

Nama: Kanza Salsabila

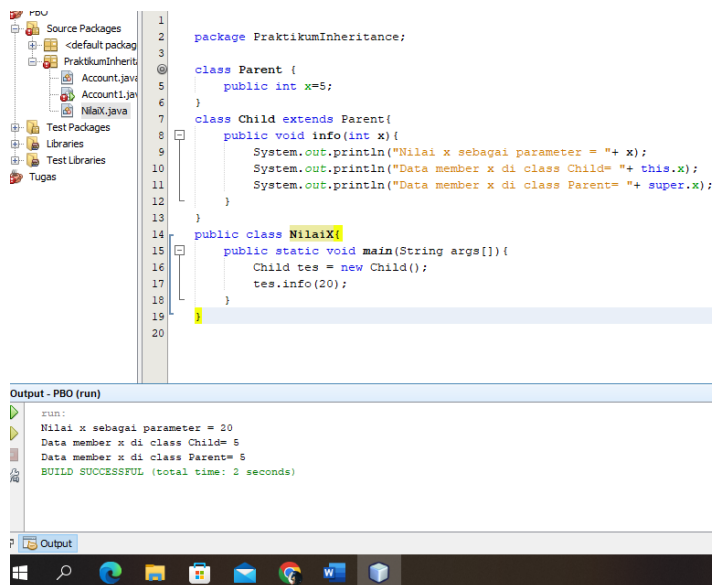
NIM: 20210040094

Kelas: TI21F

Praktikum Inheritance

Percobaan 1

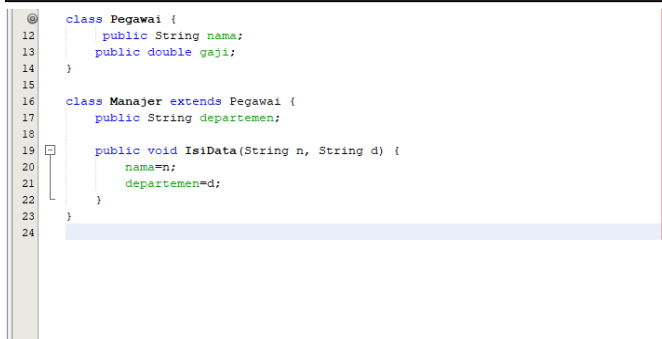
```
class Parent {  
    public int x = 5;  
}  
class Child extends Parent  
{  
    public void Info(int x)  
    {  
        System.out.println("Nilai x sebagai parameter = " + x);  
        System.out.println("Data member x di class Child = " + this.x);  
        System.out.println("Data member x di class Parent = " + super.x);  
    }  
}  
public class NilaiX {  
    public static void main(String args[]) {  
        Child tes = new Child();  
        tes.Info(20);  
    }  
}
```



class Parent sebagai induk class yang memiliki atribut integer x = 5, child sebagai sub class dan didalam class Child terdapat sebuah nilai parameter 20, karena ditentukan dari tes info, dan ada data member dari class Parent bernilai 5, kenapa nilainya 5 karena “super” mengambil nilai integer dari class Parent.

Percobaan 2

```
class Pegawai {  
    public String nama;  
    public double gaji;  
}  
class Manajer extends Pegawai {  
    public String departemen;  
  
    public void IsiData(String n, String d) {  
        nama=n; departemen=d;  
    }  
}
```

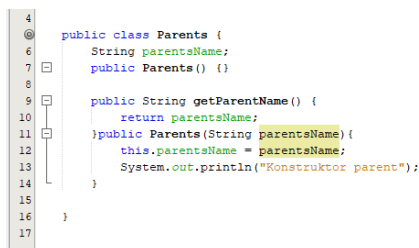


```
12 class Pegawai {  
13     public String nama;  
14     public double gaji;  
15 }  
16 class Manajer extends Pegawai {  
17     public String departemen;  
18  
19     public void IsiData(String n, String d) {  
20         nama=n;  
21         departemen=d;  
22     }  
23 }  
24
```

Solusinya yaitu mengganti “private String nama” menjadi “public String nama” tidak error namun tidak bisa di run dikarenakan tidak terdapat main method.

Percobaan 3

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



```
4 public class Parents {  
6     String parentsName;  
7     public Parents() {}  
8  
9     public String getParentName() {  
10         return parentsName;  
11     }  
12     public Parents(String parentsName) {  
13         this.parentsName = parentsName;  
14         System.out.println("Konstruktor parent");  
15     }  
16 }  
17
```

