

**LAPORAN IMPLEMENTASI AKHIR PROYEK  
BASIS DATA**

**”LIBRARY BORROWING  
TRACKER SYSTEM”**

**Disusun oleh Kelompok 12:**

- |                          |                      |
|--------------------------|----------------------|
| 1. Mahardika Ramadhana   | (24/538247/PA/22831) |
| 2. Hammam Muhammad Yazid | (24/534894/PA/22687) |
| 3. Daffa M. Siddiq       | (24/533358/PA/22569) |

**DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS GADJAH MADA  
YOGYAKARTA**

**2025**

# Contents

<b>1</b>	<b>PENDAHULUAN</b>	<b>2</b>
1.1	Latar Belakang Masalah . . . . .	2
1.2	Tujuan Pengembangan Sistem . . . . .	2
1.3	Ruang Lingkup Implementasi . . . . .	3
<b>2</b>	<b>PERANCANGAN DAN IMPLEMENTASI DATABASE</b>	<b>4</b>
2.1	Visualisasi Entity Relationship Diagram (ERD) . . . . .	4
2.2	Analisis Struktur Tabel dan Normalisasi . . . . .	4
2.2.1	Tabel Master: Books dan Members . . . . .	4
2.2.2	Tabel Transaksional: Loans (Peminjaman) . . . . .	5
<b>3</b>	<b>IMPLEMENTASI SISTEM TERINTEGRASI</b>	<b>6</b>
3.1	Arsitektur Aplikasi (Tech Stack) . . . . .	6
3.1.1	Implementasi Antarmuka Pengguna (Frontend) . . . . .	6
3.2	Manajemen Sirkulasi (Peminjaman) . . . . .	8
3.3	Manajemen Anggota . . . . .	8
<b>4</b>	<b>PENGUJIAN SISTEM DAN HASIL</b>	<b>9</b>
4.1	Dokumentasi API (Swagger Integration) . . . . .	9
4.2	Pengujian Fungsional (Black Box Testing) . . . . .	10
<b>5</b>	<b>KESIMPULAN DAN REFLEKSI</b>	<b>12</b>
5.1	Kesimpulan Proyek . . . . .	12
5.2	Refleksi dan Pembelajaran . . . . .	12

5.3	Rencana Pengembangan Selanjutnya . . . . .	13
-----	--	----

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang Masalah

Dalam ekosistem pendidikan modern, perpustakaan memegang peranan vital sebagai pusat sumber daya informasi. Namun, observasi lapangan menunjukkan bahwa banyak perpustakaan skala kecil hingga menengah masih mengandalkan metode pengelolaan sirkulasi buku yang konvensional. Proses pencatatan peminjaman yang dilakukan secara manual menggunakan buku besar atau spreadsheet sederhana seringkali menimbulkan fragmentasi data. Informasi mengenai ketersediaan buku, status peminjaman, dan riwayat anggota tersebar di berbagai dokumen fisik yang tidak saling terintegrasi, menyebabkan inefisiensi operasional yang signifikan.

Masalah paling krusial yang diidentifikasi adalah ketidakkonsistenan dalam penerapan aturan denda. Tanpa sistem yang terotomatisasi, pustakawan harus menghitung jumlah hari keterlambatan dan nominal denda secara manual untuk setiap transaksi. Hal ini sangat rentan terhadap kesalahan manusia (*human error*), yang pada akhirnya dapat merugikan institusi maupun menurunkan kepercayaan anggota perpustakaan. Selain itu, ketiadaan validasi sistem yang ketat memungkinkan terjadinya anomali data, seperti satu buku dipinjam oleh dua orang berbeda pada waktu yang sama, atau anggota yang memiliki tanggungan denda masih diperbolehkan meminjam buku baru.

