

**LAPORAN PROJECT AKHIR PRAKTIKUM WEB SERVICE : APLIKASI  
PERPUSTAKAAN**

Dosen Pengampu: Aditya Ferdiana Arief, S.Kom., M.Kom



Disusun Oleh :

Rahel Royana Siregar\_5210311032

Eunike Clarissa Br Meliala\_5210311053

Vlavia Shallom Israellie Rahamauw\_5210311056

Faiza Putri Hasna\_5210311096

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2023**

## DAFTAR ISI

LAPORAN PROJECT AKHIR PRAKTIKUM WEB SERVICE : APLIKASI PERPUSTAKAAN .....	1
DAFTAR ISI .....	2
BAB I .....	3
Sistem dan software .....	3
1. Deskripsi Topik Projek.....	3
2. Database .....	3
3. Desain Rancangan Database .....	3
4. Method .....	3
5. Framework dan Packages .....	4
❖ Backend.....	4
❖ Frontend .....	4
Hasil Create Skeleton .....	5
BAB II .....	6
Langkah-langkah Pembuatan project .....	6
❖ Backend.....	6
❖ Frontend .....	7
BAB III .....	9
1. Test Backend Restful API dengan Postman .....	9
❖ USER .....	9
❖ ANGGOTA .....	13
❖ BUKU .....	16
❖ PEMINJAMAN .....	18
2. Tes FrontEnd dengan berfungsinya CRUD pada tampilan web.....	21
a) Login .....	21
b) Dashboard .....	21
c) Menu Buku.....	22
d) Menu Anggota.....	26
e) Menu Petugas / Users.....	30
f) Menu Peminjaman .....	34

## BAB I

### Sistem dan software

Persiapan Membuat Project (System dan beberapa software yang digunakan)

- Sistem Operasi : Windows
- Teks Editor : Visual Studio Code

#### 1. Deskripsi Topik Projek

Aplikasi ini dilengkapi dengan fitur peminjaman, buku, anggota, dan users yang memungkinkan operasi CRUD (Create, Read, Update, Delete). Aplikasi ini bertujuan untuk membantu mengelola proses pekerjaan pada sebuah perpustakaan dengan efisiensi yang tinggi. Dengan adanya fitur peminjaman, pengguna dapat dengan mudah melacak buku yang dipinjam dan mengatur pengembalian. Selain itu, fitur buku memungkinkan pengelolaan koleksi buku, termasuk informasi detail tentang setiap buku di perpustakaan. Fitur anggota memungkinkan pengelolaan informasi anggota perpustakaan, seperti data pribadi. Sementara fitur pengguna/users memungkinkan pengaturan akses dan hak istimewa untuk petugas perpustakaan. Dengan perancangan aplikasi perpustakaan ini, diharapkan proses manajemen perpustakaan dapat dilakukan dengan lebih efisien dan terstruktur.

#### 2. Database

Database yang digunakan adalah Mysql2. MySQL2 adalah driver database Ruby yang sangat cepat dan cocok untuk RESTful API. Kecepatan operasi databasenya memungkinkan RESTful API merespon dengan cepat. MySQL2 juga mendukung asynchronous operations sehingga request API tetap dapat dilayani meski sedang ada operasi database yang berjalan. Selain itu fitur prepared statements dan parameterized queries meningkatkan keamanan dari SQL injection. MySQL2 juga memiliki konektivitas keepalive yang rendah overhead sehingga mengurangi beban server database.

#### 3. Desain Rancangan Database

- Users (id, nama, email, password, role)
- Buku (id, judul, penulis, penerbit, kategori, tahunTerbit, stok, userId)
- Anggota (id, nama, alamat, email)
- Peminjaman (id, tgl\_pinjam, tgl\_kembali, bukuId, anggotaId, statusPinjam, userId)

#### 4. Method

- Users:  
router.get('/users', verifyUser, adminOnly, getUsers);  
router.get('/users/:id', verifyUser, adminOnly, getUserById)  
router.post('/users', verifyUser, adminOnly, createUser);  
router.patch('/users/:id', verifyUser, adminOnly, updateUser);  
router.delete('/users/:id', verifyUser, adminOnly, deleteUser);
- Buku:  
router.get('/bukus', verifyUser, getBukus);  
router.get('/bukus/:id', verifyUser, getBukuById)

```
router.post('/bukus', verifyUser, createBuku);
router.patch('/bukus/:id', verifyUser, updateBuku);
router.delete('/bukus/:id', verifyUser, deleteBuku);
```

- Anggota:

```
router.get('/anggotas', getAnggotas);
router.get('/anggotas/:id', getAnggotaById)
router.post('/anggotas', createAnggota);
router.patch('/anggotas/:id', updateAnggota);
router.delete('/anggotas/:id', deleteAnggota);
```

- Peminjaman:

```
router.get('/peminjamans', verifyUser, getPeminjamans);
router.get('/peminjamans/:id', verifyUser, getPeminjamanById)
router.post('/peminjamans', verifyUser, createPeminjaman);
router.patch('/peminjamans/:id', verifyUser, updatePeminjaman);
router.delete('/peminjamans/:id', verifyUser, deletePeminjaman);
```

- AuthRoute:

```
router.get('/me', Me);
router.post('/login', Login)
router.delete('/logout', logOut);
```

## 5. Framework dan Packages

### ❖ Backend

#### ➤ Framework : Express

#### ➤ Packages :

- Argon2 : Package untuk mengamankan kata sandi
- Connect Session Sequelize : Penyimpanan sesi Sequelize untuk express
- Cors : Middleware untuk menangani permintaan lintas domain di sisi server
- Dotenv : Memisahkan konfigurasi dan kode
- Express Session : Menyediakan middleware manajemen sesi untuk Express
- Express : Framework web untuk nodejs
- Morgan : Melihat semua log API di nodejs
- Mysql2 : Database Mysql untuk node.js
- Sequelize : ORM (Object-Relational Mapper) untuk berinteraksi dengan database MySQL.

### ❖ Frontend

#### ➤ Framework : React

#### ➤ Packages :

- axios: Library untuk melakukan HTTP requests.
- bulma: Kerangka kerja CSS yang dapat digunakan untuk mempercantik tata letak aplikasi
- react: Library utama untuk membangun antarmuka pengguna dengan komponen-react.

- react-dom: Paket ini menyediakan metode spesifik dari DOM yang dapat digunakan oleh React.
- react-redux: Integrasi Redux untuk React, memungkinkan pengelolaan state global.
- react-router-dom: Library untuk menangani routing dalam aplikasi React.
- @reduxjs/toolkit: Alat resmi untuk Redux, menyediakan berbagai fitur untuk mengelola state.
- web-vitals: Alat untuk mengukur dan melacak kinerja aplikasi web.

Hasil Create Skeleton

```
C:\TugasBesar
├── backend/
│   ├── config/
│   ├── controllers/
│   ├── index.js
│   ├── middleware/
│   ├── models/
│   ├── node_modules/
│   ├── package-lock.json
│   ├── package.json
│   ├── request.rest
│   └── routes/
├── frontend/
│   ├── node_modules/
│   ├── package-lock.json
│   ├── package.json
│   ├── public/
│   ├── README.md
│   └── src/
└── out.txt
```

## BAB II

### Langkah-langkah Pembuatan project

#### Membuat Folder :

##### ❖ Backend

##### ➤ Buat Proyek Node.js

- Membuat direktori backend (mkdir backend)
- Pindah ke direktori (cd backend)
- Inisialisasi proyek Node.js (npm init -y)

##### ➤ Menginstal package Node.js

- npm i express mysql sequelize argon2 cors dotenv
- npm i express-session
- npm install nodemon
- npm install morgan

##### ➤ Membuat file index.js (Membuat file entry point untuk mengonfigurasi dan menjalankan aplikasi Express.js)

##### ➤ Membuat file .env (Menyimpan konfigurasi port)

##### ➤ Membuat folder model (Mendefinisikan struktur data yang akan dibuat di dalam database)

- AnggotaModel.js
- BukuModel.js
- PeminjamanModel.js
- UserModel.js

##### ➤ Membuat folder controller (Mengimplementasikan operasi CRUD yang disimpan dalam database)

- Anggota.js
- Auth.js
- Buku.js
- Peminjaman.js
- Users.js

##### ➤ Membuat folder config (Konfigurasi untuk membuat objek sequelize)

- Database.js

##### ➤ Membuat folder routes (Membuat endpoint API)

- AnggotaRoute.js
- AuthRoute.js
- BukuRoute.js
- PeminjamanRoute.js
- UserRoute.js

##### ➤ Menyimpan sesi sequelize untuk express (npm i connect-session-sequelize)

##### ➤ Membuat folder middleware (Untuk mengelola otentikasi)

- AuthUser.js

## ❖ Frontend

- Buat Proyek React.jsx
  - Inisialisasi proyek Node.js (npx create-react-app frontend --template redux)
  - Pindah ke direktori (cd frontend)
- Instal package
  - npm i react-router-dom axios bulma react-icons
- Membuat file index.js
- Menginstal ekstension
  - ES7 + React/Redux/Native snippets
  - Bulma Snippet
- Membuat folder app
  - Store.js (Untuk konfigurasi dan pembuatan store Redux)
- Membuat folder components (Desain dan rancangan tata letak UI)
  - AnggotaList.jsx
  - FormAddAnggota.jsx
  - FormEditAnggota.jsx
  - BukuList.jsx
  - FormAddBuku.jsx
  - FormEditBuku.jsx
  - PeminjamanList.jsx
  - FormAddPeminjaman.jsx
  - FormEditPeminjaman.jsx
  - UserList.jsx
  - FormAddUser.jsx
  - FormEditUser.jsx
  - Login.jsx
  - Navbar.jsx
  - Sidebar.jsx
  - Welcome.jsx
- Membuat folder features
  - authSlice.js (Menyimpan slice atau reducer Redux terkait otentikasi)
- Membuat folder pages (Menyimpan halaman-halaman aplikasi)
  - Anggota.jsx
  - AddAnggota.jsx
  - EditAnggota.jsx
  - Buku.jsx
  - AddBuku.jsx
  - EditBuku.jsx
  - Peminjaman.jsx
  - AddPeminjaman.jsx

