

**PROJECT AKHIR PRAKTIKUM WEB SERVICE**  
**SISTEM INFORMASI PERPUSTAKAAN DI UNIVERSITAS**  
**SIPERPUS**



Anggota kelompok 2 :

1. Hadiyatul Jannah (5210311014)
2. Nur Alfiyani Rakhmawati (5210311017)
3. Aulia Safira Putri (5210311021)
4. Farahdiba Amalia Almanda (5210311036)

Dosen Pengampu :

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

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

**2024**

# **BAB 1**

## **SIPERPUS**

Persiapan perangkat lunak (software) untuk proyek pengembangan perangkat lunak melibatkan pemilihan alat yang tepat untuk membantu memfasilitasi setiap tahap pengembangan. Berikut adalah beberapa persiapan software beserta beberapa software yang digunakan:

### **1. VSCode**

Visual Studio Code (VSCode) adalah sebuah editor kode sumber yang sangat dihargai oleh komunitas pengembang. Dikembangkan oleh Microsoft, VSCode menonjol karena ringan, cepat, dan sangat ekstensibel. Kelebihan utamanya terletak pada kemampuan untuk mengakomodasi berbagai bahasa pemrograman dan lingkungan pengembangan melalui sistem ekstensibilitas yang kuat. Dengan integrasi Git yang solid, pengelolaan proyek menjadi lebih efisien, sementara debugger terintegrasi membantu dalam penemuan dan perbaikan bug dengan mudah.

VSCode juga dikenal karena dukungan web development yang hebat, dengan fitur-fitur seperti penyelesaian otomatis, navigasi kode, dan analisis yang mendukung pengembangan HTML, CSS, dan JavaScript. Selain itu, editor ini menawarkan antarmuka pengguna yang intuitif untuk mengelola ekstensi dan tema, memberikan fleksibilitas dalam menyesuaikan lingkungan pengembangan sesuai kebutuhan. Dengan komunitas pengembang yang aktif, pengguna dapat dengan mudah mengakses sumber daya dan dukungan, menjadikan VSCode pilihan yang andal untuk proyek pengembangan perangkat lunak di berbagai platform, termasuk Windows, macOS, dan Linux. Sebagai produk Microsoft, Visual Studio Code terus menerima pembaruan dan perkembangan, memastikan bahwa editor ini tetap relevan dan responsif terhadap evolusi kebutuhan pengembangan perangkat lunak.

### **2. Xampp**

XAMPP adalah paket perangkat lunak yang merangkum beberapa komponen penting untuk mendukung pengembangan dan pengujian aplikasi web secara lokal. Dalam sebuah paragraf, kita dapat menjelaskan bahwa XAMPP merupakan singkatan dari "X" yang mengacu pada sistem operasi apapun (Windows, Linux, macOS), "Apache" sebagai server web, "MySQL" sebagai sistem manajemen basis data, "PHP"

sebagai bahasa pemrograman server, dan "Perl" untuk pemrosesan data dan skrip. XAMPP menyediakan lingkungan pengembangan yang lengkap dan siap pakai, memungkinkan pengembang untuk membuat, menguji, dan menyelaraskan aplikasi web tanpa perlu koneksi internet dan langsung di mesin lokal.

Dengan menggunakan XAMPP, pengguna dapat menciptakan server web lokal untuk meng-host dan menguji situs web atau aplikasi yang sedang dikembangkan. Kelebihan XAMPP termasuk kemudahan instalasi, konfigurasi yang relatif sederhana, dan dukungan untuk beberapa bahasa pemrograman. Hal ini membuatnya menjadi pilihan populer di kalangan pengembang web yang ingin membuat dan menguji proyek-proyek secara lokal sebelum di-deploy ke server produksi.

### **3. SQLyog**

SQL (Structured Query Language) adalah bahasa pemrograman khusus yang dirancang untuk mengelola dan mengakses basis data relasional. Dikembangkan pada tahun 1970-an, SQL telah menjadi standar de facto dalam interaksi dengan sistem manajemen basis data (DBMS). Fungsi utama SQL adalah untuk mengizinkan pengguna untuk mengambil, menyisipkan, mengupdate, dan menghapus data dari basis data dengan mudah dan efisien. Melalui perintah-perintah seperti SELECT, INSERT, UPDATE, dan DELETE, SQL memungkinkan para pengembang dan administrator basis data untuk menjalankan operasi yang kompleks, seperti pencarian data, perubahan struktur tabel, dan pengelolaan integritas data.

SQL memiliki struktur sintaksis yang jelas dan dapat digunakan untuk berbagai tugas, mulai dari pengelolaan basis data sederhana hingga pengolahan data yang kompleks. Selain itu, SQL juga mendukung operasi penggabungan tabel, pembuatan indeks, dan pengaturan hak akses, memberikan fleksibilitas dalam merancang dan memelihara struktur basis data. Kemampuannya yang luas dan universal membuat SQL menjadi kebutuhan esensial dalam pengembangan perangkat lunak yang melibatkan penggunaan basis data relasional.

### **4. Postman**

Postman merupakan sebuah platform yang sangat berharga dalam pengembangan dan pengujian API. Dengan antarmuka yang intuitif dan kaya fitur, Postman memberikan alat yang sangat efektif bagi para pengembang untuk merancang, menguji, dan mendokumentasikan API. Fitur utama termasuk

kemampuan untuk mengirimkan permintaan HTTP dengan mudah, memeriksa tanggapan, dan mengelola lingkungan pengembangan yang berbeda. Postman juga menyediakan kemampuan untuk mengotomatiskan proses pengujian API melalui skrip JavaScript, memungkinkan pengguna untuk membuat rangkaian pengujian yang kompleks. Selain itu, kemampuan untuk menyimpan dan berbagi koleksi permintaan API memfasilitasi kolaborasi di antara tim pengembangan. Dengan dukungan yang kuat dari komunitas pengguna, Postman telah menjadi standar industri dalam pengembangan dan pengujian API, mempercepat dan menyederhanakan siklus pengembangan perangkat lunak yang melibatkan interaksi dengan API.

## **5. Gitbash**

Git Bash adalah sebuah shell yang menyediakan lingkungan untuk menjalankan perintah Git dan perintah Unix di sistem operasi Windows. Dibangun di atas emulator terminal MinGW, Git Bash memungkinkan pengguna Windows untuk mengakses sejumlah besar perintah Git dan utilitas Unix yang sering digunakan dalam pengembangan perangkat lunak. Dengan antarmuka yang mirip dengan terminal Unix, Git Bash menyediakan pengalaman konsol yang familiar bagi pengembang yang terbiasa dengan lingkungan Unix atau Linux.

Salah satu fitur utama Git Bash adalah kemampuannya untuk memfasilitasi penggunaan Git dalam pengembangan perangkat lunak. Dengan Git Bash, pengguna dapat menjalankan perintah-perintah Git seperti `git clone`, `git commit`, dan `git push` secara langsung dari shell, menyederhanakan alur kerja pengelolaan versi proyek. Selain itu, Git Bash juga menyediakan dukungan untuk scripting menggunakan perintah-perintah shell, memberikan fleksibilitas tambahan untuk otomatisasi tugas-tugas pengembangan.

Kelebihan lain Git Bash adalah bahwa itu dapat diintegrasikan dengan baik dengan lingkungan pengembangan terpadu (IDE) atau editor kode sumber yang digunakan oleh pengembang. Dengan demikian, pengguna dapat mengintegrasikan Git Bash ke dalam alur kerja mereka tanpa harus meninggalkan lingkungan pengembangan yang mereka sukai. Dengan popularitas Git sebagai sistem kontrol versi yang luas digunakan, Git Bash menjadi pilihan populer di kalangan pengembang Windows yang mencari solusi konsol yang kuat dan mudah digunakan untuk berinteraksi dengan Git dan perintah Unix.

## 6. Chrome

Visual Studio Code (VSCode) adalah sebuah editor kode sumber yang sangat dihargai oleh komunitas pengembang. Dikembangkan oleh Microsoft, VSCode menonjol karena ringan, cepat, dan sangat ekstensibel. Kelebihan utamanya terletak pada kemampuan untuk mengakomodasi berbagai bahasa pemrograman dan lingkungan pengembangan melalui sistem ekstensibilitas yang kuat. Dengan integrasi Git yang solid, pengelolaan proyek menjadi lebih efisien, sementara debugger terintegrasi membantu dalam penemuan dan perbaikan bug dengan mudah.

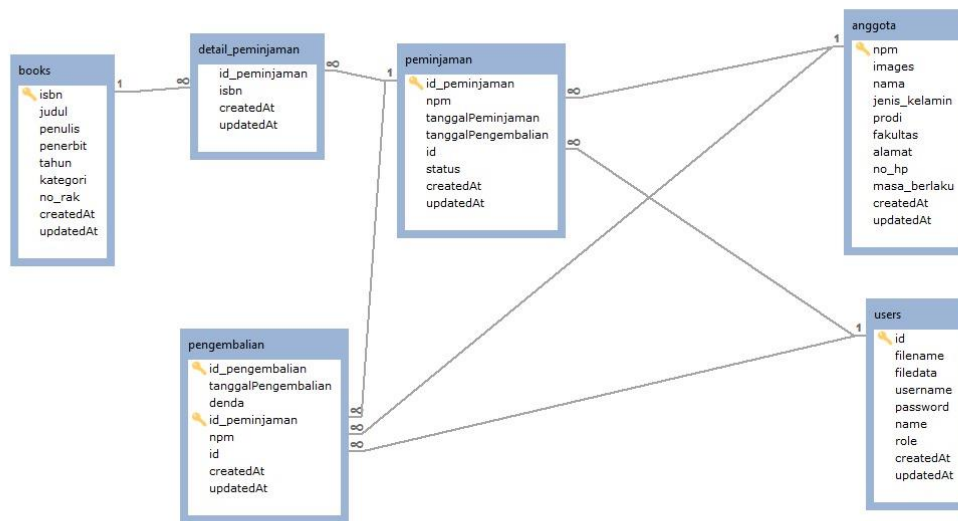
VSCode juga dikenal karena dukungan web development yang hebat, dengan fitur-fitur seperti penyelesaian otomatis, navigasi kode, dan analisis yang mendukung pengembangan HTML, CSS, dan JavaScript. Selain itu, editor ini menawarkan antarmuka pengguna yang intuitif untuk mengelola ekstensi dan tema, memberikan fleksibilitas dalam menyesuaikan lingkungan pengembangan sesuai kebutuhan. Dengan komunitas pengembang yang aktif, pengguna dapat dengan mudah mengakses sumber daya dan dukungan, menjadikan VSCode pilihan yang andal untuk proyek pengembangan perangkat lunak di berbagai platform, termasuk Windows, macOS, dan Linux. Sebagai produk Microsoft, Visual Studio Code terus menerima pembaruan dan perkembangan, memastikan bahwa editor ini tetap relevan dan responsif terhadap evolusi kebutuhan pengembangan perangkat lunak.

Pemilihan topik "SISTEM INFORMASI PERPUSTAKAAN DI UNIVERSITAS" dipandu oleh pertimbangan yang mendalam terkait dengan kebutuhan dan tantangan konkret yang dihadapi oleh lingkungan pendidikan tinggi. Perpustakaan merupakan inti dari ekosistem akademis, menyediakan akses ke pengetahuan dan sumber daya yang mendukung proses pembelajaran, penelitian, dan pengembangan ilmu pengetahuan. Pertama-tama, pengelolaan perpustakaan di universitas seringkali melibatkan volume yang besar dan beragam dari koleksi buku, jurnal, dan sumber daya lainnya. Dengan adanya Sistem Informasi Perpustakaan, kami bertujuan untuk meningkatkan efisiensi dalam manajemen koleksi ini. Sistem ini akan memberikan solusi terpadu untuk pemeliharaan, pencarian, dan pemanfaatan sumber daya, memastikan bahwa mahasiswa, dosen, dan peneliti dapat dengan mudah

mengakses informasi yang mereka butuhkan. Selain itu, perpustakaan yang dilengkapi dengan teknologi informasi terkini dapat memberikan kemudahan akses melalui pencarian online, peminjaman, dan pengembalian buku yang lebih efisien. Dengan demikian, Sistem Informasi Perpustakaan diharapkan dapat memberikan pengalaman pengguna yang lebih baik, meningkatkan kepuasan pengguna, dan mempercepat proses-proses administratif yang terkait. Pemilihan topik ini juga mencerminkan arah evolusi perpustakaan menuju model digital. Integrasi teknologi seperti basis data terpusat, analisis data, dan otomatisasi dapat membantu perpustakaan untuk tetap relevan di era digital, menjaga dan meningkatkan peran strategisnya dalam mendukung pembelajaran dan penelitian di lingkungan akademis.

Dalam keseluruhan, pemilihan topik ini diarahkan untuk menciptakan sebuah ekosistem informasi yang lebih efisien, responsif, dan terkoneksi di universitas, mendukung misi pendidikan tinggi dalam memberikan layanan terbaik bagi komunitas akademisnya dan mempersiapkan mereka untuk tuntutan masa depan yang semakin digital.

