

LAPORAN PROJECT AKHIR PRAKTIKUM

“ BOOK MANAGEMENT “

Mata kuliah : Web Service Praktik II



Dosen Pengampu :

Aditya Ferdiana Arief, S.Kom., M.Kom

Anggota Kelompok 8 :

Putri Nur Alifah 5210311035

Mufti Shofiur Rohman 5210311052

Reffa Javami Kurnia 5210311080

Novendika Rizki Ramadhan 5210311086

PROGRAM STUDI SISTEM INFORMASI SARJANA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

2024

BAB I

PENDAHULUAN

1.1. Software yang di gunakan

Berikut adalah beberapa software yang di gunakan untuk membuat project “Books Management “:

1.1.1. Visual Studio Code

Visual Studio Code adalah Software yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan built-in dukungan untuk JavaScript, naskah dan Node.js dan memiliki array beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C #, Python, dan PHP. Hal ini didasarkan sekitar Github ini Elektron, yang merupakan versi cross-platform dari Atom komponen kode- editing, berdasarkan JavaScript dan HTML5. Editor ini adalah fitur lengkap lingkungan pengembangan terpadu (IDE) dirancang untuk pengembang yang bekerja dengan teknologi cloud yang terbuka Microsoft.

1.1.2. POSTMAN

Postman adalah aplikasi komputer yang digunakan untuk pengujian API (Application Programming Interface). Cara kerjanya, Postman akan mengirimkan permintaan dari API ke server web dan menerima respons, apa pun itu. Postman menawarkan para developer berbagai fitur dan fungsi yang penting sehingga kinerjanya dapat berlangsung mudah dan sederhana

1.1.3. SQLyog

SQLyog adalah aplikasi client MySQL yang sangat berguna untuk pengelolaan database dan mudah digunakan

1.1.4. XAMPP

XAMPP adalah aplikasi berbasis web server yang bersifat open source dan dapat berjalan pada berbagai sistem operasi, seperti Windows, Linux, dan MacOS. XAMPP digunakan sebagai standalone server atau biasa disebut dengan localhost

1.2. Deskripsi Project

"Books Management" adalah project yang bertujuan untuk mengelola catatan buku di perpustakaan. Ini melibatkan pembuatan aplikasi web yang memungkinkan karyawan melihat semua buku yang tersedia, menambah buku baru, menghapus buku, mengupdate buku. Proyek ini diimplementasikan menggunakan Node.js, Express, JavaScript, dan mySql. Beberapa fitur dari proyek ini yaitu beranda, data pelanggan, dan data pegawai, menu navigasi yang tersedia yaitu setiap halaman terdapat halaman daftar buku, fungsi pencarian buku, tombol untuk menambah buku baru, begitupun pada pelanggan dan pegawai.



1.3. Database

Database yang di gunakan adalah SQL, SQL merupakan sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language).






1.4. Desain Rancangan Database

Database “tugasbesar” memiliki 3 tabel yaitu tabel buku, tabel pelanggan, dan tabel karyawan. setiap tabel memiliki atribut tertentu, berikut desain rancangan database “tugasbesar” pada project “ Books Management” :






- Tabel Buku

  tugasbesar buku
 kode : int(11)
 judul : varchar(255)
 penulis : varchar(255)
 penerbit : varchar(255)
 qty : int(11)

- Tabel Pelanggan

  tugasbesar pelanggan
 id : int(11)
 nama : varchar(255)
 alamat : varchar(255)

- Tabel Pegawai

  tugasbesar pegawai
 nip : int(50)
 nama : varchar(200)
 alamat : varchar(300)

1.5. Desain Endpoint

- Buku

GET: / --> end point untuk mengakses data buku

POST: /buku/save --> end point untuk menambahkan data buku

PATCH: /buku/update --> end point untuk update data buku

DELETE: /buku/:kode --> end point untuk hapus data buku

- Pelanggan

GET: / --> end point untuk mengakses data

POST: / pelanggan/save--> end point untuk menambahkan data pelanggan

PATCH: /pelanggan/update --> end point untuk update data pelanggan

DELETE: /pelanggan/:id --> end point untuk hapus data pelanggan

- Pegawai

GET: /--> end point untuk mengakses data pegawai

POST: /pegawai/save --> end point untuk menambahkan data pegawai

PATCH: /pegawai/update --> end point untuk update data pegawai

DELETE: /pegawai/:id --> end point untuk hapus data pegawai

1.6. Framework dan Packages

1.6.1. Framework

- Node Js : Node.js adalah runtime JavaScript yang memungkinkan pengembang untuk menulis kode server-side menggunakan JavaScript.
- Express.js : Express.js adalah framework web untuk Node.js yang menyediakan fitur-fitur untuk membangun aplikasi web dan API dengan cepat dan mudah
- React.js: React.js adalah library JavaScript yang dikembangkan oleh Facebook untuk membangun antarmuka pengguna

1.6.2. Packages

- Nodemon: otomatis restart terhadap aplikasi yang sedang dijalankan.
- Body-parser: Packages yang digunakan untuk memproses data yang dikirim ke dalam HTTP request-body.
- Mysql: Packages database pada nodejs
- Cors: Packages yang memungkinkan server untuk menunjukkan asal mana saja (domain, skema, atau port) selain miliknya sendiri dari mana browser harus mengizinkan pemuatan sumber daya
- Axios: Axios adalah library JavaScript yang digunakan untuk membuat permintaan HTTP
- Bootstrap: adalah framework CSS yang digunakan untuk membangun antarmuka pengguna responsif

BAB II

LANGKAH LANGKAH

2.1. Langkah langkah membuat Backend

- Membuat project

Membuat direktori proyek menggunakan terminal atau command prompt gitBash seperti berikut :

mkdir ProjectAkhir

- Inisialisasi Proyek Node.js

Masuk ke direktori proyek yang baru dibuat dan inisialisasikan proyek Node.js menggunakan perintah **npm init**. Perintah ini akan membuat file package.json.

cd books-management-backend

npm init -y

- Instalasi Dependensi / Packages

Selanjutnya, instal dependensi yang diperlukan untuk proyek, seperti Express.js, MySQL, dan body-parser. Dependensi ini membantu dalam membangun server web, mengelola koneksi ke database, dan memproses data yang dikirim melalui permintaan HTTP.

