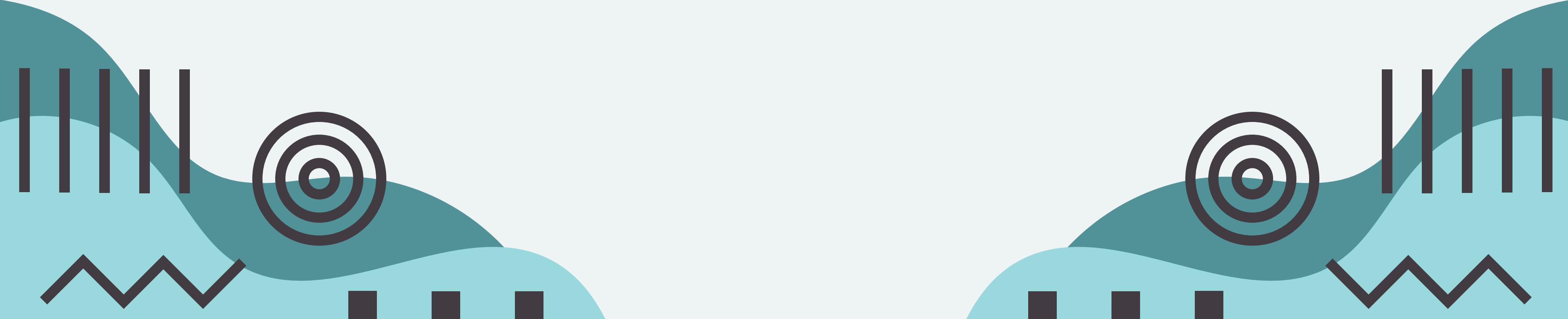


# **PERANCANGAN DAN IMPLEMENTASI REST API PERPUSTAKAAN ONLINE MENGGUNAKAN NODE.JS DAN MYSQL**

Disusun oleh :  
Ridho Safutra  
Farisa Azka Takiya



# Anggota Tim

1. *Ridho Safutra*

2. *Farisa Azka*

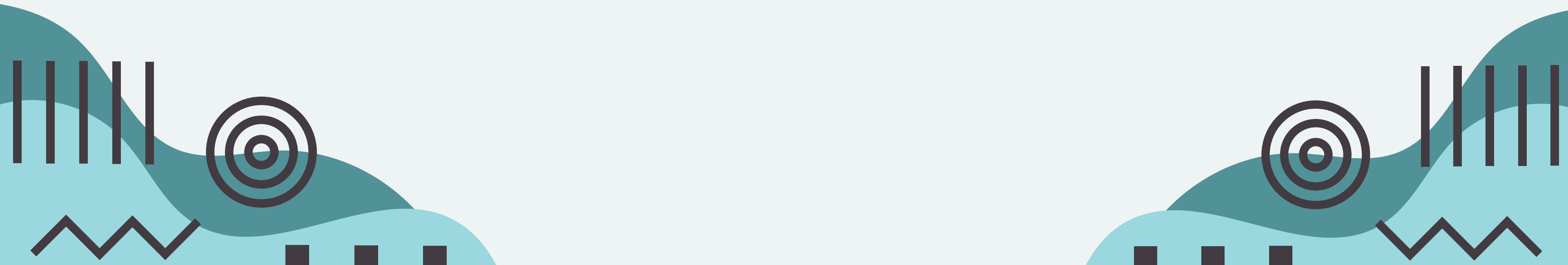
