BAB III
METODE PENELITIAN

3.1 Desain Penelitian

Metode yang dipakai untuk melakukan penelitian ini adalah metode *design and development*. Menurut Ellis dan Levy (2010) metode ini dimulai dari proses konseptualisasi sebuah masalah dan berujung dengan evaluasi upaya yang dilakukan dalam menyelesaikan masalah tersebut, Metode penelitian *design and development* bertujuan untuk membangun atau mengembangkan model, alat, dan prosedur baru yang dapat diandalkan dalam menjawab permasalahan dalam bidang yang diteliti. Proses *design and development* tidak hanya sekedar menciptakan sebuah produk, tetapi juga melibatkan tahapan yang sistematis. Oleh karena itu, pendekatan ini selaras dengan penelitian yang akan dilakukan untuk membangun sistem yang diawali dengan identifikasi masalah, dilanjutkan dengan desain sistem, hingga tahapan evaluasi. Untuk lebih memperjelas alur penelitian yang dilakukan, pada Gambar 3.1 adalah alur yang digunakan untuk menggambarkan tahapan-tahapan yang dilalui dalam penelitian ini.

Gambar 3.1 Alur Proses *Design and Development*

Pada Gambar 3.1 dalam konteks penelitian ini, metode tersebut diterapkan melalui tahapan yang dimulai dari identifikasi masalah yang akan dijadikan latar belakang yang akan dikaji pada penelitian ini, diikuti oleh deskripsi tujuan sebagai solusi yang dirancang secara spesifik untuk menjawab permasalahan tersebut. Kemudian, melakukan perancangan dan pengembangan sistem yang sesuai dengan tujuan baik itu dari sisi aplikasi web atau perangkat keras, setelah itu sistem yang telah dikembangkan akan diuji secara keseluruhan apakah berjalan dengan baik, jika pengujian telah usai dilanjutkan pada tahap evaluasi untuk menilai efektivitas dan efisiensi sistem yang dikembangkan, sehingga dapat memastikan bahwa solusi

yang dihasilkan mampu memberikan dampak positif terhadap masalah yang dihadapi. Tahap terakhir, melaporkan atau mendokumentasikan seluruh hasil pengujian dalam bentuk tulisan penelitian ini.

3.2 Identifikasi Masalah

Sejalan dengan perkembangan digitalisasi, teknologi berpengaruh besar untuk semua sektor salah satunya yaitu sistem sirkulasi di perpustakaan. Layanan sirkulasi di perpustakaan yang telah bertransformasi ke dalam bentuk digital untuk memberikan kemudahan bagi pengguna dalam mengakses informasi. Namun, sistem digital ini juga melibatkan pemrosesan data pribadi. Berkaitan dengan data pribadi, tentunya terdapat tantangan yang sangat besar berkaitan dengan kerahasiaan dan integritas data. Berdasarkan laporan publik Badan Siber dan Sandi Negara (BSSN) pada bulan Maret 2025, negara Indonesia menempati posisi dua dari top lima negara dengan tujuan anomali *traffic* kejahatan siber terbanyak sebesar 73.882.259 anomali. Tragedi ini tentunya harus menjadi perhatian serius bagi semua pihak, mengingat jumlah ancaman siber yang terus meningkat dapat berdampak signifikan pada keamanan data dan sistem digital di berbagai sektor yang ada di Indonesia. Berdasarkan masalah yang diangkat maka dibutuhkan sebuah proteksi untuk menjaga keamanan data. Tentunya sistem harus memiliki kebijakan privasi yang sesuai dengan peraturan perlindungan data yang berlaku untuk memastikan data terlindungi dari penyalahgunaan orang asing.

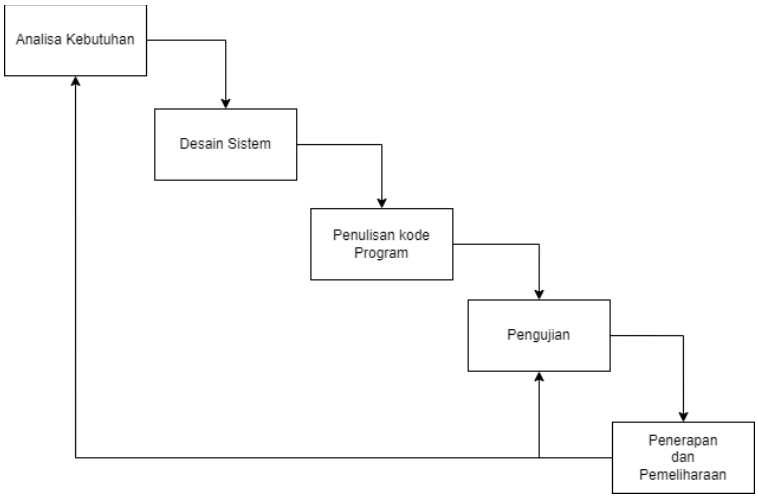
3.3 Deskripsi Tujuan

Dari hasil temuan masalah, fokus penelitian ini bertujuan untuk merancang serta membangun sistem pengamanan sirkulasi buku perpustakaan berbasis kriptografi AES-128 dengan RFID terintegrasi IoT. Sistem ini diharapkan mampu berjalan optimal dalam mendukung aktivitas sirkulasi buku. Selain itu, penelitian ini juga bertujuan untuk memastikan bahwa penerapan algoritma kriptografi AES-128 pada sisi aplikasi web mampu memberikan perlindungan data pribadi terhadap serangan siber.

3.4 Desain dan Pengembangan Sistem

Tahapan ini membahas perancangan desain dan pengembangan sistem yang akan dilakukan dalam penelitian ini. Untuk memastikan proses pengembangan

berjalan dengan terarah dan terstruktur, model Waterfall dipilih karena menurut Wahid (2020) model pengembangan ini menerapkan pendekatan yang sistematis dan terstruktur secara berurutan. Disebut sebagai Waterfall karena setiap tahap harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya, mengikuti alur yang linear. Prosesnya dimulai dari tahap awal, yaitu perencanaan, hingga tahap akhir, yaitu pemeliharaan. Setiap tahapan tidak dapat dilompati, diulang, atau kembali ke tahap sebelumnya setelah selesai dilaksanakan. Model Waterfall yang terbagi menjadi 5 tahapan yaitu analisa kebutuhan, desain sistem, penulisan kode program, pengujian, dan penerapan serta pemeliharaan. Untuk lebih jelasnya alur Waterfall ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Alur Kerja Model Waterfall

3.4.1 Analisa Kebutuhan

Pada tahapan ini penulis melakukan analisa serta menetapkan kebutuhan teknologi alat dan bahan yang akan digunakan dalam proses pengembangan sistem pengamanan sirkulasi buku perpustakaan berbasis kriptografi AES-128 dengan RFID terintegrasi IoT. Berikut adalah kebutuhan teknologi perangkat lunak dan perangkat keras yang sesuai dengan sistem yang akan dibangun, tersaji pada Tabel 3.1 dan Tabel 3.2

Tabel 3.1
Rancangan Kebutuhan Teknologi Perangkat Lunak

Nama Teknologi	Fungsi
Visual Studio Code dan PlatformIo	Visual Studio Code berfungsi sebagai *editor* kode JavaScript dan C++, PlatformIO *plugin* untuk mempermudah pengelolaan proyek dan manajemen *library* perangkat keras.
Behance	*Platform* referensi dalam perancangan tampilan antarmuka *dashboard* dan *landing page*.
React Js	*Framework* JavaScript untuk membangun antarmuka.
Tailwind CSS	*Framework* CSS untuk membuat desain antarmuka.
Node Js dan Express Js	Node Js digunakan sebagai *runtime* JavaScript untuk membangun aplikasi *server-side*, sedangkan Express Js *framework* berbasis Node Js untuk pengembangan RESTful API dan pengelolaan *routing*.
MySQL	Digunakan untuk manajemen basis data relasional.
DrawIo dan Fritzing	DrawIo berguna untuk membuat diagram alur sistem seperti flowchart atau arsitektur, sedangkan Fritzing digunakan untuk mendesain *wiring* perangkat keras.
HiveMQ Cloud	Broker MQTT berbasis *cloud* yang digunakan untuk mengelola topik *subscribe* dan *publish* data dalam protokol komunikasi IoT.
Github	Untuk pengelolaan repositori kode.
Vercel	*Platform cloud* untuk *hosting* aplikasi *frontend* dengan integrasi repositori github.
DigitalOcean	Layanan *Platform as a Service* (Paas) untuk *deployment backend* secara otomatis dari repositori github.
AWS RDS	Layanan basis data *cloud* untuk penyimpanan dan pengelolan *database* SQL.

Nama Teknologi	Fungsi
Docker	*Platform containerization* untuk mengemas aplikasi *backend* beserta dependensinya agar dapat *deploy* langsung ke DigitalOcean melalui Dockerfile.

