

**IMPLEMENTASI *FRAMEWORK LARAVEL* PADA SISTEM INFORMASI
PERPUSTAKAAN DAERAH KABUPATEN LUWU UTARA**

SKRIPSI

“Diajukan untuk memenuhi salah satu persyaratan memperoleh gelar Sarjana Komputer
pada Program Studi Informatika”



Disusun Oleh :

IRA WIDYASTUTI

NIM : IK.22.11.022

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS MEGA BUANA PALOPO
TAHUN 2024**

HALAMAN PERSETUJUAN

SKRIPSI

**IMPLEMENTASI *FRAMEWORK LARAVEL* PADA SISTEM INFORMASI
PERPUSTAKAAN DAERAH KABUPATEN LUWU UTARA**

yang disusun dan diajukan oleh

Ira Widyastuti
IK.22.11.022

telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 8 Juni 2024

Pembimbing I



Abdul Malik, S. Kom., M.Cs
NIDN : 0910049004

Pembimbing II



Muh. Hajar Harike, S. Kom., M.Si.
NIDN : 0912069201

HALAMAN PENGESAHAN

SKRIPSI

**IMPLEMENTASI FRAMEWORK LARAVEL PADA SISTEM INFORMASI
PERPUSTAKAAN DAERAH KABUPATEN LUWU UTARA**

Telah dipersiapkan dan disusun oleh

IRA WIDYASTUTI
NIM : IK.22.011.022

Telah dipertahankan didepan Dewan Penguji
pada tanggal 8 Juni 2024

Susunan Dewan Penguji

Pembimbing I



Abdul Malik, S. Kom., M.Cs
NIDN. 0910049004

Ketua Penguji



M. Hasanuddin, S. Kom., M.Kom.
NIDN. 0927098905

Pembimbing II



Muh. Hajar Harike, S. Kom., M.Si.
NIDN. 0912069201

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar *Sarjana Komputer*
Tanggal, 8 Juni 2024

Dekan Fakultas Ilmu Komputer
Universitas Mega Buana Palopo



Abdul Malik, S. Kom., M.Cs
NIDN. 0910049004

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama : Ira Widyastuti
NIM : IK.22.11.022
Tahun terdaftar : 2022 / 2023
Program Studi : SI Informatika
Fakultas : Fakultas Ilmu Komputer
Universitas : Universitas Mega Buana Palopo

Menyatakan bahwa dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu Lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/ lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Palopo, 8 Juni 2024



IRA WIDYASTUTI
Nim. IK.22.11.022

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT, yang telah memberikan rahmat dan karunia-Nya, sehingga skripsi ini dapat diselesaikan dengan baik dan tepat waktu. Saya mempersembahkan skripsi ini kepada:

1. Tuhan Yang Maha Esa, karena dengan izin dan karunianya sehingga memudahkan saya dapat menyelesaikan skripsi ini dengan tepat waktu.
2. Keluarga tercinta, terutama Suami, Orang tua, Mertua dan Semua Saudara yang selalu memberikan doa, dukungan, dan kasih sayang tiada henti. Terima kasih atas segala pengorbanan dan dorongan yang tak pernah lelah diberikan kepada saya.
3. Dosen Pembimbing, yang dengan sabar memberikan bimbingan, ilmu, dan waktu untuk membantu saya menyelesaikan skripsi ini. Terima kasih atas arahan dan koreksi yang sangat berharga.
4. Sahabat dan Teman-Teman, yang selalu memberikan semangat, dukungan moral, dan kebersamaan yang tidak ternilai selama masa studi ini. Terima kasih atas keceriaan dan motivasinya.
5. Almamater Tercinta, yang telah menjadi tempat saya menimba ilmu dan mengembangkan diri selama ini.
6. Semua Pihak yang Telah Membantu, baik secara langsung maupun tidak langsung dalam penyelesaian skripsi ini. Terima kasih atas segala bentuk bantuan dan dukungannya.

Semoga skripsi ini dapat memberikan manfaat dan kontribusi positif bagi pembaca dan perkembangan ilmu pengetahuan.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarokatuh

Puji dan syukur penulis panjatkan kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **"Implementasi Framework Laravel pada Sistem Informasi Perpustakaan Daerah Kabupaten Luwu Utara"** ini dengan baik. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan studi jenjang strata satu (S1) pada program studi di Informatika, Fakultas Ilmu Komputer, Universitas Mega Buana Palopo.

Penulisan skripsi ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. **Ibu Prof. Dr. Hj. Nilawati Uly, S.Si, Apt, M.Kes, CIPA.**, selaku Rektor Universitas Mega Buana Palopo, atas kesempatan yang diberikan untuk menimba ilmu di universitas ini.
2. **Bapak Abdul Malik, S.Kom., M.Cs**, selaku Dekan Fakultas Ilmu Komputer, atas segala fasilitas dan dukungan yang diberikan selama penulis menempuh studi.
3. **Bapak Abdul Malik, S.Kom., M.Cs** dan **Bapak Muh. Hajar Harike, S.Kom., M.Si**, selaku dosen pembimbing skripsi, atas segala bimbingan, saran, dan masukan yang sangat berharga dalam proses penyusunan skripsi ini.
4. **Kepala Dinas dan Staf Pengelola Perpustakaan Daerah Kabupaten Luwu Utara**, atas kerjasama dan bantuan yang diberikan selama proses penelitian.
5. **Keluarga** tercinta terutama **Suami** dan **Kedua Orang Tua**, yang selalu memberikan doa, kasih sayang, serta dukungan moril dan materil yang tiada henti.
6. **Teman-teman dan semua pihak** yang telah memberikan dukungan, baik secara langsung maupun tidak langsung, dalam penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan dan keterbatasan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa yang akan datang. Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan dan teknologi, khususnya di bidang sistem informasi perpustakaan.

Wassalamu`alaikum Warahmatullahi Wabarakatuh

Palopo, 8 Juni 2024

Penulis,



Ira Widyastuti

DAFTAR ISI

| | |
|--|-----|
| HALAMAN PERSETUJUAN..... | ii |
| HALAMAN PERNYATAAN KEASLIAN SKRIPSI..... | iv |
| HALAMAN PERSEMBAHAN | v |
| KATA PENGANTAR..... | vi |
| DAFTAR ISI..... | vii |
| DAFTAR TABEL | ix |
| DAFTAR GAMBAR | x |
| DAFTAR ISTILAH | xi |
| INTI SARI | xiv |
| ABSTRACT..... | xv |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Perumusan Masalah | 2 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan Penelitian | 3 |
| 1.5 Manfaat Penelitian | 3 |
| 1.6 Metode Penelitian | 4 |
| 1.6.1 Metode Pengumpulan Data | 4 |
| 1.6.2 Metode Penelitian/Pengembangan Sistem | 4 |
| 1.7 Sistematika Penulisan..... | 6 |
| BAB II TINJAUAN PUSTAKA..... | 8 |
| 2.1 Studi Literatur | 8 |
| 2.2 Dasar Teori | 11 |
| 2.2.1 Perpustakaan | 11 |
| 2.2.2 Perpustakaan Daerah Kabupaten Luwu Utara | 11 |
| 2.2.3 Profil Kab. Luwu Utara | 12 |
| 2.2.4 Sistem Informasi | 12 |
| 2.2.5 Sistem Informasi Perpustakaan..... | 13 |
| 2.2.6 <i>Framework Laravel</i> | 13 |
| 2.2.7 Konsep MVC | 14 |
| 2.2.8 Bahasa Pemrograman | 15 |
| 2.2.9 PHP (<i>Hypertext Preprocessor</i>) | 16 |
| 2.2.10 MySQL | 17 |
| 2.2.11 XAMPP | 18 |
| 2.2.12 <i>Unified Modelling Language (UML)</i> | 18 |
| 2.2.13 <i>Flowchart</i> | 22 |
| 2.2.14 Perancangan Sistem | 24 |
| 2.2.15 Metode <i>Extreme Programming (XP)</i> | 24 |
| 2.2.16 Pengujian Sistem..... | 25 |
| 2.3 Kerangka Pikir | 28 |
| BAB III ANALISIS DAN RANCANGAN SISTEM | 29 |

| | | |
|---------------------------|--|----|
| 3.1 | Analisis Sistem..... | 29 |
| 3.1.1 | Analisis Sistem yang sedang berjalan | 29 |
| 3.1.2 | Sistem yang diusulkan..... | 30 |
| 3.2 | Deskripsi Data..... | 31 |
| 3.3 | Arsitektur Sistem..... | 32 |
| 3.4 | Rancangan Model Sistem | 34 |
| 3.4.1 | Perancangan UML (Unified Modeling Language) | 34 |
| 3.4.2 | Perancangan <i>Database</i> | 49 |
| 3.5 | Rancangan <i>User Interface</i> (Antarmuka Pengguna) | 51 |
| 3.6 | Rancangan Pengujian Sistem..... | 54 |
| 3.6.1 | Rancangan Pengujian <i>Black Box</i> | 54 |
| 3.6.2 | Rancangan Pengujian <i>White Box</i> | 55 |
| BAB IV IMPLEMENTASI | | 56 |
| 4.1 | Implementasi..... | 56 |
| 4.1.1 | Implementasi Sistem | 56 |
| 4.1.2 | Implementasi <i>Database</i> | 57 |
| 4.1.3 | Implementasi Program | 57 |
| 4.2 | Hasil Pengujian | 62 |
| 4.2.1 | Hasil Pengujian <i>Black Box</i> | 62 |
| 4.2.2 | Hasil Pengujian <i>White Box</i> | 63 |
| BAB V PENUTUP | | 73 |
| 3.1 | Kesimpulan..... | 73 |
| 3.2 | Saran..... | 73 |
| DAFTAR PUSTAKA..... | | 75 |
| LAMPIRAN..... | | 78 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 2.1 Studi Literatur | 10 |
| Tabel 2.2 <i>Simbol Use Case Diagram</i> | 19 |
| Tabel 2.3 <i>Simbol Activity Diagram</i> | 21 |
| Tabel 2.4 <i>Simbol Flowchart</i> | 23 |
| Tabel 2.5 Hubungan <i>Cyclomatic complexity</i> | 27 |
| Tabel 3.1 Skenario <i>Use Case</i> Mengelola Data Buku..... | 36 |
| Tabel 3.2 Skenario <i>Use Case</i> Mengelola Data Anggota | 37 |
| Tabel 3.3 Skenario <i>Use Case</i> Mengelola Data User..... | 38 |
| Tabel 3.4 Skenario <i>Use Case</i> Mengelola Data Transaksi..... | 39 |
| Tabel 3.5 Skenario <i>Use Case</i> Mengelola Data Laporan Transaksi | 40 |
| Tabel 3.6 Skenario <i>Use Case</i> Mengelola Data Laporan Buku | 41 |
| Tabel 3.7 Skenario <i>Use Case</i> Melihat data buku..... | 41 |
| Tabel 3.8 Struktur Tabel Buku | 50 |
| Tabel 3.9 Sturuktur Tabel <i>User</i> | 50 |
| Tabel 3.10 Sturuktur Tabel Transaksi..... | 51 |
| Tabel 3.11 Tabel Perancangan <i>Black Box Testing</i> | 54 |
| Tabel 4.1 Pengujian <i>Black Box</i> | 62 |
| Tabel 4.2 <i>Cyclomatic Complexity Route Form Login</i> | 64 |
| Tabel 4.3 <i>Graph Matrix Route Form Login</i> | 64 |
| Tabel 4.4 Pengujian Jalur <i>Login</i> | 65 |
| Tabel 4.5 <i>Cyclomatic Complexity Route Form Login</i> | 66 |
| Tabel 4.6 <i>Graph Matrix Route Form</i> tambah data buku | 66 |
| Tabel 4.7 Pengujian Jalur tambah data buku | 67 |
| Tabel 4.8 <i>Cyclomatic Complexity Route Form</i> tambah anggota | 68 |
| Tabel 4.9 <i>Graph Matrix Route Form</i> tambah data anggota..... | 69 |
| Tabel 4.10 Pengujian Jalur tambah data anggota | 69 |
| Tabel 4.11 <i>Cyclomatic Complexity Route Form</i> tambah transaksi..... | 71 |
| Tabel 4.12 <i>Graph Matrix Route Form</i> tambah transaksi..... | 71 |
| Tabel 4. 13 Pengujian Jalur tambah data transaksi..... | 72 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 1.1 <i>Alur Extreme Programming</i> | 6 |
| Gambar 2.1 <i>Konsep model MVC</i> | 15 |
| Gambar 2.2 <i>Skema definisi bahasa pemrograman</i> | 16 |
| Gambar 2.3 <i>Contoh skrip PHP sederhana</i> | 17 |
| Gambar 2.4 <i>Tampilan XAMPP control panel</i> | 18 |
| Gambar 2.5 <i>Struktur Basic Control Flow Graph</i> | 26 |
| Gambar 2.6 <i>Kerangka Pikir</i> | 28 |
| Gambar 3.1 <i>Analisis Sistem yang sedang berjalan</i> | 29 |
| Gambar 3.2 <i>Flowchart Peminjaman Buku yang akan diusulkan</i> | 30 |
| Gambar 3.3 <i>Flowchart Pengembalian Buku yang akan diusulkan</i> | 31 |
| Gambar 3.4 <i>Diagram Arsitekur Sistem</i> | 33 |
| Gambar 3.5 <i>Use Case Diagram</i> | 35 |
| Gambar 3.6 <i>Class Diagram</i> | 42 |
| Gambar 3.7 <i>Activity diagram mengelola data buku</i> | 43 |
| Gambar 3.8 <i>Activity diagram mengelola data anggota</i> | 44 |
| Gambar 3.9 <i>Activity diagram mengelola data user</i> | 45 |
| Gambar 3.10 <i>Activity diagram mengelola data transaksi</i> | 46 |
| Gambar 3.11 <i>Activity diagram mengelola laporan transaksi</i> | 47 |
| Gambar 3.12 <i>Activity diagram mengelola laporan buku</i> | 48 |
| Gambar 3.13 <i>Activity diagram melihat data buku</i> | 49 |
| Gambar 3.14 <i>Halaman Login</i> | 51 |
| Gambar 3.15 <i>Halaman Dashboard</i> | 52 |
| Gambar 3.16 <i>Halaman Data Buku</i> | 52 |
| Gambar 3.17 <i>Halaman Data Anggota</i> | 53 |
| Gambar 3.18 <i>Halaman Data Transaksi</i> | 53 |
| Gambar 3.19 <i>Halaman Laporan</i> | 54 |
| Gambar 4.1 <i>Database</i> | 57 |
| Gambar 4.2 <i>Tampilan Halaman Login</i> | 58 |
| Gambar 4.3 <i>Tampilan Halaman Dashboard</i> | 59 |
| Gambar 4.4 <i>Tampilan Halaman Buku</i> | 59 |
| Gambar 4.5 <i>Tampilan Halaman Anggota</i> | 60 |
| Gambar 4.6 <i>Tampilan Halaman Transaksi</i> | 61 |
| Gambar 4.7 <i>Tampilan Halaman Riwayat</i> | 61 |
| Gambar 4.8 <i>Tampilan Halaman Laporan</i> | 62 |
| Gambar 4.9 <i>Flow graph login</i> | 64 |
| Gambar 4.10 <i>Flow graph tambah buku</i> | 66 |
| Gambar 4.11 <i>Flow graph tambah anggota</i> | 68 |
| Gambar 4.12 <i>Flow graph tambah transaksi</i> | 70 |

DAFTAR ISTILAH

API (Application Programming Interface)

Antarmuka yang memungkinkan berbagai aplikasi atau komponen perangkat lunak untuk berkomunikasi satu sama lain.

Authentication

Proses verifikasi identitas pengguna sebelum memberikan akses ke sistem.

Authorization

Proses menentukan hak akses atau izin pengguna untuk melakukan tindakan tertentu dalam sistem.

Artisan

Interface command line dalam Laravel yang menyediakan berbagai perintah untuk membantu pengembangan aplikasi.

Backend

Bagian dari aplikasi yang berfungsi di belakang layar dan menangani logika, basis data, dan integrasi sistem.

Blade

Mesin template bawaan Laravel yang digunakan untuk merancang tampilan halaman web.

Composer

Manajer ketergantungan untuk PHP yang digunakan untuk mengelola paket dan pustaka yang dibutuhkan oleh aplikasi.

CRUD (Create, Read, Update, Delete)

Singkatan dari Create, Read, Update, Delete; operasi dasar yang dilakukan dalam pengelolaan data pada aplikasi.

Database

Kumpulan data yang terorganisir yang disimpan dan diakses secara elektronik dari suatu sistem komputer.

Deployment

Proses memindahkan aplikasi dari lingkungan pengembangan ke lingkungan produksi sehingga dapat digunakan oleh pengguna akhir.

Eloquent

ORM bawaan Laravel yang menyediakan antarmuka aktif untuk berinteraksi dengan database.

Framework Laravel

Sebuah kerangka kerja PHP yang digunakan untuk membangun aplikasi web yang elegan dan memiliki sintaksis ekspresif.

Frontend

Bagian dari aplikasi yang berinteraksi langsung dengan pengguna, mencakup desain antarmuka dan pengalaman pengguna.

Middleware

Komponen yang berada di antara aplikasi dan server yang menangani request dan response, serta berbagai tugas lainnya seperti autentikasi.

Migration

Fitur dalam Laravel yang digunakan untuk membuat, mengubah, dan mengelola tabel dalam database secara versi.

MVC (Model-View-Controller)

Sebuah pola arsitektur perangkat lunak yang memisahkan aplikasi menjadi tiga komponen utama: Model (data), View (antarmuka pengguna), dan Controller (logika aplikasi).

ORM (Object-Relational Mapping)

Teknik pemrograman yang digunakan untuk mengkonversi data antara sistem tipe objek dan sistem tipe relasional seperti database.

Perpustakaan Daerah

Institusi yang menyediakan layanan penyimpanan, pengelolaan, dan peminjaman buku serta sumber daya informasi lainnya kepada masyarakat di wilayah tertentu.

Routing

Proses menentukan jalur request dalam aplikasi web sehingga permintaan pengguna diarahkan ke bagian aplikasi yang tepat.

Seeder

Fitur dalam Laravel yang digunakan untuk mengisi database dengan data awal atau data uji.

SSL (Secure Socket Layer)

Protokol keamanan yang digunakan untuk mengenkripsi data yang dikirim antara server dan pengguna untuk menjaga privasi dan integritas data.

User Experience (UX)

Pengalaman pengguna; keseluruhan pengalaman dan kepuasan pengguna saat berinteraksi dengan produk atau sistem.

User Interface (UI)

Antarmuka pengguna; komponen visual yang memungkinkan pengguna berinteraksi dengan perangkat lunak atau aplikasi.

INTI SARI

Sistem informasi perpustakaan yang efektif dan efisien diperlukan untuk meningkatkan kualitas layanan, mempermudah akses informasi, dan mengoptimalkan manajemen data perpustakaan. *Laravel* dipilih sebagai *framework* karena keunggulannya dalam hal keamanan, kinerja, dan kemudahan pengembangan aplikasi *web*.

Metodologi yang digunakan dalam penelitian ini meliputi tahap analisis kebutuhan, perancangan sistem, pengembangan, pengujian, dan implementasi. Pada tahap analisis kebutuhan, dilakukan wawancara dan observasi untuk memahami proses bisnis dan kebutuhan pengguna. Tahap perancangan sistem melibatkan pembuatan diagram alir data dan perancangan antarmuka pengguna. Pengembangan sistem dilakukan dengan mengimplementasikan fungsionalitas utama seperti pengelolaan buku, anggota, peminjaman, dan pengembalian. Pengujian sistem dilakukan untuk memastikan bahwa sistem berjalan sesuai dengan spesifikasi yang telah ditetapkan. Implementasi melibatkan pelatihan pengguna dan migrasi data dari sistem lama ke sistem baru.

Hasil penelitian menunjukkan bahwa sistem informasi perpustakaan berbasis *Laravel* mampu meningkatkan efisiensi dan akurasi pengelolaan data perpustakaan. Sistem ini juga memberikan kemudahan akses informasi bagi pengguna, baik petugas perpustakaan maupun anggota perpustakaan. Pengguna merespon positif terhadap sistem baru ini karena antarmuka yang *user-friendly* dan fitur-fitur yang lengkap.

Implementasi *framework Laravel* pada sistem informasi perpustakaan daerah Kabupaten Luwu Utara berhasil meningkatkan kualitas layanan perpustakaan. Penelitian ini diharapkan dapat menjadi *referensi* bagi pengembangan sistem informasi perpustakaan di daerah lain.

Kata kunci: Sistem informasi, Perpustakaan, *Laravel*, Metode *Extreme Programming*.

ABSTRACT

An effective and efficient library information system is needed to improve service quality, facilitate information access, and optimize library data management. Laravel was chosen as the framework due to its advantages in security, performance, and ease of web application development.

The methodology used in this study includes requirement analysis, system design, development, testing, and implementation. In the requirement analysis phase, interviews and observations were conducted to understand the business processes and user needs. The system design phase involved creating data flow diagrams and user interface designs. System development was carried out by implementing key functionalities such as book management, member management, borrowing, and returning. System testing was conducted to ensure the system operates according to the specified requirements. Implementation involved user training and data migration from the old system to the new system.

The results of the study indicate that the Laravel-based library information system can improve the efficiency and accuracy of library data management. The system also provides easy access to information for both library staff and members. Users responded positively to the new system due to its user-friendly interface and comprehensive features.

The implementation of the Laravel framework in the regional library information system of North Luwu Regency has successfully enhanced the quality of library services. This study is expected to serve as a reference for the development of library information systems in other regions.

Key Word : Information system, Library, Laravel, Extreme Programming Method.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Efektifitas pengelolaan informasi dan data menjadi krusial, terutama dalam konteks pelayanan perpustakaan. Perpustakaan merupakan salah satu tempat yang digunakan untuk menyimpan, mengelola, membaca dan mencari informasi dari bahan pustakan seperti buku, jurnal, referensi, skripsi dan bahan pustaka lainnya (Nur, 2020). Efektifitas pengelolaan data dan informasi Perpustakaan merupakan kebutuhan yang sangat penting, karena dengan efektifitas pengelolaan data dan informasi tentu akan meningkatkan proses layanan administrasi Perpustakaan. Efektifitas layanan dan data informasi digital dapat ditingkatkan dengan adanya sistem informasi perpustakaan berbasis web.(A'yuniyah et al., 2022)

Kabupaten Luwu Utara, sebagai salah satu wilayah di Indonesia, juga memerlukan sistem informasi perpustakaan yang memadai untuk meningkatkan pelayanan perpustakaan kepada masyarakatnya. Perpustakaan Daerah Kabupaten Luwu Utara, sebagai pusat informasi dan pengetahuan, memiliki peran penting dalam mendukung proses pembelajaran masyarakat serta meningkatkan tingkat literasi. Namun, tantangan timbul dari sistem informasi yang saat ini digunakan, yang masih belum optimal, kurang fleksibel, dan seringkali tidak stabil dalam mengelola data yang ada.

Layanan Perpustakaan Daerah Kab. Luwu Utara saat ini masih dilakukan secara manual, yang mengakibatkan proses peminjaman dan pengembalian buku menjadi lambat dan rawan kesalahan. Sistem penyimpanan data yang belum terintegrasi dengan baik juga menyebabkan kesulitan dalam memperbaharui dan mengakses informasi tentang ketersediaan buku secara real-time, sehingga pengguna sering kali merasa frustrasi dan tidak terlayani dengan baik.

Dalam rangka meningkatkan kualitas layanan dan efisiensi operasional, diperlukan pengembangan sistem informasi perpustakaan yang mampu mengatasi permasalahan tersebut. *Framework Laravel* merupakan sebuah kerangka kerja PHP (*Hypertext Preprocessor*), menonjolkan kesederhanaan dan fleksibilitas dalam desainnya. (Naja et al., 2023). Laravel memiliki banyak fitur modern yang

sangat membantu developer dalam membuat aplikasi, serta yang dikenal dengan arsitektur yang elegan dan fitur yang modern, menawarkan banyak keunggulan seperti keamanan yang lebih baik, kemudahan dalam pengelolaan database, serta dukungan komunitas yang luas. Penerapan *Framework Laravel* dalam mengembangkan sistem informasi perpustakaan diharapkan dapat meningkatkan efisiensi dan kualitas layanan kepada pengguna.

Selain itu, *Framework Laravel* juga mendukung pengembangan yang lebih cepat dan efisien dengan menyediakan berbagai *tools* dan *package* yang dapat dengan mudah diintegrasikan. Dengan mengadopsi *Framework* ini, diharapkan sistem informasi perpustakaan akan lebih mudah dalam skalabilitas, pemeliharaan, dan juga peningkatan di masa mendatang. Penggunaan *Laravel* dapat meningkatkan interaksi pengguna melalui antarmuka yang ramah pengguna dan responsif, sehingga mampu meningkatkan kepuasan pengunjung perpustakaan dan memperluas akses terhadap sumber-sumber informasi.

Dengan mempertimbangkan potensi perbaikan yang signifikan melalui penerapan teknologi dan metodologi yang tepat, penelitian ini bertujuan untuk mengimplementasikan dan menilai efektivitas penggunaan *framework laravel*. Penelitian ini diharapkan tidak hanya memberikan solusi teknis yang efektif tetapi juga memberikan kontribusi pada literatur akademis tentang pengembangan sistem informasi yang adaptif dan responsif terhadap kebutuhan pengguna.

Dengan memperhatikan latar belakang diatas, maka penulis ingin melakukan penelitian dengan judul **“Implementasi *Framework Laravel* Pada Sistem Informasi Perpustakaan Daerah Kab. Luwu Utara”**.

1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, maka penulis merumuskan permasalahan yaitu:

- 1) Bagaimana cara mengimplementasikan *framework Laravel* untuk mengembangkan sistem informasi perpustakaan berbasis web yang sesuai dengan kebutuhan Perpustakaan Daerah Kabupaten Luwu Utara?
- 2) Bagaimana cara mengembangkan sistem informasi perpustakaan daerah Kabupaten Luwu Utara yang mampu melakukan klasifikasi buku secara efektif

dan terintegrasi dengan kategori buku yang sudah ada? Bagaimana pengembangan sistem informasi perpustakaan Kabupaten Luwu Utara?

- 3) Seberapa efektif pengembangan sistem informasi perpustakaan daerah Kabupaten Luwu Utara menggunakan *framework Laravel* dalam meningkatkan manajemen dan pelayanan perpustakaan?