- EditPeminjaman.jsx
  - User.jsx
  - AddUser.jsx
  - EditUser.jsx
  - Dashboard.jsx
- Mengisi file App.js (Konfigurasi routing)

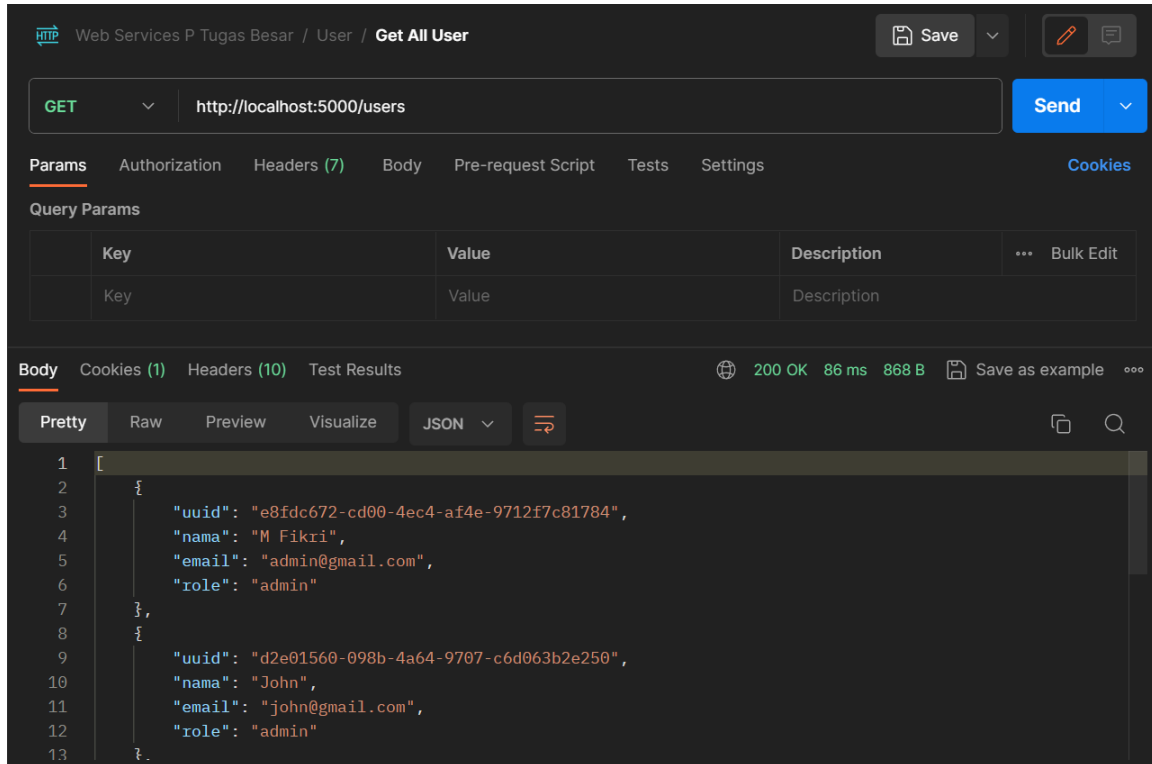


## BAB III

### 1. Test Backend Restful API dengan Postman

#### ❖ USER

##### - Get All User



Postman interface showing a GET request to `http://localhost:5000/users`. The response is a JSON array of two user objects, returned with a 200 OK status.

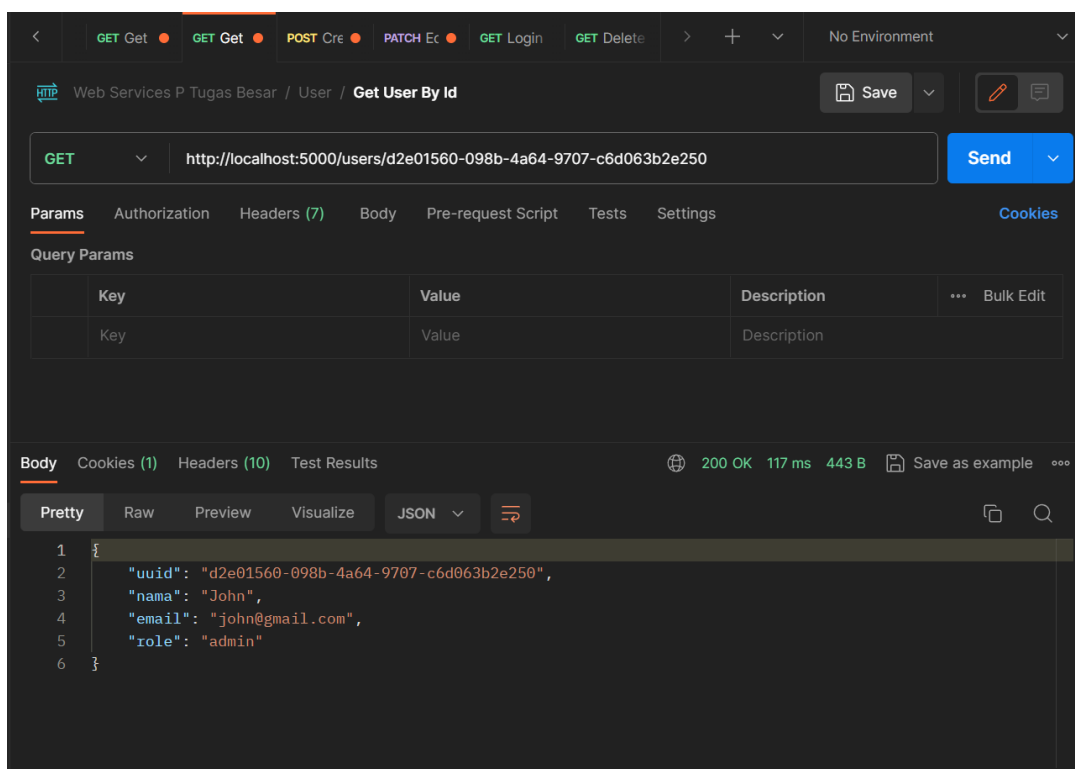
**Query Params**

Key	Value	Description
Key	Value	Description

**Body**

```
1 [
2   {
3     "uuid": "e8fdc672-cd00-4ec4-af4e-9712f7c81784",
4     "nama": "M Fikri",
5     "email": "admin@gmail.com",
6     "role": "admin"
7   },
8   {
9     "uuid": "d2e01560-098b-4a64-9707-c6d063b2e250",
10    "nama": "John",
11    "email": "john@gmail.com",
12    "role": "admin"
13  }
```

##### - Get User By Id



Postman interface showing a GET request to `http://localhost:5000/users/d2e01560-098b-4a64-9707-c6d063b2e250`. The response is a single user object, returned with a 200 OK status.

**Query Params**

Key	Value	Description
Key	Value	Description

**Body**

```
1 {
2   "uuid": "d2e01560-098b-4a64-9707-c6d063b2e250",
3   "nama": "John",
4   "email": "john@gmail.com",
5   "role": "admin"
6 }
```

## - Create User

The screenshot shows the Postman interface for a POST request to `http://localhost:5000/users`. The request body is a JSON object with the following fields: `nama` (rahelsss), `email` (rahels@gmail.com), `password` (123456), `confPassword` (123456), and `role` (admin). The response status is 201 Created, with a response time of 546 ms and a body size of 373 B. The response body is a JSON object with the field `msg` (Register Berhasil).

```
1 {
2   "nama": "rahelsss",
3   "email": "rahels@gmail.com",
4   "password": "123456",
5   "confPassword": "123456",
6   "role": "admin"
7 }
```

```
1 {
2   "msg": "Register Berhasil"
3 }
```

## - Edit User

The screenshot shows the Postman interface for a PATCH request to `http://localhost:5000/users/cf333228-52ba-44bb-9387-be838b878788`. The request body is a JSON object with the following fields: `nama` (Rahels Updated), `email` (rahels@gmail.com), `password` (empty), `confPassword` (empty), and `role` (admin). The response status is 200 OK, with a response time of 96 ms and a body size of 363 B. The response body is a JSON object with the field `msg` (User Updated).

```
1 {
2   "nama": "Rahels Updated",
3   "email": "rahels@gmail.com",
4   "password": "",
5   "confPassword": "",
6   "role": "admin"
7 }
```

```
1 {
2   "msg": "User Updated"
3 }
```

## - Delete User

The screenshot shows the Postman interface for a DELETE request. The URL is `http://localhost:5000/users/cf333228-52ba-44bb-9387-be838b878788`. The response status is 200 OK, with a response time of 91 ms and a body size of 363 B. The response body is a JSON object: `{ "msg": "User Deleted" }`.

