

## Test 1

```
- package praktikum_inheritance;
- class Mobil {
-     public int x = 5;
- }
- class Motor extends Mobil {
-     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 Motor = " + this.x);
-         System.out.println("Data member x di class Mobil = " + super.x);
-     }
- }
- public class Test_1 {
-     public static void main(String args[]) {
-         Motor test = new Motor();
-         test.Info(20);
-     }
- }
```

- Pada percobaan pertama ini class mobil sebagai induk class yang memiliki atribut integer = 5, motor sebagai subclass dan di dalam class child ada sebuah nilai parameter karena ditentukan dari tes.info dan ada data member dari class mobil yang memiliki nilai 5 karena super mengambil nilai integer dari class mobil.

## Test 2

```
- package praktikum_inheritance;  
- public 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;  
-     }  
- }
```

- Menunjukkan penggunaan kontrol akses terhadap atribut class pegawai.
- Solusinya yaitu menghapus “public” pada public class begitu pula dengan class manajer dan ditambahkan String nama agar nama = n di public void isi.Data tidak terjadi error.

### Test 3

```
- package praktikum_inheritance;  
- public class Mobil {  
-     public class Motor extends Mobil {  
-         int x;  
-         public Motor() {  
-             x = 5;  
-         }  
-     }  
- }
```

- Ini menunjukkan bahwa penggunaan konstruktor yang tidak diwariskan
- Mengapa konstruktor terjadi eror? Karena konstruktor tersebut berada di subclass
- Solusinya menambahkan class utama yaitu Test\_3

## Test 4

```
- package praktikum_inheritance;
- import java.util.Date;
- public class Pekerja {
-     private static final double BASE_SALARY = 10000.00;
-     private String Name = "";
-     private double Salary = 0.0;
-     private Date birthDate;
-
-     public Pekerja(String name, double salary, Date DoB){
-         this.Name=name;
-         this.Salary=salary;
-         this.birthDate=DoB;
-     }
-     public Pekerja(String name,double salary){
-         this(name,salary,null);
-     }
-     public Pekerja(String name, Date DoB){
-         this(name,BASE_SALARY,DoB);
-     }
-     public Pekerja(String name){
-         this(name,BASE_SALARY);
-     }
-
-     public String GetName(){ return Name;}
-     public double GetSalary(){ return Salary; }
-     public Date GetbirthDate(){return birthDate; }
- }
- class Manager extends Pekerja {
-
-     //menambahkan atribut 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("Ahmad","Bagian Gudang");
-         System.out.println("Name:"+ Utama.GetName());
-         System.out.println("Salary:"+ Utama.GetSalary());
-         System.out.println("Department:"+ Utama.GetDept());
-
-         Utama = new Manager("Udin","Obe");
-         System.out.println("Name:"+ Utama.GetName());
-         System.out.println("Salary:"+ Utama.GetSalary());
-         System.out.println("Department:"+ Utama.GetDept());
-     }
- }
```

- Pada percobaan ini penggunaan kelas Pekerja dan subclass Manager sebagai turunannya
- Class TestManager digunakan untuk menguji

## Test 5

```
- package praktikum_inheritance;
- public class Pedas extends Ekspresi{
-
-     protected String getSad(){
-         return "Pedas";
-     }
-     public void cry(){
-         System.out.println("HAAAAAAAAAH : "+ getSad());
-     }
- }
```

```
- package praktikum_inheritance;
- public class Test_Ekspresi {
-     public static void main(String[] args) {
-         Ekspresi eks = new Ekspresi();
-         Pedas pds = new Pedas();
-         Senang sng = new Senang();
-
-         eks.speak();
-         pds.cry();
-         sng.laugh();
-     }
- }
```

```
- package praktikum_inheritance;
- public class Ekspresi {
-     protected String getEkspresi(){
-         return "Manis";
-     }
-     public void speak(){
-         System.out.println("Saya itu "+getEkspresi());
-     }
- }
```

- Percobaan ini adalah tentang penggunaan class Test\_Ekspresi dengan subclass Senang dan Pedas
- Class Test\_Ekspresi digunakan untuk menguji class dan subclass-nya

## Test 6

```
- package praktikum_inheritance;
- public class Inher_Ovrid {
-
-     public static void main(String[] args) {
-         Guru objectGuru = new Guru();
-         Murid objectMurid = new Murid();
-
-
-
-
-         objectGuru.a=1;
-         objectGuru.b=1;
-         System.out.println("Object Guru (Superclass)      :");
-         objectGuru.show_variabel();
-
-
-         objectMurid.c=5;
-         System.out.println("Object Murid (Superclass dari Guru) :");
-         objectMurid.show_Variabel();
-     }
- }
```

```
- package praktikum_inheritance;
- public class Guru {
-     int a;
-     int b;
-
-     public void show_variabel(){
-         System.out.println("Nilai a="+ a);
-         System.out.println("Nilai b="+ b);
-     }
- }
```

```
- package praktikum_inheritance;
- public class Murid extends Guru{
-     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);
-     }
- }
```

- Inheritance dan Overriding method pada class Dosen dan subclass Murid
- Akan terjadi override pada method show\_Variabel
- Subclass Murid nilai a dan b mewarisi nilai Guru, dan c adalah nilai dari objek Murid