|  |  |
| --- | --- |
| Materi | Nilai |
| REACT-JS-V51-58 | **98** |

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

1. Sebelum memanggil komponen ke dalam suatu file di react js, kita harus mengimportnya terlebih dahulu, saya sudah bisa mengimport komponen js ke dalam suatu file contohnya

import Order from './Order';

1. Saat membuat status pembaran order namun ada yang belum lunas, kita bisa menggantinyaa dengan button, dengan memanfaatkan if contohnya seperti

val.status===0 ? <button onClick={ ()=>filterData(val.idorder) } className='btn btn-danger'>Belum Bayar</button> : <p>Lunas</p>

1. Saya sudah bisa menginstal ract modal, untuk membuat form pembayaran, caranya yaitu dengan mengetikkan kode beriut ke dalam cmd

yarn add react-modal

1. Sepertihalnya library lain, modal ini bisa diinstal dengan du acara yaitu online dan offline, untuk onlinenya kita bisa menggunakan cdn, dan untuk offlinenya terdapat du acara yaitu dengan menggunakan npm ataupun dengan menggunakan yarn.
2. Sebelum menggunakan modal, kita harus mengimportnya terlebih dahulu dari react-modal, yaitu dengan menggunakan kode

import Modal from 'react-modal';

1. Saya sudah mengerti bahwa kita harus menggunakan komponen modal supaya modal tersebut bisa tampil di halaman web kita, kemudian kita bisa mengisi apapun di dalam komponen modal tersebut seperti mengisinya dengan form atau yang lainnya, contohnya

<Modal> <h2>pembayaran order</h2> </Modal>

1. Saya sudah mengerti bahwa kita bisa menggunakan **setMopen(true)** untuk mengaktifkan modal(memunculkannya ke halaman web) dan kita bisa memanfaatkan **setMopen(false)** untuk menonaktifkan modal.
2. Saya sudah memahami bahwa dengan menggunakan kode

onRequestClose={ ()=>setMopen(false) }

kita bisa menutup modal atau keluar dari moda ketika kita mengklik bagian luar dari modal.

1. Saya sudah mengerti bahwa kita harus menggunakan reister pada form yang kita buat supaya saat melakukan setValue, valuenya bis dimasukkan ke dalam form, contoh penggunaan register yaitu

{...register("total", { required: null })}

1. Saat melakuka proses pembayaran, nominal yang kita masukkan harus tidak boleh kurang dari nominal total, untuk melakukannya kita bisa menambahkan perintah **min:** di daam register, contohnya

{...register("bayar", { required: true, min:total })}

1. Saya sudah mengerti bahwa untuk mengupdate inputan yang memiliki tipe file, kita harus menguploadnya dengan menggunakan perintah post, jika inputannya hanya berupa teks biasa atau numerik kita bisa langsung menggunakan perintah put. Contohnya yaitu

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

1. Saya sudah bisa melakukan join table di dalam file controller api, misalnya saya ingin menggabungkan kolom idorder dari table orders ke dalam table details saya bisa menggunakan kode

$data = DB::table('details')

        ->join('orders','orders.idorder', '=', 'details.idorder'

        ->join('menus','menus.idmenu', '=', 'details.idmenu')

1. Saya sudah mengerti bahwa sebelum melakukan join table, kita harus memanggilnya terlebih dahulu, yaitu dengan menggunakan kode

use Illuminate\Support\Facades\DB;

1. Saat melakukan proses join table, langkah selanjutnya adalah kita harus menselect table dari database, kita bisa menselect semua atau beberapa saja, untuk menselect semua data kita bisa menggunakan tanda bintang **( \* ),** contohnya

->select('orders.\*', 'details.\*', 'menus.\*', 'pelanggans.\*', 'kategoris.\*')

1. Saya sudah bisa membuat carousel dengan menggunakan react bootstrap, caranya yaitu kita harus mengimport terlebih dahulu komponen carousel dari react-bootstrap, contohnya

import Carousel from 'react-bootstrap/Carousel';

