

PROPOSAL TUGAS BESAR
PEMROGRAMAN BERORIENTASI OBJEK

KELAS : IT-47-04
TAHUN AKADEMIK : GANJIL 2025/2026



Oleh:

ID KELOMPOK : 3
ANGGOTA :

- 1. Syahril Arfian Almazril (103032300013)**
- 2. Shania Rahmalia (103032300018)**
- 3. Rara Nur Annisa (103032300077)**
- 4. Galuh Ajeng (103032300087)**
- 5. Neng Intan Nuraeini (103032330031)**

S1 TEKNOLOGI INFORMASI
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM

2025/2026

DAFTAR ISI

A. Deskripsi Singkat.....	2
B. Daftar Fitur Utama.....	2
C. Rancangan Kelas	3
D. Rincian Kelas.....	5
E. Pembagian Tugas	5
F. Rancangan Entity Relationship Diagram (ERD).....	6
G. Tools	6
H. Lampiran.....	7

A. Deskripsi Singkat

Pengelolaan perpustakaan yang efektif merupakan bagian penting dalam mendukung proses pembelajaran di lingkungan akademik modern. Untuk menjawab kebutuhan tersebut, kami mengembangkan Digilibz, sebuah aplikasi Digital Library System yang dibangun dengan pendekatan Pemrograman Berorientasi Objek (Object Oriented Programming/OOP).

Aplikasi Digilibz hadir sebagai solusi digital terpadu untuk mempermudah proses administrasi, peminjaman, serta pengelolaan koleksi buku di perpustakaan. Melalui sistem ini, pengguna yang terdiri dari mahasiswa dan dosen dapat dengan mudah meminjam, mengembalikan, serta memberikan ulasan terhadap buku secara digital. Sementara itu, admin memiliki kontrol penuh untuk mengelola data buku, memverifikasi transaksi, dan menghasilkan laporan analitik secara efisien dan terstruktur.

Dibangun dengan konsep OOP seperti *encapsulation*, *inheritance*, *polymorphism*, dan *abstraction*, aplikasi ini mengedepankan arsitektur yang modular, terorganisir, serta mudah dikembangkan. Setiap komponen dalam sistem, mulai dari entitas pengguna, buku, hingga transaksi diimplementasikan sebagai objek yang saling berinteraksi sesuai prinsip desain berorientasi objek, sehingga sistem menjadi lebih terstruktur, reusable, dan maintainable.

Dengan antarmuka yang sederhana dan mudah digunakan, Digilibz tidak hanya memudahkan pengelolaan data perpustakaan, tetapi juga menjadi contoh nyata penerapan konsep OOP dalam pengembangan perangkat lunak. Aplikasi ini diharapkan dapat menciptakan ekosistem perpustakaan cerdas (Smart Library System) yang efisien, aman, serta mampu beradaptasi terhadap kebutuhan digitalisasi kampus.

