

**Tugas Besar**  
**Tahap 3 - Model Praktis dan Kelas Diagram**  
**BookWise**  
**Platform Peminjaman Buku Digital**



**Disusun Oleh:**

Laras Hati Mahendra

18223118

**Program Studi Sistem dan Teknologi Informasi**  
**Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung**  
**Jl. Ganesha 10, Bandung 40132**  
**2025**

# Daftar Isi

<b>Daftar Isi.....</b>	<b>2</b>
<b>Daftar Gambar.....</b>	<b>3</b>
<b>Daftar Tabel.....</b>	<b>4</b>
<b>1. Deskripsi Umum.....</b>	<b>5</b>
1.1. Latar Belakang.....	5
1.2. Deskripsi Sistem.....	5
1.3. Tujuan Sistem.....	5
<b>2. Business Capability.....</b>	<b>6</b>
2.1.1 Lending Management.....	6
2.1.2 Notification System.....	6
2.1.3 Book Management.....	6
2.1.4 User Management.....	6
<b>3. Domain.....</b>	<b>7</b>
<b>4. Bounded Context.....</b>	<b>8</b>
<b>5. Arsitektur Aplikasi dan Implementasi.....</b>	<b>10</b>

## **Daftar Gambar**

Gambar 2.1 Business Capability BookWise	6
Gambar 4.1 Bounded Context BookWise	10
Gambar 5.4 Class Diagram Book Wise	14

## **Daftar Tabel**

Tabel 3.1. Domain	7
Tabel 4.1 Lingkup (Boundary) Setiap Bounded Context)	10
Tabel 4.2 Pola Integrasi Antar Bounded Context	10
Tabel 5.1 Identifikasi Class Diagram	12
Tabel 5.2 Glosarium Ubiquitous Language (Loan BC)	12
Tabel 5.3 Relasi antar Entity/Value Object	13

# 1. Deskripsi Umum

## 1.1. Latar Belakang

Seiring dengan pesatnya perkembangan teknologi digital di era modern, kebutuhan masyarakat terhadap akses literatur yang cepat, mudah, dan efisien semakin meningkat. Aktivitas membaca dan mencari referensi kini tidak lagi terbatas pada ruang fisik perpustakaan, melainkan telah bergeser menuju ekosistem digital yang lebih *fleksibel* dan dinamis. Namun, perpustakaan tradisional masih menghadapi berbagai kendala, seperti keterbatasan kapasitas penyimpanan, jam operasional yang terbatas, serta biaya perawatan yang tinggi. Kondisi tersebut seringkali menghambat proses pembelajaran dan pencarian informasi, terutama bagi pengguna yang memiliki keterbatasan waktu dan lokasi.

**BookWise** hadir sebagai solusi inovatif berupa platform perpustakaan digital yang memungkinkan pengguna untuk meminjam dan mengelola buku secara daring. Sistem ini dirancang dengan prinsip kemudahan akses dan efisiensi, sehingga pengguna dapat menikmati pengalaman pinjaman tanpa batasan waktu maupun tempat.

