

TUGAS 3 PEMROGRAMAN BERORIENTASI OBJEK

Praktikum-Inheritance

Nama : Aditya Dwi Permana

NIM : 20210040181

Kelas : TI 21 F

1. Percobaan berikut ini menunjukkan penggunaan kata kunci “super”.

Source code:

```
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);
    }
}
```

Jawaban : Karena mewariskan dari NilaiX ke Class Parent

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

Source code:

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

\\
public class Manajer extends
    Pegawai{ public String
        departemen;

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

Jawaban : Solusi nya ada lah mengubah atribut nama menjadi public dari private karena tidakakan bisa di pakai method nya di class lain jika private.

```

class Employee {
    public static final double BASE_SALARY =
        15000.00; public String Name = "";
    public double Salary =
        0.0; public Date
        birthDate; public
        Employee() { }
    public Employee(String name, double salary, Date DoB){

        this.Name=name;
        this.Salary=salary;
        this.birthDate=Do
        B;
    }
    public Employee(String name,double
        salary){ this(name,salary,null);
    }
    public Employee(String name, Date
        DoB){
        this(name,BASE_SALARY,DoB);
    }
    public Employee(String
        name){
        this(name,BASE_SALARY
        );
    }
    public String GetName(){ return
        Name;} public double GetSalary(){
        return Salary; }
}
\\
public class TestManager {
    public static void main(String[] args) {
        Manager Utama = new
        Manager("John",5000000,"Financial");
        System.out.println("Name:"+ Utama.GetName());
        System.out.println("Salary:"+ Utama.GetSalary());
        System.out.println("Department:"+ Utama.GetDept());
        Utama = new Manager("Michael","Accounting");
        System.out.println("Name:"+ Utama.GetName());
        System.out.println("Salary:"+ Utama.GetSalary());
        System.out.println("Department:"+ Utama.GetDept());

    }
}

```

Jawaban : Mengubah atribut yang private menjadi public

5. Percobaan berikut ini menunjukkan penggunaan kelas MoodyObject dengan subkelas HappyObject dan SadObject. Kelas MoodyTest digunakan untuk menguji kelas dan subkelas.

Source Code :

Superclass

```
public class MoodyObject {  
    protected String  
    getMood(){  
        return "moody";  
    }  
    public void speak(){  
        System.out.println("I am "+getMood());  
    }  
    void laugh(){ }  
    void cry(){ }  
}
```

Subclass

Menggunkaan metode override

```
public class SadObject extends  
    MoodyObject { @Override
```

```
    protected String  
        getMood(){ return  
            "sad";  
        }  
    @Override  
    public void  
        cry(){  
            System.out.println("Hoo hoo");  
        }  
    }
```

Subclass

```
public class MoodyTest {  
    public static void main(String[] args){  
        MoodyObject m = new MoodyObject();
```

//test parent

```
class
```

```
m.speak();
```

//test inheritance class

```
m = new
```

```
HappyObject();
```

```
m.speak();
```

```
m.cry();
```

//test inheritance

```
classm = new
```

```
SadObject();
```

```

m.speak();
m.cry();
}
}

```

6. **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.**

Source Code :

```

Superclass
public 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");
}

}
Subclass

```

Menggunakan metode

overridepublic class B

extends A {

```

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

```

public static void main(String args[]){

```

System.out.println("Object 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("Object      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
");

}
}

```

7. Percobaan berikut ini menunjukkan penggunaan Inheritance dan Overriding method padakelas Bapak dan subkelas Anak. Terjadi override pada method show variabel.Perhatikan

Perubahan nilai pada variabel a, b, dan c.x

Source Code :

```

Superclass
public class
Bapak { int a;
int b;
void show_variable(){
    System.out.println("Nilai a="+ a);
    System.out.println("Nilai b="+ b);
}
}

```

```

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

```

```

    }
}
Subclass
public class InheritExample {
    public static void main(String[]
        args){ Bapak objectBapak =
        new Bapak(); Anak objectAnak
        = new Anak();

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

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

```

