

# Software Requirements Specifications

Source: IEEE Std 830-1998 (**Revision of IEEE Std 830-1993**)

Edited by: Tim Dosen Oktober 2024

---

## Implementasi Paradigma Object Oriented: Sistem Perpustakaan Digital (Digilibz)

Digunakan untuk tugas Mata Kuliah  
Rekayasa Perangkat Lunak  
(CBK3FAB3)

Disiapkan oleh:

Kelompok : 2 (Dua)  
Kelas : IT-47-04

Anggota:

Syahril Arfian Almazril	103032300013
Shania Rahmalia	103032300018
Rara Nur Annisa	103032300077
Galuh Ajeng	103032300087
Neng Intan Nuraeini	103032330031

**Program Studi Sarjana Teknologi Informasi  
FIF – Telkom University  
Bandung 2025**

## Revision History

Author Name	Date	Description / Reason for Changes	Version
Kelompok digilibz	-	-	-

## Daftar Isi

<b>Daftar Isi .....</b>	<b>3</b>
<b>1. Introduction.....</b>	<b>4</b>
1.1 Purpose .....	4
1.2 Scope .....	4
1.3 Definitions, Acronyms, and Abbreviations.....	5
1.4 References .....	6
1.5 Overview.....	6
<b>2. Overall Description.....</b>	<b>7</b>
2.1 Product Perspective.....	7
2.2 Product Functions.....	7
2.3 User Characteristics .....	11
2.4 Constraints.....	12
2.5 Assumptions and Dependencies.....	12
<b>3. Specific Requirements .....</b>	<b>13</b>
3.1 External Interface Requirements .....	13
3.1.1 User Interfaces.....	13
3.1.2 Hardware Interfaces .....	14
3.1.3 Software Interfaces .....	14
3.1.4 Communications Interfaces .....	15
3.2 Functional Requirements.....	15
3.2.1 Information Flows.....	25
3.2.2 Process Description .....	34
3.2.3 Data Construct Spesifications .....	91
3.2.4 Data Dictionary .....	98
3.3 Performance Requirements .....	98
3.3.1 Design Constraints .....	98
3.3.2 Software System Attributes.....	99
3.3.3 Other Requirements.....	106

## **1. Introduction**

### **1.1 Purpose**

Dokumen ini disusun untuk memberikan pemahaman yang jelas dan mendalam mengenai kebutuhan perangkat lunak dari Sistem Perpustakaan Digital (Digilibz). Sistem ini dirancang dengan tujuan untuk meningkatkan efisiensi pengelolaan peminjaman buku melalui platform digital yang dapat diakses secara online. Sistem ini bertujuan untuk menyediakan layanan yang mempermudah User dalam mencari, meminjam, dan memberikan ulasan terhadap buku yang ada, serta memberikan Admin kontrol penuh atas pengelolaan data buku, verifikasi peminjaman, dan pembuatan laporan statistik.

Dokumen Software Requirements Specification ini ditujukan untuk:

- Tim Pengembang: Sebagai panduan teknis dalam membangun sistem sesuai dengan spesifikasi yang telah ditentukan
- Project Manager: Untuk memahami scope proyek dan mengalokasikan resources dengan tepat
- Stakeholder: Untuk memahami fungsi dan kemampuan sistem yang akan dikembangkan

### **1.2 Scope**

Sistem ini dirancang untuk digunakan oleh dua jenis pengguna utama: User dan Admin.

1. User: Pengguna yang terdaftar dan memiliki akses untuk mencari buku, meminjam buku, serta memberikan review dan rating terhadap buku yang telah dipinjam.
2. Admin: Pengelola sistem yang memiliki hak penuh untuk mengelola data buku, memverifikasi peminjaman, serta menghasilkan laporan terkait aktivitas perpustakaan.

Proses utama dalam sistem ini terdiri dari:

1. Registrasi Pengguna (Proses 1): Pengguna baru mendaftar dengan memasukkan data seperti nama, email, password, dan nomor telepon untuk membuat akun.
2. Pencarian dan Peminjaman Buku (Proses 2): Pengguna mencari buku yang tersedia dan mengajukan permintaan peminjaman. Admin memverifikasi dan menyetujui atau menolak permintaan peminjaman.

3. Pengelolaan Data Buku (Proses 3): Admin menambah, mengedit, atau menghapus data buku dalam sistem, serta mengelola stok buku yang tersedia untuk dipinjam.
4. Review dan Rating Buku (Proses 4): Pengguna yang telah meminjam buku dapat memberikan ulasan dan rating berdasarkan pengalaman mereka.

Fitur utama sistem:

1. Registrasi dan Login Pengguna: Pengguna dapat mendaftar dan login untuk mengakses berbagai layanan perpustakaan.
2. Pencarian Buku: Pengguna dapat mencari buku berdasarkan judul, penulis, kategori, dan tahun terbit.
3. Peminjaman Buku: Pengguna dapat mengajukan permintaan peminjaman buku dengan memilih buku dan menentukan tanggal peminjaman.
4. Manajemen Buku: Admin dapat menambah, mengedit, atau menghapus buku yang tersedia di sistem.
5. Sistem Review dan Rating: Pengguna dapat memberikan ulasan dan rating terhadap buku yang telah dipinjam.
6. Pembuatan Laporan: Admin dapat menghasilkan laporan mengenai statistik peminjaman, buku populer, danitas pengguna dalam periode tertentu (harian, mingguan, atau bulanan).

### 1.3 Definitions, Acronyms, and Abbreviations

Dalam pengembangan Sistem Perpustakaan Digital (Digilibz), terdapat berbagai istilah dan singkatan yang digunakan untuk mempermudah komunikasi antar tim pengembang dan pengguna sistem. Berikut adalah definisi dari beberapa istilah, akronim, dan singkatan yang penting dalam konteks sistem ini:

1. User: Pengguna yang terdaftar dalam sistem yang dapat melakukan pencarian dan peminjaman buku.
2. Admin: Pengelola sistem yang memiliki hak akses penuh untuk mengelola data buku dan transaksi peminjaman.
3. API: Antarmuka pemrograman aplikasi yang memungkinkan interaksi antara berbagai sistem.

4. UI: Antarmuka pengguna yang merujuk pada desain tampilan dan interaksi dengan pengguna.
5. CRUD: Operasi dasar dalam sistem database (Create, Read, Update, Delete).

## 1.4 References

Dokumen ini mengacu pada berbagai sumber dan standar industri sebagai berikut:

Academic References:

1. Mata Kuliah Pemrograman Berbasis Objek (PBO), Fakultas Informatika, Universitas Telkom
2. Software Engineering: A Practitioner's Approach by Roger S. Pressman
3. IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

Technical Documentation:

1. Spring Boot Official Documentation - <https://spring.io/projects/spring-boot>
2. Next.js Official Documentation - <https://nextjs.org/docs>
3. MySQL Documentation - <https://dev.mysql.com/doc>
4. MVC (Model-View-Controller) Architecture Pattern
5. Repository Pattern Documentation

## 1.5 Overview

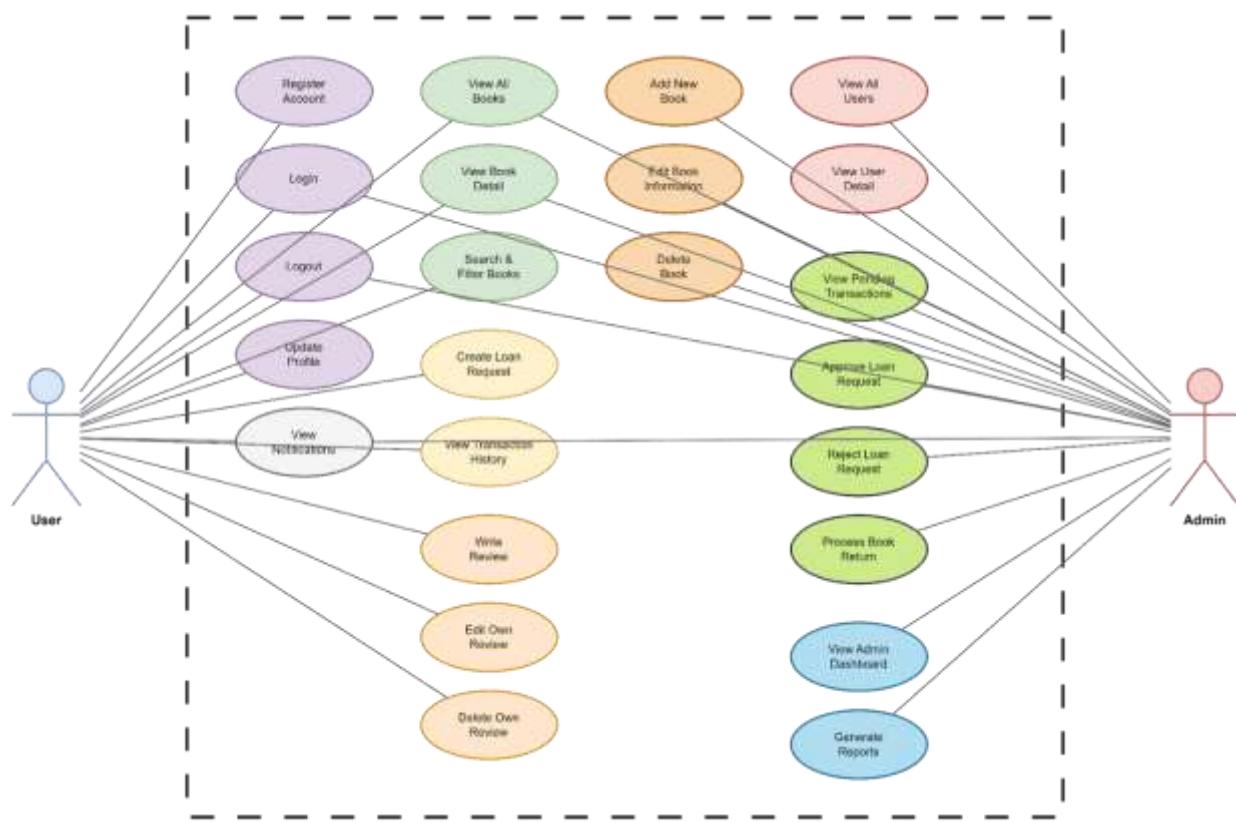
Dokumen ini memberikan penjelasan yang komprehensif mengenai kebutuhan perangkat lunak untuk pengembangan Sistem Perpustakaan Digital (Digilibz). Di dalamnya dijelaskan berbagai persyaratan fungsional dan non-fungsional yang harus dipenuhi oleh sistem, serta alur data yang menggambarkan interaksi antara komponen-komponen dalam sistem. Selain itu, dokumen ini juga memuat berbagai diagram yang menggambarkan interaksi antara pengguna (User) dan admin, serta proses yang terjadi dalam pengelolaan perpustakaan digital. Tujuan dari dokumentasi ini adalah untuk memastikan bahwa pengembangan sistem dilakukan secara terstruktur dan memenuhi kebutuhan pengguna, dengan hasil akhir yang efektif dan efisien dalam pengelolaan perpustakaan berbasis digital.

## 2. Overall Description

### 2.1 Product Perspective

Sistem Perpustakaan Digital (Digilibz) adalah aplikasi berbasis web yang dirancang untuk mengelola operasional perpustakaan dengan cara yang efisien. Sistem ini merupakan bagian dari mata kuliah Pemrograman Berbasis Objek (PBO) dan Rekayasa Perangkat Lunak (RPL) di Universitas Telkom, yang bertujuan untuk memberikan solusi dalam manajemen perpustakaan berbasis digital.

Berikut Adalah Use Case Diagram yang ada pada system kami:



### 2.2 Product Functions

Sistem Perpustakaan Digital (Digilibz) menyediakan berbagai fungsi utama yang dikelompokkan berdasarkan kategori proses bisnis:

#### 2.2.1 User Management Functions

##### 2.2.1.1 Authentication & Authorization:

- a. User Registration: Pendaftaran akun baru dengan email verification
- b. User Login: Autentikasi dengan email dan password
- c. User Logout: Mengakhiri session pengguna
- d. Password Hashing: Keamanan password menggunakan bcrypt
- e. Role-Based Access Control: Pembatasan akses berdasarkan role (USER/ADMIN)

#### 2.2.1.2 Profile Management:

- a. View Profile: Melihat informasi profil pengguna
- b. Update Profile: Mengubah nama, nomor telepon, dan lainnya
- c. Profile Statistics: Melihat statistik peminjaman pribadi

### 2.2.2 Book Management Functions

#### 2.2.2.1 Book Catalog (User & Admin):

- a. View All Books: Menampilkan katalog lengkap buku perpustakaan
- b. View Book Detail: Melihat informasi lengkap sebuah buku
- c. Filter Books: Filter berdasarkan kategori dan tahun terbit
- d. Book Availability Status: Menampilkan status ketersediaan real-time

#### 2.2.2.2 Book Administration (Admin Only):

- a. Add New Book: Menambahkan buku baru ke dalam sistem
- b. Edit Book Information: Memperbarui informasi buku existing
- c. Delete Book: Menghapus buku dengan soft delete mechanism
- d. Upload Book Cover: Mengelola gambar cover buku
- e. Manage Book Stock: Mengatur jumlah total dan available copies

## **2.2.3 Transaction Management Functions**

### **2.2.3.1 Borrowing Process (User):**

- a. Create Loan Request: Mengajukan permintaan peminjaman buku
- b. Select Loan Duration: Memilih rentang tanggal peminjaman (max 5 hari)
- c. View Transaction History: Melihat riwayat semua transaksi pribadi
- d. Check Active Loans: Melihat daftar buku yang sedang dipinjam

### **2.2.3.2 Transaction Processing (Admin):**

- a. View Pending Transactions: Melihat daftar permintaan yang menunggu approval
- b. Approve Loan Request: Menyetujui permintaan peminjaman
- c. Process Book Return: Memproses pengembalian buku
- d. Record Payment: Mencatat metode pembayaran dan bukti transfer
- e. Update Transaction Status: Mengelola status transaksi (PENDING → APPROVED → RETURNED)

### **2.2.3.3 Transaction Support:**

- a. Generate Invoice Code: Membuat kode invoice unik (INV-YYYYMMDD-XXX)
- b. Multi-Book Transaction: Mendukung peminjaman multiple buku dalam satu transaksi
- c. Transaction Detail View: Menampilkan detail lengkap transaksi
- d. Stock Update Automation: Update stok buku otomatis saat approve/return

## **2.2.4 Review & Rating Functions**

### **2.2.4.1 Review Management (User):**

- a. Write Review: Menulis review untuk buku yang pernah dipinjam
- b. Review Eligibility Check: Validasi user hanya bisa review buku yang pernah dipinjam

#### 2.2.4.2 Review Display (All Users):

- a. View All Reviews: Melihat semua review untuk sebuah buku
- b. View Average Rating: Menampilkan rata-rata rating dengan visualisasi bintang
- c. One Review Per User Rule: Sistem memastikan tidak ada duplicate review

### 2.2.5 Notification System Functions

#### 2.2.5.1 In-App Notifications:

- a. Transaction Status Notification: Notifikasi saat status transaksi berubah
- b. Overdue Alert: Alert jika user terlambat mengembalikan buku
- c. Admin New Request Alert: Notifikasi ke admin saat ada loan request baru
- d. View Notification List: Menampilkan daftar semua notifikasi
- e. Mark as Read: Menandai notifikasi sudah dibaca

### 2.2.6 Dashboard & Reporting Functions

#### 2.2.6.1 Admin Dashboard:

- a. View Statistics Summary: Total buku, total user, total transaksi aktif
- b. View Pending Approvals: Quick view permintaan yang perlu di-approve

#### 2.2.6.2 Report Generation (Admin):

- a. User Activity Report: Statistik aktivitas dan registrasi user

- b. Revenue Report: Laporan total pendapatan dari denda keterlambatan
- c. Export Reports: Generate laporan dalam format PDF atau Excel

#### 2.2.6.3 User Dashboard:

- a. View Active Loans: Melihat daftar buku yang sedang dipinjam
- b. View Borrowing History: Riwayat peminjaman
- c. View Personal Statistics: Total peminjaman, total harga, dll
- d. View Recommended Books: Rekomendasi buku berdasarkan history

### 2.2.7 Search & Recommendation Functions

#### 2.2.7.1 Search Functionality:

- a. Search by Title: Pencarian berdasarkan judul buku (partial matching)
- b. Search by Author: Pencarian berdasarkan nama penulis
- c. Advanced Filter: Kombinasi search + filter kategori/tahun

#### 2.2.7.2 Recommendation System:

- a. Random Recommendation: Menampilkan 5-10 buku secara acak di homepage
- b. Popular Books Display: Daftar buku populer berdasarkan jumlah peminjaman

### 2.3 User Characteristics

Sistem ini dirancang untuk memenuhi kebutuhan dua jenis pengguna utama:

1. User (Pengguna): Pengguna yang terdaftar dalam sistem dan dapat mencari buku, meminjam buku, serta memberikan ulasan dan rating terhadap buku yang telah dipinjam.

2. Admin (Administrator): Pengelola sistem yang memiliki hak akses penuh untuk mengelola buku, memverifikasi transaksi peminjaman, serta menghasilkan laporan dan statistik.

## 2.4 Constraints

Beberapa batasan dalam pengembangan dan penggunaan Sistem Perpustakaan Digital (Digilibz) antara lain:

1. Ketersediaan Akses Internet: Sistem ini membutuhkan koneksi internet yang stabil untuk dapat diakses oleh pengguna dan admin, karena sistem berbasis web.
2. Kompatibilitas Perangkat: Sistem harus kompatibel dengan perangkat komputer atau smartphone yang dapat mengakses platform berbasis web, namun mungkin tidak optimal di perangkat dengan kapasitas rendah atau koneksi internet yang buruk.
3. Keamanan Data: Data pribadi pengguna dan informasi peminjaman buku harus dijaga kerahasiaannya. Sistem ini perlu dilengkapi dengan langkah-langkah keamanan yang memadai, termasuk enkripsi data dan penggunaan protokol aman (HTTPS).
4. Skalabilitas Sistem: Sistem ini harus dirancang untuk mendukung peningkatan jumlah pengguna dan buku yang terdaftar tanpa mengalami penurunan performa.

Alur interaksi pengguna dengan sistem serta proses bisnis utama akan dijelaskan secara rinci melalui Use Case Diagram dan Sequence Diagram pada bagian proses sistem. Oleh karena itu, pada bagian ini pembahasan difokuskan pada batasan sistem tanpa menampilkan diagram tersebut, guna menjaga kejelasan struktur dokumen dan menghindari pengulangan informasi.

## 2.5 Assumptions and Dependencies

Beberapa asumsi dan ketergantungan yang mendasari pengembangan dan operasional Sistem Perpustakaan Digital (Digilibz) adalah:

1. Ketersediaan Sumber Daya Teknologi: Sistem ini bergantung pada ketersediaan infrastruktur teknologi yang memadai, termasuk server dan database untuk menyimpan informasi buku dan transaksi pengguna.

2. Akses Internet Pengguna: Diharapkan bahwa pengguna memiliki akses internet yang cukup untuk dapat mengakses sistem, baik menggunakan perangkat desktop maupun mobile.
3. Pemeliharaan dan Pembaruan Sistem: Sistem ini mengasumsikan bahwa pemeliharaan dan pembaruan perangkat lunak dilakukan secara berkala untuk memastikan kinerja yang optimal dan melindungi dari potensi kerentanannya.
4. Dukungan Pengguna: Diharapkan adanya dukungan teknis yang tersedia bagi pengguna dan admin jika terjadi masalah dalam penggunaan sistem, seperti masalah login, kesalahan data, atau kendala lainnya.

### **3. Specific Requirements**

#### **3.1 External Interface Requirements**

Bagian ini menjelaskan kebutuhan antarmuka eksternal sistem, meliputi antarmuka pengguna, perangkat keras, perangkat lunak, komunikasi, serta batasan dan aspek keamanan yang berkaitan dengan interaksi sistem dengan lingkungan eksternal.

##### **3.1.1 User Interfaces**

<b>Kode</b>	<b>Antarmuka</b>	<b>Deskripsi</b>
<b>UI-01</b>	Landing Page	Halaman awal sistem yang dapat diakses tanpa autentikasi dan berfungsi sebagai pengenalan sistem
<b>UI-02</b>	Halaman Registrasi	Antarmuka untuk pendaftaran pengguna baru
<b>UI-03</b>	Halaman Login	Antarmuka untuk autentikasi pengguna
<b>UI-04</b>	Dashboard User	Halaman utama setelah pengguna berhasil login
<b>UI-05</b>	Katalog Buku	Antarmuka untuk menampilkan daftar buku
<b>UI-06</b>	Detail Buku	Antarmuka untuk menampilkan informasi detail buku

<b>UI-07</b>	Transaksi Peminjaman	Antarmuka pengajuan dan pemantauan peminjaman
<b>UI-08</b>	Riwayat Transaksi	Antarmuka histori peminjaman pengguna
<b>UI-09</b>	Profil Pengguna	Antarmuka pengelolaan data akun pengguna
<b>UI-10</b>	Notifikasi	Antarmuka pemberitahuan sistem
<b>UI-11</b>	Dashboard Admin	Halaman ringkasan sistem untuk Admin
<b>UI-12</b>	Manajemen Buku	Antarmuka Admin untuk pengelolaan data buku
<b>UI-13</b>	Manajemen Pengguna	Antarmuka Admin untuk pengelolaan akun pengguna

### 3.1.2 Hardware Interfaces

Sistem tidak memiliki ketergantungan terhadap perangkat keras khusus. Sistem dijalankan pada server dengan spesifikasi standar yang mendukung aplikasi backend dan basis data.

Pengguna mengakses system menggunakan perangkat umum seperti computer, laptop, tablet, atau telepon pintar yang memiliki peramban web dan koneksi internet. Sistem tidak terintegrasi dengan perangkat keras eksternal seperti pemindai barcode, perangkat RFID, atau perangkat otomatis lainnya. Seluruh proses peminjaman dan pengembalian buku dilakukan secara manual melalui antarmuka system.

### 3.1.3 Software Interfaces

Sistem menggunakan arsitektur berbasis client-server. Pada sisi klien, sistem berjalan pada peramban web dengan frontend yang dikembangkan menggunakan framework Next.js. Pada sisi server, sistem berjalan menggunakan framework backend Spring Boot dengan bahasa pemrograman Java versi 17 atau lebih baru.

Sistem menggunakan sistem manajemen basis data relasional berbasis SQL, yaitu PostgreSQL atau MySQL. Akses basis data dilakukan menggunakan Spring Data JPA

(Hibernate) sebagai ORM. Sistem tidak menggunakan pendekatan akses basis data langsung melalui JDBC mentah maupun query manual.

Komunikasi antara frontend dan backend dilakukan melalui RESTful API dengan format data JSON. Sistem tidak mendukung penggunaan GraphQL, gRPC, atau arsitektur API lainnya di luar REST. Sistem menggunakan Git dan Github sebagai system version control untuk pengelolaan kode sumber. Process build dan dependency management dilakukan menggunakan Maven.

### **3.1.4 Communications Interfaces**

Komunikasi antara klien dan server dilakukan melalui jaringan internet menggunakan protokol HTTP yang diamankan dengan HTTPS. Seluruh pertukaran data menggunakan format JSON. Setiap permintaan yang dikirimkan ke server harus melalui mekanisme autentikasi untuk memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses sumber daya sistem. Sistem tidak menyediakan fitur komunikasi real-time seperti WebSocket atau Server-Sent Events. Pembaruan notifikasi dilakukan melalui mekanisme in-application notification dan diperbarui saat pengguna melakukan navigasi atau pemuatan ulang halaman.

Unggahan berkas, seperti gambar sampul buku, dilakukan melalui mekanisme multipart request dan disimpan pada sistem penyimpanan lokal atau penyimpanan objek sederhana tanpa penggunaan CDN.

