

Nama: Iman Alfathan Yudhanto

NPM: 1406623524

APAP-A

Write-up Tutorial 8

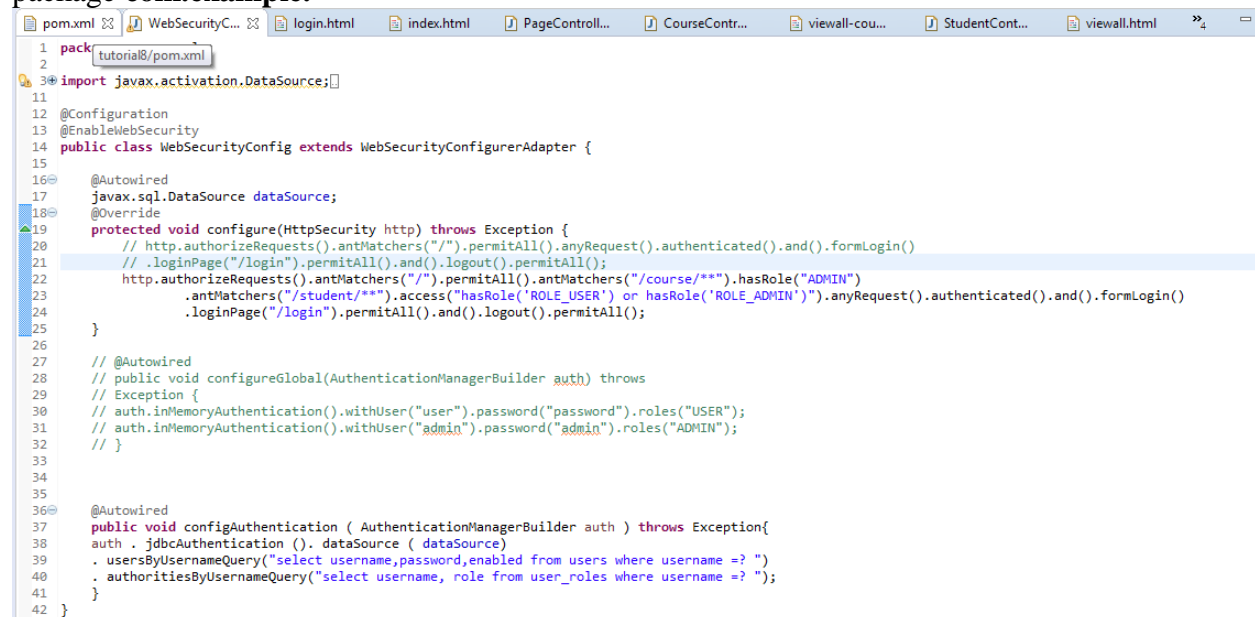
Latihan

1. Buatlah hal yang sama dengan menambahkan tombol Sign Out dan cetak nama user untuk halaman course/viewall.

Hal yang penulis lakukan pertama kali adalah menambahkan dependency untuk security pada pom.xml yang dimiliki.

```
<dependency>
  <groupId> org.springframework.boot </groupId>
  <artifactId> spring-boot-starter-security </artifactId>
</dependency>
```

Dependency di atas berguna untuk menambahkan fungsi security pada Springboot. Setelah membuat dependency, penulis membuat class baru yaitu, **WebSecurityConfig.java** pada package **com.example**.



```
1  pack tutorial8/pom.xml
2
3  import javax.activation.DataSource;
11
12 @Configuration
13 @EnableWebSecurity
14 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
15
16     @Autowired
17     javax.sql.DataSource dataSource;
18
19     @Override
20     protected void configure(HttpSecurity http) throws Exception {
21         // http.authorizeRequests().antMatchers("/").permitAll().anyRequest().authenticated().and().formLogin()
22         // .loginPage("/login").permitAll().and().logout().permitAll();
23         http.authorizeRequests().antMatchers("/").permitAll().antMatchers("/course/**").hasRole("ADMIN")
24             .antMatchers("/student/**").access("hasRole('ROLE_USER') or hasRole('ROLE_ADMIN')").anyRequest().authenticated().and().formLogin()
25             .loginPage("/login").permitAll().and().logout().permitAll();
26
27         // @Autowired
28         // public void configureGlobal(AuthenticationManagerBuilder auth) throws
29         // Exception {
30         //     auth.inMemoryAuthentication().withUser("user").password("password").roles("USER");
31         //     auth.inMemoryAuthentication().withUser("admin").password("admin").roles("ADMIN");
32         // }
33
34
35
36     @Autowired
37     public void configureAuthentication ( AuthenticationManagerBuilder auth ) throws Exception{
38         auth . jdbcAuthentication (). dataSource ( dataSource )
39             . usersByUsernameQuery ("select username,password,enabled from users where username =? ")
40             . authoritiesByUsernameQuery ("select username, role from user_roles where username =? ");
41     }
42 }
```