1.3 Batasan Masalah

- 1) Penelitian ini menggunakan implementasi *framework laravel* dalam pengembangan sistem informasi perpustakaan
- 2) Sistem informasi yang dikembangkan berlaku untuk Perpustakaan Daerah Kab. Luwu Utara dan tidak mempertimbangkan variasi jenis perpustakaan lainnya.
- 3) Penelitian ini membatasi pada dasar sistem informasi perpustakaan, seperti manajemen koleksi, pencarian buku, dan manajemen peminjaman.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

- 1) Mengimplementasikan *framework laravel* untuk mengembangkan sistem informasi Perpustakaan Daerah Kabupaten Luwu Utara berbasis *web*.
- 2) Mengembangkan sistem informasi Perpustakaan Daerah Kabupaten Luwu Utara dengan model klasifikasi buku terintegrasi dengan kategori buku.
- 3) Mengukur tingkat efektifitas pengembangan sistem informasi Perpustakaan Daerah Kabupaten Luwu Utara menggunakan *framework Laravel*.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

- 1) Untuk perpustakaan: Implementasi *Framework Laravel* pada sistem informasi perpustakaan daerah kabupaten Luwu Utara dapat meningkatkan efisiensi operasional, peningkatan layanan perpustakaan dan manajemen koleksi yang lebih efektif.

- 2) Untuk pengguna perpustakaan: Memudahkan pencarian dan akses terhadap koleksi buku, serta meningkatkan kepuasan pengguna melalui layanan yang lebih cepat dan akurat.
- 3) Untuk peneliti dan pengembang sistem informasi: Penelitian ini menambahkan pengetahuan baru pada bidang teknologi informasi dan manajemen perpustakaan serta dan sebagai dasar untuk riset lebih lanjut dalam pengembangan sistem informasi dengan laravel.

1.6 Metode Penelitian

1.6.1 Metode Pengumpulan Data

Metode Pengumpulan data yang digunakan dalam penelitian ini yaitu:

1. Metode Observasi

Observasi yang dilakukan penulis yaitu melakukan pengamatan langsung di Perustakaan Daerah Kab. Luwu Utara terhadap kegiatan peminjaman dan pengembalian buku.

2. Wawancara

Untuk mendapatkan informasi yang lebih lengkap sehingga penulis melakukan proses pengumpulan data yang dilakukan dengan cara Tanya jawab pada staf bidang pengolahan, layanan dan pelestarian bahan perpustakaan di Perpustakaan daerah Kab. Luwu Utara.

3. Studi Pustaka

Penulis mengumpulkan data terkait penelitian melalui pencairan informasi dari buku, jurnal, dan literatur lainnya. Sehingga bertujuan untuk membentuk landasan teori yang kokoh.

1.6.2 Metode Penelitian/Pengembangan Sistem

Metode pengembangan sistem untuk mengimplementasikan Framework Laravel mengacu pada Metode Pengembangan XP (*Extreme Programming*). Metode ini merupakan metodologi yang relative lebih mudah dibandingkan dengan metode lainnya. *Extreme Programming* (XP) merupakan pendekatan pengembangan perangkat lunak atau manajemen proyek yang bersifat agile, dengan tujuan menciptakan perangkat lunak yang lebih unggul dalam hal

kualitas. XP menggunakan siklus pengembangan yang pendek, sehingga mampu menanggapi secara cepat terhadap perubahan keinginan pengguna yang berubah-ubah (*Extreme Programming: 5 Tahapan, Kelebihan, Dan Kekurangan*, n.d.). Metode XP tidak hanya memfokuskan pada *coding* tetapi juga ke seluruh bagian area pengembangan perangkat lunak. Nilai utama yang mendasari dari metode ini adalah *communications* (komunikasi), *courage* (Keberanian), *simplicit* (Kesederhanaan), dan *feedback* (Umpan balik) (*Extreme Programming (XP) - Definisi, Nilai, Tahapan - Serupa.Id*, n.d.)

Tahapan metode *Extreme Programming* (XP) yaitu:

1. *Planning*

Pada tahap perencanaan, penulis mengumpulkan data-data yang dibutuhkan dalam membuat *website*. Dalam mengumpulkan data maka penulis melakukan observasi dan wawancara dengan staf Perpustakaan Daerah Kab. Luwu Utara.

Setelah data terkumpul barulah penulis mendefinisikan *output* yang akan dihasilkan, fitur yang dimiliki oleh aplikasi dan fungsi dari aplikasi yang dikembangkan.

2. *Design*

Pada tahap ini penulis mulai melakukan perancangan atau membuat *design* berdasarkan konsep dan data yang telah dikumpulkan pada tahap *Planning*. Penulis membuat alur sistem dalam *usecase diagram*, *class diagram*, dan *activity diagram*.

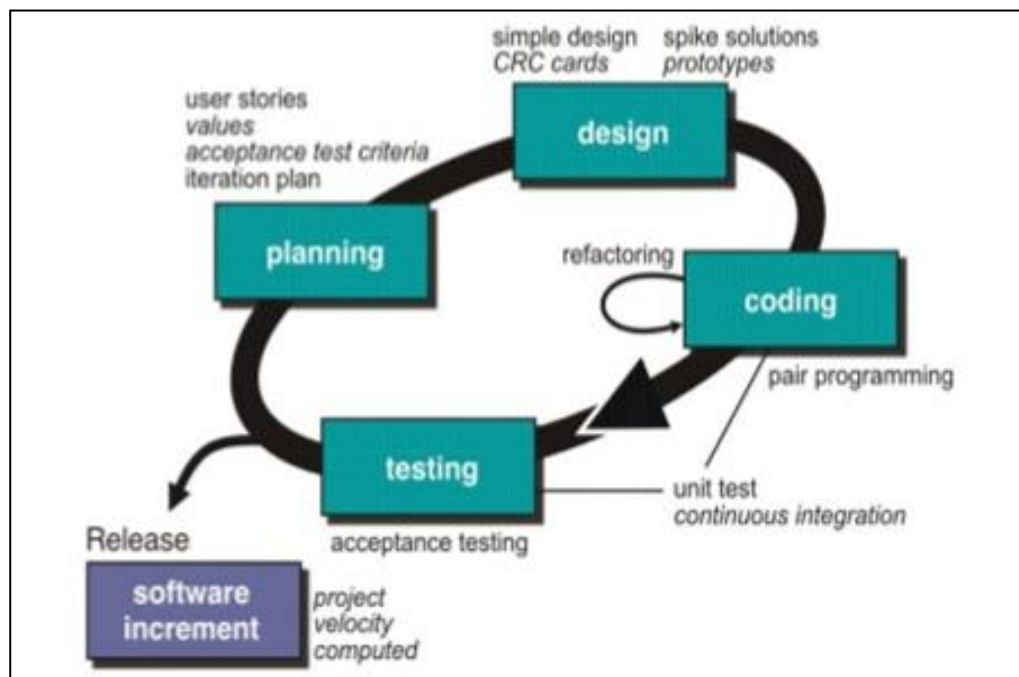
3. *Coding*

Pada tahap ini penulis mulai melakukan pengkodean berdasarkan *design* yang sudah ditentukan. Pada tahap ini penulis menggunakan *Framework Laravel*. Proses pengkodean ini menggunakan paradigma pemrograman berorientasi objek dan berbagai bahasa pemrograman *web* seperti *HTML*, *CSS*, *PHP*, dan *Javascript*.

4. *Testing*

Pada tahap ini penulis lebih fokus pada pengujian fitur dan fungsionalitas dari aplikasi. Metode pengujian yang digunakan pada tahap ini adalah metode *whitebox* dan *blackbox testing*, di mana pengujian dilakukan

terhadap *input* dari beberapa formulir untuk memeriksa apakah fungsi-fungsinya berjalan sesuai dengan yang diharapkan.



Gambar 1.1 Alur Extreme Programming(Ariyanti et al., 2020)

1.7 Sistematika Penulisan

Berdasarkan pedoman penulisan, penyusunan sistematika penulisan dapat dipilih menjadi 5 (Lima) Bab, sebagai berikut:

BAB I PENDAHULUAN

Pada bagian ini penelitian ini meliputi informasi secara umum tentang latar belakang masalah yang dihadapi oleh Perpustakaan Daerah Kab. Luwu Utara ditempat penelitian dan mengapa harus dipecahkan.

BAB II TINJAUAN PUSTAKA

Pada bab ini membandingkan penelitian-penelitian sebelumnya yang terkait dengan penelitian, menjelaskan secara umum tempat penelitian skripsi dilaksanakan dan menjelaskan semua materi yang digunakan dalam menyusun skripsi

BAB III ANALISIS DAN RANCANGAN SISTEM

Pada bab ini membahas secara bertahap tentang perencanaan dan analisis data saat pembuatan aplikasi sistem informasi Perpustakaan Daerah Kab. Luwu Utara menggunakan *framework Laravel*.

BAB IV IMPLEMENTASI

Pada bab ini diterangkan cara implementasi sistem yang digunakan untuk mengaplikasikan perancangan, dan uji coba sistem yang dilakukan oleh petugas perpustakaan untuk mengetahui sejauh mana sistem tersebut berjalan.

BAB V KESIMPULAN DAN SARAN

Pada bab ini menyajikan kesimpulan secara keseluruhan dari penelitian yang dilakukan dan juga terdapat beberapa saran yang diajukan berkaitan dengan penelitian skripsi ini.

DAFTAR PUSTAKA

Pada bagian ini dipaparkan tentang sumber-sumber *literatur* yang digunakan dalam penyusunan penelitian Skripsi ini.

LAMPIRAN LAMPIRAN

Pada bagian ini berisi lembar lempiran-lampiran yang dibutuhkan dalam penelitian Skripsi ini.

BAB II

TINJAUAN PUSTAKA

2.1 Studi Literatur

Penelitian tentang Sistem informasi perpustakaan berbasis *web* menggunakan *framework Laravel*, berdasarkan tinjauan pustaka yang dilakukan peneliti, telah aktif dilakukan sejak tahun 2019 sampai saat ini. pada bagian tinjauan pustaka ini diuraikan penelitian-penelitian tentang sistem informasi Perpustakaan berbasis web menggunakan *Framework Laravel* yang telah dilakukan oleh para peneliti terdahulu.

Leo et al., 2021, melakukan penelitian tentang Penembangan sistem informasi perpustakaan SMA Muhammadiyah 1 Bandar Sribahawono Kab. Lampung Timur. Dalam penelitian ini menggunakan *Rapid Application Development (RAD)* sebagai metode dalam pengembangan sistem. Selanjutnya, digunakan visualisasi pemodelan *UML (Unified Modelling Language)*, yang merupakan standar bahasa pemodelan untuk pengembangan perangkat lunak berbasis objek. Diharapkan dapat mempermudah staf perpustakaan dalam mengelola data transaksi seperti peminjaman, pengembalian, persediaan buku, pencarian buku, dan pembuatan laporan-laporan. (Leo et al., 2021).

Nofiati dan Daru, 2021, melakukan penelitian tentang sistem informasi Perpustakaan berbasis *Web* dengan *Framework Laravel* di Perpustakaan SMP Muhammadiyah 7 Semarang, pengumpulan data dilakukan melalui observasi, wawancara, dan penelitian literatur. Metode pengembangannya menggunakan *Extreme Programming (XP)*, dan program dibuat menggunakan *editor* kode *Visual Studio Code* dan *XAMPP*. Tujuan dari pembuatan sistem perpustakaan ini adalah untuk menghasilkan program yang dapat membantu sekolah dalam pengelolaan data di perpustakaan SMP Muhammadiyah 7 Semarang (Nofiati & Daru, 2021).

Maghfiroh dan Hidayat, 2021 melakukan penelitian tentang sistem informasi perpustakaan di Pondok Pesantren Luhur Wahid Hasyim Putri Semarang yang dapat mengelola data buku, transaksi, dan laporan. Metode pengembangan sistem yang digunakan adalah model *waterfall*, yang melibatkan tahapan analisis, desain,

pengkodean, dan pengujian. Perbedaan utama dengan sistem informasi perpustakaan lain adalah adanya fitur notifikasi pesan kepada peminjam mengenai waktu pengembalian yang sudah lewat menggunakan aplikasi *WhatsApp* yang dioperasikan oleh petugas perpustakaan (Maghfiroh & Hidayat, 2021).

A'yuniyah et al., 2022, melakukan penelitian tentang pengembangan sistem informasi inventaris perpustakaan berbasis *web* menggunakan *metode waterfall*, serta menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*) dan *MySQL (My Structure Query Language)*. Hasil pembahasan yang telah dijelaskan dalam latar belakang menyoroti masih banyaknya perpustakaan yang mengelola data secara manual. Penelitian ini bertujuan untuk menggantikan sistem manual tersebut dengan sistem komputerisasi. Metode pengembangan sistem yang digunakan adalah metode *Waterfall*. Tahapan metode *waterfall* dimulai dari perencanaan, analisis, pengumpulan data, desain, pengkodean, hingga implementasi, sehingga menghasilkan sebuah sistem inventaris perpustakaan berbasis *web* (A'yuniyah et al., 2022).

Naja et al., 2023, dalam penelitiannya tentang sistem informasi perpustakaan di SMA Negeri 1 Pecangaan Jepara. Pelayanan yang optimal akan memberikan kemudahan bagi pengunjung dan petugas perpustakaan. Pengujian aplikasi ini menggunakan metode *black box* dan berhasil dilakukan dengan baik pada setiap bagian menu. Implementasi aplikasi perpustakaan pada Perpustakaan SMA Negeri 1 Pecangaan Jepara menggunakan *Framework Laravel* memiliki beberapa fitur antara lain menu pengembalian dan peminjaman buku, data anggota, data buku, serta data pengunjung dengan menggunakan metode *waterfall* sebagai metode pengembangan sistemnya (Naja et al., 2023).

Pengembangan sistem dilakukan melalui pendekatan analisis berorientasi objek menggunakan *Unified Modeling Language (UML)*, serta menggunakan bahasa pemrograman *PHP (Hypertext Preprocessor)* dan basis data *MySQL (My Structure Query Language)*. Dinas Perpustakaan dan Arsip Daerah (DPAD) Kota Tangerang menerapkan sistem berbasis *web*. Metode pengembangan sistem menggunakan *extreme Programming (XP)* dengan tahapannya yaitu *Planning, Design, Coding, dan Testing*. Metode pengujian yang dilakukan *black box testing* yang berfokus pada fungsionalitas eksternal suatu program (Adensa et al., 2024).

Tabel 2.1 Studi Literatur

| No | Referensi | Metode | Sumber data | Keterangan | Perbedaan Penelitian |
|----|-----------------------------|--|--|--|---|
| 1 | (Leo et al., 2021) | <i>Rapid Application Development</i> (RAD) | Perpustakaan SMA Muhammadiyah 1 Bandar Sribhawono Kab. Lampung Timur | Pengelolaan proses peminjaman dan pengembalian buku menggunakan sistem <i>Rapid Application Development</i> serta sistem informasi dikembangkan menggunakan <i>framework laravel</i> . | Metode Pengembangan sistem berbeda |
| 2 | (Nofiati & Daru, 2021) | <i>Extreme Programming</i> (XP) | Perpustakaan SMP Muhammadiyah 7 Semarang | Sistem Informasi Perpustakaan menggunakan metode pengembangannya <i>Extreme Programming</i> (XP) dan pembuatan program menggunakan <i>code editor Visual Studio Code</i> , XAMPP dengan <i>Framework Laravel</i> | Model sistem dan model data yang teliti berbeda |
| 3 | (Maghfiroh & Hidayat, 2021) | <i>Waterfall</i> | Pondok Pesantren Luhur Wahid Hasyim Semarang | Metode pengembangan sistem menggunakan <i>waterfall</i> , hasil penelitiannya itu sebuah sistem informasi perpustakaan dalam mengelola data buku, data transaksi dan data laporan. | Metode pengembangan sistem berbeda |
| 4 | (A'yuniyah et al., 2022) | <i>Waterfall</i> | Perpustakaan Sekolah | Sistem ini dirancang menggunakan bahasa pemrograman PHP (<i>Hypertext Preprocessor</i>) dan MySQL dengan metode <i>waterfall</i> | Metode Pengembangan sistem berbeda |
| 5 | (Naja et al., 2023) | <i>Waterfall</i> | SMA Negeri 1 Pecangaan Jepara | Sistem informasi menggunakan <i>framework laravel</i> dengan metode <i>waterfall</i> dan pengujian menggunakan <i>black box</i> . | Metode Pengembangan Sistem berbeda |
| 6 | (Adensa et al., 2024) | <i>Extreme Programming</i> (XP) | Dinas Perpustakaan dan Kearsipan Kota Tanggerang | Pengembangan sistem ini adalah merancang <i>website</i> Dinas Perpustakaan dan Kearsipan Daerah menggunakan metode <i>Extreme Programming</i> (XP) | Model data yang diteliti berbeda |

2.2 Dasar Teori

2.2.1 Perpustakaan

Perpustakaan berasal dari istilah "pustaka", yang mengacu pada buku. Perpustakaan dapat berupa ruang, bagian dari bangunan, atau bahkan bangunan itu sendiri yang digunakan untuk menyimpan buku dan materi lainnya dengan susunan tertentu untuk penggunaan pembaca, bukan untuk tujuan komersial (Sutarno NS, 2006).

Menurut Undang-undang Nomor 43 Tahun 2007 tentang Perpustakaan menyebutkan bahwa Perpustakaan adalah institusi pengelola koleksi karya tulis, karya cetak, dan/atau karya rekam secara profesional dengan sistem yang baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka.

Menurut Undang-undang Nomor 43 Tahun 2007 ada berbagai jenis perpustakaan yaitu :

1. Perpustakaan Nasional merupakan Lembaga Pemerintah Non Departemen (LPND) yang melaksanakan tugas pemerintahan dalam bidang perpustakaan dan berkedudukan di ibukota Negara Indonesia.
2. Perpustakaan Umum adalah perpustakaan yang diselenggarakan oleh Pemerintah, yaitu pemerintah Provinsi, Kabupaten/Kota, Kecamatan, dan desa, serta dapat diselenggarakan oleh masyarakat.
3. Perpustakaan Sekolah/Madrasah yaitu perpustakaan yang berada di sekolah atau madrasah serta dikelola oleh sekolah/madrasah.
4. Perpustakaan Perguruan Tinggi yaitu perpustakaan yang berada di perguruan tinggi baik Universitas, Akademika maupun Sekolah Tinggi atau Institut.
5. Perpustakaan Khusus yaitu perpustakaan yang menyediakan bahan pustaka sesuai dengan kebutuhan pemustaka di lingkungannya.

2.2.2 Perpustakaan Daerah Kabupaten Luwu Utara

Berdasarkan Peraturan Bupati Luwu Utara No. 69 Tahun 2016 tentang Kedudukan, susunan organisasi, tugas, fungsi dan Uraian tugas serta tata kerja

Dinas Perpustakaan dan Kearsipan Daerah Kab. Luwu Utara menyediakan layanan berupa layanan Perpustakaan serta layanan penataan dan pendataan Arsip. Layanan perpustakaan adalah kegiatan yang memberikan kemudahan bagi para pemustaka untuk memanfaatkan atau mendayagunakan bahan pustaka yang tersedia di Perpustakaan. Jenis-jenis layanan yang disediakan di Dinas Perpustakaan dan Kearsipan Daerah yaitu layanan sirkulasi, layanan perpustakaan keliling (Puskel), Layanan Baca ditempat, dan Layanan internet (*Hotspot*).

Dalam Undang-undang No. 43 tahun 2007 BAB V Pasal 14 ayat (1) dan (3) yaitu :

- a) Ayat (1) yang menyatakan bahwa : “Layanan perpustakaan dilakukan secara prima, dan berorientasi bagi kepentingan pemustaka”
- b) Ayat (3) yang menyatakan bahwa : “Setiap perpustakaan mengembangkan layanan perpustakaan sesuai dengan kemajuan teknologi informasi dan komunikasi”.

2.2.3 Profil Kab. Luwu Utara

Kabupaten Luwu Utara, sebuah wilayah administratif di Provinsi Sulawesi Selatan, Indonesia, memiliki pusat pemerintahan di Masamba. Terbentuk berdasarkan Undang-Undang Nomor 19 Tahun 1999, Kabupaten ini merupakan hasil pemekaran dari Kabupaten Luwu. Luwu Utara terletak antara koordinat 2°30'45"–2°37'30" LS dan 119°41'15"–121°43'11" BT. Secara geografis, wilayah ini berbatasan dengan Provinsi Sulawesi Tengah di utara, Kabupaten Luwu Timur di timur, Kabupaten Luwu di selatan, dan Kabupaten Mamuju di barat. Jumlah penduduk Kabupaten Luwu Utara tahun 2023 sejumlah 330.576 jiwa (BPS Sulawesi Selatan, 2022)

2.2.4 Sistem Informasi

Sistem merujuk pada serangkaian prosedur yang saling terkait dan diorganisir secara bersama-sama untuk melaksanakan suatu tindakan atau mencapai tujuan tertentu. (Nurhayati et al., 2023). Sebuah sistem merupakan gabungan elemen fisik, prosedur, perangkat lunak, tenaga kerja manusia, dan

sumber daya lainnya yang berinteraksi bersama untuk mencapai tujuan yang diinginkan.

Informasi adalah hasil pengolahan data sehingga memiliki nilai dan kepentingan yang bermanfaat bagi penerima, juga dapat mengurangi ketidakpastian dalam pengambilan keputusan terkait situasi tertentu. (Adensa et al., 2024). Informasi biasa dalam bentuk teks, gambar, audio, atau gabungan berbagai jenis media lainnya.

Sistem informasi adalah serangkaian komponen yang terhubung satu sama lain dan beroperasi untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi guna mendukung pengambilan keputusan, analisis, serta pemetaan masalah kompleks dalam suatu organisasi. (Nurhayati et al., 2023). Sistem informasi terdiri dari perangkat keras, perangkat lunak, basis data, jaringan komunikasi, dan juga melibatkan orang-orang yang terlibat dalam pembuatan, pengelolaan, dan penggunaan sistem tersebut.

2.2.5 Sistem Informasi Perpustakaan

Sistem informasi Perpustakaan adalah sistem yang dibuat untuk memudahkan petugas perpustakaan dalam mengelola suatu perpustakaan. Sistem informasi perpustakaan adalah sistem yang mencakup elemen-elemen manusia, perangkat keras, perangkat lunak, prosedur, dan data yang saling terintegrasi. Sistem ini digunakan untuk mengelola secara otomatis perpustakaan, sehingga menghasilkan informasi yang berharga bagi pengguna, baik pustakawan maupun pengunjung perpustakaan. (Listiyono et al., 2022). Sistem informasi perpustakaan merupakan *platform* atau infrastruktur yang spesifik dibangun untuk membantu dalam menjalankan operasional dan mengelola perpustakaan.

2.2.6 Framework Laravel

Framework merupakan gabungan kode pustaka dan alat yang dirangkai secara terstruktur untuk membentuk suatu kerangka kerja. Tujuannya adalah untuk menyederhanakan dan mempercepat proses pengembangan aplikasi *web* (Nwabudike, 2018). *Framework* bisa dalam bentuk kerangka kerja perangkat lunak yang menyajikan fungsi-fungsi standar dan konsep-konsep yang bisa dipakai lagi dalam pembuatan program, atau bisa dalam bentuk kerangka kerja

konseptual yang memberikan panduan atau metode spesifik dalam menyelesaikan masalah atau membuat keputusan di suatu bidang tertentu.

Laravel merupakan sebuah *framework* PHP (*Hypertext Propocessor*) yang dirilis di bawah lisensi MIT (*Massachusetts Institute of Technology*), dibangun dengan konsep MVC (*Model View Controller*). *Laravel* adalah pembangunan situs berbasis MVP (*Minimum Viable Product*) yang disusun dengan bahasa pemrograman PHP (*Hypertext Preprocessor*), dimaksudkan untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya awal dan biaya pemeliharaan. Selain itu, *Laravel* bertujuan untuk memperbaiki pengalaman pengembangan aplikasi dengan menyediakan sintaks yang ekspresif, mudah dimengerti, dan menghemat waktu (Supriyadi, 2023).

Laravel memiliki banyak fitur modern yang sangat membantu *developer* dalam membuat aplikasi. *Laravel* juga memiliki sejumlah kelebihan, seperti memanfaatkan *Command Line Interface* (CLI), *Artisan*, menggunakan manajer paket PHP (*Hypertext Preprocessor*) *Composer*, serta memungkinkan penulisan kode program yang lebih singkat, mudah dimengerti, dan ekspresif (Naja et al., 2023)

2.2.7 Konsep MVC

MVC (*Model View Controller*) adalah suatu metode pengembangan perangkat lunak yang memisahkan logika aplikasi dari tampilan. MVC memisahkan komponen-komponen aplikasi, antara lain manipulasi data, *controller*, dan *user interface* (Nwabudike, 2018)

a. *Model*

Model merupakan komponen awal dalam pola MVC (*Model View Controller*). Ini menggambarkan data dan logika bisnis aplikasi. *Model* berfungsi untuk mengelola informasi dan aturan yang diperlukan oleh aplikasi. Data dalam *Model* bisa berasal dari berbagai sumber seperti *database*, *file*, atau layanan *web*. *Model* ini tidak langsung berhubungan dengan pengguna, sehingga memudahkan pengujian karena tidak tergantung pada antarmuka pengguna. Selain itu, *Model* juga mengelola perubahan data dan memberitahu komponen lain jika terjadi perubahan.

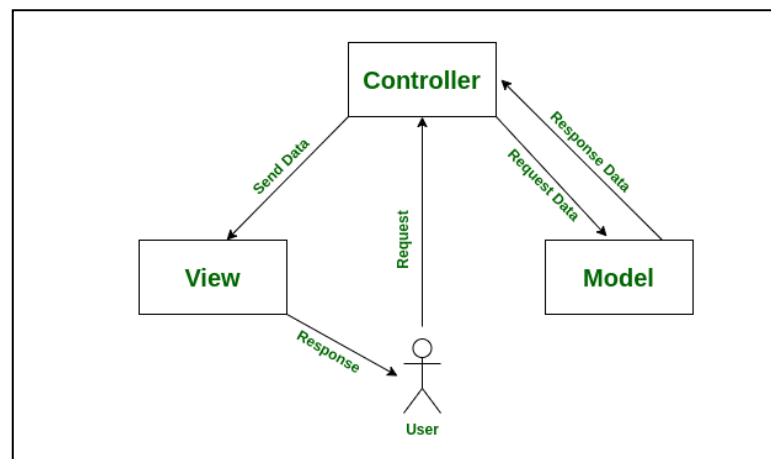
b. *View*

View adalah komponen yang menangani tampilan dan penyajian data kepada pengguna. Dalam konteks aplikasi *web*, *View* bisa berupa halaman HTML atau elemen antarmuka pengguna. *View* menerima data dari *Model* dan menampilkan informasi ini kepada pengguna. Pentingnya *View* dalam pola MVC (*Model View Controller*) adalah bahwa *View* tidak mengandung logika aplikasi. Fungsinya hanya untuk menampilkan data secara sesuai.

c. *Controller*

Controller merupakan elemen ketiga dalam pola MVC (*Model View Controller*). Fungsinya sebagai perantara antara *Model* dan *View*. *Controller* menerima permintaan dari pengguna dan menentukan tindakan yang harus dilakukan. Setelah itu, *Controller* berinteraksi dengan *Model* untuk mengambil atau memperbarui data yang dibutuhkan, dan akhirnya mengirimkan data tersebut ke *View* yang sesuai. Dengan kata lain, *Controller* mengatur aliran data antara *Model* dan *View*.

