

## A. RINGKASAN MATERI

Pada tutorial kali ini mempelajari mengembangkan sistem yang akan digunakan oleh tingkatan pengguna yang berbeda yaitu user dan admin yang memiliki aturan otentikasi dan otorisasi menggunakan dependency Spring Boot Starter Security untuk mengimplementasikan otentikasi dan otorisasi pada sistem. Otentikasi diperlukan untuk melakukan verifikasi bahwa pengguna mempunyai akses untuk masuk ke dalam sistem. Otorisasi merupakan pengaturan hak pengguna yang telah memiliki otentikasi. Pengguna dengan tingkat Admin memiliki hak otorisasi yang berbeda dengan user. Otorisasi menentukan fitur apa saja yang dapat diakses oleh seorang pengguna yang telah diotentikasi.

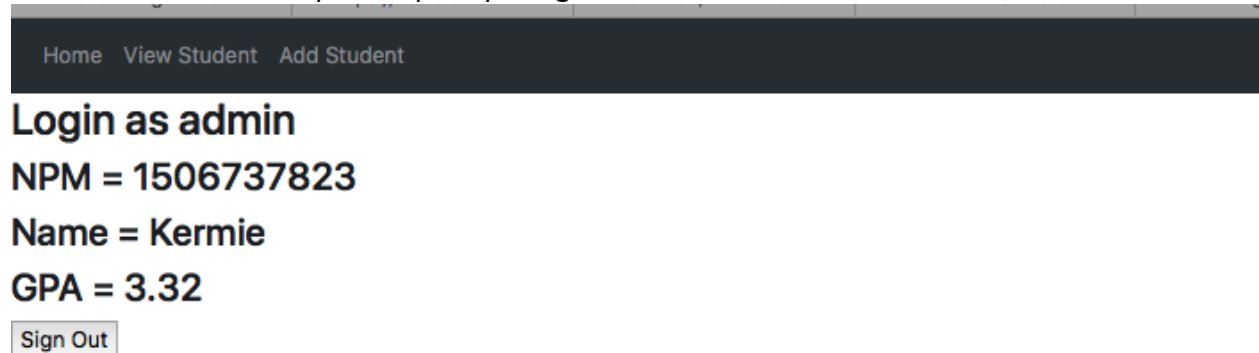
## B. LATIHAN

### 1. Buatlah hal yang sama dengan menambahkan tombol Sign Out dan cetak nama user untuk halaman `student/view/{npm}`.

Pada `view.html` akan ditambahkan kode berikut

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>View Student by NPM</title>
5   </head>
6   <body>
7     <div th:replace = "fragments/fragment :: header" ></div>
8     <h2 th:text= "'Login as ' + ${#httpServletRequest.remoteUser}'>Login as </h2>
9
10    <h3 th:text= "'NPM = ' + ${student.npm}>Student NPM</h3>
11    <h3 th:text= "'Name = ' + ${student.name}>Student Name</h3>
12    <h3 th:text= "'GPA = ' + ${student.gpa}>Student GPA</h3>
13
14    <form th:action="@{/logout}" method="post">
15      <input type="submit" value="Sign Out"/>
16    </form>
17    <div th:replace="fragments/fragment :: footer"></div>
18  </body>
19 </html>
20
```

Maka akan muncul tampilan seperti pada gambar berikut



### 2. Buatlah user baru dengan username "user", password "user", dan role "USER". Coba apakah bisa login dengan akun tersebut.

Shinta Kusuma Wardhani

1606955044

Tutorial 8

Pada WebSecurityConfig tambahkan kode berikut untuk memberi user baru, password baru dan role baru.

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

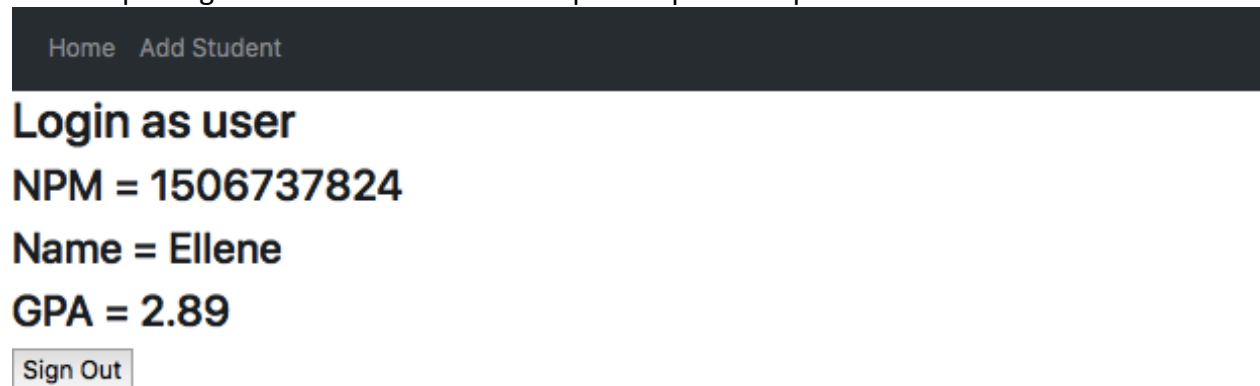
Maka akan muncul tampilan seperti pada gambar berikut.



The screenshot shows a web application interface with a dark header bar containing links: Home, View Student, and Add Student. Below the header, there is a login form with the following elements:

- A label "User Name :" followed by a text input field containing the value "user".
- A label "Password:" followed by a password input field with masked characters "...." and a toggle icon.
- A "Sign In" button.

Jika kita pilih Sign in maka akan muncul tampilan seperti tampilan berikut



The screenshot shows the user profile page after a successful login. It features a dark header bar with links: Home and Add Student. The main content area displays the following information:

- Login as user**
