|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| React JS Semester-4 Minggu ke-12 Video 47-54 | 95 |

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

1. Mengatur menu ‘Order Detail’ dengan menggunakan table details. Karena di dalam table details kolom nya tidak lengkap hanya ada iddetail, idorder, dan idmenu maka perlu di dalam DetailController perlu melakukan join table dulu agar bisa terhubung dengan table lain. Penempatannya yaitu di bagian function show yang parameter nya diisi dengan $a dan $b, untuk isinya seperti ini

$data = DB::table('details')

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

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

            ->join('pelanggans', 'orders.idpelanggan', '=', 'pelanggans.idpelanggan')

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

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

            ->where('tglorder', '>=', $a)

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

            ->orderBy('orders.tglorder', 'asc')

            ->get();

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

$data digunakan untuk menyimpan data, DB digunakan sebagai penyambung dengan database table bagian details. Untuk bisa menggunakan DB maka di dalam DetailController harus ditambahkan use Illuminate\Support\Facades\DB; terlebih dahulu. Setelah mengakses table details, kemudian terdapat panah join yang di dalam kurungnya berisi ‘orders’ berarti menjoinkan atau menggabungkan dengan data dari table orders dengan patokan idorder dari table details sama dengan idorder dari table orders. Begitu juga dengan join-join yang lainnya. Pada bagian select adalah bagian pemilihan data mana saja yang akan digunakan misalkan ‘orders.\*’ yang memiliki .\* berarti mengambil semua data dari table orders nya. lalu memilih table yang memiliki tglorder lebih besar dari $a dan lebih kecil dari $b dan terakhir diurutkan dari tglorder yang terkecil.

2. Mengatur Menu Order Detail agar tampil ke dalam browser nya. Caranya seperti biasa yaitu menggunakan useGet, useState, link, dll. Di dalam Detail.js atau file react nya. Karena sebelumnya di dalam DetailController.php sudah dijoinkan table, jadi di bagian file react nya tinggal menggunakan kolom-kolom dari database untuk ditampilkan di table nya.

3. Menambahkan form tawal dan takhir ke dalam file react Detail.js dengan mengcopy nya dari file react Order.js pada bagian form, useForm, dan function yang bersangkutan agar bisa digunakan sama seperti pada menu Order.

4. Mengatur lumen untuk react user di dalam LoginController.php nya, membuatkan function index yang berisi seperti ini

$data = User::where('level', '<>', 'pelanggan')->get();

return response()->json($data);

Jadi ketika function ini dijalankan atau dipanggil maka $data akan mengeluarkan response apabila database User yang isi dari kolom level nya bukan pelanggan (ditandai <>) berhasil di method kan dengan get.

5. Menambahkan data user melalui input submit tambah yang ada di dalam tampilan menu usernya. Ketika button tambah diklik maka akan tampil modal, untuk itu diperlukan import modal dan const untuk open modal yang digunakan ketika tambah diklik. Ketika Modal terbuka akan muncul sebuah form seperti yang ada pada modal belum bayar di menu order. Dari form ini bisa diisi data yang nantinya akan diarahkan ke function simpan, di function simpan ini adalah tempat untuk mengambil data yang dimasukkan tadi lalu dipost menggunakan link axios melalui lumen register.

6. Mengubah button banned yang ada di kolom status menu user menjadi aktif dan sebaliknya dengan cara pada bagian input submit atau button ini tadi ditambahkan onClick={()=>status(val.id)} yang berarti apabila menekan button nya dia akan menjalankan function status dengan mengirimi id dari data yang diklik button banned/aktif nya. setelah itu pada bagian function statusnya seperti ini

async function status(id) {

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

  let start = 0;

  if (data[0].status === 1) {

   start = 0;

  } else {

   start = 1;

  }

  let kirim = {

   status: start,

  };

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

 }

Yang berarti function status ini mendapat kiriman id dari data button yang dipilih tadi, lalu dengan id ini bisa difilter di dalam const data untuk mencocokkan dengan data yang ada di database yang memiliki id sama, misalkan id yang dikirim dari button 16 maka akan difilter atau dicarikan id yang sama di dalam databasenya, jika sudah ketemu id yang sama maka data dari semua kolom yang ber-id 16 ini akan disimpan di dalam const data. Setelah itu ada variable start yang berisi 0, digunakan apabila status dari data yang diambil tadi sama dengan 1 maka start akan diisi dengan 0, dan jika tidak maka start akan diisi dengan 1. Variable kirim berfungsi untuk menyimpan data status yang diisi oleh start dan terakhir ada const res yang berfungsi untuk menyambungkan dengan link bermethod put atau update dengan url /user/id (sesuai id data yang dipilih) dan ,kirim berarti data akan diupdate sesuai isi dari variable kirimnya. Pada routes nya untuk url user/id dengan method put berarti dia akan menjalankan lumen LoginController bagian function update, di function inilah tempat mengatur data yang direquest kan tadi untuk diupdate ke dalam database nya. Dengan begitu ketika banned (status = 0) di klik maka start akan diisi dengan 1, dan akan otomatis berubah menjadi aktif.

7. Mengatur page untuk login yang sebelumnya routes dan LoginController.php nya sudah diatur, yang pertama adalah memastikan routes sudah mengarah ke LoginController pada bagian function login. Lalu membuat file react untuk loginnya yang berisi code untuk form, form ini menggunakan handleSubmit yang berarti ketika disubmit dia akan mengarahkan kemana, form ini akan diarahkan ke function login yang berfungsi mengambil data dari form yang dikirimkan lalu menjalankan link atau routes untuk loginnya yang berarti menjalankan lumen LoginController bagian function login. Di dalam function login yang ada di api\_lumen ini berisi code untuk menjalankan login dengan ketentuan

if (isset($user)) {

            if ($user->status === 1) {

                if ($user->password === $password)

yang berarti apabila data yang dimasukkan sudah terdaftar maka dia akan menjalankan if selanjutnya yaitu apakah data yang dimasukkan statusnya 1 yang berarti aktif maka dia akan menjalankan if selanjutnya yaitu apakah password yang dimasukkan sesuai dengan password yang terdaftar di database nya. Setelah melewati 3 if dan berhasil semua maka proses login berhasil, tetapi jika ada salah satu if yang tidak bisa dilewati atau tidak masuk ketentuan if maka proses login gagal.

**Saya Belum Mengerti**

1.