DIGILIBZ

SISTEM PERPUSTAKAAN DIGITAL

REQUIREMENTS & USE CASE DIAGRAM

Kelompok	Digilibz
Kelas	IT4704

Anggota:

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

S1 Teknologi Informasi
Fakultas Informatika
Universitas Telkom
2025/2026

DAFTAR ISI

1. 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	Medium	Sistem harus

	Calculation		otomatis menghitung denda keterlambatan berdasarkan jumlah hari terlambat × 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 ringkasan: 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

2. NON-FUNCTIONAL REQUIREMENTS

NFR-PERF: Performance Requirements

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

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

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

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)

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)

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

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+

TOTAL NON-FUNCTIONAL REQUIREMENTS: 38 NFR

- Performance: 6 NFR
- Security: 10 NFR
- Usability: 9 NFR
- Reliability: 6 NFR
- Maintainability: 7 NFR
- Scalability: 4 NFR
- Compatibility: 3 NFR

3. CONSTRAINTS

CON-TECH: Technical Constraints

ID	Constraint	Description	Impact
CON-TECH-01	Backend Framework	Backend HARUS menggunakan Spring Boot (Java 17+) sebagai framework utama	Technology stack fixed, tidak bisa ganti framework
CON-TECH-02	Frontend Framework	Frontend HARUS menggunakan Next.js (React framework) untuk development	UI development mengikuti React paradigm
CON-TECH-03	Database Type	Database HARUS SQL-based: PostgreSQL atau MySQL	Tidak bisa menggunakan NoSQL database
CON-TECH-04	ORM Requirement	Backend HARUS menggunakan Spring Data JPA (Hibernate) untuk database access	Tidak boleh raw JDBC atau query manual
CON-TECH-05	Version Control	Project HARUS menggunakan Git	Semua code harus di-commit ke

		dan GitHub untuk version control dan collaboration	repository
CON-TECH-06	API Architecture	Backend HARUS menyediakan RESTful API dengan JSON response format	Tidak bisa menggunakan GraphQL atau gRPC
CON-TECH-07	Build Tool	Backend HARUS menggunakan Maven atau Gradle untuk dependency management	Build process harus automated
CON-TECH-08	Java Version	Backend HARUS menggunakan Java 17 atau yang lebih baru (LTS version)	Tidak bisa menggunakan Java 8 atau versi lama

CON-BUS: Business Constraints

ID	Constraint	Description	Impact
CON-BUS-01	User Roles Only	Sistem hanya memiliki 2 role: USER dan ADMIN (tidak ada role Mahasiswa/Dosen terpisah)	Semua peminjam diperlakukan sama, tidak ada privilege khusus
CON-BUS-02	Physical Books Only	Sistem untuk manajemen peminjaman buku fisik, bukan e-book atau digital library	Tidak ada fitur baca online atau download e-book
CON-BUS-03	Single Library	Sistem untuk single library location, bukan multi-branch library system	Tidak ada fitur inter-branch transfer atau location management

CON-BUS-04	Academic Context	Sistem dirancang untuk lingkungan akademik (universitas/sekolah)	User management menggunakan id sebagai identifier
CON-BUS-05	Loan Duration Limit	Maksimal durasi peminjaman per transaksi adalah 30 hari	User tidak bisa meminjam lebih dari 30 hari sekali pinjam
CON-BUS-06	Loan Quota	Maksimal peminjaman bersamaan untuk semua user adalah 5 buku aktif	User tidak bisa pinjam buku ke-6 sebelum mengembalikan yang lain
CON-BUS-07	Manual Approval	Semua transaksi peminjaman harus di-approve manual oleh Admin	Tidak ada auto-approval system
CON-BUS-08	No Renewal Feature	Sistem tidak memiliki fitur perpanjangan peminjaman (renewal)	User harus kembalikan buku dulu, baru bisa pinjam lagi

CON-DEV: Development Constraints

ID	Constraint	Description	Impact
CON-DEV-01	Team Size	Tim terdiri dari 5 anggota dengan pembagian role yang fixed	Resource terbatas, tidak bisa menambah anggota tim
CON-DEV-02	Development Timeline	Proyek harus selesai dalam 1 semester akademik (14 minggu)	Timeline ketat, harus prioritize features

