



Write Tututorial 8

APAP-B

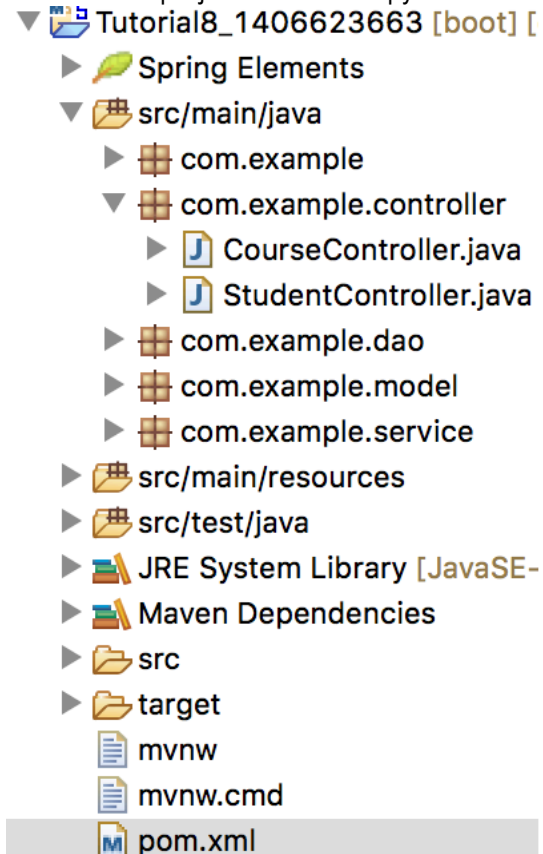
Reitasya Devina

1406623663

Simple Web Security

LOGIN

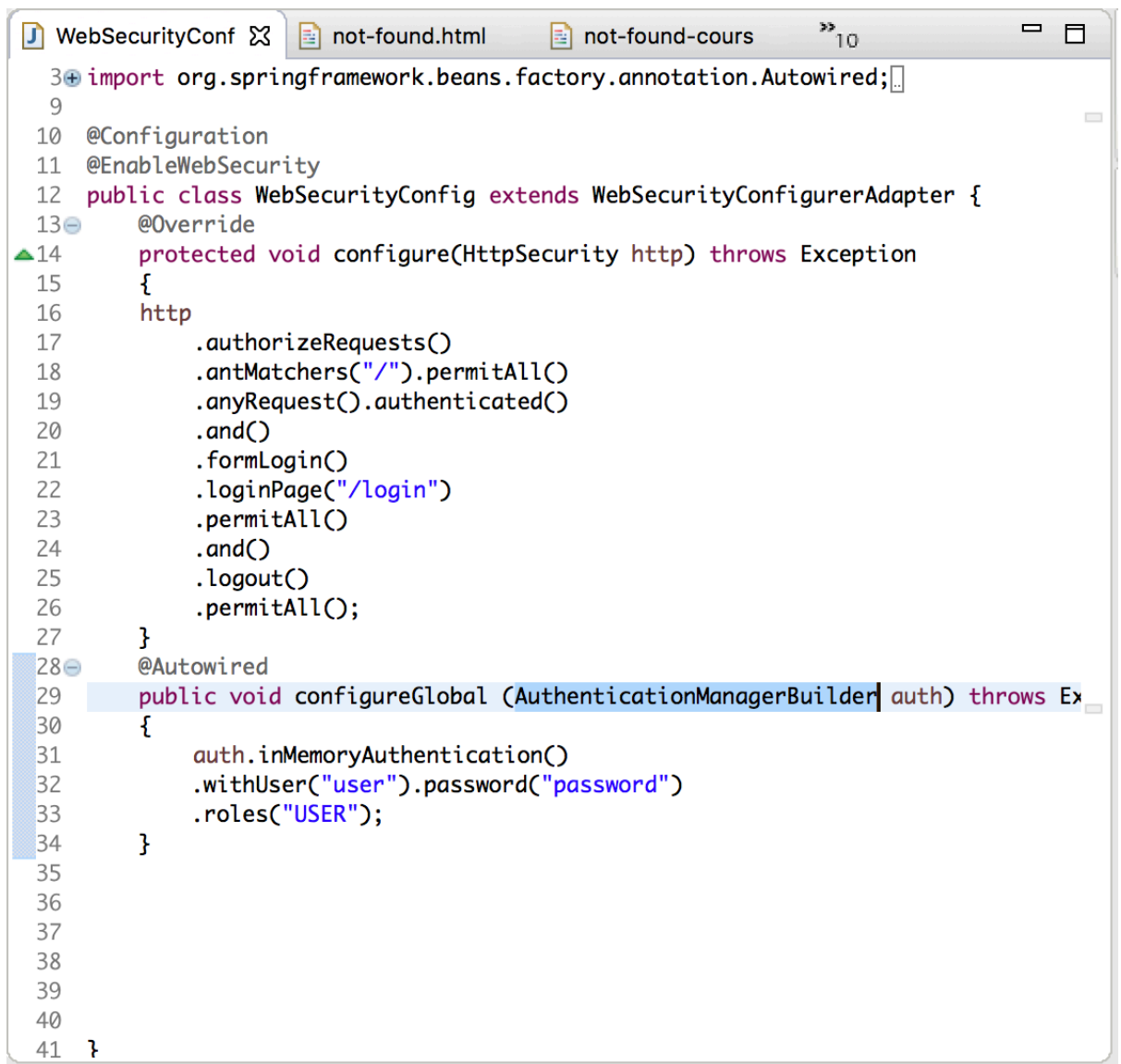
1. Buat sebuah project baru dan copy berkas-berkas yang ada di tutorial 5 untuk tutorial ini