*Takiya*

# 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

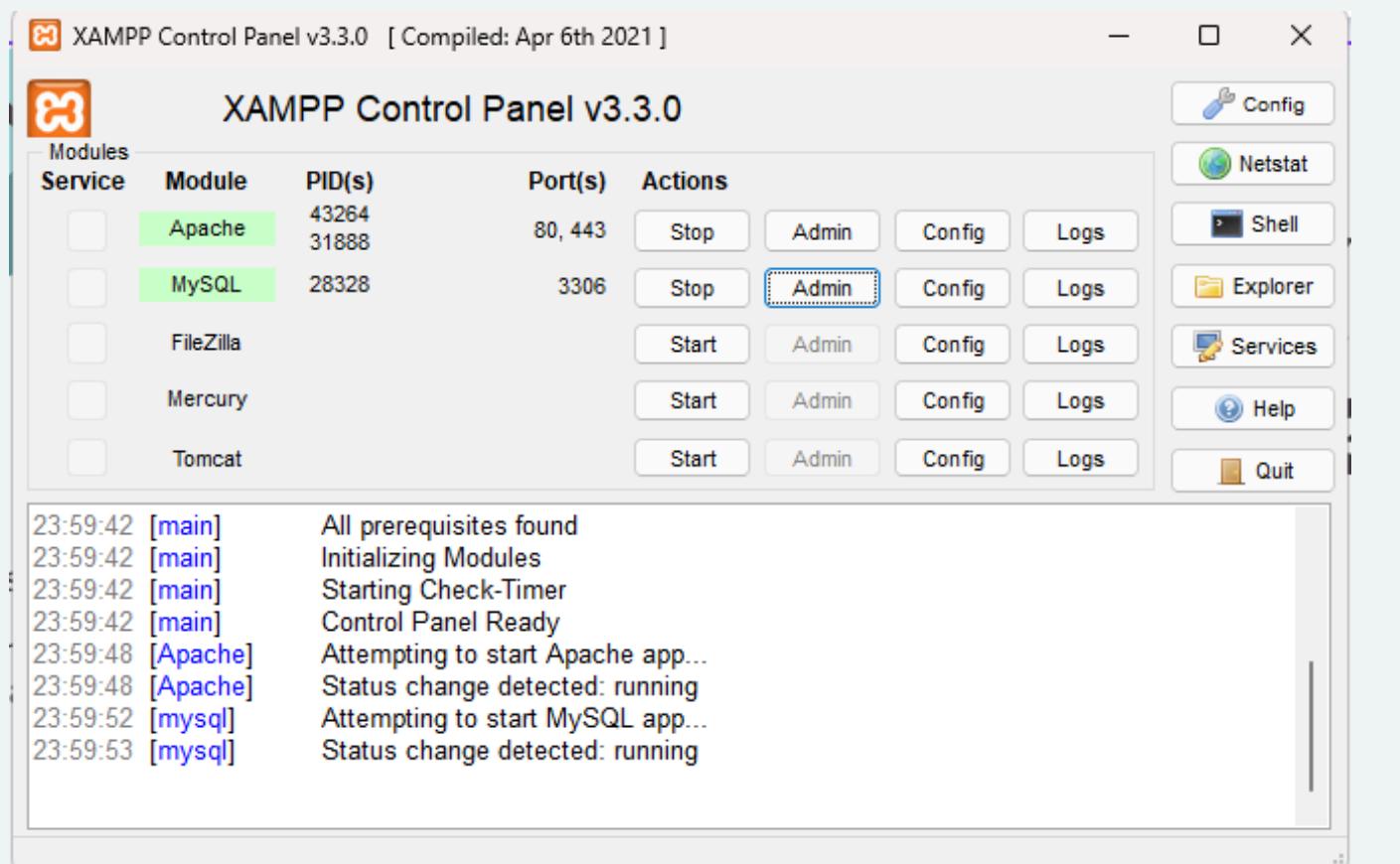
- Node.js → Backend runtime
- Express.js → Framework REST API
- MySQL → Database relasional
- JWT → Autentikasi pengguna
- Open Library API → Data buku eksternal
- Postman → Testing API