CON-DEV-03	OOP Paradigm	Kode HARUS menggunakan Object-Oriented Programming paradigm	Design harus mengikuti prinsip OOP (encapsulation, inheritance, polymorphism, abstraction)
CON-DEV-04	Class Structure	Sistem HARUS memiliki minimal 6 class utama sesuai proposal: User, Book, Transaction, TransactionItem, Review, Notification	Class diagram sudah fixed, tidak bisa mengurangi atau mengubah core classes
CON-DEV-05	Academic Project	Proyek ini adalah tugas akademik untuk mata kuliah PBO	Harus memenuhi requirement akademik, termasuk dokumentasi lengkap dan presentasi
CON-DEV-06	No Third-Party Payment	Sistem tidak terintegrasi dengan payment gateway (Midtrans, Xendit, dll)	Payment recording manual dengan upload bukti transfer
CON-DEV-07	Local Deployment	Deployment untuk demo akademik di local atau basic cloud platform (Heroku, Railway)	Tidak ada budget untuk enterprise hosting atau infrastructure

CON-DATA: Data Constraints

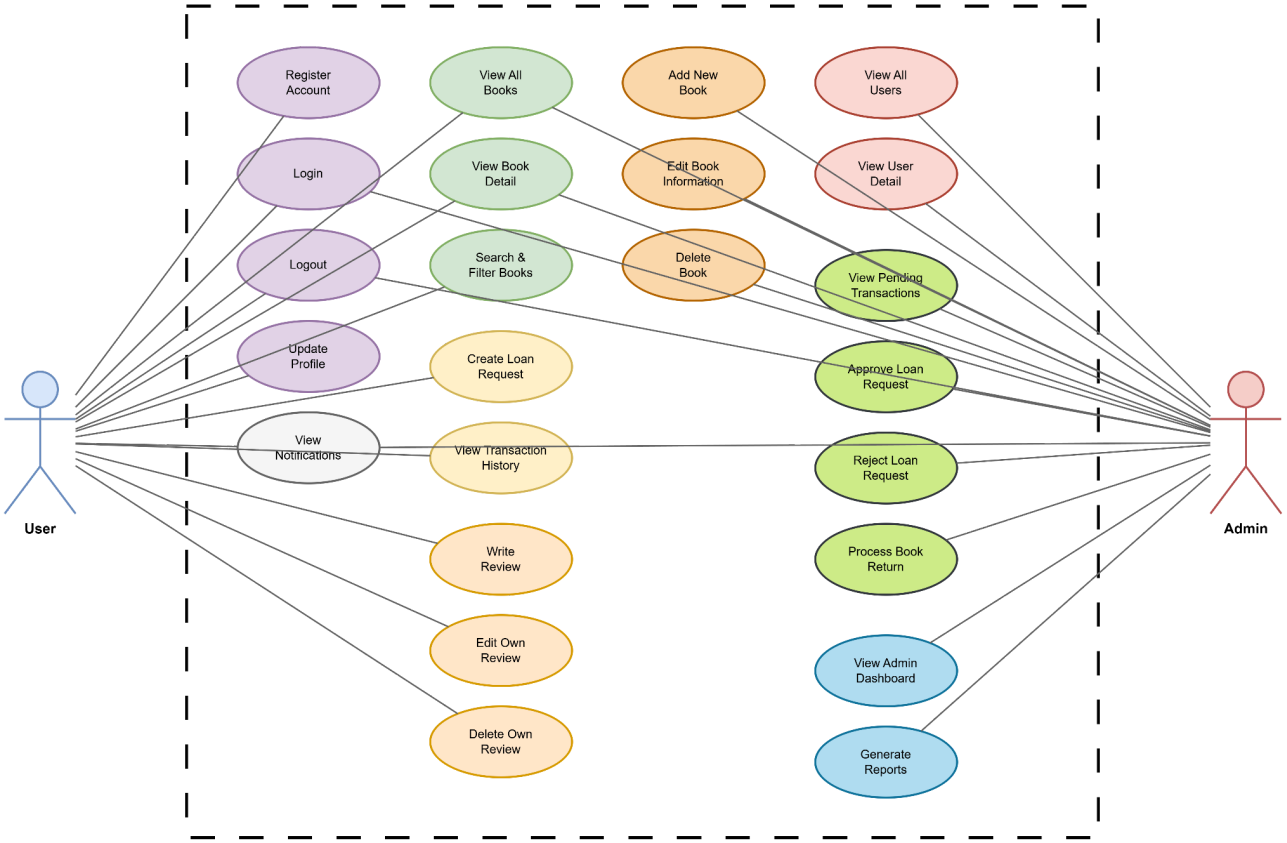
ID	Constraint	Description	Impact
CON-DATA-01	No Real Email Service	Sistem tidak terintegrasi dengan email service (SendGrid, Mailgun) untuk production	Notifikasi hanya in-app notification, tidak ada email notification real
CON-DATA-02	Local File Storage	File upload (book covers) disimpan di local file system atau basic object storage	Tidak ada CDN untuk image delivery
CON-DATA-03	Seed Data Required	Database harus di-seed dengan sample data untuk demo purpose	Perlu prepare dummy data untuk books, users, transactions
CON-DATA-04	No Real-time Features	Sistem tidak memiliki real-time notification (WebSocket, Server-Sent Events)	Notification hanya update saat user refresh atau navigation