2. Check dependency Security pada saat Anda membuat project baru atau tambahkan dependency Spring Boot Security berikut pada pom.xml

```
Tutorial8_1406623663/pom.xml
45         <scope>runtime</scope>
46     </dependency>
47 <dependency>
48     <groupId>mysql</groupId>
49     <artifactId>mysql-connector-java</artifactId>
50     <scope>runtime</scope>
51 </dependency>
52 <dependency>
53     <groupId>org.projectlombok</groupId>
54     <artifactId>lombok</artifactId>
55     <optional>true</optional>
56 </dependency>
57 <dependency>
58     <groupId>org.springframework.boot</groupId>
59     <artifactId>spring-boot-starter-test</artifactId>
60     <scope>test</scope>
61 </dependency>
62 <dependency>
63     <groupId>org.springframework.boot</groupId>
64     <artifactId>spring-boot-starter-security</artifactId>
65 </dependency>
66 </dependencies>
67
68 <build>
```

3. Buat class baru yaitu WebSecurityConfig.java pada package com.example (setara dengan main class) yang berisi:

A screenshot of an IDE window showing a Java file named 'WebSecurityConf'. The code is for a Spring Security configuration class. It includes an import for 'org.springframework.beans.factory.annotation.Autowired', annotations for '@Configuration' and '@EnableWebSecurity', and a class definition 'WebSecurityConfig' extending 'WebSecurityConfigurerAdapter'. The class has two methods: 'configure' which configures 'HttpSecurity' with various settings like 'authorizeRequests', 'antMatchers', 'anyRequest', 'formLogin', 'loginPage', 'logout', and 'permitAll'; and 'configureGlobal' which sets up in-memory authentication for a user named 'user' with password 'password' and role 'USER'. The code is syntax-highlighted and has line numbers on the left.

```
3+ import org.springframework.beans.factory.annotation.Autowired;
9
10 @Configuration
11 @EnableWebSecurity
12 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
13     @Override
14     protected void configure(HttpSecurity http) throws Exception
15     {
16         http
17             .authorizeRequests()
18             .antMatchers("/").permitAll()
19             .anyRequest().authenticated()
20             .and()
21             .formLogin()
22             .loginPage("/login")
23             .permitAll()
24             .and()
25             .logout()
26             .permitAll();
27     }
28     @Autowired
29     public void configureGlobal (AuthenticationManagerBuilder auth) throws Ex
30     {
31         auth.inMemoryAuthentication()
32             .withUser("user").password("password")
33             .roles("USER");
34     }
35
36
37
38
39
40
41 }
```