Konsep Model MVC (*Model View Controller*) dapat dilihat pada gambar 2.1.



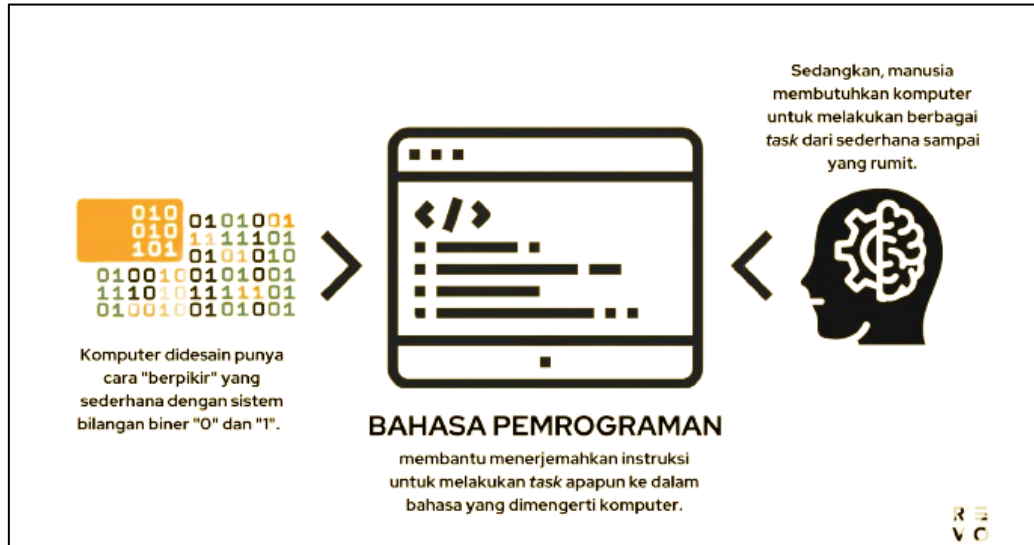
Gambar 2.1 Konsep model MVC (Bakri, 2023)

2.2.8 Bahasa Pemrograman

Bahasa pemrograman merupakan sarana yang dipakai oleh pengembang untuk menulis perintah-perintah yang kemudian diinterpretasikan oleh komputer. Bahasa pemrograman memiliki 3 *record* yaitu : *Syntax* (aturan penulisan suatu Bahasa Pemrograman atau dikenal dengan tata bahasa), kemudian *Semantic* (arti atau maksud yang terkandung pada suatu

statement/pernyataan) dan kebenaran logika (benar tidaknya urutan *statement*) (Latifatunnisa, 2022a).

Skema definisi bahasa pemrograman dapat lihat pada gambar 2.2



Gambar 2.2 Skema definisi bahasa pemrograman (Latifatunnisa, 2022b)

2.2.9 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *open-source* yang sering digunakan untuk membuat aplikasi *web* yang dinamis dan interaktif. PHP merupakan salah satu bahasa pemrograman yang telah ada sejak lama. PHP bisa dijalankan di *server web* dan digabungkan dengan HTML (*Hypertext Markup Language*), CSS (*Cascading Style Sheet*), dan JavaScript untuk menciptakan halaman *web* yang dinamis.

Saat ini, PHP sangat diminati oleh para pengembang *web* karena kemudahannya dalam pembelajaran serta kemampuannya yang kuat. PHP juga mendukung berbagai jenis *database*, seperti MySQL (*My Structured Query Language*), PostgreSQL, dan Oracle, memungkinkan pengembang untuk membuat aplikasi *web* yang lebih canggih dan bermanfaat.

Selain itu, bahasa pemrograman ini memiliki beragam kerangka kerja (*framework*) yang bisa dimanfaatkan oleh pengembang untuk mempercepat proses pembuatan aplikasi *web*. Beberapa kerangka kerja PHP yang terkenal meliputi *Laravel*, *CodeIgniter*, dan *Symfony*.

Sebagai bahasa skrip atau bahasa otomatisasi eksekusi tugas, PHP memiliki kemiripan dengan *JavaScript* dan *Python*. Tetapi perbedaannya terletak pada penggunaan PHP untuk komunikasi di sisi *server*, sedangkan *JavaScript* digunakan baik di *frontend* maupun *backend*, dan *Python* khusus digunakan di sisi *server (backend)* (News, 2023) .

Contoh skrip PHP (*Hypertext Preprocessor*) sederhana dapat dilihat pada gambar 2.3.

```
<!doctype html>
  <body>
    <?php
      echo "Hello World!";
    ?>
  </body>
</html>
```

Gambar 2.3 Contoh skrip PHP sederhana

2.2.10 MySQL

MySQL (*My Structure Query Language*) merupakan salah satu Sistem Manajemen Basis Data (DBMS) yang berperan dalam pengelolaan *database* dengan menggunakan bahasa SQL (*Structure Query Language*). MySQL merupakan sistem yang *open source*, sehingga dapat digunakan tanpa biaya. (Nwabudike, 2018). MySQL ini mendukung bahasa pemrograman PHP (*Hypertext Preprocessor*).

Selain itu, MySQL telah menjadi salah satu *database* yang sangat populer setelah *Oracle Database*. Hal ini disebabkan oleh sejumlah keunggulan yang dimilikinya (Dondy Bappedyanto, 2023). Berikut adalah beberapa keunggulan tersebut:

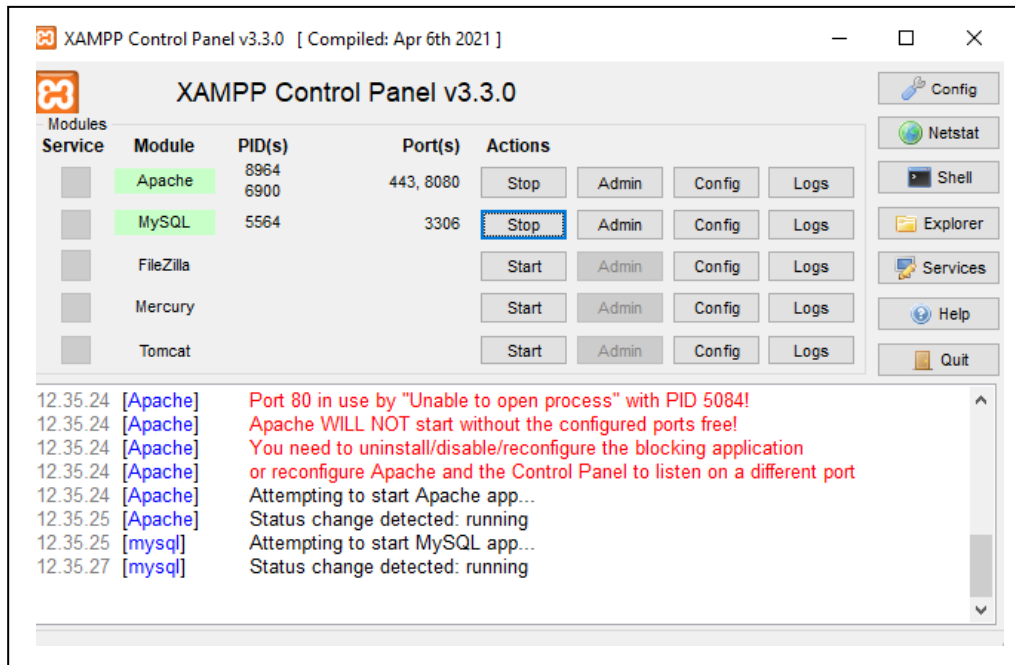
- a. *Open-source*
- b. Keamanan terjamin
- c. *Multi-user*
- d. Mendukung bahasa pemrograman lain
- e. Tidak membutuhkan RAM yang besar
- f. Mendukung berbagai macam data
- g. Struktur tabel *fleksibel*
- h. Standar industri

2.2.11 XAMPP

Adapun XAMPP memiliki kepanjangan X (*cross platform*), A (*Apache*), M (*MySQL/MariaDB*), P (*PHP*), dan P (*Perl*). XAMPP adalah perangkat lunak yang berisi berbagai aplikasi pemrograman dan basis data, termasuk *Apache HTTP Server*, *MySQL database*, serta bahasa pemrograman *PHP* dan *Perl* (Kesuma & Kholifah, 2019).

XAMPP disebut juga sebagai *standalone server* atau *server* yang dapat berdiri sendiri sehingga memudahkan pengguna saat menjalankan proses pengeditan, desain, dan pengembangan aplikasi.

Tampilan *XAMPP control panel* dapat dilihat pada gambar 2.4.



Gambar 2.4 Tampilan *XAMPP control panel*

2.2.12 Unified Modelling Language (UML)

UML (*Unified Modelling Language*) merupakan suatu bahasa berbasis grafik yang digunakan untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem pengembangan perangkat melalui gambaran grafis. *Unified Modeling Language* (UML) adalah salah satu alat yang berguna dalam pemrograman berbasis objek, dan saat ini, UML (*Unified Modelling Language*) dianggap akan menjadi standar bagi industri


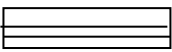

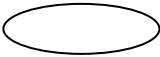

pengembangan sistem atau perangkat lunak yang mengadopsi pendekatan berorientasi objek (Harike, 2019).

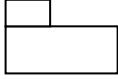
Ada beberapa macam diagram dalam *Unified Modeling Language* (UML), yaitu :

1. *Use Case Diagram*

Diagram *Use Case* merupakan representasi abstrak dari interaksi antara sistem dan aktor. Diagram ini digunakan untuk mendeskripsikan bagaimana pengguna atau aktor berinteraksi dengan sistem melalui skenario penggunaan. *Diagram Use Case* sangat penting dalam menyusun kebutuhan sistem, berkomunikasi dengan klien, dan merancang kasus uji untuk fitur-fitur sistem. *Use case* adalah cara untuk menggambarkan bagaimana sistem akan dilihat oleh pengguna, sementara diagram *use case* membantu dalam komunikasi antara analis, pengguna, dan klien.

Tabel 2.2 *Simbol Use Case Diagram*

| No | Simbol | Nama | Fungsi |
|----|---|--------------------|--|
| 1 |  | <i>Actor</i> | Semua entitas yang berhubungan dengan sistem aplikasi komputer. |
| 2 |  | <i>Class</i> | Membentuk sebuah sistem yang berpusat pada atribut dan operasional yang serupa. |
| 3 |  | <i>Interface</i> | Sekelompok operasi yang tidak diimplementasikan dari sebuah kelas. |
| 4 |  | <i>Use Case</i> | Menggambarkan tindakan yang dilakukan oleh aktor dalam sistem untuk mencapai suatu tujuan tertentu.. |
| 5 |  | <i>Interaction</i> | Untuk menampilkan arus pesan atau informasi baik antara objek atau hubungan di antara objek. |

| | | | |
|---|---|----------------|---|
| 6 |  | <i>Package</i> | Komentar atau kerangka konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dikembangkan, sehingga memungkinkan pembuatan model yang lebih sederhana. |
|---|---|----------------|---|

2. *Class Diagram*

Class adalah penggambaran sekumpulan objek dengan sifat, perilaku (operasi), dan hubungan yang serupa. Dengan menggunakan diagram kelas, dapat diperoleh gambaran menyeluruh tentang suatu sistem. Hal ini terlihat dari kelas-kelas yang ada dan hubungannya satu sama lain. Didalam sebuah sistem memiliki beberapa *class diagram*. Diagram kelas sangat berguna dalam menggambarkan struktur kelas dari sistem tertentu secara visual.

3. *Component Software*

Component software (Komponen perangkat lunak) adalah elemen fisik dari sistem yang berada di komputer, dan seringkali tidak disadari oleh para analis karena itu merupakan bagian dari implementasi perangkat lunak dari satu atau lebih kelas. Komponen dapat berupa *source code*, komponen *biner*, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari sebuah *logical view* ke komponen *view*. Oleh karena itu, diagram komponen dapat mencerminkan realitas yaitu sebuah *component software* yang mengandung *component*, *interface* dan *relationship*.

4. *Deployment Diagram*

Menyajikan secara visual susunan fisik suatu sistem, memperlihatkan bagaimana perangkat lunak berinteraksi dengan perangkat keras, serta mengilustrasikan hubungan antar komputer (*node*) dan jenis interaksinya. Dalam satu *node*, komponen yang dapat dieksekusi dan objek yang dialokasikan menunjukkan unit perangkat lunak yang telah dijalankan oleh *node* tersebut dan ketergantungannya pada komponen lain.

5. *State Diagram*

Menjelaskan seluruh keadaan atau kondisi yang dimiliki oleh objek dari suatu kelas serta peristiwa-peristiwa yang menyebabkan perubahan keadaan. Perubahan ini dapat disebabkan oleh objek lain yang mengirim pesan. Kondisi kelas tidak dijelaskan untuk semua kelas, hanya untuk yang memiliki sejumlah keadaan yang jelas dan kondisi kelas berubah oleh keadaan yang berbeda.





6. *Collaboration Diagram*

Menggambarkan interaksi dinamis seperti diagram urutan. Dalam menggambarkan alur pertukaran pesan, diagram kolaborasi mengilustrasikan objek beserta relasinya (dalam konteks yang relevan). Jika fokusnya pada urutan waktu, gunakan diagram urutan, tetapi jika fokusnya pada konteks, gunakan diagram kolaborasi.

7. *Activity Diagram*

Melukiskan urutan aliran aktivitas, digunakan untuk menjelaskan aktivitas yang terbentuk dalam suatu operasi, dan bisa juga diterapkan pada aktivitas lainnya seperti kasus penggunaan atau interaksi. Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem.

Tabel 2.3 *Simbol Activity Diagram*

| No | Simbol | Nama | Fungsi |
|----|---|--------------|--|
| 1 |  | Status Awal | Keadaan awal aktivitas sistem pada sebuah diagram aktivitas memiliki sebuah status awal. |
| 2 |  | Aktivitas | Tindakan yang dilakukan oleh sistem, biasanya dimulai dengan kata kerja. |
| 3 |  | Percabangan | Cabang asosiatif yang terjadi ketika terdapat lebih dari satu pilihan aktivitas. |
| 4 |  | Status akhir | Keadaan terakhir yang dilakukan oleh sistem. |

2.2.13 *Flowchart*

Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah. *Flowchart* memiliki peran kunci dalam menentukan langkah-langkah atau fungsi dari sebuah proyek pembuatan program yang melibatkan banyak orang secara bersamaan. Selain itu, menggunakan diagram alur proses dari program akan membuatnya lebih jelas, padat, dan mengurangi risiko salah penafsiran. Penerapan *flowchart* dalam dunia pemrograman juga merupakan metode yang efektif untuk menyatukan kebutuhan teknis dan non-teknis (Setiawan, 2021).

Flowchart sendiri terdiri dari lima jenis, masing-masing jenis memiliki karakteristik dalam penggunaannya. Berikut adalah jenis-jenisnya:

1. *Flowchart* dokumen

Flowchart dokumen (*document flowchart*) atau bisa juga disebut dengan *paperwork flowchart*. *Flowchart* dokumen berfungsi untuk menelusuri alur form dari satu bagian ke bagian yang lain, termasuk bagaimana laporan diproses, dicatat, dan disimpan.

2. *Flowchart* program

Flowchart ini menggambarkan secara rinci prosedur dari proses program. *Flowchart* program terdiri dari dua macam, antara lain: *flowchart* logika program (*program logic flowchart*) dan *flowchart* program komputer terinci (*detailed computer program flowchart*).

3. *Flowchart* proses

Flowchart proses adalah cara penggambaran rekayasa industrial dengan cara merinci dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem.

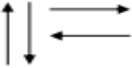




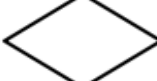
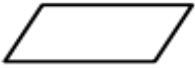




4. *Flowchart* sistem



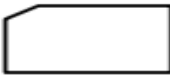

Flowchart sistem adalah *flowchart* yang menampilkan tahapan atau proses kerja yang sedang berlangsung di dalam sistem secara menyeluruh. Selain itu *flowchart* sistem juga menguraikan urutan dari setiap prosedur yang ada di dalam sistem.

5. Flowchart skematik

Flowchart ini menampilkan alur prosedur suatu sistem, hampir sama dengan *flowchart* sistem. Namun, ada perbedaan dalam penggunaan simbol-simbol dalam menggambarkan alur. Selain simbol-simbol, *flowchart* skematik juga menggunakan gambar-gambar komputer serta peralatan lainnya untuk mempermudah dalam pembacaan *flowchart* untuk orang awam.

Tabel 2.4 Simbol Flowchart

| Gambar | Keterangan |
|---|---|
|  | Flow Direction Symbol Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain |
|  | Terminator Symbol Terminal simbol untuk permulaan (start) atau akhir (stop) dari suatu diagram flowchart.. |
|  | Connector Symbol Simbol keluar – masuk atau penyambungan proses dalam lembar atau halaman yang sama. |
|  | Connector Symbol Simbol keluar – masuk atau penyambungan proses pada lembar atau halaman yang berbeda. |
|  | Processing Symbol Simbol indikasi suatu proses pengolahan fungsi pada program. |
|  | Decision Symbol Simbol pemilihan keputusan berdasarkan dua kondisi benar dan salah pada flowchart |
|  | Input – Output Symbol Simbol yang menyatakan fungsi input (masukan) atau output (keluaran) dari suatu program |
|  | Manual Input Symbol Simbol indikasi manual input data melalui keyboard jika dalam program membutuhkan masukan data secara manual. |
|  | Preparation Symbol Simbol inisialisasi atau pemberian nilai awal untuk persiapan langkah proses selanjutnya. |
|  | Predefined Process Symbol (subroutine) Simbol untuk memanggil sub-process atau prosedur ditempat yang berbeda. |
|  | Display Symbol Simbol yang berguna untuk menunjukkan di mana informasi akan ditampilkan dalam aliran proses. |

| | |
|---|--|
|  | Stored Data Symbol Simbol yang menunjukkan objek penyimpanan data umum yang digunakan dalam alur proses contohnya hardisk, flashdisk atau perangkat penyimpanan lainnya. |
|  | Sequential Access Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik. |
|  | Card Symbol Simbol yang menunjukkan input berasal dari kartu atau output disimpan ke kartu |
|  | Document Symbol Simbol yang menyatakan langkah proses yang akan menghasilkan dokumen. |

2.2.14 Perancangan Sistem

Perancangan sistem adalah proses perancangan untuk merancang sistem atau memperbaiki sistem yang telah ada sehingga sistem menjadi lebih baik serta dapat mengerjakan pekerjaan secara efektif dan efisien, proses rancangan bisa berupa rancangan input, rancangan output, rancangan file. Perancangan sistem dimulai dengan pembuatan pemodelan menggunakan diagram kasus pengguna untuk menggambarkan interaksi antara pemanggil dan fungsi-fitur yang akan diimplementasikan dalam sistem kedepannya (Malik et al., 2023).

2.2.15 Metode *Extreme Programming* (XP)

Extreme Programming (XP) merupakan pendekatan pengembangan perangkat lunak atau manajemen proyek yang bersifat *agile*, dengan tujuan menciptakan perangkat lunak yang lebih unggul dalam hal kualitas. XP menggunakan siklus pengembangan yang pendek, sehingga mampu menanggapi secara cepat terhadap perubahan keinginan pengguna yang berubah-ubah (*Extreme Programming: 5 Tahapan, Kelebihan, Dan Kekurangan*, n.d.). Metode XP tidak hanya memfokuskan pada *coding* tetapi juga ke seluruh bagian area pengembangan perangkat lunak. Nilai utama yang mendasari dari metode ini adalah *communications* (komunikasi), *courage* (Keberanian), *simplicit* (Kesederhanaan), dan *feedback* (Umpan balik) (*Extreme Programming (XP) - Definisi, Nilai, Tahapan - Serupa.Id*, n.d.).

Tahapan *Extreme Programming* yaitu *Planning*, *design*, *coding* dan *Testing*. Adapun kelebihan dan kekurangan metode ini adalah sebagai berikut:

1. Kelebihan *Extreme Programming* (XP) yaitu :
 - a. Efisiensi waktu dan biaya

- b. Minim resiko
 - c. Sederhana
 - d. Prosesnya transparan dan dapat dipertanggungjawabkan
 - e. Konstan *feedback*
 - f. Perangkat lunak yang berfungsi lebih cepat
 - g. *Teamwork*
 - h. Meningkatkan kepuasan tim
2. Kekurangan *Extreme Programming* (XP) yaitu:
- a. Kurangnya dokumentasi
 - b. Minim desain
 - c. Tidak cocok dikerjakan secara remote

2.2.16 Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi sejauh mana sistem yang dibuat memenuhi kebutuhan yang telah ditetapkan pada fase analisis *prosedur* sebelumnya. Pengujian sistem bertujuan untuk mengidentifikasi cacat dan kesalahan serta memeriksa kualitas perangkat lunak. Ada dua teknik dalam pengujian sistem yaitu pengujian kotak hitam (*Black Box*) dan pengujian kotak putih (*White Box*).

1. *Black Box testing* (Pengujian Kotak Hitam)

Black box testing merupakan pengujian perangkat lunak/sistem dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian ini dilakukan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari sistem sesuai dengan spesifikasi yang dibutuhkan (Nwabudike, 2018). Pengujian kotak hitam digunakan untuk menemukan kesalahan dalam kategori berikut:

- a) Fungsi yang salah atau hilang
- b) Kesalahan antarmuka
- c) Kesalahan dalam struktur data atau akses basis data eksternal
- d) Kesalahan perilaku atau kinerja
- e) Kesalahan insialisasi dan penghentian.

2. *White Box testing* (Pengujian Kotak Putih)

White box testing dikenal sebagai *clear box testing*, *glass box testing* atau *structural testing* merupakan metode pengujian perangkat lunak yang

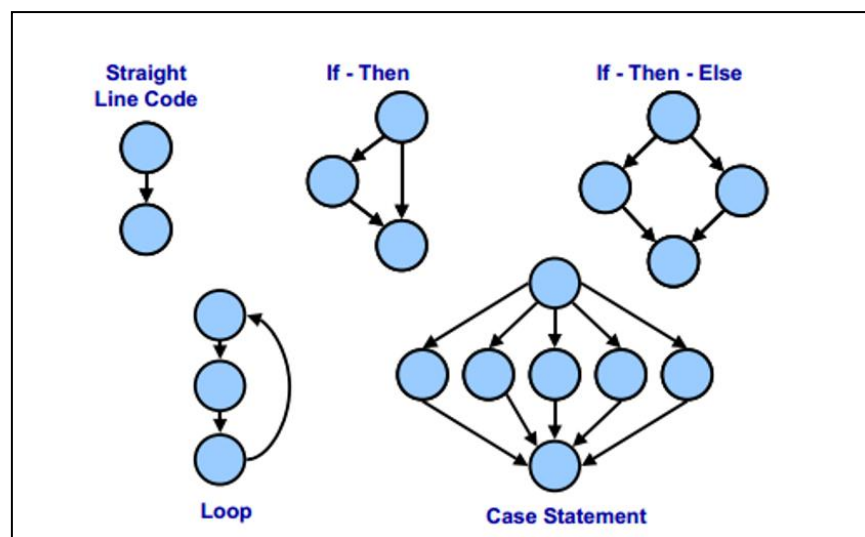
mengevaluasi struktur internal dan kode dari aplikasi (Wintana et al., 2022). Tujuan utama *white box testing* adalah untuk memastikan bahwa semua jalur logika dalam kode sistem telah diuji dan berfungsi dengan benar, mencakup pengecekan terhadap struktur internal dan aliran kontrol program untuk mengidentifikasi bug atau kesalahan dalam implementasi.

Teknik pengujian *white box* salah satunya adalah *basis patch testing* (Menguji semua pernyataan atau statement dalam program setidaknya satu kali). Basis path testing diusulkan pertama kali oleh Tom McCabe.

1. Notasi grafik alir (flow Graph)

Merupakan sebuah notasi sederhana untuk mempersentasikan aliran control logis suatu program.

Struktur basic control flow graph dapat dilihat pada gambar 2.5



Gambar 2.5 Struktur *Basic Control Flow Graph*(*Bean Test: Basis Path Testing*, n.d.)

2. Cyclomatic Complexity

Adalah matric perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas suatu program. Dalam konteks metode pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independent dalam basis set suatu program dan menyediakan batas atas untuk jumlah pengujian yang harus dilakukan guna memastikan bahwa semua pernyataan telah dieksekusi minimal sekali. (Sie et al., 2022)

Rumus:

$$V(G) = E - N + 2$$

dimana :

E = jumlah edge pada grafik alir

N = Jumlah node pada grafik alir

2 = Salah kondisi, kondisi yang dibuat dikarenakan pada kode program terdapat kelebihan atau kesalahan implementasi kode.

$$V(G) = P + 1$$

Dimana, P = Jumlah predikat *node* pada grafik alir.

$$V(G) = R$$

Dimana, R = *Region* yang menandakan suatu area dalam *flow graph*.

Untuk mengetahui tingkat resiko dari suatu skenario berdasarkan dari jumlah *Cyclomatic complecity*-nya dapat dilihat pada tabel berikut ini:

Tabel 2.5 Hubungan *Cyclomatic complecity* dengan resiko (Sie et al., 2022)

| Nilai CC | Tipe Prosedur | Tingkat Resiko |
|----------|---|----------------|
| 1 – 4 | Prosedur sederhana | Rendah |
| 5 – 10 | Prosedur yang tersusun dengan baik dan stabil | Rendah |
| 11 – 20 | Prosedur yang lebih kompleks | Menengah |
| 21 – 50 | Prosedur yang kompleks dan kritis | Tinggi |
| >50 | Rentan kesalahan, sangat mengganggu, prosedur tidak dapat diuji | Sangat Tinggi |

3. Graph Matrik

Merupakan matrik segi empat sama sisi, mempunyai jumlah kolom dan baris yang sama dengan jumlah node yang didapatkan dalam pemetaan kode program, isi dari data merupakan keberadaan penghubung antar node. Pembobotan pada hubungan antar node (Vikasari, 2023), sebagai berikut:

- a. Jalur (Edge) yang pada saat dilakukan pengujian kemungkinan akan dieksekusi.