Tabel 3.2
Rancangan Kebutuhan Teknologi Perangkat Keras

Nama Teknologi	Fungsi
Mikrokontroler ESP32 Devkit V1	Digunakan untuk menjalankan logika program serta mengelola komunikasi dengan aplikasi web.
ESP32 Expansion Board	*Board* tambahan untuk ESP32 terhubung dengan catu daya.
Modul RFID RC522	Modul pembaca kartu/stiker RFID. Digunakan untuk membaca ID unik dari kartu RFID *user* dan stiker buku.
LCD I2C 16X2	Digunakan untuk menampilkan pesan teks seperti status berhasil atau gagal meminjam buku.
Push Button	Sebagai input untuk memilih mode peminjaman buku atau pengembalian buku.
Catu Daya 12V	Digunakan untuk memberikan suplai energi ke ESP32

3.4.2 Desain Sistem

Sebelum tahapan penulisan program, pada tahapan kedua ini dilakukan perancangan desain sistem dalam membantu mendefinisikan arsitektur sistem secara keseluruhan perangkat keras dan perangkat lunak. Proses ini mencakup pembuatan desain sistem menggunakan diagram *Unified Modeling Language* (UML), seperti *activity diagram*, perancangan *database* menggunakan *Entity Relationship Diagram* (ERD), diagram arsitektur sistem, blok diagram perangkat, *wiring* diagram perangkat, dan *flow chart* sistem (Kurniawan dkk., 2020).

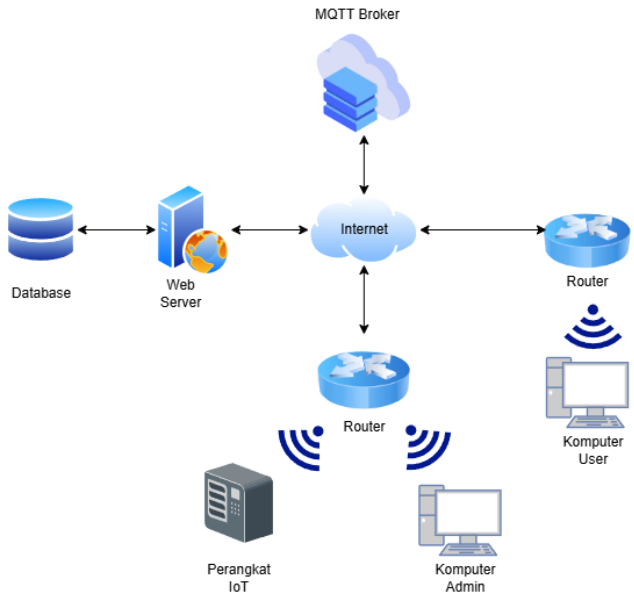
3.4.2.1 Perancangan Sistem Perangkat Keras

Pada tahap ini, penulis merancang sistem perangkat keras untuk memastikan semua komponen bekerja sesuai kebutuhan sistem. Langkah ini mencakup identifikasi perangkat keras utama yang akan digunakan dalam membangun sistem

pengamanan sirkulasi buku perpustakaan berbasis kriptografi AES-128 dengan RFID terintegrasi IoT. Berikut adalah desain dari sistem yang akan disajikan dalam bentuk UML.

1. Arsitektur Diagram

Arsitektur diagram berfungsi untuk memvisualisasikan keterhubungan antara perangkat keras dan perangkat lunak dalam sistem. Dengan adanya diagram ini, penulis dapat lebih mudah memahami bagaimana perangkat seperti *server*, *database*, jaringan, dan perangkat pengguna terintegrasi melalui koneksi internet. Selain itu, arsitektur ini juga membantu memastikan setiap elemen sistem dapat bekerja secara sinkron untuk mendukung fungsionalitas utama, seperti pengolahan data, komunikasi antar perangkat, dan penyampaian informasi ke pengguna akhir. Untuk lebih detailnya dapat dilihat pada Gambar 3.3.

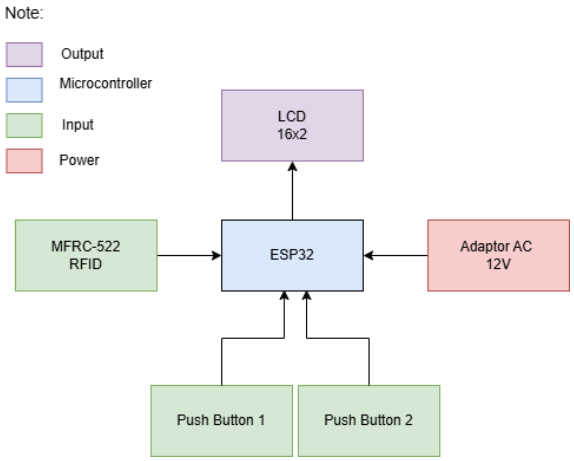


Gambar 3.3 Arsitektur Diagram Perangkat Keras

2. Blok Diagram

Dalam blok diagram ini, menggambarkan hubungan antar komponen utama, termasuk mikrokontroler, modul RFID, LCD, *push button*, dan adaptor daya. Diagram ini memberikan representasi visual untuk mempermudah pemahaman

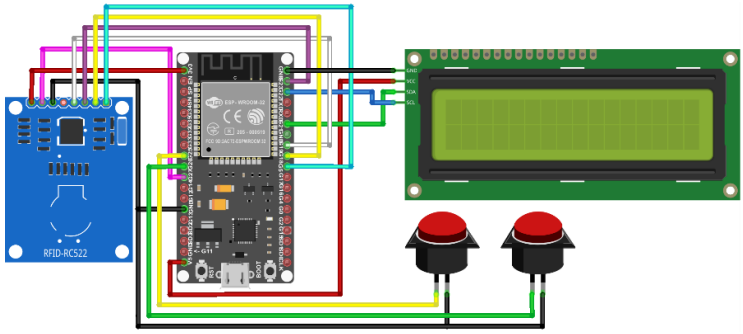
terhadap integrasi dan koneksi antar komponen yang membentuk sistem keseluruhan perangkat keras dapat dilihat pada Gambar 3.4.



Gambar 3.4 Blok Diagram Perangkat Keras

3. *Wiring* Diagram

Berikut ini adalah *wiring* diagram yang menggambarkan koneksi antar komponen perangkat keras, seperti ESP32, modul RFID RC522, LCD 16x2, dan *push button*. *Wiring* diagram ini di desain untuk memberikan gambaran teknis dalam proses perakitan perangkat keras agar sistem dapat berfungsi dengan baik sesuai rancangan untuk detailnya dapat dilihat pada Gambar 3.5.

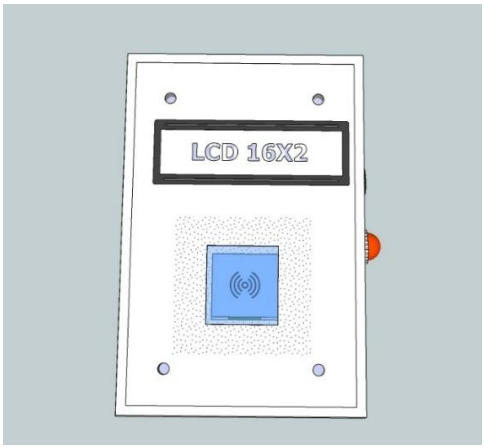


Gambar 3.5 *Wiring* Diagram Perangkat Keras

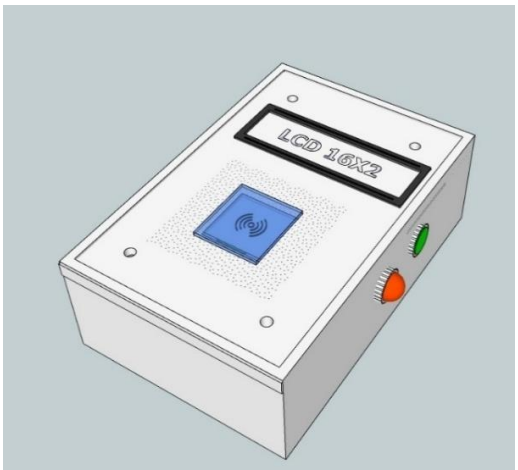
Gambar 3.5 menunjukkan hubungan kabel antar komponen untuk memastikan integrasi berjalan dengan lancar. Setiap koneksi telah dirancang untuk mendukung fungsi utama dari sistem, seperti membaca data RFID kartu atau stiker, menampilkan pesan pada LCD, serta menerima input melalui *push button* untuk memilih mode.

