
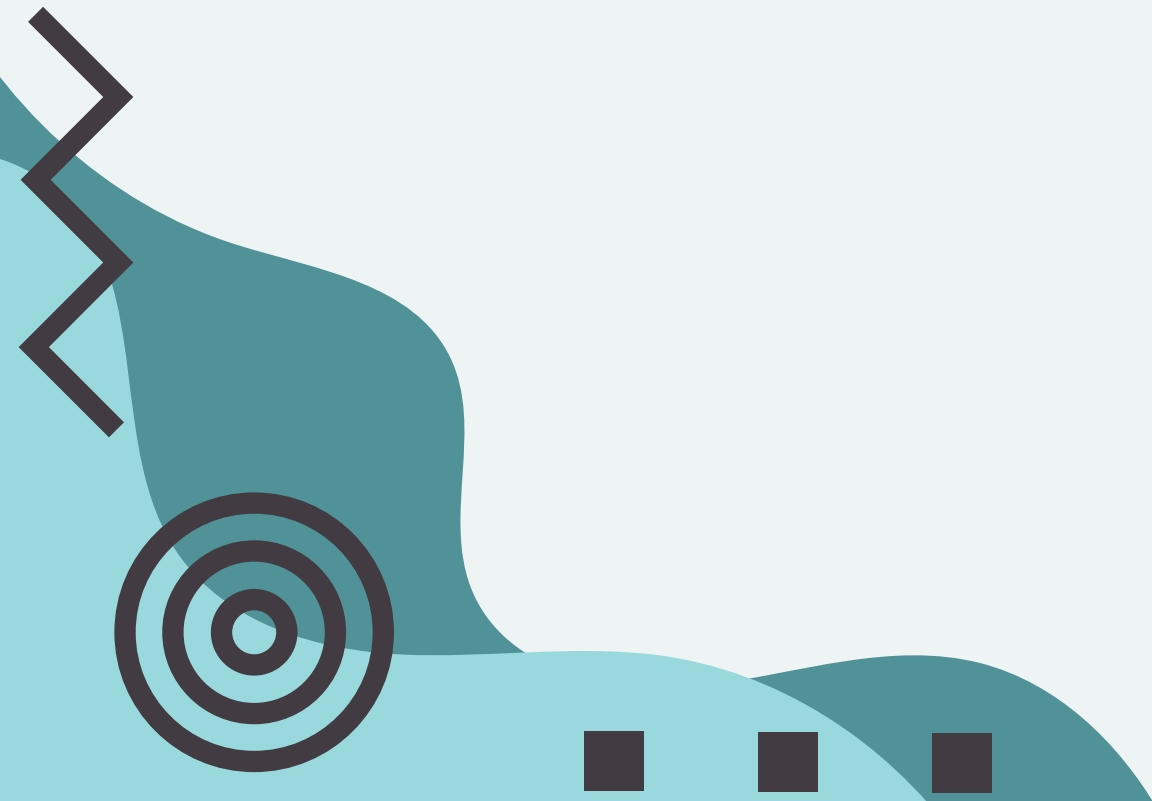
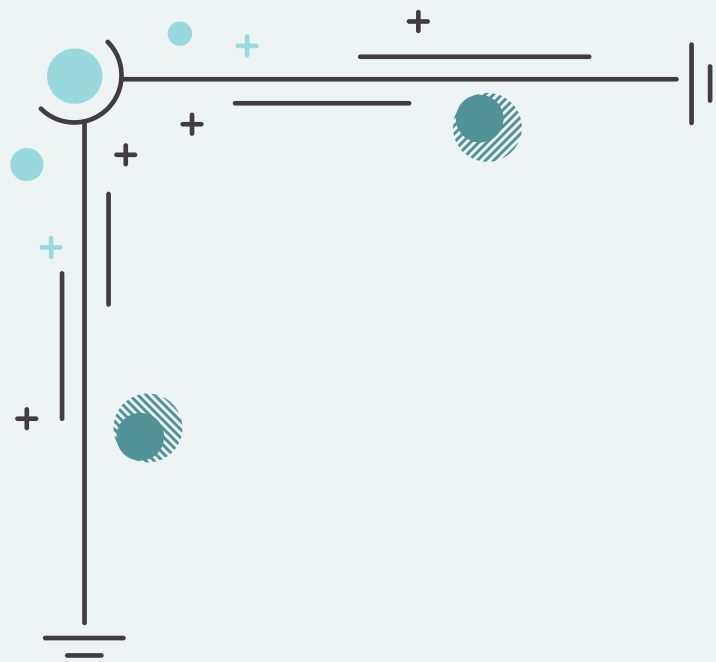




