Tutorial 08 – Web Security
CSIM603026 – Arsitektur dan Pemrograman Aplikasi Perusahaan Semester Genap 2017/2018

Mengimplementasikan Web Security Pada Spring Boot Framework

Simple Web Security

Dalam mengembangkan sebuah sistem yang digunakan oleh tingkatan pengguna yang berbeda, Anda perlu membuat aturan otentikasi dan otorisasi. Otentikasi diperlukan untuk melakukan verifikasi bahwa pengguna mempunyai akses untuk masuk ke dalam sistem. Otorisasi merupakan pengaturan hak pengguna yang telah memiliki otentikasi.

Tambahkan dependency Spring Boot Security berikut pada pom.xml untuk menambahkan dependency security.

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

Buat class baru yaitu WebSecurityConfig.java pada package com.example (setara dengan main class) yang berisi dua method yaitu WebSecurityConfig yang melakukan setting configuration dan method configureGlobal untuk mengeset otentikasi dimana user yang dapat mengakses aplikasi adalah user dengan username “admin” dan password “admin”.

```
public class WebSecurityConfig {
    protected void configure(HttpSecurity http) throws Exception
    {
        http
        .authorizeRequests()
        .antMatchers("/").permitAll()
        .anyRequest().authenticated()
        .and()
        .formLogin()
        .loginPage("/login")
        .permitAll()
        .and()
        .logout()
        .permitAll();
    }

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

Buatlah halaman login.html dengan isi sebagai berikut yang mengecek apakah username dan password yang diinput sesuai dengan konfigurasi atau tidak. Jika tidak maka akan menampilkan pesan invalid username and password dan jika sudah logout menampilkan you have been logged out.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org"
xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
    <head>
     <title>Login Page</title>
    </head>
    <body>
         <div th:if="${param.error}">
         Invalid username and password.
         </div>
         <div th:if="${param.logout}">
            You have been logged out.
         </div>
            <form th:action="@{/login}" method="post">
                <div><label> User Name : <input type="text" name="username"/>
                </label></div>
                <div><label> Password: <input type="password" name="password"/>
                </label></div>
                <div><input type="submit" value="Sign In"/></div>
            </form>
    </body>
</html>
```

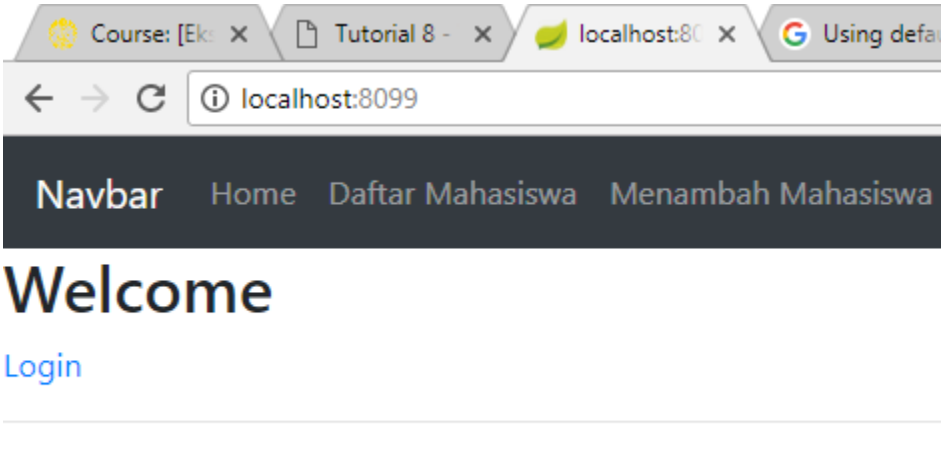
Buatlah juga halaman index.html (jika belum) yang berisi sapaan dan link untuk login. Contoh:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head th:replace="fragments/fragment :: assets" />
    <body>
         <div th:replace="fragments/fragment :: header"></div>
            <h2>Welcome</h2>
            <a href="/login">Login</a>
         <div th:replace="fragments/fragment :: footer"></div>
    </body>
</html>
```

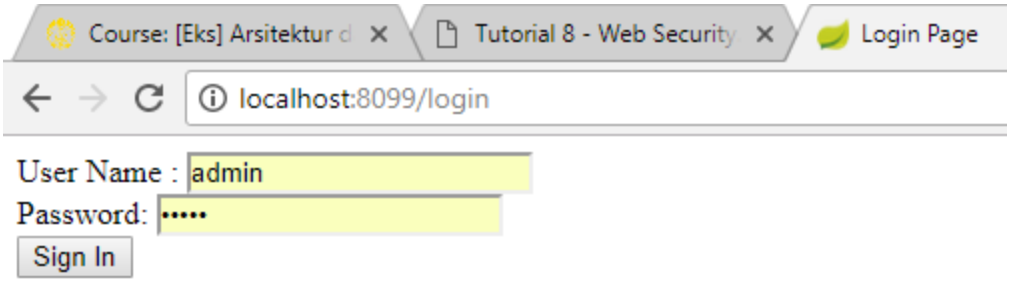
Buatlah class PageController.java dan tambahkan berikut:

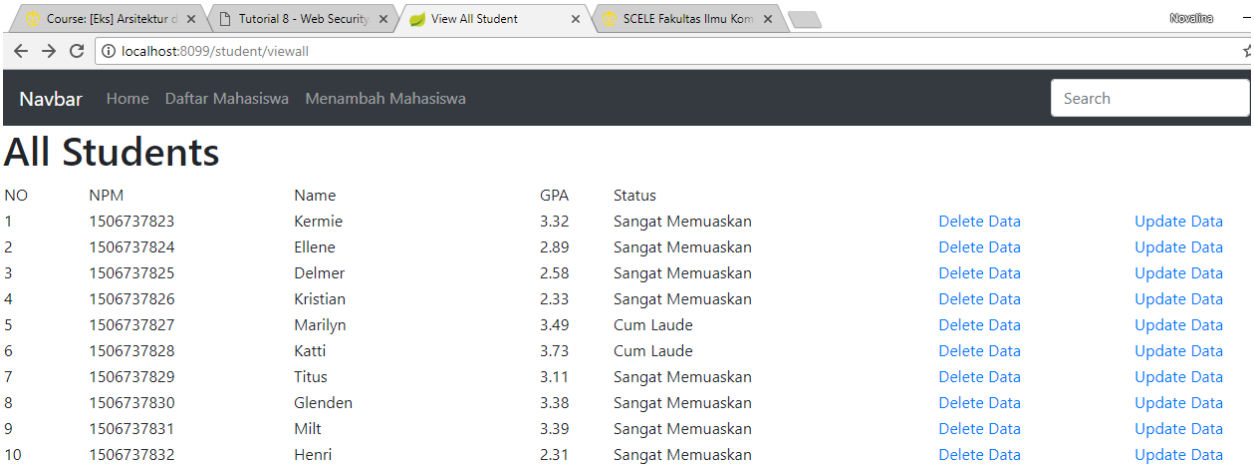
```
package com.example.controller;

import org.springframework.web.bind.annotation.RequestMapping;

public class PageController {
    @RequestMapping("/")
    public String index () {
        return "index";
    }