4. Desain Sistem Perangkat Keras

Berikut adalah desain prototipe sistem perangkat keras yang telah dibuat untuk mendukung mengembangkan sistem pengamanan sirkulasi buku perpustakaan berbasis kriptografi AES-128 dengan RFID terintegrasi IoT. Pada bagian atas prototipe ini, terdapat LCD 16x2, dan modul RFID RC522 dapat dilihat pada Gambar 3.6. Selain itu, di sisi samping prototipe ini dilengkapi dengan dua *push button* yang berfungsi sebagai input manual untuk memilih mode tertentu dengan detail pada Gambar 3.7.



Gambar 3.6 Desain Tampak Atas Perangkat Keras



Gambar 3.7 Desain Tampak Samping Perangkat Keras

3.4.2.2 Perancangan Sistem Perangkat Lunak

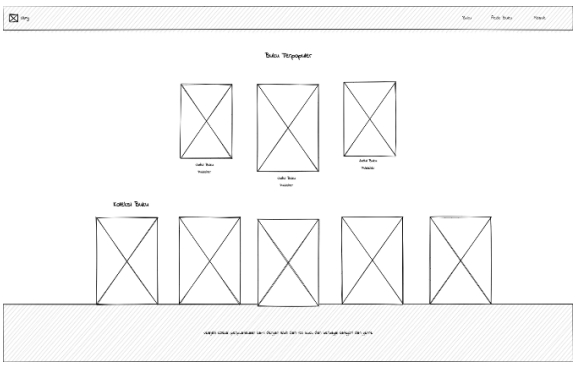
Pada tahap ini, penulis merancang sistem perangkat lunak aplikasi web dikembangkan dengan menggunakan Visual Studio Code sebagai editor utama untuk menulis kode JavaScript, baik di sisi *frontend* maupun *backend*. Pada bagian

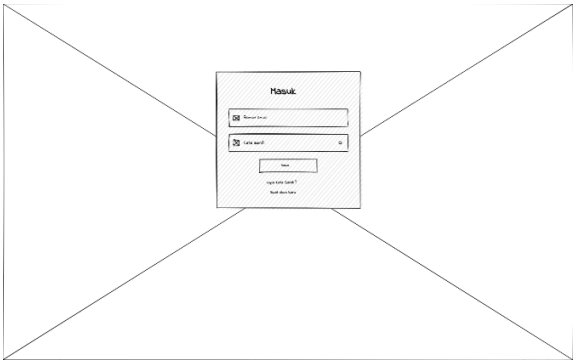
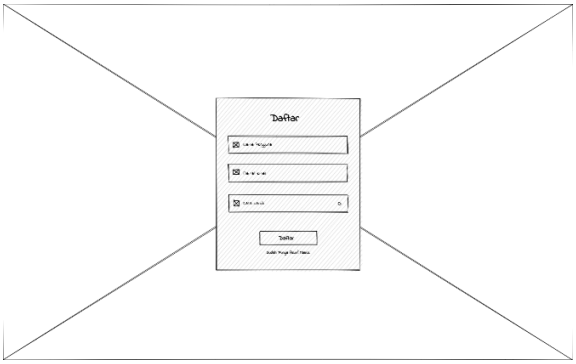
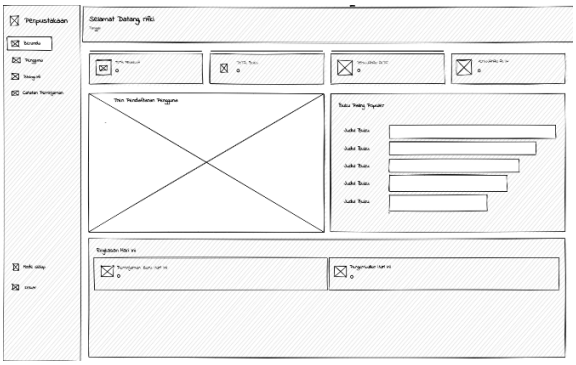
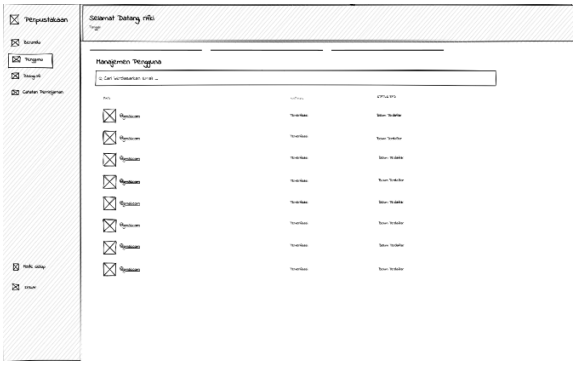
frontend, digunakan React.js untuk membangun antarmuka, serta Tailwind CSS untuk mendesain tampilan antarmuka yang responsif dan modern. Di sisi *backend*, pengembangan dilakukan menggunakan Node.js sebagai *runtime* Express.js sebagai *framework* untuk membangun RESTful API serta mengelola *routing*. Untuk komunikasi data aplikasi web dengan perangkat keras memakai protokol MQTT dengan broker HiveMQ *cloud* untuk mempublikasikan dan menerima data. Proses perancangan ini mendukung dalam membangun sistem pengamanan sirkulasi buku perpustakaan berbasis kriptografi AES-128 dengan RFID terintegrasi IoT. Rancangan sistem perangkat lunak ini disajikan dalam bentuk *wireframe* dan diagram UML untuk memberikan gambaran menyeluruh mengenai tata letak antarmuka, alur kerja dan interaksi dalam sistem.

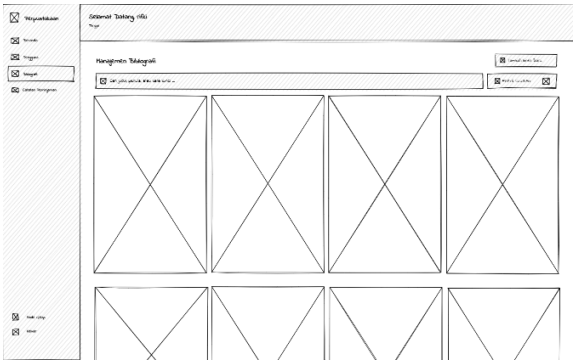
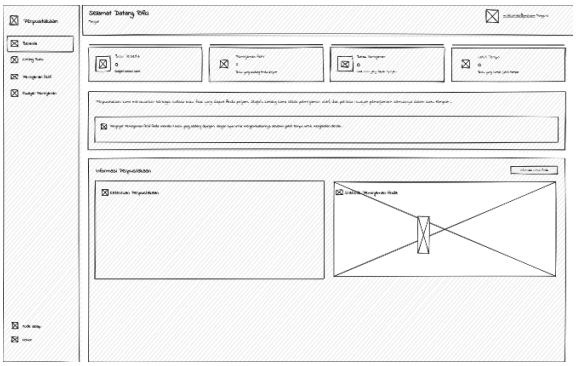
1. *Wireframe*

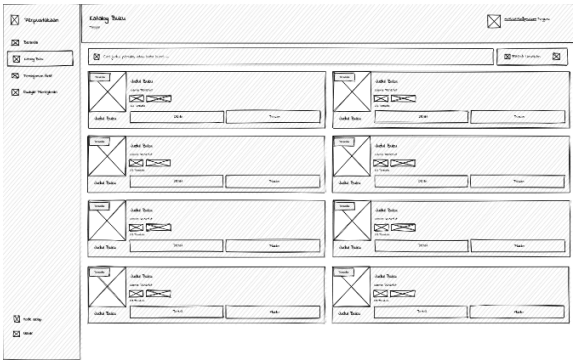
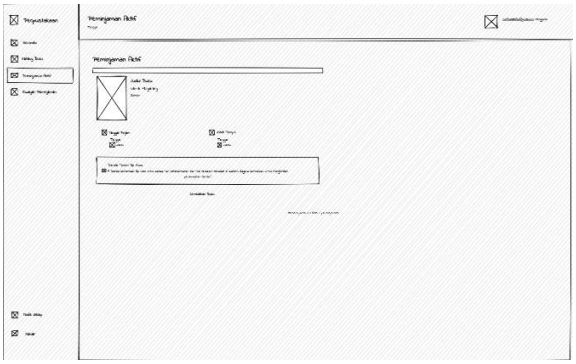
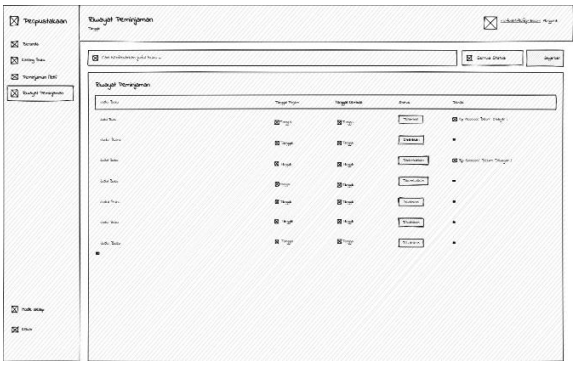
Wireframe merupakan kerangka desain suatu aplikasi untuk penataan elemen-elemen pada halaman aplikasi sebelum proses *coding* dimulai. *Wireframe* secara visual digambarkan berupa garis dan kotak yang mengatur tata letak elemen-elemen pada aplikasi. *Wireframe* terbagi menjadi dua jenis yaitu *wireframe low-fidelity* dan *high-fidelity*. Jenis *low-fidelity* merupakan desain dasar yang belum menampilkan warna dan elemen lainnya, sedangkan *high-fidelity* sudah menampilkan warna serta elemen lainnya (Fadilah & Sweetania, 2023). Detail *wireframe* yang telah dirancang dapat dilihat pada Tabel 3.3 berikut.

