BAB I

PENDAHULUAN

A. Latar Belakang

Dalam era globalisasi sekarang ini dunia informasi berkembang begitu pesat karena ditunjang dengan perkembangan teknilogi yang semakin canggih. Komputer Merupakan salah satu alat yang berperang penting dalam perkembangan teknologi informasi tersebut. Oleh karena itu suatu lembaga yang menggunkan komputer sebagai media pengolahan data dengan sistem secara manual. Dapat dikatakan pengolahan data dengan menggunkan komputer akan menunjang efisien dan produktivitas yang tinggi.

Penerapan teknologi informasi saat ini telah menyebar hampir disemua bidang, tidak terkecuali dibidang perpustakaan. Perpustakaan merupakan intitusi/lembaga yang menyediakan koleksi bahan perpustakaan tertulis, tercetak dan terekam sebagai pusat sumber informasi untuk keperluan pendidikan,penelitian dan rekreasi intelektual.

Dalam peminjaman buku biasanya mahasiswa sering lupa dengan batas pengembalian buku. Oleh karena itu sistem memanfaatkan teknologi pemindaian *barcode* untuk membaca dan mengambil informasi yang terkandung dalam *barcode*. Apabila mahasiswa meminjam buku, petugas perpustakaan dapat dengan cepat memindai *barcode* buku menggunakan pemindai *barcode* dan mengupdate status buku tersebut dalam sistem, apabila batas waktu peminjaman buku akan berakhir muncul nontifikasi di wa dengan keterangan “batas waktu peminjaman

buku anda akan berakhir besok,mohon segera mengembalikan bukunya". Hal ini mengurangi keterlambatan mahasiswa dalam mengembalikan buku.

Selain itu, pengembangan sistem informasi perpustakaan dengan *barcode* memungkinkan adanya sistem penelusuran yang lebih cepat dan akurat. Petugas perpustakaan dapat dengan mudah mencari buku berdasarkan *barcode* atau menggunakan *barcode* untuk menemukan informasi terkait buku tersebut dalam sistem.

Secara keseluruhan, penggunaan sistem *barcode* dalam pengembangan sistem informasi perpustakaan membantu meningkatkan efisiensi, keakuratan, dan kemudahan dalam pengelolaan dan penelusuran informasi perpustakaan.

B. Rumusan Masalah

Dari uraian yang telah dijelaskan pada latar belakang maka di identifikasi masalah sebagai berikut:

1. Bagaimana merancang dan mengembangkan sistem informasi perpustakaan dengan teknologi barcode untuk meningkatkan efisiensi pengelolaan data peminjaman dan pengembalian buku?
2. Bagaimana sistem informasi perpustakaan berbasis barcode dapat membantu mengurangi keterlambatan pengembalian buku oleh mahasiswa?

C. Batasan Masalah

Berdasarkan rumusan masalah di atas, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi ini hanya berbasis *desktop*.
2. Bahasa pemrograman yang digunakan adalah *Delphi* 2010.

D. Tujuan Penelitian

1. Merancang suatu aplikasi sistem informasi perpustakaan menggunakan sistem *barcode*.
2. Mengingatkan batas waktu peminjaman buku melalui notifikasi *whatsapp*.

E. Manfaat Penelitian

Adapun manfaat yang dapat diberikan dari penelitian ini adalah sebagai berikut:

1. Bagi akademik

   Sebagai bahan referensi bagi peneliti lain untuk mengembangkan kemampuan di bidang pengembangan sistem informasi perpustakaan.

2. Bagi penulis

   Penulis dapat mengimplementasikan ilmu yang didapat selama proses perkuliahan.

3. Bagi masyarakat

   Mempercepat proses peminjaman dan pengembalian buku dan mengurangi mahasiswa lupa dalam peminjaman buku.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Pengembangan Sistem Informasi

Pengembangan sistem informasi adalah proses pencarian solusi atau pemecahan dari suatu masalah baik secara terstruktur maupun berorientasi objek. Pengembangan secara terstruktur biasanya lebih menekankan pembuatan sistem berdasarkan proses kerja/prosedur yang telah ditetapkan. Sedangkan pengembangan sistem berorientasi objek lebih menekankan pembuatan sistem terhadap peranan objek yang terlibat dalam sistem tersebut. Pengembangan sistem informasi secara terstruktur terdiri dari beberapa kegiatan / tahapan (*phased*), yaitu tahap analisis sistem (*analysist*), konstruksi sistem (*construction*), pengkodean (*coding*), uji sistem (*testing*), dan tahap pemeliharaan sistem (*maintenance*). Sedangkan pengembangan sisteminformasi berorientasi objek terdiri dari tahap analisis (*Inception*), design (*elaboration*), konstruksi (*construction*) dan penggantian sistem (*Transition*). (Arbani, 2022).

2. Sistem

Definisi sistem telah banyak dikemukakan oleh para ahli dari berbagai disiplin ilmu. Walaupun definisi sistem begitu banyak namun sistem dapat diartikan dengan mengacu pada pendapat para ahli.

(Susanti, 2020) sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Suatu sistem dibangun bukan dinilai dari segi tampilan semata, namun dinilai dari pola aliran informasi dan implementasi. Faktor-faktor yang harus dipertimbangkan dalam pembangunan sistem, yaitu:

a. Efisiensi dan efektifitas
b. Prosedur yang singkat. Sistem yang dirancang tidak menjemukan pada saat melakukan *input* data.
c. Sistem yang optimal. Sistem dapat bekerja secara optimal dalam pemanfaatan sumber daya yang dimiliki oleh perusahaan.
d. Tren di masa depan. Sistem yang dibangun mampu beradaptasi dengan perkembangan jaman.
e. Efisiensi dalam biaya. Sistem dibangun berdasarkan pada perencanaan dan perancangan secara matang guna menghemat biaya.
f. Integritas dan keamanan data. Sistem yang baru dibangun harus memenuhi standar integritas dan keamanan data.
g. Interaktif. Sistem yang dibangun harus mudah dipahami dengan cara mengidentifikasi, menyeleksi dan menyusun komponen-komponen yang sesuai dengan latar belakang pemakaiannya.

3. Informasi

Dalam hal ini penulis akan mendeskripsikan definisi dari Informasi menurut pendapat ahli yang memiliki gagasan yang sama yaitu :

Informasi adalah data yang telah diorganisasi dan telah memiliki kegunaan dan manfaat (Dewi Maharani, 2019). Informasi adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan, sebagaimana perannya, pengguna membuat keputusan yang lebih baik sebagai kuantitas dan kualitas dari peninggkatan informasi (Retno sari, 2020). Dari beberapa pengertian informasi menurut pendapat ahli diatas dapat penulis simpulkan, bahwa informasi adalah data mentah yang telah diolah sedemikian rupa sehingga menghasilkan sesuatu yang bermakna bagi penggunanya dalam mengambil sebuah keputusan.

4. Perpustakaan

Perpustakaan merupakan suatu lembaga atau unit kerja yang berfungsi sebagai tempat untuk menghimpun, mengolah, menyimpan, serta mendayagunakan berbagai sumber informasi baik dalam bentuk tercetak maupun noncetak untuk menunjang kegiatan belajar, penelitian, pendidikan, dan rekreasi intelektual. Menurut Undang-Undang Republik Indonesia Nomor 43 Tahun 2007 tentang Perpustakaan, perpustakaan adalah institusi pengelola koleksi karya tulis, karya cetak, dan/atau karya rekam secara profesional dengan sistem baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka. Dengan demikian, perpustakaan tidak hanya berfungsi sebagai gudang buku, melainkan juga sebagai pusat informasi dan ilmu pengetahuan. Tahir, F., Hastuty, A., & Yunus, M. (2025).

5. *Database*

Database dalam sistem informasi perpustakaan berbasis *barcode* dirancang untuk mengelola data penting seperti koleksi buku, data anggota, transaksi peminjaman atau pengembalian, serta pengguna sistem. Tabel utama dalam *database* ini meliputi tabel buku, anggota, transaksi, dan pengguna. Tabel buku menyimpan informasi detail mengenai koleksi buku, termasuk ID buku sebagai kunci utama, judul, pengarang, penerbit, tahun terbit, kategori, jumlah eksemplar, dan kode *barcode* unik untuk setiap buku. Tabel anggota berisi data anggota perpustakaan seperti *ID* anggota sebagai kunci utama, nama lengkap, alamat, nomor telepon, tanggal pendaftaran, dan *barcode* yang dicetak pada kartu anggota.

Tabel transaksi digunakan untuk mencatat data peminjaman dan pengembalian buku. Kolom-kolomnya mencakup ID transaksi sebagai kunci utama, ID buku, ID anggota, tanggal peminjaman, tanggal pengembalian, dan status transaksi seperti "Dipinjam" atau "Dikembalikan." Selain itu, tabel pengguna menyimpan informasi petugas perpustakaan, termasuk ID pengguna, nama lengkap, username untuk *login*, kata sandi yang disimpan dalam format hash untuk keamanan, serta peran pengguna seperti admin atau petugas biasa.

Database adalah sistem terorganisasi yang dirancang untuk menyimpan, mengelola, dan mengakses data secara efisien. Data dalam *database* dapat berupa informasi berupa angka, teks, gambar, atau format lain yang terstruktur untuk memudahkan pengguna atau perangkat lunak dalam memproses informasi tersebut. *Database* umumnya digunakan dalam berbagai

aplikasi, seperti sistem keuangan, *e-commerce*, manajemen pelanggan, hingga analitik data besar. Untuk mengelola *database*, digunakan perangkat lunak yang disebut *Database Management System (DBMS*), seperti *MySQL, PostgreSQL, Oracle Database,* dan *MongoDB*. *DBMS* memungkinkan pengguna untuk menambahkan, menghapus, memperbarui, dan mengambil data menggunakan perintah khusus seperti *Structured Query Language (SQL)*.

Database dalam sistem informasi perpustakaan berbasis *barcode* adalah elemen utama yang memungkinkan pengelolaan data secara terstruktur, efisien, dan mudah diakses. Sistem ini dirancang menggunakan pendekatan relasional, yang menghubungkan berbagai tabel berdasarkan kunci utama (*Primary Key*) dan kunci asing (*Foreign Key*) untuk memastikan integritas data dan mencegah redundansi.

Tabel buku adalah inti dari database ini, menyimpan data penting mengenai koleksi perpustakaan. Setiap buku memiliki ID unik yang berfungsi sebagai kunci utama, serta informasi lainnya seperti judul, pengarang, penerbit, tahun terbit, kategori, jumlah eksemplar, dan *barcode* unik. *Barcode* ini digunakan untuk mengidentifikasi setiap buku secara cepat dan akurat, terutama saat proses peminjaman atau pengembalian.

Tabel anggota dirancang untuk menyimpan informasi mengenai pengguna perpustakaan. Setiap anggota memiliki ID unik sebagai kunci utama, nama lengkap, alamat, nomor telepon, tanggal pendaftaran, dan *barcode* yang dihasilkan untuk dicetak pada kartu anggota. *Barcode* ini berfungsi untuk

mempermudah proses identifikasi anggota saat mereka meminjam atau mengembalikan buku.

Tabel transaksi memainkan peran penting dalam mencatat setiap aktivitas peminjaman dan pengembalian buku. Kolom-kolomnya mencakup ID transaksi sebagai kunci utama, ID buku dan ID anggota sebagai kunci asing yang menghubungkan tabel ini dengan tabel buku dan anggota, serta informasi waktu seperti tanggal peminjaman dan tanggal pengembalian. Status transaksi juga dicatat, misalnya "Dipinjam" untuk buku yang sedang dipinjam atau "Dikembalikan" untuk buku yang telah dikembalikan. Tabel ini memastikan pelacakan yang rinci dan terorganisir untuk setiap buku dan anggota.

Tabel pengguna dirancang untuk mengelola data petugas perpustakaan yang bertugas menggunakan sistem. Setiap petugas memiliki ID unik sebagai kunci utama, nama pengguna, *username*, *password* yang disimpan dalam format *hash* untuk keamanan, serta peran atau hak akses mereka (misalnya, *admin* atau petugas). Hal ini memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses fitur-fitur tertentu dari sistem.

Relasi antar tabel dirancang dengan cermat untuk memastikan efisiensi dan keamanan data. Misalnya, ID buku di tabel transaksi terhubung ke tabel buku untuk mencatat transaksi spesifik, sedangkan ID anggota menghubungkan tabel transaksi dengan tabel anggota. Pendekatan ini memungkinkan data diintegrasikan tanpa redundansi, sehingga sistem dapat mengelola data yang besar dengan efisien.

Selain itu, *database* ini mendukung pengembangan fitur tambahan seperti perhitungan denda keterlambatan berdasarkan tanggal pengembalian, pemberitahuan otomatis kepada anggota mengenai status peminjaman, dan laporan statistik yang mencakup informasi tentang buku yang paling sering dipinjam atau anggota yang paling aktif. Dengan struktur yang fleksibel dan terorganisir, database ini dapat diperluas untuk mendukung integrasi dengan teknologi lain seperti *cloud storage* atau analitik berbasis AI, menjadikan perpustakaan lebih modern dan responsif terhadap kebutuhan penggunanya

6. *Borland Delphi*

Borland Delphi adalah lingkungan pengembangan perangkat lunak (IDE) dan bahasa pemrograman yang awalnya dikembangkan oleh *Borland* pada tahun 1995 dan kini dikelola oleh *Embarcadero Technologies. Delphi* menggunakan *Object Pascal* sebagai bahasa pemrogramannya, yang dikenal karena sintaksnya yang sederhana dan dukungan pemrograman berbasis objek. Dengan fitur utama seperti pustaka komponen *visual (VCL), Delphi* mempermudah pembuatan antarmuka pengguna melalui metode *drag-and-drop*, sehingga mempercepat pengembangan aplikasi *desktop* berbasis *Windows. IDE Delphi* dilengkapi dengan *editor* kode, *debugger,* dan *desainer visual* yang memungkinkan pengembang melihat hasil desain secara langsung. Selain itu, *Delphi* mendukung pengembangan aplikasi berbasis *database,* seperti *MySQL, SQLite, Oracle,* dan *SQL Server,* melalui komponen seperti *BDE (Borland Database Engine), ADO,* dan *FireDAC*.

Delphi memiliki kemampuan *multi-platform* dalam versi modernnya, memungkinkan pengembang membuat aplikasi untuk *Windows, macOS, iOS, Android,* dan *Linux.* Aplikasi yang dibuat dengan *Delphi* memiliki performa tinggi karena dikompilasi langsung menjadi kode mesin. Keunggulan lain dari *Delphi* adalah kemudahan penggunaannya, efisiensi pengembangan, serta kompatibilitasnya dengan berbagai teknologi seperti *Windows* API dan *database.* Namun, *Delphi* memiliki beberapa kekurangan, seperti biaya lisensi yang relatif mahal dan popularitas yang menurun dibandingkan dengan *IDE modern* lainnya. Meskipun demikian*, Delphi* tetap menjadi alat yang andal untuk mengembangkan aplikasi *desktop*, terutama yang memerlukan performa tinggi dan antarmuka pengguna yang kompleks. Saat ini, *Delphi* banyak digunakan untuk aplikasi *desktop* berbasis *Windows,* sistem manajemen *database,* dan perangkat lunak *enterprise*.

Borland Delphi adalah lingkungan pengembangan perangkat lunak *(IDE)* yang dirancang untuk mempermudah dan mempercepat proses pembuatan aplikasi *desktop* berbasis *Windows*, menggunakan bahasa pemrograman *Object Pascal*. Pertama kali dirilis pada tahun 1995 oleh *Borland*, *Delphi* menawarkan pendekatan pemrograman visual yang intuitif, memungkinkan pengembang untuk mendesain antarmuka pengguna dengan *drag-and-drop* menggunakan pustaka komponen *visual* atau *VCL (Visual Component Library). IDE* ini terintegrasi dengan editor kode yang canggih, *debugger*, dan *desainer visual,* sehingga pengembang dapat merancang aplikasi secara interaktif dan melihat hasilnya secara langsung.

Salah satu fitur unggulan *Delphi* adalah dukungannya terhadap pengembangan aplikasi berbasis *database. Delphi* mendukung integrasi dengan berbagai jenis *database* seperti *MySQL, Oracle, SQL Server,* dan *SQLite*, yang memungkinkan pengembang membangun aplikasi dengan manajemen data yang kuat. Komponen bawaan seperti *Borland Database Engine (BDE), ADO (ActiveX Data Objects),* dan *FireDAC* memberikan fleksibilitas dan performa tinggi dalam mengelola data. Selain itu, *Delphi* mendukung berbagai protokol koneksi dan pengelolaan transaksi, sehingga sangat cocok untuk aplikasi enterprise yang memerlukan pengelolaan data skala besar.

Dalam versi modernnya yang dikelola oleh *Embarcadero Technologies, Delphi* telah berkembang menjadi alat pengembangan *multi-platform.* Dengan fitur seperti *FireMonkey (FMX),* pengembang dapat membuat aplikasi yang tidak hanya berjalan di *Windows*, tetapi juga di *macOS, iOS, Android,* dan *Linux* dengan basis kode yang sama. *Delphi* juga mendukung pengembangan aplikasi berbasis teknologi terbaru, seperti *RESTful API, cloud computing,* dan *IoT (Internet of Things),* menjadikannya relevan untuk kebutuhan pengembangan modern.

Keunggulan *Delphi* meliputi kecepatan pengembangan aplikasi, performa tinggi karena aplikasi dikompilasi menjadi kode mesin (*native code)*, kompatibilitas luas dengan berbagai teknologi, dan pendekatan pemrograman visual yang mempermudah pengembangan. *Delphi* juga memiliki komunitas pengguna yang besar, dokumentasi yang luas, serta dukungan teknis dari

Embarcadero. Namun, *Delphi* memiliki beberapa kekurangan, seperti biaya lisensi yang relatif mahal dan popularitasnya yang menurun dibandingkan dengan *IDE open-source* atau modern lainnya.

Borland Delphi telah digunakan secara luas dalam berbagai bidang, seperti pengembangan aplikasi *desktop*, sistem manajemen *database,* perangkat lunak teknis, serta aplikasi *enterprise* dengan antarmuka pengguna yang kompleks. Meski menghadapi persaingan ketat dari alat pengembangan lain, *Delphi* tetap menjadi pilihan utama untuk pengembang yang mengutamakan stabilitas, efisiensi, dan kemampuan lintas platform dalam membangun aplikasi. Dengan struktur yang fleksibel dan terus diperbarui, *Delphi* mampu mendukung pengembangan perangkat lunak dari skala kecil hingga *enterprise.*

7. *SQLite 3*

SQLite adalah salah satu solusi penyimpanan data yang paling ringan dan mudah digunakan, tapi tetap andal. Berbeda dengan sistem basis data yang kompleks seperti *MySQL* atau *PostgreSQL, SQLite* langsung bekerja dari satu file kecil, tanpa perlu *server*. Pengertian dari *SQLite* adalah sistem manajemen basis data atau *database* yang sangat ringan dan tidak membutuhkan *server* untuk menjalankannya. Berbeda dengan beberapa database lain seperti *MySQL* atau *PostgreSQL* yang memerlukan instalasi dan pengaturan server, *SQLite* dapat berjalan langsung di dalam aplikasi yang menggunakannya. Data disimpan dalam satu file yang bisa diakses kapan saja tanpa pengaturan

tambahan. Karena kepraktisannya, *SQLite* banyak digunakan dalam aplikasi *mobile*, perangkat *IoT*, dan program *desktop*.