## **3.2 Functional Requirements**

### **FR-AUTH: Authentication & Authorization**

ID	Requirement	Priority	Description
FR-AUTH-01	User Registration	High	Sistem harus memungkinkan pengguna baru mendaftar dengan email, password, nama, dan nomor telepon

FR-AUTH-02	User Login	High	Sistem harus memvalidasi kredensial pengguna dan memberikan akses sesuai role (Admin atau User)
FR-AUTH-03	User Log Out	Medium	Sistem harus menyediakan fungsi log out yang menghapus session pengguna
FR-AUTH-04	Password Encryption	High	Sistem harus menyimpan password dalam bentuk terenkripsi menggunakan bcrypt atau algoritma hash yang aman
FR-AUTH-05	Role-Based Access Control	High	Sistem harus membatasi akses fitur berdasarkan role pengguna (Admin memiliki akses penuh, User memiliki akses terbatas)
FR-AUTH-06	Session Management	High	Sistem harus mengelola session pengguna dengan timeout 30 menit untuk keamanan

### FR-BOOK: Book Management

ID	Requirement	Priority	Description
FR-BOOK-01	Add Book	High	Admin dapat menambahkan buku baru dengan detail: judul, penulis, kategori, tahun terbit, ISBN, deskripsi, cover image, jumlah stok total, jumlah tersedia, biaya denda per hari
FR-BOOK-02	View All Books	High	Semua pengguna (User dan Admin) dapat melihat katalog lengkap buku yang tersedia di perpustakaan

FR-BOOK-03	View Book Detail	High	Semua pengguna dapat melihat detail lengkap sebuah buku termasuk informasi buku, ketersediaan stok, dan daftar review dari pengguna lain
FR-BOOK-04	Update Book	Medium	Admin dapat memperbarui informasi buku yang sudah ada dalam sistem
FR-BOOK-05	Delete Book	Medium	Admin dapat menghapus buku dari sistem menggunakan soft delete (data tidak sepenuhnya terhapus, hanya ditandai sebagai deleted)
FR-BOOK-06	Book Stock Management	High	Sistem harus otomatis mengurangi jumlah stok available saat buku dipinjam dan menambah kembali saat buku dikembalikan
FR-BOOK-07	Book Availability Status	High	Sistem harus menampilkan status ketersediaan buku secara real-time (tersedia/tidak tersedia) berdasarkan jumlah available_copies
FR-BOOK-08	Book Cover Upload	Medium	Admin dapat mengunggah gambar cover buku dengan format JPG/PNG maksimal 5MB

#### FR-SEARCH: Search & Filter

ID	Requirement	Priority	Description
FR-SEARCH-01	Search by Title	High	User dapat mencari buku berdasarkan judul dengan partial matching (case-insensitive)

FR-SEARCH-02	Search by Author	High	User dapat mencari buku berdasarkan nama penulis dengan partial matching
FR-SEARCH-03	Filter by Category	Medium	User dapat memfilter daftar buku berdasarkan kategori (contoh: Teknologi, Bisnis, Sastra, dll)
FR-SEARCH-04	Filter by Year	Low	User dapat memfilter buku berdasarkan tahun terbit
FR-SEARCH-05	Sort Results	Medium	User dapat mengurutkan hasil pencarian berdasarkan: judul (A-Z), tahun terbit (terbaru), atau rating (tertinggi)
FR-SEARCH-06	Combined Search	Medium	User dapat mengkombinasikan pencarian dan filter (contoh: cari "Java" dalam kategori "Teknologi")

### FR-RECOM: Recommendation System

ID	Requirement	Priority	Description
FR-RECOM-01	Random Recommendation	Medium	Sistem menampilkan 5-10 rekomendasi buku secara acak di homepage untuk membantu user discover buku baru
FR-RECOM-02	Popular Books	Medium	Sistem menampilkan daftar buku populer berdasarkan jumlah peminjaman

FR-RECOM-03	Highest Rated Books	Low	Sistem menampilkan buku dengan rating tertinggi berdasarkan average rating dari user reviews
-------------	---------------------	-----	--

### FR-USER: User Management

ID	Requirement	Priority	Description
FR-USER-01	View All Users	Medium	Admin dapat melihat daftar lengkap semua pengguna terdaftar di sistem dengan informasi: nama, email, role, tanggal registrasi
FR-USER-02	Update User Profile	Medium	User dapat memperbarui profil mereka sendiri (nama, nomor telepon, foto profil)
FR-USER-03	Delete/Deactivate User	Low	Admin dapat menonaktifkan akun pengguna (soft delete) jika diperlukan
FR-USER-04	View User Detail	Medium	Admin dapat melihat detail lengkap user termasuk riwayat peminjaman dan statistik aktivitas
FR-USER-05	User Profile Page	Medium	User dapat mengakses halaman profil untuk melihat informasi akun dan statistik peminjaman mereka

### FR-TRANS: Transaction Management

ID	Requirement	Priority	Description
FR-TRANS-01	Create Loan Request	High	User dapat membuat permintaan peminjaman buku dengan memilih

			buku dan menentukan rentang tanggal peminjaman (date_from dan date_to)
FR-TRANS-02	Generate Invoice Code	High	Sistem otomatis generate invoice code unik untuk setiap transaksi dengan format: INV-YYYYMMDD-XXX
FR-TRANS-03	Transaction Status Management	High	Sistem harus melacak status transaksi: PENDING (menunggu approval), APPROVED (disetujui admin), REJECTED (ditolak admin), RETURNED (sudah dikembalikan)
FR-TRANS-04	Admin Approve Loan	High	Admin dapat menyetujui permintaan peminjaman dari user setelah melakukan verifikasi
FR-TRANS-05	Admin Reject Loan	High	Admin dapat menolak permintaan peminjaman dengan memberikan alasan penolakan
FR-TRANS-06	Return Book Process	High	Admin dapat memproses pengembalian buku dan sistem otomatis update status transaksi dan stok buku
FR-TRANS-07	Late Fee Calculation	Medium	Sistem harus otomatis menghitung denda keterlambatan berdasarkan jumlah hari terlambat $\times$ late_fee per hari buku
FR-TRANS-08	Payment Recording	Medium	Sistem harus mencatat metode pembayaran (cash/transfer) dan bukti pembayaran jika ada

FR-TRANS-09	Transaction History	Medium	User dapat melihat riwayat lengkap semua transaksi peminjaman mereka (pending, approved, rejected, returned)
FR-TRANS-10	Multi-Book Transaction	Medium	Satu transaksi dapat berisi multiple buku (melalui TransactionItem)
FR-TRANS-11	Transaction Detail View	High	User dan Admin dapat melihat detail lengkap transaksi termasuk buku yang dipinjam, tanggal, biaya, dan status
FR-TRANS-12	Loan Duration Validation	Medium	Sistem harus memvalidasi durasi peminjaman maksimal 30 hari per transaksi
FR-TRANS-13	User Quota Validation	High	Sistem harus memvalidasi bahwa user tidak melebihi kuota peminjaman (maksimal 5 buku aktif bersamaan)

#### FR-REVIEW: Review & Rating System

ID	Requirement	Priority	Description
FR-REVIEW-01	Submit Review	Medium	User yang pernah meminjam buku dapat memberikan review dan rating (1-5 bintang) untuk buku tersebut
FR-REVIEW-02	View All Reviews	Medium	Semua pengguna dapat melihat daftar review untuk sebuah buku di halaman detail buku

FR-REVIEW-03	Edit Own Review	Low	User dapat mengedit review dan rating yang pernah mereka buat
FR-REVIEW-04	Delete Own Review	Low	User dapat menghapus review mereka sendiri
FR-REVIEW-05	Average Rating Calculation	Medium	Sistem otomatis menghitung rata-rata rating untuk setiap buku dari semua review yang ada
FR-REVIEW-06	Review Eligibility Check	Medium	Sistem harus memvalidasi bahwa user hanya dapat review buku yang pernah mereka pinjam (cek riwayat transaksi)
FR-REVIEW-07	Review Timestamp	Low	Sistem mencatat waktu review dibuat dan waktu terakhir di update
FR-REVIEW-08	One Review Per User	Medium	Sistem harus memastikan satu user hanya bisa membuat satu review per buku (tidak boleh duplicate)

### FR-NOTIF: Notification System

ID	Requirement	Priority	Description
FR-NOTIF-01	Transaction Status Notification	High	Sistem otomatis mengirim notifikasi ke user saat status transaksi berubah (approved, rejected, atau returned)

FR-NOTIF-02	Due Date Reminder	Medium	Sistem mengirim notifikasi reminder H-2 sebelum tanggal pengembalian (date_to) untuk mengingatkan user
FR-NOTIF-03	Overdue Alert	High	Sistem mengirim notifikasi alert jika user terlambat mengembalikan buku (melewati date_to)
FR-NOTIF-04	Notification List	Medium	User dapat melihat daftar semua notifikasi mereka di halaman notifications
FR-NOTIF-05	Mark as Read	Low	User dapat menandai notifikasi sebagai sudah dibaca (is_read = true)
FR-NOTIF-06	Unread Count	Low	Sistem menampilkan jumlah notifikasi yang belum dibaca di UI (badge notification)
FR-NOTIF-07	Admin Notification	Medium	Admin menerima notifikasi saat ada loan request baru yang perlu di-approve
FR-NOTIF-08	System Announcement	Low	Admin dapat mengirim pengumuman sistem ke semua user (contoh: maintenance notice)

### FR-DASH: Dashboard & Analytics

ID	Requirement	Priority	Description
FR-DASH-01	Admin Dashboard	High	Admin dapat melihat dashboard dengan statistik ringkas: total

			buku, total user, total transaksi aktif, pending approvals
FR-DASH-02	Recent Reviews Display	Medium	Dashboard admin menampilkan 5 review terbaru dari user untuk monitoring feedback
FR-DASH-03	Transaction Report	Medium	Admin dapat generate laporan transaksi berdasarkan periode tertentu (daily, weekly, monthly)
FR-DASH-04	Popular Books Report	Low	Admin dapat melihat laporan buku paling sering dipinjam (top borrowers)
FR-DASH-05	User Activity Report	Low	Admin dapat melihat statistik aktivitas user (most active users, registration trends)
FR-DASH-06	Revenue Report	Low	Admin dapat melihat laporan total pendapatan dari denda keterlambatan
FR-DASH-07	Book Availability Overview	Medium	Dashboard menampilkan overview ketersediaan buku (total available vs total borrowed)

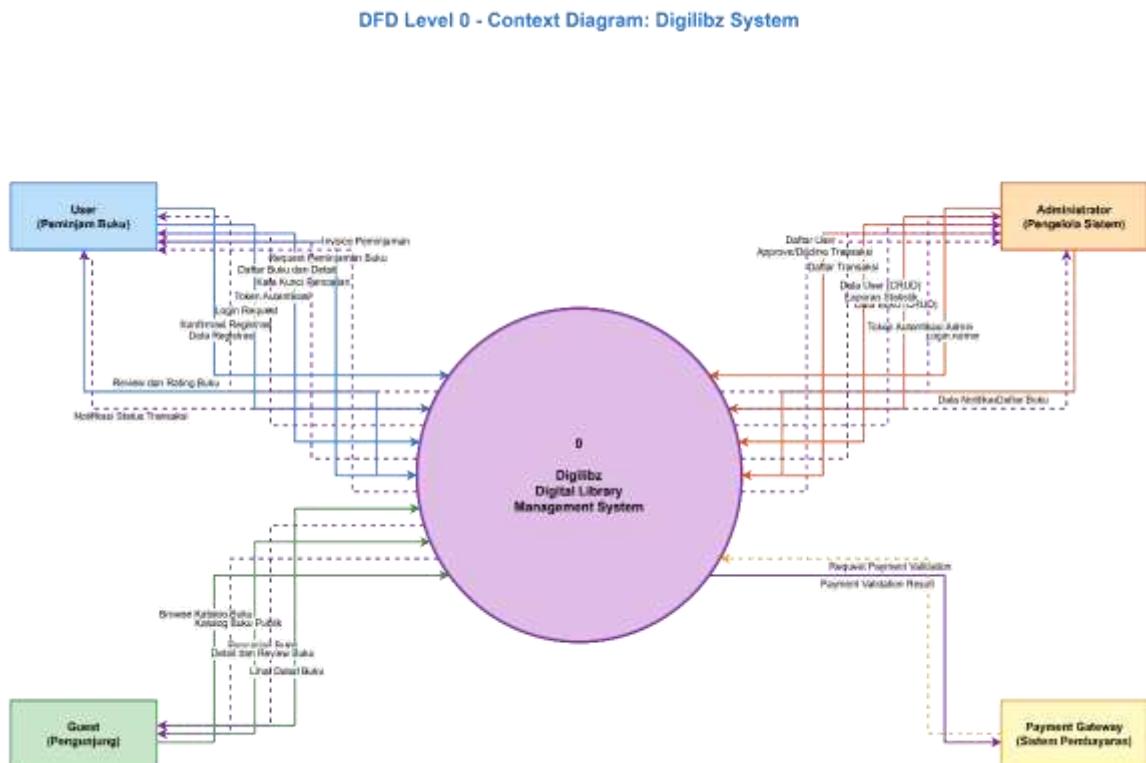
## TOTAL FUNCTIONAL REQUIREMENTS: 67 FR

- Authentication: 6 FR
- Book Management: 8 FR
- Search & Filter: 6 FR

- Recommendation: 3 FR
- User Management: 5 FR
- Transaction Management: 13 FR
- Review & Rating: 8 FR
- Notification: 8 FR
- Dashboard & Analytics: 7 FR

### 3.2.1 Information Flows

#### 3.2.1.1 Data Flow Diagram 1



##### 3.2.1.1.1 Data Entities

Berdasarkan DFD Level 0 Digilibz, data entities adalah sebagai berikut:

1. User (Peminjam Buku)

Mengelola data registrasi, login, token autentikasi, data peminjaman buku, invoice peminjaman, review dan rating buku, serta menerima notifikasi status transaksi.

2. Guest (Pengunjung)

Mengakses data katalog buku publik, detail buku, serta melihat review buku tanpa harus melakukan autentikasi.

3. Administrator (Pengelola Sistem)

Mengelola data user, data buku, data transaksi peminjaman, laporan statistik, autentikasi admin, serta menerima dan mengirim data notifikasi sistem.

4. Payment Gateway (Sistem Pembayaran)

Mengelola data validasi pembayaran dan mengirimkan hasil validasi pembayaran ke sistem Digilibz.

### 3.2.1.1.2 Pertinent Processes

Proses utama yang terjadi dalam sistem Digilibz berdasarkan DFD Level:

1. Manajemen Autentikasi

Proses login, registrasi user, serta pengelolaan token autentikasi untuk user dan administrator.

2. Manajemen Katalog Buku

Proses penelusuran katalog buku, pencarian berdasarkan kata kunci, serta penyajian detail buku.

3. Manajemen Peminjaman Buku

Proses pengajuan peminjaman buku, pembuatan invoice peminjaman, serta pembaruan status transaksi.

4. Manajemen Pembayaran

Proses pengiriman permintaan validasi pembayaran ke payment gateway dan penerimaan hasil validasi pembayaran.

5. Manajemen Review dan Rating

Proses pemberian review dan rating buku oleh user serta penyajian review kepada user lain dan guest.

#### 6. Manajemen Administrasi Sistem

Proses pengelolaan data user, data buku, data transaksi, laporan statistik, serta persetujuan atau penolakan transaksi oleh administrator.

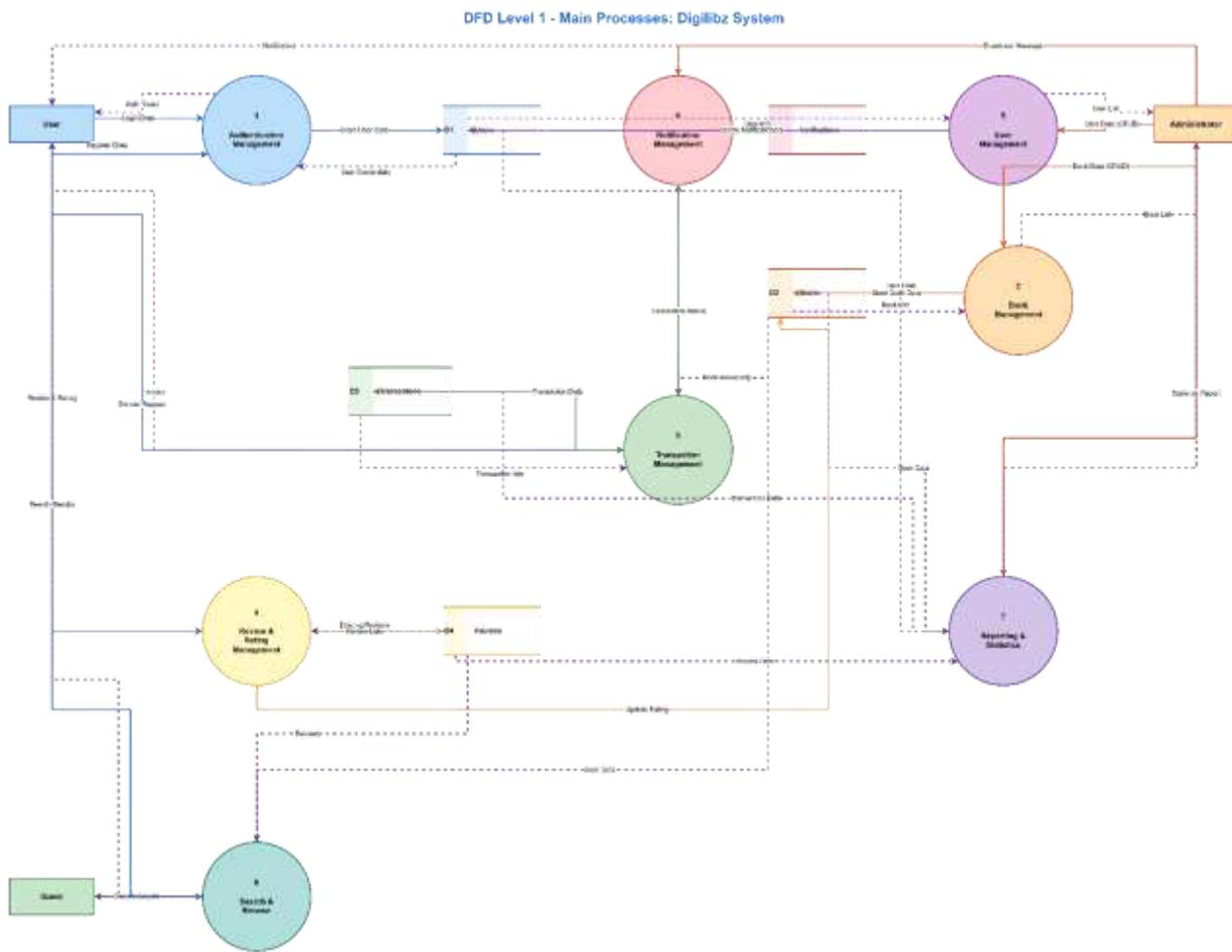
##### **3.2.1.1.3 Topology**

Dalam ini Digilibz menggunakan arsitektur client-server terpusat, yaitu:

- Digilibz Digital Library Management System berperan sebagai pusat proses (centralized system).
- User, Guest, dan Administrator berinteraksi dengan sistem melalui antarmuka aplikasi.
- Payment Gateway terhubung sebagai sistem eksternal untuk validasi pembayaran.
- Seluruh pertukaran data dilakukan secara dua arah antara entitas eksternal dan sistem Digilibz melalui aliran data yang terdefinisi pada DFD Level 0.
- Topologi ini memungkinkan pengelolaan data terpusat, keamanan akses yang terkontrol, serta integrasi yang efisien dengan sistem pembayaran eksternal.

Topologi ini memungkinkan pengelolaan data terpusat, keamanan akses yang terkontrol, serta integrasi yang efisien dengan sistem pembayaran eksternal.

### 3.2.1.2 Data Flow Diagram 2



#### 3.2.1.2.1 Data Entities

Data entities yang digunakan dan dikelola oleh sistem meliputi:

##### 1. User Data

Menyimpan informasi user seperti data registrasi, kredensial login, dan status autentikasi.

##### 2. Book Data

Berisi informasi buku meliputi judul, penulis, kategori, stok buku, dan detail buku lainnya.

##### 3. Transaction Data

Menyimpan data peminjaman buku, status transaksi, invoice peminjaman, serta data pembayaran.

4. Review & Rating Data

Menyimpan data ulasan dan penilaian buku yang diberikan oleh user.

5. Notification Data

Menyimpan informasi notifikasi terkait status transaksi, peminjaman, dan aktivitas sistem lainnya.

6. Report & Statistics Data

Menyimpan data laporan statistik penggunaan sistem, transaksi, dan data pendukung evaluasi sistem.

### 3.2.1.2.2 Pertinent Processes

Berikut adalah proses utama yang terjadi pada DFD Level 1 Digilibz:

1. Authentication Management

Mengelola proses registrasi, login user, validasi kredensial, serta pembuatan dan pengelolaan token autentikasi.

2. Book Management

Mengelola data buku termasuk tambah, ubah, hapus, dan melihat data buku oleh administrator.

3. Transaction Management

Mengelola proses peminjaman buku, pencatatan transaksi, pembaruan status transaksi, serta integrasi pembayaran.

4. Review & Rating Management

Mengelola proses input review dan rating buku oleh user serta penyajian data review.

5. Search & Browse

Mengelola pencarian dan penelusuran katalog buku yang dapat diakses oleh user dan guest.

6. User Management

Mengelola data user oleh administrator termasuk melihat, memperbarui, dan menghapus data user.

7. Reporting & Statistics

Mengelola pembuatan laporan statistik terkait transaksi, buku, dan aktivitas pengguna.

8. Notification Management

Mengelola pengiriman notifikasi kepada user dan administrator terkait status transaksi dan aktivitas sistem.

### 3.2.1.2.3 Topology

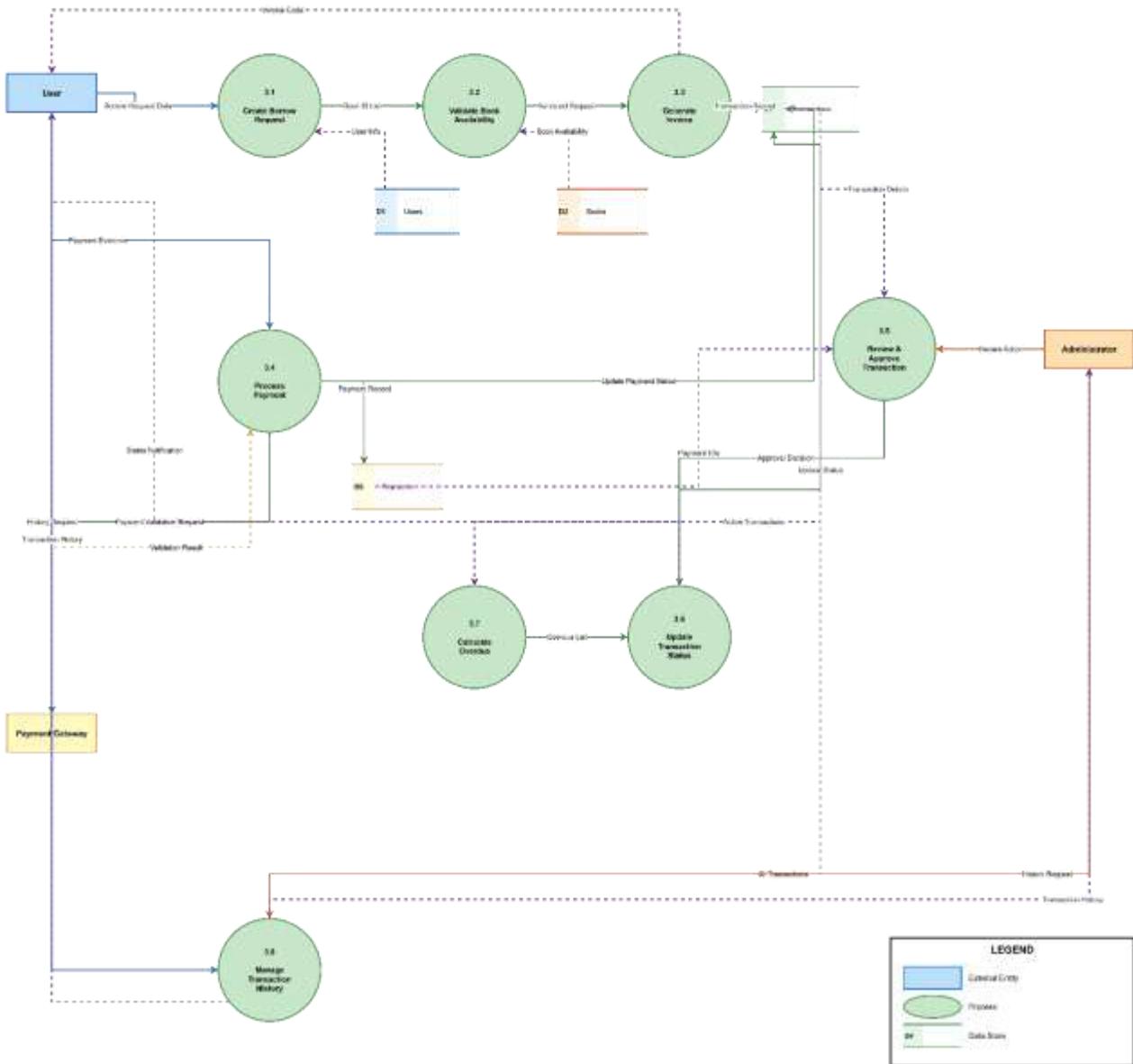
Pada DFD Level 1 Digilibz menggunakan arsitektur client-server dengan sistem terpusat, di mana:

- Sistem Digilibz Digital Library Management System berperan sebagai pusat pemrosesan data.
- User, Guest, dan Administrator berinteraksi dengan sistem melalui modul-modul fungsional yang berbeda.
- Setiap proses utama saling terhubung melalui aliran data internal sistem.
- Sistem terintegrasi dengan komponen eksternal (payment/validasi transaksi) melalui proses Transaction Management.
- Data disimpan dan dikelola dalam data store terpisah sesuai fungsi masing-masing proses.

Topologi ini mendukung modularitas sistem, kemudahan pengelolaan data, serta skalabilitas pengembangan sistem Digilibz.

