

Ringkasan

Pada tutorial kali ini, dipelajari bagaimana cara membuat sebuah web security sederhana dengan memberikan otentikasi dan otorisasi pada sebuah sistem. Di mana, pemberian otentikasi dan otorisasi tersebut dilakukan dengan menggunakan dependency Spring Boot Starter Security. Kemudian, proses otentikasi yang dipelajari adalah proses otentikasi yang menggunakan memory system serta menggunakan database sebagai data acuan untuk otentikasinya. Selanjutnya, proses pemberian otorisasi dengan memetakan fitur-fitur mana saja yang dapat digunakan oleh sebuah role tertentu dengan menggunakan url sebagai acuan bahwa sebuah fitur dapat diakses oleh role tertentu.

Latihan 1 – Membuat tombol Sign Out pada Halaman student/view/{npm}

Pembuatan tombol **Sign Out** pada halaman **student/view/{npm}** hampir sama seperti pembuatan tombol tersebut pada halaman **student/viewall**, cukup dengan menambahkan baris kode seperti pada Gambar 1. Adapun hasilnya dapat dilihat pada Gambar 2.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head th:replace="fragments/fragment :: assets" />
  <body>
    <div th:replace="fragments/fragment :: header"></div>
    <h2 th:text="'Login as ' + ${#httpServletRequest.remoteUser}">Login as</h2>
    <form th:action="@{/Logout}" method="post">
      <input type="submit" value="Sign Out" />
    </form>
    <hr />
    <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>
    <div th:replace="fragments/fragment :: footer"></div>
  </body>
</html>
```

Gambar 1. Penambahan Kode untuk Membuat Tombol Sign Out pada Halaman student/view/{npm}

Student Web Home Daftar Mahasiswa Menambah Mahasiswa

Search Search

Login as admin

Sign Out

NPM = 1606954735

Name = Arri Kurniawan

GPA = 3.9

Mata Kuliah APAP

Gambar 2. Hasil dari Latihan Ke-1

Latihan 2 – Menambahkan user baru dengan username “user”, password “user”, dan role “USER”.

Untuk menambahkan user baru pada *memory system* dapat dilihat potongan kode pada Gambar 2.

```
.authorizeRequests()
.antMatchers("/").permitAll()
.antMatchers("/student/viewall").hasRole("ADMIN")
.antMatchers("/student/view/**").hasAnyRole("ADMIN", "USER")
.anyRequest().authenticated()
.and()
.formLogin()
.loginPage("/login")
.permitAll()
.and()
.logout()
.permitAll();
}

// Don't use database
@Autowired
public void configureGlobal (AuthenticationManagerBuilder auth) throws Exception
{
    auth.inMemoryAuthentication()
        .withUser("admin").password("admin")
        .roles("ADMIN");
    auth.inMemoryAuthentication()
        .withUser("user").password("user")
        .roles("USER");
}