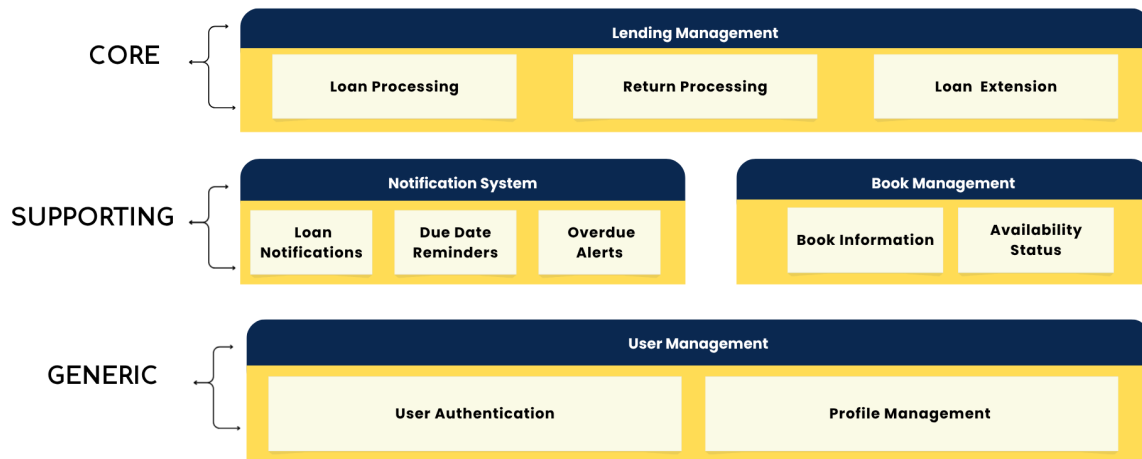
## 1.2. Deskripsi Sistem

**BookWise** merupakan platform peminjaman buku digital yang memungkinkan pengguna untuk meminjam dan mengelola buku secara daring. Sistem ini mendukung proses peminjaman dan pengembalian buku, pengelolaan informasi buku oleh admin, serta pengiriman notifikasi terkait status peminjaman.

## 1.3. Tujuan Sistem

1. Mempermudah pengguna dalam meminjam buku digital tanpa perlu kunjungan fisik ke perpustakaan
2. Memastikan pengelolaan buku yang efisien bagi admin
3. Memberikan notifikasi otomatis kepada pengguna terkait status peminjaman

## 2. Business Capability



Gambar 2.1 Business Capability BookWise

Sistem **BookWise** terbagi menjadi empat kapabilitas utama yang mewakili area fungsional berbeda di dalam arsitektur. Setiap kapabilitas memiliki tanggung jawab jelas dan bekerja melalui integrasi antar bounded context. Pada tahap desain taktis (M3), hanya kapabilitas *Lending Management* yang dimodelkan dalam bentuk *class diagram*, sedangkan kapabilitas lainnya berada di bounded context eksternal dan hanya berinteraksi melalui referensi atau event.

### 2.1.1 Lending Management

Kapabilitas inti yang mengatur seluruh proses peminjaman buku digital. Area ini bertanggung jawab atas pembuatan pinjaman, pencatatan tanggal jatuh tempo, pengembalian buku, perpanjangan pinjaman, serta penandaan pinjaman yang melewati batas waktu. *Lending Management* juga memelihara riwayat peminjaman pengguna. *Lending Management* tidak menyimpan detail buku maupun data pengguna, ia hanya menyimpan sebagai referensi (BookId, UserID).

### 2.1.2 Notification System

Kapabilitas pendukung yang mengirimkan notifikasi kepada pengguna terkait aktivitas peminjaman. aktivitas utama meliputi: notifikasi pinjaman baru, pengingat tanggal jatuh tempo, dan pemberitahuan keterlambatan

*Notification* berada pada *bounded context* terpisah. Ia hanya mengonsumsi event yang dipublikasikan oleh *Lending Management*, misalnya *LoanCreated*, *LoanReturned*, atau *LoanOverdue*.

### 2.1.3 Book Management

Kapabilitas pendukung untuk pengelolaan informasi buku, mulai dari detail buku (judul, pengarang, deskripsi), pengelolaan ketersediaan buku, perubahan metadata buku oleh admin. **BookWise** menampilkan informasi buku kepada pengguna sebelum mereka meminjam. Namun dalam domain *Lending*, konteks ini tidak dimodelkan. *Lending* hanya memanfaatkan **BookId** sebagai referensi tanpa mengetahui isi atau struktur data buku.

### 2.1.4 User Management

Kapabilitas dasar yang mengelola data dan akses pengguna ke dalam platform. Kapabilitas ini tidak dimodelkan pada *class diagram*. *Lending Management* tidak menyimpan informasi profil. Ia hanya berinteraksi melalui *UserId*, sehingga *User Management* tidak perlu dimodelkan pada desain taktis. Seluruh detail pengguna disimpan pada *bounded context* eksternal ini.

## 3. Domain

Tabel 3.1. Domain

Jenis Domain	Capability	Subdomain	Deskripsi
<b>Core Domain</b>	<i>Lending Management</i>	<i>Loan Processing</i>	Mengatur proses peminjaman buku digital, termasuk pembuatan pinjaman baru, validasi, dan pencatatan tanggal jatuh tempo. Subdomain ini adalah pusat logika inti aplikasi.
		<i>Return Processing</i>	Mengelola proses pengembalian buku, memperbarui status pinjaman, dan memastikan konsistensi data setelah buku dikembalikan.

