|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| React JS Semester-4 Minggu ke-13 Video 55-58 | 95 |

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

1. Mengarahkan login ke dalam page admin dengan cara pada bagian function login atau function yang digunakan untuk mengarahkan submit login. Di dalam function ini dibuatkan variable bernama token seperti ini let token = await res.data.token; yang berarti variable token akan diisi dengan data login bagian tokennya. Lalu ada sessionStorage.setItem("token", token); yang berarti data dari variable token akan dimasukkan ke dalam session token. Session ini sebagai penyimpanan sementara, jadi hanya berfungsi ketika login dijalankan, jika login sudah tidak dijalankan maka session tidak digunakan. Dan session token dimasukkan ke dalam const gettoken. Setelah itu const gettoken akan diuji menggunakan if seperti ini

if (gettoken() != "undefined") {

      history.push("/admin");

    }

Yang berarti apabila gettoken tidak sama dengan undefined atau token nya benar sesuai dengan user yang telah terdaftar di database, maka akan diarahkan ke dalam page admin menggunakan history.push. History ini adalah link yang sudah ada sebelumnya atau riwayat link terakhir dan dipush berarti ditambahkan dengan apa yang ada di dalam kurung nya push. Dan apabila login gagal atau token undefined maka di dalam Back.js nya ditambahkan seperti ini

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

    return <Redirect to="/login" />;

  }

Yang berarti jika session token sama dnegan undefined atau tokennya tidak ditemukan dari database maka dia akan redirect atau mengarahkan lagi ke dalam page loginnya.

2. Mengatur button logout yang ditambahkan di nav menggunakan bootstrap. Pada code button ditambahkan onClick yang menyambungkan dengan function hapus. Di dalam function hapus nya diisi dengan

function hapus() {

    sessionStorage.clear();

    history.push("/login");

  }

Yang berarti session yang menyimpan data user sementara tadi di clear, jadi session nya sudah tidak menyimpan data apapun lagi. Setelah logout akan diarahkan ke dalam page login sesuai dengan history.push nya. Jadi ketika mencoba masuk ke page admin lagi tidak bisa karena sessionnya sudah tidak ada.

3. Mengisi nav bagian li text Email dan Level sesuai dengan akun yang digunakan untuk login. Caranya pada bagian file Login.js di function loginnya ditambahkan

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

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

yang berarti ditambahkan session bernama email dan level yang isinya diambil dari data email sesuai dengan akun yang dimasukkan untuk login. Lalu pada bagian li Email ditambahkan : {sessionStorage.getItem("email")} dan li Level nya ditambahkan {sessionStorage.getItem("level")} yang berarti akan ditambahkan session email dan session level untuk ditampilkan dalam browser nya. Apabila di logout maka semua session, dari session login, email, dan level akan terhapus semua dan session siap diisi dengan data akun baru untuk diloginkan.

4. Menambahkan Hash pada password yang digunakan ketika mendaftar atau menambahkan user. Caranya pada bagian Login.php atau file lumen nya yaitu di function register pada bagian $data

'password' => Hash::make($request->input('password')),

Jadi ketika ada data baru yang dipost untuk menambahkan user, pada bagian password nya akan di hash secara otomatis. Jadi apabila password dilihat dari database bukan lagi berisi password aslinya tetapi sebuah code-code yang diberikan oleh hash. Selain menggunakan Hash:make ada juga Hash:check yang digunakan untuk mengecek atau mencocokkan password yang dimasukkan untuk login apakah sama dengan password yang terhash di dalam database nya. Untuk menggunakan Hash diperlukan use Illuminate\Support\Facades\Hash;

5. terdapat window.location.reload(); yang berfungsi untuk mereload page secara otomatis ketika sudah melakukan login(tergantung penempatannya untuk apa).

6. Menambahkan hak ases untuk setiap user nya sesuai dengan level nya. Misalkan admin mengakses side atau menu-menu tertentu seperti user-admin, maka pada Side.js yang mengatur menu-menu nya ditambahkan seperti ini

<Link to={`${url}/user`}>{sessionStorage.getItem("level") === "admin" ? (<li className="list-group-item">User Admin</li>) : ("")}</Link>

Yang berarti apabila session level nya adalah admin maka Link untuk User Admin hanya akan muncul ketika yang login admin, jika selain admin yang login, menu user-admin nya tidak akan muncul. Apabila satu menu digunakan untuk tiga user yaitu admin, koki, dan kasir maka codenya seperti ini

<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>

Jadi ada session untuk pengaturan level admin, koki, dan kasir yang dipisahkan dengan tanda || saja.

**Saya Belum Mengerti**

1.