// Use database
// @Autowired
// DataSource dataSource;
//
// @Autowired
// public void configAuthentication (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=?");
// }
}
```

Gambar 3. Menambah User Baru dalam Memory System

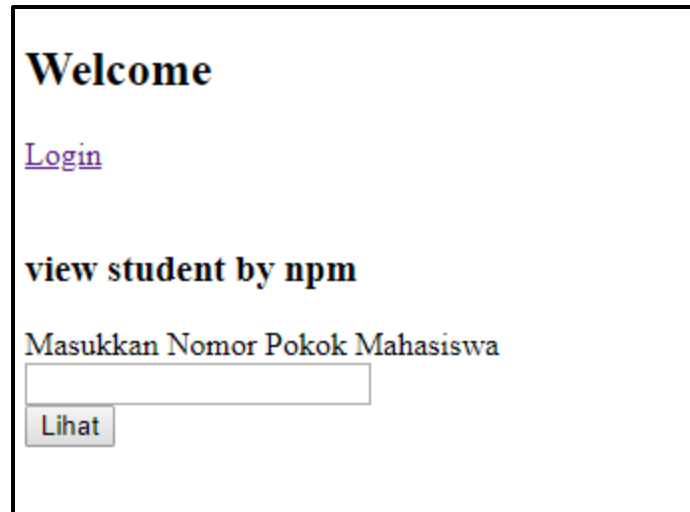
Selanjutnya, hasilnya dicoba dengan memasukkan username dan password yang telah ditambahkan tersebut seperti Gambar 4. Kemudian, hasilnya dilihat pada Gambar 5 dan 6 yang mana user yang telah ditambahkan tersebut telah bisa melakukan otentikasi dan otorisasi terhadap sistem.

You have been logged out.

User Name:

Password:

Gambar 4. Halaman Login



Welcome

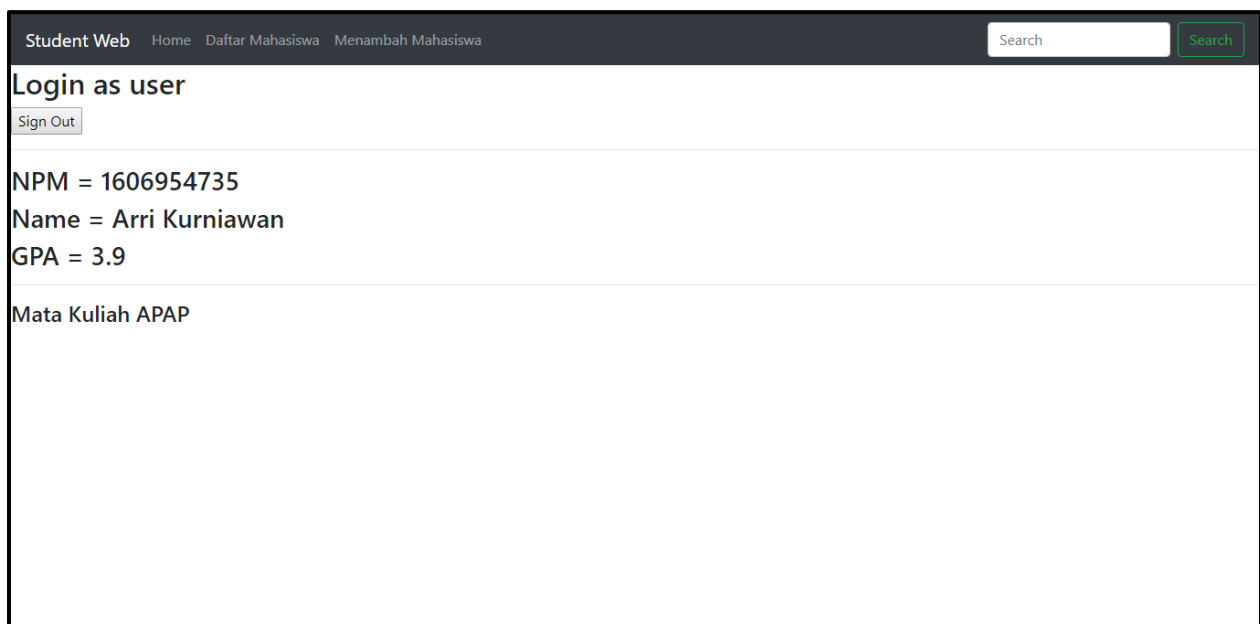
[Login](#)

view student by npm

Masukkan Nomor Pokok Mahasiswa

Lihat

Gambar 5. Hasil Penambahan User Ke-1



Student Web Home Daftar Mahasiswa Menambah Mahasiswa Search Search

Login as user

Sign Out

NPM = 1606954735
Name = Arri Kurniawan
GPA = 3.9

Mata Kuliah APAP

Gambar 6. Hasil Penambahan User Ke-2

Latihan 3 – Menampilkan menu view student by npm beserta input field dan button untuk memasukkan nomor npm student yang ingin dicari pada halaman index user dengan role USER.

Untuk menambahkan hal tersebut, diperlukan potongan kode seperti pada Gambar 7. Kemudian, hasilnya dapat dilihat pada Gambar 5. Selanjutnya, menambahkan fitur yang dapat diaksesnya dengan menambahkan potongan kode pada WebSecurityConfig.java seperti pada Gambar 8.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head><title>Index</title></head>
  <body>
    <h2>Welcome</h2>
    <a href="/Login">Login</a><br />
    <th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</th>
    <div th:if="${#httpServletRequest.isUserInRole('ADMIN')} || #httpServletRequest.isUserInRole('USER')}">
      <form th:action="@{/student/view/}" method="post">
        <h3>view student by npm</h3>
        <div class="form-row">
          <div class="form-group col-md-4">
            <label for="npm">Masukkan Nomor Pokok Mahasiswa</label>
          </div>
          <div class="form-group col-md-4">
            <input type="text" name="npm" class="form-control"/>
          </div>
          <div class="col-md-4">
            <button type="submit" class="btn btn-primary">Lihat</button>
          </div>
        </div>
      </form>
    </div>
  </body>
</html>
```

Gambar 7. Penambahan Kode pada index.html untuk Latiha Ke-3

```
package com.example;

import org.apache.tomcat.jdbc.pool.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/**").hasAnyRole("ADMIN", "USER")
            .anyRequest().authenticated()
            .and()
            .formLogin()
            .loginPage("/login")
            .permitAll()
            .and()
            .logout()
            .permitAll();
    }

    // Don't use database
    // @Autowired
    // public void configureGlobal (AuthenticationManagerBuilder auth) throws Exception
    // {
    //     auth.inMemoryAuthentication()
    //         .withUser("admin").password("admin")
    //         .roles("ADMIN");
    //     auth.inMemoryAuthentication()
    //         .withUser("user").password("user")
    //         .roles("USER");
    // }

    // Use database
}
```

Gambar 8. Penambahan Kode pada WebConfigSecurity.java

Latihan 4 Buat agar user dengan role ADMIN dapat melihat view all student dan view student by npm juga.

Untuk mengerjakan Latihan 4 ini, perlu ditambahkan potongan Kode seperti pada Gambar 9 dan hasilnya dapat dilihat pada Gambar 10.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head><title>Index</title></head>
  <body>
    <h2>Welcome</h2>
    <a href="/login">Login</a><br />
    <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br>
    <div th:if="${#httpServletRequest.isUserInRole('ADMIN')} || #httpServletRequest.isUserInRole('USER')}">
      <form th:action="@{/student/view/}" method="post">
        <h3>view student by npm</h3>
        <div class="form-row">
          <div class="form-group col-md-4">
            <label for="npm">Masukkan Nomor Pokok Mahasiswa</label>
          </div>
          <div class="form-group col-md-4">
            <input type="text" name="npm" class="form-control"/>
          </div>
          <div class="col-md-4">
            <button type="submit" class="btn btn-primary">Lihat</button>
          </div>
        </div>
      </form>
    </div>
  </body>
</html>
```

Gambar 9. Potongan Kode untuk Latihan Ke-4

Welcome

[Login](#)
[View All Student](#)

view student by npm

Masukkan Nomor Pokok Mahasiswa

Gambar 10. Hasil Latihan Ke-4