

## Pertemuan 8: **POLIMORFISMA**

PRAKTIKUM ALGORITMA DAN  
PEMROGRAMAN KOMPUTER 2

**Mohammad Febryan Khamim** (5002221072)  
**Putri Ghaida Tsurayya** (5002221053)



Genap **2024/2025**

# OUTLINE PEMBAHASAN



Berikut ini adalah **outline** pembahasan pada pertemuan hari ini:

**01**

**POLIMORFISMA**

**02**

**MARI MENCoba**

**03**

**PENUGASAN**



# POLIMORFISMA

## DEFINISI

**Polimorfisma** adalah suatu konsep yang memungkinkan entitas (method, operator, atau objek) untuk memiliki perbedaan perilaku atau operasi berdasarkan spesifikasi yang dibutuhkan pada tiap-tiap entitas.

**Nama sama → Perilaku berbeda**

## Hubungan dengan Inheritance

**Inheritance** (mewariskan atribut atau method dari kelas induk ke kelas anak) → **Polimorfisme** (method dan atribut tersebut diberikan perintah untuk menampilkan perilaku berbeda).

## Contoh Polimorfisma

```
class Animal {  
    public void animalSound() {  
        System.out.println("The animal makes a sound");  
    }  
}
```

Kelas induk mempunyai method 'animalSound', lalu pada masing-masing kelas anak, method tersebut dispesifikasi.

```
class Pig extends Animal {  
    public void animalSound() {  
        System.out.println("The pig says: wee wee");  
    }  
}
```

```
class Dog extends Animal {  
    public void animalSound() {  
        System.out.println("The dog says: bow wow");  
    }  
}
```

## Fitur-Fitur Kunci Polimorfisma

- **Multiple Behaviors.** Method yang sama memiliki perilaku berbeda sesuai objek yang memanggil method ini.
- **Method overriding.** Kelas anak dapat mendefinisikan ulang method dari kelas induk
- **Method overloading** Mendefinisikan beberapa method dengan nama sama, tetapi parameter berbeda.

## Kenapa harus Polimorfisma

- **Code reusability.** Polimorfisma memungkinkan method atau class yang sama dapat digunakan kembali oleh beberapa objek
- **Flexibility.** Memungkinkan objek dari kelas yang berbeda menambahkan fitur atau subclass baru tanpa mengubah kode lama.
- **Abstraction.** Memungkinkan penggunaan Class abstract atau interface untuk mendeklarasikan perilaku umum.



## Abstract Class

- **Abstract class** adalah kelas yang digunakan sebagai kerangka dasar (*template*) untuk kelas-kelas turunannya. Kelas ini berisi **method abstract** (tanpa isi) maupun **method biasa**.
- Tidak memungkinkan membuat objek dari **abstract class** → harus mendefinisikan / mengisi **method abstract** dulu pada kelas anak.
- Semua **Class** yang memiliki Abstract method merupakan Abstract Class

## Abstract Class

```
abstract class Animal {  
    abstract void makeSound(); // metode abstrak  
  
    void sleep() {  
        System.out.println("Sleeping...");  
    }  
}  
  
class Dog extends Animal {  
    void makeSound() {  
        System.out.println("Bark");  
    }  
}
```



## MARI MENCoba



## PENUGASAN

Penugasan **pertemuan** ini dapat diakses melalui pranala berikut:

[Intip.in/Pertemuan8Alprokom2](http://Intip.in/Pertemuan8Alprokom2)



# TERIMA KASIH