## 1.2 Tujuan Pengembangan Sistem

Proyek "Library Borrowing Tracker System" dikembangkan sebagai solusi teknologi terintegrasi untuk menjawab permasalahan tersebut. Tujuan utama dari pengembangan sistem ini adalah membangun sebuah platform berbasis web yang mampu:

1. **Otomasi Sirkulasi:** Menggantikan pencatatan manual dengan sistem database relasional yang mampu memproses peminjaman dan pengembalian secara *real-time*.
2. **Penegakan Aturan Bisnis:** Menerapkan logika bisnis secara ketat, yaitu batas waktu peminjaman 14 hari dan denda otomatis Rp1.000 per hari, yang dieksekusi langsung oleh sistem tanpa intervensi manual.
3. **Integritas Data:** Menjamin keakuratan data inventaris melalui penerapan *Foreign Key Constraints* dan normalisasi basis data hingga tingkat ketiga (3NF).

## 1.3 Ruang Lingkup Implementasi

Laporan ini merangkum hasil implementasi tahap akhir (*Final Integration*) yang mencakup pengembangan penuh dari sisi Backend, Frontend, hingga Database. Teknologi yang digunakan meliputi:

- **Database Tier:** PostgreSQL (di-hosting pada Neon) untuk penyimpanan data relasional yang persisten.
- **Logic Tier (Backend):** NestJS framework untuk menangani API, validasi DTO, dan logika bisnis denda.
- **Presentation Tier (Frontend):** React dengan Vite dan Tailwind CSS untuk antarmuka pengguna yang responsif.

## BAB 2 PERANCANGAN DAN IMPLEMENTASI DATABASE

### 2.1 Visualisasi Entity Relationship Diagram (ERD)

Gambar di bawah ini adalah *Entity Relationship Diagram* (ERD) yang menjadi jembatan visual antara desain konseptual dan implementasi fisik database. ERD ini dirancang untuk memetakan hubungan antara entitas Buku, Anggota, Peminjaman, dan Denda.

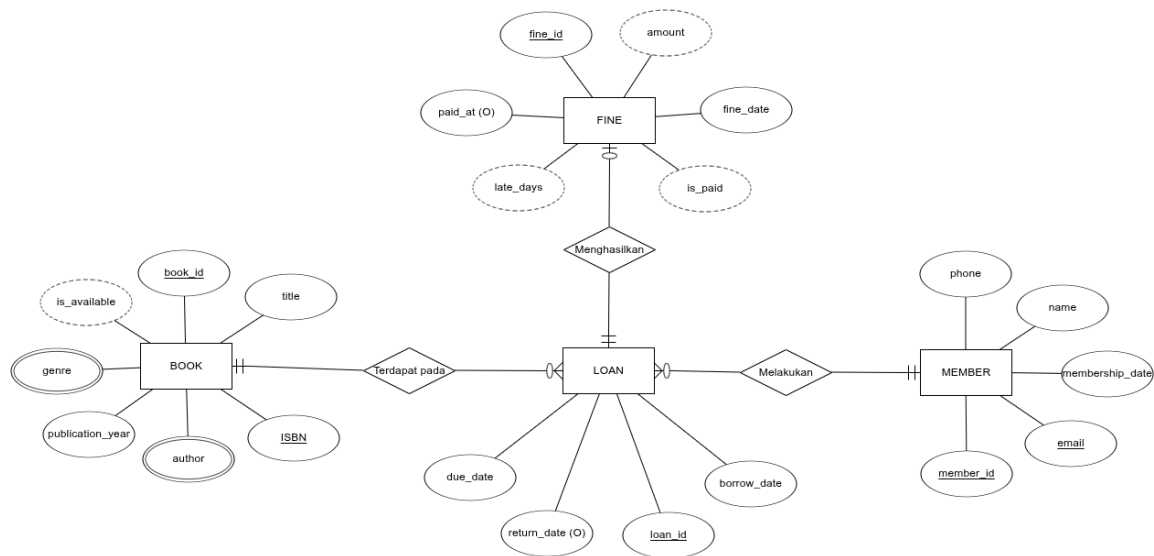


Figure 1: Entity Relationship Diagram (ERD) Final - 3NF

### 2.2 Analisis Struktur Tabel dan Normalisasi

Struktur database ini telah melalui proses normalisasi yang ketat untuk menghilangkan redundansi dan menjaga konsistensi. Berikut adalah analisis mendalam untuk setiap entitas utama:

#### 2.2.1 Tabel Master: Books dan Members

Tabel ini menyimpan data referensi utama yang jarang berubah namun sering diakses (Read-Heavy).

