

LAPORAN PERANCANGAN & DOKUMENTASI REST API

LIBRARY API SYSTEM (Berdasarkan Source Code)

1. Pendahuluan

Library API System merupakan aplikasi backend berbasis **Node.js** dan **Express.js** yang menyediakan layanan REST API untuk sistem perpustakaan. Sistem ini dirancang untuk mengelola data buku, anggota, proses peminjaman, serta autentikasi pengguna dengan menggunakan database MySQL.

REST API ini memungkinkan integrasi dengan aplikasi frontend maupun aplikasi pihak ketiga melalui komunikasi berbasis HTTP dan format data JSON.

2. Tujuan Pengembangan Sistem

Tujuan dari pengembangan Library API System adalah:

1. Menyediakan layanan API untuk manajemen data perpustakaan.
 2. Mengelola data buku dan anggota secara terpusat.
 3. Menangani proses peminjaman dan pengembalian buku.
 4. Menyediakan sistem autentikasi pengguna.
 5. Mengimplementasikan backend yang scalable dan mudah dikembangkan.
-

3. Arsitektur Sistem

Sistem menggunakan arsitektur **RESTful Client-Server** dengan komponen utama sebagai berikut:

3.1 Client

Client berupa aplikasi web atau mobile yang mengakses REST API.

3.2 Backend Server (Node.js + Express.js)

Backend server bertanggung jawab dalam:

- Menyediakan endpoint REST API
- Menjalankan business logic
- Mengelola koneksi database
- Melakukan autentikasi dan otorisasi

3.3 Database Server (MySQL)

Database digunakan untuk menyimpan data:

- Buku
- Anggota

- Transaksi peminjaman
 - Data pengguna (user/admin)
-

4. Struktur Project

Berdasarkan source code, struktur utama project adalah:

```
library-api/
├── routes/
│   ├── authRoutes.js
│   ├── bookRoutes.js
│   ├── memberRoutes.js
│   ├── borrowingRoutes.js
│   ├── externalRoutes.js
│   └── test-db.js
└── node_modules/
    └── package.json
└── server/app.js (atau index.js)
```

Folder **routes/** berisi seluruh endpoint REST API yang dikelompokkan berdasarkan fitur.

5. Modul dan Fungsionalitas Sistem

5.1 Modul Autentikasi (authRoutes.js)

Modul ini menangani proses login dan autentikasi pengguna.

Fungsi utama:

- Login user
- Validasi kredensial
- Pengelolaan session atau token

Contoh endpoint:

Method	Endpoint	Deskripsi
POST	/api/auth/login	Login pengguna
POST	/api/auth/register	Registrasi user

5.2 Modul Buku (bookRoutes.js)

Modul ini digunakan untuk mengelola data buku.

Fungsi utama:

- Menambah data buku
- Mengambil daftar buku
- Mengubah data buku
- Menghapus data buku

Contoh endpoint:

Method	Endpoint	Deskripsi
GET	/api/books	Mengambil semua buku
GET	/api/books/:id	Mengambil detail buku
POST	/api/books	Menambah buku baru
PUT	/api/books/:id	Mengubah data buku
DELETE	/api/books/:id	Menghapus data buku

5.3 Modul Anggota (memberRoutes.js)

Modul ini menangani manajemen data anggota perpustakaan.

Fungsi utama:

- Menambah anggota
- Melihat daftar anggota
- Mengubah data anggota
- Menghapus data anggota

Contoh endpoint:

Method	Endpoint	Deskripsi
GET	/api/members	Mengambil semua anggota
GET	/api/members/:id	Mengambil detail anggota
POST	/api/members	Menambah anggota
PUT	/api/members/:id	Mengubah data anggota

Method	Endpoint	Deskripsi
DELETE	/api/members/:id	Menghapus data anggota

5.4 Modul Peminjaman (borrowingRoutes.js)

Modul ini digunakan untuk mengelola transaksi peminjaman buku.

Fungsi utama:

- Membuat transaksi peminjaman
- Melihat data peminjaman
- Mengelola status peminjaman

Contoh endpoint:

Method	Endpoint	Deskripsi
POST	/api/borrowings	Membuat transaksi peminjaman
GET	/api/borrowings	Mengambil daftar peminjaman
GET	/api/borrowings/:id	Mengambil detail peminjaman

5.5 Modul External API (externalRoutes.js)

Modul ini digunakan untuk integrasi dengan sistem atau API eksternal, misalnya untuk mengambil data tambahan atau sinkronisasi data.

6. Desain Database (Konseptual)

Berdasarkan fungsionalitas sistem, terdapat beberapa tabel utama:

Tabel books

- id
- title
- author
- publisher
- year
- stock

Tabel members

- id

- name
- email
- phone
- address

Tabel borrowings

- id
- member_id
- book_id
- borrow_date
- return_date
- status

Tabel users

- id
- username
- password
- role

7. Keamanan Sistem

Keamanan sistem diimplementasikan melalui:

1. Autentikasi user (login)
2. Enkripsi password (bcrypt atau sejenis)
3. Penggunaan middleware untuk proteksi endpoint
4. Validasi input user
5. Pembatasan hak akses berdasarkan role

8. Contoh Alur Proses Peminjaman

1. User melakukan login melalui endpoint auth.
2. User memilih buku melalui endpoint books.
3. User melakukan peminjaman melalui endpoint borrowings.
4. Sistem menyimpan transaksi peminjaman ke database.
5. Sistem mengubah stok buku.

9. Penutup

Library API System dirancang sebagai backend RESTful API yang modular dan terstruktur dengan menggunakan Node.js, Express.js, dan MySQL. Sistem ini mampu mendukung pengelolaan perpustakaan secara digital serta memudahkan integrasi dengan berbagai aplikasi client.

Dokumentasi ini diharapkan dapat menjadi acuan dalam pengembangan, pemeliharaan, dan pengembangan lanjutan sistem.