|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| React JS Semester-4 Minggu ke-10 Video 29-38 | 95 |

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

1. Menggunakan use Illuminate\Support\Facedes\DB; di dalam file controller untuk mengambil dari objek DB nya. Setelah menambahkan use itu maka untuk mengatur tabel dari db nya bisa dilakukan misalkan seperti ini

$data = DB::table('menus')

->join('kategoris', 'kategoris.idkategori', '=', 'menus.idkategori')

->select('menus.\*', 'kategoris.kategori')

->orderBy('menus.menu', 'asc')

->get();

Yang berarti $data ini menyimpan database yang menyambungkan dengan table bernama menus yang akan diatur dengan yang pertam ada join yang berarti table menus akan digabungkan dengan tabel kategoris yaitu pada bagian idkategori yang ada di kategoris dengan idkategori yang ada di menus adalah sama. Yang kedua ada select yang berarti setelah idkategori di menus memiliki data yang sama dengan idkategori yang ada di tabel kategori sama maka kolom kategori akan dimasukkan ke dalam tabel menus untuk ditampilkan. Yang ketiga ada orderBy yang berarti akan diurutkan sesuai dengan kolom menu nya berdasarkan dari yang paling kecil (asc) dan terakhir ada get() yang berarti method nya get yang berfungsi untuk menampilkan. Jadi ketika ditampilkan tabel menus pada postman nya, data yang ada di dalam tabel menus akan memiliki tambahan kategori nya apa sesuai dengan idkategori yang ada.

2. useEffect adalah function yang diberikan oleh react, useEffect diimport dari react. Function useEffect ini bekerja atau berjalan sesuai dengan triggernya atau pemicunya.

3. Mengcustom Hook dengan cara membuatkan folder Hook yang isinya file useGet.js. Di dalam file ini terdapat 2 import yaitu useState, useEffect dan link dari Axios/link setelah itu const useGet = () tanda kurungnya diisi dengan (url) dengan url ini bisa digunakan untuk file lain. Jadi isi dari const useGet ini bisa digunakan di dalam file lain apabila ada return nya juga. Jika sebelumnya isi dari useGet ini hanya ada di dalam file kategeori, dengan cara ini semua file selain kategori juga dapat menggunakan useGet nya. Misalkan di dalam file ini terdaoat useEffect yang isinya adalah fetchData dan di dalam fetchData ini ada const request yang menyimpan await link.get(url) dimana ini nanti bisa digunakan untuk mengambil tindakan sesuai method dan url yang diberikan di file lain. Const request ini dimasukkan ke dalam setIsi jadi isi lah yang menyimpan request nya dan isi yang direturn untuk bisa digunakan diluar file.

4. Karena Hook sudah dicustom, ketika membuat tampilan dari Data Menu seperti Data Kategori, variable isi tadi bisa digunakan untuk mengambil semua data dari menu sesuai dengan url yang sudah diatur di Axios nya seperti ini Const [isi] = useGet(‘/menu’) yang berarti nanti apabila url nya localhost:3000/admin/menu maka dia akan menjalankan url sesuai dengan pengaturan axios dengan method get. Karena sudah tersambung dengan table menus yang ada di db, maka table yang dibuatkan di vs code untuk data menu nya bisa mengambil data dari table menus nya. Pada bagian tbody bisa menggunakan {isi.map(val,index)=>()} dan didalam kurung setelah panah itulah tempat meletakkan tr dan td nya yang dimana tr sebagai key yaitu menggunakan index dan td nya sebagai value dengan menggunakan val. Seperti biasa untuk pengambilan data misalkan bagian kolom menu maka penulisannya adalah {val.menu}. Pada bagian kolom gambar agar menampilkan gambarnya tidak hanya url gambar nya saja maka menggunakan img src yang src nya diisi dengan {val.gambar}.

5. Menggunakan useEffect cleanUp, Contohnya di dalam useEffect seperti ini

useEffect(() => {

  let ambil = true;

  async function fetchData() {

   const res = await link.get(url);

   if (ambil) {

    setIsi(res.data);

   }

  }

  fetchData();

  return () => {

   ambil = false;

  };

 }, [isi]);

Jadi useEffect ini awalnya adalah true yang diletakkan di variable ambil, lalu terdapat function detchData yang mengisi const res dan ada if ambil atau jika ambil berisi true maka isi akan di setIsi dengan res.data setelah itu functionnya dijalankan. Di dalam return terdapat ambil = false; yang berfungsi apabila useEffect tidak dijalankan maka ambil akan berisi false, jadi pengambilan data dari database cukup sekali ketika yang pertama itu dan tidak perlu mengambil lagi sehingga mengakibatkan loading terlalu lama karena memuat data terus menerus.