PERANCANGAN DAN IMPLEMENTASI REST API PERPUSTAKAAN ONLINE MENGUNAKAN NODE.JS DAN MYSQL

Disusun oleh :
Ridho Safutra
Farisa Azka Takiya



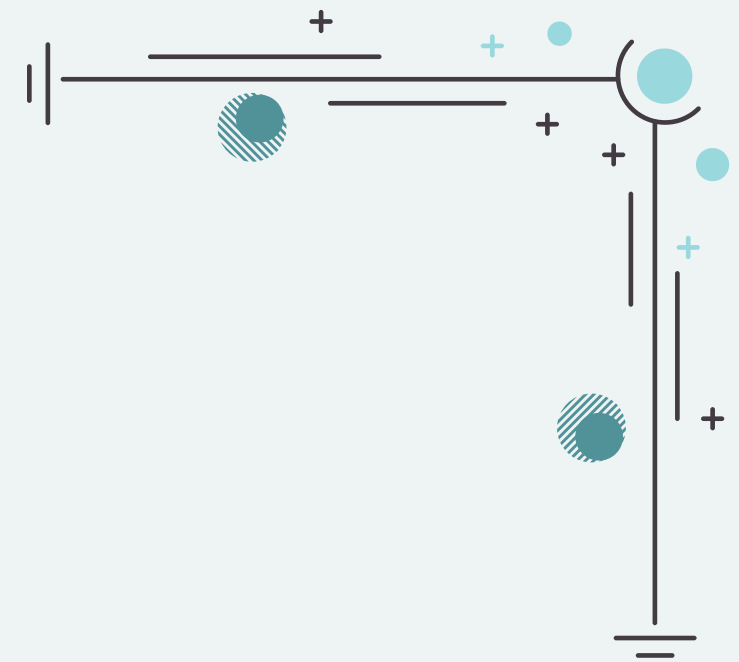


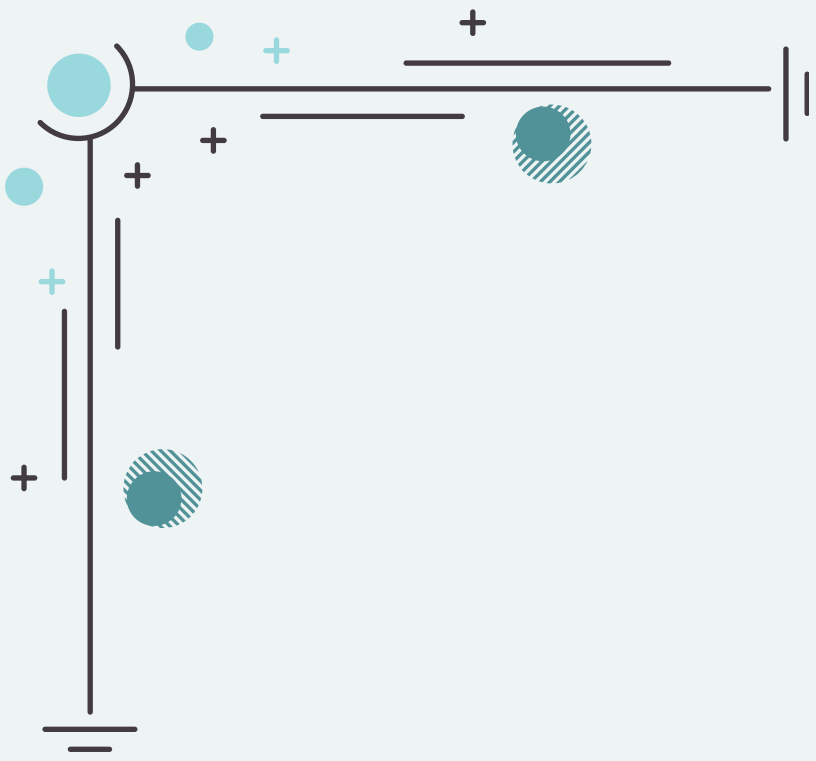
Anggota Tim

1. *Ridho Safutra*

2. *Farisa Azka*

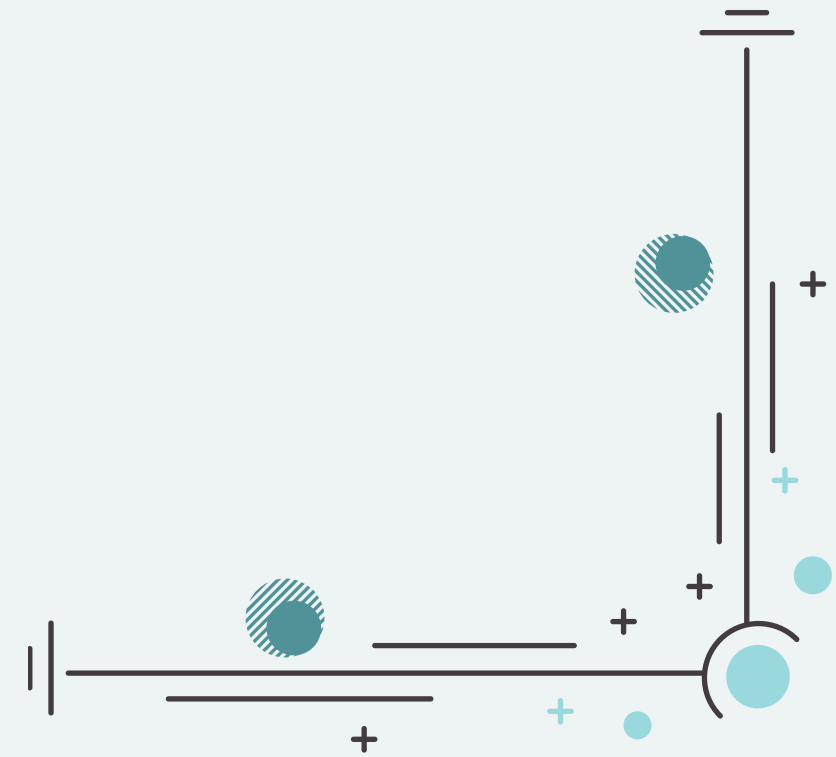
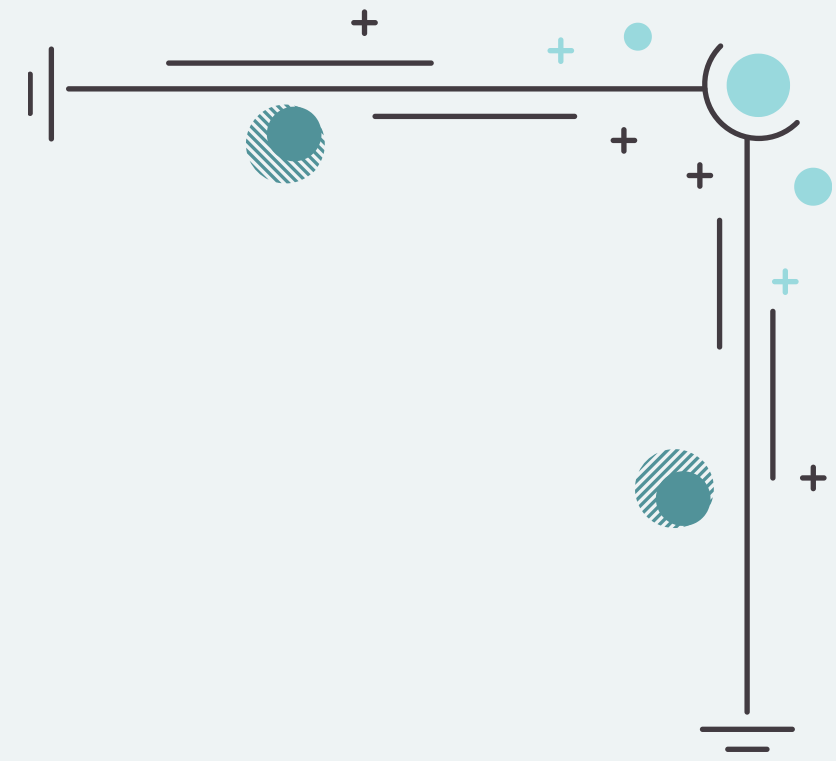
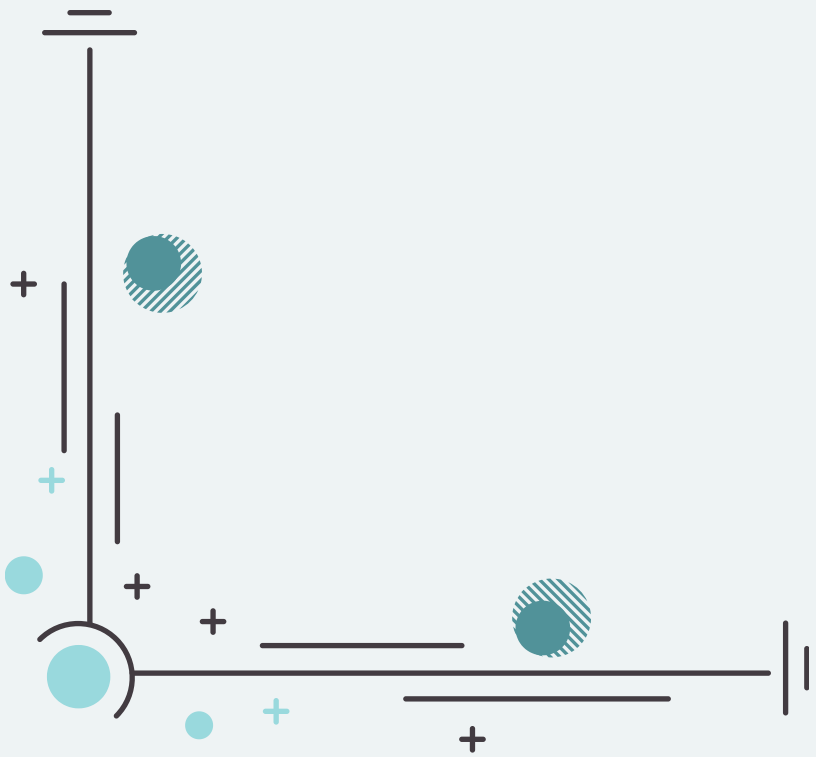
Takiya





