Nama : Alfiansyah Hidayat

Nim : 20210040098

Kelas : TI21F

Matkul : Pemrograman Berorientasi Objek

Sesi : Praktikum

Percobaan 1 :

Percobaan ini menunjukan penggunaan kata kunci “super”.

```
class Parent {
    public int x = 5;
}

class Child extends Parent {
    public int x = 10;
    public void Info(int x) {
    System.out.println("Nilai x sebagai parameter = " + x);
    System.out.println("Data member x di class Child = " + this.x);
    System.out.println("Data member x di class Parent = " +
super.x);
    }
}

public class NilaiX {
    public static void main(String args[]) {
        Child tes = new Child();
        tes.Info(20);
    }
}
```

Analisa: ketika objek “tes” dibuat dan objek itu memanggil fungsi info maka output yang akandihasilkan yaitu 20, 10, 5. Jika “this.x” maka nilai yang diambil adalah nilai x yang menempel pada objek itu karena itu bernilai 10. Sedang “super.x” dia aka mengambil nilai pada parent class karena itu bernilai 5

Percobaan 2

```
public class Pegawai {
   private String nama;
   public double gaji;
}

public class Manajer extends Pegawai {
   public String departemen;

   public void IsiData(String n, String d) {
      nama=n;
      departemen=d;
   }
}
```

Percobaan berikut ini menunjukkan penggunaan kontrol akses terhadap atribut parent class.Mengapa terjadi error, dan bagaimana solusinya?

Hasil Analisa :

Error yang pertama terjadi karena class Pegawai dinyatakan sebagai public seharusnya public digunakan ketika kelas pegawai diletakan pada filenya sendiri. Error kedua terjadi karena fungsi IsiData di class Manajer memanggil variabel nama dari class Pegawai, akan tetapi variabel nama diclass Pegawai di private sehingga terjadi error.

Solusinya

atribut nama pada kelas pegawai access modifier diganti dari private menjadi public.Dan pemanggilan nama pada fungsi IsiData diganti menjadi super.nama = n

Percobaan 3 :

```
public class Parent {
     // kosong
}

public class Child extends Parent {
    int x;
    public Child() {
        x = 5;
    }
}
```

Percobaan berikut ini menunjukkan penggunaan konstruktor yang tidak diwariskan. Mengapa terjadi error, dan bagaimana solusinya?

Jawab

Tidak terjadi error walaupun class Parent mempunyai constructor

Percobaan 4

Percobaan berikut ini menunjukkan penggunaan kelas Employee dan subkelas Manager yang merupakan turunannya. Kelas TestManager digunakan untuk menguji kelas Manager.

analisa

Karena semua penggunaan sudah benar, pemanggilan objek pertama menggunakan dengan 3 parameter yaitu nama, salary dan dept sedangkan objek kedua menggunakan konstruktor dengan 2 parameter yaitu nama dan dept.

Percobaan 5.

```
        //test inheritance class
        m = new HappyObject();
        m.speak();
        m.laugh();

        //test inheritance class
        m=new SadObject();
        m.speak();
        m.cry();
    }
}
```

Hasil Analisa :

Analisis: Tidak ada masalah dalam program ini, program ini akan menjalankan kelas yangdibuat menjalankan fungsi fungsinya.

Percobaan 6 :

```
class A {
    String var_a = "Variabel A";
    String var_b = "Variabel B";
    String var_c = "Variabel C";
    String var_d = "Variabel D";

    A(){
        System.out.println("Konstruktor A dijalankan");
    }
}

class B extends A{
    B(){
        System.out.println("Konstruktor B dijalankan ");
        var_a = "Var_a dari class B";
        var_b = "Var_a dari class B";
    }

    public static void main(String args[]){

        System.out.println("Objek A dibuat");
        A aa= new A();
        System.out.println("menampilkan nama variabel obyek aa");
        System.out.println(aa.var_a);
        System.out.println(aa.var_b);
        System.out.println(aa.var_c);
        System.out.println(aa.var_d);
        System.out.println("");

        System.out.println("Objek B dibuat");
        B bb= new B();
        System.out.println("menampilkan nama variabel obyek bb");
        System.out.println(bb.var_a);
        System.out.println(bb.var_b);
        System.out.println(bb.var_c);
        System.out.println(bb.var_d);
    }
}
```

Percobaan berikut ini menunjukkan penggunaan kelas A dan dengan subkelas B. Simpan kedua kelas ini dalam 2 file yang berbeda (A.java dan B.java) dan dalam satu package. Perhatikan proses pemanggilan konstruktor dan pemanggilan variabel

Hasil Analisa :

Terdapat dua class yaitu kelas A sebagai parent dan class B sebagai subclass dari parent A.Pada percobaan ini class A dan class B dijalankan dalam file berbeda. Class B masih dapat mengakses kelas A karena pada dasarnya modifier default membuat class B dapat mengakses class A yang terdapat pada satu package yang sama.

Percobaan 7 :

```
class Bapak {
        int a;
        int b;

        void show_variabel(){
            System.out.println("Nilai a=" + a);
            System.out.println("Nilai b=" + b);
        }
}
class Anak extends Bapak{
    int c;
    int b;
    int a;
    void show_variabel(){
        System.out.println("Nilai a=" + super.a);
        System.out.println("Nilai b=" + super.b);
        System.out.println("Nilai c=" + c);
    }
}

    public class inheritExample{

        public static void main(String[] args) {

            Bapak objectBapak = new Bapak();
            Anak objectAnak = new Anak();

            objectBapak.a=1;
            objectBapak.b=2;
            System.out.println("Object Bapak (Superclass)");

            objectBapak.show_variabel();
            objectAnak.c=5;
            System.out.println("Object Anak (Superclass dari Bapak)");
```

```
pbo_inheritance.Anak > show_variabel >
ut ×
PBO_Inheritance (run) × PBO - D:\PBO ×
Object Bapak (Superclass)
Nilai a=1
Nilai b=2
Object Anak (Superclass dari Bapak)
Nilai a=0
Nilai b=0
Nilai c=5
BUILD SUCCESSFUL (total time: 0 seconds)
```

Hasil Analisa :

Walaupun sudah melakukan modifikasi pada method show_variabel pada class anak dengan menggunakan super untuk menampilkan nilai a dan b nilainya akan tetap 0. Karena nilai dasarnya 0. Jadi, objek subclass tidak akan melakukan “Override” pada objek Bapak selama masih dalam bentuk objek.

Percobaan 8

```
public class Parent {
    String parentName;
    Parent(){}

    Parent(String parentName){
        this.parentName = parentName;
        System.out.println("Konstruktor parent");
    }

}

class Baby extends Parent {
    String babyName;

    Baby(String babyName){
        super();
        this.babyName = babyName;
        System.out.println("Konstruktor Baby");
        System.out.println(babyName);
    }

    public void Cry(){
        System.out.println("Owek owek");
    }
```

Hasil Analisa :

Pada class Baby terdapat super untuk meng-override class Parent nya. Atribut babyName diset pada constructor. Ketika class Baby dibuat akan menampilkan print dari constructor kelas Parent juga.