Pada dasarnya, *SQLite* adalah sebuah *software* atau sistem manajemen *relational database* yang bersifat *open-source, serverless,* dan *portabel* yang digunakan untuk mempermudah akses dan mengelola penyimpanan data. *Software* ini didesain untuk tujuan utama dalam hal kecepatan, kemudahan, *portabel,* dan dapat diandalkan ketika kita butuh untuk menyimpan data hanya 1 *kilobytes* ataupun ratusan *gigabytes.*

Penggunanya dapat dengan mudah menggunakan *SQLite* tanpa perlu melakukan instalasi *environment* atau konfigurasi apapun karena *SQLite* merupakan *stand-alone software* yang memang didesain untuk disematkan (*embedded*) ke dalam sebuah aplikasi. Selain itu, beberapa produk sistem manajemen *relational database* yang cukup populer lainnya adalah *MySQL, Oracle Database, PostgreSQL* memiliki ukuran yang jauh lebih besar dibandingkan *SQLite*.

Disamping itu *SQLite* juga tidak menggunakan model arsitektur *client-server* seperti *MySQL* dan *PostgreSQL* karena seluruh programnya berisi *library* Bahasa pemrograman C yang menyematkan *database* ke dalam sebuah aplikasi. Ini berarti Anda tidak membutuhkan konfigurasi *server* yang rumit untuk dapat menjalankan sistem manajemen *database* pada aplikasi.

8. *NodeJS*

Node.js merupakan sebuah lingkungan runtime yang memungkinkan eksekusi kode *JavaScript* di luar lingkungan browser. Lingkungan ini dibangun

di atas V8 *JavaScript Engine*, yaitu mesin pemrosesan *JavaScript* yang dikembangkan oleh Google dan digunakan juga dalam browser *Google Chrome. Node.js* pertama kali diperkenalkan oleh Ryan Dahl pada tahun 2009, dengan tujuan utama menyediakan *platform* yang efisien dan ringan untuk membangun aplikasi server-side yang mampu menangani sejumlah besar koneksi secara bersamaan dengan penggunaan sumber daya yang minimal.

Salah satu keunggulan utama *Node.js* terletak pada penggunaan arsitektur *non-blocking* dan *event-driven. Node.js* hanya menggunakan satu *thread* utama *(single-threaded*), namun mampu menangani banyak koneksi secara simultan berkat mekanisme yang disebut *event loop*. Mekanisme ini memungkinkan *Node.js* untuk menunda (*defer)* proses-proses I/O seperti pembacaan *file* atau permintaan jaringan, tanpa harus menghentikan eksekusi kode lainnya. Untuk itu, *Node.js* sangat cocok digunakan pada aplikasi yang membutuhkan respons cepat dan bersifat *real-time*, seperti layanan chat, notifikasi, *dashboard monitoring*, dan *API* berbasis *REST* maupun *GraphQL.*

Node.js juga memiliki ekosistem yang sangat luas melalui *Node Package Manager* (*NPM*), yang menyediakan ribuan modul dan pustaka open-source yang dapat digunakan untuk mempercepat proses pengembangan aplikasi. Kemudahan dalam penggunaan *JavaScript* secara menyeluruh, baik di sisi klien (*frontend*) maupun *server* (*backend*), menjadi salah satu alasan utama mengapa *Node.js* banyak diadopsi oleh pengembang *web* masa kini.

Meskipun memiliki banyak keunggulan, *Node.js* memiliki beberapa keterbatasan, khususnya dalam menangani proses yang bersifat CPU-*intensive*

atau memerlukan komputasi berat, seperti pengolahan gambar dalam jumlah besar, enkripsi kompleks, atau proses perhitungan ilmiah. Hal ini dikarenakan *Node.js* hanya berjalan dalam satu thread utama, sehingga apabila terdapat satu proses yang memakan waktu lama dan tidak bersifat *asynchronous*, dapat menyebabkan keterlambatan atau bahkan penghentian sementara proses lainnya (*blocking*). Oleh karena itu, pemilihan *Node.js* sebagai platform pengembangan harus disesuaikan dengan kebutuhan dan karakteristik dari aplikasi yang akan dibangun.

9. *Visual Studio*

Visual Studio adalah sebuah *Integrated Development Environment (IDE)* yang dikembangkan oleh *Microsoft. IDE* ini dirancang untuk membantu para pengembang dalam membangun berbagai jenis aplikasi, mulai dari aplikasi *desktop, web, mobile*, hingga *game. Visual Studio* dikenal karena kelengkapan fiturnya yang mendukung proses pengembangan perangkat lunak secara menyeluruh, dari penulisan kode, *debugging*, hingga *deployment.* Salah satu keunggulan utamanya adalah editor kode yang canggih, dilengkapi dengan fitur *IntelliSense* yang memberikan saran otomatis dan mempercepat proses penulisan kode. Selain itu, *Visual Studio* juga dilengkapi *debugger* yang kuat, memungkinkan pengembang melacak kesalahan secara detail melalui fitur *breakpoint, watch,* dan *call stack*.

IDE ini mendukung berbagai bahasa pemrograman seperti *C#, C++, VB.NET, JavaScript, Python*, dan masih banyak lagi. Dengan bantuan ekstensi, *Visual Studio* juga bisa digunakan untuk bahasa lain seperti *Java* dan *PHP.*

Terdapat banyak jenis proyek yang bisa dibuat dengan *Visual Studio*, di antaranya aplikasi *desktop (Windows Forms dan WPF*), aplikasi *web (ASP.NET dan Blazor*), *aplikasi mobile (Xamarin dan MAUI*), layanan web atau API, bahkan aplikasi untuk *machine learning* dan *IoT*.

Visual Studio hadir dalam beberapa edisi, yaitu *Community* (gratis untuk individu, pelajar, dan proyek *open-source*), *Professional* (berbayar, dengan fitur tambahan untuk tim kecil hingga menengah), dan *Enterprise* (dengan fitur lengkap untuk kebutuhan perusahaan besar, termasuk *tools* analisis performa dan pengujian otomatis). Walau sering disamakan dengan *Visual Studio Code*, sebenarnya keduanya sangat berbeda. *Visual Studio* adalah IDE yang lengkap dan cocok untuk proyek besar, sedangkan *Visual Studio Code* adalah editor ringan yang fleksibel dan cocok untuk berbagai bahasa dengan bantuan ekstensi.

Beberapa fitur unggulan *Visual Studio* meliputi *Live Share* untuk kolaborasi *real-time, CodeLens* yang menampilkan informasi langsung di atas baris kode, dan *Test Explorer* untuk menjalankan serta memantau unit testing. Meski *Visual Studio* sangat powerful, kekurangannya adalah ukuran file instalasi yang besar dan kebutuhan spesifikasi komputer yang cukup tinggi. Namun demikian, bagi pengembang yang membutuhkan lingkungan pengembangan yang lengkap dan terintegrasi, *Visual Studio* tetap menjadi salah satu pilihan terbaik di industri.

10. *Barcode* 128

Dalam konteks sistem informasi perpustakaan, penggunaan *Barcode* 128 sangat bermanfaat untuk mempercepat dan mempermudah proses pengelolaan data buku dan anggota. Setiap buku atau kartu anggota dapat diberi label *barcode* yang menyimpan informasi unik, seperti nomor inventaris atau nomor identitas anggota. Ketika dilakukan peminjaman atau pengembalian, petugas hanya perlu memindai *barcode* tersebut menggunakan alat pemindai (*barcode scanner*), sehingga proses input data menjadi lebih cepat, akurat, dan efisien dibandingkan metode manual. Selain itu, dengan format *Code* 128 yang mampu menyandikan informasi dalam bentuk teks dan angka, sistem dapat mengintegrasikan *barcode* secara langsung ke dalam database, sehingga meminimalisir kesalahan input dan meningkatkan keamanan data. *Barcode* ini juga dapat dicetak dengan mudah menggunakan printer label biasa, tanpa memerlukan peralatan khusus. Keunggulan lain dari *Code* 128 adalah kemampuannya untuk bekerja dengan baik meskipun dicetak dalam ukuran kecil, yang sangat sesuai dengan keterbatasan ruang pada label buku atau kartu anggota. Oleh karena itu, *Barcode* 128 menjadi standar yang sangat direkomendasikan dalam pembangunan sistem perpustakaan berbasis teknologi informasi, baik di tingkat sekolah, perguruan tinggi, maupun perpustakaan umum.

Barcode 128 merupakan salah satu jenis *barcode* satu dimensi (1D) yang dirancang untuk menyimpan data dalam bentuk alfanumerik secara efisien dan padat. *Barcode* ini banyak digunakan dalam berbagai sistem seperti

perpustakaan, logistik, manajemen persediaan, dan pelacakan barang karena kemampuannya dalam menyimpan karakter dalam jumlah besar dengan ukuran label yang relatif kecil. *Code* 128 dapat memuat hingga 128 karakter ASCII, termasuk angka (0-9), huruf besar dan kecil (A-Z, a-z), serta berbagai simbol khusus. Dalam struktur penyusunannya, barcode ini terdiri dari kode awal (*Start Code*), rangkaian data, checksum (kode pemeriksaan untuk memastikan



Gambar 2. 1 *Barcode* 128

akurasi data), dan kode akhir (*Stop Code*). *Barcode* 128 juga terbagi dalam tiga subset karakter, yaitu Code Set A, B, dan C, yang masing-masing digunakan untuk menyandikan karakter tertentu; misalnya, Set C digunakan secara efisien untuk menyandikan angka dalam pasangan dua digit. Keunggulan utama Code 128 terletak pada kepadatan datanya yang tinggi, fleksibilitas dalam menyandikan berbagai jenis karakter, serta sistem validasi data otomatis melalui checksum, sehingga menjadikannya sangat andal dalam penggunaan. Oleh karena itu, *Barcode* 128 menjadi pilihan populer dalam sistem informasi perpustakaan untuk pelabelan buku maupun anggota, karena mampu meningkatkan efisiensi dan keakuratan proses pencatatan dan pencarian data.

11. *Whatsapp Gateway*

WhatsApp Gateway merupakan sebuah sistem aplikasi yang digunakan untuk mengirim dan menerima pesan *WhatsApp* dari *web* atau aplikasi lain ke perangkat *WhatsApp* yang dituju. Dengan sistem tersebut setiap gawai dan

aplikasi dapat terhubung dengan perangkat *WhatsApp* untuk saling bertukar informasi dan data. API berpesan sebagai pembawa pesan yang menerima permintaan pengguna dan memberitahu sistem apa yang harus dilakukan sehingga memberikan respon yang sesuai untuk permintaan tersebut.

WhatsApp Gateway dapat menyebarkan pesan ke ratusan nomor secara otomatis dan cepat serta terhubung langsung dengan nomor-nomor kontak yang tersimpan tanpa harus mengetik ratusan nomor dan pesan pada ponsel. Semua nomor akan diambil secara otomatis dari basis data tersebut sehingga dapat menghemat waktu. Biasanya *WhatsApp Geteway* digunakan untuk keperluan aplikasi bisnis, baik untuk kepentingan broadcast promosi, pelayanan informasi terhadap pengguna, penyebaran kontek produk atau jasa dan lain-lain(Kompasiana 2020).

12. Dasar Pemrograman

Dasar pemrograman adalah konsep dan prinsip fundamental yang perlu dipahami oleh seseorang yang ingin belajar atau bekerja dalam bidang pemrograman. Berikut hal yang perlu dipahami dalam pada dasar pemrograman:

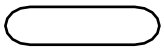
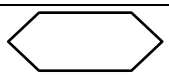
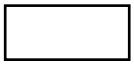
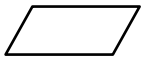
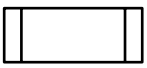
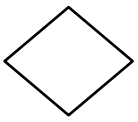
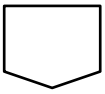
a. Algoritma

Algoritma ialah susunan secara berurut langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Artinya algoritma akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan (Ade Hastuty Hasyim, 2021).

b. Flowchart

Flowchart dalam bahasa indonesia diterjemahkan sebagai *diagram* alir. Dari dua kata ini, maka dapat kita bayangkan bahwa *flowchart* ini berbentuk *diagram* yang bentuknya dapat mengalirkan sesuatu. *Flowchart* adalah refresentasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol, dimana masing-masing simbol mempersentasikan suatu kegiatan tertentu(Ade Hastuty Hasyim, 2021).

Tabel 2. 1 *Flowchart*

No.	Simbol	Nama	Fungsi
1		*Terminator*	Permulaan/akhir program
2		Garis Alir (*Flow Line*)	Arah aliran program
3		*Preparation*	Proses inisialisasi program/pemberian nilai awal
4		Proses	Proses perhitungan/proses pengolahan data
5		Input/Output Data	Proses input/output data, parameter, informasi
6		*Predefined process (sub program)*	Permulaan sub program/proses menjalankan sub program
7		*Decision*	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk data langkah selanjutnya
8		*On Page Connector*	Penghubung bagian-bagian *Flowchart* yang berada pada halaman berbeda
9		*Off Page Connector*	Penghubung bagian-bagian *Flowchart* yang pada halaman berbeda

13. *Unified Modelling Language*

UML (*Unified Modeling Language*) adalah alat desain sistem berorientasi objek. Secara filosofis kemunculan *UML* terinspirasi dari konsep yang sudah ada yaitu konsep pemodelan *Object Oriented* (*OO*), karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh objek dan digambarkan atau dicatat dalam simbol-simbol yang cukup spesifik. proses standar dan independen.

UML adalah pemodelan untuk membantu proses perancangan sistem sehingga meminimalisir kesalahan dalam membuat program. Penerapan *UML* ini menggambarkan struktur aktor yang terlibat, aktifitas setiap aktor, proses dan mekanisme dimana memberikan kemudahan perancangan dalam membatu proses pengkodingan menjadi sebuah aplikasi (Voutama, 2022).

UML memiliki tiga kategori utama yaitu *diagram* struktur, *diagram* perilaku, dan *diagram* interaksi. Adapun daftar simbol *UML* yaitu:

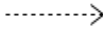
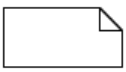
a. *Use case* diagram

Use case Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan fungsionalitas atau perilaku sistem dari sudut pandang pengguna eksternal (aktor). Diagram ini menjelaskan interaksi antara aktor dan sistem dalam bentuk skenario (*use case*) yang merepresentasikan kebutuhan atau fungsi yang diinginkan.

Use case diagram bersifat statis, artinya tidak menggambarkan urutan atau alur waktu, melainkan hanya menunjukkan apa yang bisa dilakukan

sistem, bukan bagaimana sistem melakukannya. Oleh karena itu, diagram ini menjadi alat bantu penting dalam proses analisis kebutuhan dan perancangan sistem. Berikut simbol *use case* diagram.

Tabel 2. 2 *Use case* diagram

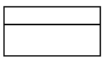
NO.	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
4		*Generalization*	Menspesifikasikan bahwa *use case* sumber secara eksplisit.
5		*Include*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Extend*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*Association*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. *Class* diagram

Class Diagram merupakan salah satu diagram dalam pemodelan *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan struktur statis sistem dalam bentuk kelas-kelas (*class*) beserta atribut, operasi (*method*), dan hubungan antar kelas dalam suatu sistem.

Class diagram sangat penting dalam pengembangan sistem berbasis objek karena menjadi fondasi utama dalam desain perangkat lunak, menggambarkan entitas sistem dan bagaimana entitas tersebut saling berinteraksi. Diagram ini tidak hanya mencerminkan struktur data, tetapi juga perilaku yang dimiliki oleh masing-masing kelas. Berikut symbol class diagram.

Tabel 2. 3 Symbol *Class Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

c. Seguence diagram

Sequence diagram merupakan salah satu jenis diagram interaksi dalam pemodelan *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan urutan interaksi antar objek dalam suatu sistem berdasarkan alur waktu. Diagram ini menampilkan bagaimana objek berkomunikasi satu sama lain melalui pengiriman pesan dalam suatu skenario tertentu, mulai dari awal hingga akhir proses.

Sequence diagram bersifat dinamis karena memodelkan perilaku sistem yang berubah dari waktu ke waktu. Setiap objek dalam diagram memiliki garis hidup (*lifeline*) yang menunjukkan keberadaannya selama proses berlangsung. Pesan-pesan dikirimkan dari satu objek ke objek lainnya, dan pesan tersebut ditampilkan dalam bentuk panah horizontal yang menghubungkan *lifeline*. Berikut symbol sequence diagram.

Tabel 2. 4 Symbol *Sequence* Diagram

No	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktivitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktivitas yang terjadi

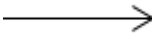
d. *State chart* diagram

State chart Diagram, juga dikenal sebagai *State* Diagram, adalah salah satu jenis diagram perilaku dalam *Unified Modeling Language* (UML) yang

digunakan untuk memodelkan perubahan status *(state*) dari sebuah objek selama siklus hidupnya sebagai respons terhadap peristiwa (*event*) tertentu.

Diagram ini menunjukkan bagaimana sebuah objek berpindah dari satu keadaan ke keadaan lain, berdasarkan aksi atau kondisi yang terjadi dalam sistem. *State chart* diagram sangat berguna untuk memodelkan perilaku dinamis dari sistem atau entitas tertentu, terutama jika objek memiliki banyak status yang saling terkait. Berikut symbol state chart.

Tabel 2. 5 Symbol *State Chart* Diagram

No	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai *Link* pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

e. *Activity* diagram

Activity Diagram merupakan salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur kerja (*workflow*) atau urutan aktivitas dalam suatu proses bisnis atau sistem. Diagram ini menunjukkan bagaimana aktivitas dilakukan, dari satu langkah ke langkah

berikutnya, lengkap dengan percabangan (*decision*), paralelisme (*fork/join*), dan alur keputusan.

Berbeda dengan diagram urutan *(sequence* diagram) yang berfokus pada interaksi antar objek, *activity* diagram lebih menekankan pada proses dan logika alur kerja, sehingga sangat efektif digunakan untuk menganalisis alur sistem secara fungsional dan menggambarkan perilaku dinamis sistem. Berikut symbol *activity* diagram.

Tabel 2. 6 Symbol *Activity* diagram

No.	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

14. *White box Testing*

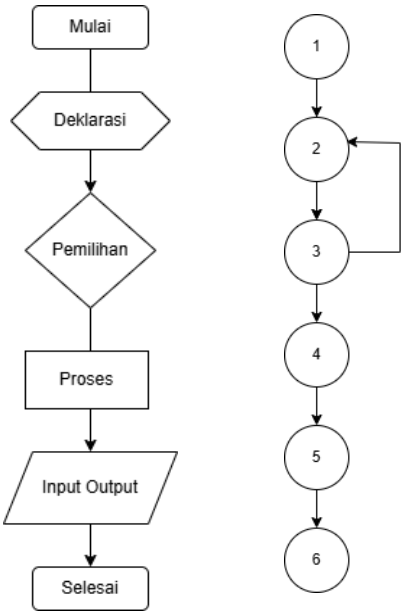
White box Testing disebut sebagai pemeriksaan struktur. Penguji dapat melihat dengan tepat program mana yang sedang diuji. Karena struktur internal kode diketahui dengan menguji setiap bagian yang dapat dievaluasi untuk mengidentifikasi masalah logis dalam kode sumber perangkat lunak, pengujian dibuat dari sudut pandang pengembang. Jovanovic dalam jurnal (Praniffa et al., 2023).