npm install express mysql body-parser

- Buat Server Express

Buat file express berupa app.js dan atur server express dengan mengimpor Modul Express, menambahkan middleware body-parser, membuat routes awal, dan menentukan port

```
const express = require("express");
const app = express();
const bodyParser = require("body-parser");
const cors = require("cors");

// Menggunakan Cors untuk mengizinkan akses dari berbagai domain
app.use(cors());
```

```
// Menggunakan bodyParser untuk mengelola data yang masuk
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

// Mengaktifkan server untuk mendengarkan permintaan pada port 2910
app.listen(1928, () => {
  console.log("Server run on port 1928");
});
```

- Koneksikan ke Database MySQL

Untuk dapat mengkoneksikan ke database Mysql buatlah file koneksi dbConection.js dan mengganti nilai host, user, password, dan database sesuai dengan pengaturan database MySQL

```
const mysql = require("mysql");

const db = mysql.createConnection({
  host: "localhost",
  password: "",
  user: "root",
  database: "tugasbesar"
});

db.connect(err => {
  if (err) console.log(err.message);
  else console.log("koneksi berhasil");
});

module.exports = db;
```

- Buat API Endpoints

Tambahkan endpoint untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada data buku, pelanggan, pegawai dalam database

- Atur Middleware dan Pengolahan Permintaan

Gunakan middleware Express.js untuk memproses permintaan. Misalnya, untuk mengizinkan aplikasi untuk mengonsumsi data dalam format JSON

```
// Menggunakan bodyParser untuk mengelola data yang masuk
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
```

- Jalankan Server

Tambahkan kode untuk menjalankan server pada port tertentu

```
/ Mengaktifkan server untuk mendengarkan permintaan pada port
2910
app.listen(1928, () => {
  console.log("Server run on port 1928");
});
```

2.2. Langkah Langkah membuat Frontend

- Buat folder proyek baru untuk frontend

mkdir books-management-frontend

cd books-management-frontend

- Inisialisasi proyek React dengan Create React App atau alat sejenis.

npx create-react-app .

- Instalasi Paket-paket yang Diperlukan

Instalasi paket-paket yang diperlukan untuk proyek "books management" melibatkan penggunaan Node Package Manager (npm). Dalam proyek ini paket yang diperlukan yaitu Axios untuk melakukan permintaan HTTP ke backend, Bootstrap untuk desain antarmuka yang responsif. instalasi paket di lakukan pada terminal atau command prompt (gunakan gitbash)

npm install axios bootstrap

- Struktur Project

Membuat folder src , setelah membuat folder src tambahkan di dalamnya folder **Component**, file **App.js** atau **App/index.js**, file **Index.js**

- Menambahkan Component/Page
- Menambahkan Index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
```



```
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import {BrowserRouter} from 'react-router-dom';

ReactDOM.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>, document.getElementById('root'));

reportWebVitals();
```

- Menambahkan App.js

```
import React from 'react';
import Main from './pages/Main';
import { Link, NavLink } from 'react-router-dom';
import { Container, Nav, Navbar } from 'react-bootstrap';

class App extends React.Component {
  render() {
    return (
      <div>
        { /* Navbar */ }
        <nav className="navbar navbar-expand-lg navbar-light bg-white py-0" bg="dark" variant="dark">
          <div className="container-fluid">
            { /* Navbar Brand */ }
            <NavLink className="navbar-brand fs-4" to="/">
              { /* Logo */ }
              
            </NavLink>

            { /* Navbar Toggler */ }
            <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
              <span className="navbar-toggler-icon"></span>
            </button>
          </div>
        </nav>
      </div>
    );
  }
}
```

```

        {/* Navbar Links */}
        <div className="collapse navbar-collapse"
id="navbarSupportedContent">
            <ul className="navbar-nav mx-auto mb-2 mb-lg-0">
                {/* Beranda */}
                <li className="nav-item">
                    <NavLink className="nav-link"
to="/">Beranda</NavLink>
                </li>

                {/* Galeri Buku */}
                <li className="nav-item">
                    <NavLink className="nav-link" to="/Buku">Galeri
Buku</NavLink>
                </li>

                {/* Pelanggan */}
                <li className="nav-item">
                    <NavLink className="nav-link"
to="/pelanggan">Pelanggan</NavLink>
                </li>

                {/* Karyawan */}
                <li className="nav-item">
                    <NavLink className="nav-link"
to="/pegawai">Karyawan</NavLink>
                </li>
            </ul>
        </div>
    </div>
</nav>