Pada kelas tersebut, terdapat 2 method, yaitu configureAuthentication dan configure. Method configureAuthentication berfungsi dalam memberi akses ke url yang telah dibuat. Akses url dipengaruhi oleh *role* milik user. Sedangkan method configure berfungsi dalam menyimpan autentikasi. Autentikasi diambil dari database yang mana ada query untuk memanggil database pada method tersebut.

```

pom.xml WebSecurityC... login.html index.html PageControll... CourseContr... viewall-cou...
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml"
3     xmlns:th="http://www.thymeleaf.org" xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
4 <head>
5 <title>Login Page</title>
6 </head>
7 <body>
8
9 <div th:if="${param.error}">Invalid username and password.</div>
10 <div th:if="${param.logout}">You have been logged out.</div>
11 <form th:action="@{/login}" method="post">
12 <div>
13 <label> User Name : <input type="text" name="username" />
14 </label>
15 </div>
16 <div>
17 <label> Password: <input type="password" name="password" />
18 </label>
19 </div>
20 <div>
21 <input type="submit" value="Sign In" />
22 </div>
23 </form>
24
25 </body>
26 </html>

```

Setelah itu, penulis membuat halaman html untuk melakukan login.

```

pom.xml WebSecurityC... login.html index.html PageControll... CourseContr... viewall-cou... Studer
1 <html xmlns:th="http://www.thymeleaf.org">
2 <head>
3 <title>Index</title>
4 </head>
5 <body>
6 <h2>Welcome to APAP Tutorial 8</h2>
7 <a href="/login"> Login </a>
8 <p>
9 <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/course/viewall"> ViewAll Course </a><br/>
10 <a th:if="${#httpServletRequest.isUserInRole('USER')}" href="/student/viewall"> ViewAll Student </a><br/></p>
11 </body>
12 </html>

```

Tidak lupa penulis menambahkan beberapa kode untuk fitur menu di index.html. Jika pengguna login dengan role USER maka hanya akan muncul link ke view all student, sedangkan jika pengguna login dengan *role* ADMIN maka akan muncul link ke view all course. Sama halnya jika user memiliki dua *role*.

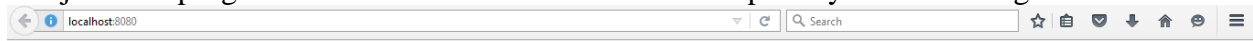
```

pom.xml WebSecurityC... login.html index.html PageControll...
1 package com.example.controller;
2
3 import org.springframework.stereotype.Controller;
4
5
6 @Controller
7 public class PageController {
8     @RequestMapping("/")
9     public String index() {
10         return "index";
11     }
12
13     @RequestMapping("/login")
14     public String login() {
15         return "login";
16     }
17 }

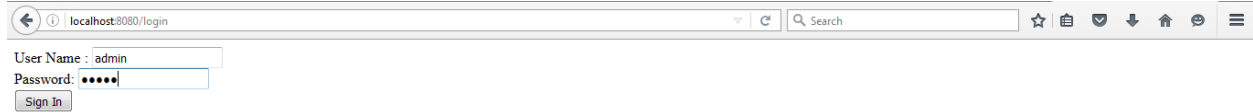
```

Yang penulis lakukan berikutnya adalah membuat kelas PageController. Lalu penulis

menjalankan program dan buka **localhost:8080** maka tampilannya adalah sebagai berikut:



Halaman index.html ketika berhasil di akses.



Halaman login.html ketika berhasil di akses. Pada kali ini, penulis login sebagai admin yang memiliki role admin.



Halaman ketika penulis berhasil melakukan login. Lalu penulis ingin mengakses halaman View all Course.