- **Books (Buku):** Tabel ini dirancang untuk menyimpan metadata inventaris.

Kolom `is_available` (Boolean) bertindak sebagai "bendera" status yang krusial; jika bernilai *false*, maka buku tersebut tidak akan muncul dalam daftar pencarian peminjaman. Kami memisahkan atribut penulis dan genre untuk memenuhi standar 1NF.

- **Members (Anggota):** Tabel ini menyimpan identitas peminjam. Penggunaan `email` yang bersifat *UNIQUE* mencegah pendaftaran ganda oleh satu individu.

### 2.2.2 Tabel Transaksional: Loans (Peminjaman)

Tabel Loan adalah jantung dari sistem ini yang menghubungkan Anggota dan Buku.

- **Relasi:** Tabel ini memiliki dua *Foreign Key*, yaitu `member_id` dan `book_id`. Relasi ini menerapkan kaidah *Referential Integrity*; sistem akan menolak transaksi jika ID buku atau ID anggota yang dimasukkan tidak valid.
- **Logika Tanggal:** Kolom `borrow_date` dan `due_date` adalah kunci dari fitur otomasi. Saat record baru dibuat, `due_date` secara otomatis diisi dengan nilai (`borrow_date` + 14 hari) melalui logika aplikasi, menghilangkan kebutuhan pustakawan untuk menghitung tanggal kembali secara manual.

# BAB 3 IMPLEMENTASI SISTEM TERINTEGRASI

## 3.1 Arsitektur Aplikasi (Tech Stack)

Sistem ini dibangun di atas arsitektur modern yang memisahkan tanggung jawab antara antarmuka dan logika data, memastikan skalabilitas dan kemudahan pemeliharaan.

### 3.1.1 Implementasi Antarmuka Pengguna (Frontend)

Sisi klien (Frontend) bertugas menerjemahkan data JSON dari Backend menjadi visual yang interaktif. Berikut adalah dokumentasi implementasi UI:

#### 1. Dashboard Statistik (Overview)

Halaman Dashboard berfungsi sebagai pusat kendali bagi administrator. Kartu statistik di bagian atas memberikan wawasan instan mengenai total buku, anggota aktif, dan jumlah peminjaman yang sedang berjalan. Grafik visual memudahkan pengambilan keputusan manajerial.

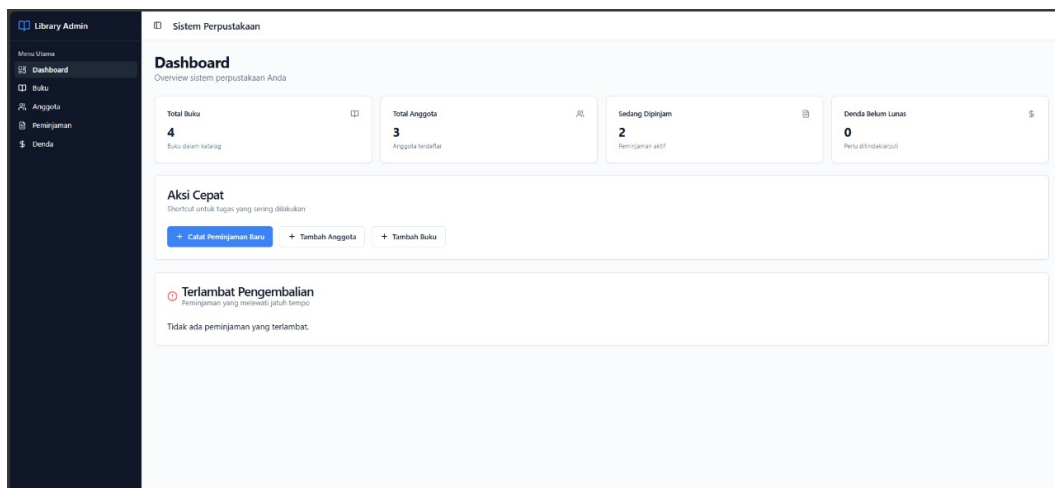


Figure 2: Dashboard Admin dengan Ringkasan Statistik Real-time

#### 2. Manajemen Katalog Buku

Halaman ini memungkinkan admin untuk melihat seluruh koleksi perpustakaan. Fitur pencarian dan filter status ketersediaan diimplementasikan untuk mempercepat proses temu kembali informasi.