    {/* Main Component */}
    <p><Main /></p>
</div>
);
}
}

export default App;

```

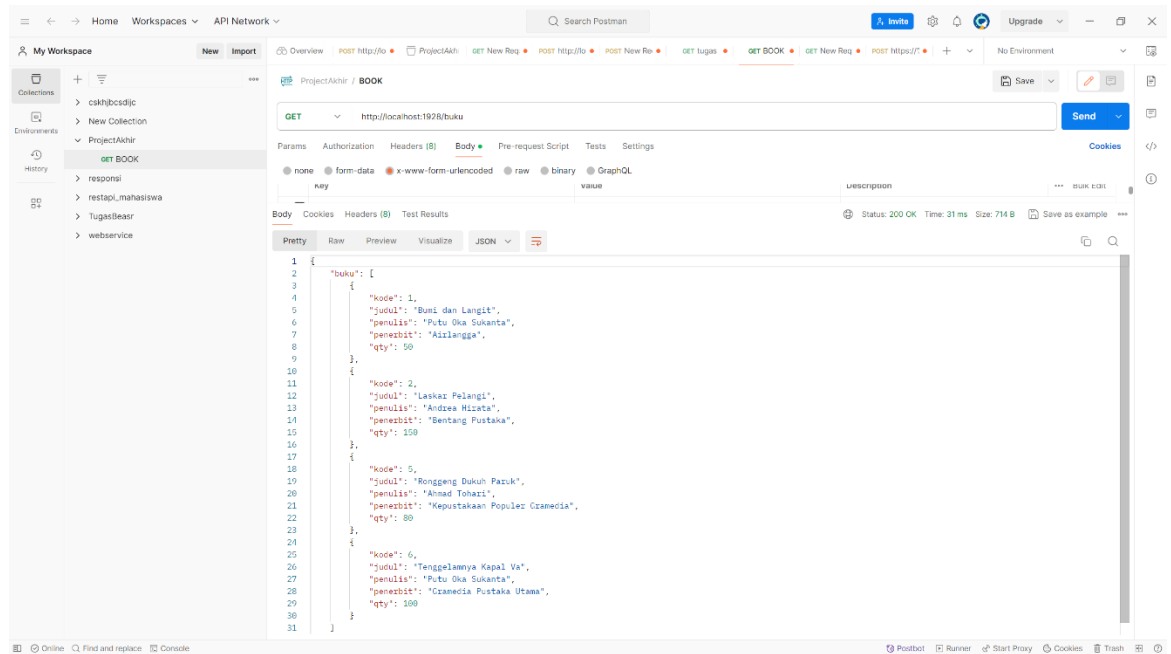
BAB III

IMPLEMENTASI

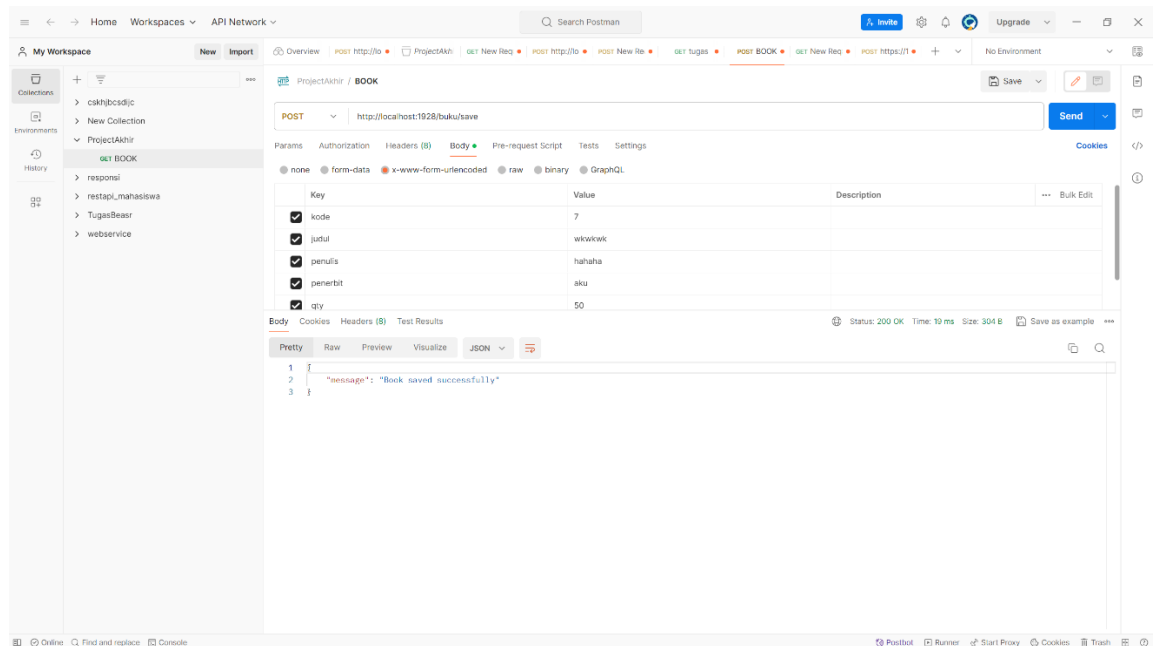
