

## ACTIVITY PERTEMUAN 7

NAMA : Muhammad Tarmidzi Bariq

NPM : 51422161

KELAS : 3IA11

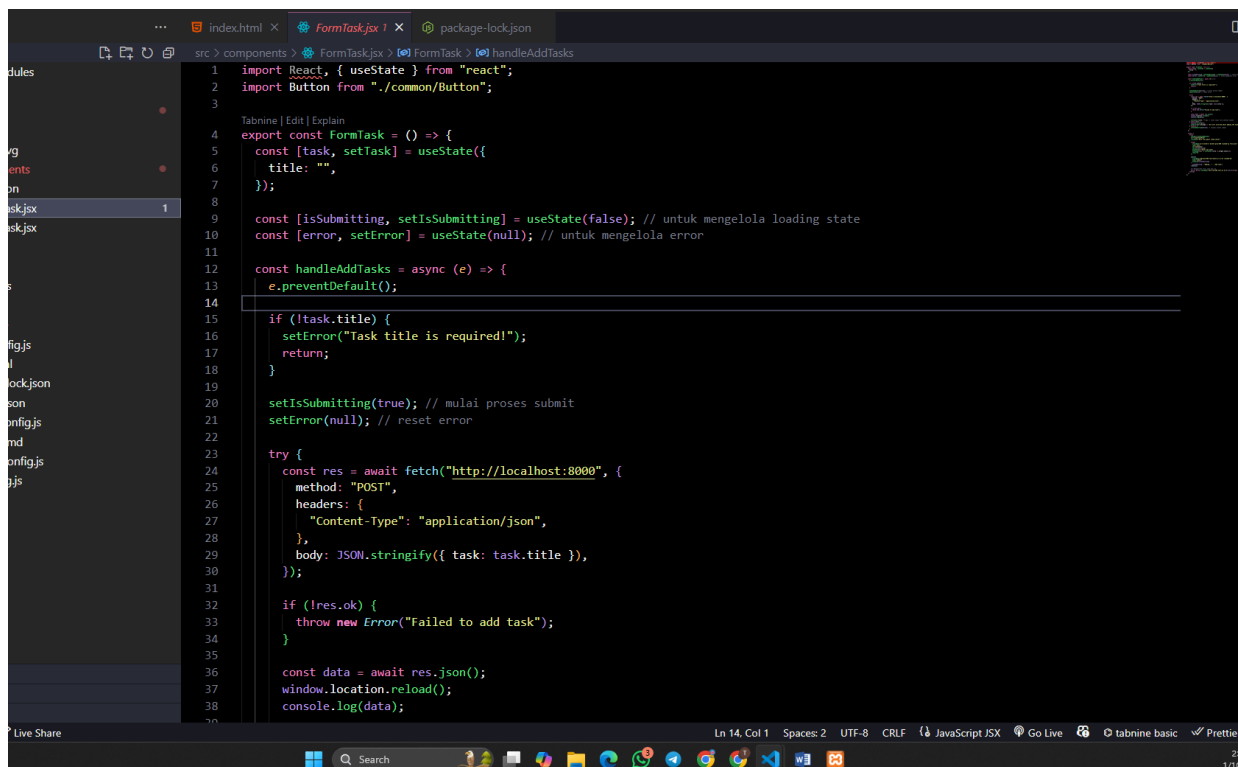
MATERI : PENGENALAN JSON API

MATA PRAKTIKUM : PEMOGRAMAN WEBSITE

---

(Screenshoot langkah-langkah sesuai video pembelajaran dan jelaskan dengan ringkas)

