LEMBAR ANALISIS

Praktikum Rekayasa Perangkat Lunak Berbasis Komponen

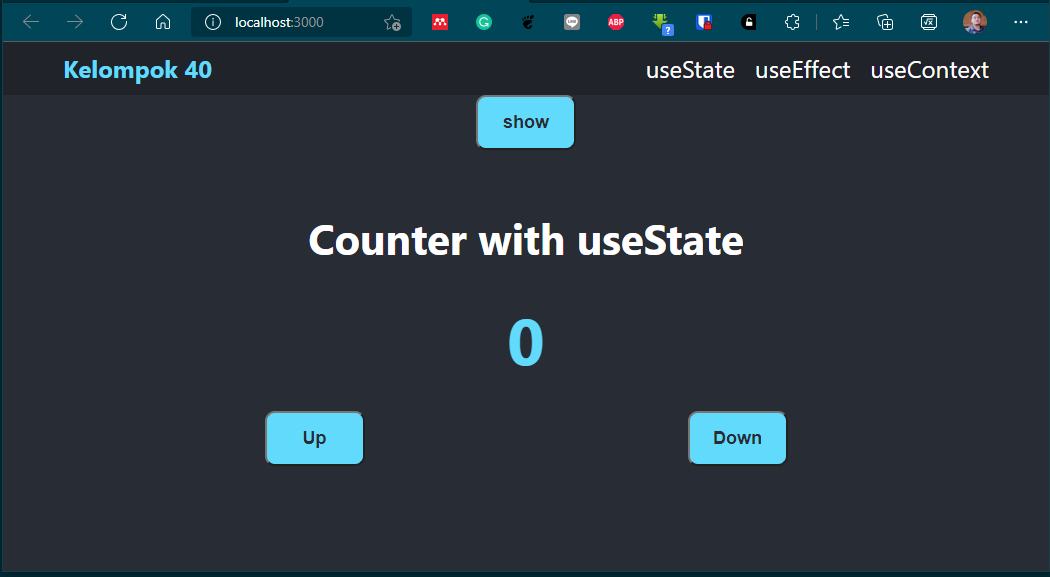
Modul 4

React Hooks

|  |  |
| --- | --- |
| Nama (NIM) Praktikan | Kelompok 40 |
| Firsta Adi Pradana (21120118130059) |

Hasil Analisis

Pada praktikum modul 4 RPLBK yang dilaksanakan pada hari Sabtu, 11 September 2021, praktikan mencoba mempraktikakan tentang useEffect, useState, dan useContext pada React. Berikut adalah hasil dari percobaan tersebut:



Pada percobaan pertama, kami menggunakan useState. useState digunakan untuk memudahkan dalam mengatur nilai variable dalam component tersebut. Misalnya untuk mengatur nilai count, digunakan useState(0) sehingga default nilai dari count adalah 0. Kemudian dibuat funsi countUp dan countDown. Saat countUp dijalankan, maka nilai count sebelumnya akan bertambah 1, dan apabila countDown dijalankan, maka nilai count sebelumnya akan berkurang 1. Source codenya adalah sebagai berikut:

import { useState } from "react";

import "./State.css";

export default *function* Index() {

*const* [show, setShow] = useState(false);

*const* [count, setCount] = useState(0);

*const* [user, setUser] = useState("");

*const* [password, setPassword] = useState("");

*const* countUp = () => setCount(count + 1);

*const* countDown = () => setCount(count - 1);

  return (

    <div *className*="container">

      {show && (

        <>

          <div *className*="titleWrapper">

            <p *className*="title">input with useState</p>

          </div>

          <div *className*="inputWrapper">

            <div>

              <input

*placeHolder*="name"

*size*="50"

*value*={user}

*onChange*={(*event*) => setUser(event.target.value)}

              />

            </div>

            <div>

              <input

*placeholder*="password"

*size*="50"

