

Bab 5

Inheritance dan Hierarki Kelas

5.1 Konsep Inheritance dan Relasi "is-a"

Sub-CPMK yang Dicakup dalam Bab Ini:

- Sub-CPMK 3.1: Mengimplementasikan inheritance dan method overriding

5.1.1 Konsep Pewarisan

Konsep Penting

Inheritance memungkinkan kita untuk membuat class baru berdasarkan class yang sudah ada, mewarisi semua attributes dan methods-nya [1]. Relasi ini dikenal sebagai relasi *is-a*.

Keuntungan utama inheritance:

- **Code reuse**: mengurangi duplikasi kode
- **Extensibility**: menambah fitur dengan subclass
- **Polymorphism**: memungkinkan penggunaan reference superclass

5.1.2 Keyword `extends`

Dalam Java, inheritance dideklarasikan dengan keyword `extends`.

```
1 class Kendaraan {  
2     protected String merk;  
3 }
```

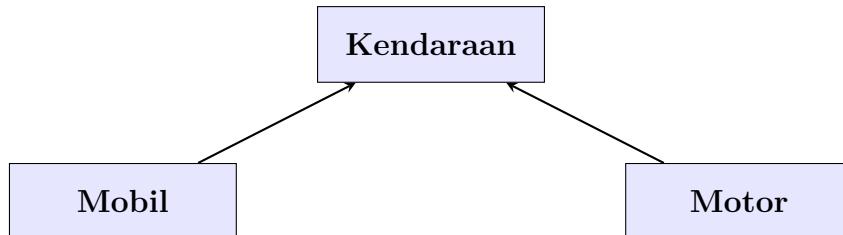
```

4     public Kendaraan(String merk) {
5         this.merk = merk;
6     }
7
8     public void nyalakan() {
9         System.out.println("Kendaraan dinyalakan");
10    }
11 }
12
13 class Mobil extends Kendaraan {
14     private int jumlahPintu;
15
16     public Mobil(String merk, int jumlahPintu) {
17         super(merk);
18         this.jumlahPintu = jumlahPintu;
19     }
20 }
```

Kode Program 5.1: Inheritance Sederhana

5.1.3 Hierarki Kelas

Hierarki kelas membantu kita mengorganisasi konsep dalam struktur bertingkat:



Gambar 5.1: Contoh Hierarki Kelas (Relasi is-a)

5.2 Overriding, `super`, dan Desain Hierarki

5.2.1 Method Overriding

Subclass dapat mendefinisikan ulang method dari superclass. Gunakan anotasi `@Override` untuk memperjelas niat.

```

1 class Kendaraan {
2     public void nyalakan() {
3         System.out.println("Kendaraan dinyalakan");
4     }
5 }
6
7 class Mobil extends Kendaraan {
8     @Override
```

```

9     public void nyalakan() {
10        System.out.println("Mobil dinyalakan dengan kunci");
11    }
12 }
```

Kode Program 5.2: Contoh Method Overriding

5.2.2 Penggunaan super

`super` digunakan untuk memanggil constructor atau method dari superclass.

```

1 class Dosen {
2     protected String nama;
3
4     public Dosen(String nama) {
5         this.nama = nama;
6     }
7 }
8
9 class DosenTetap extends Dosen {
10    private String nidn;
11
12    public DosenTetap(String nama, String nidn) {
13        super(nama);
14        this.nidn = nidn;
15    }
16 }
```

Kode Program 5.3: Penggunaan super

5.2.3 Composition vs Inheritance

Gunakan inheritance jika relasi *is-a* kuat. Jika tidak, gunakan composition (*has-a*) untuk fleksibilitas.

Catatan

Contoh: **Mobil** *has-a* **Mesin**. Lebih tepat menggunakan composition.

Aktivitas Pembelajaran

- Hierarki Kelas:** Buat hierarki class untuk sistem akademik (Person → Mahasiswa/Dosen).
- Overriding:** Implementasikan method `hitungGaji()` berbeda di class **Pegawai** dan **Manajer**.

3. **Diskusi:** Identifikasi contoh *is-a* dan *has-a* dalam aplikasi e-commerce.
4. **Code Review:** Cari potensi penyalahgunaan inheritance pada kode yang diberikan dosen.

Latihan dan Refleksi

1. Jelaskan perbedaan inheritance dan composition.
2. Buat class **Hewan** dan subclass **Kucing**. Tambahkan method **suara()** dan override di subclass.
3. Kapan penggunaan inheritance dapat menyebabkan masalah desain? Berikan contoh.
4. Buat class **Pegawai** dan subclass **PegawaiKontrak**. Tambahkan atribut dan method spesifik.
5. **Refleksi:** Apa tantangan terbesar Anda saat menyusun hierarki kelas?

Asesmen (Evaluasi Kinerja)

Instrumen Penilaian untuk Sub-CPMK 3.1

A. Pilihan Ganda

1. Keyword untuk melakukan inheritance di Java adalah:
 - (a) inherit
 - (b) extends
 - (c) implements
 - (d) override
2. Method overriding berarti:
 - (a) Dua method dengan nama sama dalam class yang sama
 - (b) Subclass mendefinisikan ulang method superclass
 - (c) Method dipanggil berulang
 - (d) Method private diakses dari luar class

B. Tugas Praktik

- Buat hierarki class **Produk** → **ProdukDigital** dan **ProdukFisik** dengan atribut dan method yang sesuai.

- Demonstrasikan overriding pada method `hitungHargaAkhir()`.

Rubrik Penilaian: Lihat Lampiran A

Checklist Pencapaian Kompetensi

Centang item berikut setelah Anda yakin telah menguasainya:

- Saya memahami konsep inheritance dan relasi *is-a*
- Saya dapat membuat subclass dengan **extends**
- Saya mampu menerapkan method overriding
- Saya memahami penggunaan **super**
- Saya dapat memilih inheritance atau composition dengan tepat

Rangkuman

Inheritance memungkinkan class baru mewarisi perilaku dan data dari class lain, mendukung code reuse dan desain hierarki [1]. Overriding dan **super** adalah kunci untuk menyesuaikan perilaku subclass.

Kata Kunci: *Inheritance, extends, overriding, super, is-a, composition*

Daftar Pustaka

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