### 3.2.1.3 Data Flow Diagram 3

DFD Level 2 - Transaction Management (Process 3 Decomposition)



### **3.2.1.n.1 Data Entities**

Berdasarkan DFD Level 2 – Transaction Management Digilibz, data entities yang terlibat dalam proses transaksi adalah sebagai berikut:

1. User  
Mengirimkan data permintaan peminjaman buku, bukti pembayaran, serta menerima informasi status transaksi dan riwayat transaksi.
2. Book (D2 – Books)  
Menyimpan data buku dan status ketersediaan buku yang digunakan dalam proses validasi peminjaman.
3. User Data (D1 – Users)  
Menyimpan data identitas user yang digunakan saat pembuatan permintaan peminjaman.
4. Transaction Data (D3 – Transactions)  
Menyimpan data transaksi peminjaman, invoice, status transaksi, serta detail transaksi lainnya.
5. Payment Data (D6 – Payments)  
Menyimpan data pembayaran dan hasil validasi pembayaran dari sistem payment gateway.

### **3.2.1.n.2 Pertinent Processes**

Proses utama yang terjadi pada DFD Level 2 Transaction Management meliputi:

1. Create Borrow Request (3.1)  
Proses pengajuan permintaan peminjaman buku oleh user.
2. Validate Book Availability (3.2)  
Proses pengecekan ketersediaan buku berdasarkan data buku yang tersimpan.
3. Generate Invoice (3.3)

Proses pembuatan invoice peminjaman setelah buku dinyatakan tersedia.

4. Process Payment (3.4)

Proses pengiriman data pembayaran ke payment gateway dan penerimaan hasil validasi pembayaran.

5. Review & Approve Transaction (3.5)

Proses peninjauan dan persetujuan transaksi peminjaman oleh administrator.

6. Update Transaction Status (3.6)

Proses pembaruan status transaksi berdasarkan hasil pembayaran dan persetujuan admin.

7. Calculate Overdue (3.7)

Proses perhitungan denda keterlambatan pengembalian buku.

8. Manage Transaction History (3.8)

Proses pencatatan dan pengelolaan riwayat transaksi peminjaman buku.

### 3.2.1.n.3 Topology

Topologi pada DFD Level 2 Transaction Management Digilibz menggunakan arsitektur client-server terpusat, di mana:

- User berinteraksi dengan sistem untuk melakukan permintaan peminjaman dan pembayaran.
- Transaction Management berperan sebagai pusat pengolahan data transaksi.
- Administrator terlibat dalam proses review dan persetujuan transaksi.
- Payment Gateway berfungsi sebagai sistem eksternal untuk validasi pembayaran.

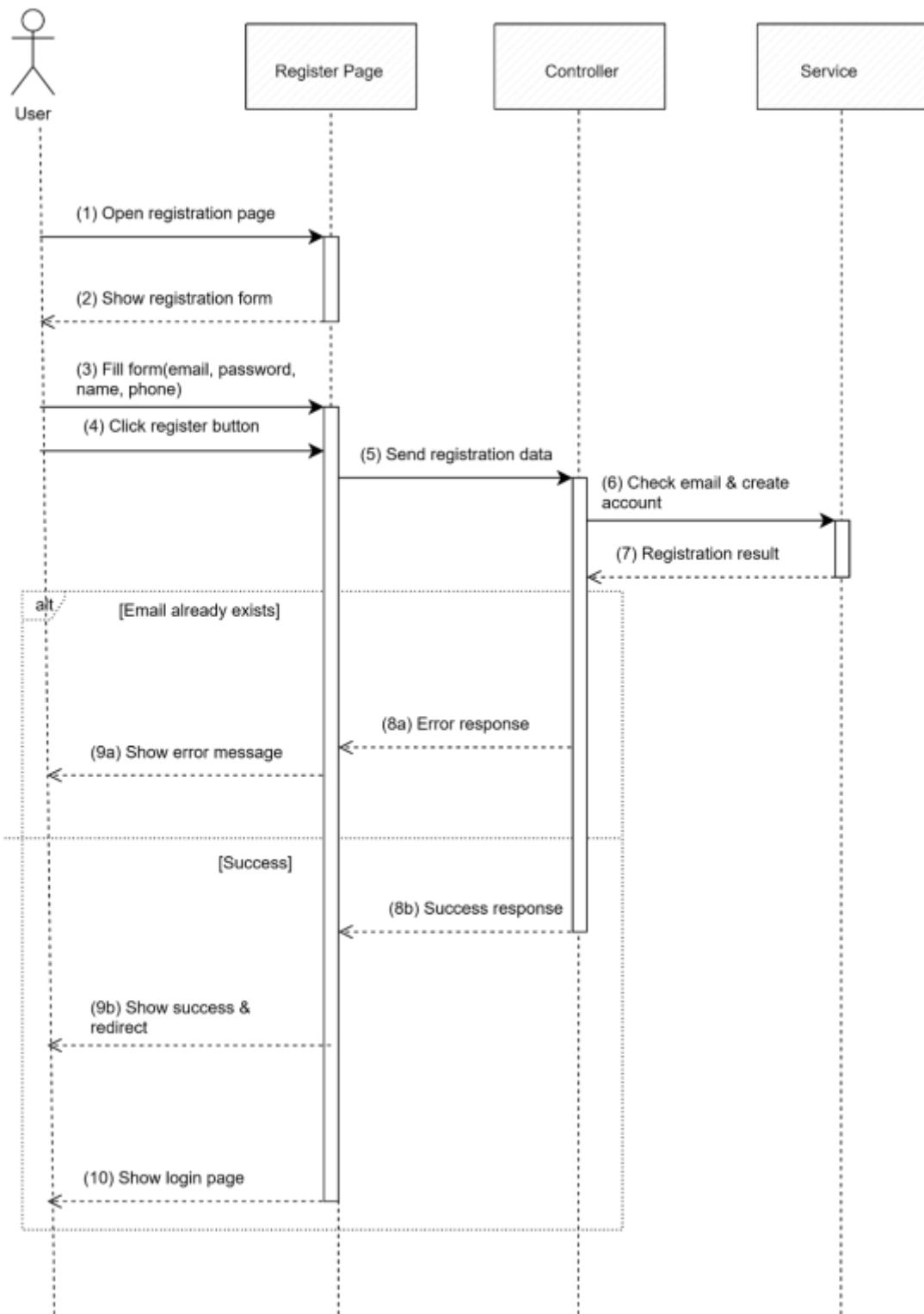
- Data transaksi, buku, user, dan pembayaran disimpan pada data store terpisah sesuai fungsinya.

Topologi ini memastikan proses transaksi berjalan terstruktur, terkontrol, dan terintegrasi dengan sistem pembayaran eksternal.

### 3.2.2 Process Description

#### 3.2.2.1 Process 1

Aspek	Detail
<b>Use case name</b>	Register Account
<b>Description</b>	User baru mendaftar akun di sistem Digilibz dengan mengisi form registrasi lengkap
<b>Pre-condition</b>	User belum memiliki akun di sistem Digilibz
<b>Post-condition</b>	Akun user berhasil dibuat dan tersimpan di database dengan status active



### **3.2.2.1.1 Input Data Entities**

Input data entitas pada use case “Register Account” mencakup data identitas pengguna yang digunakan untuk pembuatan akun baru pada sistem Digilibz. Data yang dimasukkan oleh user meliputi informasi dasar akun seperti nama lengkap, alamat email, dan kata sandi. Data email digunakan sebagai identitas unik pengguna di dalam sistem, sedangkan kata sandi akan diproses dan disimpan dalam bentuk terenkripsi untuk menjaga keamanan data. Selain data yang diinput langsung oleh user, sistem juga menghasilkan data tambahan berupa peran pengguna dan status akun yang secara default ditetapkan sebagai USER dan active. Seluruh data ini akan divalidasi untuk memastikan kelengkapan dan keunikan sebelum disimpan ke dalam database sistem.

### **3.2.2.1.2 Algorithm or Formula of Process**

Step performed UCS 01: Register Account

- a. User mengakses halaman registrasi pada sistem Digilibz.
- b. User mengisi form registrasi dengan data yang diperlukan.
- c. Sistem melakukan validasi terhadap kelengkapan dan format data yang diinput.
- d. Sistem memeriksa apakah email yang digunakan sudah terdaftar atau belum.
- e. Jika data valid dan email belum terdaftar:
  - Sistem mengenkripsi kata sandi pengguna.
  - Sistem menetapkan role pengguna sebagai USER dan status akun sebagai active.
  - Sistem menyimpan data akun pengguna ke dalam database.

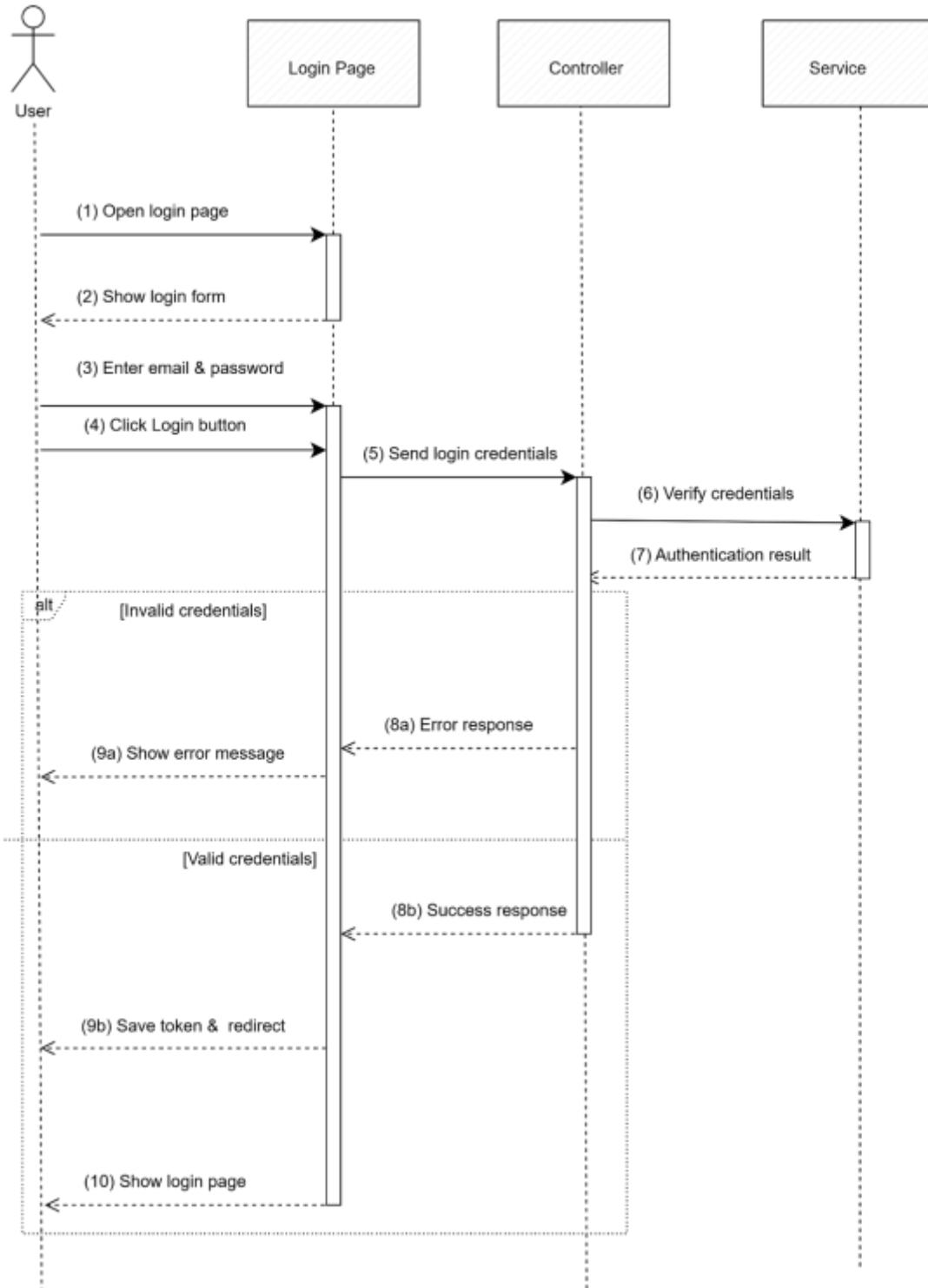
- f. Sistem menampilkan notifikasi bahwa proses registrasi berhasil.

### 3.2.2.1.3 Affected Data Entities

Use case “Register Account” mempengaruhi entitas data user dalam sistem Digilibz/ Proses ini menyebabkan penambahan satu data pengguna baru ke dalam database dengan status akun aktif. Data user yang tersimpan akan digunakan pada proses selanjutnya, seperti autentikasi login, pengajuan peminjaman buku, pengembalian buku, serta penerimaan notifikasi sistem. Sekain itu, sistem dapat membantu data notifikasi internal sebagai penanda bahwa akun pengguna berhasil dibuat.

### 3.2.2.2 Process 2

Aspek	Detail
<b>Use case name</b>	Login
<b>Description</b>	User atau Admin login ke sistem menggunakan email dan password
<b>Pre-condition</b>	User sudah terdaftar di sistem dengan status active
<b>Post-condition</b>	User berhasil login dan mendapat akses sesuai role (USER atau ADMIN)



### **3.2.2.2.1 Input Data Entities**

Input data entitas pada use case “Login” mencakup data kredensial pengguna yang digunakan untuk proses autentikasi ke dalam sistem Digilibz. Data yang diinput oleh user atau admin berupa alamat email dan kata sandi yang telah didaftarkan sebelumnya. Sistem menggunakan data tersebut untuk melakukan proses verifikasi identitas pengguna dengan mencocokkannya terhadap data yang tersimpan di dalam database. Selain data kredensial, sistem juga memanfaatkan data status akun untuk memastikan bahwa akun yang melakukan login berada dalam kondisi active. Data role pengguna digunakan untuk menentukan hak akses dan tampilan antarmuka yang sesuai setelah proses login berhasil.

### **3.2.2.2 Algorithm or Formula of Process**

Step performed UCS 02: Login

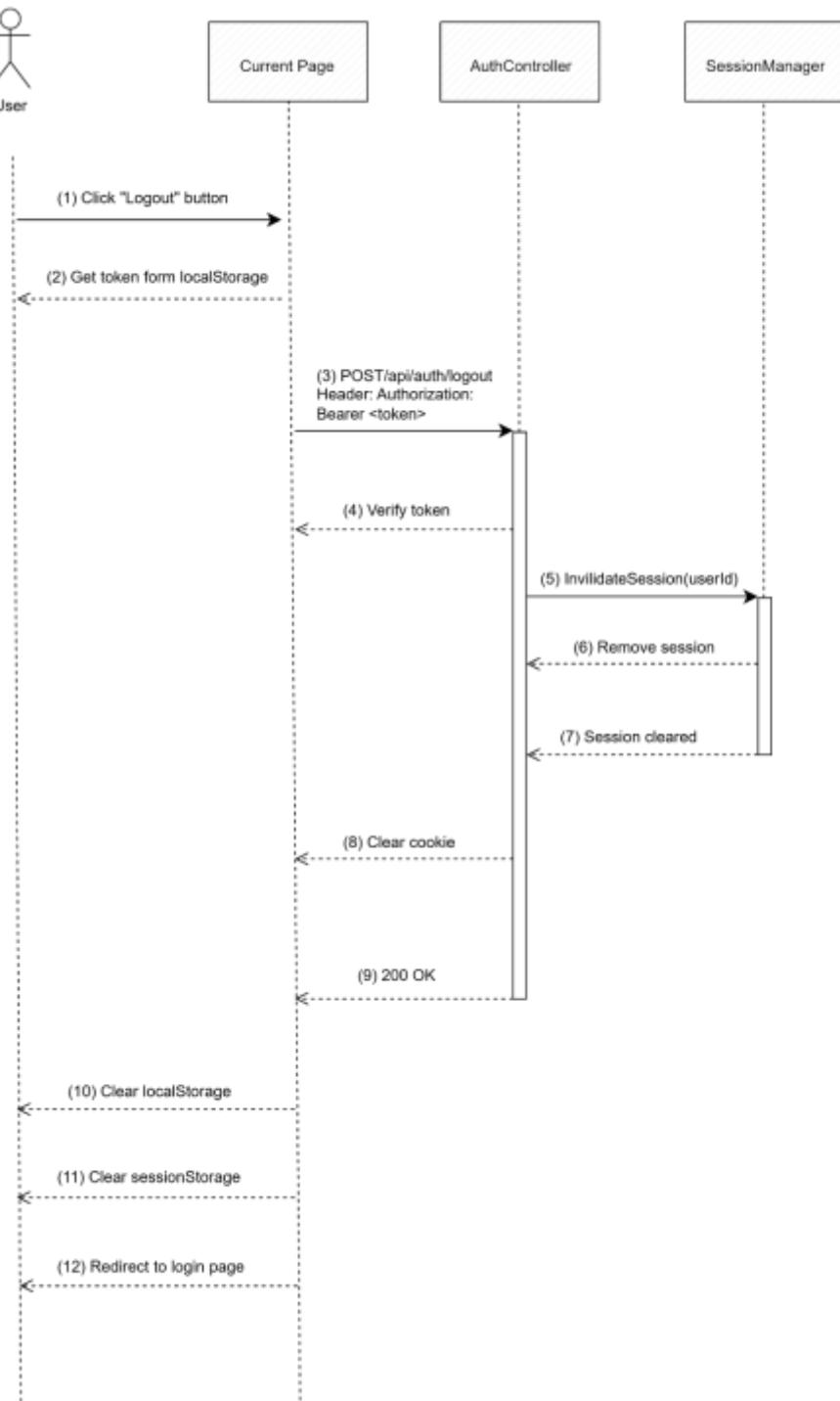
- a. User atau Admin mengakses halaman login sistem Digilibz.
- b. User atau Admin memasukkan alamat email dan kata sandi pada form login.
- c. Sistem melakukan validasi terhadap kelengkapan data yang diinput.
- d. Sistem memeriksa kecocokan email dan kata sandi dengan data pengguna di database.
- e. Sistem memverifikasi status akun pengguna.
- f. Jika kredensial valid dan status akun active:
  - Sistem mengautentikasi pengguna.
  - Sistem membuat sesi login.
  - Sistem mengarahkan pengguna ke halaman utama sesuai dengan role (USER atau ADMIN).
- g. Jika proses login gagal, sistem menampilkan pesan kesalahan kepada pengguna.

### **3.2.2.2.3 Affected Data Entities**

Use case “Login” mempengaruhi entitas data User dan Session dalam sistem Digilibz. Proses login tidak mengubah data utama pengguna, namun memperbarui status sesi autentikasi yang menandakan bahwa pengguna sedang aktif menggunakan sistem. Informasi role pengguna yang tersimpan pada entitas User digunakan untuk menentukan hak akses terhadap fitur sistem. Data sesi ini akan digunakan untuk mengontrol akses pengguna selama berada di dalam sistem hingga pengguna melakukan logout.

### **3.2.2.3 Process 3**

<b>Aspek</b>	<b>Detail</b>
<b>Use case name</b>	Logout
<b>Description</b>	User atau Admin keluar dari sistem dan mengakhiri session
<b>Pre-condition</b>	User sudah login dan memiliki session aktif
<b>Post-condition</b>	Session user dihapus dan user dikembalikan ke halaman login



### 3.2.2.3.1 Input Data Entities

Input data entitas pada use case “Logout” mencakup data sesi autentikasi yang sedang aktif pada pengguna. Data ini meliputi informasi identitas pengguna, token sesi, serta role pengguna yang tersimpan selama proses login berlangsung.

Sistem menggunakan data sesi tersebut untuk mengidentifikasi pengguna yang sedang login dan memastikan bahwa proses logout dilakukan pada sesi yang valid. Tidak terdapat input data manual dari pengguna selain aksi logout yang dipicu melalui antarmuka sistem.

### **3.2.2.3.2 Algorithm or Formula of Process**

Step performed UCS 03: Logout

- a. User atau Admin memilih opsi logout pada sistem Digilibz.
- b. Sistem memverifikasi keberadaan sesi login yang masih aktif.
- c. Sistem menghapus atau menonaktifkan sesi autentikasi pengguna.
- d. Sistem membersihkan data sesi yang tersimpan di sisi server maupun klien.
- e. Sistem mengakhiri proses autentikasi pengguna.
- f. Sistem mengarahkan pengguna kembali ke halaman login.

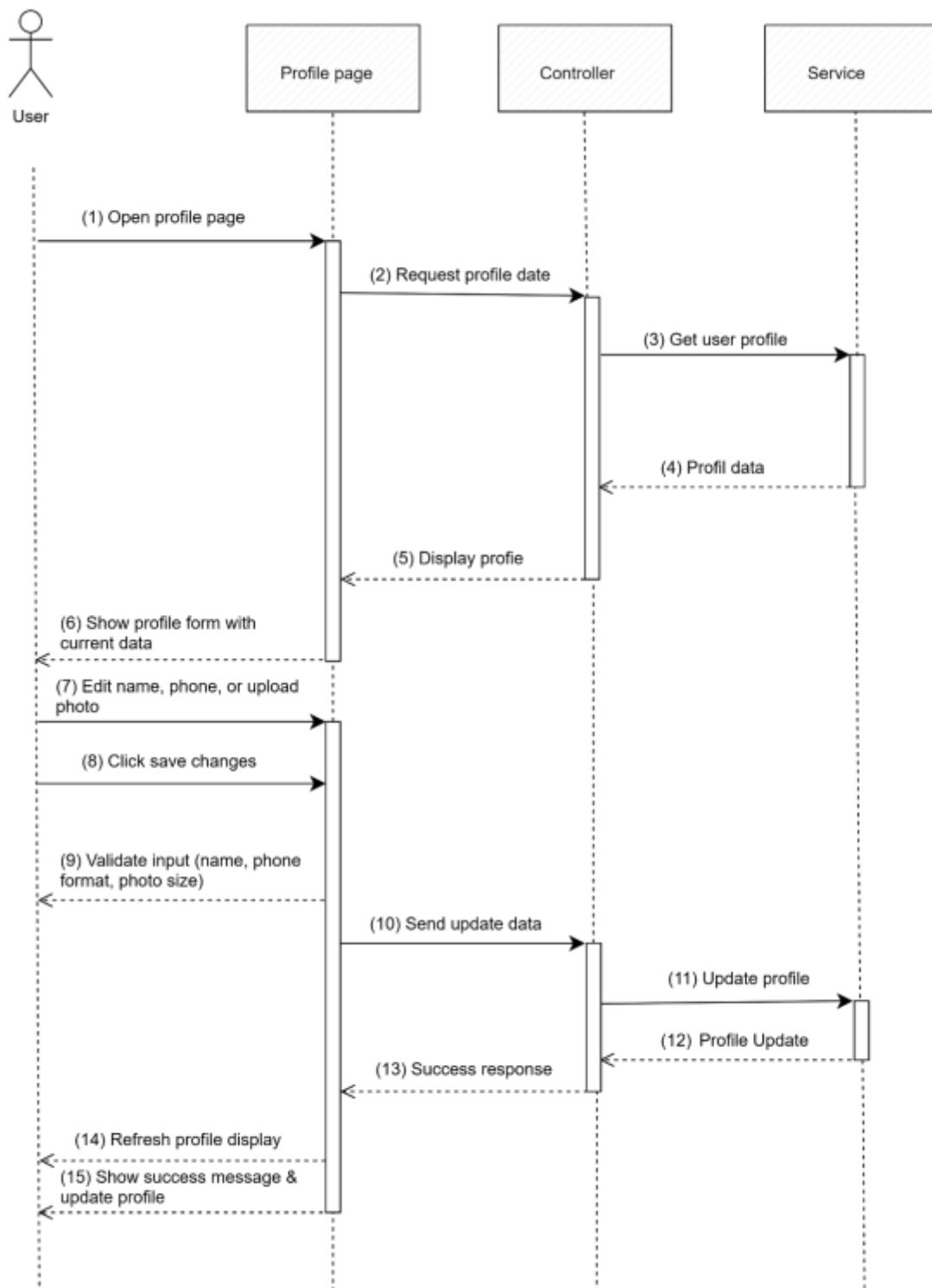
### **3.2.2.3.3 Affected Data Entities**

Use case “Logout” mempengaruhi entitas data Session dalam sistem Digilibz. Proses ini menyebabkan sesi autentikasi pengguna dihapus atau dinonaktifkan sehingga pengguna tidak lagi memiliki akses ke fitur sistem. Tidak terdapat perubahan pada data utama pengguna maupun data transaksi, dan seluruh data akun tetap tersimpan di dalam database sistem.

### **3.2.2.4 Process 4**

Aspek	Detail
<b>Use case name</b>	Update Profile
<b>Description</b>	User memperbarui informasi profil mereka sendiri
<b>Pre-condition</b>	User sudah login

<b>Post-condition</b>	Data profil user berhasil diupdate di database
-----------------------	--



#### **3.2.2.4.1 Input Data Entities**

Input data entitas pada use case “Update Profile” mencakup data profil pengguna yang akan diperbarui oleh user. Data tersebut meliputi informasi identitas dasar seperti nama lengkap, nomor telepon, dan foto profil. Sistem juga menggunakan data identitas user yang tersimpan dalam sesi login aktif untuk memastikan bahwa proses pembaruan hanya dapat dilakukan pada akun milik user yang sedang login. Seluruh data input divalidasi untuk memastikan kesesuaian format dan kelengkapan sebelum disimpan ke dalam database.