Menggunakan SQL digunakan untuk membuat data menjadi terstruktur melalui model data relasional. Dengan memanfaatkan tabel dan relasi antar tabel, SQL membantu mengorganisir data dengan jelas dan terstruktur. Keterkaitan antar entitas dijaga melalui konsep kunci asing, yang menghasilkan hubungan yang terstruktur dalam basis data. Normalisasi, yang didukung oleh SQL, membantu mengurangi redundansi data dan memastikan efisiensi dan integritas basis data. SQL memberikan perintah-perintah seperti SELECT, JOIN, dan GROUP BY untuk melakukan query dan analisis data dengan efisien, menghasilkan informasi yang terstruktur. Konsep integritas referensial dan kunci utama yang diterapkan oleh SQL membantu menjaga konsistensi dan integritas data. Prinsip ACID dalam SQL memastikan bahwa setiap transaksi, baik berhasil maupun gagal, menjaga keberlanjutan data dan konsistensi dalam basis data. Dengan skalabilitas dan performa yang baik, SQL memberikan fondasi yang kuat untuk manajemen data yang efisien dan keterkaitan yang jelas, penting dalam proyek sistem informasi perpustakaan yang membutuhkan data terstruktur dan terorganisir.



Dalam diagram ERD ini, terdapat enam entitas, yaitu:

- Books
- Detail Peminjaman
- Detail Pengembalian
- Peminjaman
- Pengembalian
- Anggota

Entitas Books mewakili buku-buku yang dimiliki oleh perpustakaan. Entitas Detail Peminjaman mewakili detail peminjaman buku, seperti nomor ID peminjaman, nomor ID buku, tanggal peminjaman, dan tanggal pengembalian. Entitas Detail Pengembalian mewakili detail pengembalian buku, seperti nomor ID pengembalian, nomor ID peminjaman, tanggal pengembalian, dan denda. Entitas Peminjaman mewakili peminjaman buku, seperti nomor ID peminjaman, nomor ID anggota, dan nomor ID buku. Entitas Pengembalian mewakili pengembalian buku, seperti nomor ID pengembalian, nomor ID peminjaman, dan nomor ID anggota. Entitas Anggota mewakili anggota perpustakaan, seperti nomor ID anggota, nomor induk mahasiswa, nama, fakultas, dan program studi.

Relasi yang menghubungkan entitas-entitas tersebut adalah sebagai berikut:

- **\*\*Relasi Books - Detail Peminjaman.** Relasi ini menghubungkan entitas Books dengan entitas Detail Peminjaman. Relasi ini menggunakan kunci primer id pada entitas Books sebagai kunci asing pada entitas Detail Peminjaman. Relasi ini menunjukkan bahwa setiap buku dapat dipinjam oleh beberapa anggota.
- **\*\*Relasi Detail Peminjaman - Peminjaman.** Relasi ini menghubungkan entitas Detail Peminjaman dengan entitas Peminjaman. Relasi ini menggunakan kunci primer id\_peminjaman pada entitas Detail Peminjaman sebagai kunci asing pada entitas Peminjaman. Relasi ini menunjukkan bahwa setiap peminjaman memiliki beberapa detail peminjaman.
- **\*\*Relasi Peminjaman - Anggota.** Relasi ini menghubungkan entitas Peminjaman dengan entitas Anggota. Relasi ini menggunakan kunci primer id\_anggota pada entitas Anggota sebagai kunci asing pada entitas Peminjaman. Relasi ini menunjukkan bahwa setiap peminjaman dilakukan oleh satu anggota.
- **\*\*Relasi Peminjaman - Pengembalian.** Relasi ini menghubungkan entitas Peminjaman dengan entitas Pengembalian. Relasi ini menggunakan kunci primer id\_peminjaman pada entitas Peminjaman sebagai kunci asing pada entitas Pengembalian. Relasi ini menunjukkan bahwa setiap peminjaman memiliki satu pengembalian.

Endpoint adalah titik akhir atau URL tertentu yang digunakan untuk berkomunikasi dengan suatu aplikasi atau layanan melalui protokol HTTP. Setiap endpoint memiliki beberapa metode yang menentukan jenis operasi atau aksi yang ingin dilakukan pada sumber daya yang terkait. Berikut adalah penjelasan singkat untuk empat metode umum: GET, POST, PATCH, dan DELETE.

#### 1. GET:

Metode GET digunakan untuk mengambil atau membaca data dari sumber daya yang ditentukan oleh URL. Ini adalah metode yang bersifat idempoten, artinya tidak mengubah status sumber daya dan dapat dipanggil berkali-kali tanpa efek samping. Contohnya, endpoint GET dapat digunakan untuk mengambil informasi tentang buku tertentu dalam sistem perpustakaan.



```

router.get("/buku", async (req, res) => {
  const { page = 1, limit = 10, order = "judul" } = req.query;

  try {
    const offset = (page - 1) * limit;

    // Mengambil data buku dari tabel 'Books' dengan paginasi dan pengurutan
    const buku = await Book.findAll({
      offset,
      limit: parseInt(limit),
      order: [[order, "ASC"]],
    });

    if (!buku || buku.length === 0) {
      return res.status(404).json({
        success: false,
        error: "No books found",
      });
    }

    res.json({
      success: true,
      data: buku,
    });
  } catch (error) {
    console.error("Error:", error.message);
    res.status(500).json({
      success: false,
      error: "Internal Server Error",
    });
  }
});

```

## 2. POST:

Metode POST digunakan untuk membuat sumber daya baru atau mengirimkan data ke sumber daya tertentu. Ini biasanya digunakan ketika ingin menambahkan entitas baru ke dalam sistem. Misalnya, endpoint POST dapat digunakan untuk menambahkan buku baru ke dalam perpustakaan dengan mengirimkan data seperti judul, pengarang, dan nomor ISBN.

```

router.post("/tambahBuku", async (req, res) => {
  // Mendapatkan data buku dari body request
  const { isbn, judul, penulis, penerbit, tahun, kategori, no_rak } = req.body;
  // Menambahkan buku ke dalam tabel 'Books'
  try {
    const bukuBaru = await Book.create({
      isbn,
      judul,
      penulis,
      penerbit,
      tahun,
      kategori,
      no_rak
    });

    res.json({
      success: true,
      data: bukuBaru,
    });
  } catch (error) {
    console.error("Error:", error.message);

    // Menangani kesalahan validasi Sequelize
    if (error.name === 'SequelizeValidationError') {
      console.error("Validation Errors:", error.errors);
      res.status(400).json({
        success: false,
        error: "Validation Error",
        details: error.errors.map(err => ({
          field: err.path,
          message: err.message,
        })),
      });
    }
  }
});

```

```

    res.json({
      success: true,
      data: bukuBaru,
    });
  } catch (error) {
    console.error("Error:", error.message);

    // Menangani kesalahan validasi Sequelize
    if (error.name === 'SequelizeValidationError') {
      console.error("Validation Errors:", error.errors);
      res.status(400).json({
        success: false,
        error: "Validation Error",
        details: error.errors.map(err => ({
          field: err.path,
          message: err.message,
        })),
      });
    } else {
      res.status(500).json({
        success: false,
        error: "Internal Server Error",
      });
    }
  }
}

```

### 3. PATCH:

Metode PATCH digunakan untuk memperbarui sebagian dari sumber daya yang ada. Dengan kata lain, PATCH digunakan untuk melakukan pembaruan parsial terhadap suatu sumber daya tanpa menggantinya sepenuhnya. Misalnya, endpoint PATCH dapat digunakan untuk mengubah status peminjaman buku atau mengupdate informasi tertentu tanpa merubah seluruh entitas buku.

```
router.patch("/updateBuku/:isbn", async (req, res) => {
  try {
    const { isbn, judul, penulis, penerbit, tahun, kategori, no_rak } = req.body;

    // Cari buku berdasarkan ISBN
    const existingBook = await Book.findOne({ where: { isbn } });

    if (!existingBook) {
      return res.status(404).json({ message: "Buku tidak ditemukan" });
    }

    // Lakukan pembaruan data buku
    const updatedBook = await existingBook.update({
      judul,
      penulis,
      penerbit,
      tahun,
      kategori,
      no_rak
    });

    // Mengembalikan data buku yang baru diperbarui sebagai respons API
    res.json({
      success: true,
      data: updatedBook,
    });
  } catch (error) {
    console.error("Error:", error.message);
    // Mengembalikan respons API yang menyatakan kegagalan jika terjadi kesalahan
    res.status(500).json({
      success: false,
      error: "Internal Server Error",
    });
  }
});
```

```
// Mengembalikan data buku yang baru diperbarui sebagai respons API
res.json({
  success: true,
  data: updatedBook,
});
} catch (error) {
  console.error("Error:", error.message);
  // Mengembalikan respons API yang menyatakan kegagalan jika terjadi kesalahan
  res.status(500).json({
    success: false,
    error: "Internal Server Error",
  });
}
});
```

#### 4. DELETE:

Metode DELETE digunakan untuk menghapus sumber daya yang ditentukan oleh URL. Penggunaan metode ini akan menghapus data sumber daya tersebut dari sistem. Sebagai contoh, endpoint DELETE dapat digunakan untuk menghapus entitas buku dari perpustakaan jika buku tersebut sudah tidak tersedia atau tidak diperlukan lagi.

```
router.delete("/hapusBuku/:isbn", async (req, res) => {
  try {
    const { isbn } = req.params;

    // Cari buku berdasarkan ISBN
    const existingBook = await Book.findOne({ where: { isbn } });

    if (!existingBook) {
      return res.status(404).json({ success: false, error: 'Buku tidak ditemukan' });
    }

    // Hapus buku dari database
    await existingBook.destroy();

    // Mengembalikan respons sukses
    res.json({
      success: true,
      message: 'Buku berhasil dihapus',
    });
  } catch (error) {
    console.error('Error:', error.message);
    // Mengembalikan respons API yang menyatakan kegagalan jika terjadi kesalahan
    res.status(500).json({
      success: false,
      error: 'Internal Server Error',
    });
  }
});
```

React.js, yang digunakan untuk pengembangan frontend, merupakan sebuah pustaka JavaScript yang sangat populer dan powerful untuk membangun antarmuka pengguna (UI) yang dinamis dan interaktif dalam konteks pengembangan aplikasi web. Dikembangkan oleh Facebook, React.js mengusung pendekatan deklaratif yang efisien, memungkinkan pengembang untuk mengelola UI sebagai serangkaian komponen terpisah. Pendekatan utama dalam pengembangan dengan React adalah pemecahan UI menjadi komponen-komponen kecil yang dapat diulang. Setiap komponen bertanggung jawab atas tampilan dan logika mereka sendiri, memudahkan dalam pengorganisasian dan pemeliharaan kode. Pendekatan ini tidak hanya meningkatkan modularitas, tetapi juga memfasilitasi penggunaan kembali komponen-komponen yang telah dibuat, menghasilkan pengembangan yang lebih cepat dan efisien. Salah satu fitur kunci React adalah penggunaan JSX (JavaScript XML), yang memungkinkan penulisan sintaks HTML di dalam JavaScript. Ini meningkatkan keterbacaan kode dan mempermudah pengembang dalam merancang tampilan komponen. JSX kemudian dikompilasi menjadi JavaScript standar untuk dijalankan di browser. React juga dikenal dengan konsep Virtual DOM. Dengan Virtual DOM, React dapat melakukan pembaruan UI dengan lebih efisien. Ketika ada perubahan, React membandingkan Virtual DOM dengan DOM aktual, dan hanya merender perubahan yang diperlukan, menghindari pembaruan lengkap yang dapat membebani performa.

Express adalah kerangka kerja (framework) backend yang populer dalam pengembangan aplikasi web dengan menggunakan bahasa pemrograman JavaScript dan berbasis Node.js. Dikembangkan sebagai kerangka kerja minimalis, Express memberikan pendekatan yang sederhana namun sangat fleksibel untuk membangun server dan API. Dengan filosofi "menjadi minimalis", Express memberikan kebebasan kepada pengembang untuk memilih dan mengintegrasikan berbagai komponen atau modul tambahan sesuai dengan kebutuhan proyek. Meskipun bersifat minimalis, Express menyediakan fungsi-fungsi dasar yang dibutuhkan dalam pembangunan aplikasi backend, seperti routing, middleware, dan manajemen permintaan dan respons HTTP. Salah satu fitur utama Express adalah sistem routing yang kuat. Dengan Express, pengembang dapat dengan mudah menentukan rute-rute (routes) yang menghubungkan URL dengan fungsionalitas spesifik di server. Hal ini mempermudah pengelolaan endpoint dan menangani permintaan HTTP yang masuk. Express juga mendukung penggunaan middleware, yang memungkinkan pengembang menyisipkan fungsionalitas tambahan di antara proses permintaan dan respons. Middleware dapat digunakan untuk keperluan seperti otentikasi, logging, kompresi, dan banyak lagi.

Kemampuan ini memberikan fleksibilitas tambahan dalam mengelola aliran data dalam server. Kerangka kerja Express bersifat terbuka dan memungkinkan integrasi mudah dengan berbagai pustaka dan modul pihak ketiga melalui manajer paket Node.js, seperti npm. Ini memudahkan pengembang untuk menambahkan fitur tambahan sesuai kebutuhan proyek tanpa harus membangun semuanya dari awal.