    @RequestMapping("/login")
    public String login () {
        return "login";
    }

}
```

Ketika akses localhost:



Buka localhost:8080/student/viewall maka Anda akan dialihkan ke halaman login sebagai berikut, masukkan username dengan “admin” dan password dengan “admin” seperti paga gambar di bawah ini.





NO	NPM	Name	GPA	Status		
1	1506737823	Kermie	3.32	Sangat Memuaskan	Delete Data	Update Data
2	1506737824	Ellene	2.89	Sangat Memuaskan	Delete Data	Update Data
3	1506737825	Delmer	2.58	Sangat Memuaskan	Delete Data	Update Data
4	1506737826	Kristian	2.33	Sangat Memuaskan	Delete Data	Update Data
5	1506737827	Marilyn	3.49	Cum Laude	Delete Data	Update Data
6	1506737828	Katti	3.73	Cum Laude	Delete Data	Update Data
7	1506737829	Titus	3.11	Sangat Memuaskan	Delete Data	Update Data
8	1506737830	Glenden	3.38	Sangat Memuaskan	Delete Data	Update Data
9	1506737831	Milt	3.39	Sangat Memuaskan	Delete Data	Update Data
10	1506737832	Henri	2.31	Sangat Memuaskan	Delete Data	Update Data

Sign Out dan Nama Username

Pada bagian ini kita akan menambahkan tombol sign out dan mencetak nama pengguna pada halaman "student/viewall"

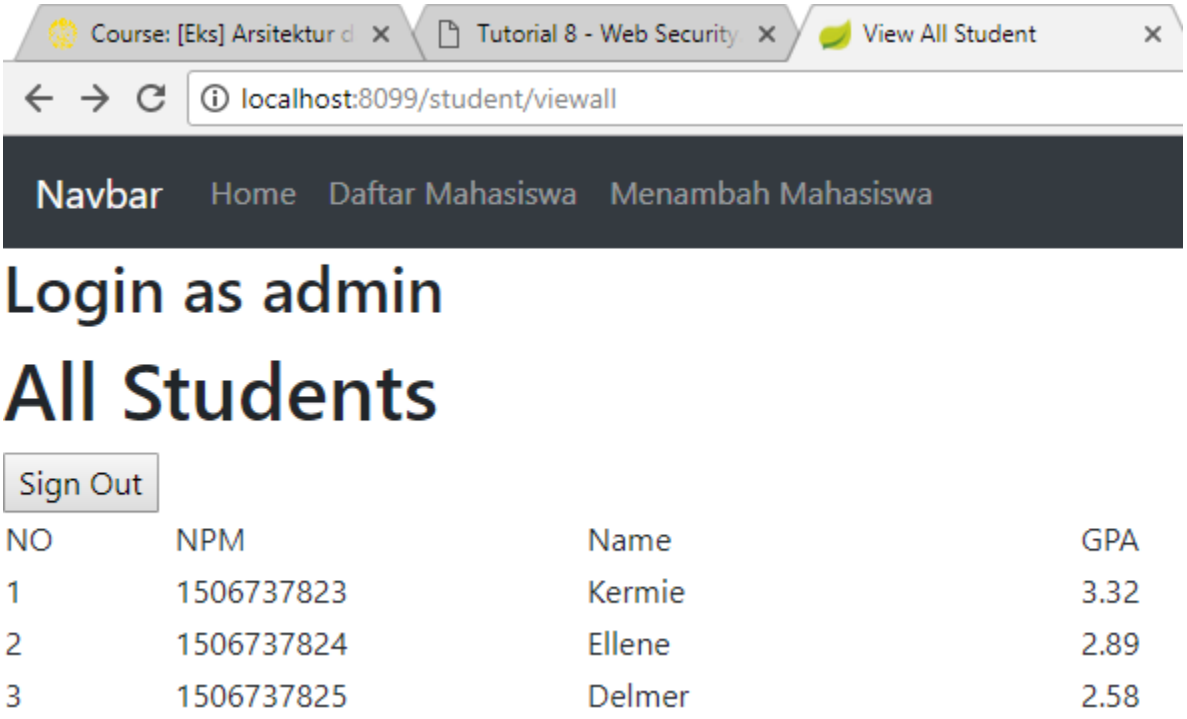
- Pada berkas viewall.html tambahkan baris berikut di bawah tag body:

```
<h2 th:text=" 'Login as ' + ${#httpServletRequest.remoteUser}">Login as</h2>
```

- Untuk tombol logout tambahkan baris berikut:

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

Hasilnya sebagai berikut dimana pengguna dapat mengakses menu viewall yang menampilkan pesan login as admin dan table berisi data student.



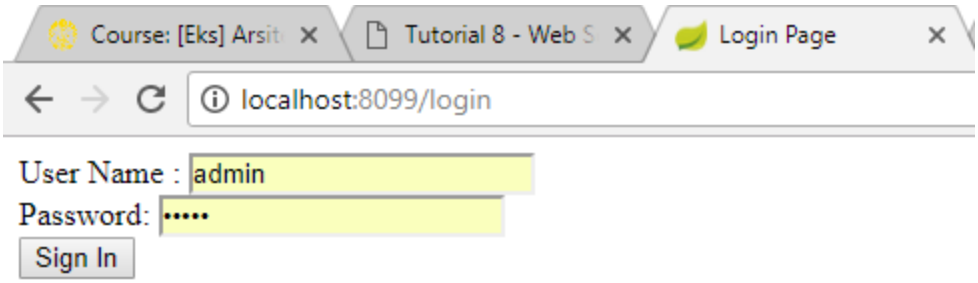
Latihan

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

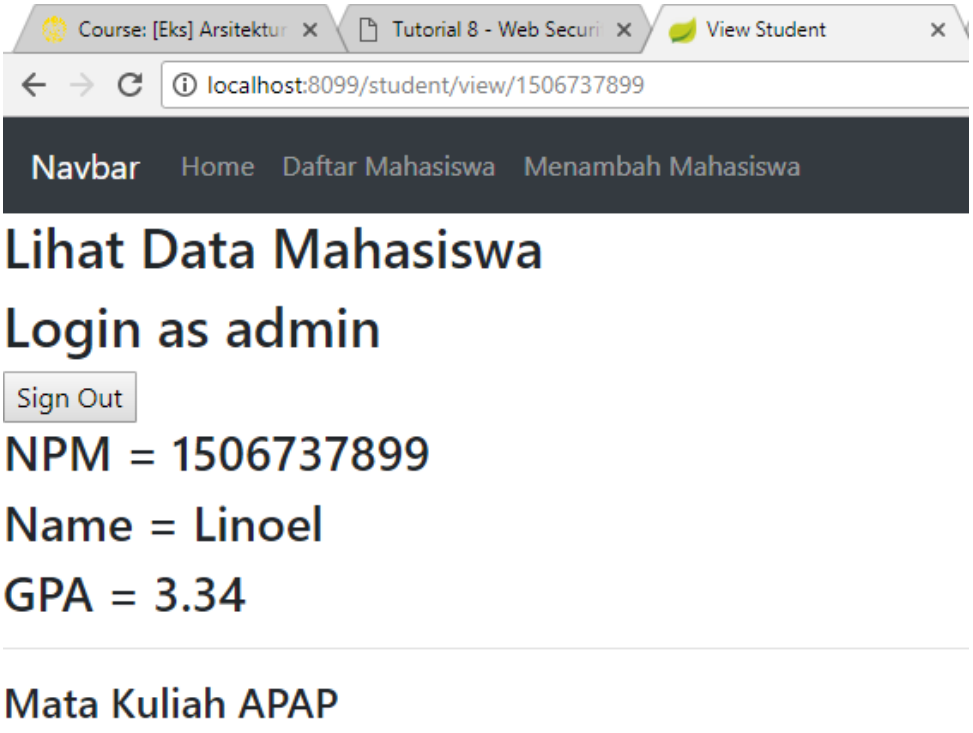
Cara yang dilakukan adalah sebagai berikut.

- Menambahkan button logout dan cetak nama user seperti paga gambar di bawah ini.

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



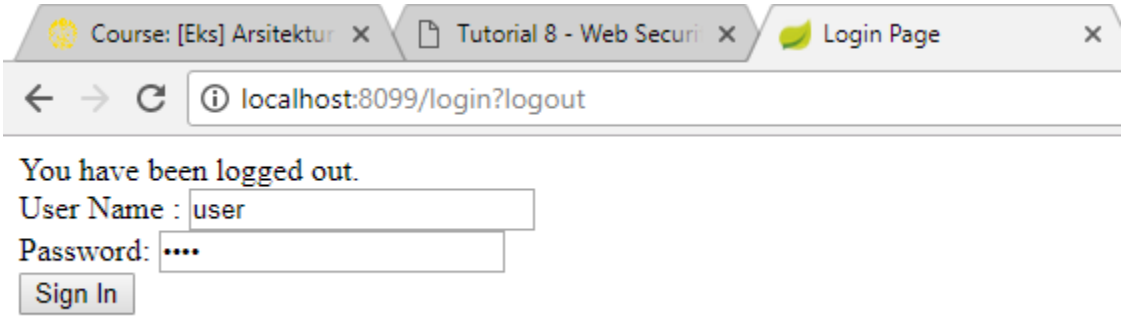
Hasilnya yaitu adanya button sign out dan pesan login as admin seperti pada gambar di bawah ini.



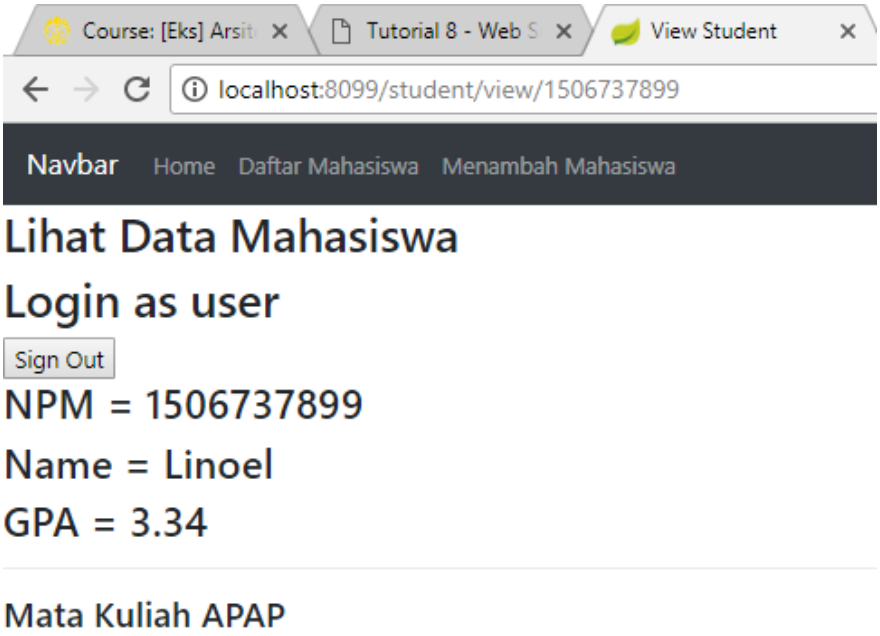
2. Buatlah user baru dengan username “user”, password “user”, dan role “USER”. Coba apakah bisa login dengan akun tersebut.