View All Course

No	ID Course	Nama	Student yang mengambil
1	aota	Apresiasi Orang yang Tidak Bisa Apresiasi	<ul style="list-style-type: none">• 29-Munaroh• 25-Kuma
2	bodoh	Management Kebodohan	<ul style="list-style-type: none">• andi-19
3	CSC123	PSP	<ul style="list-style-type: none">• 1-Chanek Jr.• 2-Valdi Rachman• 12-Sanklin• 29-Munaroh• 29-Munaroh
4	CSC124	SDA	<ul style="list-style-type: none">• 1-Chanek Jr.• 2-Valdi Rachman

Gambar di atas merupakan tampilan ketika penulis berhasil masuk ke halaman view All Course. Pada bagian atas terdapat penjelasan bahwa penulis masuk sebagai admin. Di bawahnya terdapat tombol sign out.



Tampilan ketika penulis berhasil melakukan signout.



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sat Nov 18 21:48:56 ICT 2017

There was an unexpected error (type=Forbidden, status=403).

Access is denied

Jika user dengan role “user” ingin mengakses halaman yang dilarang oleh sistem, maka akan muncul pemberitahuan seperti gambar di atas.

2. Buatlah user baru dengan username “admin”, password “admin”, dan role “ADMIN”. Cobakan apakah bisa login dengan akun tersebut.

Untuk membuat user baru dengan username dan password yang ada di soal. Cukup menambahkan kode pada bagian pada kelas WebSecurityConfig.

```
1 package com.example;
2
3 import javax.activation.DataSource;
4
5 @Configuration
6 @EnableWebSecurity
7 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
8
9     @Autowired
10    javax.sql.DataSource dataSource;
11
12    @Override
13    protected void configure(HttpSecurity http) throws Exception {
14        // http.authorizeRequests().antMatchers("/").permitAll().and().formLogin()
15        // .loginPage("/login").permitAll().and().logout().permitAll();
16        http.authorizeRequests().antMatchers("/").permitAll().antMatchers("/course/**").hasRole("ADMIN")
17        .antMatchers("/student/**").access("hasRole('ROLE_USER') or hasRole('ROLE_ADMIN')").anyRequest().authenticated().and().formLogin()
18        .loginPage("/login").permitAll().and().logout().permitAll();
19    }
20
21    // @Autowired
22    // public void configureGlobal(AuthenticationManagerBuilder auth) throws
23    // Exception {
24    //     auth.inMemoryAuthentication().withUser("user").password("password").roles("USER");
25    //     auth.inMemoryAuthentication().withUser("admin").password("admin").roles("ADMIN");
26    // }
27
28    @Autowired
29    public void configAuthentication ( AuthenticationManagerBuilder auth ) throws Exception{
30        auth . jdbcAuthentication ( ). dataSource ( dataSource )
31        . usersByUsernameQuery ("select username,password,enabled from users where username =? ")
32        . authoritiesByUsernameQuery ("select username, role from user_roles where username =? ");
33    }
34
35    }
36 }
```

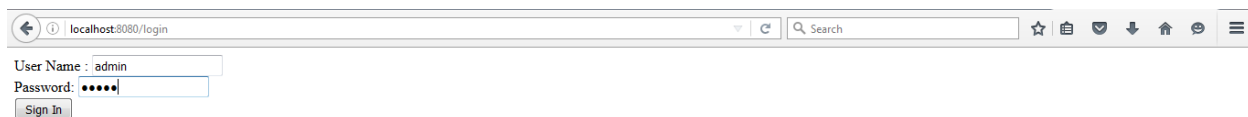
Cara menambahkan secara manual, dapat dilihat pada bagian yang di-comment dan diberi lingkaran warna merah. Jika ingin meng-config secara database, dapat dilihat pada method configAuthentication.

3. Ubah agar user dengan role ADMIN hanya dapat melihat view all course dan view all student. Sedangkan user dengan role USER hanya dapat melihat view all student saja. Anda juga harus mengubah menu di depan.