#### **3.2.2.4.2 Algorithm or Formula of Process**

Step performed UCS 04: Update Profile

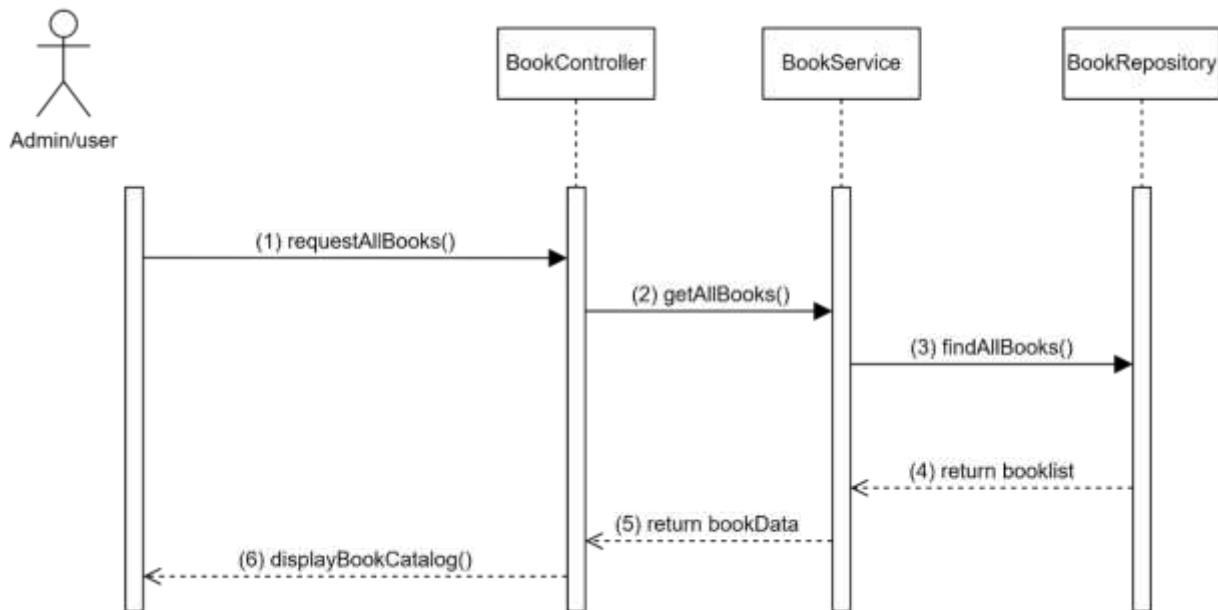
- a. User mengakses halaman profil pada sistem Digilibz.
- b. Sistem menampilkan data profil user yang tersimpan saat ini.
- c. User melakukan perubahan pada data profil yang diinginkan.
- d. User mengirimkan permintaan pembaruan data profil.
- e. Sistem memvalidasi data input yang diberikan oleh user.
- f. Jika data valid, sistem memperbarui data profil user di database.
- g. Sistem menyimpan perubahan dan memperbarui data sesi jika diperlukan.
- h. Sistem menampilkan notifikasi bahwa data profil berhasil diperbarui.

#### **3.2.2.4.3 Affected Data Entities**

Use case “Update Profile” mempengaruhi entitas data User dalam sistem Digilibz. Proses ini memperbarui atribut profil pengguna tanpa mengubah kredensial autentikasi atau role pengguna. Data lain seperti transaksi, review, dan notifikasi tidak terpengaruh oleh proses ini.

### 3.2.2.5 Process 5

Aspek	Detail
<b>Use case name</b>	View All Books
<b>Description</b>	User atau Admin melihat katalog lengkap semua buku yang tersedia di perpustakaan
<b>Pre-condition</b>	Tidak ada (dapat diakses tanpa login atau setelah login)
<b>Post-condition</b>	Daftar buku ditampilkan dengan pagination



#### 3.2.2.5.1 Input Data Entities

Input data entitas pada use case “View All Books” mencakup parameter permintaan data buku yang dikirimkan oleh pengguna ke sistem. Data ini meliputi informasi pagination seperti nomor halaman dan jumlah data per halaman. Sistem

juga dapat menerima parameter tambahan berupa pengurutan data buku sesuai konfigurasi default. Use case ini tidak memerlukan data autentikasi karena dapat diakses oleh user yang belum login maupun user yang sudah login.

### 3.2.2.5.2 Algorithm or Formula of Process

Step performed UCS 05: View All Books

- a. User atau Admin mengakses halaman katalog buku pada sistem Digilibz.
- b. Sistem menerima permintaan untuk menampilkan daftar buku.
- c. Sistem mengambil data buku dari database berdasarkan parameter pagination.
- d. Sistem memfilter buku yang berstatus aktif dan tidak terhapus (soft delete).
- e. Sistem mengurutkan data buku sesuai dengan pengaturan default.
- f. Sistem membagi data buku ke dalam halaman (pagination).
- g. Sistem menampilkan daftar buku beserta informasi dasar setiap buku kepada pengguna.

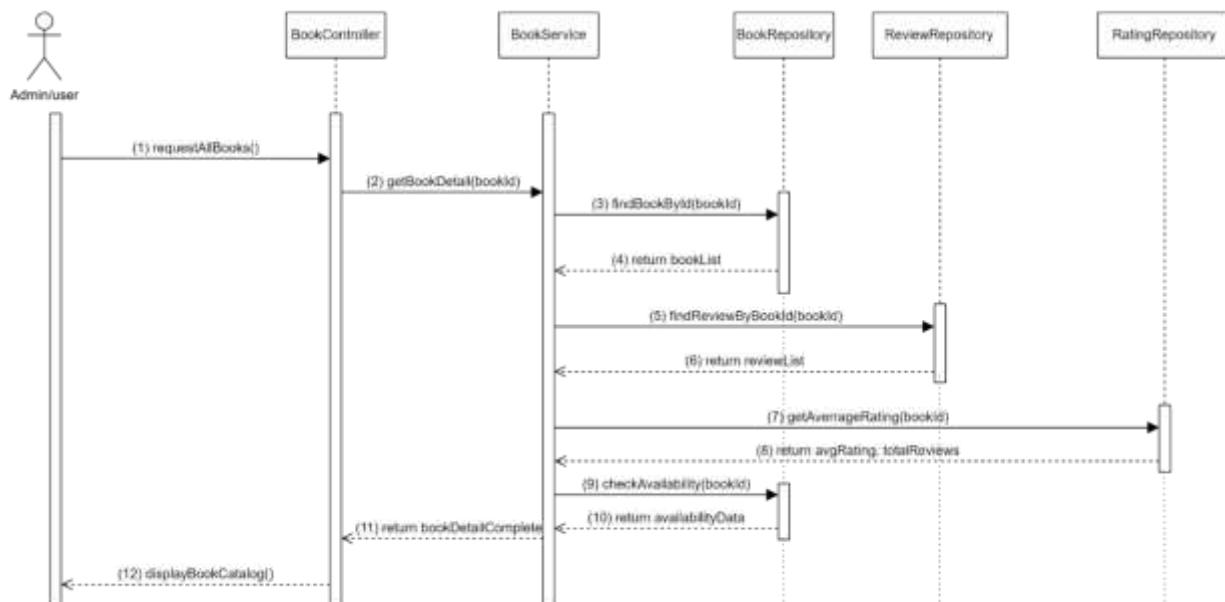
### 3.2.2.5.3 Affected Data Entities

Use case “View All Books” tidak menyebabkan perubahan pada data yang tersimpan di dalam sistem. Proses ini hanya membaca data dari entitas Book untuk ditampilkan kepada pengguna. Tidak terdapat proses insert, update, atau delete terhadap database.

### 3.2.2.6 Process 6

Aspek	Detail

<b>Use case name</b>	View Book Detail
<b>Description</b>	User atau Admin melihat informasi lengkap sebuah buku termasuk deskripsi, rating, dan review
<b>Pre-condition</b>	Buku dengan ID tersebut ada di database
<b>Post-condition</b>	Detail buku ditampilkan lengkap dengan review list



### 3.2.2.6.1 Input Data Entities

Input data entitas pada use case “View Book Detail” mencakup identitas buku yang akan ditampilkan detailnya. Data utama yang digunakan adalah Book ID yang dikirimkan melalui permintaan sistem ketika user atau admin memilih salah satu buku dari katalog. Sistem juga menggunakan parameter tambahan untuk pengambilan data terkait, seperti daftar review dan informasi rating buku. Use case ini dapat diakses tanpa autentikasi, selama buku yang diminta tersedia dan berstatus aktif di dalam database.

### **3.2.2.6.2 Algorithm or Formula of Process**

Step performed UCS 06: View Book Detail

- a. User atau Admin memilih salah satu buku dari daftar katalog.
- b. Sistem menerima permintaan detail buku berdasarkan Book ID.
- c. Sistem memverifikasi keberadaan buku di database dan memastikan status buku aktif.
- d. Sistem mengambil data detail buku, termasuk informasi umum dan deskripsi.
- e. Sistem mengambil daftar review dan rating yang terkait dengan buku tersebut.
- f. Sistem menghitung dan menampilkan nilai rata-rata rating buku.
- g. Sistem menampilkan seluruh informasi detail buku beserta daftar review kepada pengguna.

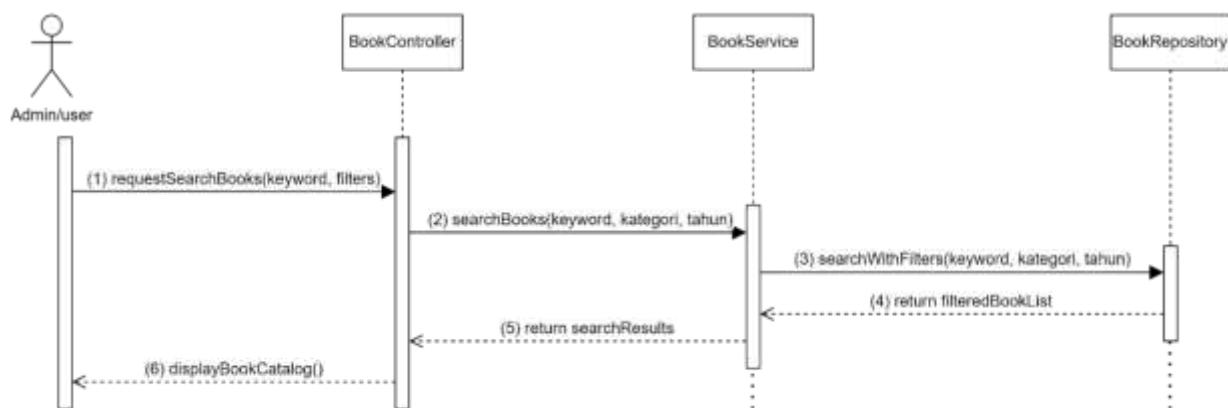
### **3.2.2.6.3 Affected Data Entities**

Use case “View Book Detail” hanya mempengaruhi proses pembacaan data pada sistem. Entitas data yang terlibat adalah Book dan Review, tanpa adanya perubahan terhadap data yang tersimpan di database. Tidak terjadi proses penambahan, pembaruan, maupun penghapusan data pada use case ini.

### **3.2.2.7 Process 7**

<b>Aspek</b>	<b>Detail</b>
<b>Use case name</b>	Search & Filter Books

<b>Description</b>	User atau Admin mencari buku dengan keyword dan memfilter berdasarkan kategori atau tahun
<b>Pre-condition</b>	User berada di halaman katalog buku
<b>Post-condition</b>	Hasil pencarian dan filter ditampilkan



### 3.2.2.7.1 Input Data Entities

Input data entitas pada use case “Search & Filter Books” mencakup parameter pencarian dan penyaringan data buku yang dimasukkan oleh user atau admin. Data ini meliputi kata kunci pencarian (keyword) untuk judul atau nama penulis, serta parameter filter tambahan seperti kategori buku dan tahun terbit. Sistem juga menerima parameter pengurutan dan pagination untuk mengelola tampilan hasil pencarian. Seluruh input digunakan sebagai kriteria untuk proses pencarian data buku yang tersimpan di database.

### 3.2.2.7.2 Algorithm or Formula of Process

Step performed UCS 07: Search & Filter Books

- User atau Admin mengakses halaman katalog buku.

- b. User atau Admin memasukkan keyword pencarian dan/atau memilih filter yang tersedia.
- c. Sistem menerima parameter pencarian dan filter dari pengguna.
- d. Sistem memvalidasi parameter input untuk memastikan format yang sesuai.
- e. Sistem melakukan pencarian data buku berdasarkan keyword menggunakan partial matching.
- f. Sistem menerapkan filter kategori dan tahun terbit sesuai dengan parameter yang dipilih.
- g. Sistem mengurutkan hasil pencarian berdasarkan pengaturan default atau pilihan pengguna.
- h. Sistem membagi hasil pencarian ke dalam halaman (pagination).
- i. Sistem menampilkan daftar buku yang sesuai dengan kriteria pencarian dan filter.

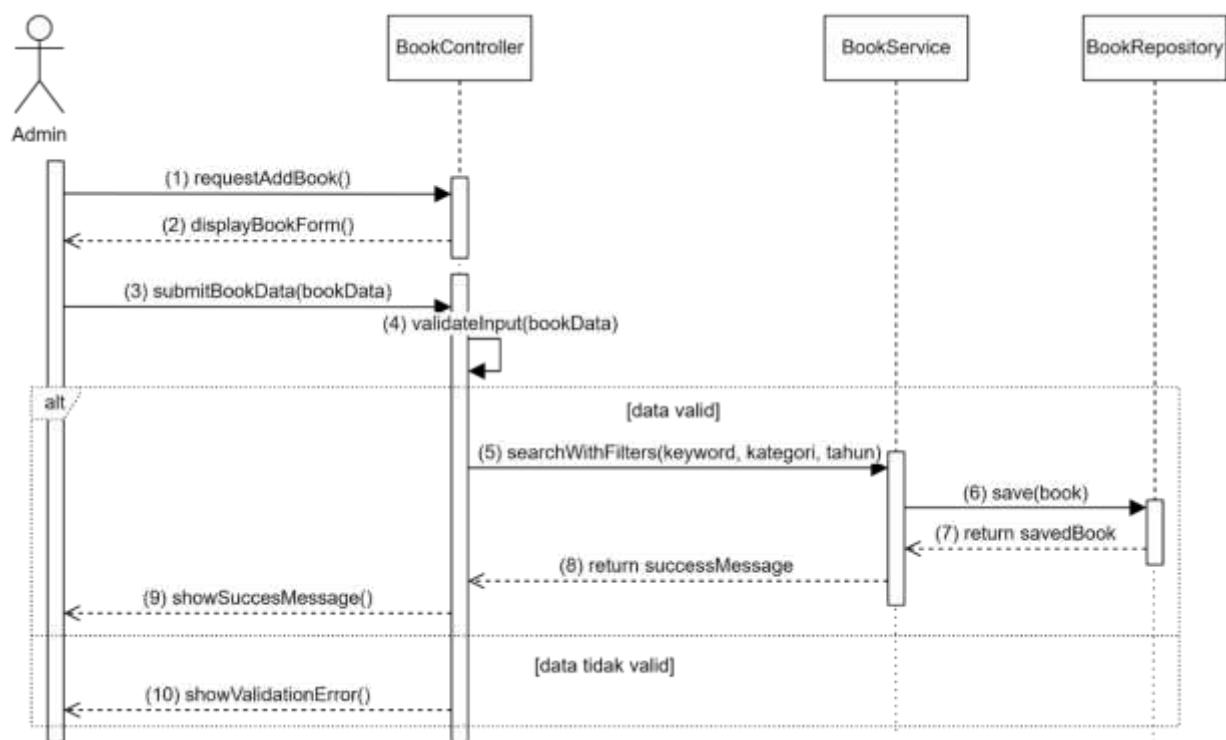
#### **3.2.2.7.3 Affected Data Entities**

Use case “Search & Filter Books” tidak menyebabkan perubahan pada data yang tersimpan di dalam sistem. Proses ini hanya melibatkan pembacaan data dari entitas Book dan atribut terkait lainnya. Tidak terdapat proses insert, update, maupun delete terhadap database.

#### **3.2.2.8 Process 8**

Aspek	Detail
<b>Use case name</b>	Add New Book

<b>Description</b>	Admin menambahkan buku baru ke sistem perpustakaan
<b>Pre-condition</b>	User login sebagai ADMIN
<b>Post-condition</b>	Buku baru berhasil tersimpan di database dan muncul di katalog



### 3.2.2.8.1 Input Data Entities

Input data entitas pada use case “Add New Book” mencakup data informasi buku yang akan ditambahkan oleh Admin ke dalam sistem Digilibz. Data tersebut meliputi judul buku, nama penulis, kategori, tahun terbit, ISBN, deskripsi buku, gambar sampul buku, jumlah stok total, jumlah buku yang tersedia, serta informasi biaya denda per hari. Sistem juga menggunakan data autentikasi Admin yang tersimpan dalam sesi login aktif untuk memverifikasi hak akses sebelum

proses penambahan buku dilakukan. Seluruh data input akan divalidasi untuk memastikan kelengkapan, format, dan batasan ukuran file yang sesuai dengan ketentuan sistem.

### **3.2.2.8.2 Algorithm or Formula of Process**

Step performed UCS 08: Add New Book

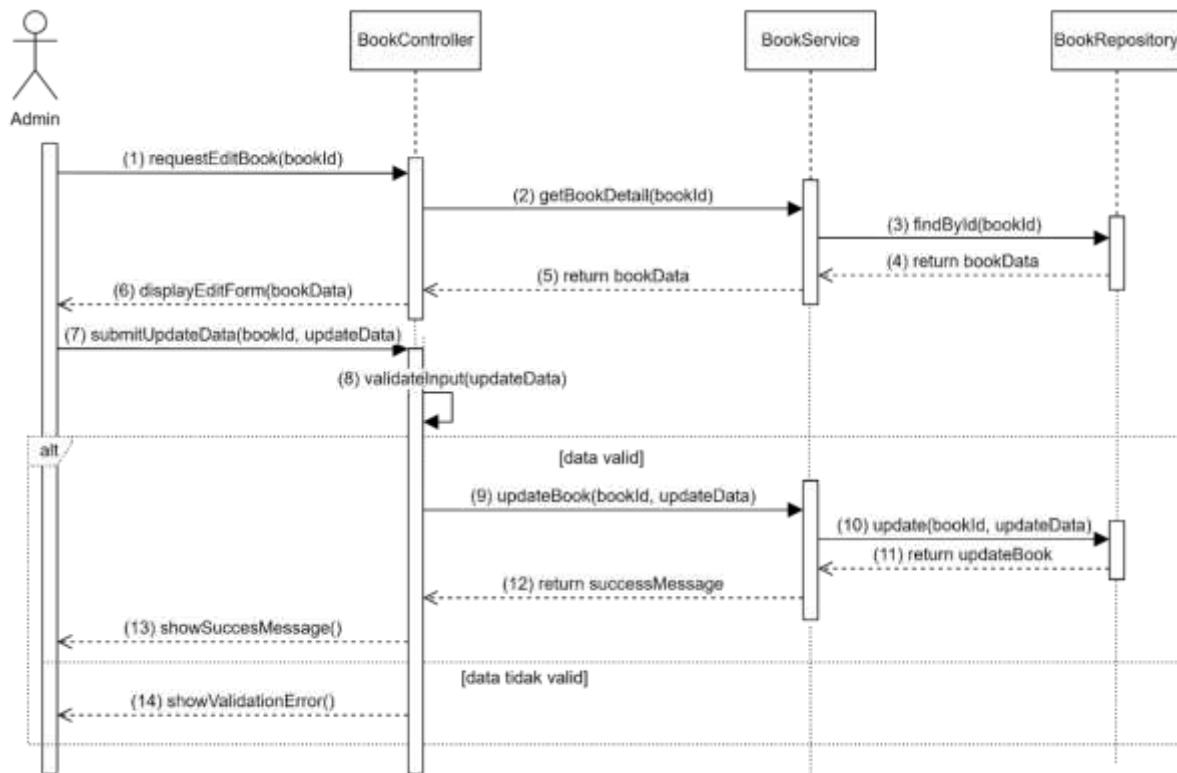
- a. Admin mengakses halaman manajemen buku pada sistem Digilibz.
- b. Sistem menampilkan form penambahan buku baru.
- c. Admin mengisi seluruh data buku yang diperlukan.
- d. Admin mengunggah gambar sampul buku sesuai format yang diizinkan.
- e. Admin mengirimkan permintaan penambahan buku ke sistem.
- f. Sistem memverifikasi hak akses Admin berdasarkan sesi login.
- g. Sistem memvalidasi seluruh data input buku.
- h. Jika data valid, sistem menyimpan data buku baru ke dalam database.
- i. Sistem menyimpan file gambar sampul buku ke media penyimpanan.
- j. Sistem menampilkan notifikasi bahwa buku berhasil ditambahkan dan memperbarui katalog buku.

### **3.2.2.8.3 Affected Data Entities**

Use case “Add New Book” mempengaruhi entitas data Book dalam sistem Digilibz. Proses ini menambahkan satu data buku baru ke database dan memperbarui informasi katalog buku yang dapat diakses oleh seluruh pengguna. Tidak terdapat perubahan pada entitas data lainnya seperti transaksi atau review.

### 3.2.2.9 Process 9

Aspek	Detail
<b>Use case name</b>	Edit Book Information
<b>Description</b>	Admin memperbarui informasi buku yang sudah ada
<b>Pre-condition</b>	User login sebagai ADMIN, buku dengan ID tersebut ada
<b>Post-condition</b>	Data buku berhasil diupdate di database



#### 3.2.2.9.1 Input Data Entities

Input data entitas pada use case “Edit Book Information” mencakup identitas buku yang akan diperbarui beserta data informasi baru yang dimasukkan oleh

Admin. Data utama yang digunakan adalah Book ID sebagai penanda buku yang akan diedit. Informasi yang dapat diperbarui meliputi judul buku, nama penulis, kategori, tahun terbit, ISBN, deskripsi, gambar sampul, jumlah stok total, jumlah buku yang tersedia, serta biaya denda per hari. Sistem juga menggunakan data autentikasi Admin dari sesi login aktif untuk memastikan bahwa hanya pengguna dengan hak akses ADMIN yang dapat melakukan perubahan data buku.

### **3.2.2.9.2 Algorithm or Formula of Process**

Step performed UCS 09: Edit Book Information

- a. Admin mengakses halaman manajemen buku pada sistem Digilibz.
- b. Admin memilih buku yang akan diperbarui dari daftar buku.
- c. Sistem mengambil dan menampilkan data buku berdasarkan Book ID.
- d. Admin melakukan perubahan pada informasi buku yang diperlukan.
- e. Admin mengirimkan permintaan pembaruan data buku.
- f. Sistem memverifikasi hak akses Admin.
- g. Sistem memvalidasi data input yang diperbarui.
- h. Jika data valid, sistem memperbarui data buku di database.
- i. Sistem menyimpan perubahan data dan, jika ada, memperbarui file gambar sampul.
- j. Sistem menampilkan notifikasi bahwa data buku berhasil diperbarui.