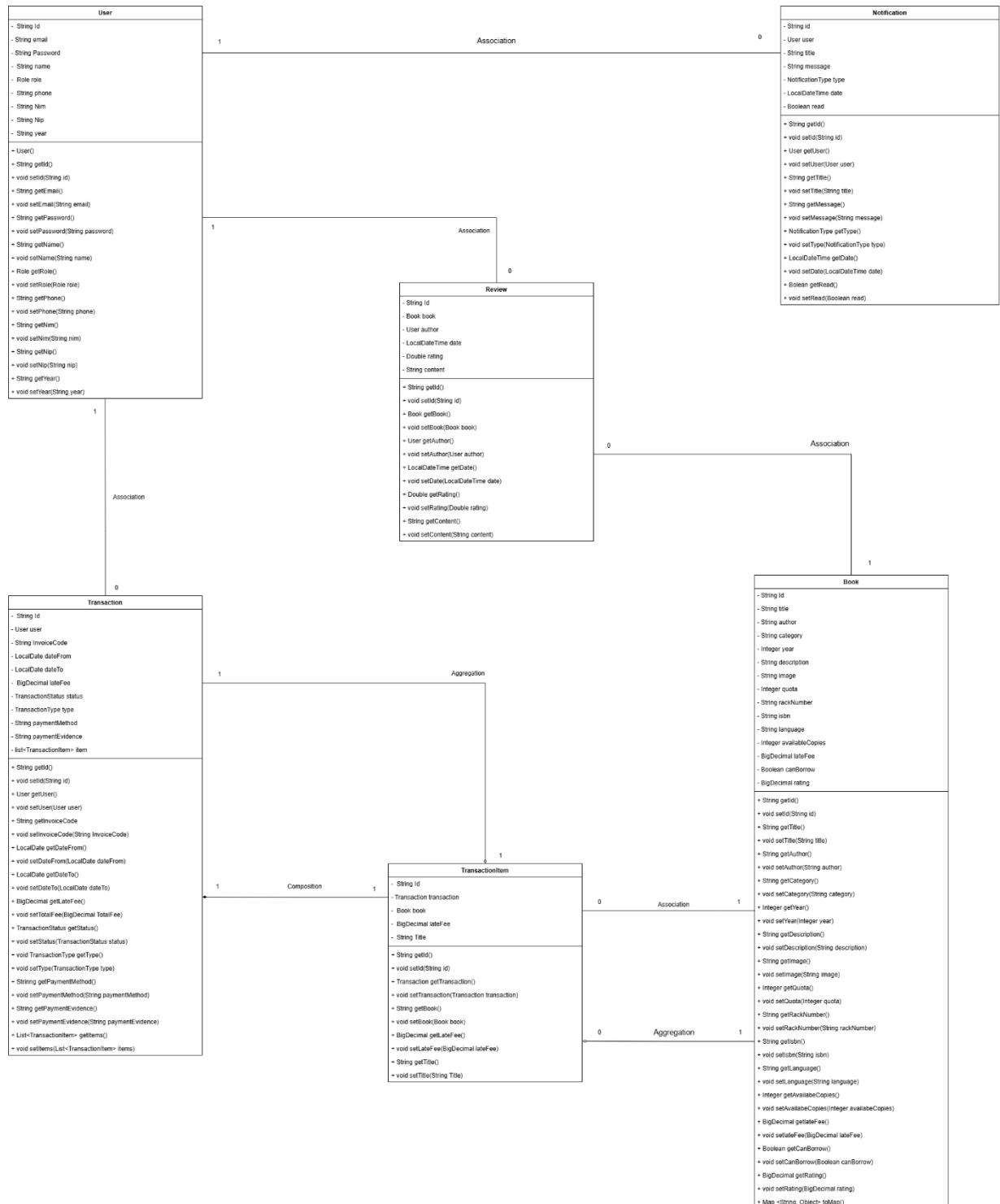
B. Daftar Fitur Utama

No.	Nama Fitur	Deskripsi
1.	Manajemen Buku	Admin dapat menambah, melihat, memperbarui, dan menghapus data buku (CRUD).
2.	Pencarian & Filter Buku	Pengguna dapat mencari buku berdasarkan judul/penulis dan memfilter berdasarkan kategori atau tahun terbit.
3.	Rekomendasi Buku	Sistem menampilkan rekomendasi buku secara acak.
4.	Manajemen Pengguna	Admin dapat mengelola data pengguna (mahasiswa, dosen, admin). Pengguna dapat mendaftar dan login sesuai perannya.
5.	Proses Peminjaman	Pengguna dapat meminjam buku fisik dengan memilih rentang tanggal
6.	Manajemen Transaksi	Admin dapat menyetujui atau menolak permintaan peminjaman dan mengelola proses pengembalian buku.
7.	Review & Rating Buku	Pengguna dapat memberikan ulasan dan rating untuk buku yang telah dipinjam.
8.	Notifikasi Pengguna	Sistem mengirimkan notifikasi kepada pengguna terkait status transaksi, pengingat, atau informasi lainnya.

9.	Dashboard Admin	Admin dapat melihat statistik ringkasan seperti jumlah total buku, pengguna, transaksi, dan review terbaru.
10.	Bookmark Buku	Pengguna dapat menyimpan buku favorit untuk dilihat nanti.

C. Rancangan Kelas

1. Gambar Class Diagram



2. Penjelasan Hubungan antar Kelas

2.1 Hubungan antara User dan Notification

Hubungan antara User dan Notification merupakan Association kardinalitas $1 \rightarrow 0$. Kelas User berhubungan dengan Notification karena notification hanya dimiliki oleh satu user. Dan relasi ini juga menunjukkan bahwa sistem dapat mengirimkan banyak notification kepada user, seperti pemberitahuan transaksi berhasil, pesan sistem, atau pembaruan lainnya.

2.2 Hubungan antara User dan Review

Hubungan antara User dan Review merupakan association dengan kardinalitas $1 \rightarrow 0$. Kelas user terhubung dengan review karena setiap user dapat menulis banyak ulasan terhadap buku yang telah dibaca atau dibeli. Hubungan ini memungkinkan sistem menampilkan daftar ulasan yang telah dibuat oleh seorang user, serta menghubungkan setiap review dengan akun penulisnya.

2.3 Hubungan antara User dan Transaction

Hubungan antara User dan Transaction merupakan association dengan kardinalitas $1 \rightarrow 0$. Kelas User berhubungan dengan kelas Transaction karena satu user dapat melakukan banyak transaction pembelian e-book, namun setiap transaction hanya dilakukan oleh satu user.