Judul Buku	Penulis	Tahun Terbit	Genre	Status	Aksi
Database	David	2022	Computer Science	Tersedia	[Edit] [Hapus]
Laut Bercerita	Lella S. Chudori	2020	History	Dipinjam	[Edit] [Hapus]
OS	Mahardika	2009	Horor	Dipinjam	[Edit] [Hapus]
TPSC	Tufel Bark Buhir	2024	Fiction	Tersedia	[Edit] [Hapus]

Figure 3: Daftar Katalog Buku dengan Status Ketersediaan

### 3. Modul Input Data (Forms)

Formulir input dirancang dengan validasi sisi klien. Sistem tidak akan mengizinkan pengiriman data jika kolom wajib (seperti Judul atau Nama Anggota) masih kosong.

Figure 4: Modal Input Data Buku Baru

Figure 5: Modal Registrasi Anggota Baru

### 3.2 Manajemen Sirkulasi (Peminjaman)

Fitur paling kompleks dalam sistem ini adalah pencatatan peminjaman. Halaman ini menggabungkan data dari tabel Buku dan Anggota. Admin cukup memilih ID Anggota dan ID Buku, dan sistem akan secara otomatis menetapkan status buku menjadi "Dipinjam" (Unavailable).

**Manajemen Peminjaman**  
Catat transaksi peminjaman dan pengembalian buku

**+ Catat Peminjaman Baru**  
Masukkan informasi peminjaman buku

ID Anggota:

ID Buku:

Tanggal Pinjam:

Tanggal Jatuh Tempo:

[+ Simpan Transaksi](#)

**Daftar Peminjaman**  
Kelola peminjaman aktif dan riwayat

[Peminjaman Aktif](#) [Riwayat](#)

Judul Buku	Peminjam	Tgl Pinjam	Tgl Jatuh Tempo	Tgl Kembali	Status
Database	dafa maul	25/11/2025	9/12/2025	25/11/2025	<a href="#">Returned On Time</a>
OS	marco ramadhana	25/11/2025	9/12/2025	1/12/2025	<a href="#">Returned On Time</a>
Laut Berserita	dika ramadhana	24/11/2025	8/12/2025	24/11/2025	<a href="#">Returned On Time</a>

Figure 6: Antarmuka Transaksi Peminjaman dan Riwayat Pengembalian

### 3.3 Manajemen Anggota

Halaman ini menampilkan basis data anggota perpustakaan yang terdaftar. Fitur edit dan hapus disediakan untuk mengelola siklus hidup keanggotaan.

**Manajemen Anggota**  
Kelola data anggota perpustakaan

[+ Tambah Anggota](#)

**Daftar Anggota**  
Cari anggota berdasarkan nama atau email

Nama	Email	No. Telepon	Tanggal Bergabung	Aksi
dafa maul	dafa@gmail.com	085236067545	25/11/2025	<a href="#">Edit</a> <a href="#">Hapus</a>
dika ramadhana	dika@gmail.com	085321456745	24/11/2025	<a href="#">Edit</a> <a href="#">Hapus</a>
marco ramadhana	marco@gmail.com	085478946543	25/11/2025	<a href="#">Edit</a> <a href="#">Hapus</a>

Figure 7: Daftar Anggota Aktif

# BAB 4 PENGUJIAN SISTEM DAN HASIL

## 4.1 Dokumentasi API (Swagger Integration)

Dokumentasi API menjadi krusial dalam memastikan komunikasi yang lancar antara Frontend dan Backend. Kami menggunakan Swagger (OpenAPI) untuk mendefinisikan kontrak data secara transparan.

### 1. Endpoint Resources

Gambar di bawah menampilkan daftar endpoint yang tersedia untuk entitas Books dan Members. Setiap endpoint dilengkapi dengan metode HTTP yang sesuai (GET, POST, PATCH, DELETE).