Tabel 3.3 Wireframe Sistem Perangkat Lunak

Wireframe	**Deskripsi**
	Struktur tata letak halaman *landing page* sebelum *login* atau register. Menampilkan koleksi buku, buku populer, dan *navigation bar*.

Wireframe	**Deskripsi**
	Susunan elemen pada halaman *login.* Menampilkan *Form* input alamat email dan kata sandi, serta tombol masuk untuk masuk ke halaman *dashboard*.
	Susunan elemen pada halaman *register.* Menampilkan *Form* input nama, alamat email dan kata sandi, serta tombol daftar untuk masuk ke halaman *login*.
	Struktur tata letak halaman beranda *dashboard* admin. Menampilkan data jumlah total buku, *user*, peminjaman, dan pengembalian. Selain itu ada visualisasi data dalam bentuk *chart*.
	Susunan elemen pada *dashboard* admin menu manajemen pengguna. Menampilkan seluruh *user*, status, status RFID, dan *search box* untuk mencari *user*.

Wireframe	**Deskripsi**
	Susunan elemen pada *dashboard* admin menu manajemen bibliografi. Menampilkan seluruh buku, tombol untuk menambahkan buku, tombol filter buku, dan s*earch box* untuk mencari buku.
	Struktur tata letak halaman catatan peminjaman pada *dashboard* admin. Menampilkan informasi *user* yang melakukan peminjaman, tombol *filter*, dan tombol *export to* csv.
	Struktur tata letak halaman beranda pada *dashboard user*. Menampilkan profil pribadi serta jumlah total data buku, peminjaman, pengembalian, informasi pengguna, dan visualisasi dalam bentuk *chart*.

Wireframe	Deskripsi
	Struktur tata letak halaman katalog buku pada *dashboard user*. Menampilkan koleksi buku, s*earch box* untuk mencari buku, *filter* buku, tombol pesan buku, dan tombol detail buku.
	Struktur tata letak halaman katalog buku pada *dashboard user*. Menampilkan informasi buku yang dipinjam.
	Struktur tata letak halaman riwayat peminjaman pada *dashboard user*. Menampilkan informasi buku yang pernah dipinjam, tombol *filter* status, dan s*earch box* mencari judul buku.

2. *Use Case Diagram*

Use case diagram adalah representasi dari skenario interaksi yang terjadi antara pengguna dengan sistem. Konsep ini menggambarkan bagaimana aktor berinteraksi dengan fitur atau fungsi tertentu dalam sistem. Diagram *use case* digunakan untuk memvisualisasikan hubungan antara aktor dan kegiatan yang dapat dilakukan melalui sistem (Ardiansah & Hidayatullah, 2023). Detail *use case diagram* yang telah dirancang dapat dilihat pada Gambar 3.8.

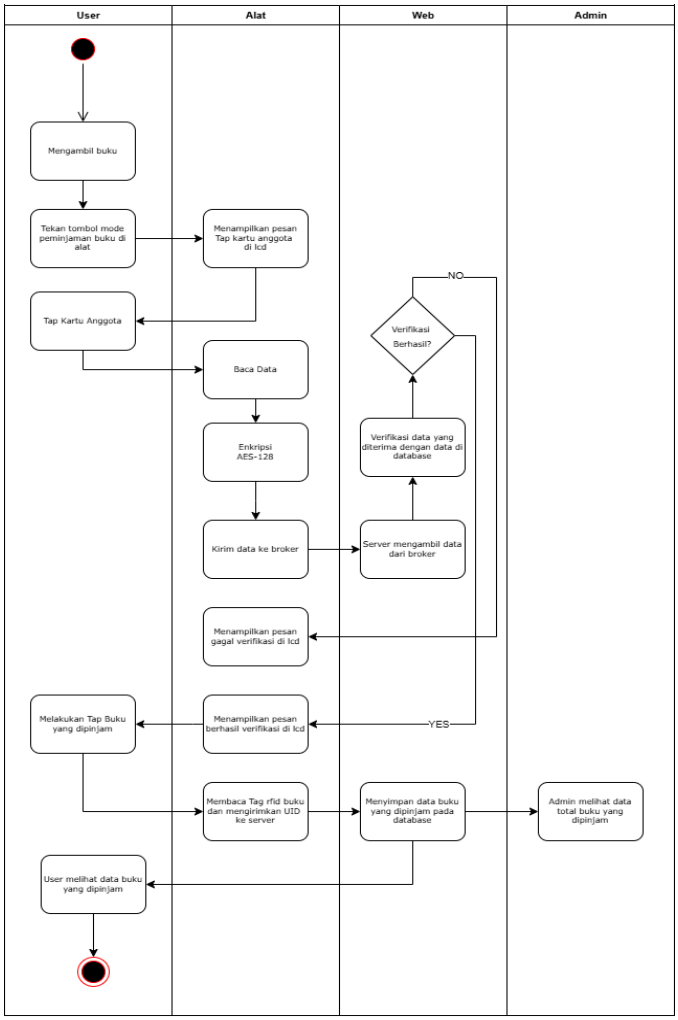
Gambar 3.8 *Use Case Diagram* Sistem Perangkat Lunak

Penulis membuat dua aktor untuk memberikan akses otorisasi tertentu pada fitur-fitur aplikasi web sesuai dengan kewenangan yang diberikan. Dapat dilihat pada Gambar 3.8 bahwa terdapat dua *dashboard* atau halaman berbeda antara admin dan *user*. Untuk *user* aktivitas yang dapat dilakukan yaitu registrasi, *login*, melihat katalog buku, mencari dan memfilter buku, melihat detail buku, meminjam buku, mengembalikan buku, melihat riwayat peminjaman, serta mengelola profil pribadi. Sementara itu, admin memiliki akses untuk mengelola baik itu pengguna, kartu atau stiker RFID, buku, serta memonitor catatan peminjaman.

3. *Activity* Diagram

Activity Diagram digunakan untuk menggambarkan alur aktivitas yang dilakukan oleh pengguna dalam sebuah sistem. Diagram ini menunjukkan langkah-langkah proses secara visual, mulai dari interaksi awal hingga hasil akhir yang diharapkan. Dengan menggunakan simbol-simbol tertentu, diagram

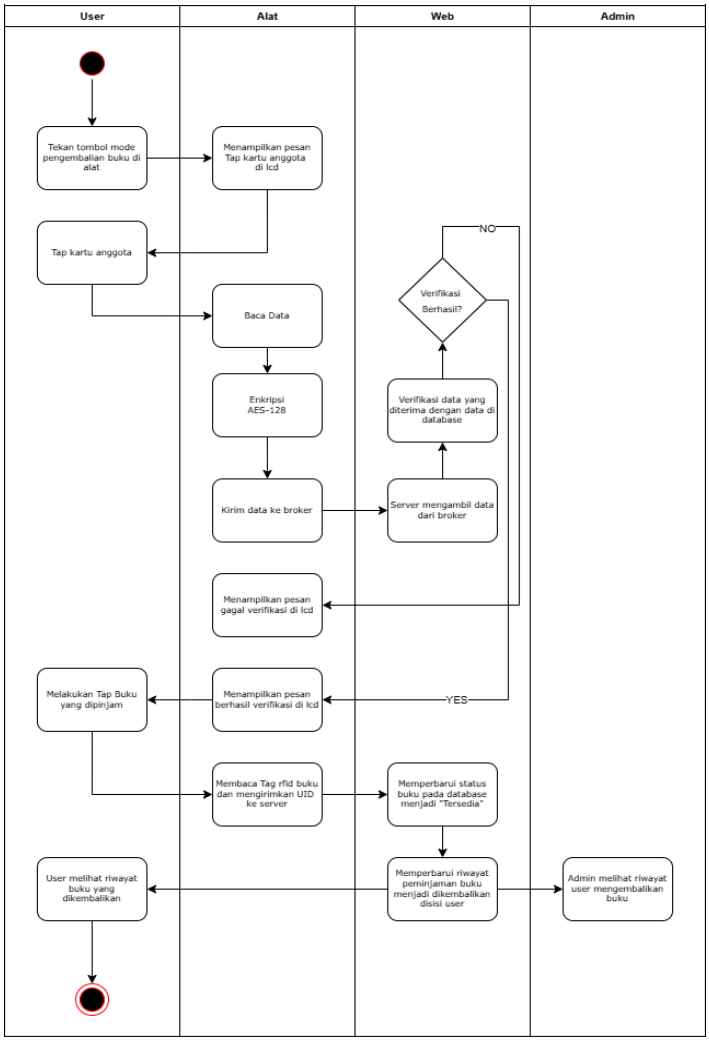
ini membantu memetakan logika alur kerja secara terstruktur. Selain itu, *activity* diagram juga mempermudah pengembang dalam memahami dan merancang fungsi sistem sesuai kebutuhan pengguna (Bintari dkk., 2024). Alur aktivitas yang dilakukan oleh *user* ketika meminjam buku dapat dilihat pada Gambar 3.9 serta mengembalikan buku pada Gambar 3.10, selain itu di sisi admin dapat memonitoring *user* yang sedang meminjam atau mengembalikan buku. Sistem yang dibuat terintegrasi antara *user* dan admin terutama pada fitur peminjaman serta pengembalian buku.



