|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| React JS Semester-4 Minggu ke-9 Video 21-28 | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Mengimport css bootstrap dari react-bootstrap.github.io ke dalam vs code yang nantinya digunakan secara online.

2. Di dalam file react dapat diketikkan code rfc lalu enter yang nanti akan otomatis terbentuk seperti ini

import React from 'react';

const Back = () => {

    return (

        <div>

        </div>

    );

}

export default Back;

nama const diambil sesuai dengan nama file nya, misalkan file nya bernama Back.js maka const nya akan diberi nama Back.

3. File yang digunakan untuk mengatur tampilan dari browser React nya adalah file App.js di dalam file ini biasanya mengimport dari file lainnya dengan menuliskan import nama front nya lalu from “./namafolder/namafile” setelah itu untuk ditampilkan di browser pada function app nya dituliskan <NamaConst />

4. Semua tampilan yang ada di browser berasal dari App.js dengan url yang sama hanya localhost:3000, untuk itu apabila url dibagi agar bagian front end dan back end dalam url yang berbeda maka harus diberikan router dan route nya dulu. Di dalam file App.js diketikkan import { BrowserRouter as Router, Route } from "react-router-dom"; yang berarti dari file ini bisa diberikan sebuah Router dan Route. Contohnya di dalam function App nya diberikan open Router dan close Router nya <Router></Router> lalu Router ini diisi dengan Route untuk pengaturan setiap route nya misalkan <Route path=”/admin” component={Back} /> yang berarti apabila url nya diketikkan localhost:3000 dengan tambahan dari path yaitu /admin maka yang akan muncul di browser nya adalah dari page Back nya. Apabila di dalam Route ada tambahan exact berarti Route itu khusus sesuai path nya, contohnya Route pertama path nya “/” lalu diberi exact dan Route kedua path nya “/home” tanpa diberi exact dan path milik Route pertama ada / nya dimana di path Route kedua juga ada. Agar tampilan dari Route pertama tidak keluar dalam Route kedua maka ditambahkan exact yang mengkhususkan Route itu tampilannya hanya akan keluar sesuai path nya.

5. Nested Route artinya Route bersarang, jadi misalkan ada sebuah url seperti ini localhost:3000/admin maka sarangnya adalah admin dan di dalam admin dibagi menjadi beberapa page lagi misalkan page kategori, menu, dll. Untuk masuk ke dalam page yang ada di dalam admin maka url nya ditambah menjadi /kategori seperti ini localhost:3000/admin/kategori jadi yang dimaksud kategori adalah kategori milik admin nya bukan hanya /kategori tanpa /admin. Untuk pengetikkannya, pada file yang digunakan untuk mengatur sebuah link ditambahkan import seperti ini dulu import { Link, useRouteMatch } from "react-router-dom"; lalu di dalam const nya ditambahkan const { url, path } = useRouteMatch(); yang berarti const url dan path adalah const yang menyimpan useRouteMatch dari import tadi. Setelah itu pada bagian element yang diberikan link seperti ini <Link></Link> pada Link open nya dijadikan seperti ini misalkan menambahkan /kategori maka <Link to={`${url}/kategori`}> ini berarti url atau link yang ada di browser diisi dengan ${url} atau link awal yang sudah ada disitu sebelumnya lalu ditambahi dengan /kategori. Jadi misalkan user sudah ada di admin page dengan url awal localhost:3000/admin maka ketika menekan element yang memiliki link seperti itu url nya akan tertambah /kategori menjadi seperti ini localhost:3000/admin/kategori.