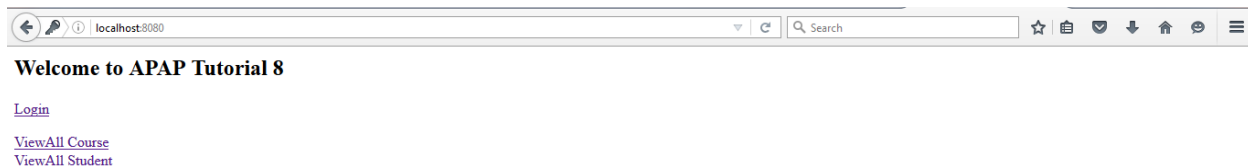
Untuk menambahkan url yang bisa diakses oleh dua role, maka bisa dengan menambahkan kode access(hasRole(Role) or hasRole(Role)).

```
pom.xml WebSecurityC... login.html index.html PageControll... CourseContr... viewall-cou... StudentCont... viewall.html 4
1 package com.example;
2
3 import javax.activation.DataSource;
11
12 @Configuration
13 @EnableWebSecurity
14 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
15
16     @Autowired
17     javax.sql.DataSource dataSource;
18     @Override
19     protected void configure(HttpSecurity http) throws Exception {
20         // http.authorizeRequests().antMatchers("/").permitAll().anyRequest().authenticated().and().formLogin()
21         // .loginPage("/login").permitAll().and().logout().permitAll();
22         http.authorizeRequests().antMatchers("/").permitAll().antMatchers("/course/**").hasRole("ADMIN")
23         .antMatchers("/student/**").access("hasRole('ROLE_USER') or hasRole('ROLE_ADMIN')").anyRequest().authenticated().and().formLogin()
24         .loginPage("/login").permitAll().and().logout().permitAll();
25     }
26
27     // @Autowired
28     // public void configureGlobal(AuthenticationManagerBuilder auth) throws
29     // Exception {
30     //     auth.inMemoryAuthentication().withUser("user").password("password").roles("USER");
31     //     auth.inMemoryAuthentication().withUser("admin").password("admin").roles("ADMIN");
32     // }
33
34
35
36     @Autowired
37     public void configAuthentication ( AuthenticationManagerBuilder auth ) throws Exception{
38         auth . jdbcAuthentication (). dataSource ( dataSource )
39         . usersByUsernameQuery ("select username,password,enabled from users where username =? ")
40         . authoritiesByUsernameQuery ("select username, role from user_roles where username =? ");
41     }
42 }
```

Fungsi kode tersebut untuk menambahkan url yang bisa diakses oleh banyak Role.



Halaman login.html ketika berhasil di akses. Pada kali ini, penulis login sebagai admin yang memiliki role admin.



Maka akan muncul dua menu yang sudah disetting sesuai role-nya.

localhost:8080/course/viewall

Login as admin

Sign Out

View All Course

No	ID Course	Nama	Student yang mengambil
1	aota	Apresiasi Orang yang Tidak Bisa Apresiasi	<ul style="list-style-type: none"> • 29-Munaroh • 25-Kuma
2	bodoh	Management Kebodohan	<ul style="list-style-type: none"> • andi-19
3	CSC123	PSP	<ul style="list-style-type: none"> • 1-Chanek Jr. • 2-Valdi Rachman • 12-Sanklin • 29-Munaroh • 29-Munaroh
4	CSC124	SDA	<ul style="list-style-type: none"> • 1-Chanek Jr. • 2-Valdi Rachman

user yang memiliki role admin dapat mengakses halaman course/viewall.

localhost:8080/student/viewall

Login as admin

Sign Out

All Students

No. 1
NPM = 1
Name = Chanek Jr.
GPA = 3.41
Kuliah yang diambil
<ul style="list-style-type: none"> • PSP-4 sks • SDA-3 sks
Delete Data Update Data Update Data Object
No. 2
NPM = 11
Name = Dudung

user yang memiliki role admin dapat mengakses halaman course/viewall.

Hal yang dipelajari:

- Membuat autentikasi dengan menggunakan method-method security pada Spring.
- Membuat autentikasi secara manual (tanpa database) atau berdasarkan user yang tersimpan di database dan cara mengakses user beserta role-nya.
- Mengetahui pembuatan method untuk membatasi role mana yang bisa mengakses sebuah url.

- Mengetahui cara membuat logout pada Spring. Sebelumnya penulis hanya paham membuat logout pada php.
- Menampilkan nama/keterangan mengenai user yang melakukan login.