Gambar 3.9 *Activity* Diagram Fitur Peminjaman Buku

Pada proses peminjaman buku seperti yang digambarkan pada Gambar 3.9, pengguna diarahkan untuk menekan tombol mode peminjaman, lalu tap kartu anggota pada alat. Kemudian, alat membaca data dan mengenkripsi UID kartu

anggota dengan algoritma kriptografi AES-128 sebelum mengirimkannya ke broker. Server web mengambil data dari broker dan memverifikasi. Jika berhasil, pengguna melihat pesan berhasil verifikasi pada LCD dan sistem menyimpan data buku yang dipinjam di *database*.

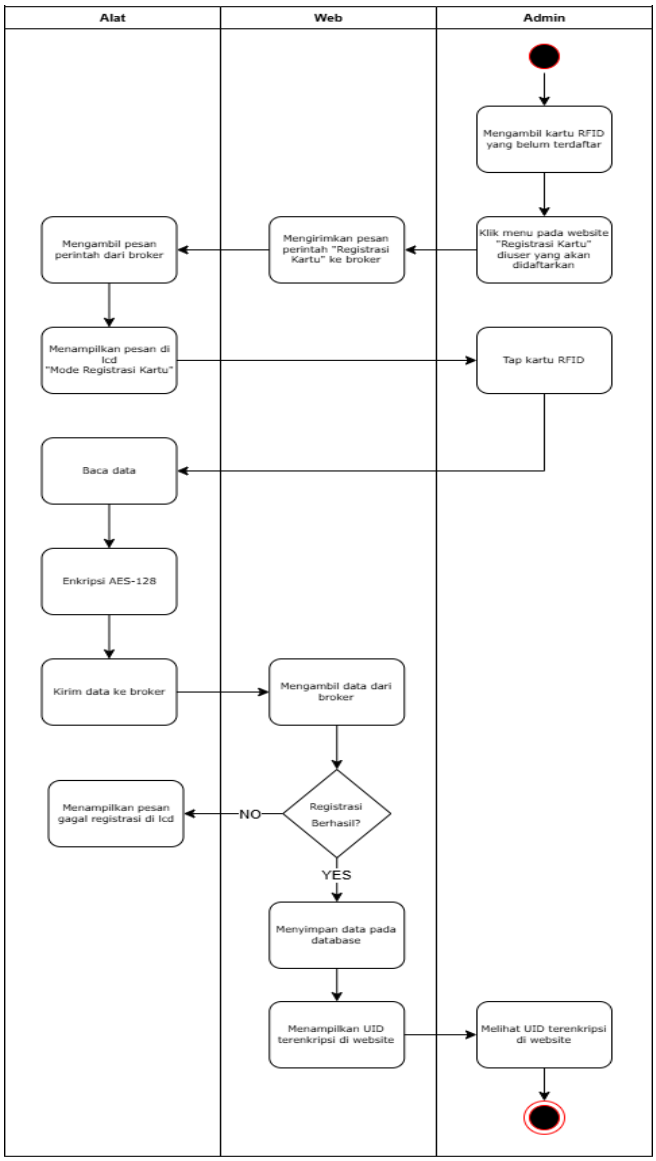


Gambar 3.10 *Activity* Diagram Fitur Pengembalian Buku

Pada proses pengembalian buku seperti yang digambarkan pada Gambar 3.10, pengguna diarahkan untuk menekan tombol mode pengembalian, lalu tap kartu anggota pada alat. Kemudian, alat membaca data dan mengenkripsi UID kartu anggota dengan algoritma kriptografi AES-128 sebelum mengirimkannya ke broker. Server web mengambil data dari broker dan memverifikasi. Jika berhasil, pengguna melihat pesan berhasil verifikasi pada

LCD, kemudian tap buku yang dipinjam dan sistem memperbarui status buku menjadi tersedia pada *database*.

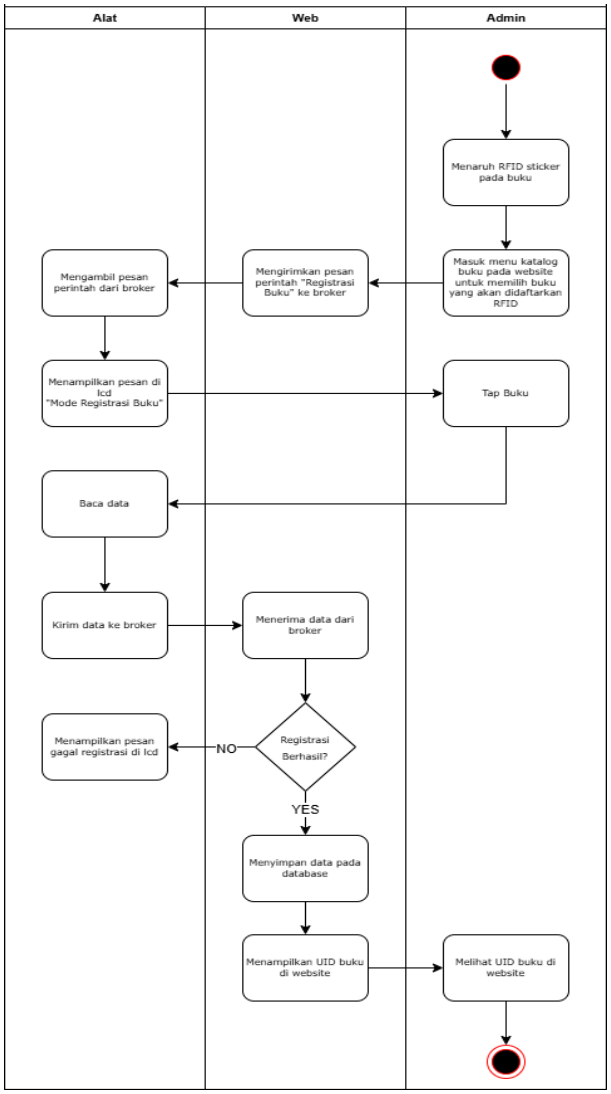
Selain fitur utama sistem, yaitu peminjaman dan pengembalian buku. Untuk mempermudah admin dalam mengatur kartu anggota perpustakaan dan buku, sistem IoT ini dirancang dengan fitur pendaftaran kartu anggota berbasis RFID serta penggunaan stiker RFID pada buku, sehingga proses manajemen koleksi buku dan data anggota perpustakaan dapat dilakukan dengan lebih praktis dan terorganisir alur aktivitas lebih detail dapat dilihat pada Gambar 3.11 dan Gambar 3.12.



Gambar 3.11 *Activity Diagram* Registrasi RFID Kartu Anggota

Seperti yang digambarkan pada Gambar 3.11, alur sistem registrasi kartu RFID anggota baru dilakukan oleh admin. proses dimulai dengan admin mengambil kartu RFID yang belum terdaftar dan memulai pendaftaran pengguna baru melalui aplikasi web. Secara bersamaan, aplikasi web mengirimkan pesan perintah "Registrasi Kartu" ke broker dan alat mengambil pesan dari broker, kemudian menampilkan pesan “Mode Registrasi Kartu” pada layar LCD 16x2. Admin melakukan tap kartu, kemudian alat membaca UID kartu RFID dan dienkripsi menggunakan AES-128 sebelum data dikirim ke broker. Jika registrasi berhasil, data akan disimpan ke dalam *database*, dan UID terenkripsi akan ditampilkan pada aplikasi web.