TOTAL CONSTRAINTS: 23 Constraints

- Technical Constraints: 8
- Business Constraints: 8
- Development Constraints: 7
- Data Constraints: 4

4. USE CASE DIAGRAM

4.1 Use Case Diagram



5. USE CASE LIST

Daftar Lengkap Use Case

UC ID	Use Case Name	Actor	Priority	Description
AUTHENTICATION & PROFILE				
UC-01	Register Account	User	High	User baru mendaftar akun dengan mengisi form registrasi (email, password, nama, phone)

UC-02	Login	User, Admin	High	User atau Admin login ke sistem menggunakan email dan password
UC-03	Logout	User, Admin	Medium	User atau Admin logout dari sistem
UC-04	Update Profile	User	Medium	User update informasi profil (nama, phone, foto)
BOOK BROWSING & SEARCH				
UC-05	View All Books	User, Admin	High	Melihat katalog lengkap semua buku di perpustakaan
UC-06	View Book Detail	User, Admin	High	Melihat detail lengkap buku termasuk deskripsi, rating, review, dan ketersediaan
UC-07	Search & Filter Books	User, Admin	High	Mencari buku dengan keyword dan memfilter berdasarkan kategori/tahun
BOOK MANAGEMENT (ADMIN)				
UC-08	Add New Book	Admin	High	Admin menambah buku baru ke

				sistem dengan form input lengkap
UC-09	Edit Book Information	Admin	Medium	Admin mengedit informasi buku yang sudah ada
UC-10	Delete Book	Admin	Medium	Admin menghapus buku dari sistem (soft delete)
TRANSACTION - BORROWING				
UC-11	Create Loan Request	User	High	User membuat permintaan peminjaman dengan memilih buku dan rentang tanggal
UC-12	View Transaction History	User	Medium	User melihat riwayat semua transaksi peminjaman
UC-13	View Pending Transactions	Admin	High	Admin melihat daftar transaksi yang menunggu approval
UC-14	Approve Loan Request	Admin	High	Admin menyetujui permintaan peminjaman dari user
UC-15	Reject Loan	Admin	High	Admin

	Request			menolak permintaan peminjaman dengan memberikan alasan
UC-16	Process Book Return	Admin	High	Admin memproses pengembalian buku dan kalkulasi denda jika terlambat
REVIEW & RATING				
UC-17	Write Review	User	Medium	User menulis review dan memberikan rating untuk buku yang pernah dipinjam
UC-18	Edit Own Review	User	Low	User mengedit review yang pernah dibuat
UC-19	Delete Own Review	User	Low	User menghapus review sendiri
USER MANAGEMENT (ADMIN)				
UC-20	View All Users	Admin	Medium	Admin melihat daftar semua user terdaftar
UC-21	View User Detail	Admin	Medium	Admin melihat detail lengkap user termasuk transaction history

NOTIFICATION				
UC-22	View Notifications	User, Admin	Medium	User/Admin melihat daftar notifikasi sistem
DASHBOARD & REPORTS				
UC-23	View Admin Dashboard	Admin	High	Admin melihat dashboard dengan statistik perpustakaan
UC-24	Generate Reports	Admin	Medium	Admin generate laporan (transaction report, popular books, dll)

5.1 REALISASI USE CASE

5.1.1 Use Case #1: Register Account

Use Case Scenario #1: Register Account

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

Main Scenario:

User	System
1. Mengakses halaman registrasi	
	2. Menampilkan form registrasi (email, password, nama, phone)
3. Mengisi semua field yang required	
4. Menekan tombol "Register"	
	5. Validasi format email dan password strength
	6. Check apakah email sudah terdaftar
	7. Hash password menggunakan bcrypt
	8. Simpan data user ke database dengan role "USER"
	9. Menampilkan pesan sukses "Registrasi berhasil"
	10. Redirect ke halaman login

Alternative Flow: Email Sudah Terdaftar

User	System
3. Mengisi email yang sudah ada di database	
	6. Deteksi email duplicate
	7. Menampilkan error "Email sudah terdaftar, gunakan email lain"
	8. Tetap di halaman registrasi dengan data form dipertahankan

5.1.2 Use Case #2: Login

Use Case Scenario #2: Login

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

Main Scenario:

User	System
1. Mengakses halaman login	
	2. Menampilkan form login (email, password)
3. Memasukkan email dan password	
4. Menekan tombol "Login"	
	5. Validasi format input
	6. Query database untuk mencari user berdasarkan email
	7. Verify password menggunakan bcrypt compare
	8. Generate session token atau JWT
	9. Simpan session di server atau return JWT ke client

	10. Menampilkan pesan sukses "Login berhasil"
	11. Redirect ke dashboard sesuai role (USER → User Dashboard, ADMIN → Admin Dashboard)

Alternative Flow: Login Gagal - Password Salah

User	System
3. Memasukkan email benar tapi password salah	
	7. Password verification failed
	8. Log failed login attempt
	9. Menampilkan error message "Email atau password salah"
	10. Tetap di halaman login, form dikosongkan

Alternative Flow: Login Gagal - Email Tidak Ditemukan

User	System
3. Memasukkan email yang tidak terdaftar	
	6. User not found in database
	7. Menampilkan error message "Email atau password salah"
	8. Tetap di halaman login

5.1.3 Use Case #3: Logout

Use Case Scenario #3: Logout

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

Main Scenario:

User	System
1. Klik button/link "Logout" di menu	
	2. Menampilkan confirmation dialog "Apakah Anda yakin ingin logout?"
3. Klik "Ya" untuk konfirmasi	
	4. Invalidate session token atau delete JWT
	5. Clear session data di server
	6. Clear cookies di browser
	7. Menampilkan pesan "Anda telah logout"
	8. Redirect ke halaman login

Alternative Flow: Cancel Logout

User	System
3. Klik "Batal" pada confirmation dialog	
	4. Close dialog
	5. User tetap di halaman sebelumnya dengan session tetap aktif

5.1.4 Use Case #4: Update Profile

Use Case Scenario #4: Update Profile

Aspek	Detail
Use case name	Update Profile
Description	User memperbarui informasi profil mereka sendiri
Pre-condition	User sudah login
Post-condition	Data profil user berhasil diupdate di database

Main Scenario:

User	System
1. Mengakses menu "Profile" atau "Edit Profile"	
	2. Query database untuk mengambil data user saat ini
	3. Menampilkan form edit profile dengan data existing
4. Mengubah field yang ingin diupdate (nama, phone, foto)	

5. Klik tombol "Save Changes"	
	6. Validasi input data baru
	7. Check apakah ada perubahan data
	8. Update data di database
	9. Update session data jika ada field yang berubah
	10. Menampilkan success message "Profil berhasil diperbarui"
	11. Refresh halaman profile dengan data terbaru

Alternative Flow: No Changes Made

User	System
4. Tidak mengubah apapun	
5. Klik "Save Changes"	
	7. Deteksi tidak ada perubahan
	8. Menampilkan info message "Tidak ada perubahan yang disimpan"
	9. Tetap di halaman profile

5.1.5 Use Case #5: View All Books

Use Case Scenario #5: View All Books

Aspek	Detail
Use case name	View All Books
Description	User atau Admin melihat katalog lengkap semua buku yang tersedia di perpustakaan

Pre-condition	Tidak ada (dapat diakses tanpa login atau setelah login)
Post-condition	Daftar buku ditampilkan dengan pagination

Main Scenario:

User	System
1. Mengakses halaman "Books" atau "Katalog"	
	2. Query database untuk mengambil semua buku aktif (not deleted)
	3. Load book cover images
	4. Hitung total available copies per buku
	5. Menampilkan daftar buku dalam card/grid layout
	6. Tampilkan info: judul, penulis, kategori, rating, status availability
	7. Implementasi pagination (20 buku per halaman)
8. Scroll atau klik pagination untuk melihat lebih banyak	
	9. Load data halaman berikutnya

Alternative Flow: No Books Available

User	System
1. Mengakses halaman katalog buku	
	2. Query database - tidak ada buku
	3. Menampilkan empty state "Belum ada buku tersedia"

5.1.6 Use Case #6: View Book Detail

Use Case Scenario #6: View Book Detail

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

Main Scenario:

User	System
1. Klik card buku dari halaman katalog	
	2. Get book ID dari parameter URL
	3. Query database untuk detail buku lengkap
	4. Load book cover image
	5. Query reviews untuk buku tersebut
	6. Calculate average rating dari semua review
	7. Check book availability status
	8. Menampilkan detail buku: judul, penulis, kategori, tahun, ISBN, deskripsi, stok tersedia, biaya denda
	9. Menampilkan rating rata-rata dengan visualisasi bintang

