

# **SISTEM PEMINJAMAN ALAT BANTU MEDIS BERBASIS WEB MENGGUNAKAN REACT JS DAN GO**

Diajukan untuk Memenuhi Kelulusan Matakuliah Proyek 3  
Program Studi DIV Teknik Informatika

## **DISUSUN OLEH :**

Moch Restu Agis Burhanudin (714230059)

Aghni Hasna Mufida (714230069)



**Universitas Logistik & Bisnis Internasional**

PROGRAM STUDI DIV TEKNIK INFORMATIKA  
UNIVERSITAS LOGISTIK DAN BISNIS INTERNASIONAL  
BANDUNG  
2026

# **LEMBAR PENGESAHAN**

Laporan Proyek III ini telah diperiksa, disetujui dan disidangkan  
di Bandung, ...../...../2026

**Oleh:**

Penguji Pendamping,

Penguji Utama,

Rolly Maulana Awangga, S.T., M.T.    Nisa Hanum Harani, S.Kom., M.T.

NIK : 117.86.219

NIK : 117.89.223

Pembimbing,

Koordinator Proyek II

Nisa Hanum Harani, S.Kom., M.T.    Roni Habibi, S.Kom., M.T., SFPC

NIK : 117.89.223

NIK : 103.78069

Menyetujui,

Ketua Program Studi D-IV

Teknik Informatika

Roni Andarsyah, S.T., M.Kom.

NIK : 115.88.193

# **SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME**

Nama : Moch Restu Agis Burhanudin  
NPM : 714230059  
Program Studi : D-IV Teknik Informatika  
Judul : Sistem Pengembangan dan Rental Media Berbasis Web  
Menggunakan React.js dan GO

Menyatakan bahwa:

1. Proyek Pengembangan Aplikasi (Proyek 3) yang saya susun merupakan karya asli yang dibuat secara mandiri dan belum pernah dipublikasi sebagai syarat kelulusan pada Program Studi D-IV Teknik Informatika, Politeknik Logistik dan Bisnis Internasional seyogyanya di persyaratan tinggi lainnya. Proyek ini berkenaan dengan sistem, perangkat, dan teknologi yang terkini dan lainnya sesuai dengan aturan dan dosen pembimbing sesuai ketentuan yang berlaku.
2. Seluruh isi dalam proyek ini tidak sepenuhnya karya saya sendiri melainkan hasil kerja pengembangan seminar secara jujur. Setiap rujukan terhadap karya ilmiah atau pendapat orang lain belum didokumentasikan secara tertulis dan dalam rangka membatilkan sama penggunaan yang didekumsikan ke dalam daftar pustaka sesuai kaidah penulisan ilmiah.
3. Pernyataan ini saya buat dengan sebetar-benarnya. Apabila di kemudian hari terbukti pernyataan saya tidak benar, maka saya bersedia menerima sanksi terbulat yang berlaku di lingkungan persyaratan tinggi.

BANDUNG, 2026  
Yang membuat pernyataan,

Moch Restu Agis Burhanudin  
NPM : 714230059

# **SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME**

Nama : Aghni Hasna Mufida  
NPM : 714230069  
Program Studi : D-IV Teknik Informatika  
Judul : Sistem Pengembangan dan Rental Media Berbasis Web  
Menggunakan React.js dan GO

Menyatakan bahwa:

1. Proyek Pengembangan Aplikasi (Proyek 3) yang saya susun merupakan karya asli yang dibuat secara mandiri dan belum pernah dipublikasi sebagai syarat kelulusan pada Program Studi D-IV Teknik Informatika, Politeknik Logistik dan Bisnis Internasional seyogyanya di persyaratan tinggi lainnya. Proyek ini berkenaan dengan sistem, perangkat, dan teknologi yang terkini dan lainnya sesuai dengan aturan dan dosen pembimbing sesuai ketentuan yang berlaku.
2. Seluruh isi dalam proyek ini tidak sepenuhnya karya saya sendiri melainkan hasil kerja pengembangan seminar secara jujur. Setiap rujukan terhadap karya ilmiah atau pendapat orang lain belum didokumentasikan secara tertulis dan dalam rangka membatilkan sama penggunaan yang didekumsikan ke dalam daftar pustaka sesuai kaidah penulisan ilmiah.
3. Pernyataan ini saya buat dengan sebetar-benarnya. Apabila di kemudian hari terbukti pernyataan saya tidak benar, maka saya bersedia menerima sanksi terbulat yang berlaku di lingkungan persyaratan tinggi.

BANDUNG, 2026  
Yang membuat pernyataan,

Aghni Hasna Mufida  
NPM : 714230069

## ABSTRAK

Sebagian besar masyarakat menghadapi masalah aksesibilitas alat bantu medis, terutama terkait biaya dan keterbatasan informasi stok di fasilitas kesehatan. Tujuan dari penelitian ini adalah untuk membangun MedisLink, sebuah sistem informasi berbasis *web* yang menghubungkan peminjaman dan donasi alat medis. Sistem ini akan memudahkan hubungan antara donatur dan pengguna yang membutuhkan secara jelas dan efektif. Pengembangan teknologi sistem ini menggunakan arsitektur yang terpisah dengan *React.js* di sisi depan, *Golang (Fiber)* di sisi *backend*, dan *MongoDB* sebagai basis data non-relasional. Struktur *API* yang responsif dan terbagi berdasarkan layanan fungsional mendukung efisiensi sistem dan menjaga keamanan akses pengguna melalui autentikasi *JSON Web Token (JWT)*. Sistem notifikasi pengembalian alat otomatis adalah salah satu fitur unggulan yang memastikan waktu distribusi inventaris yang tepat dengan menggunakan mekanisme *Cron Job* dengan interval waktu harian.

**Kata Kunci:** MedisLink, *React.js*, *Go (Fiber)*, *MongoDB*, *JWT*, *Cron Job*.

## ABSTRACT

Most people face accessibility issues for medical devices, particularly regarding costs and limited stock information at healthcare facilities. The purpose of this research is to build *MedisLink*, a *web-based information system* that connects medical equipment lending and donation. This system will facilitate clear and effective relationships between donors and users in need. The system technology development uses a *separated architecture* with *React.js* on the *frontend*, *Golang (Fiber)* on the *backend*, and *MongoDB* as a *non-relational database*. The responsive *API* structure divided by *functional services* supports *system efficiency* and maintains *user access security* through *JSON Web Token (JWT) authentication*. The automatic equipment return *notification system* is one of the featured capabilities that ensures proper *inventory distribution* timing using *Cron Job mechanisms* with daily intervals.

**Keywords:** *MedisLink, React.js, Go (Fiber), MongoDB, JWT, Cron Job.*

## **KATA PENGANTAR**

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga kami dapat menyelesaikan Proyek III ini. Penulisan proyek ini merupakan bagian dari upaya saya menuju gelar Sarjana Terapan Program Studi Teknik Informatika Universitas Logistik & Bisnis Internasional Bandung. Kami menyadari bahwa hasil ini merupakan hasil dukungan dan bimbingan yang kami terima dari semua pihak mulai dari masa perkuliahan hingga penyusunan proyek III ini. Oleh karena itu, kami mengucapkan terima kasih kepada semua pihak yang telah mendukung dan membimbing kami.

## **DAFTAR ISI**

# **Daftar Isi**

<b>ABSTRAK</b>	<b>1</b>
<b>ABSTRACT</b>	<b>2</b>
<b>KATA PENGANTAR</b>	<b>3</b>
<b>BAB I PENDAHULUAN</b>	<b>10</b>
<b>1 Pendahuluan</b>	<b>11</b>
1.1 Latar Belakang . . . . .	11
1.2 Deskripsi Aplikasi . . . . .	12
1.3 Identifikasi Masalah . . . . .	12
1.4 Tujuan . . . . .	13
1.5 Ruang Lingkup . . . . .	13
1.5.1 Pengembangan Teknologi . . . . .	13
1.5.2 Peningkatan Akses dan Efisiensi Layanan . . . . .	13
<b>BAB II LANDASAN TEORI</b>	<b>14</b>
<b>2 Tinjauan Pustaka</b>	<b>15</b>
2.1 Landasan Teori . . . . .	15
2.1.1 API . . . . .	15
2.1.2 Node.js . . . . .	15
2.1.3 Go (Golang) . . . . .	16
2.1.4 React.js . . . . .	16
2.1.5 JavaScript . . . . .	16

2.1.6	MongoDB . . . . .	16
2.1.7	DaisyUI . . . . .	17
2.1.8	Axios . . . . .	17
2.1.9	UML . . . . .	17
2.2	Use Case Diagram . . . . .	17
2.3	Diagram Aktivitas (Activity Diagram) . . . . .	18
2.4	Diagram Urutan (Sequence Diagram) . . . . .	19
2.5	Deployment Diagram . . . . .	21
<b>BAB III METODE PENELITIAN</b>		<b>22</b>
<b>3 Analisis dan Perancangan</b>		<b>23</b>
3.1	Analisis Sistem pada Aplikasi . . . . .	23
3.1.1	Analisis Sistem yang Sedang Berjalan . . . . .	23
3.2	Analisis Sistem yang Akan Dibangun . . . . .	24
3.2.1	Alur Sistem yang Akan Dibangun . . . . .	24
3.2.2	Kebutuhan Fungsional . . . . .	24
3.2.3	Kebutuhan Nonfungsional . . . . .	25
3.3	Use Case Diagram . . . . .	26
3.4	Class Diagram . . . . .	27
3.5	Activity Diagram . . . . .	28
3.5.1	Tamu Melihat Landing Page . . . . .	28
3.5.2	Registrasi . . . . .	29
3.5.3	Login User . . . . .	30
3.5.4	Peminjaman Alat Bantu Medis . . . . .	31
3.5.5	Donasi Alat . . . . .	32
3.5.6	Riwayat Peminjaman User . . . . .	33
3.5.7	Riwayat Donasi User . . . . .	34
3.5.8	User Melihat Notifikasi . . . . .	35
3.5.9	Mengelola Akun . . . . .	36
3.5.10	Login Admin . . . . .	37
3.5.11	Admin Mengelola Peminjaman . . . . .	39
3.5.12	Admin Mengelola Inventaris . . . . .	40
3.5.13	Admin Mengelola Berita . . . . .	41
3.5.14	Admin Mengelola Iklan . . . . .	42
3.5.15	Admin Konfirmasi Donasi . . . . .	43
3.6	Sequence Diagram . . . . .	44
3.6.1	Tamu Melihat Landing Page . . . . .	44

3.6.2	Login . . . . .	45
3.6.3	Peminjaman Alat Bantu Medis . . . . .	45
3.6.4	User Donasi Alat . . . . .	46
3.6.5	Riwayat Peminjaman User . . . . .	47
3.6.6	Riwayat Donasi User . . . . .	47
3.6.7	Mengelola Akun User . . . . .	48
3.6.8	User Melihat Notifikasi . . . . .	48
3.6.9	Admin Mengelola Peminjaman . . . . .	49
3.6.10	Admin Mengelola Inventaris . . . . .	50
3.6.11	Admin Mengelola Berita . . . . .	51
3.6.12	Admin Mengelola Iklan . . . . .	52
3.6.13	Admin Konfirmasi Donasi . . . . .	53
3.7	Deployment . . . . .	54
<b>BAB IV HASIL DAN PEMBAHASAN</b>		<b>55</b>
<b>4 IMPLEMENTASI DAN PENGUJIAN</b>		<b>56</b>
4.1	Pembahasan Hasil Implementasi . . . . .	56
4.1.1	Landing Page . . . . .	56
<b>BAB V KESIMPULAN DAN SARAN</b>		<b>57</b>
<b>Bibliografi</b>		<b>58</b>

## **DAFTAR TABEL**

# **Daftar Tabel**

2.1	Simbol Use Case Diagram . . . . .	18
2.2	Simbol Activity Diagram . . . . .	19
2.3	Simbol Sequence Diagram . . . . .	20
2.4	Simbol Deployment Diagram . . . . .	21

## **DAFTAR GAMBAR**

# **Daftar Gambar**

3.1	Use Case Diagram Sistem MedisLink . . . . .	26
3.2	Class Diagram Sistem MedisLink . . . . .	27
3.3	Activity Diagram Tamu Melihat Landing Page . . . . .	28
3.4	Activity Diagram Registrasi . . . . .	29
3.5	Activity Diagram Login User . . . . .	30
3.6	Activity Diagram Peminjaman Alat Bantu Medis . . . . .	31
3.7	Activity Diagram Donasi Alat . . . . .	32
3.8	Activity Diagram Riwayat Peminjaman . . . . .	33
3.9	Activity Diagram Riwayat Donasi User . . . . .	34
3.10	Activity Diagram User Melihat Notifikasi . . . . .	35
3.11	Activity Diagram Mengelola Akun . . . . .	36
3.12	Activity Diagram Login Admin . . . . .	37
3.13	Activity Diagram Admin Mengelola Peminjaman . . . . .	39
3.14	Activity Diagram Admin Mengelola Inventaris . . . . .	40
3.15	Activity Diagram Admin Mengelola Berita . . . . .	41
3.16	Activity Diagram Admin Mengelola Iklan . . . . .	42
3.17	Activity Diagram Admin Konfirmasi Donasi . . . . .	43
3.18	Sequence Diagram Tamu Melihat Landing Page . . . . .	44
3.19	Sequence Diagram Login . . . . .	45
3.20	Sequence Diagram Peminjaman Alat Bantu Medis . . . . .	45
3.21	Sequence Diagram User Donasi Alat . . . . .	46
3.22	Sequence Diagram Riwayat Peminjaman User . . . . .	47
3.23	Sequence Diagram Riwayat Donasi User . . . . .	47
3.24	Sequence Diagram Mengelola Akun User . . . . .	48

3.25 Sequence Diagram User Melihat Notifikasi . . . . .	48
3.26 Sequence Diagram Admin Mengelola Peminjaman . . . . .	49
3.27 Sequence Diagram Admin Mengelola Inventaris . . . . .	50
3.28 Sequence Diagram Admin Mengelola Berita . . . . .	51
3.29 Sequence Diagram Admin Mengelola Iklan . . . . .	52
3.30 Sequence Diagram Admin Konfirmasi Donasi . . . . .	53
3.31 Deployment Diagram MedisLink . . . . .	54
4.1 Halaman Landing Page . . . . .	56

# **BAB I**

## **PENDAHULUAN**

# Bab 1

## Pendahuluan

### 1.1 Latar Belakang

Keterbatasan akses terhadap alat bantu medis di Indonesia masih menjadi masalah kesehatan masyarakat yang signifikan. Banyak penyandang disabilitas dan pasien dengan kebutuhan medis khusus mengalami kesulitan mendapatkan alat bantu seperti kursi roda, tabung oksigen, dan alat rehabilitasi lainnya karena faktor ekonomi dan keterbatasan ketersediaan alat di fasilitas kesehatan. Akses yang terbatas ini tidak hanya memperburuk kondisi kesehatan pengguna tetapi juga berdampak pada tingkat kecacatan dan risiko kematian akibat tidak terpenuhinya kebutuhan dasar medis (T. N. P. et al., 2022).

Banyak fasilitas kesehatan, terutama rumah sakit umum, sering kali kehabisan stok alat bantu medis yang dibutuhkan oleh pasien, sementara kemampuan masyarakat untuk membeli alat tersebut secara mandiri masih rendah. Hal ini menunjukkan perlunya solusi alternatif agar alat bantu medis dapat diakses oleh masyarakat tanpa harus membeli dengan biaya tinggi (F. et al., 2022).

Dalam konteks tersebut, pengembangan sistem peminjaman alat bantu medis menjadi solusi inovatif untuk mengatasi keterbatasan akses tersebut. Sistem ini memungkinkan masyarakat memperoleh alat bantu medis secara temporer melalui mekanisme peminjaman, terutama bagi mereka yang belum mampu membeli alat tersebut atau ketika stok di rumah sakit habis. Dengan adanya sistem ini, diharapkan ketersediaan alat bantu medis bagi masyarakat dapat meningkat serta risiko komplikasi kesehatan akibat keterbatasan alat dapat dikurangi.

## 1.2 Deskripsi Aplikasi

Aplikasi *MedisLink* merupakan platform berbasis web yang dirancang untuk membantu masyarakat memperoleh akses alat bantu medis secara lebih mudah, cepat, dan efisien melalui sistem peminjaman. Aplikasi ini menjadi solusi bagi individu yang membutuhkan alat bantu medis seperti kursi roda, tongkat kruk, *walker*, atau alat pernapasan namun terkendala biaya atau ketersediaan alat di fasilitas kesehatan.