6. Bagian path contohnya dibuatkan sebuah const dulu seperti ini const { path } = useRouteMatch(); jadi const path ini menggunakan useRouteMatch lalu di dalam return ada

<Switch>

    <Route path={`${path}/:isi`}>

     <Content />

    </Route>

   </Switch>

Yang ketika diketikkan switch dan route nya dia akan otomatis mengisi import

import { useRouteMatch } from "react-router-dom"; menjadi import { Route, Switch, useRouteMatch } from "react-router-dom";

setelah itu pada bagian Route path={``$path/:isi} yang berarti isi ini adalah sebuah variable yang menyimpan data setelah / sesuai dengan url yang dipilih misalkan url awal localhost:3000/admin lalu ketika menekan element kategori maka url nya menjadi localhost:3000/admin/kategori maka variable isi akan terisi dengan kategori. Setelah itu di file lain isi ini bisa digunakan dengan cara dibuatkan ini dulu const { isi } = useParams(); jadi isi ini bisa digunakan di file ini, misalkan <h1>{isi}</h1> maka h1 ini akan tampil sesuai dengan isi nya yang apabila pada url localhost:3000/admin/kategori maka isi nya kategori dan apabila url nya localhost:3000/admin/menu maka isi nya menu.

7. Mengetikkan console.log dengan cepat yaitu dengan mengetikkan clg lalu enter maka akan langsung otomatis tertera console.log(object).

8. Mengetikkan imr lalu enter akan otomatis muncul code import react from ‘react’.

9. Untuk menampilkan sebuah data yang ada di dalam database ke dalam browser nya react dengan memilih kolom yang akan dipilih, jadi tidak semua data nya akan ditampilkan. Caranya dengan menggunakan axios, file axios yang sebelumnya sudah dibuat pada folder layout dengan nama file link.js bisa dipindah ke folder axios yang ada di src react-resto, lalu di dalam file yang akan digunakan untuk menampilkan data misalnya data kategori berarti di file Kategori.js menggunakan useEffect yang didalamnya ada function async dan di dalam functionnya ada const request. Const request ini berisi await link.get(‘./kategori’) yang berarti nanti axios akan menjalankan link nya kategori dengan method get lalu dibuatkan setIsi dimana setIsi ini adalah useState yang digunakan untuk menyimpan ini setIsi(request.data); yang berarti diambil data yang ada di kategori tadi. Lalu functionnya ditampilkan dan masih di dalam useEffect. Kemudian pada return nya adalah tempat untuk dibuatkan sebuah tabel dan pemanggilan data nya seperti ini

<table>

    <tr>

     <th>Kategori</th>

     <th>Keterangan</th>

    </tr>

    {isi.map((val, index) => (

     <tr key={index}>

      <td>{val.kategori}</td>

      <td>{val.keterangan}</td>

     </tr>

    ))}

   </table>

Ada dua kolom yaitu kategori dan keterangan, dan untuk barisnya diisi dengan isi.map yang dibagi menjadi val dan index, index nya digunakan untuk key dan val nya digunakan untuk td nya. Jadi data yang diambil berdasarkan urutan index nya dan val nya untuk memilih kolom mana yang akan digunakan misalnya val kategori maka yang diambil adalah data dari kolom kategorinya.

10. Membuat sebuah form dengan menggunakan function register dari react-hook-form. Pertama-tama yaitu mengetikkan import {useForm} from “react-hook-from” setelah itu buat sebuah form dulu dan disetiap inputnya ditambahkan sebuah ref={register({ required: true })} yang berarti dia menerima data yang dimasukkan oleh user, lalu pada code form nya ditambahkan code menjadi seperti ini <form onSubmit={handleSubmit(simpan)}> yang berarti apabila form sudah disubmit maka dia akan menjalankan function bernama simpan atau handleSubmit nya mengambil data dan diarahkan ke function simpan. Function ini akan menyimpan data yang dimasukkan user dan function ini sebagai tempat pengaturan data nya akan diapakan.