4. Buatlah halaman login.html

WebSecurityConf login.html index.html not-found.html not-found.html

```

1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml"
3     xmlns:th="http://www.thymeleaf.org"
4     xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
5 <head>
6 <title>Login Page</title>
7 </head>
8 <body>
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 </body>
25 </html>

```

5. Buatlah juga halaman index.html

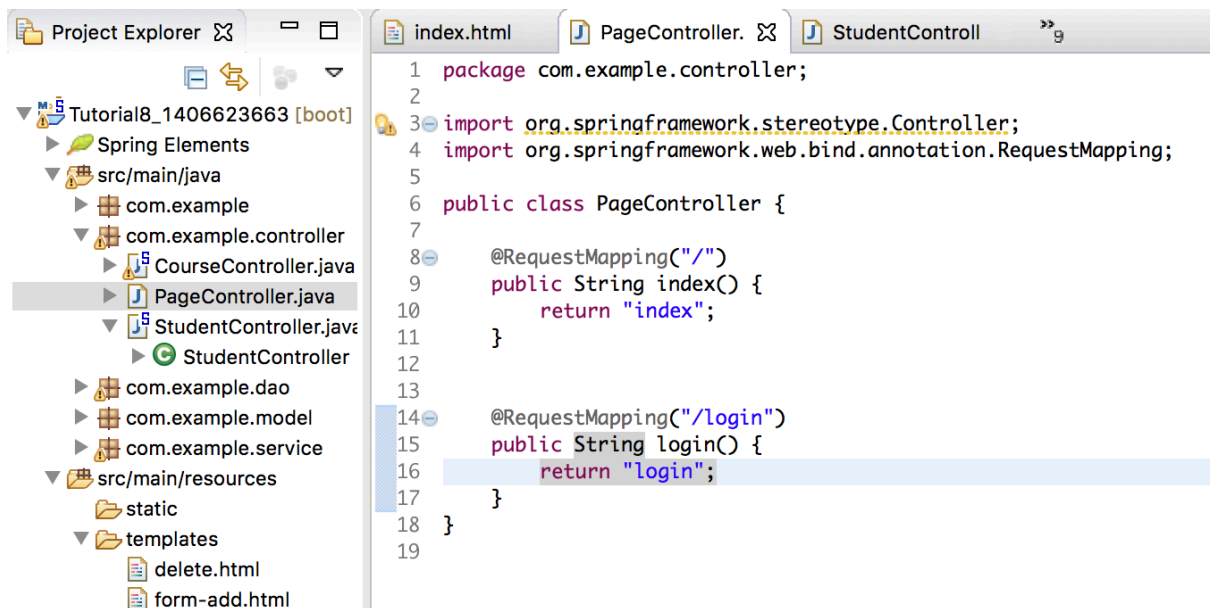
WebSecurityConf login.html index.html not-found.html

```

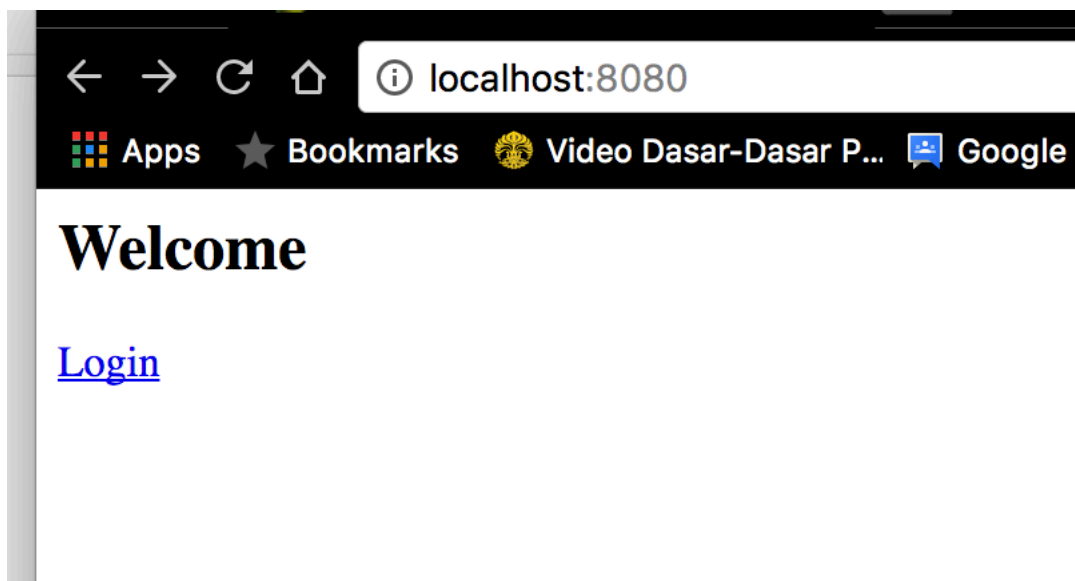
1 <html xmlns:th = "http://www.thymeleaf.org" >
2 <head>
3 <title>Index</title>
4 </head>
5 <body>
6     <h2>Welcome</h2>
7     <a href="/login"> Login </a>
8 </body>
9 </html>

