

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Pemrograman Web
Kelas : 3IA11
Praktikum ke- : 7
Tanggal : 10/01/2024
Materi : JSON API
NPM : 51422161
Nama : MUHAMMAD TARMIDZI BARIQ
Ketua Asisten : Fadhil Muhammad
Jumlah Lembar : 7



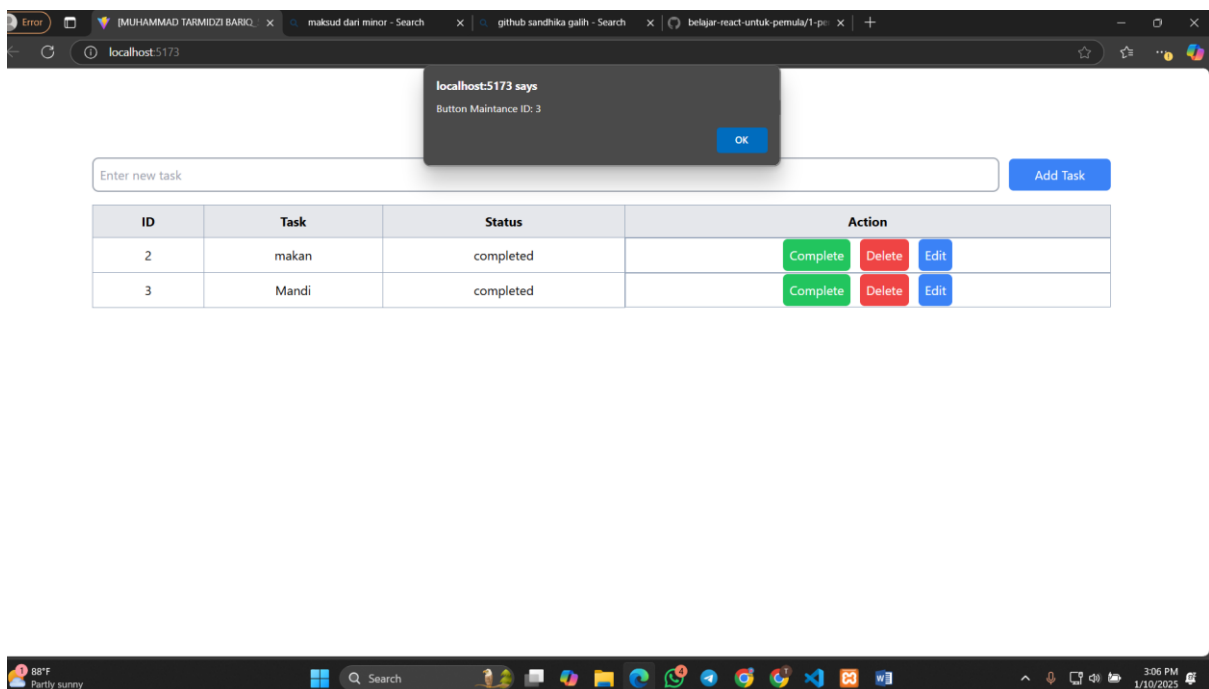
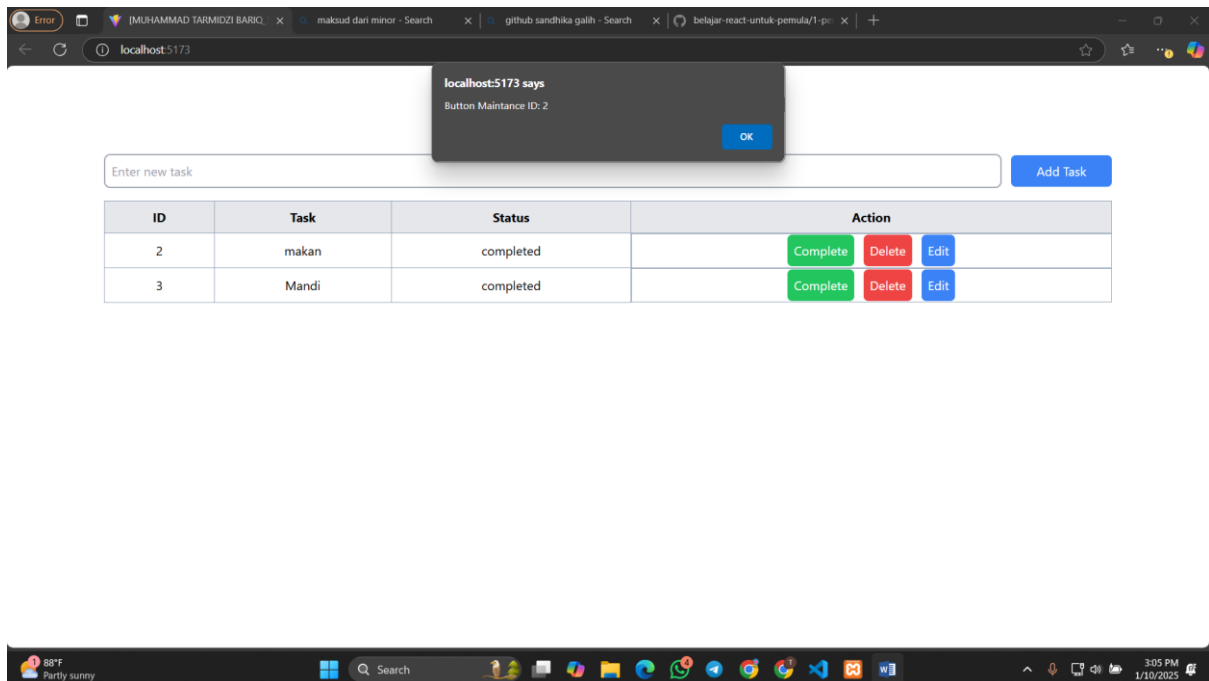
LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2024

Ubah secara minor tampilan website dan jelaskan ketiga file kode tersebut!