Melalui MedisLink, pengguna dapat melihat ketersediaan alat, melakukan peminjaman, melacak status pinjaman, serta menerima notifikasi pengembalian secara otomatis. Selain itu, sistem juga mempermudah pengelolaan inventaris bagi admin sehingga pemantauan stok dapat dilakukan secara transparan dan akurat.

Fitur utama aplikasi MedisLink meliputi:

### 1. Peminjaman Alat Bantu Medis

Pengguna dapat memilih jenis alat yang tersedia, melihat detail alat, dan melakukan peminjaman secara daring. Setiap transaksi tersimpan dalam sistem sehingga status peminjaman dapat dipantau.

### 2. Pengelolaan Inventaris Alat

Admin dapat mengelola stok alat, menambahkan alat baru, memperbaiki kondisi alat, serta mencatat riwayat keluar-masuk alat melalui fitur *Inventory Log*.

### 3. Notifikasi Pengembalian

Sistem mengirimkan notifikasi kepada pengguna untuk mengingatkan jadwal pengembalian alat agar dapat digunakan kembali oleh peminjam lainnya.

## 1.3 Identifikasi Masalah

Berdasarkan latar belakang tersebut, permasalahan yang diidentifikasi adalah:

1. Bagaimana membangun sistem peminjaman alat bantu medis yang memudahkan masyarakat mengakses alat tanpa harus membelinya?
2. Bagaimana sistem dapat membantu mengatasi keterbatasan stok alat bantu medis di rumah sakit atau fasilitas kesehatan?

3. Bagaimana memfasilitasi masyarakat yang memiliki alat bantu medis tidak terpakai agar dapat mendonasikannya secara tepat sasaran?

## 1.4 Tujuan

Tujuan pengembangan sistem ini adalah menyediakan solusi berupa sistem peminjaman alat bantu medis yang dapat memperluas akses masyarakat terhadap alat kesehatan.

Tujuan khusus proyek ini adalah:

1. Merancang dan membangun platform peminjaman alat bantu medis berbasis web.
2. Mengembangkan sistem manajemen inventaris untuk melacak perputaran dan ketersediaan alat.
3. Menyediakan fitur donasi yang memfasilitasi penyaluran alat bantu medis yang tidak terpakai.

## 1.5 Ruang Lingkup

### 1.5.1 Pengembangan Teknologi

Dokumentasi ini membahas proses pengembangan sistem peminjaman alat bantu medis berbasis web menggunakan Golang (Fiber) sebagai *backend*, React.js sebagai *frontend*, dan MongoDB sebagai basis data. Sistem dilengkapi dengan autentikasi JWT untuk menjaga keamanan akses pengguna serta struktur API yang efisien (Fiber, 2023).

Selain itu, sistem menyediakan fitur notifikasi pengembalian alat untuk mengingatkan pengguna mengenai batas waktu pengembalian. Dokumentasi ini mencakup perancangan arsitektur, desain basis data, serta implementasi fitur aplikasi.

### 1.5.2 Peningkatan Akses dan Efisiensi Layanan

Melalui pengembangan sistem MedisLink, aplikasi ini diharapkan dapat meningkatkan akses masyarakat terhadap alat bantu medis. Dengan fitur peminjaman daring, pengguna tidak perlu datang langsung ke fasilitas kesehatan sehingga proses menjadi lebih cepat dan efisien. Sistem juga dilengkapi dengan kategori alat, ulasan pengguna, serta pengelolaan inventaris yang terstruktur.

## **BAB II**

### **LANDASAN TEORI**

## Bab 2

# Tinjauan Pustaka

### 2.1 Landasan Teori

#### 2.1.1 API

REST API (*Representational State Transfer API*) merupakan kerangka kerja yang memungkinkan layanan diakses di berbagai platform dan lingkungan, mendukung interoperabilitas, serta mematuhi standar web. REST API menjadi metode umum untuk mempublikasikan layanan di internet karena sifatnya yang tidak terikat pada platform tertentu dan kemudahan implementasinya (Banubakode & Chore, 2022). RESTful API terdiri dari berbagai *endpoint* yang menunjukkan fungsi spesifik dari suatu proses bisnis dan dapat diakses melalui protokol HTTP menggunakan metode seperti *GET*, *POST*, *PUT*, dan *DELETE*.

JSON (*JavaScript Object Notation*) menjadi format komunikasi standar dalam REST API karena kesederhanaan dan kompatibilitas lintas platform (K. et al., 2022). JSON banyak digunakan dalam arsitektur mikro-layanan karena memungkinkan pengembangan aplikasi yang fleksibel dan cepat (Habib, 2019).

#### 2.1.2 Node.js

Node.js adalah lingkungan *runtime* JavaScript yang memungkinkan pengembang menulis kode sisi server menggunakan JavaScript (Gurusamy, 2020). Dibangun di atas mesin V8 milik Google Chrome, Node.js mampu menangani operasi I/O non-blocking secara efisien. Platform ini menggunakan model berbasis peristiwa dengan mekanisme *callback*, sehingga mampu menangani ribuan koneksi secara bersamaan tanpa memerlukan *threading* kompleks (Yee, 2019).

Node.js banyak digunakan oleh perusahaan besar seperti PayPal, LinkedIn, Medium, dan Netflix karena performanya yang tinggi (Gurusamy, 2020).

### **2.1.3 Go (Golang)**

Go adalah bahasa pemrograman sumber terbuka yang dikompilasi dengan performa tinggi dan sintaks ringkas. Go mendukung pemrograman konkuren melalui *goroutines* dan *channel*, serta memiliki sistem tipe yang kuat dan modularitas berbasis paket. Selain cepat dalam kompilasi, Go menyediakan *garbage collection* dan refleksi sehingga efektif untuk membangun layanan backend berskala besar (Authors, 2023).

### **2.1.4 React.js**

React.js merupakan pustaka JavaScript untuk membangun antarmuka pengguna berbasis komponen yang modular dan efisien (Annaram, 2024). Salah satu keunggulan React adalah penggunaan *Virtual DOM*, yang memungkinkan pembaruan tampilan secara cepat tanpa merender ulang seluruh halaman. Hal ini menjadikan React cocok untuk aplikasi web yang interaktif dan responsif.

### **2.1.5 JavaScript**

JavaScript adalah bahasa pemrograman tingkat tinggi yang digunakan untuk membuat halaman web menjadi interaktif (W. et al., 2022). Bahasa ini memungkinkan pembuatan komponen seperti validasi formulir, galeri gambar responsif, dan animasi. Selain berjalan di sisi klien melalui browser, JavaScript juga dapat dijalankan di sisi server menggunakan lingkungan seperti Node.js (Shoikhedbrod, 2023).

### **2.1.6 MongoDB**

MongoDB adalah basis data NoSQL berbasis dokumen yang menyimpan data dalam format BSON/JSON. MongoDB mendukung skalabilitas horizontal dan fleksibilitas skema, sehingga cocok untuk aplikasi yang berkembang pesat. Namun, fleksibilitas ini memerlukan perencanaan data yang baik agar tidak menimbulkan inkonsistensi (MongoDB, 2023).

### **2.1.7 DaisyUI**

DaisyUI adalah *component library* berbasis Tailwind CSS yang menyediakan berbagai komponen antarmuka siap pakai seperti tombol, formulir, dan navigasi. DaisyUI mendukung berbagai tema serta mempermudah pengembangan UI modern tanpa menambah ukuran file CSS secara signifikan (DaisyUI, 2023).

### **2.1.8 Axios**

Axios adalah *HTTP client library* berbasis JavaScript untuk melakukan permintaan HTTP seperti *GET*, *POST*, *PUT*, dan *DELETE*. Axios mendukung Promise serta menyediakan fitur seperti *interceptors*, pengaturan *timeout*, dan otomatis konversi JSON (Axios, 2023).

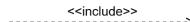
### **2.1.9 UML**

Unified Modeling Language (UML) merupakan bahasa pemodelan standar dalam pengembangan perangkat lunak berorientasi objek. UML digunakan untuk mendokumentasikan, menspesifikasikan, dan membangun sistem secara terstruktur sehingga proses pengembangan dapat dilakukan secara sistematis (Systems, 2021).

## **2.2 Use Case Diagram**

Use Case Diagram merupakan pemodelan yang menggambarkan perilaku sistem informasi yang akan dibuat. Use case digunakan untuk mengidentifikasi fungsi-fungsi dalam sistem serta aktor yang memiliki hak akses terhadap fungsi tersebut (Systems, 2021). Simbol-simbol yang digunakan dalam Use Case Diagram ditampilkan pada Tabel 2.1.

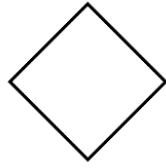
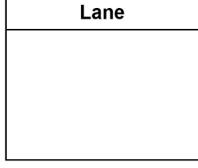
Tabel 2.1: Simbol Use Case Diagram

Gambar	Keterangan
	Use Case menggambarkan fungsionalitas yang disediakan oleh sistem dalam bentuk unit-unit interaksi antara sistem dan aktor.
	Actor merupakan entitas yang berinteraksi dengan sistem untuk menjalankan proses tertentu.
	Asosiasi menunjukkan hubungan interaksi antara aktor dan use case.
	Asosiasi dengan panah terbuka menunjukkan interaksi pasif aktor terhadap sistem.
	Include menunjukkan bahwa suatu use case memanggil use case lain.
	Extend menunjukkan perluasan use case berdasarkan kondisi tertentu.

## 2.3 Diagram Aktivitas (Activity Diagram)

Activity Diagram menggambarkan alur kerja atau aktivitas dari suatu sistem maupun proses bisnis (Systems, 2021). Simbol-simbol yang digunakan dalam Activity Diagram ditampilkan pada Tabel 2.2.

Tabel 2.2: Simbol Activity Diagram

Gambar	Keterangan
	Start Point menandakan awal dari suatu aktivitas atau aliran kerja.
	End Point menandakan akhir dari suatu aktivitas atau proses.
	Activities menggambarkan proses atau kegiatan bisnis yang dilakukan dalam sistem.
	Fork menunjukkan percabangan proses yang berjalan secara paralel dalam sistem.
	Decision Point menunjukkan titik pengambilan keputusan (true/false) dalam alur kerja.
	Swimlane menunjukkan pembagian tanggung jawab aktor dalam suatu proses.

## 2.4 Diagram Urutan (Sequence Diagram)

Sequence Diagram menggambarkan interaksi antar objek dalam suatu use case dengan memperlihatkan waktu hidup objek serta pesan yang dikirim dan diterima (Systems, 2021). Simbol-simbol yang digunakan dalam Sequence Diagram ditampilkan pada Tabel 2.3.

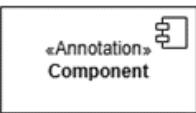
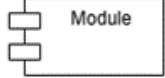
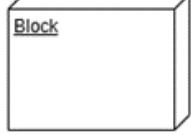
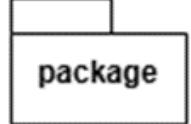
Tabel 2.3: Simbol Sequence Diagram

Gambar	Keterangan
	Entity Class merepresentasikan entitas utama dalam sistem dan menjadi dasar struktur basis data.
	Boundary Class berfungsi sebagai antarmuka antara aktor dan sistem.
	Control Class bertanggung jawab mengatur logika aplikasi dan koordinasi antar objek.
	Message merepresentasikan komunikasi antar objek dalam sistem.
	Recursive menunjukkan objek mengirim pesan kepada dirinya sendiri.
	Activation menunjukkan durasi eksekusi suatu operasi pada objek.
	20

## 2.5 Deployment Diagram

Deployment Diagram menunjukkan bagaimana dan di mana sistem diterapkan, termasuk arsitektur eksekusi sistem. Perangkat keras, prosesor, dan lingkungan eksekusi perangkat lunak direpresentasikan sebagai *node*. Diagram ini juga dapat menampilkan relasi *deployment* dan *manifest* antar artefak serta node (Systems, 2021).

Tabel 2.4: Simbol Deployment Diagram

Simbol	Keterangan
	Komponen merupakan bagian modular dari sistem yang perlakunya ditentukan oleh antarmuka yang disediakan dan dibutuhkan.
	Komponen perangkat lunak yang merepresentasikan unit dengan fungsi tertentu dalam struktur sistem.
	Node merepresentasikan perangkat keras atau perangkat lunak eksternal dalam sistem.
	Package adalah wadah untuk mengelompokkan elemen-elemen sistem dalam struktur terorganisir.
	Kebergantungan menunjukkan hubungan antar node, dengan arah panah menuju node yang digunakan.
	Relasi menunjukkan hubungan antar node dalam menjalankan sistem.

## **BAB III**

### **METODE PENELITIAN**

# Bab 3

## Analisis dan Perancangan

### 3.1 Analisis Sistem pada Aplikasi

Dalam pengembangan aplikasi, analisis terhadap sistem yang berjalan diperlukan untuk memahami alur kerja secara menyeluruh sebagai dasar usulan pengembangan. Analisis dilakukan dengan merujuk pada alur bisnis atau urutan operasional sistem yang digambarkan melalui use case diagram, activity diagram, sequence diagram, dan deployment diagram. Pemahaman yang jelas terhadap alur bisnis akan mendukung pengembangan sistem yang lebih efektif.

#### 3.1.1 Analisis Sistem yang Sedang Berjalan

Sistem pada aplikasi MedisLink dimulai ketika pengguna membutuhkan alat bantu medis. Pengguna melakukan registrasi dan login, kemudian memilih alat berdasarkan kategori yang tersedia. Sistem menampilkan informasi ketersediaan alat untuk menentukan apakah alat dapat dipinjam.

Jika alat tidak tersedia, pengguna dapat menunggu atau memilih alternatif lain. Jika tersedia, pengguna mengisi data peminjaman sesuai ketentuan. Setelah peminjaman berhasil, alat digunakan selama periode tertentu. Sistem mengirimkan notifikasi pengingat sebelum batas waktu pengembalian.

Apabila alat dikembalikan tepat waktu, proses dinyatakan selesai dan data dicatat. Jika terjadi keterlambatan, status diperbarui hingga alat dikembalikan. Proses berakhir ketika alat siap dipinjam kembali oleh pengguna lain.

## **3.2 Analisis Sistem yang Akan Dibangun**

Sistem yang akan dibangun merupakan pengembangan dari sistem sebelumnya dengan penambahan fitur sesuai hasil analisis kebutuhan.

### **3.2.1 Alur Sistem yang Akan Dibangun**

Sistem dimulai dengan registrasi dan login pengguna. Setelah berhasil masuk, pengguna diarahkan ke dashboard untuk melihat informasi akun dan daftar alat bantu medis. Pengguna dapat memilih alat berdasarkan kategori, melihat detail, dan mengajukan peminjaman secara online.

Sistem memproses permintaan dengan memeriksa ketersediaan dan menca-tat transaksi. Selama masa peminjaman, sistem mengirim notifikasi pengingat sesuai jadwal. Pengguna dapat melihat status serta riwayat peminjaman.

Admin bertugas mengelola data pengguna, alat, kategori, serta memantau proses peminjaman dan pengembalian. Proses berakhir saat alat dikembalikan dan status dinyatakan selesai.

### **3.2.2 Kebutuhan Fungsional**

Berikut kebutuhan fungsional sistem:

1. Aplikasi berbasis website yang dapat diakses melalui internet.
2. Fitur pendaftaran dan login pengguna.
3. Menampilkan daftar alat bantu medis berdasarkan kategori.
4. Menampilkan detail alat sebelum peminjaman.
5. Pengajuan peminjaman secara online.
6. Pencatatan dan penyimpanan transaksi peminjaman.
7. Notifikasi pengingat pengembalian alat.
8. Riwayat peminjaman pengguna.
9. Fitur ulasan dan penilaian setelah peminjaman.
10. Admin dapat mengelola alat, kategori, serta memantau peminjaman dan pengembalian.