   - Caranya adalah dengan menambahkan akun dan password “user” dengan role “USER” pada method configureGlobal seperti pada gambar di bawah ini.

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

Hasilnya adalah sebagai berikut.



Berikut ini adalah hasilnya dimana user dengan username dan password user berhasil login dan menampilkan halaman.



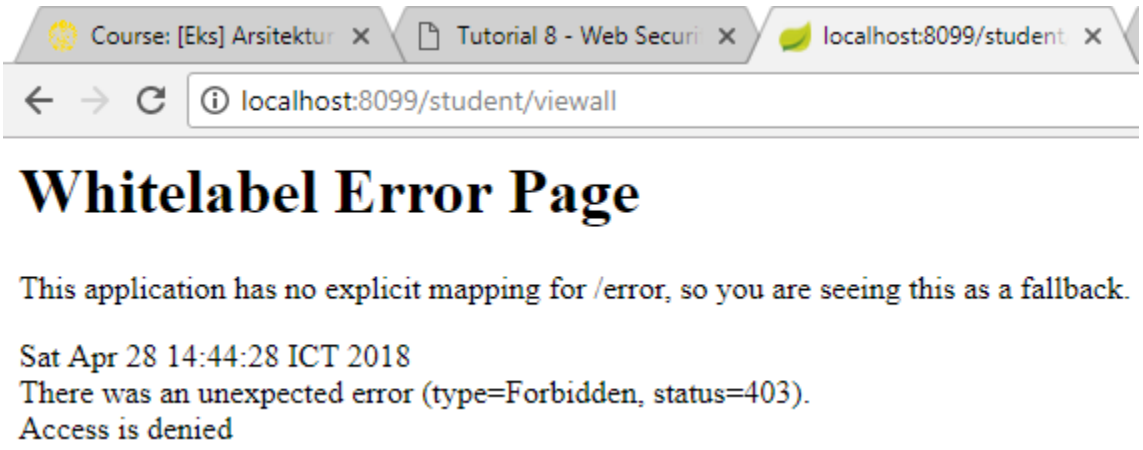
Membuat Menu dan Memanfaatkan Role

Pada bagian ini kita akan membuat menu sesuai dengan role. Selanjutnya kita menginginkan agar halaman untuk view all student hanya dapat dilihat oleh user dengan role ADMIN. Sedangkan halaman untuk view student by npm hanya dapat dilihat oleh user dengan role USER.

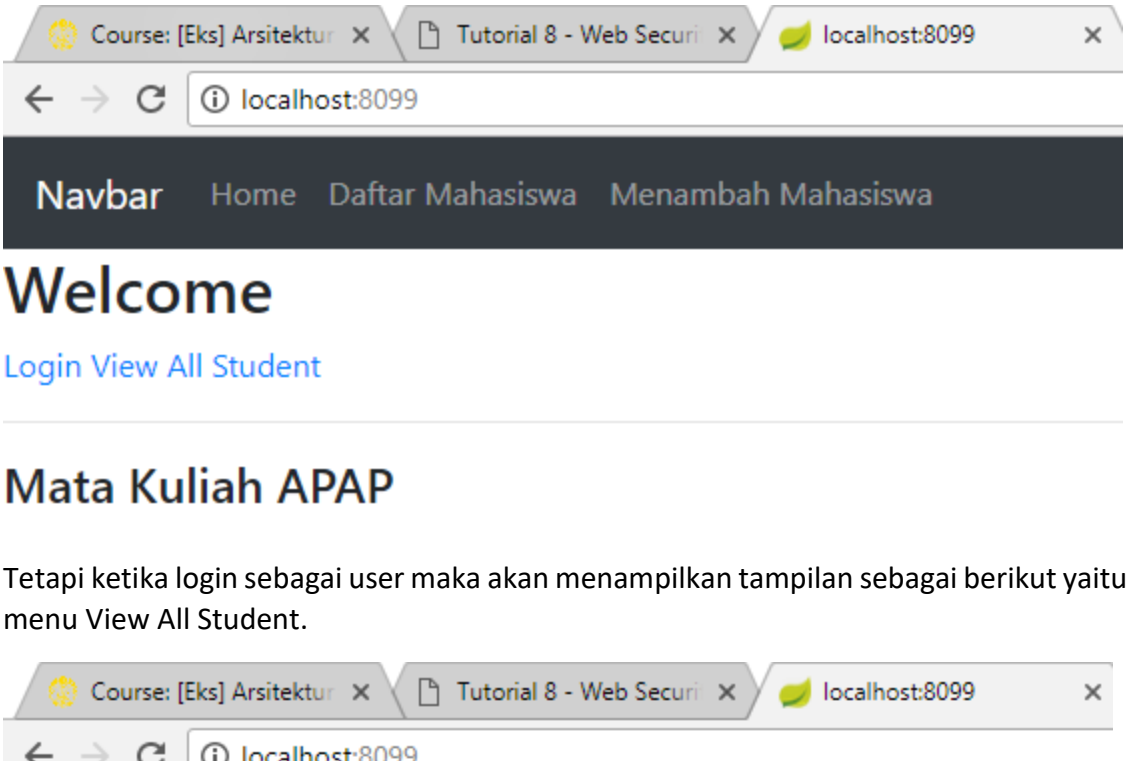
Pada method configure pada class WebSecurityConfig harus ditambahkan antMatchers dengan nama menu dan role yang dapat mengakses menu tersebut.

