Pada tutorial ini kita belajar mengenai web security. Dalam mengembangkan web kita perlu melakukan verifikasi terhadap pengguna sistem kita. Sehingga tidak semua user bisa masuk kehalaman yang tidak sesuai dengan rule nya. Setelah mem verifikasi/otentikasi terhadap siapa saja pengguna yang berhak untuk masuk kedalam sistem, langkah selanjutnya adalah melakukan otorisasi. Otorisasi merupakan pengaturan hak pengguna yang sebelumnya telah di otentikasi. Pengguna dibedakan berdasarkan levelnya masing – masing agar tujuan dari sistem tercapai.

Latihan

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

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

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<div th:replace="fragments/fragment :: header"></div>
<body>
<h4>Cari mahasiswa dengan NPM</h4>
<form action="/student/view" method="get">
    <input type="number" name="npm" placeholder="NPM">
    <input type="submit" value="Cari">
</form>
</body>
<div th:replace="fragments/fragment :: footer"></div>
</html>
```

Pada form tersebut kita mengarahkan action nya pada /student/view/npm dengan method nya adalah get. Kemudian untuk akses role nya diatur pada class WebSecurityConfig.

```
.antMatchers( ...antPatterns: "/student/search").hasRole("USER")
```

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

Menggunakan method hasAnyRole , memungkinkan memiliki lebih dari satu role untuk setiap Request Mapping.