### **3.2.3 Kebutuhan Nonfungsional**

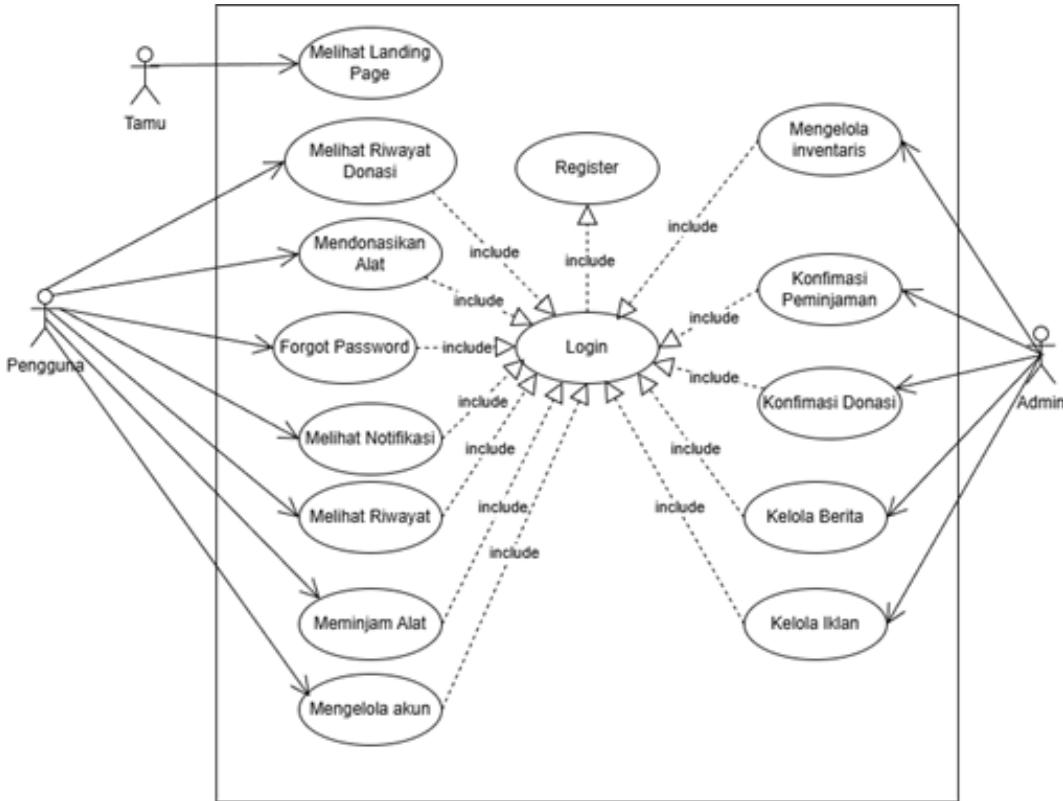
#### **Perangkat Lunak:**

- Sistem Operasi: Windows 11 atau Linux Ubuntu 20.04
- Editor Kode: Visual Studio Code
- Browser: Google Chrome atau Mozilla Firefox
- Database: MongoDB
- Backend: Golang dengan framework Fiber
- Frontend: React.js
- Autentikasi: JSON Web Token (JWT)

#### **Perangkat Keras:**

- Prosesor: Intel Core i5 Gen-10 atau AMD Ryzen 5 5000 series
- RAM: Minimal 8 GB
- Penyimpanan: SSD 256 GB
- Koneksi internet stabil

### 3.3 Use Case Diagram



Gambar 3.1: Use Case Diagram Sistem MedisLink

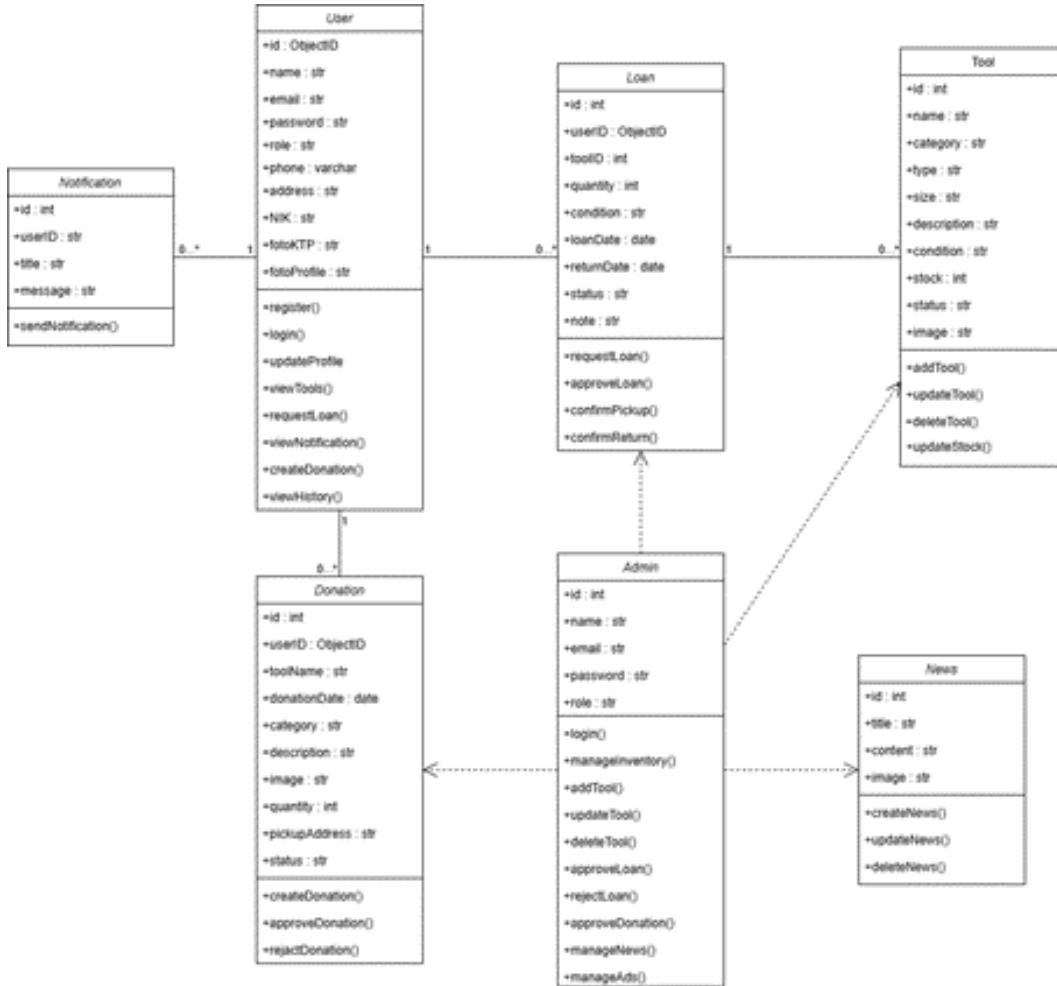
Gambar 3.1 menggambarkan interaksi antara aktor dan sistem MedisLink dalam menjalankan seluruh fitur aplikasi. Terdapat tiga aktor utama, yaitu Tamu, Pengguna, dan Admin, yang masing-masing memiliki peran berbeda dalam sistem.

Tamu hanya dapat melihat halaman *landing page* tanpa perlu login. Pengguna memiliki akses untuk melakukan register, login, melihat riwayat donasi, mendonasikan alat, melakukan *forgot password*, melihat notifikasi, melihat riwayat peminjaman, meminjam alat, serta mengelola akun.

Admin bertanggung jawab mengelola inventaris, melakukan konfirmasi peminjaman dan donasi, serta mengelola berita dan iklan.

Proses *Login* menjadi use case utama karena hampir seluruh fitur lainnya memiliki relasi *include* terhadap login, yang berarti autentikasi diperlukan sebelum fitur dapat diakses. Diagram ini menunjukkan sistem dirancang dengan kontrol akses yang terstruktur sesuai peran masing-masing aktor.

## 3.4 Class Diagram



Gambar 3.2: Class Diagram Sistem MedisLink

Gambar 3.2 menunjukkan struktur class utama beserta hubungan antar class dalam sistem MedisLink.

Class *User* berperan sebagai entitas utama yang dapat melakukan registrasi, login, melihat alat, mengajukan peminjaman, melakukan donasi, serta menerima notifikasi. Class *Loan* merepresentasikan proses peminjaman alat medis yang menghubungkan *User* dengan class *Tool*, yang menyimpan data alat seperti kategori, kondisi, stok, dan status.

Class *Donation* mencatat proses donasi alat medis yang dilakukan oleh user, termasuk informasi alat dan status donasi. Class *Admin* bertanggung jawab dalam pengelolaan sistem seperti mengelola inventaris, menyetujui atau menolak peminjaman dan donasi, serta mengelola konten berita.

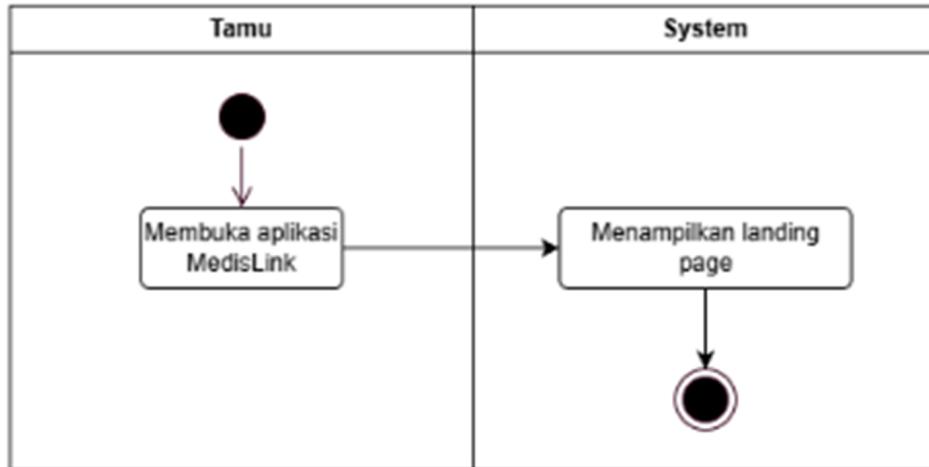
Class *Notification* digunakan untuk mengirim informasi terkait status peminjaman dan donasi kepada user, sedangkan class *News* berfungsi untuk mengelola dan menampilkan informasi pada aplikasi.

Relasi antar class menunjukkan bahwa satu user dapat memiliki banyak peminjaman, donasi, dan notifikasi, sementara admin berperan dalam validasi dan pengelolaan data sistem.

## 3.5 Activity Diagram

Activity diagram digunakan untuk memodelkan perilaku sistem dalam suatu alur proses. Diagram ini menggambarkan interaksi elemen dinamis serta jalur logis berdasarkan kondisi, percabangan, maupun proses paralel.

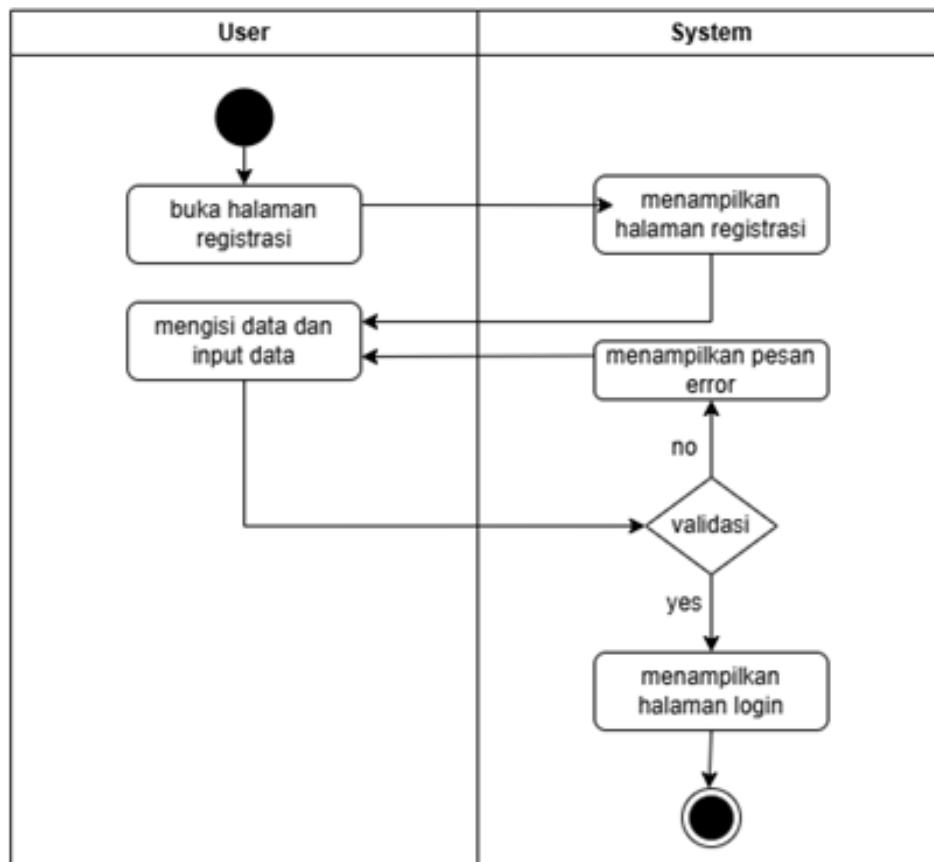
### 3.5.1 Tamu Melihat Landing Page



Gambar 3.3: Activity Diagram Tamu Melihat Landing Page

Gambar 3.3 menunjukkan alur interaksi antara aktor Tamu dan sistem Medis-Link. Proses dimulai ketika tamu membuka aplikasi tanpa melakukan login. Sistem kemudian menampilkan halaman *landing page* sebagai halaman awal aplikasi. Proses berakhir setelah halaman berhasil ditampilkan tanpa akses ke fitur internal sistem.

### 3.5.2 Registrasi

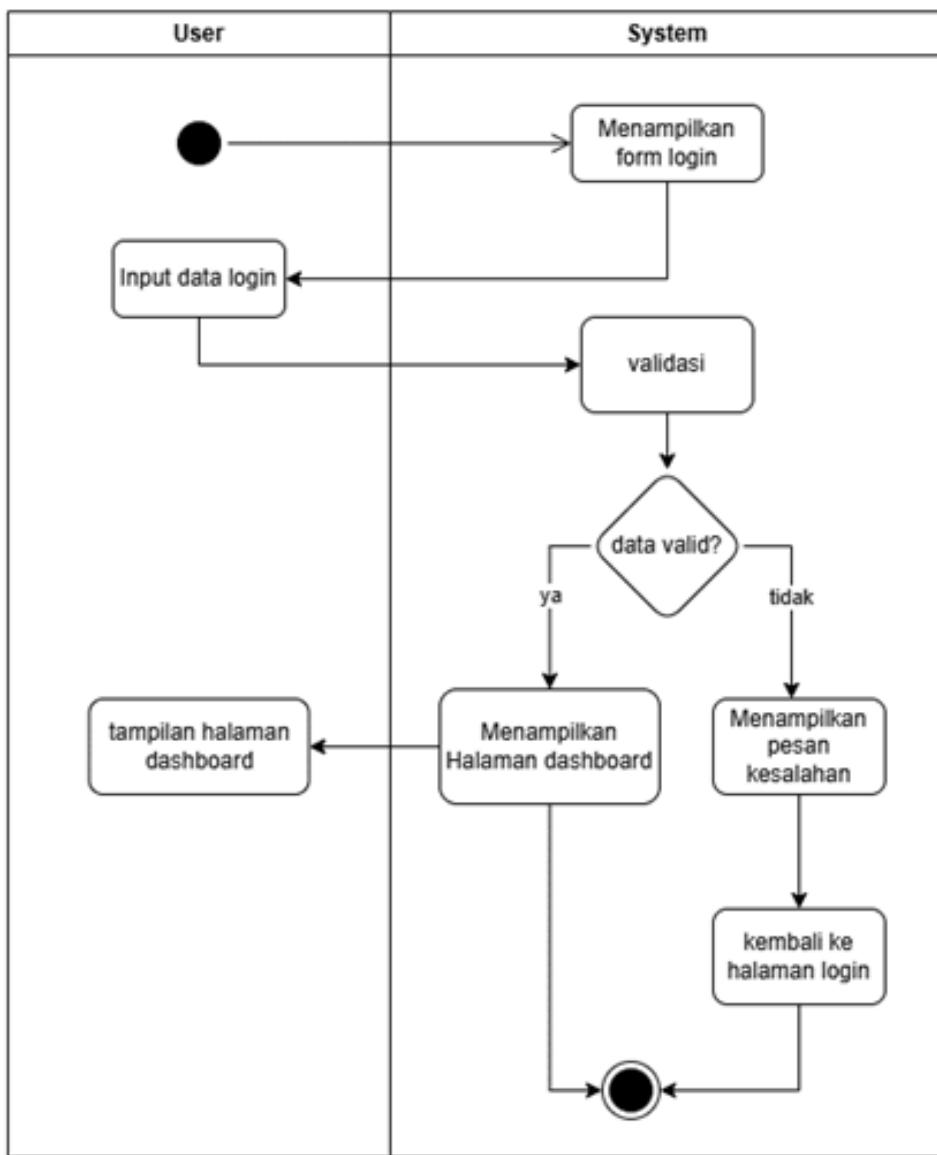


Gambar 3.4: Activity Diagram Registrasi

Gambar 3.4 menggambarkan alur proses pendaftaran pengguna dalam sistem. Proses dimulai ketika pengguna membuka halaman registrasi yang ditampilkan oleh sistem, kemudian pengguna mengisi data yang diperlukan.

