

TUGAS TUTORIAL 8
ARSITEKTUR DAN PEMROGRAMAN APLIKASI
ENTERPRISE



Disusun oleh :

ANDESTA PUTRA
NPM. 1706106633

PROGRAM SARJANA 1 EKSTENSI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS INDONESIA
2018

A. Ringkasan Materi

Pada Tutorial ini kita akan membahas tentang Web Security, yaitu pengembangan sistem yang dapat melakukan otentikasi dan juga otorisasi. Otentikasi sendiri adalah verifikasi yang dilakukan oleh sistem apakah pengguna mempunyai hak akses kedalam sistem atau tidak. Kemudian otorisasi adalah pengaturan dari hak pengguna yang memiliki otentikasi. Pada sistem juga dihipotesiskan memiliki level user yang berbeda menyesuaikan dengan kebutuhan sistem

B. Tutorial

Pada tutorial ini kita akan menggunakan project dari tutorial 5 untuk kita kembangkan dengan menambahkan fitur otentikasi dan juga otorisasi.

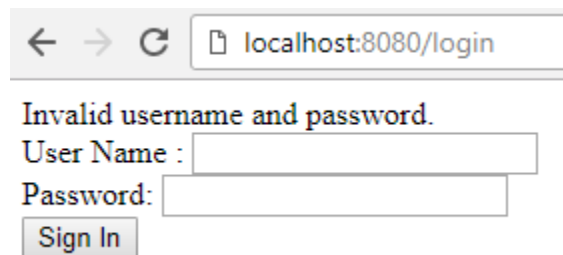
1. Simple Web Security

Untuk membuat otentikasi kita perlu menambahkan dependency Security sebagai berikut.

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

Selanjutnya kita akan membuat WebSecurityConfig.java pada package com.example dengan code seperti yang tertera pada tutorial. Adapun yang dapat kita pelajari dari WebSecurityConfig ini adalah pada method configure berisikan bentuk konfigurasi dari setiap request yang dicantumkan. Seperti pada request “/” dapat diakses tanpa membutuhkan permission, sedangkan request lainnya diperlukan otentikasi sehingga jika user belum memilikinya maka akan di alihkan ke request login.

Berikut ini merupakan tampilan login ketika user belum memiliki otentikasi terhadap sistem.



← → ↻

Invalid username and password.

User Name :

Password:

Ketika kita mengakses index atau request “/” maka sistem akan menampilkan index.html karena tidak ada proses otentikasi didalamnya.

Welcome To Tutorial 05 APAP

[Login](#)

Masukkan Nomor Pokok Mahasiswa

Search

Mata Kuliah APAP

2. Sign Out dan Nama username

Pada tahap ini kita akan membuat button untuk melakukan sign out dan juga menampilkan username sebagai tanda bahwa kita telah melakukan autentikasi.

Yang perlu dilakukan adalah menambahkan kode `${#HttpServletRequest.remoteUser}` untuk dapat menampilkan username yang kita gunakan untuk masuk ke dalam sistem.

Kemudian untuk logout sendiri kita perlu membuat form yang mengarahkan ke request logout untuk kemudian keluar dari autentikasi.

Tampilan button sign out dan juga username sebagai admin.

Navbar

Home

Daftar Mahasiswa

Menambah Mahasiswa

Search

Search

Sign Out

Tampil Semua Data

Login as admin

No	NPM	Name	GPA	Cum Laude	Delete	Update
1	1506737823	Kermie	3.32	Sangat Memuaskan!	Update Data	Delete Data
2	1506737824	Ellene	2.89	Sangat Memuaskan!	Update Data	Delete Data
3	1506737825	Delmer	2.58	Sangat Memuaskan!	Update Data	Delete Data
4	1506737826	Kristian	2.33	Sangat Memuaskan!	Update Data	Delete Data

Tampilan berhasil logout.

You have been logged out.

User Name :

Password:

Sign In

3. Membuat Menu dan Memanfaatkan Role

Pada tahap ini kita akan melakukan otorisasi dimana setiap user memiliki level yang berbeda terhadap hak akses ke dalam sistem. Kita dapat menambahkan kode pada method configure sebagai berikut pada WebSecurityConfig seperti dibawah ini.