FormTask.jsx



```
1 import React, { useState } from "react";
2 import Button from "../common/Button";
3
4 export const FormTask = () => {
5   const [task, setTask] = useState({
6     title: "",
7   });
8
9   const [isSubmitting, setIsSubmitting] = useState(false); // untuk mengelola loading state
10  const [error, setError] = useState(null); // untuk mengelola error
11
12  const handleAddTasks = async (e) => {
13    e.preventDefault();
14
15    if (!task.title) {
16      setError("Task title is required!");
17      return;
18    }
19
20    setIsSubmitting(true); // mulai proses submit
21    setError(null); // reset error
22
23    try {
24      const res = await fetch("http://localhost:8000", {
25        method: "POST",
26        headers: {
27          "Content-Type": "application/json",
28        },
29        body: JSON.stringify({ task: task.title }),
30      });
31
32      if (!res.ok) {
33        throw new Error("Failed to add task");
34      }
35
36      const data = await res.json();
37      window.location.reload();
38      console.log(data);
39    }
40  }
41}
```

```
File Edit Selection View Go Run Terminal Help
< >
pert7.coding
EXPLOER
... index.html FormTask.jsx package-lock.json
PERT7 CODING
node_modules
public
src
assets
react.svg
components
FormTask.jsx
TableTask.jsx
App.css
App.jsx
index.css
main.jsx
.gitignore
eslint.config.js
index.html
package-lock.json
package.json
postcss.config.js
README.md
tailwind.config.js
vite.config.js
OUTLINE
TIMELINE
NPM SCRIPTS
Live Share
FormTask.jsx
4 export const FormTask = () => {
12 const handleAddTasks = async (e) => {
38 console.log(data);
39
40 setTask({ title: "" }); // reset input form setelah sukses
41 } catch (err) {
42 console.error(err);
43 setError(err.message || "An error occurred while adding the task.");
44 } finally {
45 setIsSubmitting(false); // selesai proses submit
46 }
47 };
48
49 return (
50 <form
51   onSubmit={handleAddTasks}
52   id="createTaskForm"
53   className="mb-4 flex gap-3 items-center"
54 >
55   <input
56     className="p-2 border-2 border-gray-400 rounded-lg flex-grow"
57     type="text"
58     id="taskInput"
59     value={task.title}
60     placeholder="Enter new task"
61     onChange={(e) => setTask({ title: e.target.value })}
62     required
63   />
64
65   <button
66     className="bg-blue-500 text-white p-2 w-32 rounded-md"
67     type="submit"
68     disabled={isSubmitting}
69   >
70     {isSubmitting ? "Adding..." : "Add Task"}
71   </button>
72
73   { /* Menampilkan error jika ada */ }
74   {error && <p className="text-red-500 text-sm ml-2">{error}</p>}
75 </form>
76 );
77 };
78
```

```
73 { /* Menampilkan error jika ada */ }
74 {error && <p className="text-red-500 text-sm ml-2">{error}</p>}
75 </form>
76 );
77 };
78
```

