

Inheritance & Polymorphism

Teosofi Hidayah Agung
Hafidz Mulia

Departemen Matematika
Institut Teknologi Sepuluh Nopember

19 Mei 2025

Zombie
- HP : int
Zombie() # Zombie(int hp) ~ setHP(int hp) ~ getHP() : int + walk() : void + attack() : void + voice() : void

Creeper
- HP : int
Creeper() # Creeper(int hp) ~ setHP(int hp) ~ getHP() : int + walk() : void + explode() : void + voice() : void

Villager
- profession : String
- HP : int
Villager() # Villager(String job, int hp) + walk() : void + trade() : void + voice() : void

Masalah

Menurut kalian struktur class seperti diatas apakah sudah efisien/sederhana? Jika masih belum, bagaimana cara agar pembuatan class diatas lebih efisien?

Daftar isi

1 Inheritance

- Constructor Child Class

2 Polymorphism

3 Diagram Kelas

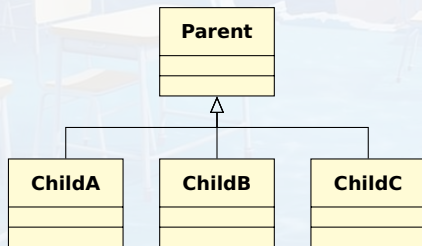
- Hubungan/Relasi

4 Latihan

Inheritance

Definisi

Inheritance adalah sebuah konsep dalam pemrograman berorientasi objek yang memungkinkan sebuah kelas untuk mewarisi atribut dan metode dari kelas lain. Kelas yang mewarisi disebut sebagai subclass atau child class, sedangkan kelas yang diwarisi disebut sebagai superclass atau parent class.



Gambar: Struktur inheritance

Inheritance

Manfaat

- Mengurangi duplikasi kode
- Memudahkan pemeliharaan kode
- Memungkinkan penggunaan polymorphism

Catatan

Sama seperti anak yang hanya memiliki satu ibu kandung, subclass hanya dapat mewarisi dari satu superclass. Disisi lain superclass dapat memiliki banyak subclass.

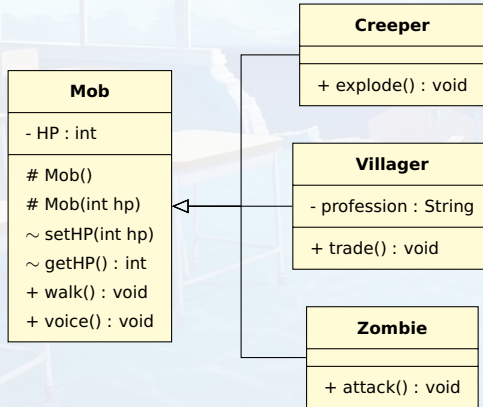
Inheritance

Kode: Struktur syntax inheritance

```
1 class Parent {
2     void display() {
3         System.out.println("This is the Parent class");
4     }
5 }
6
7 class Child extends Parent {
8     void display() {
9         System.out.println("This is the Child class");
10    }
11 }
```

Inheritance

Gambar: Efisiensi struktur class



Inheritance

Constructor Child Class

Constructor Child Class

- Constructor child class akan memanggil constructor parent class
- Jika constructor parent class tidak ada, maka constructor child class tidak dapat diakses

Biasanya constructor parent class dipanggil dengan menggunakan keyword `super()`. Method ini harus dipanggil ketika parent class punya konstruktor dengan parameter atau ketika mau memanggil method atau atribut dari parent class yang di-override di child class.

Inheritance

Constructor Child Class

Kode: Contoh constructor child class

```
1 class Parent {
2     int x;
3     Parent(int x) {
4         this.x = x;
5     }
6 }
7 class Child extends Parent {
8     int y;
9     Child(int x, int y) {
10         super(x); // Memanggil constructor Parent
11         this.y = y;
12     }
13 }
```

Daftar isi

1 Inheritance

- Constructor Child Class

2 Polymorphism

3 Diagram Kelas

- Hubungan/Relasi

4 Latihan

Polymorphism

Definisi

Polymorphism adalah kemampuan suatu objek untuk memiliki banyak bentuk. Dalam konteks pemrograman berorientasi objek, polymorphism memungkinkan kita untuk menggunakan metode yang sama pada objek yang berbeda, meskipun implementasinya berbeda.