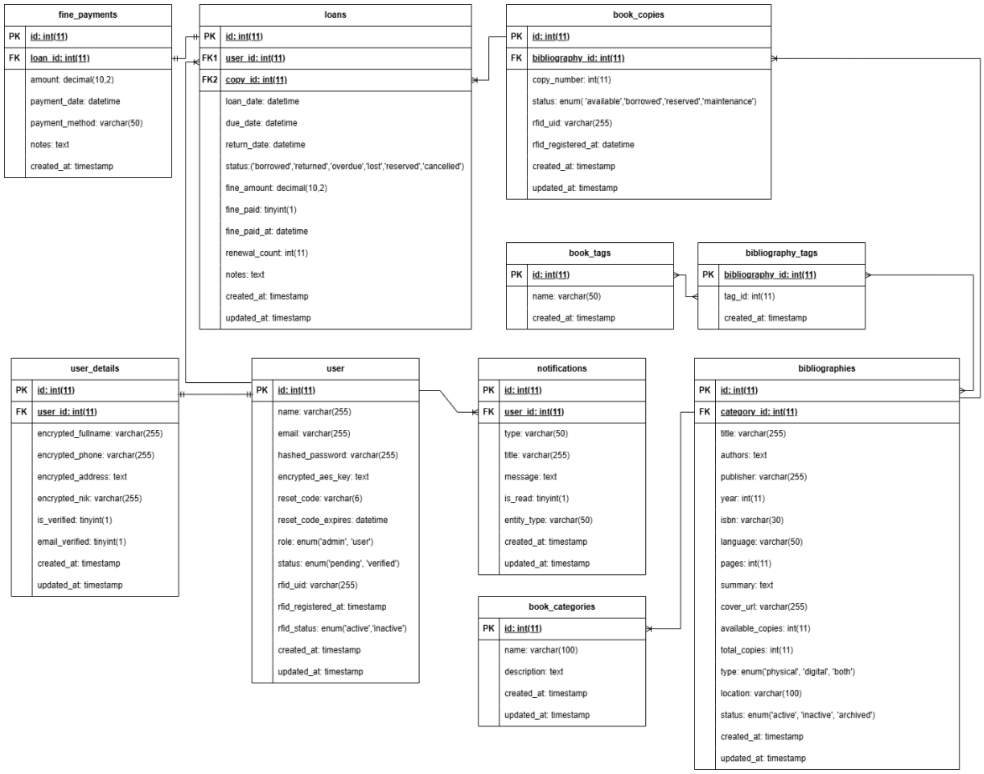
Selain fitur registrasi kartu anggota yang dilengkapi oleh RFID, untuk buku pun dilengkapi oleh RFID berbentuk stiker untuk memudahkan peminjaman, pengembalian, dan manajemen buku. Alur sistem registrasi RFID stiker buku dimulai dengan admin menaruh stiker RFID pada buku yang belum terdaftar dan memulai pendaftaran buku melalui aplikasi web dengan memilih buku mana yang akan ditambahkan stiker RFID. Secara bersamaan, aplikasi web mengirimkan pesan perintah "Registrasi Buku" ke broker dan alat mengambil pesan dari broker, kemudian menampilkan pesan “Mode Registrasi Buku” pada layar LCD 16x2. Admin melakukan tap buku, kemudian alat membaca UID stiker RFID. Jika registrasi berhasil, data akan disimpan ke dalam *database*, dan UID buku akan ditampilkan pada aplikasi web. Untuk alur detail aktivitas proses registrasi RFID stiker buku dapat dilihat pada Gambar 3.12.



Gambar 3.12 *Activity Diagram* Registrasi RFID Stiker Buku

4. *Entity Relationship* Diagram

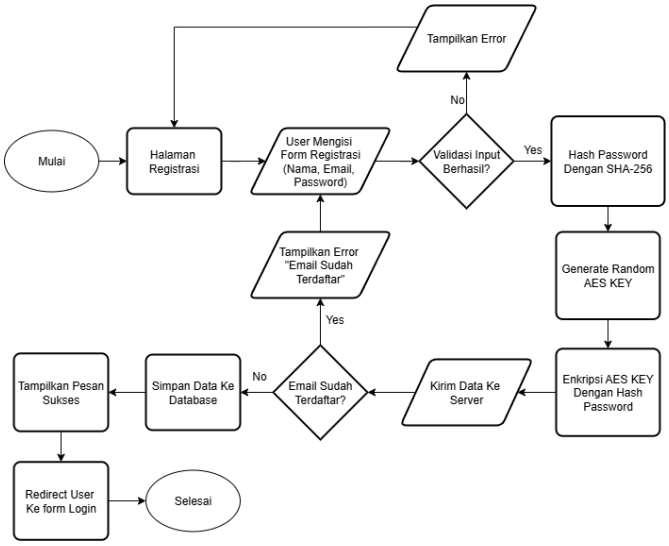
Entity relationship diagram (ERD) digunakan untuk memvisualisasikan basis data yang menunjukkan hubungan atau relasi antara entitas dan atributnya secara terstruktur dan jelas, ERD memanfaatkan notasi dan simbol untuk menjelaskan komponen-komponen tersebut. Relasi dalam ERD menunjukkan bahwa terdapat hubungan antara beberapa entitas yang berasal dari kumpulan entitas berbeda (Mukhlis & Santoso, 2023). Dalam konteks penelitian ini, jenis relasi yang dibutuhkan meliputi *one to one*, *many to many*, *many to one*. Adapun rancangan ERD yang dibuat untuk mendukung pengembangan sistem ini terlihat pada Gambar 3.13 berikut.



Gambar 3.13 ERD Sistem Perangkat Lunak

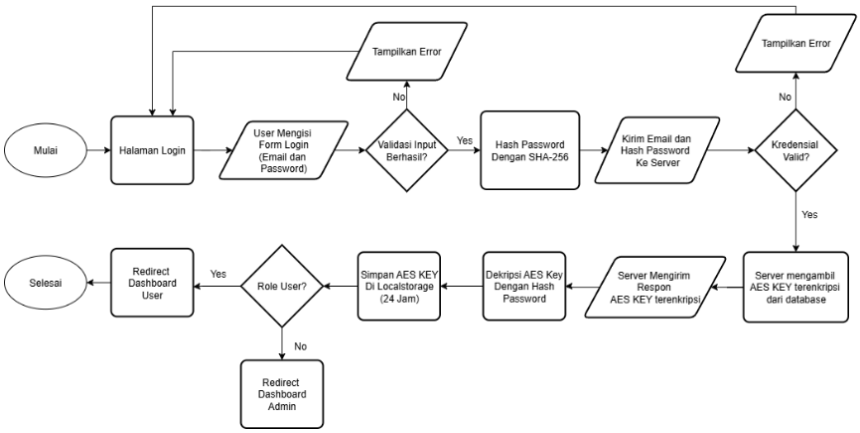
5. *Flowchart* Sistem

Diagram alir atau *flowchart* adalah representasi visual dari algoritma yang menggunakan bentuk-bentuk grafik yang saling terhubung dengan garis. Setiap bentuk grafik menggambarkan operasi tertentu dan dilengkapi teks. Diagram ini membantu proses pengembangan sistem atau algoritma, sehingga dapat dengan mudah memahami cara kerja program tanpa harus melihat kode secara langsung (Suryawan, 2023). Perancangan *flowchart* dalam penulisan ini bertujuan untuk mendukung penyusunan sistem secara terstruktur dan mendukung tahapan berikutnya, yaitu penulisan kode pada fitur-fitur utama aplikasi web yang akan dikembangkan. Fitur tersebut mencakup proses pembuatan AES *Key* yang terintegrasi dengan *Hash Password* setiap pengguna dan enkripsi serta dekripsi data pribadi anggota perpustakaan. Pada Gambar 3.14 menunjukkan alur detail proses registrasi akun.