Setelah data diinput, sistem melakukan validasi untuk memastikan kelengkapan dan kebenaran data. Jika data tidak valid, sistem menampilkan pesan kesalahan. Jika data valid, sistem mengarahkan pengguna ke halaman login. Diagram ini menunjukkan interaksi antara pengguna dan sistem berdasarkan kondisi yang terjadi.

### 3.5.3 Login User

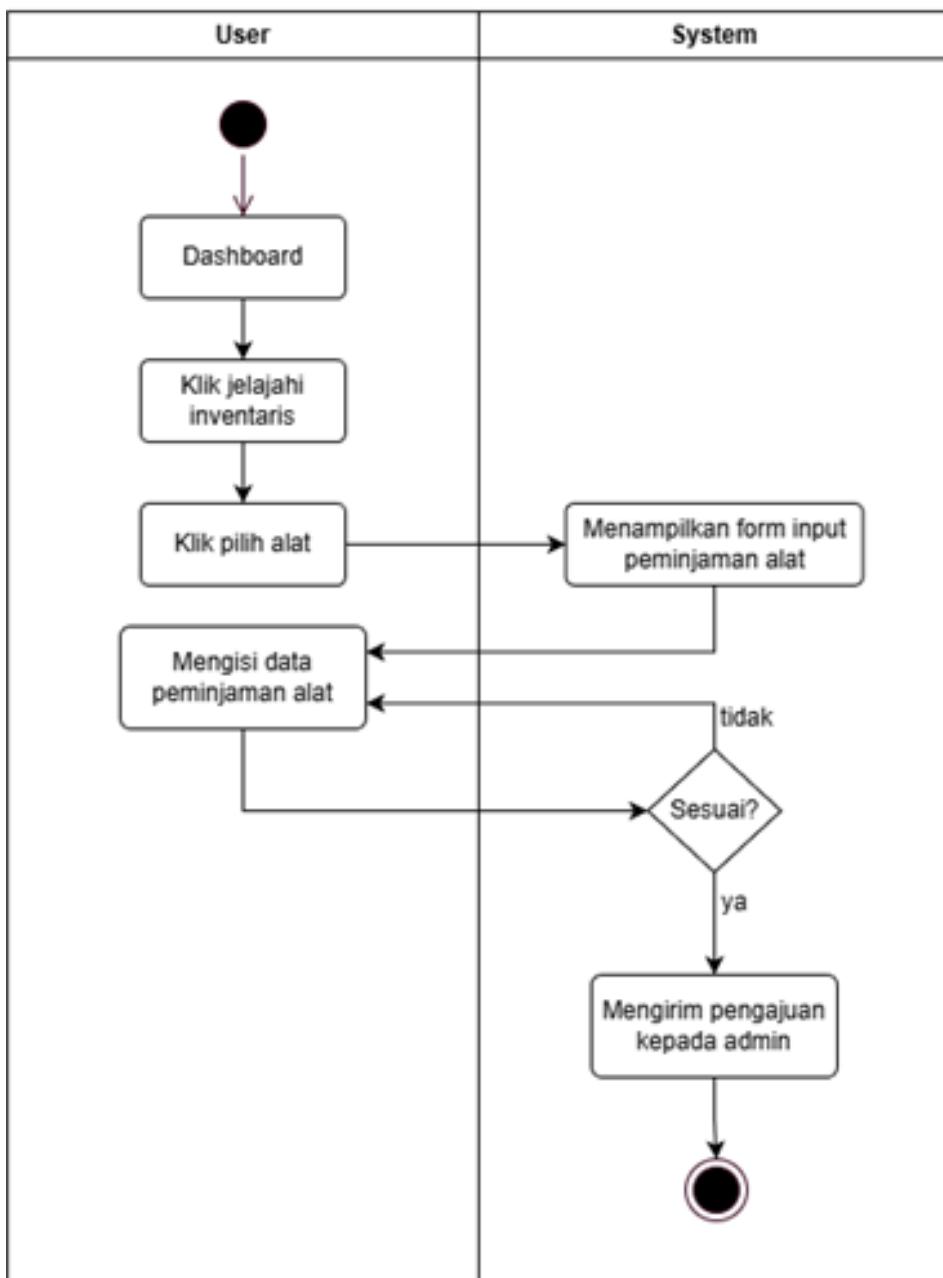


Gambar 3.5: Activity Diagram Login User

Gambar 3.5 menggambarkan proses autentikasi pengguna dalam sistem. Proses dimulai ketika sistem menampilkan form login dan pengguna menginput data yang diperlukan.

Sistem kemudian melakukan validasi terhadap data login. Jika data valid, sistem menampilkan halaman dashboard. Jika data tidak valid, sistem menampilkan pesan kesalahan dan mengarahkan pengguna kembali ke halaman login. Diagram ini menunjukkan interaksi antara pengguna dan sistem berdasarkan hasil validasi.

### 3.5.4 Peminjaman Alat Bantu Medis



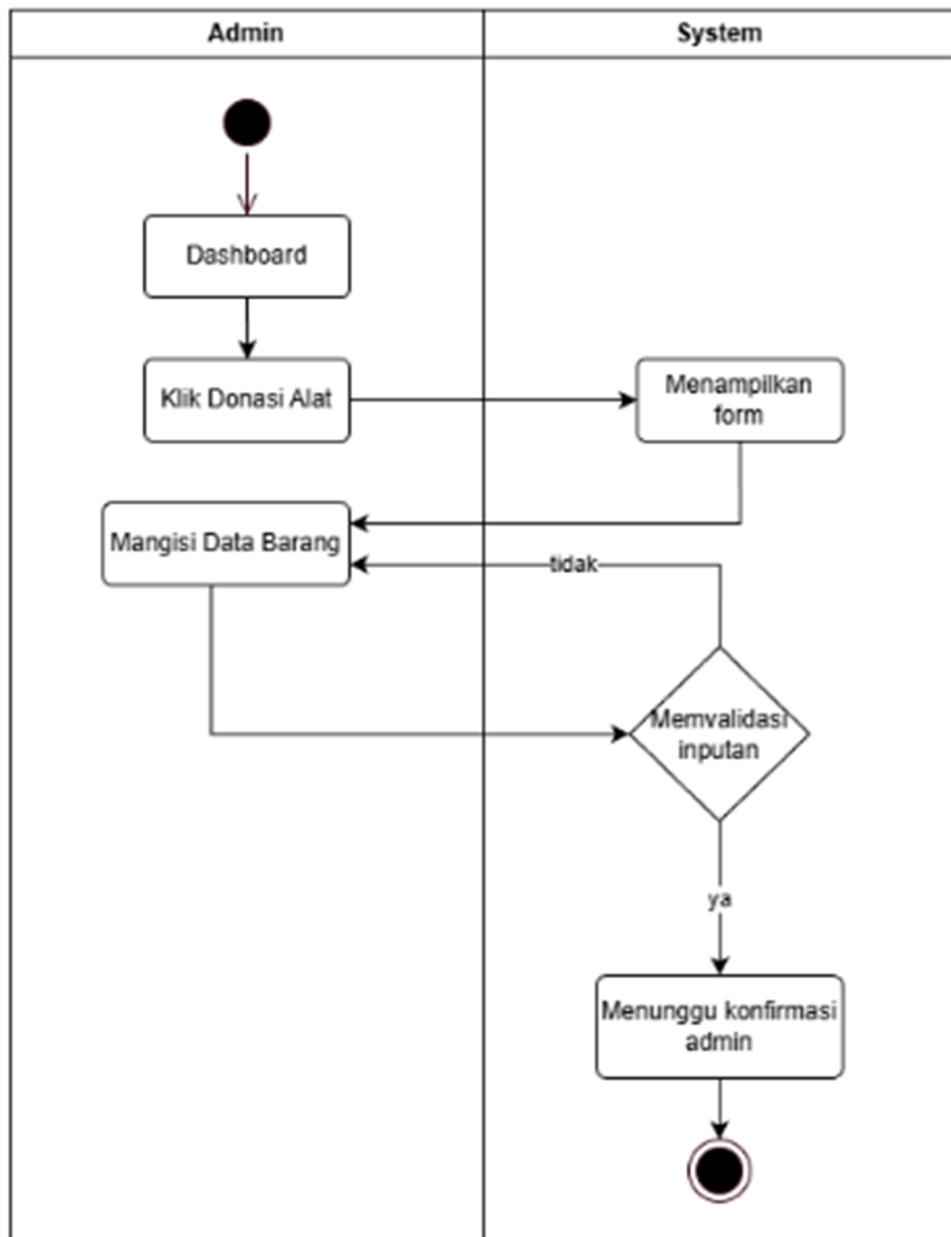
Gambar 3.6: Activity Diagram Peminjaman Alat Bantu Medis

Gambar 3.6 menggambarkan alur proses pengajuan peminjaman alat oleh pengguna dalam sistem. Proses dimulai dari dashboard, di mana pengguna memilih opsi untuk menjelajahi inventaris dan memilih alat yang diinginkan.

Sistem kemudian menampilkan form input peminjaman. Setelah pengguna mengisi data, sistem melakukan validasi untuk memastikan kesesuaian data. Jika data tidak sesuai, pengguna diminta untuk mengisi kembali. Jika data

valid, sistem mengirimkan pengajuan kepada admin sebagai tahap akhir proses peminjaman.

### 3.5.5 Donasi Alat

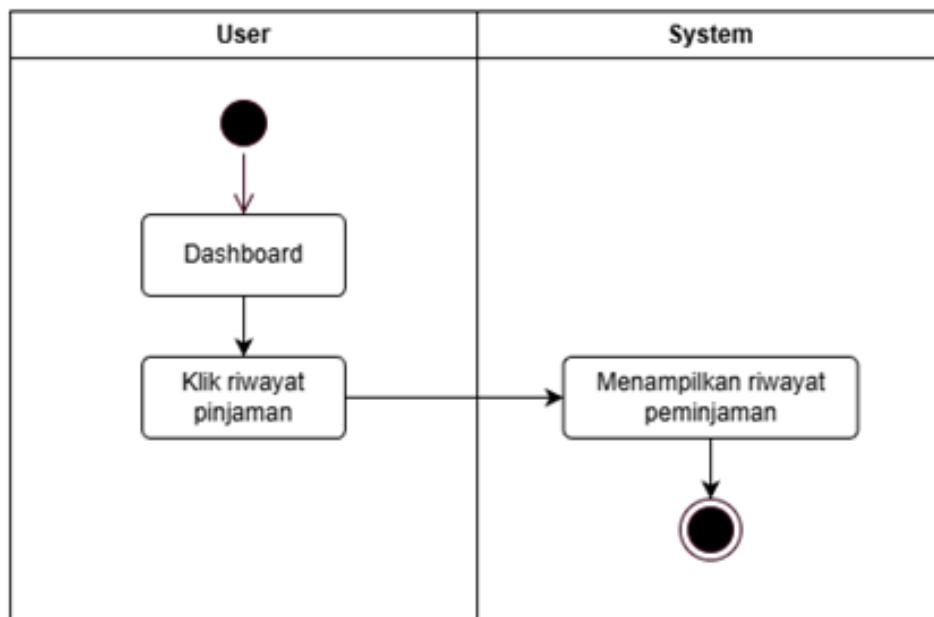


Gambar 3.7: Activity Diagram Donasi Alat

Gambar 3.7 menggambarkan proses donasi alat medis dalam sistem MedisLink. Proses dimulai ketika admin mengakses dashboard dan memilih menu donasi alat.

Sistem kemudian menampilkan form input donasi yang harus diisi. Setelah data diinput, sistem melakukan validasi. Jika data belum sesuai, admin diminta memperbaiki input. Jika data valid, sistem memproses data dan status donasi menunggu konfirmasi hingga proses selesai.

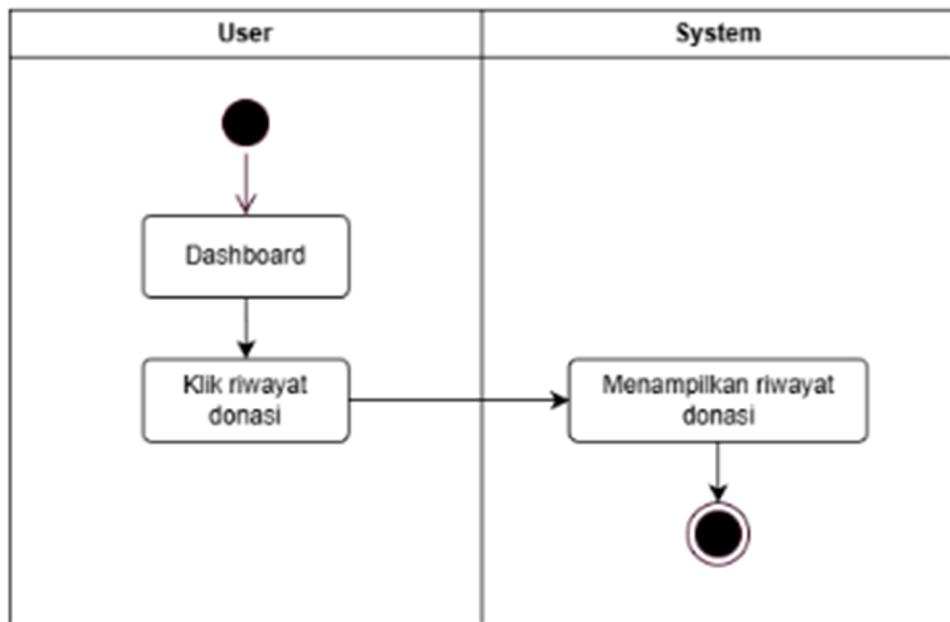
### 3.5.6 Riwayat Peminjaman User



Gambar 3.8: Activity Diagram Riwayat Peminjaman

Gambar 3.8 menggambarkan proses melihat riwayat transaksi peminjaman. Pengguna memulai dari dashboard dan memilih menu riwayat pinjaman. Sistem kemudian menampilkan riwayat peminjaman yang telah dilakukan. Proses berakhir setelah data berhasil ditampilkan.

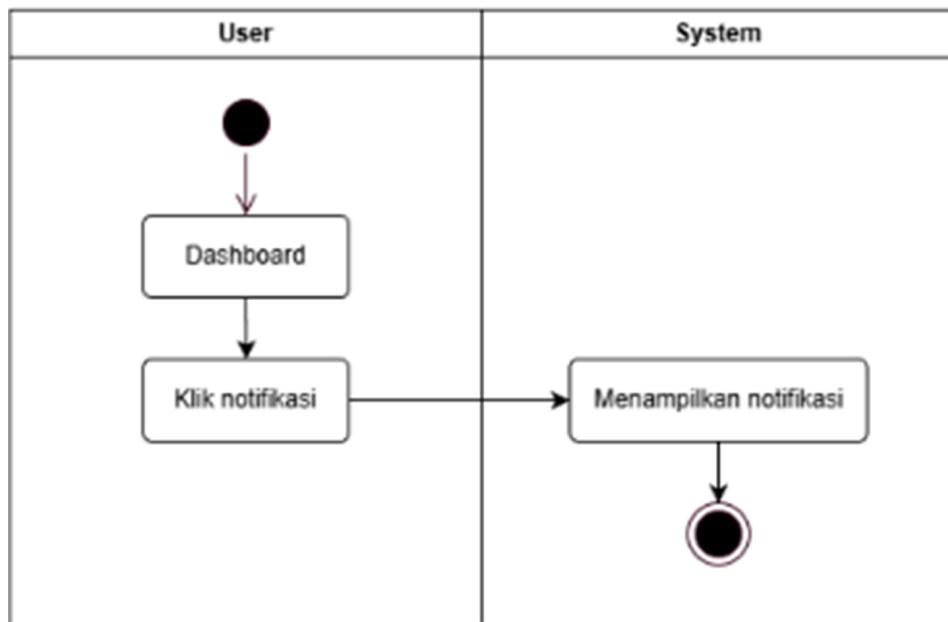
### 3.5.7 Riwayat Donasi User



Gambar 3.9: Activity Diagram Riwayat Donasi User

Gambar 3.9 menggambarkan alur aktivitas pengguna dalam melihat riwayat donasi. Pengguna memilih menu riwayat donasi pada dashboard, kemudian sistem menampilkan daftar riwayat donasi yang pernah dilakukan. Proses berakhir setelah riwayat berhasil ditampilkan.