Intinya sebuah objek induk bisa memanggil method yang ada pada kelas anaknya. Misalkan kita punya class induk Hewan dan class anak Sapi. Kita bisa memanggil method yang ada di class Sapi dengan menggunakan objek dari class Hewan. Contohnya seperti berikut:

Polymorphism

Kode: Contoh polymorphism

```
1 class Parent {
2     void method1() {
3         System.out.println("Parent method");
4     }
5 }
6 class Child extends Parent {
7     void method1() {
8         System.out.println("Child method");
9     }
10 }
11 class Main {
12     public static void main(String[] args) {
13         Parent obj = new Child(); // Polymorphism
14         obj.method1(); // Error: methodChild() not found
15     }
16 }
```

Polymorphism

Sekarang coba ingat kembali tentang materi Logika Matematika. Misalkan aku mempunyai kalimat kebenaran:

"Semua sapi adalah Hewan"

$$\forall x(S(x) \implies H(x))$$

Sekarang ketika aku bilang "Semua sapi itu pasti hewan", apakah itu benar?

Polymorphism

Sekarang coba ingat kembali tentang materi Logika Matematika. Misalkan aku mempunyai kalimat kebenaran:

"Semua sapi adalah Hewan"

$$\forall x(S(x) \implies H(x))$$

Sekarang ketika aku bilang "Semua sapi itu pasti hewan", apakah itu benar? **Ya, jelas benar.** Sekarang coba kita ubah kalimatnya menjadi "Semua hewan adalah sapi". Apakah itu benar?

Polymorphism

Sekarang coba ingat kembali tentang materi Logika Matematika. Misalkan aku mempunyai kalimat kebenaran:

"Semua sapi adalah Hewan"

$$\forall x(S(x) \implies H(x))$$

Sekarang ketika aku bilang "Semua sapi itu pasti hewan", apakah itu benar? **Ya, jelas benar**. Sekarang coba kita ubah kalimatnya menjadi "Semua hewan adalah sapi". Apakah itu benar? **Tidak, jelas itu kurang tepat**.

Polymorphism

Contoh

Misal kita punya class Hewan dan class anaknya adalah Sapi, Kucing, dan Ayam.

Kode: Aplikasi polymorphism

```
1  Hewan[] hewan2 = { new Sapi(), new Kucing(), new Ayam() };
2
3  for (Hewan h : hewan2) {
4      h.bersuara();
5  }
```


Polymorphism

Contoh

Atau misal dalam *minecraft* dimana kita dapat mengatur difficulty dari game tersebut.

Kode: Aplikasi dalam game

```
1  Mob[] mobs = {new Villager(),new Zombie(),new Creeper()};
2
3  for (Mob m : mobs) {
4      if (difficulty.equals("peaceful") && (m instanceof Zombie || m
5          instanceof Creeper)) {
6          continue;
7      } else {
8          m.spawn();
9      }
10 }
```

Daftar isi

1 Inheritance

- Constructor Child Class

2 Polymorphism

3 Diagram Kelas

- Hubungan/Relasi

4 Latihan

Diagram Kelas

Hubungan/Relasi

Hubungan/Relasi

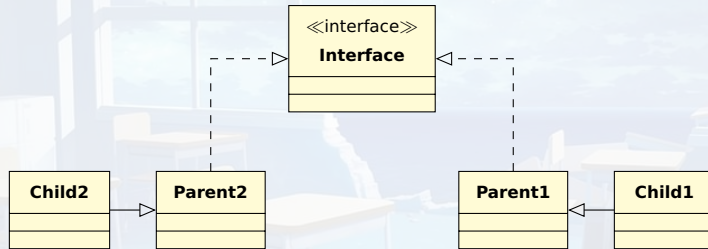
- Pewarisan : hubungan antara dua class di mana satu class mewarisi atribut dan metode dari class lain.
- Implementasi : hubungan antara class dan interface di mana class mengimplementasikan method yang didefinisikan dalam interface.

Catatan

Sebenarnya cukup banyak relasi yang ada dalam diagram kelas, tetapi kita hanya akan membahas dua relasi ini saja.

Diagram Kelas

Hubungan/Relasi



Gambar: Hubungan/Relasi

Contoh

- Pewarisan : class Child mewarisi atribut dan metode dari class Parent.
- Implementasi : class Child mengimplementasikan metode yang didefinisikan dalam interface Interface.

Daftar isi

1 Inheritance

- Constructor Child Class

2 Polymorphism

3 Diagram Kelas

- Hubungan/Relasi

4 Latihan

Latihan 1

Buatlah class dalam *Java* yang merepresentasikan diagram kelas berikut:

