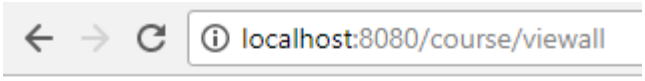
Abdul Raziq Andri
1506757560
Tutorial 8

1. Latihan 1

   Barisan code yang sama ditambahkan ke dalam file view-course-all.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View Course Info</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"></input>
        </form>
        <h1>All Students</h1>
```

   Sehingga ketika diakses, halaman akan menghasilkan tampilan sebagai berikut



Login as user

Sign Out

All Courses

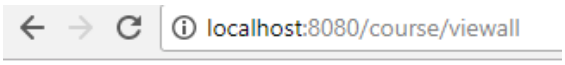
2. Latihan 2

   Tambahkan baris code seperti yang digunakan untuk user dengan username “user” pada class WebSecurityConfig

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

    auth.inMemoryAuthentication()
    .withUser("admin").password("admin")
    .roles("ADMIN");
}
```

   Sehingga terdapat akun dengan username “admin” yang dapat melakukan login



Login as admin

Sign Out

All Courses

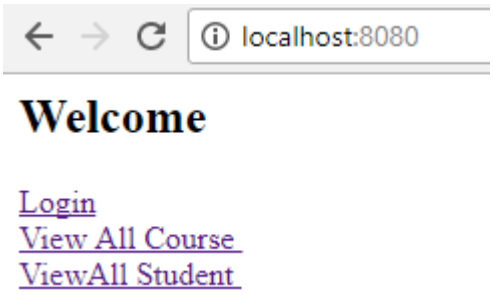
ID = CSC123

Abdul Raziq Andri
1506757560
Tutorial 8

3. Latihan 3
   Tambahkan barisan code sesuai dengan permintaan soal, dimana akun dengan role "ADMIN" hanya diizinkan untuk membuka halaman "/course/viewall" dan "/student/viewall", sedangkan akun dengan role "USER" hanya diizinkan untuk membuka halaman "/student/viewall"

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

   Sehingga ketika akun dengan role "ADMIN" melakukan login, akan ditampilkan halaman sebagai berikut



Abdul Raziq Andri
1506757560
Tutorial 8

4. Hal yang dipelajari
   Dari tutorial 8 ini, saya mengetahui bagaimana penerapan dari security web menggunakan spring. Dimana security yang dimaksud menerapkan:
   a. Otentikasi
      Otentikasi diterapkan melalui fitur login dengan barisan code berikut yang menyimpan informasi pada memory

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

    auth.inMemoryAuthentication()
    .withUser("admin").password("admin")
    .roles("ADMIN");
}
```

      Ataupun menggunakan barisan code berikut yang mengambil data dari database

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

   b. Otorisasi
      Otorisasi diterapkan melalui barisan code berikut yang hanya mengizinkan user untuk mengakses halaman tertentu saja

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