|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| React Js Project Resto bag.2 (Vid 29-38) | 90 |

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

1. Memahami & dapat meng-inputkan data menu melalui postman seperti pada sebelumnya :

* Menggunakan perintah post
* Isi link sesuai pada react
* Menggunakan form-data untuk mengisi key & value sesuai pada database menus
* Jangan lupa menggunakan api\_token pada header agar akses data dapat dilakukan

2. Memahami bahwa dalam lumen(laravel) untuk menggabungkan tabel antar database dapat menggunakan method join :

* Import terlebih dahulu syntax untuk perintah join table use Illuminate\Support\Facades\DB;
* Syntax join tabel adalah $var = DB:table(‘nama table yg akan dijoin’)
* Lalu di bawahnya ->join(‘table yg akan di ambil kolomnya’,’tabel.kolomnya’, ‘=’, ‘tabel awal.kolom tabel utama yang akan di join dengan tabel yg di tuju’)
* Lalu untuk menampilkan semua isi tabel nya ->select(‘tabel utama.\*’,’tabel tujuan.kolom yang akan di tampilkan’)
* Untuk mengurutkannya saat ditampilkan menggunakan ->OrderBy(‘tabel utama.kolom order’,’asc/desc’)
* Menggunakan function get() untuk menampilkannya

3. Memahami bahwa useEffect merupakan function yang akan berjalan apabila triggernya(const yang di set dengan useState) mengalami perubahan.

4. Memahami bahwa dengan menggunakan useEffect maka dapat dimanfaatkan untuk melakukan perubahan data hanya sekali, jadi load page tidak akan terlalu lama dan hanya akan ter-refresh saat terjadi perubahan data.

5. Memahami cara menggunakan hook untuk function2 agar tinggal pakai saja dengan memanggilnya :

* Membuat file hook dari function lalu import keperluan yang dibutuhkan
* Lalu buat script function yang dibutuhkan dan return
* Kemudian impor file hook pada file yang akan menggunakan hook dan tinggal gunakan function yang dibuat pada hook tadi.

6. Memahami & dapat menampilkan data menu :

* Pada file menu import useGet yaitu hook function untuk fetchData yg sudah dibuat sebelumnya untuk menampilkan data
* Buat const bentuk array [isi] = untuk memanggil useGet(‘/Menu’); {const dibungkus array karena data yg akan di oper berbentuk array}
* Buat variabel untuk nomer tabel
* Di dalam return buat 2 rom untuk judul dan untuk tabel
* Untuk table buat dengan tag table className=”table” lalu isi dengan tag thead dan tbody
* Untuk thead isi dengan tr th untuk judul kolom2 tabel no,kategori,menu,gambar,harga
* Untuk tbody isi dengan map untuk menampilkan isi data yang berbentuk array menggunakan key {const get isi.map((key, index) => ( <tr key={index}> lalu isi untuk memanggil misal <td>{val.kolom tabel}</td>
* Untuk gambar isi terlebih dahulu dengan tag img lalu isi key dan kolom tabel gambar pada src nya
* Sama saja seperti saat menampilkan data kategori

7. Memahami penggunaan useEffect clean up untuk membersihkan data sisa/cache setelah proses input data agar page ter-resfresh

8. Memahami & dapat menampilkan data pelanggan yaitu :

* Pada file pelanggan impor hook useGet
* Buat const berbentuk array untuk menggunakan useGet yg telah diisi function fetch data untuk menampilkan isi data
* Pada return buat tabel untuk menampilkan data pelanggan sama seperti saat menampilkan data kategori dan menu yaitu menggunakan map dan key index karena data berbentuk array