```
@Override
protected void configure(HttpSecurity http) throws Exception
{
    http
    .authorizeRequests()
    .antMatchers("/").permitAll()
    .antMatchers("/student/viewall").hasRole("ADMIN")
    .antMatchers("/student/view/*").hasRole("USER")
    .anyRequest().authenticated()
    .and()
    .formLogin()
    .loginPage("/login")
    .permitAll()
    .and()
    .logout()
    .permitAll();
}
```

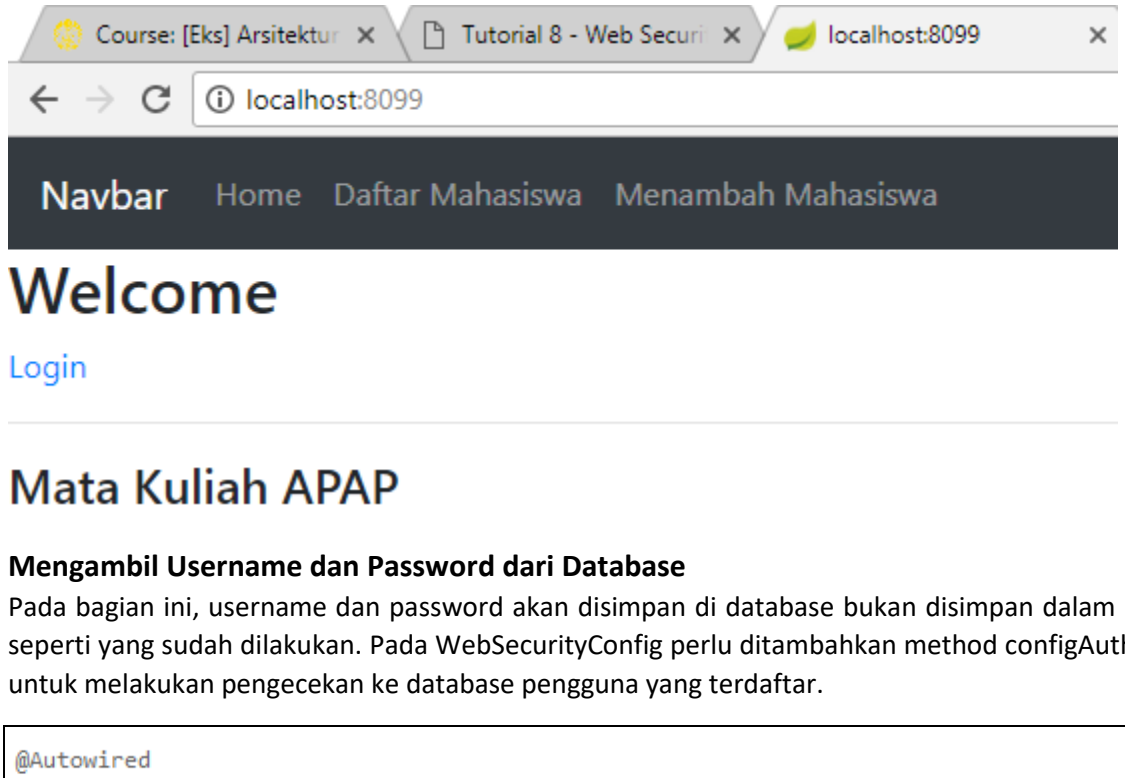
Ketika login dengan user dan membuka menu viewall, maka akan menampilkan gambar seperti berikut ini. Ini membuktikan bahwa user dengan username “user” tidak dapat mengakses menu viewall.



Sedangkan ketika login sebagai admin maka muncul tampilan berikut dimana terdapat link menuju menu View All Student yang dapat diakses hanya pengguna dengan role admin.



Tetapi ketika login sebagai user maka akan menampilkan tampilan sebagai berikut yaitu tidak ada menu View All Student.

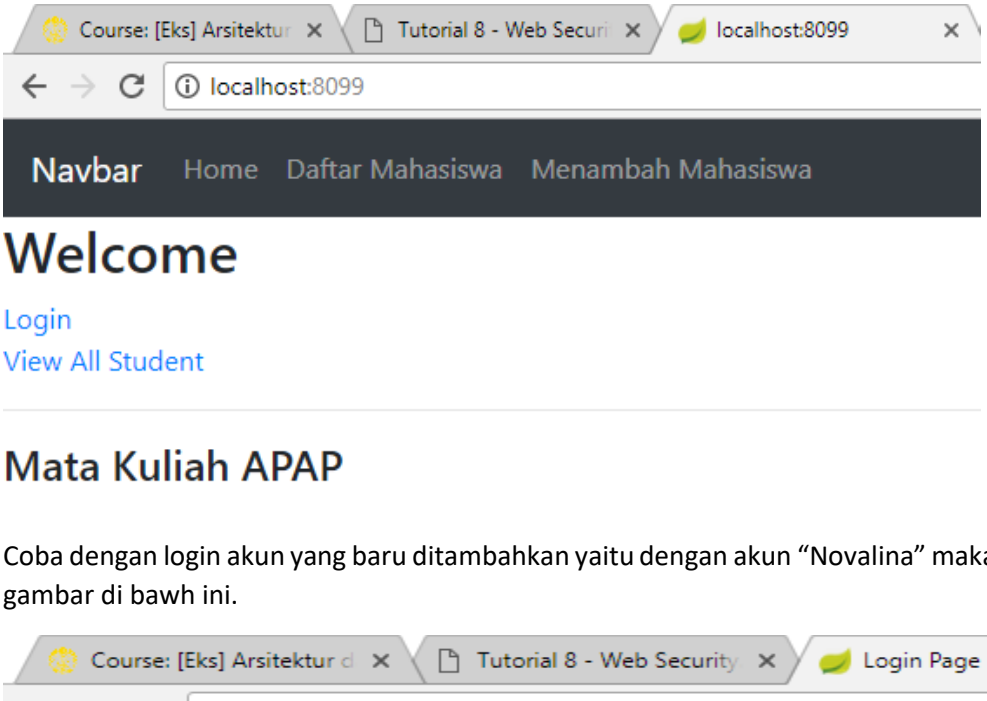


Mengambil Username dan Password dari Database

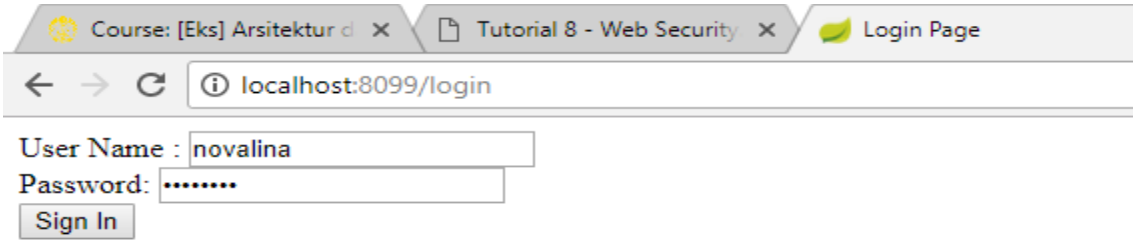
Pada bagian ini, username dan password akan disimpan di database bukan disimpan dalam in-memory seperti yang sudah dilakukan. Pada WebSecurityConfig perlu ditambahkan method configAuthentication untuk melakukan pengecekan ke database pengguna yang terdaftar.

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

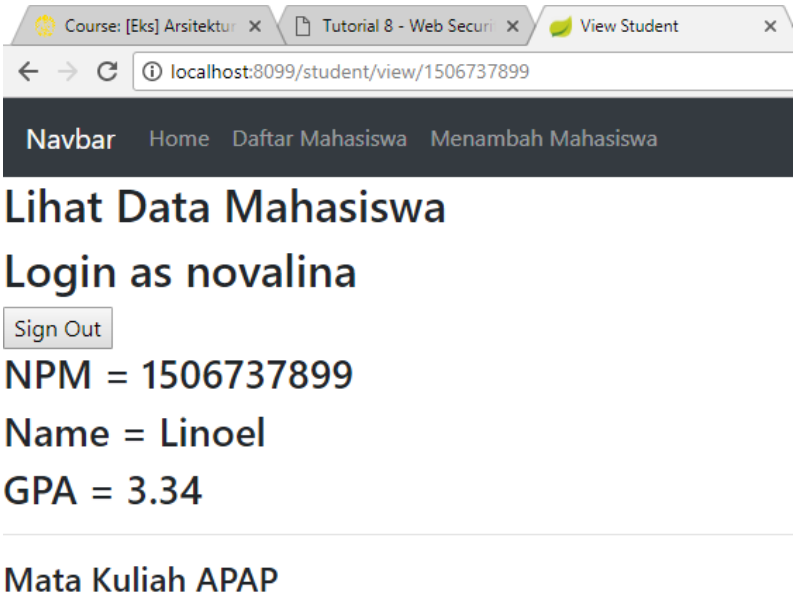
Setelah dijalankan maka ketika login admin akan muncul seperti tampilan seperti berikut.



Coba dengan login akun yang baru ditambahkan yaitu dengan akun “Novalina” maka hasilnya seperti pada gambar di bawh ini.



Ketika berhasil login maka user dapat mengakses menu yang dapat diakses dengan role “USER” seperti pada gambar berikut.



Latihan

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.
   - Pertama tambahkan link untuk menu view student by npm pada halaman index dengan pengecekan role "USER" yang dapat mengakses.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head th:replace="fragments/fragment :: assets" />
    <body>
         <div th:replace="fragments/fragment :: header"></div>
            <h2>Welcome</h2>
            <a href="/login">Login</a>
            <br/>
            <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
            <a th:if="${#httpServletRequest.isUserInRole('USER')}" href="/student/viewall">View Student by NPM</a><br/>
         <div th:replace="fragments/fragment :: footer"></div>
    </body>
</html>
```