		<i>Loan Extension</i>	Menangani permintaan perpanjangan masa pinjam dan mengatur penyesuaian tanggal jatuh tempo sesuai kebijakan.
<b>Supporting Domain</b>	<i>Notification System</i>	<i>Loan Notifications</i>	Mengirimkan notifikasi ketika pinjaman dibuat, diperpanjang, atau dikembalikan. Mengonsumsi <i>event</i> dari <i>Lending</i> .
		<i>Due Date Reminders</i>	Mengirim pengingat otomatis kepada pengguna sebelum jatuh tempo.
		<i>Overdue Alerts</i>	Mengirim peringatan kepada pengguna saat pinjaman melewati tanggal jatuh tempo.
	<i>Book Management</i>	<i>Book Information</i>	Mengelola detail buku seperti judul, penulis, kategori, deskripsi, dan metadata lainnya.
		<i>Availability Status</i>	Mengatur status ketersediaan buku (tersedia, sedang dipinjam, tidak aktif), digunakan sebelum proses peminjaman.
<b>Generic Domain</b>	Manajemen Pengguna	<i>User Authentication</i>	Mengatur login, akses pengguna, dan validasi identitas. Tidak terhubung langsung ke logika <i>Lending</i> .
		<i>Profile Management</i>	Menyimpan dan memperbarui data profil pengguna. <i>Lending</i> hanya memakai <i>UserId</i> sebagai referensi.

Pembagian ini menggambarkan area-area utama dalam domain bisnis peminjaman buku. Pada tahap M2, subdomain tersebut tidak diterjemahkan secara satu-ke-satu menjadi *bounded context*. Hal ini sesuai prinsip *Domain-Driven Design*, dimana subdomain mewakili

area masalah bisnis, sedangkan *bounded context* menentukan batasan linguistik dan model yang digunakan dalam implementasi.

*Loan Processing*, sebagai *core domain*, menjadi *bounded context* utama, yaitu *Lending / Loan Processing BC*, yang menjadi pusat dari desain dan implementasi *final project*. Sementara itu, tiga subdomain lainnya berperan sebagai *bounded context* eksternal:

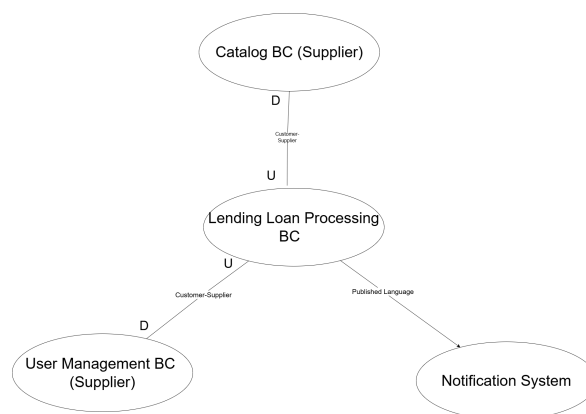
1. *Catalog / Book Management BC* menyediakan model buku yang digunakan *Lending BC*.
2. *User Management BC* menyediakan identitas dan data pengguna.
3. *Notification BC* mengonsumsi *event domain* dari *Lending BC* untuk mengirimkan pemberitahuan otomatis.

Oleh sebab itu, tugas kedua hanya memfokuskan desain hanya pada satu *bounded context* inti sementara *bounded context* lainnya akan diposisikan sebagai layanan eksternal/pendukung.

Dalam pendekatan *Domain-Driven Design*, subdomain menggambarkan area masalah bisnis secara konseptual, sedangkan *bounded context* menentukan batas linguistik dan batas model yang digunakan dalam implementasi. Oleh karena itu, satu subdomain tidak selalu diterjemahkan secara langsung menjadi satu *bounded context*. Pemisahan ini penting untuk menjaga agar model di setiap konteks tetap konsisten, terdefinisi dengan jelas, serta tidak saling bertabrakan dalam hal istilah dan aturan domain.