9. Memahami & dapat membuat custom hook untuk delete data

10. Dapat menambahkan opsi delete data pada tabel menu & pelanggan dengan memanfaatkan hook delete data yaitu :

* Membuat file untuk hook delete lalu mengiis dengan async function untuk delete data, menggunakan asynchronus karena perlu menunggu agar semua sistem data terload baru akan dilakukan penghapusan
* Media penghapusan melalui perantara iddata baik menu maupun pelanggan
* Pada hook delete impor file link dari folder axios karena akan menggunakan link yg ada di dalam file link karena perlu akses data menggunakan api\_token yang ada di file link
* Impor useState lalu buat const pesan dan setPesan dalam bentuk array untuk menggunakan useState untuk menampilkan pesan penghapusan yg berasal dari api lumen di function destroy
* Lalu letakkan hapus dan pesan pada return untuk menampilkan respon apabila function berjalan
* Pada file pelanggan dan menu impor use untuk delete lalu tambahkan const untuk menambahkan hapus dan pesan dari hook delete beserta alamat url nya const { hapus, pesan } = useDelete('/Pelanggan/menu/');
* Tambahkan kolom untuk hapus data pada tabel
* Pada tabel isinya tambahkan button delete lalu beri class dengan bootstrap, gunakan onClick untuk pemicu hook hapus onClick={() => hapus(val.idmenu/idpelanggan)}
* Tampilkan pesan di atas tabel agar warning message ketika akan menghapus data dapat terlihat

11. Memahami & dapat membuat form untuk insert data menu :

* Impor useForm dari react-hook-form karena akan menggunakan form
* Tambahkan form untuk input data seperti pada kategori lalu atur untuk menu,harga,gambar,kategori dan submit
* Tambahkan const untuk useState dan useForm seperti pada kategori
* Pada form bagian kategori buat menjadi optiom lalu isi dengan map untuk menampilkan data yang bisa di pilih dengan value idkategori dan yg tampil menggunakan kategori juga tambahkann register
* Untuk upload gambar buat function simpan dengan params data lalu buat const formData = menggunakan new FormData();
* Append data agar bisa di masukkan ke database formData.append('menu', data.menu); khusus untuk gambar tambahkan array index ke 0 -> data.gambar[0]
* Tambahkan link untuk repon apabila data berhasil di masukkan link.post('./Menu', formData).then(res => setPesan(res.data.pesan));
* Untuk menggunakan setPesan, tambahkan pada useDelete
* Tambahkan reset agar data ter-reset apabila telah berhasil di inputkan.

12. Memahami & dapat memasukkan value data ke form yang akan update/ubah :

* Tambahkan button untuk update pada tabel menu, arahkan onClick pada function showData
* Pada function show di controller menu api tambahkan merger database karena akan digunakan untuk menampilkan kategori menu, juga tambahkan where agar id nya lebih cepat di dapatkan karena lebih spesifik ->where('idmenu', '=', $id)
* Buat function showData dengan params (id) lalu isi dengan const await untuk mendapatkan idmenu const res = await link.get('/Menu/' + id);
* Atur agar value dari data yang di klik untuk di ubah maka datanya akan di masukkan ke dalam form untuk di ubah setValue('menu', res.data[0].menu); untuk gambar menggunakan tag img src setGambar(<img src={res.data[0].gambar} alt="" height="150" />);
* Tambahkan useState untuk const gambar,setGambar dan idkategori, setIdkategori
* Atur pula dalam routing untuk akses data

13. Memahami & dapat menerapkan opsi untuk update gambar atau tidak :

* Pada function update di api controller menu isi params dengan request dan id karena akan mengubah data
* Lanjut is dengan validasi data untuk tulisan saja gambar tidak perlu
* Buat if request hasFile gambar dengan getClientOriginal yang makshudnya apabila ada gambar baru yang di upload (gambar menu di update) maka gambar tersebut akan di proses dengan request $data untuk di inputkan ke database dan disimpan di folder upload seperti pada proses input data
* Apabila tidak ada perubahan untuk gambar, maka else nya adalah array $data yang berisi perubahan untuk teks saja menu,harga dsb tanpa perlu gambar
* Panggil modal untuk menggunakan function update dan tambahkan pesan apabila data berhasil di update.

14. Memahami bahwa untuk update data yang mengandung file maka di dalam router nya menggunakan post bukan put karena bukan bentuk text.

15. Memahami & dapat megatur untuk finishing update data pada react.js :

* Tambahkan const useState untuk idmenu, setIdmenu dan pilihan, setPilihan seperti pada kategori untuk proses penyimpanan update data apakah ada data yang diupdate atau tidak
* Pada function simpan tambahkan if untuk pilihan lalu isi dengan link.post untuk update data formData lalu else link.post untuk pesan update data dengan idmenu dan formData lalu setPilihan(true) karena tidak ada data yang di ubah
* Pada required untuk gambar di form hilangkan dan hanya gunakan register