3.1. Test Backend Restful API dengan Postman

3.1.1. Buku

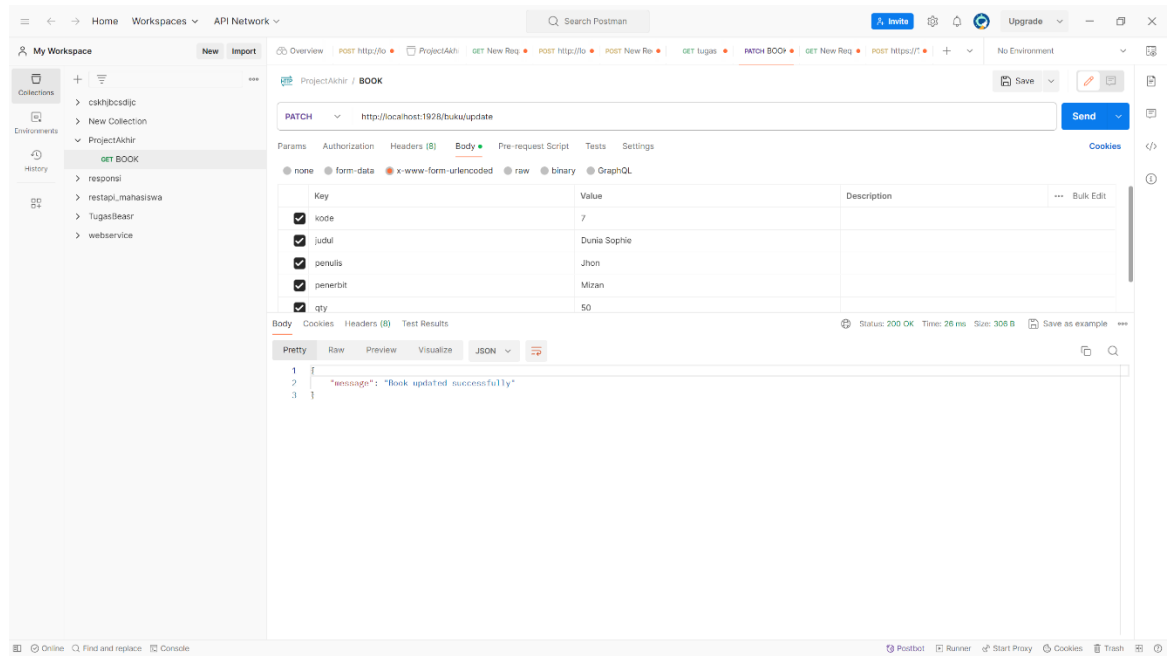
- Get



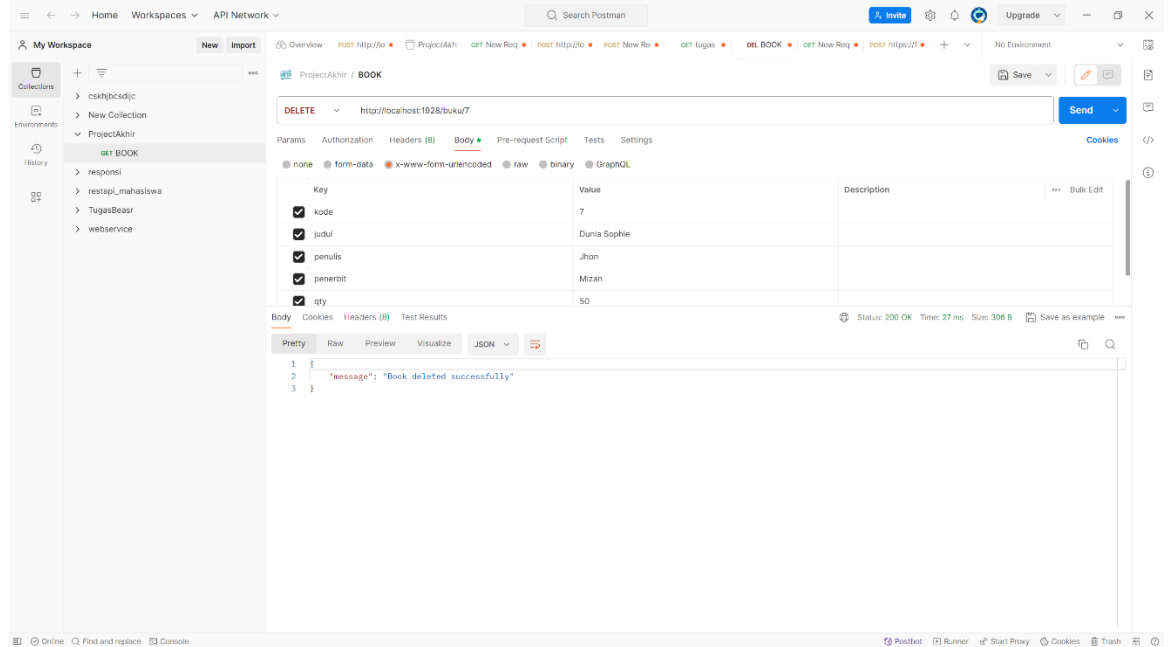
- Post



