

Nama : Olvi Lora S

NPM : 1606954943

Tutorial 8

Latihan

1. Menambahkan tombol Sign Out dan mencetak nama user yang sedang login

- Menambahkan code berikut di view.html

```
<body>
<h2 th:text=" 'Login as ' + ${#httpServletRequest.remoteUser}">Login as</h2>
<form th:action="@{/logout}" method="post">
  <input type="submit" value="Sign Out"/>
</form>
<h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
<h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
<h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
</body>
```

- Apabila mengakses url <http://localhost:8080/student/view/12345> maka akan tampil seperti berikut

Home Daftar Mahasiswa Menambah Mahasiswa

Login as admin

Sign Out

NPM = 12345

Name = olvi

GPA = 3.4

Mata Kuliah APAP

- Apabila tombol Sign Out diklik maka akan keluar dari aplikasi dan tampil pesan You have been logged out seperti yang didefinisikan di view login.html

```
<div th:if="${param.logout}">
  You have been logged out.
</div>
```

You have been logged out.

User Name :

Password:

2. Membuat user baru

- Menambah code berikut untuk menambahkan username dan password baru di WebSecurityConfig.java dengan roles sebagai USER

```
@Autowired
public void configureGlobal (AuthenticationManagerBuilder auth) throws Exception
{
    auth.inMemoryAuthentication()
        .withUser("admin").password("admin")
        .roles("ADMIN")
        .and().withUser("olvi").password("olvi")
        .roles("USER");
}
```

- Kemudian login menggunakan akun olvi, maka akan berhasil masuk dan apabila mengakses view all student maka akan ditampilkan notifikasi user login sebagai olvi

Home Daftar Mahasiswa Menambah Mahasiswa

Login as olvi

All Students

No	NPM	Name	GPA	Status	Action	Action
1	12345	olvi	3.4	Sangat Memuaskan!	Delete Data	Update Data
2	1506737823	Kermie	3.32	Sangat Memuaskan!	Delete Data	Update Data
3	1506737824	Ellene	2.89	Sangat Memuaskan!	Delete Data	Update Data
4	1506737825	Delmer	2.58	Sangat Memuaskan!	Delete Data	Update Data

3. Menampilkan menu view student by npm

- Pada file index.html saya menambahkan menu baru yang dapat diakses oleh pengguna dengan role user. Menu nya adalah View Student by NPM

```
<body>
    <h2>Welcome To Tutorial 05 APAP</h2>
    <a href="/Login">Login</a>
    <!-- <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
    <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
    <a th:if="${#httpServletRequest.isUserInRole('USER')}" href="/student/viewform">View Student by NPM</a><br/>
    <div th:replace="fragments/fragment :: footer"></div>
</body>
</html>
```

- Jika diakses melalui URL <http://localhost:8080>, tampilannya adalah sebagai berikut

Welcome To Tutorial 05 APAP

[Login](#)

[View Student by NPM](#)

Mata Kuliah APAP

- Menu tersebut jika diklik akan redirect ke mapping /student/viewform. Oleh sebab itu saya menambahkan mapping /student/viewform di StudentController

```
@RequestMapping("/student/viewform")
public String viewForm (Model model)
{
    return "form-view-student";
}
```

- Code nya memanggil view form-view-student.html yang berisi html sebagai berikut.

```
1 <!DOCTYPE HTML>
2 <html xmlns="http://www.w3.org/1999/xhtml"
3     xmlns:th="http://www.thymeleaf.org">
4 <head>
5
6 <title>Add student</title>
7 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
8 </head>
9
10 <body>
11
12     <h1 class="page-header">Add Student</h1>
13
14     <form action="/student/view" method="get">
15         <div>
16             <label for="npm">NPM</label> <input type="text" name="npm" />
17         </div>
18         <div>
19             <button type="submit" name="action" value="save">View</button>
20         </div>
21     </form>
22
23 </body>
24
25 </html>
```

- Apabila view nya diakses melalui url maka tampilannya adalah sebagai berikut

< > ↺ ☰ | localhost:8080/student/viewform

Add Student

NPM

View

- Kemudian isi npm yang ingin dilihat di field yang telah disediakan dan klik tombol View maka akan ditampilkan data sesuai npm yang dimasukkan.

Home Daftar Mahasiswa Menambah Mahasiswa

Login as lora

Sign Out

NPM = 12345

Name = olvi

GPA = 3.4

Mata Kuliah APAP

4. Membuat agar role ADMIN dapat melihat view all student dan view student by npm juga
- Command role untuk USER untuk menampilkan view student by npm sehingga fitur tersebut dapat diakses oleh pengguna dengan ADMIN

```

@Override
protected void configure(HttpSecurity http) throws Exception{
    http
        .authorizeRequests()
        .antMatchers("/").permitAll()
        .antMatchers("/student/viewall").hasRole("ADMIN")
        // .antMatchers("/student/view/**").hasRole("USER")
        .anyRequest().authenticated()
        .and()
        .formLogin()
        .loginPage("/login")
        .permitAll()
        .and()
        .logout()
        .permitAll();
}

```

- Kemudian di file index.html kondisi untuk menampilkan menu View Student by NPM dihapuskan sehingga bisa diakses oleh semua role pengguna.

```

<body>
    <h2>Welcome To Tutorial 05 APAP</h2>
    <a href="/Login">Login</a>
    <!-- <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
    <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
    <a href="/student/viewform">View Student by NPM</a><br/>
    <div th:replace="fragments/footer":: footer"></div>
</body>

```

- Apabila pengguna admin masuk ke aplikasi maka akan ditampilkan menu View All Student dan View Student By NPM

Home Daftar Mahasiswa Menambah Mahasiswa

Welcome To Tutorial 05 APAP

Login

View All Student

View Student by NPM

Mata Kuliah APAP

- Apabila pengguna user masuk ke aplikasi maka akan ditampilkan View Student By NPM

Welcome To Tutorial 05 APAP

[Login](#)

[View Student by NPM](#)

Mata Kuliah APAP

Yang saya pelajari dalam pengerjaan Tutorial 8 ini adalah sebagai berikut:

- Saya mempelajari tentang Web Security, yaitu mengimplementasikan tingkatan pengguna yang berbeda sesuai kebutuhan. Pada tutorial ini saya belajar membuat 2 tingkatan role yaitu sebagai ADMIN dan USER. Untuk mendefinisikan role tersebut menyediakan class `WebSecurityConfig.java` yang digunakan untuk mengkonfigurasi url yang dapat diakses oleh semua role pengguna dan url tertentu yang dapat diakses oleh role tertentu
- Di class `WebSecurityConfig.java` tersebut juga didefinisikan konfigurasi untuk meng-set username dan password yang disimpan di memory melalui method `configureGlobal`. Kemudian terdapat konfigurasi untuk meng-set pengguna yang login berdasarkan data yang tersimpan di database melalui method `configAuthentication`.
- Untuk mengimplementasikan security, hal yang pertama dilakukan adalah menambahkan dependencies `spring boot starter security`