|  |  |
| --- | --- |
| Materi | Nilai |
| REACT-JS-V21-30 | **98** |

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

1. Saya sudah bisa menginstal React JS dengan menggunakan yarn, yaitu dengan mengetikkan kode berikut di dalam cmd kemudian enter dan tunggu hasilnya,

yarn create react-app my-app

1. Saya sudah mengerti bahwa kita harus menyalakan yarn tersebut terlebih dahulu saat ingin menjalankn react js, karena fungsinya sama seperti XAMPP atau sebagai localhost.
2. Saya sudah bisa menyalakan node js dengan menggunakan yarn, yaitu dengan mengetikkan di cmd berupa perintah **yarn start**. Lalu klik enter dan tunggu hingga browser membuka halaman baru.
3. Saya sudah mengerti bahwa untuk menggunakan bootstrap yang sudah kita install kita harus mengimportnya terlebih dahulu menggunakan kode.

import 'bootstrap/dist/css/bootstrap.min.css';

1. Setelah kita memasukkan kode import bootstrap tersebut dan lalu kita membuka browser dan font atau pun posisi element yang kita buat sebelumnya sudah berubah itu artinya kita sudah berhasil mengambil css dari bootstrap, jika tak berubah kita bisa mengecek routesnya.
2. Kita bisa mengecek apakah react bootstrap yang kita install sudah benar-benar terinstal dan dapat digunakan dengan membuka folder **node\_modules** kemudian cari folder bernama **bootstrap**, jika ada maka bootstrap tersebut sudah terinstal dan sudah dapat digunakan.
3. Saya sudah mengerti bahwa kita bisa merapihkan codingan kit dalam sekejap dengan menggunakan format dokumen, caranya blok semua bagian yang ingin dirapihkan lalu klik kanan dan pilih format document.
4. Saya sekarang sudah mengerti bahwa misal tag **<Front />** itu artinya sama seperti tag **<Front></Front>,** seperti penyederhanaan kode. jadi tampilan coding kita akan terlihat lebih simple.
5. Untuk versi react js yang sekarang menurut saya sudah tidak bisa menggunakan ‘as’ , **missal {BrowserRouter as Router, Route}**, karena berbed versi maka penulisannya menjadi berbeda yaitu **{BrowserRouter, Routes, Route}** dan untuk memanggilnya hars menggunakan element BrowserRouter terlebih dahulu.
6. Saya sudah bisa mengonversi codingan dari yang semula berbentuk html menjadi berbentuk JSX dengan menggunakan snippets yang sudah saya install sebelumnya, caranya tinggal blok bagian yang diinginkan lalu klik kanan dan pilih **convert HTML to JSX**.
7. Saya sudah mengerti bahwa di react js versi 6 kita tidak lagi memerlukan useRouterMatch namun kita bisa langsung bisa menambahkan url namun tanpa disertai dengan slash, contohnya

<Link to="kategori">

1. Saya sudah mengerti bahwa untuk menandai parent routes di react js versi 6 itu lebih simple dan lebih mudah, yaitu kita hanya perlu menambahkan slash dan bintang **( /\*)** disamping paent routes. Contohnya

<Route path='admin/\*' element={<Back/>} />

1. Saya sudah mengerti bahwa di dalam react js versi 6 itu sudah tidak menggunakan Switch, jika kita menggunakan Switch maka akan muncul error, sebagai gantinya kita bisa menggunakan Routes, contohnya

<Routes></Routes>

1. Saya sudah mengerti bahwa di dalam nested routes itu terdapat dua konsep utama, yaitu parent routes dan children routes, children routes berada di dalam parent routes yang ditandahi dengan slash bintang **( /\*)**  seperti yang sudah dijelaskan sebelumnya.
2. Saya sudah bisa memanggil komponen saat menggunakan navbar. Jadi ketika kita mengetik pilihan di navbar akan membawa kita menuju komponen file yang sudah diimport. Caranya kita bisa menggabungkannya dengan percabangan if, contoh

if (isi==='kategori') {

        tampil = <Kategori />

    }

1. Saya sudah mengerti bahwa dalam pembuatan useState itu dibutuhkan dua variable penting yaitu variable untuk menampung nilai state saat ini dan variabel untuk memperbarui nilai tersebut atau untuk meng-set var tersebut, contohnya yaitu

const [isi,setIsi ]= useState([]);

1. Saya sudah mengerti bahwa kita bisa memanfaatkan *hook* useEffect() untuk mengambil atau mengfetch data dari database, caranya kita harus membuat function terlebih dahulu kemudian mengisinya dengan axios, contoh penggunaanya yaitu

useEffect(() => {

        async function fetchData() {

            const request = await link.get('/kategori'); }}

1. Saya sudah mengerti ahwa fitur async dan await itusaling terhubung, await hanya bisa digunakan pada fungsi yang menggunakan async.
2. Saya sudah mengerti bahwa fitur await itu bisa digunakan untuk menunda eksekusi hingga proses asynchronous selesai.
3. Saya sudah mengerti bahwa untuk menggunakan react-hook-form kita harus mengimportnya terlebih dahulu, caranya yaitu kita harus mengetikan kode berikut

import { useForm } from "react-hook-form";

1. Saya sudah mengerti bahwa hook **error** dalam react versi 5 sudah diganti menjadi **formState: {errors}** di ract versi 6.
2. Saya sudah mengerti bahwa register itu dapat digunakan untuk mengampil nilai, bisa nilai inputan dari form ataupun yang lainnya dan dapat pula dimasukkan ke dalam console.log.
3. Saya sudah bisa membuat register dengan menggunakan kode

{...register('kategori', {required:true})}

Kode dari register itu terlihat sangat berbeda dari react versi sebelumnya yang menggunakan kode **ref={register}.** Required itu digunakan sebagai validation.

1. Saya sudah bis menampilkan eror dengan hook dengan menggunakan kode sebagai berikut

<h5>{errors.kategori && <span>Kategori is required</span>}</h5>

Tag h5 itu digunakan untuk menampilkan error di halaman web. Kategori itu diambil dari name form yang sudah kita buat.

1. Saya sudah bisa mengupload atau menginsert data ke dalam database dengan menggunakan kode

link.post('/kategori', data).then(res=>console.log(res));

kode diatas menggunakan axios sepeti sebelumnya.

1. Supaya idkategori tidak muncul semua secraa bersamaan, kita bisa menggunakan arrow function untuk mengatasiya, jadidengan begitu idkategori hanya akan muncul ketika di klik, contoh kodenya yaitu

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

1. Saya sudah bisa membuat konfirmasi seebagai verifikasi dalam program delete data, sehingga kecil kemungkinan data terhapus karena terpencet, untuk membuatnya kita bisa menggunakan kode

window.confirm("Anda akan menghapus item ini?")

1. Saya sudah mengerti bahwa kita bisa memanfaatkan trigger dari useEffect supaya kita tidak perlu memanggil function berkali-kali, jadi hanya cukup sekali saja sehingga proses load akan menjadi lebih cepat, contoh kodenya yaitu

useEffect(() => {

    fetchData();

  }, [isi]);

1. l

**Saya kurang paham**

1. Saya agak sedikit kesulitan saat pengerjaan karena perbedaan versi react, tapi dapat saya atasi dengan melihat langsung petunjuknya di web dari react.