- Patch

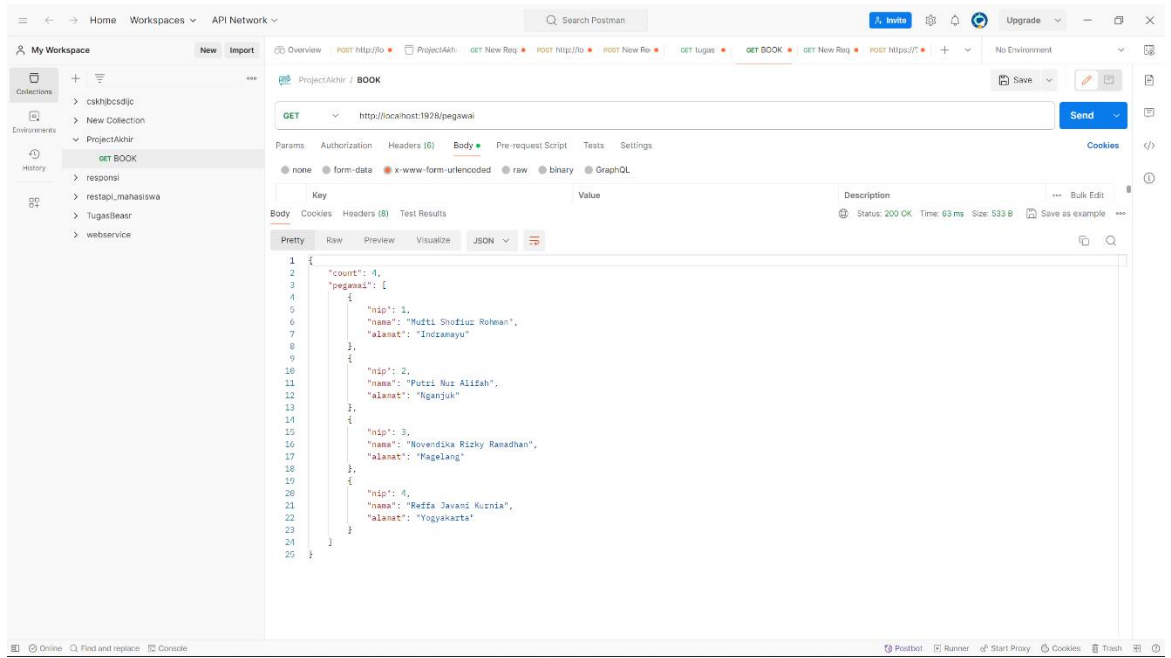


- Delete

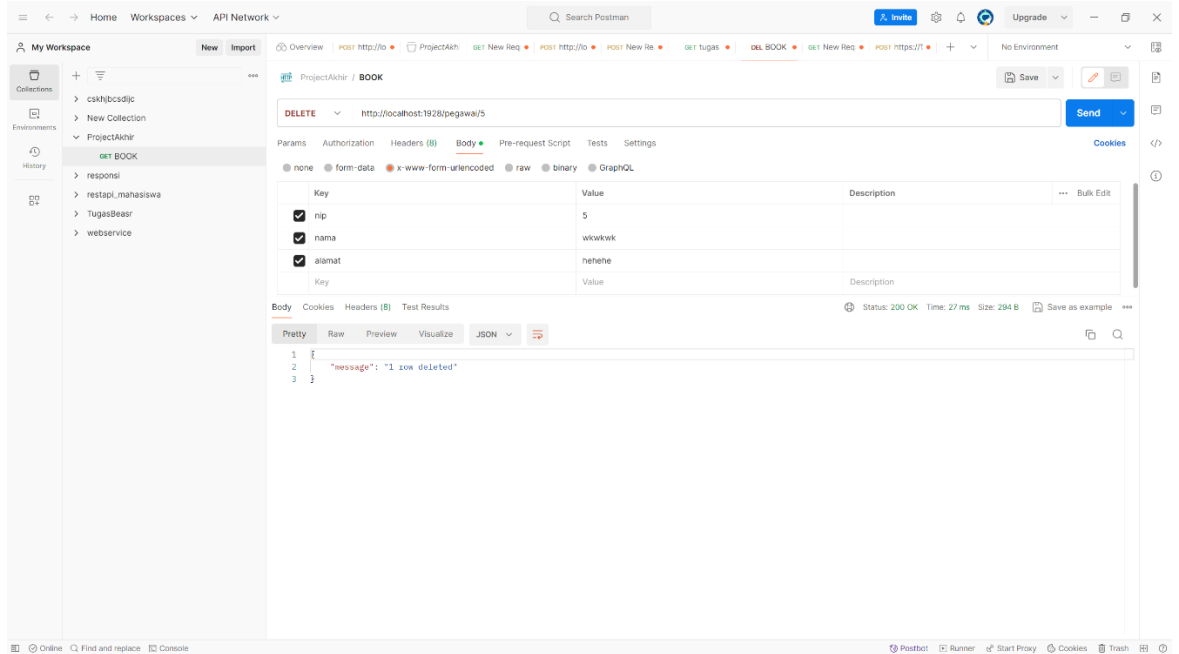


3.1.2. Pegawai

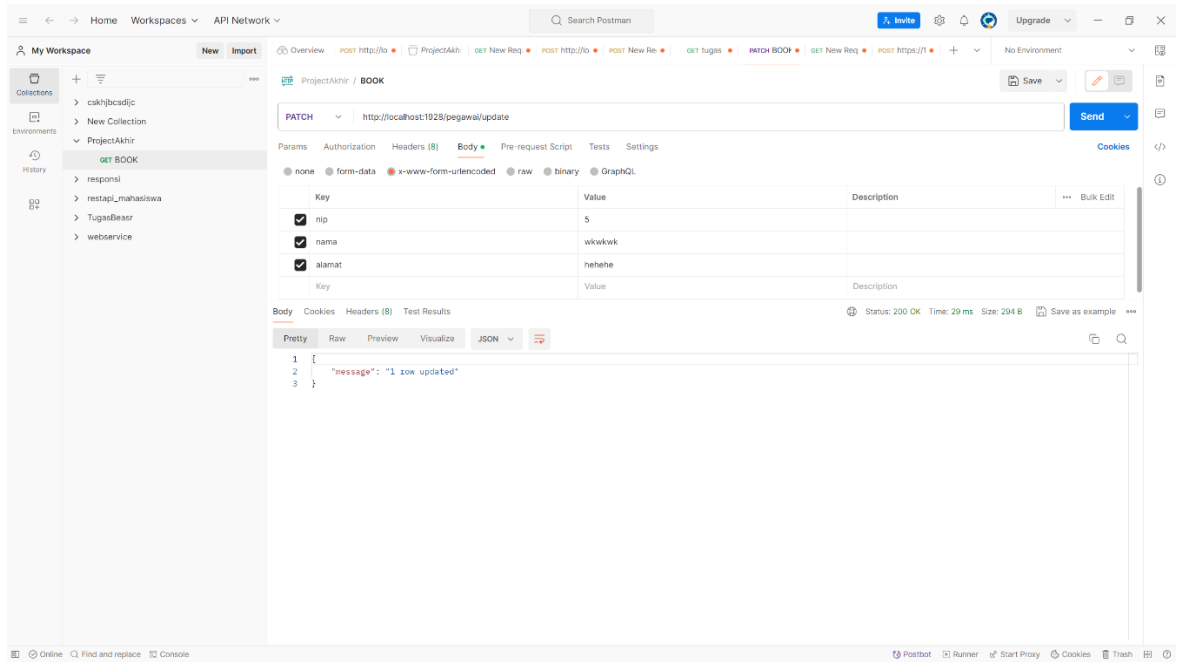
- Get