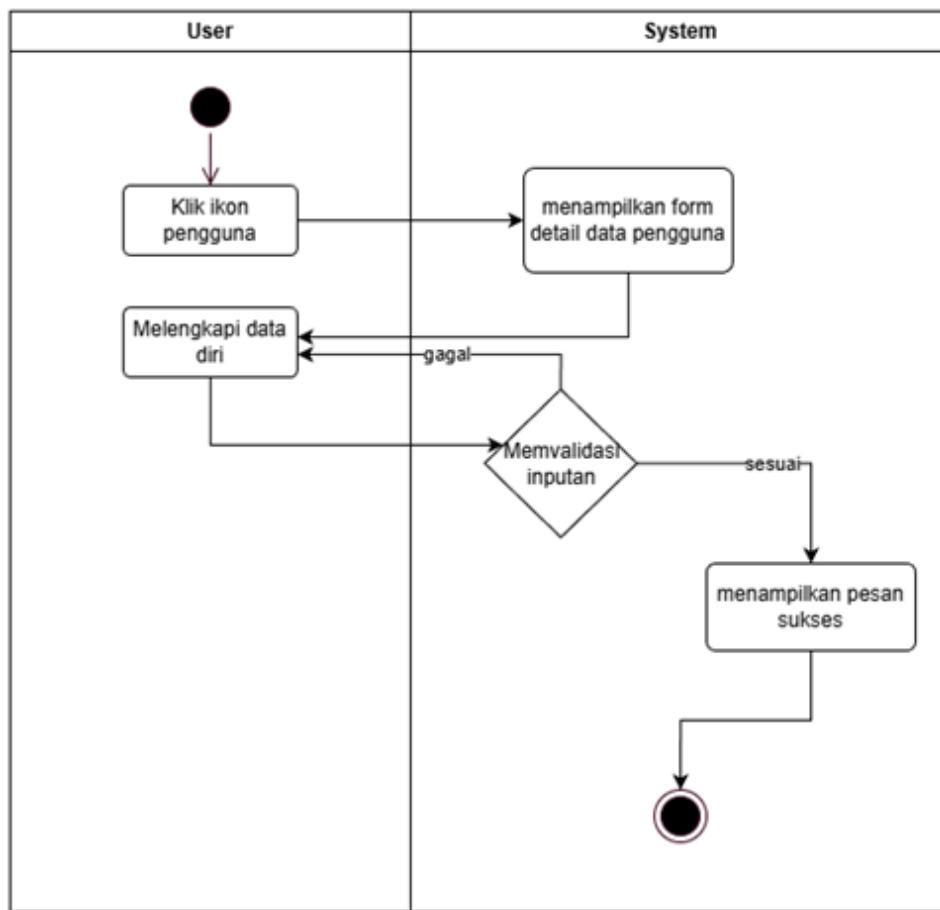
### 3.5.8 User Melihat Notifikasi



Gambar 3.10: Activity Diagram User Melihat Notifikasi

Gambar 3.10 menunjukkan proses ketika user berada di dashboard dan memilih menu notifikasi. Sistem kemudian menampilkan daftar notifikasi terkait aktivitas peminjaman atau donasi. Proses berakhir setelah notifikasi berhasil ditampilkan.

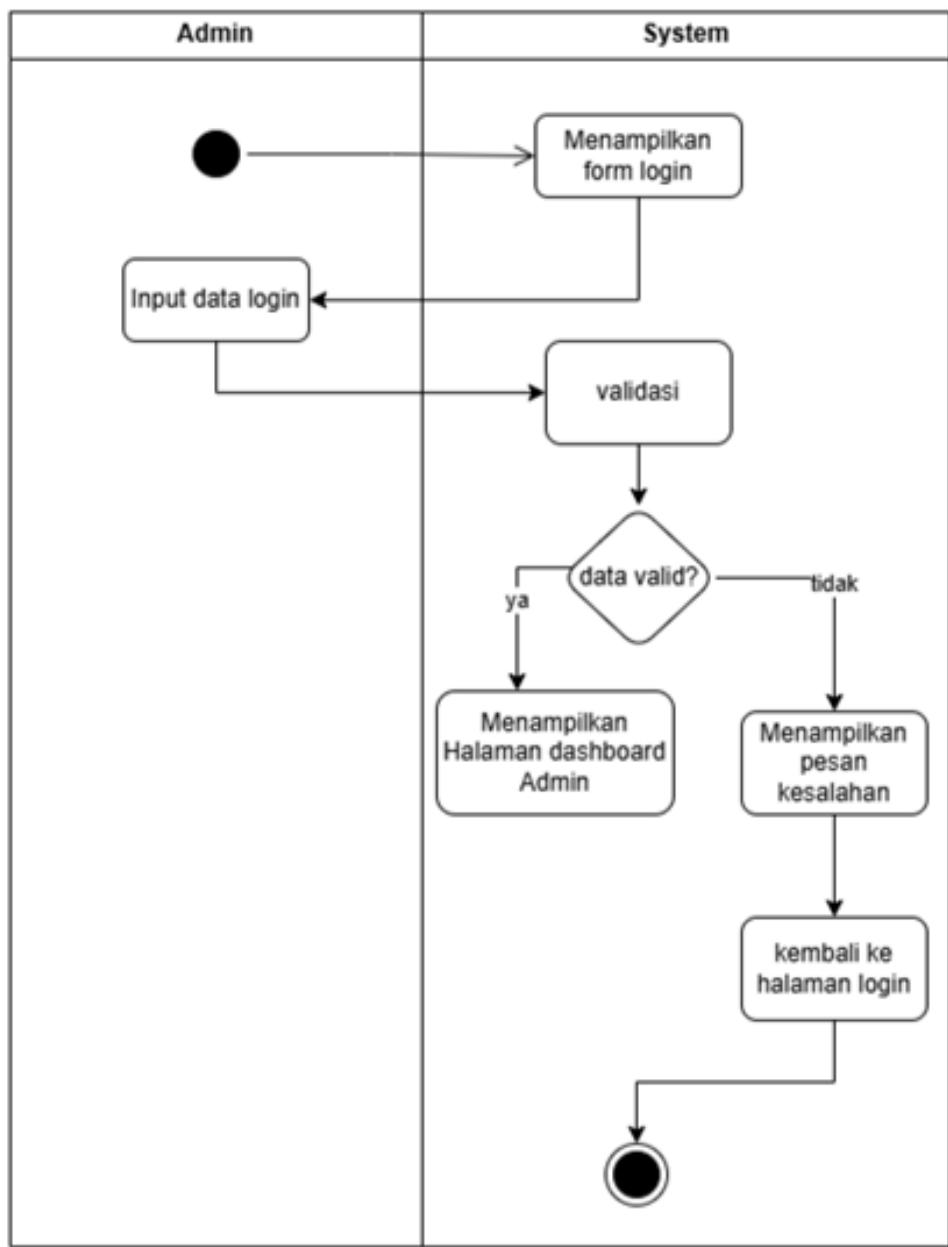
### 3.5.9 Mengelola Akun



Gambar 3.11: Activity Diagram Mengelola Akun

Gambar 3.11 menggambarkan proses pembaruan data pribadi pengguna. Pengguna memilih ikon akun, sistem menampilkan form detail data, kemudian pengguna memperbarui data. Sistem melakukan validasi; jika data tidak valid, pengguna diminta memperbaiki input. Jika valid, sistem menampilkan pesan sukses sebagai tanda pembaruan berhasil.

### 3.5.10 Login Admin

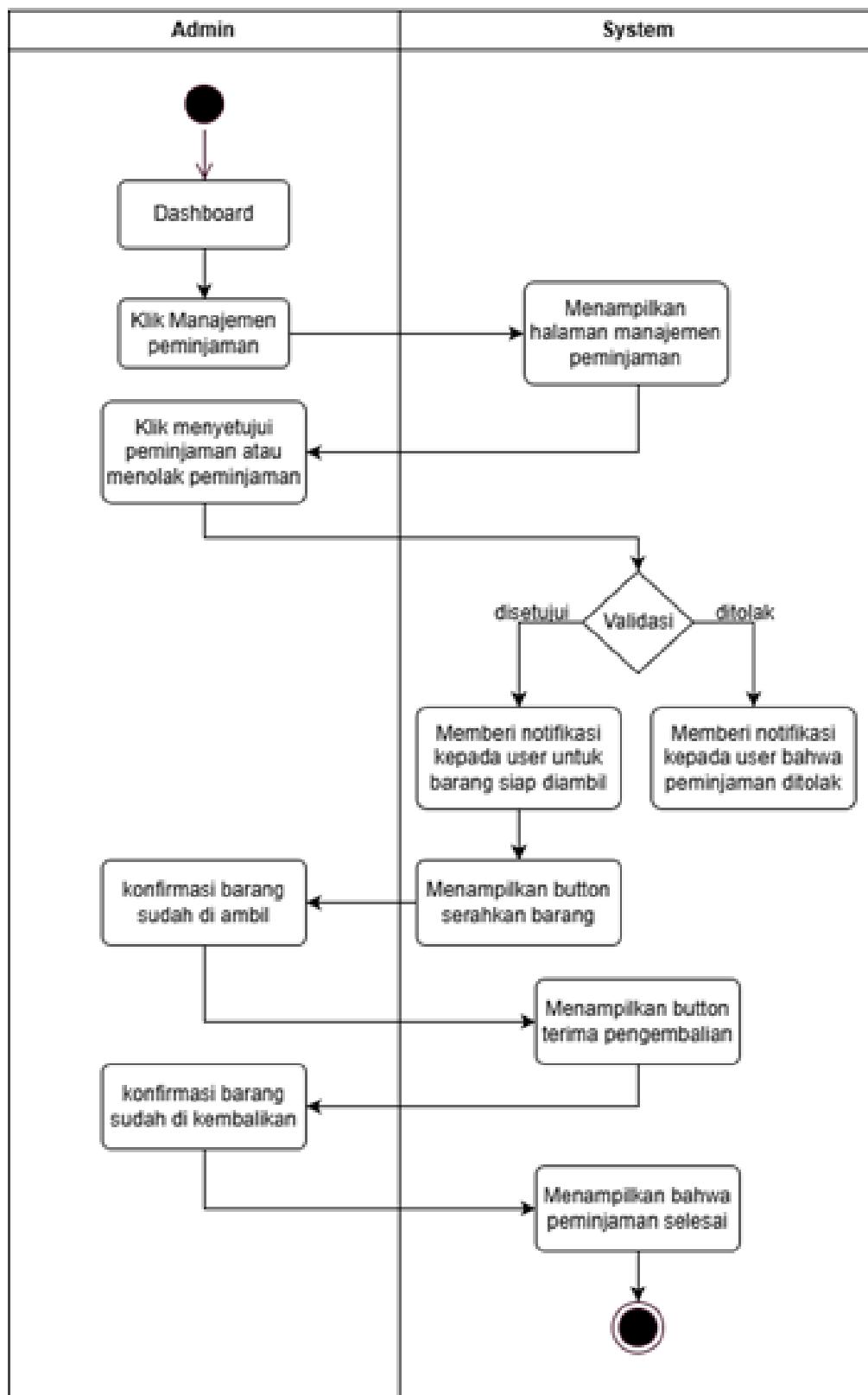


Gambar 3.12: Activity Diagram Login Admin

Gambar 3.12 menggambarkan proses login admin dalam aplikasi MedisLink. Admin menginput data login, kemudian sistem melakukan validasi. Jika data valid, sistem menampilkan halaman dashboard admin. Jika tidak valid, sistem menampilkan pesan kesalahan dan mengarahkan kembali ke halaman login.



### 3.5.11 Admin Mengelola Peminjaman

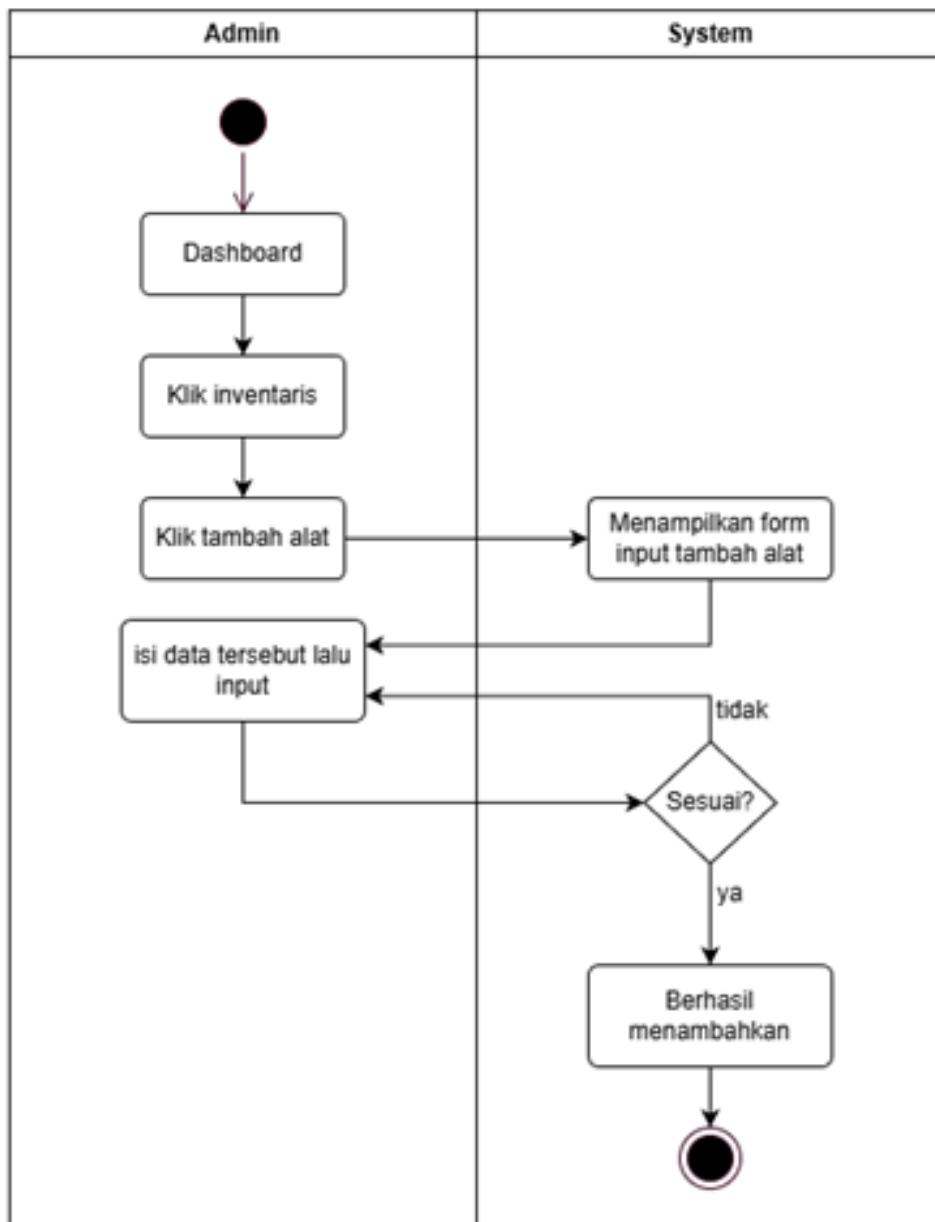


Gambar 3.13: Activity Diagram Admin Mengelola Peminjaman

Gambar 3.13 menggambarkan proses pengelolaan peminjaman oleh admin. Admin memilih menu manajemen peminjaman, kemudian sistem menampilkan daftar peminjaman untuk diverifikasi.

Admin dapat menyetujui atau menolak peminjaman, dan sistem mengirim notifikasi kepada pengguna sesuai keputusan tersebut. Jika disetujui, admin mengonfirmasi penyerahan barang dan pengembalian barang hingga sistem menampilkan status bahwa peminjaman telah selesai.