- b. Waktu yang digunakan dalam memproses pengujian yang diharapkan ada pada edge selama proses pengujian
- c. Memori yang dibutuhkan selama proses transfer lakukan pada edge
- d. Resource yang dibutuhkan dalam proses transfer pada edge.

2.3 Kerangka Pikir



Gambar 2.6 Kerangka Pikir

BAB III

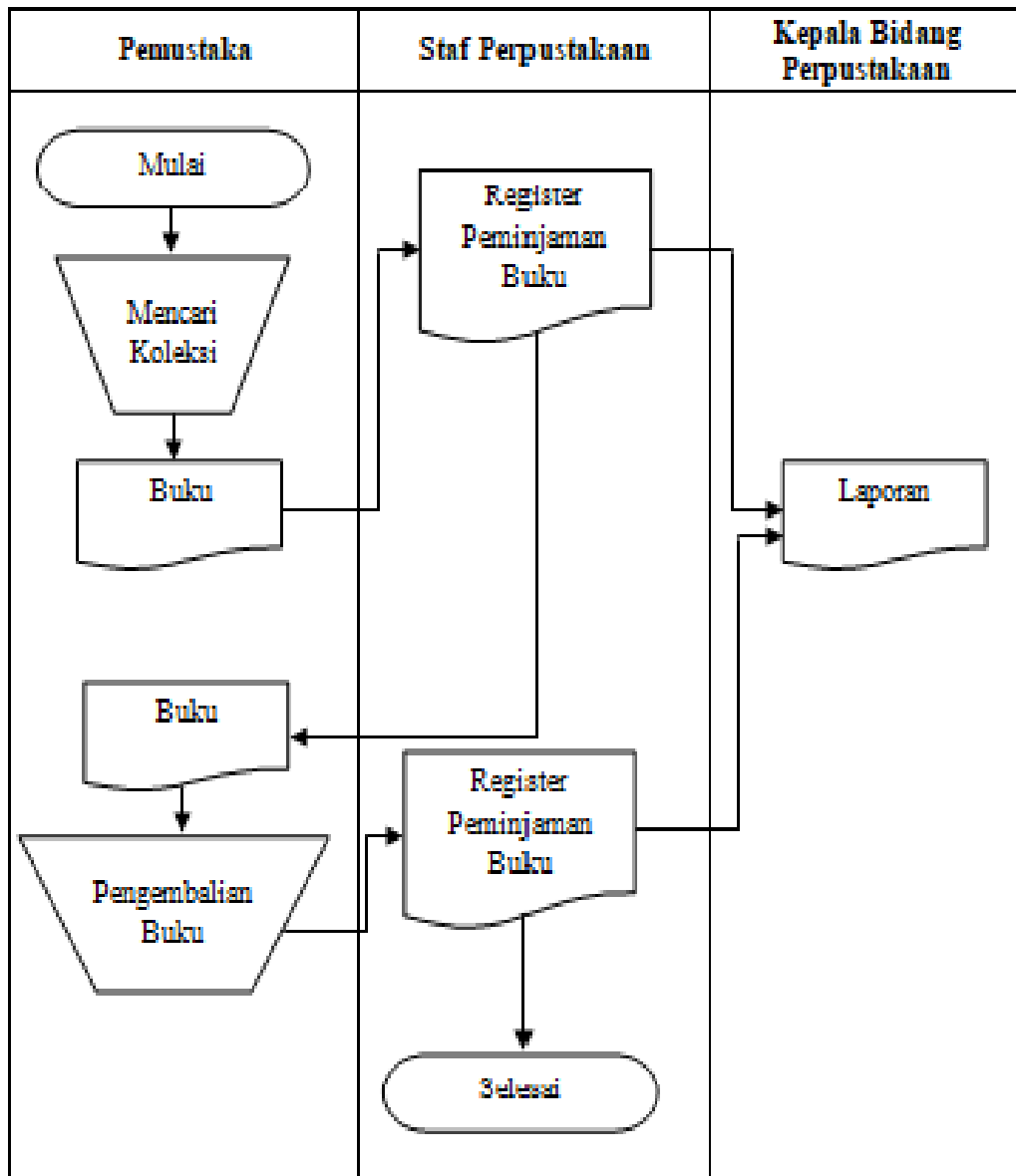
ANALISIS DAN RANCANGAN SISTEM

3.1 Analisis Sistem

3.1.1 Analisis Sistem yang sedang berjalan

Dari analisis yang dilakukan oleh penulis terhadap Perpustakaan Daerah Kabupaten Luwu Utara, disimpulkan bahwa sistem yang digunakan masih bersifat manual.

Rincian lebih lanjut dapat ditemukan dalam gambar 3.1

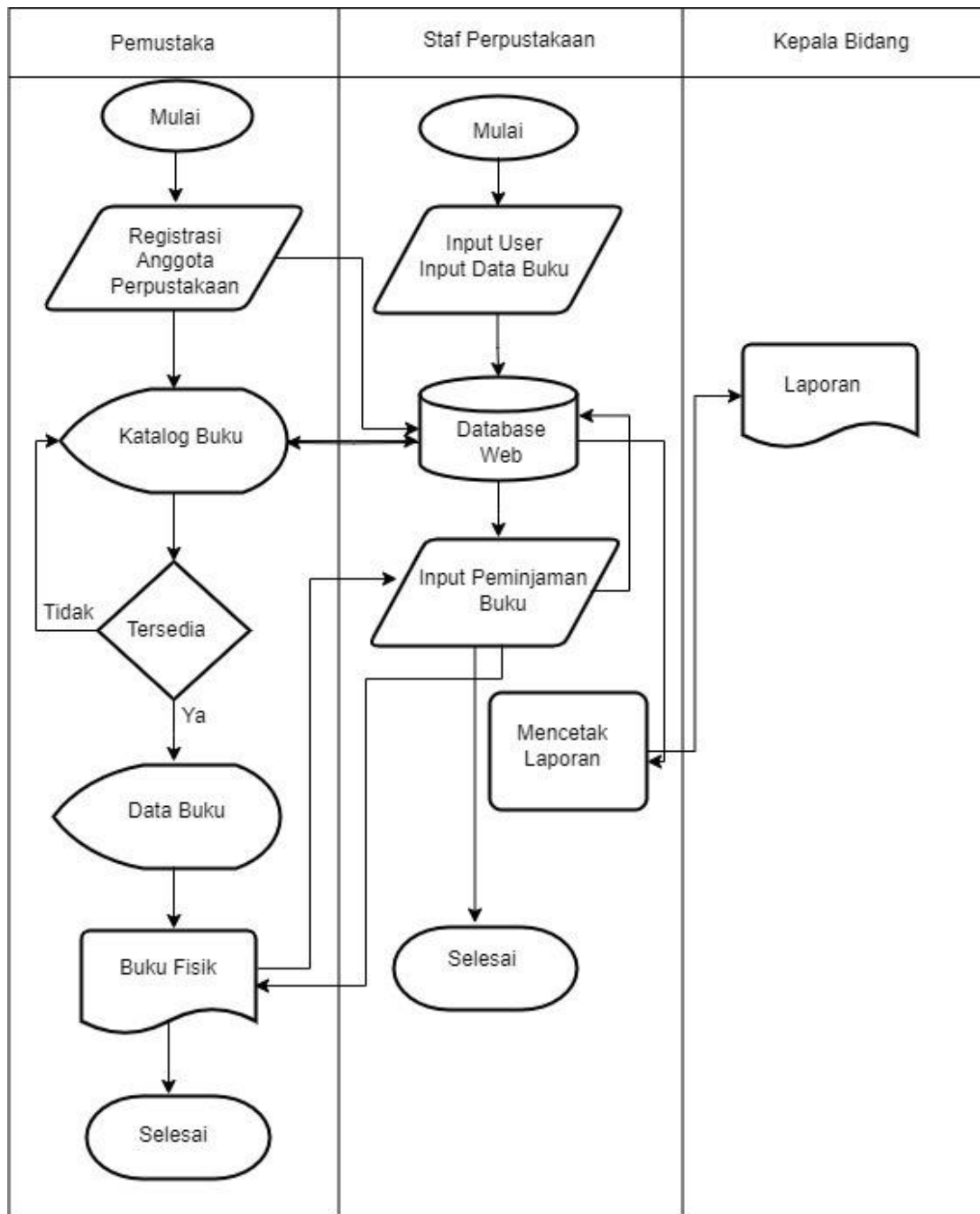


Gambar 3.1 Analisis Sistem yang sedang berjalan

3.1.2 Sistem yang diusulkan

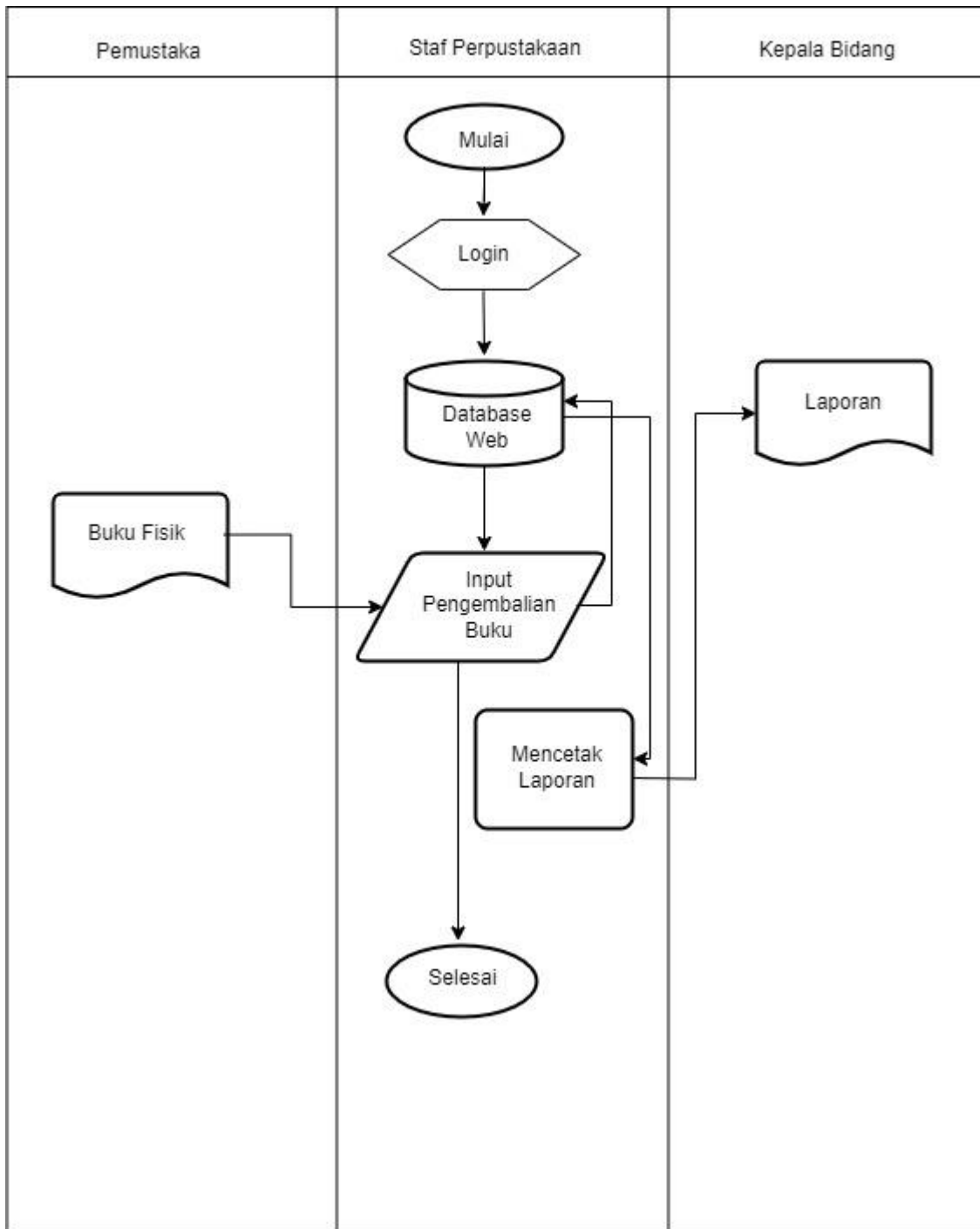
Adapun diagram alir dokumen sistem yang diusulkan dibagi menjadi 2 (dua) diagram alir yaitu diagram alir sistem peminjaman buku dan diagram alir sistem pengembalian buku.

Sistem diagram alir peminjaman buku dapat dilihat pada gambar 3.2



Gambar 3.2 *Flowchart Peminjaman Buku yang akan diusulkan*

Sistem diagram alir pengembalian buku dapat dilihat pada gambar 3.3



Gambar 3.3 Flowchart Pengembalian Buku yang akan diusulkan

3.2 Deskripsi Data

Data yang digunakan dan diproses dalam penelitian ini yaitu Data Buku, data Anggota dan data transaksi .

1. Data buku

Data buku terdiri dari *field Id* buku, judul buku, penulis, penerbit, ISBN, tahun terbit, Kategori, dan Klasifikasi.

2. Data Anggota

Data anggota terdiri dari *field Id*, nama, NIK, alamat, Jenis Kelamin, Tanggal lahir, dan No. Hp.

3. Data Transaksi

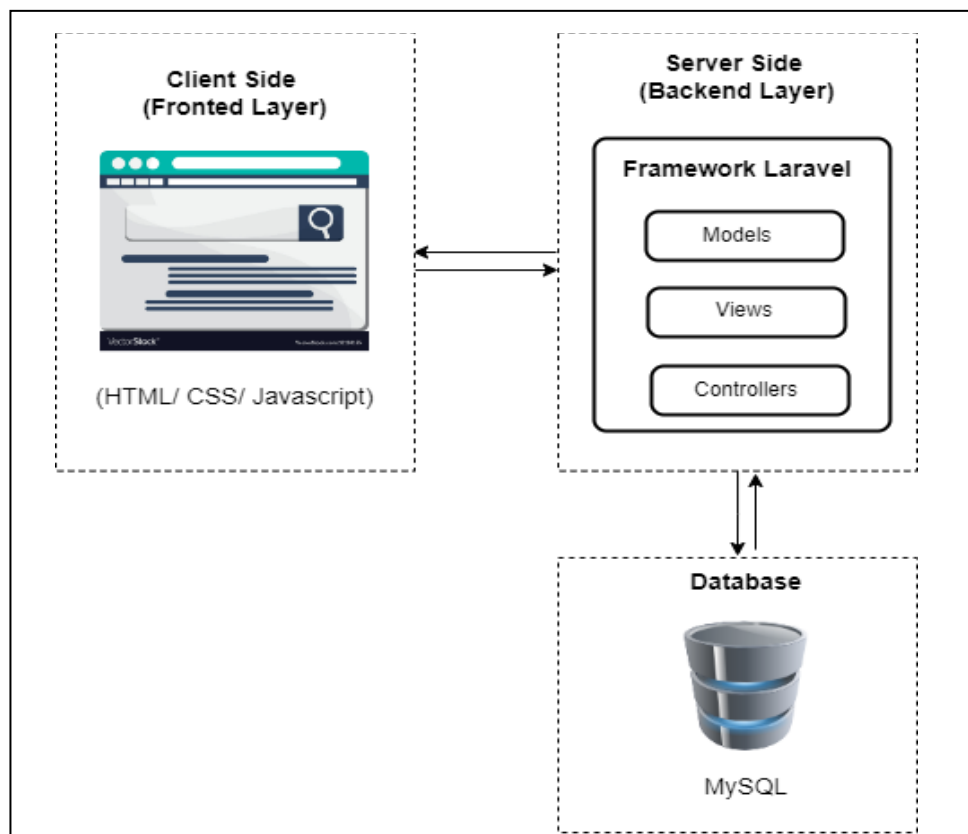
Data transaksi terdiri dari *Field Id*, Judul buku, tanggal peminjaman dan tanggal pengembalian.

3.3 Arsitektur Sistem

Arsitektur sistem menentukan bagaimana komponen-komponen tersebut diorganisasikan dan bagaimana mereka saling berinteraksi untuk memastikan kinerja, skalabilitas, keandalan, dan keamanan sistem secara keseluruhan. Arsitektur sistem dalam penelitian ini terdiri dari 3 bagian yaitu *Client Side*, *Server Side* dan *Database*.

1. *Client Side (Frontend)* merupakan jenis website yang memproses data dan menampilkan halaman *web* melalui *browser* pengguna (*Client*) seperti Antarmuka Pengguna (*User Interface*) terdiri dari :
 - a) Desain antarmuka menggunakan HTML, CSS dan *JavaScript*.
 - b) Fitur utama: Pencarian buku, peminjaman dan pengembalian buku, pendaftaran pengguna, dan tampilan Katalog buku.
2. *Server Side (Backend)* merupakan suatu istilah yang merujuk pada pemrosesan dan pengelolaan data yang terjadi di sisi server dalam sebuah aplikasi *web* atau sistem komputer. Dalam penelitian ini menggunakan *Framework Laravel* yang terdiri dari :
 - a) *Laravel* digunakan untuk mengelola logika aplikasi, *autentikasi* pengguna, dan *routing*.
 - b) Struktur MVC (*Model-View-Controller*) memisahkan logika bisnis, tampilan, dan pengelolaan data.

- c) Fitur Utama: Manajemen pengguna, manajemen koleksi buku, manajemen transaksi peminjaman dan pengembalian, serta laporan statistik.
3. *Database* merupakan data yang disimpan dan dikelola sedemikian rupa sehingga dapat diakses, diperbaharui, dan dikelola dengan mudah. *Database* yang digunakan dalam penelitian ini yaitu menggunakan *MySQL* yang terdiri dari:
- MySQL* digunakan untuk menyimpan data.
 - Struktur *database* relasional dengan tabel utama: *Users* (Pengguna) , *Books* (Buku), *Transactions* (Transaksi), *Categories* (Kategori), Nomor Klasifikasi dan *Authors*.
 - ORM (*Object-Relational Mapping*) *Eloquent* dari *Laravel* digunakan untuk interaksi dengan *database*.



Gambar 3.4 Diagram *Arsitektur* Sistem

Alur arsitektur sistem dimulai dengan interaksi pengguna melalui antarmuka pengguna yang responsif, melalui *web browser* yang

mengirimkan permintaan ke *backend*. *Backend*, yang menggunakan *framework Laravel* dan bahasa pemrograman PHP, menerima permintaan tersebut dan memprosesnya. Permintaan tersebut kemudian diarahkan ke layanan *backend* yang sesuai, seperti manajemen buku, manajemen anggota, atau transaksi peminjaman, di mana data yang diperlukan diambil dari basis data yang terhubung (seperti *MySQL* atau *PostgreSQL*). Setelah permintaan diproses dan data diperoleh, *backend* mengirimkan respons kembali ke antarmuka pengguna, yang kemudian menampilkan informasi yang diminta kepada pengguna. Data yang diperoleh juga dapat digunakan untuk analisis lebih lanjut atau dilaporkan. Selain itu, sistem ini memperhatikan aspek keamanan dengan melakukan otentikasi pengguna, otorisasi akses, dan penggunaan *HTTPS* untuk menyandikan komunikasi antara *klien* dan *server*. Dengan alur ini, sistem memastikan pengguna mendapatkan pengalaman yang responsif, aman, dan informatif saat berinteraksi dengan sistem informasi perpustakaan di Kabupaten Luwu Utara.

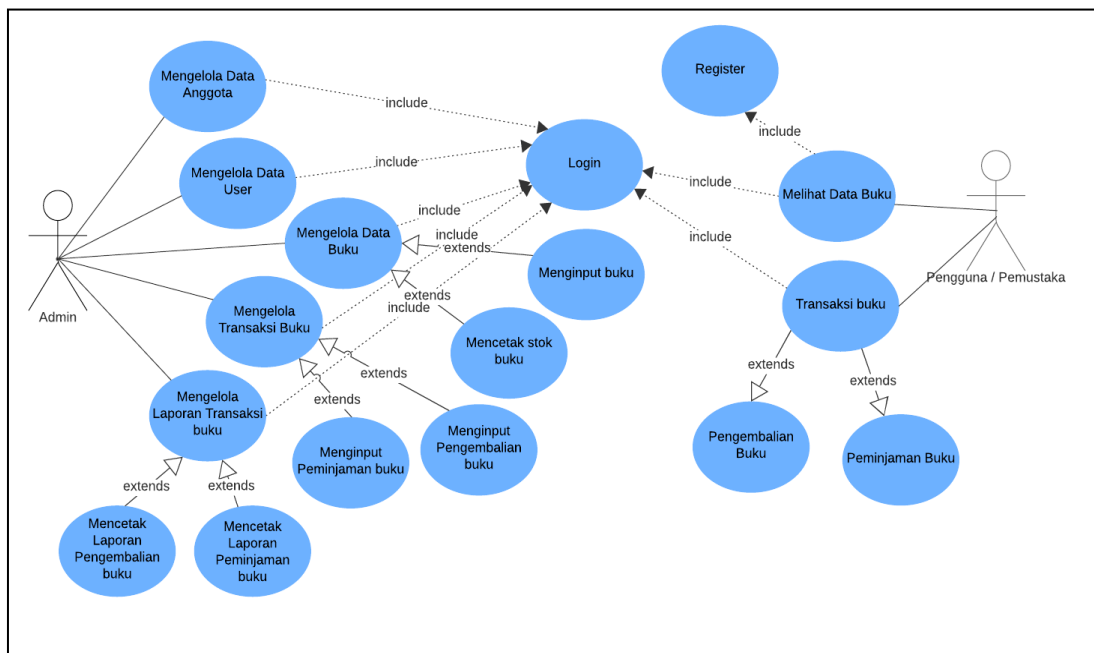
3.4 Rancangan Model Sistem

3.4.1 Perancangan UML (Unified Modeling Language)

Pada tahap perancangan model sistem akan memberikan gambaran tentang model sistem yang akan dibuat dan yang akan dikembangkan. Pada bagian ini perancangan ini akan digambarkan melalui diagram UML serta desain antarmuka program.

1. Use Case Diagram

Pada tahapan ini menggambarkan siapa saja *actor* yang terlibat di dalam sistem dan apa yang bisa dilakukan oleh *actor* di dalam sistem. Adapun gambar dari *use case* dapat dilihat pada gambar 3.5 dibawah ini:



Gambar 3.5 Use Case Diagram

Use case diagram untuk sistem informasi perpustakaan daerah kabupaten Luwu Utara menggunakan *framework laravel* melibatkan dua (2) aktor utama yaitu *Admin* dan *Pegguna (Pemustaka)*. *Admin* memiliki wewenang untuk mengelola data anggota, *user*, data buku (terdiri dari menginput data buku, dan mencetak stok buku), transaksi buku (terdiri dari menginput peminjaman buku dan menginput pengembalian buku), dan laporan transaksi buku (terdiri dari mencetak laporan peminjaman dan laporan pengembalian buku). Sedangkan untuk *Pegguna/pemustaka* harus terdaftar sebagai anggota perpustakaan untuk dapat mencari buku dan melakukan transaksi buku (peminjaman dan pengembalian buku) diperpustakaan melalui sistem informasi ini.

2. Skenario Use Case

Skenario *use case* adalah deskripsi tentang bagaimana pengguna atau aktor berinteraksi dengan sistem untuk mencapai tujuan tertentu. Ini digunakan dalam analisis dan desain sistem perangkat lunak untuk memahami kebutuhan fungsional sistem dari *perspektif* pengguna. Adapun Skenario *use case* yang akan dilakukan yaitu:

1) Skenario Use Case mengelola data buku

Nama Use Case : Mengelola data buku

Aktor : *Admin*

Tujuan : Pengelolaan data buku
 Kondisi Awal : *Admin* belum melakukan *login*
 Kondisi : *Admin* sudah *login*
 Kondisi Akhir : *Admin* berhasil *login*
 Deskripsi : Proses mengelola data buku adalah proses untuk mengatur dan mengelola data buku pada sistem yang meliputi tambah, *Edit*, *Delete*, dan Cari.

Tabel 3.1 Skenario *Use Case* Mengelola Data Buku

| <i>Admin</i> | Reaksi Sistem |
|--|---|
| 1. Memilih menu data master, data buku | 2. Sistem menampilkan data buku |
| Skenario alternatif <i>input</i> (Tambah) | |
| 3. Menekan tombol Tambahkan data | 4. Menampilkan <i>form</i> tambah buku |
| 5. Mengisi form data buku dan menekan tombol <i>submit</i> | 6. Sistem menampilkan data berhasil ditambahkan dan disimpan oleh sistem. |
| Skenario alternatif <i>Edit</i> data | |
| 7. Menekan tombol <i>Edit</i> data | 8. Sistem menampilkan data yang telah dipilih |
| 9. Mengubah data pada <i>form</i> data buku dan menekan tombol <i>update</i> | 10. Sistem menampilkan pesan data berhasil diubah oleh sistem. |
| Skenario alternatif <i>Delete</i> / Hapus data | |
| 11. Menekan tombol <i>Delete</i> /Hapus | 12. Sistem menampilkan pilihan hapus atau batal |
| 13. Menekan tombol <i>Delete</i> | 14. Sistem menampilkan pesan data berhasil dihapus oleh sistem. |
| Skenario Alternatif <i>Search</i> / Cari data | |
| 15. Memasukkan kata kunci pada kolom pencarian. | 16. Sistem menampilkan data yang dicari. |

| | |
|----------------------|----------------------|
| Kondisi Akhir | Data berhasil diolah |
|----------------------|----------------------|

2) Skenario *Use Case* mengelola data anggota

Nama *Use Case* : Mengelola Data anggota

Aktor : *Admin*

Tujuan : Pengelolaan data anggota

Kondisi Awal : *Admin* belum melakukan *login*

Kondisi : *Admin* sudah *login*

Kondisi Akhir : *Admin* berhasil *login*

Deskripsi : Proses mengelola data anggota adalah proses untuk mengatur dan mengelola data anggota pada sistem yang meliputi tambah, *Edit*, dan *Delete*.