White box testing memiliki belebihan, seperti mampu menghapus bagian asing dari kode –kode yang tersembunyi, melakukan pengujian secara menyeluruh

karena seluruh bagian dari struktur atau logika dieksplorasi, membantu dalam mengoptimalkan kode, dan pengujian *white box testing* dapat dimulai walaupun GUI masih dalam masa pengembangan(Praniffa et al., 2023).

Adapun contoh dari pengujian *white box* dapat kita lihat di bawah

a. Pertama membuat flowchart dan flowgram activity user



Gambar 2. 2 *Flowchart dan Flowgraph*

b. Kemudian menghitung *Cyclomatic Complexcity* V(G) dari *Edge* dan Node dengan rumus

c. V(G) = E – N + 2

d. Ket :

E = Edge

N = Node

e. Menentukan region berdasarkan *Cyclomatic Complexcity* V(G)

f. Membuat Independent Path Berdasarkan flowgraph

g. Terakhir membuat matriks aktivasi user berdasarkan Independent Path

15. *Black box Testing*

Metode *Black box Testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan. Estimasi banyaknya data uji dapat dihitung melalui banyaknya field data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid. Mz, MM, Hastuty, A., & Yunus, M.

Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian pada sistem menggunakann metode *Black Box*, tujuannya mengetahui kelemahan dari sistem agar data yang dihasilkan sesuai dengan data yang dimasukkan setelah data dieksekusi dan menghindari kekurangan dan kesalahan pada aplikasi sebelum digunakan oleh user

Tujuan *Black box* Testing adalah untuk mengevaluasi setiap fitur infrastruktur program aplikasi. *Black box Testing* bertujuan untuk menentukan apakah perangkat lunak beroperasi sebagaimana mestinya tanpa memiliki akses ke kode sumber yang sebenarnya (Ismail & Efendi, 2020). Contoh table *blackbox* sebagai berikut:

Tabel 2. 7 Contoh *Blackbox*

Test Faktor	**Hasil**	**Kesimpulan**
Halaman yang diuji	Berhasil/gagal	Penjelesan jika berhasil atau tidak
Screenshoot		
Source Gambar atau Screenshoot Halaman yang diuji		

B. Kajian Hasil Penelitian Terdahulu

Untuk dapat digunakan sebagai data pendukung, diperlukan berbagai penelitian sebelumnya. Studi sebelumnya yang berkaitan dengan masalah yang dibahas dalam penelitian ini merupakan salah satu informasi pendukung yang peneliti butuhkan. Dalam hal ini, referensi yang berkaitan dengan tantangan teknologi informasi dari studi sebelumnya digunakan. Oleh karena itu penelitian dilakukan secara online dengan menggunakan berbagai temuan penelitian yang disajikan dalam makalah akhir, tesis, atau publikasi.

(Restu Panji Amarta, Budiman, 2021) Pengembangan Sistem Informasi Perpustakaan Berbasis *QR code* (*Quick Response Code*). Penelitian ini bertujuan untuk meningkatkan kualitas layanan pada perpustakaan Fakultas Teknik Universitas Majalengka. Karena dengan menggunakan *QR code* dapat memudahkan atau mempersingkat waktu bagi mahasiswa yang ingin meminjam buku. Metode yang digunakan pada penelitian ini yaitu Metode Lapangan *(Field Research*), Metode Kepustakaan *(Library Researc*), *Unified Modeling Language (UML), Rapid Application Development (RAD*). Berdasarkan penelitian yang dilakukan terhadap sistem informasi perpustakaan berbasis *QR code* (*Quick Response Code*) sebagai berikut Sistem Informasi Perpustakaan Berbasis *QR code* (*Quick Response Code*)yang telah dibuat berhasil untuk pendaftaran anggota,

peminjaman buku, pengembalian buku dan pelaporan, Fungsi dari sistem informasi perpustakaan ini berjalan dengan baik dan sesuai yang diharapkan oleh peneliti.

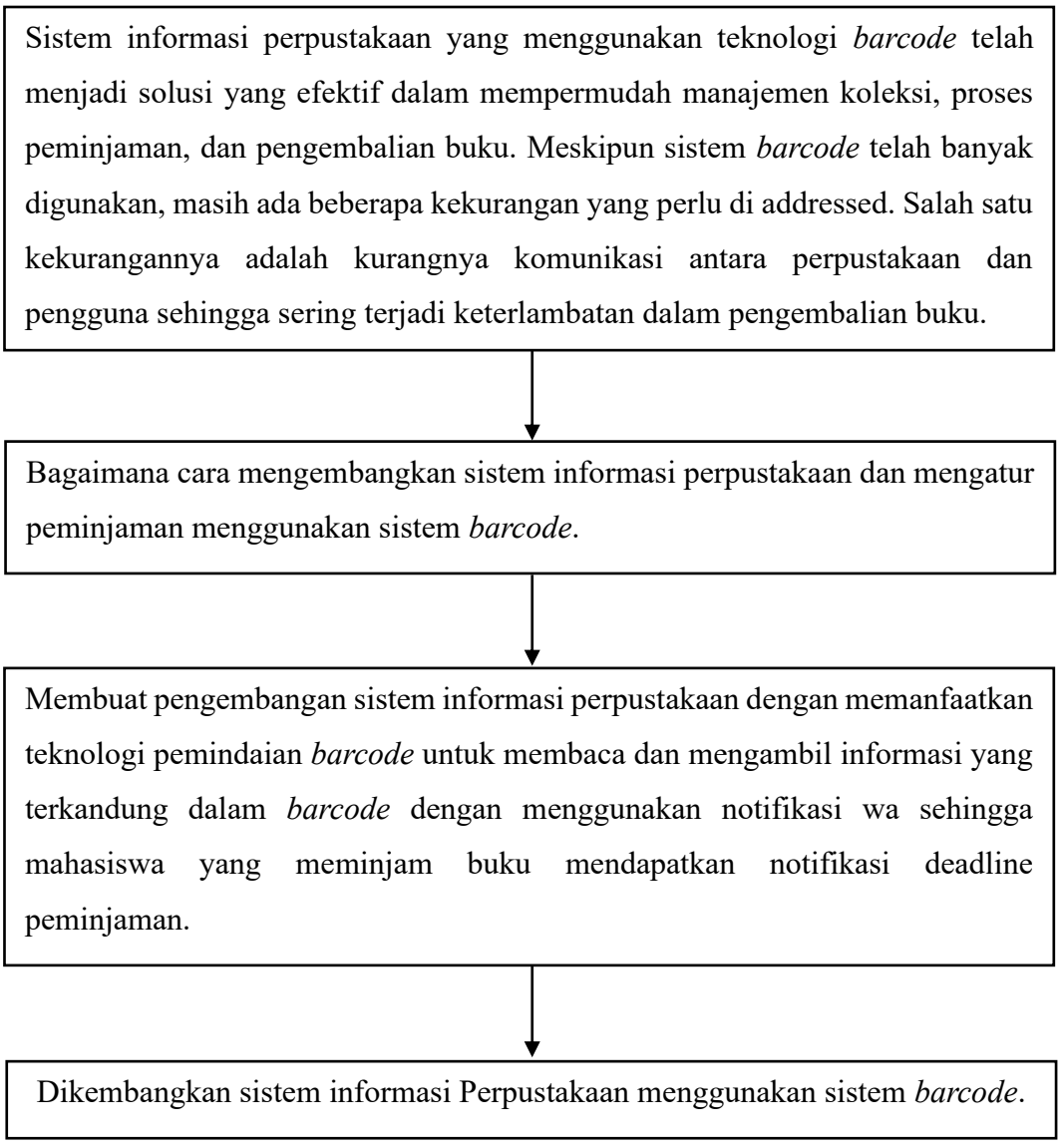
(Asdar, dalam Muhammad Zikrurridho Afwani, Royana Afwani, Farid Wahyudi, 2021) Sistem Informasi Perpustakaan Sekolah SMPN 4 Gunungsari. Penelitian ini bertujuan memudahkan dalam pengelolahan data,peminjaman buku, pengembalian buku dan manajemen buku. Metode yang digunakan adalah *waterfall* dengan beberapa tahapan yang runtut dari analisis kebutuhan, desain sistem, *coding* dan testing, penerapan program, serta pemeliharaan. Penelitian ini menggunakan penelitian lapangan (*Field Research*) dan kajian kepustakaan (*Literature study*).

(Hadid Putri B Zurna, Faiza Rini, Ade Pratama 2022) Sistem Informasi Perpustakaan Berbasis Web. Tujuan penelitian yang akan dicapai adalah untuk memudahkan peminjaman dan pengembalian buku pada perpustakaan SMP Negeri 1 Lubuk Sikaping diantaranya adalah siswa, guru, dan tenaga kerja. Metode penelitian yang digunakan *System Development Life Cycle (SDLC)* adalah beberapa langkah yang menunjukan seluruh proses untuk membangun sebuah sistem. Adapun harapanya semoga perpustakaan smp negeri 1 lubuk sikaping dapat meningkat dengan semakin singkatnya waktu pengerjaan, menigkatnya keakuratan pengelolaan data dan memotivasi staf perpustakaan dalam pekerjaannya.

(Satria Novari, Taufik Sobri, Anika Dayaka, 2021) Sistem Informasi Perpustakaan Berbasis Dekstop Pada SMK NEGERI 4 Ogan Komering Ulu Menggunakan *Embarcadero XE2 dan MySql* Analisis. Penelitian ini bertujuan untuk memudahkan staf perpustakaan dalam mengelolaan pencarian dan peminjaman buku. Penelitian ini menggunakan metode observasi, referensi dan

interview. Teknik pengumpulan data dalam penelitian ini melalui dengan cara dilakukan pengamatan dan tanya jawab langsung kepada pihak-pihak sekolah SMK 4 Negeri OKU.

C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian deskriptif, dimana memberikan gambaran mengenai fenomena yang sesungguhnya terjadi dan menggunakaan pendekatan kualitatif. Metode ini dimulai dengan mengumpulkan data, menganalisis data dan menginterpresentasikannya.

B. Lokasi dan Waktu Penelitian

Penelitian dilakukan di perpustakaan Universitas Muhammadiyah Parepare dan waktu penelitian dilakukan selama ± 3 (dua) bulan di tahun 2024.

C. Alat dan Bahan Penelitian

1. Perangkat keras

a. Laptop Acer Aspire Vero AV16 Ulltra 7 155U:

1) *Processor* : Intel® Celeron N4020

2) *Installed RAM* : 4GB

3) *HDD* : 1TB

2. Perangkat lunak

a. Windows 10 *Home* Single Language

b. Visual Studio Code

c. Delphi

d. Whatsapp Gateway

D. Teknik Pengumpulan Data

1. Metode Wawancara

Penulis melakukan tanya jawab dan wawancara dengan pihak yang dibutuhkan dalam penelitian ini seperti staf perpustakaan dan mahasiswa sehingga dari pertanyaan tersebut diharapkan dapat memberikan sebuah informasi yang dapat digunakan dalam penelitian.

2. Observasi

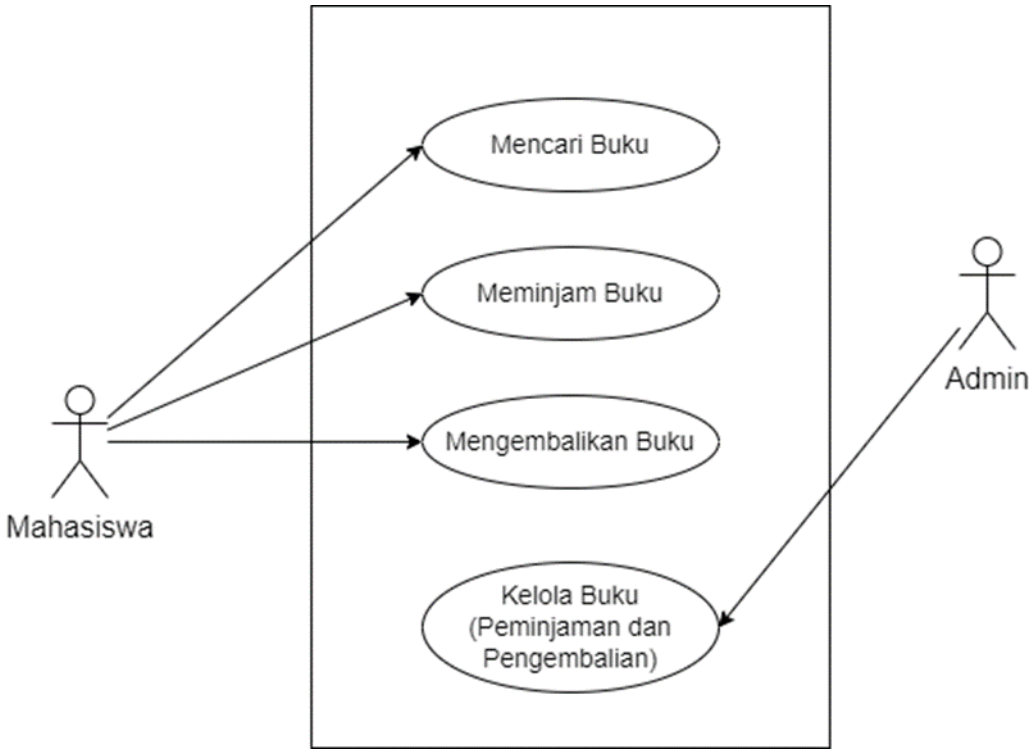
Observasi merupakan teknik pengumpulan data dengan cara mengamati secara langsung maupun tidak langsung. Observais tidak langsung dilakukan dengan cara mengamati objek penelitian melalui media tertentu, seperti video atau foto.

3. Dokumentasi

Pada tahap ini dilakukan mengumpulkan data yang berkaitan dalam bentuk dokumen. Dokumen-dokumen tersebut dapat berupa dokumen tertulis, dokumen elektronik.

E. Rancangan Sistem

1. Sistem yang berjalan

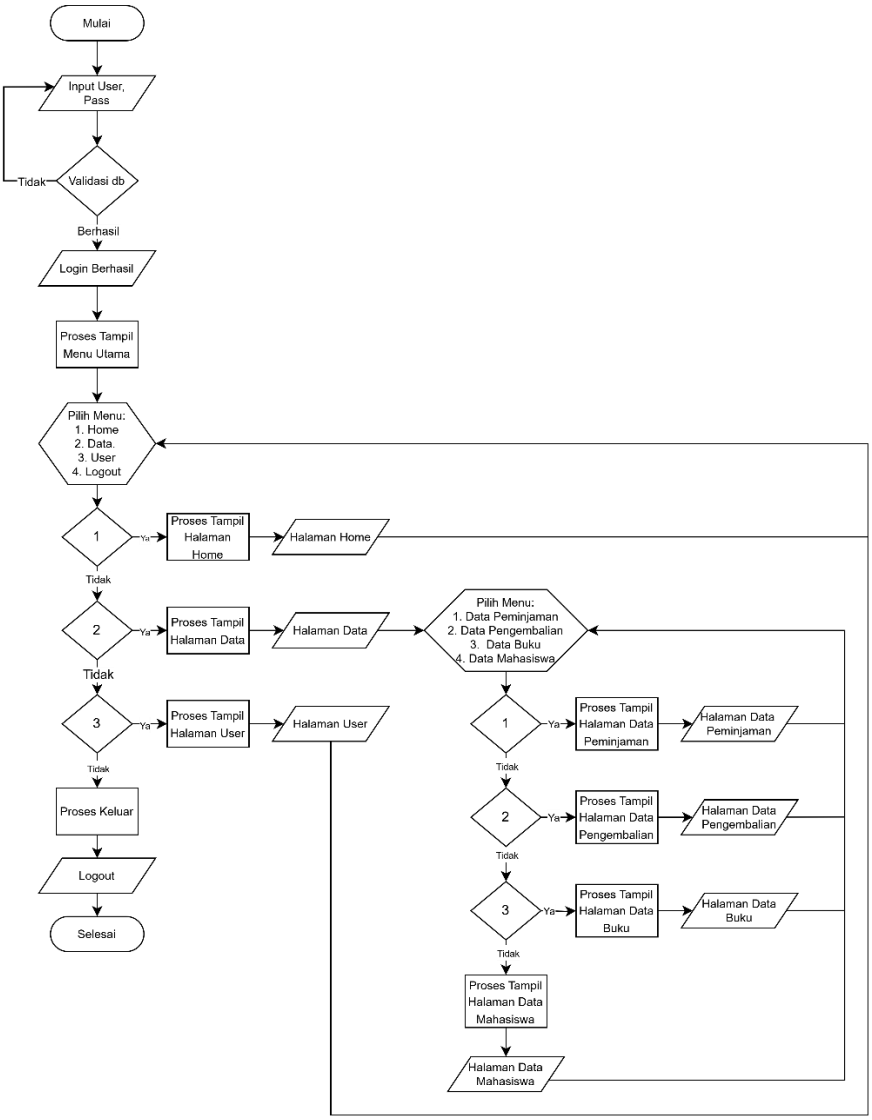


Gambar 3. 1 Rancangan sistem yang berjalan

Sistem peminjaman buku di perpustakaan dilakukan melalui interaksi langsung antara mahasiswa dan petugas perpustakaan. Proses dimulai ketika mahasiswa datang ke perpustakaan dan mencari buku yang dibutuhkan. Setelah menemukan buku yang dicari, mahasiswa membawa buku tersebut ke meja layanan. Di sana, petugas perpustakaan mencatat data peminjaman secara manual, biasanya menggunakan kartu peminjaman atau buku log. Mahasiswa kemudian diperbolehkan membawa buku tersebut selama periode yang telah ditentukan. Ketika masa peminjaman selesai, mahasiswa kembali ke perpustakaan untuk mengembalikan buku.