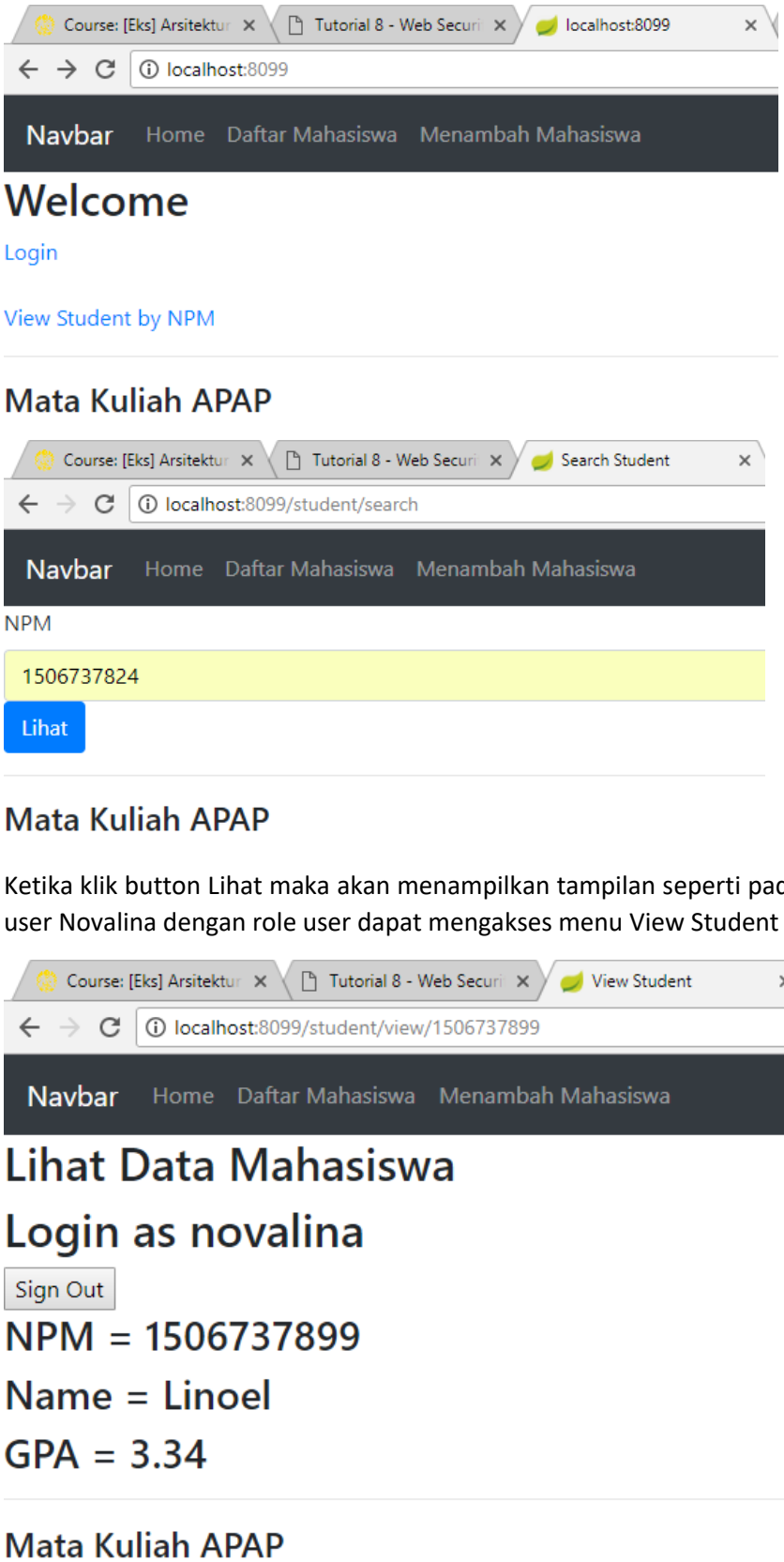
   - Kemudian menambahkan form untuk meminta inputan mencari student berdasarkan npm