# 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

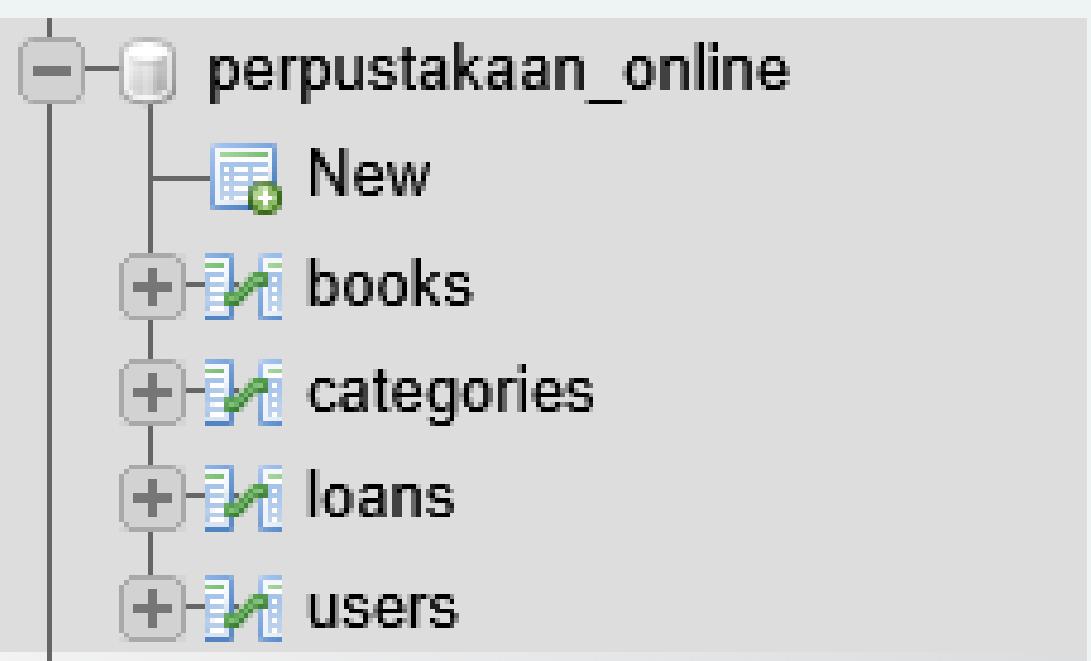
# Perancangan Database

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



Tabel utama:

- users
- books
- categories
- loans

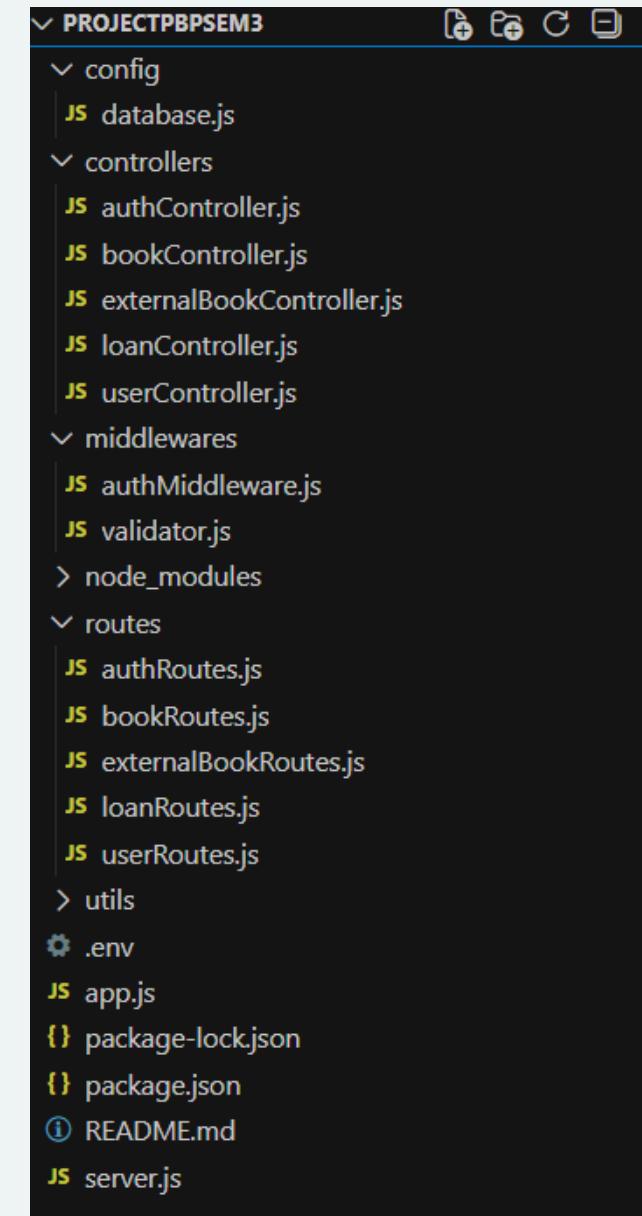


# Struktur Folder Project

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

Folder utama:

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

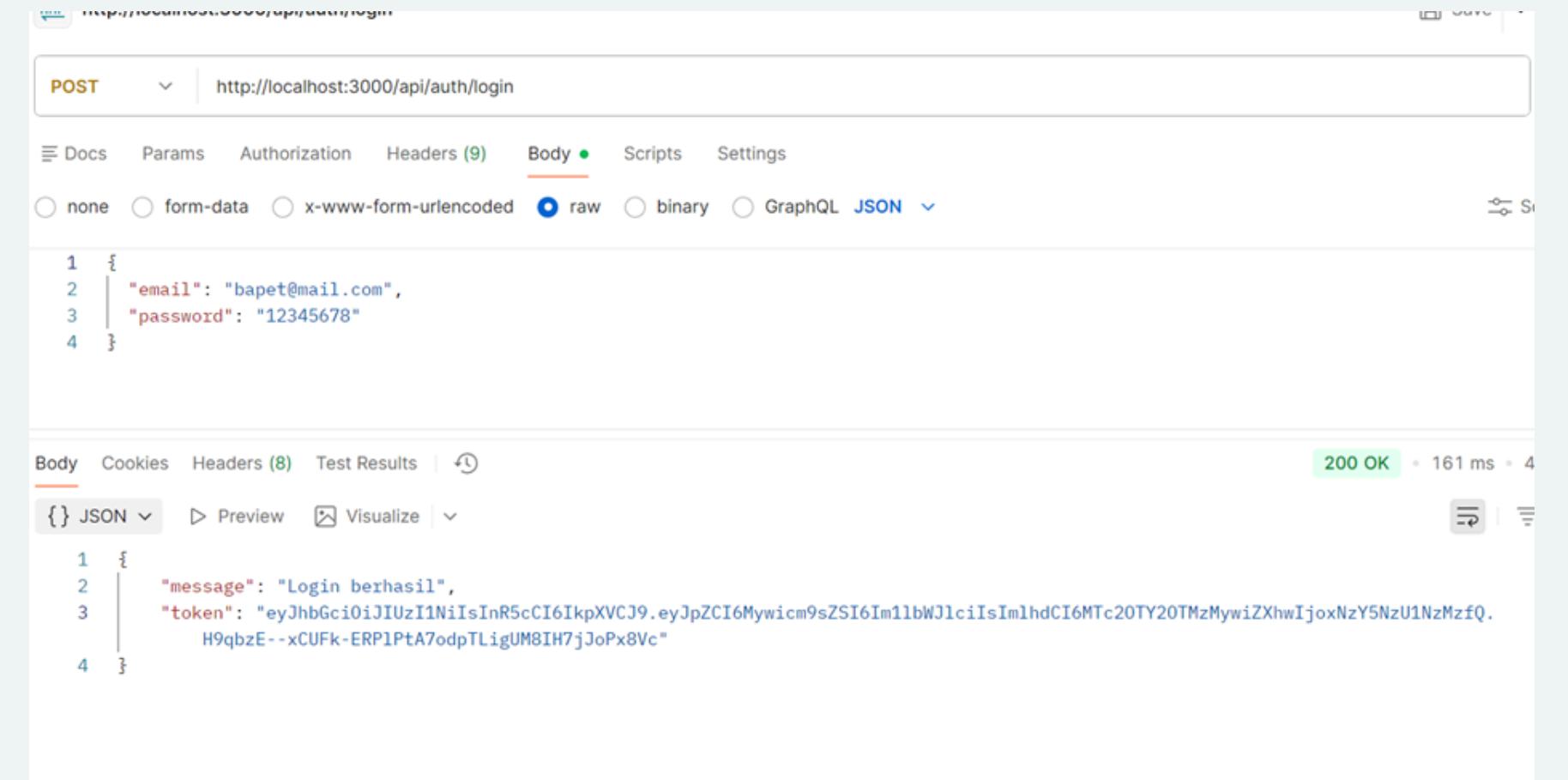


```
PROJECTPBSEM3
  config
    database.js
  controllers
    authController.js
    bookController.js
    externalBookController.js
    loanController.js
    userController.js
  middlewares
    authMiddleware.js
    validator.js
  routes
    authRoutes.js
    bookRoutes.js
    externalBookRoutes.js
    loanRoutes.js
    userRoutes.js
  node_modules
  utils
  .env
  app.js
  package-lock.json
  package.json
  README.md
  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

The screenshot shows a REST API testing interface with the following details:

- Method:** POST
- URL:** http://localhost:3000/api/loans
- Headers:** Authorization, Headers (9), Body (selected), Script
- Body:** Raw JSON (selected) containing the following data:

```
1 {  
2   "user_id": 3,  
3   "book_id": 1,  
4   "loan_date": "2026-01-29",  
5   "return_date": "2026-02-05"  
6 }  
7
```

- Response Headers:** Headers (8)
- Response Body:** JSON (selected) containing the message "Book loaned successfully".
- Response Preview:** A JSON object with the key "message" and the value "Book loaned successfully".
- Response Visualize:** A JSON object with the key "message" and the value "Book loaned successfully".

# Fitur Peminjaman Buku

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

The screenshot shows the MySQL Workbench interface for the 'perpustakaan\_online' database. The 'loans' table is selected. The table has columns: id, user\_id, book\_id, loan\_date, return\_date, and status. There are two rows of data: one where the loan date is 2026-01-28 and the return date is 2026-02-05, and another where the loan date is 2026-01-29 and the return date is 2026-02-05, both marked as 'borrowed'.

The screenshot shows the MySQL Workbench interface for the 'perpustakaan\_online' database. The 'users' table is selected. The table has columns: id, name, email, password, role, and created\_at. There are three rows of data: Ridho (ridho@gmail.com), azka (azka@mail.com), and bapet (bapet@mail.com). All users are listed as 'member' and were created on different dates in January 2026.

The screenshot shows a Postman request to the 'http://localhost:3000/api/loans' endpoint using the 'POST' method. The 'Body' tab is selected and contains the following JSON payload:

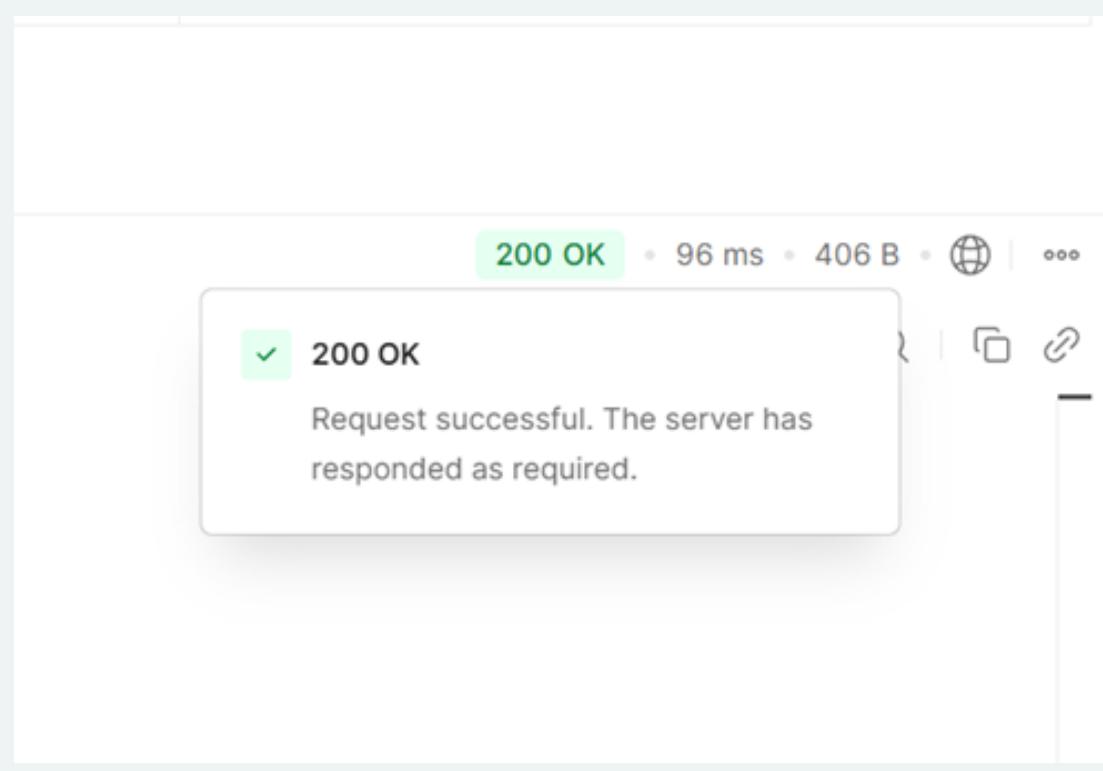
```
1 {  
2   "user_id": 3,  
3   "book_id": 1,  
4   "loan_date": "2026-01-29",  
5   "return_date": "2026-02-05"  
6 }  
7
```

The response body shows a success message:

```
1 {  
2   ... "message": "Book loaned successfully"  
3 }
```

# Pengujian Sistem

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

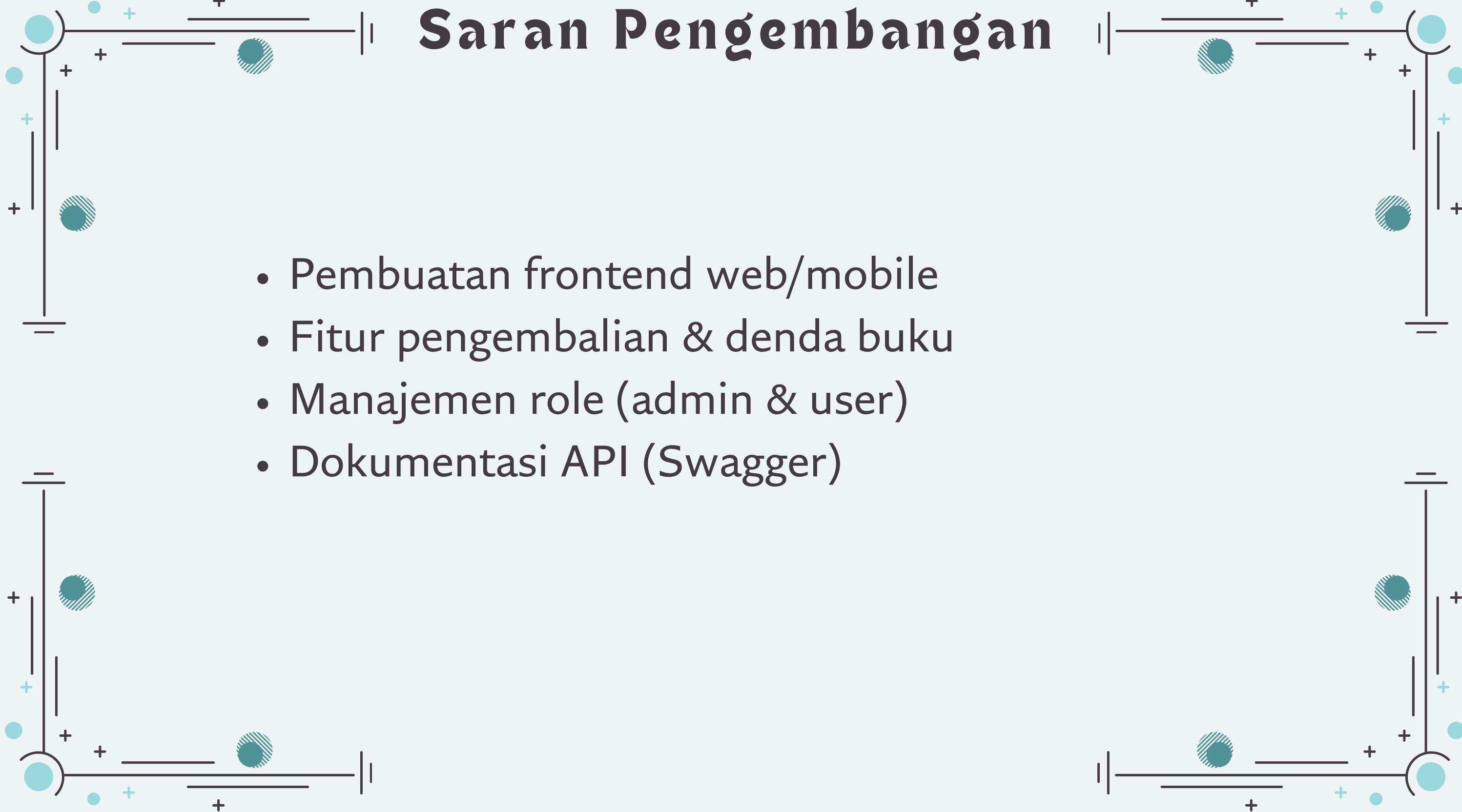


A screenshot of the Postman application interface. The top bar shows "POST http://localhost:3000/t" and "GET http://localhost:3000/a". The main area shows a GET request to "http://localhost:3000/api/external-books/books?q=laskar pelangi". The response status is "200 OK" with a response time of "823 ms" and a size of "1.1 KB". The response body is a JSON object:

```
1 {  
2   "source": "Open Library API",  
3   "total": 8,  
4   "data": [  
5     {  
6       "title": "Laskar Pelangi",  
7       "author": "Andrea Hirata",  
8       "year": 2005,  
9       "isbn": "N/A"  
10    },  
11    {  
12      "title": "Laskar pelangi",  
13      "author": "Aszori S. Kaini",  
14      "year": 2008,  
15      "isbn": "N/A"  
16    },  
17    {  
18      "title": "Di balik layar laskar pelangi",  
19      "author": "Rita Triana Budiarti",  
20      "year": 2008,  
21      "isbn": "N/A"  
22    },  
23    {  
24      "title": "Laskar pelangi song book",  
25      "author": "",  
26      "year": 2008,  
27      "isbn": "N/A"  
28    }  
29  ]  
30}  
31
```

# 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