## **BAB II**

### **Langkah-langkah membuat project (Backend dan Frontend)**

Langkah-langkah membuat proyek backend menggunakan Node.js dengan Express dan beberapa paket tambahan. Berikut adalah penjelasan prosesnya :

- Pertama, menginisialisasi proyek Node.js dengan menjalankan perintah `npm init` untuk membuat file `package.json`.
- Setelah itu menggunakan `nodemon` sebagai alat pemantau perubahan untuk mempermudah pengembangan dengan menjalankan perintah `npm i nodemon --save`.
- Kemudian, membuat folder `client` dan `src` untuk menyimpan file-file proyek. Di dalam folder `src`, membuat folder `routes` yang nantinya akan digunakan untuk menyimpan rute-rute aplikasi.
- Menginstal beberapa paket yang dibutuhkan untuk proyek ini menggunakan perintah `npm i express morgan dotenv body-parser cors fastest-validator bcrypt --save`. Paket-paket ini termasuk Express sebagai framework web, Morgan untuk logging, dotenv untuk mengelola variabel lingkungan, body-parser untuk membaca data dari body permintaan, cors untuk mengelola permintaan lintas domain, fastest-validator untuk validasi cepat, dan bcrypt untuk pengamanan password.
- Menginstal paket seperti `express`, `http-errors`, dan `cookie-parser` yang akan digunakan untuk mengelola permintaan HTTP dan cookies.
- Menginstal paket-paket terkait database dan autentikasi, yaitu `mysql2`, `jsonwebtoken`, `sequelize`, dan `sequelize-cli`.
- Setelah instalasi, menginisialisasi pengaturan Sequelize dengan menjalankan perintah `npx sequelize init`. Langkah ini membuat struktur proyek Sequelize dan file konfigurasi. Kemudian, Anda membuat migrasi tabel pengguna (`users`) dengan perintah `npx sequelize migration:create --name=create-table-users`, yang akan digunakan untuk membuat dan mengelola tabel pengguna di database.

Proses membuat proyek frontend React dengan menggunakan Create React App dan beberapa paket pendukung seperti react-router-dom, axios, date-fns, react-icons, dan tailwindcss.

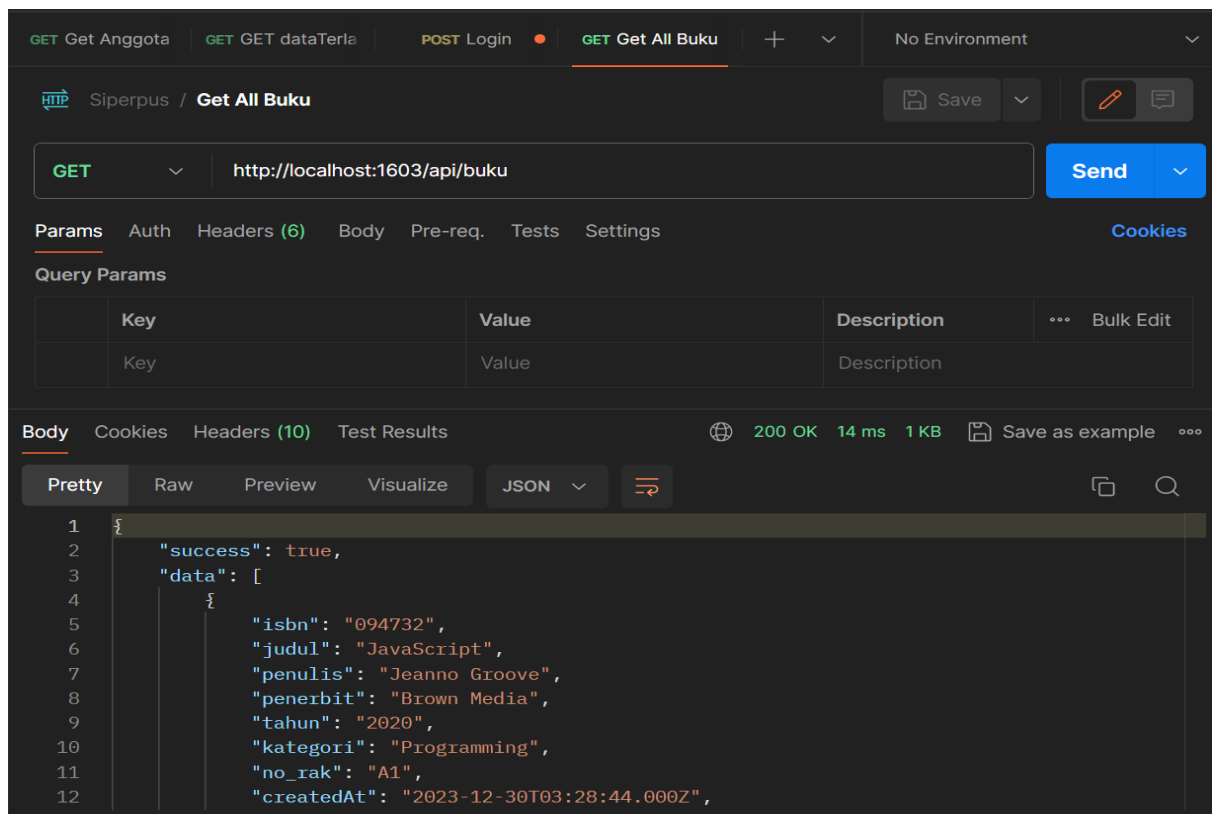
- Inisialisasi Proyek React: `npx create-react-app client`  
Penginstallan reactJs dimana meng generate folder-folder yang ada pada framework reactJs kecuali Pages dan Components itu membuat manual.
- Instalasi React Router Dom: `npm install react-router-dom`  
Menginstall function sebuah rute untuk page frontend
- Instalasi Axios : `npm install axios`  
Menginstal paket axios untuk melakukan permintaan HTTP seperti pengambilan data dari server atau API.
- Instalasi Date-fns: `npm install date-fns`  
Menginstal paket date-fns untuk memudahkan manipulasi dan format tanggal dalam proyek React.
- Instalasi React Icons : `npm install react-icons`  
Menginstal paket react-icons untuk menggunakan ikon secara mudah di aplikasi React tanpa harus mengunduh ikon-ikon secara terpisah.
- Instalasi Tailwind CSS: `npm install tailwindcss`  
Menginstal paket tailwindcss untuk menggunakan Tailwind CSS, sebuah framework CSS yang memungkinkan pengembang untuk membuat antarmuka pengguna dengan cepat.
- Konfigurasi Tailwind CSS  
Menambahkan konfigurasi Tailwind CSS di file `tailwind.config.js` dan mengimpornya di dalam file CSS. Menyertakan dan mengonfigurasi Tailwind CSS agar dapat digunakan dalam proyek React.  
`content: ["/src/**/*.html.js"],` untuk memuat jenis file di reactJs  
`@tailwind base` : ini memuat aturan dasar untuk elemen HTML, memberikan gaya dasar untuk elemen seperti heading, paragraph, dll.  
`@tailwind components` : ini memuat aturan untuk komponen-komponen UI yang dihasilkan oleh Tailwind CSS, seperti tombol, form, dll.  
`@tailwind utilities` : untuk menggunakan tailwind pada project reactJs. Ini memuat utilitas yang dapat digunakan untuk mengatur properti khusus seperti margin, padding, warna, dan sebagainya.

## BAB III

### Lakukan Test Backend Restful API dengan Postman dan dokumentasikan.

Request Get All buku akan menampilkan semua data buku yang tersimpan di API. Jika permintaan ini berhasil, API akan mengembalikan respons JSON yang berisi data buku. Data buku ini akan mencakup informasi seperti

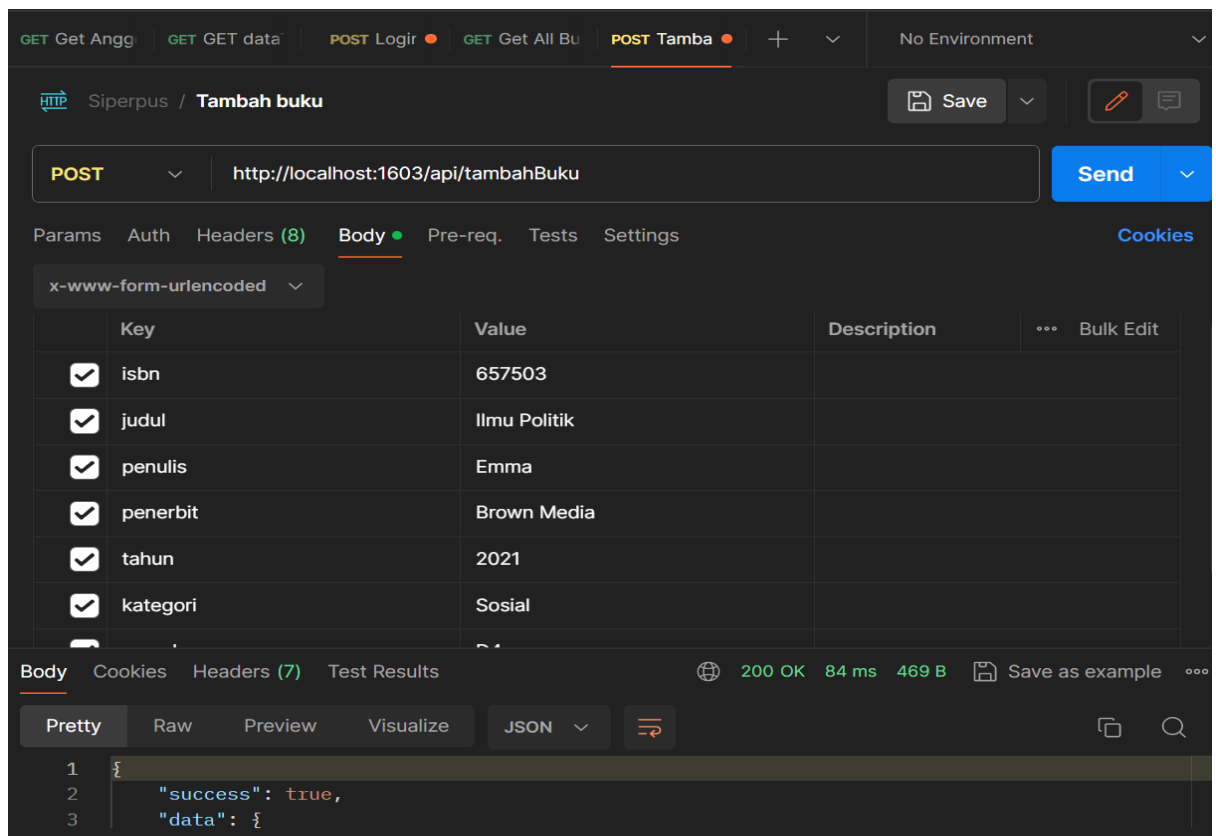
- ISBN
- Judul
- Penulis
- Penerbit
- tahun
- kategori, dan
- no rak.



Request Post Tambahbuku digunakan untuk menambahkan data buku baru ke API. Data buku yang akan ditambahkan harus mencakup informasi seperti

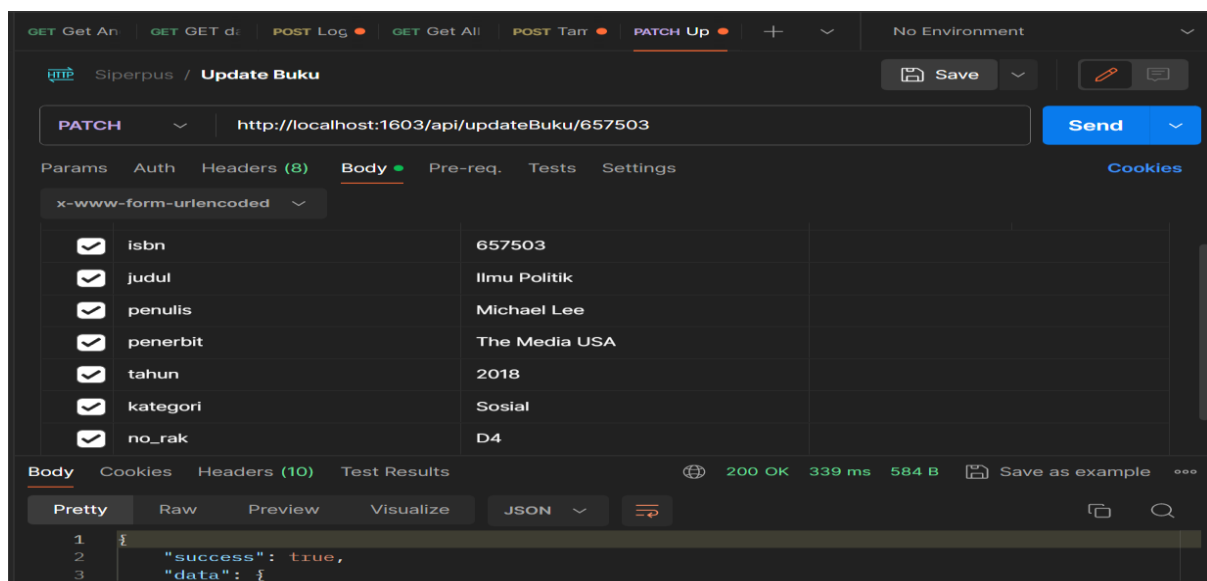
- ISBN
- Judul
- Penulis
- Penerbit
- tahun
- kategori, dan
- no rak





Request Patch updatebuku digunakan untuk mengupdate data buku ke API. Data buku yang akan ditambahkan harus mencakup informasi seperti