## TableTask.jsx

```
src > components > TableTask.jsx > ...
1 import React, { useEffect, useState } from "react";
2 import Button from "../common/Button";
3 import axios from "axios";
4
5 export const TableTask = () => {
6   const [tasks, setTasks] = useState([]);
7
8   // Mendapatkan data dari API dan menyimpannya dalam state
9   useEffect(() => {
10     const getTasks = async () => {
11       try {
12         const res = await axios.get("http://localhost:8000"); // pastikan endpoint sesuai
13         setTasks(res.data);
14         console.log(res.data);
15       } catch (error) {
16         console.error("Failed to fetch tasks:", error);
17       }
18     };
19     getTasks();
20   }, []);
21
22   // Fungsi untuk memperbarui task menjadi complete
23   const updateTask = async (taskId) => {
24     try {
25       const res = await fetch("http://localhost:8000", {
26         method: "PUT",
27         headers: {
28           "Content-Type": "application/json",
29         },
30         body: JSON.stringify({ id: taskId }), // update status
31       });
32       const data = await res.json();
33       window.location.reload();
34       console.log(data);
35
36       setTasks((prevTasks) =>
37         prevTasks.map((task) =>
38           task.id === taskId ? { ...task, status: "completed" } : task
39         )
40       );
41     } catch (error) {
42       console.error("Failed to update task:", error);
43     }
44   };
45
46   // Fungsi untuk menghapus task
47   const deleteTask = async (taskId) => {
48     try {
49       const res = await fetch("http://localhost:8000", {
50         method: "DELETE",
51         body: JSON.stringify({ id: taskId }),
52       });
53       const data = await res.json();
54       console.log(data);
55       setTasks((prevTasks) => prevTasks.filter((task) => task.id !== taskId));
56     } catch (error) {
57       console.error("Failed to delete task:", error);
58     }
59   };
60
61   return (
62     <table className="w-full table-auto border-collapse">
63       <thead className="bg-gray-200">
64         <tr className="text-center">
65           <th className="border px-4 py-2 border-slate-400">ID</th>
66           <th className="border px-4 py-2 border-slate-400">Task</th>
67           <th className="border px-4 py-2 border-slate-400">Status</th>
68           <th className="border px-4 py-2 border-slate-400">Action</th>
69         </tr>
70       </thead>
71       <tbody>
72         {tasks.length > 0 ? (
73           tasks.map((task) => (
74             <tr className="text-center" key={task.id}>
```

```
5 export const TableTask = () => {
23   const updateTask = async (taskId) => {
36     setTasks((prevTasks) =>
37       prevTasks.map((task) =>
38         task.id === taskId ? { ...task, status: "completed" } : task
39       )
40     );
41   } catch (error) {
42     console.error("Failed to update task:", error);
43   }
44 };
45
46 // Fungsi untuk menghapus task
47 const deleteTask = async (taskId) => {
48   try {
49     const res = await fetch("http://localhost:8000", {
50       method: "DELETE",
51       body: JSON.stringify({ id: taskId }),
52     });
53     const data = await res.json();
54     console.log(data);
55     setTasks((prevTasks) => prevTasks.filter((task) => task.id !== taskId));
56   } catch (error) {
57     console.error("Failed to delete task:", error);
58   }
59 };
60
61 return (
62   <table className="w-full table-auto border-collapse">
63     <thead className="bg-gray-200">
64       <tr className="text-center">
65         <th className="border px-4 py-2 border-slate-400">ID</th>
66         <th className="border px-4 py-2 border-slate-400">Task</th>
67         <th className="border px-4 py-2 border-slate-400">Status</th>
68         <th className="border px-4 py-2 border-slate-400">Action</th>
69       </tr>
70     </thead>
71     <tbody>
72       {tasks.length > 0 ? (
73         tasks.map((task) => (
74           <tr className="text-center" key={task.id}>
```

```
terminal Help ← → per7-coding
... index.html TableTask.jsx 1 x package-lock.json
src > components > TableTask.jsx > ...
5 export const TableTask = () => {
63   <thead className="bg-gray-200">
64     <tr className="text-center">
65       <th className="border px-4 py-2 border-slate-400">ID</th>
66       <th className="border px-4 py-2 border-slate-400">Task</th>
67       <th className="border px-4 py-2 border-slate-400">Status</th>
68       <th className="border px-4 py-2 border-slate-400">Action</th>
69     </tr>
70   </thead>
71   <tbody>
72     {tasks.length > 0 ? (
73       tasks.map((task) => (
74         <tr className="text-center" key={task.id}>
75           <td className="border px-4 py-2 border-slate-400">{task.id}</td>
76           <td className="border px-4 py-2 border-slate-400">{task.task}</td>
77           <td className="border px-4 py-2 border-slate-400">
78             {task.completed ? "completed" : "incomplete"}
79           </td>
80           <td className="flex gap-3 justify-center border border-slate-400 h-full">
81             <Button
82               className={"bg-green-500 w-fit"}
83               onClick={() => updateTask(task.id)}
84             >
85               Complete
86             </Button>
87             <Button
88               className={"bg-red-500 w-fit"}
89               onClick={() => deleteTask(task.id)}
90             >
91               Delete
92             </Button>
93           </td>
94         </tr>
95       )
96     ) : (
97       <tr>
98         <td
99           colSpan="4"
100           className="text-center py-4 border border-slate-400"
101         >
102           No tasks available
103         </td>
104       </tr>
105     )
106   </tbody>
107 </table>
108 );
109 };
```

```
99   <td
100     colSpan="4"
101     className="text-center py-4 border border-slate-400"
102   >
103     No tasks available
104   </td>
105 </tr>
106 </tbody>
107 </table>
108 );
109 };
```

## Button.jsx

```
1 const Button = ({ className, children, ...props }) => {
2   return (
3     <button
4       {...props}
5       className={`bg-blue-500 text-white p-2 w-32 rounded-md ${className}`}
6     >
7       {children}
8     </button>
9   );
10 };
11
12 export default Button;
13
```

## HASIL