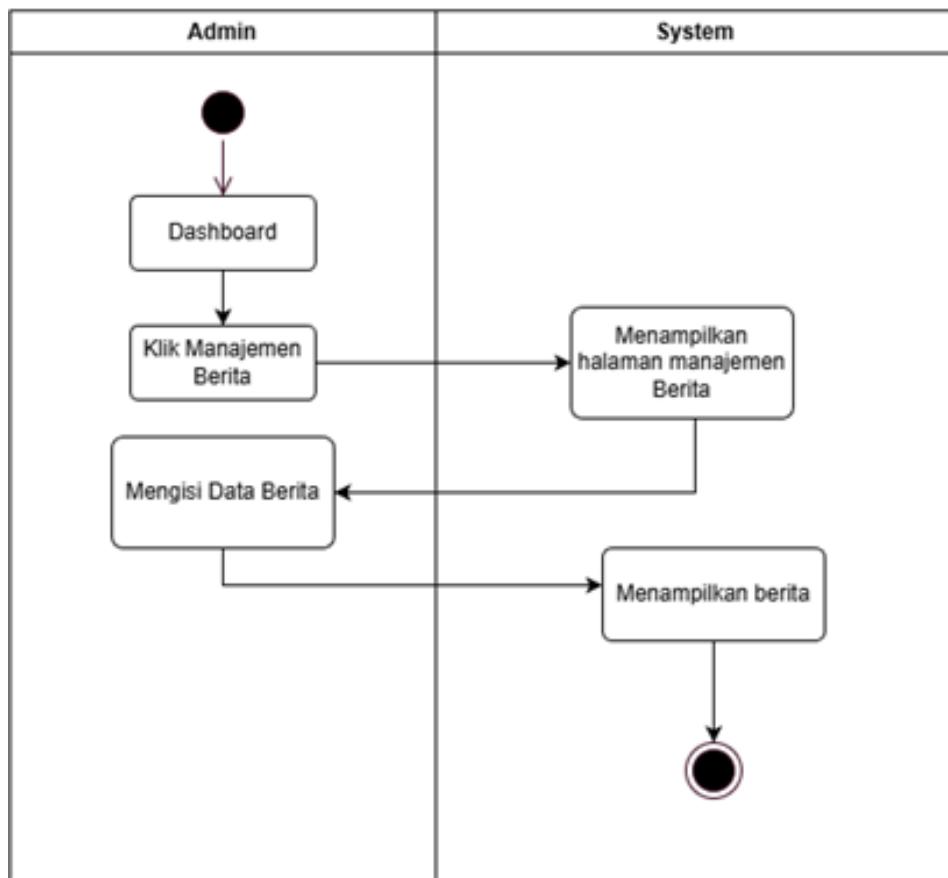
### 3.5.12 Admin Mengelola Inventaris



Gambar 3.14: Activity Diagram Admin Mengelola Inventaris

Gambar 3.14 menggambarkan proses penambahan data alat medis oleh admin. Admin memilih menu inventaris dan menekan tombol tambah alat, kemudian sistem menampilkan form input. Setelah data dimasukkan, sistem melakukan validasi. Jika data tidak sesuai, admin diminta memperbaiki input. Jika valid, sistem menyimpan data dan menampilkan notifikasi bahwa penambahan alat berhasil.

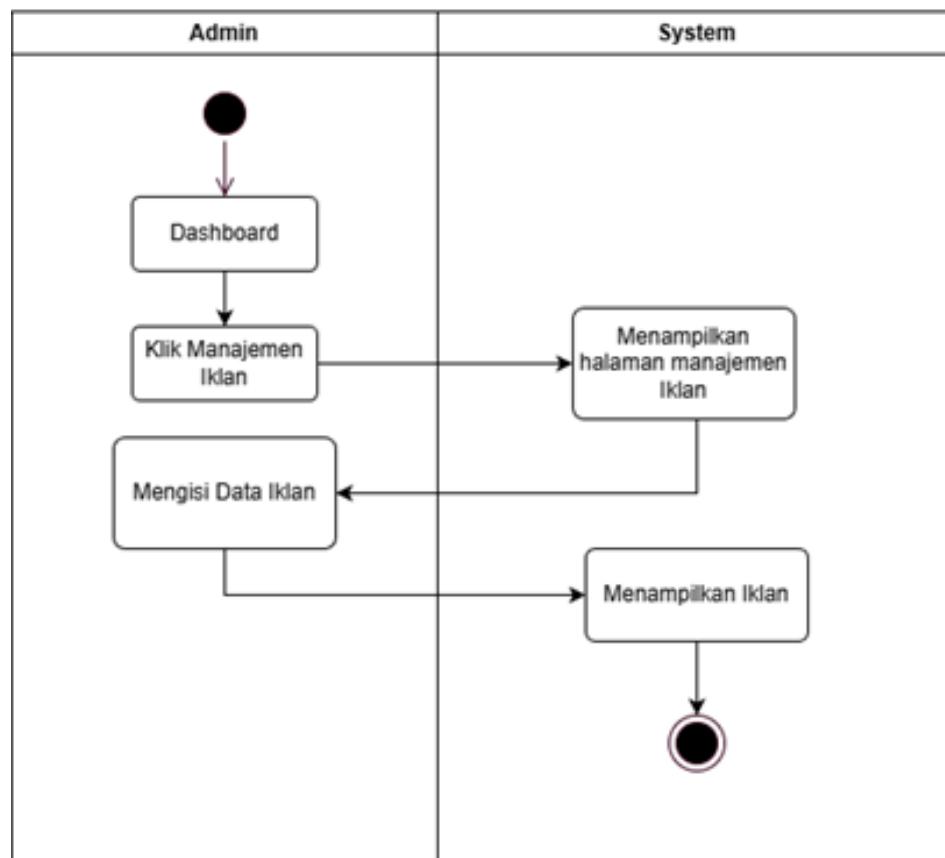
### 3.5.13 Admin Mengelola Berita



Gambar 3.15: Activity Diagram Admin Mengelola Berita

Gambar 3.15 menggambarkan proses pengelolaan berita oleh admin. Admin memilih menu manajemen berita, kemudian sistem menampilkan halaman pengelolaan. Admin mengisi data berita dan sistem menampilkan berita kepada pengguna setelah proses selesai.

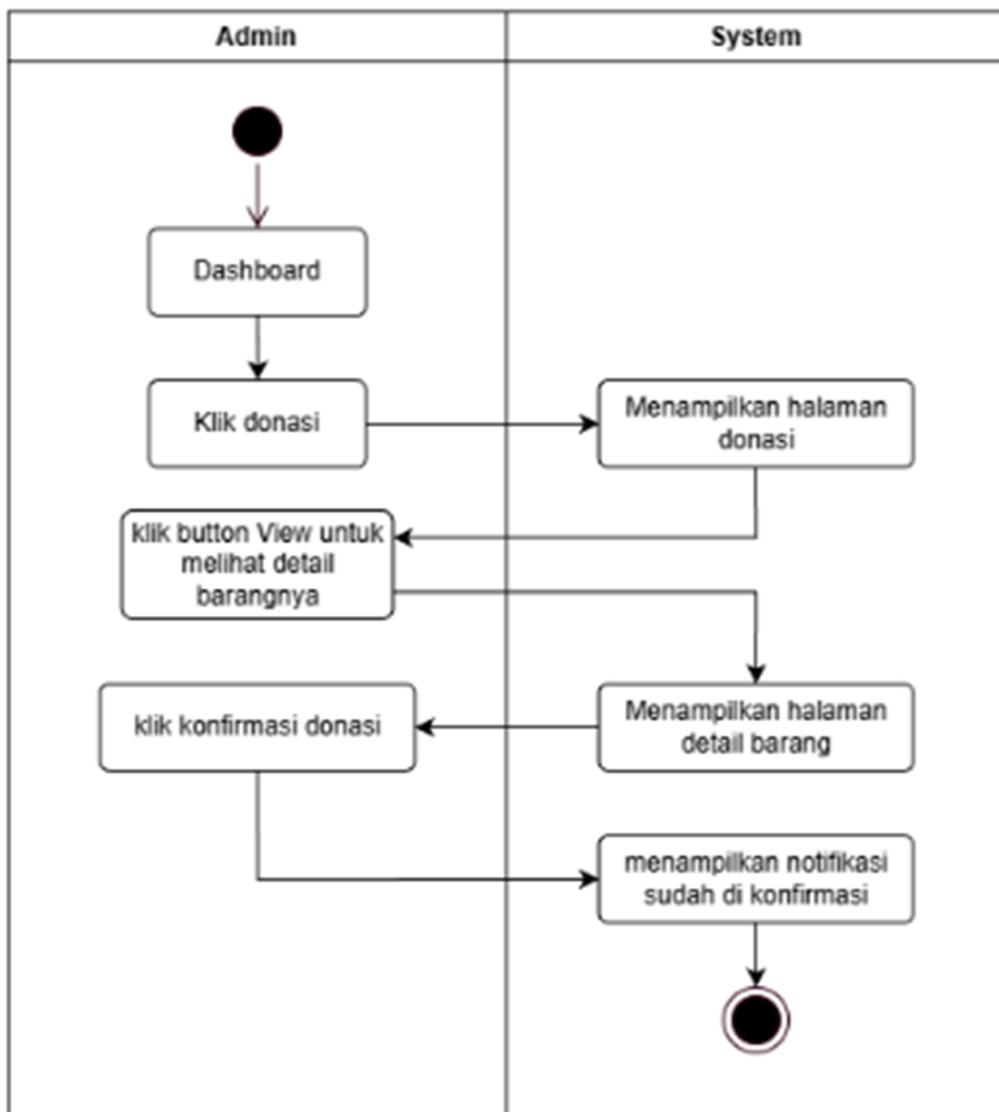
### 3.5.14 Admin Mengelola Iklan



Gambar 3.16: Activity Diagram Admin Mengelola Iklan

Gambar 3.16 menggambarkan proses pengelolaan iklan oleh admin. Admin memilih menu manajemen iklan pada dashboard, kemudian sistem menampilkan halaman pengelolaan iklan. Admin mengisi data iklan yang akan dipublikasikan dan sistem menampilkan iklan setelah proses selesai.

### 3.5.15 Admin Konfirmasi Donasi



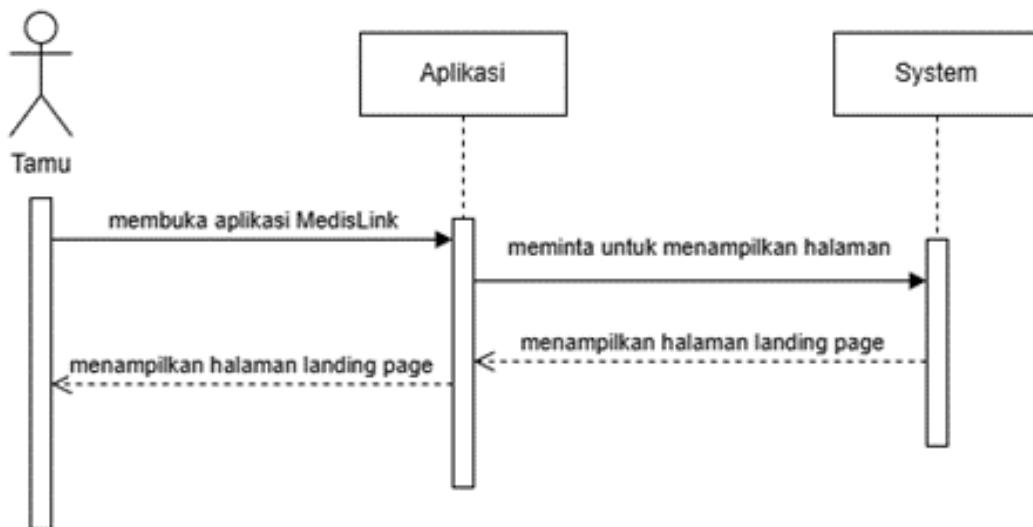
Gambar 3.17: Activity Diagram Admin Konfirmasi Donasi

Gambar 3.17 menggambarkan proses admin dalam mengonfirmasi donasi alat medis. Admin memilih menu donasi, sistem menampilkan daftar donasi, kemudian admin dapat melihat detail barang sebelum melakukan konfirmasi. Setelah donasi dikonfirmasi, sistem menampilkan notifikasi bahwa donasi berhasil dan proses dinyatakan selesai.

## 3.6 Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan interaksi antar objek dalam sistem berdasarkan urutan waktu eksekusi. Diagram ini menunjukkan komunikasi antar objek pada setiap use case yang dijalankan.

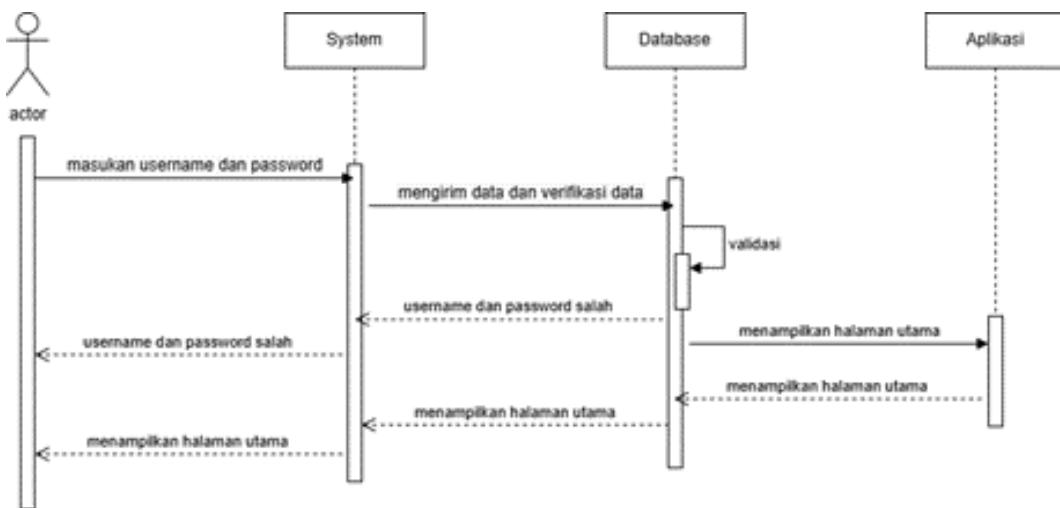
### 3.6.1 Tamu Melihat Landing Page



Gambar 3.18: Sequence Diagram Tamu Melihat Landing Page

Gambar 3.18 menggambarkan interaksi antara Tamu, Aplikasi, dan Sistem saat mengakses MedisLink. Proses dimulai ketika tamu membuka aplikasi, kemudian aplikasi meminta sistem untuk menampilkan halaman awal. Sistem merespons dengan mengirimkan halaman landing page yang kemudian ditampilkan kepada tamu. Proses berakhir setelah landing page berhasil ditampilkan.

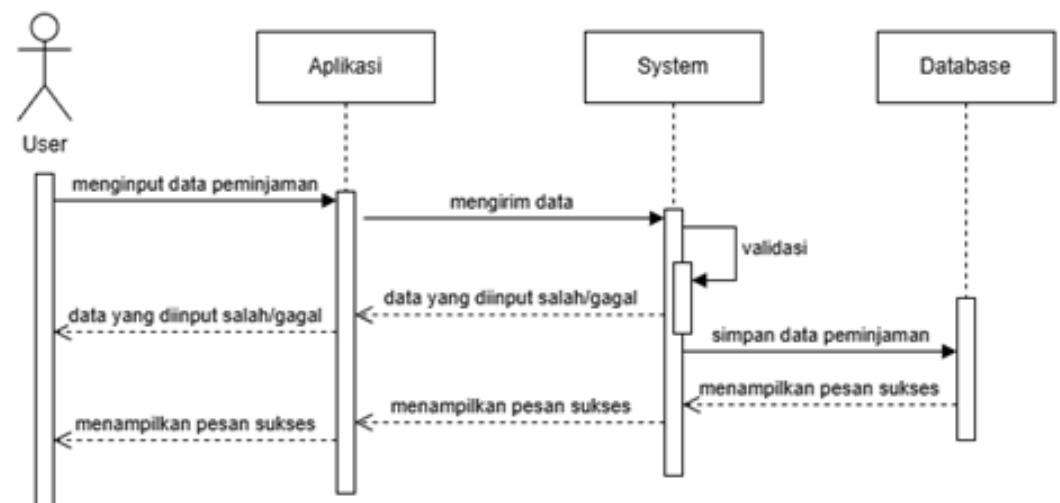
### 3.6.2 Login



Gambar 3.19: Sequence Diagram Login

Gambar 3.19 menggambarkan proses autentikasi pengguna atau admin. Aktor memasukkan username dan password melalui aplikasi, kemudian sistem memverifikasi data ke database. Jika data tidak valid, sistem mengembalikan pesan kesalahan. Jika valid, sistem menampilkan halaman utama sesuai peran pengguna.