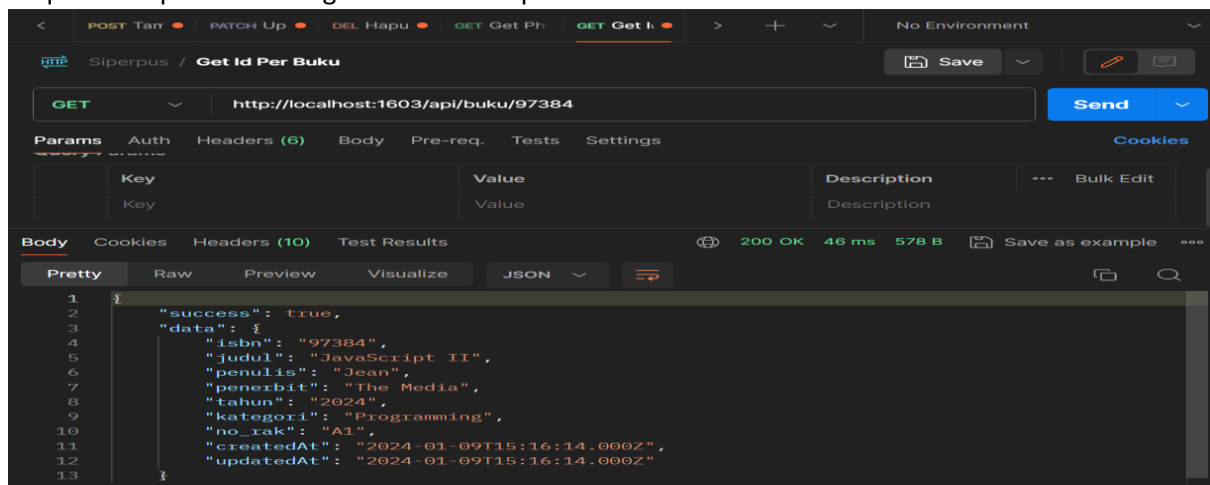
- ISBN
- Judul
- Penulis
- Penerbit
- tahun
- kategori, dan
- no\_rak



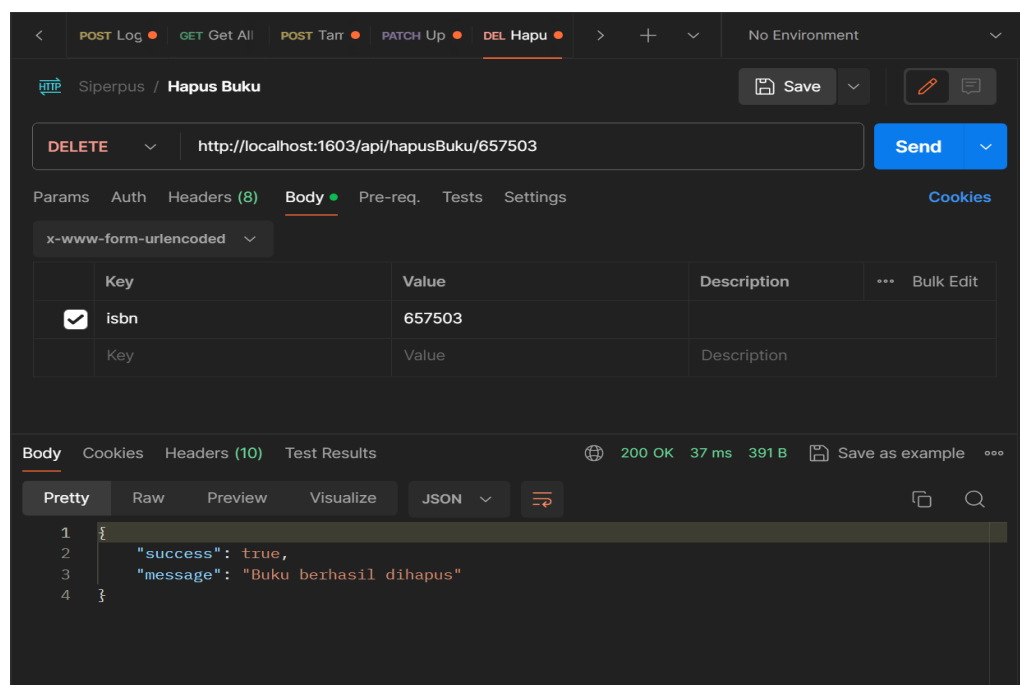
Request Delete hapus buku ini digunakan menghapus data pada data buku ke API. Data buku yang akan dihapus harus mencakup informasi seperti

- ISBN
- Judul
- Penulis
- Penerbit
- tahun
- kategori, dan
- no\_rak

Request Get per id buku digunakan untuk pencarian data buku berdasarkan id atau isbn

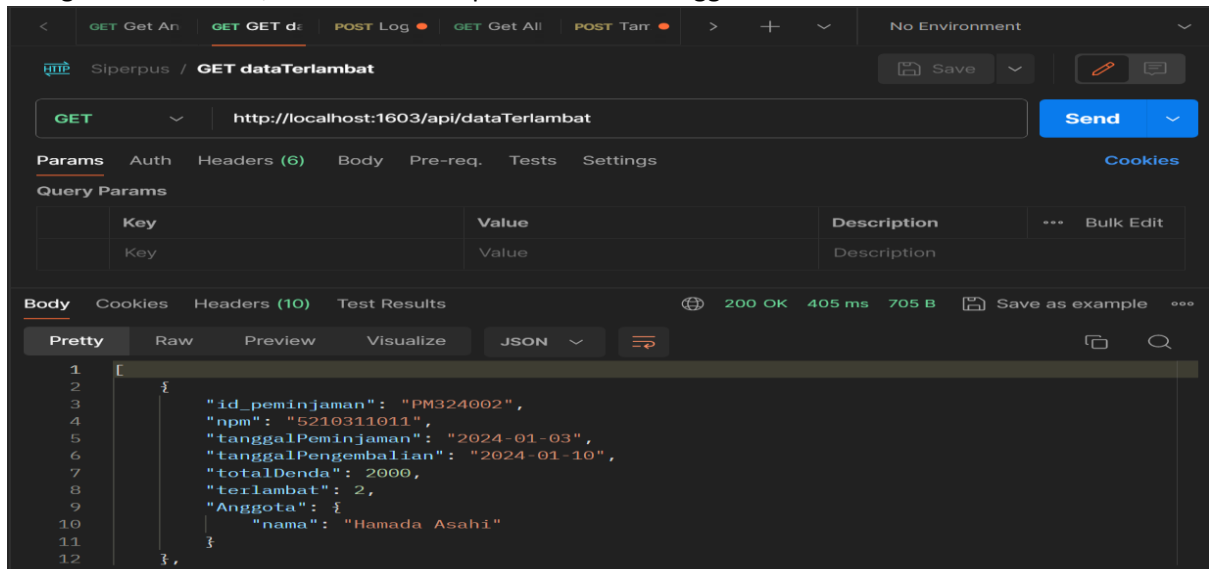


Permintaan ini akan menampilkan semua data anggota yang tersimpan di API. Data anggota ini akan mencakup informasi seperti npm, nama, prodi, jenis kelamin, masa berlaku, no hp, dan gambar.



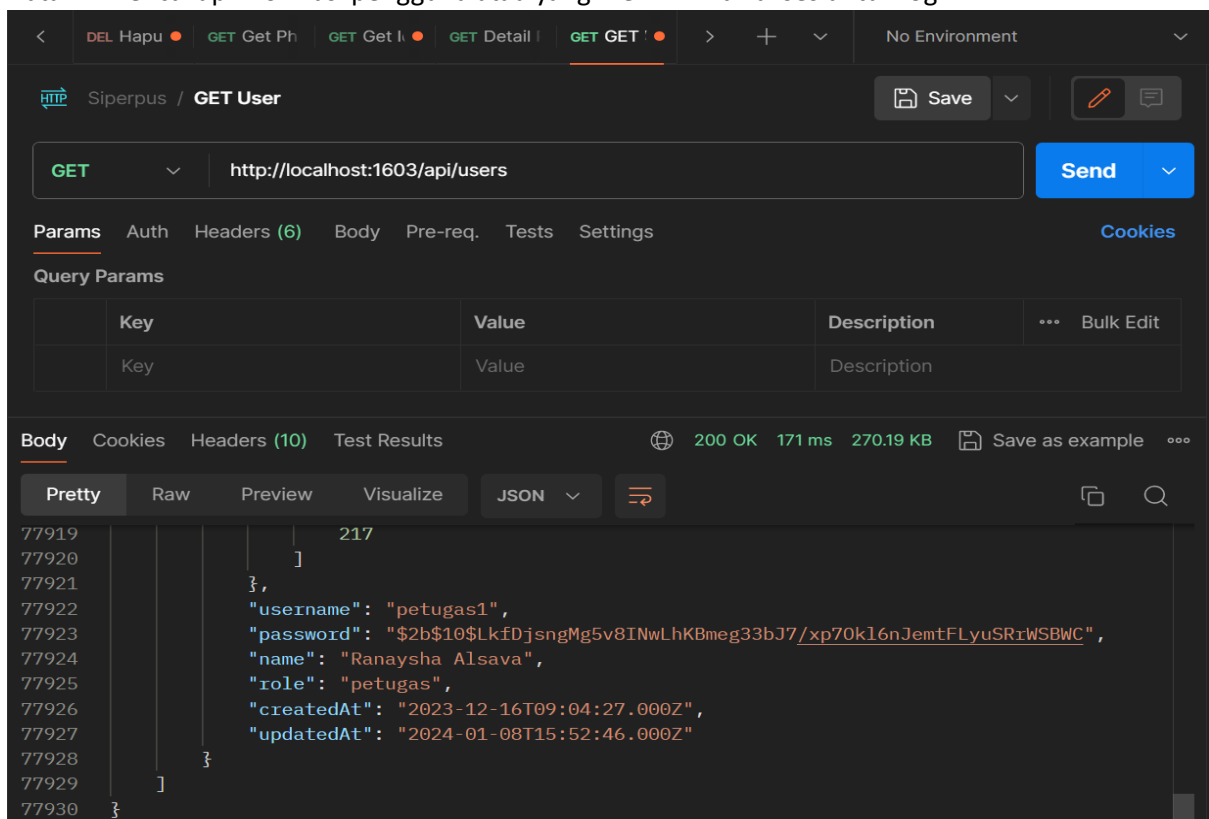


Get data terlambat di API. Permintaan ini akan menampilkan semua data peminjaman buku yang telah terlambat. get data terlambat ini digunakan untuk mendapatkan semua data peminjaman buku yang telah terlambat. Data peminjaman buku yang telah terlambat ini dapat digunakan untuk berbagai keperluan, misalnya untuk mengirimkan email peringatan ke anggota yang terlambat mengembalikan buku, atau untuk memproses denda keanggotaan.



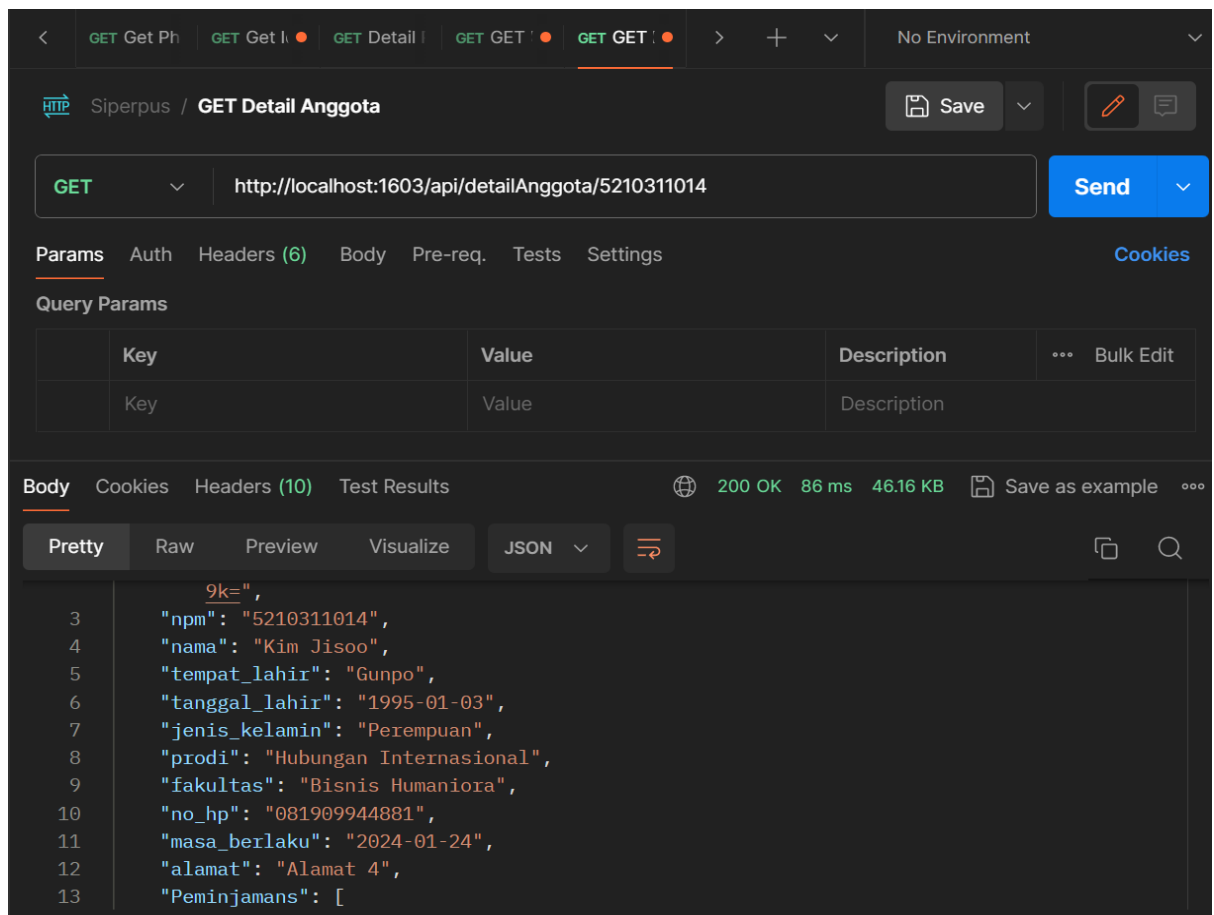
## USER

Get users di API. Permintaan ini akan mengembalikan semua data pengguna yang tersimpan di API. Data ini mencakup informasi pengguna atau yang memiliki hak akses untuk login