2.4 Hubungan antara Review dan Book

Hubungan antara review dan book merupakan association dengan kardinalitas $1 \rightarrow 0$. Kelas Review memiliki hubungan dengan kelas book, karena setiap review ditulis untuk satu buku tertentu, dan satu buku juga dapat memiliki banyak review dari user yang berbeda. Hubungan ini memungkinkan sistem menampilkan seluruh review yang terkait dengan satu buku, serta menghitung rata-rata rating buku berdasarkan ulasan yang diterima.

2.5 Hubungan antara Transaction dan TransactionItem

Hubungan antara transaction dan transactionitem merupakan aggregation dengan kardinalitas $1 \rightarrow 0$. Kelas transaction memiliki relasi agregasi terhadap transactionitem, yang berarti bahwa setiap transaksi dapat terdiri dari beberapa item dan menjelaskan item-item buku apa saja yang akan user beli.

2.6 Hubungan antara TransactionItem dan Book

Hubungan antara transactionitem dan book merupakan aggregation dengan kardinalitas $1 \rightarrow 0$. Setiap transactionItem mengacu pada satu book tertentu yang akan dibeli oleh user dalam transaksi tersebut. Namun, satu buku dapat muncul berbagai detail transaksi yang berbeda-beda. Misalnya, buku yang sama banyak dibeli oleh user. Relasi aggregation ini menunjukkan bahwa objek book tidak bergantung pada transactionItem, tapi menjadi referensi didalamnya.

2.7 Hubungan antara Transaction dan TransactionItem

Hubungan antara Transaction dan TransactionItem merupakan composition dengan kardinalitas $1 \rightarrow 1$. Setiap transaction memiliki tepat satu TransactionItem yang menjadi bagian tak terpisahkan dari transaction tersebut. Karena relasi nya composition, keberadaan TransactionItem sepenuhnya bergantung pada transaction. Jika transaction dihapus, maka transactionItem juga ikut terhapus. Dengan kata lain, TransactionItem tidak bisa berdiri sendiri tanpa transaction.

2.8 Hubungan antara Transaction dan Book

Hubungan antara Transaction dan Book merupakan aggregation (tidak langsung/ melalui transactionItem dengan kardinalitas $1 \rightarrow 1$). Relasi ini menunjukkan bahwa satu transaksi bisa mengagregasi informasi dari satu atau beberapa buku melalui transactionItem. Transaction tidak memiliki hubungan langsung dengan book, melainkan melalui transactionItem sebagai perantara.