2. Sistem yang diusulkan

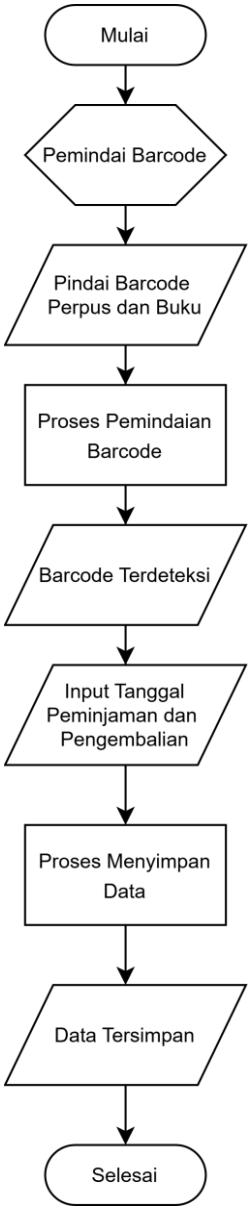
a. Menu Sistem



Gambar 3. 2 Flowchart Menu Sistem

Pada gambar di atas menggambarkan alur kerja sistem informasi perpustakaan mulai dari proses login hingga navigasi menu dan sub-menu. Proses diawali dengan pengguna melakukan input username dan password, kemudian sistem melakukan validasi data ke database. Apabila validasi gagal, pengguna tidak dapat mengakses sistem dan proses berhenti. Namun, jika validasi berhasil, sistem menampilkan notifikasi login berhasil dan pengguna diarahkan ke halaman menu utama. Pada menu utama, tersedia empat pilihan yaitu Home, Data, User, dan Logout. Jika pengguna memilih menu Home, sistem akan menampilkan halaman utama (home page). Apabila memilih menu Data, sistem menampilkan halaman data yang memiliki empat sub-menu, yaitu Data Peminjaman, Data Pengembalian, Data Buku, dan Data Mahasiswa. Masing-masing sub-menu ini menampilkan halaman yang sesuai dengan data yang dipilih. Jika pengguna memilih menu User, sistem akan menampilkan halaman data pengguna. Sedangkan jika memilih Logout, sistem akan menjalankan proses keluar, menutup sesi pengguna, dan mengakhiri proses dengan status selesai. Alur ini menunjukkan bahwa sistem dirancang untuk memberikan navigasi yang terstruktur, memudahkan pengguna dalam mengakses informasi sesuai kebutuhan, serta menjaga keamanan akses melalui proses login dan validasi.

b. Sistem peminjaman

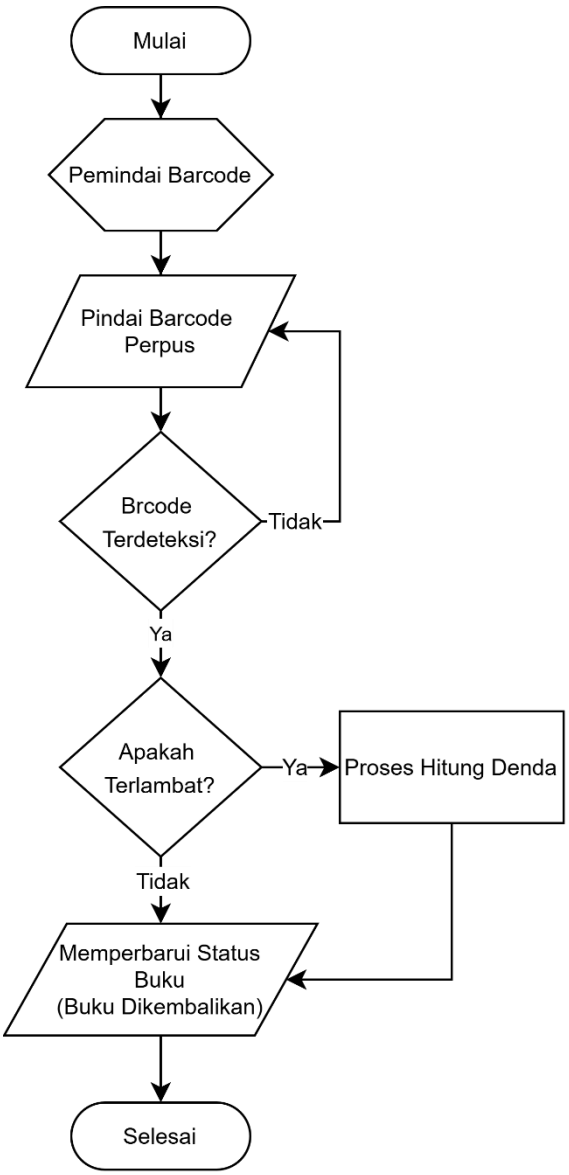


Gambar 3. 3 Sistem peminjaman yang diusulkan

Proses peminjaman buku di atas menggambarkan alur proses peminjaman buku menggunakan sistem pemindaian barcode di perpustakaan. Proses dimulai dengan tahap awal (Mulai) di mana pengguna atau petugas menggunakan alat pemindai barcode. Selanjutnya, dilakukan

proses pemindaian barcode pada buku dan kartu anggota perpustakaan. Sistem kemudian memproses hasil pemindaian untuk mendeteksi barcode yang terbaca. Setelah barcode terdeteksi, pengguna diminta untuk memasukkan tanggal peminjaman dan tanggal pengembalian buku. Data yang telah dimasukkan kemudian diproses oleh sistem untuk disimpan ke dalam database. Setelah proses penyimpanan selesai, status data tersimpan akan ditampilkan sebagai konfirmasi. Proses kemudian diakhiri dengan tahap selesai. Alur ini menunjukkan bahwa sistem dirancang untuk mempermudah dan mempercepat proses peminjaman buku melalui teknologi barcode, sekaligus memastikan keakuratan data peminjaman dan pengembalian.

c. Sistem pengembalian

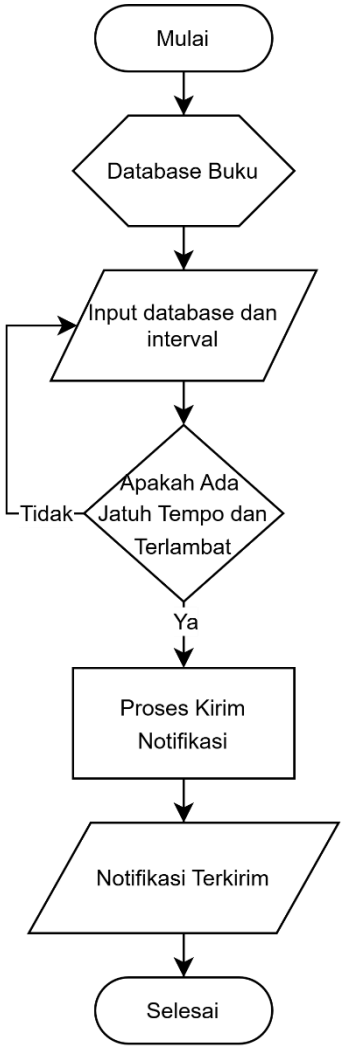


Gambar 3. 4 Sistem pengembalian yang diusulkan

Pada gambar di atas menjelaskan alur proses pengembalian buku di perpustakaan dengan menggunakan sistem pemindaian barcode. Proses dimulai dari tahap awal (Mulai), di mana petugas atau pengguna melakukan pemindaian barcode pada buku yang akan dikembalikan menggunakan alat pemindai barcode. Sistem kemudian memproses hasil pemindaian untuk

mendeteksi apakah barcode buku terdeteksi atau tidak. Jika barcode tidak terdeteksi, proses berhenti karena data buku tidak ditemukan. Namun, jika barcode terdeteksi, sistem akan memeriksa apakah pengembalian buku dilakukan tepat waktu atau terlambat. Apabila pengembalian terlambat, sistem menjalankan proses perhitungan denda sesuai dengan kebijakan perpustakaan. Setelah itu, atau jika buku dikembalikan tepat waktu, sistem memperbarui status buku menjadi “Dikembalikan” pada database. Proses kemudian diakhiri dengan status selesai. Alur ini menunjukkan bahwa sistem dirancang untuk memastikan proses pengembalian buku berjalan efisien, sekaligus dapat menghitung denda keterlambatan secara otomatis dan memperbarui status buku secara real-time.

d. Sistem notifikasi



Gambar 3. 5 Sistem notifikasi yang diusulkan

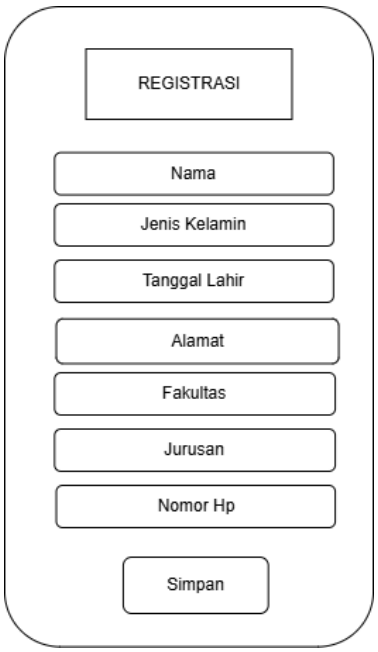
Sistem notifikasi yang diusulkan dalam pengembangan sistem informasi perpustakaan ini bertujuan untuk mengingatkan pengguna terkait batas waktu pengembalian buku dan menginformasikan apabila terjadi keterlambatan. Proses dimulai dengan tahap awal (Mulai), di mana sistem mengakses database buku untuk memperoleh data peminjaman. Selanjutnya, dilakukan proses input database dan interval yang berisi data batas waktu pengembalian (jatuh tempo) serta periode pengecekan yang

telah ditentukan. Setelah itu, sistem memeriksa apakah terdapat buku yang sudah mendekati jatuh tempo atau telah terlambat dikembalikan. Jika tidak ada buku yang memenuhi kondisi tersebut, proses berakhir. Namun, jika ditemukan buku yang sudah jatuh tempo atau terlambat, sistem melanjutkan ke tahap proses kirim notifikasi kepada pengguna atau anggota perpustakaan yang bersangkutan. Setelah notifikasi berhasil terkirim, sistem menampilkan status notifikasi terkirim sebagai konfirmasi, kemudian proses diakhiri dengan tahap selesai. Alur ini dirancang untuk membantu perpustakaan dalam mengingatkan anggota secara otomatis mengenai kewajiban pengembalian buku, sehingga dapat mengurangi tingkat keterlambatan dan menjaga kelancaran sirkulasi koleksi buku

F. Desain Interface

Desain antarmuka sistem berfungsi sebagai alat bantu dalam proses merancang antarmuka sistem yang akan dikembangkan. Proses perancangan antarmuka sistem membantu pengembang dalam membuat representasi visual yang sederhana mengenai tampilan aplikasi yang akan dibuat. Desain antarmuka juga berperan penting dalam memastikan bahwa aplikasi mudah dan sesuai dengan kebutuhan pengguna. Tahapan ini biasanya dilakukan sebelum pengkodean dimulai, guna meminimalkan kesalahan dan memastikan bahwa pengalaman pengguna (*user experience*) menjadi prioritas utama dalam pengembangan sistem.

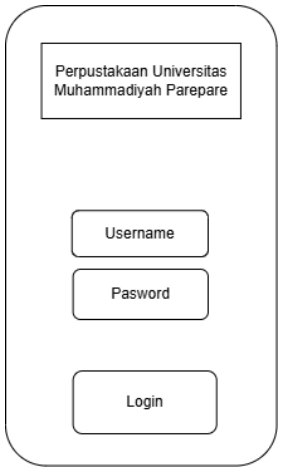
1. **Registrasi kartu perpustakaan**



Gambar 3. 6 Desain registrasi kartu perpustakaan

Pada gambar diatas menjelaskan tampilan registrasi dimana mahasiswa belum memiliki kartu perpustkaan. Pada tahap ini, mahasiswa akan mengisi data diri seperti nama, jenis kelamin, tanggal lahir alamat, fakultas, jurusan, nomor handphon (pastikan aktif dan memiliki *whatsapp*.

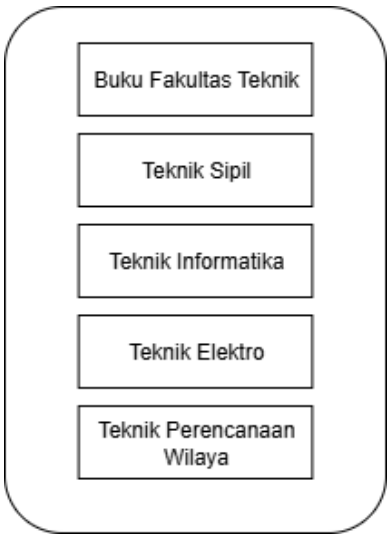
2. ***Login***



Gambar 3. 7 Desain *interface login*

Gambar diatas merupakan tampilan *login* ketika sudah mempunyai akun, pastikan username dan password anda benar.

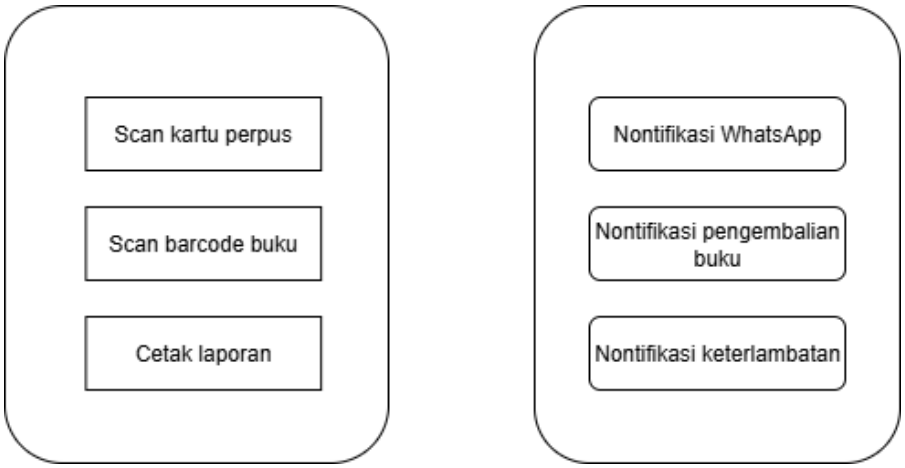
3. Data buku



Gambar 3. 8 Desain *interface* data buku

Halaman ini menampilkan daftar buku yang tersedia di perpustakaan. Informasi buku meliputi judul, pengarang, tahun terbit, dan status ketersediaan.

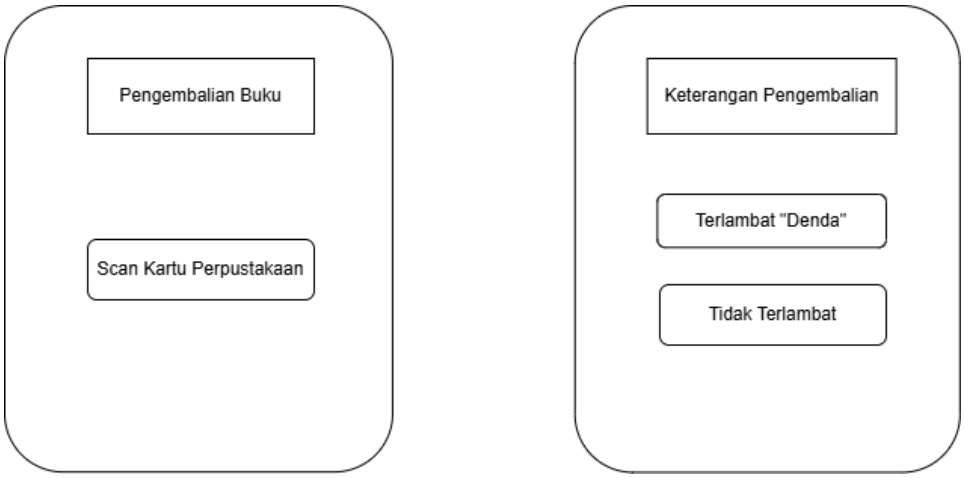
4. Peminjaman buku



Gambar 3. 9 Desain *interface* peminjaman buku

Pada halaman ini, mahasiswa dapat melakukan proses peminjaman buku. Mahasiswa dapat memilih buku yang tersedia dan akan menerima notifikasi pengingat pengembalian buku -1 hari sebelum hari terakhir peminjaman buku.

5. Pengembalian buku



Gambar 3. 10 Desain *interface* pengembalian buku

Tampilan ini memungkinkan pengguna untuk mengembalikan buku yang telah dipinjam. Sistem akan mencatat tanggal pengembalian dan menghitung apakah terdapat keterlambatan yang mungkin dikenai denda.

BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Aliran Data UML

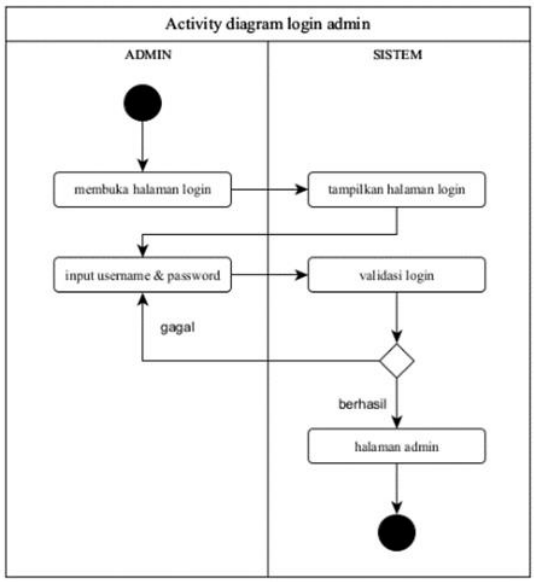
Dalam bagian ini, akan dibahas dua *diagram* UML yang menggambarkan aliran data dan interaksi dalam Sistem Informasi Perpustakaan Menggunakan Sistem *Barcode*, yaitu *Activity diagram* dan *Sequence diagram*.

1. *Activity diagram*

Activity diagram digunakan untuk memodelkan aliran kerja atau aliran kontrol dari sebuah proses bisnis atau *use case*. *Diagram* ini menunjukkan aktivitas-aktivitas yang terjadi dalam sistem dan urutan eksekusinya.

a. *Activity diagram admin*

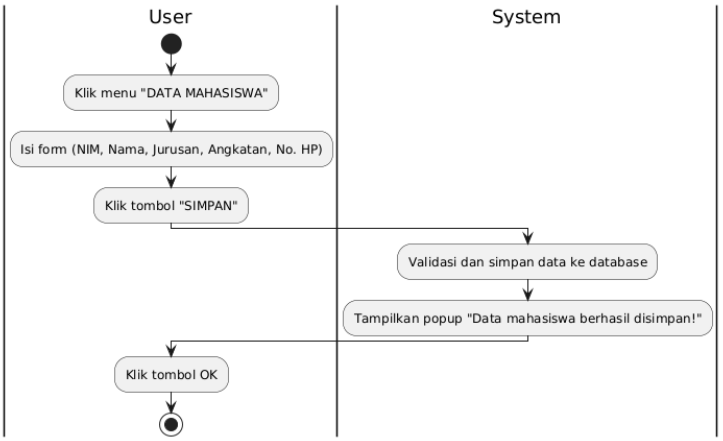
1) *Activity diagram login*



Gambar 4. 1 *Activity diagram login*

Pada gambar 4. 1 menjelaskan cara masuk sebagai *admin*. *Admin* harus terlebih dahulu membuka aplikasi perpustakaan, setelah itu sistem akan menampilkan *form login* dan *admin* harus memasukkan nama pengguna dan kata sandi. Sistem kemudian akan melakukan validasi; jika informasi benar, maka akan masuk ke halaman *admin*; jika tidak, maka akan kembali ke halaman *login*.

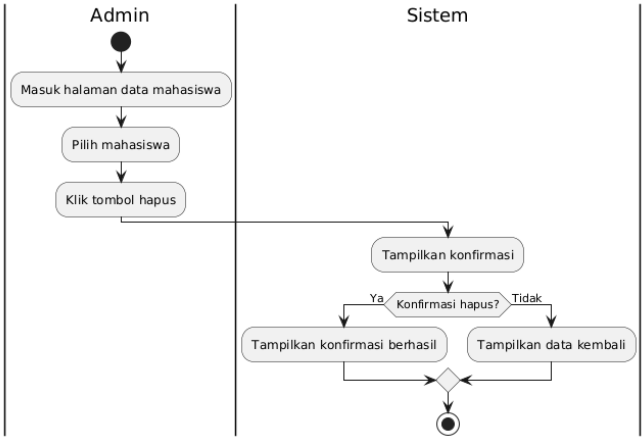
2) *Activity diagram* tambah data mahasiswa



Gambar 4. 2 *Activity diagram* tambah mahasiswa

Pada gambar 4. 2 *Activity diagram* ini menggambarkan proses penambahan data mahasiswa pada Sistem Informasi Perpustakaan. Proses dimulai saat pengguna memilih menu "DATA MAHASISWA", kemudian mengisi form yang berisi NIM, nama, jurusan, angkatan, dan nomor HP. Setelah itu, pengguna menekan tombol "SIMPAN". Sistem akan memvalidasi dan menyimpan data ke dalam database, lalu menampilkan notifikasi bahwa data berhasil disimpan. Pengguna menekan tombol "OK" pada popup untuk mengakhiri proses.