	10. Menampilkan daftar review dari user lain
	11. Jika user login dan eligible, tampilkan button "Pinjam" atau "Write Review"

Alternative Flow: Book Not Found

User	System
1. Akses URL dengan invalid book ID	
	3. Book not found in database
	4. Redirect ke halaman 404 "Buku tidak ditemukan"

5.1.7 Use Case #7: Search & Filter Books

Use Case Scenario #7: Search & Filter Books

Aspek	Detail
Use case name	Search & Filter Books
Description	User atau Admin mencari buku dengan keyword dan memfilter berdasarkan kategori atau tahun
Pre-condition	User berada di halaman katalog buku
Post-condition	Hasil pencarian dan filter ditampilkan

Main Scenario:

User	System
1. Ketik keyword di search bar (contoh: "Java Programming")	
2. Pilih filter kategori (contoh: "Teknologi")	
3. Klik tombol "Search" atau tekan Enter	
	4. Build SQL query dengan WHERE clauses untuk search dan filter
	5. Execute query: LIKE '%keyword%' pada title dan author
	6. Apply filter kategori jika dipilih
	7. Sort results berdasarkan relevance atau rating
	8. Menampilkan hasil pencarian dengan highlight keyword
	9. Show result count: "Ditemukan 15 buku"
10. Lihat hasil dan klik detail buku	

Alternative Flow: No Results Found

User	**System**
1. Search dengan keyword yang tidak ada	
	5. Query returns empty result
	6. Menampilkan "Tidak ditemukan buku dengan keyword 'xyz'"
	7. Tampilkan suggestions atau rekomendasi buku populer

5.1.8 Use Case #8: Add New Book (Admin)

Use Case Scenario #8: Add New Book

Aspek	Detail
Use case name	Add New Book
Description	Admin menambahkan buku baru ke sistem perpustakaan
Pre-condition	User login sebagai ADMIN
Post-condition	Buku baru berhasil tersimpan di database dan muncul di katalog

Main Scenario:

Admin	System
1. Klik menu "Manage Books" → "Add New Book"	
	2. Verify admin role authorization
	3. Menampilkan form tambah buku kosong
4. Mengisi semua field: judul, penulis, kategori, tahun, ISBN, deskripsi, stok total, late fee	
5. Upload book cover image	
	6. Validate image format (JPG/PNG) dan size (<5MB)
7. Klik "Add Book"	
	8. Validasi semua required fields terisi
	9. Check ISBN tidak duplicate
	10. Save image file ke storage dengan

	unique filename
	11. Insert book data ke database
	12. Set available_copies = stock_total
	13. Set can_borrow = true
	14. Menampilkan success message "Buku berhasil ditambahkan"
	15. Redirect ke book detail page atau book list

Alternative Flow: ISBN Already Exists

Admin	System
4. Mengisi ISBN yang sudah ada di database	
	9. Detect duplicate ISBN
	10. Menampilkan error "ISBN sudah terdaftar"
	11. Tetap di form dengan data dipertahankan

5.1.9 Use Case #9: Edit Book Information (Admin)

Use Case Scenario #9: Edit Book Information

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

Main Scenario:

Admin	System
1. Di halaman book detail, klik button "Edit"	
	2. Verify admin authorization
	3. Load existing book data dari database
	4. Menampilkan form edit dengan data pre-filled
5. Mengubah field yang ingin diupdate	
6. Upload new cover image (opsional)	
7. Klik "Save Changes"	
	8. Validasi input data
	9. Check apakah ada perubahan
	10. Jika ada new image, delete old image dan save new
	11. Update book record di database
	12. Log update activity
	13. Menampilkan success "Buku berhasil diperbarui"
	14. Redirect ke book detail page dengan data terbaru

Alternative Flow: No Changes Detected

Admin	System
7. Klik save tanpa mengubah apapun	
	9. Detect no changes

	10. Menampilkan info "Tidak ada perubahan"

5.1.10 Use Case #10: Delete Book (Admin)

Use Case Scenario #10: Delete Book

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

Main Scenario:

Admin	System
1. Di halaman book detail, klik button "Delete"	
	2. Menampilkan confirmation dialog "Apakah Anda yakin ingin menghapus buku ini?"
3. Klik "Yes, Delete"	
	4. Verify admin authorization
	5. Check apakah ada transaksi aktif untuk buku ini
	6. Set is_deleted = true (soft delete)
	7. Set deleted_at = current timestamp
	8. Set can_borrow = false

