

## **Tugas Pendahuluan: Modul 8**

**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 15 Mei 2025, pukul 13:00 (sebelum praktikum). Link pengumpulan: <https://forms.gle/AcPBWgx7is21tJUi7>

### **Soal**

1. Jelaskan konsep IS-A relationship dalam Java! Berikan contoh implementasinya dalam bentuk kode program sederhana. Apa manfaat dari penggunaan hubungan ini dalam pemrograman berorientasi objek?

**Jawab:**

#### **Contoh Is-a relationship dalam Java**

```
class Hewan {  
    void makan() {  
        System.out.println("Hewan sedang makan");  
    }  
}  
  
class Kucing extends Hewan {  
    void meong() {  
        System.out.println("Kucing meong");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Kucing k = new Kucing();  
        k.makan(); // pewarisan dari Hewan  
        k.meong();  
    }  
}
```

2. Apa yang dimaksud dengan Has-A relationship dalam Java? Jelaskan dengan contoh kode dan bandingkan perbedaannya dengan IS-A relationship!

**Jawab:**

#### **Has A relationship dalam Java**

```
class Mesin {  
    void hidupkan() {  
        System.out.println("Mesin hidup");  
    }  
}
```

```

    }
}

class Mobil {
    Mesin mesin = new Mesin(); // HAS-A relationship

    void nyalakanMobil() {
        mesin.hidupkan();
    }
}

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

```

### IS-A

Menggunakan extends

Hubungan pewarisan

Polimorfisme & warisan

### HAS-A

Menggunakan objek sebagai atribut

Hubungan kepemilikan

Komposisi/Agregasi

3. Kata kunci instanceof sering digunakan dalam Java. Jelaskan fungsi dari keyword ini dan berikan contoh penggunaannya! Dalam situasi seperti apa instanceof sebaiknya digunakan?

**Jawab:**

### Keyword instanceof

```

class Hewan {}
class Anjing extends Hewan {}

public class Main {
    public static void main(String[] args) {
        Hewan h = new Anjing();

        if (h instanceof Anjing) {
            System.out.println("Ini adalah anjing");
        }
    }
}

```

```

        } else {
            System.out.println("Bukan anjing");
        }
    }
}

```

Digunakan Saat:

- Saat melakukan casting objek turunan ke induk/superclass.
- Untuk menghindari ClassCastException.
- Saat mengimplementasikan polimorfisme.

4. Java mendukung beberapa tipe pewarisan. Jelaskan tipe-tipe pewarisan dalam Java dan berikan contoh kode untuk masing-masingnya! Sertakan penjelasan tentang pewarisan tunggal, pewarisan multilevel, dan pewarisan hierarkis. Mengapa Java tidak mendukung pewarisan berganda secara langsung?

**Jawab:**

## **Tipe” pewarisan dalam Java**

### **1.Single Inheritance**

```

class A {
    void tampil() {
        System.out.println("Dari kelas A");
    }
}

class B extends A {}

public class Main {
    public static void main(String[] args) {
        B obj = new B();
        obj.tampil();
    }
}

```

### **2.Multilevel Inheritance**

```

class A {
    void methodA() {
        System.out.println("Dari A");
    }
}

class B extends A {
    void methodB() {
        System.out.println("Dari B");
    }
}

```

```

    }

    class C extends B {
        void methodC() {
            System.out.println("Dari C");
        }
    }

    public class Main {
        public static void main(String[] args) {
            C obj = new C();
            obj.methodA();
            obj.methodB();
            obj.methodC();
        }
    }

```

---

### 3.Hierarchical inheritance

---

```

    class A {
        void tampil() {
            System.out.println("Dari kelas A");
        }
    }

    class B extends A {}
    class C extends A {}

    public class Main {
        public static void main(String[] args) {
            B obj1 = new B();
            C obj2 = new C();
            obj1.tampil();
            obj2.tampil();
        }
    }

```

---

5. Jelaskan perbedaan antara agregasi dan komposisi dalam konteks relasi antar objek dalam Java! Berikan contoh implementasi kode untuk masing-masing relasi tersebut. Kapan sebaiknya menggunakan agregasi dan kapan menggunakan komposisi?

**Jawab:**

Kriteria	Agregasi	Komposisi
Kepemilikan	Lemah (dapat berdiri sendiri)	Kuat (tergantung induknya)
Kehidupan objek	Tidak bergantung	Bergantung
Contoh	Dosen memiliki mahasiswa	Rumah memiliki kamar

## **A.Agregasi Java**

---

```
class Mahasiswa {
    String nama;
    Mahasiswa(String nama) {
        this.nama = nama;
    }
}

class Dosen {
    String nama;
    Mahasiswa mhs;

    Dosen(String nama, Mahasiswa mhs) {
        this.nama = nama;
        this.mhs = mhs;
    }

    void info() {
        System.out.println(nama + " membimbing " + mhs.nama);
    }
}

public class Main {
    public static void main(String[] args) {
        Mahasiswa m = new Mahasiswa("Andi");
        Dosen d = new Dosen("Bu Sari", m);
        d.info();
    }
}
```

---

## **B.Komposisi**

---

```
class Kamar {
    String nama;
    Kamar(String nama) {
        this.nama = nama;
    }
}

class Rumah {
    Kamar kamar;

    Rumah(String namaKamar) {
        this.kamar = new Kamar(namaKamar);
    }

    void info() {
        System.out.println("Rumah memiliki kamar: " + kamar.nama);
    }
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Rumah r = new Rumah("Tidur");  
        r.info();  
    }  
}
```

---

---