

Nama : Julio Muhammad Pranadano
NPM : 1706106785

Ringkasan Materi

Pada tutorial ini mempelajari mengenai web security, dimana dalam pengembangan sebuah sistem pengguna atau *user* memerlukan sebuah autentikasi dan otorisasi. Autentikasi diperlukan untuk melakukan verifikasi apakah pengguna tersebut memiliki hak akses untuk mengakses sistem. Sedangkan otorisasi merupakan hak yang dimiliki pengguna tergantung dari *role* pengguna tersebut. Dalam hal ini admin dan *user* memiliki hak akses yang berbeda.

Tutorial

Pada tutorial ini membuat sebuah web security dengan menggunakan kembali project 5 yang telah dibuat. Pada saat membuat otentikasi diperlukan tambahan dependency pada pom.xml seperti berikut :

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Setelah itu membuat WebSecurityConfig.java pada package com.example dengan mengikuti langkah pada modul. Pada WebSecurityConfig, terdapat method yang berisi konfigurasi dari setiap request yang dicantumkan. Seperti `"/login"` untuk melakukan login dan `"/"` untuk mengakses index.html yang dapat diakses oleh siapapun karena tidak ada autentikasi di dalamnya

Latihan sign out dan username baru.

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

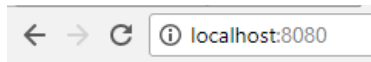


2. Buatlah user baru dengan username `"user"`, password `"user"`, dan role `"USER"`. Coba apakah bisa login dengan akun tersebut.

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

Nama : Julio Muhammad Pranadano
NPM : 1706106785

Tampilan Login sebagai user :



Welcome

[Login](#)

View by NPM

Menambahkan user baru dengan username “user” dan password “user” yang memiliki role sebagai “USER” yang didefinisikan di `configureGlobal()`.

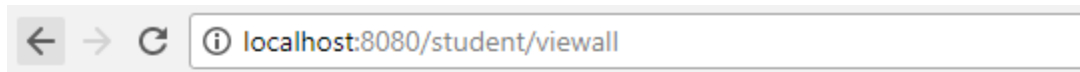
Membuat Menu dan Memanfaatkan Role

Tahap ini kita melakukan otorisasi dari setiap pengguna berdasarkan rolenya untuk dapat mengakses halaman tertentu. Kita dapat menambahkan kode pada method `configure` di `WebSecurityConfig.java` seperti berikut :

```
http
    .authorizeRequests()
    .antMatchers("/", "/resources/**").permitAll()
    .antMatchers("/student/viewall").hasRole("ADMIN")
    .antMatchers("/student/view/**").hasRole("USER")
    .anyRequest().authenticated()
    ...
```

Pada code diatas, role admin dapat menjalankan dengan url `/student/viewall` (dapat melihat seluruh data siswa) sedangkan role user dapat melihat berdasarkan npm user tersebut.

Jika login menggunakan user dengan role = USER ,maka saat akan membuka `student/viewall` muncul pesan error seperti dibawah



Whitelabel Error Page

This application has no explicit mapping for `/error`, so you are seeing this as a fallback.

Sat Apr 28 23:43:10 ICT 2018

There was an unexpected error (type=Forbidden, status=403).

Forbidden

Latihan

Nama : Julio Muhammad Pranadano
NPM : 1706106785

1. Tampilkan menu view student by npm beserta input field dan button untuk memasukan nomor npm student yang ingin dicari pada halaman index user dengan role USER .

The image shows two screenshots of a web application. The top screenshot is a login page at `localhost:8080/login` with a 'User Name' field containing 'user', a 'Password' field with masked characters, and a 'Sign In' button. The bottom screenshot shows the application after login, displaying a 'Welcome' message, a 'Login' link, and a 'View by NPM' section. The 'View by NPM' section has an input field with '1706106785' and a 'Cari' button. Below this is a 'Navbar' with links for 'Home', 'Daftar Mahasiswa', and 'Menambah Mahasiswa'. The main content area shows 'Login as user' with a 'Sign Out' button, and displays the user's details: 'NPM = 1706106785', 'Name = Julio Muhammad Pranadano', and 'GPA = 3.65'.

