**Laporan Proyek CRUD**

**1. Pendahuluan**

Proyek ini merupakan implementasi aplikasi CRUD (Create, Read, Update, Delete) menggunakan React untuk frontend dan Node.js dengan Express sebagai backend. Aplikasi ini memungkinkan pengguna untuk menambahkan, melihat, mengedit, dan menghapus data pengguna.

**2. Tujuan Proyek**

* **Create**: Menambahkan pengguna baru.
* **Read**: Menampilkan daftar pengguna.
* **Update**: Memperbarui informasi pengguna yang ada.
* **Delete**: Menghapus pengguna dari daftar.

**3. Arsitektur Sistem**

* **Frontend**:
  + **React.js**: Digunakan untuk membangun antarmuka pengguna.
  + **Axios**: Library HTTP client untuk berkomunikasi dengan server backend.
* **Backend**:
  + **Node.js**: Platform untuk menjalankan server JavaScript.
  + **Express.js**: Framework untuk membangun API RESTful.
  + **MySQL**: Database untuk menyimpan data pengguna.
  + **XAMPP**: Digunakan sebagai paket pengelola database dan server lokal.

**4. Deskripsi Komponen**

**4.1 Frontend**

* **UserForm.js**: Komponen formulir untuk menambahkan dan mengedit pengguna. Menggunakan state untuk menyimpan data input dan Axios untuk melakukan permintaan HTTP ke backend.
* **UserList.js**: Komponen untuk menampilkan daftar pengguna. Menggunakan Axios untuk mengambil data dari backend dan menangani operasi hapus.
* **App.js**: Komponen utama yang mengintegrasikan UserForm dan UserList. Mengelola state pengguna yang sedang diedit dan mengatur fungsi handleEdit dan handleSave.

**4.2 Backend**

* **index.js**: File utama server yang menyetel rute API untuk operasi CRUD (POST, GET, PUT, DELETE). Menggunakan Express untuk menangani permintaan HTTP dan MySQL untuk operasi database.

**5. Langkah-langkah Implementasi**

1. **Membuat Proyek React:**
   * Digunakan perintah npx create-react-app frontend untuk membuat aplikasi React baru.
2. **Menambahkan Axios:**
   * Diinstal menggunakan npm install axios untuk komunikasi dengan backend.
3. **Membuat Komponen React:**
   * UserForm.js untuk formulir pengguna.
   * UserList.js untuk menampilkan daftar pengguna.
   * App.js untuk mengintegrasikan komponen.
4. **Membangun Backend Node.js:**
   * Menyusun rute API dengan Express.
   * Mengatur koneksi ke database MySQL.
   * Mengimplementasikan operasi CRUD di backend.
5. **Pengujian Aplikasi:**
   * Menjalankan aplikasi React dengan npm start.
   * Memastikan backend Node.js berjalan.
   * Menguji fitur CRUD untuk memastikan semua operasi bekerja dengan baik.

**6. Hasil dan Penilaian**

* **Fungsi Create**: Data pengguna baru dapat ditambahkan ke database dan ditampilkan di antarmuka pengguna.
* **Fungsi Read**: Daftar pengguna ditampilkan dengan benar di antarmuka pengguna.
* **Fungsi Update**: Data pengguna yang ada dapat diperbarui dan perubahan terlihat di antarmuka pengguna.
* **Fungsi Delete**: Pengguna dapat dihapus dari daftar dan perubahan terlihat di antarmuka pengguna.

**7. Masalah dan Solusi**

* **Masalah Koneksi**: Terjadi kesalahan ECONNRESET saat menginstal dependensi. Solusi: Memastikan koneksi internet stabil dan membersihkan cache npm.
* **Data Tidak Muncul Setelah Save**: Setelah menyimpan data, daftar pengguna tidak diperbarui. Solusi: Memanggil fungsi fetchUsers setelah menyimpan data untuk memperbarui daftar pengguna.

**8. Kesimpulan**

Proyek ini berhasil diimplementasikan dengan semua fungsi CRUD berjalan dengan baik. Aplikasi memungkinkan pengguna untuk menambahkan, melihat, mengedit, dan menghapus data pengguna dengan antarmuka yang responsif.