**DELETE** `http://localhost:5000/users/cf333228-52ba-44bb-9387-be838b878788` **Send**

**Params** Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

**Query Params**

Key	Value	Description
Key	Value	Description

**Body** Cookies (1) Headers (10) Test Results 200 OK 91 ms 363 B Save as example

**Pretty** Raw Preview Visualize JSON `{`  
1  `"msg": "User Deleted"`  
2 `}`  
3

## - Login

The screenshot shows the Postman interface for a POST request. The URL is `http://localhost:5000/login`. The response status is 200 OK, with a response time of 91 ms and a body size of 363 B. The response body is a JSON object: `{ "email": "ichaa@gmail.com", "password": "123456" }`.

**POST** `http://localhost:5000/login` **Send**

**Params** Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON **Beautify**

`{`  
1  `"email": "ichaa@gmail.com",`  
2  `"password": "123456"`  
3 `}`  
4

## - Me

Web Services P Tugas Besar / User / Me

GET http://localhost:5000/me

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies (1) Headers (10) Test Results 200 OK 32 ms 445 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "uuid": "3c3c29a2-33ac-47bc-9671-215e28912e1b",
3   "nama": "ichaa",
4   "email": "ichaa@gmail.com",
5   "role": "admin"
6 }
```

## - Logout

Web Services P Tugas Besar / User / Logout

DELETE http://localhost:5000/logout

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies (1) Headers (10) Test Results 200 OK 83 ms 368 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "Anda telah logout"
3 }
```

## ❖ ANGGOTA

### - Get All Anggota

Web Services P Tugas Besar / Anggota / **Get All Anggota**

GET `http://localhost:5000/anggotas` **Send**

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies (1) Headers (10) Test Results 200 OK 79 ms 410 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "nama": "icha",
5     "alamat": "sleman",
6     "email": "vlavia@gmail.com"
7   }
8 ]
```

### - Get Anggota by Id

Web Services P Tugas Besar / Anggota / **Get Anggota By Id**

GET `http://localhost:5000/anggotas/1` **Send**

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies (1) Headers (10) Test Results 200 OK 37 ms 408 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "nama": "icha",
4   "alamat": "sleman",
5   "email": "vlavia@gmail.com"
6 }
```

- Create new Anggota

The screenshot shows the Postman interface for a POST request. The URL is `http://localhost:5000/anggotas`. The request body is a JSON object: `{ "nama": "ichami", "alamat": "Yogya", "email": "ichami@gmail.com" }`. The response status is 201 Created, with a response time of 94 ms and a body size of 384 B. The response body is a JSON object: `{ "msg": "Anggota Berhasil Ditambahkan" }`.

```
1 {
2   "nama": "ichami",
3   "alamat": "Yogya",
4   "email": "ichami@gmail.com"
5 }
```

```
1 {
2   "msg": "Anggota Berhasil Ditambahkan"
3 }
```

- Edit Anggota

The screenshot shows the Postman interface for a PATCH request. The URL is `http://localhost:5000/anggotas/5`. The request body is a JSON object: `{ "nama": "ichamilsss", "alamat": "Bandung", "email": "ichamils@gmail.com" }`. The response status is 200 OK, with a response time of 97 ms and a body size of 382 B. The response body is a JSON object: `{ "msg": "Data anggota berhasil diUpdate" }`.

```
1 {
2   "nama": "ichamilsss",
3   "alamat": "Bandung",
4   "email": "ichamils@gmail.com"
5 }
```

```
1 {
2   "msg": "Data anggota berhasil diUpdate"
3 }
```

## - Delete Anggota

Web Services P Tugas Besar / Anggota / Delete Anggota

DELETE  Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies (1) Headers (10) Test Results 200 OK 40 ms 366 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "Anggota Deleted"
3 }
```

## ❖ BUKU

### - Get All Buku

Web Services P Tugas Besar / Buku / **Get All Buku**

GET  **Send**

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies (1) Headers (10) Test Results 200 OK 43 ms 999 B Save as example

Pretty Raw Preview Visualize JSON

```
27  {,
28  {
29    "uuid": "cb515281-3d5d-4c7a-9150-3497d7e6dd93",
30    "judul": "Matematika",
31    "penulis": "Eunike",
32    "penerbit": "Erlangga",
33    "kategori": "Pendidikan",
34    "tahunTerbit": 2019,
35    "stok": 10,
36    "user": {
37      "nama": "ichaa",
38      "email": "ichaa@gmail.com"
39    }
  }
```

### - Get Buku By Id

Web Services P Tugas Besar / Buku / **Get Buku By Id**

GET  **Send**

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies (1) Headers (10) Test Results 200 OK 58 ms 570 B Save as example

Pretty Raw Preview Visualize JSON

```
1  {
2    "uuid": "a111d982-5bc9-4494-8a84-bfe78bc47cbd",
3    "judul": "Habis Gelap terbitlah terang",
4    "penulis": "Rahel",
5    "penerbit": "Erlangga",
6    "kategori": "Pendidikan",
7    "tahunTerbit": 1999,
8    "stok": 5,
9    "user": {
10     "nama": "ichaa",
11     "email": "ichaa@gmail.com"
12   }
13 }
```



## - Create new Buku

The screenshot shows the Postman interface for a POST request. The URL is `http://localhost:5000/bukus`. The request body is a JSON object with the following fields: `judul` (Web Service Praktik 2), `penulis` (Afwan Angga), `penerbit` (UTY), `kategori` (Komputer), `tahunTerbit` (2021), and `stok` (7). The response status is 201 Created, with a response time of 50 ms and a body size of 379 B. The response body is a JSON object with a `msg` field containing the text "Buku Sukses Ditambahkan".

```
1 {
2   "judul": "Web Service Praktik 2",
3   "penulis": "Afwan Angga",
4   "penerbit": "UTY",
5   "kategori": "Komputer",
6   "tahunTerbit": 2021,
7   "stok": 7
8 }
```

```
1 {
2   "msg": "Buku Sukses Ditambahkan"
3 }
```

## - Edit Buku

The screenshot shows the Postman interface for a PATCH request. The URL is `http://localhost:5000/bukus/f4e1a2b0-d95c-4ea1-b41a-534bb8533d6c`. The request body is a JSON object with the following fields: `judul` (Web Service Praktik 2 baru), `penulis` (Afwan Angga), `penerbit` (UTY), `kategori` (Komputer), `tahunTerbit` (2021), and `stok` (6). The response status is 200 OK, with a response time of 66 ms and a body size of 377 B. The response body is a JSON object with a `msg` field containing the text "Buku berhasil diperbaharui".

```
1 {
2   "judul": "Web Service Praktik 2 baru",
3   "penulis": "Afwan Angga",
4   "penerbit": "UTY",
5   "kategori": "Komputer",
6   "tahunTerbit": 2021,
7   "stok": 6
8 }
```

```
1 {
2   "msg": "Buku berhasil diperbaharui"
3 }
```

## - Delete Buku

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** `http://localhost:5000/bukus/f4e1a2b0-d95c-4ea1-b41a-534bb8533d6c`
- Status:** 200 OK, 95 ms, 372 B
- Response Body (JSON):**

```
{  "msg": "Buku berhasil dihapus"}
```

## ❖ PEMINJAMAN

### - Get All Peminjaman

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `http://localhost:5000/peminjamans`
- Status:** 200 OK, 93 ms, 1.04 KB
- Response Body (JSON):**

```
[  {    "uuid": "03317c1c-5a87-4ba9-9cbb-b4f3e9c0a507",    "tgl_pinjam": "2024-11-12T00:00:00.000Z",    "tgl_kembali": "2024-01-26T00:00:00.000Z",    "bukuId": 1,    "anggotaId": 1,    "statusPinjam": "Telah dikembalikan",    "user": {      "nama": "ichaa",      "email": "ichaa@gmail.com"    }  }]
```

## - Get Peminjaman by Id

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `http://localhost:5000/peminjamans/be07c662-5208-4e61-8b0b-9dd8583987fa`
- Params:** Query Params table with columns: Key, Value, Description.
- Body:** JSON response showing loan details:

```
1 {
2   "uuid": "be07c662-5208-4e61-8b0b-9dd8583987fa",
3   "tgl_pinjam": "2023-01-29T00:00:00.000Z",
4   "tgl_kembali": "2023-01-31T00:00:00.000Z",
5   "bukuId": 1,
6   "anggotaId": 1,
7   "statusPinjam": "Telah dikembalikan",
8   "user": {
9     "nama": "ichaa",
10    "email": "ichaa@gmail.com"
11  }
12 }
```
- Status:** 200 OK, 76 ms, 581 B

## - Create new Peminjaman

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** `http://localhost:5000/peminjamans`
- Body:** JSON request body:

```
1 {
2   ... "tgl_pinjam": "2023-01-29",
3   ... "tgl_kembali": "2023-01-31",
4   ... "bukuId": 1,
5   ... "anggotaId": 1
6 }
```
- Status:** 201 Created, 71 ms, 385 B
- Body:** JSON response showing success message:

```
1 {
2   "msg": "Peminjaman Sukses Ditambahkan"
3 }
```

## - Edit Peminjaman

The screenshot shows a Postman interface for a PATCH request. The URL is `http://localhost:5000/peminjamans/be07c662-5208-4e61-8b0b-9dd8583987fa`. The request body is a JSON object with the following fields:

```
1 {
2   "tgl_pinjam": "2023-01-29",
3   "tgl_kembali": "2023-01-31",
4   "bukuId": 1,
5   "anggotaId": 1
6 }
```

The response is a JSON object with a success message:

```
1 {
2   "msg": "Peminjaman berhasil diperbaharui"
3 }
```

The status bar indicates a 200 OK response with a 99 ms response time and 383 B of data.

## - Delete Peminjaman

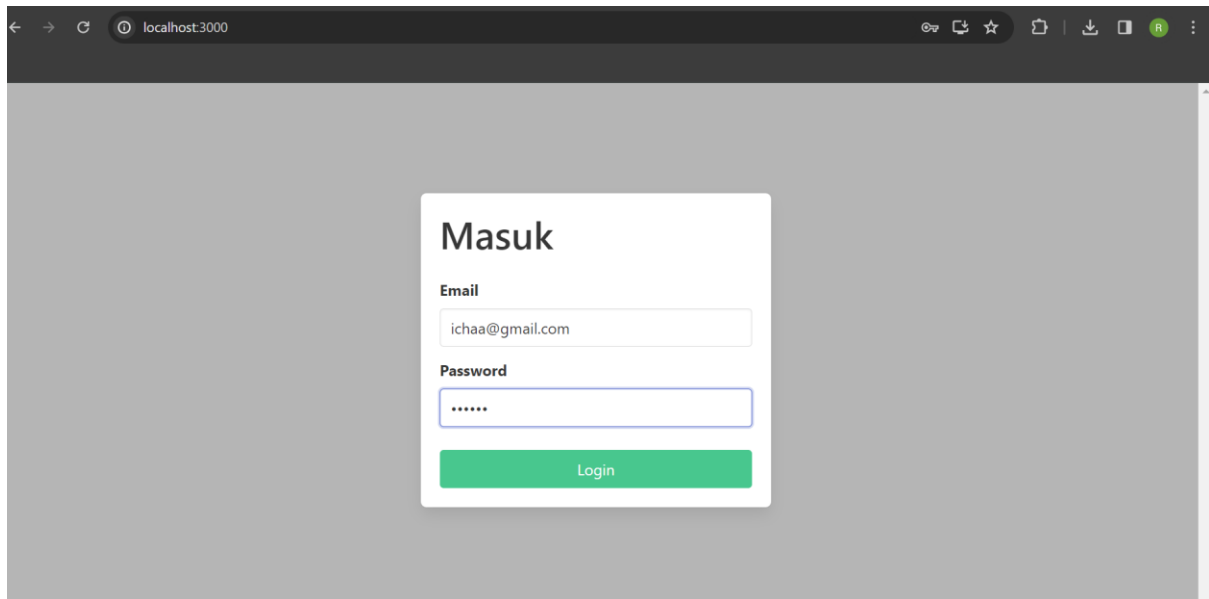
The screenshot shows a Postman interface for a DELETE request. The URL is `http://localhost:5000/peminjamans/be07c662-5208-4e61-8b0b-9dd8583987fa`. The response is a JSON object with a success message:

```
1 {
2   "msg": "Peminjaman berhasil dihapus"
3 }
```

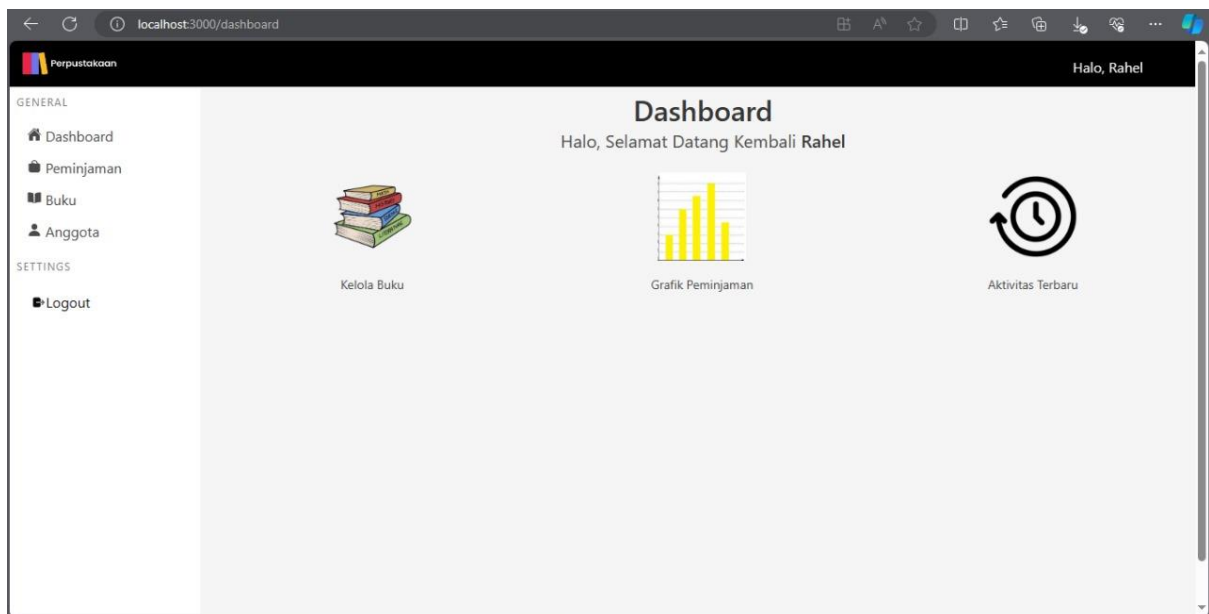
The status bar indicates a 200 OK response with a 63 ms response time and 378 B of data.

## 2. Tes FrontEnd dengan berfungsinya CRUD pada tampilan web

### a) Login



### b) Dashboard



### c) Menu Buku

- Get All Buku

The screenshot shows a web application interface for a library. The top navigation bar includes a logo, the name 'Perpustakaan', and a user greeting 'Halo, ichaa'. The left sidebar contains navigation links under 'GENERAL' (Dashboard, Peminjaman, Buku, Anggota), 'ADMIN' (Petugas), and 'SETTINGS' (Logout). The main content area is titled 'Daftar Buku' and features a 'Tambah Buku Baru' button. Below the button is a table listing books.

No	Judul	Penulis	Penerbit	Kategori	Tahun Terbit	Stok	Actions
1	Habis Gelap terbitlah terang	Rahel	Erlangga	Pendidikan	1999	5	<a href="#">Edit</a> <a href="#">Delete</a>
2	Matematika	Eunike	Erlangga	Pendidikan	2019	10	<a href="#">Edit</a> <a href="#">Delete</a>
3	Web Service Praktik 2	Afwan Angga	UTY	Komputer	2021	7	<a href="#">Edit</a> <a href="#">Delete</a>

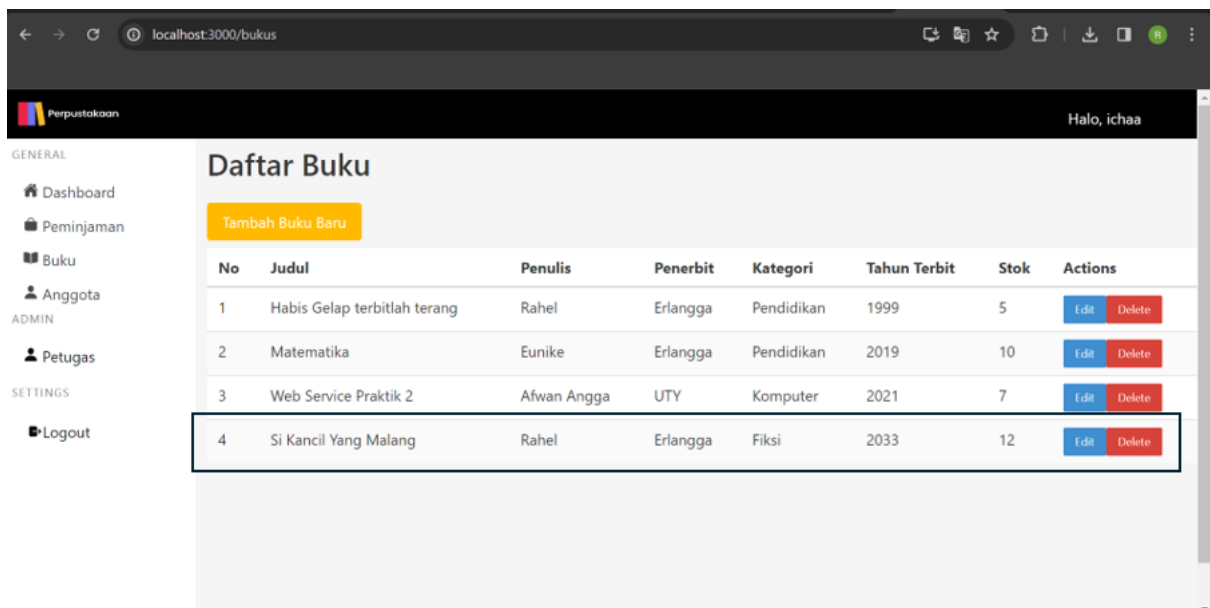
- Create New Buku

The screenshot shows the 'Create New Buku' form in the web application. The form is open, displaying fields for book details. The background shows the 'Daftar Buku' table from the previous screenshot.

Form Fields:

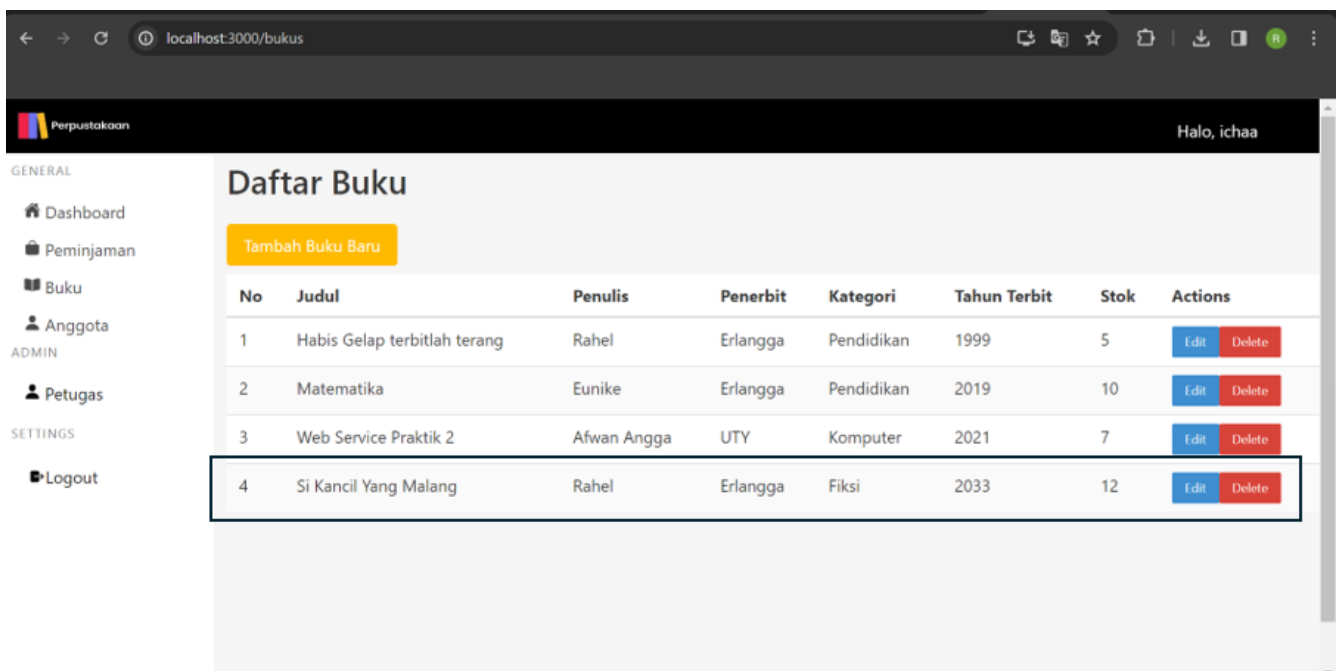
- Judul: Si Kancil Yang Malang
- Penulis: Rahel
- Penerbit: Erlangga
- Kategori: Fiksi (dropdown menu)
- Tahun Terbit: 2033
- Stok: 12

A 'Save' button is located at the bottom of the form.



Data Buku Berhasil Ditambahkan

- Edit Buku



Perpustakaan Halo, ichaa

### Update Buku

**Judul**  
Si Kancil dan Buaya

**Penerbit**  
Erlangga

**Penulis**  
Rahel

**Kategori**  
Fiksi

**Tahun Terbit**  
2033

**Stok**  
12

Update

Data buku berhasil diedit

Perpustakaan Halo, ichaa

### Daftar Buku

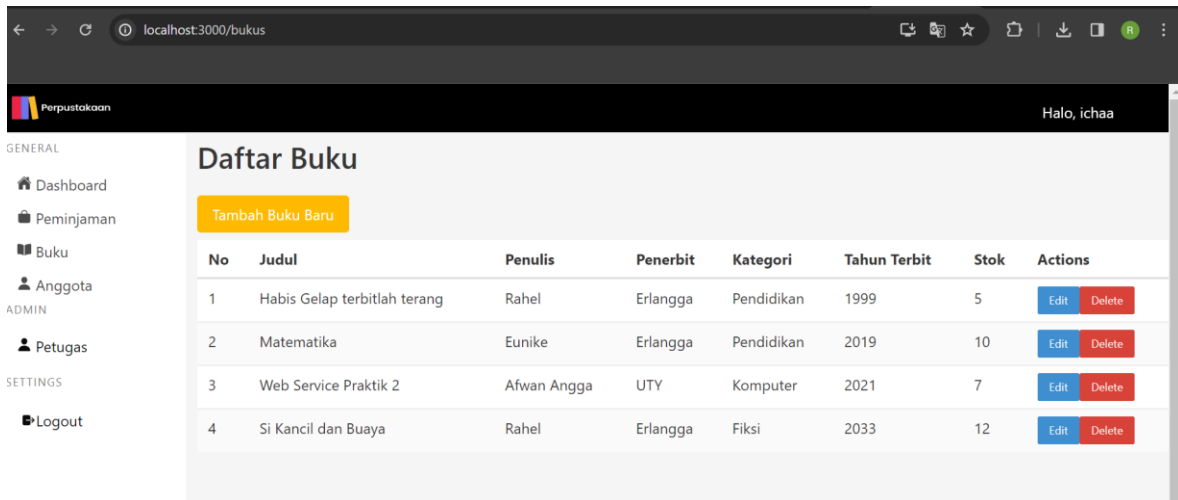
Tambah Buku Baru

No	Judul	Penulis	Penerbit	Kategori	Tahun Terbit	Stok	Actions
1	Habis Gelap terbitlah terang	Rahel	Erlangga	Pendidikan	1999	5	<a href="#">Edit</a> <a href="#">Delete</a>
2	Matematika	Eunike	Erlangga	Pendidikan	2019	10	<a href="#">Edit</a> <a href="#">Delete</a>
3	Web Service Praktik 2	Afwan Angga	UTY	Komputer	2021	7	<a href="#">Edit</a> <a href="#">Delete</a>
4	Si Kancil dan Buaya	Rahel	Erlangga	Fiksi	2033	12	<a href="#">Edit</a> <a href="#">Delete</a>



- Delete Data Buku

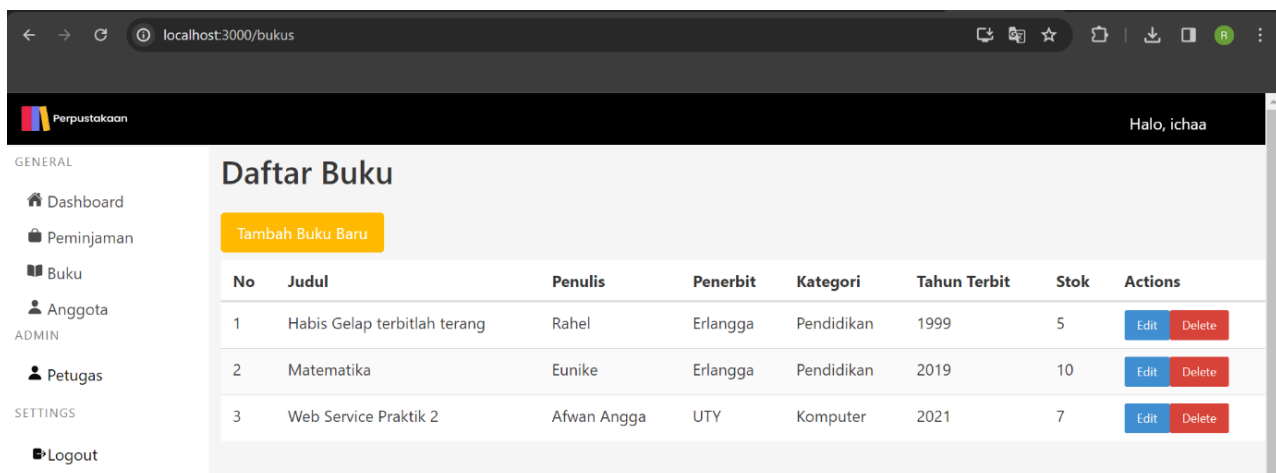
Sebelum data nomor 4 dihapus:



The screenshot shows a web browser at localhost:3000/bukus. The application has a dark header with 'Perpustakaan' and a user greeting 'Halo, ichaa'. A sidebar on the left contains navigation links: Dashboard, Peminjaman, Buku, Anggota, Petugas, and Logout. The main content area is titled 'Daftar Buku' and features a yellow 'Tambah Buku Baru' button. Below the button is a table with 4 rows of book data. Each row has 'Edit' and 'Delete' buttons in the 'Actions' column.

No	Judul	Penulis	Penerbit	Kategori	Tahun Terbit	Stok	Actions
1	Habis Gelap terbitlah terang	Rahel	Erlangga	Pendidikan	1999	5	<a href="#">Edit</a> <a href="#">Delete</a>
2	Matematika	Eunike	Erlangga	Pendidikan	2019	10	<a href="#">Edit</a> <a href="#">Delete</a>
3	Web Service Praktik 2	Afwan Angga	UTY	Komputer	2021	7	<a href="#">Edit</a> <a href="#">Delete</a>
4	Si Kancil dan Buaya	Rahel	Erlangga	Fiksi	2033	12	<a href="#">Edit</a> <a href="#">Delete</a>

Setelah data nomor 4 berhasil dihapus:

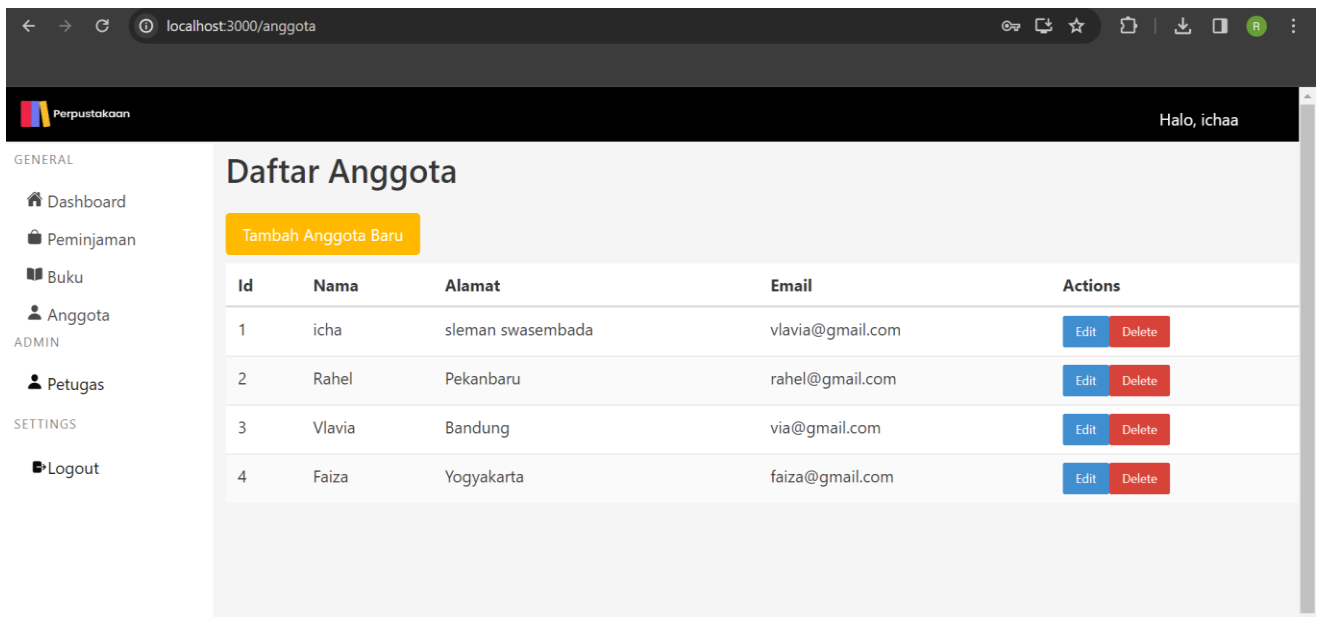