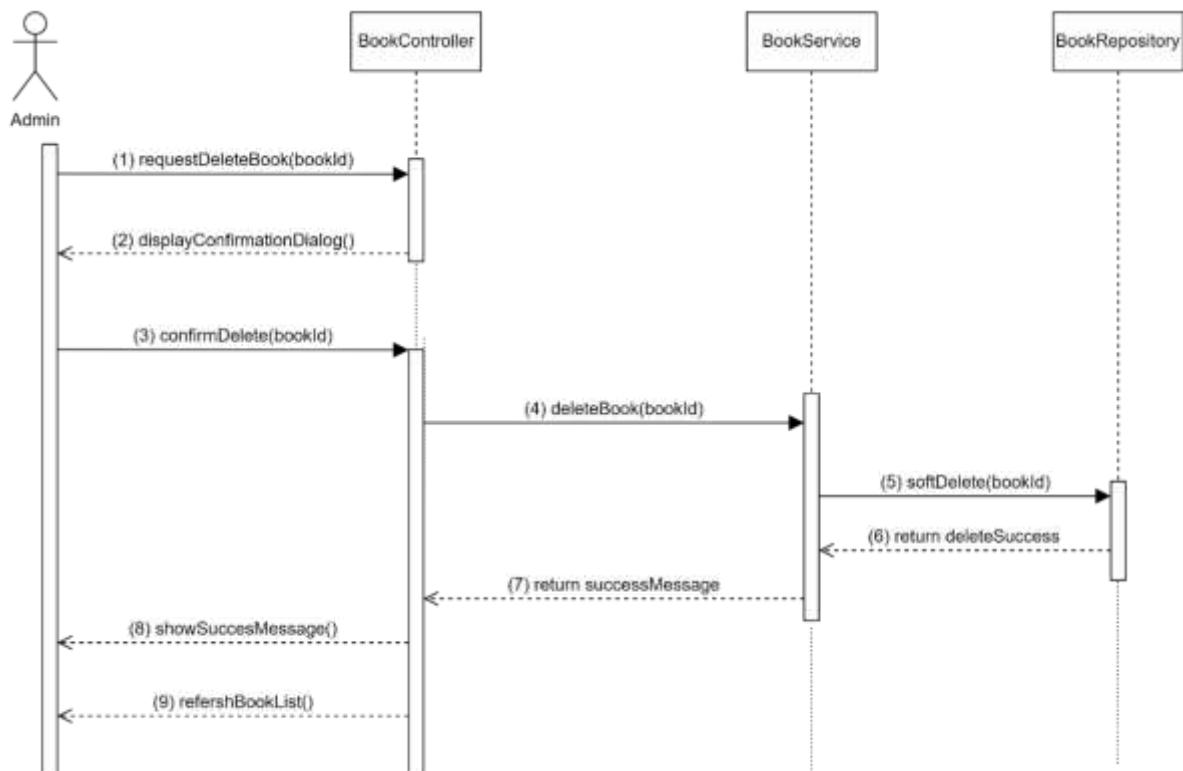
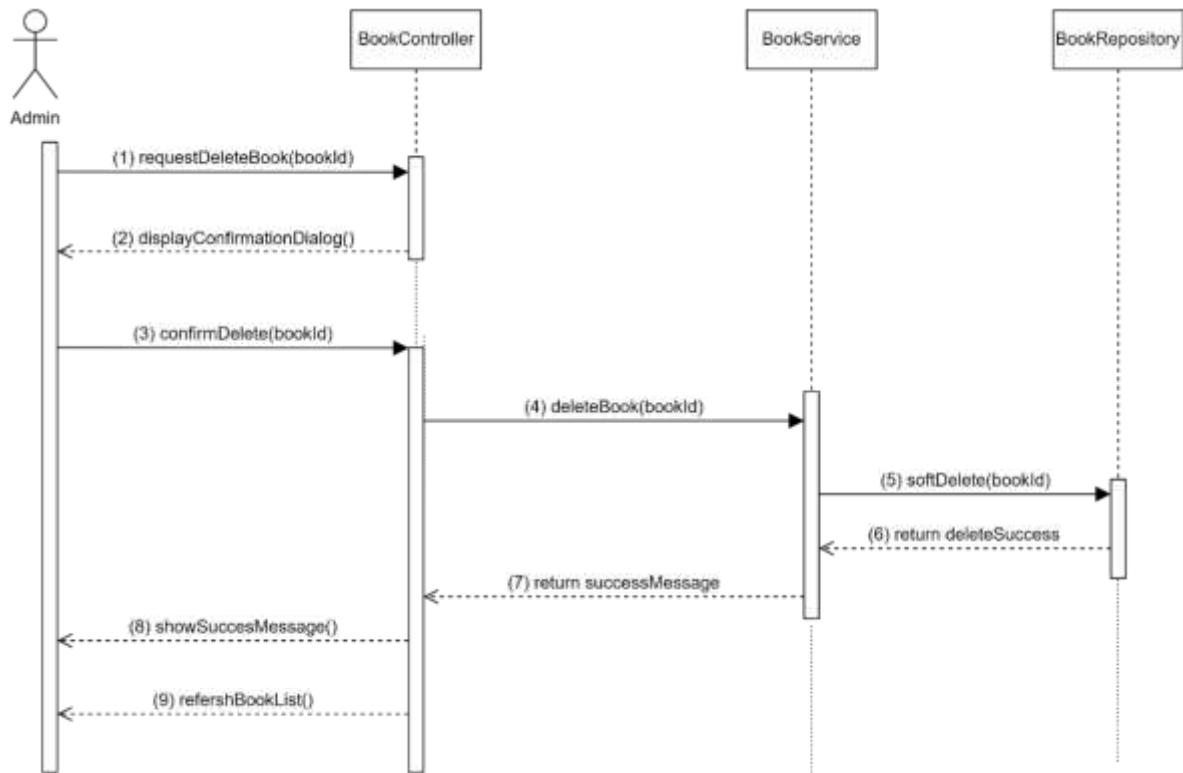
### **3.2.2.9.3 Affected Data Entities**

Use case “Edit Book Information” mempengaruhi entitas data Book dalam sistem Digilibz. Proses ini mengubah data buku yang sudah ada tanpa menghapus data

sebelumnya. Entitas data lain seperti transaksi, review, dan notifikasi tidak terpengaruh secara langsung oleh proses ini.

### 3.2.2.10 Process 10

Aspek	Detail
<b>Use case name</b>	Delete Book
<b>Description</b>	Admin menghapus buku dari sistem menggunakan soft delete
<b>Pre-condition</b>	User login sebagai ADMIN, buku ada di database
<b>Post-condition</b>	Buku ditandai sebagai deleted (soft delete) dan tidak muncul di katalog



### **3.2.2.10.1 Input Data Entities**

Input data entitas pada use case “Delete Book” mencakup identitas buku yang akan dihapus dari sistem. Data utama yang digunakan adalah Book ID sebagai penanda buku yang akan diproses. Sistem juga menggunakan data autentikasi Admin yang tersimpan dalam sesi login aktif untuk memverifikasi hak akses sebelum proses penghapusan dilakukan. Tidak terdapat input data tambahan selain aksi penghapusan yang dipicu oleh Admin.

### **3.2.2.10.2 Algorithm or Formula of Process**

Step performed UCS 10: Delete Book

- a. Admin mengakses halaman manajemen buku pada sistem Digilibz.
- b. Admin memilih buku yang akan dihapus dari daftar buku.
- c. Sistem menampilkan konfirmasi penghapusan buku.
- d. Admin mengonfirmasi tindakan penghapusan buku.
- e. Sistem memverifikasi hak akses Admin berdasarkan sesi login.
- f. Sistem memverifikasi keberadaan buku di database.
- g. Sistem menandai status buku sebagai deleted (soft delete).
- h. Sistem menyimpan perubahan status buku di database.
- i. Sistem memperbarui katalog buku sehingga buku tidak lagi ditampilkan kepada pengguna.
- j. Sistem menampilkan notifikasi bahwa buku berhasil dihapus dari katalog.

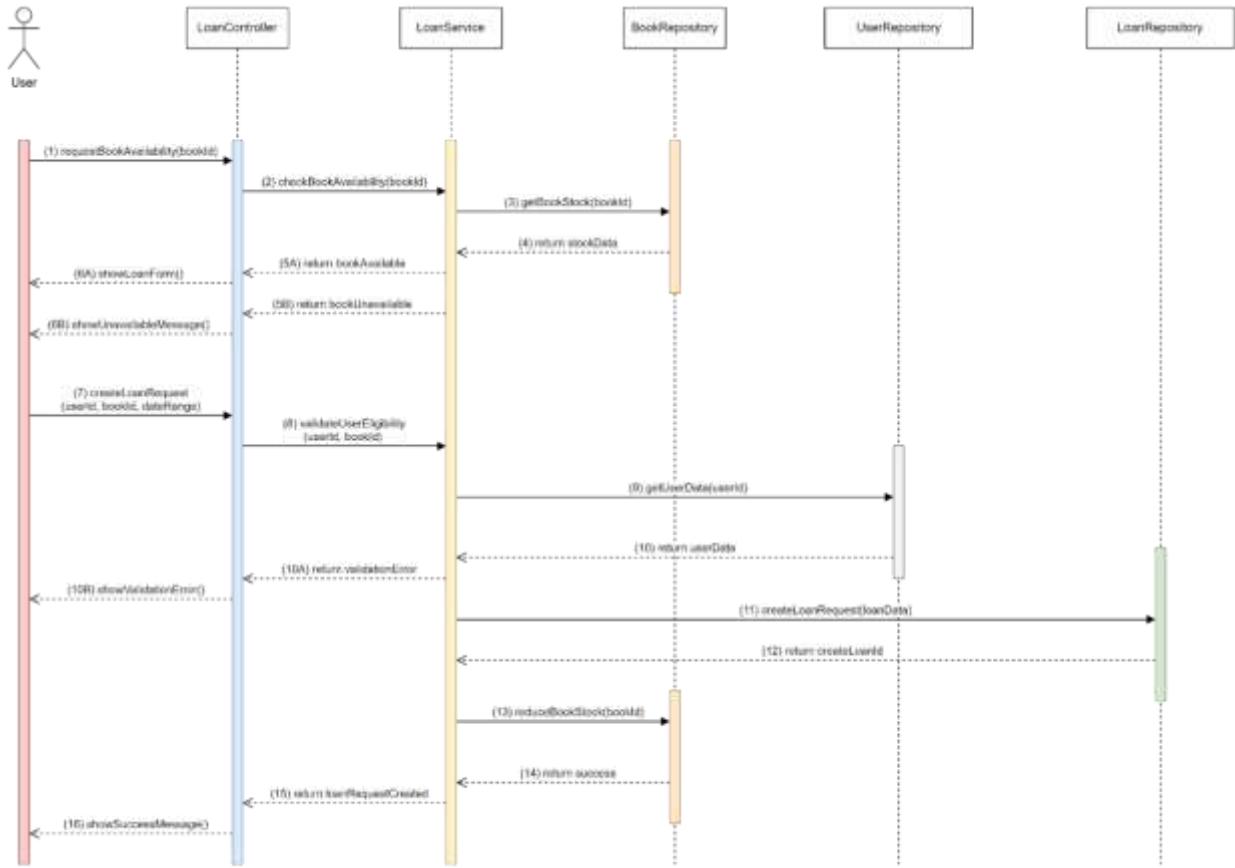
### **3.2.2.10.3 Affected Data Entities**

Use case “Delete Book” mempengaruhi entitas data Book dengan cara mengubah status aktif buku menjadi status terhapus (soft delete). Data buku

tidak dihapus secara permanen dari database, sehingga riwayat transaksi dan data terkait lainnya tetap terjaga. Tidak terdapat perubahan langsung pada entitas data transaksi atau review.

### 3.2.2.10 Process 11

Aspek	Detail
<b>Use case name</b>	Create Loan Request
<b>Description</b>	User membuat permintaan peminjaman buku dengan menentukan rentang tanggal
<b>Pre-condition</b>	User login, buku tersedia ( <code>available_copies &gt; 0</code> ), user belum melebihi quota (< 5 active loans)
<b>Post-condition</b>	Transaksi peminjaman dibuat dengan status PENDING menunggu approval admin



### 3.2.2.11.1 Input Data Entities

Input data entitas pada use case "Create Loan Request" mencakup identitas user yang akan meminjam buku. Data utama yang digunakan adalah user\_id sebagai penanda user yang login, book\_id sebagai identitas buku yang akan dipinjam, date\_from dan date\_to sebagai rentang waktu peminjaman. Sistem juga membutuhkan data ketersediaan buku (available\_copies) dan data peminjaman aktif user untuk verifikasi quota. Admin yang tersimpan dalam sistem akan menerima data loan request untuk verifikasi hak akses sebelum proses peminjaman dilakukan.

### 3.2.2.11.2 Algorithm or Formula of Process

Step performed UCS 11: Create Loan Request

- a. User memilih buku yang ingin dipinjam dari katalog.
- b. Sistem menampilkan ketersediaan buku dan informasi detail.
- c. User mengisi form peminjaman dengan rentang tanggal (date\_from dan date\_to).
- d. Sistem memvalidasi ketersediaan buku (available\_copies > 0).
- e. Sistem memvalidasi quota user (active\_loans < 5).
- f. Sistem memvalidasi durasi peminjaman (maksimal 30 hari).
- g. Sistem generate invoice code unik dengan format INV-YYYYMMDD-XXX.
- h. Sistem membuat record transaction baru dengan status PENDING.
- i. Sistem mengurangi available\_copies buku sebanyak 1.
- j. Sistem mengirim notifikasi ke user bahwa loan request berhasil dibuat.
- k. Sistem mengirim notifikasi ke admin bahwa ada loan request baru yang perlu di-approve.
- l. Sistem menampilkan pesan sukses dan invoice code ke user.

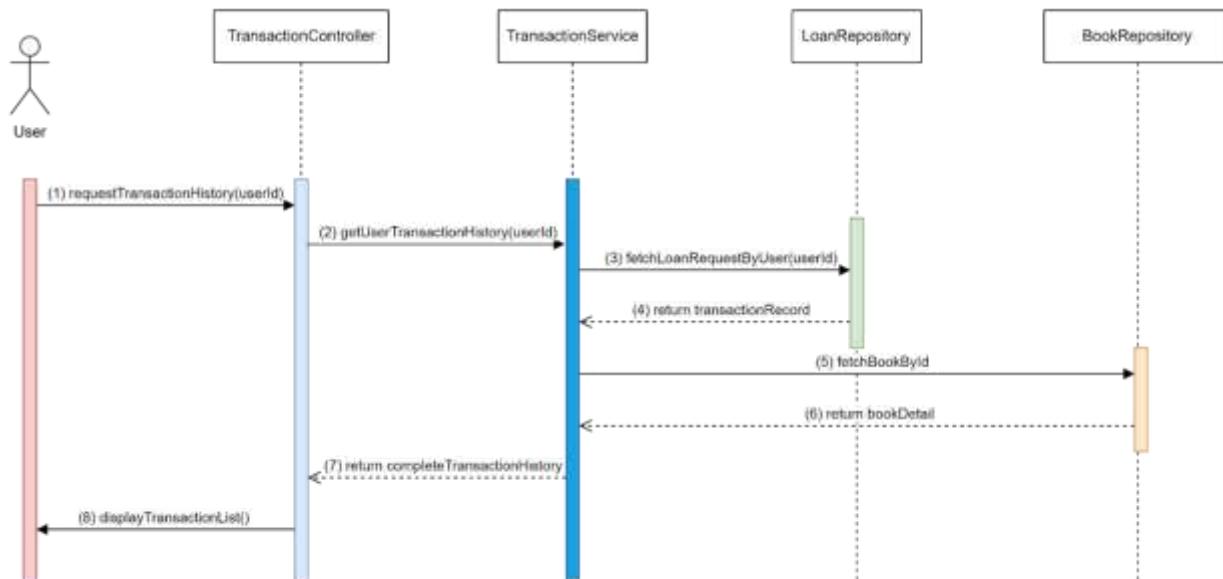
### 3.2.2.11.3 Affected Data Entities

Use case "Create Loan Request" mempengaruhi entitas data Transaction dengan cara menambah 1 record baru dengan status PENDING. Status buku menjadi aktif dengan available\_copies berkurang sebanyak 1 dari total sebelumnya, sehingga riwayat transaksi dan data terkait lainnya tetap terjaga. Tidak terdapat perubahan langsung pada entitas data transaksi atau review. Sistem juga menambah 2 record baru pada tabel Notification, satu untuk user dan satu untuk admin.

### 3.2.2.12 Process 12

Aspek	Detail

<b>Use case name</b>	View Transaction History
<b>Description</b>	User melihat riwayat semua transaksi peminjaman mereka
<b>Pre-condition</b>	User sudah login
<b>Post-condition</b>	Daftar transaksi user ditampilkan dengan filter dan pagination



### 3.2.2.12.1 Input Data Entities

Input data entitas pada use case "View Transaction History" mencakup identitas user yang login. Data utama yang digunakan adalah user\_id sebagai penanda user yang akan melihat riwayat transaksinya. Sistem juga dapat menerima parameter filter status (PENDING, APPROVED, REJECTED, RETURNED) dan parameter pagination (page\_number, page\_size) untuk menampilkan data secara terorganisir. Tidak diperlukan data tambahan selain aksi pengguna yang ingin melihat riwayat peminjaman mereka.

### **3.2.2.12.2 Algorithm or Formula of Process**

Step performed UCS 12: View Transaction History

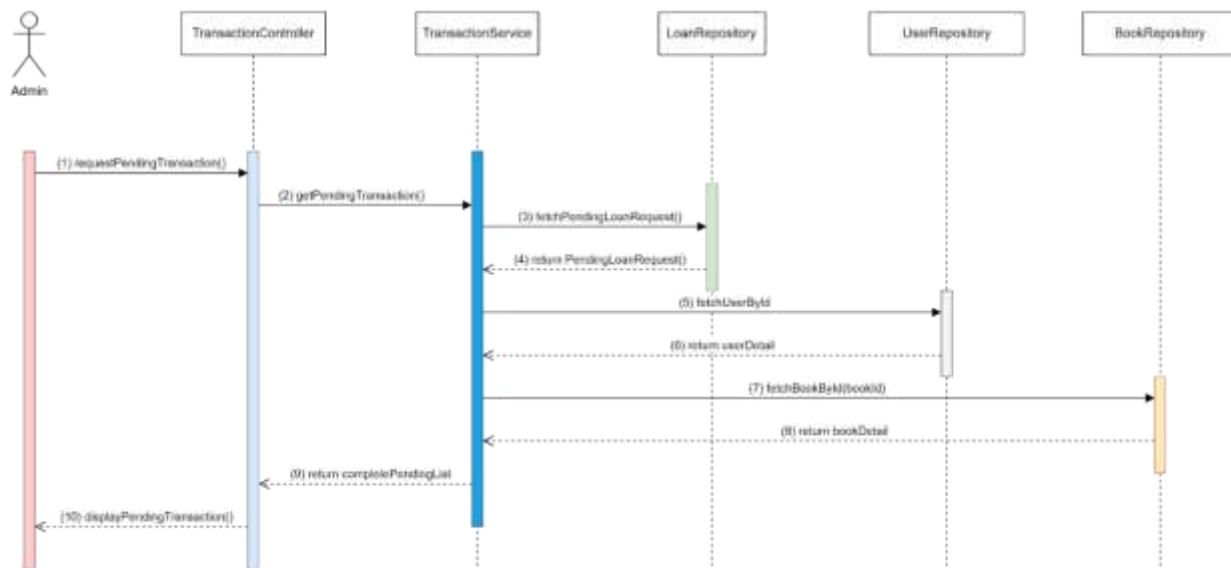
- a. User mengakses halaman "My Transactions" atau "Riwayat Peminjaman".
- b. Sistem memverifikasi user sudah login.
- c. Sistem query database untuk mengambil semua transaksi milik user (WHERE user\_id = current\_user).
- d. Sistem melakukan JOIN dengan tabel transaction\_items untuk mendapatkan book\_id.
- e. Sistem melakukan JOIN dengan tabel books untuk mendapatkan detail buku (title, author, cover).
- f. Sistem mengurutkan data berdasarkan tanggal pembuatan (created\_at DESC).
- g. Sistem menerapkan filter status jika user memilih filter tertentu.
- h. Sistem menerapkan pagination dengan menampilkan 20 transaksi per halaman.
- i. Sistem memformat data untuk display (format tanggal, currency, status badge).
- j. Sistem menampilkan list transaksi lengkap dengan informasi buku dan status.

### **3.2.2.12.3 Affected Data Entities**

Use case "View Transaction History" tidak mempengaruhi entitas data apapun karena merupakan operasi READ-only. Sistem hanya mengambil data dari tabel transactions, transaction\_items, dan books tanpa melakukan perubahan. Tidak terdapat perubahan pada status buku, data transaksi, atau entitas lainnya. Proses ini murni untuk menampilkan informasi kepada user tanpa modifikasi data.

### 3.2.2.13 Process 13

Aspek	Detail
<b>Use case name</b>	View Pending Transactions
<b>Description</b>	Admin melihat daftar semua permintaan peminjaman yang menunggu approval
<b>Pre-condition</b>	User login sebagai ADMIN
<b>Post-condition</b>	Daftar pending transactions ditampilkan



#### 3.2.2.13.1 Input Data Entities

Input data entitas pada use case "View Pending Transactions" mencakup identitas admin yang login. Data utama yang digunakan adalah `admin_id` dan verifikasi `role = 'ADMIN'` untuk otorisasi akses. Sistem secara otomatis menggunakan filter `status = 'PENDING'` untuk menampilkan hanya transaksi yang menunggu approval. Parameter tambahan seperti `sort_order` dan `pagination`

juga digunakan untuk mengatur tampilan data. Admin tidak perlu input data tambahan selain aksi untuk melihat daftar pending transactions.

### **3.2.2.13.2 Algorithm or Formula of Process**

Step performed UCS 13: View Pending Transactions

- a. Admin login dan mengakses admin dashboard.
- b. Sistem memverifikasi role admin (role = 'ADMIN').
- c. Admin mengklik menu "Pending Transactions" atau "Pending Approvals".
- d. Sistem query database untuk mengambil semua transaksi dengan status PENDING.
- e. Sistem melakukan JOIN dengan tabel users untuk mendapatkan data peminjam.
- f. Sistem melakukan JOIN dengan tabel transaction\_items dan books untuk detail buku.
- g. Sistem menghitung waiting time untuk setiap request (DATEDIFF dari created\_at).
- h. Sistem menandai request yang urgent (waiting time > 24 jam).
- i. Sistem mengurutkan data berdasarkan created\_at ASC (request lama di atas).
- j. Sistem menerapkan pagination dengan 50 transaksi per halaman.
- k. Sistem menampilkan summary statistics (total pending, urgent count).
- l. Sistem menampilkan tabel pending requests dengan action buttons (Approve/Reject).

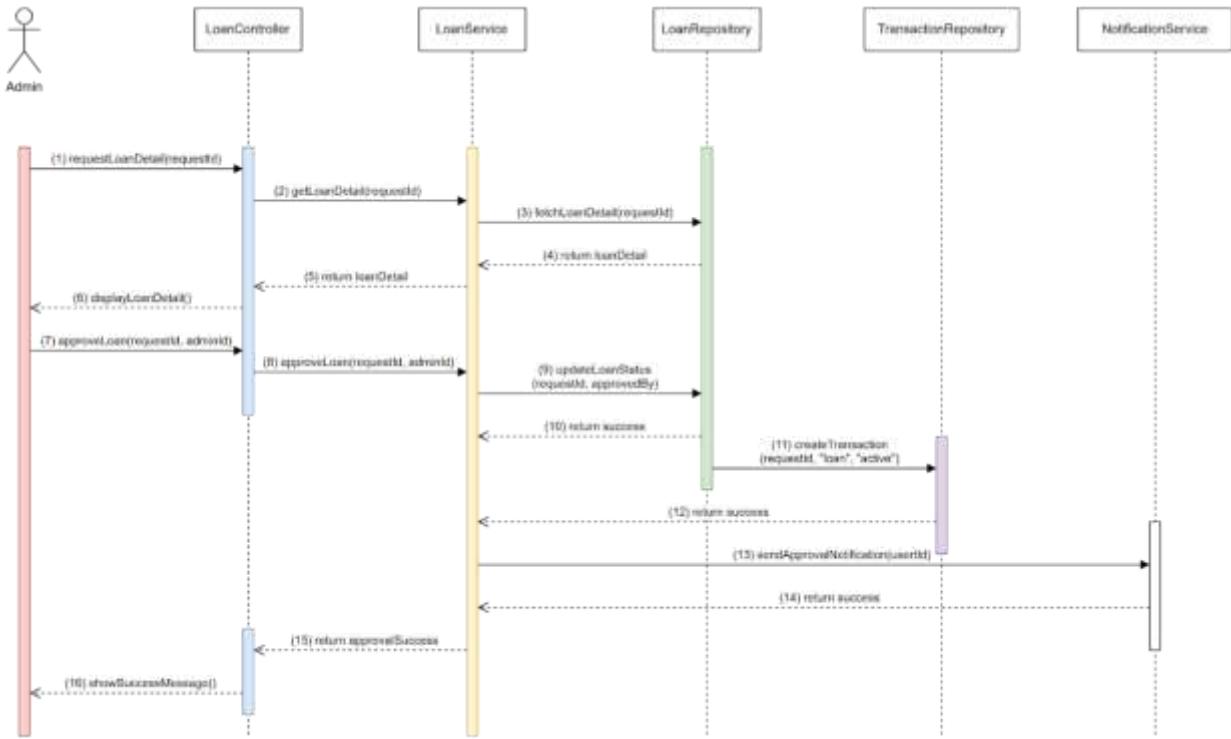
### **3.2.2.13.3 Affected Data Entities**

Use case "View Pending Transactions" tidak mempengaruhi entitas data apapun karena merupakan operasi READ-only. Sistem hanya mengambil data dari tabel transactions, users, transaction\_items, dan books tanpa melakukan perubahan.

Tidak terdapat perubahan pada status transaksi, data user, atau entitas lainnya. Proses ini murni untuk menampilkan informasi pending requests kepada admin tanpa modifikasi data.

### 3.2.2.14 Process 14

Aspek	Detail
<b>Use case name</b>	Approve Loan Request
<b>Description</b>	Admin menyetujui permintaan peminjaman dari user
<b>Pre-condition</b>	User login sebagai ADMIN, transaksi berstatus PENDING
<b>Post-condition</b>	Status transaksi berubah menjadi APPROVED, user dapat mengambil buku



### 3.2.2.14.1 Input Data Entities

Input data entitas pada use case "Approve Loan Request" mencakup identitas admin yang melakukan approval dan identitas transaksi yang akan disetujui. Data utama yang digunakan adalah admin\_id sebagai penanda admin yang login dan request\_id (transaction\_id) sebagai identitas transaksi yang akan di-approve. Sistem juga membutuhkan data transaksi lengkap untuk verifikasi status PENDING sebelum approval. Admin tidak perlu input data tambahan selain konfirmasi approval melalui tombol atau dialog konfirmasi.