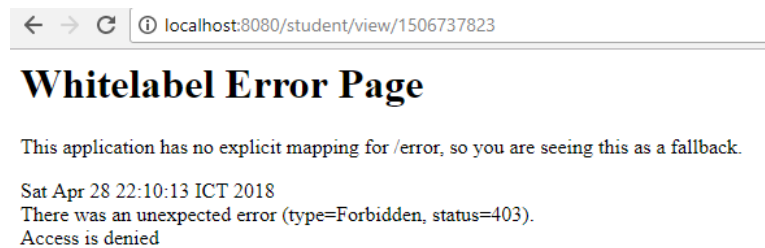
```

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

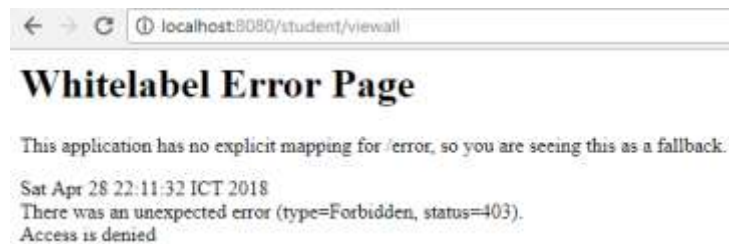
```

Adapun arti dari kode tersebut adalah untuk mengakses request student/viewall dibutuhkan role ADMIN, kemudian untuk student/view/** dibutuhkan role USER.

Sehingga ketika kita login sebagai admin dan membukan view/npm dan lainnya akan mendapati tampilan seperti ini.



Sedangkan apabila kita login sebagai USER kita tidak dapat mengakses student/viewall.



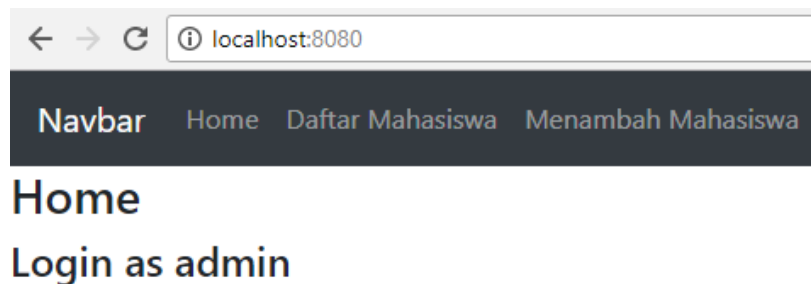
Selanjutnya kita juga dapat membuat kondisi agar ketika user login sebagai admin akan muncul navigasi view all student dengan menggunakan kode sebagai berikut.

```

<a th:if="${#httpServletRequest.isUserInRole('ADMIN')}}" class="nav-link"
href="/student/viewall">Daftar Mahasiswa</a>

```

Maka tampilan admin akan menjadi seperti berikut.



4. Mengambil Username dan Password dari Database

Pada langkah ini kita akan mencoba mengambil username dan password dari database. Untuk itu kita perlu membuat tabl users dan user_roles seperti yang tertera pada tutorial. Kemudian agar dapat mengaksesnya kita dapat mengganti method configureGlobal dengan method configAuthentication sebagai berikut.

```
@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=?");
}
```

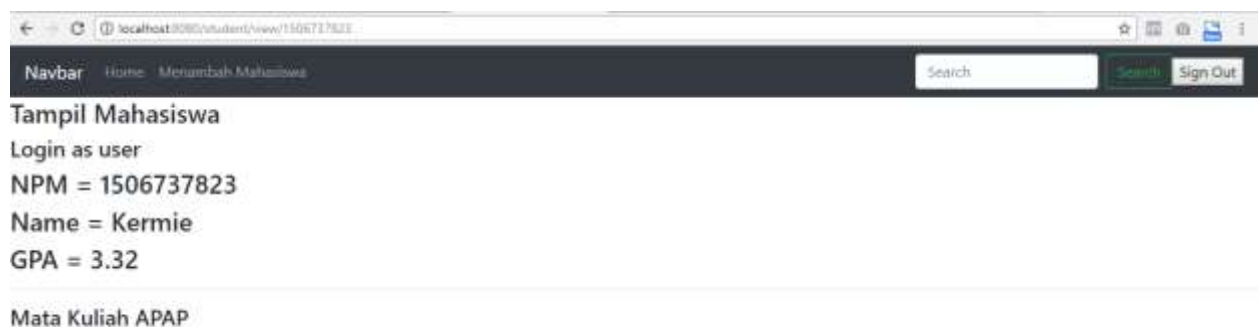
C. Pertanyaan

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

Kita dapat menambahkan butto dan nama user pada fragment header seperti berikut ini.

```
<form th:action="@{/logout}" method="post">
    <input type="submit" value="Sign Out"/>
</form>

</nav>
<h3 th:text="${title}">Ini Title</h3>
<h4 th:text=" 'Login as ' + ${#httpServletRequest.remoteUser}">Login as</h4>
```

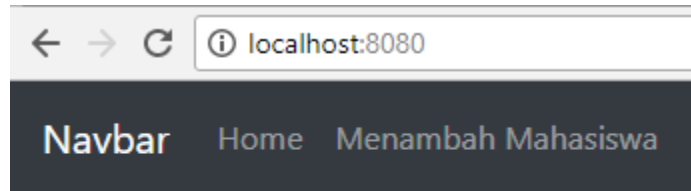


2. Buatlah user baru dengan username "user", password "user", dan role "USER". Coba apakah bisa login dengan akun tersebut.