Figure 8: Dokumentasi Endpoint untuk Modul Buku dan Anggota

### 2. Struktur Data (DTO)

Data Transfer Object (DTO) digunakan untuk memvalidasi data yang masuk ke server. Skema ini memastikan bahwa data yang dikirim oleh client memiliki tipe dan format yang benar sebelum diproses oleh database.

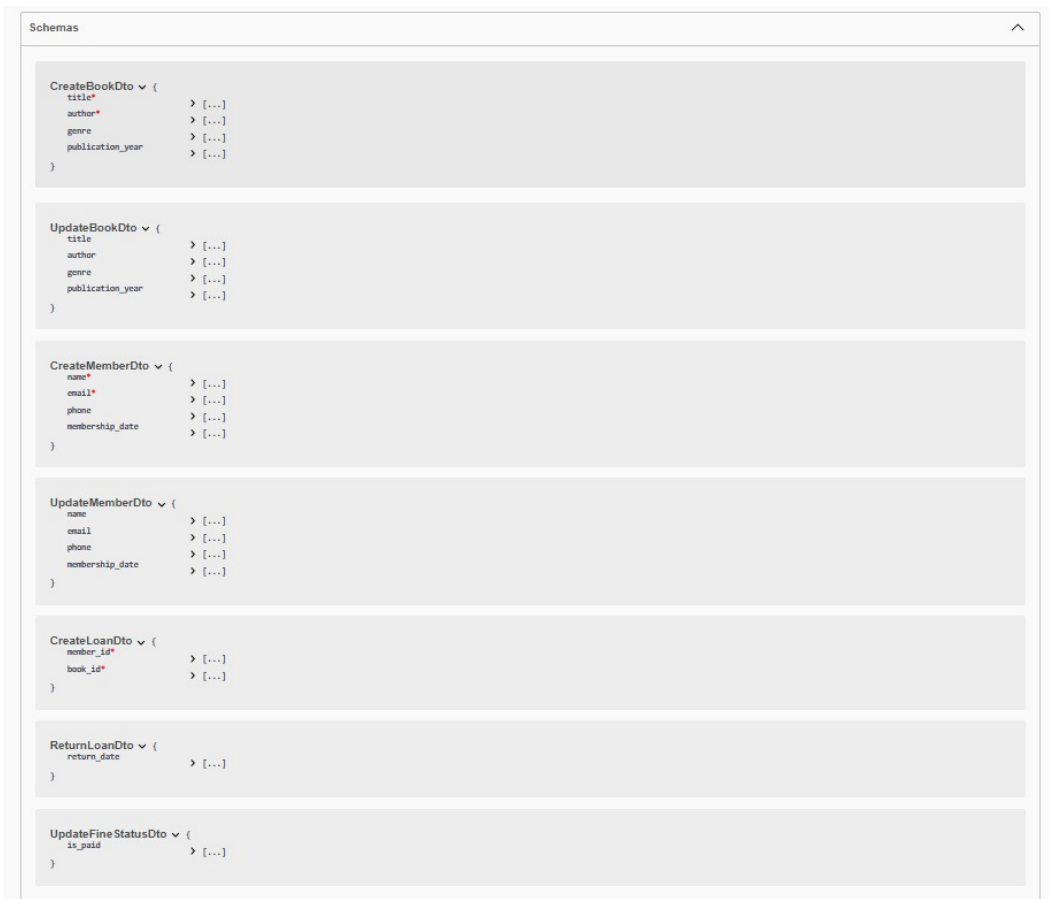


Figure 9: Definisi Skema DTO untuk Validasi Input

### 3. Modul Transaksi

Endpoint khusus untuk menangani peminjaman dan denda dipisahkan untuk menjaga modularitas kode.

