|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran React-JS Video ke 47 – 54  (Tugas semester 4 #12) | 95 |

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

1. Menggabungkan tabel menu, order, pelanggan, dan kategori ke dalam tabel detail dengan perintah join().

Cth :

$data = DB::table('details')

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

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

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

Artinya, di dalam $data terdapat perintah memanggil DB dengan table details.

Selanjutnya, terdapat perintah untuk join() yang pertama, dengan memilih tabel orders dan memilih kolom idorder pada tabel orders yang digabungkan dengan kolom idorder pada tabel details.

Perintah join yang kedua, dengan memilih tabel menus dan memilih kolom idmenu pada tabel menus lalu digabungkan dengan kolom idmenu pada tabel details.

Perintah join yang ketiga, dengan memilih tabel pelanggans dan memilih kolom idpelanggan pada tabel pelanggans yang digabungkan dengan kolom idpelanggan pada tabel orders.

Perintah join yang keempat, dengan memilih tabel kategoris dan memilih kolom idkategori pada tabel kategoris yang digabungkan dengan kolom idkategori pada tabel menus.

Terdapat perintah select untuk memilih semua data dari kolom orders, details, menus, pelanggans, dan kategoris.

Lalu where digunakan untuk memilih data pada kolom tglorder yang nilainya lebih besar samadengan dari $a, dan juga memilih data pada kolom tglorder yang nilainya lebih kecil sama dengan dari $b.

Dan orderBy digunakan untuk mengurutkan penampilan dengan kolom tglorder dari tabel orders dengan menggunakan asc / diurutkan dari nilai kecil ke nilai besar.

Untuk menampilkannya menggunakan fungsi get().

Nilai akan dikeluarkan dengan return dan response akan dikirimkan dalam bentuk json yang akan mengirimkan $data.

1. IdOrder sering disebut dengan faktur.
2. Mengambil data user pada LUMEN.

Cth :

public function index()

    {

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

*return* response()->json($data);

    }

Artinya, di dalam function index terdapat $data yang memanggil Model User dengan perintah where yang memilih data yang akan ditampilkan dengan apabila kolom level tidak sama dengan pelanggan, maka data akan ditampilkan.

Lalu, nilai akan dikeluarkan dan response pada $data akan dikirimkan menggunakan json.

1. Mengubah status user dari aktif menjadi banned apabila button di klik.

Cth :

<td>

                    {val.status === 1 ? (

                      <input

                        type="submit"

                        value="AKTIF"

                        className="btn btn-success btn-sm"

                        onClick={() => status(val.id)}

                      />

                    ) : (

                      <input

                        type="submit"

                        value="BANNED"

                        className="btn btn-danger btn-sm"

                        onClick={() => status(val.id)}

                      />

                    )}

                  </td>

Artinya, di dalam td terdapat pengujian yaitu dengan, apabila nilai dari status pada variabel val sama dengan 1, maka mempunyai button berwarna hijau dengan value aktif, tetapi jika nilai dari val.status tidak sama dengan 1, maka button akan berwarna merah dengan value adalah banned. Lalu, kedua button apabila di klik, akan menjalankan function status dengan parameter yang mengambil nilai dari val.id.

  async function status(id) {

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

    let stat = 0;

*if* (data[0].status === 1) {

      stat = 0;

    } *else* {

      stat = 1;

    }

    let kirim = {

      status: stat,

    };

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

  }

Artinya, terdapat const data yang memilih data pada const isi menggunakan filter dengan variabel val untuk menampung data yang di dalamnya terdapat pengujian untuk melihat nilai dari val.id apakah sama nilai nya dengan id.

Variabel stat berisi nilai awal 0.

Terdapat pengujian yaitu apabila nilai dari objek status pada const data[0].status adalah 1, maka nilai dari variabel stat akan diubah menjadi 0. Tetapi jika nilai dari objek status pada const data[0].status adalah 0, maka nilai dari variabel stat akan diubah menjadi 1.

Lalu pada variabel kirim berisi objek status dengan mengambil nilai yang disimpan ke dalam variabel stat. Jadi, ketika nilai dari data[0].status adalah 1, maka berubah menjadi 0 dimana jika nilai dari data[0].status adalah 0, maka button akan berubah menjadi merah. Dan jika nilai dari data[0].status adalah 0, maka nilai berubah menjadi 1, dimana button akan berubah menjadi hijau. Jadi, nilai perubahan akan disimpan ke dalam variabel stat, lalu akan dimasukkan ke dalam objek status pada variabel kirim.

Setelah itu, const res berisi link yang disambungkan pada perintah put untuk melakukan update data yang sudah disimpan pada variabel kirim, dimana setelah itu membaca url /user/id lalu mengirimkan variabel kirim.

1. Mengatur kondisi – kondisi ke dalam LUMEN, kondisi pertama, apabila email sudah terdaftar, maka data akan masuk dan menampilkan pesan, login anda sukses. Tetapi, jika email tidak terdaftar, maka gagal masuk dan menampilkan pesan, login anda gagal, email belom terdaftar.

Kondisi kedua, apabila email terdaftar tetapi status dari user adalah 0 maka login gagal, tetapi jika status dari user adalah 1, maka login berhasil.

Kondisi ketiga, apabila password yang dimasukkan sama dengan password yang ada pada database, maka login berhasil. Tetapi, jika password yang dimasukkan tidak sama dengan password yang ada di dalam database, maka login gagal.

**Saya Belum Mengerti**