|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| React JS Semester-4 Minggu ke-11 Video 39-46 | 95 |

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

1. Seperti pada materi api\_lumen yaitu membuat tabel baru ke dalam database dengan menggunakan php artisan. Caranya dengan mengetikkan di dalam terminalnya seperti ini php artisan make:model Order -mc --resource yang berarti akan terbentuk file model dengan nama Order beserta -mc atau migration dan controller nya juga lalu --resource ini berfungsi agar file yang dibuat tadi otomatis sudah terisi sesuai penggunaan file nya.

2. Mengupdate data order pada bagian kolom bayar, kembali, dan status menggunakan OrderController pada bagian function update. Caranya sama seperti sebelumnya yaitu membutuhkan validate atau validations untuk memvalidasi/menyeleksi data yang akan dimasukkan, $data yang menyimpan data dari request user, dan menu “Order” untuk menyambungkan dengan db nya.

3. Menampilkan Data Order atau Show Data Order seperti biasanya yaitu dengan menggunakan useGet untuk url nya dan hasilnya diletakkan di dalam tabel yang dibagi menjadi 7 kolom yaitu no, pelanggan, tgl order, total, bayar, kembali, dan status. Untuk bagian status akan dibuatkan button belum bayar apabila statusnya 0 dan apabila statusnya 1 yang tampil adalah tulisan Lunas. Caranya dibagian <td>{val.status}</td> diubah menjadi <td>{val.status===0 ? <button className=”btn btn-danger”>Belum Bayar</button> : <p>Lunas</p>} yang artinya apabila status = 0 maka dia akan menampilkan button danger beruliskan Belum Bayar lalu ada tanda : yang berarti apabila status tidak 0 maka dia akan menampilkan tulisan Lunas.

4. Menampilkan data order dengan acuan tanggal order nya, sama halnya seperti show data namun tidak hanya 1 data tetapi data-data yang memiliki tanggal sama, menggunakan method get. Caranya di dalam controller order bagian function show diisi seperti ini

public function show($a, $b)

    {

        $order = Order::where('tglorder', '>=', $a)

            ->where('tglorder', '<=', $b)

            ->get();

        return response()->json($order);

    }

Yang berarti $order menyimpan data yang diambil dari tabel order dengan ketentuan data ini memiliki tanggal lebih besar sama dengan dari $a dan lebih kecil sama dengan dari $b, setelah itu $a dan $b fungsinya sebagai objek yang diletakkan di url untuk menerima data yang dimasukkan dari user nya dan dikirimkan ke controller sesuai route nya. Di router url nya seperti ini $router->get('order/{a}/{b}', ['uses' => 'OrderController@show']);

jadi ketika diketikkan url localhost:3000/api/order/2021-03-18/2021-03-19 berarti a diisi dengan tanggal 2021-03-18 dan b diisi dengan 2021-03-19 dan url ini akan menjalankan controller order bagian function show. Setelah send url akan tampil response json berupa data yang memenuhi ketentuan tanggal tadi.

5. Menampilkan data sesuai dengan tawal dan takhir dari form date nya. Jadi dibuatkan sebuah form untuk input date tawal dan takhir lalu input submit dan pada form nya ditambahkan onSubmit={handleSubmit(cari)} yang berarti apabila submit ditekan maka dia akan menjalankan function cari. Function cari diletakkan diatas seperti ini

function cari(data) {

  setawal(data.tawal);

  setAkhir(data.takhir);

 }

Yang berarti dia akan mengisi awal dengan form dari tawal dan mengisi akhir dengan form dari takhir. Karena sebelumnya sudah diatur untuk awal dan akhir maka dia tinggal menjalankan sesuai dengan form nya. Agar ketika direfresh dia tampil semuanya maka pada

const [awal, setawal] = useState("2021-03-01");

const [akhir, setAkhir] = useState(today);

Yang berarti apabila form belum diset maka dia akan menampilkan data mulai dari tanggal 1 maret 2021 sampai today. Today ini adalah variable yang mengatur agar sesuai dengan tanggal hari ini. Di variable today dibuat seperti ini let today = new Date().toISOString().slice(0, 10); dengan begitu takhir akan terisi sesuai tanggal yang hari ini.

