

- **Hal yang dipelajari dalam tutorial**

Pada tutorial 8 kali ini mempelajari mengenai Web Service. Web Service berguna untuk melakukan otentikasi dan otorisasi terhadap pengguna (biasanya admin dan user) yang akan mengakses sebuah sistem. Untuk membuat Web Service dalam Spring memerlukan *dependency* Security yang ditambahkan pada pom.xml, setelah itu bisa membuat sebuah *class config*. Untuk melakukan otentikasi dan otorisasi, caranya dapat mengambil *username* dan *password* yang ada pada *database* atau bisa pada *in-memory*.

- **Latihan**

- ❖ **Latihan 1**

Pada latihan 1 ini membuat tombol Sign Out dan mencetak nama 'user' untuk halaman student/view/{npm}. Untuk menambahkan tombol Sign Out pada halaman view.html caranya dengan membuat *code* seperti pada gambar dibawah ini.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>View Student by NPM</title>
  </head>
  <body>
    <div th:if="${#httpServletRequest.isUserInRole('ADMIN')}">
      <div th:replace="fragments/fragment :: header"></div>
      <div style="padding-left: 2.7ch;">
        <h2 th:text="'Login as ' + ${#httpServletRequest.remoteUser}">Login as</h2><br/>
        <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
        <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
        <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
        <form th:action="@{/logout}" method="post">
          <input type="submit" value="Sign Out" class="btn btn-success"/>
        </form>
      </div>
    </div>
    <div th:unless="${#httpServletRequest.isUserInRole('ADMIN')}">
      <div th:replace="fragments/fragment-user :: header"></div>
      <div style="padding-left: 2.7ch;">
        <h2 th:text="'Login as ' + ${#httpServletRequest.remoteUser}">Login as</h2><br/>
        <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
        <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
        <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
        <form th:action="@{/logout}" method="post">
          <input type="submit" value="Sign Out" class="btn btn-success"/>
        </form>
      </div>
    </div>
  </body>
</html>
```

Sedangkan untuk menampilkan nama role yang mengakses halaman tersebut dengan cara menambahkan *code* pada gambar dibawah ini.

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>View Student by NPM</title>
  </head>
  <body>
    <div th:if="${#httpServletRequest.isUserInRole('ADMIN')}">
      <div th:replace="fragments/fragment :: header"></div>
      <div style="padding-left: 2.7ch;">
        <h2 th:text="'Login as ' + ${#httpServletRequest.remoteUser}">Login as</h2><br/>
        <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
        <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
        <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
        <form th:action="@{/logout}" method="post">
          <input type="submit" value="Sign Out" class="btn btn-success"/>
        </form>
      </div>
    </div>
    <div th:unless="${#httpServletRequest.isUserInRole('ADMIN')}">
      <div th:replace="fragments/fragment-user :: header"></div>
      <div style="padding-left: 2.7ch;">
        <h2 th:text="'Login as ' + ${#httpServletRequest.remoteUser}">Login as</h2><br/>
        <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
        <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
        <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
        <form th:action="@{/logout}" method="post">
          <input type="submit" value="Sign Out" class="btn btn-success"/>
        </form>
      </div>
    </div>
  </body>
</html>

```

❖ Latihan 2

Pada latihan ini membuat sebuah *user* baru dengan peran “USER”. Caranya dengan menambahkan *code* berikut pada config.

```

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

```

❖ Latihan 3

Pada latihan ini membuat menu view student by NPM beserta *field* input dan *button* pada halaman index.html dengan *role* ‘USER’. Caranya dengan membuat *code* berikut pada halaman index.html.

```

<html>
  <head>
    <title>Index</title>
  </head>
  <body>
    <div th:if="${#httpServletRequest.isUserInRole('ADMIN')}">
      <div th:replace="fragments/fragment :: header"></div>
      <h2 align="center" th:text="'Welcome ' + ${#httpServletRequest.remoteUser}"></h2>
      <div style="padding-left: 2.7ch;">
        <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall/">View All Student</a>
      <p></p>
      <label th:if="${#httpServletRequest.isUserInRole('ADMIN')}"><b>View Student by NPM</b></label>
      <form th:if="${#httpServletRequest.isUserInRole('ADMIN')}" action="/student/view" method="get">
        <input type="text" name="npm" placeholder="Masukan NPM"></input>
        <p></p>
        <button type="submit" class="btn btn-success">Save</button><br/>
      </form>
    </div>
    <div th:unless="${#httpServletRequest.isUserInRole('ADMIN')}">
      <div th:replace="fragments/fragment-user :: header"></div>
      <h2 align="center" th:text="'Welcome ' + ${#httpServletRequest.remoteUser}"></h2>
      <div style="padding-left: 2.7ch;">
        <label th:if="${#httpServletRequest.isUserInRole('USER')}"><b>View Student by NPM</b></label>
        <form th:if="${#httpServletRequest.isUserInRole('USER')}" action="/student/view" method="get">
          <input type="text" name="npm" placeholder="Masukan NPM"></input>
          <p></p>
          <button type="submit" class="btn btn-success">Save</button><br/>
        </form>
      </div>
    </div>
    <div th:replace="fragments/fragment :: footer"></div>
  </body>
</html>