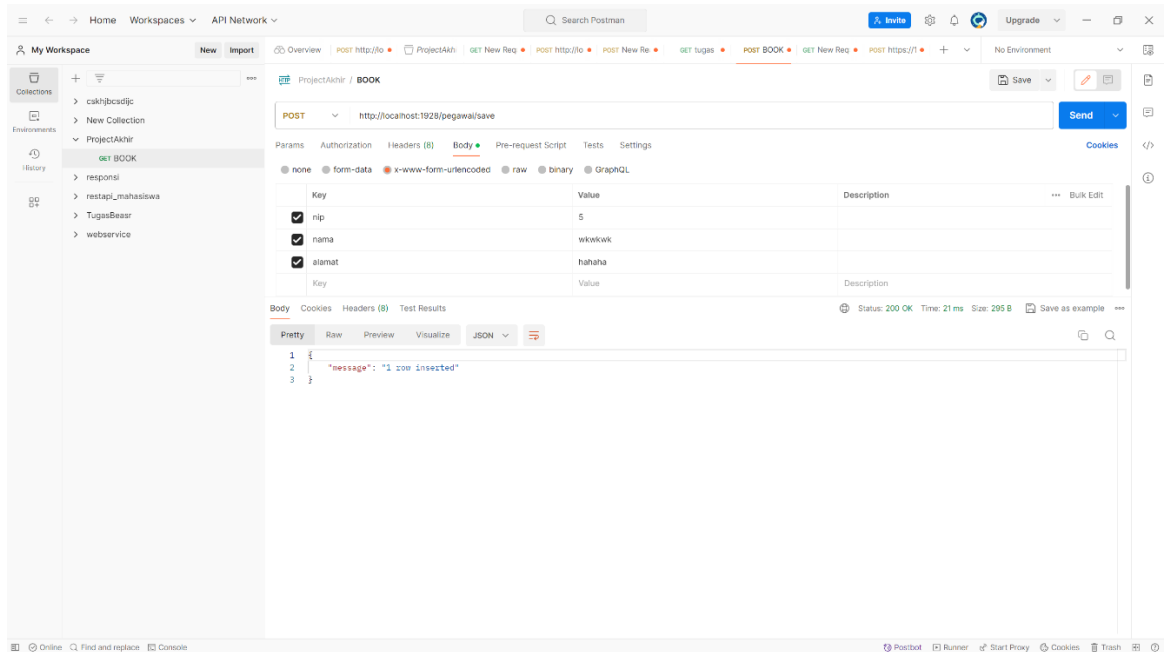
- Post



- Patch

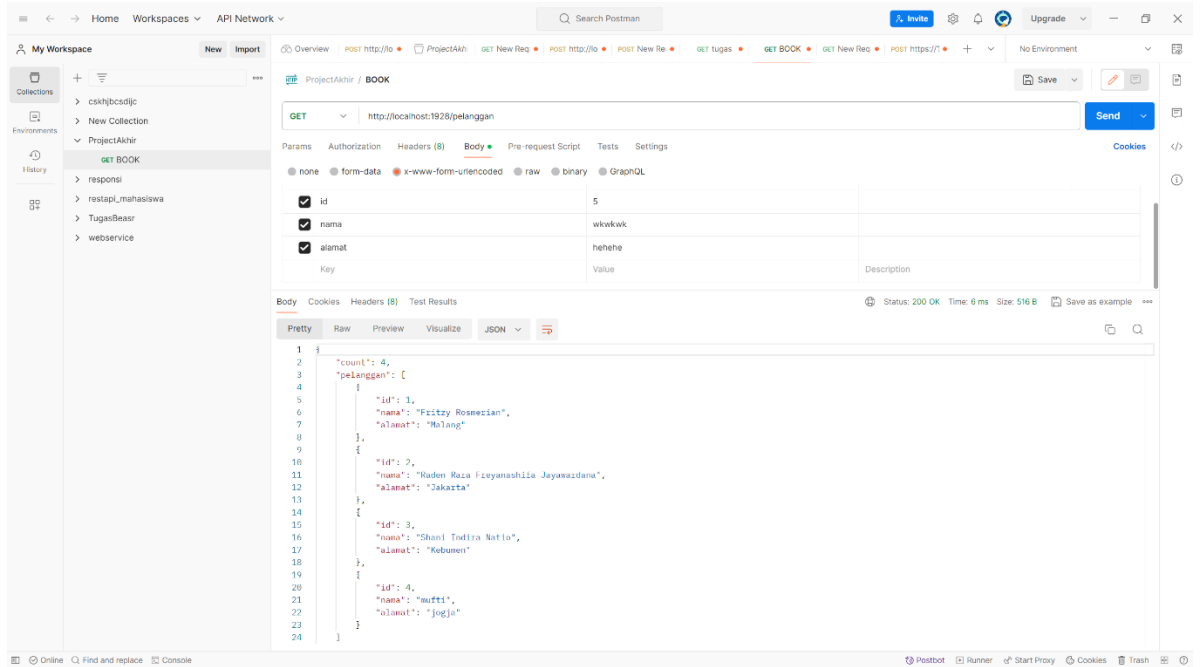


- Delete

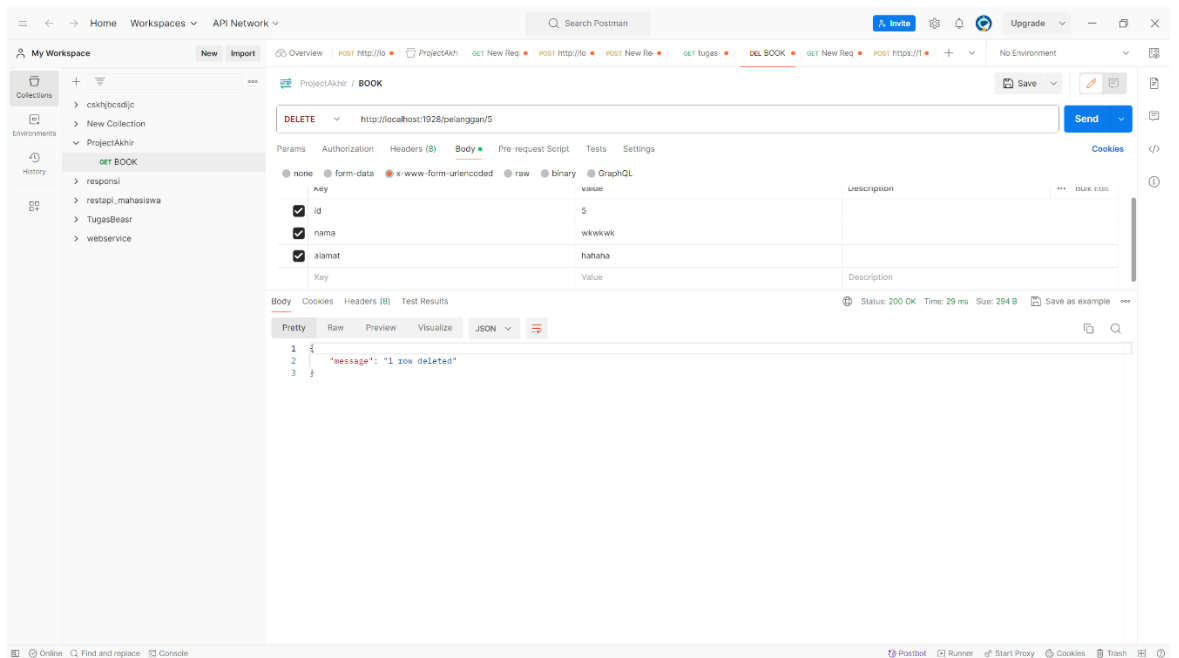


3.1.3. Pelanggan

