

Rangkuman Tutorial 8

Materi yang dapat dipelajari dari tutorial 8 ini adalah mengenal *web security*. Suatu aplikasi yang dikembangkan akan digunakan oleh *user*, namun *user* ini merupakan *user* yang memiliki wewenang dalam mengaplikasikan sistem tersebut. Dalam *web security* ini ialah membangun sebuah sistem yang menerapkan sistem otentikasi dan otorisasi. Otentikasi ini merupakan sistem untuk seorang *user*, apakah *user* ini adalah pengguna yang valid atau tidak. Otorisasi ialah memberikan *role* atau peran atau wewenang yang dapat dilakukan dalam mengoperasikan sistem.

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

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>View Student by NPM</title>
</head>
<body>
<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/navbar :: footer"></div>
</body>
</html>
```

Berikut adalah kode penambahan sebuah tombol SignOut pada halaman `/student/view/{npm}`.

← → ↻ localhost:8080/student/view/1506742809

Login as admin

NPM = 1506742809

Name = Hastings

GPA = 2.04

Mata Kuliah APAP

Berikut adalah tampilan pada 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");
}
```

Berikut ini adalah kode penambahan role untuk ‘USER’ pada class `WebSecurityConfig`. Pada `auth` dan setelah `roles(‘ADMIN’)` ditambahkan method `and()` untuk menambahkan otentikasi dengan kasus ini adalah otentikasi untuk ‘USER’.

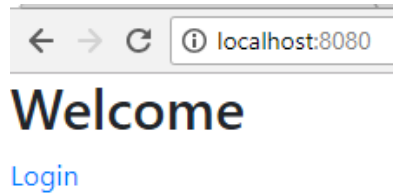
← → ↻ ⓘ localhost:8080/login

User Name : user

Password:

Sign In

Berikut ini tes pada halaman login , login disini akan menggunakan role 'USER' dengan username dan password 'user', dimana yang sudah tambahkan kode sebelumnya di class WebSecurityConfig.



Login tersebut berhasil mengotentikasi 'user' dan pergi ke halaman index.

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.

```
<!doctype HTML>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:fragment="common_header(content)">
<title>Index</title>
<link th:fragment="script" rel="stylesheet" href="/bootstrap-4.0.0-dist/css/bootstrap.min.css" />
</head>
<body>
<h2>Welcome</h2>
<div>
<a href="/Login">Login</a>
</div>
<div>
<a th:if="${HttpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All S
</div>
<div th:if="${HttpServletRequest.isUserInRole('USER')}">
<h2>View by NPM</h2>
<form action="/student/view/" method="get">
<input type="text" name="npm"/>
<button type="submit">Cari</button>
</form>
</div>
</body>
</html>
```

Berikut ini adalah kode pada halaman login, dimana ditambahkan sebiah input field dan button untuk memasukkan npm dan men-submitnya.

← → ↻ ⓘ localhost:8080/login

User Name : user

Password:

Sign In

Kasus ini menggunakan akun 'user'.

← → ↻ ⓘ localhost:8080

Welcome

[Login](#)

View by NPM

Setelah berhasil masuk sebagai 'user' maka *user* tersebut akan masuk ke halaman index dan halaman tersebut memunculkan sebuah text field dan tombol untuk mencari seorang student berdasarkan NPM.

← → ↻ ⓘ localhost:8080/student/view/?npm=1506742809

Login as user

NPM = 1506742809

Name = Hastings

GPA = 2.04

Mata Kuliah APAP

Berikut adalah ketika sukses memasukkan NPM dan mencari seorang student sesuai dengan NPM yang diberikan.

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

```

<!doctype HTML>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:fragment="common_header(content)">
<title>Index</title>
<link th:fragment="script" rel="stylesheet" href="/bootstrap-4.0.0-dist/css/bootstrap.min.css" />
</head>
<body>
<h2>Welcome</h2>
<div>
<a href="/Login">Login</a>
</div>
<div>
<a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
</div>
<div th:if="${#httpServletRequest.isUserInRole('USER')} or ${#httpServletRequest.isUserInRole('ADMIN')}">
<h2>View by NPM</h2>
<form action="/student/view/" method="get">
<input type="text" name="npm"/>
<button type="submit">Cari</button>
</form>
</div>
</body>
</html>

```

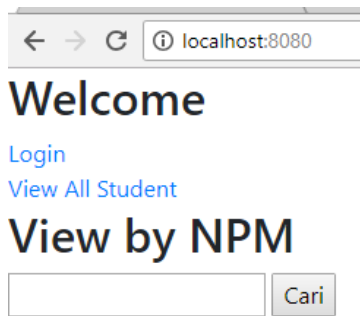
Berikut adalah kode pada halaman index. Pada halaman ditambahkan sebuah 'or', dimana untuk memunculkan halaman view by npm ketika masuk sebagai 'admin'.