11. Untuk mengatur errors pada form misalkan pada bagian input kategori harus diisi dan tidak boleh kosong, maka ditambahkan {errors.kategori && "Kategori harus diisi"} yang berarti apabila input bernam kategori kosong maka akan muncul pesan “Kategori harus diisi” dibawah inputnya. Agar form, error, dll berjalan maka harus ditambahkan const seperti ini const { register, handleSubmit, reset, errors } = useForm();

12. Untuk mengepost data yang dimasukkan dari form tadi, caranya pada bagian function simpan atau function yang digunakan untuk mengatur data ini akan diapakan diisi dengan link.post(‘/kategori’, data).then(res=>console.log(res)); yang berarti data ini akan dipost ke dalam kategorinya sesuai dengan routes di api\_lumen waktu itu dengan adanya penambahan /kategori pada link bermethod post maka dia akan mempost datanya lalu ada res yang berarti respon nya akan muncul pada console.log apabila proses post berhasil.

13. Menghapus sebuah data dari tabel yang sudah tampil di browser react nya. Caranya dengan menambahkan kolom baru untuk tempat button hapus nya lalu mengatur button agar ketika ditekan dia mengambil id dan mengarahkan ke function hapus seperti ini

<button onClick={() => hapus(val.idkategori)} className="btn btn-danger"> Hapus </button>

Jadi ketika dia diklik dia akan mengarahkan ke function bernama hapus dan mengirimkan data idkategori dari data yang dipilih untuk dihapus. Lalu di dalam function hapus nya parameter diisi dengan id yang didapat dari button tadi lalu isi functionnya seperti ini

if (window.confirm("Yakin akan menghapus?")) {

   const res = await link.delete("/kategori/" + id);

   setPesan(res.data.pesan);

  }

If(window.confirm) merupakan sebuah konfirmasi setelah menekan button hapus akan muncul seperti notifikasi yang berfungsi untuk mengkonfirmasi dengan tulisan “Yakin akan menghapus?” const res memiliki isi untuk menyambungkan dengan link dan method nya delete dan memilih bagian kategori beserta id nya. Jika proses hapus berhasil maka setPesan akan dijalankan yaitu munculnya pesan yang ada dibawah judul Data Kategori.

14. Pada bagian showdata menggunakan setValue yang berfungsi untuk mengambil isi sesuai kolom yang ditentukan dari sebuah data yang dipilih untuk diletakkan ke dalam form nya misalkan setValue(‘kategori’, res.data[0].kategori); yang berarti form kategori akan diisi dengan data dari kolom kategori yang diambil dari data yang dipilih untuk diambil data kategorinya.

15. Pada bagian sebelumnya sudah dilakukan proses pengambilan data dan diletakkan ke dalam form nya yang nantinya akan diupdate. Untuk melakukan Update tinggal mengatur pada bagian submitnya saja yaitu ada di function simpan. Di dalam function simpan diisi dengan sebuah if seperti ini

if (pilihan) {

   link.post("/kategori", data).then((res) => setPesan(res.data.pesan));

  } else {

   link

    .put("/kategori/" + idkategori, data)

    .then((res) => setPesan(res.data.pesan));

   setPilihan(true);

  }

  reset();

  fetchData();

yang berarti apabila pilihan nya true maka dia akan menjalankan method post atau simpan ketika menambahkan sebuah data. Dan apabila pilihan false atau tidak true maka dia akan menjalankan method put atau update sesuai dengan idkategori dan data yang ada lalu ketika sudah melakukan update akan otomatis pilihannya kembali ke true lagi. Pada bagian sebelumnya sudah diatur untuk pilihan apakah true atau false, jadi di bagian atas ditambahkan const [pilihan, setPilihan] = useState(true); dulu yang berarti pilihan ini awalnya adalah true, jadi apabila form belum digunakan maka status pilihannya adalah true lalu pada bagian function showData atau letaknya ubah ada setPilihan(false); karena itu ketika menekan button ubah dan data nya tampil di form maka status pilihannya adalah false. Dari sini didapatkan adanya pilihan true dan false yang digunakan untuk menentukan tindakan apa yang akan diambil oleh submit sesuai dengan tujuannya.

**Saya Belum Mengerti**

1.