6. Mengcustom Hook onClick untuk button hapus agar dapat digunakan untuk banyak file. Karena ini button hapus maka diperlukan id dari data yang akan dihapus, jadi button yang ada di dalam file misalkan file menu.js pada bagian buttonnya disetting agar ketika mengklik dia mengambil id dari data yang dipilih seperti ini

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

Jadi ketika button delete ini diclick maka dia akan mengirimkan sebuah id yang diambil dari data yang dipilih ke function hapus. Function hapus ini terletak di file useDelete.js yang ada di folder Hook. Sebelumnya agar tersambung dengan file ini maka harus mengimport nya terlebih dahulu seperti mengimport useGet dari Hook. Di dalam file useDelete nya membutuhkan 2 import yaitu untuk link axios agar tersambung dengan db nya dan useState untuk pesannya nanti. Lalu di dalam const useDelete nya berisi seperti ini

const useDelete = (url) => {

 const [pesan, setPesan] = useState("");

 async function hapus(id) {

  if (window.confirm("Yakin akan menghapus?")) {

   const res = await link.delete(url + id);

   setPesan(res.data.pesan);

  }

 }

 return { hapus, pesan };

};

Jadi useDelete ini menggunakan url yang nanti bisa diisi dengan url yang berbeda-beda pada bagian const res nya. Ketika function hapus menerima id yang dikirimkan dari button delete dia akan menjalankan confirm dulu jika confirm oke maka const res akan dijalankan dengan menyertakan id untuk syarat penghapusannya. Lalu ada setPesan yang berfungsi mengisi pesan. Lalu function hapus dan pesan direturn agar bisa digunakan di file lain. Setelah itu di dalam file react nya ditambahkan const { hapus, pesan } = useDelete("/menu/"); yang berarti untuk function hapus dan pesan bisa digunakan di file ini dengan ketentuan url yang diberikan ke useDelete adalah /menu/ sesuai dengan routes untuk mengatur delete menu. Jadi file useDelete yang dibuat di folder Hook bisa digunakan untuk semua file react.

7. Mengupload gambar atau data yang lainnya ke dalam database menggunakan append, ketika sebuah form sudah diisi data oleh user lalu submit maka form ini akan menjalankan function bernama simpan sesuai dengan <form onSubmit={handleSubmit(simpan)}> dan function simpan ini berisi seperti ini

function simpan(data) {

  const formData = new FormData();

  formData.append("idkategori", data.idkategori);

  formData.append("menu", data.menu);

  formData.append("harga", data.harga);

  formData.append("gambar", data.gambar[0]);

  link.post("/menu", formData).then((res) => setPesan(res.data.pesan));

  reset();

 }

Jadi function ini menerima data yang diberikan oleh form lalu membuatkan objek yang menyimpan data ini dan formData.append itu untuk memasukkan data sesuai dengan nama kolomnya. Misalkan formData.append(‘Menu’,data.menu); berarti kolom menu akan diisi dengan data yang diambil dari form input bernama menu. Sedangkan pada bagian gambar ditambahkan [0] karena dia menyimpan sebuah file dan file ini memiliki rincian maka dari itu dimasukkan ke dalam array. Lalu untuk mempost menggunakan link.post itu dan ketika berhasil akan munul pesan yang sudah disetPesan. Terakhir ada reset(); yang fungsinya untuk mereset form yang tadinya terisi menjadi kembali kosong.

8. Show data Menu caranya sama seperti show data pada kategori. Ada beberapa yang berbeda yaitu jika di kategori tidak ada select option dan input file pada form nya sedangkan menu ada select option dan input file maka untuk menampilkan data menu kedalam optionnya seperti ini

<select name="idkategori" ref={register} className="form-control">

{kategori.map((val, index) => val.idkategori == idkategori ?

(<option key={index} selected value={val.idkategori}> {val.kategori} </option>) :

(<option key={index} value={val.idkategori}> {val.kategori} </option>)

)}

</select>

Jadi ketika idkategori mendapat kiriman data idkategori dari data yang dipilih untuk diubah dan idkategori ini ada yang sama dengan yang dioption maka dia akan menggunakan kategori itu untuk ditampilkan, cirinya yaitu ada selected nya sedangkan jika belum ada data yang mau diubah maka select option akan tampil tanpa selected.

9. Biasanya ketika mengupdate sebuah data pada lumen menggunakan method put, tetapi apabila ada data yang perlu diupdate dan ada file nya yang diupdate maka method nya diganti menjadi post.

10. Untuk membedakan submit agar melakukan proses insert atau update pada data menu maka sama halnya ketika mengatur di data kategori yaitu menggunakan useState pilihan dan setPilihan.

**Saya Belum Mengerti**

1.