← → ↻ ⓘ localhost:8080/login

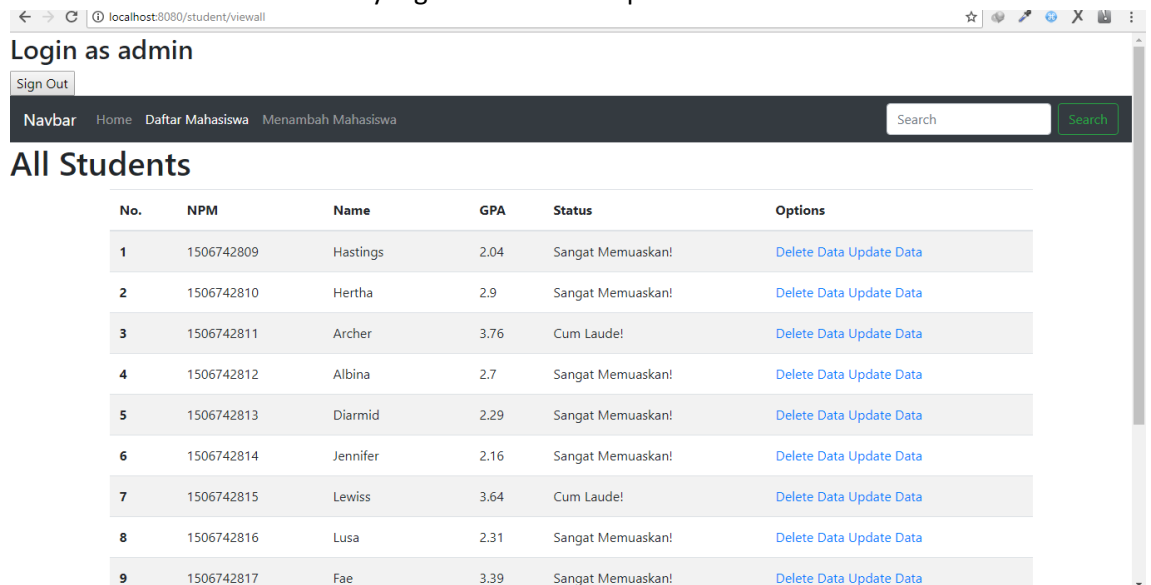
User Name :

Password:

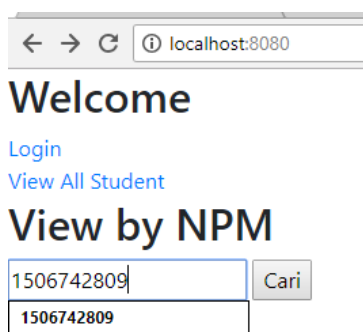
Tes otentikasi dengan pengguna sebagai admin.



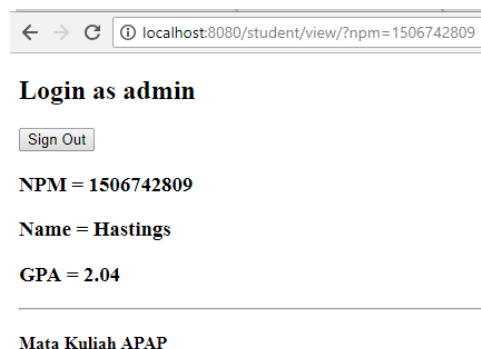
Berikut adalah halaman index yang berhasil menampilkan menu view dan viewall.



Berikut ketika pengguna masuk ke link viewall dan berhasil menampilkan semua data student.



Berikut ketika tes untuk mencari student berdasarkan NPM.



Hasilnya seperti diatas.