

Bab 7

Abstract Class dan Interface

7.1 Abstract Class dan Abstraksi

Sub-CPMK yang Dicakup dalam Bab Ini:

- Sub-CPMK 3.2: Menerapkan polymorphism dalam desain aplikasi

Konsep Penting

Abstraksi melalui abstract class dan interface adalah mekanisme utama untuk mencapai desain yang fleksibel dalam Java [1]. Abstract class tidak dapat diinstantiasi dan dapat memiliki method abstrak serta konkret.

7.1.1 Karakteristik Abstract Class

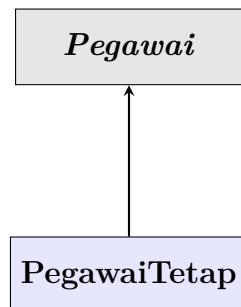
- Dideklarasikan dengan keyword **abstract**
- Dapat memiliki atribut dan constructor
- Dapat memiliki method abstract dan non-abstract

```
1 abstract class Pegawai {  
2     protected String nama;  
3  
4     public Pegawai(String nama) {  
5         this.nama = nama;  
6     }  
7  
8     public abstract double hitungGaji();  
9  
10    public void tampilkanInfo() {
```

```

11         System.out.println("Nama: " + nama);
12     }
13 }
14
15 class PegawaiTetap extends Pegawai {
16     private double gajiBulanan;
17
18     public PegawaiTetap(String nama, double gajiBulanan) {
19         super(nama);
20         this.gajiBulanan = gajiBulanan;
21     }
22
23     @Override
24     public double hitungGaji() {
25         return gajiBulanan;
26     }
27 }
```

Kode Program 7.1: Contoh Abstract Class



Gambar 7.1: Hierarki Class Pegawai (Abstract)

7.1.2 Kapan Menggunakan Abstract Class

Gunakan abstract class ketika:

- Ada perilaku umum yang ingin diwariskan
- Ada method yang wajib diimplementasikan subclass
- Dibutuhkan constructor atau state bersama

7.2 Interface dan Perbandingannya dengan Abstract Class

7.2.1 Konsep Interface

Interface mendefinisikan kontrak yang harus diimplementasikan oleh class.

```

1  interface Pembayaran {
2      void proses();
3      void verifikasi();
4  }
5
6  class PembayaranKartu implements Pembayaran {
7      @Override
8      public void proses() {
9          System.out.println("Proses kartu kredit");
10     }
11
12     @Override
13     public void verifikasi() {
14         System.out.println("Verifikasi kartu kredit");
15     }
16 }
```

Kode Program 7.2: Contoh Interface

7.2.2 Perbandingan Abstract Class vs Interface

Aspek	Abstract Class	Interface
Instansiasi	Tidak dapat diinstansiasi	Tidak dapat diinstansiasi
Inheritance	Single inheritance	Multiple inheritance of type
State	Dapat memiliki attribute	Hanya constant (public static final)
Method	Abstract dan konkret	Abstract, default, static

Tabel 7.1: Perbandingan Abstract Class dan Interface

Aktivitas Pembelajaran

- Diskusi:** Tentukan kapan menggunakan abstract class dan kapan interface.
- Implementasi:** Buat interface **Notifikasi**. Lalu buat class **EmailNotifikasi** dan **SMSNotifikasi**.
- Studi Kasus:** Rancang abstract class **Bentuk** dan implementasi **Lingkaran**, **Segitiga**.
- Review:** Analisis kode yang menggunakan inheritance berlebihan dan usulkan perbaikan dengan interface.

Latihan dan Refleksi

1. Jelaskan perbedaan utama antara abstract class dan interface.
2. Buat abstract class **Transportasi** dan subclass **Kereta**, **Bus**.
3. Buat interface **Diskon** dan implementasikan pada class **Produk**.
4. Jelaskan manfaat default method pada interface.
5. **Refleksi:** Bagaimana Anda menentukan pilihan desain antara abstract class dan interface?

Asesmen (Evaluasi Kinerja)

Instrumen Penilaian untuk Sub-CPMK 3.2

A. Pilihan Ganda

1. Interface di Java dapat diimplementasikan oleh:
 - (a) Satu class saja
 - (b) Banyak class
 - (c) Hanya abstract class
 - (d) Hanya class final
2. Abstract class dapat memiliki:
 - (a) Hanya method abstract
 - (b) Hanya method static
 - (c) Method abstract dan method konkret
 - (d) Hanya constructor private

B. Tugas Praktik

- Rancang interface **Layanan** dan dua implementasi berbeda. Tunjukkan penggunaan polymorphism.

Rubrik Penilaian: Lihat Lampiran A

Checklist Pencapaian Kompetensi

Centang item berikut setelah Anda yakin telah menguasainya:

- Saya memahami konsep abstract class
- Saya memahami konsep interface

- Saya dapat memilih antara abstract class dan interface
- Saya dapat mengimplementasikan interface di Java
- Saya dapat menggunakan polymorphism dengan abstract class dan interface

Rangkuman

Abstract class menyediakan dasar perilaku dan state bersama, sedangkan interface menyediakan kontrak untuk perilaku [1]. Keduanya mendukung polymorphism dan desain yang fleksibel.

Kata Kunci: *Abstract Class, Interface, implements, default method, polymorphism*

Daftar Pustaka

- [1] Cay S. Horstmann. *Core Java Volume I-Fundamentals*. Prentice Hall, 11th edition, 2019.