Tabel 3.2 Skenario *Use Case* Mengelola Data Anggota

| Admin | Reaksi Sistem |
|---|---|
| 1. Memilih menu data master, data anggota | 2. Sistem menampilkan data anggota |
| Skenario alternatif <i>input</i> (Tambah) | |
| 3. Menekan tombol Tambahkan data | 4. Menampilkan form tambah anggota |
| 5. Mengisi <i>form</i> data anggota dan menekan tombol <i>submit</i> | 6. Sistem menampilkan data berhasil ditambahkan dan disimpan oleh sistem. |
| Skenario alternatif <i>Edit</i> data | |
| 7. Menekan tombol <i>Edit</i> data | 8. Sistem menampilkan data yang telah dipilih |
| 9. Mengubah data pada <i>form</i> data anggota dan menekan tombol <i>update</i> | 10. Sistem menampilkan pesan data berhasil diubah oleh sistem. |
| Skenario alternatif <i>Delete</i> / Hapus data | |
| 11. Menekan tombol <i>Delete</i> /Hapus | 12. Sistem menampilkan pilihan hapus atau batal |
| 13. Menekan tombol <i>Delete</i> | 14. Sistem menampilkan pesan data berhasil dihapus oleh |

| | |
|----------------------|----------------------|
| | sistem. |
| Kondisi Akhir | Data berhasil diolah |

3) Skenario *Use Case* mengelola data *user*

Nama *Use Case* : Mengelola Data *user*

Aktor : *Admin*

Tujuan : Pengelolaan data *user*

Kondisi Awal : *Admin* belum melakukan *login*

Kondisi : *Admin* sudah *login*

Kondisi Akhir : *Admin* berhasil *login*

Deskripsi : Proses mengelola data *user* adalah proses untuk Mengatur dan mengelola data *user* pada sistem yang meliputi tambah, *submit*, *Edit*, *Delete*, dan *Search/Cari*

Tabel 3.3 Skenario *Use Case* Mengelola Data User

| <i>Admin</i> | Reaksi Sistem |
|---|---|
| 1. Memilih menu data master, data <i>user</i> | 2. Sistem menampilkan data <i>user</i> |
| Skenario alternatif <i>input</i> (Tambah) | |
| 3. Menekan tombol Tambahkan data | 4. Menampilkan <i>form</i> tambah <i>user</i> |
| 5. Mengisi form data <i>user</i> dan menekan tombol <i>submit</i> | 6. Sistem menampilkan data berhasil ditambahkan dan disimpan oleh sistem. |
| Skenario alternatif <i>Edit</i> data | |
| 7. Menekan tombol <i>Edit</i> data | 8. Sistem menampilkan data yang telah dipilih |
| 9. Mengubah data pada <i>form</i> data <i>user</i> dan menekan tombol <i>update</i> | 10. Sistem menampilkan pesan data berhasil diubah oleh sistem. |
| Skenario alternatif <i>Delete</i> / Hapus data | |
| 11. Menekan tombol <i>Delete</i> /Hapus | 12. Sistem menampilkan pilihan hapus atau batal |

| | |
|---|---|
| 13. Menekan tombol <i>Delete</i> | 14. Sistem menampilkan pesan data berhasil dihapus oleh sistem. |
| Skenario Alternatif <i>Search</i> data | |
| 15. Memasukkan kata kunci pada kolom pencarian. | 16. Sistem menampilkan data yang dicari. |
| Kondisi Akhir | Data berhasil diolah |

4) Skenario *Use Case* mengelola data transaksi

Nama *Use Case* : Mengelola Data Transaksi

Aktor : *Admin*

Tujuan : Pengelolaan data transaksi

Kondisi Awal : *Admin* belum melakukan *login*

Kondisi : *Admin* sudah *login*

Kondisi Akhir : *Admin* berhasil *login*

Deskripsi : Proses mengelola data transaksi adalah proses Untuk mengelola data pinjaman buku, sistem yang meliputi tombol tambah, *submit*, sudah Kembali, dan *Delete*.

Tabel 3.4 Skenario *Use Case* Mengelola Data Transaksi

| <i>Admin</i> | Reaksi Sistem |
|--|---|
| 1. Memilih menu transaksi pada <i>dashboard</i> | 2. Sistem menampilkan data transaksi |
| Skenario alternatif <i>input</i> (Tambah) | |
| 3. Menekan tombol Tambahkan data | 4. Menampilkan <i>form</i> tambah transaksi |
| 5. Mengisi <i>form</i> data transaksi dan menekan tombol <i>submit</i> | 6. Sistem menampilkan data berhasil ditambahkan dan disimpan oleh sistem. |
| Skenario alternatif Sudah Kembali | |
| 7. Menekan tombol Sudah Kembali | 8. Sistem menampilkan pilihan OK atau <i>Cancel</i> |
| 9. Menekan tombol OK | 10. Sistem menampilkan pesan |

| | |
|---|---|
| | data berhasil diubah oleh sistem. |
| Skenario alternatif <i>Delete</i> / Hapus data | |
| 11. Menekan tombol <i>Delete</i> /Hapus | 12. Sistem menampilkan pilihan hapus atau <i>cancel</i> . |
| 13. Menekan tombol <i>Delete</i> | 14. Sistem menampilkan pesan data berhasil dihapus oleh sistem. |
| Kondisi Akhir | Data berhasil diolah |

5) Skenario *Use Case* mengelola data laporan transaksi

Nama *Use Case* : Mengelola laporan transaksi

Aktor : *Admin*

Tujuan : Pengelolaan laporan transaksi

Kondisi Awal : *Admin* belum melakukan *login*

Kondisi : *Admin* sudah *login*

Kondisi Akhir : *Admin* berhasil *login*

Deskripsi : Proses mengelola laporan transaksi adalah proses Untuk mengekspor semua data transaksi menjadi Laporan pada sistem, yang meliputi *ekspor* Laporan pdf atau *excel*.

Tabel 3.5 Skenario *Use Case* Mengelola Data Laporan Transaksi

| <i>Admin</i> | Reaksi Sistem |
|-----------------------------------|--|
| 1. Memilih menu laporan | 2. Sistem menampilkan pilihan laporan transaksi |
| 3. Memilih data laporan transaksi | 4. Sistem menampilkan pilihan ekspor laporan PDF |
| 5. Pilih salah satu tombol | 6. Sistem menampilkan data laporan transaksi |
| Kondisi Akhir | Data berhasil diolah |

6) Skenario *Use Case* mengelola data laporan buku

Nama *Use Case* : Mengelola laporan buku

Aktor : *Admin*
 Tujuan : Pengelolaan laporan buku
 Kondisi Awal : *Admin* belum melakukan *login*
 Kondisi : *Admin* sudah *login*
 Kondisi Akhir : *Admin* berhasil *login*
 Deskripsi : Proses mengelola laporan buku adalah proses Untuk mengekspor semua data transaksi menjadi Laporan pada sistem, yang meliputi *ekspor* Laporan pdf, *ekspor* laporan *excel*.

Tabel 3.6 Skenario *Use Case* Mengelola Data Laporan Buku

| <i>Admin</i> | Reaksi Sistem |
|------------------------------|---|
| 1. Memilih menu laporan | 2. Sistem menampilkan pilihan laporan buku |
| 3. Memilih data laporan buku | 4. Sistem menampilkan pilihan <i>ekspor</i> laporan PDF atau <i>ekspor</i> laporan <i>excel</i> |
| 5. Pilih salah satu tombol | 6. Sistem menampilkan data laporan buku. |
| Kondisi Akhir | Data berhasil diolah |

7) Skenario *Use Case* melihat data buku

Nama *Use Case* : Melihat data buku
 Aktor : *Admin*
 Tujuan : Melihat data buku
 Kondisi Awal : *Admin* belum melakukan *login*
 Kondisi : *Admin* sudah *login*
 Kondisi Akhir : *Admin* berhasil *login*
 Deskripsi : Proses untuk dapat melihat data buku

Tabel 3.7 Skenario *Use Case* Melihat data buku

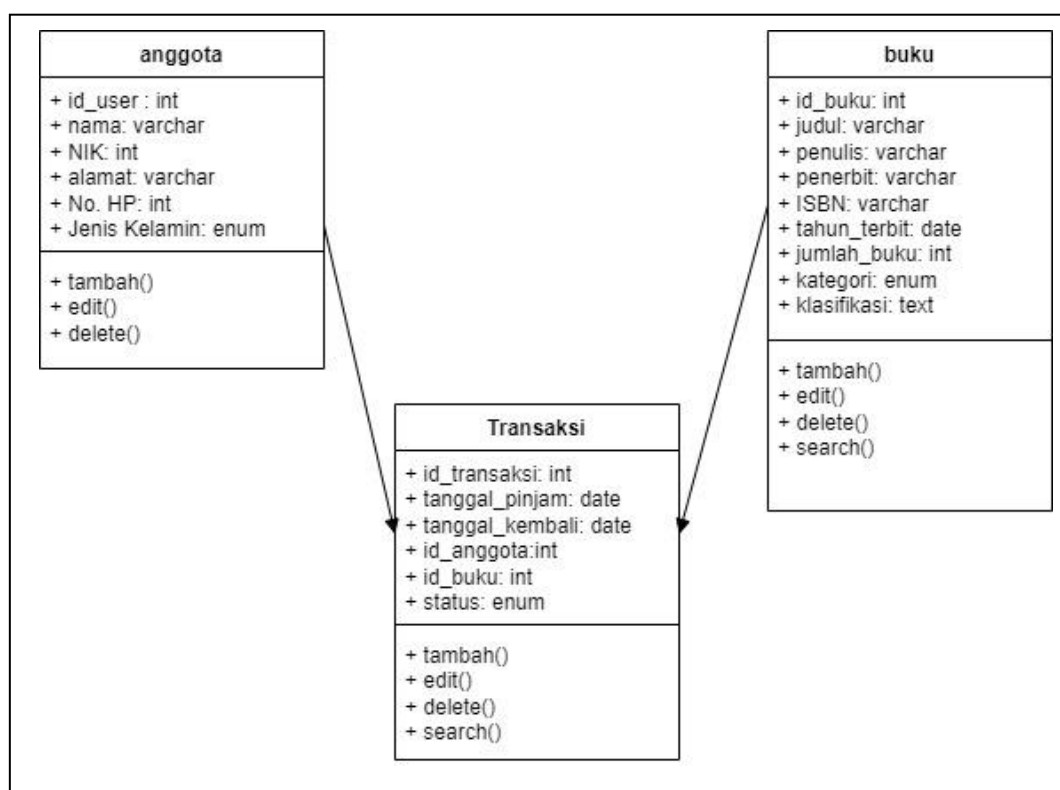
| <i>Admin</i> | Reaksi Sistem |
|-------------------------------|--|
| 1. Memasukkan data judul buku | 2. Menampilkan halaman data buku yang dicari |
| 3. Melihat data buku | 4. Sistem menampilkan data |

| | |
|----------------------|----------------------|
| | buku |
| Kondisi Akhir | Data berhasil diolah |

3. Class Diagram

Diagram kelas (*class diagram*) merepresentasikan kelas-kelas dalam sistem beserta hubungan antar kelas tersebut. *Class diagram* juga menunjukkan *property* dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

Class Diagram dalam penelitian ini dapat dilihat pada gambar 3.6



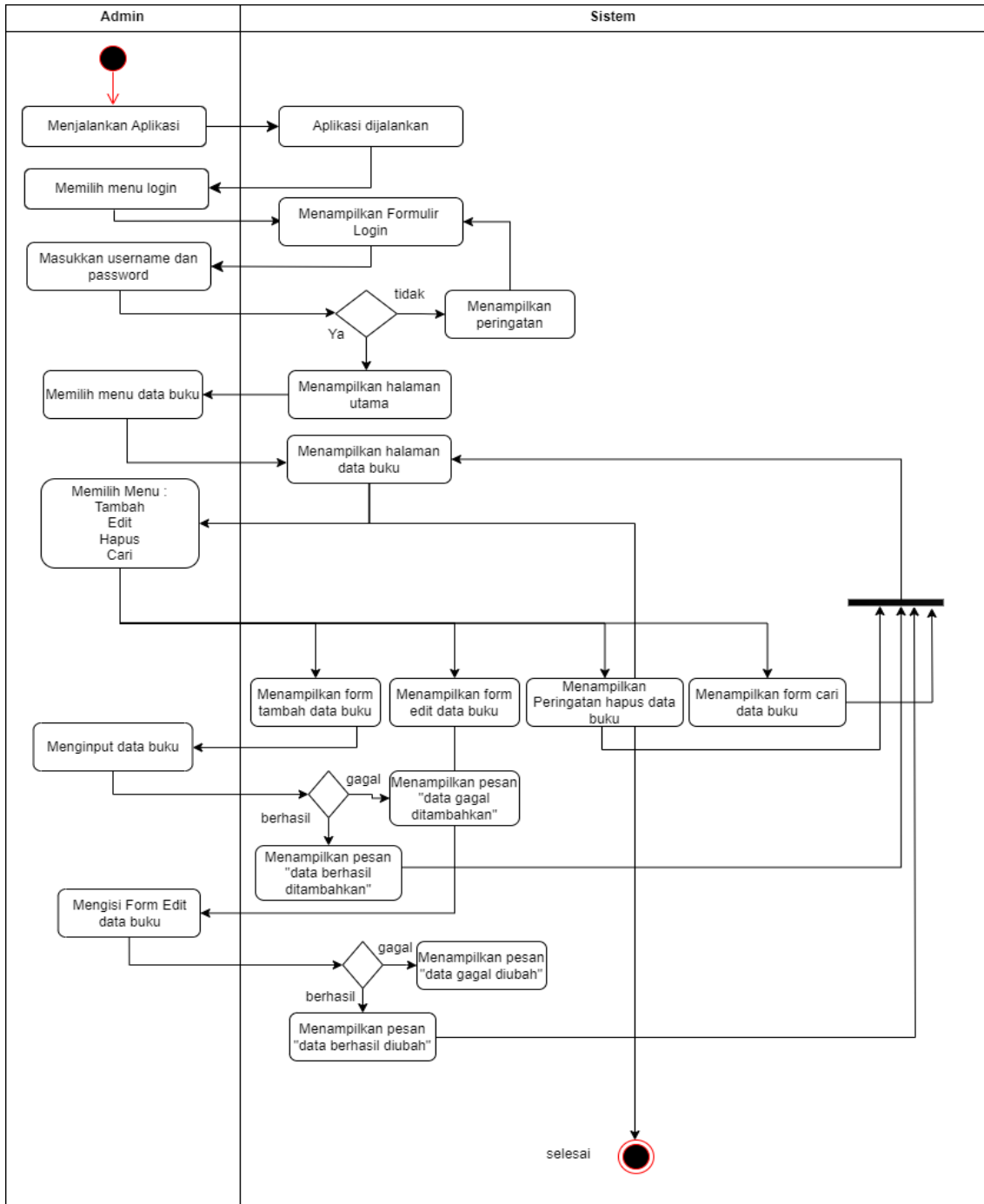
Gambar 3.6 Class Diagram

4. Activity Diagram

Activity diagram menunjukkan aktivitas yang terlibat di pengolahan data atau didalam proses. Diagram ini membantu dalam visualisasi langkah-langkah yang terlibat dalam suatu proses bisnis atau penggunaan sistem, serta bagaimana alur kerja tersebut berpindah dari satu aktivitas ke aktivitas lainnya. Adapun activity diagram yang akan dilaksanakan yaitu:

1. Activity diagram mengelola data buku

Adapun gambar diagram mengelola data buku dapat dilihat pada gambar 3.7.



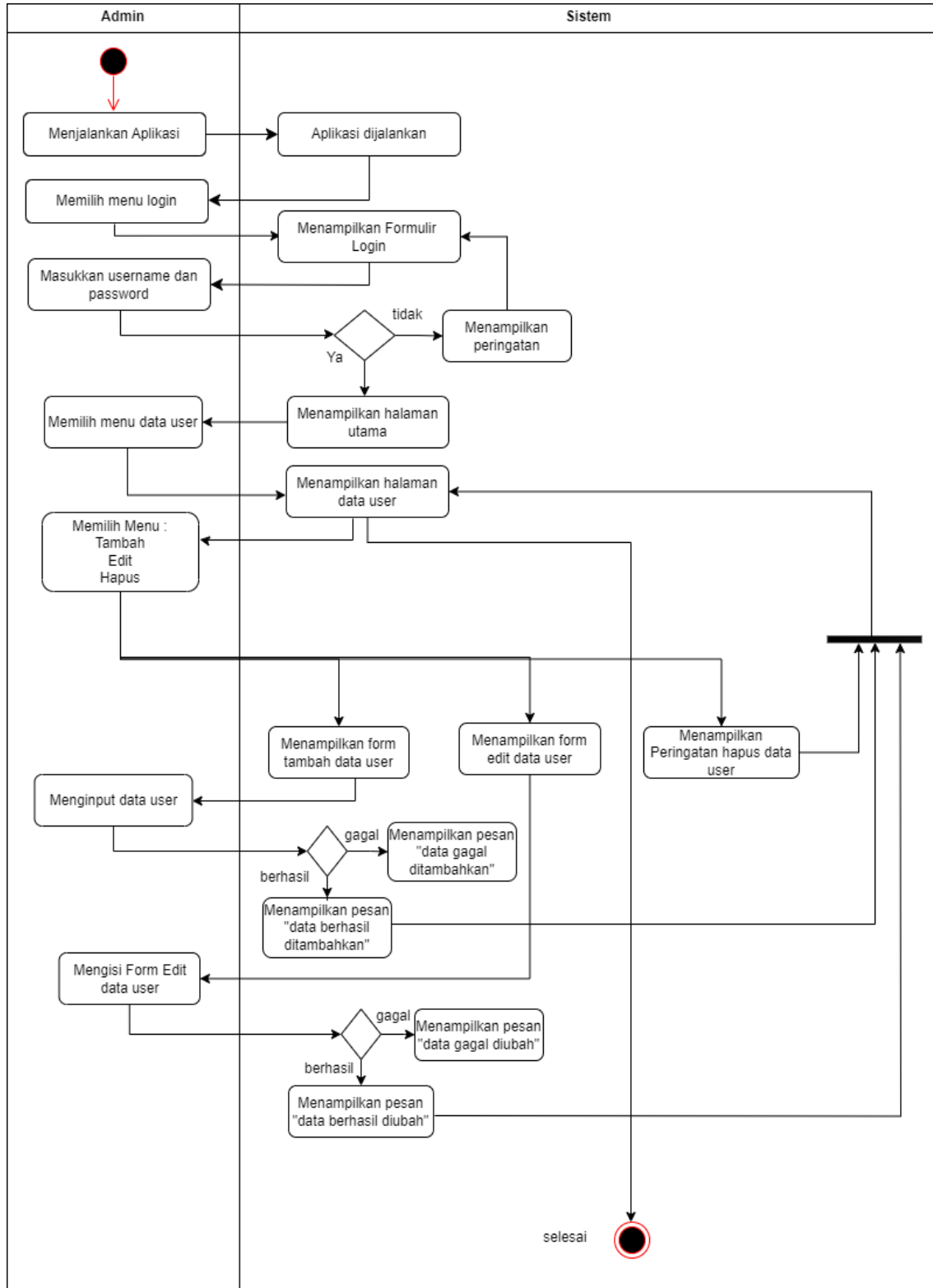
Gambar 3.7 Activity diagram mengelola data buku

2. Activity diagram mengelola data anggota

Gambar Activity diagram mengelola data anggota dapat dilihat pada gambar 3.8.

3. Activity diagram mengelola data user

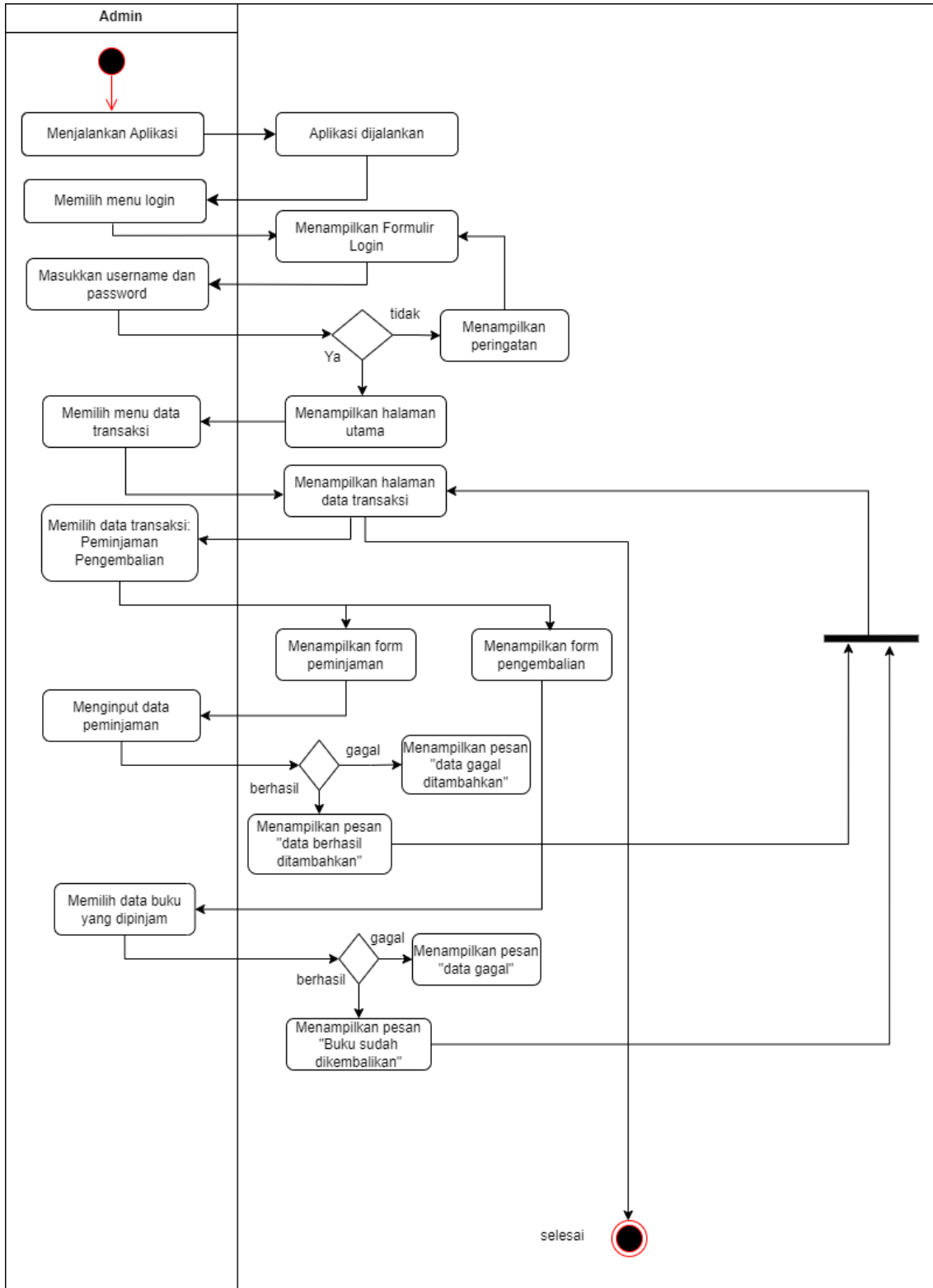
Gambar Activity diagram mengelola data user dapat dilihat pada gambar 3.9 dibawah ini.



Gambar 3.9 Activity diagram mengelola data user

4. Activity diagram mengelola data transaksi

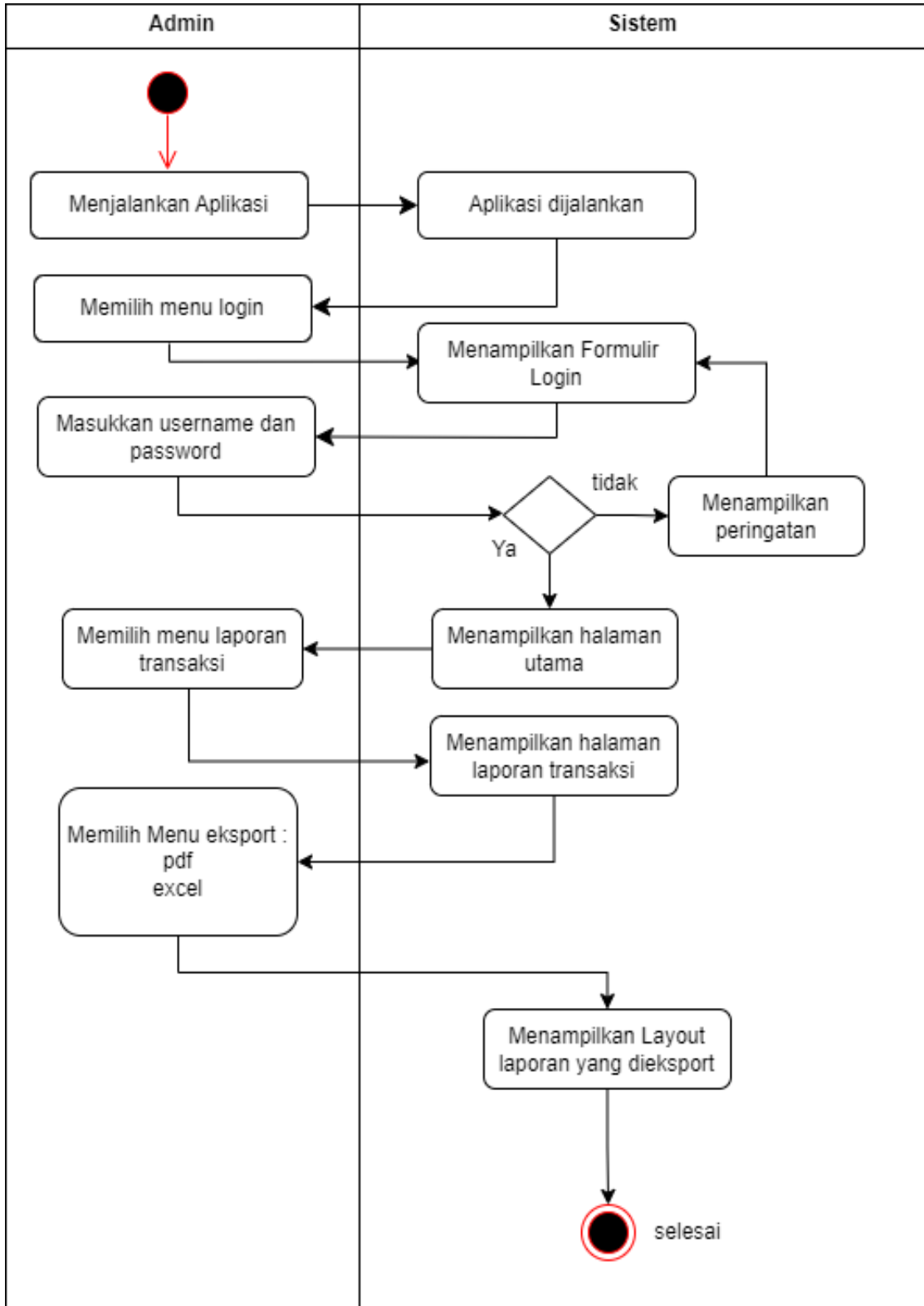
Gambar Activity diagram mengelola data transaksi dapat dilihat pada gambar 3.10 dibawah ini.