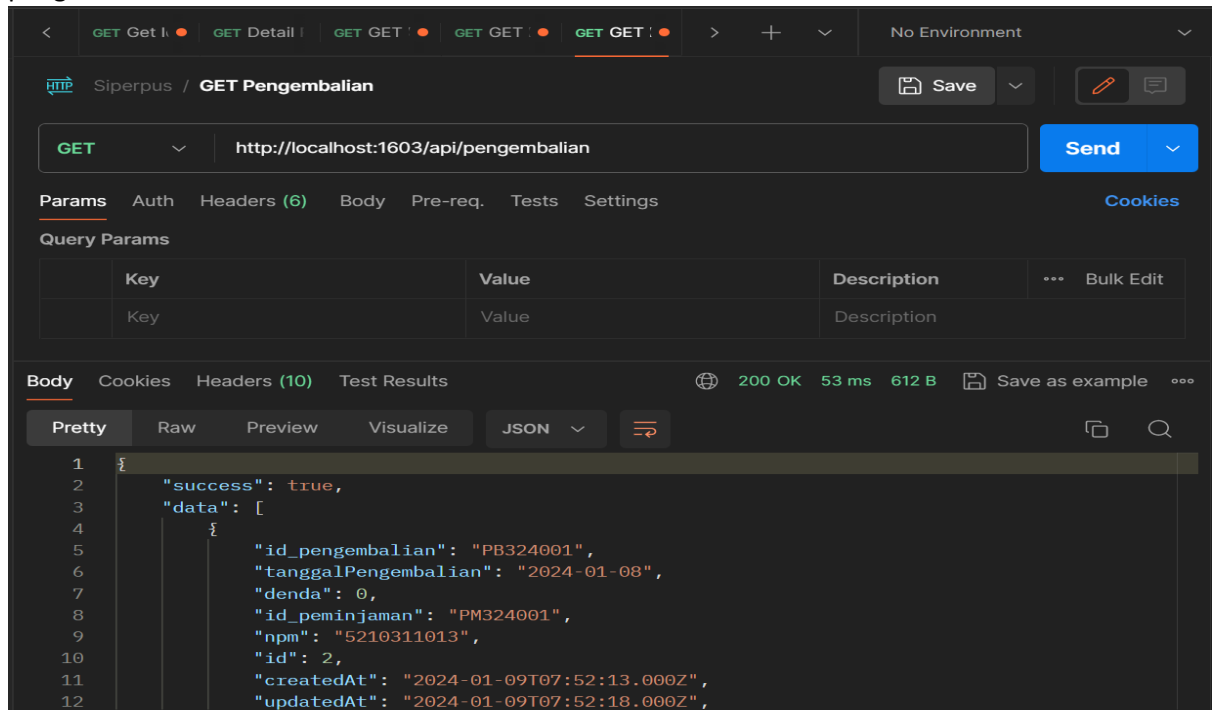
Get detail anggota API. Permintaan ini akan mengembalikan semua data detail anggota dengan ID anggota. Data detail anggota ini mencakup

- npm
- nama
- tempat lahir
- tanggal lahir
- jenis kelamin
- prodi
- fakultas
- no hp
- masa berlaku
- alamat
- id peminjaman

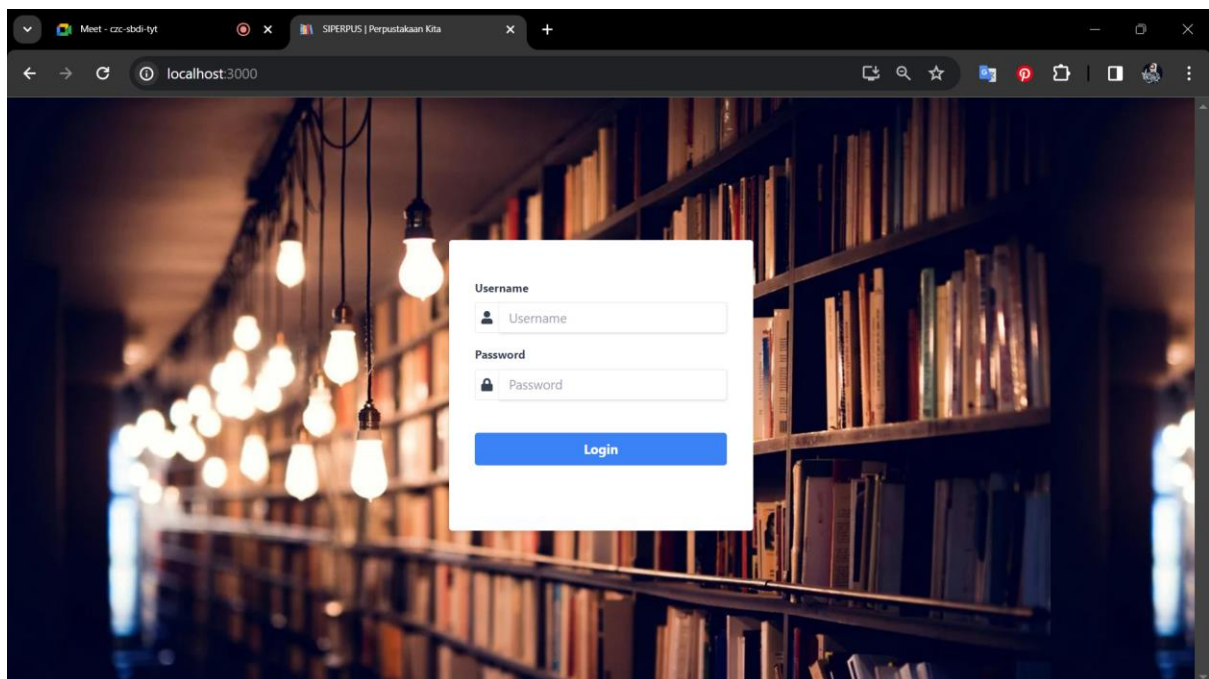


Get pengebalan API. Permintaan ini akan menampilkan semua data pengebalan buku yang telah dilakukan oleh anggota. digunakan untuk mendapatkan semua data pengebalan buku yang telah dilakukan oleh anggota. Data pengebalan buku ini dapat digunakan untuk berbagai keperluan, misalnya untuk menampilkan daftar pengebalan buku, atau untuk memproses transaksi

pengembalian buku.

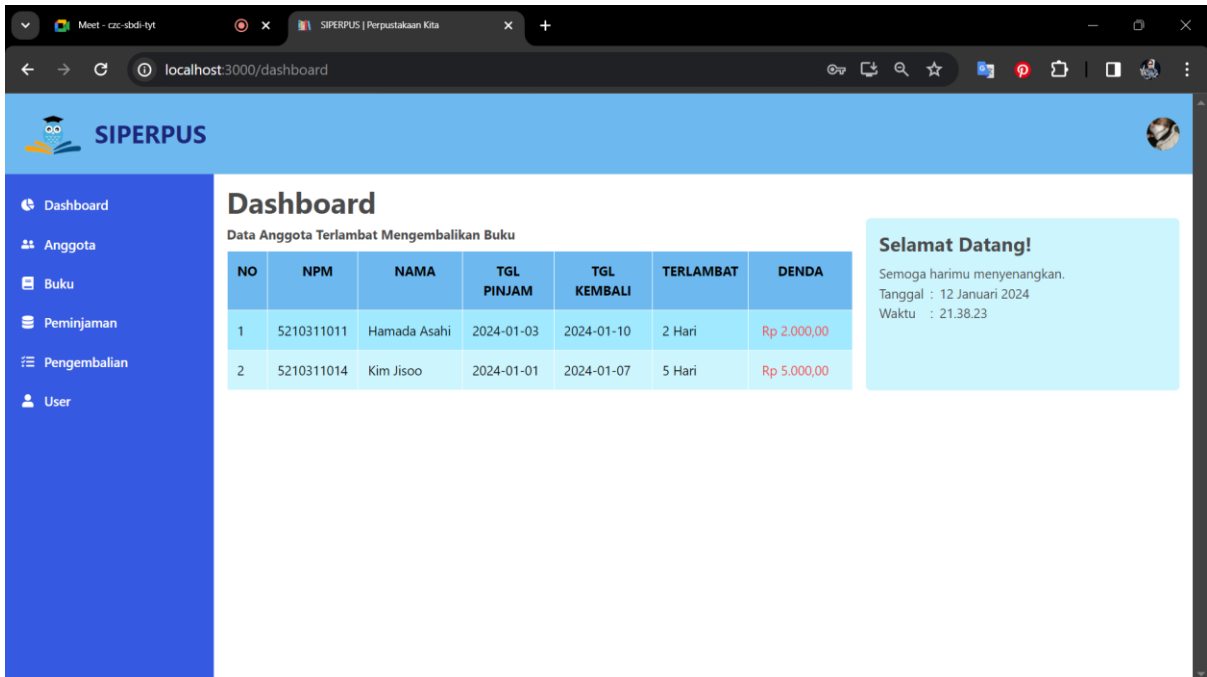


Test Frontend dengan berfungsinya CRUD pada tampilan web.



Untuk masuk ke dalam sistem perpustakaan tersebut, pengguna harus memasukkan username dan password yang benar. Dengan menginputkan username = admin1, password =

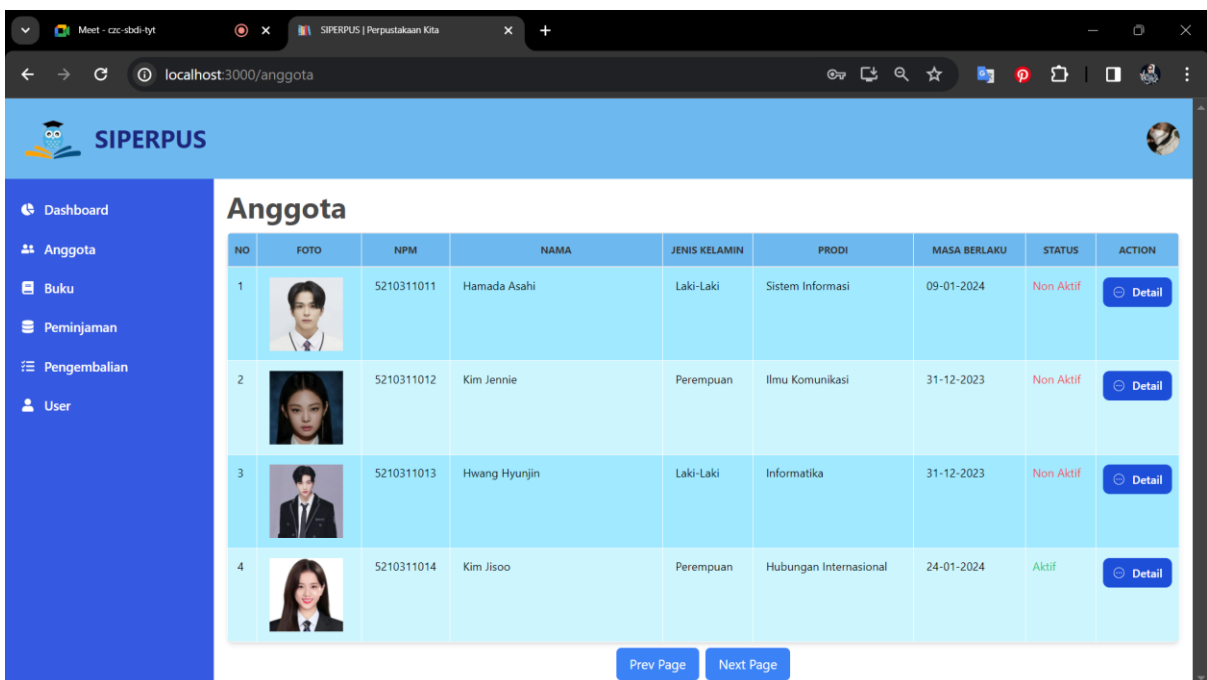
password123. Jika salah menginputkan maka akan muncul pesan “salah”. Jika benar maka akan menampilkan halaman selanjutnya.