	9. Log deletion activity
	10. Menampilkan success "Buku berhasil dihapus"
	11. Redirect ke book list page

Alternative Flow: Active Transactions Exist

Admin	System
3. Konfirmasi delete	
	5. Detect ada transaksi pending/approved untuk buku ini
	6. Menampilkan error "Tidak dapat menghapus buku. Masih ada transaksi aktif."
	7. Tetap di halaman book detail

5.1.11 Use Case #11: Create Loan Request

Use Case Scenario #11: Create Loan Request

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

Main Scenario:

User	System
1. Di halaman book detail, klik button "Pinjam Buku"	
	2. Check user login status
	3. Validate book availability
	4. Check user active loan quota
	5. Menampilkan form peminjaman dengan date pickers
6. Pilih tanggal mulai (date_from)	
7. Pilih tanggal selesai (date_to)	
	8. Validasi date_to > date_from
	9. Validasi durasi max 30 hari
	10. Calculate loan duration (days)
	11. Calculate estimated fee (duration × late_fee)
	12. Display fee estimate
13. Review dan klik "Submit Request"	
	14. Generate unique invoice code (INV-YYYYMMDD-XXX)
	15. Create Transaction record dengan status PENDING
	16. Create TransactionItem record
	17. Update book: available_copies - 1
	18. Create notification untuk user

	"Permintaan peminjaman dibuat"
	19. Create notification untuk admin "Ada permintaan peminjaman baru"
	20. Menampilkan success dengan invoice number
	21. Redirect ke "My Transactions" page

Alternative Flow: Quota Exceeded

User	System
1. Klik "Pinjam Buku"	
	4. Check quota - user sudah punya 5 active loans
	5. Menampilkan error "Anda telah mencapai batas peminjaman (5 buku aktif)"
	6. Button "Pinjam" disabled

Alternative Flow: Book Not Available

User	System
1. Klik "Pinjam Buku"	
	3. Check availability - available_copies = 0
	4. Menampilkan error "Buku sedang tidak tersedia"

5.1.12 Use Case #12: View Transaction History

Use Case Scenario #12: View Transaction History

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

Main Scenario:

User	System
1. Klik menu "My Transactions" atau "Riwayat Peminjaman"	
	2. Query database: SELECT * FROM transactions WHERE user_id = current_user
	3. Join dengan transaction_items dan books
	4. Sort by created_at DESC (terbaru dulu)
	5. Menampilkan list transaksi dengan info: invoice, tanggal, judul buku, status, total fee
	6. Apply color coding untuk status (PENDING=kuning, APPROVED=hijau, REJECTED=merah, RETURNED=biru)
	7. Tampilkan filter: All, Pending, Approved, Returned
8. Klik salah satu transaksi untuk detail	
	9. Redirect ke transaction detail page

Alternative Flow: No Transactions Yet

User	System
1. Akses transaction history	
	2. Query returns empty
	3. Menampilkan empty state "Anda belum pernah meminjam buku"
	4. Tampilkan button "Browse Books"

5.1.13 Use Case #13: View Pending Transactions (Admin)

Use Case Scenario #13: View Pending Transactions

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

Main Scenario:

Admin	System
1. Login ke admin dashboard	
	2. Menampilkan summary: "X Pending Approvals"
3. Klik menu "Pending Transactions"	
	4. Query: SELECT * FROM transactions WHERE status = 'PENDING' ORDER BY created_at ASC

	5. Join dengan users dan books untuk detail
	6. Menampilkan tabel pending requests dengan kolom: Invoice, User Name, Book Title, Request Date, Actions
	7. Highlight request yang sudah > 24 jam (urgent)
8. Klik "Review" pada salah satu request	
	9. Redirect ke transaction detail page untuk approval

5.1.14 Use Case #14: Approve Loan Request (Admin)

Use Case Scenario #14: Approve Loan Request

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

Main Scenario:

Admin	System
1. Buka transaction detail dari pending list	
	2. Menampilkan detail lengkap: user info, book info, dates, calculated fee
3. Review informasi peminjaman	
4. Klik button "Approve"	

	5. Menampilkan confirmation "Setujui peminjaman ini?"
6. Konfirmasi "Yes"	
	7. Update transaction: status = 'APPROVED'
	8. Set approved_by = admin_id
	9. Set approved_at = current timestamp
	10. Create notification untuk user: "Peminjaman Anda disetujui. Silakan ambil buku di perpustakaan."
	11. Log approval action
	12. Menampilkan success "Transaksi berhasil disetujui"
	13. Remove dari pending list, pindah ke approved list

