

Nama : Angga Saputra Ramadhan
Kelas : 2B
NIM : 244107020065
NoAbsen : 05

1. Perhatikan kode java berikut:

```
class Hewan {  
    private String nama = "Harimau";  
}  
  
class Kucing extends Hewan {  
    void tampilkan() {  
        System.out.println(super.nama);  
    }  
}
```

Akan muncul error, mengapa? Jelaskan dan perbaiki

Jwb.

Variabel nama dideklarasikan sebagai private di class hewan, sehingga tidak bisa langsung bisa diakses dari class kucing bahkan dengan super maka jika ingin menggunakan superclass nama harus menggunakan getter untuk mengakses nama seperti yang sudah perbaiki pada dibawah ini:

```
class Hewan {  
    private String nama = "Harimau";  
  
    public String getNama() {  
        return nama;  
    }  
}  
  
class Kucing extends Hewan {  
    void tampilkan() {  
        System.out.println(super.getNama());  
    }  
}
```

2. Perhatikan kode java berikut kemudian jelaskan konsep inheritance/pewarisan dalam code java tersebut secara runtut termasuk penggunaan super dalam pewarisan.

```

class Hewan2 {
    private String nama;

    Hewan2(String nama) {
        this.nama = nama;
    }

    public String getNama() {
        return nama;
    }

    public void setName(String nama) {
        this.nama = nama;
    }
}

class Kucing extends Hewan {
    private String ras;

    Kucing(String nama, String ras) {
        super(nama);
        this.ras = ras;
    }

    void tampilkanData() {
        System.out.println("Nama: " + super.getNama());
        System.out.println("Ras: " + ras);
    }
}

public class DemoSuperPrivate {
    public static void main(String[] args) {
        Kucing k = new kucing("Miko", "Persia");
        k.tampilkanData();
    }
}

```

Jwb.

Disini pada super() digunakan untuk memanggil constructor class induk sedangkan kalau super.getNama() untuk memanggil method lalu untuk subclass digunakan untuk mewarisi atribut dan method dari Hewan.

- Perhatikan kode java berikut, jelaskan konsep komposisi dalam kode tersebut kemudian hapus objek "Manusia" dalam kode tersebut dan jalankan kembali untuk membuktikan teori komposisi

```

class Jantung3 {
    public void berdetak() {
        System.out.println(x:"♥ Jantung berdetak...");
    }
}

class Manusia {
    private String nama;
    private Jantung3 jantung;

    public Manusia(String nama) {
        this.nama = nama;
        this.jantung = new Jantung3();
    }

    public void hidup() {
        System.out.println(nama + " sedang hidup.");
        jantung.berdetak();
    }
}

public class DemoKomposisi {
    Run | Debug
    public static void main(String[] args) {
        Manusia m1 = new Manusia(nama:"Andri");
        m1.hidup();
    }
}

```

Jwb.

Komposisi berarti hubungan kepemilikan total jika objek manusia dihapus maka objek jantung juga ikut hilang karena jika m1 dihapus maka jantung tidak ada karena ia di buat dalam class Manusia.

4. Lakukan hal sama untuk konsep agregasi. Buktikan teori agregasi dikode java berikut dengan menghapus salah satu objeknya.

```

class Guru4 {
    private String nama;

    public Guru4(String nama) {
        this.nama = nama;
    }

    public String getNama() {
        return nama;
    }

    public void mengajar() {
        System.out.println(nama + " sedang mengajar.");
    }
}

class Sekolah {
    private String namaSekolah;
    private Guru4 guru;

    public Sekolah(String namaSekolah, Guru4 guru) {
        this.namaSekolah = namaSekolah;
        this.guru = guru;
    }

    public void infoSekolah() {
        System.out.println("Nama Sekolah: " + namaSekolah);
        System.out.println("Guru: " + guru.getNama());
    }
}

public class DemoAgregasiSederhana {
    Run | Debug
    public static void main(String[] args) {
        Guru4 guru1 = new Guru4(nama:"Pak Budi");
        Sekolah sekolah1 = new Sekolah(namaSekolah:"SD Negeri Ngijo", guru1);

        sekolah1.infoSekolah();
        guru1.mengajar();

        sekolah1 = null;
        System.out.println(x:"\nSetelah sekolah dihapus, guru masih ada:");
        guru1.mengajar();
    }
}

```

Jwb.

Agregasi adalah hubungan yang memiliki tetapi tidak tergantung. Jika objek sekolah dihapus maka objek guru tetap bisa digunakan karena sekolah hanya menyimpan referensi ke objek guru yang sudah ada