The screenshot shows the same web application after deleting the book with ID 4. The table now only contains 3 rows of data. The 'Delete' button for the last row (ID 3) is no longer visible.

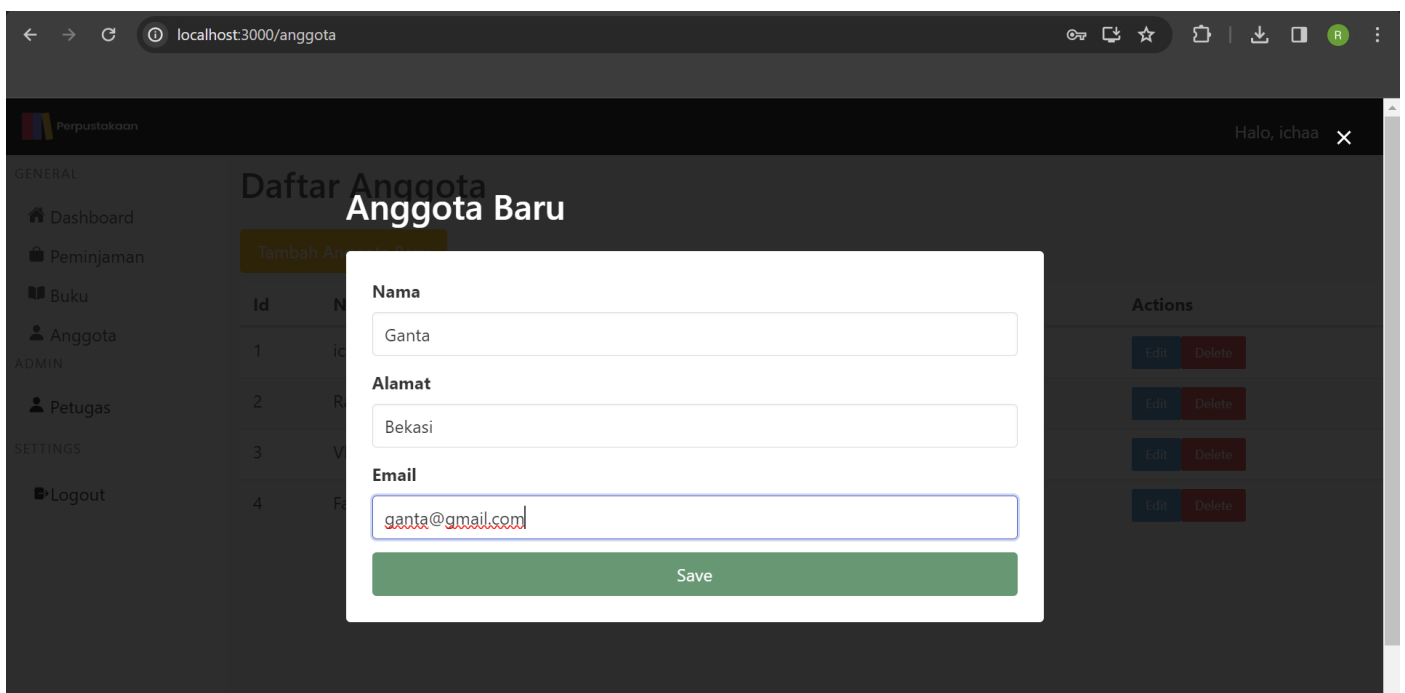
No	Judul	Penulis	Penerbit	Kategori	Tahun Terbit	Stok	Actions
1	Habis Gelap terbitlah terang	Rahel	Erlangga	Pendidikan	1999	5	<a href="#">Edit</a> <a href="#">Delete</a>
2	Matematika	Eunike	Erlangga	Pendidikan	2019	10	<a href="#">Edit</a> <a href="#">Delete</a>
3	Web Service Praktik 2	Afwan Angga	UTY	Komputer	2021	7	<a href="#">Edit</a>

d) Menu Anggota

- Get All Anggota



- Create New Anggota



Data anggota berhasil ditambahkan:

localhost:3000/anggota

Pengusfokkoon Halo, ichaa

GENERAL

- Dashboard
- Peminjaman
- Buku
- Anggota

ADMIN

- Petugas

SETTINGS

- Logout

## Daftar Anggota

Tambah Anggota Baru

Id	Nama	Alamat	Email	Actions
1	icha	sieman swasembada	vlavia@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
2	Rahel	Pekanbaru	rahel@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
3	Vlavia	Bandung	via@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
4	Faiza	Yogyakarta	faiza@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
5	Ganta	Bekasi	ganta@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>

- Edit Data Anggota

localhost:3000/anggota/edit/30

Pengusfokkoon Halo, ichaa

GENERAL

- Dashboard
- Peminjaman
- Buku
- Anggota

ADMIN

- Petugas

SETTINGS

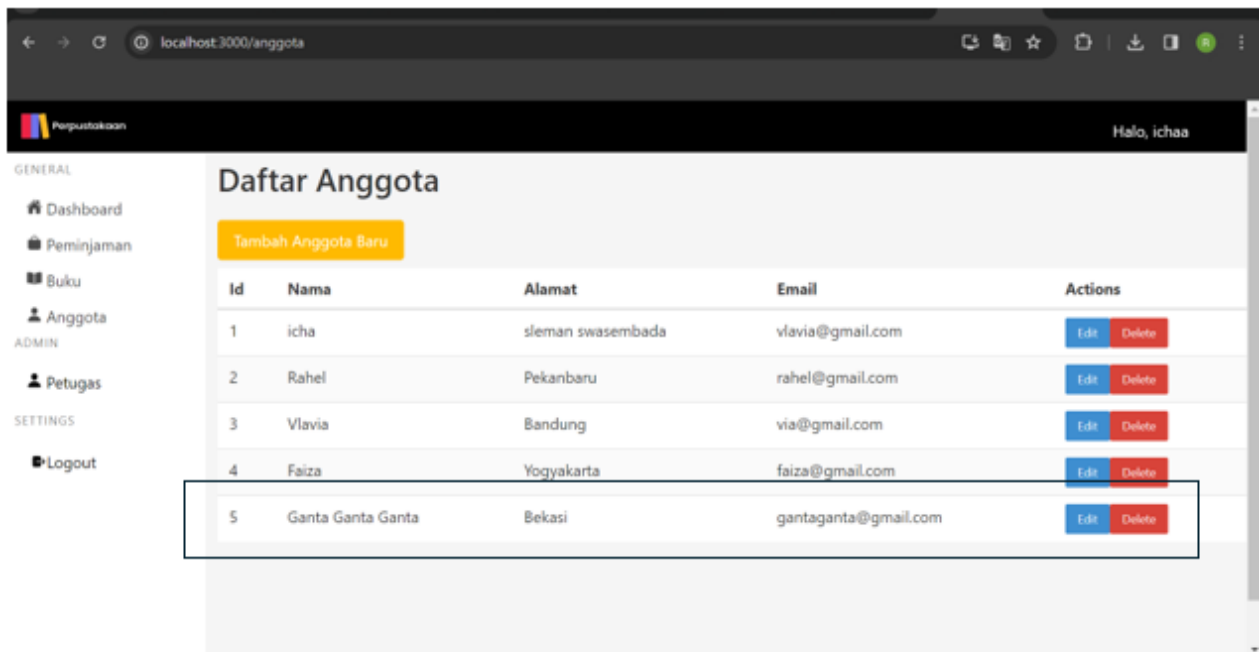
- Logout

## Update Anggota

Name: Ganta Ganta Ganta Email: gantaganta@gmail.com

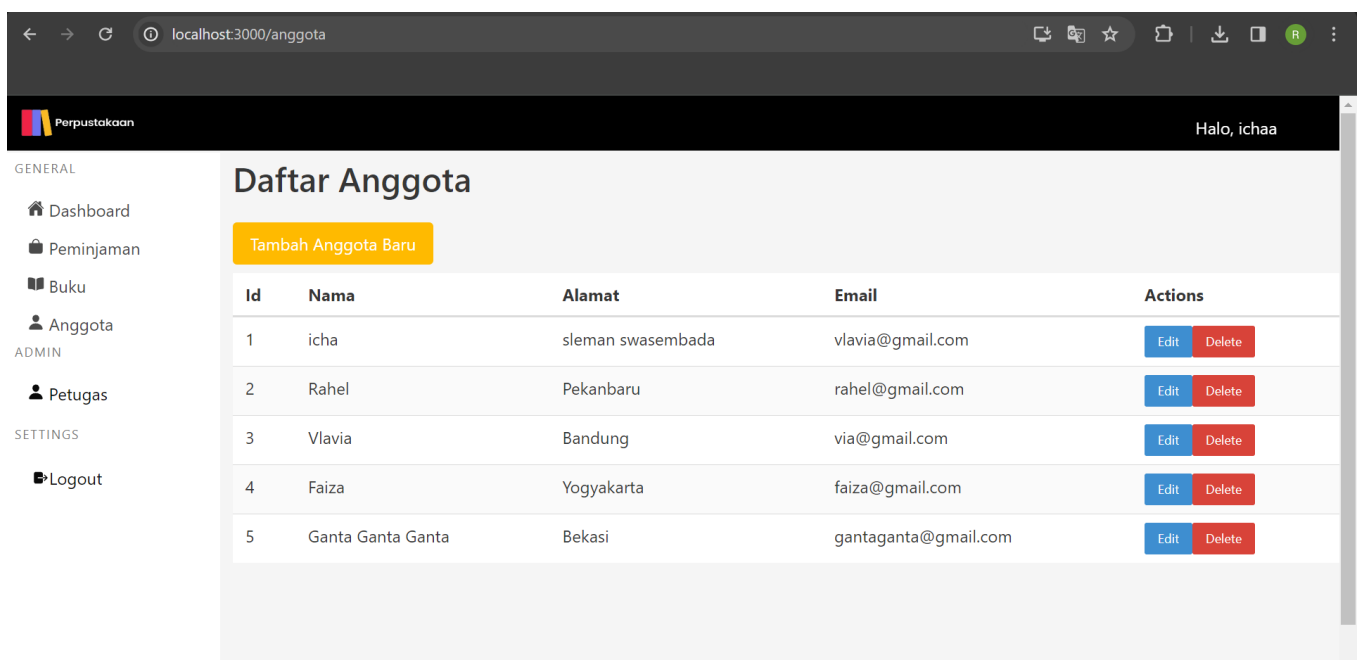
Alamat: Bekasi

[Update](#)



- Delete Data Anggota

Sebelum menghapus data baris ke 5:



Setelah berhasil menghapus data baris ke 5:

The screenshot shows a web application interface for a library (Perpustakaan). The browser address bar indicates the URL is localhost:3000/anggota. The user is logged in as 'ichaa'. The main content area is titled 'Daftar Anggota' and features a yellow button labeled 'Tambah Anggota Baru'. Below this is a table with 5 columns: Id, Nama, Alamat, Email, and Actions. The table contains 4 rows of data. The left sidebar shows navigation links for GENERAL (Dashboard, Peminjaman, Buku, Anggota), ADMIN (Petugas), and SETTINGS (Logout).

Id	Nama	Alamat	Email	Actions
1	icha	sleman swasembada	vlavia@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
2	Rahel	Pekanbaru	rahel@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
3	Vlavia	Bandung	via@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
4	Faiza	Yogyakarta	faiza@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>

### e) Menu Petugas / Users

- Get all Users

The screenshot shows a web application interface for managing users. The browser address bar indicates the URL is `localhost:3000/users`. The page title is "Daftar Users". A sidebar on the left contains navigation links: "Dashboard", "Peminjaman", "Buku", "Anggota", "Petugas", and "Logout". The main content area displays a table of users with columns: "No", "Name", "Email", "Role", and "Actions". The table contains 4 rows of data. A "Tambah Users Baru" button is located above the table. The user "ichaa" is logged in, as indicated by the "Halo, ichaa" message in the top right corner.

No	Name	Email	Role	Actions
1	M Fikri	admin@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
2	John	john@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
3	ichaa	ichaa@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
4	ichaa	ichaa@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>

- Create New User

Form tambah user baru:

The screenshot shows the "Form tambah user baru" (Create New User) form. The form is open, and the user "ichaa" is logged in, as indicated by the "Halo, ichaa" message in the top right corner. The form fields are: "Name" (Faizza), "Email" (faiza@gmail.com), "Password" (masked with dots), "Confirm Password" (masked with dots), and "Role" (Admin). A "Save" button is at the bottom of the form. The background shows the "Daftar Users" table and the sidebar.

**Form Fields:**

- Name: Faizza
- Email: faiza@gmail.com
- Password: .....
- Confirm Password: .....
- Role: Admin

**Buttons:** Save

User baru berhasil ditambahkan:

Daftar Users

Tambah Users Baru

No	Name	Email	Role	Actions
1	M Fikri	admin@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
2	John	john@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
3	ichaa	ichaa@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
4	ichaa	ichaa@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
5	Faiza	faiza@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>

- Edit Data User

Update data baris ke 5:

Update Petugas

Name: Faiza Faiza Faiza

Email: faizafaiza@gmail.com

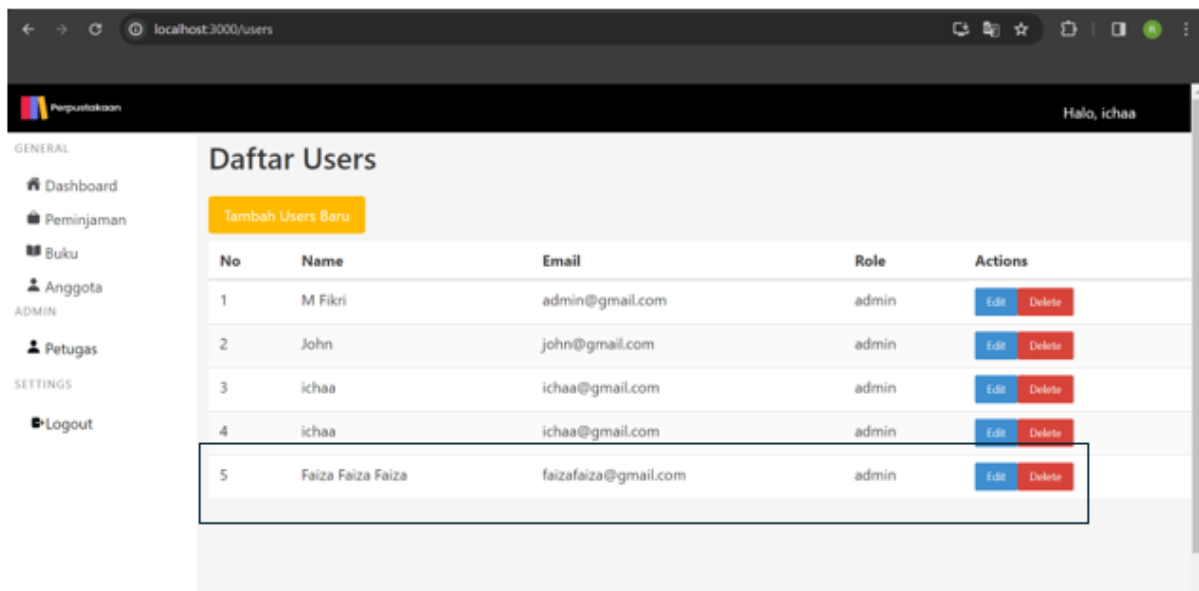
Password: .....

Confirm Password: .....

Role: Admin

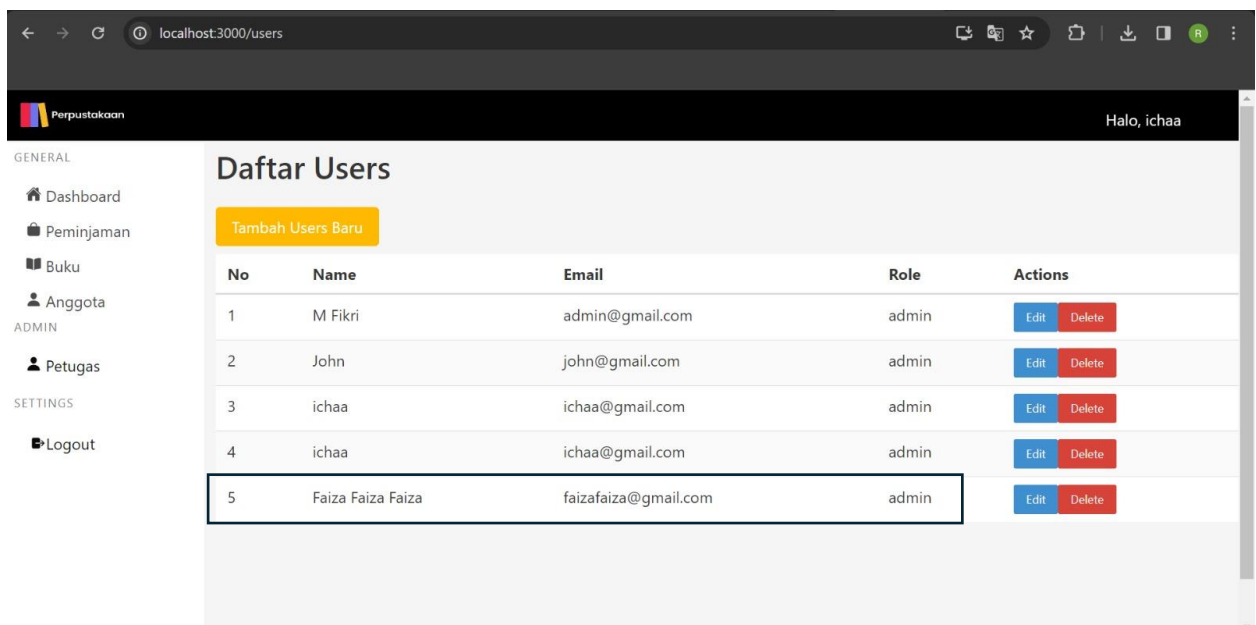
Update

Data baris ke 5 berhasil diupdate:



No	Name	Email	Role	Actions
1	M Fikri	admin@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
2	John	john@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
3	ichaa	ichaa@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
4	ichaa	ichaa@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
5	Faiza Faiza Faiza	faizafaiza@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>

- Delete data users



No	Name	Email	Role	Actions
1	M Fikri	admin@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
2	John	john@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
3	ichaa	ichaa@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
4	ichaa	ichaa@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
5	Faiza Faiza Faiza	faizafaiza@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>

Tampilan data user yang belum didelete, dapat dilihat data dengan id 5 masih berada di daftar user



localhost:3000/users

Perustakaan

Halo, ichaa

GENERAL

- Dashboard
- Peminjaman
- Buku
- Anggota

ADMIN

- Petugas

SETTINGS

- Logout

## Daftar Users

Tambah Users Baru

No	Name	Email	Role	Actions
1	M Fikri	admin@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
2	John	john@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
3	ichaa	ichaa@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>
4	ichaa	ichaa@gmail.com	admin	<a href="#">Edit</a> <a href="#">Delete</a>

Tampilan setelah menghapus, data users no 5 sudah terhapus.

- f) Menu Peminjaman
- Get All Peminjaman

Daftar Peminjaman

Tambah Peminjaman Baru

No	Tanggal Pinjam	Tanggal Kembali	Id Buku	Id Anggota	Status	Actions
1	2024-01-13	2024-01-15	1	1	Sedang dipinjam	<a href="#">Edit</a> <a href="#">Delete</a>
2	2024-01-17	2024-01-24	1	1	Telah dikembalikan	<a href="#">Edit</a> <a href="#">Delete</a>
3	2024-01-26	2024-01-30	1	1	Sedang dipinjam	<a href="#">Edit</a> <a href="#">Delete</a>
4	2024-01-22	2024-01-15	1	3	Telah dikembalikan	<a href="#">Edit</a> <a href="#">Delete</a>
5	2024-01-09	2024-01-15	1	1	Sedang dipinjam	<a href="#">Edit</a> <a href="#">Delete</a>
6	2024-01-23	2024-01-30	1	1	Sedang dipinjam	<a href="#">Edit</a> <a href="#">Delete</a>

- Create Peminjaman

Peminjaman Baru

Tambah Peminjaman Baru

Tanggal Pinjam

Tanggal Kembali

Id Buku

Id Anggota

Save

Setelah berhasil menambahkan data

Perpustakaan

