|  |  |
| --- | --- |
| Materi | Nilai |
| REACT-JS-V41-50 | **98** |

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

1. Saya sudah mengerti bahwa dalam pembuatan filter data berdasarkan tanggal, kita membutuhkan dua buah variable, satu variable untuk tanggal awal dan satu variable lain untuk tanggal akhir, missal **$a** dan **$b**
2. Supaya program cari tanggal tersebut dapat bejalan, kita bisa memanfaat method where(‘’), untuk penulisannya kurang lebih mirip seperti program untuk join table, contohnya

$order = Order::where(‘tglorder’,’<=’,$a)

->where(‘tglorder’,’>=’, $b)

1. 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.
2. 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**.
3. Saya sudah mengerti ahwa fitur async dan await itusaling terhubung, await hanya bisa digunakan pada fungsi yang menggunakan async.
4. Saya sudah mengerti bahwa fitur await itu bisa digunakan untuk menunda eksekusi hingga proses asynchronous selesai.
5. Sebelum mengetest program filter berdasarakan tanggal tersebut kita tidak boleh lupa untuk mengatur routes nya terlebih dahulu, kita bisa melakukan perintah get kemudian diisi dengan nama url kemudian variable tanggal awal dan variable untuk tanggal akhir, contohnya yaitu

$router->get('order/{a}/{b}' , ['uses' => 'OrderController@show']);

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 ketika kita membuat program backend dengan menggunakan react js dan API seperti Lumen, kita harus membuatnya di API dulu kemudian memastikan bahwa program tersebut sudah benar. Baru setelah itu kita bisa memasukkannya ke react js.
2. Saya sudah bisa membuat form dengan Bahasa pemrograman HTML di file react js, saya juga menggabungkannya dengan class dari bootstrap supaya form tersebut bisa menjadi lebih rapih.
3. Saya sudah bisa memunculkan content sesuai url yang ada, caranya yang pertama kita harus membuat konstanta dan memasukkannya ke **useParams(),** jika sudah maka kita bisa langsung membuat if dengan kondisi apabila konstanta yang kita buat tadi sama dengan url, maka akan tampil komponen yang diinginkan , contoh programnya yaitu

if (isi==='order') { tampil = <Order /> }

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. 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.\*')

**Saya kurang paham**

* + - 1. -