### 3.2.2.14.2 Algorithm or Formula of Process

Step performed UCS 14: Approve Loan Request

- Admin membuka detail loan request dari daftar pending transactions.
- Sistem menampilkan detail lengkap transaksi (user, buku, tanggal, durasi).
- Admin menekan tombol "Approve" pada transaksi tersebut.

- d. Sistem menampilkan dialog konfirmasi "Apakah Anda yakin ingin menyetujui peminjaman ini?".
- e. Admin mengkonfirmasi approval dengan klik "Yes" atau "Confirm".
- f. Sistem memverifikasi status transaksi masih PENDING.
- g. Sistem memverifikasi admin memiliki hak akses (role = 'ADMIN').
- h. Sistem update status transaksi menjadi APPROVED di database.
- i. Sistem menyimpan informasi approved\_by (admin\_id) dan approved\_at (timestamp).
- j. Sistem membuat log activity untuk audit trail (transaction\_log).
- k. Sistem mengirim notifikasi ke user bahwa peminjaman disetujui.
- l. Sistem menampilkan pesan sukses ke admin.
- m. Sistem menghapus transaksi dari daftar pending dan memperbarui dashboard statistics.

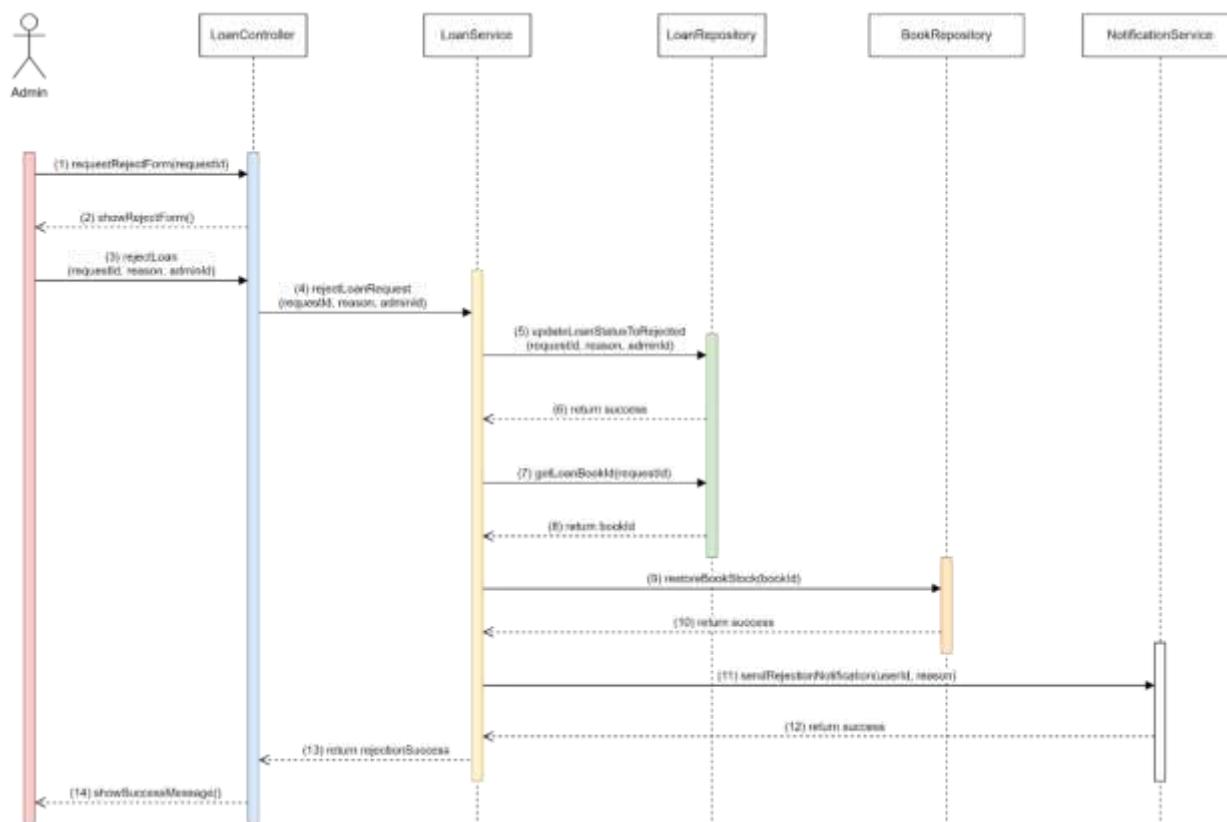
#### **3.2.2.14.3 Affected Data Entities**

Use case "Approve Loan Request" mempengaruhi entitas data Transaction dengan cara mengubah status dari PENDING menjadi APPROVED. Field approved\_by diisi dengan admin\_id dan approved\_at diisi dengan timestamp approval. Sistem juga menambah 1 record baru pada tabel Notification untuk user dengan pesan bahwa peminjaman telah disetujui. Tabel transaction\_log bertambah 1 record untuk audit trail. Tidak terdapat perubahan pada entitas Book karena available\_copies sudah dikurangi saat create loan request. Data user tetap tidak berubah kecuali mendapat notifikasi baru.

#### **3.2.2.15 Process 15**

Aspek	Detail
<b>Use case name</b>	Reject Loan Request

<b>Description</b>	Admin menolak permintaan peminjaman dengan memberikan alasan
<b>Pre-condition</b>	User login sebagai ADMIN, transaksi berstatus PENDING
<b>Post-condition</b>	Status transaksi berubah menjadi REJECTED, stok buku dikembalikan



### 3.2.2.15.1 Input Data Entities

Input data entities pada use case "Reject Loan Request" mencakup identitas permintaan peminjaman yang akan ditolak. Data utama yang digunakan adalah requestId sebagai penanda unik permintaan yang akan diproses, reason sebagai

alasan penolakan, dan adminId untuk autentikasi Admin yang melakukan tindakan penolakan.

### 3.2.2.15.2 Algorithm or Formula of Process

Step performed UCS 15: Reject Loan Request

- a. Admin mengakses form penolakan permintaan peminjaman.
- b. Admin memasukkan requestId dan reason untuk penolakan.
- c. Sistem memvalidasi requestId dan status permintaan (harus PENDING).
- d. Sistem mengupdate status permintaan menjadi REJECTED di database.
- e. Sistem memperbarui stok buku yang dikembalikan ke katalog.
- f. Sistem mengirim notifikasi penolakan kepada user.
- g. Sistem menampilkan pesan sukses kepada Admin.

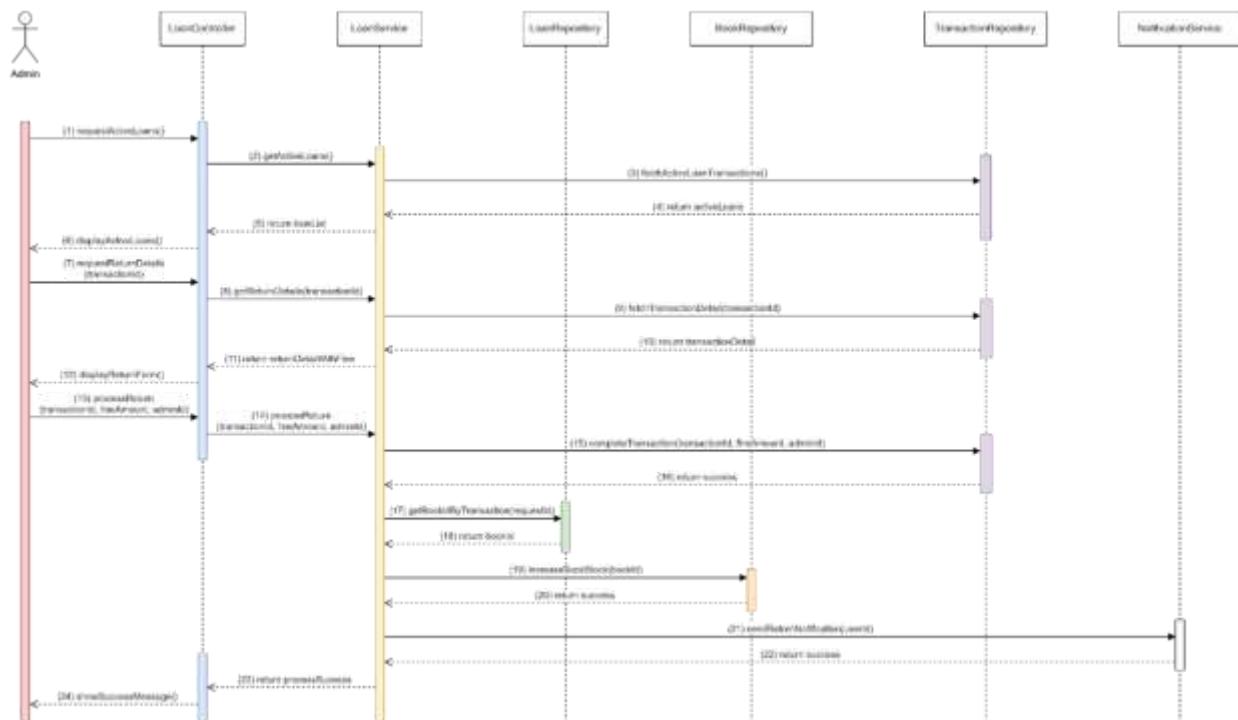
### 3.2.2.15.3 Affected Data Entities

Use case "Reject Loan Request" mempengaruhi entitas data Loan Request dengan cara mengubah status aktif request menjadi REJECTED secara permanen di database. Entitas Book juga terpengaruh karena stok buku dikembalikan ke katalog dan tersedia kembali untuk peminjaman. Entitas Notification terbuat baru untuk memberi tahu user bahwa permintaan mereka ditolak.

### 3.2.2.16 Process 16

Aspek	Detail
Use case name	Process Book Return

<b>Description</b>	Admin memproses pengembalian buku dan menghitung denda jika terlambat
<b>Pre-condition</b>	User login sebagai ADMIN, transaksi berstatus APPROVED
<b>Post-condition</b>	Status transaksi berubah RETURNED, stok buku dikembalikan, denda dihitung



### 3.2.2.16.1 Input Data Entities

Input data entities pada use case "Process Book Return" mencakup identitas transaksi pengembalian buku. Data utama yang digunakan adalah loanId sebagai penanda transaksi peminjaman yang akan diproses pengembaliannya, returnDate sebagai tanggal pengembalian aktual, dan adminId untuk autentikasi Admin yang memproses pengembalian.

### **3.2.2.16.2 Algorithm or Formula of Process**

Step performed UCS 16: Process Book Return

- a. Admin mengakses halaman pemrosesan pengembalian buku.
- b. Admin mencari transaksi peminjaman berdasarkan loanId atau user.
- c. Sistem menampilkan detail transaksi dengan status APPROVED.
- d. Admin mengonfirmasi pengembalian buku secara fisik.
- e. Sistem memverifikasi status transaksi dan menghitung denda (jika terlambat).
- f. Sistem mengupdate status transaksi menjadi RETURNED di database.
- g. Sistem menambah stok buku di katalog.
- h. Sistem mencatat denda pada data transaksi (jika ada).
- i. Sistem mengirim notifikasi pengembalian berhasil kepada user.
- j. Sistem menampilkan konfirmasi sukses kepada Admin.

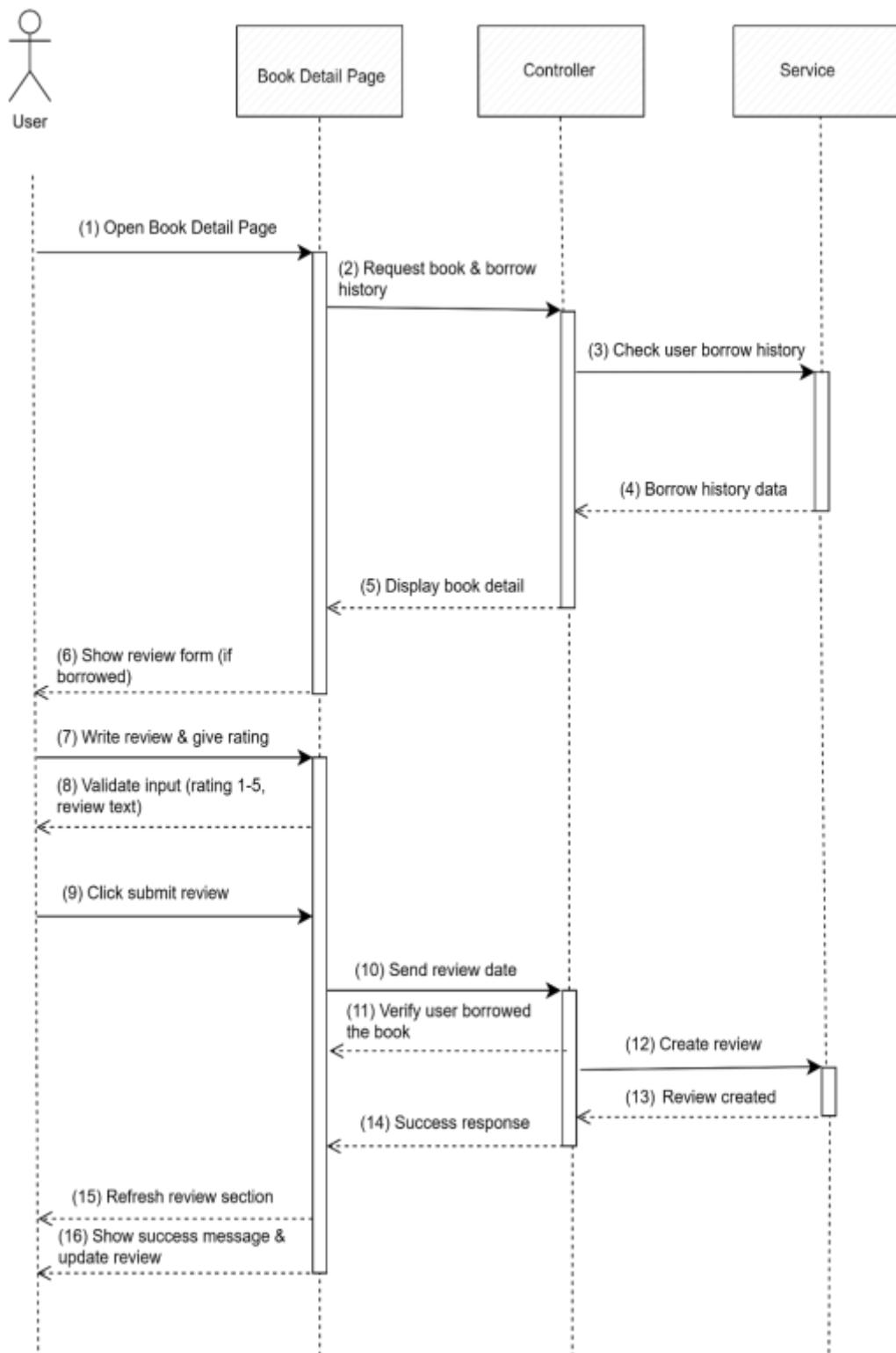
### **3.2.2.16.3 Affected Data Entities**

Use case "Process Book Return" mempengaruhi entitas data Loan Transaction dengan cara mengubah status aktif transaksi menjadi RETURNED dan mencatat returnDate di database. Entitas Book terpengaruh karena stok buku bertambah kembali setelah pengembalian. Entitas Transaction History terbuat baru untuk mencatat detail pengembalian. Entitas Notification terbuat untuk memberi tahu user bahwa pengembalian berhasil diproses.

### **3.2.2.17 Process 17**

Aspek	Detail
<b>Use case name</b>	Write Review

<b>Description</b>	User menulis review dan memberikan rating untuk buku yang pernah dipinjam
<b>Pre-condition</b>	User login, user pernah meminjam buku tersebut (transaksi RETURNED)
<b>Post-condition</b>	Review tersimpan dan muncul di halaman book detail



### **3.2.2.17.1 Input Data Entities**

Input data entities pada use case "Write Review" mencakup data ulasan yang ditulis user untuk buku yang telah dipinjam. Data utama yang digunakan adalah bookId sebagai identitas buku yang direview, userId untuk identifikasi user yang menulis review, rating dengan skala 1-5, dan reviewText sebagai isi ulasan user.

### **3.2.2.17.2 Algorithm or Formula of Process**

Step performed UCS 17: Write Review

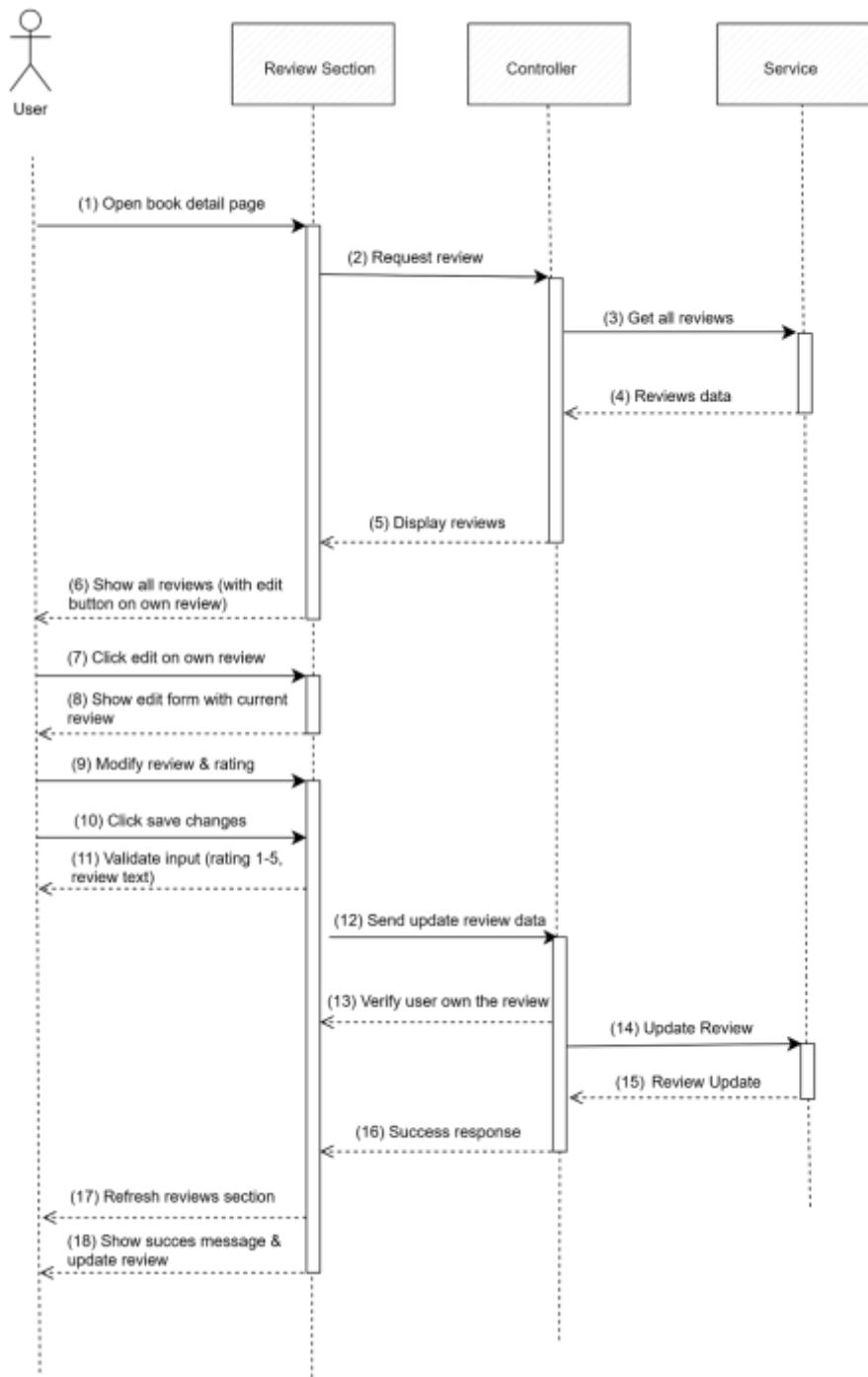
- a. User membuka halaman detail buku yang pernah dipinjam.
- b. Sistem memverifikasi riwayat peminjaman user untuk buku tersebut (status RETURNED).
- c. Sistem menampilkan form review pada halaman detail buku.
- d. User menulis review dan memberikan rating (1-5).
- e. User memvalidasi input (rating harus ada, review minimal 10 karakter).
- f. User mengklik tombol submit review.
- g. Sistem memverifikasi user telah meminjam buku tersebut.
- h. Sistem menyimpan data review ke database.
- i. Sistem menghitung ulang rata-rata rating buku.
- j. Sistem menampilkan review pada halaman detail buku.
- k. Sistem menampilkan pesan sukses kepada user.

### **3.2.2.17.3 Affected Data Entities**

Use case "Write Review" mempengaruhi entitas data Review dengan cara menambah data review baru ke database yang terkait dengan bookId dan userId. Entitas Book terpengaruh karena nilai averageRating dan totalReviews diperbarui berdasarkan review baru. Tidak terdapat perubahan langsung pada entitas transaksi, namun riwayat review tetap terjaga untuk referensi user.

### 3.2.2.18 Process 18

Aspek	Detail
<b>Use case name</b>	Edit Own Review
<b>Description</b>	User mengedit review yang pernah mereka buat
<b>Pre-condition</b>	User login, user sudah pernah membuat review untuk buku tersebut
<b>Post-condition</b>	Review berhasil diupdate dengan konten baru



### 3.2.2.18.1 Input Data Entities

Input data entitas pada use case “Edit Own Review” mencakup identitas review yang akan diubah serta konten baru yang dimasukkan oleh user. Data utama yang digunakan adalah Review ID, User ID dari sesi login aktif, serta konten review

baru termasuk rating (1–5 bintang) dan teks komentar. Sistem memverifikasi kepemilikan review agar user hanya dapat mengedit review yang mereka buat sendiri.

### 3.2.2.18.2 Algorithm or Formula of Process

Step performed UCS 18: Edit Own Review

- a. User mengakses halaman detail buku atau halaman profil untuk melihat review mereka.
- b. User memilih review yang ingin diedit.
- c. Sistem memverifikasi sesi login dan memastikan bahwa review tersebut milik user yang bersangkutan.
- d. Sistem menampilkan form edit review dengan konten lama yang sudah terisi.
- e. User memperbarui konten review dan rating sesuai keinginan.
- f. User mengirimkan permintaan pembaruan review ke sistem.
- g. Sistem memvalidasi konten baru untuk memastikan format dan panjang teks sesuai aturan.
- h. Sistem memperbarui data review di database dengan konten baru.
- i. Sistem menampilkan notifikasi bahwa review berhasil diperbarui dan memperbarui tampilan review di halaman buku.

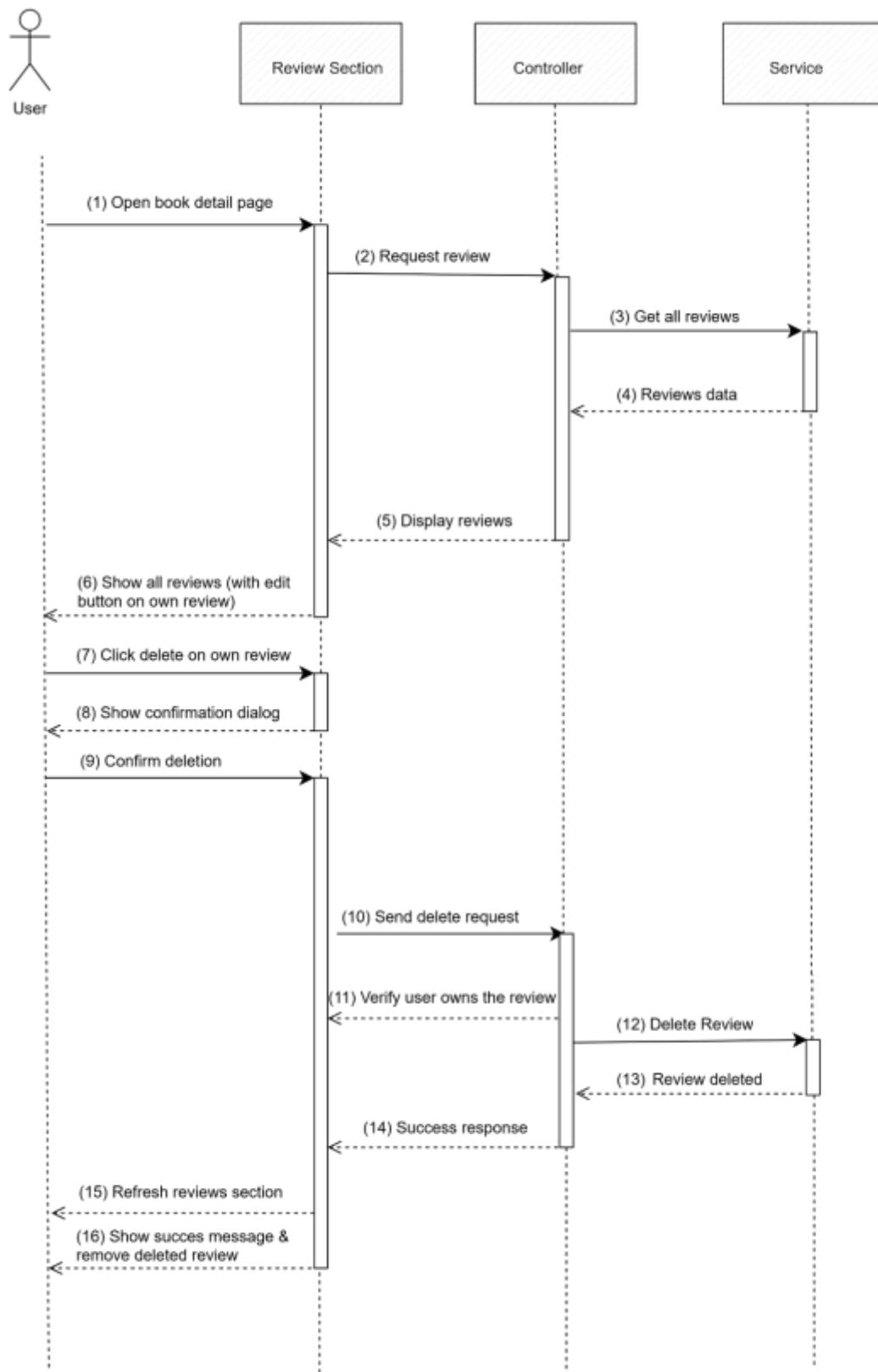
### 3.2.2.18.3 Affected Data Entities

Use case “Edit Own Review” mempengaruhi entitas data Review. Proses ini memperbarui konten review milik user di database. Entitas lain seperti Book, User, dan Transaction tidak mengalami perubahan langsung akibat use case ini.

