

Tugas Pratikum Pemrograman Berorientasi Objek

Nama : M. Yusgi Padilllah

Kelas : TI21E

NIM : 20210040126

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);
    }
}
```

Analisis: Saat objek "test" dibuat, objek memanggil fungsi info, dan output yang dihasilkan adalah 20, 10, 5. Walaupun variabelnya sama yaitu "x", yang membedakan adalah pemanggilan variabelnya. Jika hanya "x" yang dipanggil, maka x hanyalah nilai parameter (jika ada dalam fungsi), karena nilainya 20. Jika "this.x", nilai yang diambil adalah nilai x yang melekat pada objek karena memiliki nilai 10. Dan "super.x" akan mengambil nilai dari kelas induk, karena memiliki nilai 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;
    }
}
```

Analisis: Kesalahan karena variabel nama dipanggil di kelas Manajer dan fungsi Isi Data, tetapi tidak ada variabel nama di kelas Manajer. Solusi: Ubah atribut nama di kelas pengubah akses

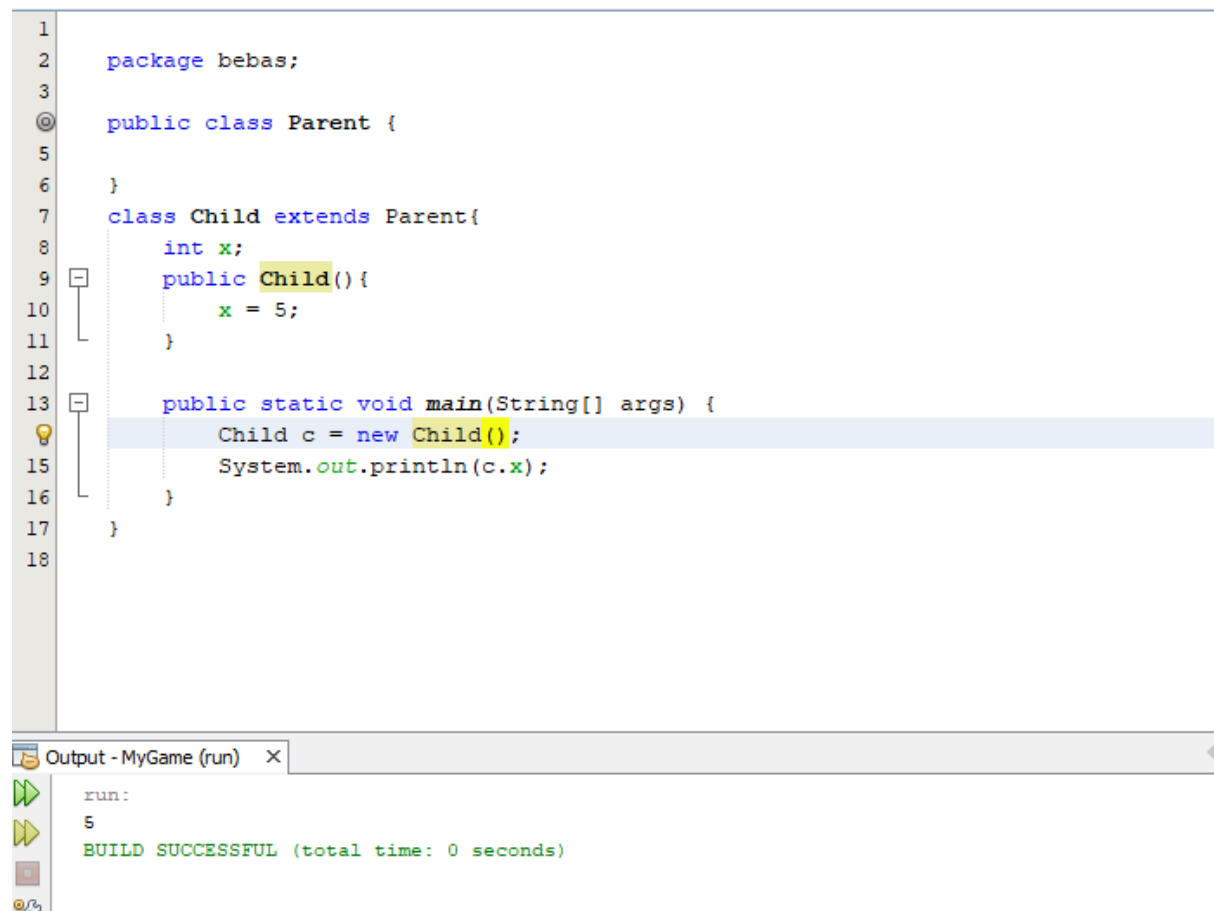
karyawan dari privat ke publik. dan ubah pemanggilan nama pada fungsi IsiData menjadi super.name = n

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;  
    }  
}
```

Analisis: Meskipun kelas induk tidak memiliki konstruktor, tidak ada kesalahan yang dilaporkan



```
1  
2 package bebas;  
3  
4 public class Parent {  
5  
6 }  
7 class Child extends Parent{  
8     int x;  
9     public Child() {  
10         x = 5;  
11     }  
12  
13     public static void main(String[] args) {  
14         Child c = new Child();  
15         System.out.println(c.x);  
16     }  
17 }  
18
```

Output - MyGame (run) x

```
run:  
5  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Percobaan 4 :

Karena semua penggunaan sudah benar, Memanggil objek pertama menggunakan konstruktor dengan 3 parameter, nama, salary, dan Dept. Sedangkan objek kedua menggunakan konstruktor dengan 2 parameter, nama dan Dept.

Percobaan 5 :

Analisis: Tidak ada masalah dengan program ini, program ini akan menjalankan kelas yang dibuat untuk menjalankan fungsinya.

Percobaan 6 :

Analisis: ada 2 kelas yaitu kelas A sebagai kelas induk dan kelas B subkelas A, subkelas A akan menggantikan nilai var_a dan var_b kelas induknya. Saat objek B dibuat, konstruktor A akan dieksekusi.

Percobaan 7:

Analisis: Walaupun sudah menggunakan super pada kelas anak untuk mengakses nilai dari parent kelas, nilai a dan b dari kelas anak akan tetap 0 karena pada dasarnya blueprint memiliki nilai 0. Dengan demikian, objek anak tidak akan "override" objek bapak, asalkan objek tersebut berbentuk objek.

Percobaan 8 :

```
public class Parent {
    String parentName;
    Parent() {}

    Parent(String parentName) {
        this.parentName = parentName;
        System.out.println("Konstruktor parent");
    }
}

class Baby extends Parent {
    String 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");
    }
}
```

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