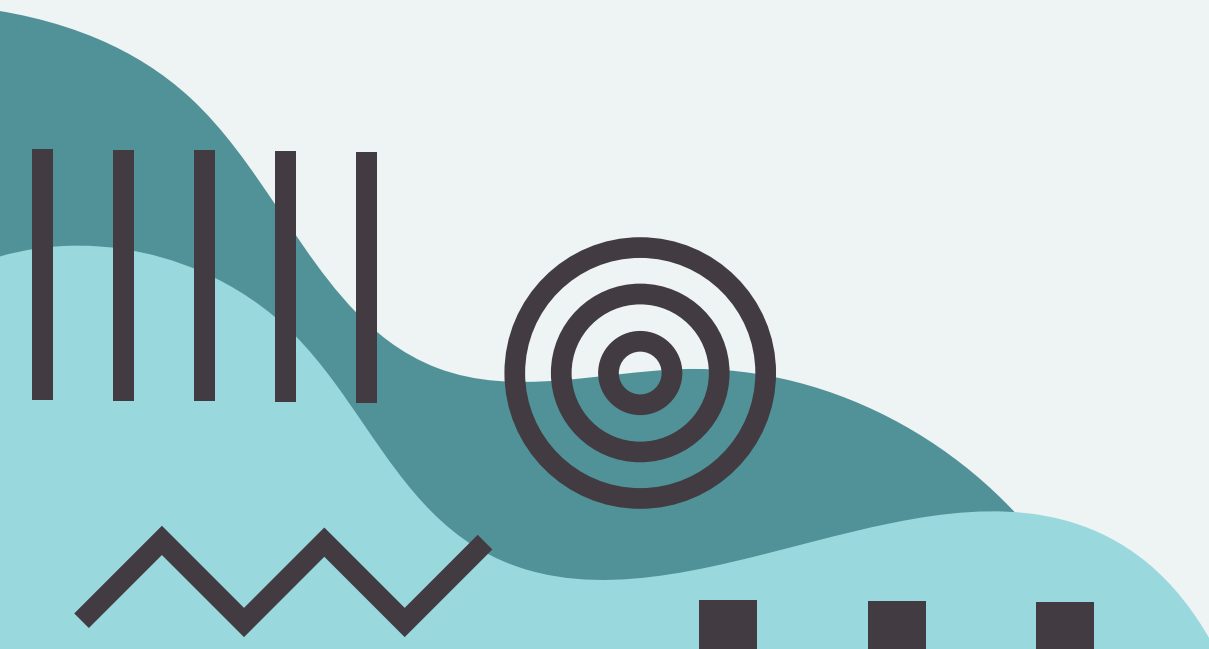
Latar Belakang

- Sistem perpustakaan manual memiliki banyak keterbatasan
- Pengelolaan data buku & peminjaman tidak efisien
- Dibutuhkan sistem backend berbasis REST API
- REST API memudahkan integrasi web & mobile



Tujuan Project

- Membangun REST API Perpustakaan Online
- Menggunakan Node.js & Express.js
- Mengelola data buku, user, dan peminjaman
- Menerapkan autentikasi JWT
- Integrasi API publik Open Library



Teknologi yang Digunakan

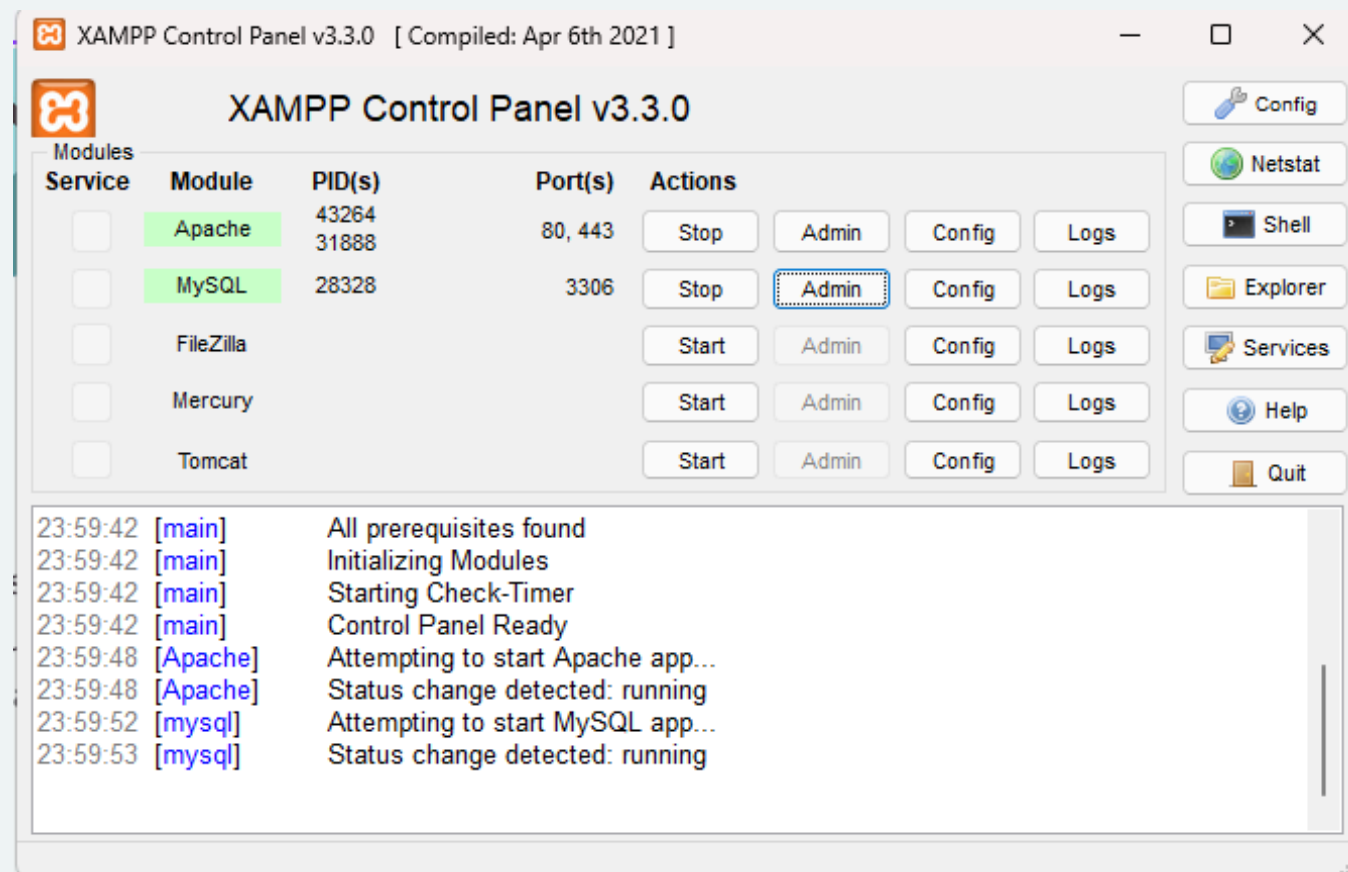
- Node.js → Backend runtime
- Express.js → Framework REST API
- MySQL → Database relasional
- JWT → Autentikasi pengguna
- Open Library API → Data buku eksternal
- Postman → Testing API