*value*={password}

*onChange*={(*event*) => setPassword(event.target.value)}

              />

            </div>

          </div>

        </>

      )}

      <button *className*="Button" *onClick*={() => setShow(!show)}>

        {show ? "hide" : "show"}

      </button>

      <div *style*={{ marginBlock: 30 }} />

      <p *className*="title">Counter with useState</p>

      <p *className*="countValue">{count}</p>

      <div *className*="btnWrapper">

        <button *className*="Button" *onClick*={countUp}>

          Up

        </button>

        <button *className*="Button" *onClick*={countDown}>

          Down

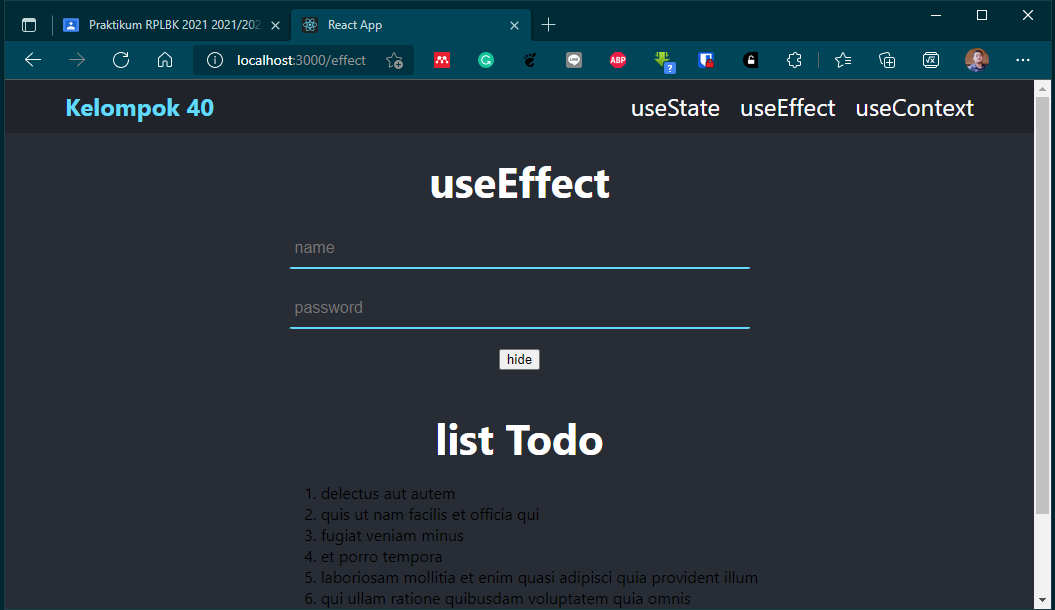
        </button>

      </div>

    </div>

  );

}



useEffect pada percobaan kedua digunakan untuk melakukan fetch dari suatu url. Response dari url tersebut akan dijadikan json, kemudian dengan useState digunakan untuk mengubah nilai dari data. Kemudian data tersebut ditampilkan sebagai list Todo. Fungsinya mirip dengan componentDidUpdate, contohnya untuk useEffect pada input password, akan dijalankan tiap ada update pada *component* .Berikut adalah source codenya:

import { useState, useEffect } from "react";

export default *function* Index() {

*const* [show, setShow] = useState(false);

*const* [valueInput, setValueInput] = useState({

    user: "",

    password: "",

  });

*const* [data, setData] = useState([]);

  useEffect(() => {

    fetch("https://jsonplaceholder.typicode.com/todos")

      .then((*response*) => response.json())

      .then((*data*) => {

        console.log("[ini data]", data);

        setData(data);

      })

      .catch((*err*) => {

        console.log(err);

      });

  }, []);

  useEffect(() => {

    console.log("\*[spam console]");

  });

  useEffect(() => {

    if (valueInput.password !== "") {

      alert(`component will update & value ${valueInput.password}`);

    }

  }, [valueInput.password]);

*const* inputHandler = (*event*, *type*) => {

    if (type === "user") {

      setValueInput({ ...valueInput, user: event.target.value });

    } else {

      setValueInput({ ...valueInput, password: event.target.value });

    }

  };

  return (

    <div *className*="container">

      {show && (

        <>

          <div *className*="titleWrapper">

            <p *className*="title">useEffect</p>

          </div>

          <div *className*="inputWrapper">

            <div>

              <input

*placeholder*="name"

*size*="50"

*value*={valueInput.user}

*onChange*={(*event*) => inputHandler(event, "user")}

              />

            </div>

            <div>

              <input

*placeholder*="password"

*size*="50"