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head th:replace="fragments/fragment :: assets" />
    <body>
         <div th:replace="fragments/fragment :: header"></div>
            <form action="/student/view" method="get">
                <div>
                    <input type="text" name="npm" class="form-control" id="npm" width="5px" />
                </div>
                <button type="submit" class="btn btn-primary">Lihat</button>

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

   - Perlu menambahkan method pada controller yang akan memanggil form search tersebut.

```
@RequestMapping("/student/search")
public String search (Model model)
{
    model.addAttribute("title", "Search Student");
    return "search";
}
```

   - Hasilnya seperti pada gambar di bawah ini yaitu setelah login maka pada tampilan index akan menampilkan menu View Student by NPM yang jika diklik akan menampilkan form input NPM seprti pada gambar di bawah ini.



Ketika klik button Lihat maka akan menampilkan tampilan seperti pada gambar di bawah ini dimana user Novalina dengan role user dapat mengakses menu View Student by NPM.

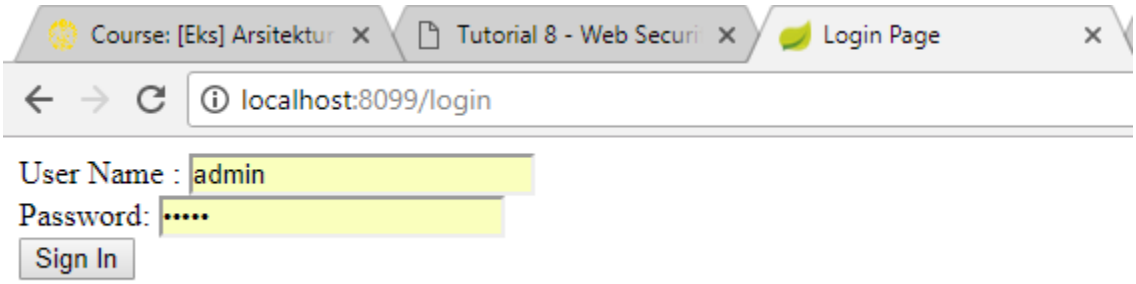
4. Buat agar user dengan role ADMIN dapat melihat view all student dan view student by npm juga.
   - Pertama sekali setingan pada Web Security Config untuk menu student/view harus dicomment dengan demikian maka tidak ada lagi batasan untuk pengguna yang membuka email tersebut.

```
@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")
        .anyRequest().authenticated()
        .and()
        .formLogin()
        .loginPage("/login")
        .permitAll()
        .and()
        .logout()
        .permitAll();
    }
```