Arsitektur Sistem

- Client mengirim request HTTP
- Backend memproses request melalui REST API
- Database MySQL sebagai penyimpanan data
- Response dikirim dalam format JSON
- Autentikasi menggunakan JWT

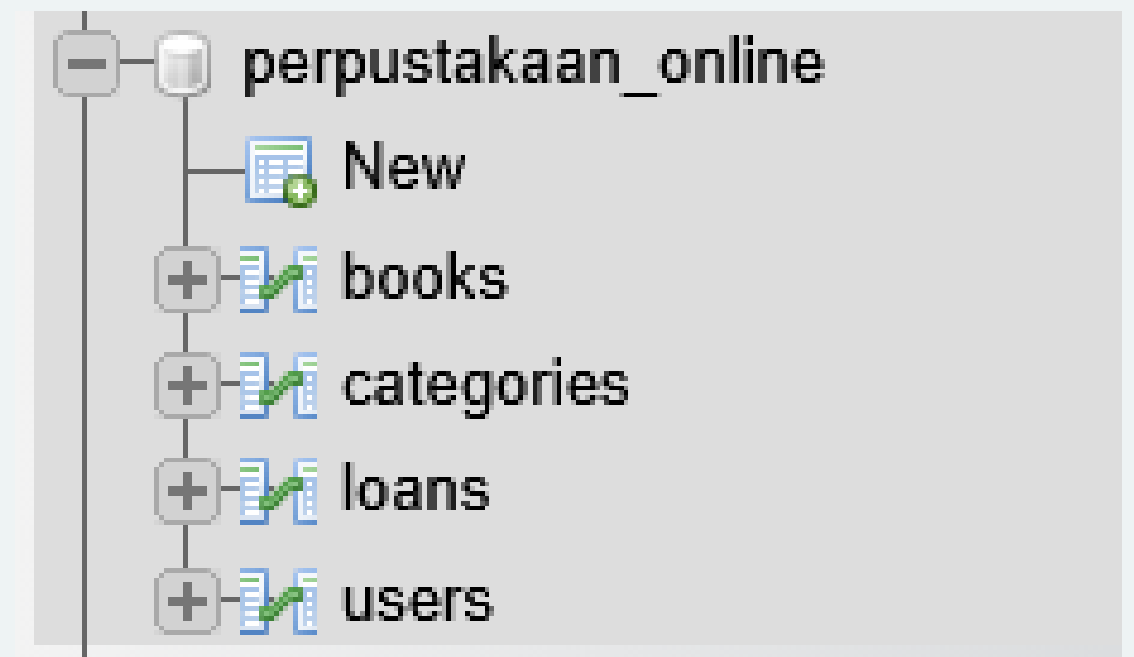
Perancangan Database

- Database menggunakan MySQL
- Menggunakan relasi antar tabel
- Menjaga konsistensi & integritas data



Tabel utama:

- users
- books
- categories
- loans

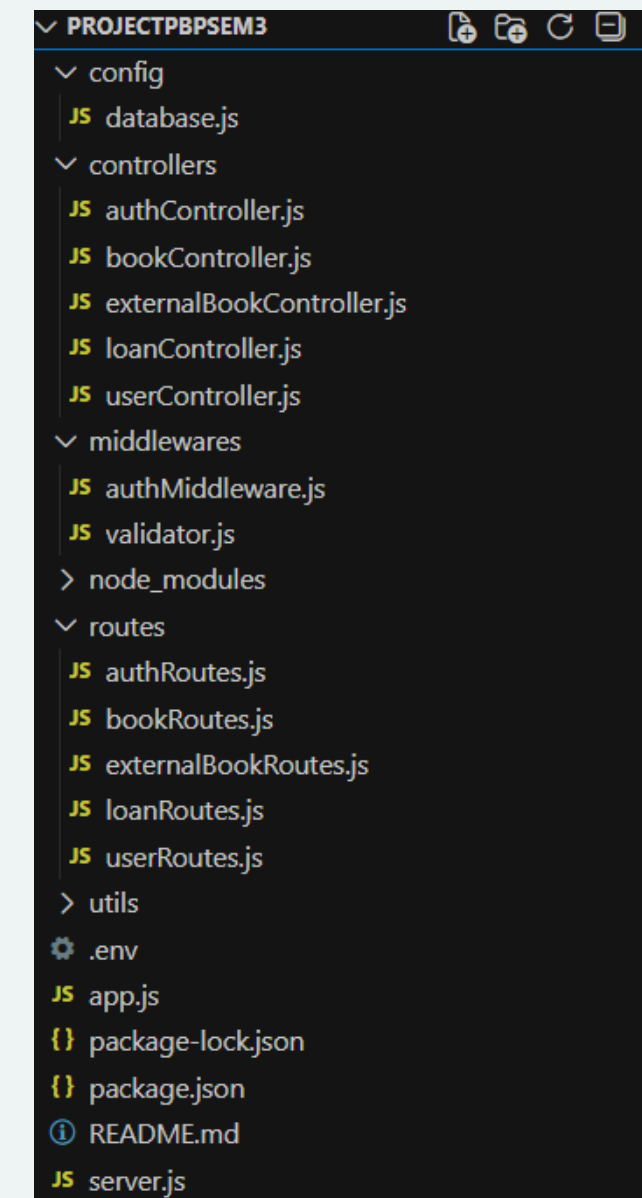


Struktur Folder Project

- Menggunakan pola MVC sederhana
- Memisahkan logic, routing, dan konfigurasi

Folder utama:

- controllers
- routes
- middlewares
- config
- app.js / server.js



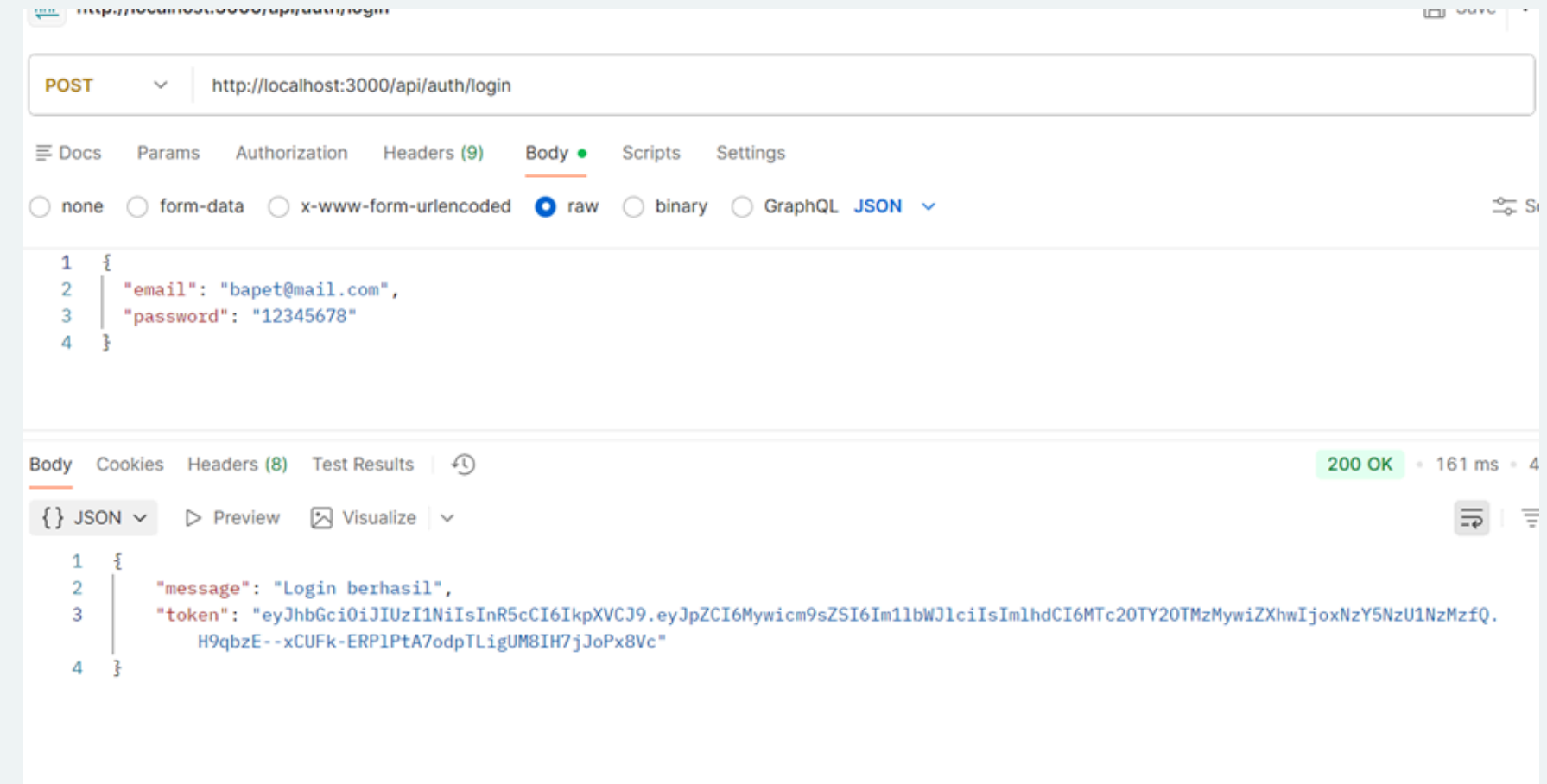
Autentikasi JWT

- User melakukan login
- Server menghasilkan JWT
- Token dikirim ke client
- Token digunakan untuk akses endpoint protected
- Middleware memvalidasi token

```

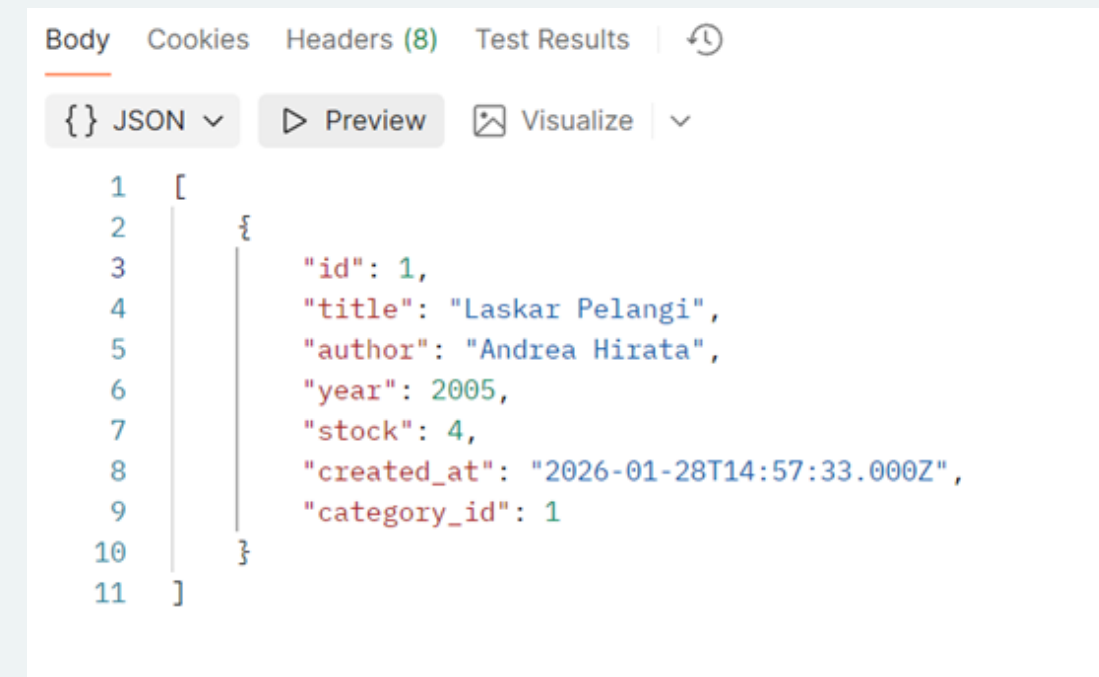
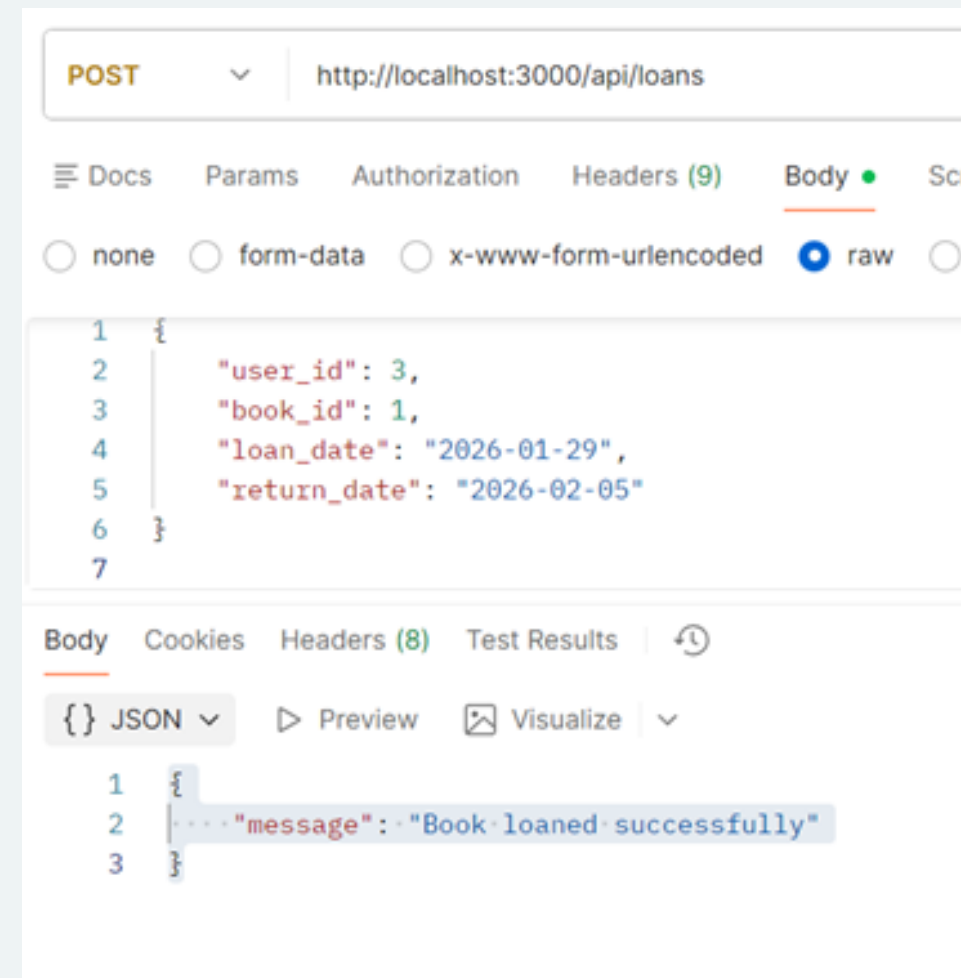
middlewares > JS authMiddleware.js > ...
1  const jwt = require("jsonwebtoken");
2
3  const authMiddleware = (req, res, next) => {
4    const authHeader = req.headers.authorization;
5
6    if (!authHeader) {
7      return res.status(401).json({
8        message: "Token tidak ditemukan",
9      });
10   }
11
12   const token = authHeader.split(" ")[1];
13
14   try {
15     const decoded = jwt.verify(token, process.env.JWT_SECRET);
16     req.user = decoded; // { id, role }
17     next();
18   } catch (error) {
19     return res.status(401).json({
20       message: "Token tidak valid",
21     });
22   }
23 };
24
25 module.exports = authMiddleware;
26