Login as admin

GENERAL

Dashboard

Peminjaman

Buku

Anggota

ADMIN

Petugas

SETTINGS

Logout

Daftar Peminjaman

Tambah Peminjaman Baru

No	Tanggal Pinjam	Tanggal Kembali	Id Buku	Id Anggota	Status	Actions
1	2024-01-13	2024-01-15	1	1	Sedang dipinjam	<a>Edit</a> <a>Delete</a>
2	2024-01-17	2024-01-24	1	1	Telah dikembalikan	<a>Edit</a> <a>Delete</a>
3	2024-01-26	2024-01-30	1	1	Sedang dipinjam	<a>Edit</a> <a>Delete</a>
4	2024-01-22	2024-01-15	1	3	Telah dikembalikan	<a>Edit</a> <a>Delete</a>
5	2024-01-09	2024-01-15	1	1	Sedang dipinjam	<a>Edit</a> <a>Delete</a>
6	2024-01-23	2024-01-30	1	1	Sedang dipinjam	<a>Edit</a> <a>Delete</a>
7	2024-01-16	2024-01-22	19	3	Sedang dipinjam	<a>Edit</a> <a>Delete</a>

- Edit data peminjaman

Perpustakaan

Login as admin

GENERAL

Dashboard

Peminjaman

Buku

Anggota

ADMIN

Petugas

SETTINGS

Logout

Update Peminjaman

Tanggal Pinjam

2024-01-16

Tanggal Kembali

2024-01-19

Id Buku

19

Id Anggota

3

Status

Telah dikembalikan

Update

Perpustakaan

Login as admin

GENERAL

Dashboard

Peminjaman

Buku

Anggota

ADMIN

Petugas

SETTINGS

Logout

Daftar Peminjaman

Tambah Peminjaman Baru

No	Tanggal Pinjam	Tanggal Kembali	Id Buku	Id Anggota	Status	Actions
1	2024-01-13	2024-01-15	1	1	Sedang dipinjam	<a>Edit</a> <a>Delete</a>
2	2024-01-17	2024-01-24	1	1	Telah dikembalikan	<a>Edit</a> <a>Delete</a>
3	2024-01-26	2024-01-30	1	1	Sedang dipinjam	<a>Edit</a> <a>Delete</a>
4	2024-01-22	2024-01-15	1	3	Telah dikembalikan	<a>Edit</a> <a>Delete</a>
5	2024-01-09	2024-01-15	1	1	Sedang dipinjam	<a>Edit</a> <a>Delete</a>
6	2024-01-23	2024-01-30	1	1	Sedang dipinjam	<a>Edit</a> <a>Delete</a>
7	2024-01-16	2024-01-19	19	3	Telah dikembalikan	<a>Edit</a> <a>Delete</a>

Setelah diedit, data pada tanggal kembali dan status telah berhasil diubah

- Delete peminjaman

Perpustakaan

Login as admin

GENERAL

Dashboard

Peminjaman

Buku

Anggota

ADMIN

Petugas

SETTINGS

Logout

Daftar Peminjaman

Tambah Peminjaman Baru

No	Tanggal Pinjam	Tanggal Kembali	Id Buku	Id Anggota	Status	Actions
1	2024-01-13	2024-01-15	1	1	Sedang dipinjam	<div>EditDelete</div>
2	2024-01-17	2024-01-24	1	1	Telah dikembalikan	<div>EditDelete</div>
3	2024-01-26	2024-01-30	1	1	Sedang dipinjam	<div>EditDelete</div>
4	2024-01-22	2024-01-15	1	3	Telah dikembalikan	<div>EditDelete</div>
5	2024-01-09	2024-01-15	1	1	Sedang dipinjam	<div>EditDelete</div>
6	2024-01-23	2024-01-30	1	1	Sedang dipinjam	<div>EditDelete</div>
7	2024-01-16	2024-01-19	19	3	Telah dikembalikan	<div>EditDelete</div>

Tampilan Sebelum dihapus

Perpustakaan

Login as admin

GENERAL

Dashboard

Peminjaman

Buku

Anggota

ADMIN

Petugas

SETTINGS

Logout

Daftar Peminjaman

Tambah Peminjaman Baru

No	Tanggal Pinjam	Tanggal Kembali	Id Buku	Id Anggota	Status	Actions
1	2024-01-13	2024-01-15	1	1	Sedang dipinjam	<div>EditDelete</div>
2	2024-01-17	2024-01-24	1	1	Telah dikembalikan	<div>EditDelete</div>
3	2024-01-26	2024-01-30	1	1	Sedang dipinjam	<div>EditDelete</div>
4	2024-01-22	2024-01-15	1	3	Telah dikembalikan	<div>EditDelete</div>
5	2024-01-09	2024-01-15	1	1	Sedang dipinjam	<div>EditDelete</div>
6	2024-01-16	2024-01-19	19	3	Telah dikembalikan	<div>EditDelete</div>

Tampilan setelah dihapus.

## Hasil Create Skeleton

```
C:\TugasBesar
├── backend/
│   ├── config/
│   ├── controllers/
│   ├── index.js
│   ├── middleware/
│   ├── models/
│   ├── node_modules/
│   ├── package-lock.json
│   ├── package.json
│   ├── request.rest
│   └── routes/
├── frontend/
│   ├── node_modules/
│   ├── package-lock.json
│   ├── package.json
│   ├── public/
│   ├── README.md
│   └── src/
└── out.txt
```

Gambar Hasil Create Skeleton