Percobaan ke 3 sama dengan percobaan 2 tidak error namun tidak bisadi run dikarenakan tidak terdapat main method.

```

import java.util.Date;
public class Employee {
    private static final double BASE_SALARY =
15000.00; private String Name = ""; private double
Salary = 0.0;
    private Date birthDate;

    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; }
    public Date GetbirthDate(){return birthDate; }
}
class Manager extends Employee {
//tambahan attribrute untuk kelas manager
    private String department;
    public Manager(String name, double salary, Date DoB) {
super(name, salary, DoB);
}
    public Manager(String n,String dept){
super(n);
department=dept;
} public Manager(String dept, int par, String
financial){ super(dept); department=dept;
}
    public String GetDept(){
return department;
}

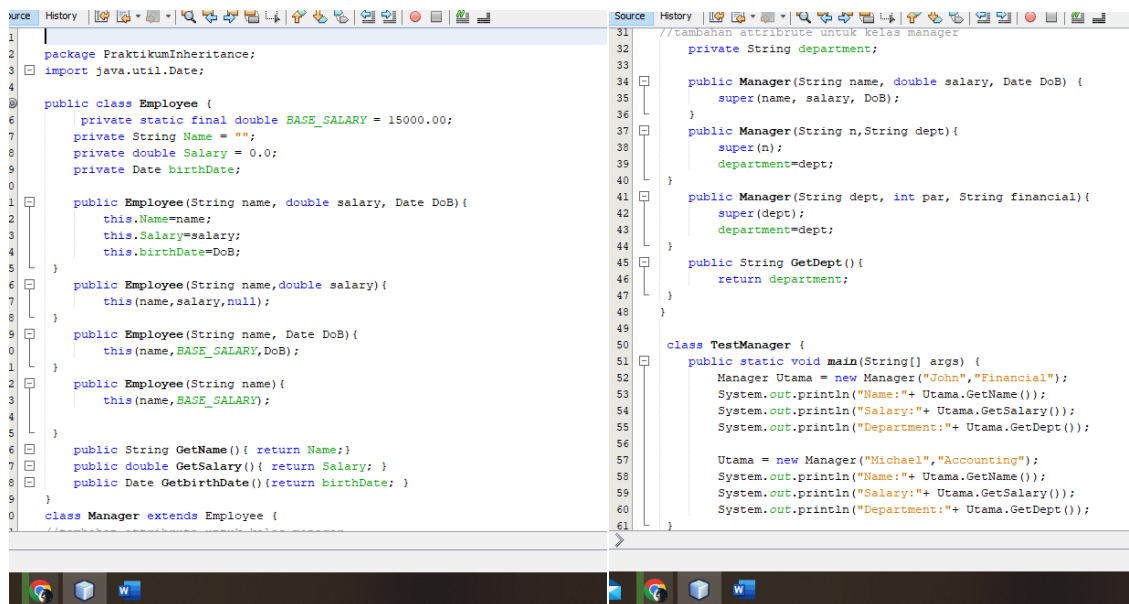
class TestManager {
    public static void main(String[] args) {
        Manager Utama = new Manager("John","Financial");
        System.out.println("Name:"+ Utama.GetName());
        System.out.println("Salary:"+ Utama.GetSalary());
        System.out.println("Department:"+ Utama.GetDept());
    }
}

```

```

Utama = new Manager("Michael","Accounting");
System.out.println("Name:"+ Utama.GetName());
System.out.println("Salary:"+ Utama.GetSalary());
System.out.println("Department:"+ Utama.GetDept());
}
}

```



Percobaan ini menunjukkan penggunaan kelas Employee dan subkelas Manager yang merupakan turunannya. Kelas TestManager digunakan untuk menguji jalannya sebuah program tersebut.

Percobaan 5

```

public class MoodyObject {
    protected String getMood(){
        return "moody";
    }
    public void speak(){
        System.out.println("I am : "+getMood());
    }
}
public class SadObject extends MoodyObject{

    protected String getSad(){
        return "sad";
    }
    public void cry(){
        System.out.println("Hoo hoo : "+ getSad());
    }
}
public class HappyObject extends MoodyObject{

    protected String getHappy(){
        return "happy";
    }
}

```

```

public void laugh(){
    System.out.println("Hahaha : " +
        getHappy());
}
}
public class MoodyTest {
    public static void main(String[] args) {
        MoodyObject m = new
            MoodyObject();
        SadObject Sad = new SadObject();
        HappyObject Happy = new
            HappyObject();

        m.speak();
        Sad.cry();
        Happy.laugh();
    }
}

```

```

26 }
27 public class MoodyTest {
28     public static void main(String[] args) {
29         MoodyObject m = new MoodyObject();
30         SadObject sad = new SadObject();
31         HappyObject happy = new HappyObject();
32
33         m.speak();
34         sad.cry();
35         happy.laugh();
36     }
37 }

```

Output :

```

I am : moody
Hoo hoo : sad
Hahaha : happy
BUILD SUCCESSFUL (total time: 0 seconds)

```

Pada Percobaan ini menunjukkan penggunaan kelas MoodyObject dengan subkelas HappyObject dan SadObject. Kelas MoodyTest digunakan untuk menguji kelas dan subkelas dalam menjalankan sebuah Program

1. SadObject berisi : sad, method untuk menampilkan pesan, tipe public
2. HappyObject berisi : laugh, method untuk menampilkan pesan, tipe public
3. MoodyObject berisi : • getMood, memberi nilai mood sekarang, tipe public, return type string • Speak, menampilkan mood, tipe public

