

PENJELASAN TEORI

Benony Gabriel

105222002

1. Jelaskan pengaruh keyword super dalam inheritance!

Jawab:

Keyword super dalam Java digunakan untuk merujuk ke superclass atau parent class dari objek saat ini. Dalam konteks inheritance, super memiliki beberapa kegunaan penting:

- a) Mengakses Konstruktor Parent: super dapat digunakan untuk memanggil konstruktor dari parent class dalam konstruktor child class. Ini penting ketika parent class memiliki konstruktor yang perlu diinisialisasi sebelum child class dapat berfungsi dengan benar.

```
public Child(String name, int age) {  
    super(name); // Memanggil konstruktor Parent  
    this.age = age;  
}
```

- b) Mengakses Method dan Atribut Parent: super dapat digunakan untuk memanggil metode atau mengakses atribut dari parent class yang telah di-override atau disembunyikan di child class. Ini memungkinkan child class untuk memperluas atau mengubah perilaku dari method parent tanpa sepenuhnya menggantikan fungsionalitasnya.

```
@Override  
public void displayInfo() {  
    super.displayInfo(); // Memanggil method displayInfo dari Parent  
    System.out.println("Child Age: " + age);  
}
```

2. Jelaskan apakah terdapat perubahan di data parent jika nilai parent di ubah di child!

Jawab:

Ketika sebuah nilai atau atribut di parent class diubah melalui child class, tidak ada perubahan langsung pada data parent class kecuali atribut tersebut adalah bagian dari instance yang sama. Perubahan hanya terjadi pada instance spesifik yang diubah, bukan pada definisi class secara keseluruhan. Berikut adalah dua skenario umum:

- a) Instance-Specific Changes: Jika sebuah child class mengubah nilai atribut yang diwariskan dari parent class, perubahan ini hanya berlaku untuk instance tersebut dan bukan untuk class parent itu sendiri.

```
Child child = new Child("Bob", 20);  
child.name = "Charlie"; // Hanya mengubah nama di instance child tersebut
```

- b) Static Fields: Jika atribut yang diubah adalah static, maka perubahan tersebut akan berdampak pada semua instance dari class tersebut karena static fields adalah milik class, bukan instance.

```
public class Parent {  
    protected static String familyName;  
  
    public Parent(String familyName) {  
        Parent.familyName = familyName;  
    }  
}  
  
public class Child extends Parent {  
    public Child(String familyName) {  
        super(familyName);  
    }  
}  
  
Parent p = new Parent("Smith");  
Child c = new Child("Johnson");  
// Parent.familyName dan Child.familyName sekarang akan menjadi "Johnson"
```

Dengan demikian, perubahan yang dilakukan di child class hanya mempengaruhi instance spesifik yang diubah, kecuali untuk atribut static yang mempengaruhi seluruh class.