#### 4. *Bounded Context*

*Bounded Context* merupakan bagian penting dalam pendekatan *Domain-Driven Design* (DDD) yang digunakan untuk memetakan batas tanggung jawab dari setiap domain di dalam sistem **Bookwise**. Melalui diagram *bounded context*, dapat diketahui bagaimana setiap domain saling berhubungan dan berinteraksi untuk mewujudkan fungsi utama sistem, yaitu peminjaman serta pembacaan buku digital.



Gambar 4.1 Bounded Context BookWise

Tabel 4.1 Lingkup (*Boundary*) Setiap *Bounded Context*)

<b><i>Bounded Context</i></b>	<b>Apa yang Termasuk di Dalam</b>	<b>Apa yang Tidak Termasuk / Dipegang BC Lain</b>
<b><i>Lending BC (Core)</i></b>	<i>Loan, LoanStatus, DueDate</i> , aturan peminjaman, <i>behavior</i> peminjaman ( <i>borrow, returnBook, extendLoan, markOverdue</i> ), <i>LoanPolicyService, LoanRepository</i>	Detail buku, detail user, notifikasi, autentikasi
<b><i>Book Management BC (Catalog)</i></b>	Informasi buku (judul, penulis, ISBN), metadata buku, status ketersediaan	Peminjaman, pengembalian, <i>due date</i> , identitas pengguna
<b><i>Notification BC</i></b>	<i>Event listener</i> , pengirim notifikasi, template notifikasi	Peminjaman, manajemen buku, autentikasi
<b><i>User Management BC (Identity &amp; Access)</i></b>	Data pengguna, <i>autentikasi, otorisasi, profile management</i>	Detail peminjaman, detail buku, pengelolaan notifikasi

Tabel 4.2 Pola Integrasi Antar *Bounded Context*

<b><i>Source BC</i></b>	<b><i>Target BC</i></b>	<b><i>Pattern</i></b>	<b><i>Makna Relasi</i></b>
<b><i>Lending BC (Core)</i></b>	<i>Book Management BC</i>	Customer–Supplier (Reference Model)	<i>Book Management</i> menyediakan struktur dan data buku. <i>Lending</i> hanya menggunakan <i>BookId</i> tanpa mengetahui detail buku.
<b><i>Lending BC</i></b>	<i>User Management BC</i>	Customer–Supplier (Reference Model)	<i>User Management</i> menyediakan data identitas pengguna. <i>Lending</i> hanya menggunakan <i>UserId</i> sebagai referensi.

<b>Lending BC</b>	<i>Notification BC</i>	<i>Published Language / Domain Event</i>	Lending BC menerbitkan event seperti <i>LoanCreated</i> , <i>LoanReturned</i> , <i>LoanOverdue</i> . <i>Notification BC</i> mendengarkan dan memproses event tersebut untuk mengirim notifikasi.
-------------------	------------------------	--	--

Dalam pendekatan Domain-Driven Design (DDD), **bounded context** digunakan untuk memetakan batas tanggung jawab setiap domain di sistem **BookWise**. Setiap bounded context memiliki peran spesifik, dan interaksi antar konteks sering mengikuti pola tertentu. Salah satu konsep penting adalah **upstream dan downstream**, yang menunjukkan arah ketergantungan antar domain. **Upstream** adalah bounded context yang menyediakan model atau informasi, sementara **downstream** adalah bounded context yang mengonsumsi model tersebut tanpa mengubahnya. Dalam **BookWise**, **Catalog / Book Management BC** dan **User Management BC** berperan sebagai upstream karena mereka menetapkan struktur utama buku dan pengguna. Sebaliknya, **Lending BC** menjadi downstream karena menggunakan model buku dan identitas pengguna untuk proses peminjaman, pengembalian, dan perpanjangan pinjaman, namun tidak memiliki otoritas untuk mengubah model tersebut. Pola integrasi yang tepat untuk hubungan ini adalah **Customer–Supplier**, di mana upstream bertindak sebagai supplier model dan downstream sebagai customer yang mengonsumsinya.