```

Setelah itu, perlu menambahkan *code* pada WebSecurityConfig seperti pada gambar dibawah ini

```

package com.example;

import javax.sql.DataSource;

@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure (HttpSecurity http) throws Exception {
        http
            .authorizeRequests ()
            .antMatchers ("/").permitAll ()
            .antMatchers ("/student/viewall").hasRole("ADMIN")
            .antMatchers ("/student/view/**").hasRole("USER")
            .antMatchers ("/student/view").hasRole("USER")
        //
        .antMatchers("/student/view").access("hasRole('USER') or hasRole('ADMIN')")
        .antMatchers("/login").permitAll()
        .antMatchers("/css/**").permitAll()
        .antMatchers("/js/**").permitAll()
        .anyRequest ().authenticated ()
        .and ()
        .formLogin ()
        .loginPage ( "/login")
        .permitAll ()
        .and ()
        .logout ()
        .permitAll ();
    }
}

```

❖ Latihan 4

Latihan terakhir pada tutorial ini membuat pengguna dengan *role* 'ADMIN' bisa melakukan view all student dan view student by NPM. Caranya, pertama dengan menambahkan *code* pada halaman index.html seperti pada gambar dibawah ini.

```
<html>
  <head>
    <title>Index</title>
  </head>
  <body>
    <div th:if="${#httpServletRequest.isUserInRole('ADMIN')}">
      <div th:replace="fragments/fragment :: header"></div>
      <h2 align="center" th:text="'Welcome ' + ${#httpServletRequest.remoteUser}"></h2>
      <div style="padding-left: 2.7ch;">
        <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall/">View All Student</a>
        <p></p>
        <label th:if="${#httpServletRequest.isUserInRole('ADMIN')}"><b>View Student by NPM</b></label>
        <form th:if="${#httpServletRequest.isUserInRole('ADMIN')}" action="/student/view" method="get">
          <input type="text" name="npm" placeholder="Masukan NPM"></input>
          <p></p>
          <button type="submit" class="btn btn-success">Save</button><br/>
        </form>
      </div>
    </div>
    <div th:unless="${#httpServletRequest.isUserInRole('ADMIN')}">
      <div th:replace="fragments/fragment-user :: header"></div>
      <h2 align="center" th:text="'Welcome ' + ${#httpServletRequest.remoteUser}"></h2>
      <div style="padding-left: 2.7ch;">
        <label th:if="${#httpServletRequest.isUserInRole('USER')}"><b>View Student by NPM</b></label>
        <form th:if="${#httpServletRequest.isUserInRole('USER')}" action="/student/view" method="get">
          <input type="text" name="npm" placeholder="Masukan NPM"></input>
          <p></p>
          <button type="submit" class="btn btn-success">Save</button><br/>
        </form>
      </div>
    </div>
    <div th:replace="fragments/fragment :: footer"></div>
  </body>
</html>
```

Setelah itu, pada *class config* ada sedikit penambahan *code* supaya pengguna dengan *role* 'ADMIN' bisa melakukan view all student dan view student by NPM. Untuk lebih jelasnya dapat dilihat pada gambar dibawah.

```
package com.example;

import javax.sql.DataSource;

@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure (HttpSecurity http) throws Exception {
        http
            .authorizeRequests ()
            .antMatchers ("/").permitAll ()
            .antMatchers ("/student/viewall").hasRole("ADMIN")
            .antMatchers ("/student/view/**").access("hasRole('USER') or hasRole('ADMIN')")
            .antMatchers("/student/view").access("hasRole('USER') or hasRole('ADMIN')")
            .antMatchers("/login").permitAll()
            .antMatchers("/css/**").permitAll()
            .antMatchers("/js/**").permitAll()
            .anyRequest ().authenticated ()
            .and ()
            .formLogin ()
            .loginPage ( "/login")
            .permitAll ()
            .and ()
            .logout ()
            .permitAll ();
    }
}
```