

2024

PBW

Modul 9 – Aspect-Oriented Programming



Modul Aspect-Oriented Programming

Pada modul ini akan dilakukan proses otorisasi menggunakan *aspect-oriented programming* (AOP) dengan Spring AspectJ:

Persiapan

1. Buatlah salinan hasil pengerjaan Modul 08 yang Anda buat sebelumnya.
2. Jadikan Modul 08 sebagai *template* untuk Modul 09 yang Anda akan kerjakan hari ini.
3. Pastikan Anda telah menambah *dependency* untuk AOP:

```
implementation("org.springframework.boot:spring-boot-starter-aop")
```

4. Pastikan Anda telah menambahkan anotasi berikut pada aplikasi Anda:

```
@EnableAspectJAutoProxy
```

Anotasi

Langkah pertama, buatlah *interface* anotasi **RequiredRole** sebagai anotasi yang akan digunakan sebagai target dari *pointcut* yang akan digunakan:

```
package com.example.m08;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

@Target(ElementType.METHOD)
@Retention(RetentionPolicy.RUNTIME)
public @interface RequiredRole {
    String[] value();
}
```

Aspek

Selanjutnya buatlah sebuah kelas aspek bernama **AuthorizationAspect**, aspek ini harus dapat mengambil atribut “username” dan atribut “role” dari *session*. Terdapat tiga hal yang perlu dilakukan oleh aspek ini:

1. Memeriksa apakah terdapat atribut *session* dengan nama “username”, jika tidak ada, berarti pengguna belum login, maka lakukan *throw* sebuah *exception*.
2. Memeriksa apakah terdapat *role* dengan nama “*” yang berarti *role* apapun akan diperbolehkan.
3. Jika tidak terdapat “*”, maka periksalah apakah *role* dari pengguna ada pada *array* yang berasal anotasi **RequiredRole**.

LoginController

Terdapat beberapa perubahan yang perlu dilakukan pada **LoginController**:

1. Pada modul sebelumnya proses *login* hanya menyimpan “username” pada *session*. Ubahlah agar proses *login* juga menyimpan “role” dari pengguna.

2. Buatlah *method handler* untuk URL `"/dashboard"` menjadi sebagai berikut, agar URL ini dapat diakses oleh *role* apapun asalkan pengguna telah melakukan *login*:

```
@GetMapping("/dashboard")
@RequiredRole("*")
public String index(HttpSession session, Model model) {
    return "dashboard";
}
```

HomeController

Buatlah sebuah *controller* baru dengan nama HomeController, terdapat dua hal yang perlu Anda buat pada kelas ini, yaitu:

1. Sebuah halaman pada URL `"/public"` yang dapat diakses baik oleh *role* `"user"` maupun `"admin"`.
2. Sebuah halaman pada URL `"/admin"` yang hanya dapat diakses oleh *role* `"admin"`.

Seluruh pekerjaan disatukan pada sebuah zip dan dikumpulkan dengan nama M09xyyy.zip. Pastikan ukuran file yang diunggah cukup dengan membersihkan folder build.

= SELAMAT MENGERJAKAN =