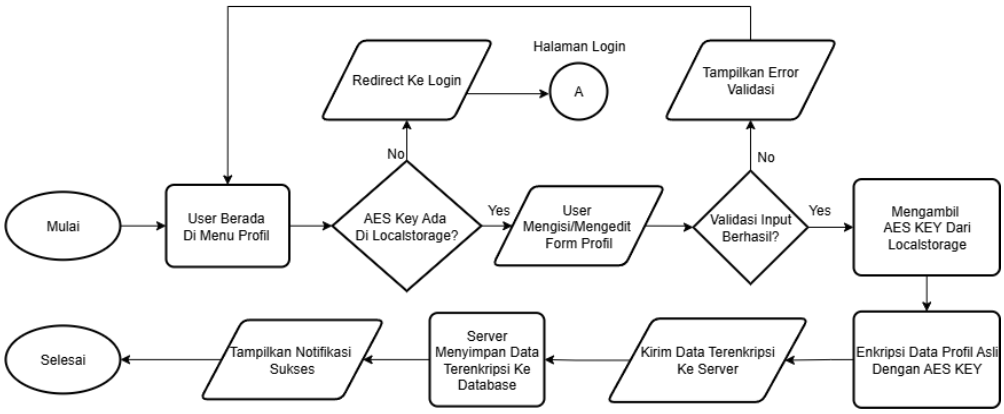
Gambar 3.14 *Flowchart* Registrasi Akun dan Pembuatan AES *Key*

Pembuatan AES *Key* bertujuan untuk menghasilkan kunci yang digunakan dalam proses enkripsi dan dekripsi data pribadi anggota perpustakaan. Seperti ditunjukkan pada Gambar 3.14, proses ini dimulai ketika pengguna melakukan registrasi akun dengan memasukkan nama, email, dan *password.* Sistem kemudian melakukan *hash password* menggunakan algoritma SHA-256, dilanjutkan dengan pembuatan AES *Key* secara acak. Hasil *generate* AES *Key* tersebut dienkripsi dengan *hash password* sebelum dikirimkan ke server. Selanjutnya, sistem memvalidasi apakah email yang didaftarkan sudah terdaftar. Jika belum, data pengguna akan disimpan ke dalam *database* dan jika sudah pengguna akan melihat pesan gagal "Email Sudah Terdaftar". Kemudian, pada Gambar 3.15 menunjukkan alur detail proses *login*.



Gambar 3.15 *Flowchart* Login dan Pengambilan AES *Key*

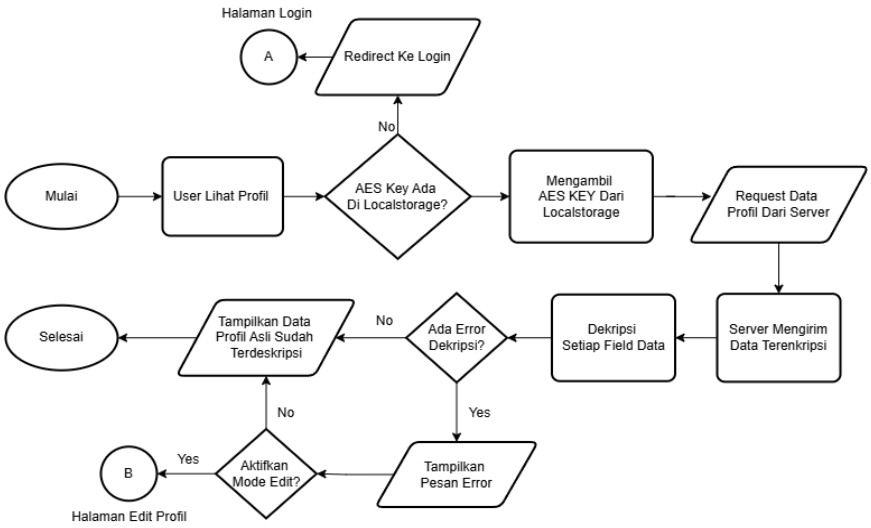
Saat proses login terjadi, sistem juga melakukan dekripsi AES *Key* dipakai untuk proses dekripsi data pribadi *user* yang akan ditampilkan pada halaman profil *user.* Pada Gambar 3.15 menunjukkan setelah pengguna mengisi formulir login dengan email dan *password* yang kemudian divalidasi oleh sistem. Jika validasi berhasil, *password* akan di *hash* menggunakan algoritma SHA-256, lalu email dan *hash password* dikirim ke server untuk diverifikasi. Setelah kredensial dinyatakan valid, server mengambil AES *Key* terenkripsi dari *database* dan mengirimkan respons kembali ke klien. Sistem klien kemudian mendekripsi AES *Key* menggunakan *hash password* yang telah dibuat sebelumnya dan menyimpannya sementara di *local storage* selama 24 jam. Setelah itu, pengguna akan diarahkan ke *dashboard* sesuai dengan perannya, apakah sebagai admin atau pengguna biasa. Jika terjadi kesalahan pada salah satu tahap, sistem akan menampilkan pesan *error* kepada pengguna. Gambar 3.16 menunjukkan alur detail proses Enkripsi data pada aplikasi web.



Gambar 3.16 *Flowchart* Enkripsi Pada Aplikasi Web Di sisi *Client*

Sesuai dengan Gambar 3.16 setelah pengguna berhasil *login*, akan diarahkan ke menu profil untuk mengisi data pribadi sebagai bentuk identifikasi anggota perpustakaan atau edit profil jika sudah mengisi data pribadi. Sistem akan memeriksa apakah AES *Key* tersedia di *local storage*. Jika tidak, pengguna akan diarahkan kembali ke halaman login. Setelah itu, pengguna mengisi formulir data pribadi, dan sistem memvalidasi input yang dimasukkan. Jika validasi berhasil, data pribadi asli akan dienkripsi menggunakan AES *Key* yang diambil dari *local storage*, kemudian data terenkripsi tersebut dikirim ke

server. *Server* akan menyimpan data terenkripsi ke *database*, dan sistem akan menampilkan notifikasi bahwa proses berhasil. Jika terjadi kesalahan dalam proses validasi, sistem akan menampilkan pesan *error* kepada pengguna. Pada Gambar 3.17 menunjukkan alur detail proses dekripsi data pada aplikasi web.



Gambar 3.17 *Flowchart* Dekripsi Pada Aplikasi Web Di sisi *Client*

Setelah pengguna mengisi atau mengedit data pribadi, jika mereka ingin melihat data tersebut, sistem akan menjalankan proses dekripsi terlebih dahulu. Sesuai dengan Gambar 3.17 proses dekripsi dimulai dengan memeriksa keberadaan AES *Key* di *local storage*. Jika AES *Key* tidak tersedia, pengguna akan diarahkan kembali ke halaman login. Jika AES *Key* ditemukan, sistem akan mengambilnya dari *local storage* dan mengirimkan permintaan data profil terenkripsi ke server. Server akan merespons dengan mengirimkan data pribadi yang terenkripsi, kemudian data didekripsi oleh sistem setiap *field* data. Jika proses dekripsi berhasil, data pribadi asli akan ditampilkan. Namun, jika terjadi kesalahan selama dekripsi, sistem akan menampilkan pesan *error*. Pengguna juga memiliki opsi untuk mengaktifkan mode edit jika diperlukan.

3.4.3 Penulisan Kode Program

Setelah melakukan rancangan desain sistem, tahapan selanjutnya adalah implementasi melalui penulisan kode program. Dalam penelitian ini, hasil

rancangan desain diimplementasikan menjadi produk berupa perangkat keras dan perangkat lunak berbasis aplikasi web. Untuk pengembangan perangkat lunak, penulis menggunakan bahasa pemrograman JavaScript dengan *Framework* ReactJs untuk membangun antarmuka, didukung oleh Tailwind CSS sebagai *framework* untuk membuat desain antarmuka aplikasi web. Penulis memanfaatkan Visual Studio Code sebagai *editor* kode JavaScript dan bahasa C++ untuk pengembangan perangkat keras dengan bantuan *plugin* PlatformIO untuk manajemen *library* perangkat keras. Selain itu, penulis menggunakan Node JS sebagai *runtime* javascript dalam membangun aplikasi *server-side* serta Express JS untuk membangun RESTful API. Selanjutnya, untuk manajemen basis data relasional penulis menggunakan MySQL dan HiveMQ Cloud sebagai broker untuk mengelola topik *subscribe* dan *publish* data antara aplikasi web dengan perangkat keras berbasis protokol MQTT.

3.4.4 Pengujian

Pada tahapan pengujian, penulis melakukan pengujian sistem *black box testing* merupakan cara pengujian berdasarkan kepada spesifikasi kebutuhan dan tidak perlu dilakukan analisis kode. Teknik ini dilakukan hanya dari sudut pandang pengguna. *black box testing* menilai tingkat keefektifan sistem berdasarkan bekerja atau tidaknya fitur, dengan melakukan pengujian *black box* peneliti dapat mengidentifikasi kesalahan fungsionalitas pada sistem yang dikembangkan.

3.4.5 Penerapan dan Pemeliharaan

Pada tahap terakhir metode Waterfall yaitu penerapan, aplikasi web yang sebelumnya dikembangkan pada lingkungan lokal diimplementasikan ke dalam lingkungan produksi sesuai dengan rancangan yang telah ditetapkan. Aplikasi web pada sisi *frontend* diimplementasikan pada layanan Vercel untuk proses *hosting*, sementara *backend* dijalankan pada DigitalOcean melalui Dockerfile, dan *database* SQL disimpan serta dikelola pada *platform* AWS RDS.