1. Setelah kita mengimport carousel tersebut, kita bisa menuliskan komponen carousel **( <Carousel> </Carousel> )** yang di dalamnya terdapat carouselItem **( <Carousel.Item> </Carousel.Item> )** yang dapat digunakan untuk menampung gambar untuk carousel dan juga terdapat komponen Carousel.Caption **( <Carousel.Caption> </Carousel.Caption> )** yang dapat digunakan untuk menampung keterangan/Caption dari carousel tersebut.
2. Saya sudah bisa membuat modal untuk menu user, caranya sama saja seperti membuat modal pada menu pembayaan yaitu yang pertama kita harus mengimport Modal dari React modal, yaitu dengan menggunakan kode

import Modal from 'react-modal’;

1. Kemudian kita bisa mengisinya dengan form untuk insert data da membuat function untuk melakuka proses input data e dalam database.
2. Untuk mengupdate keterangan atau status dari user, yang semula statusnya di banned kemudian sat di klik menjadi akstif, dan sebaliknya cara pertama yang bisa kita lakukan yaitu kita bisa membuat function kemudian mengisinya dengan pengujian untuk mengambil value status tersebut dari database lalu kit bisa membuat variable untuk media perintah update, misalnya

let kirim = { status: statue }

const res = await link.put('/user/'+id, kirim);

1. Nah, jika sudah kita hanya perlu membuat pengujian di dalam button tersebut, midsal jika value dari row status adalah bernilai satu maka akan menampilkan button dengan tulisan aktif, dan jika value dari row status adalah nol maka akan menampilkan butto dengan tulisan banned, contoh kodenya yaitu

{val.status === 1 ? ( <input

className="btn btn-success” type="submit" value="ACTIVE" onClick={ ()=>status(val.id) } /> : <input

className="btn btn-danger” type="submit" value="BANNED" onClick={ ()=>status(val.id) } /> )}

1. Saya sudah bisa mengatur hak akses pada web yang saya buat, missal jika yang login adalah admin maka dapat memiliki hak penuh atas data dari web tersebut, dan jika yang login adalah kasir/koki hanya akan bisa mengakses menu tertentu saja. Untuk melakukan hal tersebut tentunya kita harus melakukan pengujian, contohnya

<Link to="kategori" className="text-decoration-none"> sessionStorage.getItem('level')==='admin' ) ? (<li className="list-group-item">Kategori</li>) : "" } </Link>

1. Saya sudah mengerti bahwa untuk membuat redirect kita membutuhkan useNavigate yang sebelumnya sudah diimport dari react-router-dom.
2. Saya sudah mengerti bahwa di dalam react js versi 6, kita sudah tidak perlu lagi menggunakan useHistory untuk membuat proses redirect, tapi kita bisa menggunakan useNavigate, supaya kita mengetahui berbagai updatean function dari react kita harus membaca dan melihat dokumentasi di web remi reat js.
3. Untuk mengarahkan atau meredirect halaman menuju halaman lain yang sudah ditentukan, mial selepas kita melakukan proses login maka akan mengarahkan kita ke dalam menu utama, kita bisa menggunakan perintah **Navigate(url halaman yang diinginkan).** contohnya

navigate('/admin/home');

window.location.reload();

1. Saya sudah bisa membua menu logout, caranya sangat mudah, yaitu kita hanya perlu membersihkan sessionstorage dengan perintah clear(), kemudian kita bisa meredirectnya menuju halaman login, contoh kodenya yaitu

sessionStorage.clear();

navigate('/login');

1. Saya sudah mengerti bahwa hash dalam password digunakan untuk melakukan enkripsi pada password supaya keamanan password menjadi lebih terjaga dan tidak terlihat walai sudah dibuka di dddalam database.
2. Saya sudah bisa membuat hash pada password, caranya kita harus menambah Hash::make pada file controller di dalam API, contohnya yaitu

'password'=>Hash::make($request->input('password')),

1. Saat melakukan proses login, kita tentunya harus mengambil satu baris yang sama dari database dan menccokkannya dengan inputtan yang dimasukkan di dalam form. Untuk melakukannya kita bisa menggunakan function atau method first(), contoh penggunaanya yaitu

$user = User::where('email',$email)->first();

1. Supaya password yang dimasukkan di dalam form tidak harus sama persis seperti kode enkripsi yang ada di database, kita bisa menggunakan hashcheck, contoh penggunaanya yaitu

Hash::check($password, $user->password

**Saya kurang paham**

* + - 1. -