NO	NPM	NAMA	TGL PINJAM	TGL KEMBALI	TERLAMBAT	DENDA
1	5210311011	Hamada Asahi	2024-01-03	2024-01-10	2 Hari	Rp 2.000,00
2	5210311014	Kim Jisoo	2024-01-01	2024-01-07	5 Hari	Rp 5.000,00

**Selamat Datang!**  
Semoga harimu menyenangkan.  
Tanggal : 12 Januari 2024  
Waktu : 21.38.23

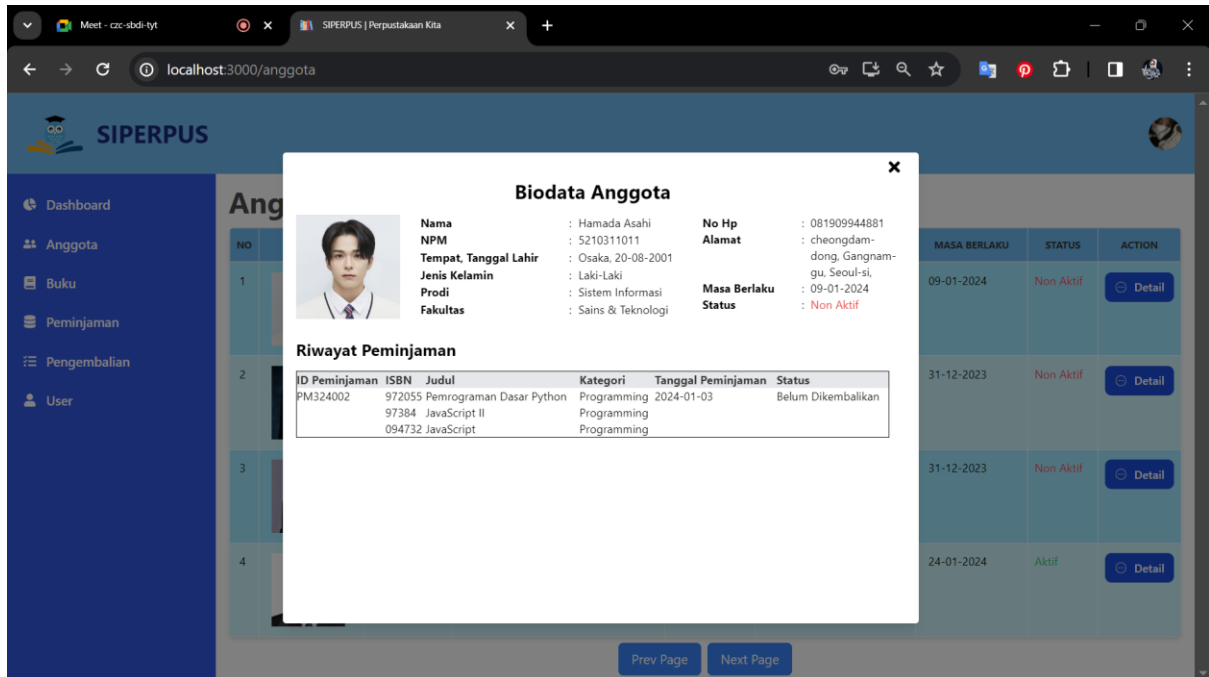
Setelah login maka akan ke halaman dashboard, dengan disambut tulisan selamat datang dan menampilkan tanggal dan waktu saat mengaksesnya, dan didalamnya terdapat data anggota yang terlambat mengembalikan buku. Informasi ini dapat digunakan oleh petugas perpustakaan untuk memantau status peminjaman dan pengembalian buku, serta untuk menghitung denda yang harus dibayar oleh anggota yang terlambat mengembalikan buku.



NO	FOTO	NPM	NAMA	JENIS KELAMIN	PRODI	MASA BERLAKU	STATUS	ACTION
1		5210311011	Hamada Asahi	Laki-Laki	Sistem Informasi	09-01-2024	Non Aktif	<a href="#">Detail</a>
2		5210311012	Kim Jennie	Perempuan	Ilmu Komunikasi	31-12-2023	Non Aktif	<a href="#">Detail</a>
3		5210311013	Hwang Hyunjin	Laki-Laki	Informatika	31-12-2023	Non Aktif	<a href="#">Detail</a>
4		5210311014	Kim Jisoo	Perempuan	Hubungan Internasional	24-01-2024	Aktif	<a href="#">Detail</a>


[Prev Page](#) [Next Page](#)

Pada halaman anggota menampilkan data anggota perpustakaan, status diambil berdasarkan masa berlaku, jika melebihi tgl masa berlaku maka status non aktif.



The screenshot shows the SIPERUS web application interface. The left sidebar contains navigation links: Dashboard, Anggota, Buku, Peminjaman, Pengembalian, and User. The main content area displays the 'Anggota' page. A modal window titled 'Biodata Anggota' is open, showing member details and a table of borrowing history.

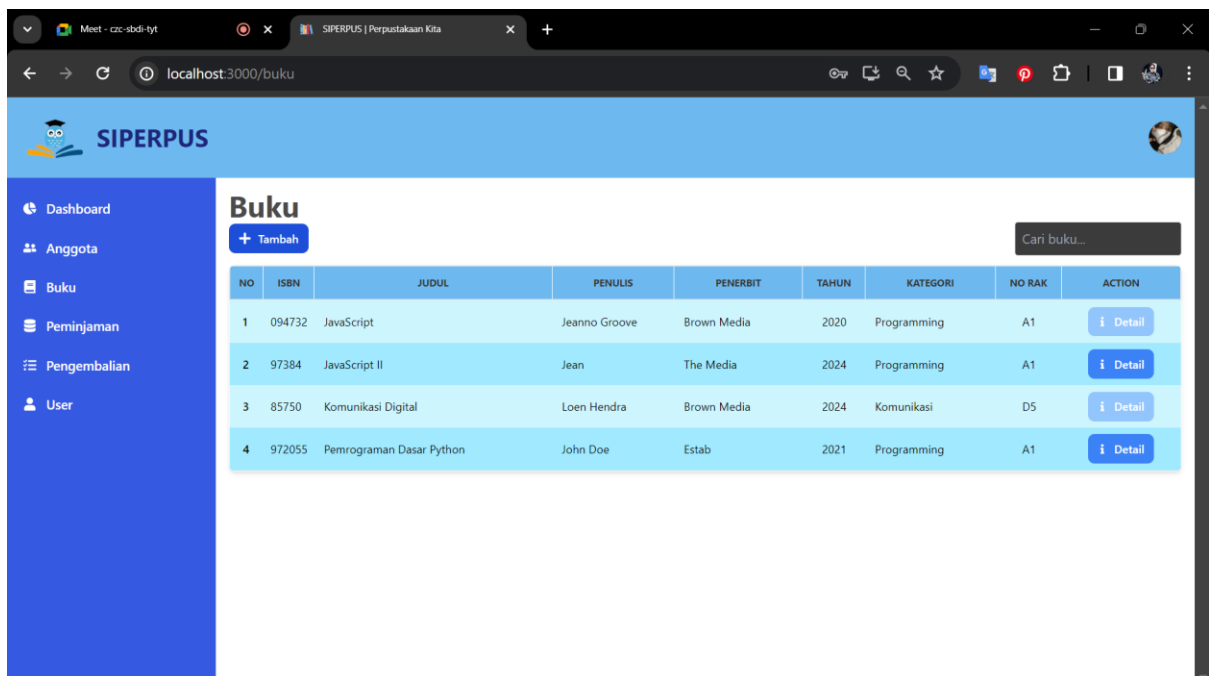
**Biodata Anggota**

	<b>Nama</b> : Hamada Asahi <b>NPM</b> : 5210311011 <b>Tempat, Tanggal Lahir</b> : Osaka, 20-08-2001 <b>Jenis Kelamin</b> : Laki-Laki <b>Prodi</b> : Sistem Informasi <b>Fakultas</b> : Sains & Teknologi	<b>No Hp</b> : 081909944881 <b>Alamat</b> : cheongdam-dong, Gangnam-gu, Seoul-si <b>Masa Berlaku</b> : 09-01-2024 <b>Status</b> : Non Aktif
---	---	--

**Riwayat Peminjaman**

ID Peminjaman	ISBN	Judul	Kategori	Tanggal Peminjaman	Status
PM324002	972055	Pemrograman Dasar Python	Programming	2024-01-03	Belum Dikembalikan
	97384	JavaScript II	Programming		
	094732	JavaScript	Programming		

Detail anggota menyajikan informasi lengkap mengenai seorang anggota, termasuk biodata pribadi dan riwayat peminjaman buku yang dilakukan oleh anggota tersebut. Sistem memungkinkan satu ID anggota menampilkan riwayat peminjaman lebih dari satu buku.



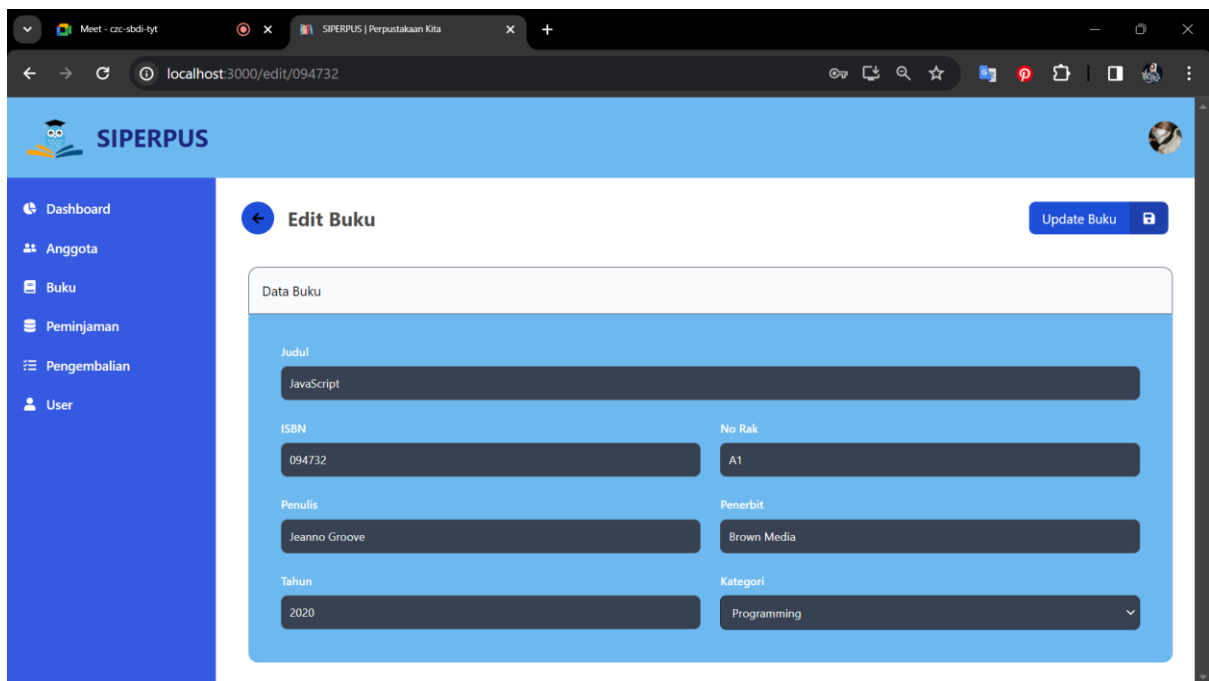
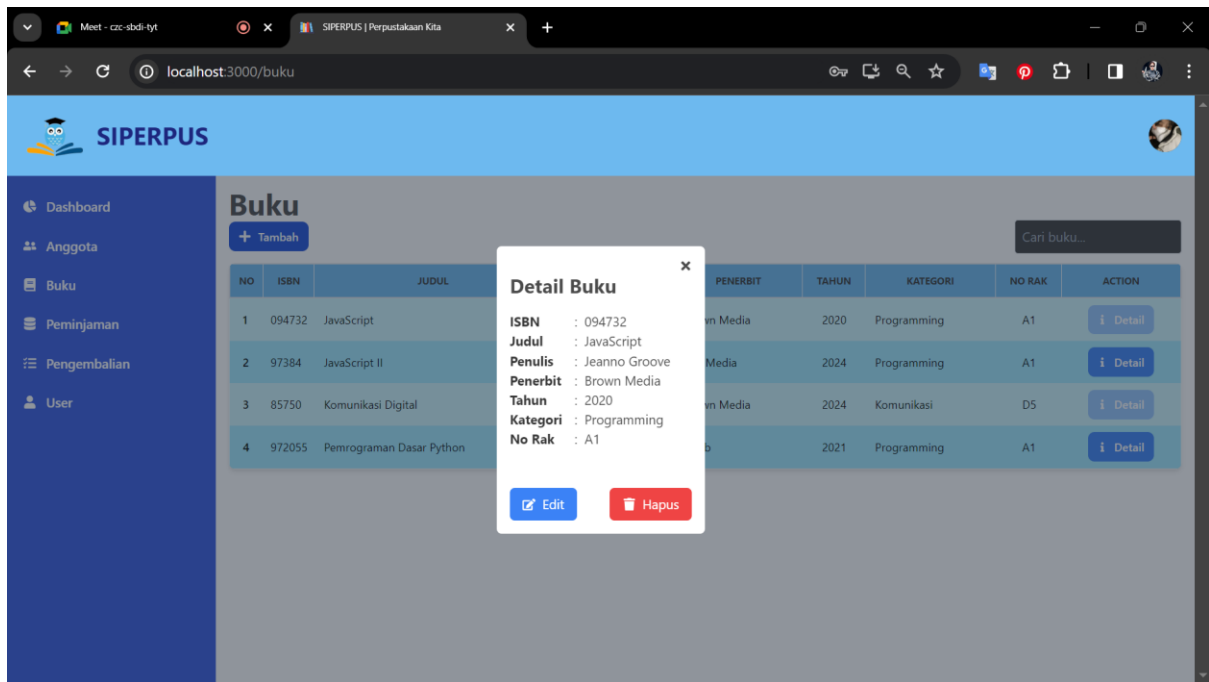
The screenshot shows the SIPERUS web application interface. The left sidebar contains navigation links: Dashboard, Anggota, Buku, Peminjaman, Pengembalian, and User. The main content area displays the 'Buku' page. A table of books is shown with columns for NO, ISBN, JUDUL, PENULIS, PENERBIT, TAHUN, KATEGORI, NO RAK, and ACTION.