Seluruh komponen diuji secara menyeluruh untuk memastikan seluruh fitur bekerja sesuai dengan rancangan dan memenuhi kebutuhan pengguna. Setelah penerapan, tahapan pemeliharaan dilakukan melalui evaluasi berkelanjutan,

termasuk perbaikan *bug* dan pengoptimalan fitur berdasarkan hasil pengujian, guna memastikan sistem tetap relevan dan stabil sebagai bagian dari penelitian akademik.

3.5 Uji Sistem

Uji coba sistem dilakukan dengan menggunakan metode *black box testing* pengujian ini bertujuan untuk memvalidasi fungsionalitas aplikasi tanpa perlu menguji kode atau struktur internal aplikasi. Metode ini mengevaluasi sistem berdasarkan kesesuaian antara input yang diberikan dan *output* yang diharapkan. Pengujian ini sangat efektif untuk memastikan kualitas antarmuka pengguna serta interaksi sistem (Ar Rachman dkk., 2025).

Setelah dilakukan desain untuk pengembangan sistem pengamanan sirkulasi buku perpustakaan berbasis kriptografi AES-128 dengan RFID terintegrasi IoT. Selanjutnya, dilakukan tahapan pengujian yang dilakukan dengan *black box testing* yang mencakup fitur-fitur utama sistem yang menjadi fondasi dari seluruh fungsionalitas aplikasi web. Tabel 3.4 menjabarkan setiap serangkaian kasus uji yang dirancang untuk mengevaluasi respons sistem terhadap input tertentu dan membandingkannya dengan *output* yang diharapkan. Perancangan pengujian ini menjadi dasar untuk validasi fungsionalitas yang lebih komprehensif pada Bab IV.

Tabel 3.4 Rancangan Pengujian Komponen Utama Sistem

Komponen	**Fitur Pengujian**	**Skenario Pengujian**	**Hasil yang Diharapkan**
Sistem autentikasi dan pembuatan AES *key*	Registrasi pengguna	*User* mengisi *form* registrasi dengan data valid (nama, email, dan *password*)	Akun berhasil dibuat, AES *Key* berhasil dibuat dan dienkripsi dengan *Hash Password,* data disimpan di *database*
	Login Pengguna	*User* mengisi email dan *password* valid	*Login* berhasil, AES *Key* didekripsi dari *database* dan disimpan di *local storage*, dan *redirect* ke *dashboard*

Komponen	Fitur Pengujian	Skenario Pengujian	Hasil yang Diharapkan
	Login dengan kredensial salah	*User* mengisi email dan *password* invalid	Sistem menolak *login* dan menampilkan pesan *error* "Email atau Password Salah"
	Logout Pengguna	*User* menekan tombol *logout*	*Session* dan AES KEY dihapus dari *local storage.*
Proses peminjaman dan pengembalian buku	Peminjaman Buku	*User* meminjam buku yang tersedia	Status buku berubah menjadi "dipinjam" dan tercatat pada halaman peminjaman aktif serta histori peminjaman
	Pengembalian Buku	*User* mengembalikan buku yang dipinjam	Status buku berubah menjadi "tersedia" dan Status catatan peminjaman diperbarui menjadi "Dikembalikan"
Sistem perangkat keras RFID	Registrasi kartu RFID anggota perpustakaan	Admin menginisiasi registrasi RFID dari *web interface* dan menempelkan kartu	Sistem mengenkripsi UID kartu dan menyimpan di *database*
	Registrasi stiker RFID buku	Admin menempelkan stiker RFID pada buku dan menginisiasi	UID RFID stiker disimpan di *database* dan terhubung dengan data buku di sistem

Komponen	Fitur Pengujian	Skenario Pengujian	Hasil yang Diharapkan
		registrasi RFID dari *web interface*	
Komunikasi perangkat keras dan server	Komunikasi MQTT ke broker HiveMQ *cloud*	ESP32 mengirim data ke server melalui protokol MQTT	Koneksi SSL/TSL berhasil dan data dikirim serta diterima dengan benar
	Pengiriman dari aplikasi web ke ESP32	Aplikasi web mengirim perintah mode ke ESP32	ESP32 berhasil menerima dan mengeksekusi perintah

Dengan pengujian yang dirancang seperti pada Tabel 3.4, sistem diharapkan dapat divalidasi keandalannya terutama pada aspek keamanan data pribadi anggota perpustakaan. Seperti, proses enkripsi data pribadi di sisi aplikasi web menggunakan algoritma kriptografi AES-128, sistem autentikasi, pembuatan AES *Key*, proses peminjaman dan pengembalian buku, dan komunikasi antara perangkat keras dengan aplikasi web. Rancangan ini sejalan dengan desain pengembangan sistem yang telah dilakukan sebelumnya, validasi komponen inti dilakukan terlebih dahulu sebelum pengujian fungsional keseluruhan sistem.

3.6 Evaluasi Hasil Sistem

Dalam metode *design and development* evaluasi bertujuan untuk menganalisis hasil pengujian sistem yang telah dibangun. Hasil evaluasi dilakukan melalui tiga tahapan pengujian utama yaitu *black Box testing*, pengujian menggunakan Wireshark, dan perhitungan korelasi pearson.

Pada penelitian ini, *black box testing* digunakan untuk memvalidasi fungsionalitas sistem pengamanan sirkulasi buku perpustakaan berbasis kriptografi AES-128 dengan RFID terintegrasi IoT secara keseluruhan, baik pada komponen aplikasi web ataupun perangkat keras, tanpa meninjau struktur kode internal. Fokus pengujian ini adalah memastikan fitur-fitur berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Pada aplikasi web perpustakaan, pengujian mencakup proses autentikasi, algoritma AES-128 untuk enkripsi data pribadi

anggota perpustakaan, manajemen koleksi buku, sistem peminjaman atau pengembalian buku, dan lainnya. Sementara pada perangkat keras, pengujian meliputi proses pengimplementasian protokol SSL/TSL dan sertifikat CA pada protokol MQTT, dikombinasikan dengan enkripsi UID kartu RFID anggota perpustakaan menggunakan AES-128 pada perangkat ESP32 sebelum data dikirimkan ke broker, validasi kartu RFID anggota atau stiker RFID buku, proses peminjaman atau pengembalian buku secara langsung dengan alat, dan lainnya. Tabel pengujian *black box* merupakan parameter keberhasilan bahwa kedua sistem dapat berfungsi saling terintegrasi.

Untuk menganalisis serta memverifikasi keamanan data selama proses pengiriman dan penerimaan di lalu lintas jaringan, dilakukan pengujian menggunakan aplikasi Wireshark untuk memvalidasi bahwa data pribadi anggota perpustakaan selama proses transmisi tidak terekspos sebagai data asli atau *plaintext*. Hasil pengujian ini nantinya diharapkan bahwa pendekatan enkripsi data selama proses transmisi data dalam jaringan, dapat memberikan perlindungan komprehensif terhadap data pribadi pengguna.

Pengujian korelasi pearson dalam penelitian ini digunakan untuk mengevaluasi kualitas dan kekuatan algoritma enkripsi yang diterapkan pada data pribadi anggota perpustakaan. Korelasi pearson digunakan untuk mengukur hubungan antara *plaintext* dan *chipertext*, proses analisis ini nantinya menggunakan data uji NIK pengguna dalam bentuk numerik berbasis nilai ASC. Data ini nantinya akan dihitung koefisien korelasinya. Jika hasil perhitungan korelasi ini 0 maka tidak ada korelasi antara *plaintext* dengan *chipertext*, sedangkan jika nilai 1 maupun -1 berarti terdapat hubungan linear antara *plaintext* dengan *chipertext*.

Dari hasil ketiga pengujian ini nantinya diharapkan dapat diperoleh evaluasi yang sesuai dengan parameter yang telah ditetapkan dalam penelitian ini. Hasil pengujian *black box* memverifikasi kesesuaian fungsionalitas sistem dengan kebutuhan pengguna, analisis Wireshark memberikan bukti keamanan transmisi data antara komponen sistem, dan pengujian korelasi pearson memvalidasi kualitas implementasi algoritma kriptografi AES-128 untuk melindungi data pribadi

anggota perpustakaan. Keseluruhan evaluasi ini akan menjadi dasar untuk menilai keberhasilan penelitian yang telah dilakukan.

3.7 Laporan Hasil Pengujian

Tahapan terakhir dalam metode penelitian *design and development* adalah penyusunan laporan hasil pengujian. Dari hasil pengujian serta evaluasi kemudian dapat disimpulkan menjadi bentuk laporan tertulis yang terstruktur sebagai dokumen skripsi. Laporan ini tidak hanya berfungsi sebagai dokumen penelitian, tetapi juga memberikan dasar yang kuat untuk pengembangan lebih lanjut, baik dalam peningkatan sistem yang telah ada maupun eksplorasi fitur-fitur baru di masa depan.