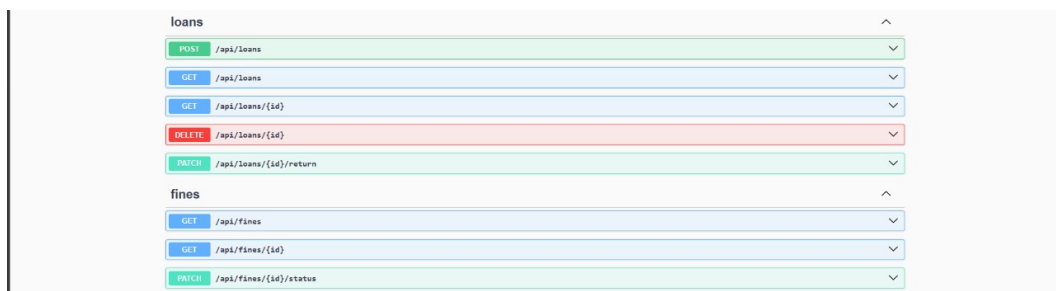


Figure 10: Dokumentasi Endpoint untuk Modul Peminjaman dan Denda

## 4.2 Pengujian Fungsional (Black Box Testing)

Untuk memverifikasi keandalan Backend, kami melakukan pengujian fungsional menggunakan `curl`. Pengujian ini mensimulasikan permintaan HTTP dari aplikasi luar.

## Skenario 1: Penambahan Data Buku (POST)

Pengujian dilakukan dengan mengirimkan payload JSON berisi data buku baru. Server merespon dengan kode status **201 Created**, yang menandakan data berhasil disimpan ke PostgreSQL dan ID unik telah digenerate.

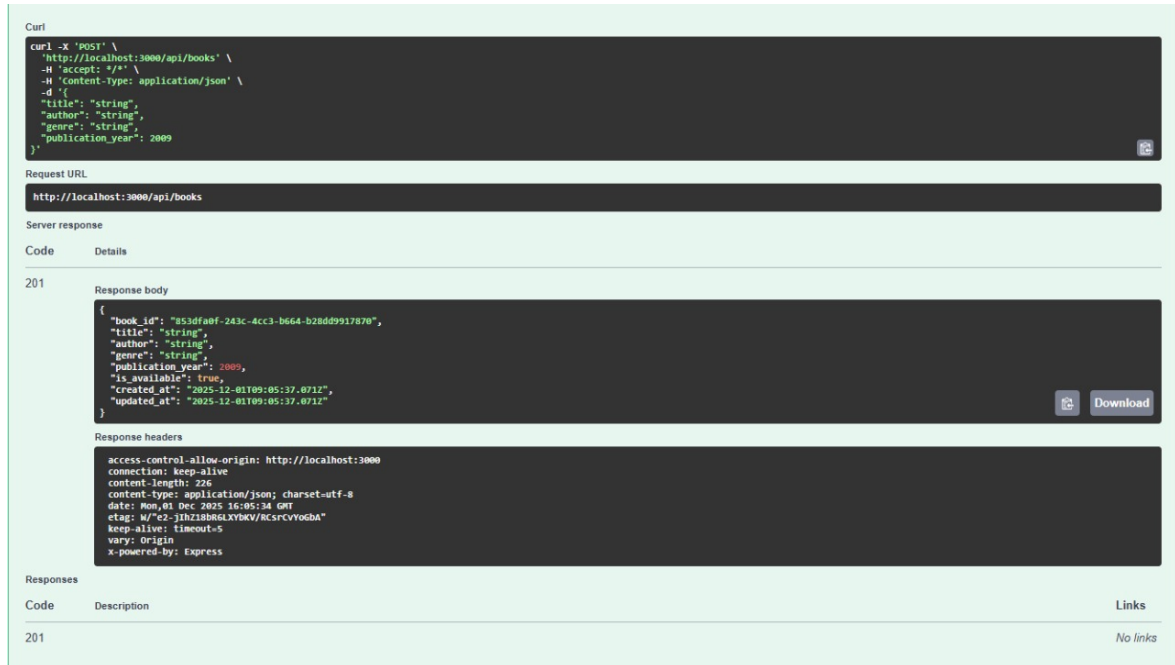


Figure 11: Hasil Uji Coba POST Request - Sukses 201

## Skenario 2: Penghapusan Data (DELETE)

Pengujian ini memverifikasi fitur hapus. Server merespon dengan kode **200 OK**, dan data buku terkait dipastikan hilang dari database serta tidak lagi muncul di hasil pencarian.