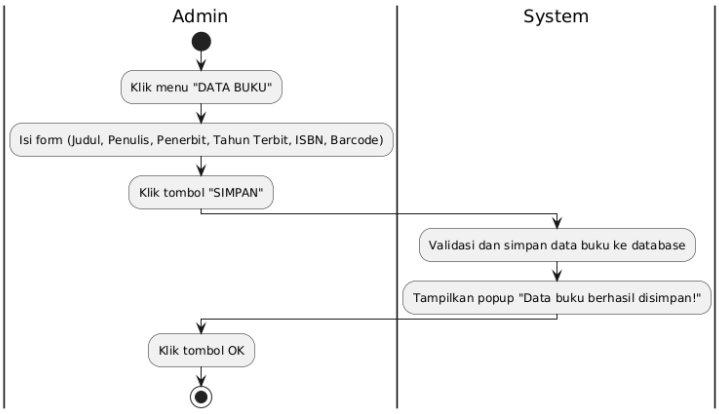
3) *Activity diagram* hapus mahasiswa



Gambar 4. 3 *Activity diagram* hapus data mahasiswa

Pada gambar 4. 3 menjelaskan prosedur yang digunakan *admin* untuk menghapus data mahasiswa. Halaman *admin* ditampilkan oleh sistem ketika telah dibuka oleh *admin*. *Admin* kemudian memilih menu yang datanya akan dimusnahkan, *admin* mengklik tombol hapus data mahasiswa. Sistem kemudian akan menampilkan konfirmasi penghapusan. *Admin* kemudian melakukan konfirmasi. Jika admin konfirmasi, data akan terhapus, jika tidak maka data batal dihapus.

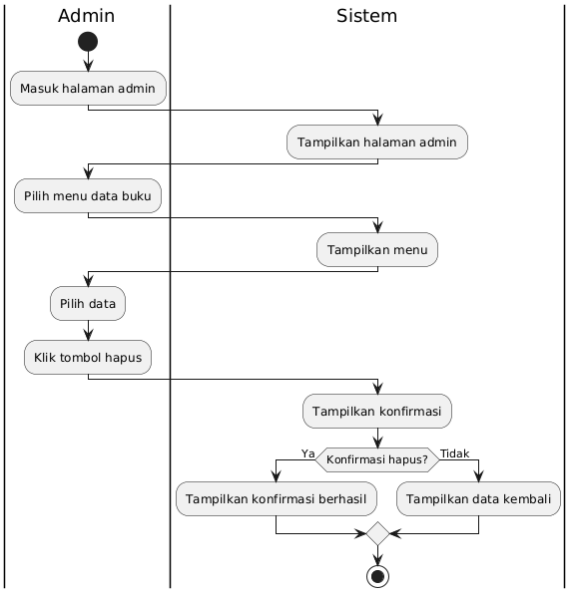
4) *Activity diagram* tambah buku



Gambar 4. 4 *Activity diagram* tambah buku

Pada gambar 4. 4 menggambarkan alur penambahan data buku pada Sistem Informasi Perpustakaan. Proses dimulai ketika pengguna memilih menu "DATA BUKU", lalu mengisi form yang terdiri dari judul, penulis, penerbit, tahun terbit, ISBN, dan *barcode*. Setelah itu, pengguna menekan tombol "SIMPAN". Sistem akan memvalidasi dan menyimpan data buku ke dalam *database*, kemudian menampilkan notifikasi bahwa data berhasil disimpan. Proses ditutup dengan pengguna menekan tombol "OK" pada popup konfirmasi.

5) *Activity diagram* hapus buku

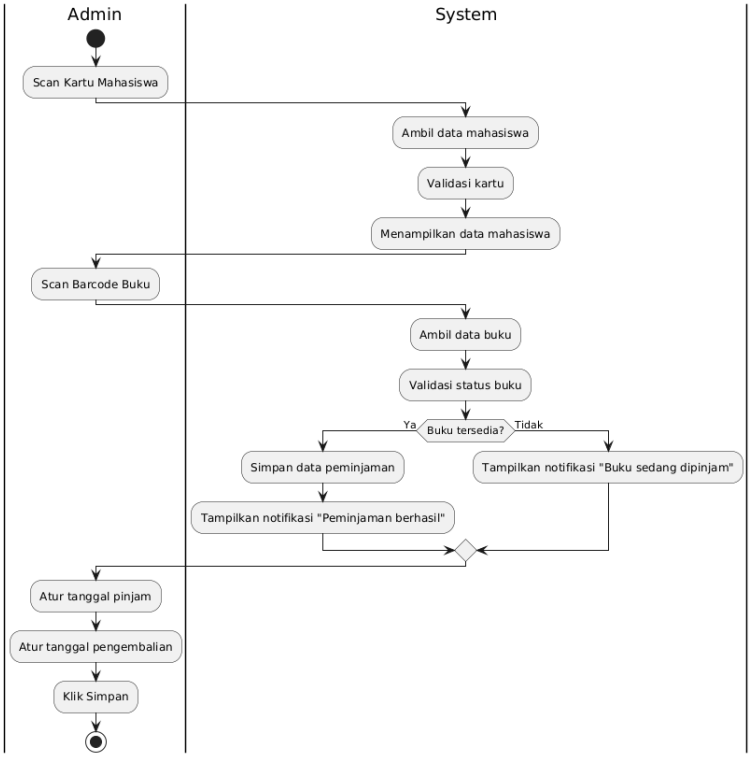


Gambar 4. 5 *Activity diagram* hapus buku

Pada gambar 4.5 menjelaskan prosedur yang digunakan *admin* untuk menghapus data buku. Halaman *admin* ditampilkan oleh sistem ketika telah dibuka oleh *admin*. *Admin* kemudian memilih menu yang datanya akan dimusnahkan, *admin* mengklik tombol hapus. Sistem kemudian akan menampilkan konfirmasi penghapusan. *Admin*

kemudian melakukan konfirmasi. Jika admin konfirmasi, data akan terhapus, jika tidak maka data batal dihapus.

6) *Activity diagram* peminjaman buku

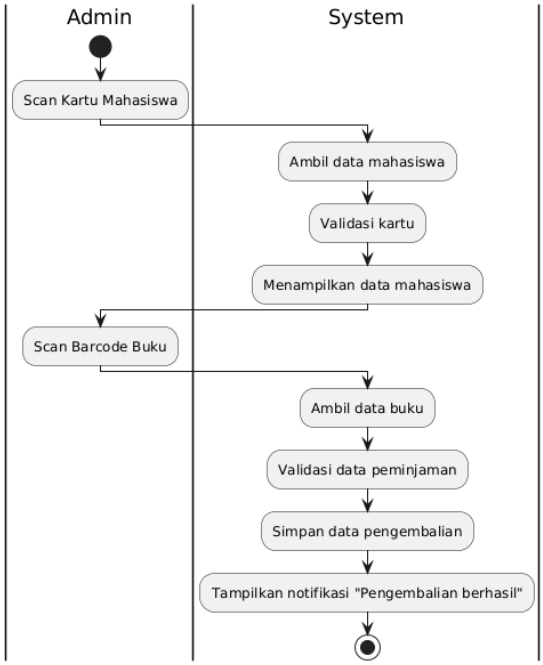


Gambar 4. 6 *Activity diagram* peminjaman buku

Pada Gambar 4.6 *Activity diagram* ini menggambarkan alur proses peminjaman buku di sistem perpustakaan. Proses dimulai ketika admin melakukan pemindaian kartu mahasiswa. Sistem kemudian mengambil dan memvalidasi data mahasiswa serta menampilkan informasi tersebut. Selanjutnya, admin memindai *barcode* buku yang ingin dipinjam. Sistem mengambil data buku dan memeriksa ketersediaannya. Jika buku tersedia, sistem menyimpan data peminjaman dan menampilkan notifikasi bahwa peminjaman berhasil. Jika buku tidak tersedia, sistem menampilkan notifikasi bahwa buku

sedang dipinjam. Setelah itu, admin mengatur tanggal pinjam dan tanggal pengembalian, lalu mengklik tombol simpan untuk menyelesaikan proses peminjaman buku.

7) *Activity diagram* pengembalian buku



Gambar 4. 7 *Activity diagram* pengembalian buku

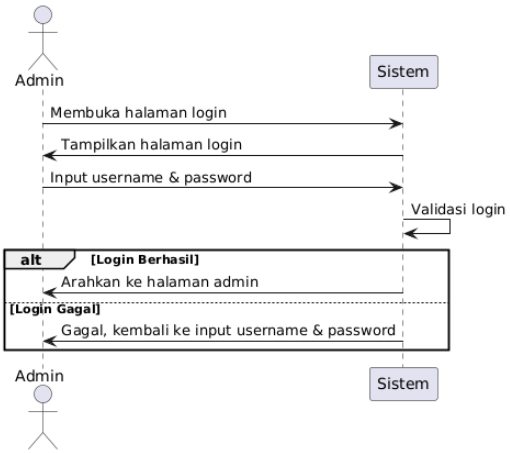
Pada gambar 4.7 ini proses pengembalian, admin memindai *barcode* buku yang dikembalikan. Sistem mengambil data buku dan memverifikasi statusnya. Setelah itu, sistem menghitung apakah terdapat keterlambatan dan menentukan denda jika perlu. Kemudian data pengembalian disimpan, dan sistem menampilkan notifikasi bahwa pengembalian berhasil dilakukan.

2. *Sequence diagram*

Sequence diagram merupakan *diagram* yang menggambarkan interaksi antara objek-objek dalam sistem secara berurutan berdasarkan waktu. *Diagram* ini menunjukkan bagaimana objek-objek saling bertukar pesan atau data untuk menyelesaikan suatu tugas atau *use case*.

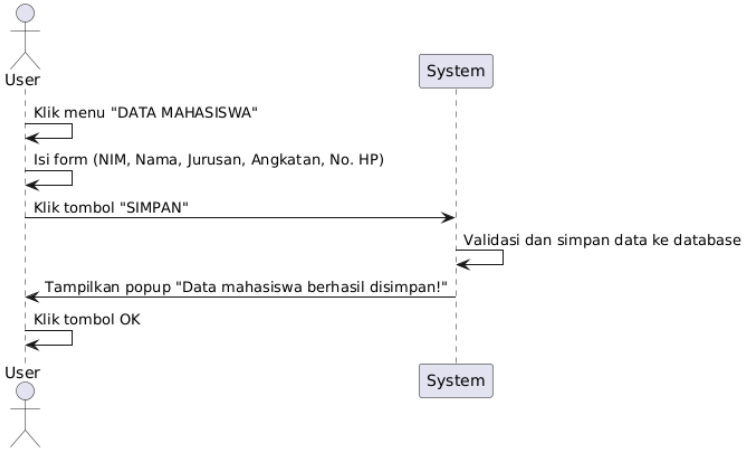
a. *Sequence diagram admin*

1) *Sequence diagram login*



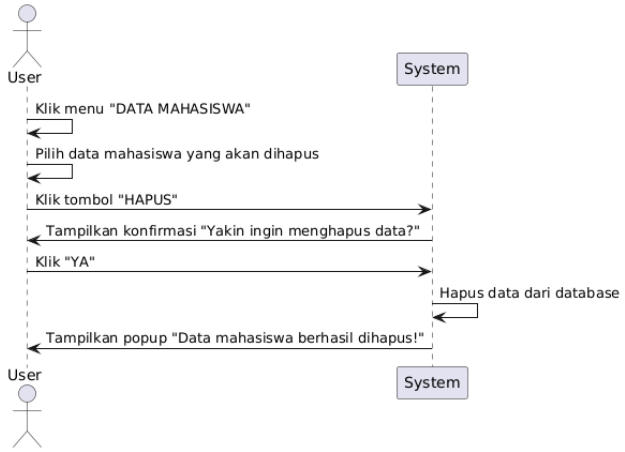
Gambar 4. 8 *Sequence diagram login*

2) *Sequence diagram* tambah data mahasiswa



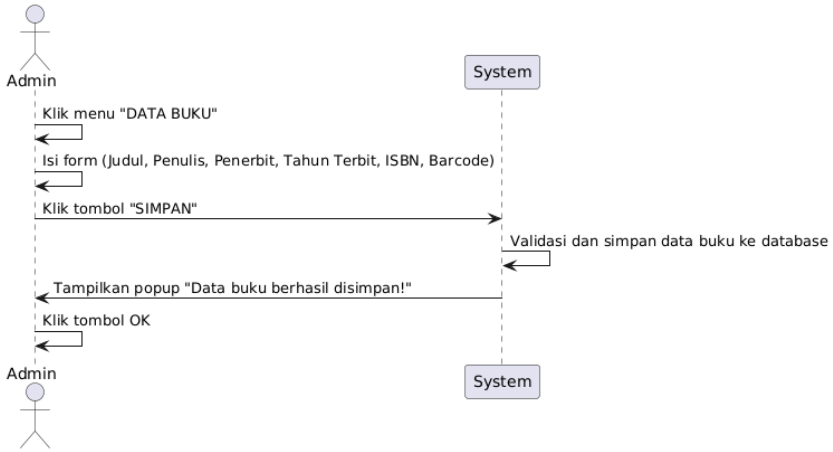
Gambar 4. 9 *Sequence diagram* tambah data mahasiswa

3) *Sequence diagram* hapus data mahasiswa



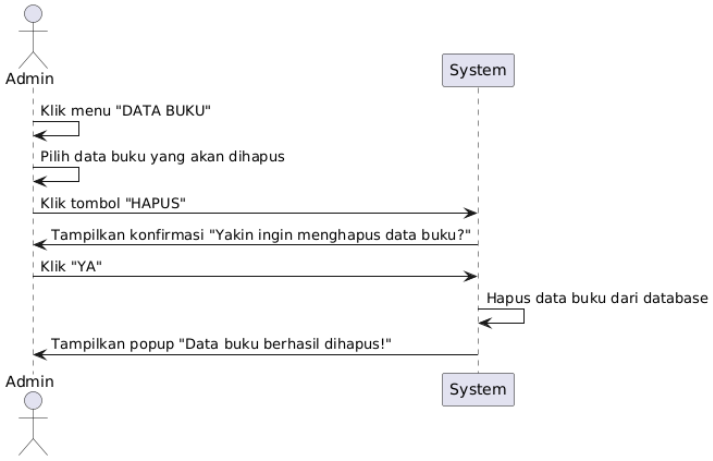
Gambar 4. 10 *Sequence diagram* hapus data mahasiswa

4) *Sequence diagram* tambah buku



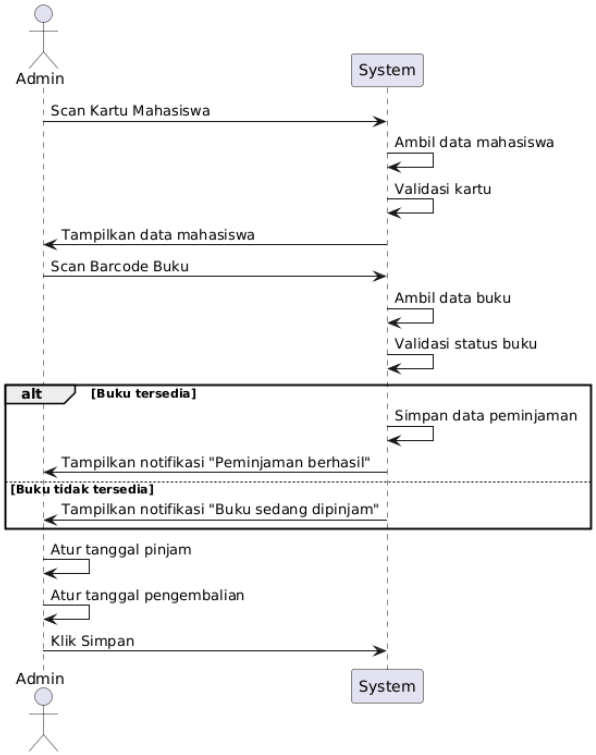
Gambar 4. 11 *Sequence diagram* tambah buku

5) *Sequence diagram* hapus buku



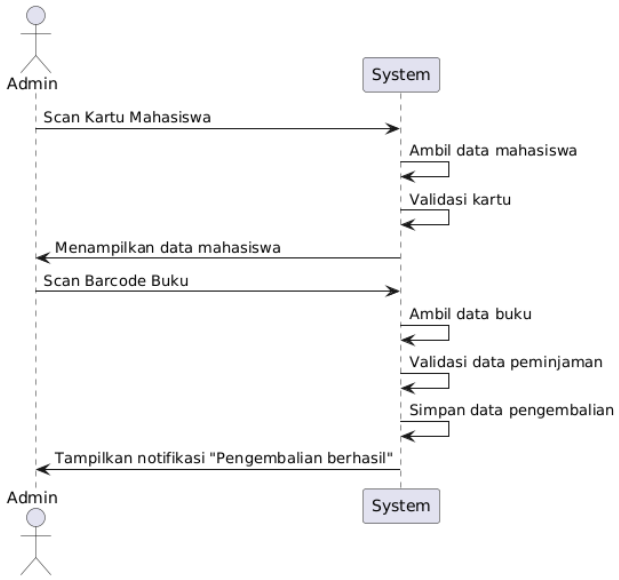
Gambar 4. 12 *Sequence diagram* hapus buku

6) *Sequence diagram* peminjaman buku



Gambar 4. 13 *Sequence diagram* peminjaman buku

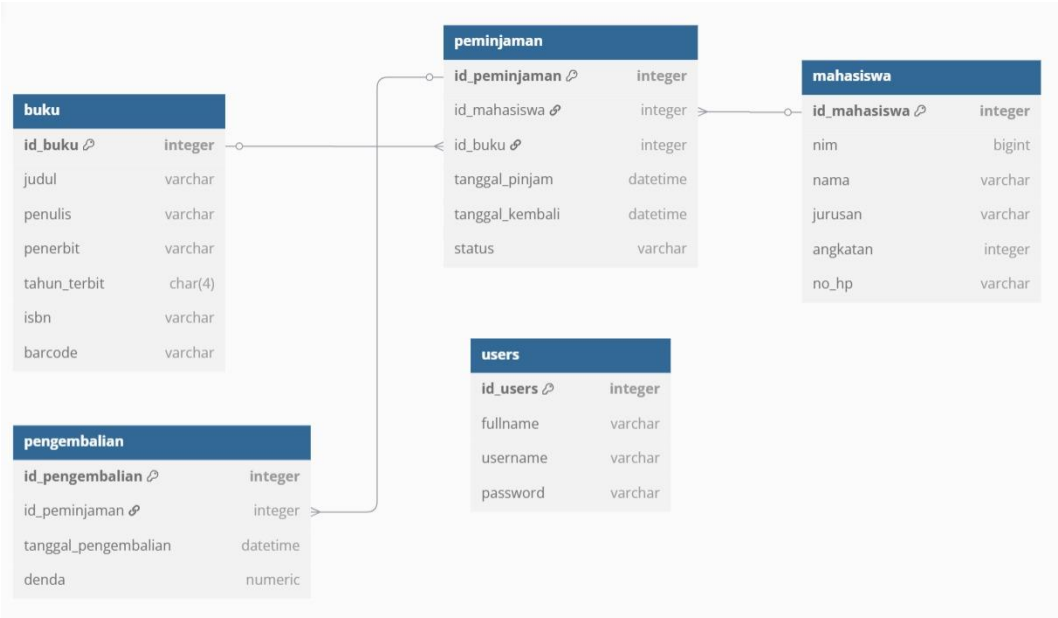
7) *Sequence diagram* pengembalian buku



Gambar 4. 14 *Sequence diagram* pengembalian buku

B. Rancangan Database

1. Relasi tabel



Gambar 4. 15 Relasi tabel *database*

Struktur *database* sistem informasi perpustakaan ini terdiri dari lima tabel utama, yaitu: mahasiswa, buku, peminjaman, pengembalian, dan *users*. Setiap tabel memiliki hubungan dan peran masing-masing dalam mendukung fungsi manajemen perpustakaan secara digital.

