

Yang Saya Pelajari

Saya mempelajari bagaimana cara mengimplementasikan security pada Java Spring, dimana pada tutorial ini saya membuat halaman login untuk masuk ke halaman course/viewall dan student/viewall dan hanya dapat masuk jika user dan password yang diinput sama seperti user dan password yang sudah terdaftar di dalam WebSecurityConfig.

Latihan 1

1. Karena pada tutorial 5 tidak ada halaman /course/viewall, maka dibuatkan terlebih dahulu.
2. Pada StudentMapper.java, dibuat method di bawah ini

```
@Select ("select id_course, name, credits from course")
@Results (value = {
    @Result (property = "idCourse" , column = "id_course" ),
    @Result (property = "name" , column = "name" ),
    @Result (property = "credits" , column = "credits" )
})
List<CourseModel> selectAllCourses();
```

3. Pada StudentServiceDatabase.java, dibuat method seperti di bawah ini

```
public List<CourseModel> selectAllCourses() {
    return studentMapper.selectAllCourses();
}
```

4. Pada StudentController.java, dibuat method dengan Request Mapping /course/viewall seperti di bawah ini

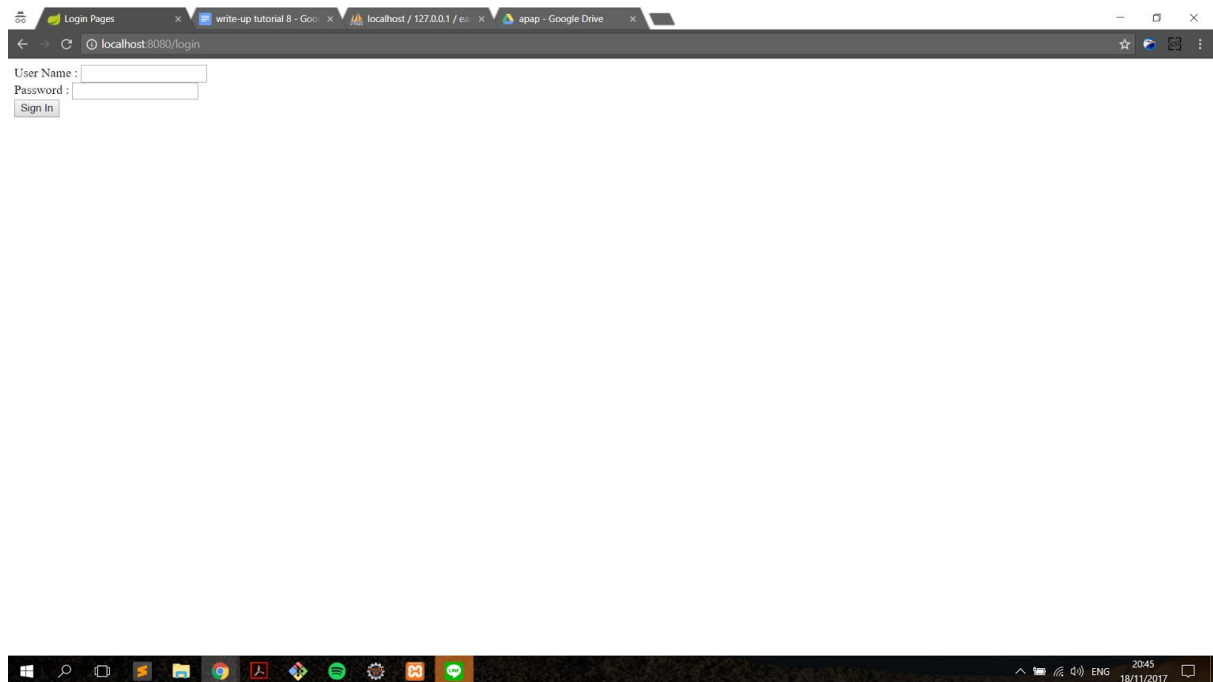
```
@RequestMapping("/course/viewall")
public String viewAllCourse (Model model)
{
    List<CourseModel> courses = studentDAO.selectAllCourses();
    model.addAttribute ("course", courses);

    return "course-viewall";
}
```

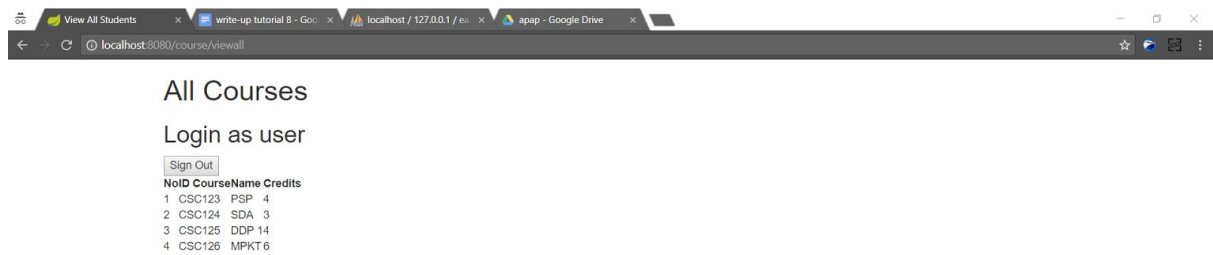
5. Kemudian membuat file course-viewall.html seperti di bawah ini

