

**TUGAS BESAR TAHAP - 3 SISTEM MODEL *BOOK WISE*  
TEKNOLOGI SISTEM TERINTEGRASI**

Mata Kuliah : II3160 Teknologi Sistem Terintegrasi  
Dosen : Daniel Wiyogo Dwiputro, S.T., M.T.



Disiapkan oleh  
Nama : Laras Hati Mahendra  
NIM : 18223118

**PROGRAM STUDI SISTEM DAN TEKNOLOGI INFORMASI  
FAKULTAS SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
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. Arsitektur Aplikasi dan Implementasi.....</b>	<b>5</b>

## **Daftar Gambar**

Gambar 5.4 Class Diagram Book Wise

8

## **Daftar Tabel**

Tabel 5.1 Identifikasi Class Diagram	5
Tabel 5.2 Glosarium Ubiquitous Language (Loan BC)	6
Tabel 5.3 Relasi antar Entity/Value Object	7

Sebagai kelanjutan dari desain kontekstual pada Milestone 2, pembahasan berikutnya mengenai arsitektur teknis BookWise. Intinya, bagian ini menjembatani rancangan konseptual yang dibuat sebelumnya dengan implementasi, supaya setiap domain, terutama Loan sebagai inti peminjaman, bisa bekerja terstruktur, dan bisa dikembangkan.

## 1. 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 BookWise. Berikut adalah identifikasi kelas-kelas yang terdapat dalam sistem BookWise beserta jenis, atribut, dan methodnya:

Tabel 1.1 *Identifikasi Class Diagram*

Nama Kelas	Jenis Kelas	Atribut (Tipe Data)	Method (Acces)
<b>Loan</b>	Entity	borrowedId: String (private) bookId: String (private) userId: String (private) loanStatus: Enum (private) dueDate: Date (private)	-
<b>LoanProcessingService</b>	Service	-	createLoan() (public) verifyLoan() (public) distributeBook() (public)
<b>ReturnProcessingService</b>	Service	-	initiateReturn() (public) verifyReturn() (public) finalizeReturn() (public)
<b>LoanExtensionService</b>	Service	-	verifyLoanExtension()

Berdasarkan Tabel 1.1, struktur domain Loan menjadi jelas. Loan sebagai aggregate root menjaga state peminjaman melalui behavior yang ada, memastikan

proses peminjaman, pengembalian, dan perpanjangan sesuai alur BookWise. Atribut entity dibuat private supaya data internal terlindungi, dan setiap perubahan hanya bisa dilakukan melalui method resmi dari Lending Loan Processing BC. Hal ini menjamin konsistensi data dan memastikan status pinjaman selalu valid.

Tabel 1.2 *Glosarium Ubiquitous Language (Loan BC)*

Istilah	Definisi / Penjelasan
<b>Loan</b>	Entitas yang merepresentasikan pinjaman buku oleh pengguna.
<b>BorrowedId</b>	Identifier unik transaksi pinjaman.
<b>BookId</b>	Identifier unik buku yang dipinjam.
<b>UserId</b>	Identifier unik pengguna yang meminjam buku.
<b>LoanStatus</b>	Status pinjaman, misal PENDING, APPROVED, RETURNED, OVERDUE.
<b>DueDate</b>	Tanggal jatuh tempo pengembalian buku.
<b>LoanProcessingService</b>	Service yang mengatur proses pengajuan, verifikasi, dan penyaluran pinjaman.
<b>ReturnProcessingService</b>	Service yang mengatur proses pengembalian buku dan penyelesaian pinjaman.
<b>LoanExtensionService</b>	Service yang mengatur proses perpanjangan peminjaman buku.
<b>CreateLoan</b>	Method untuk mengajukan pinjaman baru.
<b>VerifyLoan</b>	Method untuk memverifikasi kelayakan pinjaman.
<b>DistributeBook</b>	Method untuk menyalurkan buku ke peminjam setelah disetujui.
<b>InitiateReturn</b>	Method untuk memulai proses pengembalian buku.
<b>VerifyReturn</b>	Method untuk memeriksa kondisi buku dan status pengembalian.

<b>FinalizeReturn</b>	Method untuk menyelesaikan dan mencatat pengembalian buku.
<b>VerifyLoanExtension</b>	Method untuk memverifikasi dan menyetujui perpanjangan pinjaman.

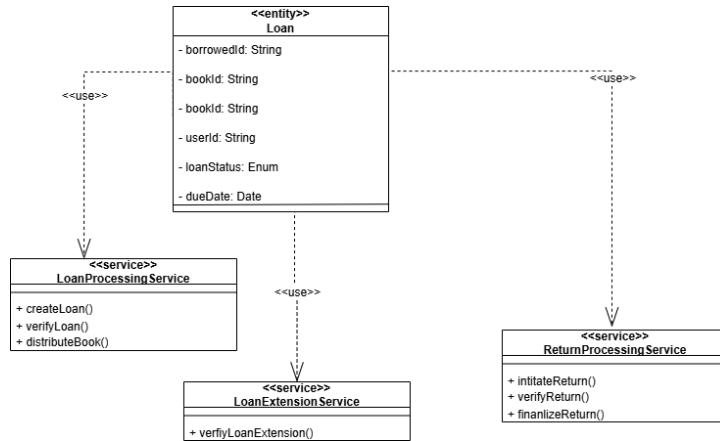
Tabel 1.2 menampilkan glosarium Ubiquitous Language dalam Loan Bounded Context, yang menjamin istilah konsisten dan class diagram sesuai proses bisnis. Setelah istilah utama dijelaskan, langkah berikutnya adalah memeriksa relasi antar entitas, yang dirangkum pada tabel berikut.

Tabel 1.3 Relasi antar *Entity/Value Object*

<b>Relasi</b>	<b>Tipe Relasi UML</b>	<b>Penjelasan</b>
<i>Loan</i> → <i>LoanProcessingService</i>	Usage(Dependency)	Service bergantung pada <i>Loan</i> untuk proses pinjaman.
<i>Loan</i> → <i>ReturnProcessingService</i>	Usage(Dependency)	Service bergantung pada <i>Loan</i> untuk proses pengembalian.
<i>Loan</i> → <i>LoanExtensionService</i>	Usage(Dependency)	Service bergantung pada <i>Loan</i> untuk memverifikasi perpanjangan.

Dari relasi pada Tabel 1.3 terlihat bahwa semua service bergantung pada entity *Loan* melalui dependency. *Loan* menjadi pusat yang mengendalikan state, sementara service hanya menggunakan entity untuk menjalankan behavior, tanpa komposisi, agregasi, atau pewarisan.

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



Gambar 1.4 *Class Diagram Book Wise*

Class diagram ini menampilkan struktur Lending Loan Processing BC pada BookWise. Loan berperan sebagai aggregate root yang mengendalikan state peminjaman, termasuk status, tanggal jatuh tempo, dan identitas peminjam dan buku. Semua service (LoanProcessingService, ReturnProcessingService, LoanExtensionService) memiliki dependency terhadap Loan, artinya mereka menggunakan entity Loan untuk menjalankan behavior masing-masing (pengajuan, pengembalian, dan perpanjangan pinjaman). Tidak terdapat relasi agregasi, komposisi, pewarisan, atau asosiasi lain, sehingga alur domain tetap sederhana dan terfokus pada pengelolaan pinjaman.