*value*={valueInput.password}

*onChange*={(*event*) => inputHandler(event, "password")}

              />

            </div>

          </div>

        </>

      )}

      <button *className*="button" *onClick*={() => setShow(!show)}>

        {show ? "hide" : "show"}

      </button>

      <p *className*="title" *style*={{ marginTop: 40 }}>

        list Todo

      </p>

      <ol>

        {data.slice(0, 10).map((*value*) => (

          <li *key*={value.id}>{value.title}</li>

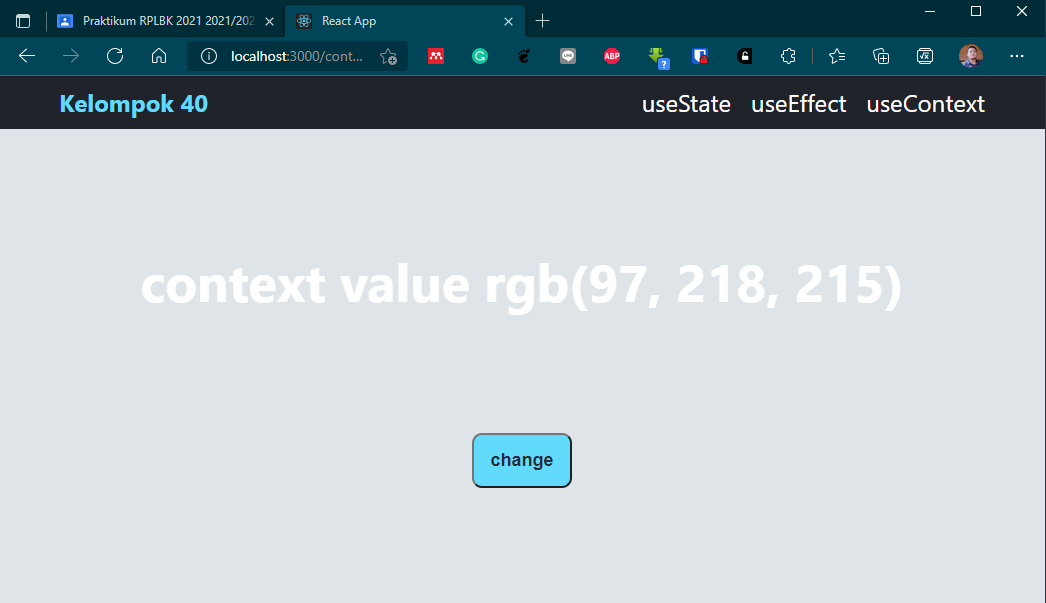
        ))}

      </ol>

    </div>

  );

}



Percobaan ketiga adalah untuk useContext(). Use context ini digunakan untuk mengoper data tanpa melakukan oper props secara manual. Pada percobaan, digunakan untuk mengoper nilai dari theme pada ThemeContext ke *component* Text yang akan dioper dan ditampilkan pada *component* Content. Berikut adalah source codenya(sudah diubah berdasarkan challenge tapi belum sempat njelasin dah abis waktunya):

import { useContext, createContext, useState } from "react";

import "./Context.css";

*const* themes = {

  light: {

    text: "rgb(97, 218, 215)",

    backgroud: "#DEE4E7",

  },

  dark: {

    text: "#DEE4E7",

    background: "rgb(40, 44, 52)",

  },

};

*const* ThemeContext = createContext();

export default *function* Context() {

*const* [valueTheme, setValueTheme] = useState(themes.dark);

  return (

    <ThemeContext.Provider *value*={valueTheme}>

      <div *className*="contentWrapper" *style*={{ backgroundColor: valueTheme.backgroud}}>

        <Content />

        <button

*className*="Button"

*onClick*={() => setValueTheme( valueTheme === themes.light ? themes.dark : themes.light)}

        >

          change

        </button>

      </div>

    </ThemeContext.Provider>

  );

}

*function* Content(*props*) {

  return (

    <div>

      <Text />

    </div>

  );

}

*function* Text(*props*) {

*const* theme = useContext(ThemeContext);

  console.log("[context value]", theme);

  console.log(theme.text)

  return <p *className*="titleContext">context value {theme.text}</p>;

}

|  |
| --- |
| Firsta Adi Pradana  21120118130059 |
|  |