

KUSUMA WEB ACADEMY

DASAR-DASAR JALUR JAVA

Modul 12: Pilar OOP 2 - Pewarisan (Inheritance) & Polimorfisme

Hak Cipta © 2026 Kusuma Web. All Rights Reserved.

Konsep Pewarisan (Inheritance - Keyword extends)

Definisi Pewarisan

- ▶ Proses penurunan semua properti/atribut publik & dilindungi serta method milik kelas induk (**Superclass**) ke kelas anak baru (**Subclass**).
- ▶ **Keyword:** Menggunakan keyword **extends**.
- ▶ *Keuntungan:* Mengeliminasi penulisan ulang fungsionalitas umum (*code reuse*).

Implementasi Pewarisan

```
// Superclass (Induk)
public class Binatang {
    public void makan() {
        System.out.println("Makan...");
    }
}

// Subclass (Anak)
public class Kucing extends Binatang {
    public void meong() {
        System.out.println("Meong!");
    }
}
```

Penggunaan Keyword `super` di Java

Fungsi Keyword `super`

- ▶ Merujuk secara langsung pada atribut, method, atau konstruktor milik superclass.
- ▶ **Aplikasi Umum:**
 - ▶ Memanggil konstruktor superclass dari dalam subclass: `super(param);`
 - ▶ Memanggil method induk yang telah ditindih (**overridden**) oleh anak kelas.

Implementasi `super`

```
public class Hewan {
    String nama;
    Hewan(String nama) {
        this.nama = nama;
    }
}

public class Anjing extends Hewan {
    String ras;
    Anjing(String nama, String ras) {
        super(nama); // Konstruktor
                    // Induk wajib
        this.ras = ras;
    }
}
```

Pilar OOP 3: Polimorfisme Dinamis (Method Overriding)

Konsep Polimorfisme

- ▶ Karakteristik di mana satu rujukan tipe superclass dapat menampung objek subclass yang berbeda-beda.
- ▶ **Method Overriding:** Anak kelas menulis ulang secara menyeluruh method milik induknya dengan anotasi `@Override`.

Demo Overriding & Polimorfisme

```
class Cetak {  
    void print() { System.out.println("Asal"); }  
}  
class CetakKertas extends Cetak {  
    @Override  
    void print() { System.out.println("Kertas"); }  
}  
  
// Implementasi Polimorfisme Dinamis  
Cetak obj = new CetakKertas();  
obj.print(); // Output: Kertas (Dinamis runtime)
```

1. Single Inheritance

- ▶ Java tidak mendukung *Multiple Inheritance* kelas. Sebuah kelas anak **hanya boleh** mengekstensi tepat satu kelas induk.
- ▶ Solusi integrasi banyak perilaku diakomodasi melalui **Interface**.

2. Penggunaan Keyword `final`

- ▶ Jika sebuah class bertanda `final` class, maka ia tidak dapat diwariskan sama sekali.
- ▶ Jika method bertanda `final` void, method tersebut dilarang keras untuk dioverride oleh subclass mana pun.