|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran React-JS Video ke 55 – 58  (Tugas semester 4 #13) | 95 |

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

1. Melakukan login dan apabila login berhasil, maka akan diarahkan ke dalam halaman admin, tetapi jika tidak, maka akan tetap di dalam halaman login.

Cth :

let token = *await* res.data.token;

    sessionStorage.setItem("token", token);

    reset();

*if* (gettoken() != "undefined") {

      history.push("/admin");

    }

  }

  const gettoken = () => sessionStorage.getItem("token");

Artinya, terdapat variabel token yang mengambil variabel token dari variabel data yang ada di dalam variabel res.

sessionStorage.setItem() akan mengatur data ke dalam ‘token’ dengan mengambil nilai dari variabel token.

Pada function anonimus dengan const gettoken berisi sessionStorage.getItem yang mengambil nilai / data dari ‘token’.

Kemudian terdapat pengujian pada if, yaitu apabila const gettoken tidak sama dengan ‘undifined’, maka akan memasukkan link dengan history.push() yang memanggil /admin. Dimana variabel history ini berisi useHistory{} yang berisi localhost:3000. Jadi ketika login yang dimasukkan adalah benar, maka akan diarahkan ke halaman selanjutnya, yaitu /admin dengan menyambungkan dengan history dimana berisi localhost:3000.

*if* (sessionStorage.getItem("token") === "undefined") {

*return* <Redirect to="/login" />;

  }

Artinya, tetapi jika nilai token sama dengan undefined, maka halaman akan diarahkan menggunakan redirect ke dalam file login.

1. sesssionStorage() digunakan untuk menyimpan data sementara dimana data akan menghilang ketika tab ditutup atau browser ditutup.
2. setItem() digunakan untuk mengatur data / menyimpan data.
3. getItemn() digunakan untuk mengambil data dari session storage.
4. Push() digunakan untuk memasukkan link baru dan menampilkan data yang ada di dalam link tersebut.
5. sessionStorage.clear() digunakan untuk menghapus seluruh data yang telah tersimpan di sessionStorage.
6. Menambahkan button logout

Cth :

* Mengatur sessionStorage dengan menggunakan perintah setItem :

    sessionStorage.setItem("email", res.data.data.email);

    sessionStorage.setItem("level", res.data.data.level);

Artinya, yaitu pada sessionStorage pertama mengatur data pada ‘email’ dengan mengambil nilai pada variabel email pada variabel data yang ada di dalam variabel data pada variabel res.

Dan di dalam pada sessionStorage pertama mengatur data pada ‘level’ dengan mengambil nilai pada variabel level pada variabel data yang ada di dalam variabel data pada variabel res.

* Mengatur button dan mengambil nilai menggunakan sessionStorage untuk mengisi email dan level.

<li className="nav-item list-unstyled">

            Email : {sessionStorage.getItem("email")}

          </li>

          <li className="nav-item list-unstyled">

            Level : {sessionStorage.getItem("level")}

          </li>

<button

            onClick={hapus}

            className="btn btn-outline-success"

            type="submit"

          >

            LogOut

          </button>

Artinya, pada <li> pertama terdapat Email dengan mengambil data melalui sessionStorage.getItem pada ‘email’. Dan <li> kedua terdapat Level dengan mengambil data melalui sessionStorage.getItem pada ‘level’. Button logout terdapat event onClick, yaitu apabila button logout di klik, maka akan menjalankan function hapus.

* Function hapus

function hapus() {

    sessionStorage.clear();

    history.push("/login");

  }

Artinya, di dalam function hapus berisi sessionStorage.clear yang digunakan untuk menghapus semua data yang tersimpan pada sessionStorage. Setelah data terhapus, maka akan memasukkan link dengan history.push() yang memanggil /login. Jadi setelah button logout di klik, maka seluruh data yang disimpan ke dalam function sessionStorage akan dihapus dan akan langsung diarahkan ke dalam halaman login.

1. hash::make digunakan untuk membuat password yang dimasukkan menjadi acak secara otomatis sehingga tidak mudah dilacak.

Cth :

$data = [ 'password' => Hash::make($request->input('password'))];

Artinya, di dalam $data terdapat key password dengan value yaitu, hash::make yang digunakan untuk membuat password yang dimasukkan pada $request(‘password’) agar menjadi acak.

1. hash::check digunakan untuk mengecek password yang dimasukkan apakah sama dengan password asli yang ada pada database.

Cth :

Hash::check($password, $user->password)

Artinya, terdapat perintah hash::check() untuk mengecek password, pada $password apakah sama dengan password yang ada pada $user->password

1. Untuk menggunakan hash di dalam lumen, harus ditambahkan :

use Illuminate\Support\Facades\Hash;

1. Membatasi hak akses sesuai dengan level user yang login.

Cth :

* admin dapat mengakses, Kategori, Menu, Pelanggan, Order, Order-Detail, dan User Admin.
* koki dapat mengakses, Order dan Order-Detail
* kasir dapat mengakses, Order-Detail.

Cth :

<Link to={`${url}/detail`}>

            {sessionStorage.getItem("level") === "admin" ||

            sessionStorage.getItem("level") === "kasir" ||

            sessionStorage.getItem("level") === "koki" ? (

              <li className="list-group-item">Order-Detail</li>

            ) : (

              ""

            )}

          </Link>

Artinya, di dalam link detail, yaitu terdapat pengujian menggunakan if, yaitu dengan mengambil sessionStorage.getItem (‘level’) apabila sama dengan admin, kasir, dan koki, maka akan menjalankan <li> yang berisi Order-Detail.

Jika level yang dimasukkan tidak sesuai dengan sessionStorage.getItem yang diambil, maka akan menampilkan kosong.

**Saya Belum Mengerti**