Gambar 3.10 Activity diagram mengelola data transaksi

5. Activity diagram mengelola laporan transaksi

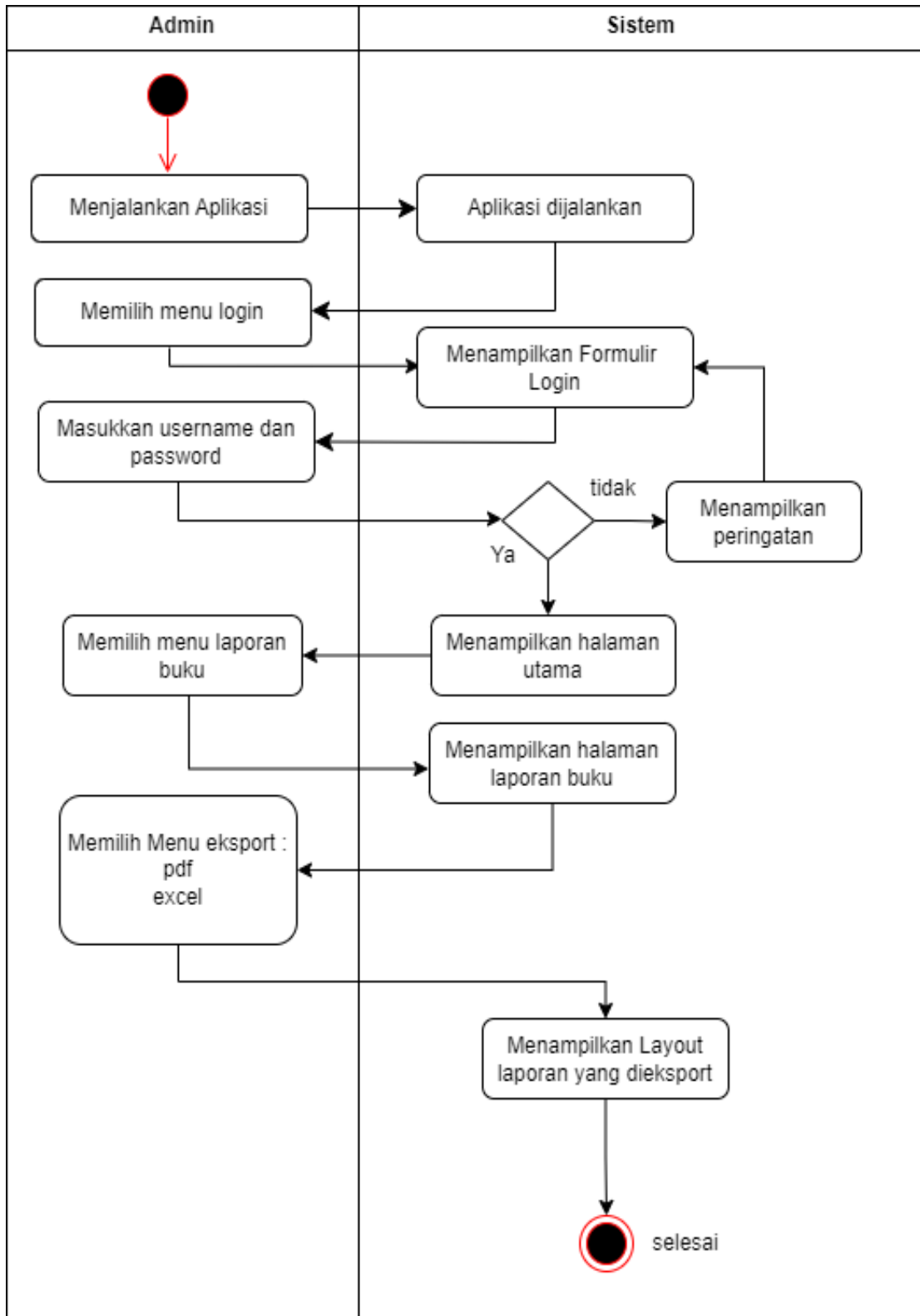
Gambar Activity diagram mengelola laporan transaksi dapat dilihat pada gambar 3.11 dibawah ini.



Gambar 3.11 Activity diagram mengelola laporan transaksi

6. Activity diagram mengelola laporan buku

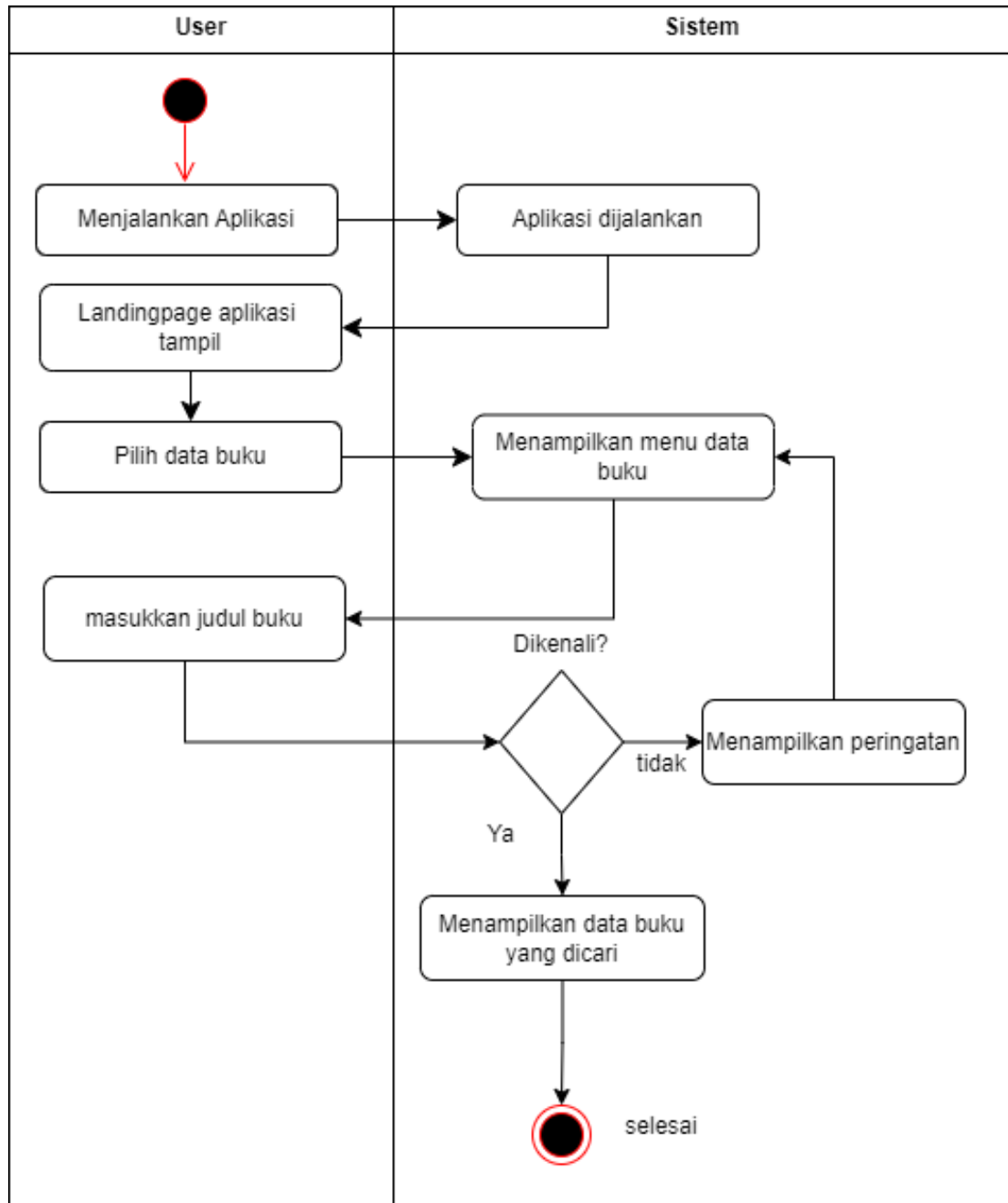
Gambar Activity diagram mengelola lapora buku dapat dilihat pada gambar 3.12 dibawah ini.



Gambar 3.12 Activity diagram mengelola laporan buku

7. Activity diagram melihat data buku

Gambar Activity diagram mengelola laporan transaksi dapat dilihat pada gambar 3.12 dibawah ini.



Gambar 3.13 Activity diagram melihat data buku

3.4.2 Perancangan Database

Perancangan *database* adalah proses yang melibatkan pembuatan struktur logis dari sistem basis data yang akan digunakan untuk menyimpan, mengelola, dan mengakses data. Proses ini melibatkan beberapa langkah penting untuk

memastikan bahwa *database* yang dibuat efektif, efisien, dan sesuai dengan kebutuhan pengguna.

1. Struktur Tabel Buku

Fungsi : Menambah, Menyimpan, Mengedit, Menghapus.

Tabel 3.8 Struktur Tabel Buku

| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | <i>Keterangan</i> |
|--------------|------------------|-------------|--------------------|
| Id_buku | <i>Integer</i> | 10 | <i>Primary Key</i> |
| Judul | <i>Varchar</i> | 255 | Judul Buku |
| Penulis | <i>Varchar</i> | 255 | Penulis Buku |
| Penerbit | <i>Varchar</i> | 255 | Penerbit Buku |
| ISBN | <i>Varchar</i> | 255 | ISBN Buku |
| Tahun_terbit | <i>Date</i> | - | Tahun Terbit Buku |
| Jumlah_buku | <i>Integer</i> | 5 | Jumlah Stok Buku |
| Kategori | <i>Enum</i> | - | Kategori Buku |
| klasifikasi | <i>Varchar</i> | 10 | Klasifikasi Buku |

2. Struktur Tabel *User*

Fungsi : Menambah, Menyimpan, Mengedit, Menghapus.

Tabel 3.9 Struktur Tabel *User*

| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | <i>Keterangan</i> |
|---------------|------------------|-----------------|--------------------|
| Id_user | <i>Integer</i> | 10 | <i>Primary Key</i> |
| Nama | <i>Varchar</i> | 255 | Nama |
| NIK | <i>Varchar</i> | 255 | NIK/NISN |
| Alamat | <i>Varchar</i> | 255 | Alamat User |
| Tanggal_lahir | <i>Date</i> | - | Tanggal Lahir |
| No_hp | <i>Varchar</i> | 13 | No HP |
| Jenis_kelamin | <i>Enum</i> | Pria, Wanita | Jenis Kelamin |

3. Struktur Tabel Transaksi

Fungsi : Menambah, Menyimpan, Mengedit, Menghapus.

Tabel 3.10 Sturuktur Tabel Transaksi

| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | Keterangan |
|--------------------|------------------|--------------------|-----------------------|
| Id_transaksi | <i>Integer</i> | 10 | <i>Primary Key</i> |
| Id_user | <i>Integer</i> | 10 | Id User |
| Id_buku | <i>Integer</i> | 10 | Id Buku |
| Tanggal_peminjaman | <i>Date</i> | - | Tanggal peminjaman |
| Tanggal_kembali | <i>Date</i> | - | Tanggal Pengembalian |
| Status | <i>Enum</i> | Pinjam, kembali | Status Transaksi Buku |

3.5 Rancangan *User Interface* (Antarmuka Pengguna)

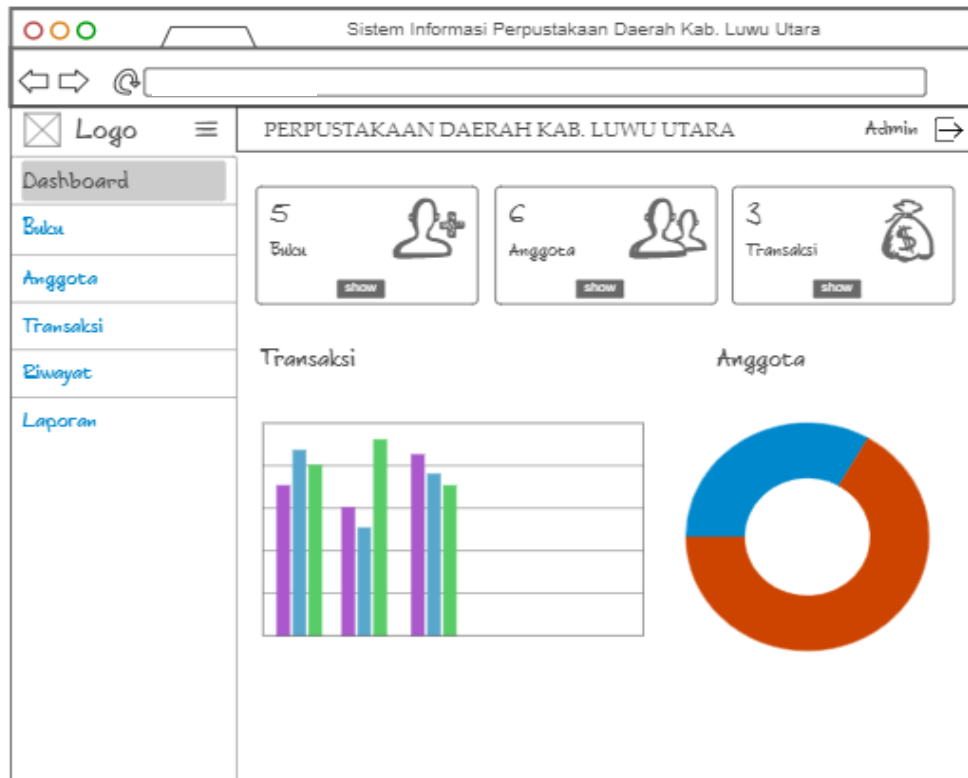
Rancangan *User Interface* (UI) atau antarmuka pengguna adalah proses mendesain tampilan dan interaksi antara pengguna dan sistem. Tujuan utama dari rancangan UI adalah menciptakan pengalaman pengguna yang efektif, efisien, dan menyenangkan. Proses ini melibatkan beberapa langkah penting yang memastikan bahwa antarmuka tidak hanya menarik secara visual tetapi juga fungsional dan mudah digunakan. Desain antarmuka dapat dilihat pada gambar di bawah ini.

1. Halaman *Login*



Gambar 3.14 Halaman *Login*

2. Halaman Utama



Gambar 3.15 Halaman *Dashboard*

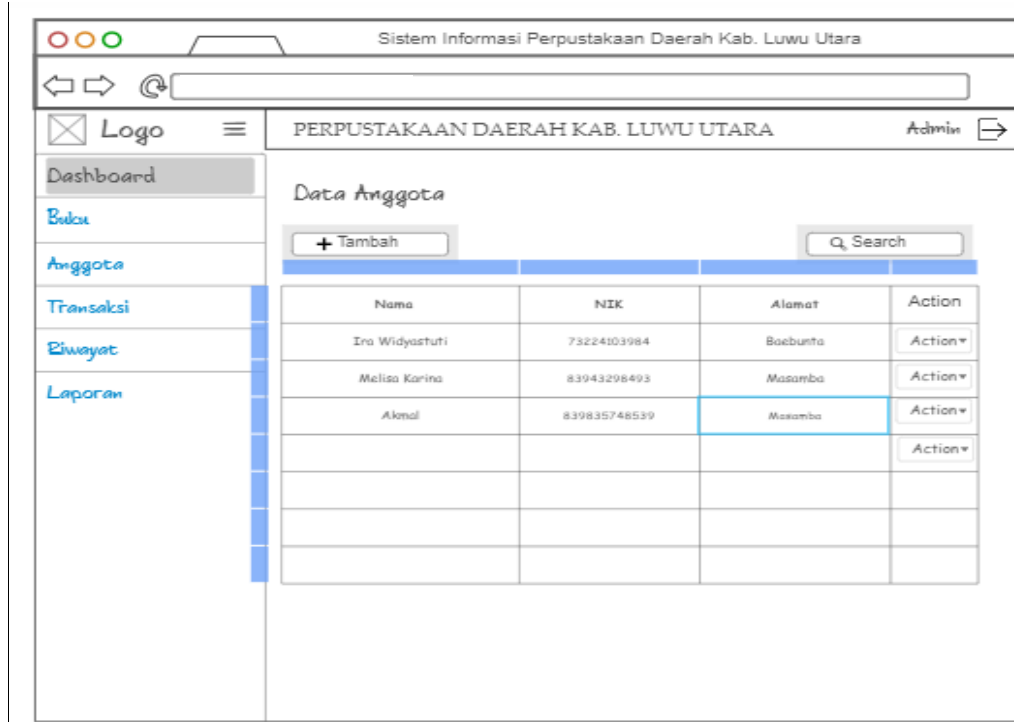
3. Halaman Buku

The screenshot shows a web browser window with the title 'Sistem Informasi Perpustakaan Daerah Kab. Luwu Utara'. The page header includes the logo, the name of the library, and the user 'Admin'. The sidebar on the left lists navigation options: Dashboard, Buku, Anggota, Transaksi, Ewayat, and Laporan. The main content area displays a table titled 'Data Buku' with a '+ Tambah' button and a search bar. The table has four columns: Judul Buku, ISBN, Penulis, and Action. The table contains five rows of book information.

| Judul Buku | ISBN | Penulis | Action |
|-----------------------------|-------------------|-------------------------------|---------|
| Kisah : Zaman Jahiliah | 978-602-6633-3 | Safina | Action▼ |
| Amazing Science from Qur'an | 978-603-317-829-9 | Gina Sania | Action▼ |
| Fikih Penempuan | 979-9392-23-3 | Syaikh Mutaawalli As-sya'rani | Action▼ |
| Serpihan Rindu | 978-602-398-028-4 | Eembart | Action▼ |
| | | | |
| | | | |

Gambar 3.16 Halaman *Data Buku*

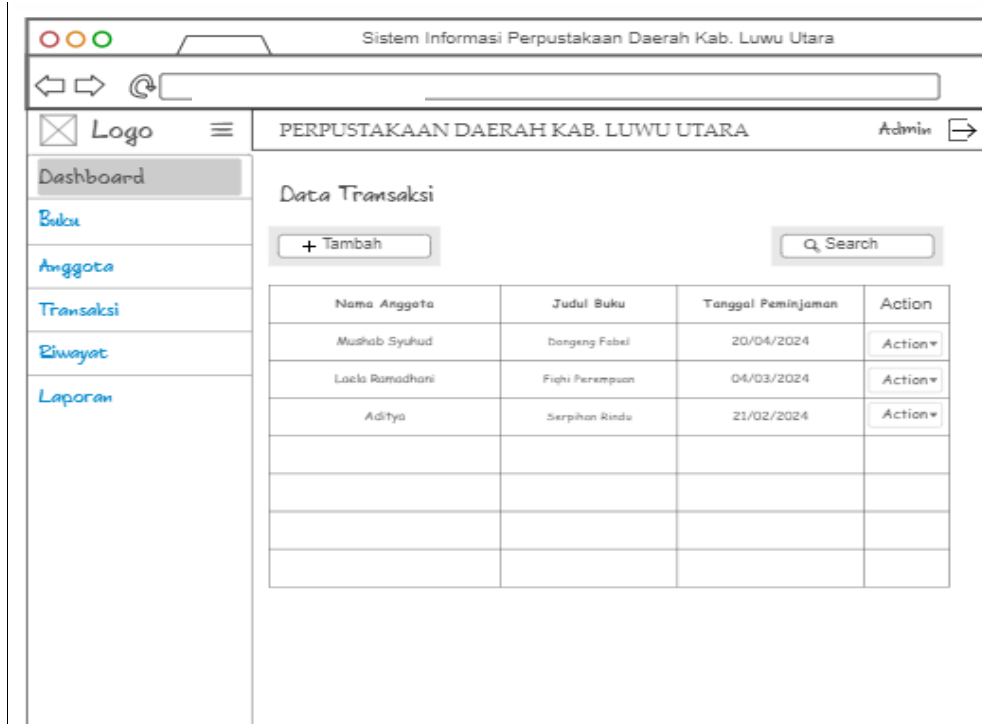
4. Halaman Anggota



Gambar 3.17 Halaman Data Anggota

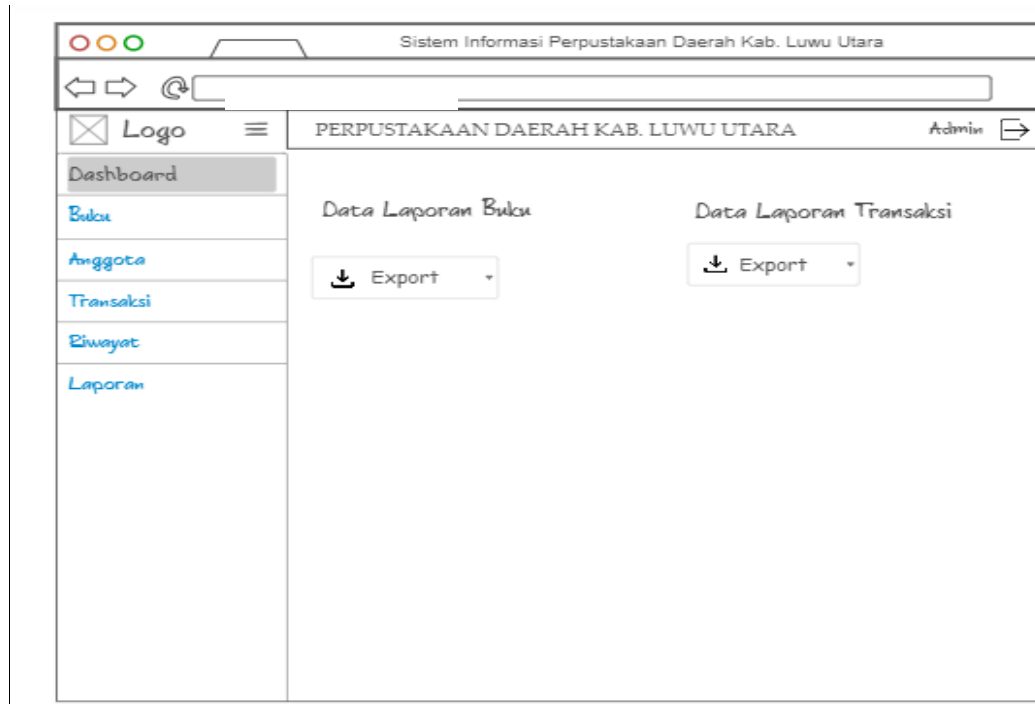
5. Halaman Data Transaksi

Halaman data transaksi terdiri dari halaman peminjaman buku dan halaman pengembalian buku



Gambar 3.18 Halaman Data Transaksi

6. Halaman Laporan



Gambar 3.19 Halaman Laporan

3.6 Rancangan Pengujian Sistem

Pengujian sistem merupakan hal yang penting dan bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak. Pengujian sistem yang digunakan yaitu menggunakan *Black Box Testing* dan *White Box Testing*.

3.6.1 Rancangan Pengujian *Black Box*

Pengujian menggunakan *Black Box* digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Berikut ini adalah tabel rencana pengujian *black box*.

Tabel 3.11 Tabel Perancangan *Black Box Testing*

| No | Komponen yang diuji | Butir Uji | Jenis Pengujian |
|----|---------------------|-------------------------|------------------|
| 1 | Fungsi menu buku | Menambah buku baru | <i>Black Box</i> |
| | | Mengedit Informasi buku | <i>Black Box</i> |
| | | Menghapus buku | <i>Black Box</i> |

| | | | |
|---|-----------------------|--|------------------|
| 2 | Fungsi Menu Anggota | Menambah anggota baru | <i>Black Box</i> |
| | | Mengedit informasi anggota | <i>Black Box</i> |
| | | Menghapus anggota | <i>Black Box</i> |
| 3 | Fungsi Menu Transaksi | Melakukan Peminjaman buku | <i>Black Box</i> |
| | | Melakukan Pengembalian Buku | <i>Black Box</i> |
| 4 | Fungsi Menu Riwayat | Melihat riwayat transaksi | <i>Black Box</i> |
| 5 | Fungsi Menu Laporan | Melakukan <i>Eksport</i> Laporan Buku | <i>Black Box</i> |
| | | Melakukan <i>Eksport</i> Laporan Transaksi | <i>Black Box</i> |

3.6.2 Rancangan Pengujian *White Box*

Pengujian *white box* berfokus pada pengujian struktur internal kode sumber. Dalam konteks implementasi *framework laravel* untuk sistem informasi perpustakaan. Adapun metode pengujian sistemnya yaitu:

1. Flow graph
2. Cyclomatic Complexity
3. Graph matrix
4. Pengujian jalur independent path dan test case

BAB IV

IMPLEMENTASI

4.1 Implementasi

Implementasi merupakan tahap dalam pengembangan proyek di mana rencana, desain, dan spesifikasi yang telah dibuat sebelumnya diterapkan ke dalam sistem nyata. Pada tahap ini, konsep dan desain diubah menjadi bentuk operasional yang dapat digunakan oleh pengguna akhir. Implementasi mencakup berbagai aspek seperti pengkodean, pengujian, dan pengaturan sistem.