Menambahkan Tombol Edit dan muncul alert ketika Tombol Edit diklik dengan ID sesuai dengan data yang muncul



```

    </Button>
    <Button
      className={"bg-blue-500 w-fit"}
      onClick={() => editData(task.id)}
    >
      Edit
    </Button>
  </td>

```

```

// Implement the editData function
const editData = (taskId) => {
  // Implement your logic for editing tasks here
  alert(`Button Maintenance ID: ${taskId}`);
};

return (

```

1. TableTask.jsx

Isi file kode tersebut ialah komponen React bernama TableTask yang menampilkan tabel daftar tugas (tasks) dan memungkinkan pengguna untuk melakukan beberapa aksi terhadap tugas-tugas tersebut

- Menampilkan data tugas dari API dalam bentuk tabel.
- Memberikan opsi mengubah status tugas (menjadi "completed"), menghapus tugas, atau mengedit tugas.

Mengambil data dari API:

Menggunakan useEffect dan axios untuk mem-fetch data dari API (<http://localhost:8000>) dan menyimpan datanya dalam state tasks.

Aksi pada tugas:

updateTask(taskId):

Mengubah status tugas menjadi "completed" dengan metode HTTP PUT.

Memperbarui status dalam UI.

deleteTask(taskId):

Menghapus tugas dari server menggunakan HTTP DELETE.

Menghapus tugas tersebut dari state tasks agar hilang dari tabel.

editData(taskId):

Menampilkan pesan alert sementara untuk tombol edit.

FormTask.jsx

File kode tersebut ialah komponen React bernama FormTask yang menyediakan form untuk menambahkan data kesistem.

```
export const FormTask = () => {  
  const [task, setTask] = useState({  
    title: "",  
  });  
  
  const [isSubmitting, setIsSubmitting] = useState(false); // untuk mengelola loading state  
  const [error, setError] = useState(null); // untuk mengelola error
```

- useState digunakan untuk membuat dan mengelola state dalam komponen.
- task menyimpan informasi tugas (saat ini hanya title).
- isSubmitting melacak apakah form sedang dalam proses pengiriman.
- error menyimpan pesan error yang terkait dengan pengiriman form.

```
const handleAddTasks = async (e) => {  
  e.preventDefault();  
  
  if (!task.title) {  
    setError("Task title is required!");  
    return;  
  }  
  
  setIsSubmitting(true); // mulai proses submit  
  setError(null); // reset error  
  
  try {  
    const res = await fetch("http://localhost:8000", {  
      method: "POST",  
      headers: {  
        "Content-Type": "application/json",
```

```

    },
    body: JSON.stringify({ task: task.title }),
  });

  if (!res.ok) {
    throw new Error("Failed to add task");
  }

  const data = await res.json();
  window.location.reload();
  console.log(data);

  setTask({ title: "" }); // reset input form setelah sukses
} catch (err) {
  console.error(err);
  setError(err.message || "An error occurred while adding the task.");
} finally {
  setIsSubmitting(false); // selesai proses submit
}
};

```

- handleAddTasks adalah fungsi asinkron yang menangani pengiriman form.
- e.preventDefault() mencegah perilaku default pengiriman form.
- Jika title tugas kosong, error diset dan fungsi keluar lebih awal.
- Status loading form diset menjadi true dan error sebelumnya direset.
- Data tugas dikirim ke server melalui permintaan POST ke http://localhost:8000.
- Jika permintaan gagal, error dilemparkan.
- Jika permintaan berhasil, halaman direfresh dan input tugas direset.
- Setiap error ditangkap dan ditampilkan, serta status loading direset.

```

return (
  <form
    onSubmit={handleAddTasks}
    id="createTaskForm"
    className="mb-4 flex gap-3 items-center"
  >
    <input
      className="p-2 border-2 border-gray-400 rounded-lg flex-grow"
      type="text"
      id="taskInput"
      value={task.title}
      placeholder="Enter new task"
      onChange={(e) => setTask({ title: e.target.value })}
      required
    />
  </form>
)

```

```

    />

    <Button
      className="bg-blue-500 text-white p-2 w-32 rounded-md"
      type="submit"
      disabled={isSubmitting}
    >
      {isSubmitting ? "Adding..." : "Add Task"}
    </Button>

    { /* Menampilkan error jika ada */ }
    {error && <p className="text-red-500 text-sm ml-2">{error}</p>}}
  </form>
);

```

- Form ini mencakup field input untuk title tugas dan tombol submit.
- Field input terhubung dengan state task.title dan memperbaruinya saat ada perubahan.
- Tombol submit dinonaktifkan saat form sedang dikirim.
- Jika ada error, error tersebut ditampilkan di bawah form.

Button.jsx

Kode ini mendefinisikan sebuah komponen Button yang dapat digunakan kembali dalam React.

```

const Button = ({ className, children, ...props }) => {
  return (
    <button
      {...props}
      className={`bg-blue-500 text-white p-2 w-32 rounded-md ${className}`}
    >
      {children}
    </button>
  );
};

export default Button;

```