```

6. Buatlah class PageController.java



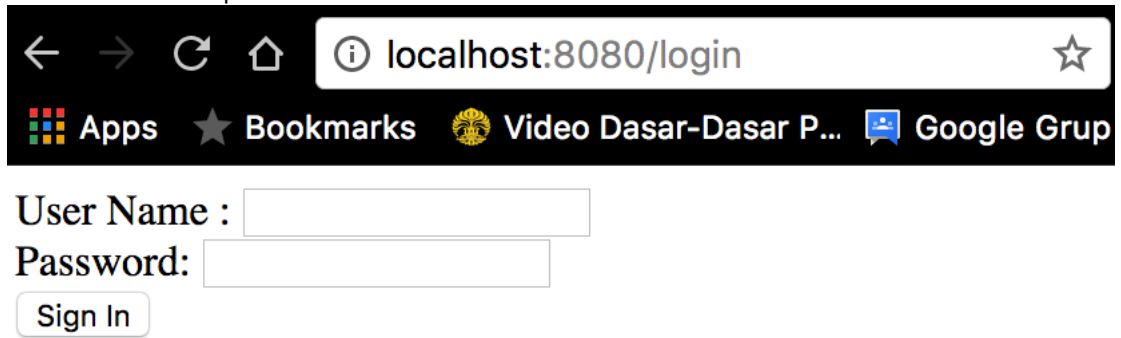
7. Jalankan program dan buka localhost:8080



8. Buka localhost:8080/student/viewall

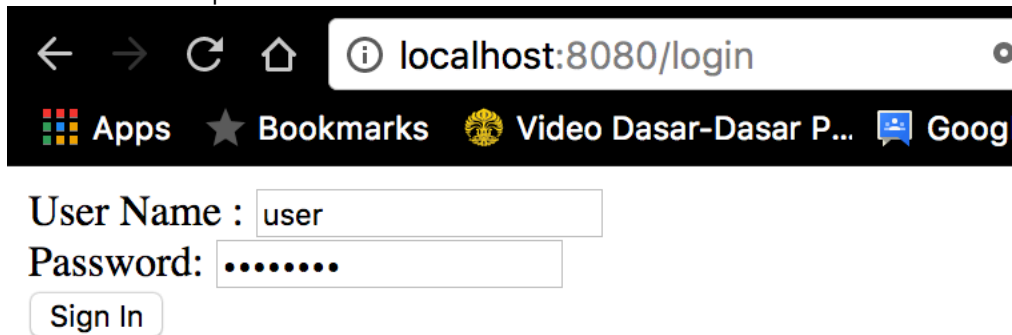
Akan mendirec ke <http://localhost:8080/login>

