

MODIFIER

PENGERTIAN

- ✗ Modifier merupakan keyword yang diletakkan di depan class, interface, field, maupun metode.
- ✗ modifier yang perlu diketahui yakni,
 - + Access modifier
 - + Static modifier
 - + Final modifier

ACCESS MODIFIER

- ✗ Digunakan untuk mengatur pengaksesan.
- ✗ Fungsi : data dan metode dari suatu class tidak selalu bisa diakses secara bebas.
- ✗ Terdapat empat jenis modifier yaitu :
 - + Public
 - + Protected
 - + (default)
 - + Private
- © Kecuali (default), access modifier harus dituliskan.

ACCESS MODIFIER (2)

Jenis Modifier	Dalam class yang sama	Dalam package yang sama	Dalam subclass	Dalam package lain
public	√	√	√	√
protected	√	√	√	-
(default)	√	√	-	-
private	√	-	-	-

- ✗ Modifier private dan protected hanya dapat diapakai pada elemen class, bukan pada class itu sendiri
- ✗ Dalam konsep pemrograman berorientasi objek, access modifier merupakan penerapan dari konsep enkapsulasi (encapsulation)

ACCESS MODIFIER (3)

- ✗ Prinsip penggunaan access modifier :
 - + Gunakan private untuk menyembunyikan bagian class secara penuh dari class lainnya
 - + Gunakan modifier (default) untuk membuatnya hanya dapat diakses oleh class dalam package yang sama
 - + Gunakan protected untuk membuatnya hanya dapat diakses dalam class dan turunannya (subclass)
 - + Gunakan modifier public untuk membuatnya dapat diakses secara penuh

STATIC MODIFIER

- ✖ Digunakan pada field dan metode, agar dapat menjadi milik class, bukan objek.
- ✖ Dengan demikian, penggunaan field dan metode tersebut tidak harus membuat objeknya terlebih dahulu.
- ✖ Selain itu, nilai field class akan sama semua, tidak bergantung objeknya.

STATIC MODIFIER (2)

- ✗ Semua variable dan metode yang bergantung pada objek, harus didefinisikan sebagai instance variabel dan instance metode.
- ✗ Static variable hanya dapat dibuat pada variable dan metode yang nilainya sama, tidak bergantung pada masing-masing objek.

```
[modifier] static identifier () {  
    // metode body  
}
```

```
[modifier] static identifier;
```

STATIC MODIFIER (3)

- ✖ Static variable sering digunakan untuk menyimpan data bersama dalam satu class.

```
1 public class Laptop {  
2     public static int jumlah = 0;  
3  
4     public Laptop() {  
5         jumlah++;  
6         System.out.println(jumlah);  
7     }  
8 }
```

```
1 public class DemoLaptop {  
2     public static void main(String[] args){  
3         new Laptop();  
4         new Laptop();  
5         new Laptop();  
6         new Laptop();  
7     }  
8 }
```

Output - Modul2 (run)

```
run:  
1  
2  
3  
4  
BUILD SUCCESSFUL (total time
```


STATIC MODIFIER (4)

- ✖ Setiap kali Laptop dibuat, variabel konstruktor akan dijalankan.
- ✖ Sehingga output yang diperoleh, jika ada pembuatan objek Laptop, variable jumlah Laptop akan bertambah dengan satu.

FINAL MODIFIER

- ✗ Digunakan untuk menyatakan bahwa sesuatu sudah final dan tidak dapat diubah lagi.
- ✗ Arti final sendiri berbeda-beda, tergantung apakah digunakan bersama class, metode, atau variable.
 - + Final class, tidak dapat dibuat turunannya (subclass)
 - + Final metode, tidak dapat di-override
 - + Final variable, tidak dapat diubah nilainya

FINAL MODIFIER (2) ~ FINAL CLASS

- ✗ Untuk membuat class yang tidak akan dapat dibuat subclassnya.
- ✗ Semua metode dalam final class otomatis akan bersifat final. Sehingga tidak perlu menuliskan metode final di depan metode tersebut.

```
[modifier] final class identifier {  
    //class body  
}
```

- ✗ Penggunaan final class harus dilakukan dengan tepat.

FINAL MODIFIER (3) ~ FINAL METODE

- ✗ Final metode tidak dapat ditimpa (di-override) oleh subclass di bawahnya.

```
[modifier] final metode identifier (parameter) {  
    //metode body  
}
```

- ✗ Alasan menggunakan final metode : demi kecepatan.
- ✗ Dalam java menemukan metode dengan cara mencarinya dari subclass kemudian naik menuju superclass hingga metode tersebut ditemukan.

FINAL METODE (LANJUTAN)

- ✖ Jika metode dideklarasikan sebagai final, kompiler dapat menggunakan bytecode dari metode tersebut tanpa takut akan di-override.

FINAL MODIFIER (5) ~ FINAL VARIABEL

- ✖ Variable yang dideklarasikan sebagai final tidak dapat diubah nilainya, sehingga variable jenis ini disebut dengan konstanta.
- ✖ Umumnya, final variabel juga dideklarasikan sebagai static.
- ✖ Dengan demikian, nilai pada final variable dapat dipakai tanpa harus membuat objeknya. (cara ini membuat program lebih efisien)

TERIMA KASIH...

- ✖ Silahkan beralih ke praktikum dan kerjakan tugas-tugas yang disediakan...
- ✖ Semangat!!

