

## B. Perancangan Database

Berikut adalah desain database yang digunakan dalam sistem ini, dengan minimal 4 tabel yang saling berelasi:

### 1. Tabel users

Tabel ini digunakan untuk menyimpan informasi pengguna.

Field	Tipe Data	Keterangan
id	INT	Primary Key, Auto Increment
username	VARCHAR	Nama pengguna (unik)
email	VARCHAR	Email pengguna (unik)
password	VARCHAR	Password pengguna
role	ENUM	Peran pengguna (user atau admin)

### 2. Tabel books

Tabel ini menyimpan informasi tentang buku yang tersedia di perpustakaan.

Field	Tipe Data	Keterangan
id	INT	Primary Key, Auto Increment
title	VARCHAR	Judul buku
author	VARCHAR	Penulis buku
genre	VARCHAR	Genre buku
year_published	INT	Tahun penerbitan
quantity_available	INT	Jumlah buku yang tersedia

### 3. Tabel borrowed\_books

Tabel ini menyimpan informasi tentang buku yang sedang dipinjam oleh pengguna.

Field	Tipe Data	Keterangan
id	INT	Primary Key, Auto Increment
user_id	INT	Foreign Key (mengacu ke tabel users)
book_id	INT	Foreign Key (mengacu ke tabel books)
borrow_date	DATETIME	Tanggal peminjaman
return_date	DATETIME	Tanggal pengembalian
status	ENUM	Status peminjaman (borrowed, returned)

### 4. Tabel reviews

Tabel ini digunakan untuk menyimpan ulasan yang diberikan oleh pengguna terhadap buku.

Field	Tipe Data	Keterangan
id	INT	Primary Key, Auto Increment
user_id	INT	Foreign Key (mengacu ke tabel users)
book_id	INT	Foreign Key (mengacu ke tabel books)
rating	INT	Rating buku antara 1 sampai 5
comment	TEXT	Komentar atau ulasan dari pengguna
review_date	DATETIME	Tanggal ulasan dibuat

Relasi antar tabel:

- Tabel users berelasi dengan tabel borrowed\_books dan reviews melalui user\_id.
- Tabel books berelasi dengan tabel borrowed\_books dan reviews melalui book\_id.

### C. Perancangan Endpoint

Berikut adalah minimal 20 endpoint untuk REST API sistem manajemen perpustakaan:

1. **POST** /api/v1/auth/register - Registrasi pengguna baru
2. **POST** /api/v1/auth/login - Login pengguna dan mendapatkan JWT
3. **GET** /api/v1/books - Menampilkan daftar buku
4. **POST** /api/v1/books - Menambahkan buku baru
5. **GET** /api/v1/books/{book\_id} - Menampilkan detail buku
6. **PUT** /api/v1/books/{book\_id} - Mengupdate detail buku
7. **DELETE** /api/v1/books/{book\_id} - Menghapus buku
8. **POST** /api/v1/borrow - Meminjam buku
9. **PUT** /api/v1/borrow/{borrow\_id} - Mengembalikan buku
10. **GET** /api/v1/borrowed-books - Menampilkan daftar buku yang dipinjam
11. **GET** /api/v1/borrowed-books/{borrow\_id} - Menampilkan detail peminjaman
12. **POST** /api/v1/reviews - Menambahkan ulasan untuk buku
13. **GET** /api/v1/reviews/{book\_id} - Menampilkan ulasan buku
14. **GET** /api/v1/users/{user\_id} - Menampilkan data pengguna
15. **PUT** /api/v1/users/{user\_id} - Mengupdate data pengguna
16. **DELETE** /api/v1/users/{user\_id} - Menghapus pengguna
17. **GET** /api/v1/users - Menampilkan daftar semua pengguna (admin only)
18. **POST** /api/v1/auth/logout - Logout dan menghapus token JWT
19. **GET** /api/v1/search-books - Mencari buku berdasarkan kata kunci
20. **POST** /api/v1/admin/ban-user - Admin membekukan akun pengguna

### D. Dokumentasi Backend API

**Endpoint: Registrasi Pengguna**

**Endpoint:** /api/v1/auth/register

**Method:** POST

**Body Parameter:**

- username: (string) Nama pengguna
- email: (string) Email pengguna
- password: (string) Password pengguna
- role: (string) Peran pengguna (user atau admin)

**Response:**

```
{  
  "message": "User registered successfully"
```

**Endpoint: Login Pengguna**

**Endpoint:** /api/v1/auth/login

**Method:** POST

**Body Parameter:**

- email: (string) Email pengguna
- password: (string) Password pengguna

**Response:**

```
{  
  "token": "JWT_Token_Here"}
```

**E. Integrasi API Publik**

API dapat mengintegrasikan beberapa layanan publik, seperti:

**Google Boks API:** Untuk mencari buku berdasarkan judul atau penulis.

- **Endpoint:** [https://www.googleapis.com/books/v1/volumes?q={search\\_query}](https://www.googleapis.com/books/v1/volumes?q={search_query})
- Menggunakan endpoint ini untuk mencari buku dari Google Books API dan menambahkannya ke dalam sistem.

**Open Library API:** Untuk mengakses informasi buku yang lebih lengkap.

- **Endpoint:** [https://openlibrary.org/search.json?q={search\\_query}](https://openlibrary.org/search.json?q={search_query})

**F. Middleware dan Validasi JWT**

**Middleware** digunakan untuk mengamankan endpoint yang membutuhkan autentikasi. Middleware akan memvalidasi JWT token di setiap request.

**JWT Middleware:**

```

const jwt = require('jsonwebtoken');
const authMiddleware = (req, res, next) => {
  const token = req.header('Authorization')?.replace('Bearer ', '');

  if (!token) {
    return res.status(401).json({ message: 'Access denied' });
  }

  try {
    const decoded = jwt.verify(token, process.env.JWT_SECRET);
    req.user = decoded;
    next();
  } catch (error) {
    res.status(400).json({ message: 'Invalid token' });
  }
};

```

## G. Pengujian Endpoint

- Pengujian dilakukan menggunakan **Postman** untuk menguji setiap endpoint API.
- **Test Cases:**
  1. Cek endpoint **POST /api/v1/auth/register** untuk registrasi pengguna baru.
  2. Cek endpoint **POST /api/v1/auth/login** untuk login dan mendapatkan token.
  3. Cek endpoint **GET /api/v1/books** untuk melihat daftar buku yang tersedia.
  4. Cek endpoint **POST /api/v1/books** untuk menambah buku baru ke perpustakaan.
  5. Cek endpoint **POST /api/v1/borrow** untuk meminjam buku.