

Praktikum Inheritance

Name : Ahmad busaeri

Class : TI21 F

Nim : 20210040175

Matkul : OOP

Percobaan 1:

Percobaan berikut ini menunjukkan penggunaan kata kunci “super”.

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

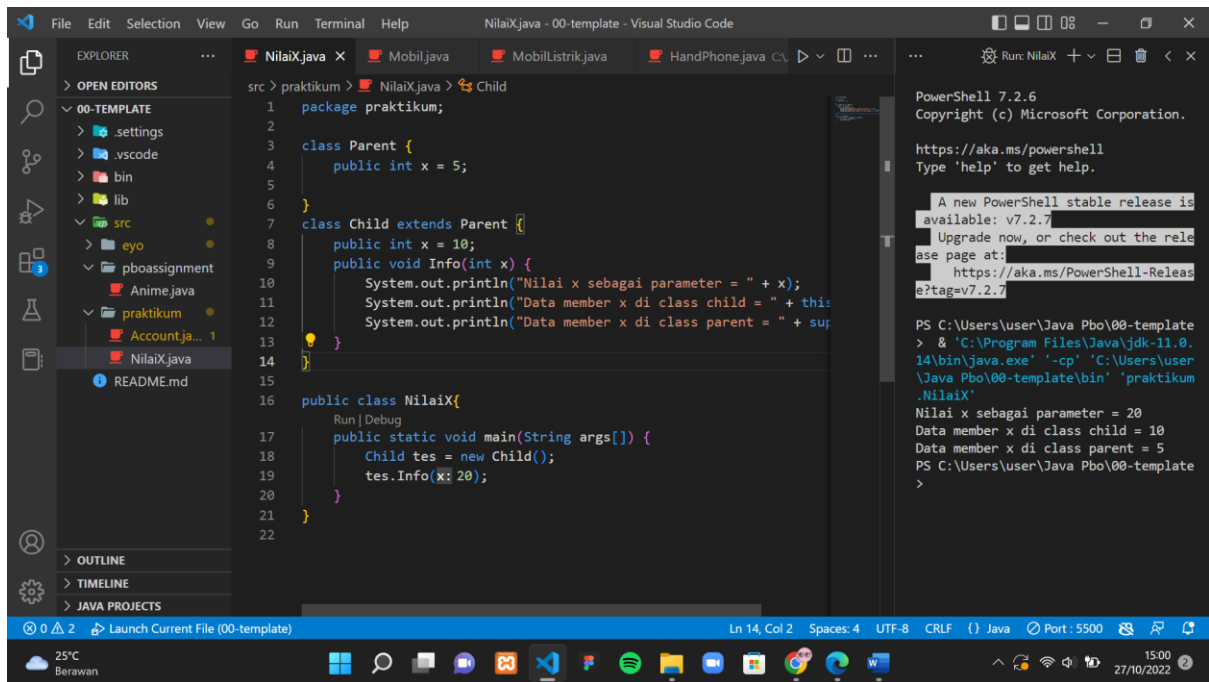
class Child extends Parent {
    public int x = 10;
    public void Info(int x) {
        System.out.println("Nilai x sebagai parameter = " + x);
        System.out.println("Data member x di class Child = " + this.x);
        System.out.println("Data member x di class Parent = " +
            super.x);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

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 akan dihasilkan yaitu 20, 10, 5. Walaupun variabel yang sama yaitu “x” tetapi yang membedakan adalah pemanggil variabelnya. Jika hanya “x” saja yang dipanggil maka x itu hanya nilai dari parameter (jika itu dalam sebuah fungsi) karena itu bernilai 20. Jika “this.x” maka nilai yang diambil adalah nilai x yang menempel pada objek itu karena itu bernilai 10. Sedang “super.x” dia akan mengambil nilai pada parent class karena itu bernilai 5

Output :



Percobaan 2:

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

```
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;
    }
}
```

Analisa: terjadi error karena pada kelas Manajer dan dalam fungsi Isi Data memanggil variabel nama sedangkan dalam class Manajer tidak ada variabel nama. Solusi: 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:

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

```
public class Parent {
    // kosong
}

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

Hasil Tidak terjadi error walaupun parent class tidak mempunyai konstruktor

Percobaan 4:

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

```
class Employee {
    private static final double BASE_SALARY = 15000.00;
    private String Name = "";
    private double Salary = 0.0;
    private Date birthDate;

    public Employee() {}
    public Employee(String name, double salary, Date DoB){
        this.Name=name;
        this.Salary=salary;
        this.birthDate=DoB;
    }
    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; }
}

class Manager extends Employee {

    //tambahan attribrute untuk kelas manager
    private String department;

    public Manager(String name,double salary,String dept){
        super(name,salary);
        department=dept;
    }
    public Manager(String n,String dept){
        super(n);
        department=dept;
    }
    public Manager(String dept){
        super();
        department=dept;
    }
    public String GetDept(){
        return department;
    }
}

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());
    }
}

```

Hasil analisa :

Tidak ditemukan error Pemanggilan objek pertama menggunakan konstruktor dengan 3 parameter yaitu nama , salary, dan Dept sedangkan object kedua menggunakan konstruktor dengan 2 parameter yaitu nama dan Dept

```

1  package praktikum;
2
3  import java.sql.Date;
4
5  class Employee {
6      private static final double BASE_SALARY = 15000.00;
7      private String Name = "";
8      private double Salary = 0.0;
9      private Date birthDate;
10
11     public Employee(){}
12     public Employee(String name, double salary, Date DoB){
13         this.Name=name;
14         this.Salary=salary;
15         this.birthDate=DoB;
16     }
17     public Employee(String name,double salary){
18         this(name,salary,null);
19     }
20     public Employee(String name, Date DoB){
21         this(name,BASE_SALARY,DoB);
22     }
23     public Employee(String name){
24         this(name,BASE_SALARY);
25     }
26     public String GetName(){ return Name;}
27     public double GetSalary(){ return Salary; }
28 }
29
30 class Manager extends Employee {
31     //tambahan attribrute untuk kelas manager
32     private String department;
33
34     public Manager(String name,double salary,String dept){
35         super(name,salary);
36         department=dept;
37     }
38     public Manager(String n,String dept){
39         super(n);
40         department=dept;
41     }
42     public Manager(String dept){
43         super();
44         department=dept;
45     }
46     public String GetDept(){
47         return department;
48     }
49 }
50 public class TestManager {
51     public static void main(String[] args) {
52         Manager Utama = new Manager("John",5000000,"Financial"); System.out.println("Name:"+ Utama.GetName());
53         System.out.println("Salary:"+ Utama.GetSalary());
54
55         System.out.println("Department:"+ Utama.GetDept());
56         Utama = new Manager("Michael","Accounting");
57         System.out.println("Name:"+ Utama.GetName());
58         System.out.println("Salary:"+ Utama.GetSalary());
59         System.out.println("Department:"+ Utama.GetDept());
60     }
61 }
62

```

Percobaan 5:

Hasil Analisa :

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

Percobaan 6:

Hasil Analisis: terdapat 2 kelas yaitu kelas A sebagai parent class dan kelas B sebagai subclass dari A, sub class A akan mengganti nilai var_a dan var_b dari parent kelas nya. Ketika objek B dibuat, constuktor A akan tetap dijalankan

Percobaan 7 :

Percobaan 8:

Analisis: pada kelas Baby mewariskan Parent. terdapat super() pada fungsi konstruktor yang akan mengoverride kelas parentnya. this.babyName = babyName untuk passing nilai babyName pada objek dengan parameter konstruktor babyName