Berbeda dengan pola *Customer–Supplier*, hubungan antara **Lending BC** dan **Notification BC** menggunakan pola **Published Language**, karena yang dibagikan bukan model domain, melainkan *event*. *Notification BC* hanya mendengarkan *event* untuk mengirim notifikasi, tanpa memerlukan penyesuaian model atau bahasa domain. Dalam pola ini, konsep *upstream downstream* tidak relevan, dan aliran informasi cukup direpresentasikan sebagai event. Dengan demikian, kombinasi pola *Customer–Supplier* dan *Published Language* memungkinkan sistem *BookWise* menjaga konsistensi data dan aliran logika, sekaligus memisahkan tanggung jawab inti dan layanan pendukung secara jelas.

## 5. Arsitektur Aplikasi dan Implementasi

*Class diagram BookWise* disusun berdasarkan prinsip *Domain-Driven Design (DDD)* dengan mengidentifikasi entitas utama, *value object*, serta hubungan antar kelas yang mencerminkan proses bisnis pada sistem peminjaman buku digital. Berikut adalah identifikasi kelas-kelas yang terdapat dalam sistem **BookWise** beserta jenis dan deskripsinya:

Tabel 5.1 *Identifikasi Class Diagram*

<b>Nama Kelas</b>	<b>Jenis Kelas</b>	<b>Deskripsi</b>
<b><i>Loan</i></b>	<i>Aggregate Root / Entity</i>	Entitas utama peminjaman yang mengelola seluruh state pinjaman.
<b><i>LoanStatus</i></b>	<i>Value Object</i>	Merepresentasikan status peminjaman ( <i>Requested, Borrowed, Returned, Overdue</i> ).
<b><i>DueDate</i></b>	<i>Value Object</i>	Batas waktu pengembalian serta logika <i>overdue</i> .
<b><i>BookId</i></b>	<i>Value Object</i>	Identitas buku yang dipinjam, sebagai referensi ke <i>Catalog BC</i> .
<b><i>UserId</i></b>	<i>Value Object</i>	Identitas pengguna yang meminjam, sebagai referensi ke <i>User Management BC</i> .
<b><i>LoanRepository</i></b>	<i>Repository</i>	Antarmuka untuk menyimpan dan mengambil <i>Loan</i> .
<b><i>LoanPolicyService</i></b>	<i>Domain Service</i>	Mengelola aturan peminjaman seperti perhitungan <i>due date</i> atau <i>validasi policy</i> .

Berdasarkan Tabel 5.1, struktur domain *Loan* menjadi jelas. ***Loan*** sebagai *aggregate root* menjaga *state* peminjaman melalui *behavior*. Behavior ini memastikan aturan domain tetap valid, misalnya status awal harus ***Requested***, transisi status harus sesuai urutan, ***DueDate*** tidak boleh di masa lalu, dan pengembalian hanya dapat dilakukan jika status sudah ***Borrowed***. Value object seperti ***LoanStatus***, ***DueDate***, ***BookId***, dan ***UserId*** menjaga konsistensi data, sementara ***LoanRepository*** dan ***LoanPolicyService*** memisahkan logika penyimpanan dan bisnis.

Tabel 5.2 *Glosarium Ubiquitous Language (Loan BC)*

<b>Istilah</b>	<b>Deskripsi</b>
<i>Loan</i>	Transaksi peminjaman buku.
<i>BookId</i>	Identitas unik buku.
<i>UserId</i>	Identitas unik pengguna.
<i>LoanStatus</i>	Status pinjaman.
<i>DueDate</i>	Tenggat pengembalian.
<i>Borrow</i>	Proses membuat pinjaman aktif.

<i>ReturnBook</i>	Proses mengembalikan buku.
<i>Overdue</i>	Kondisi lewat batas waktu.
<i>LoanRepository</i>	Tempat simpan Loan.
<i>LoanPolicy</i>	Kebijakan peminjaman.