**Buku**

NO	ISBN	JUDUL	PENULIS	PENERBIT	TAHUN	KATEGORI	NO RAK	ACTION
1	094732	JavaScript	Jeanno Groove	Brown Media	2020	Programming	A1	<a href="#">Detail</a>
2	97384	JavaScript II	Jean	The Media	2024	Programming	A1	<a href="#">Detail</a>
3	85750	Komunikasi Digital	Loen Hendra	Brown Media	2024	Komunikasi	D5	<a href="#">Detail</a>
4	972055	Pemrograman Dasar Python	John Doe	Estab	2021	Programming	A1	<a href="#">Detail</a>

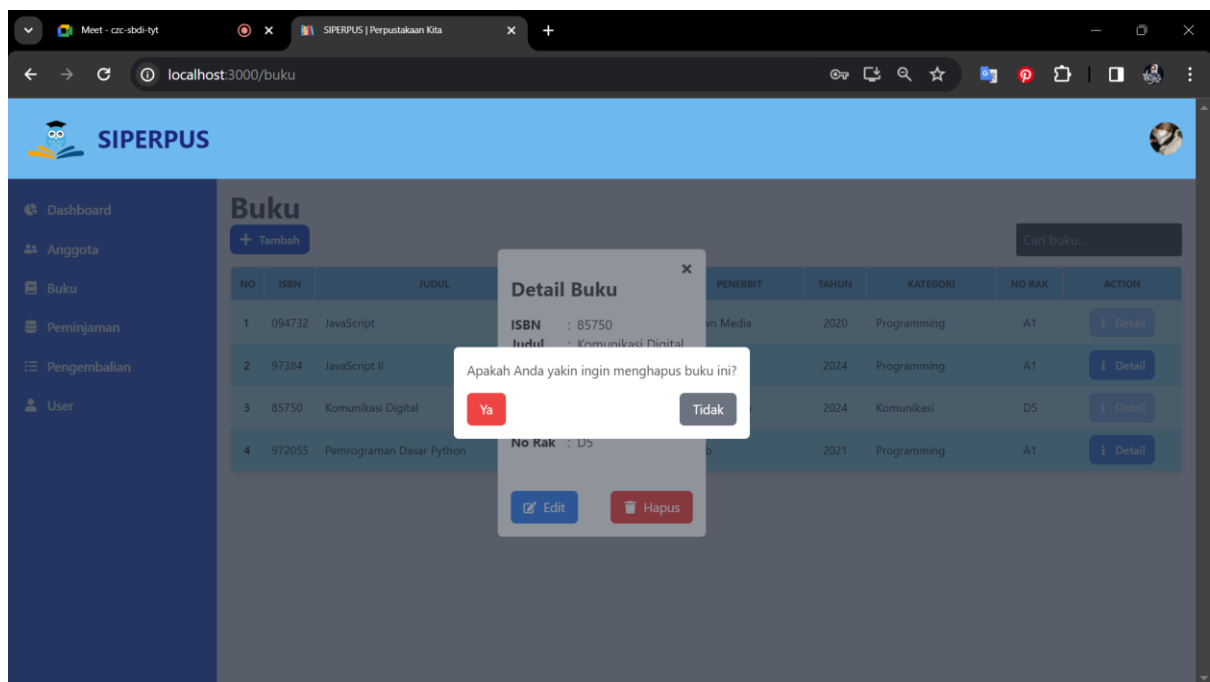
Pada halaman buku menampilkan data buku dan juga bisa menambahkan, cari, dan jika mengklik action detail maka akan ada button untuk mengedit ataupun hapus buku.



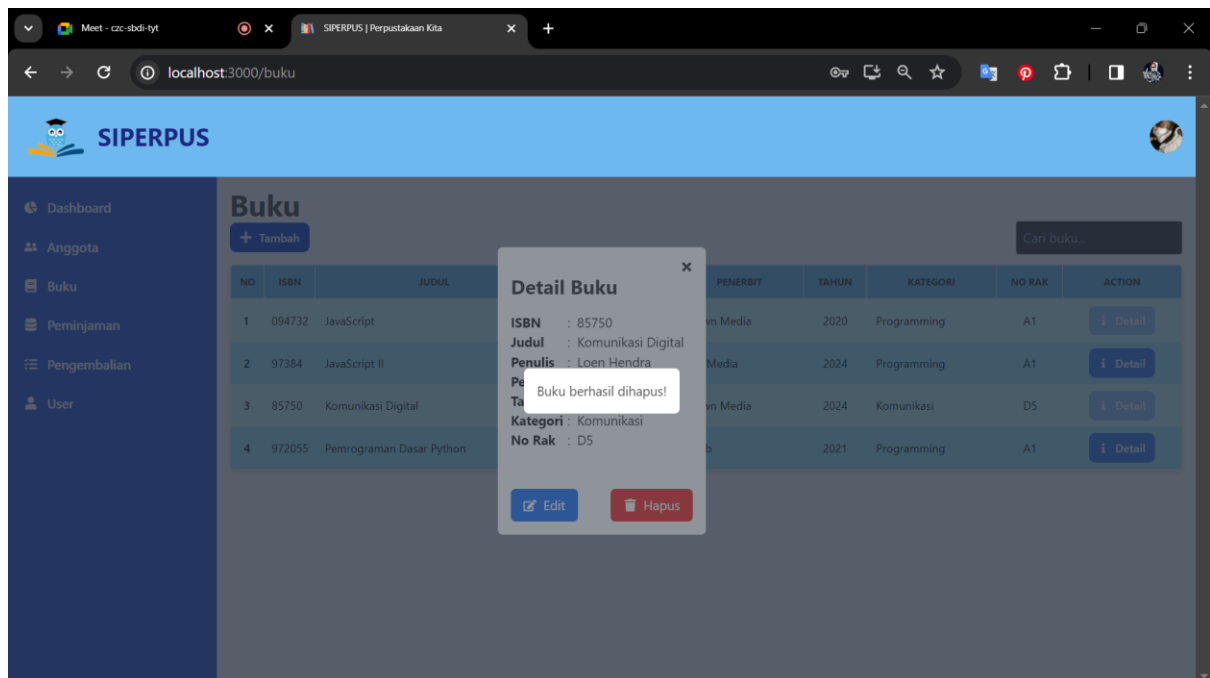


Ketika pengguna mengklik opsi "Edit Buku," sistem akan membawa ke halaman pengeditan buku di mana kolom-kolom yang berisi data terpilih untuk diedit akan ditampilkan. Pada halaman ini, pengguna dapat melakukan perubahan yang diinginkan terhadap informasi buku, seperti judul, penulis, atau atribut lainnya. Setelah menyelesaikan proses pengeditan, pengguna dapat menekan tombol "Update Buku." Setelah pengguna menekan tombol "Update Buku," sistem akan mengolah

perubahantersebut.

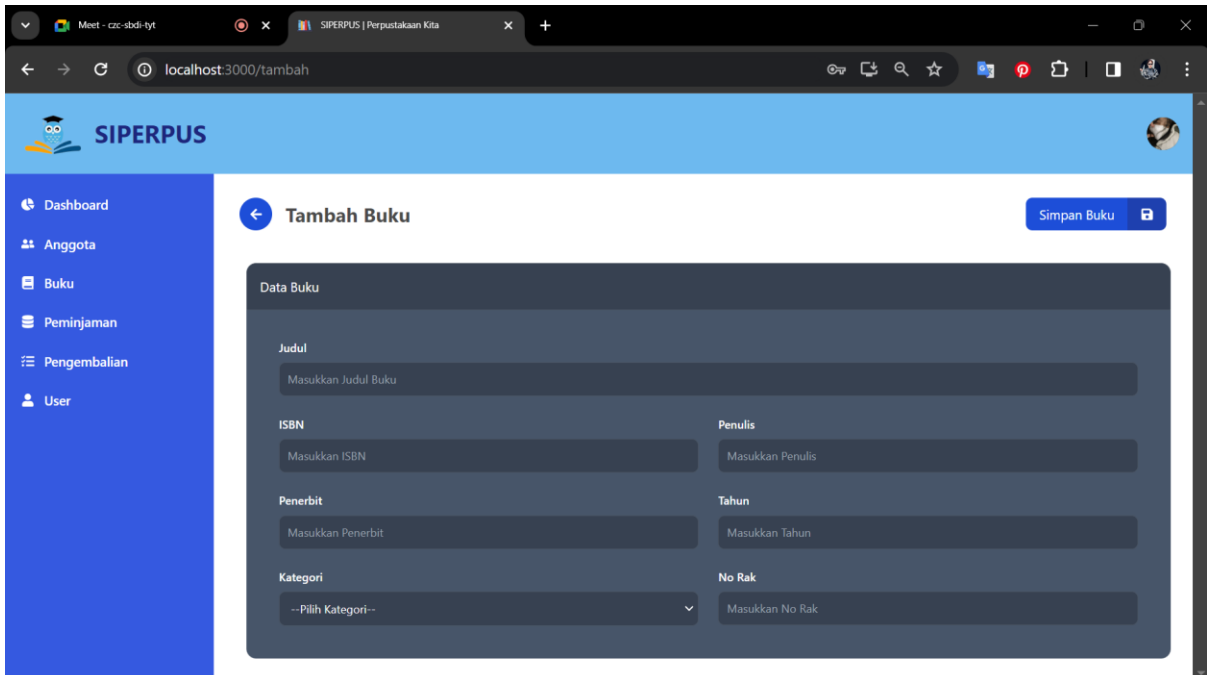


Untuk hapus maka sebelum data dihapus dan klik hapus maka akan muncul konfirmasi “apakah anda yakin ingin menghapus buku ini?” jika ya aka akan ada “buku berhasil dihapus” dan kembali ke halaman data buku dan data yang dihapus akan hilang didalam data buku.



konfirmasi sebelum penghapusan memberikan perlindungan kepada pengguna dari penghapusan tidak sengaja atau tidak diinginkan. Pesan sukses memberikan umpan balik positif kepada pengguna bahwa buku telah dihapus dengan sukses. Pengalaman ini dirancang untuk menjaga keamanan dan

memberikan kejelasan pada pengguna terkait tindakan penghapusan yang dilakukan dalam sistem manajemen buku.

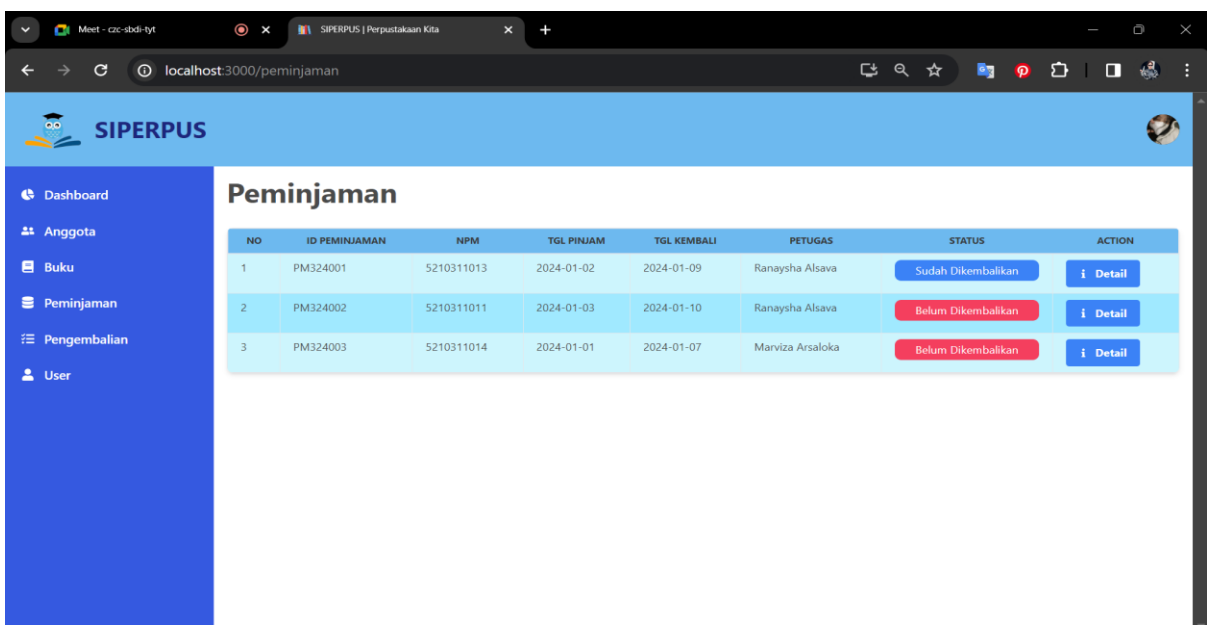


The screenshot displays the 'Tambah Buku' (Add Book) page in the SIPERUS application. The page features a sidebar with navigation links: Dashboard, Anggota, Buku, Peminjaman, Pengembalian, and User. The main content area is titled 'Tambah Buku' and contains a form for adding a new book. The form includes the following fields:

- Judul:** A text input field with the placeholder 'Masukkan Judul Buku'.
- ISBN:** A text input field with the placeholder 'Masukkan ISBN'.
- Penulis:** A text input field with the placeholder 'Masukkan Penulis'.
- Penerbit:** A text input field with the placeholder 'Masukkan Penerbit'.
- Tahun:** A text input field with the placeholder 'Masukkan Tahun'.
- Kategori:** A dropdown menu with the placeholder '--Pilih Kategori--'.
- No Rak:** A text input field with the placeholder 'Masukkan No Rak'.

