

Nama : Mohamad Reza Ashari  
Nim : 20210040090  
Kelas : TI 21 E  
Matkul : Pemrograman Beroientasi Objek

### **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);  
    }  
}
```

Ketika objek "tes" dibuat dan memanggil fungsi info maka output yang dihasilkan : 20,10,5.  
Walaupun "X" variabel nya sama tapi pemanggilan variabelnya yang beda.

Jika "X" nya diganti "this.x" nilai yang diambil ialah nilai x yang menempel pada objek itu karena bernilai 10. Jika "super.x" maka akan mengambil nilai pada parent class bernilai 5.

### **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;  
    }  
}
```

Adanya error dalam fungsi Data memanggil variabel nama pada kelas Manajer, di class Manajer tidak ada variabel nama.

Atribut nama di kelas pegawai access modifier diubah menjadi public dan pemanggilan nama 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;  
    }  
}
```

Tidak ada 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;  
    }  
}
```

Tidak ada error atau masalah. Objek pertama pemanggilan menggunakan konstruktor dengan 3 parameter : nama, salary, dan dept. Objek kedua menggunakan konstruktor 2 parameter : nama, dept

**Percobaan 5:**

Program ini tidak ada masalah. Di dalam program ini akan menjalankan kelas yang dibuat menjalankan fungsi-fungsinya.

**Percobaan 6:**

Ada 2 kelas : A Parent class dan B subclass dari A. Sub class A akan menggantikan nilai var\_a dan var\_b parent kelasnya ketika objek B dibuat, konstruktor A tetap dijalankan.

**Percobaan 7:**

Nilai a dan b dari kelas anak akan tetap 0 walaupun sudah menggunakan super pada kelas anak agar dapat mengakses nilai dari parent kelas karena blueprint nya bernilai 0. Objek anak akan melakukan override pada objek bapak selama dalam bentuk objek.

**Percobaan 8:**

Pada kelas Baby menurunkan Parent. Terdapat super() pada fungsi konstruktor yang akan mengoverride kelas parentnya. This.babyName = babyName untuk passing nilai babyName pada objek dengan parameter konstruktor babyName