

Nama : Haryo Parigroho  
NPM : 1506689603  
Kelas : B

Hal yang saya pelajari dari tutorial ini adalah mengenai web security otentikasi dan otorisasi program dengan menggunakan Spring Boot Framework. Untuk menggunakan fitur security program harus menggunakan dependency Spring Boot Starter Security.

Selanjutnya untuk mengatur mengenai otorisasi dan otentikasi, perlu dibuat class yang meng-extends WebSecurityConfigureAdapter dan meng-overide method configure. Method configure ini adalah yang mengatur siapa-siapa saja yang dapat mengakses halaman tertentu. Lalu method configAuthentication dengan parameter AuthenticationManagerBuilder adalah yang merekam siapa saja usernya dari database.

Lalu pada view, untuk mengetahui siapa yang login digunakan thymeleaf `#httpServletRequest.isUserInRole()`.

## Latihan

1. Menambahkan tombol Sign Out dan cetak nama user untuk halaman course/viewall  
Pertama-tama saya membuat mapping untuk "course/viewall" lalu membuat method yang mengambil semua course pada database.

```
@Select("select * from course")
@Results(value = {
    @Result(property="idCourse", column="id_course", javaType=String.class),
    @Result(property="name", column="name", javaType=String.class),
    @Result(property="credits", column="credits", javaType=Integer.class),
    @Result(property="students", column="id_course", javaType=List.class, many=@Many(select="selectStudents"))
})
List<CourseModel> selectAllCourse();
```

Method mapping yang akan mengembalikan halaman view-all-course.html:

```
@RequestMapping("/course/viewall")
public String viewall(Model model) {
    List<CourseModel> courses = courseDAO.selectAllCourse();
    model.addAttribute("courses", courses);
    return "view-all-course";
}
```

Halaman view-all-course.html menampilkan informasi user dan logout:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head><title>View All Courses</title></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>All Courses</h1>
```

Nama : Haryo Parigroho

NPM : 1506689603

Kelas : B

2. Membuat user baru dengan username = admin, password = admin, dan role = ADMIN

```
"USER");  
auth.inMemoryAuthentication().withUser("admin").password("admin").roles(  
"ADMIN"); }
```

Saya menambahkan inMemoryAuthentication untuk melakukan ini, jadi terdapat 2 inMemoryAuthentication.

3. Mengubah agar ADMIN hanya bisa view all course dan view all student dan USER hanya bisa view all student serta mengubah tampilannya agar sesuai.

Untuk mengerjakan ini saya menggunakan method `hasAnyRole("USER", "ADMIN")` untuk `antMatchers("/student/viewall")` dan `hasRole("ADMIN")` untuk `antMatchers("/course/viewall")`

```
@Override  
protected void configure(HttpSecurity http) throws Exception {  
    http.authorizeRequests().antMatchers("/").permitAll().antMatchers("/course/viewall").hasRole("ADMIN")  
        .antMatchers("/student/viewall").hasAnyRole("USER", "ADMIN").anyRequest().authenticated().and().formL  
        .loginPage("/login").permitAll().and().logout().permitAll();  
}
```

Untuk tampilan, saya menambahkan or pada `th:if`,

```
<a href="/login">Login</a> <br />  
<a th:if="${#httpServletRequest.isUserInRole('ADMIN')}"  
    href="/course/viewall">View All Course</a>  
<br />  
<a th:if="${#httpServletRequest.isUserInRole('USER')} or #httpServletRequest.isUserInRole('ADMIN')}"  
    href="/student/viewall">View All Student</a>  
<br />
```

Jadi jika user adalah ADMIN menu yang muncul adalah View All Course dan View All Student sedangkan jika user adalah USER menu yang muncul adalah View All Student.