5.1.15 Use Case #15: Reject Loan Request (Admin)

Use Case Scenario #15: Reject Loan Request

Aspek	Detail
Use case name	Reject Loan Request
Description	Admin menolak permintaan peminjaman dengan memberikan alasan
Pre-condition	User login sebagai ADMIN, transaksi berstatus PENDING
Post-condition	Status transaksi berubah menjadi REJECTED, stok buku dikembalikan

Main Scenario:

Admin	System
1. Buka transaction detail dari pending list	
3. Klik button "Reject"	
	4. Menampilkan modal "Alasan Penolakan" dengan textarea
5. Input alasan penolakan (contoh: "Buku sedang maintenance")	
6. Klik "Submit Rejection"	
	7. Validate alasan tidak kosong
	8. Update transaction: status = 'REJECTED'
	9. Save rejection reason
	10. Set approved_by = admin_id (yang reject)
	11. Set approved_at = current timestamp
	12. Restore book stock: available_copies + 1
	13. Create notification untuk user dengan alasan penolakan
	14. Log rejection action
	15. Menampilkan success "Transaksi berhasil ditolak"
	16. Remove dari pending list

Alternative Flow: Empty Rejection Reason

Admin	System
5. Tidak mengisi alasan	
6. Klik submit	
	7. Validation error "Alasan penolakan harus diisi"
	8. Tetap di modal rejection

5.1.16 Use Case #16: Process Book Return (Admin)

Use Case Scenario #16: Process Book Return

Aspek	Detail
Use case name	Process Book Return
Description	Admin memproses pengembalian buku dan menghitung denda jika terlambat
Pre-condition	User login sebagai ADMIN, transaksi berstatus APPROVED
Post-condition	Status transaksi berubah RETURNED, stok buku dikembalikan, denda dihitung

Main Scenario:

Admin	System
1. User datang mengembalikan buku	
2. Admin cari transaksi by invoice code atau user name	
	3. Menampilkan transaction detail
	4. Check current date vs date_to (due date)

	5. Calculate days late (jika ada)
	6. Calculate late fee = days_late × book.late_fee per hari
	7. Display calculated fee
8. Verify kondisi buku OK	
9. Input payment method jika ada late fee	
10. Klik "Process Return"	
	11. Update transaction: status = 'RETURNED'
	12. Set returned_at = current timestamp
	13. Update total_fee dengan late fee (jika ada)
	14. Update book: available_copies + 1
	15. Create notification untuk user dengan info late fee (jika ada)
	16. Print receipt (optional)
	17. Menampilkan success "Pengembalian berhasil diproses"

Alternative Flow: Late Return with Fee

Admin	System
	5. Detect current_date > date_to
	6. Calculate: 3 days late × Rp 2.000 = Rp 6.000
	7. Menampilkan warning "TERLAMBAT 3 hari. Denda: Rp 6.000"
9. Input payment: "Cash"	

	13. Update total_fee = original_fee + late_fee

5.1.17 Use Case #17: Write Review

Use Case Scenario #17: Write Review

Aspek	Detail
Use case name	Write Review
Description	User menulis review dan memberikan rating untuk buku yang pernah dipinjam
Pre-condition	User login, user pernah meminjam buku tersebut (transaksi RETURNED)
Post-condition	Review tersimpan dan muncul di halaman book detail

Main Scenario:

User	System
1. Buka halaman book detail	
	2. Check apakah user pernah pinjam buku ini
	3. Query: SELECT * FROM transactions WHERE user_id=X AND book_id=Y AND status='RETURNED'
	4. Jika yes, tampilkan section "Write a Review"
5. Klik "Write Review"	
	6. Expand form review
7. Pilih rating (1-5 bintang) dengan klik	
8. Tulis review di textarea	

9. Klik "Submit Review"	
	10. Validate rating (1-5) dan content tidak kosong
	11. Check user belum pernah review buku ini (prevent duplicate)
	12. Create review record di database
	13. Recalculate book average rating
	14. Update book.rating field
	15. Menampilkan success "Review berhasil dikirim"
	16. Refresh review section, tampilkan review baru di top

Alternative Flow: User Belum Pernah Pinjam

User	System
1. Buka book detail	
	3. Query returns empty - user belum pernah pinjam
	4. Tidak tampilkan form review
	5. Tampilkan message "Pinjam buku ini terlebih dahulu untuk memberikan review"

5.1.18 Use Case #18: Edit Own Review

Use Case Scenario #18: Edit Own Review

Aspek	Detail
Use case name	Edit Own Review
Description	User mengedit review yang pernah mereka

	buat
Pre-condition	User login, user sudah pernah membuat review untuk buku tersebut
Post-condition	Review berhasil diupdate dengan konten baru

Main Scenario:

User	System
1. Lihat review sendiri di book detail page	
	2. Tampilkan button "Edit" hanya pada review milik user
3. Klik button "Edit" pada review sendiri	
	4. Load existing review data
	5. Convert review display menjadi edit form
6. Ubah rating atau content	
7. Klik "Save Changes"	
	8. Validate input
	9. Update review record: content, rating, updated_at
	10. Recalculate book average rating
	11. Update book.rating
	12. Menampilkan success "Review berhasil diperbarui"
	13. Refresh review display dengan data updated

5.1.19 Use Case #19: Delete Own Review

Use Case Scenario #19: Delete Own Review

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

Main Scenario:

User	System
1. Lihat review sendiri di book detail page	
3. Klik button "Delete"	
	4. Menampilkan confirmation "Hapus review ini?"
5. Konfirmasi "Yes, Delete"	
	6. Delete review record dari database
	7. Recalculate book average rating tanpa review ini
	8. Update book.rating
	9. Menampilkan success "Review berhasil dihapus"
	10. Remove review dari UI

5.1.20 Use Case #20: View All Users (Admin)

Use Case Scenario #20: View All Users

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

Main Scenario:

Admin	System
1. Klik menu "Manage Users" atau "View Users"	
	2. Query: SELECT * FROM users WHERE role='USER' ORDER BY created_at DESC
	3. Menampilkan tabel dengan kolom: No, Name, Email, Phone, Registration Date, Status
	4. Implementasi search user by name/email
	5. Pagination 50 users per page
6. Klik nama user untuk detail	
	7. Redirect ke user detail page

5.1.21 Use Case #21: View User Detail (Admin)

Use Case Scenario #21: View User Detail

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

Main Scenario:

Admin	System
1. Dari user list, klik user untuk detail	
	2. Query user data lengkap
	3. Query transaction history user tersebut
	4. Calculate statistics: total peminjaman, active loans, total denda, dll
	5. Menampilkan user profile: foto, nama, email, phone, join date
	6. Tampilkan statistics cards
	7. Tampilkan transaction history table
	8. Tampilkan review history

5.1.22 Use Case #22: View Notifications

Use Case Scenario #22: View Notifications

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

Main Scenario:

User	System
1. Klik icon notification bell di navbar	
	2. Query: SELECT * FROM notifications WHERE user_id=X ORDER BY date DESC LIMIT 20
	3. Menampilkan dropdown/page dengan list notifikasi
	4. Highlight notifikasi unread (is_read=false)
	5. Tampilkan icon berdasarkan notification type
6. Klik salah satu notifikasi	
	7. Mark as read: UPDATE notifications SET is_read=true WHERE id=Y
	8. Update unread count badge
	9. Redirect ke related page (transaction

	detail, book detail, dll)

5.1.23 Use Case #23: View Admin Dashboard (Admin)

Use Case Scenario #23: View Admin Dashboard

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

Main Scenario:

Admin	System
1. Login sebagai admin	
	2. Redirect otomatis ke admin dashboard
	3. Query statistics:
	- Total books: COUNT(*) FROM books WHERE is_deleted=false
	- Total users: COUNT(*) FROM users WHERE role='USER'
	- Active loans: COUNT(*) FROM transactions WHERE status='APPROVED'
	- Pending approvals: COUNT(*) FROM transactions WHERE status='PENDING'
	4. Query recent reviews (last 5)
	5. Query popular books (most borrowed)

	6. Menampilkan summary cards dengan numbers
	7. Menampilkan charts: loan trends, category distribution
	8. Tampilkan quick actions: "Approve Pending", "Add Book", dll

5.1.24 Use Case #24: Generate Reports (Admin)

Use Case Scenario #24: Generate Reports

Aspek	Detail
Use case name	Generate Reports
Description	Admin generate laporan perpustakaan untuk periode tertentu
Pre-condition	User login sebagai ADMIN
Post-condition	Report file (PDF/Excel) berhasil di-generate dan dapat didownload

Main Scenario:

Admin	System
1. Klik menu "Reports"	
	2. Menampilkan form report generator
3. Pilih report type: "Transaction Report"	
4. Pilih date range: start date dan end date	
5. Klik "Generate Report"	
	6. Validate date range
	7. Query transactions dalam periode tersebut

	8. Calculate summary: total transactions, total revenue, top books, dll
	9. Generate PDF report dengan library (jsPDF/Apache POI)
	10. Include charts dan tables
	11. Save file temporarily
	12. Return download link
	13. Menampilkan success "Report berhasil di-generate"
14. Klik "Download"	
	15. Serve file untuk download

Total Use Cases: 24