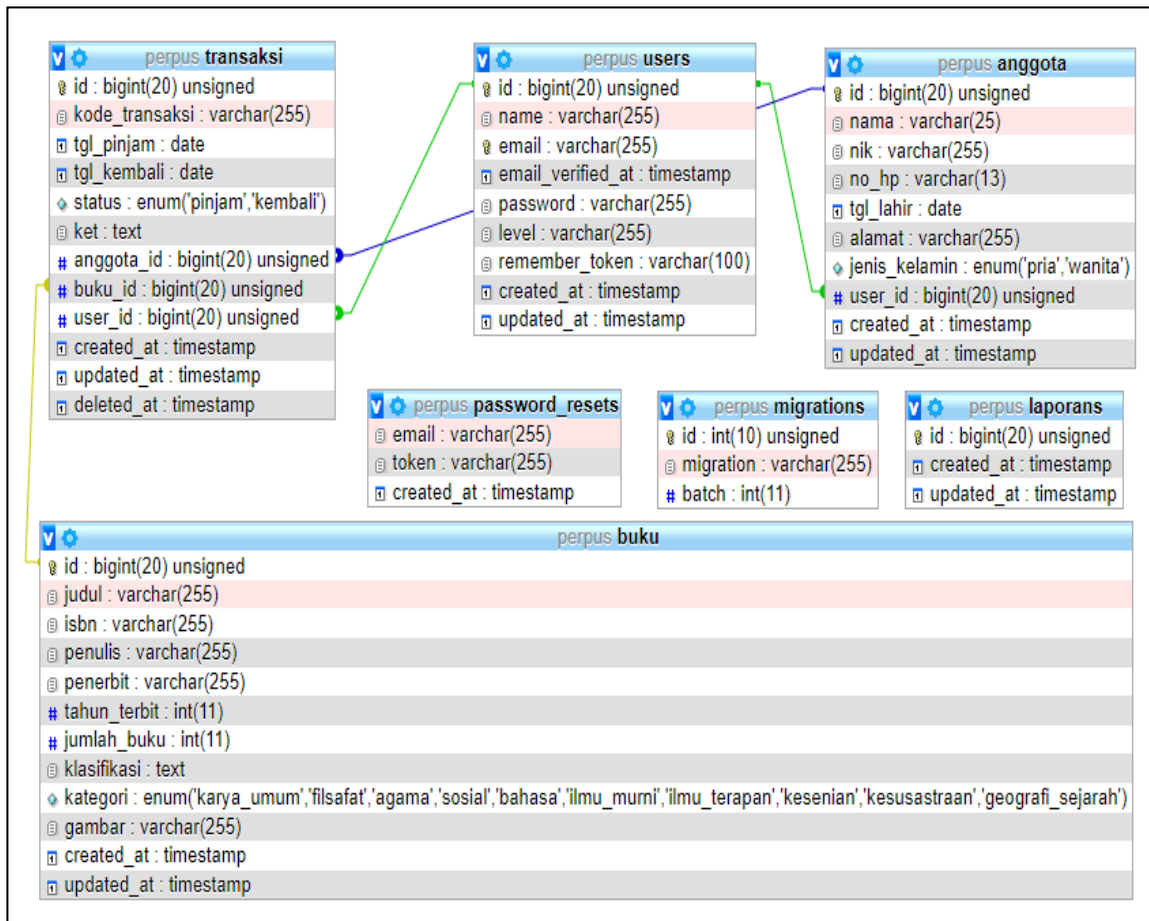
4.1.1 Implementasi Sistem

Implementasi sistem merupakan tahap penting dalam pengembangan perangkat lunak dan proyek teknologi informasi, di mana rencana, desain, dan spesifikasi yang telah disusun diterapkan ke dalam sistem operasional yang dapat digunakan oleh pengguna akhir. Proses ini mencakup berbagai aktivitas yang bertujuan untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan dan ekspektasi pengguna. Spesifikasi *Hardware* dan *software* yang digunakan dalam proses implementasi sistem yaitu:

1. Spesifikasi *Hardware*:
 - a. *Processor Intel Core i5 VPro-inside*
 - b. *Memory (RAM) 8 GB*
 - c. *Harddisk 256 GB*
2. Spesifikasi *Software*:
 - a. *Sistem Operasi Windows 10*
 - b. *Framework Laravel 7*
 - c. *XAMPP Versi 7.4.1*
 - d. *Composer Versi 2.7.6*
 - e. *Pengkodean menggunakan Visual Studio Code*
 - f. *Browser untuk menjalankan program yaitu Google Chrome.*

4.1.2 Implementasi Database

Implementasi *database* merupakan proses mendetail yang mengubah desain *database* menjadi sistem operasional yang siap digunakan untuk menyimpan dan mengelola data. Proses ini mencakup beberapa tahap, mulai dari perencanaan hingga pemeliharaan berkelanjutan. Adapun *database* dalam sistem informasi ini dapat dilihat dari gambar 4.1



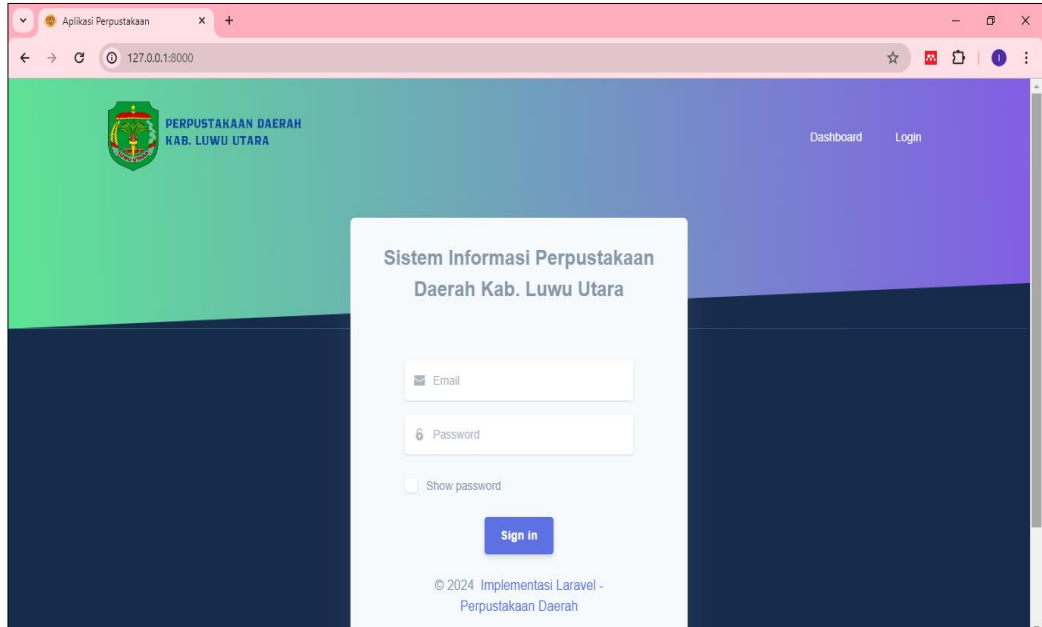
Gambar 4.1 Database

4.1.3 Implementasi Program

Implementasi program merupakan tahap dalam pengembangan perangkat lunak di mana rancangan dan spesifikasi yang telah dibuat sebelumnya diterapkan menjadi aplikasi atau sistem yang siap digunakan oleh pengguna akhir. Berikut ini adalah implementasi program pada sistem informasi Perpustakaan Daerah Kab. Luwu Utara sebagai berikut:

1. Tampilan Halaman *Login*

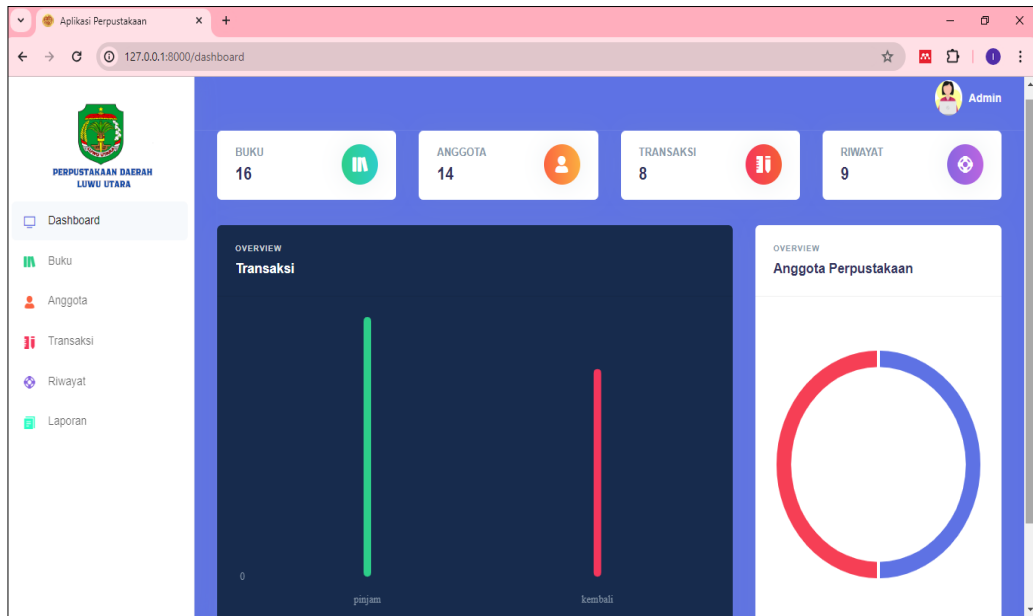
Halaman *login* adalah halaman tampilan awal ketika masuk ke dalam program sistem informasi Perpustakaan. Berikut halaman login admin ditunjukkan oleh gambar 4.2.



Gambar 4.2 Tampilan Halaman *Login*

2. Tampilan Halaman *Dashboard*

Tampilan halaman ini adalah halaman utama/*dashboard* yang akan muncul ketika *user/admin* berhasil *login*. Dalam tampilan ini terdapat beberapa menu yang berada disebelah kiri tampilan. Sedangkan disebelah kanan tampilan, terdapat nama *user/admin*. *User* juga dapat melakukan *logout* dengan melakukan *klik* pada tulisan nama tersebut untuk keluar dari aplikasi.



Gambar 4.3 Tampilan Halaman *Dashboard*

3. Tampilan Halaman Buku

Tampilan halaman buku ini berfungsi untuk memasukkan data buku koleksi Perpustakaan Daerah Kab. Luwu Utara. Dalam halaman ini terdapat beberapa *action* seperti tambah data, melihat *detail* buku, cari data buku, *edit* data buku dan hapus data buku.

The page features a search bar with the placeholder text "masukkan pencarian" and a "Tambah Buku" button. The main content is a table titled "Daftar Buku" with the following data:

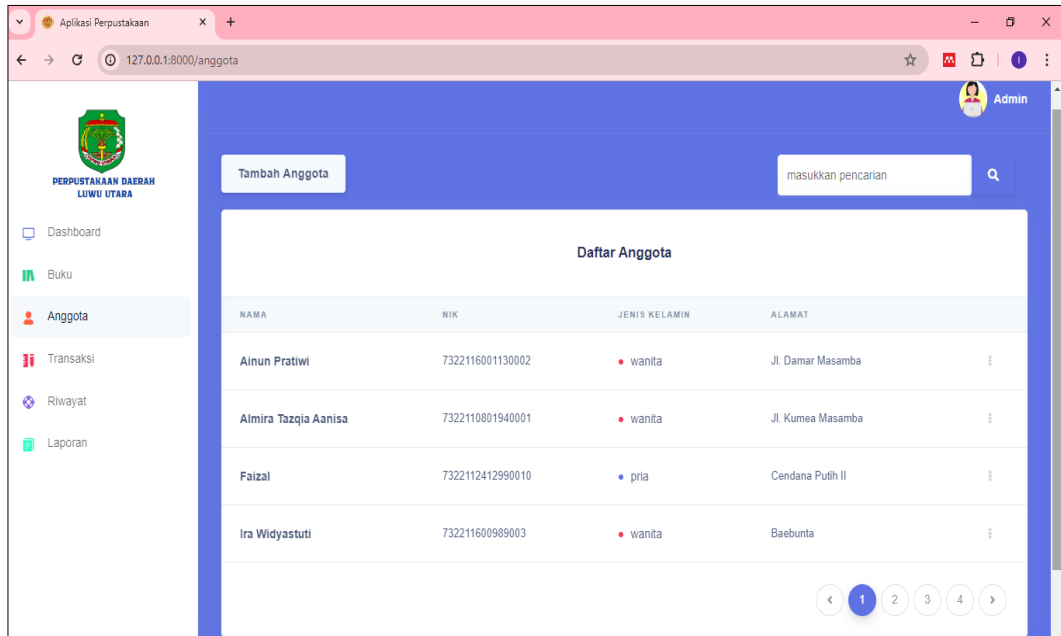
| JUDUL | PENULIS | PENERBIT | TAHUN TERBIT |
|----------------------------|------------------|------------------|--------------|
| Prasasti cinta | Aster Putih | Rumah Oranye | 2014 |
| Ibu Doa Yang Hilang | Bagas D. Bawono | Zettu | 2015 |
| Seni mengenal diri sendiri | Brian Adam | Bright Publisher | 2021 |
| Dibalik Tragedi | Eidelweis Almira | Zettu | 2014 |

At the bottom of the table, there is a pagination control showing page 1 of 4.

Gambar 4.4 Tampilan Halaman Buku

4. Tampilan Halaman Anggota

Tampilan halaman anggota ini digunakan untuk mengolah dan memproses data anggota. Dalam halaman ini terdapat beberapa *action* seperti Tambah anggota, cari data anggota, *edit* data anggota dan hapus data anggota.



Gambar 4.5 Tampilan Halaman Anggota

5. Tampilan Halaman *Transaksi*

Halaman transaksi berfungsi untuk menampilkan seluruh *transaksi* buku. Dalam halaman ini terdapat tombol *action* untuk menambahkan *transaksi*, cari transaksi, melihat transaksi buku dan *edit* transaksi. *Menu edit transaksi* terdapat pilihan *status*, dalam pilihan *status* kita dapat mengubahnya dengan *statusnya* kembali apabila buku yang dipinjamkan sudah dikembalikan.

Daftar Transaksi

| NAMA | NIK | KODE TRANSAKSI | TANGGAL PINJAM | TANGGAL KEMBALI | STATUS |
|------------------|------------------|----------------|----------------|-----------------|---------|
| Salwa | 732211031100001 | CiqFXWmZ | 2024-05-29 | 2024-06-05 | pinjam |
| Moh. Aqtar Saman | 732211803080004 | kiUqLSnJd | 2024-06-04 | 2024-06-11 | pinjam |
| Faizal | 732211241269010 | HgkwckFul | 2024-06-03 | 2024-06-10 | pinjam |
| Nuraini | 7322116911070005 | OpFvV9Uht | 2024-05-30 | 2024-06-07 | pinjam |
| Maghfira Juanda | 732111902970003 | 2V4YCJVFY | 2024-06-04 | 2024-06-11 | pinjam |
| M. Ibnu Habibi | 7322110408990006 | o8W6JVV4Z | 2024-05-12 | 2024-05-20 | kembali |

Gambar 4.6 Tampilan Halaman Transaksi

6. Tampilah Halaman Riwayat

Halaman riwayat berisi tentang riwayat *transaksi* yang dilakukan, seperti *transaksi* peminjaman dan *transaksi* pengembalian buku.

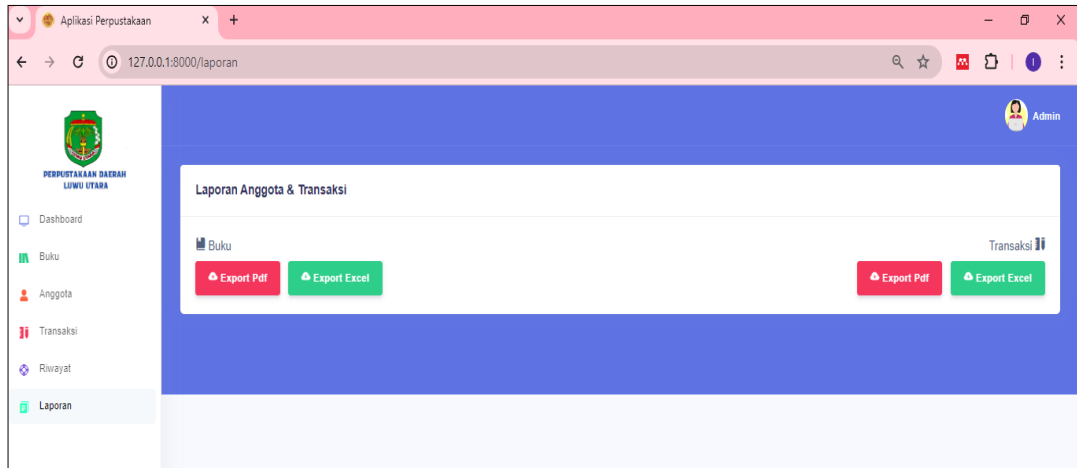
Riwayat Peminjaman

| NAMA | NIK | BUKU | TANGGAL PINJAM | TANGGAL KEMBALI | STATUS |
|---------------------|------------------|-------------------------------------|----------------|-----------------|---------|
| Zea | 123456789 | Metode Penelitian | 2023-12-15 | 2023-12-16 | kembali |
| Ainun Pratiwi | 7322118001130002 | Dibalik Tragedi | 2024-05-20 | 2024-05-28 | kembali |
| Almira Tazqia Anisa | 7322110801040001 | Push Panduan Berleksi dan Berbisnis | 2024-04-29 | 2024-05-13 | kembali |
| M. Ibnu Habibi | 7322110408990006 | Smart way to Toefl | 2024-05-12 | 2024-05-20 | kembali |
| Maghfira Juanda | 732111902970003 | Long Lost yang lama hilang | 2024-06-04 | 2024-06-11 | pinjam |
| Nuraini | 7322116911070005 | Ibu Doa yang Hilang | 2024-05-30 | 2024-06-07 | pinjam |
| Faizal | 732211241269010 | Seni mengenal diri sendiri | 2024-06-03 | 2024-06-10 | pinjam |
| Moh. Aqtar Saman | 732211803080004 | Dunia Sukabo Sejumlah Cerita | 2024-06-04 | 2024-06-11 | pinjam |
| Salwa | 732211031100001 | Cinta Bertabur dilangit Mekkah | 2024-05-29 | 2024-06-05 | pinjam |

Gambar 4.7 Tampilan Halaman Riwayat

7. Tampilan Halaman Laporan

Halaman laporan terdiri dari 2 laporan yaitu Laporan buku dan laporan transaksi yang telah dilaksanakan. Di halaman ini terdapat tombol untuk *eksport file* dalam bentuk *pdf* dan *excel*.



Gambar 4.8 Tampilan Halaman Laporan

4.2 Hasil Pengujian

4.2.1 Hasil Pengujian *Black Box*

Pengujian *black box* merupakan *metode* pengujian yang dimana peniliannya terhadap sebuah aplikasi bukan terletak pada spesifikasi logika atau fungsi aplikasi tersebut, tetapi *input* dan *output*. Dengan berbagai *input* yang diberikan akan dievaluasi apakah suatu aplikasi atau sistem dapat memberikan *output* yang sesuai dengan harapan penguji. Berikut ini merupakan hasil dari pengujian *black box* ditunjukkan oleh tabel berikut:

Tabel 4.1 Pengujian *Black Box*

| No | Kasus Uji | Langkah Uji | Hasil yang diharapkan | Hasil Aktual | Keterangan |
|----|-------------------|--|-------------------------------|------------------------|------------|
| 1 | Menu <i>Login</i> | <i>Run</i> program, muncul <i>form login</i> | Menampilkan <i>form login</i> | Tampilan halaman utama | Berhasil |

| | | | | | |
|---|---------------------|-------------------------------|--|----------------------------------|----------|
| 2 | Menu data buku | <i>Klik</i> menu data buku | Menampilkan <i>form</i> data buku | Tampilan data buku | Berhasil |
| 3 | Menu data anggota | <i>Klik</i> menu data anggota | Menampilkan <i>form</i> data anggota | Tampilan data anggota | Berhasil |
| 4 | Menu data transaksi | <i>Klik</i> Menu transaksi | Menampilkan <i>form</i> data transaksi | Tampilan data transaksi | Berhasil |
| 5 | Menu laporan | <i>Klik</i> Menu laporan | Menampilkan <i>form</i> laporan | Tampilan <i>ekspor</i> transaksi | Berhasil |

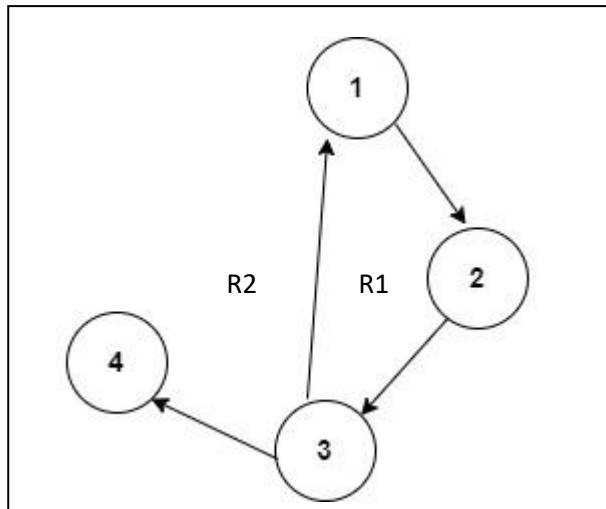
4.2.2 Hasil Pengujian *White Box*

Untuk pengujian sistem ini menggunakan *white box testing* untuk menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan.

1. Pengujian *Route form login*

Route ini memiliki fungsi untuk menampilkan halaman *login*. Selanjutnya akan dibuatkan *flow graph* berdasarkan *kode route form login* sebagai berikut:

- 1) *Input user dan password*
- 2) *Tombol Login*
- 3) *Authentifikasi*
- 4) *Dashboard*



Gambar 4.9 *Flow graph login*

Berdasarkan *flow graph* pada gambar 4.9 diketahui bahwa jumlah *edge* grafik alir (E) = 4 yang merupakan garis yang menghubungkan *node*, jumlah *node* (N) = 4 yang merupakan lingkaran yang menggambarkan suatu aktivitas. Jumlah Predicate (P) = 1 yang merupakan *node* bercabang, dan jumlah *region* (R) = 2 yang menandakan suatu area dalam *flow graph*, yang dapat dilihat dengan simbol R1 dan R2 pada gambar 4.9. Sehingga rumus perhitungan Cyclomatic complexity dan juga bantuan *graph matrix* adalah:

Tabel 4.2 *Cyclomatic Complexity Route Form Login*

| | | |
|--------------------|----------------|------------|
| $V(G) = E - N + 2$ | $V(G) = P + 1$ | $V(G) = R$ |
| $V(G) = 4 - 4 + 2$ | $V(G) = 1 + 1$ | $V(G) = 2$ |
| $V(G) = 2$ | $V(G) = 2$ | |

Tabel 4.3 *Graph Matrix Route Form Login*

| Node | 1 | 2 | 3 | 4 | Hasil |
|--------|---|---|---|---|-------|
| 1 | | 1 | | | 0 |
| 2 | | | 1 | | 0 |
| 3 | 1 | | | 1 | 1 |
| 4 | | | | | 0 |
| Jumlah | | | | | 2 |

Berdasarkan hasil perhitungan *kompleksitas siklomatis* diatas sehingga didapatkan jumlah hasil *independent path* untuk *route login* yaitu 2 dengan jalur *independent path* sebagai berikut:

- Jalur 1 : 1 – 2 – 3 – 4 Masuk halaman *Dashboard*
- Jalur 2 : 1 – 2 – 3 – 1 Gagal *Login*

Diketahui bahwa *route form login* memiliki 2 jalur. Setelah menentukan jalur *independent path* maka langkah selanjutnya yaitu membuat dan *test case* dan melaksanakan pengujian. Hasil pengujian dapat dilihat pada tabel berikut:

Tabel 4.4 Pengujian Jalur *Login*

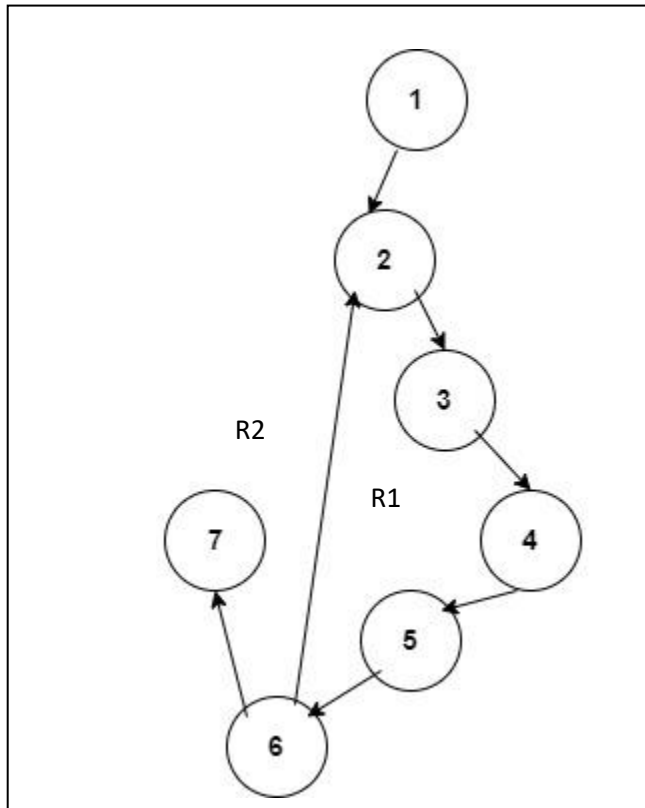
| Jalur | Input | Proses | Hasil | Ket. |
|---------|----------------------------------|----------------------------------|----------------------------------|----------|
| 1-2-3-4 | Memasukkan username dan password | Authentifikasi User dan password | Masuk ke halaman dashboard | Berhasil |
| 1-2-3-1 | Memasukkan username dan password | Authentifikasi User dan password | Muncul pesan user/password salah | Berhasil |

Berdasarkan keseluruhan hasil dari pengujian yang telah dilakukan sehingga menghasilkan nilai CC 2 adalah sehingga menunjukkan tipe prosedur sederhana yang memiliki tingkat resiko akan cacat atau error adalah rendah.

