

Pemrograman Berorientasi Aspek

Tracing with AOP, Anonymus Advice & Returning

Ramos Somya



- Suatu aspek digunakan untuk melakukan trace kejadian tertentu.
- Misal untuk proses debugging → memeriksa apakah program sudah berjalan dengan benar.
- Suatu class dan method dapat ditelusuri atau dilakukan tracing kapan diakses dan berapa kali dieksekusi.

Contoh

Buat Class sebagai berikut:

- **Point → Java Class**
- **Line → Java Class**
- **mainLine → Java Class (Main)**
- **SetsInRotateCounting → Aspect**

Class Point

- Pada class Point tambahkan 2 buah atribut int x dan int y dengan modifier private.
- Buatlah setter dan getter untuk kedua atribut tersebut.
- Tambahkan method toString dengan kembalian berupa String yang akan me-return-kan nilai x dan y.

Class Line

- Tambahkan atribut Point p1 dan Point p2 dengan modifier private.
- Buat setter dan getter untuk kedua atribut tersebut.
- Tambahkan method:

```
public void rotate(double d)
{
    p1.setX(p1.getX() + 10);
    p1.setY(p1.getY() + 10);
    p2.setX(p2.getX() + 10);
    p2.setY(p2.getY() + 10);
}
```

mainLine

- Tambahkan method main dan tambahkan kode program berikut ini:

```
Point p1 = new Point();  
Point p2 = new Point();  
Line l = new Line();
```

```
l.setP1(p1);  
l.setP2(p2);
```

```
l.rotate(10);  
l.rotate(10);
```

```
System.out.println("p1= " + p1 + "\np2= "+p2);
```

SetsInRotateCounting

```
public aspect SetsInRotateCounting {
    int rotatecount = 0;
    int setCount = 0;

    before(): call(void Line.rotate(double))
    {
        rotatecount++;
        System.out.println("Rotate Count : " + rotatecount);
    }

    before(): call (void Point.set*(int))
    {
        setCount++;
        System.out.println("setCount : " + setCount);
    }
}
```

Penjelasan

- Program tersebut akan melakukan tracing pada method rotate → akan dilacak berapa kali method ini dieksekusi/dipanggil.
- Method set (setter) juga dilakukan tracing berapa kali setter ini digunakan untuk mengeset nilai x dan y.

Anonymus Advice

- Suatu advice dapat dibuat tanpa mendeklarasikan terlebih dahulu pointcut-nya.
- Jadi langsung saja dibuat advice-nya.

Contoh

- **Class Matematika**
- **Class Test**
- **Anonymus → Aspect**



Class Matematika:

- **Buatlah sebuah method jumlah yang berisi penjumlahan dua buah bilangan float. Tipe kembalian method ini adalah float. Parameter angka1 dan angka2.**

Class Test:

- **Buat method main dan panggil method jumlah.**

Class AnonymusAdvice

```
public aspect AnonymusAdvice {
    pointcut cobaReturn(): call(float jumlah(float,float));
    after() returning(float x) : cobaReturn()
    {
        System.out.println("Nilai Penjumlahan : " + x);
    }

    // Anonymus Advice
    after() returning (float x): call(float jumlah(float, float))
    {
        System.out.println("Nilai Penjumlahan : " + x);
    }
}
```

Tugas

- **Buatlah program untuk menghitung nilai faktorial dengan menggunakan rekursi.**
- **Tambahkan aspect untuk melakukan tracing berapa kali rekursi tersebut dilakukan.**
- **Inputan dimasukkan melalui input dialog dan hasilnya (tracing dan hasil faktorial) ditampilkan dalam message dialog.**

Ketentuan

- Dikumpulkan hardcopy yang berisi kode program, penjelasan dan output program pada hari Rabu, 19 Oktober 2011 pukul 14.00 – 15.00 WIB di kantor.
- Copy paste nilai == 0 !!!

See You Next Week