Tabel mahasiswa menyimpan data mahasiswa yang menjadi pengguna layanan perpustakaan. Setiap mahasiswa diidentifikasi dengan id_mahasiswa sebagai *primary key* dan memiliki atribut seperti nim, nama, jurusan, angkatan, dan no_hp.

Tabel buku berisi data koleksi buku yang tersedia di perpustakaan. Setiap *entry* buku diidentifikasi dengan id_buku sebagai *primary key* dan mencakup atribut seperti judul, penulis, penerbit, tahun_terbit, isbn, dan *barcode*.

Tabel peminjaman mencatat transaksi peminjaman buku oleh mahasiswa. Tabel ini memiliki id_peminjaman sebagai *primary key* dan menjalin relasi ke tabel mahasiswa melalui id_mahasiswa, serta ke tabel buku melalui id_buku. Tabel ini juga menyimpan informasi tanggal peminjaman (tanggal_pinjam), tanggal pengembalian (tanggal_kembali), dan status (misalnya: "dipinjam", "dikembalikan").

Tabel pengembalian menyimpan data pengembalian buku yang terhubung langsung ke tabel peminjaman melalui id_peminjaman. Tabel ini mencatat tanggal_pengembalian dan nilai denda jika terjadi keterlambatan.

Tabel *users* berisi data pengguna sistem (admin atau petugas perpustakaan) dengan id_users sebagai primary key. Atribut lainnya meliputi fullname, username, dan password.

2. Kamus Data

a. Tabel mahasiswa

Tabel 4. 1 Kamus data tabel mahasiswa

Column	*Type*	*Length*	*Description*
id_mahasiswa	*Integer*	-	*Primary key*, ID unik mahasiswa
nim	*Bigint*	-	Nomor Induk Mahasiswa (NIM)
nama	*varchar*	255	Nama lengkap mahasiswa
jurusan	*varchar*	100	Jurusan atau program studi
angkatan	*integer*	-	Tahun angkatan masuk
no_hp	*varchar*	20	Nomor hp mahasiswa

b. Tabel buku

Tabel 4. 2 Kamus data tabel buku

Column	*Type*	*Length*	*Description*
id_buku	*Integer*	-	*Primary key*, ID unik buku
judul	*varchar*	255	Judul buku
penulis	*varchar*	100	Nama penulis
penerbit	*varchar*	100	Nama penerbit
tahun_terbit	*char*	4	Tahun terbit buku
isbn	*varchar*	20	Nomor ISBN buku
barcode	*varchar*	50	*Barcode* untuk pemindaian buku
kategori	*varchar*	100	*Jurusan judul buku*

c. Tabel peminjaman

Tabel 4. 3 Kamus data tabel peminjaman

Column	*Type*	*Length*	*Description*
id_peminjaman	*integer*	-	*Primary key*, ID unik transaksi peminjaman
id_mahasiswa	*integer*	-	*Foreign key* ke tabel mahasiswa
id_buku	*integer*	-	*Foreign key* ke tabel buku
tanggal_pinjam	*datetime*	-	Tanggal saat buku dipinjam
tanggal_kembali	*datetime*	-	Tanggal jatuh tempo pengembalian buku
status	*varchar*	50	Status peminjaman (dipinjam/dikembalikan)

d. Tabel pengembalian

Tabel 4. 4 Kamus data tabel pengembalian

Column	***Type***	***Length***	***Description***
id_pengembalian	*integer*	-	*Primary key*, ID transaksi pengembalian
id_peminjaman	*integer*	-	*Foreign key* ke tabel peminjaman
tanggal_pengembalian	*datetime*	-	Tanggal pengembalian aktual
denda	*numeric*	-	Besaran denda jika terlambat

e. Tabel *users*

Tabel 4. 5 Kamus data tabel users

Column	***Type***	***Length***	***Description***
id_users	*integer*	-	*Primary key*, ID unik pengguna
fullname	*varchar*	100	Nama lengkap pengguna (admin/petugas)
username	*varchar*	50	Nama pengguna untuk *login*
password	*varchar*	255	Kata sandi pengguna

C. Detail Sistem

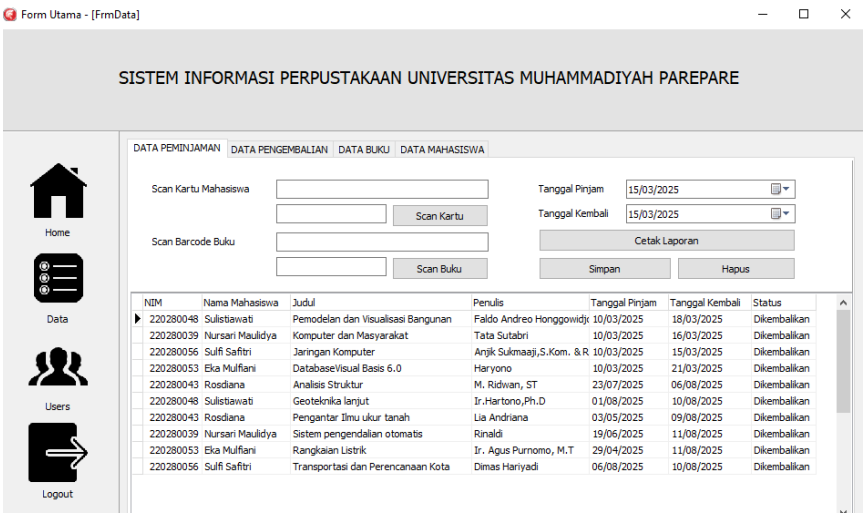
1. Halaman beranda

Halaman beranda sistem informasi perpustakaan Universitas Muhammadiyah Parepare menampilkan sambutan selamat datang dengan latar rak buku dan ilustrasi meja berisi buku serta alat tulis. Di bagian kiri terdapat menu navigasi berupa *Home*, Data, Users, dan Logout. Tampilan dirancang sederhana dan informatif untuk memudahkan pengguna mengakses fitur sistem.



Gambar 4. 16 Halaman beranda

2. Halaman data peminjaman buku



Gambar 4. 17 Halaman data peminjaman buku

Halaman Data Peminjaman Buku pada sistem informasi perpustakaan Universitas Muhammadiyah Parepare menampilkan antarmuka untuk mencatat dan mengelola data peminjaman buku. Di bagian atas terdapat formulir input yang memungkinkan petugas untuk mengisi data mahasiswa peminjam dengan memindai kartu mahasiswa serta memilih buku yang akan dipinjam. Formulir ini mencakup kolom Scan Kartu Mahasiswa, Scan Barcode Buku, Tanggal Pinjam, dan Tanggal Kembali, serta tombol Simpan dan Hapus untuk mengelola data transaksi. Di bawah formulir, terdapat tabel yang menampilkan daftar peminjaman yang telah dilakukan, dengan kolom NIM, Nama Mahasiswa, Judul Buku, Penulis, Tanggal Pinjam, Tanggal Kembali,dan Status. Selain itu, terdapat tombol Cetak Laporan untuk menghasilkan dokumen laporan peminjaman. Tabel ini menampilkan riwayat peminjaman dari beberapa mahasiswa, lengkap dengan status peminjaman seperti Dikembalikan atau Dipinjam.

Dibawah ini adalah cetakan dari laporan peminjaman buku diperpustakaan:



LAPORAN

NIM	NAMA	Judul Buku	Tanggal Pinjam
220280048	Sulistiawati	Pemodelan dan Visualisasi Bangunan	10/03/2025
220280039	Nursari Maulidya	Komputer dan Masyarakat	10/03/2025
220280056	Sulfi Safitri	Jaringan Komputer	10/03/2025
220280053	Eka Mulfiani	DatabaseVisual Basis 6.0	10/03/2025
220280043	Rosdiana	Analisis Struktur	23/07/2025
220280048	Sulistiawati	Geoteknika lanjut	01/08/2025
220280043	Rosdiana	Pengantar Ilmu ukur tanah	03/05/2025
220280039	Nursari Maulidya	Sistem pengendalian otomatis	19/06/2025
220280053	Eka Mulfiani	Rangkaian Listrik	29/04/2025
220280056	Sulfi Safitri	Transportasi dan Perencanaan Kota	06/08/2025

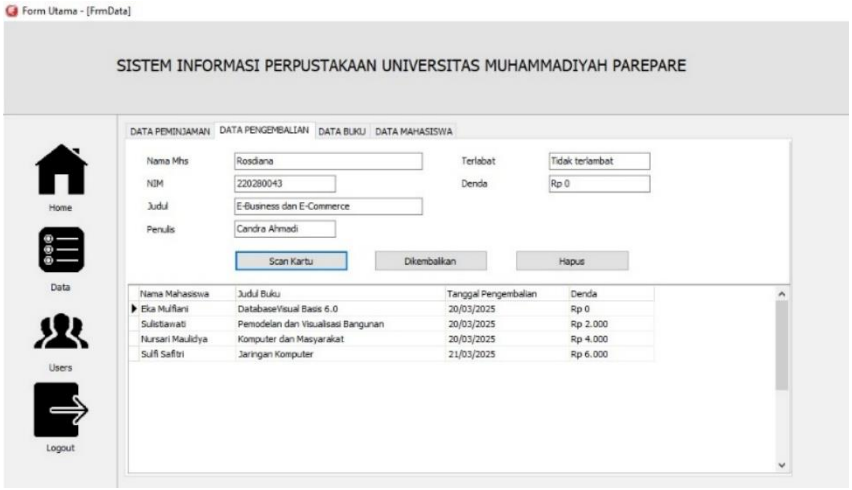
Gambar 4. 18 Laporan Peminjaman Buku

Laporan peminjaman buku pada sistem informasi perpustakaan merupakan keluaran (*output*) yang menampilkan data terkait aktivitas peminjaman oleh mahasiswa. Laporan ini berisi informasi berupa Nomor Induk Mahasiswa (NIM), nama peminjam, judul buku yang dipinjam, serta tanggal peminjaman. Data ini bertujuan untuk memudahkan pihak perpustakaan dalam memantau, mengelola, dan mengarsipkan proses peminjaman buku. Keberadaan laporan ini sangat membantu pustakawan dalam mengontrol masa pinjam, mengatur jadwal pengembalian, serta menyediakan arsip historis peminjaman yang dapat digunakan untuk analisis minat baca mahasiswa di perpustakaan.

3. Halaman pengembalian buku

Halaman Data Pengembalian Buku pada sistem informasi perpustakaan Universitas Muhammadiyah Parepare dirancang untuk mencatat proses pengembalian buku oleh mahasiswa. Di bagian atas terdapat formulir yang menampilkan informasi mahasiswa dan buku yang dikembalikan, meliputi Nama Mhs, NIM, Judul Buku, dan Penulis. Sistem juga menampilkan status keterlambatan melalui kolom Terlambat serta nominal Denda jika ada. Terdapat tombol Scan Kartu untuk mengambil data mahasiswa secara otomatis, serta tombol Dikembalikan dan Hapus untuk memproses atau menghapus data pengembalian. Di bagian bawah halaman ditampilkan tabel riwayat pengembalian buku yang berisi informasi Nama Mahasiswa, Judul Buku, Tanggal Pengembalian, dan Denda yang dikenakan. Tampilan ini membantu

petugas perpustakaan memantau status pengembalian buku secara efisien, termasuk mendeteksi keterlambatan dan menghitung denda.

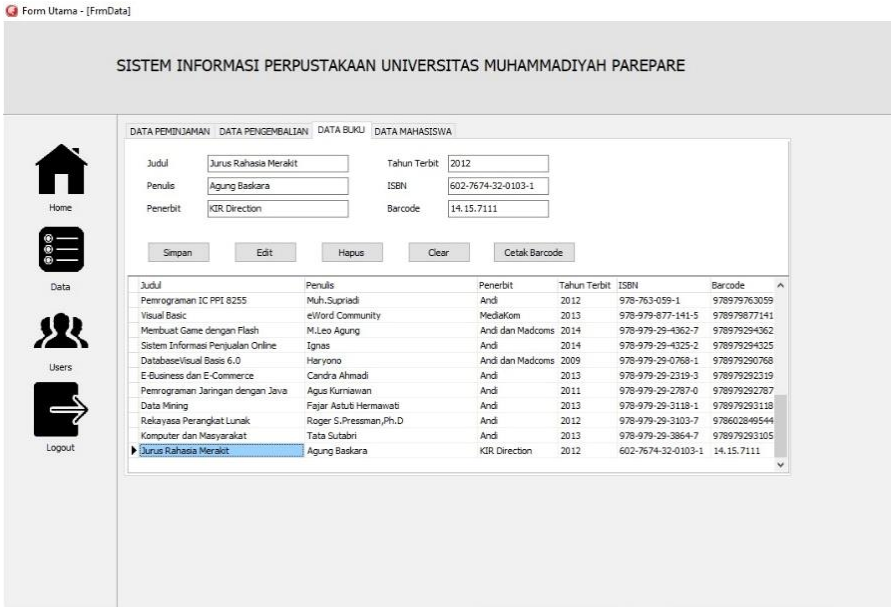


Gambar 4. 19 Halaman data peminjaman buku

4. Halaman data buku

Halaman Data Buku pada sistem informasi perpustakaan Universitas Muhammadiyah Parepare digunakan untuk mengelola informasi buku yang tersedia di perpustakaan. Di bagian atas halaman terdapat formulir input yang mencakup beberapa kolom seperti Judul, Penulis, Penerbit, Tahun Terbit, ISBN, dan *Barcode*. Pengguna dapat memasukkan data baru atau memperbarui data yang sudah ada dengan tombol Simpan, Edit, Hapus, dan Clear. Tersedia juga tombol Cetak *Barcode* untuk mencetak label *barcode* dari buku yang didaftarkan. Didalam data buku terdapat kategori yang ditandai dengan beberapa warna dan tiap warna mewakili ketegori dari buku khusus untuk jurusan tertentu, seperti jurusan teknik informatika warna biru, teknik sipil warna kuning, teknik elektro warna hijau dan teknik perencanaan wilayah

warna ungu. Di bagian bawah, terdapat tabel yang menampilkan daftar buku yang telah terdaftar dalam sistem, mencakup informasi lengkap seperti judul buku, penulis, penerbit, tahun terbit, ISBN, dan *barcode*. Antarmuka ini memudahkan pengelola perpustakaan untuk menambahkan, memperbarui, atau menghapus data buku dengan cepat dan akurat.

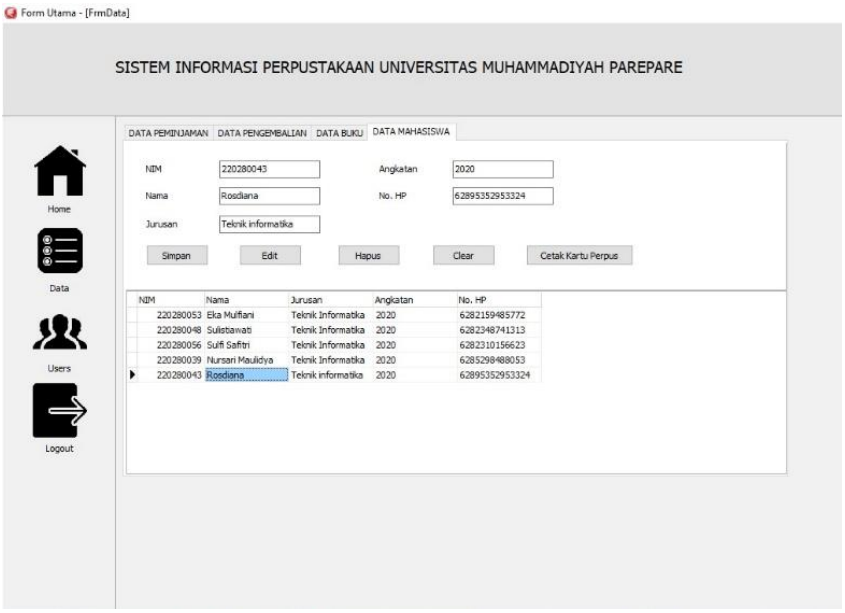


Gambar 4. 20 Halaman data buku

5. Halaman data mahasiswa

Halaman Data Mahasiswa pada sistem informasi perpustakaan Universitas Muhammadiyah Parepare digunakan untuk mengelola informasi mahasiswa sebagai pengguna layanan perpustakaan. Pada bagian atas halaman terdapat formulir isian yang mencakup kolom NIM, Nama, Jurusan, Angkatan, dan No. HP. Terdapat juga tombol fungsi seperti Simpan, Edit, Hapus, Clear, serta Cetak Kartu Perpus untuk mencetak kartu perpustakaan mahasiswa. Di bawah formulir, ditampilkan tabel berisi daftar mahasiswa yang telah terdaftar, lengkap dengan informasi seperti NIM, Nama, Jurusan, Angkatan, dan Nomor

HP. Halaman ini mempermudah petugas dalam mencatat dan memperbarui data mahasiswa yang aktif menggunakan layanan perpustakaan.



Gambar 4. 21 Halaman data mahasiswa

D. Pengujian Sistem

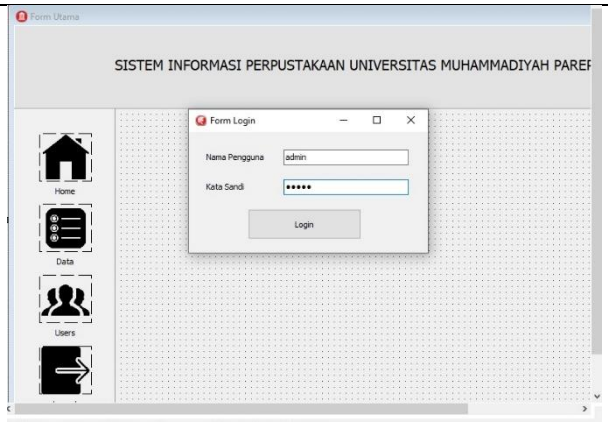
Pengujian sistem bertujuan untuk mengidentifikasi kesalahan, kesenjangan, atau kekurangan dalam sistem sebelum digunakan oleh pengguna akhir. Ada dua pendekatan utama dalam pengujian sistem ini, yaitu pengujian *Black box* dan pengujian *white box*.