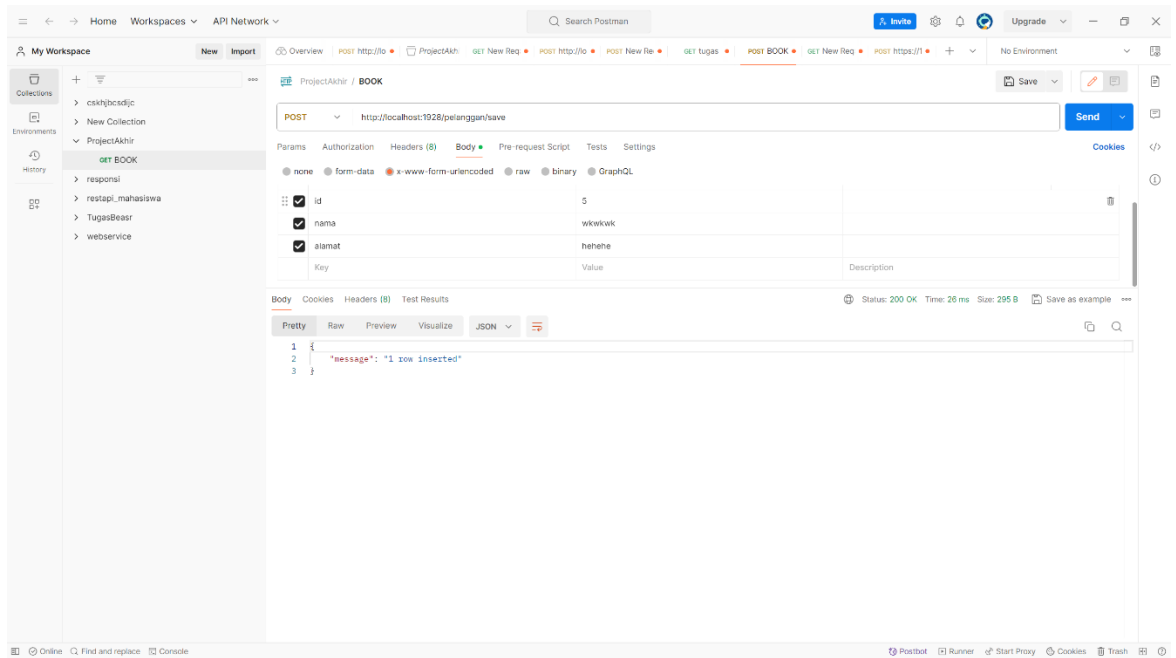
- Get



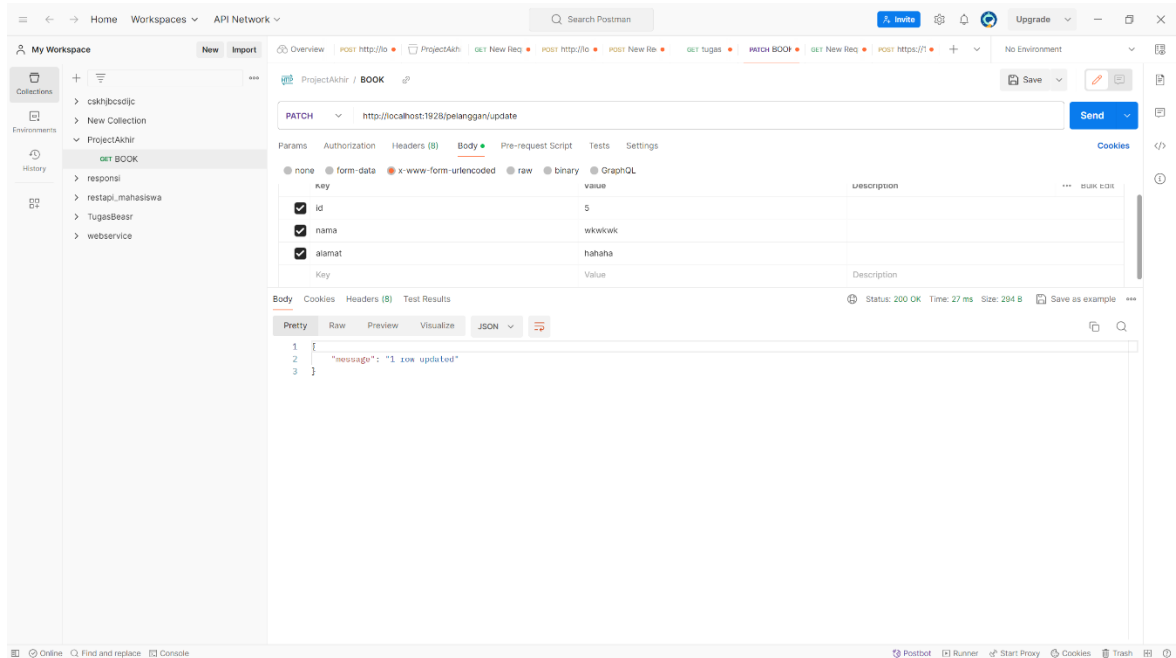
- Post



- Patch

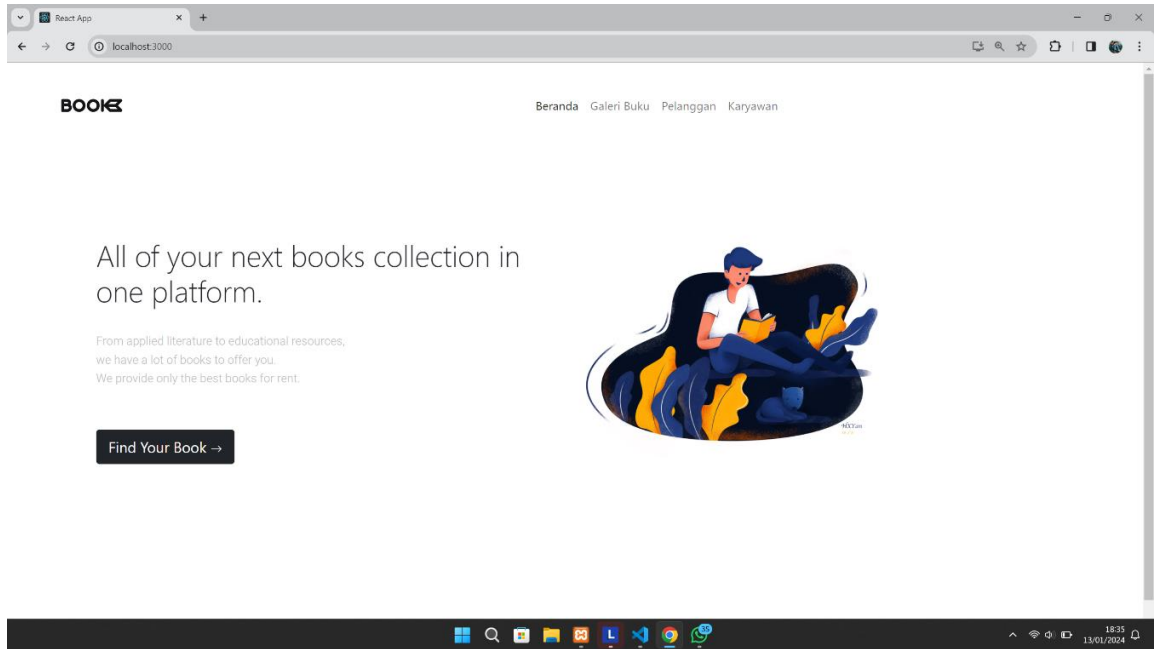


- Delete

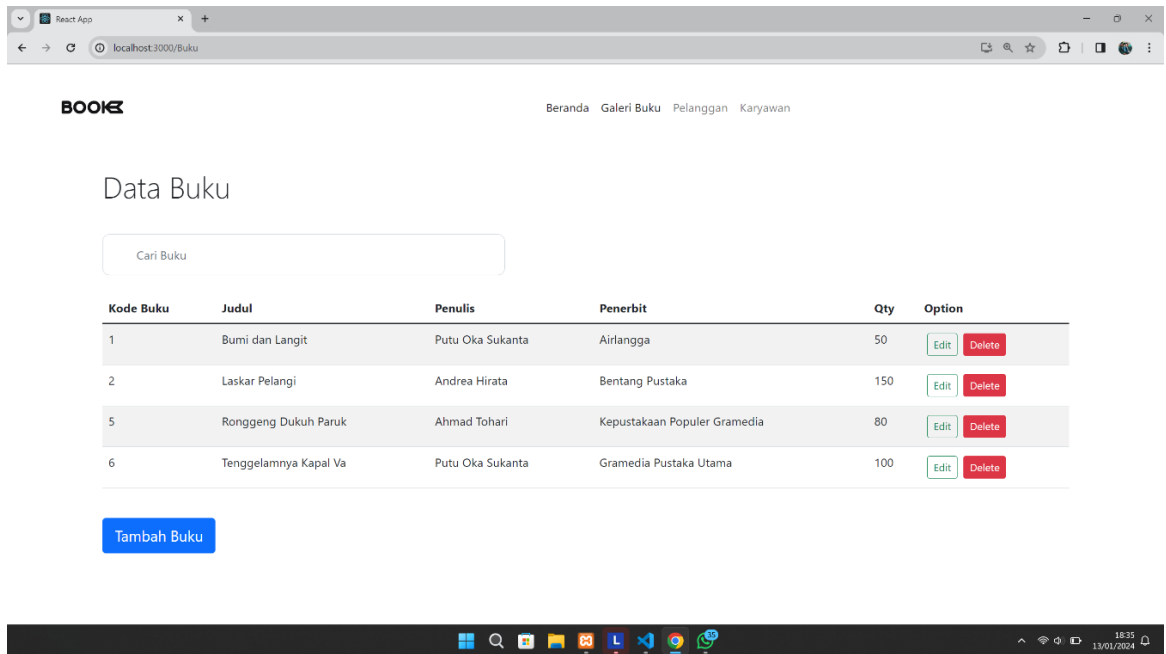