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

Untuk melakukannya dapat menambahkan kode berikut pada index.html

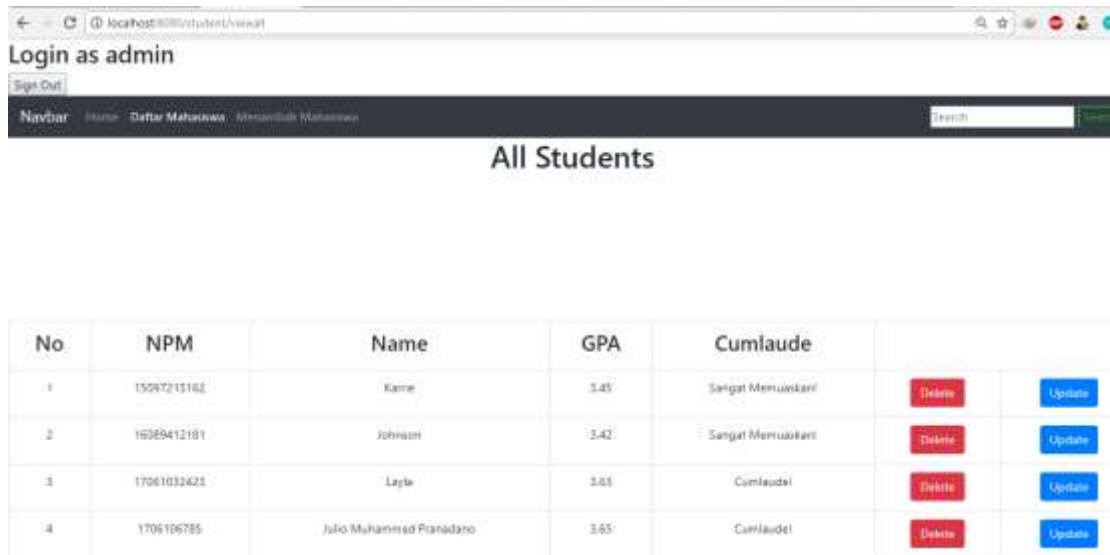
```
<div th:if="${#httpServletRequest.isUserInRole('USER')} or ${#httpServletRequest.isUserInRole('ADMIN')}">
```

Login sebagai admin

The image shows a screenshot of a web application at `localhost:8080/login?logout`. It displays a message 'You have been logged out.' followed by a login form. The 'User Name' field contains 'admin', and the 'Password' field has masked characters. There is a 'Sign In' button.

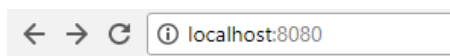
View All Student

Nama : Julio Muhammad Pranadano
NPM : 1706106785



No	NPM	Name	GPA	Cumlaude		
1	15287215162	Karne	3.45	Sangat Memuaskan	Delete	Update
2	16389412181	Johnaan	3.42	Sangat Memuaskan	Delete	Update
3	17061032622	Layla	3.65	Cumlaude	Delete	Update
4	1706106785	Julio Muhammad Pranadano	3.65	Cumlaude	Delete	Update

View by NPM

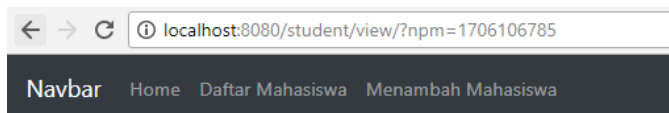
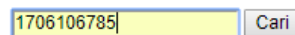


Welcome

[Login](#)

[View All Student](#)

View by NPM



Login as admin

Sign Out

NPM = 1706106785

Name = Julio Muhammad Pranadano

GPA = 3.65

Untuk melakukannya ubah method configure pada WebSecurityConfig dengan mengganti function hasRole menjadi hasAnyRole dan diisikan role tersebut dengan "Admin" dan "User".

```
.antMatchers("/student/view/**").hasAnyRole("ADMIN", "USER")
```