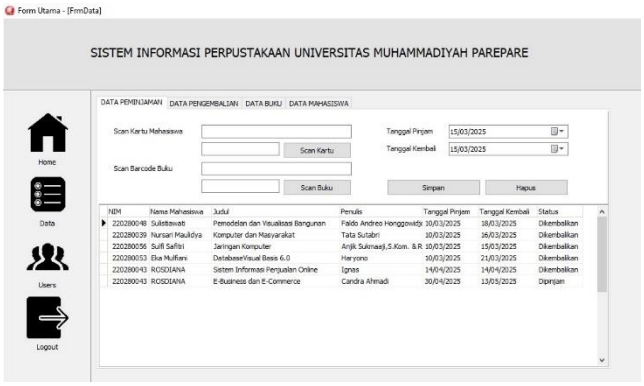
1. *Black box*

a. *Black box testing login* berhasil

Tabel 4. 6 *Black box testing login* berhasil

Tes faktor	Hasil	Keterangan
Memasukkan *email* atau *password* yang benar	✓	Sistem berhasil menampilkan halaman admin/*dashboard*.
Screenshot		

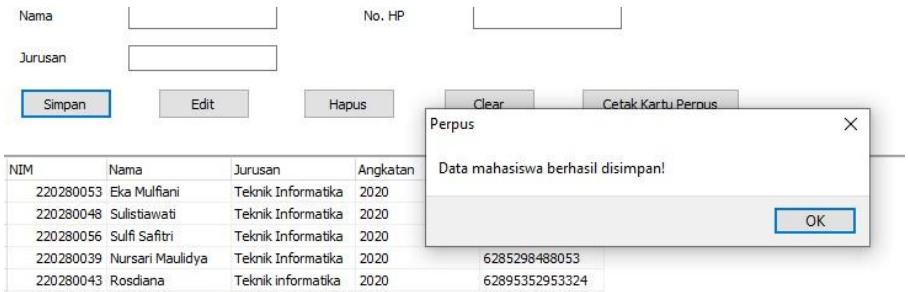




b. *Black box testing* tambah data

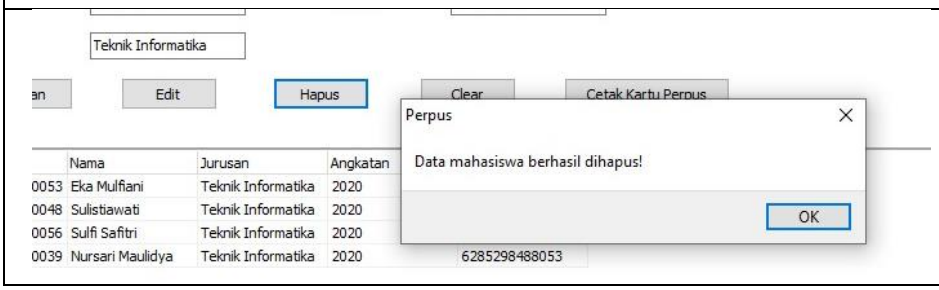
Tabel 4. 7 *Black box testing* tambah data

Tes faktor	Hasil	Keterangan
Admin mengisi *form* tambah dan menekan tombol simpan	✓	Sukses, sebuah notifikasi muncul bahwa data telah berhasil ditambahkan.
Screenshot		



c. *Black box testing* hapus data

Tabel 4. 8 *Black box testing* hapus data

Tes faktor	Hasil	Keterangan
Admin menghapus salah satu data	✓	Sukses, sebuah notifikasi muncul bahwa data telah berhasil dihapus.
Screenshot		
		

d. *Black box testing* halaman beranda

Tabel 4. 9 *Black box testing* halaman beranda

Tes faktor	Hasil	Keterangan
User pertama kali mengakses aplikasi	✓	Sukses, tampil halaman beranda
Screenshot		
		

e. *Black box testing scan barcode* kartu mahasiswa

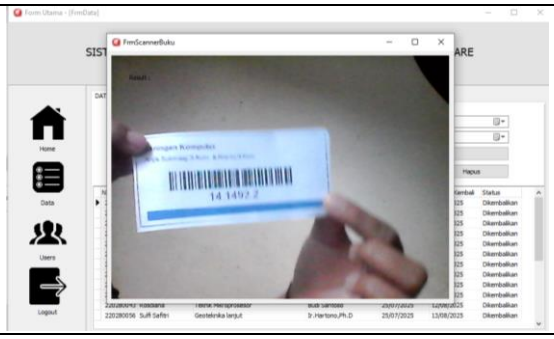
Tabel 4. 10 *Black box testing scan barcode* kartu mahasiswa

Tes faktor	Hasil	Keterangan
User menekan tombol *scan* kartu mahasiswa	✓	Sukses, tampil data mahaiswa
Screenshot		



f. *Black box testing scan barcode* kartu buku

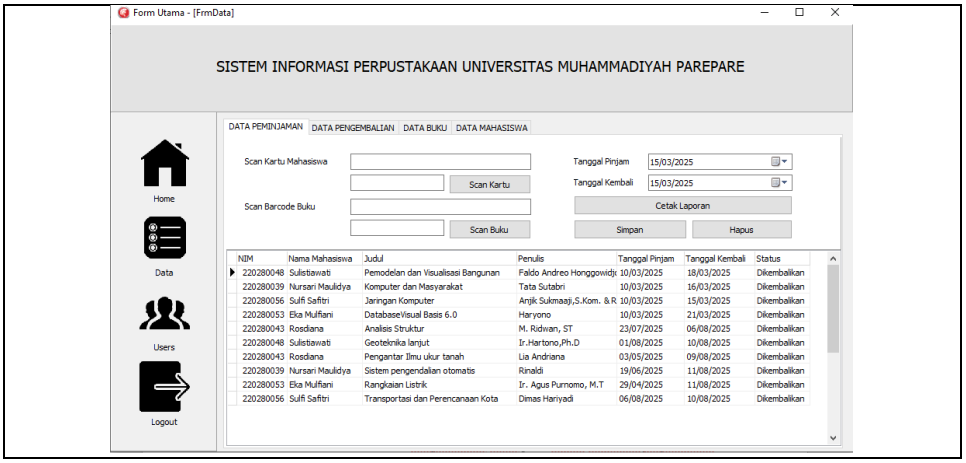
Tabel 4. 11 *Black box testing scan barcode* kartu buku

Tes faktor	Hasil	Keterangan
User menekan tombol scan buku	✓	Sukses, tampil data buku
Screenshot		
		

g. *Black box testing* peminjaman buku

Tabel 4. 12 *Black box testing* peminjaman buku

Tes faktor	Hasil	Keterangan
User selesai melakukan *scan* kartu mahasiswa dan kartu buku serta tanggal dan mengklik tombol simpan	✓	Sukses, data mahasiswa beserta buku yang dipinjam masuk ke list daftar peminjam.
Screenshot		



h. *Black box testing* pengembalian buku

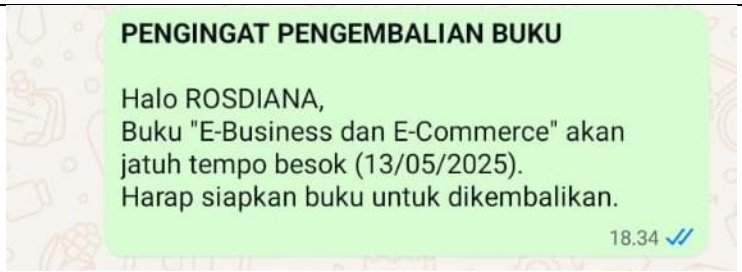
Tabel 4. 13 *Black box testing* peminjaman barang karang taruna

Tes faktor	Hasil	Keterangan
User menekan data peminjam yang akan mengembalikan buku dan mengklik tombol dikembalikan	✓	Sukses, hasil scan wajah hari ini untuk melakukan peminjaman
Screenshot		
		

i. *Black box testing* notifikasi pengingat pengembalian buku

Tabel 4. 14 *Black box testing* halaman dashboard admin

Tes faktor	Hasil	Keterangan
User meminjam buku dan belum mengembalikannya, sehingga menerima pengingat pengembalian.	✓	Notifikasi berhasil dikirim kepada peminjam satu hari sebelum tanggal jatuh tempo pengembalian buku.
Screenshot		



j. *Black box testing* notifikasi pengingat jatuh tempo

Tabel 4. 15 *Black box testing* halaman jatuh tempo

Tes faktor	Hasil	Keterangan
User belum mengembalikan buku pada hari jatuh tempo, sehingga sistem mengingatkan untuk segera mengembalikannya hari ini agar terhindar dari keterlambatan dan denda.	✓	Notifikasi jatuh tempo berhasil dikirim pada hari terakhir peminjaman untuk mengingatkan peminjam agar segera mengembalikan buku dan menghindari denda.
Screenshot		
		

k. *Black box testing* notifikasi keterlambatan pengembalian buku

Tabel 4. 16 *Black box testing* halaman pengaturan user

Tes faktor	Hasil	Keterangan
User tidak mengembalikan buku tepat waktu, sehingga sistem mendeteksi keterlambatan dan mengirim notifikasi disertai informasi denda.	✓	Notifikasi keterlambatan berhasil dikirim kepada peminjam setelah melewati tanggal jatuh tempo, disertai informasi jumlah hari

		keterlambatan dan denda.
Screenshot		
		

l. *Black box testing* Laporan Peminjaman Buku

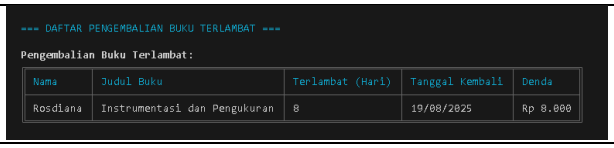
Tabel 4. 17 *Black box testing* Laporan Peminjaman Buku

Tes faktor	Hasil	Keterangan
User mencetak Laporan	✓	Rekap Data laporan dapat dicetak
Screenshot		
		

m. *Black box testing* Laporan Pengembalian Buku Terlambat

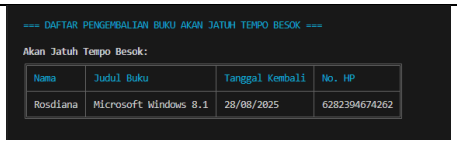
Tabel 4. 18 *Black box testing* laporan pengembalian buku terlambat

Tes faktor	Hasil	Keterangan
User mencetak Laporan	✓	Data laporan dapat dicetak
Screenshot		



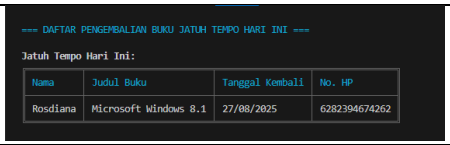
n. *Black box testing* Laporan Buku Jatuh Tempo Besok

Tabel 4. 19 *Black box testing* Laporan jatuh tempo besok

Tes faktor	Hasil	Keterangan
User mencetak Laporan	✓	Data laporan dapat dicetak
Screenshot		
		

o. Black box testing laporan Buku Jatuh Tempo Hari Ini

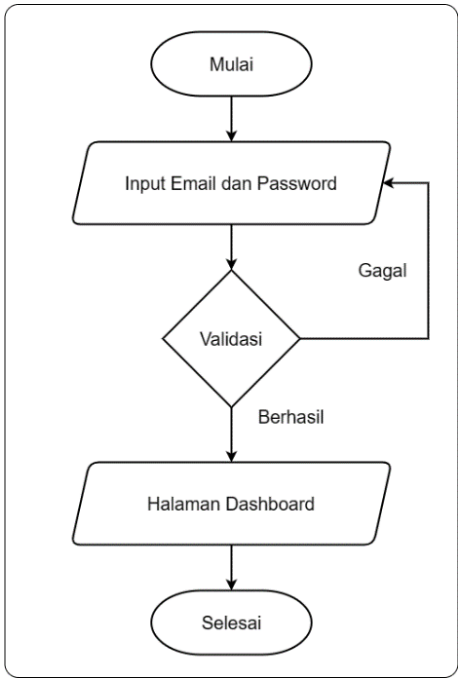
Tabel 4. 20 *Black box testing* laporan jatuh tempo hari ini

Tes faktor	Hasil	Keterangan
User mencetak Laporan	✓	Data laporan dapat dicetak
Screenshot		
		

2. *White box*

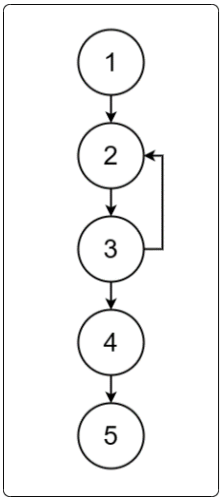
a. *White box testing login* berhasil

1) ***Flowchart***



Gambar 4. 22 *Flowchart login* berhasil

2) ***Flowgraph***



Gambar 4. 23 *Flowgraph login* berhasil

Berdasarkan gambar 4. 23 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada *edge* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

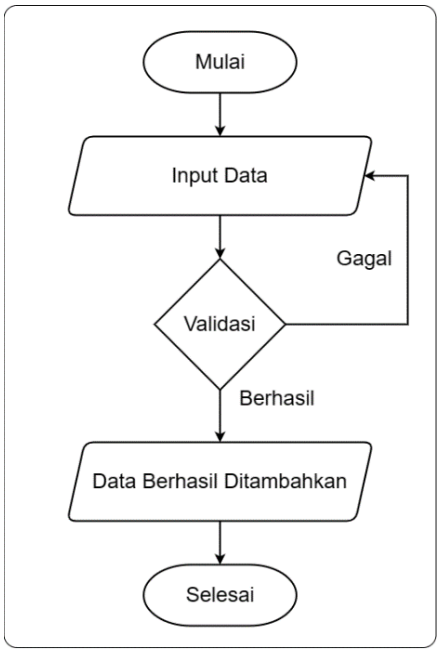
(4) Grafik matriks *login* berhasil

Tabel 4. 21 Grafik matriks *login* berhasil

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					1+ 1 = 2

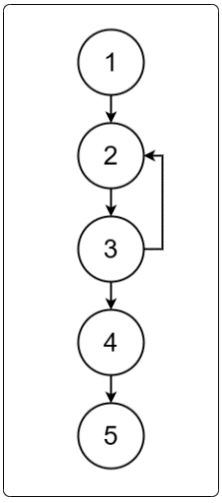
b. *White box testing* tambah data

1) *Flowchart*



Gambar 4. 24 *Flowchart* tambah data

2) *Flowgraph*



Gambar 4. 25 *Flowgraph* tambah data

Berdasarkan gambar 4. 25 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada *edge* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

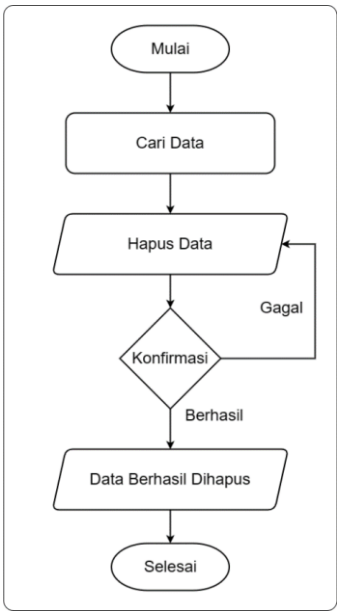
(4) Grafik matriks tambah data

Tabel 4. 22 Grafik matriks tambah data

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					1+ 1 = 2

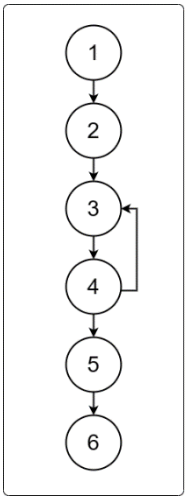
c. *White box testing* hapus data

1) *Flowchart*



Gambar 4. 26 *Flowchart* hapus data

2) *Flowgraph*



Gambar 4. 27 *Flowgraph* hapus data

Berdasarkan gambar 4. 27 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada *edge* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 6

N (*node*) = 6

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 6 – 6 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 3 – 5 – 6

Path 2 = 1 – 2 – 3 – 4 – 5 – 6

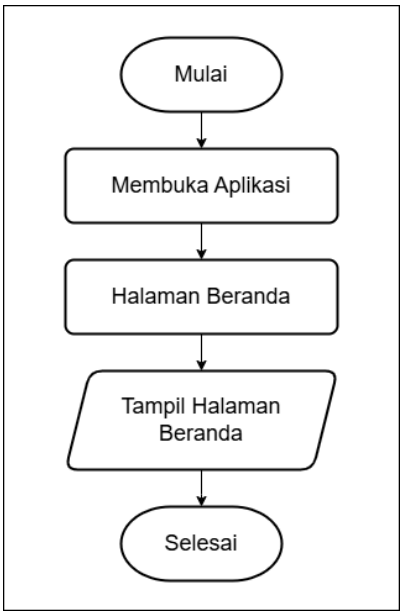
(4) Grafik matriks hapus data

Tabel 4. 23 Grafik matriks hapus data

	1	2	3	4	5	6	E-1
1		1					1 - 1 = 0
2			1				1 – 1 = 0
3				1			1 – 1 = 0
4			1		1		2 – 1 = 1
5						1	1 – 1 = 0
6							0
	SUM (E + 1)						1+ 1 = 2

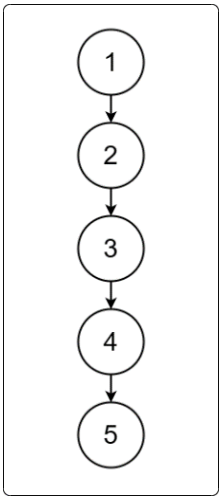
d. *White box testing* halaman beranda/*home*

1) *Flowchart*



Gambar 4. 28 *Flowchart* halaman beranda/*home*

2) *Flowgraph*



Gambar 4. 29 *Flowgraph* halaman beranda/*home*

Berdasarkan gambar 4. 29 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada *edge* dan *node.*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 4

N (*node*) = 5

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5

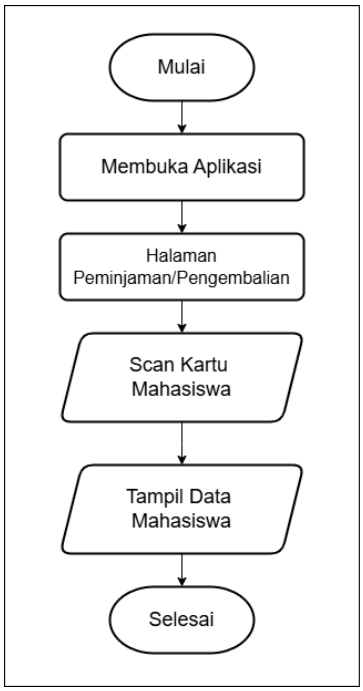
(4) Grafik matriks halaman beranda/*home*

Tabel 4. 24 Grafik matriks halaman beranda/*home*

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3				1		1 – 1 = 0
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					0+ 1 = 1

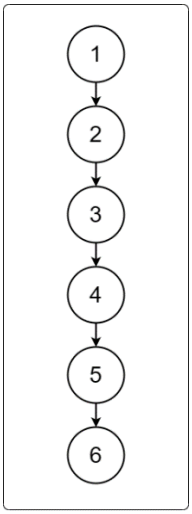
e. *White box testing scan barcode* kartu mahasiswa

