

## **Tugas Pendahuluan: Modul 7**

**NIM: 105222033**

**Nama:**

**Mohamad Dandung Sadat**

**Instruksi:** Jawablah pertanyaan-pertanyaan berikut dengan jelas dan rinci. Gunakan contoh kode jika diperlukan untuk memperjelas jawaban Anda. Deadline, Kamis 08 Mei 2025, pukul 13:00 (sebelum praktikum). Link pengumpulan: <https://forms.gle/AWxTPKBmXt2BhuCD9>

### **Soal**

1. Jelaskan dengan kata-kata Anda sendiri apa yang dimaksud dengan konsep inheritance (pewarisan) dalam pemrograman berorientasi objek (OOP) di Java. Mengapa konsep ini penting dalam pengembangan perangkat lunak?

**Jawab:**

Dalam pemrograman berorientasi objek (OOP) di Java, pewarisan (inheritance) adalah suatu hal dimana sub-class (child class) mewarisi atribut (variabel) dan metode (fungsi) dari super-class (parent class) yang lebih tinggi dari dirinya. Artinya, subclass dapat menggunakan kode yang telah dituliskan dalam superclass tanpa menuliskannya lagi.

pentingnya pengembangan perangkat lunak menggunakan inheritance

Reusabilitas: Memudahkan repurposing. Kode yang telah ditulis dalam superclass akan langsung diterapkan di subclass, sehingga tidak ada duplikasi kode.

Perawatan lebih simple: Superclass dapat diteruskan ke subclass, sehingga pewarisan kelas ini mempermudah bekerja dengan banyak subclass.

Ekstensi fungsionalitas: Lain dari itu, subclass dapat menambah atau mengubah perilaku super-class (override) untuk menambah pengembangan yang lebih fleksible.

Polimorfisme: Inheritance membuka kemungkinan polimorfisme yang merupakan kapasitas untuk menggunakan satu antarmuka dengan berbagai macam objek, sehingga mendukung pemanggilan polimorfik melalui superclass dan subclass.

2. Sebutkan dan jelaskan minimal 3 manfaat utama dari penggunaan inheritance dalam Java. Berikan masing-masing analogi nyata yang relevan.

**Jawab:**

**1. Reusabilitas Kode (Code Reusability)**

**Penjelasan:**

Subclass dapat menggunakan ulang kode dari superclass tanpa menuliskannya ulang. Reusabilitas kode dapat meningkatkan produktivitas karena mengurangi duplikasi.

**Analogi:**

Anggaplah ada pabrik mobil yang memiliki desain dasar untuk sebuah kendaraan, contoh sasis, roda serta mesin. Jika mau membuat sedan, SUV atau bahkan truk baru, mereka tinggal menggunakan sasis kustom tambahan karena mereka sudah mendapat desain master yang terbuka.

**2. Pemeliharaan Lebih Mudah (Ease of Maintenance)**

**Penjelasan:**

Jika ada perubahan pada fitur dasar yang ada pada superclass, perubahan tersebut secara otomatis akan merambat pada semua subclass yang menggunakannya. Sistem pemeliharaan dan sistem upgrade menjadi lebih efisien.

**Analogi:**

Mengganti lampu utama di sebuah gedung bertingkat, lampu tersebut berfungsi untuk semua ruangan. Saat perbaikan sistem listrik pusat, semua ruangan tidak perlu dibongkar satu per satu.

**3. Ekstensi dan Penyesuaian (Extensibility & Customization)**

**Penjelasan:**

Subclass turun dari superclass dengan menambahkan aksesori pada alat di superclass.

**Analogi:**

Seorang koki memiliki resep dasar untuk membuat sup. Murid-muridnya (subclass) dapat menggunakan resep dasar tersebut, tetapi bisa menambahkan bahan sesuai selera—misalnya menambah ayam, udang, atau rempah-rempah lokal—tanpa mengubah resep dasar secara keseluruhan.