### 3.2.2.19 Process 19

Aspek	Detail

<b>Use case name</b>	Delete Own Review
<b>Description</b>	User menghapus review yang pernah mereka buat
<b>Pre-condition</b>	User login, review milik user
<b>Post-condition</b>	Review dihapus dari database dan tidak muncul di book detail



### **3.2.2.19.1 Input Data Entities**

Input data entitas pada use case “Delete Own Review” mencakup identitas review yang ingin dihapus oleh user. Data utama yang digunakan adalah Review ID dan User ID dari sesi login aktif untuk memastikan bahwa review yang akan dihapus memang milik user yang bersangkutan. Sistem memverifikasi kepemilikan review sebelum melakukan proses penghapusan agar tidak terjadi penghapusan review milik user lain.

### **3.2.2.19.2 Algorithm or Formula of Process**

Step performed UCS 19: Delete Own Review

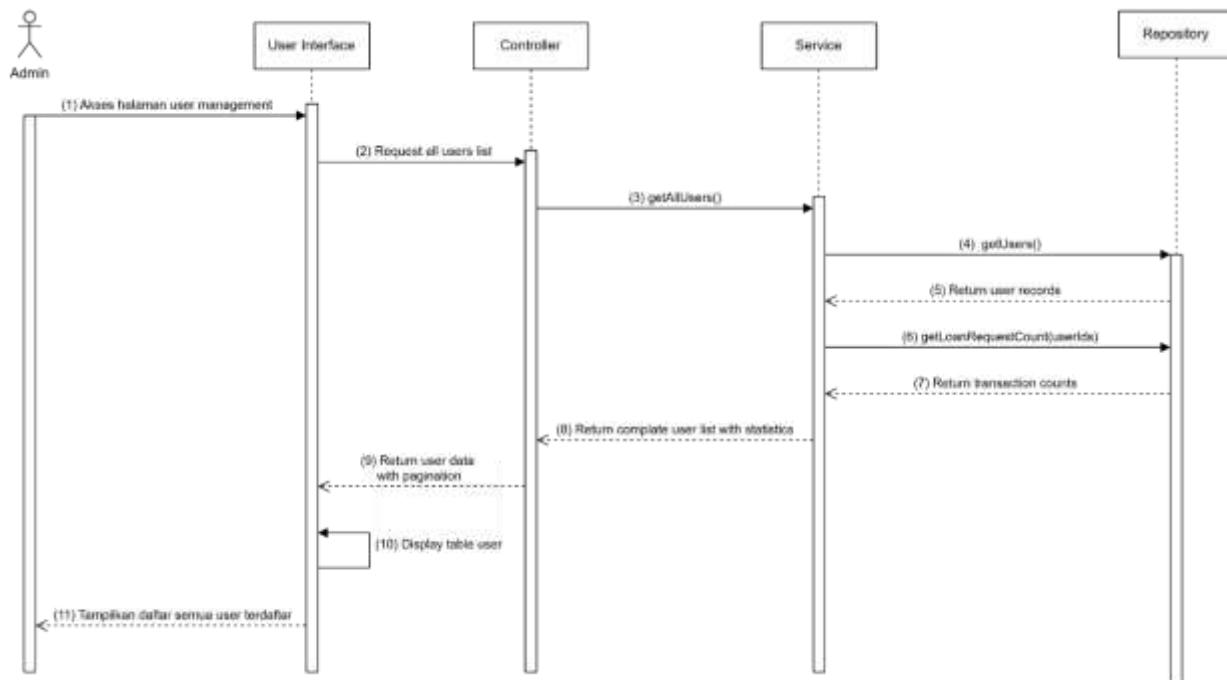
- a. User mengakses halaman detail buku atau halaman profil untuk melihat review mereka.
- b. User memilih review yang ingin dihapus.
- c. Sistem memverifikasi sesi login dan memastikan bahwa review tersebut milik user yang bersangkutan.
- d. Sistem menampilkan konfirmasi penghapusan review.
- e. User mengonfirmasi penghapusan review.
- f. Sistem menandai review sebagai dihapus atau langsung menghapus record dari database sesuai implementasi.
- g. Sistem memperbarui tampilan halaman book detail atau halaman profil, sehingga review yang dihapus tidak lagi ditampilkan.
- h. Sistem menampilkan notifikasi bahwa review berhasil dihapus.

### **3.2.2.19.3 Affected Data Entities**

Use case “Delete Own Review” mempengaruhi entitas data Review. Proses ini menghapus review milik user dari database sehingga tidak muncul lagi di halaman detail buku. Entitas lain seperti Book, User, dan Transaction tidak mengalami perubahan langsung akibat use case ini.

### 3.2.2.20 Process 20

Aspek	Detail
<b>Use case name</b>	View All Users
<b>Description</b>	Admin melihat daftar semua user terdaftar di sistem
<b>Pre-condition</b>	User login sebagai ADMIN
<b>Post-condition</b>	Daftar user ditampilkan dengan informasi ringkas



#### 3.2.2.20.1 Input Data Entities

Input data entitas pada use case “View All Users” mencakup permintaan daftar pengguna yang dikirimkan oleh Admin. Sistem menggunakan data autentikasi Admin untuk memverifikasi hak akses. Selain itu, sistem dapat menerima

parameter pagination untuk membatasi jumlah data user yang ditampilkan dalam satu halaman.

### 3.2.2.20.2 Algorithm or Formula of Process

Step performed UCS 20: View All Users

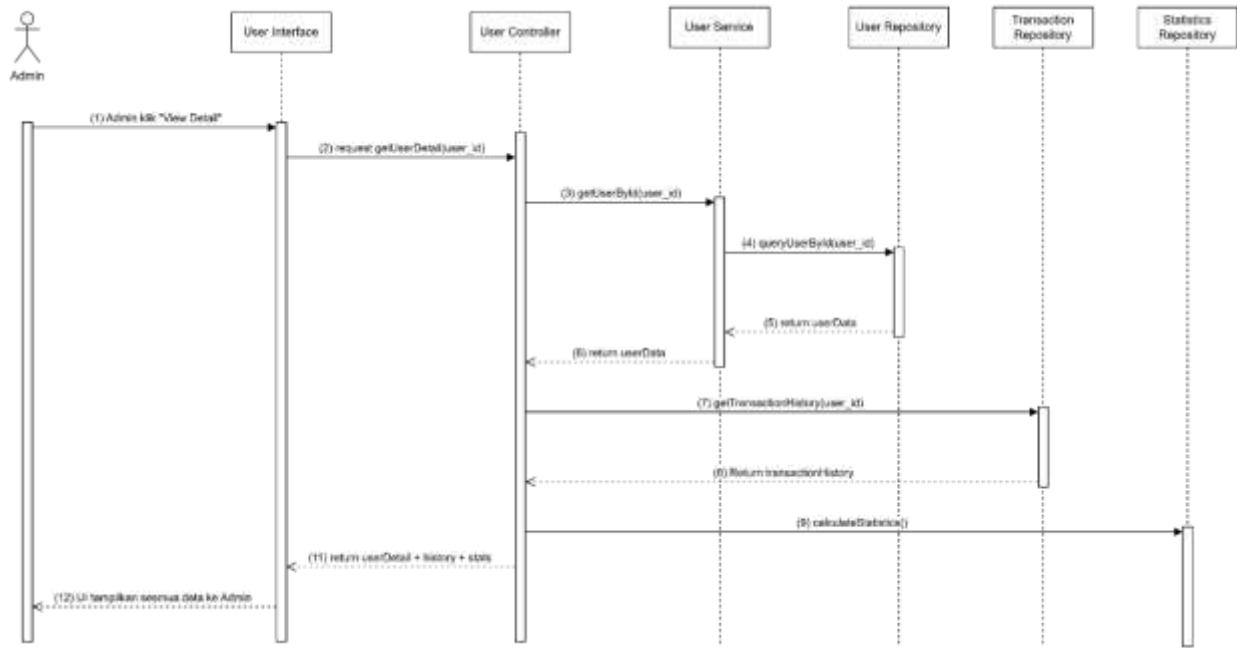
- a. Admin mengakses halaman daftar user pada sistem Digilibz.
- b. Sistem memverifikasi status login dan role Admin.
- c. Sistem mengambil data seluruh user yang terdaftar di database.
- d. Sistem memfilter user yang berstatus aktif dan tidak terhapus.
- e. Sistem membagi data user ke dalam halaman (pagination).
- f. Sistem menampilkan daftar user beserta informasi ringkas setiap user.

### 3.2.2.20.3 Affected Data Entities

Use case “View All Users” tidak mengubah data yang tersimpan di sistem. Proses ini hanya membaca data dari entitas User untuk ditampilkan kepada Admin. Tidak terdapat proses insert, update, atau delete pada database.

### 3.2.2.21 Process 21

Aspek	Detail
<b>Use case name</b>	View User Detail
<b>Description</b>	Admin melihat detail lengkap user termasuk riwayat peminjaman dan statistik
<b>Pre-condition</b>	User login sebagai ADMIN, user_id valid
<b>Post-condition</b>	Detail user dan activity log ditampilkan



### 3.2.2.21.1 Input Data Entities

Input data entitas pada use case “View User Detail” mencakup identitas pengguna yang akan dilihat detailnya, yaitu User ID yang dipilih oleh Admin. Sistem juga menggunakan data autentikasi Admin dari sesi login aktif untuk memastikan bahwa hanya pengguna dengan role ADMIN yang dapat mengakses informasi detail user. Data yang diambil meliputi informasi profil user, riwayat transaksi peminjaman, serta statistik aktivitas pengguna di dalam sistem.

### 3.2.2.21.2 Algorithm or Formula of Process

Step performed UCS 21: View User Detail

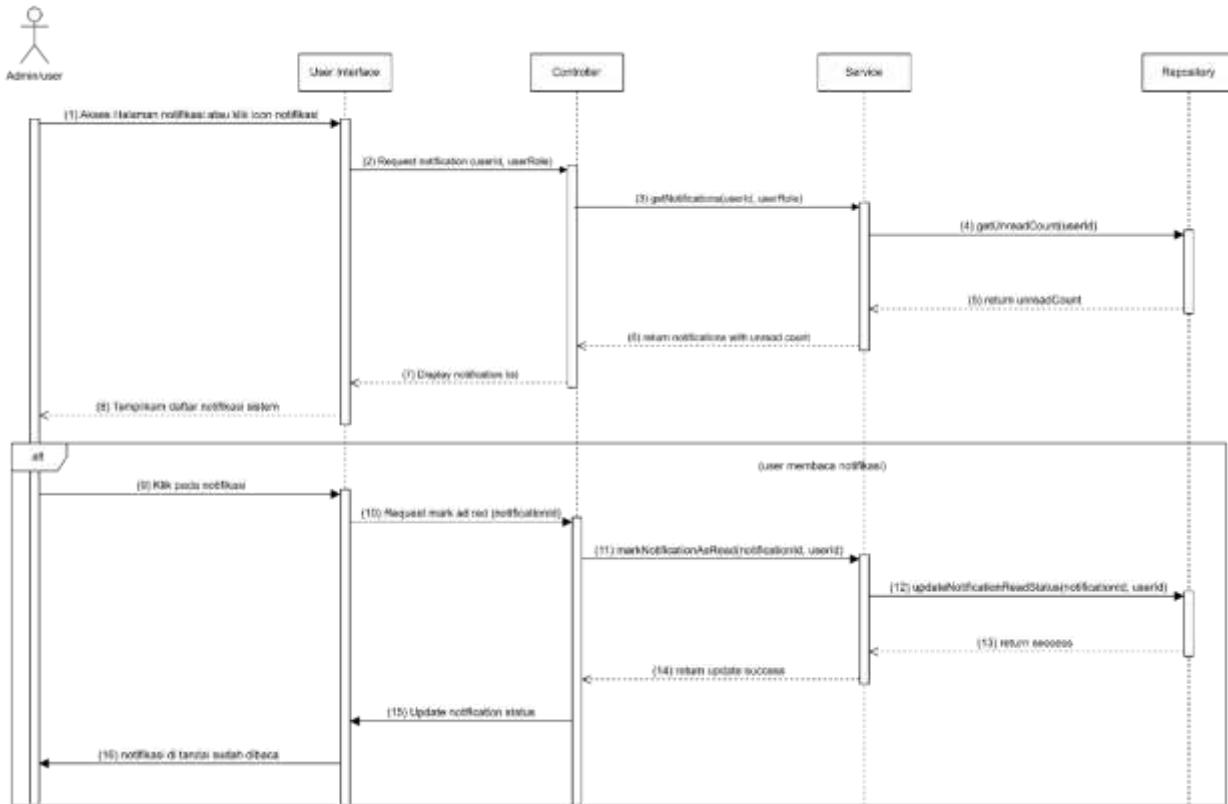
- Admin mengakses halaman manajemen pengguna pada sistem Digilibz.
- Admin memilih salah satu user dari daftar pengguna terdaftar.
- Sistem menerima permintaan detail user berdasarkan User ID.
- Sistem memverifikasi hak akses Admin.
- Sistem mengambil data profil user dari database.
- Sistem mengambil riwayat transaksi peminjaman user.
- Sistem menghitung statistik aktivitas user berdasarkan data transaksi.
- Sistem menampilkan detail user beserta riwayat peminjaman dan statistik aktivitas.

### **3.2.2.21.3 Affected Data Entities**

Use case “View User Detail” tidak menyebabkan perubahan pada data yang tersimpan di dalam sistem. Proses ini melibatkan pembacaan data dari entitas User, Transaction, dan TransactionItem untuk keperluan monitoring dan evaluasi oleh Admin. Tidak terdapat proses insert, update, atau delete terhadap database.

### **3.2.2.22 Process 22**

<b>Aspek</b>	<b>Detail</b>
<b>Use case name</b>	View Notifications
<b>Description</b>	User atau Admin melihat daftar notifikasi sistem
<b>Pre-condition</b>	User sudah login
<b>Post-condition</b>	Daftar notifikasi ditampilkan, badge unread count updated



### 3.2.2.22.1 Input Data Entities

Input data entitas pada use case “View Notifications” mencakup permintaan pengambilan daftar notifikasi yang dikirimkan oleh user atau admin ke sistem. Sistem menggunakan data identitas pengguna dan sesi login aktif untuk mengambil notifikasi yang sesuai dengan akun tersebut. Data notifikasi meliputi informasi judul notifikasi, isi pesan, waktu pengiriman, serta status notifikasi (sudah dibaca atau belum dibaca). Sistem juga menggunakan data jumlah notifikasi yang belum dibaca untuk memperbarui indikator badge pada antarmuka pengguna.

### 3.2.2.22.2 Algorithm or Formula of Process

Step performed UCS 22: View Notifications

- User atau Admin mengakses halaman notifikasi pada sistem Digilibz.

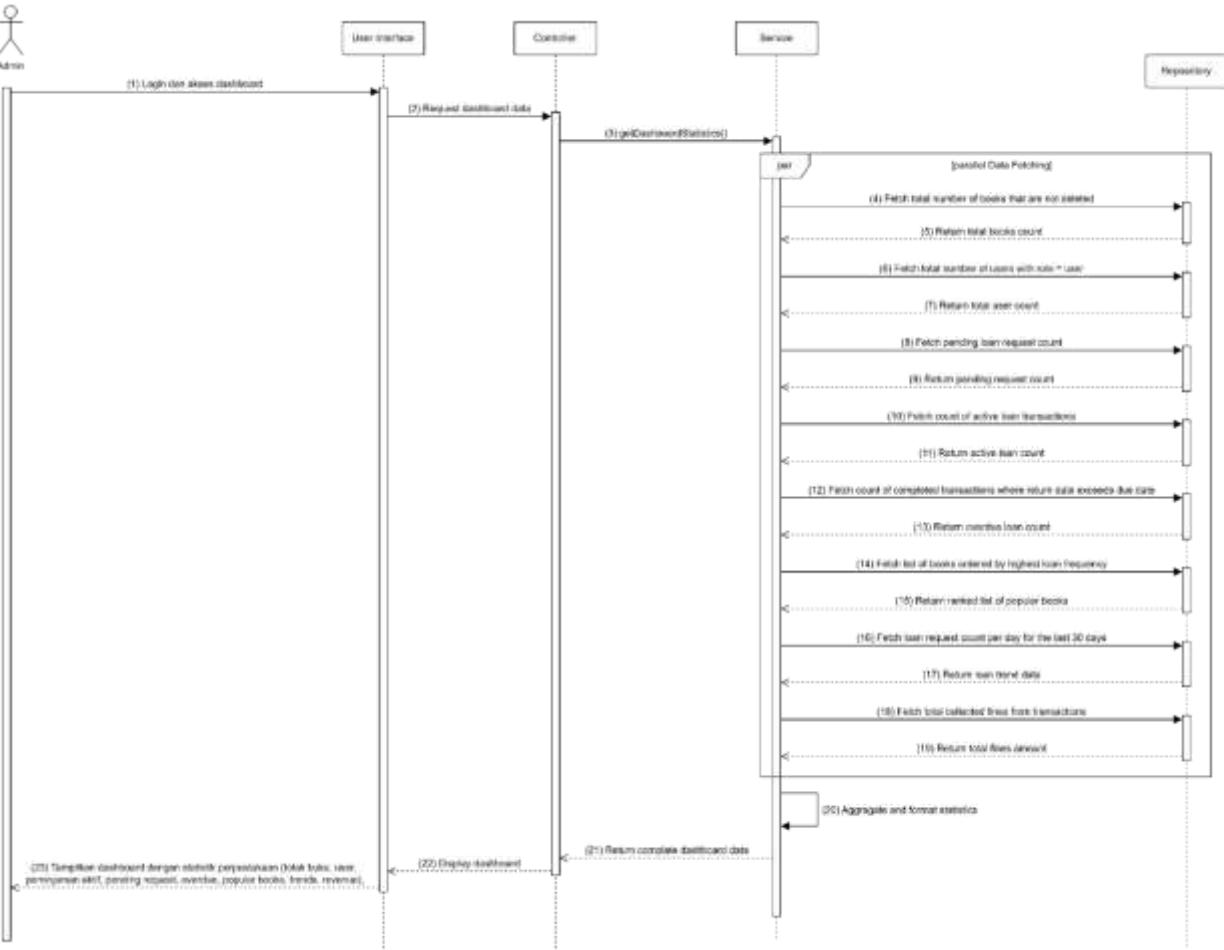
- b. Sistem memverifikasi status login pengguna.
- c. Sistem mengambil daftar notifikasi yang terkait dengan akun pengguna dari database.
- d. Sistem menghitung jumlah notifikasi yang belum dibaca.
- e. Sistem menyiapkan data notifikasi untuk ditampilkan.
- f. Sistem menampilkan daftar notifikasi kepada pengguna.
- g. Sistem memperbarui tampilan badge unread count sesuai dengan data notifikasi.

### **3.2.2.22.3 Affected Data Entities**

Use case “View Notifications” melibatkan entitas data Notification dalam sistem Digilibz. Proses ini pada umumnya hanya membaca data notifikasi tanpa mengubah isi notifikasi. Namun, sistem dapat memperbarui status tampilan badge unread count berdasarkan kondisi data notifikasi yang ada. Tidak terdapat perubahan data utama pada entitas lain di dalam database.

### **3.2.2.23 Process 23**

Aspek	Detail
<b>Use case name</b>	View Admin Dashboard
<b>Description</b>	Admin melihat dashboard dengan statistik dan ringkasan perpustakaan
<b>Pre-condition</b>	User login sebagai ADMIN
<b>Post-condition</b>	Dashboard dengan charts dan statistics ditampilkan



### 3.2.2.23.1 Input Data Entities

Input data entitas pada use case “View Admin Dashboard” mencakup permintaan akses dashboard yang dikirimkan oleh Admin ke sistem. Sistem menggunakan data autentikasi dan role pengguna yang tersimpan dalam sesi login aktif untuk memverifikasi bahwa pengguna memiliki hak akses ADMIN. Selain itu, sistem mengambil data statistik yang bersumber dari berbagai entitas seperti jumlah buku, jumlah pengguna, transaksi peminjaman aktif, permintaan peminjaman yang menunggu persetujuan, serta data ringkasan lainnya yang diperlukan untuk ditampilkan pada dashboard.

### **3.2.2.23.2 Algorithm or Formula of Process**

Step performed UCS 23: View Admin Dashboard

- a. Admin mengakses halaman dashboard pada sistem Digilibz.
- b. Sistem memverifikasi status login dan role Admin.
- c. Sistem mengambil data statistik dari database.
- d. Sistem menghitung ringkasan data perpustakaan berdasarkan informasi yang tersedia.
- e. Sistem mengolah data menjadi bentuk statistik dan ringkasan.
- f. Sistem menyiapkan data untuk ditampilkan dalam bentuk grafik dan angka ringkasan.
- g. Sistem menampilkan dashboard Admin yang berisi charts dan statistik perpustakaan.

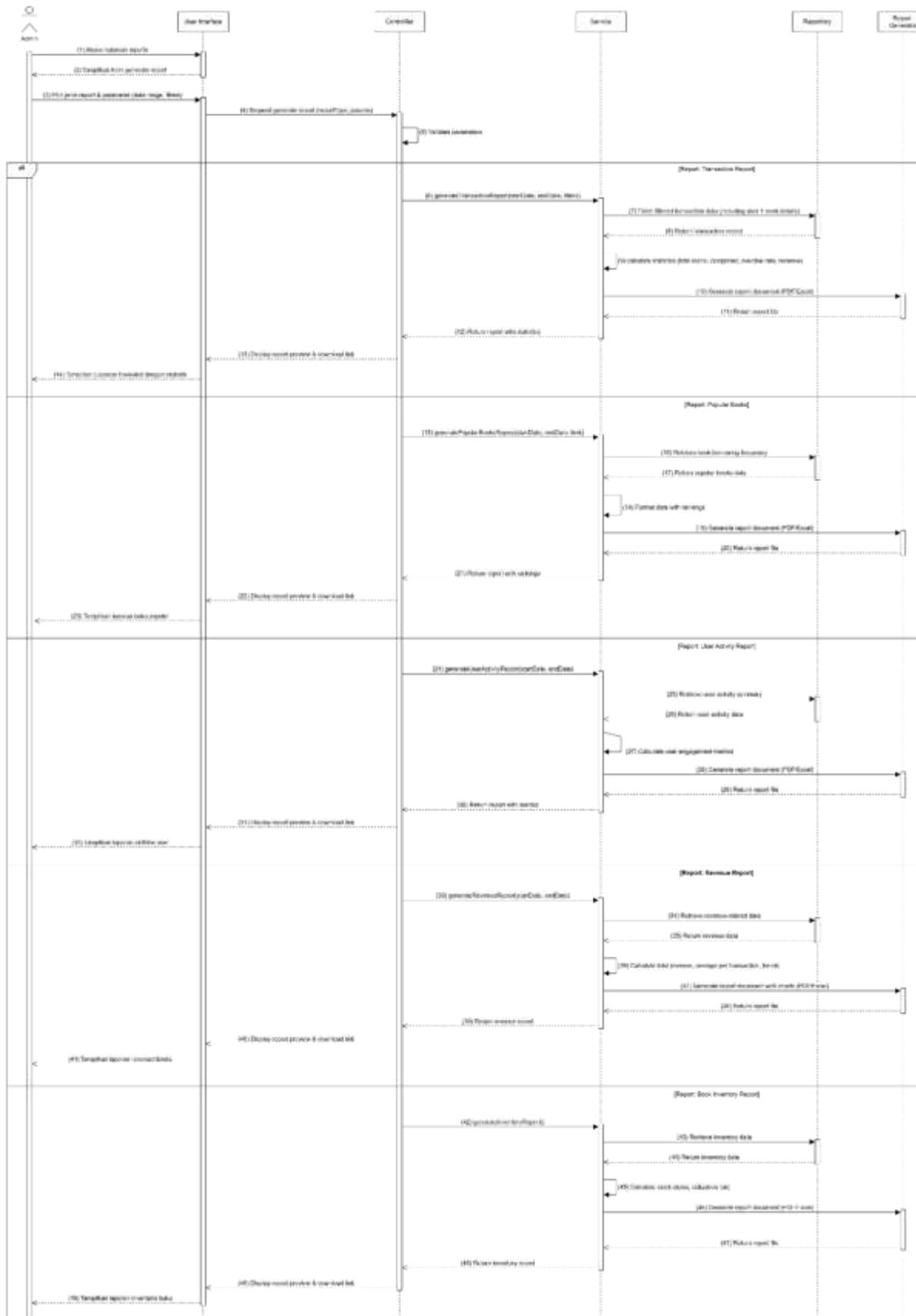
### **3.2.2.23.3 Affected Data Entities**

Use case “View Admin Dashboard” tidak menyebabkan perubahan pada data yang tersimpan di dalam sistem. Proses ini hanya melibatkan pembacaan dan pengolahan data dari entitas Book, User, Transaction, dan Review untuk keperluan visualisasi dan monitoring. Tidak terdapat proses insert, update, maupun delete pada database.