1) *Flowchart*



Gambar 4. 30 *Flowchart* scan *barcode* kartu mahasiswa

2) *Flowgraph*



Gambar 4. 31 *Flowgraph* scan *barcode* kartu mahasiswa

Berdasarkan gambar 4. 31 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada *edge* dan *node.*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 6

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 5 – 6 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6

(4) Grafik matriks scan *barcode* kartu mahasiswa

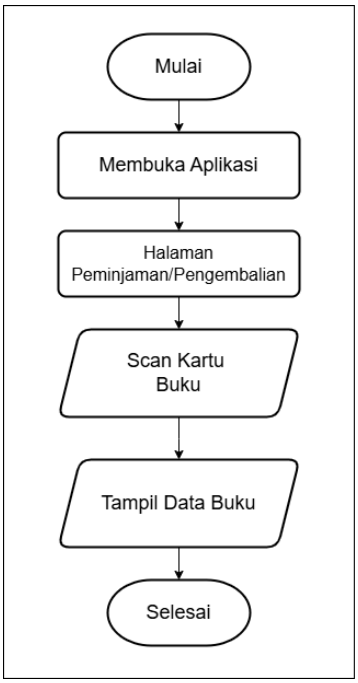
Tabel 4. 25 Grafik matriks scan *barcode* kartu mahasiswa

	1	2	3	4	5	6	E-1
1		1					1 - 1 = 0
2			1				1 – 1 = 0
3				1			1 – 1 = 0
4					1		1 – 1 = 0
5						1	1 – 1 = 0
6							0

	SUM (E + 1)		0+ 1 = 1

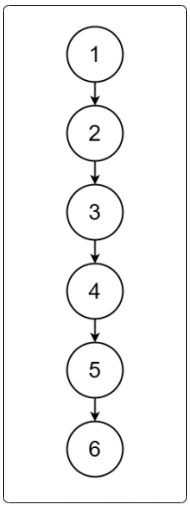
f. *White box testing scan barcode* kartu buku

1) *Flowchart*



Gambar 4. 32 *Flowchart* scan *barcode* buku

2) *Flowgraph*



Gambar 4. 33 *Flowgraph* scan *barcode* buku

Berdasarkan gambar 4. 33 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada *edge* dan *node.*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 6

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 5 – 6 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6

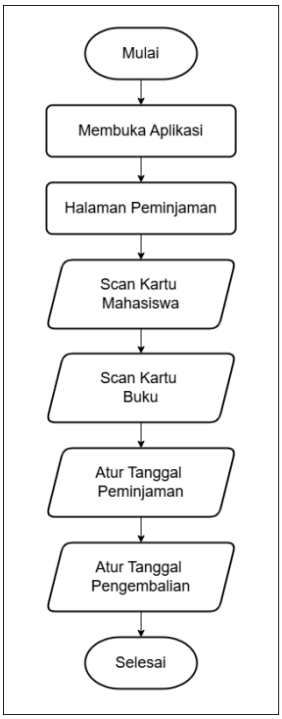
(4) Grafik matriks scan *barcode* buku

Tabel 4. 26 Grafik matriks scan *barcode* buku

	1	2	3	4	5	6	E-1
1		1					1 - 1 = 0
2			1				1 – 1 = 0
3				1			1 – 1 = 0
4					1		1 – 1 = 0
5						1	1 – 1 = 0
6							0
	SUM (E + 1)						0+ 1 = 1

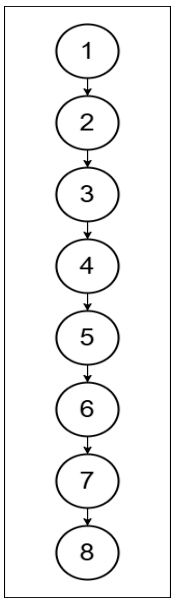
g. *White box testing* peminjaman buku

1) *Flowchart*



Gambar 4. 34 *Flowchart* peminjaman buku

2) *Flowgraph*



Gambar 4. 35 *Flowgraph* peminjaman buku

Berdasarkan gambar 4. 35 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada *edge* dan *node.*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 8

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 7 – 8 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

(4) Grafik matriks peminjaman buku

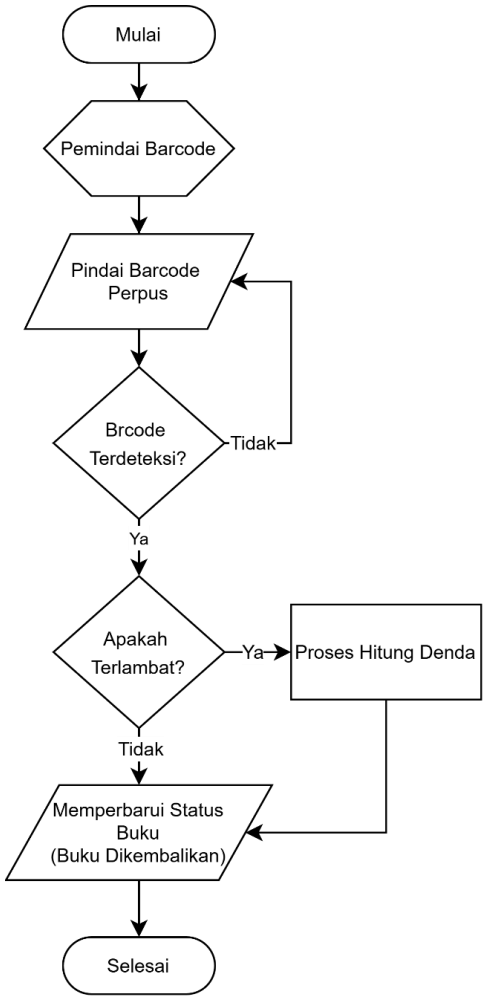
Tabel 4. 27 Grafik matriks peminjaman buku

	1	2	3	4	5	6	7	8	E-1
1		1							1 - 1 = 0
2			1						1 – 1 = 0
3				1					1 – 1 = 0
4					1				1 – 1 = 0
5						1			1 – 1 = 0
6							1		1 – 1 = 0

7								1	1 – 1 = 0
8									0
	SUM (E + 1)								0 + 1 = 1

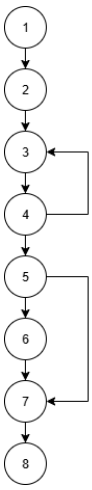
h. *White box testing* pengembalian buku

1) *Flowchart*



Gambar 4. 36 *Flowchart* pengembalian buku

2) *Flowgraph*



Gambar 4. 37 *Flowgraph* pengembalian buku

Berdasarkan gambar 4. 37 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada *edge* dan *node.*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 9

N (*node*) = 8

P (Predikat *node*) = 2

Penyelesaian :

V(G) = E – N + 2

= 9 – 8 + 2

= 3

Predikat (P) = P + 1

= 2 + 1

= 3

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region* = 3

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

Path 2 = 1 – 2 – 3 – 4 – 3

Path 3 = 1 – 2 – 3 – 4 – 5– 7

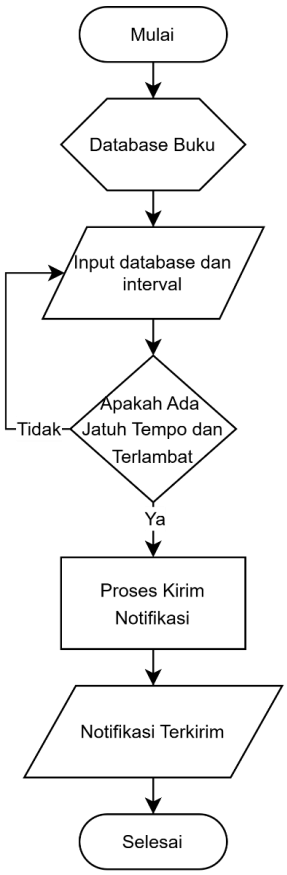
(4) Grafik matriks pengembalian buku

Tabel 4. 28 Grafik matriks pengembalian buku

	1	2	3	4	5	6	7	8	E - 1
1		1							1 – 1 = 0
2			1						1 – 1 = 0
3				1					1 – 1 = 0
4			1		1				2 – 1 = 1
5						1	1		2 – 1 = 1
6							1		1 – 1 = 0
7								1	1 – 1 = 0
8									0
	SUM (E + 1)								0 + 1 = 3

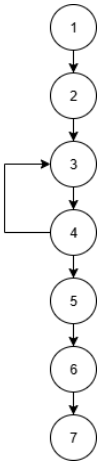
i. *White box testing* notifikasi pengingat pengembalian buku

1) ***Flowchart***



Gambar 4. 38 *Flowchart* notifikasi pengingat pengembalian buku

2) ***Flowgraph***



Gambar 4. 39 *Flowgraph* notifikasi pengingat pengembalian buku

Berdasarkan gambar 4. 39 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada *edge* dan *node.*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 7

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 7 – 7 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7

Path 2 = 1 – 2 – 3 – 4 – 3

(4) Grafik matriks notifikasi pengingat pengembalian buku

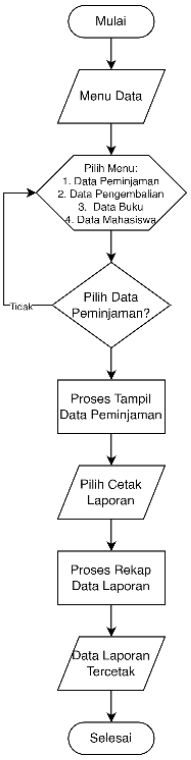
Tabel 4. 29 Grafik matriks notifikasi pengingat pengembalian buku

	1	2	3	4	5	6	7	E-1
1		1						1 - 1 = 0
2			1					1 – 1 = 0
3				1				1 – 1 = 1
4			1		1			2 – 1 = 1

5						1		1 – 1 = 0
6							1	1 – 1 = 0
7								0
	SUM (E + 1)							1+ 1 = 2

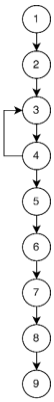
j. *White box testing* Cetak Laporan Peminjaman

(1) *Flowchart*



Gambar 4. 40 *Flowchart* Cetak Laporan Peminjaman Buku

(2) *Flowgraph*



Gambar 4. 41 *Flowgraph* Cetak Laporan Peminjaman Buku

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada *edge* dan *node.*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 9

N (*node*) = 9

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 9 – 9 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region* = 2

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 3

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

(4) Grafik matriks notifikasi pengingat pengembalian buku

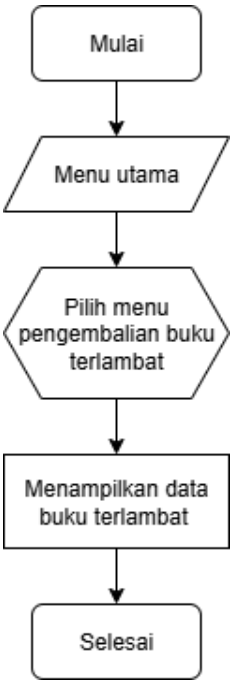
Tabel 4. 30 Grafik matriks Cetak Laporan Peminjaman Buku

	1	2	3	4	5	6	7	8	9	E-1
1		1								1 - 1 = 0
2			1							1 – 1 = 0
3				1						1 – 1 = 0
4			1		1					2 – 1 = 1
5						1				1 – 1 = 0
6							1			1 – 1 = 0

7								1		1 – 1 = 0
8									1	1 – 1 = 0
SUM (E + 1)										1+ 1 = 2

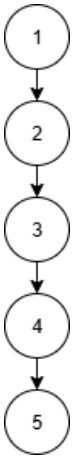
k. *White box testing* Laporan Pengembalian Buku Terlambat

1) Flowchart



Gambar 4. 42 *Flowchar*t laporan pengembalian buku terlambat

2) Flowgraph



Gambar 4. 43 *Flowgraph* laporan pengembalian buku terlambat

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada edge dan node.

Pada rumus : V(G) = E – N + 2

E (*edge*) = 4

N (*node*) = 5

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region* =1

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5

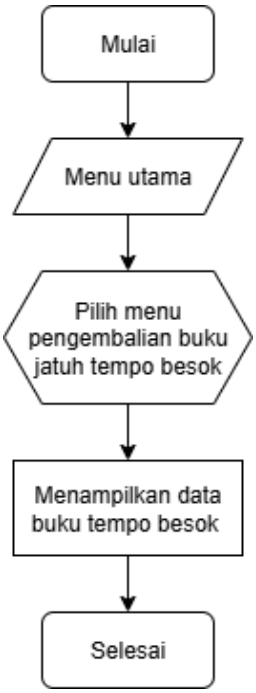
(4) Grafik matriks laporan pengembalian buku terlambat

Tabel 4. 31 Grafik matriks laporan pengembalian buku terlambat

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3				1		1 – 1 = 0
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					0+ 1 = 1

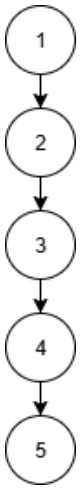
l. *White box testing* Laporan Pengembalian Jatuh Tempo Besok

1) Flowchart



Gambar 4. 44 *Flowchart* laporan jatuh tempo besok

2) Flowgraph



Gambar 4. 45 *Flowgraph* laporan jatuh tempo besok

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada edge dan node.

Pada rumus : V(G) = E – N + 2

E (*edge*) = 4

N (*node*) = 5

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region* = 1

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5

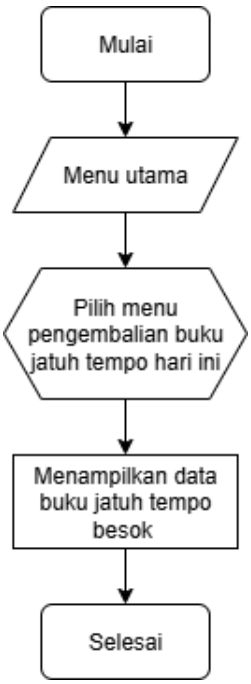
(4) Grafik matriks laporan jatuh tempo besok

Tabel 4. 32 Grafik matriks laporan jatuh tempo besok

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3				1		1 – 1 = 0
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					0+ 1 = 1

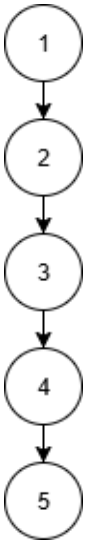
m. *White box testing* Laporan Jatuh Tempo Hari Ini

1) Flowchart



Gambar 4. 46 *Flowchart* laporan jatuh tempo hari ini

2) Flowgraph



Gambar 4. 47 *Flowgraph* laporan jatuh tempo hari ini

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* V(G) pada edge dan node.

Pada rumus : V(G) = E – N + 2

E (*edge*) = 4

N (*node*) = 5

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *Flowgraph* diatas memiliki *region* = 1

(3) *Independent path* pada *Flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks laporan jatuh tempo hari ini

Tabel 4. 33 Grafik matriks laporan jatuh tempo hari ini

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3				1		1 – 1 = 0
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					0 + 1 = 1

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa:

1. Sistem informasi perpustakaan berbasis barcode berhasil dirancang dan dikembangkan menggunakan *elphi studio XE8* dengan dukungan database SQLite. Sistem ini mampu mempercepat proses peminjaman dan pengembalian buku, meminimalisir kesalahan input manual, serta meningkatkan efisiensi dalam pengelolaan data perpustakaan.
2. Sistem informasi perpustakaan yang dikembangkan telah terintegrasi dengan *whatsApp gateway* untuk memberikan notifikasi otomatis terkait batas waktu pengembalian buku. Fitur ini terbukti efektif dalam mengingatkan mahasiswa sebelum jatuh tempo, pada hari terakhir, maupun ketika terjadi keterlambatan, sehingga keterlambatan pengembalian dapat berkurang dan kepatuhan mahasiswa meningkat.

B. Saran

1. Sistem informasi perpustakaan yang telah dirancang masih memiliki potensi untuk dikembangkan lebih lanjut, seperti penambahan fitur pencarian buku yang lebih spesifik dan pelaporan statistik peminjaman.
2. Sistem informasi perpustakaan berbasis *barcode* ini dapat diterapkan tidak hanya di lingkungan Universitas Muhammadiyah Parepare, tetapi juga di

sekolah, lembaga, atau perpustakaan lainnya yang masih menggunakan sistem manual, agar pengelolaan perpustakaan menjadi lebih efisien.

DAFTAR PUSTAKA

Amarta, R. P. (2021). Pengembangan Sistem Informasi Perpustakaan Berbasis *QR code* (Quick Response Code)(Studi Kasus Perpustakaan Fakultas Teknik Universitas Majalengka). PROSIDING SNAST, 198-209.

Ade Hastuty Hasyim. (2021). Dasar Pemograman. Cet 2-Makassar: C.V. Bangun Bumi Utama.

Duckett, J. (2022). CSS in Action. Manning Publications.

Duckett, J., & Wiley, J. (2019). HTML & CSS Design and Build Websites. John Wiley & Sons

Hartl, M., & Donahoe, L. (2020). Learn Enough HTML to Be Dangerous. Learn Enough Society

Hidayanti, N., Nuryani, E., Kania, R., & Wijaya, F. Y. (2023). Rancan Bangun Sistem Informasi Manajemen Perpustakaan Menggunakan *QR code* Berbasis Website. Jurnal Sistem Informasi dan Informatika (Simika), 6(1), 35-47.

Lamada, M., Ifani, A. Z., & Wahyuni, F. (2022). Analisis Kualitas dan Pengembangan Sistem Informasi Perpustakaan berbasis web menggunakan Teknologi *Barcode*. Journal of Embedded Systems, Security and Intelligent Systems, 3(1), 15-28.

Maidda, N. R. A. (2019). Sistem Informasi Perpustakaan Menggunakan *Barcode* Scanner di MAN Kota Magelang (Doctoral dissertation, Skripsi, Universitas Muhammadiyah Magelang).

Malau, E., Salim, A., Santoso, B., & Ramadan, R. (2020). Sistem Informasi Manajemen Perpustakaan Dengan *Barcode* Di GKI Gunung Sahari Jakarta. Bina Insani Ict Journal, 5(1), 41-50.

Meyer, E. A., & Weyl, E. (2020). CSS: The Definitive Guide (4th ed.). O'Reilly & Associates.

Mz, MM, Hastuty, A., & Yunus, M. INFORMASI SEKOLAH UPT SDN BERBASIS WEB.

Oktaviani, R. D., & Seprina, I. (2020, August). Rancang bangun sistem informasi perpustakaan sma pusri palembang menggunakan *barcode*. In Bina Darma Conference on Computer Science (BDCCS) (Vol. 2, No. 1, pp. 89-96).

Putri, H., Rini, F., & Pratama, A. (2022). Sistem Informasi Perpustakaan Berbasis Web. Jurnal Pustaka Data (Pusat Akses Kajian Database, Analisa Teknologi, Dan Arsitektur Komputer), 2(1), 5-10.

Renatha, F. A., Satoto, K. I., & Nurhayati, O. D. (2019). Perancangan dan pengembangan sistem informasi perpustakaan berbasis web (studi kasus Jurusan Sistem Komputer). Jurnal Teknologi dan Sistem Komputer, 3(3), 343-353.

Tahir, F., Hastuty, A., & Yunus, M. (2025). Sistem Informasi Pengelolaan Data pada Perpustakaan Umum Parepare. *KOMPUTEK*, *9*(1), 32-38.