3. Apa perbedaan antara superclass dan subclass? Jelaskan juga apa yang dimaksud dengan keyword extends dalam konteks pewarisan.

**Jawab:**

**Superclass** dan subclass adalah dua konsep penting dalam pewarisan (inheritance) di pemrograman berorientasi objek. Superclass adalah kelas induk yang berisi atribut dan metode dasar yang dapat digunakan oleh kelas lain. Kelas ini mendefinisikan perilaku umum yang bisa diwariskan, tetapi tidak mewarisi dari kelas manapun secara eksplisit (kecuali dari Object, kelas dasar semua kelas di Java). Sementara itu, subclass adalah kelas turunan yang mewarisi semua atribut dan metode dari superclass, kecuali yang bersifat private. **Subclass** menggunakan keyword extends untuk menunjukkan bahwa ia mewarisi dari superclass. Selain menggunakan kembali fungsionalitas yang ada, subclass juga bisa menambahkan metode baru atau memodifikasi metode yang diwarisi. Dengan demikian, superclass berfungsi sebagai dasar umum, sedangkan subclass memperluas atau menyesuaikan perilaku tersebut sesuai kebutuhan.

**Keyword extends** digunakan dalam Java untuk menyatakan bahwa suatu kelas adalah subclass dari kelas lain (superclass). Dengan menggunakan extends, subclass otomatis mewarisi semua metode dan atribut yang tidak bersifat private dari superclass.

4. Bagaimana peran keyword super dalam pewarisan di Java? Jelaskan dua kegunaan utamanya dan berikan ilustrasi dengan kode sederhana.

**Jawab:**

Keyword super dalam Java digunakan untuk merujuk ke anggota (atribut atau metode) dari superclass (kelas induk) dalam konteks pewarisan. Keyword ini sangat berguna saat subclass ingin menggunakan kembali atau mengakses hal-hal dari superclass yang mungkin telah *di-override* atau disembunyikan oleh subclass.

### 1. Mengakses Konstruktor Superclass

```
class Kendaraan {
    Kendaraan() {
        System.out.println("Ini adalah konstruktor Kendaraan");
    }
}

class Mobil extends Kendaraan {
    Mobil() {
        super(); // Memanggil konstruktor Kendaraan
        System.out.println("Ini adalah konstruktor Mobil");
    }
}
```

```

    }

    public class Tes1 {
        public static void main(String[] args) {
            Mobil m = new Mobil();
        }
    }

```

---

## 2. Mengakses Metode Superclass

---

```

class Hewan {
    void suara() {
        System.out.println("Hewan bersuara");
    }
}

class Kucing extends Hewan {
    void suara() {
        System.out.println("Kucing mengeong");
    }

    void tampilkanSuara() {
        super.suara(); // Panggil metode dari superclass
        suara();      // Panggil metode dari subclass
    }
}

public class Tes2 {
    public static void main(String[] args) {
        Kucing k = new Kucing();
        k.tampilkanSuara();
    }
}

```

---

5. Jelaskan mengapa konstruktor tidak diwarisi oleh subclass dalam Java. Lalu bagaimana cara subclass bisa tetap menggunakan konstruktor dari superclass?

**Jawab:**

Dalam Java, konstruktor tidak diwarisi oleh subclass karena konstruktor bukan anggota (member) dari suatu kelas seperti metode atau atribut. Konstruktor adalah fungsi khusus yang digunakan hanya untuk menginisialisasi objek saat pertama kali dibuat, dan namanya harus sama persis dengan nama kelas tempat ia dideklarasikan. Karena nama konstruktor tidak bisa berbeda dari nama kelas, maka secara teknis tidak mungkin diwariskan ke subclass yang memiliki nama berbeda. Subclass bisa memanggil konstruktor superclass dengan menggunakan keyword `super`, dan ini harus dilakukan di baris pertama konstruktor subclass jika ingin

mengakses konstruktor superclass (terutama jika konstruktor superclass memiliki parameter).