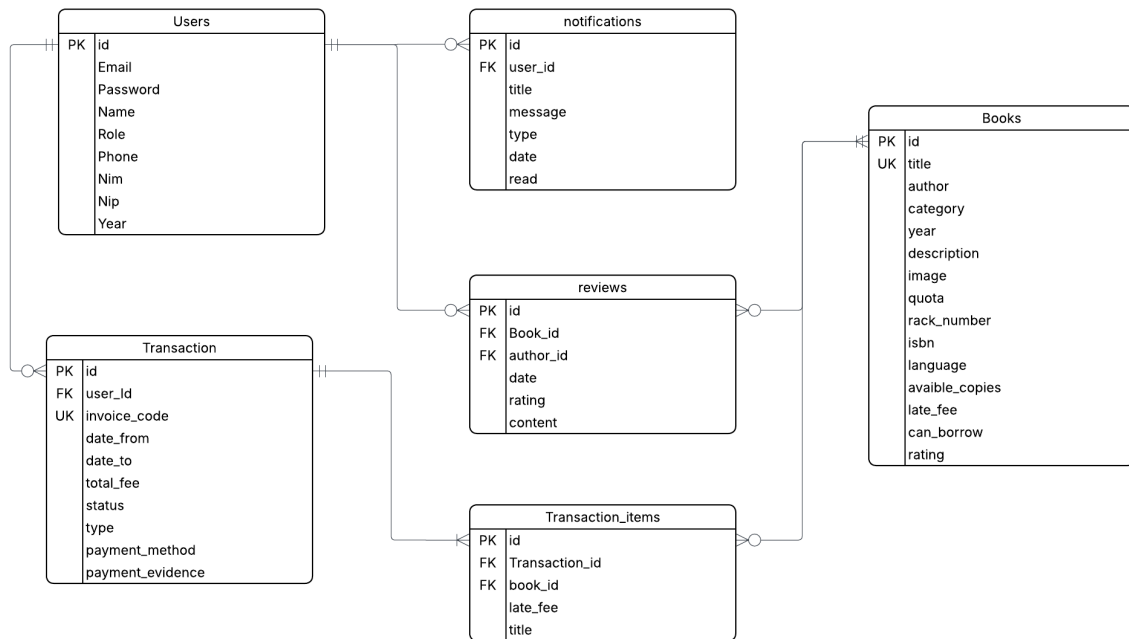
D. Rincian Kelas

No	Nama Kelas	Deskripsi Singkat
1.	User	Kelas User merepresentasikan akun pengguna Digilibz yang memuat identitas dasar yang secara fungsional digunakan untuk sumber autentikasi dan penetapan hak akses.
2.	Notification	Kelas Notification merepresentasikan pesan sistem yang ditujukan ke seorang User di Digilibz.
3.	Review	Kelas Review merepresentasikan ulasan yang ditulis pengguna terhadap sebuah Book di Digilibz. Setiap Review terhubung ke satu Book dan ditulis oleh satu User (author). Secara fungsional, Review ini dipakai untuk menampilkan daftar ulasan perbuku.
4.	Transaction	Kelas Transaction merekam aktivitas di Digilibz mulai dari proses pemesanan peminjaman/penyewaan buku secara digital hingga pelunasan denda dengan mengikat satu transaksi pada satu User, menyertakan kode invoice, rentang tanggal, jenis dan status transaksi, metode serta bukti pembayaran, nilai total yang ditagihkan.
5.	Transaction Item	Kelas Transaction Item merepresentasikan baris item di dalam transaksi Digilibz. Setiap Itemnya terikat pada satu Transaction dan merujuk ke satu Book sebagai objek yang ditransaksikan. Dengan adanya rancangan kelas ini, satu Transaction dapat berisi banyak TransactionItem sementara tiap item merangkum detail minimal yang diperlukan agar proses checkout, perhitungan denda, dan pelacakan transaksi berjalan dengan benar/konsisten.
6.	Book	Kelas Book merepresentasikan koleksi dari Digilibz sehingga dapat menjadi sumber informasi, data, dan tampilan katalog. Secara fungsionalitas, Book dipakai untuk fitur pencarian & filter, halaman detail buku, perhitungan ketersediaan saat checkout, serta peringatan denda.

E. Pembagian Tugas

No	PIC	Fitur	Kelas yang Dibuat
1.	Syahril Arfian Almazril	Mengelola semua data pengguna (termasuk otentikasi) dan sistem notifikasi.	User, Notification
2.	Shania Rahmalia	Mengelola fungsionalitas ulasan dan rating buku oleh pengguna.	Review
3.	Rara Nur Annisa	Mengelola proses inti peminjaman, termasuk pembuatan invoice dan status peminjaman.	Transaction
4.	Galuh Ajeng	Mengelola data detail buku apa saja yang ada di dalam satu transaksi.	TransactionItem
5.	Neng Intan Nuraeni	Mengelola semua data master buku, mencakup fungsionalitas serta penyajian data untuk pengguna.	Book

F. Rancangan Entity Relationship Diagram (ERD)



G. Tools

Pengembangan website Digilibz dilakukan dengan pendekatan OOP yang memanfaatkan kombinasi framework modern yang mendukung efisiensi pengembangan dan skalabilitas system. Berikut adalah alat (*tools*) dan teknologi yang digunakan dalam pembangunan sistem ini:

1. Backend

Pada pengembangan di bagian Backend digunakan framework Spring Boot untuk mengembangkan website secara cepat, efisien, dan terstruktur. Framework ini mendukung konsep OOP seperti abstraction, inheritance, dan encapsulation, serta menyediakan REST API yang menghubungkan data antara sistem backend dan frontend dengan aman dan konsisten.

2. Frontend

Pada pengembangan di bagian Frontend digunakan framework Next.js karena dapat menghasilkan tampilan antarmuka yang interaktif, responsif, dan modern.

3. Database

Pada pengembangan website Digilibz sebagai Sistem Manajemen Perpustakaan Digital, digunakan SQL sebagai sistem manajemen basis data utama. Database ini yang akan menyimpan seluruh data seperti informasi pengguna, buku digital, dan riwayat peminjaman.

4. Git dan GitHub

Penggunaan Git sebagai kontrol untuk mencatat setiap perubahan kode yang dilakukan anggota tim, sedangkan GitHub digunakan sebagai repositori online tempat menyimpan, berbagi, dan mengelola proyek secara kolaboratif sehingga setiap anggota dapat berkontribusi secara tertata.

5. Diagram Tools

Tools yang digunakan dalam pembuatan Class Diagram dan Entity Relationship Diagram (ERD) adalah draw.io agar membantu kami memahami hubungan antar kelas dan fitur sebelum tahap pengembangan kode lebih lanjut.

6. Visual Studio (VS) Code

Visual Studio Code digunakan sebagai kode editor utama dalam pengembangan website ini karena ringan, fleksibel, dan mendukung berbagai ekstensi.

H. Lampiran

Link Google Drive Diagram:

https://drive.google.com/drive/folders/1x0_PUPXaU6mCThoriSjle2TNAJVBHQX3D?usp=sharing