6. Melakukan installasi react-mdal dengan mengetikkan yarn add react-modal ke dalam cmd nya. jadi di dalam coding bisa menggunakan react-modal untuk diimport.

7. Menggunakan import Modal dari react-modal seperti ini import Modal from “react-modal”; setelah itu menambahkan Modal.setAppElement(“#root”); agar modal tersambung dengan page yang akan digunakan. Lalu ada const seperti ini const [mopen, setMopen] = useState(false); yang berarti ada const bernama mopen (modal open) dan setMopen yang digunakan untuk mengisi mopen nya. useState dari mopen pada awal membuka page adalah false. Lalu untuk meletakkan isi dari modal seperti ini

<Modal isOpen={mopen} onRequestClose={() => setMopen(false)} style={{overlay: {background: "darkgrey"}, content: {color: "palevioletred"}}}> disini terdapat isOpen yang berarti modal akan terbuka sesuai dengan isi mopen nya, apabila mopen false maka modal tidak akan terbuka. Untuk itu pada bagian button belum bayar akan di set untuk menjadikan mopen berisi true. Codenya seperti ini onClick={() => setMopen(true)}

Yang berarti mopen akan diisi dengan true ketika mengklik button belum bayar, dengan begitu setelah mengklik button akan tampil modal. Untuk mengclose modal caranya seperti ini di dalam modal ada <button className="btn btn-success" onClick={() => setMopen(false)}>Close</button> fungsi button ini adalah ketika dia diklik maka mopen akan diisi dengan false, dengan begitu modal akan otomatis tertutup.

8. Mengatur Isi dari Modal yang ada di dalam button belum bayar. Sebelumnya ditmbahkan dulu const setValue pada useForm nya. lalu menambahkan useState(0) untuk total dan setTotal, useState(‘’) untuk pelanggan dan setPelanggan nya. setelah itu pada bagian button belum bayar akan diarahkan ke function bernama filterData dengan mengirim val.idorder nya juga, jadi filterData akan mengambil id yang dikirim dari button ini. Untuk function filterData nya seperti ini

function filterData(id) {

  const data = isi.filter((val) => val.idorder === id);

  setPelanggan(data[0].pelanggan);

  setTotal(data[0].total);

  setMopen(true);

 }

Terdapat const data yang berfungsi untuk menjadikan kiriman val.idorder sebagai id nya, lalu setPelanggan untuk mengisi useState pelanggan tadi dengan pelanggan dari data yang diambil, lalu setTotal untuk mengisi useState total dengan total dari data yang diambil lalu mengisi mopen dengan true yang fungsinya untuk membuka/open modal nya. ketika modal dijalankan, pada bagian modal terdapat onAfterOpen={isiForm} yang berarti setelah modal open atau terbuka dia akan menjalankan function isiForm. Di dalam function isiForm ini seperti ini

function isiForm() {

  setValue("total", total);

 }

Yang berarti form bernama total akan diisi dengan data dari kolom total sesuai data yang ditekan button belum bayarnya.

9. Mengatur button bayar yang ada di dalam modal agar form yang dimasukkan ke dalam modal berjalan. Caranya pada form yang menyimpan button bayar modal seperti ini <form onSubmit={handleSubmit(simpan)}> yang berarti ketika form ini sdah disubmit dia akan menjalankan function yang bernama simpan. Di dalam function simpan isi berisi seperti ini

async function simpan(data) {

  let hasil = {

   bayar: data.bayar,

   kembali: data.bayar - data.total,

   status: 1,

  };

  const res = await link.put("/order/"+idorder, hasil);

  setMopen(false);

 }

Yang berarti function ini mengambil data dari form yang disubmit barusan, lalu membuat variable hasil yang berisi sebuah objek yang menyimpan data untuk dimasukkan ke table nya. kolom bayar akan diisi dengan data dari form bayar, kolom kembali akan diisi dengan data hasil dari pengurangan form bayar dikurangi form total, lalu kolom status diisi dengan 1 agar nanti jadi Lunas. Lalu ada const res yang menjalankan link dengan method put atau update ke dalam order sesuai dengan data dari variable hasil. Setelah proses update berhasil maka modal akan ditutup lagi, yaitu dengan membuat mopen menjadi false.

**Saya Belum Mengerti**

1.