

KUSUMA WEB ACADEMY

# DASAR-DASAR JALUR JAVA

Modul 13: Pilar OOP 4 - Abstraksi (Abstract Class & Interface)

Hak Cipta © 2026 Kusuma Web. All Rights Reserved.

# Konsep Abstraksi: Fokus pada Apa, Bukan Bagaimana

## Filosofi Abstraksi (Abstraction)

- ▶ Menyembunyikan kompleksitas detail implementasi sistem yang rumit di belakang antarmuka sederhana.
- ▶ Fokus penuh pada perilaku publik yang ditawarkan objek (\*what it does\*), bukan cara detail ia melaksanakannya (\*how it does it\*).

## Analogi Dunia Nyata

Saat mengendarai mobil, Anda hanya perlu menginjak pedal gas untuk berakselerasi. Anda tidak perlu memikirkan pembakaran silinder mesin di bawah kap mobil.

## Implementasi di Java

- ▶ Java memfasilitasi abstraksi melalui dua mekanisme utama:
  1. **Abstract Class** (Menawarkan abstraksi parsial).
  2. **Interface** (Menawarkan abstraksi penuh/total).

# Abstract Class: Ciri & Struktur Kode

## Ciri-Ciri Abstract Class

- ▶ Ditandai dengan keyword **abstract**.
- ▶ **Tidak bisa diinstansiasi langsung** menjadi objek (new).
- ▶ Dapat memiliki method biasa (dengan body) maupun **abstract method** (tanpa body/kosong).
- ▶ Wajib diturunkan ke subclass konkret yang bertugas mengimplementasikan semua abstract method-nya.

```
// Kelas abstrak induk
abstract class Komputer {
    // Abstract method (Tanpa body {})
    abstract void booting();

    // Regular method
    void matikan() {
        System.out.println("Shutting
            down...");
    }
}
```

# Interface: Kontrak Perilaku Sistem Sekaligus

## Konsep Interface

- ▶ Sebuah spesifikasi atau cetak biru murni yang hanya berisi abstract method publik dan konstanta statis (public static final).
- ▶ Implementasi interface ke kelas menggunakan keyword **implements**.
- ▶ Mendukung *multiple interface implementation* di satu kelas.

```
interface Koneksi {  
    void sambungInternet(); // public  
    abstract  
}  
  
// Mengimplementasikan interface  
class LaptopAsus implements Koneksi {  
    @Override  
    public void sambungInternet() {  
        System.out.println("Wifi  
            Terhubung!");  
    }  
}
```

# Perbandingan Utama: Abstract Class vs Interface

Sangat penting bagi arsitek software untuk memahami batasan fundamental berikut:

<b>Fitur Pembeda</b>	<b>Abstract Class</b>	<b>Interface</b>
<b>Abstraksi</b>	Parsial (0 s/d 100%)	Total (100% murni)
<b>Pewarisan</b>	Hanya boleh extends satu kelas induk	Boleh implements banyak inter
<b>Variabel Kelas</b>	Bebas memiliki variabel instansi	Hanya boleh konstanta (publ
<b>Default Method</b>	Tidak ada keyword khusus	Didukung sejak Java 8 lewat ke
<b>Konstruktur</b>	Bisa memiliki konstruktur	Tidak boleh memiliki konstruk