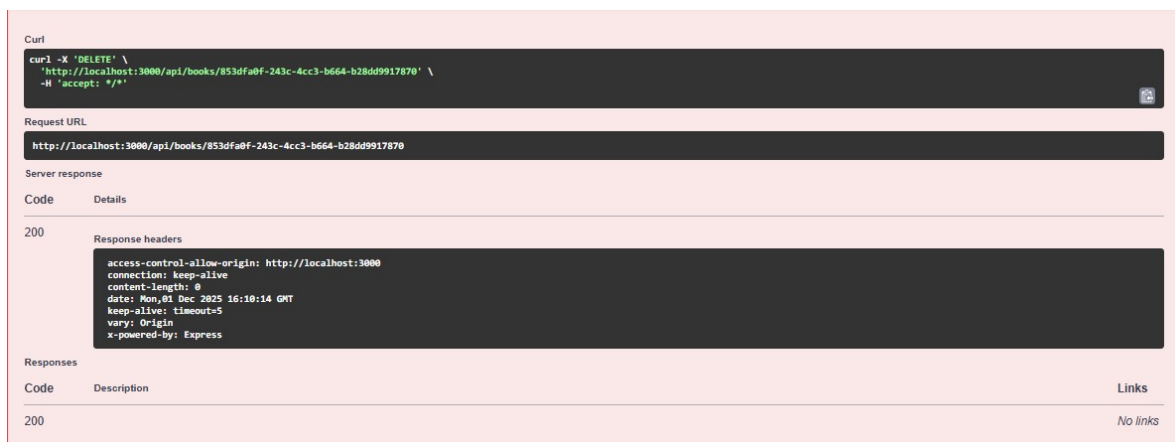


Figure 12: Hasil Uji Coba DELETE Request - Sukses 200

## BAB 5 KESIMPULAN DAN REFLEKSI

### 5.1 Kesimpulan Proyek

Pengembangan "Library Borrowing Tracker System" selama 5 minggu ini telah berhasil mentransformasi kebutuhan administrasi perpustakaan menjadi aplikasi web fungsional penuh.

1. **Transformasi Digital:** Kami berhasil menggantikan proses pencatatan manual dengan sistem database yang terstruktur, mengurangi risiko redundansi dan inkonsistensi data.
2. **Implementasi Logika Bisnis:** Fitur otomasi denda (Rp1.000/hari) dan batas waktu pinjam (14 hari) terbukti berjalan stabil, memberikan kepastian aturan bagi anggota perpustakaan.
3. **Integrasi Teknologi:** Sinergi antara PostgreSQL, NestJS, dan React membuktikan bahwa arsitektur terpisah (Decoupled Architecture) sangat efektif untuk membangun sistem yang skalabel dan mudah dipelihara.

### 5.2 Refleksi dan Pembelajaran

Perjalanan pengembangan ini memberikan banyak pelajaran berharga bagi tim kami:

- **Pentingnya Desain ERD:** Kami belajar bahwa waktu yang dihabiskan untuk merancang ERD yang matang di awal proyek sangat menghemat waktu saat fase coding. Normalisasi data mencegah kami dari masalah anomali data di kemudian hari.
- **Tantangan Zona Waktu:** Salah satu kendala teknis terbesar adalah sinkronisasi objek tanggal (*Date Objects*) antara server database (UTC) dan browser pengguna (Lokal). Kami mengatasinya dengan menstandarisasi seluruh format tanggal menggunakan ISO-8601 di seluruh lapisan aplikasi.
- **Dokumentasi API:** Penggunaan Swagger sangat membantu tim Frontend untuk memahami struktur data tanpa harus selalu bertanya kepada tim Backend, mempercepat proses integrasi tim.

### 5.3 Rencana Pengembangan Selanjutnya

Mengingat ini adalah prototipe akademis, sistem ini memiliki potensi besar untuk dikembangkan lebih lanjut dengan fitur-fitur seperti:

- **Notifikasi Email:** Mengirim pengingat otomatis kepada anggota H-1 sebelum jatuh tempo pengembalian.
- **Barcode Scanner:** Memungkinkan peminjaman buku dengan memindai barcode ISBN menggunakan kamera smartphone.
- **Dashboard Anggota:** Memberikan akses login kepada anggota untuk melihat riwayat peminjaman mereka sendiri secara mandiri.