Kita perlu menambahkan kode berikut pada method configureGlobal

```
auth.inMemoryAuthentication()
    .withUser("user").password("user")
    .roles("USER");
```

Dan hasil login adalah sebagai berikut.



Home

Login as user

3. 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.

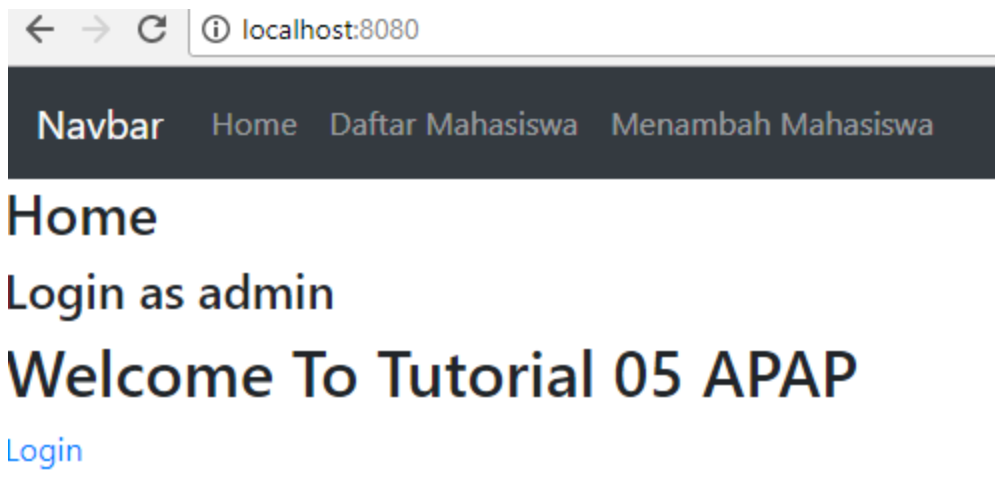
Kita dapat menambahkan kode berikut pada index.html untuk membuat form npm serta kondisi apabila user login sebagai USER.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <body>
    <div th:replace="fragments/fragment :: header"></div>

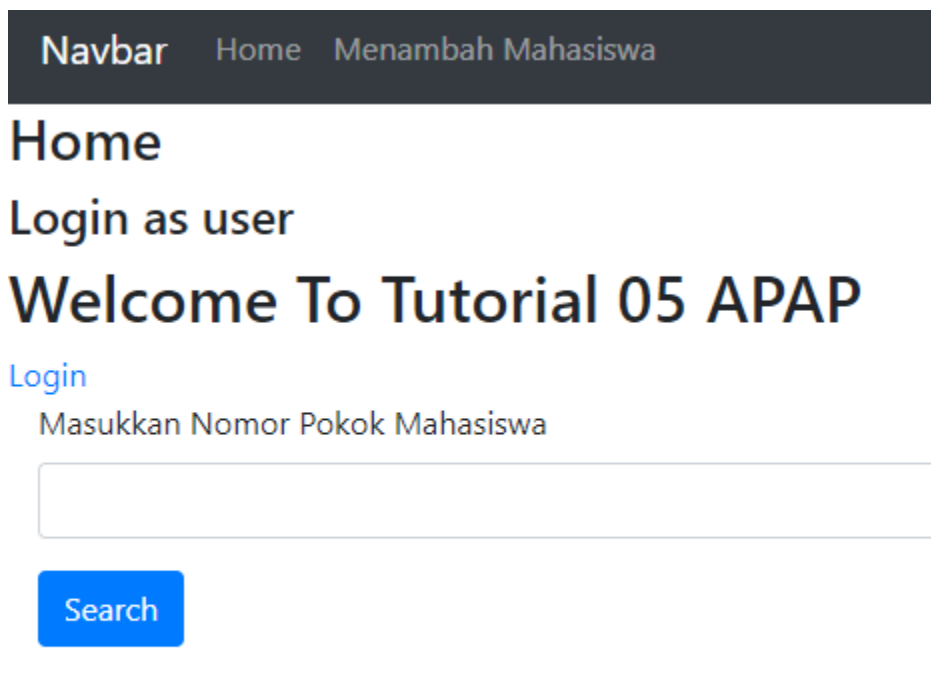
    <h2>Welcome To Tutorial 05 APAP</h2>
    <a href="/login">Login</a>