Percobaan 6

```

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

    ClassA(){
        System.out.println("Konstruktor A dijalankan"); } }
public class ClassB extends ClassA{
    ClassB(){
        System.out.println("Konstruktor B dijalankan ");
        var_a = "Var_a dari class B"; var_b = "Var_a
        dari class B"; var_c = "Var_a dari class B";
        var_d = "Var_a dari class B";
    }
    public static void main(String args[]){
        System.out.println("Objek A dibuat");
        ClassA aa= new ClassA();
        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");
ClassB bb= new ClassB();
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);
}
}

```

```

2 package PraktikumInheritance;
3
4
5 public class ClassA {
6     String var_a = "Variabel A";
7     String var_b = "Variabel B";
8     String var_c = "Variabel C";
9     String var_d = "Variabel D";
10
11     ClassA() {
12         System.out.println("Konstruktor A dijalankan");
13     }
14 }
15
16
17 public class ClassB extends ClassA {
18     ClassB() {
19         System.out.println("Konstruktor B dijalankan ");
20         var_a = "Var_a dari class B";
21         var_b = "Var_a dari class B";
22         var_c = "Var_a dari class B";
23         var_d = "Var_a dari class B";
24     }
25
26     public static void main(String args[]) {
27         System.out.println("Objek A dibuat");
28         ClassA aa = new ClassA();
29         System.out.println("menampilkan nama variabel obyek aa");
30         System.out.println(aa.var_a);
31         System.out.println(aa.var_b);
32         System.out.println(aa.var_c);
33         System.out.println(aa.var_d);
34         System.out.println("");
35
36         System.out.println("Objek B dibuat");
37         ClassB bb = new ClassB();
38         System.out.println("menampilkan nama variabel obyek bb");
39         System.out.println(bb.var_a);
40         System.out.println(bb.var_b);
41         System.out.println(bb.var_c);
42         System.out.println(bb.var_d);
43     }
44 }

```

```

Output - praktikum-inheritance (run)
SUM:
Objek A dibuat
Konstruktor A dijalankan
menampilkan nama variabel obyek aa
Variabel A
Variabel B
Variabel C
Variabel D

Objek B dibuat
Konstruktor A dijalankan
Konstruktor B dijalankan
menampilkan nama variabel obyek bb
Var_a dari class B
Var_a dari class B
Var_a dari class B
Var_a dari class B
BUILD SUCCESSFUL (total time: 0 seconds)

```

Pada percobaan ini menunjukkan penggunaan kelas A dan dengan subkelas B. kemudian simpan file tersebut dalam class yang berbeda dan dalam satu package. Kemudian proses pemanggilan konstruktor dan pemanggilan variabel dalam program tersebut.

Percobaan 7

```

public class Bapak {
    int a;
    int b;

    public void show_variabel(){ System.out.println("Nilai a="+
a);
System.out.println("Nilai b="+ b);
}
}

```

```

public class Anak extends Bapak{
    int c;

    public 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 InheritanceExample {

    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_variabel();
    }
}

```

```

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

```

```

3      public class InheritanceExample {
4
5
6      public static void main(String[] args) {
7          Bapak objectBapak = new Bapak();
8          Anak objectAnak = new Anak();
9
10
11          objectBapak.a=1;
12          objectBapak.b=1;
13          System.out.println("Object Bapak (Superclass:");
14          objectBapak.show_variabel());
15
16          objectAnak.c=5;
17          System.out.println("Object Anak dari Bapak:");
18          objectAnak.show_Variabel());
19      }
20  }
21

```

```
Output - praktikum-inheritance (run)
```

```
run:  
Object Bapak (Superclass):  
Nilai a=1  
Nilai b=1  
Object Anak (Superclass dari Bapak):  
Nilai a=0  
Nilai b=0  
Nilai c=5  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Percobaan 8

```
public class TestBaby { public static void
main(String
args[]){ Baby x = new Baby("Nurul Intan");
x.cry();
}
}
```

```
public class Baby extends Parent{
    String babyName;

    public String getBabyName() {
        return 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");
    }
}
```

```
4 public class Parents {
6     String parentsName;
7     public Parents() {}
8
9     public String getParentName() {
10         return parentsName;
11     }
12     public Parents(String parentsName) {
13         this.parentsName = parentsName;
14         System.out.println("Konstruktor parent");
15     }
16 }
17
```

```
1 public class TestBaby {
2     public static void main(String args[]) {
3         Baby x = new Baby("Nurul Intan");
4         x.cry();
5     }
6 }
```

Output - praktikum-inheritance (run)

```
run:
Konstruktor Baby
Nurul Intan
owek owek
BUILD SUCCESSFUL (total time: 0 seconds)
```

Percobaan ini menggunakan method Overriding pada Kelas Parent dan subclass Baby(extends) Kemudian cara menguji kinerja dari program tersebut dengan membuat class test baby dan programpun akhirnya dapat berjalan.