

Hal-hal yang dipelajari

- Membuat fungsi login dan logout dengan database dan tanpa database
- Mendapatkan informasi request client dengan HttpServletRequest
- Membatasi akses pada suatu halaman berdasarkan role dengan menggunakan fungsi AntMatchers dan hasRole/hasAuthority.

Ditambahkan dependency Security pada Pom.xml

Dibuat class baru dengan nama WebSecurityConfig.java

```
17 public class WebSecurityConfig extends WebSecurityConfigurerAdapter{
18
19     @Override
20     protected void configure(HttpSecurity http) throws Exception
21     {
22         http
23             .authorizeRequests()
24             .antMatchers("/").permitAll()
25             .antMatchers("/student/viewall").hasAuthority("ADMIN")
26             .antMatchers("/student/view/**").hasAnyAuthority("USER", "ADMIN")
27             .antMatchers("/student/viewall").hasRole("ADMIN")
28             .antMatchers("/student/view/**").hasAnyRole("USER", "ADMIN")
29             .anyRequest().authenticated()
30             .and()
31             .formLogin()
32             .loginPage("/login")
33             .permitAll()
34             .and()
35             .logout()
36             .permitAll();
37     }
38
39     @Autowired
40     public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception
41     {
42         auth.inMemoryAuthentication()
43             .withUser("admin").password("{noop}admin")
44             .authorities("ADMIN");
45
46         auth.inMemoryAuthentication()
47             .withUser("user").password("{noop}user")
48             .authorities("USER");
49     }
50
51     @Bean
52     public static NoOpPasswordEncoder passwordEncoder() {
53         return (NoOpPasswordEncoder) NoOpPasswordEncoder.getInstance();
54     }
55 }
```

Dibuat halaman login dengan nisi sebagai berikut

```
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org" xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
3 <head>
4 <title> Login Page </title> </head>
5 <body>
6 <div th:if = "${param.error}" >
7 Invalid username and password. </div>
8 <div th:if = "${param.logout}" > You have been logged out.
9 </div>
10 <form th:action = "@{/login}" method = "post" >
11 <div>
12 <label> User Name : <input type = "text" name = "username" /></label></div>
13 <div><label> Password: <input type = "password" name = "password" /></label></div>
14 <div><input type = "submit" value = "Sign In" /></div>
15
16 </form>
17 </body>
18 </html>
```

Dibuat class PageController dengan isi sebagai berikut :

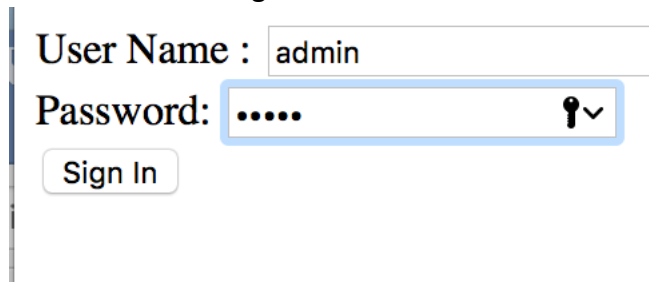
```
1 package com.fpamungkas.tutorial08.controller;
2
3 import org.springframework.stereotype.Controller;
4
5
6 public class PageController {
7
8     @RequestMapping("/")
9     public String index() {
10         return "index";
11     }
12
13
14 }
15
```

Dijalankan program dan dibuka halaman index dengan hasil seperti pada gambar dibawah

Welcome

[Login](#)

Dibuka halaman login



User Name : admin

Password:

Sign In

Sign Out dan Nama Username

Ditambahkan baris berikut pada file viewAll.html untuk mendapatkan informasi request client, dimana akan ditampilkan username dari user.

```
<h2 th:text = " 'Login as ' + ${#httpServletRequest.remoteUser}" > Login as </h2>
```

Ditambahkan baris berikut untuk menyematkan fungsi logout

```
<form th:action = "@{/logout}" method = "post">  
<input type = "submit" value = "Sign Out" /> </form>
```

Latihan

1.

Dilakukan hal yang sama untuk halaman student/view/npm

Ditambahkan baris berikut pada file viewAll.html untuk mendapatkan informasi request client, dimana akan ditampilkan username dari user.

```
<h2 th:text = " 'Login as ' + ${#httpServletRequest.remoteUser}" > Login as </h2>
```

Ditambahkan baris berikut untuk menyematkan fungsi logout

```
<form th:action = "@{/logout}" method = "post">  
<input type = "submit" value = "Sign Out" /> </form>
```

2.

Ditambahkan baris berikut pada fungsi ConfigureGlobal

```
auth.inMemoryAuthentication()  
    .withUser("user").password("{noop}user")  
    .authorities("USER");
```

Membuat Menu dan Memanfaatkan Role

Ditambahkan baris berikut pada fungsi Configure agar halaman untuk **view all student** hanya dapat dilihat oleh *user* dengan **role ADMIN** dan halaman untuk **view student by npm** hanya dapat dilihat oleh *user* dengan **role USER**.

```
.antMatchers("/student/viewall").hasAuthority("ADMIN")  
.antMatcher("/student/view/**").hasAuthority("USER");
```

Ditambahkan baris berikut pada halaman index.html agar jika pengguna *login* dengan **role ADMIN** maka hanya akan muncul link ke **view all student**

```
<a th:href="#{httpServletRequest.isUserInRole('ADMIN')}" href = "/student/viewall" > View All Student </a><br/>
```

Mengambil Username dan Password dari Database

Pada class WebSecurityConfig, beri comment pada method configureGlobal, dan dtambahkan baris dan method berikut:

```
@Autowired  
DataSource dataSource;  
  
@Autowired  
public void configureAuthentication(AuthenticationManagerBuilder auth) throws Exception{  
    auth.jdbcAuthentication().dataSource(dataSource)  
        .usersByUsernameQuery("select username, password, enabled from users where username = ?")  
        .authoritiesByUsernameQuery("select username, role from user_roles where username = ?");  
}
```

LATIHAN

1.

Ditambahkan baris berikut pada halaman index.html

```
<div th:if = "${httpServletRequest.isUserInRole('USER')}"><a href="#"> Cari </a><input type="text" value="Cari" /><input type="button" value="Cari" /></div><br/>
```

2.

Ditambahkan baris penggunaan fungsi berikut pada function configure

```
.antMatchers("/student/view/**").hasAnyAuthority("USER", "ADMIN")  
  
.antMatchers("/student/view/**").hasAnyRole("USER", "ADMIN")
```