```



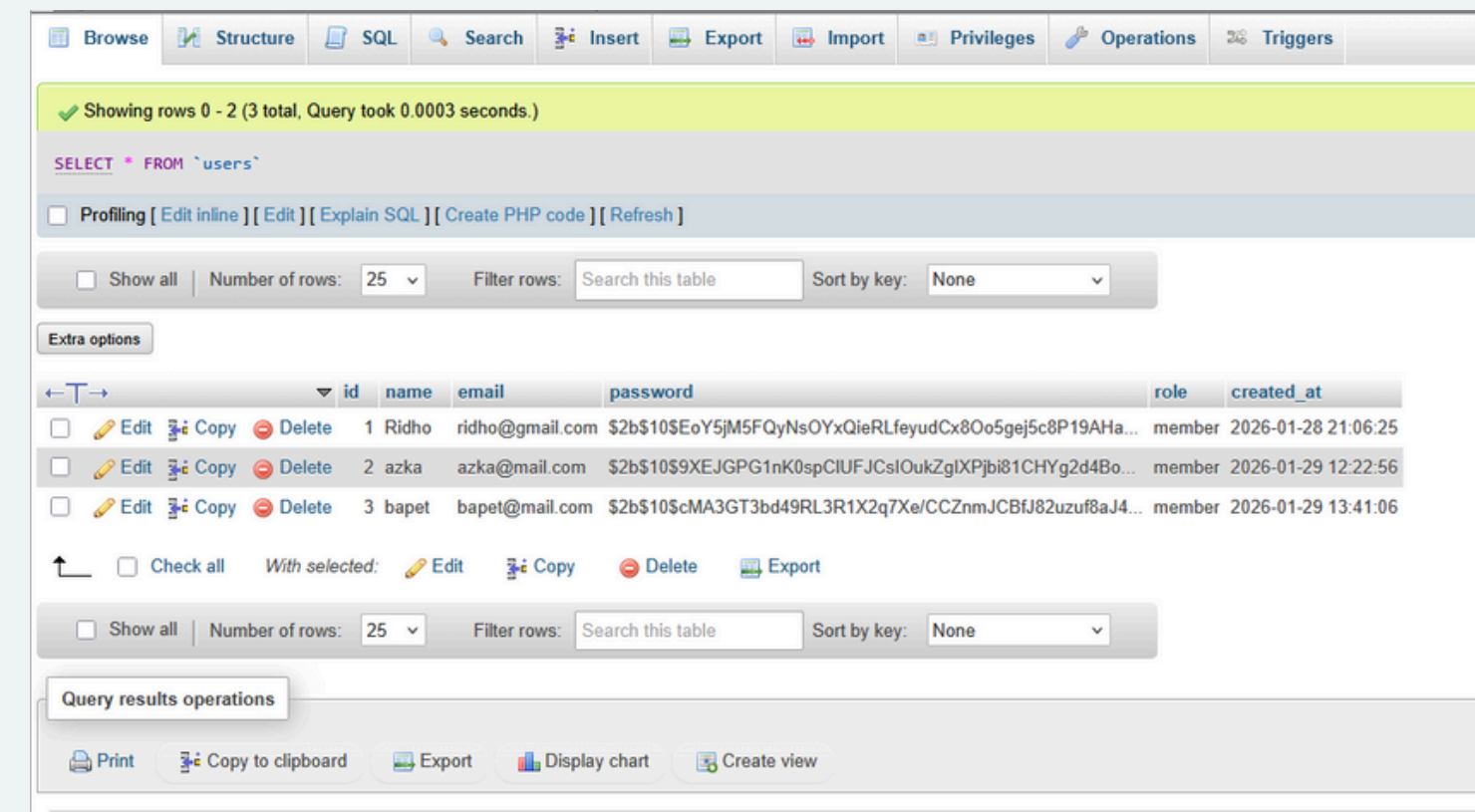
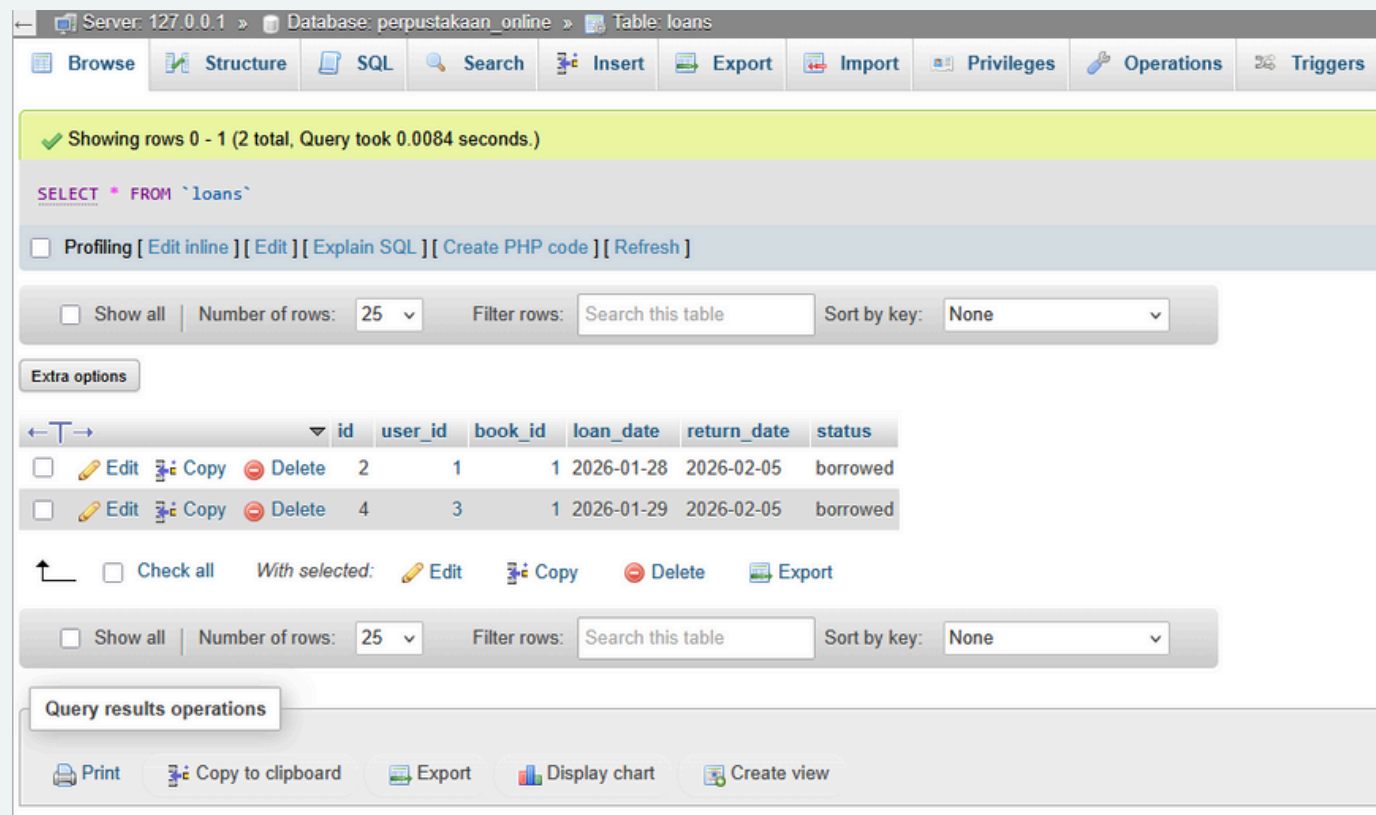
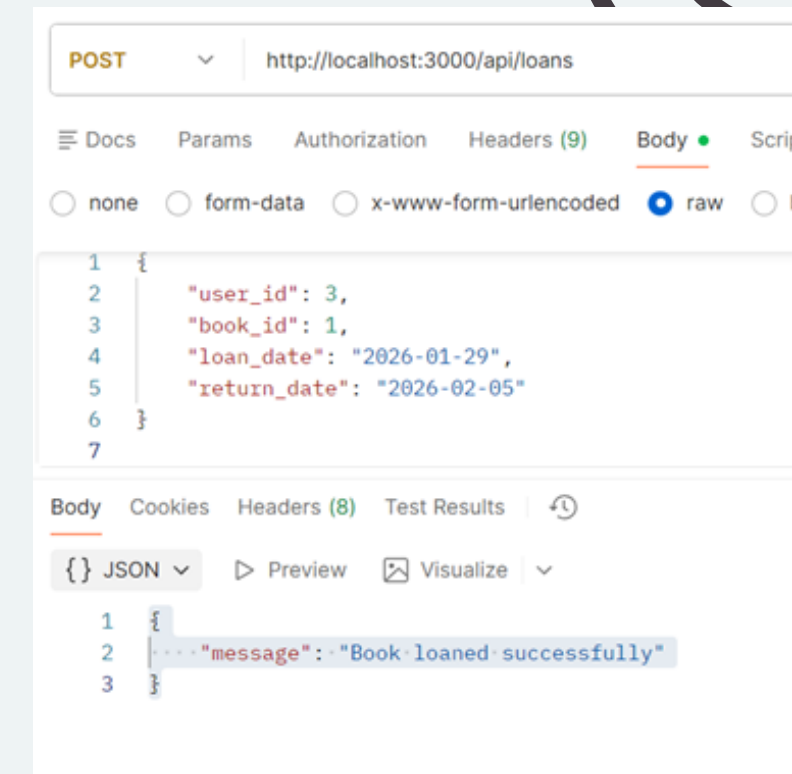
Endpoint REST API