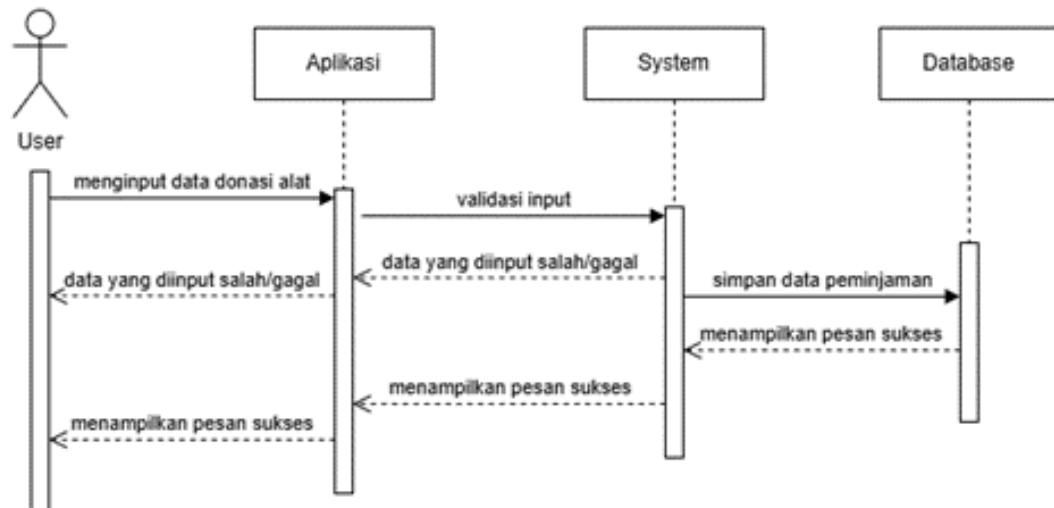
### 3.6.3 Peminjaman Alat Bantu Medis



Gambar 3.20: Sequence Diagram Peminjaman Alat Bantu Medis

Gambar 3.20 menggambarkan proses pengajuan peminjaman. User menginput data melalui aplikasi, kemudian sistem melakukan validasi. Jika gagal, sistem menampilkan pesan kesalahan. Jika berhasil, sistem menyimpan data ke database dan menampilkan pesan sukses kepada user.

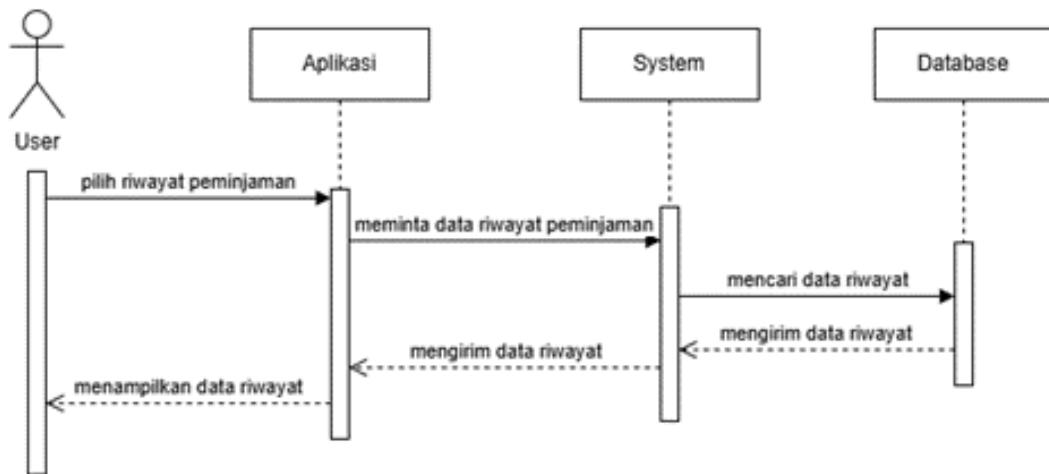
### 3.6.4 User Donasi Alat



Gambar 3.21: Sequence Diagram User Donasi Alat

Gambar 3.21 menggambarkan proses donasi alat oleh user. Data donasi dikirim ke sistem untuk diverifikasi. Jika tidak valid, sistem mengembalikan pesan kesalahan. Jika valid, data disimpan ke database dan sistem menampilkan pesan sukses.

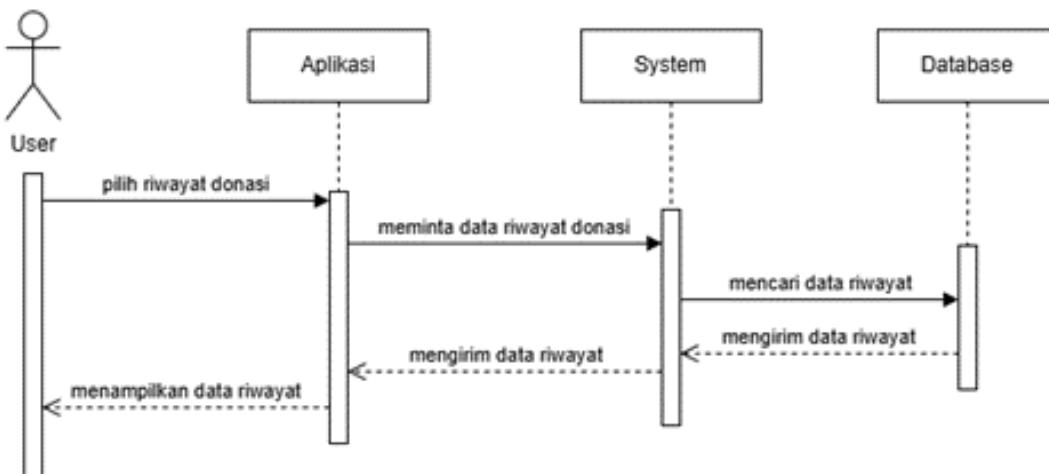
### 3.6.5 Riwayat Peminjaman User



Gambar 3.22: Sequence Diagram Riwayat Peminjaman User

Gambar 3.22 menggambarkan proses user dalam mengakses riwayat peminjaman. Aplikasi meminta data ke sistem, sistem mengambil data dari database, kemudian data dikirim kembali ke aplikasi untuk ditampilkan kepada user.

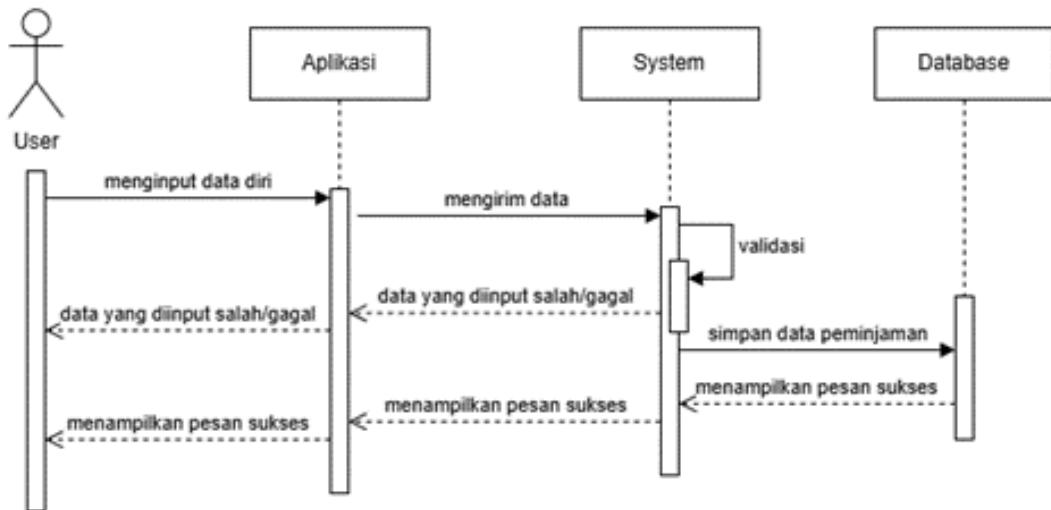
### 3.6.6 Riwayat Donasi User



Gambar 3.23: Sequence Diagram Riwayat Donasi User

Gambar 3.23 menggambarkan proses user dalam melihat riwayat donasi. Sistem mengambil data dari database dan meneruskannya ke aplikasi untuk ditampilkan kepada user.

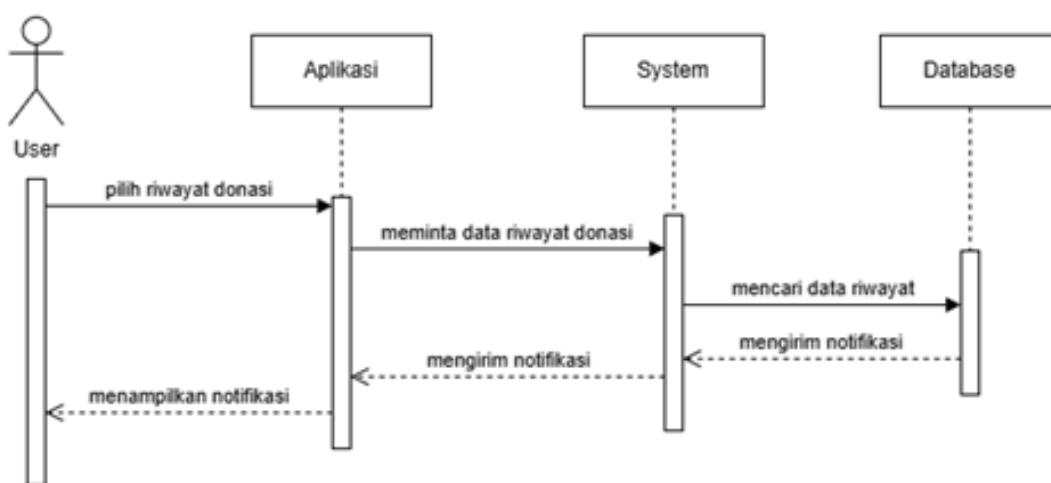
### 3.6.7 Mengelola Akun User



Gambar 3.24: Sequence Diagram Mengelola Akun User

Gambar 3.24 menggambarkan proses pembaruan data pengguna. Data dikirim ke sistem untuk diverifikasi. Jika gagal, sistem menampilkan pesan kesalahan. Jika berhasil, data disimpan ke database dan sistem menampilkan pesan sukses.

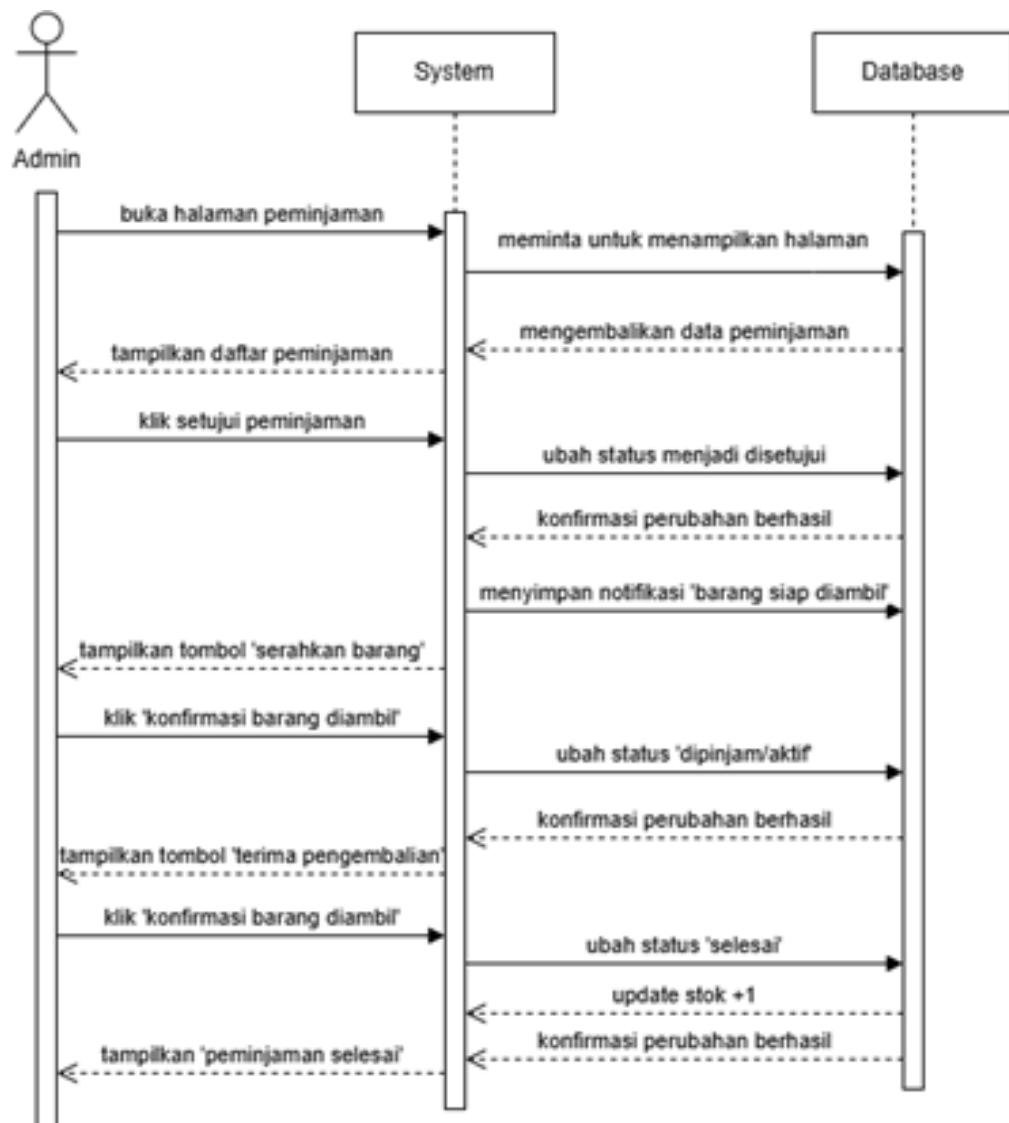
### 3.6.8 User Melihat Notifikasi



Gambar 3.25: Sequence Diagram User Melihat Notifikasi

Gambar 3.25 menggambarkan proses user dalam melihat notifikasi. Aplikasi meminta data notifikasi ke sistem, sistem mengambil data dari database, kemudian data dikirim kembali ke aplikasi untuk ditampilkan kepada user.

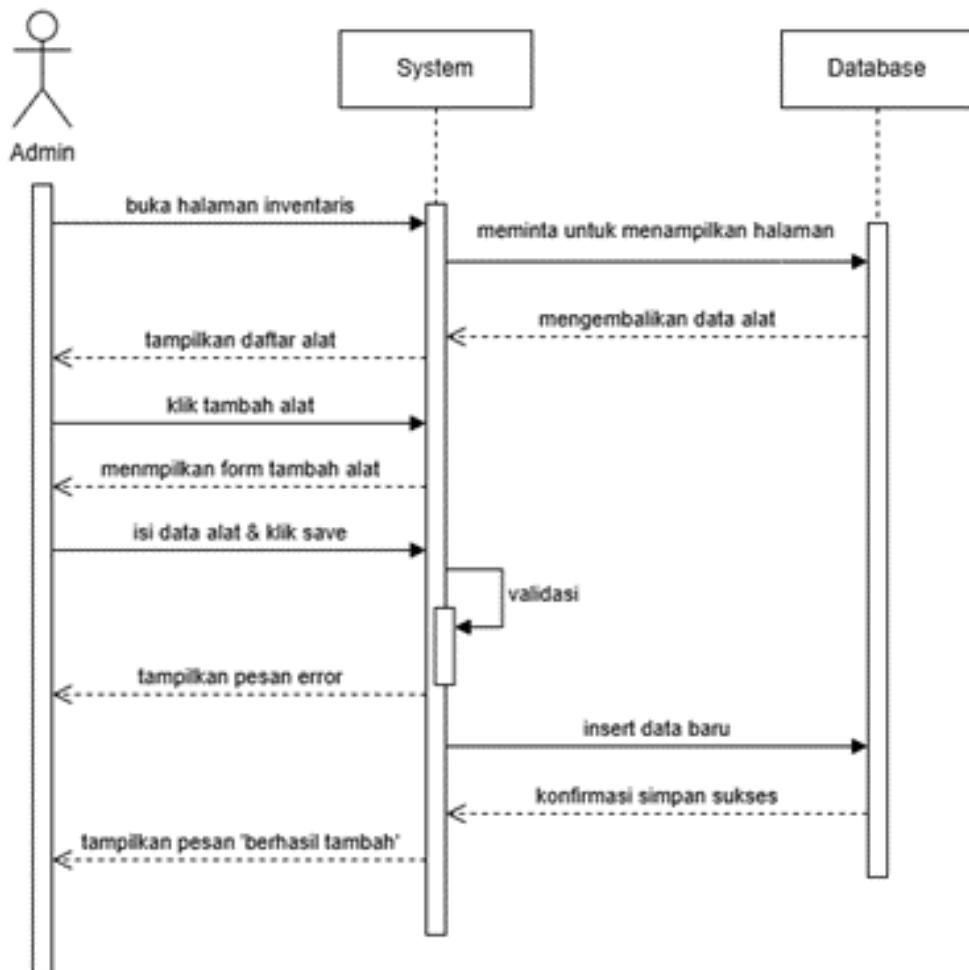
### 3.6.9 Admin Mengelola Peminjaman



Gambar 3.26: Sequence Diagram Admin Mengelola Peminjaman

Gambar 3.26 menunjukkan proses validasi peminjaman oleh admin. Sistem mengambil daftar peminjaman dari database, admin menyetujui atau menolak, kemudian sistem memperbarui status dan mengirim notifikasi. Proses berlanjut hingga konfirmasi pengembalian dan status berubah menjadi selesai serta stok alat diperbarui.