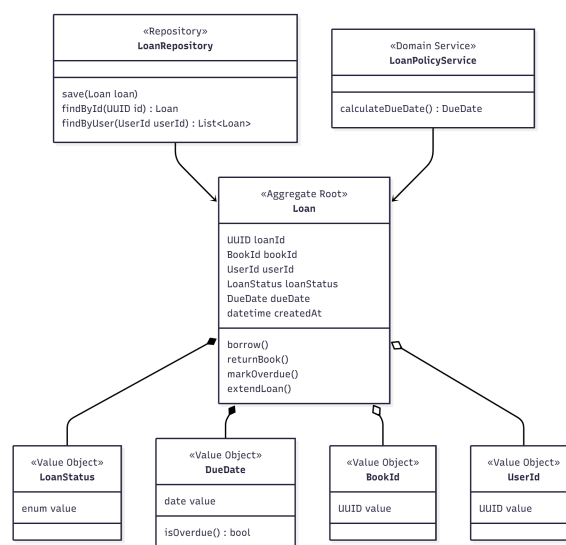
Tabel 5.2 menyajikan glosarium Ubiquitous Language yang digunakan dalam *Loan Bounded Context*. Glosarium ini berfungsi sebagai panduan istilah yang konsisten antara tim pengembang, analis, dan stakeholder, sehingga setiap konsep dalam domain memiliki makna yang sama dan tidak menimbulkan ambiguitas. Kehadiran Ubiquitous Language ini juga memastikan bahwa *class diagram* yang dikembangkan benar-benar mencerminkan proses bisnis. Dengan demikian, model yang dihasilkan menjadi lebih mudah dipahami, terdokumentasi dengan jelas, serta mendukung pengembangan berkelanjutan pada domain Loan. Setelah istilah utama didefinisikan, tahap berikutnya adalah melihat bagaimana tiap entitas saling berhubungan dalam domain. Relasi tersebut dirangkum pada tabel berikut.

Tabel 5.3 Relasi antar *Entity/Value Object*

Entity 1	Entity 2	Kardinalitas	Tipe Relasi UML	Penjelasan
<i>Loan</i>	<i>LoanStatus</i>	1 : 1	Komposisi	<i>Loan</i> memiliki status, dan status tidak hidup terpisah.
<i>Loan</i>	<i>DueDate</i>	1 : 1	Komposisi	<i>DueDate</i> bagian dari <i>Loan</i> dan tidak berdiri sendiri.
<i>Loan</i>	<i>BookId</i>	1 : 1	Agregasi	<i>Loan</i> menggunakan <i>BookId</i> , tapi <i>BookId</i> berasal dari BC lain.
<i>Loan</i>	<i>UserId</i>	1 : 1	Agregasi	Sama: <i>UserId</i> adalah identitas dari BC eksternal.
<i>LoanRepository</i>	<i>Loan</i>	1 : N	Asosiasi	<i>Repository</i> menyimpan banyak <i>Loan</i> .
<i>LoanPolicyService</i>	<i>Loan</i>	1 : N	Asosiasi	<i>Service</i> tidak memiliki <i>Loan</i> , hanya memakai.

Dari relasi pada Tabel 5.3 terlihat bahwa domain *Loan* memisahkan dengan jelas mana komponen yang menjadi bagian inti entitas dan mana yang hanya digunakan sebagai referensi. Komposisi digunakan untuk elemen yang bergantung sepenuhnya pada *Loan*, sementara agregasi menunjukkan objek dari *bounded context* lain. Asosiasi dipakai untuk komponen yang hanya berinteraksi tanpa memiliki entitas tersebut.

Berdasarkan identifikasi kelas dan relasi yang telah dijelaskan berikut adalah visualisasi lengkap *class diagram* sistem **BookWise**.



Gambar 5.4 *Class Diagram* Book Wise

*Class diagram* pada Gambar 5.4 menunjukkan struktur lengkap sistem **BookWise** dengan atribut dan method pada setiap kelas, serta garis relasi yang merepresentasikan interaksi antar kelas sesuai dengan kardinalitas dan tipe relasi yang telah didefinisikan. Diagram ini menjadi blueprint teknis untuk implementasi kode program dan memastikan bahwa setiap komponen sistem memiliki tanggung jawab yang jelas sesuai dengan domain masing-masing. Dengan desain *class diagram* yang modular dan mengikuti prinsip DDD, sistem **BookWise** dapat dikembangkan secara terstruktur, mudah dipelihara, dan siap untuk dikembangkan lebih lanjut sesuai kebutuhan bisnis di masa mendatang.