- Auth:
- POST /register
- POST /login
- Books:
- GET /books
- POST /books
- Loans:
- POST /loans
- GET /loans



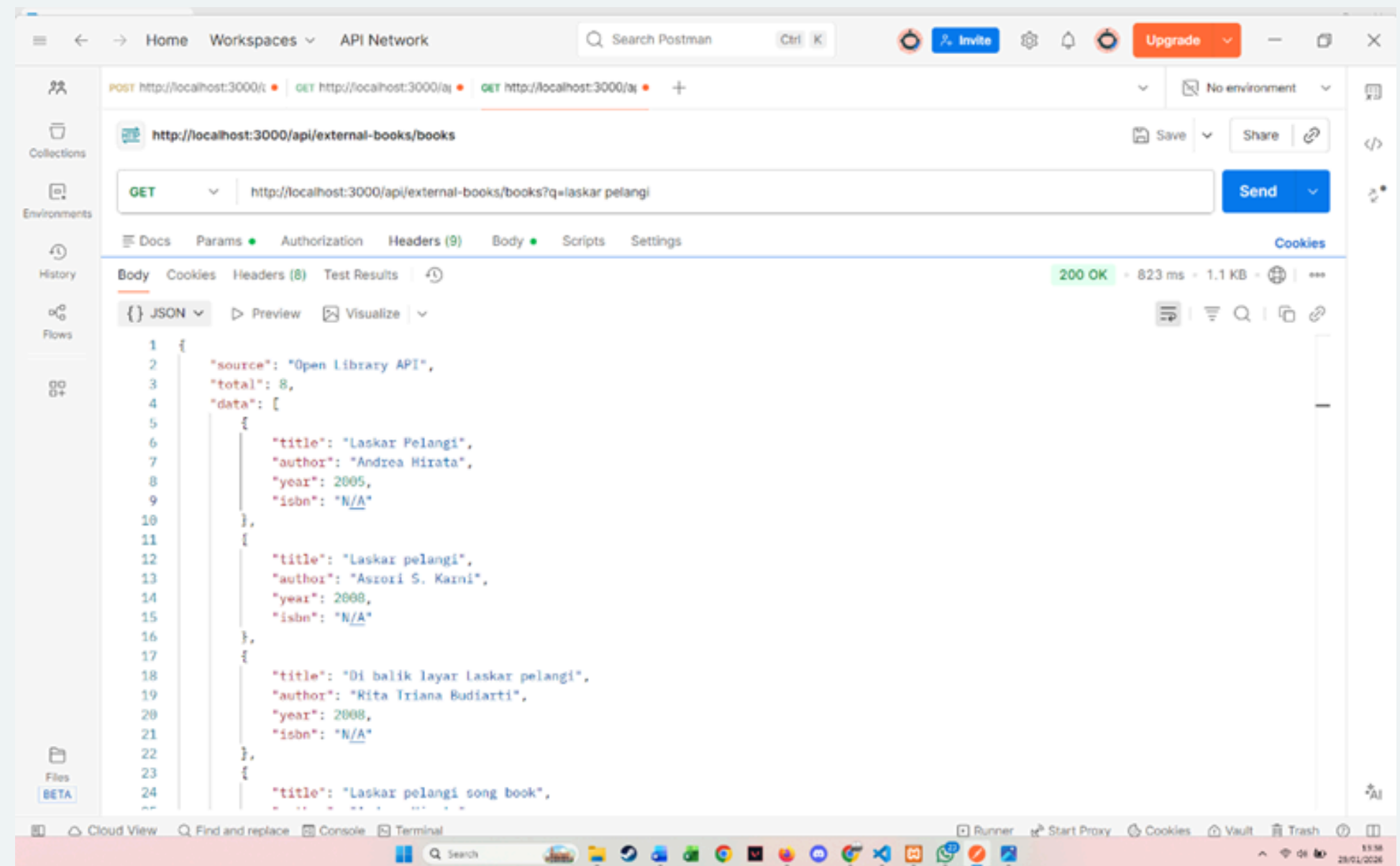
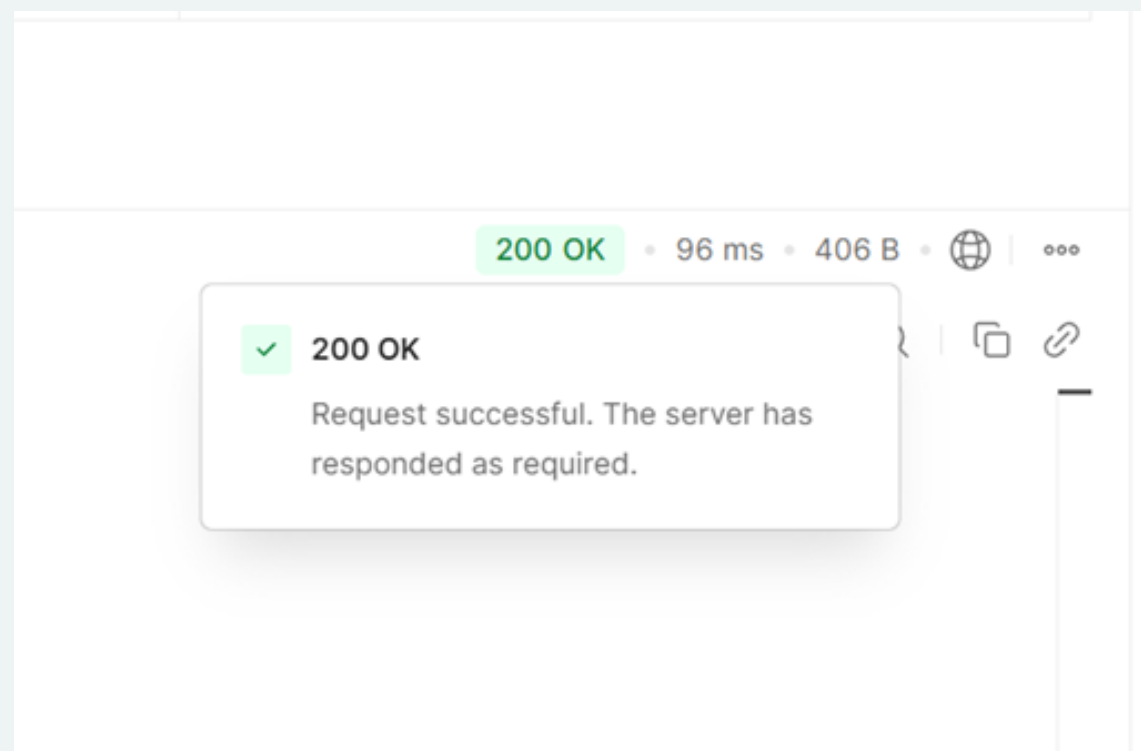
Fitur Peminjaman Buku

- User dapat meminjam buku
- Data disimpan ke tabel loans
- Terhubung dengan users & books
- Menggunakan foreign key



Pengujian Sistem

- Pengujian menggunakan Postman
- Semua endpoint berjalan normal
- Response JSON & status code sesuai

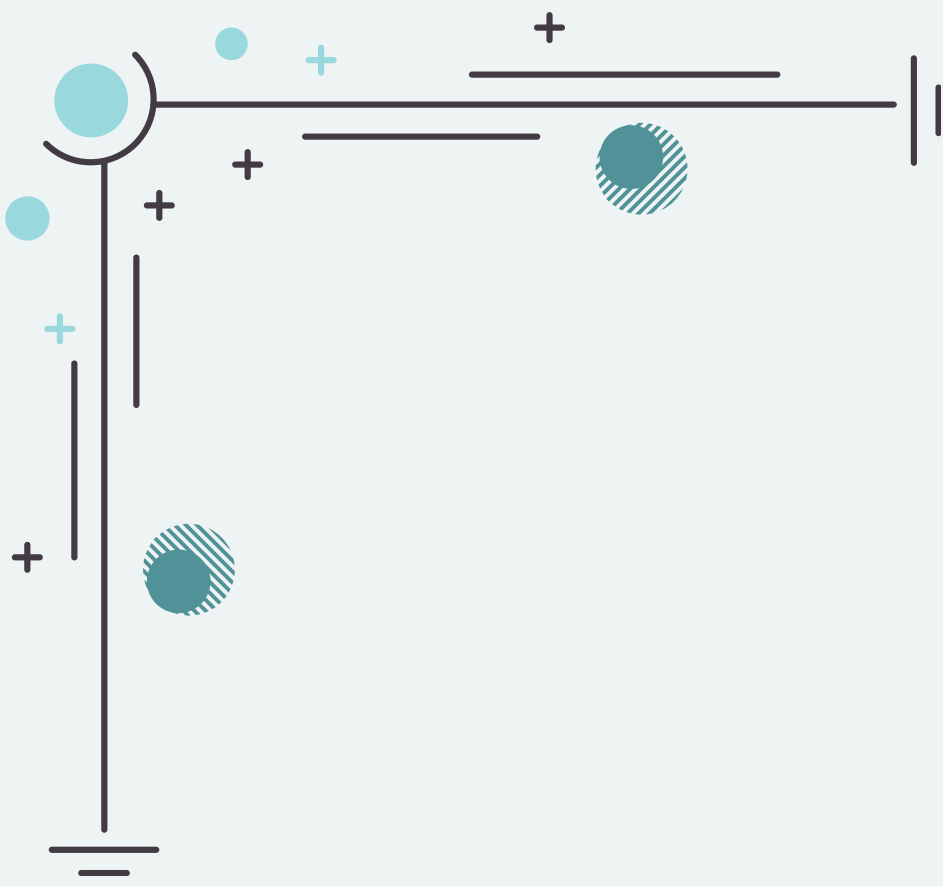


Resimpulan

- REST API berhasil dibangun
- Autentikasi JWT berjalan baik
- Database terintegrasi dengan benar
- API publik berhasil diimplementasikan
- Sistem siap dikembangkan lebih lanjut

Saran Pengembangan

- Pembuatan frontend web/mobile
- Fitur pengembalian & denda buku
- Manajemen role (admin & user)
- Dokumentasi API (Swagger)



TERIMA
KASIH

