

# PEMOGRAMAN JAVA

Yoannita

## ● Modifier



# Class

Dalam pendefinisian *class*, dituliskan :

```
<modifier> class <name> {  
    <attributeDeclaration>*  
    <constructorDeclaration>*  
    <methodDeclaration>*  
}
```

dimana :

<modifier> adalah sebuah *access modifier*, yang dapat dikombinasikan dengan tipe *modifier* lain.

# Modifier

- Access Modifier
  - default
  - public
  - private
  - protected
- Static Modifier
- Final Modifier
- Abstract Modifier
- Synchronized Modifier
- Native Modifier
- Default

# Access Modifier

- **default**

- Hanya kelas dalam paket dapat mengakses variabel dan metode dari kelas.
- Dapat diakses oleh subclass asalkan berada dalam package yang sama

- **public**

- Variabel dan metode dapat diakses dari dalam maupun luar kelas

- **private**

- Atribut kelas hanya dapat diakses oleh metode dalam kelas dimana didefinisikan
- tidak ada kode satupun dari luar class tersebut yang diizinkan mengakses / mengubah nilai dari member tersebut

- **protected**

- Atribut kelas hanya dapat diakses oleh kelas dan subkelas dari kelas tersebut

# Access Modifier

|                                     | <i>private</i> | default/package | <i>protected</i> | <i>public</i> |
|-------------------------------------|----------------|-----------------|------------------|---------------|
| class yang sama                     | Yes            | Yes             | Yes              | Yes           |
| package yang sama                   |                | Yes             | Yes              | Yes           |
| package yang berbeda (subclass)     |                |                 | Yes              | Yes           |
| package yang berbeda (non-subclass) |                |                 |                  | Yes           |

# Static Modifier

- Memberikan spesifikasi bahwa atribut atau metode sama untuk semua objek dari kelas tertentu.
- Keuntungan dari modifier static adalah dapat diakses tanpa membuat objek dalam class.

# Final Modifier

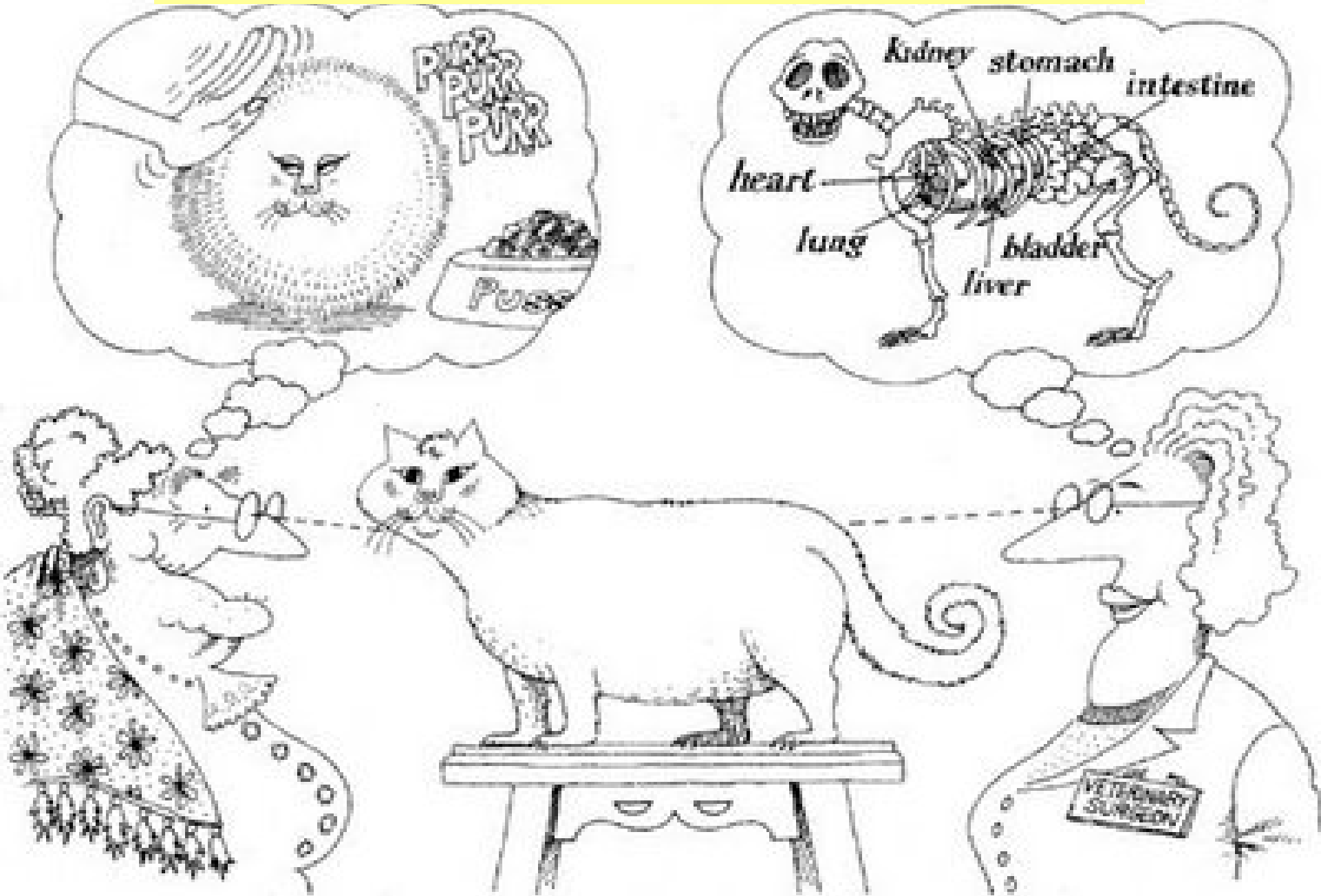
- Memberikan spesifikasi bahwa variabel mempunyai nilai konstan sama atau metode yang tidak dapat di-*overridden* dalam subkelas.
- *final int persenDiskon = 0.1;*

# Abstract Modifier

- Memberikan spesifikasi bahwa metode tidak dapat dijalankan, dan harus digunakan subkelas yang tidak abstrak



**Abstraction focuses on the essential characteristics of some object, relative to the perspective**



# Synchronized Modifier

- Memberikan spesifikasi bahwa metode thread safe, yang berarti eksekusi diperbolehkan pada synchronized method

# Native Modifier

- Mengidentifikasi metode yang mempunyai implementasi dasar
- Modifier native memberikan informasi java kompiler bahwa implementasi metode berada di luar
- Contoh : `native int nilaiTotal();`