

Implementasi Web Security pada Spring Boot Framework

Pada lab ini, yang dipelajari adalah mengembangkan sebuah sistem dengan penambahan fitur autentikasi dan otorisasi berdasarkan tingkat pengguna sebagai bagian dari implementasi web security. Fitur ini berfungsi untuk membatasi akses user terhadap pages yang diotentikasi oleh system.

Ada dua model inisialisasi user yang dipelajari:

1. Melalui aplikasi, bersifat statis karena daftar tipe user(username, password, dan role) terbatas pada list yang diinput di aplikasi
2. Melalui database, bersifat dinamis karena dapat ditambahkan pada tabel (users, dan user_role).

Ada dua jenis user yaitu admin dan user. Hasil otentikasi dapat dilihat pada page 'viewall student' dan 'view student by npm'

Latihan 1

1. Menambah Sign out pada dan cetak nama user pada halaman student/view/{npm}

```
<h2 th:text=" 'Login as ' + ${#httpServletRequest.remoteUser}">Login as</h2>
<form th:action="@{/Logout}" method="post">
  <input type="submit" value="Sign Out"/>
</form>
```

2. Membuat user baru dengan username "user", password "user, dan role "User"

```
auth.inMemoryAuthentication()
    .withUser("user").password("user")
    .roles("USER");
```

Latihan 2

1. Menampilkan menu view student by npm beserta input field dan button untuk memasukkan nomor npm student yang ingin dicari pada halaman index user dengan role USER
 - a. Tambah link baru yang akan memanggil form-search berdasarkan NPM pada index.html

```
<a th:if="${#httpServletRequest.isUserInRole('USER')}"
href="/student/search">View Student by NPM</a><br>
```

- b. Buat form-search.html

Pada html berikut, page student/view akan dipanggil ketika button submit diklik.

```
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
<title>Search student</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>

<body>
    <h2 th:text=" 'Login as ' +
    ${#httpServletRequest.remoteUser}">Login as</h2>
    <form th:action="@{/Logout}" method="post">
        <input type="submit" value="Sign Out"/>
    </form>
    <h1 class="page-header">Searching Data Student by NPM</h1>

    <form action="/student/view" >
        <div>
            <label for="npm">NPM</label> <input type="text"
name="npm" />
        </div>
        <div>
            <button type="submit" name="action"
value="search">Search</button>
        </div>
    </form>
    <div th:replace="fragments/fragment_2 :: footer"></div>
</body>

</html>
```

- c. Page StudentController > tambah mapping berikut untuk memanggil form-search

```
@RequestMapping("/student/search")
public String search ()
{
    return "form-search";
}
```

2. Membuat user dengan role ADMIN dapat melihat view all student dan view student by npm
a. Pada WebSecurityConfig > comment line berikut untuk mengizinkan admin mengakses view

```
// .antMatchers("/student/view/**").hasRole("USER")
```

- b. Pada index.html > tambahkan line berikut untuk menampilkan url dengan otorisasi admin

```
<a th:if="${#httpServletRequest.isUserInRole('ADMIN')}"
href="/student/search">View Student by NPM</a><br>
```