Kondisi sebelum input :



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/login'. Below the address bar, there are navigation icons and a list of bookmarks including 'Apps', 'Bookmarks', 'Video Dasar-Dasar P...', and 'Google Grup'. The login form consists of two input fields: 'User Name :' and 'Password:', both of which are empty. Below the password field is a 'Sign In' button.

Kondisi Setelah input:



The screenshot shows the same web browser window after input. The 'User Name :' field now contains the text 'user', and the 'Password:' field is filled with eight dots. The 'Sign In' button remains visible below the password field.

SIGN OUT dan NAMA USERNAME

1. Pada berkas viewall.html tambahkan baris berikut di bawah tag body

```

WebSecurityConf StudentControll viewall.html 12
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View All Students</title>
5 </head>
6 <body>
7 <h2 th:text=" 'Login as ' + ${#httpServletRequest.remoteUser}">
8   Login as</h2>
9
10 <form th:action="@{/logout}" method="post">
11   <input type="submit" value="Sign Out" />
12 </form>
13
14 <h1>All Students</h1>
15 <div th:each="student, iterationStatus: ${students}">
16   <h3 th:text="No. ' + ${iterationStatus.count}">No. 1</h3>
17   <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
18   <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
19   <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
20   <h3>Kuliah yang diambil</h3>
21   <ul th:each="course, iterationStatus: ${student.courses}">
22     <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">
23       Nama kuliah-X SKS</li>
24   </ul>
25   <a th:href="/student/delete/" + ${student.npm}"> Delete Data</a>
26   <a th:href="/student/update/" + ${student.npm}"> Update Data</a>
27   <hr />
28 </div>
29 </body>
30 </html>
31

```

2. Sehingga tampilannya adalah sebagai berikut:

LATIHAN

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


```

WebSecurityConf  viewall.html  viewall-course.  13
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View All Course</title>
5 </head>
6 <body>
7 <h2 th:text=" 'Login as ' + ${#httpServletRequest.remoteUser}">
8   Login as</h2>
9
10 <form th:action="@{/logout}" method="post">
11   <input type="submit" value="Sign Out" />
12 </form>
13 <h1>All Course</h1>
14
15 <div th:each="course, iterationStatus: ${courses}">
16   <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
17   <h3 th:text="'idCourse = ' + ${course.idCourse}">Student NPM</h3>
18   <h3 th:text="'Name = ' + ${course.name}">Student Name</h3>
19   <h3 th:text="'SKS = ' + ${course.credits}">Student GPA</h3>
20   <h3>Mahasiswa yang mengambil</h3>
21   <ul th:each="student, iterationStatus: ${course.students}">
22     <li th:text="${student.npm} + '-' + ${student.name}">Nama MHS
23   </ul>
24   <hr />
25 </div>
26 </body>
27 </html>
28

```

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

```

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

```

Hasil Output:

Membuat Menu dan Memanfaatkan Role

1. Pada method configure pada class WebSecurityConfig harus ditambahkan baris yang ditebalkan berikut:

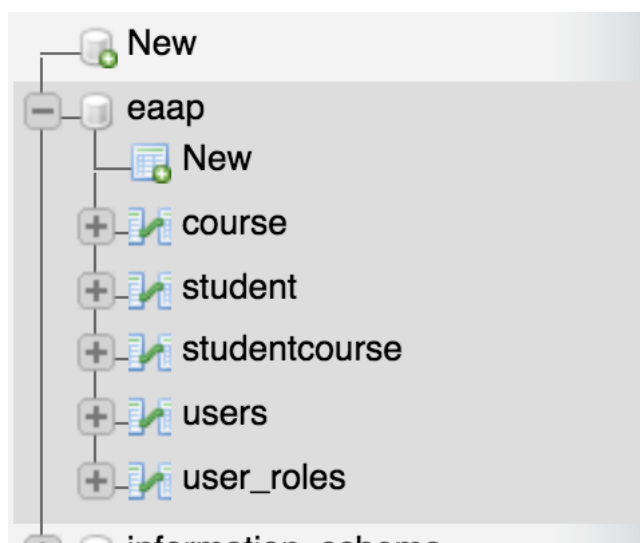
```
*WebSecurityCon  viewall.html  viewall-course.  13
11 @EnableWebSecurity
12 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
13     @Override
14     protected void configure(HttpSecurity http) throws Exception
15     {
16         http
17             .authorizeRequests()
18             .antMatchers("/").permitAll()
19             .antMatchers("/course/**").hasRole("ADMIN")
20             .antMatchers("/student/**").hasRole("USER")
21             .anyRequest().authenticated()
22             .and()
23             .formLogin()
24             .loginPage("/login")
25             .permitAll()
26             .and()
27             .logout()
28             .permitAll();
29     }
30     @Autowired
31     public void configureGlobal (AuthenticationManagerBuilder auth) throw
32     {
33         auth.inMemoryAuthentication ()
34             .withUser("user").password("password")
35             .roles("USER");
36
37         auth.inMemoryAuthentication ()
38             .withUser("admin").password("admin")
39             .roles("ADMIN");
40     }
41 }
```

2. Selanjutnya kita akan menambahkan menu. 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. Pada login.html tambahkan baris berikut di bawah link login:

```
WebSecurityConf  index.html  viewall-course. 13
1 <html xmlns:th="http://www.thymeleaf.org">
2 <head>
3 <title>Index</title>
4 </head>
5 <body>
6 <h2>Welcome</h2>
7 <a href="/login"> Login </a>
8 <br />
9 <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}"
10 href="/course/viewall"> View All Course </a>
11 <br />
12 <a
13 th:if="${#httpServletRequest.isUserInRole('USER') }"
14 href="/student/viewall"> View All Student </a>
15 <br />
16
17 </body>
18 </html>
19
```

Mengambil Username dan Password dari Database

1. Penambahan script untuk membuat table users dan user_roles



2. Pada class WebSecurityConfig, beri comment pada method configureGlobal, dan tambahkan baris dan method berikut:

```
13 // }
14 @Autowired
15 DataSource dataSource;
16
17 @Autowired
18 public void configAuthentication(AuthenticationManagerBuilder auth) throws Exception {
19     auth.jdbcAuthentication().dataSource(dataSource)
20         .usersByUsernameQuery("select username,password,enabled from users where username=?")
21         .authoritiesByUsernameQuery("select username, role from user_roles where username=?");
22 }
23 }
24 }
```

LATIHAN

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.

- Mengubah dan menambahkan syntax berikut pada WebSecurityConfig.java

```
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests().antMatchers("/").permitAll().antMatchers("/course/**").hasRole("ADMIN")
            .antMatchers("/student/**").hasAnyRole("USER", "ADMIN").anyRequest().authenticated().and().formLogin()
            .loginPage("/login").permitAll().and().logout().permitAll();
    }
}
```

- Mengubah syntax tampilan pada halaman index.html

```
WebSecurityConf index.html viewall-course. 13
1 <html xmlns:th="http://www.thymeleaf.org">
2 <head>
3 <title>Index</title>
4 </head>
5 <body>
6 <h2>Welcome</h2>
7 <a href="/login"> Login </a>
8 <br />
9 <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}"
10 href="/course/viewall"> View All Course </a>
11 <br />
12 <a
13 th:if="${#httpServletRequest.isUserInRole('USER') }"
14 href="/student/viewall"> View All Student </a>
15 <br />
16
17 </body>
18 </html>
19
```