2. Pengujian tambah buku

Route ini memiliki fungsi untuk menambah data buku. Selanjutnya akan dibuatkan *flow graph* berdasar *kode route form* tambah buku sebagai berikut:

- 1)Klik menu Buku
- 2)View daftar buku
- 3)Tambah buku
- 4)Input data buku
- 5)Pilih simpan buku
- 6)Authentifikasi inputan buku
- 7)Data buku



Gambar 4.10 *Flow graph* tambah buku

Berdasarkan *flow graph* pada gambar 4.10 diketahui bahwa jumlah *edge* grafik alir (E) = 7 yang merupakan garis yang menghubungkan *node*, jumlah *node* (N) = 7 yang merupakan lingkaran yang menggambarkan suatu aktivitas. Jumlah *Predicate* (P) = 1 yang merupakan *node* bercabang, dan jumlah *region* (R) = 2 yang menandakan suatu area dalam *flow graph*, yang dapat dilihat dengan simbol R1 dan R2 pada gambar 4.10. Sehingga rumus perhitungan *Cyclomatic complexity* dan juga bantuan *graph matrix* adalah:

Tabel 4.5 *Cyclomatic Complexity Route Form Login*

| | | |
|--------------------|----------------|------------|
| $V(G) = E - N + 2$ | $V(G) = P + 1$ | $V(G) = R$ |
| $V(G) = 7 - 7 + 2$ | $V(G) = 1 + 1$ | $V(G) = 2$ |
| $V(G) = 2$ | $V(G) = 2$ | |

Tabel 4.6 *Graph Matrix Route Form* tambah data buku

| Node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Hasil |
|------|---|---|---|---|---|---|---|-------|
| 1 | | 1 | | | | | | 0 |
| 2 | | | 1 | | | | | 0 |

| | | | | | | | | |
|---------------|--|---|--|---|---|---|---|----------|
| 3 | | | | 1 | | | | 0 |
| 4 | | | | | 1 | | | 0 |
| 5 | | | | | | 1 | | 0 |
| 6 | | 1 | | | | | 1 | 1 |
| 7 | | | | | | | | 0 |
| Jumlah | | | | | | | | 2 |

Berdasarkan hasil perhitungan *kompleksitas siklomatis* diatas sehingga didapatkan jumlah hasil *independent path* untuk *route* tambah data buku yaitu 2 dengan jalur *independent path* sebagai berikut:

- Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 Tambah buku
- Jalur 2 : 1 – 2 – 3 – 4 – 6 – 2 Gagal tambahkan buku

Diketahui bahwa *route form login* memiliki 2 jalur. Setelah menentukan jalur *independent path* maka langkah selanjutnya yaitu membuat dan *test case* dan melaksanakan pengujian. Hasil pengujian dapat dilihat pada tabel berikut:

Tabel 4.7 Pengujian Jalur tambah data buku

| Jalur | Input | Proses | Hasil | Ket. |
|-------------------|-----------------|-------------------------------|---------------------------------|----------|
| 1-2-3-4- 5-6-7 | Input data buku | Validasi data inputan buku | Muncul Pesan Input berhasil | Berhasil |
| 1-2-3-4- 6-2 | Input data buku | Validasi data inputan buku | Kembali ke Menu data buku | Berhasil |

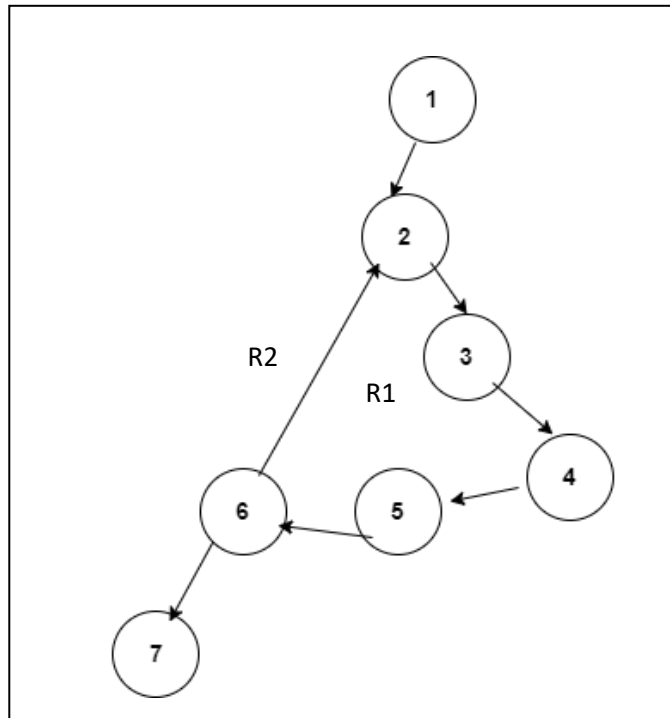
Berdasarkan keseluruhan hasil dari pengujian yang telah dilakukan sehingga menghasilkan nilai CC adalah 2 sehingga menunjukkan tipe prosedur sederhana yang memiliki tingkat resiko akan cacat atau error adalah rendah.

3. Pengujian tambah anggota

Route ini memiliki fungsi untuk menambah data anggota. Selanjutnya akan dibuatkan *flow graph* berdasrkan *kode route form* tambah buku sebagai berikut:

- 1)Klik menu Anggota
- 2)View daftar Anggota

- 3) Tambah Anggota
- 4) Input data Anggota
- 5) Pilih simpan
- 6) Autentifikasi inputan data Anggota
- 7) Data buku



Gambar 4.11 *Flow graph* tambah anggota

Berdasarkan *flow graph* pada gambar 4.11 diketahui bahwa jumlah *edge* grafik alir (E) = 7 yang merupakan garis yang menghubungkan *node*, jumlah *node* (N) = 7 yang merupakan lingkaran yang menggambarkan suatu aktivitas. Jumlah *Predicate* (P) = 1 yang merupakan *node* bercabang, dan jumlah *region* (R) = 2 yang menandakan suatu area dalam *flow graph*, yang dapat dilihat dengan simbol R1 dan R2 pada gambar 4.11. Sehingga rumus perhitungan *Cyclomatic complexity* dan juga bantuan *graph matrix* adalah:

Tabel 4.8 *Cyclomatic Complexity Route Form* tambah anggota

| | | |
|--------------------|----------------|------------|
| $V(G) = E - N + 2$ | $V(G) = P + 1$ | $V(G) = R$ |
| $V(G) = 7 - 7 + 2$ | $V(G) = 1 + 1$ | $V(G) = 2$ |
| $V(G) = 2$ | $V(G) = 2$ | |

Tabel 4.9 *Graph Matrix Route Form* tambah data anggota

| Node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Hasil |
|---------------|---|---|---|---|---|---|---|----------|
| 1 | | 1 | | | | | | 0 |
| 2 | | | 1 | | | | | 0 |
| 3 | | | | 1 | | | | 0 |
| 4 | | | | | 1 | | | 0 |
| 5 | | | | | | 1 | | 0 |
| 6 | | 1 | | | | | 1 | 1 |
| 7 | | | | | | | | 0 |
| Jumlah | | | | | | | | 2 |

Berdasarkan hasil perhitungan *kompleksitas siklomatis* diatas sehingga didapatkan jumlah hasil *independent path* untuk *route login* yaitu 2 dengan jalur *independent path* sebagai berikut:

- Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 Tambah anggota
- Jalur 2 : 1 – 2 – 3 – 4 – 6 – 2 Gagal tambahkan anggota

Diketahui bahwa *route form login* memiliki 2 jalur. Setelah menentukan jalur *independent path* maka langkah selanjutnya yaitu membuat dan *test case* dan melaksanakan pengujian. Hasil pengujian dapat dilihat pada tabel berikut:

Tabel 4.10 Pengujian Jalur tambah data anggota

| Jalur | Input | Proses | Hasil | Ket. |
|---------------|--------------------|-------------------------------|-------------------------------------|----------|
| 1-2-3-4-5-6-7 | Input data anggota | Validasi data inputan anggota | Muncul Pesan Input anggota berhasil | Berhasil |
| 1-2-3-4-6-2 | Input data anggota | Validasi data inputan anggota | Kembali ke Menu data anggota | Berhasil |

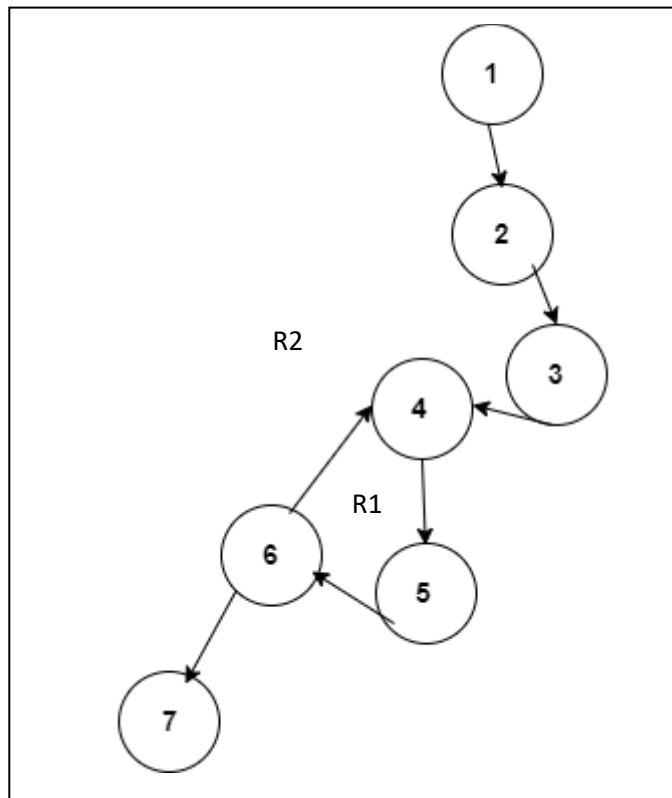
Berdasarkan keseluruhan hasil dari pengujian yang telah dilakukan sehingga menghasilkan nilai CC adalah 2 sehingga menunjukkan tipe

prosedur sederhana yang memiliki tingkat resiko akan cacat atau error adalah rendah.

4. Pengujian tambah transaksi

Route ini memiliki fungsi untuk menambah data transaksi. Selanjutnya akan dibuatkan *flow graph* berdasarkan *kode route form* tambah transaksi sebagai berikut:

- 1)Klik menu Transaksi
- 2)View daftar transaksi
- 3)Tambah Transaksi
- 4)Input data transaksi
- 5)Pilih simpan
- 6)Authentifikasi inputan data transaksi
- 7)Data transaksi



Gambar 4.12 *Flow graph* tambah transaksi

Berdasarkan *flow graph* pada gambar 4.12 diketahui bahwa jumlah *edge* grafik alir (E) = 7 yang merupakan garis yang menghubungkan *node*, jumlah *node* (N) = 7 yang merupakan lingkaran yang menggambarkan suatu

aktivitas. Jumlah *Predicate* (P) = 1 yang merupakan *node* bercabang, dan jumlah *region* (R) = 2 yang menandakan suatu area dalam *flow graph*, yang dapat dilihat dengan simbol R1 dan R2 pada gambar 4.12. Sehingga rumus perhitungan *Cyclomatic complexity* dan juga bantuan *graph matrix* adalah:

Tabel 4.11 *Cyclomatic Complexity Route Form* tambah transaksi

| | | |
|--------------------|----------------|------------|
| $V(G) = E - N + 2$ | $V(G) = P + 1$ | $V(G) = R$ |
| $V(G) = 7 - 7 + 2$ | $V(G) = 1 + 1$ | $V(G) = 2$ |
| $V(G) = 2$ | $V(G) = 2$ | |

Tabel 4.12 *Graph Matrix Route Form* tambah transaksi

| Node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Hasil |
|---------------|---|---|---|---|---|---|---|----------|
| 1 | | 1 | | | | | | 0 |
| 2 | | | 1 | | | | | 0 |
| 3 | | | | 1 | | | | 0 |
| 4 | | | | | 1 | | | 0 |
| 5 | | | | | | 1 | | 0 |
| 6 | | | | 1 | | | 1 | 1 |
| 7 | | | | | | | | 0 |
| Jumlah | | | | | | | | 2 |

Berdasarkan hasil perhitungan *kompleksitas siklomatis* diatas sehingga didapatkan jumlah hasil *independent path* untuk *route login* yaitu 2 dengan jalur *independent path* sebagai berikut:

- Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 Tambah Transaksi
- Jalur 2 : 1 – 2 – 3 – 4 – 6 – 4 Gagal tambahkan transaksi

Diketahui bahwa *route form* tambah transaksi memiliki 2 jalur. Setelah menentukan jalur *independent path* maka langkah selanjutnya yaitu membuat dan *test case* dan melaksanakan pengujian. Hasil pengujian dapat dilihat pada tabel berikut:

Tabel 4. 13 Pengujian Jalur tambah data transaksi

| Jalur | Input | Proses | Hasil | Ket. |
|-------------------|-------------------------|---------------------------------------|---|-------------|
| 1-2-3-4- 5-6-7 | Input data transaksi | Validasi data inputan transaksi | Muncul Pesan Input anggota berhasil | Berhasil |
| 1-2-3-4- 6-4 | Input data transaksi | Validasi data inputan transaksi | Tetap ke menu Input transaksi | Berhasil |

Berdasarkan keseluruhan hasil dari pengujian yang telah dilakukan sehingga menghasilkan nilai CC adalah 2 sehingga menunjukkan tipe prosedur sederhana yang memiliki tingkat resiko akan cacat atau error adalah rendah.

BAB V

PENUTUP

3.1 Kesimpulan

Berdasarkan hasil dari penelitian ini, maka dapat ditarik kesimpulan sebagai berikut:

1. Implementasi framework Laravel berhasil digunakan untuk mengembangkan sistem informasi perpustakaan berbasis web yang sesuai dengan kebutuhan perpustakaan daerah Kabupaten Luwu Utara. Dengan Laravel, sistem yang dibangun memiliki struktur yang teratur, mudah dikembangkan, dan keamanan yang lebih baik.
2. Sistem informasi perpustakaan yang dikembangkan mampu melakukan klasifikasi buku secara efektif dan terintegrasi dengan kategori buku yang ada. Fitur klasifikasi ini mempermudah pengguna dalam mencari dan mengelompokkan buku berdasarkan kategori, sehingga meningkatkan efisiensi dalam pengelolaan koleksi perpustakaan.
3. Pengembangan sistem informasi perpustakaan dengan menggunakan framework Laravel terbukti efektif dalam meningkatkan manajemen dan pelayanan perpustakaan. Hal ini terlihat dari kemudahan dalam pengelolaan data buku, peningkatan aksesibilitas pengguna, serta kemampuan sistem dalam mendukung operasional perpustakaan secara keseluruhan. Secara keseluruhan, penelitian ini menunjukkan bahwa penggunaan framework Laravel dalam pengembangan sistem informasi perpustakaan dapat memberikan solusi yang efektif dan efisien untuk meningkatkan kinerja perpustakaan daerah Kabupaten Luwu Utara.

3.2 Saran

Adapun saran untuk dapat mengembangkan aplikasi ini lebih lanjut, yaitu:

1. Pengembangan Fitur Lanjutan:
Untuk lebih meningkatkan kinerja dan fungsionalitas sistem, disarankan untuk menambahkan fitur-fitur lanjutan seperti sistem peminjaman dan pengembalian buku secara otomatis, notifikasi pengingat untuk pengembalian buku, serta integrasi dengan sistem manajemen perpustakaan lainnya.

2. Pelatihan dan Sosialisasi:

Agar sistem informasi perpustakaan dapat digunakan secara optimal, penting untuk memberikan pelatihan dan sosialisasi kepada seluruh staf perpustakaan serta pengguna perpustakaan mengenai cara penggunaan sistem ini. Hal ini akan memastikan bahwa semua pihak dapat memanfaatkan fitur-fitur yang ada dengan baik.

3. Evaluasi dan Pemeliharaan Berkala:

Sistem informasi perpustakaan perlu dievaluasi dan dipelihara secara berkala untuk memastikan bahwa sistem tetap berjalan dengan baik dan sesuai dengan kebutuhan pengguna. Evaluasi rutin dapat membantu dalam mengidentifikasi dan memperbaiki bug atau masalah teknis yang mungkin muncul.

4. Peningkatan Keamanan:

Meskipun Laravel sudah menyediakan fitur keamanan yang baik, tetap perlu dilakukan peningkatan dan pengawasan terhadap aspek keamanan sistem. Hal ini termasuk perlindungan data pengguna, backup data secara rutin, dan pemantauan terhadap potensi ancaman keamanan siber.

5. Penelitian Lanjutan:

Disarankan untuk melakukan penelitian lanjutan guna mengevaluasi dampak jangka panjang dari penerapan sistem ini terhadap kinerja perpustakaan. Penelitian lebih lanjut juga dapat mencakup eksplorasi teknologi baru yang dapat diintegrasikan ke dalam sistem untuk terus meningkatkan efisiensi dan efektivitas layanan perpustakaan.

DAFTAR PUSTAKA

- A'yuniyah, Q., Fadillah Pratama, P., Zaid, R., Ismail, I. Z., Ramadhan, R., & Fadhila, R. (2022). Library Inventory Information System Web-Based. *Seminar Nasional Penelitian Dan Pengabdian Masyarakat*, 290–298. <https://journal.irpi.or.id/index.php/sentimas>
- Adensa, A., Raihan, K., Faisal Rafi, R., Richwandi Putra, I., & Azizah, F. (2024). Pengembangan Web Dinas Perpustakaan Dan Arsip Berbasis Laravel Framework Pada Dpad Kota Tangerang. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(6), 3877–3883. <https://doi.org/10.36040/jati.v7i6.7840>
- Ariyanti, L., Satria, M. N. D., & Alita, D. (2020). Sistem Informasi Akademik Dan Administrasi Dengan Metode Extreme Programming Pada Lembaga Kursus Dan Pelatihan. *Jurnal Teknologi Dan Sistem Informasi*, 1(1), 90–96. <https://doi.org/10.33365/jtsi.v1i1.214>
- Bakri. (2023). *Konsep Model-View-Controller (MVC) dalam Pengembangan WEB - Biro Administrasi Kepegawaian, Karir dan Informasi Universitas Medan Area*. <https://bakri.uma.ac.id/konsep-model-view-controller-mvc-dalam-pengembangan-web/>
- Bean Test: Basis Path Testing*. (n.d.). Retrieved June 5, 2024, from <https://howtesting.blogspot.com/2014/11/basis-path-testing.html>
- BPS Sulawesi Selatan. (2022). Badan Pusat Statistik Provinsi Sulawesi Selatan. *Badan Pusat Statistik*, <https://sulsel.bps.go.id/index.php/linkTabelStatis>. <https://sulsel.bps.go.id/indicator/12/83/1/jumlah-penduduk-menurut-kabupaten-kota.html>
- Dondy Bappedyanto. (2023). Mengetahui MySQL, Definisi, Fungsi, hingga Cara Kerjanya. In *Bznet Go* (pp. 1–5). <https://www.biznetgio.com/news/apa-itu-mysql>
- Extreme Programming: 5 Tahapan, Kelebihan, dan Kekurangan*. (n.d.). Retrieved May 10, 2024, from <https://www.sekawanmedia.co.id/blog/extreme-programming-adalah/>
- Extreme Programming (XP) - Definisi, Nilai, Tahapan - serupa.id*. (n.d.). Retrieved May 10, 2024, from <https://serupa.id/extreme-programming-xp-definisi-nilai-tahapan/>
- Harike, M. H. (2019). Aplikasi E-voting Pemilihan OSIS pada SMA NEGERI 11 Luwu. *Seminar Nasional Teknologi Informasi Dan Komputer*, 175–183. <https://journal.uncp.ac.id/index.php/semantik/article/view/1506>
- Kesuma, C., & Kholifah, D. N. (2019). Sistem Informasi Akademik Berbasis Web Pada Lkp Rejeki Cilacap. *EVOLUSI: Jurnal Sains Dan Manajemen*, 7(1), 82–88. <https://doi.org/10.31294/evolusi.v7i1.5026>
- Latifatunnisa, H. (2022a). *Bahasa Pemrograman: Arti, Fungsi, Tingkatan, Tips Pemula 2023 | RevoU*. Revoupedia. <https://revou.co/panduan-teknis/bahasa-pemrograman>
- Latifatunnisa, H. (2022b). *Bahasa Pemrograman: Arti, Fungsi, Tingkatan, Tips Pemula*

2023 / *RevoU*. Revoupedia. <https://revou.co/panduan-teknis/bahasa-pemrograman>

- Leo, Y. L., Sudarmaji, S., & Irawan, D. (2021). Perancangan Sistem Informasi Peminjaman Dan Pengembalian Buku Perpustakaan Pada Sma Muhammadiyah 1 Bandar Sribhawono Berbasis Web Menggunakan Metode Rapid Application Development (Rad). *Jurnal Mahasiswa Ilmu Komputer*, 2(2), 233–241. <https://doi.org/10.24127/ilmukomputer.v2i2.1674>
- Listiyono, H., Sani, D. L., Khristianto, T., & Soelistijadi, R. (2022). Desain Sistem Informasi Perpustakaan Universitas Stikubank Semarang Berbasis Web. *Pixel :Jurnal Ilmiah Komputer Grafis*, 15(1), 121–131. <https://doi.org/10.51903/pixel.v15i1.742>
- Maghfiroh, I., & Hidayat, A. (2021). Sistem Informasi Perpustakaan Pesantren menggunakan Framework Laravel (Studi Kasus Pada Pondok Pesantren Luhur Wahid Hasyim Semarang (PPLWH Putri)). *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 3(1), 68. <https://doi.org/10.36499/jinrpl.v3i1.4038>
- Malik, A., Hasanuddin, M., & Kurniawan, F. (2023). Analisis dan Perancangan Sistem Informasi Desa Berbasis Web Menggunakan OpenSID dengan Metode Prototyping. *Dewantara Journal of Technology*, 4(2), 6–11.
- Naja, I. N., Teknologi, F., Dan, I., & Semarang, U. (2023). *IMPLEMENTASI FRAMEWORK LARAVEL DALAM APLIKASI PERPUSTAKAAN SMA NEGERI 1 PECANGAAN JEPARA*.
- News. (2023). *Apa Itu PHP? Pengertian, Sejarah, dan Fungsinya*. <https://www.biznetgio.com/news/apa-itu-php>. <https://www.biznetgio.com/news/apa-itu-php>
- Nofiati, N., & Daru, A. F. (2021). Sistem Informasi Perpustakaan Berbasis Web Dengan Framework Laravel. *Information Science and Library*, 2(2), 101. <https://doi.org/10.26623/jisl.v2i2.4352>
- Nur, F. (2020). Pengaruh Kewajiban Penyediaan Modal Minimum (Kpmm), Beban Operasional Pada Pendapatan Operasional (Bopo) Dan Financing To Deposit Ratio (Fdr) Terhadap Profitabilitas Pt Bank Syariah Mandiri. *Skripsi*, 21(1), 1–104. <http://journal.um-surabaya.ac.id/index.php/JKM/article/view/2203%0Ahttp://mpoc.org.my/malaysia-n-palm-oil-industry/>
- Nurhayati, W., Yanti, G., & Sari, K. (2023). *IMPLEMENTASI METODE WATERFALL PADA SISTEM INFORMASI PERPUSTAKAAN ONLINE SMK NEGERI 1 SEPULUH AGUNG*. 4(2).
- Nwabudike, L. C. (2018). Pustular Eruption (Iododerma?) in a patient with cancer treated with complementary and alternative medicine. *JAMA Dermatology*, 154(4), 495–496. <https://doi.org/10.1001/jamadermatol.2017.6164>
- Setiawan, R. (2021). Flowchart: Fungsi, Jenis, Simbol, dan Contohnya - Dicoding Blog. In *Dicoding*. <https://www.dicoding.com/blog/flowchart-adalah/>
- Sie, J. B. L., Izmy Alwiah Musdar, & Syamsul Bahri. (2022). Pengujian White Box

- Testing Terhadap Website Room Menggunakan Teknik Basis Path. *KHARISMA Tech*, 17(2), 45–57. <https://doi.org/10.55645/kharismatech.v17i2.235>
- Supriyadi. (2023). Perancangan Sistem Perpustakaan Berbasis Web Menggunakan Framework Laravel. *JSAI (Journal Scientific and Applied Informatics)*, 6(2), 105–113. <https://doi.org/10.36085/jsai.v6i2.5019>
- Sutarno NS. (2006). *Manajemen Perpustakaan*. CV. Sagung Seto.
- Vikasari, C. (2023). Cyclomatic Complexity dan Graph Matrix dalam Pengujian Sistem Informasi Manajemen Rumah Sakit. *Infotekmesin*, 14(1), 43–49. <https://doi.org/10.35970/infotekmesin.v14i1.1636>
- Wintana, D., Pribadi, D., & Nurhadi, M. Y. (2022). Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing. *Jurnal Larik: Ladang Artikel Ilmu Komputer*, 2(1), 8–16. <https://doi.org/10.31294/larik.v2i1.1382>

LAMPIRAN

KUESIONER PENGGUNA

Judul Penelitian: Implementasi Framework Laravel pada Sistem Informasi Perpustakaan Daerah Kabupaten Luwu Utara

Petunjuk Pengisian:

1. Isilah kuesioner ini dengan jujur dan sesuai dengan pengalaman Anda.
2. Pilihlah jawaban yang paling sesuai dengan kondisi dan pendapat Anda.
3. Kuesioner ini bersifat anonim dan data yang Anda berikan akan dijaga kerahasiaannya.

A. Data Diri Responden

1. Jenis Kelamin:

- Laki-laki
- Perempuan

2. Usia:

- < 20 tahun
- 20-29 tahun
- 30-39 tahun
- 40-49 tahun
- > 50 tahun

3. Tingkat Pendidikan:

- SD
- SMP
- SMA
- Diploma
- S1
- S2/S3

4. Status Pekerjaan:

- Pelajar/Mahasiswa
- Pegawai Negeri
- Pegawai Swasta
- Wirausaha
- Lainnya: _____

B. Penggunaan Sistem Informasi Perpustakaan

1. Seberapa sering Anda mengunjungi perpustakaan?

- Setiap hari
- Seminggu sekali
- Sebulan sekali
- Jarang
- Tidak pernah

2. Seberapa sering Anda menggunakan sistem informasi perpustakaan?

- Setiap hari
- Seminggu sekali
- Sebulan sekali
- Jarang
- Tidak pernah

C. Penilaian Terhadap Sistem Informasi Perpustakaan

1. Apakah tampilan antarmuka sistem mudah digunakan?

- Sangat mudah
- Mudah
- Cukup
- Sulit
- Sangat sulit

2. Apakah informasi yang disediakan oleh sistem mudah dipahami?

- Sangat mudah dipahami
- Mudah dipahami
- Cukup dipahami
- Sulit dipahami
- Sangat sulit dipahami

3. Bagaimana penilaian Anda terhadap kecepatan sistem?

- Sangat cepat
- Cepat
- Cukup cepat
- Lambat
- Sangat lambat

4. Apakah sistem informasi perpustakaan membantu Anda dalam mencari dan meminjam buku?

- Sangat membantu
- Membantu
- Cukup membantu
- Kurang membantu
- Tidak membantu

5. Seberapa puas Anda dengan keseluruhan sistem informasi perpustakaan?

- Sangat puas
- Puas
- Cukup puas
- Kurang puas
- Tidak puas

D. Saran dan Masukan

1. Apa yang Anda sukai dari sistem informasi perpustakaan ini?
 - o Memudahkan dan dalam mencari informasi
2. Apa yang Anda tidak sukai dari sistem informasi perpustakaan ini?
 - o tidak ada
3. Saran atau masukan untuk perbaikan sistem informasi perpustakaan ini:
 - o lebih di tingkatkan lagi
 - o
 - o

Terima kasih atas partisipasi Anda dalam mengisi kuesioner ini. Jawaban Anda sangat berharga bagi perbaikan dan pengembangan sistem informasi perpustakaan kami.