- NPM = 1506737824**
- Name = Ellene**
- GPA = 2.89**
- A "Sign Out" button.

### 3. Tampilkan menu view student by npm beserta input field dan button untuk memasukkannomor npm student yang ingin dicari pada halaman index user dengan role USER.

Pada index.html masukkan kode berikut

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <title>Index</title> </head>
5   <body>
6     <div th:replace="fragments/fragment :: header"></div>
7     <a href="/login">Login</a>
8
9     <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
10    <div th:if="${#httpServletRequest.isUserInRole('ADMIN')} || #httpServletRequest.isUserInRole('USER')}>
11      <form th:action="@{/student/view/}" method="post">
12        <h2 align="center">View Student by NPM</h2>
13        <div class="form-row">
14          <div class="form-group">
15            <label for="npm">NPM</label>
16          </div>
17          <div class="form-group">
18            <input type="text" name="npm" class="form-control"/>
19          </div>
20          <div class="form-group">
21            <button type="submit" class="btn btn-primary">Lihat</button>
22          </div>
23        </div>
24      </form>
25    </div>
26
27    <div th:replace="fragments/fragment :: footer"></div>
28  </body>
29 </html>
```

dan masukkan role pada WebSecurityConfig seperti pada gambar berikut

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

Maka akan muncul tampilan seperti pada gambar berikut.

[Home](#) [Add Student](#)

[Login](#) View Student by NPM

NPM

1506737824

[Home](#) [Add Student](#)

**Login as user**  
**NPM = 1506737824**  
**Name = Ellene**  
**GPA = 2.89**

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

Pada index.html tambahkan kode berikut

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>Index</title> </head>
5 <body>
6 <div th:replace="fragments/fragment :: header"></div>
7 <a href="/login">Login</a>
8
9 <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
10 <div th:if="${#httpServletRequest.isUserInRole('ADMIN')} || #httpServletRequest.isUserInRole('USER')}">
11 <form th:action="@{/student/view/}" method="post">
12 <h2 align="center">View Student by NPM</h2>
13 <div class="form-row">
14 <div class="form-group">
15 <label for="npm">NPM</label>
16 </div>
17 <div class="form-group">
18 <input type="text" name="npm" class="form-control"/>
19 </div>
20 <div class="form-group">
21 <button type="submit" class="btn btn-primary">Lihat</button>
22 </div>
23 </div>
24 </form>
25 </div>
26
27 <div th:replace="fragments/fragment :: footer"></div>
28 </body>
29 </html>
```

Maka akan menghasilkan tampilan seperti pada gambar dibawah

[Home](#) [View Student](#) [Add Student](#)

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

NPM