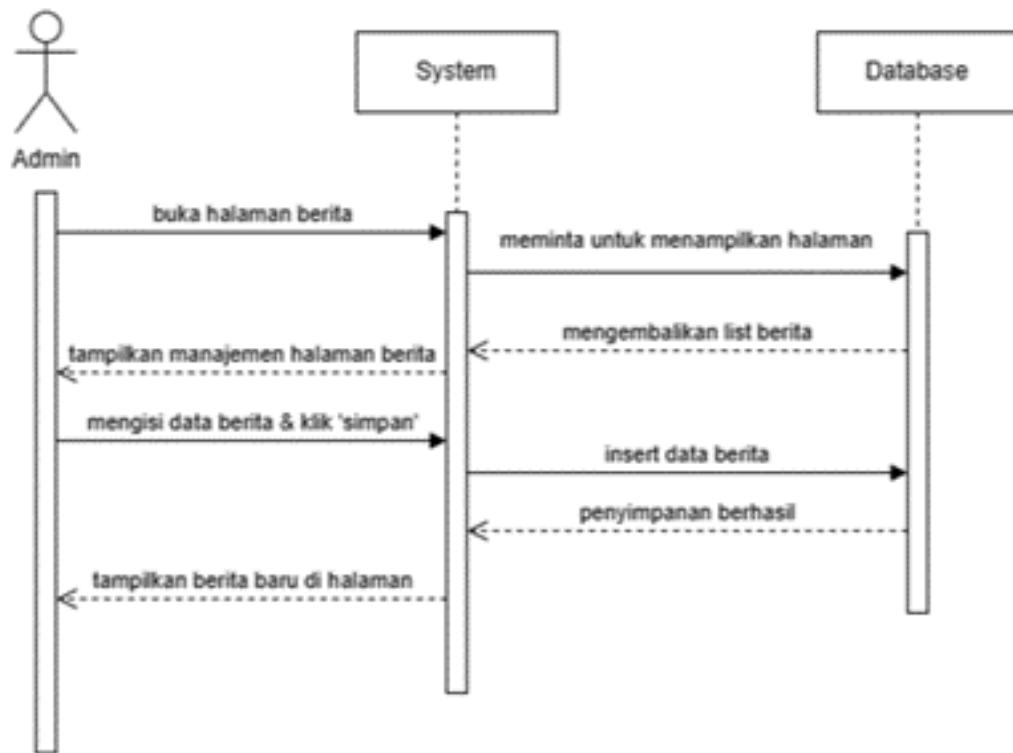
### 3.6.10 Admin Mengelola Inventaris



Gambar 3.27: Sequence Diagram Admin Mengelola Inventaris

Gambar 3.27 menggambarkan proses penambahan alat oleh admin. Sistem menampilkan daftar alat, admin mengisi form tambah alat, kemudian sistem memvalidasi dan menyimpan data ke database. Setelah berhasil, sistem menampilkan notifikasi keberhasilan.

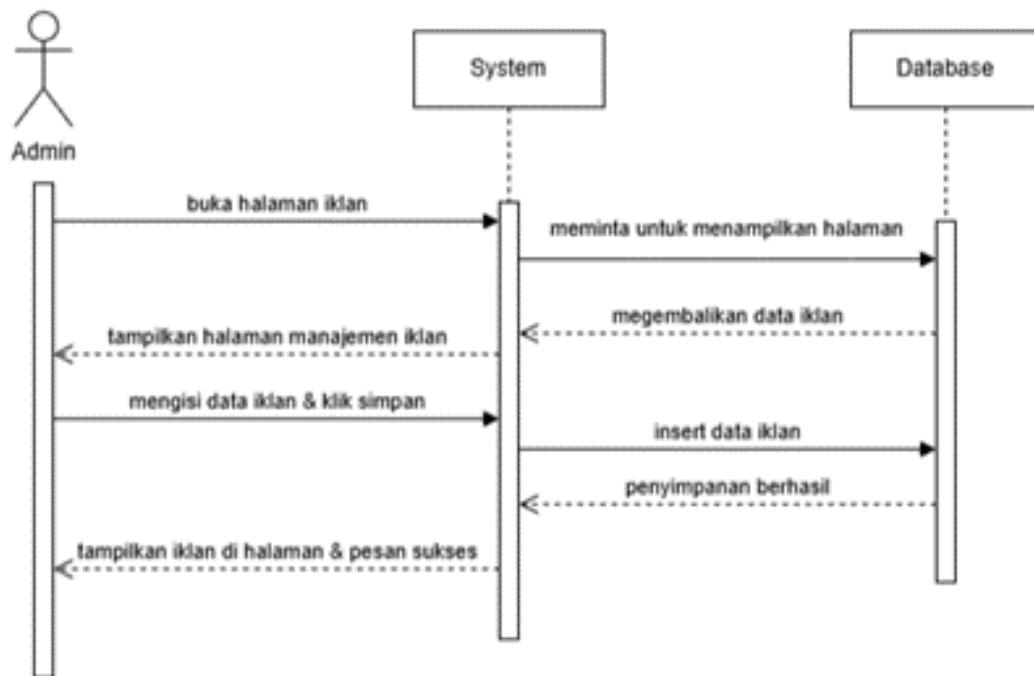
### 3.6.11 Admin Mengelola Berita



Gambar 3.28: Sequence Diagram Admin Mengelola Berita

Gambar 3.28 menunjukkan proses publikasi berita oleh admin. Sistem menampilkan daftar berita dari database, kemudian admin mengisi dan menyimpan berita baru. Setelah database mengonfirmasi penyimpanan berhasil, sistem menampilkan berita tersebut pada halaman manajemen.

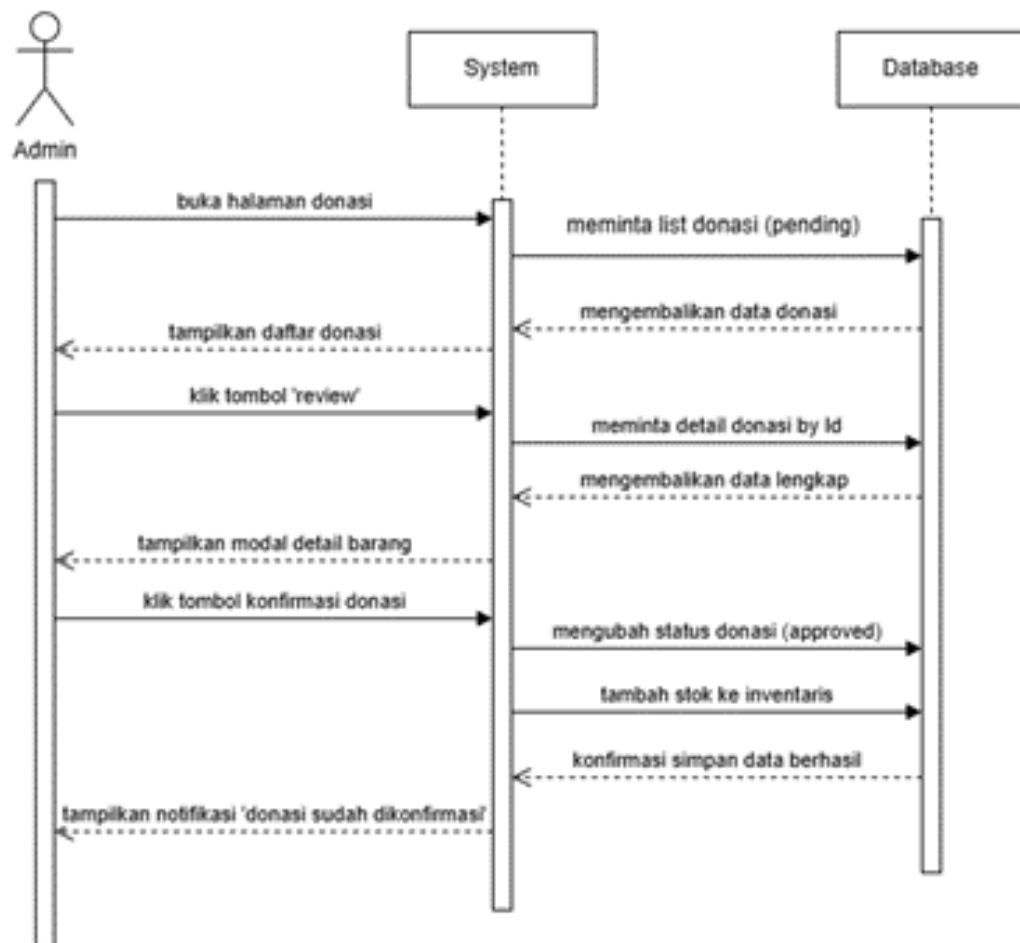
### 3.6.12 Admin Mengelola Iklan



Gambar 3.29: Sequence Diagram Admin Mengelola Iklan

Gambar 3.29 menggambarkan proses publikasi iklan oleh admin. Sistem menampilkan data iklan dari database, admin mengisi dan menyimpan data baru, kemudian sistem memperbarui tampilan setelah penyimpanan berhasil.

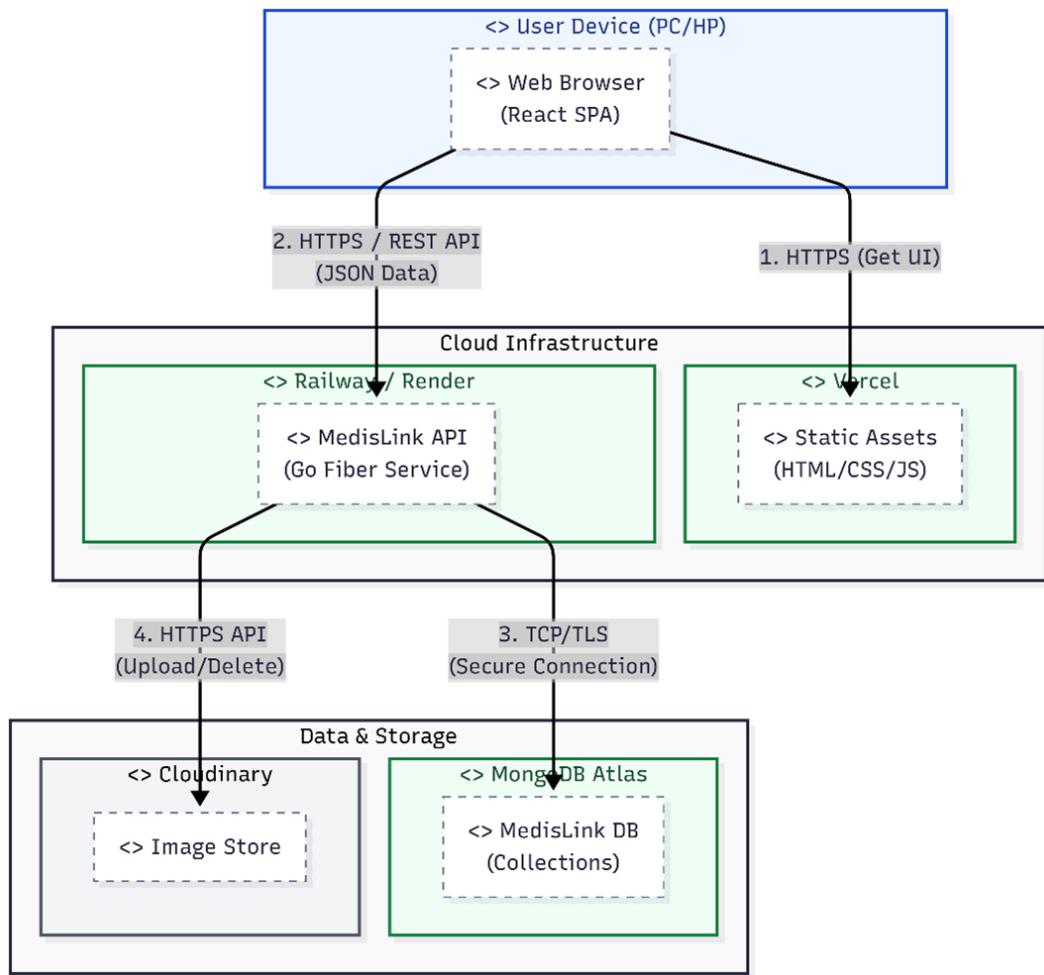
### 3.6.13 Admin Konfirmasi Donasi



Gambar 3.30: Sequence Diagram Admin Konfirmasi Donasi

Gambar 3.30 menunjukkan proses validasi donasi oleh admin. Sistem menampilkan daftar donasi pending dari database, admin meninjau detail dan melakukan konfirmasi. Sistem kemudian memperbarui status donasi menjadi disetujui serta menambahkan stok barang ke inventaris. Proses berakhir ketika notifikasi keberhasilan ditampilkan.

### 3.7 Deployment



Gambar 3.31: Deployment Diagram MedisLink

Gambar 3.31 menunjukkan arsitektur infrastruktur sistem MedisLink. Pengguna mengakses aplikasi melalui web browser pada perangkat PC atau HP. Aset frontend (React SPA) dihosting pada Vercel dan diakses melalui protokol HTTPS. Permintaan data dikirim melalui REST API ke backend MedisLink API yang berjalan pada Railway atau Render menggunakan protokol HTTPS. Backend terhubung secara aman melalui koneksi TCP/TLS ke MongoDB Atlas sebagai basis data utama. Selain itu, sistem juga terintegrasi dengan Cloudinary untuk penyimpanan dan pengelolaan aset gambar. Arsitektur ini memastikan keamanan komunikasi data serta pemisahan layanan frontend, backend, dan storage.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

## Bab 4

# IMPLEMENTASI DAN PENGUJIAN

## 4.1 Pembahasan Hasil Implementasi

### 4.1.1 Landing Page



Gambar 4.1: Halaman Landing Page

Gambar 4.1 merupakan halaman utama aplikasi MedisLink yang pertama kali ditampilkan kepada pengguna. Halaman ini berfungsi sebagai pengantar aplikasi dan menampilkan informasi singkat mengenai layanan peminjaman alat bantu medis. Pada bagian atas terdapat tombol *Masuk/Daftar* yang digunakan untuk proses autentikasi pengguna. Selain itu, tersedia tombol *Jelajahi Alat* untuk melihat daftar alat bantu medis serta tombol *Hubungi Kami* untuk memperoleh informasi lebih lanjut mengenai layanan yang tersedia.

## **BAB V**

### **KESIMPULAN DAN SARAN**

# Bibliografi

- Annaram. (2024). React virtual dom.
- Authors, T. G. (2023). *The go programming language documentation*. (<https://golang.org>)
- Axios. (2023). *Axios documentation*. (<https://axios-http.com>)
- Banubakode, & Chore. (2022). Rest api study.
- DaisyUI. (2023). *Daisyui documentation*. (<https://daisyui.com>)
- et al., F. (2022). Judul artikel.
- et al., K. (2022). Json in web services.
- et al., T. N. P. (2022). Judul artikel.
- et al., W. (2022). Javascript web development.
- Fiber. (2023). *Fiber web framework*. (<https://gofiber.io>)
- Gurusamy. (2020). Node.js performance.
- Habib. (2019). Microservices architecture.
- MongoDB. (2023). *Mongodb documentation*. (<https://www.mongodb.com>)
- Shoikhedbrod. (2023). Modern javascript.
- Systems, S. (2021). *Uml overview*.
- Yee. (2019). Event driven programming.