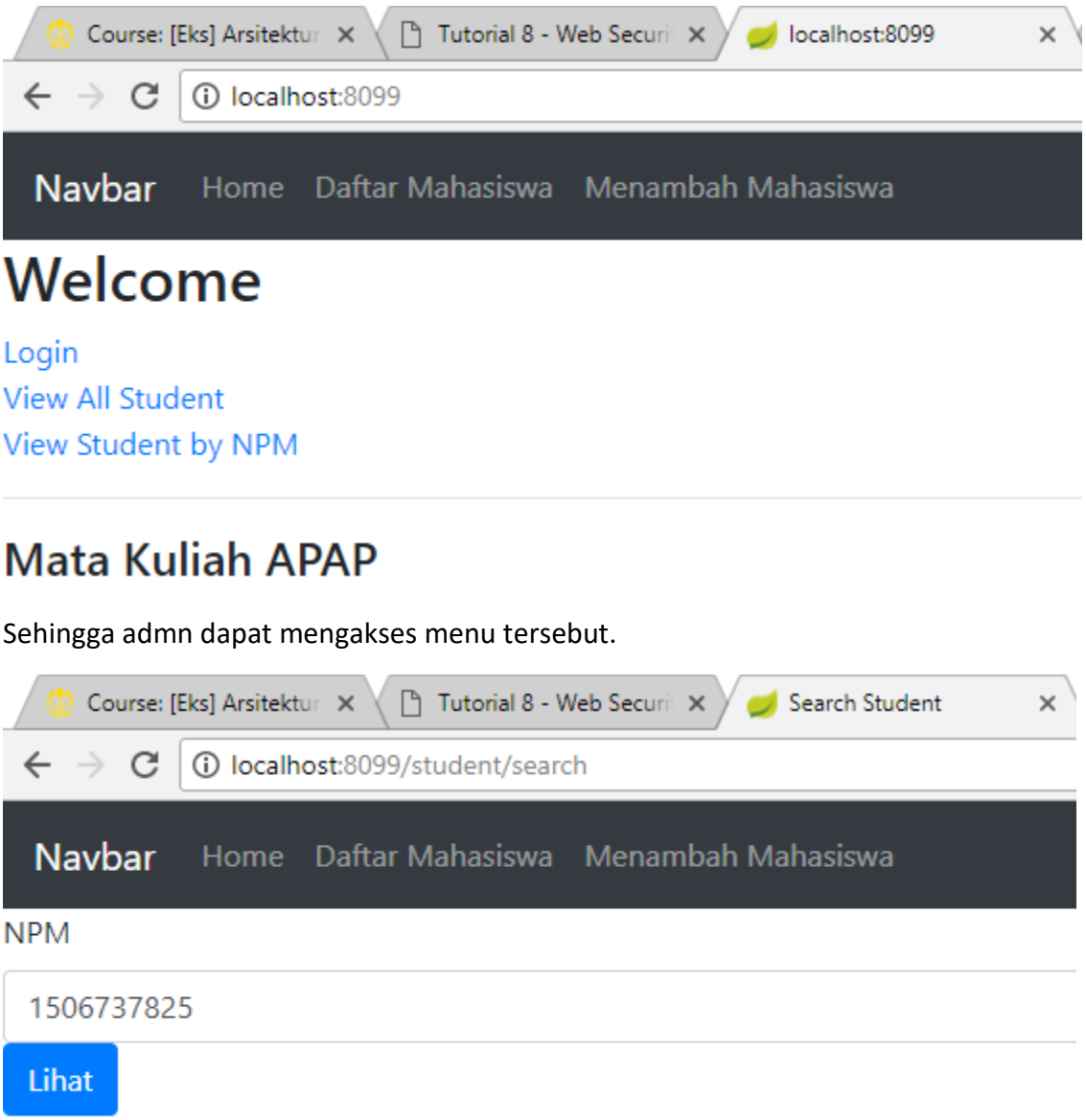
   - Kemudian pada halaman index.html pengecekan untuk user yang login dengan role USER harus dihapus sehingga memungkinkan user manapun untuk mengakses menu view student by npm tersebut.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head th:replace="fragments/fragment :: assets" />
    <body>
        <div th:replace="fragments/fragment :: header"></div>
           <h2>Welcome</h2>
           <a href="/login">Login</a>
           <br/>
           <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
           <!-- a th:if="${#httpServletRequest.isUserInRole('USER')}" href="/student/search">View Student by NPM</a><br/> -->
           <a href="/student/search">View Student by NPM</a><br/>
        <div th:replace="fragments/fragment :: footer"></div>
    </body>
</html>
```

Hasilnya adalah sebagai berikut.

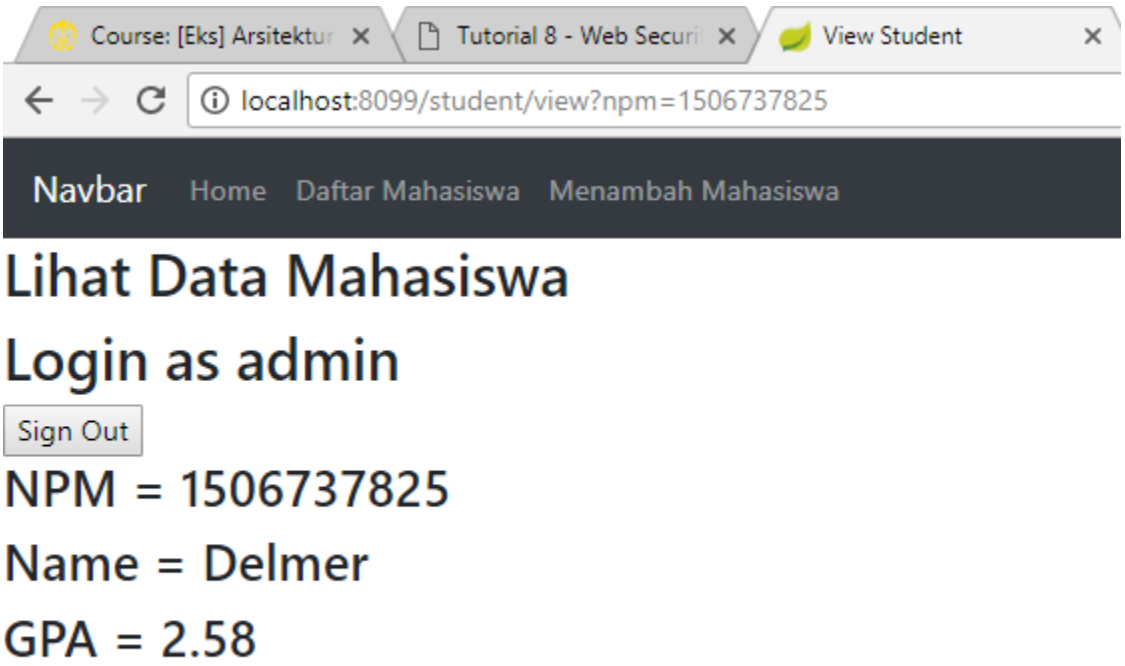


Dengan demikian setelah user admin login maka menu view all student dan view student by npm akan ditampilkan seperti pada gambar berikut ini.

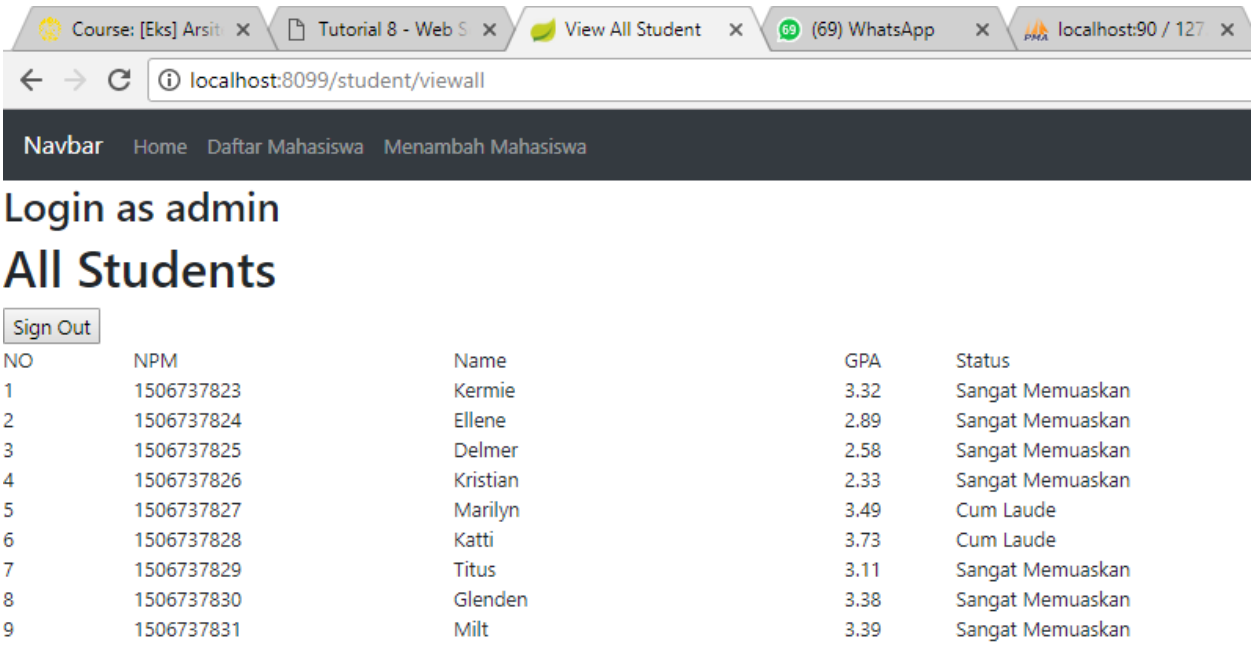


Sehingga admn dapat mengakses menu tersebut.

Berikut ini tampilan ketika user dengan username admin mengakses menu view student by NPM.



Berikut ini tampilan ketika user “admin” mengakses menu viewall.



NO	NPM	Name	GPA	Status
1	1506737823	Kermie	3.32	Sangat Memuaskan
2	1506737824	Ellene	2.89	Sangat Memuaskan
3	1506737825	Delmer	2.58	Sangat Memuaskan
4	1506737826	Kristian	2.33	Sangat Memuaskan
5	1506737827	Marilyn	3.49	Cum Laude
6	1506737828	Katti	3.73	Cum Laude
7	1506737829	Titus	3.11	Sangat Memuaskan
8	1506737830	Glenden	3.38	Sangat Memuaskan
9	1506737831	Milt	3.39	Sangat Memuaskan