3.2. Test Frontend

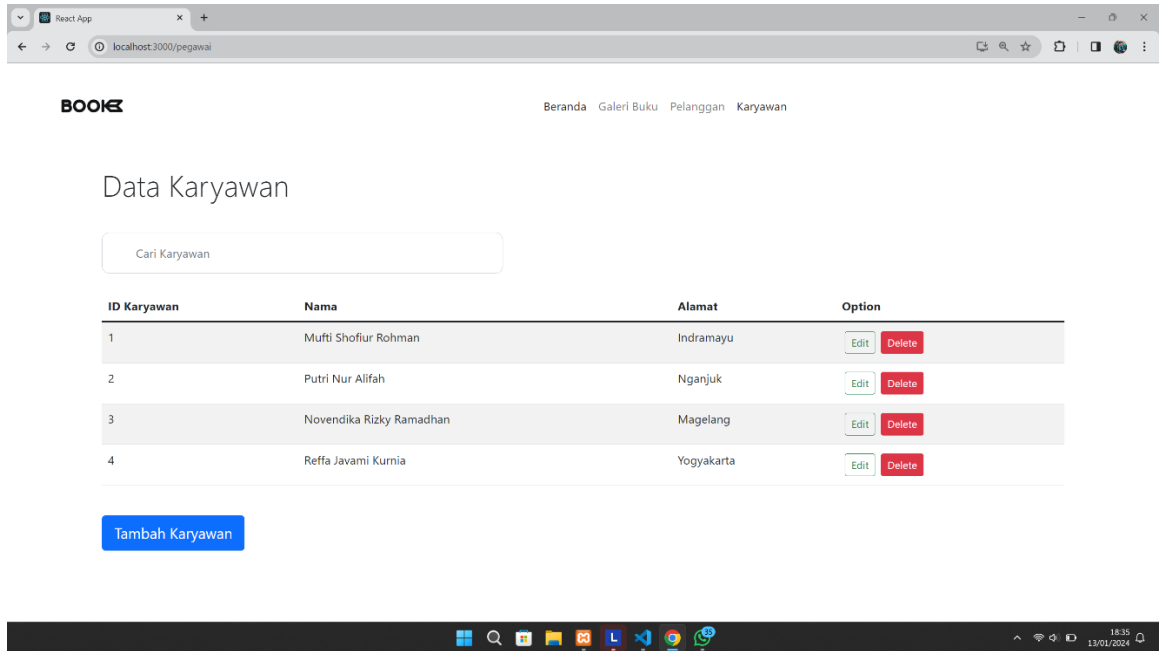
- Halaman Beranda



- Halaman Data Buku



- Halaman Data Karyawan



- Halaman Data Pelanggan