### **3.2.2.24 Process 24**

Aspek	Detail
<b>Use case name</b>	Generate Reports
<b>Description</b>	Admin generate laporan perpustakaan untuk periode tertentu
<b>Pre-condition</b>	User login sebagai ADMIN

<b>Post-condition</b>	Report file (PDF/Excel) berhasil di-generate dan dapat didownload
-----------------------	---



### **3.2.2.24.1 Input Data Entities**

Input data entitas pada use case “Generate Reports” mencakup parameter periode laporan yang dipilih oleh Admin. Data ini meliputi rentang tanggal awal dan tanggal akhir laporan, serta jenis laporan yang akan dihasilkan, seperti laporan transaksi peminjaman, laporan buku populer, laporan aktivitas pengguna, atau laporan pendapatan dari denda keterlambatan. Sistem juga menggunakan data autentikasi Admin dari sesi login aktif untuk memastikan bahwa hanya pengguna dengan role ADMIN yang dapat mengakses fitur pembuatan laporan.

### **3.2.2.24.2 Algorithm or Formula of Process**

Step performed UCS 24: Generate Reports

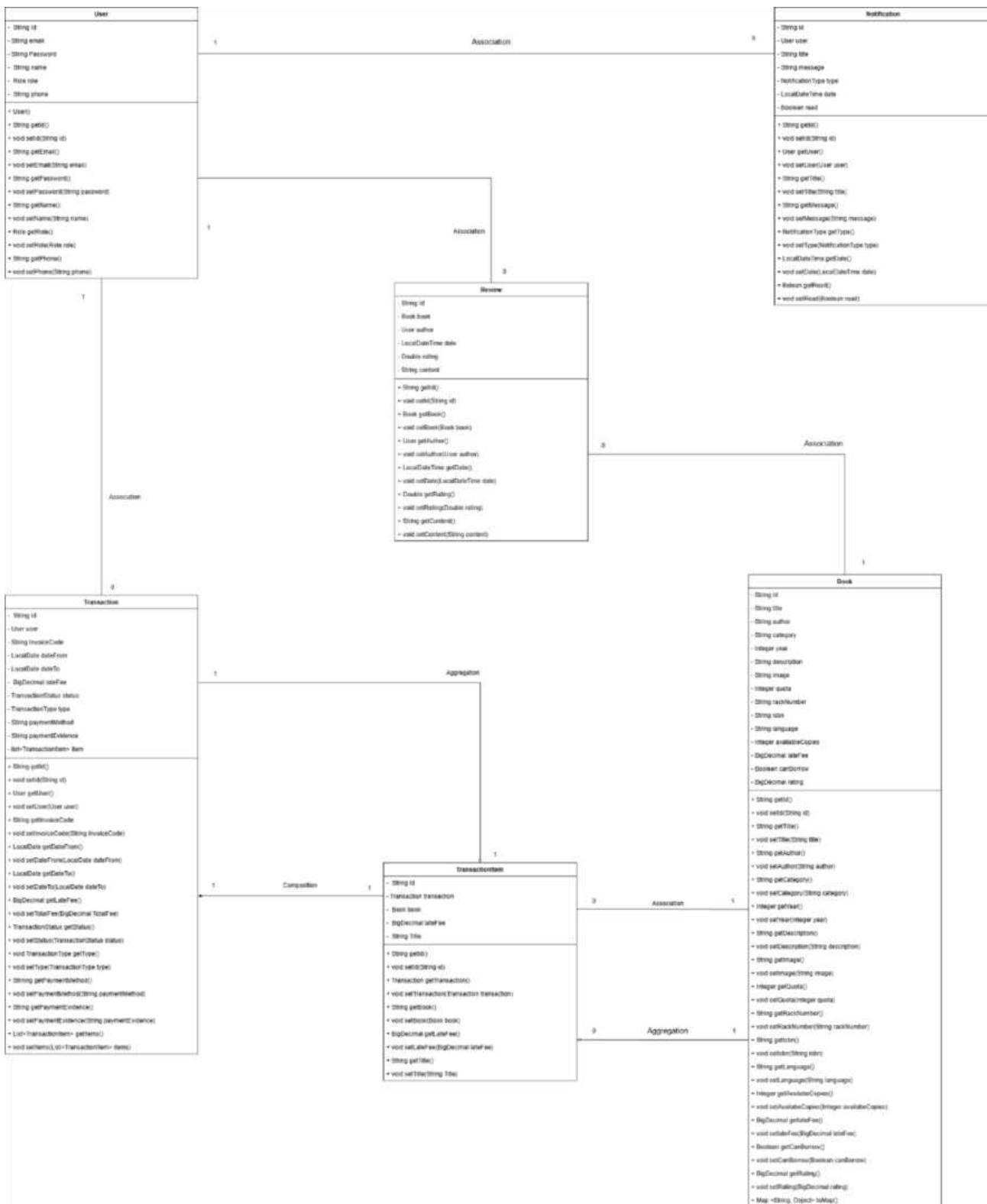
- a. Admin mengakses halaman laporan pada sistem Digilibz.
- b. Sistem menampilkan opsi pemilihan jenis laporan dan periode waktu.
- c. Admin memilih jenis laporan dan menentukan periode laporan.
- d. Admin mengirimkan permintaan pembuatan laporan ke sistem.
- e. Sistem memverifikasi hak akses Admin.
- f. Sistem memvalidasi parameter periode dan jenis laporan.
- g. Sistem mengambil data yang relevan dari database sesuai dengan parameter laporan.
- h. Sistem mengolah data dan menyusun laporan dalam format terstruktur.
- i. Sistem menghasilkan file laporan dalam format PDF atau Excel.
- j. Sistem menyediakan file laporan untuk diunduh oleh Admin.

### **3.2.2.24.3 Affected Data Entities**

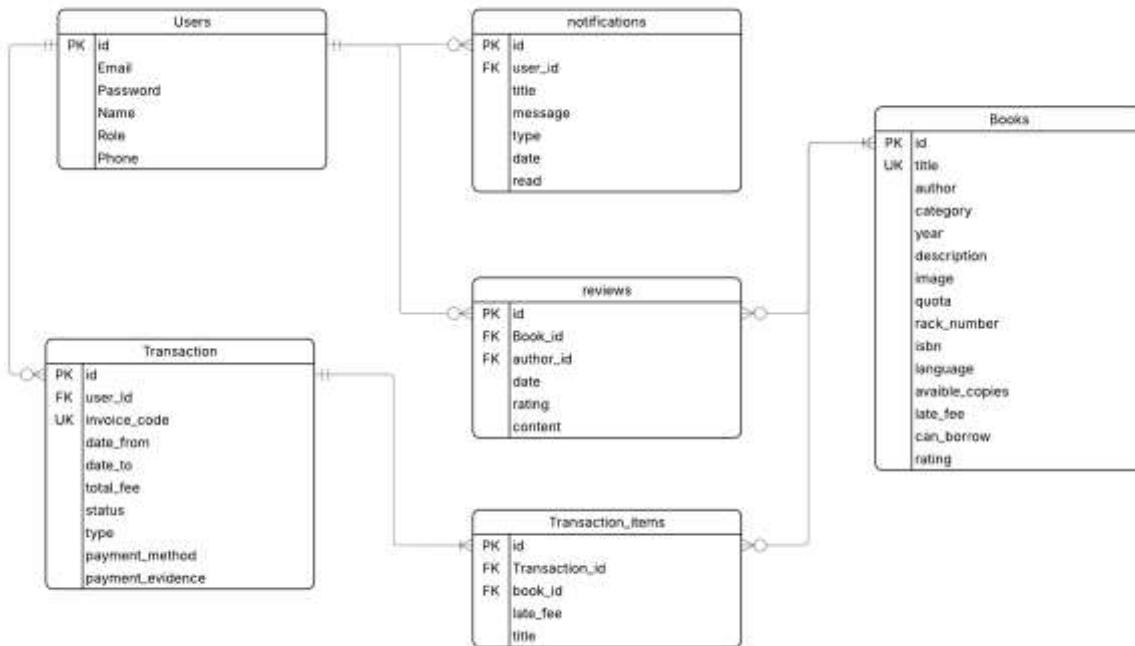
Use case “Generate Reports” tidak menyebabkan perubahan pada data utama yang tersimpan di dalam sistem. Proses ini hanya melibatkan pembacaan dan pengolahan data dari beberapa entitas, seperti Transaction, Book, dan User, untuk tujuan pelaporan. Tidak terdapat proses penambahan, pembaruan, atau penghapusan data pada database.

### **3.2.3 Data Construct Spesifications**

Berikut merupakan Class Diagram dari aplikasi Sistem Perpustakaan Digital (Digilibz).



## ERD



### 3.2.3.1 Construct 1

Merujuk pada Class Diagram, pada class pertama bernama “User” yang berfungsi untuk menyimpan data pengguna sistem. Class ini memiliki beberapa atribut utama yang digunakan untuk menyimpan informasi identitas pengguna serta sejumlah method untuk mengelola data pengguna tersebut.

#### 3.2.3.1.1 Record Type

```
id : String  
email : String  
password : String  
name : String  
role : Role  
phone : String
```

#### 3.2.3.1.2 Constituent Fields

Class ini memiliki 6 field atribut, yaitu id, email, password, name, role, phone.

### **3.2.3.2 Construct 2**

Merujuk pada Class Diagram, pada class kedua bernama “Book” yang berfungsi untuk menyimpan data buku yang tersedia dalam sistem. Class ini memiliki beberapa atribut utama yang digunakan untuk menyimpan informasi detail buku serta mendukung proses pengelolaan koleksi buku dalam aplikasi.

#### **3.2.3.2.1 Record Type**

```
id : String  
title : String  
author : String  
category : String  
year : Integer  
description : String  
image : String  
quota : Integer  
rackNumber : String  
isbn : String  
language : String  
availableCopies : Integer  
lateFee : BigDecimal  
condition : Boolean  
rating : BigDecimal
```

### **3.2.3.2.2 Constituent Fields**

Class ini memiliki 15 field atribut, yaitu id, title, author, category, year, description, image, quota, rackNumber, isbn, language, availableCopies, lateFee, condition, dan rating.

### **3.2.3.3 Construct 3**

Merujuk pada Class Diagram, pada class ketiga bernama “**Notification**” yang berfungsi untuk menyimpan data notifikasi yang dikirimkan kepada pengguna sistem. Class ini digunakan untuk menyampaikan informasi atau pemberitahuan terkait aktivitas tertentu dalam aplikasi.

#### **3.2.3.3.1 Record Type**

id : String

user : User

title : String

message : String

type : NotificationType

date : LocalDateTime

read : Boolean

#### **3.2.3.3.2 Constituent Fields**

Class ini memiliki 7 field atribut, yaitu id, user, title, message, type, date, dan read.

### **3.2.3.4 Construct 4**

Merujuk pada Class Diagram, pada class keempat bernama “**Review**” yang berfungsi untuk menyimpan data ulasan yang diberikan oleh pengguna terhadap buku. Class ini memiliki atribut yang digunakan untuk menyimpan informasi ulasan, penilaian, serta keterkaitan antara pengguna dan buku.

#### **3.2.3.4.1 Record Type**

```
id : String  
book : Book  
author : User  
date : LocalDateTime  
rating : Double  
content : String
```

#### **3.2.3.4.2 Constituent Fields**

Class ini memiliki 6 field atribut, yaitu id, book, author, date, rating, dan content.

#### **3.2.3.5 Construct 5**

Merujuk pada Class Diagram, pada class kelima bernama “Transaction” yang berfungsi untuk menyimpan data transaksi yang dilakukan oleh pengguna. Class ini mencatat informasi peminjaman atau pengembalian buku serta status transaksi yang terjadi.

#### **3.2.3.5.1 Record Type**

```
id : String  
user : User  
invoiceCode : String  
dateFrom : LocalDateTime  
dateTo : LocalDateTime  
lateFee : BigDecimal  
status : TransactionStatus  
type : TransactionType  
paymentMethod : String  
paymentEvidence : String
```

#### **3.2.3.5.2 Constituent Fields**

Class ini memiliki 10 field atribut, yaitu id, user, invoiceCode, dateFrom, dateTo, lateFee, status, type, paymentMethod, dan paymentEvidence.

### **3.2.3.6 Construct 6**

Merujuk pada Class Diagram, pada class keenam bernama “TransactionItem” yang berfungsi untuk menyimpan detail item buku dalam suatu transaksi. Class ini digunakan untuk menghubungkan data transaksi dengan data buku yang terlibat.

#### **3.2.3.6.1 Record Type**

id : String

transaction : Transaction

book : Book

lateFee : BigDecimal

title : String

#### **3.2.3.6.2 Constituent Fields**

Class ini memiliki 5 field atribut, yaitu id, transaction, book, lateFee, dan title.

### **3.2.4 Data Dictionary**

#### **3.2.4.1 Date Element 1**

##### **3.2.4.1.1 Name**

##### **3.2.4.1.2 Representation**

##### **3.2.4.1.3 Units/Format**

##### **3.2.4.1.4 Precision/Accuracy**

##### **3.2.4.1.4 Range**

#### **3.2.4.2 Date Element 2**

##### **3.2.4.2.1 Name**

##### **3.2.4.2.2 Representation**

##### **3.2.4.2.3 Units/Format**

##### **3.2.4.2.4 Precision/Accuracy**

##### **3.2.4.2.4 Range**

#### **3.2.4.q Date Element q**

##### **3.2.4.q.1 Name**

##### **3.2.4.q.2 Representation**

##### **3.2.4.q.3 Units/Format**

##### **3.2.4.q.4 Precision/Accuracy**

##### **3.2.4.1.4 Range**

### **3.3 Performance Requirements**

#### **3.3.1 Design Constraints**

Website ini dikembangkan dengan beberapa batasan desain yang harus dipenuhi selama proses perancangan dan implementasi sistem. Pada sisi frontend, website dibatasi untuk dikembangkan menggunakan framework Next.js dengan bahasa TypeScript untuk memastikan struktur kode yang konsisten, maintainable, dan mendukung pengembangan

aplikasi web modern. Aplikasi ini dirancang hanya untuk platform web, sehingga tidak mencakup pengembangan aplikasi native Android maupun iOS.

Pada sisi backend, sistem dibatasi untuk menggunakan Java dengan framework Spring Boot sebagai fondasi pengembangan layanan server. Pemilihan Spring Boot dilakukan untuk mendukung arsitektur aplikasi berbasis REST API yang modular dan mudah dikembangkan. Untuk pengelolaan data, sistem diwajibkan menggunakan Hibernate sebagai Object Relational Mapping (ORM) dan basis data MySQL versi 8 atau lebih tinggi guna menjamin kompatibilitas, stabilitas, dan performa penyimpanan data. Selain itu, aplikasi dirancang untuk dijalankan dalam lingkungan containerized menggunakan Docker. Penggunaan Docker menjadi batasan desain yang bertujuan untuk memastikan konsistensi lingkungan pengembangan, pengujian, dan deployment, serta mempermudah proses distribusi aplikasi ke berbagai lingkungan server.

Dari sisi pengguna, aplikasi dirancang untuk melayani dua jenis pengguna, yaitu pengguna umum dan administrator, sehingga desain sistem harus mendukung pengelolaan hak akses dan peran pengguna secara terpisah. Seluruh batasan desain ini ditetapkan untuk menjaga keseragaman teknologi, kemudahan pemeliharaan sistem, serta memastikan aplikasi dapat dikembangkan dan dioperasikan secara optimal.

### 3.3.2 Software System Attributes

#### 3.3.2.1 NFR-PERF: Performance

ID	Requirement	Target	Description
NFR-PERF-01	Page Load Time	< 3 seconds	Setiap halaman aplikasi harus load dalam waktu kurang dari 3 detik pada koneksi internet normal (4G/WiFi)

NFR-PERF-02	API Response Time	< 500ms	REST API endpoint harus merespons dalam waktu kurang dari 500 milliseconds untuk operasi CRUD standar
NFR-PERF-03	Database Query Time	< 200ms	Query database harus eksekusi dalam waktu kurang dari 200 ms untuk query kompleks
NFR-PERF-04	Concurrent Users	100+ users	Sistem harus dapat menangani minimal 100 concurrent users tanpa degradasi performa signifikan
NFR-PERF-05	Image Loading	< 2 seconds	Book cover image harus load dalam waktu kurang dari 2 detik
NFR-PERF -06	Search Performance	< 1 second	Fitur search harus mengembalikan hasil dalam waktu kurang dari 1 detik untuk database dengan 10,000+ buku

### 3.3.2.2 NFR-SEC: Security Requirements

ID	Requirement	Description
NFR-SEC-01	HTTPS Protocol	Semua komunikasi antara client dan server harus menggunakan HTTPS/TLS untuk enkripsi data in transit
NFR-SEC-02	Password Security	Password user harus di-hash menggunakan bcrypt dengan minimum 10 rounds untuk keamanan

NFR-SEC-03	SQL Injection Prevention	Sistem harus menggunakan prepared statements atau ORM (JPA) untuk mencegah SQL injection attacks
NFR-SEC-04	XSS Prevention	Input dari user harus di-sanitize dan di-validate untuk mencegah Cross-Site Scripting (XSS) attacks
NFR-SEC-05	CSRF Protection	Sistem harus mengimplementasikan CSRF token protection untuk form submissions
NFR-SEC-06	Session Management	Session harus expire setelah 30 menit inactivity dan harus di-invalidate saat logout
NFR-SEC-07	File Upload Security	File upload harus divalidasi: hanya allow image (JPG, PNG), maksimal size 5MB, dan check MIME type
NFR-SEC-08	API Authentication	REST API harus menggunakan JWT token atau session-based authentication
NFR-SEC-09	Role-Based Authorization	Setiap endpoint harus memiliki authorization check berdasarkan role user (Admin/User)
NFR-SEC-10	Sensitive Data Protection	Data sensitif (password, payment info) tidak boleh di-log atau di-expose di error messages

### 3.3.2.3 NFR-USAB: Usability Requirements

ID	Requirement	Description

NFR-USAB-01	Responsive Design	UI harus responsive dan berfungsi dengan baik di berbagai ukuran layar (desktop 1920px, tablet 768px, mobile 375px)
NFR-USAB-02	Intuitive Navigation	Navigation menu harus jelas dan mudah dipahami dengan maksimal 3 level hierarchy
NFR-USAB-03	Error Messages	Error messages harus informatif, user-friendly, dan memberikan guidance cara memperbaiki error
NFR-USAB-04	Form Validation	Form harus memiliki client-side validation dengan feedback langsung (real-time validation)
NFR-USAB-05	Loading Indicators	Sistem harus menampilkan loading spinner atau progress indicator saat proses berlangsung
NFR-USAB-06	Consistent UI	Design pattern, color scheme, typography, dan component styling harus konsisten di seluruh aplikasi
NFR-USAB-07	Accessibility	Sistem harus memenuhi basic WCAG 2.0 guidelines Level A (keyboard navigation, alt text untuk images)
NFR-USAB-08	Success Feedback	Sistem harus memberikan feedback sukses yang jelas setelah user melakukan aksi (toast notification, success message)
NFR-USAB-09	Confirmation Dialogs	Aksi destructive (delete, reject) harus memiliki confirmation dialog sebelum eksekusi

### 3.3.2.4 NFR-RELI: Reliability Requirements

ID	Requirement	Description
NFR-RELI-01	System Uptime	Sistem harus memiliki uptime minimal 99% (maksimal downtime 7.2 jam per bulan)
NFR-RELI-02	Error Handling	Sistem harus gracefully handle errors tanpa crash dan menampilkan error page yang informatif
NFR-RELI-03	Data Backup	Database harus di-backup secara otomatis setiap hari (daily backup) dan disimpan selama 30 hari
NFR-RELI-04	Transaction Atomicity	Database transaction harus memenuhi ACID properties untuk menjaga data consistency
NFR-RELI-05	Data Validation	Semua input data harus divalidasi di backend untuk memastikan data integrity
NFR-RELI-06	Failover Mechanism	Sistem harus memiliki basic error recovery mechanism (retry logic untuk transient failures)

### 3.3.2.5 NFR-MAIN: Maintainability Requirements

ID	Requirement	Description
NFR-MAIN-01	Code Documentation	Kode harus memiliki JavaDoc comments untuk class dan method penting (minimal 70% coverage)

NFR- MAIN-02	Modular Architecture	Sistem harus menggunakan layered architecture (Controller-Service-Repository pattern) untuk separation of concerns
NFR- MAIN-03	Version Control	Semua kode harus di-track menggunakan Git dengan commit message yang clear dan descriptive
NFR- MAIN-04	Coding Standards	Kode harus mengikuti Java/JavaScript coding conventions (Google Java Style Guide atau Airbnb JavaScript Style)
NFR- MAIN-05	Logging System	Sistem harus memiliki comprehensive logging untuk debugging (log level: DEBUG, INFO, WARN, ERROR)
NFR- MAIN-06	Code Reusability	Kode harus DRY (Don't Repeat Yourself) dengan utility classes dan helper methods untuk logic yang reusable
NFR- MAIN-07	Unit Test Coverage	Minimal 70% code coverage untuk Service layer dengan unit tests (JUnit)

### 3.3.2.6 NFR-SCALE: Scalability Requirements

ID	Requirement	Description
NFR- SCALE-01	Horizontal Scalability	Arsitektur backend harus stateless untuk mendukung horizontal scaling dengan load balancer

NFR-SCALE-02	Database Scalability	Database design harus normalized dan indexed untuk mendukung pertumbuhan data hingga 100,000+ records
NFR-SCALE-03	Stateless API	REST API harus stateless (tidak menyimpan session di server) untuk mendukung distributed deployment
NFR-SCALE-04	Caching Strategy	Sistem harus menggunakan caching (Redis atau in-memory cache) untuk data yang frequently accessed

### 3.3.2.7 NFR-COMPAT: Compatibility Requirements

ID	Requirement	Description
NFR-COMPAT-01	Browser Compatibility	Aplikasi harus support browser modern: Chrome (latest 2 versions), Firefox (latest 2 versions), Safari (latest 2 versions), Edge (latest 2 versions)
NFR-COMPAT-02	Mobile Compatibility	UI harus berfungsi dengan baik di mobile devices (iOS Safari, Android Chrome)
NFR-COMPAT-03	Database Compatibility	Aplikasi harus compatible dengan PostgreSQL 13+ atau MySQL 8+

### **3.3.3 Other Requirements**

## **PENUTUP & KESIMPULAN**

Dokumen *Software Requirements Specification (SRS)* ini disusun sebagai acuan utama dalam pengembangan Sistem Perpustakaan Digital (Digilibz) agar proses perancangan dan implementasi sistem dapat dilakukan secara terstruktur, sistematis, dan sesuai dengan kebutuhan pengguna. Seluruh kebutuhan fungsional dan non-fungsional telah dianalisis dan didokumentasikan secara jelas, didukung dengan berbagai model visual seperti *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Data Flow Diagram (DFD)*, *Entity Relationship Diagram (ERD)*, dan *Class Diagram* untuk memberikan gambaran menyeluruh mengenai perilaku, alur data, serta struktur sistem.

Diharapkan dokumen SRS ini dapat menjadi pedoman yang konsisten bagi tim pengembang, dosen pengampu, dan seluruh pemangku kepentingan dalam memahami ruang lingkup dan spesifikasi sistem Digilibz. Dengan adanya dokumentasi yang lengkap dan terstandarisasi, pengembangan sistem diharapkan dapat berjalan lebih efektif, meminimalkan kesalahan implementasi, serta menghasilkan sistem perpustakaan digital yang andal, mudah digunakan, dan mampu mendukung proses pengelolaan perpustakaan secara optimal.