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<link rel="stylesheet" type="text/css" href="//cdn.datatables.net/1.10.16/css/jquery.dataTables.css" />
<script type="text/javascript" charset="utf8" src="//cdn.datatables.net/1.10.16/js/jquery.dataTables.js"></script>
</head>
<body class="container">
  <h1>All Courses</h1>
  <h2 th:text=" 'Login as ' + ${httpServletRequest.remoteUser}">Login as</h2>
  <form th:action="@{/Logout}" method="post">
    <input type="submit" value="Sign Out"></input>
  </form>
  <div>
    <table id="table_id" class="display">
      <thead>
        <tr>
          <th>No</th>
          <th>ID Course</th>
          <th>Name</th>
          <th>Credits</th>
        </tr>
      </thead>
      <tbody>
        <tr th:each="course, iterationStatus: ${course}">
          <td th:text="${iterationStatus.index + 1}">No</td>
          <td th:text="${course.idCourse}"></td>
          <td th:text="${course.name}"></td>
          <td th:text="${course.credits}"></td>
        </tr>
      </tbody>
    </table>
  </div>
</body>
```

6. Menjalankan program di /course/viewall. Diminta untuk input password



7. Masuk ke halaman /course/viewall



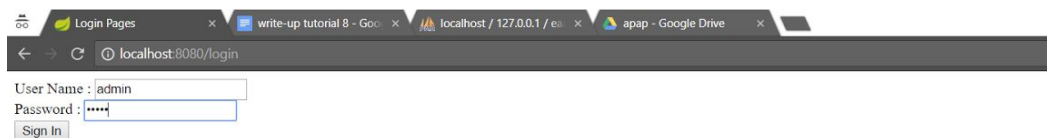
Latihan 2

1. Untuk membuat user baru, saya cukup menambahkan user dan password yang diinginkan di dalam method configureGlobal()

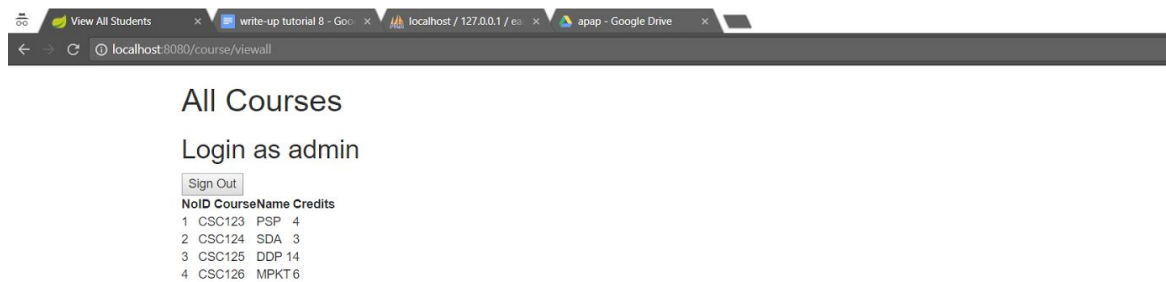
```
@Autowired
public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception
{
    auth.inMemoryAuthentication()
        .withUser("user").password("password")
        .roles("USER");

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

2. Mencoba masuk ke halaman /course/viewall. Memasukan user "admin" dan password "admin"



3. Berhasil masuk sebagai admin



Latihan 3

1. Pada WebSecurityConfig.java, saya membuat method configure() seperti di bawah ini.

```
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {  
    @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()  
            .formLogin()  
            .loginPage("/login")  
            .permitAll()  
            .and()  
            .logout()  
            .permitAll();  
    }  
}
```

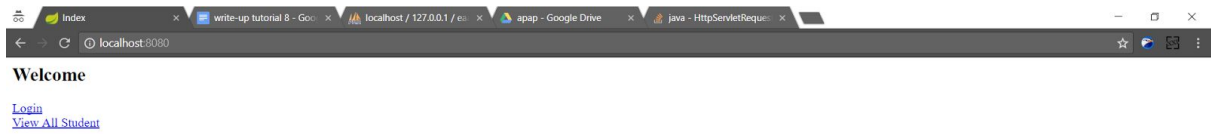
Untuk /course/viewall hanya dapat diakses oleh admin, sedangkan untuk student/viewall hanya dapat diakses oleh user dan admin

2. Di file index.html dibuat code seperti di bawah ini

```
1<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org"  
2    xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">  
3<head>  
4    <title>Index</title>  
5</head>  
6<body>  
7    <h2>Welcome</h2>  
8    <a href="/login">Login</a>  
9    <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/course/viewall"><br /> View All Course </a>  
10    <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall"><br /> View All Student</a>  
11    <a th:if="${#httpServletRequest.isUserInRole('USER')}" href="/student/viewall"><br /> View All Student</a>  
12</body>  
13</html>
```

3. Dijalankan di browser menuju halaman <http://localhost:8080/> dan berikut tampilannya jika masuk sebagai role user

MUHAMMAD AKBAR SETIADI
APAP B
1506737810



4. Berikut tampilannya jika masuk sebagai role admin