A 'Simpan Buku' button is located in the top right corner of the form area.

Untuk menambahkan data buku, pengguna harus mengisi beberapa informasi penting seperti judul, ISBN, penerbit, kategori, penulis, tahun terbit, dan rak buku. Setelah semua informasi tersebut diisi dengan benar, pengguna dapat menyimpan data buku dengan mengklik tombol "Simpan Buku." Data yang baru ditambahkan akan otomatis ditampilkan di halaman data buku. Dengan implementasi ini, pengguna dapat dengan mudah menambahkan buku baru ke dalam sistem dengan mengikuti langkah-langkah yang jelas. Pesan konfirmasi memberikan umpan balik positif kepada pengguna dan mengonfirmasi bahwa data buku telah berhasil ditambahkan ke dalam sistem.



The screenshot displays the 'Peminjaman' (Borrowing) page in the SIPERUS application. The page features a sidebar with navigation links: Dashboard, Anggota, Buku, Peminjaman, Pengembalian, and User. The main content area is titled 'Peminjaman' and contains a table with the following columns:

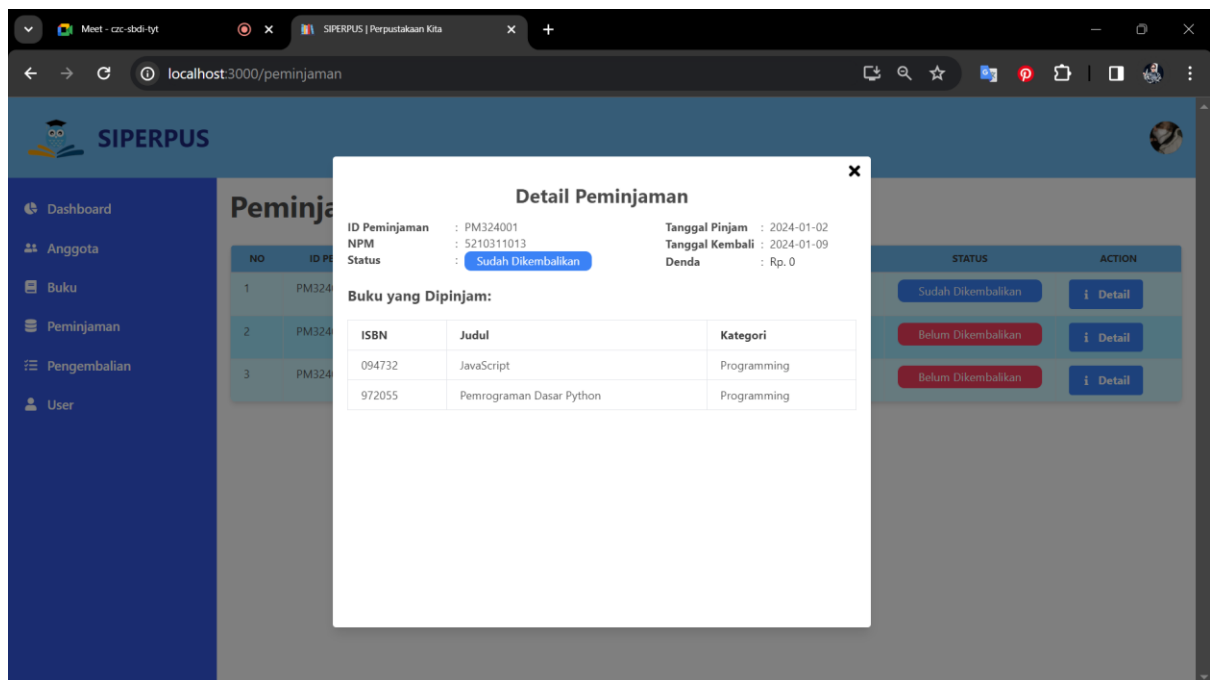
- NO**
- ID PEMINJAMAN**
- NPM**
- TGL PINJAM**
- TGL KEMBALI**
- PETUGAS**
- STATUS**
- ACTION**

The table contains three rows of borrowing data:

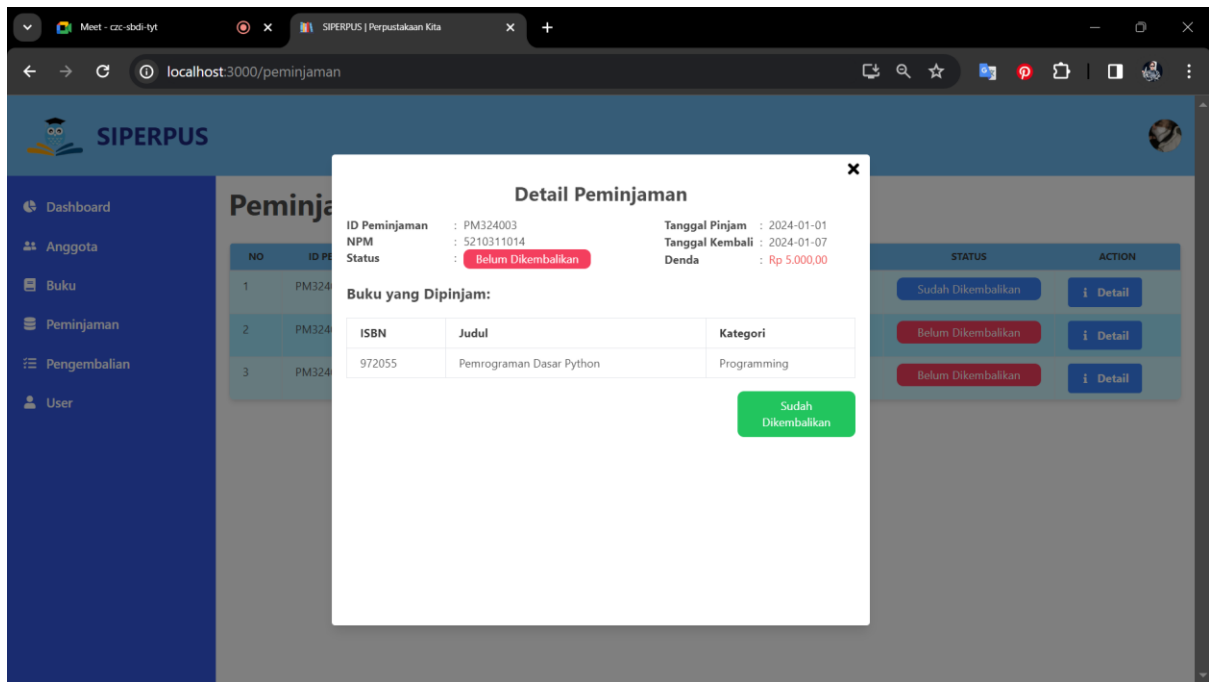
NO	ID PEMINJAMAN	NPM	TGL PINJAM	TGL KEMBALI	PETUGAS	STATUS	ACTION
1	PM324001	5210311013	2024-01-02	2024-01-09	Ranaysha Alsava	Sudah Dikembalikan	<a href="#">Detail</a>
2	PM324002	5210311011	2024-01-03	2024-01-10	Ranaysha Alsava	Belum Dikembalikan	<a href="#">Detail</a>
3	PM324003	5210311014	2024-01-01	2024-01-07	Marviza Arsaloka	Belum Dikembalikan	<a href="#">Detail</a>

Pada halaman peminjaman akan menampilkan data peminjaman anggota. Terdapat id\_peminjaman, npm, tgl\_peminjam, tgl\_kembali, status, dan action detail. Pada status akan mempengaruhi keterlambatan denda.

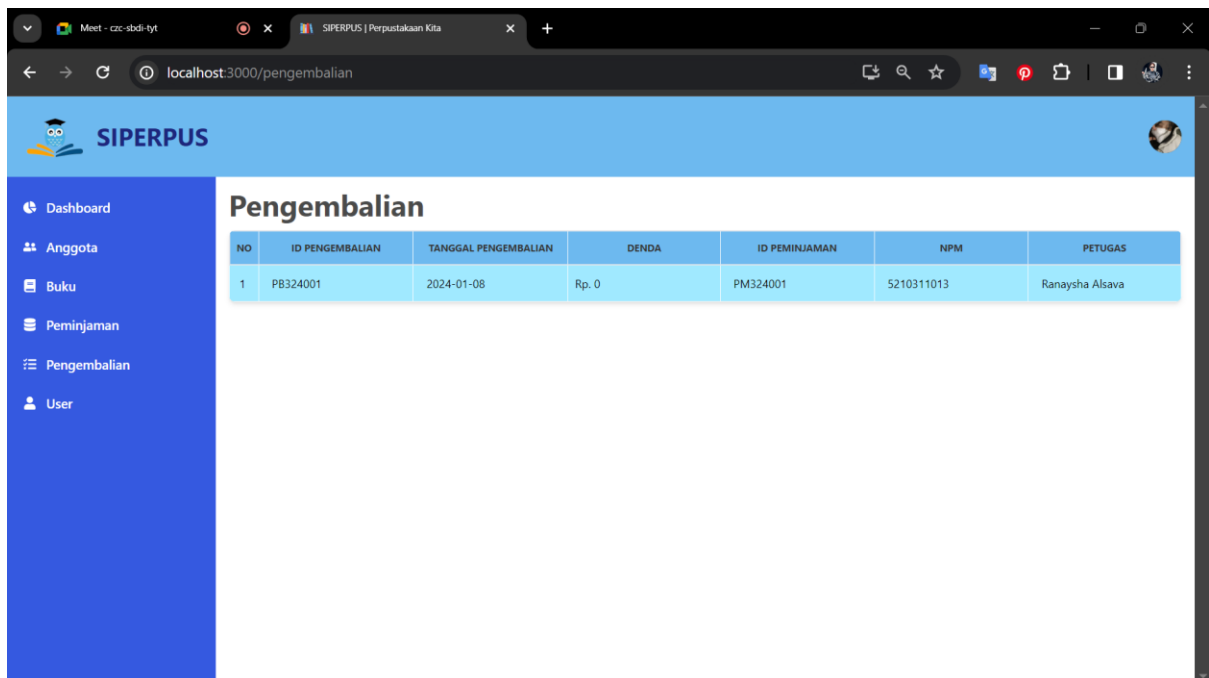
Status peminjaman yang mempengaruhi keterlambatan dan denda adalah cara untuk menentukan apakah buku telah dikembalikan tepat waktu atau tidak. Dalam konteks ini, beberapa status peminjaman yang umumnya digunakan adalah "Dipinjam" (belum dikembalikan) dan "Dikembalikan" (telah dikembalikan). Proses pengembalian buku yang dilakukan setelah batas waktu yang ditentukan dapat mengakibatkan keterlambatan, yang pada gilirannya dapat menimbulkan denda.



Pada action detail peminjaman menampilkan detail peminjaman yang terdiri dari id\_peminjaman, npm, status, tgl\_pinjam, tgl\_kembali, denda, dan juga buku yang dipinjam.



Jika status belum dikembalikan, maka akan terdapat button "Sudah Dikembalikan" untuk mengindikasikan bahwa item atau barang tersebut telah dikembalikan ke pemiliknya atau tempat asalnya. Dengan menekan tombol "Sudah Dikembalikan", sistem atau aplikasi dapat memperbarui status item tersebut menjadi "Sudah Dikembalikan"



Pada halaman pengembalian menampilkan data pengembalian. Terdapat id\_pengembalian, tanggal\_pengembalian, denda, id\_peminjaman, npm, dan petugas.

The screenshot shows a web browser window with the URL `localhost:3000/user`. The application is titled "SIPERPUS" and features a blue sidebar with the following menu items: Dashboard, Anggota, Buku, Peminjaman, Pengembalian, and User. The main content area displays a table titled "User" with the following data:

NO	NAME	ROLE	USERNAME
1	Marviza Arsaloka	admin	admin1
2	Ranaysya Alsava	petugas	petugas1

Pada halaman user menampilkan data user. Terdapat name, role, dan username.