    <div class="col-md-12">
      <form th:if="${#httpServletRequest.isUserInRole('USER')}}" action="/student/view/" method="get">
        <div class="form-group">
          <label for="npm">Masukkan Nomor Pokok Mahasiswa</label> <input class="form-control" type="text" name="npm">
        </div>
        <div class="form-group">
          <button type="submit" class="btn btn-primary">Search</button>
        </div>
      </form>
    </div>

    <div th:replace="fragments/fragment :: footer"></div>
  </body>
</html>
```



Mata Kuliah APAP

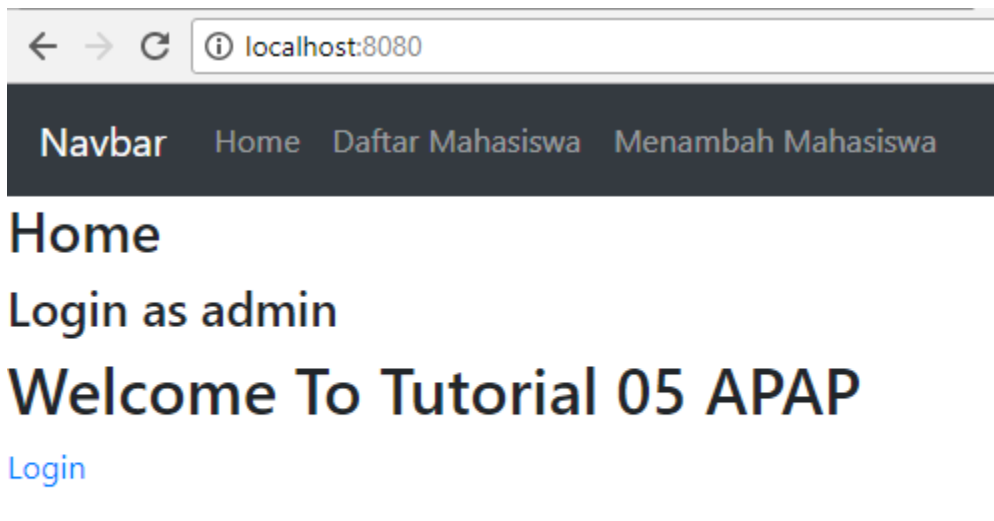


Mata Kuliah APAP

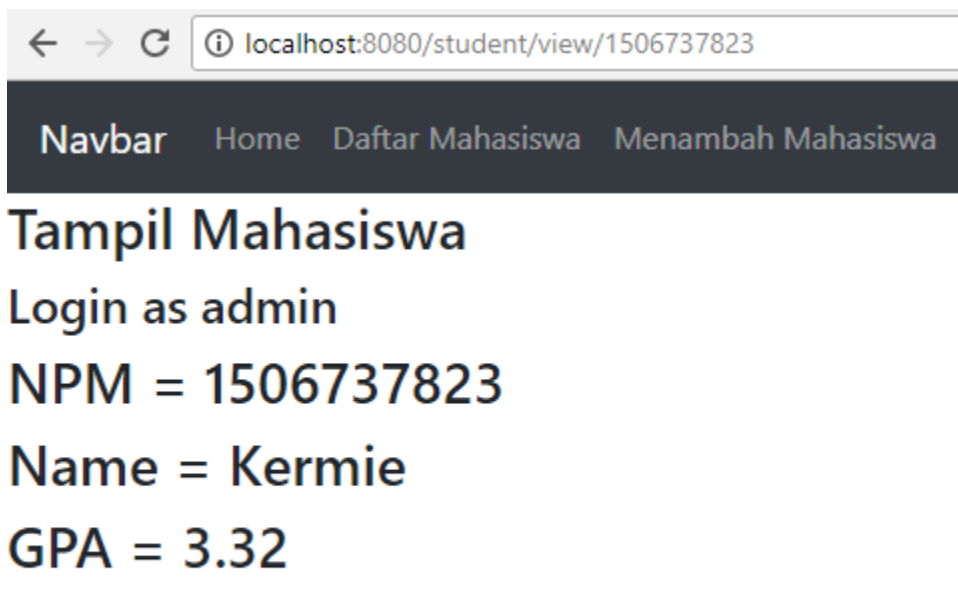
4. Buat agar user dengan role ADMIN dapat melihat view all student dan view student by npm juga.

Untuk melakukan ini kita hanya perlu mengubah kembali method configure pada WebSecurityConfig. Pada kasus ini kita dapat mengganti function hasRole menjadi hasAnyRole dan diisi dengan role ADMIN dan USER sebagai berikut.

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



Mata Kuliah APAP



Mata Kuliah APAP