

Nama : Muhammad Syarif Hidayat

NIM : 20210040123

Kelas : TI21E

Tugas Praktikum

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 dan 5. Walaupun variabel yang sama yaitu “x” tetapi yang membedakan adalah pemanggilan variabelnya. Jika hanya “x” saja yang dipanggil maka x itu hanya nilai dari parameter (jika itu dalam fungsi) karena itu bernilai 20. Jika “this.x” maka nilai yang diambil adalah nilai x yang menempel pada objek itu karena itu bernilai 10. Sedangkan “super.x” dia akan mengambil nilai pada parent class karena itu 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;
    }
}
```

Analisa : Terjadi error karena pada kelas Manajer dan dalam fungsi IsiData memanggil variabel nama sedangkan dalam class Manajer tidak ada variabel nama.

Solusi : Atribut nama pada kelas Pegawai access modifiernya diganti dari private menjadi public. Dan pemanggilan nama pada fungsi IsiData diganti menjadi super.nama=n.

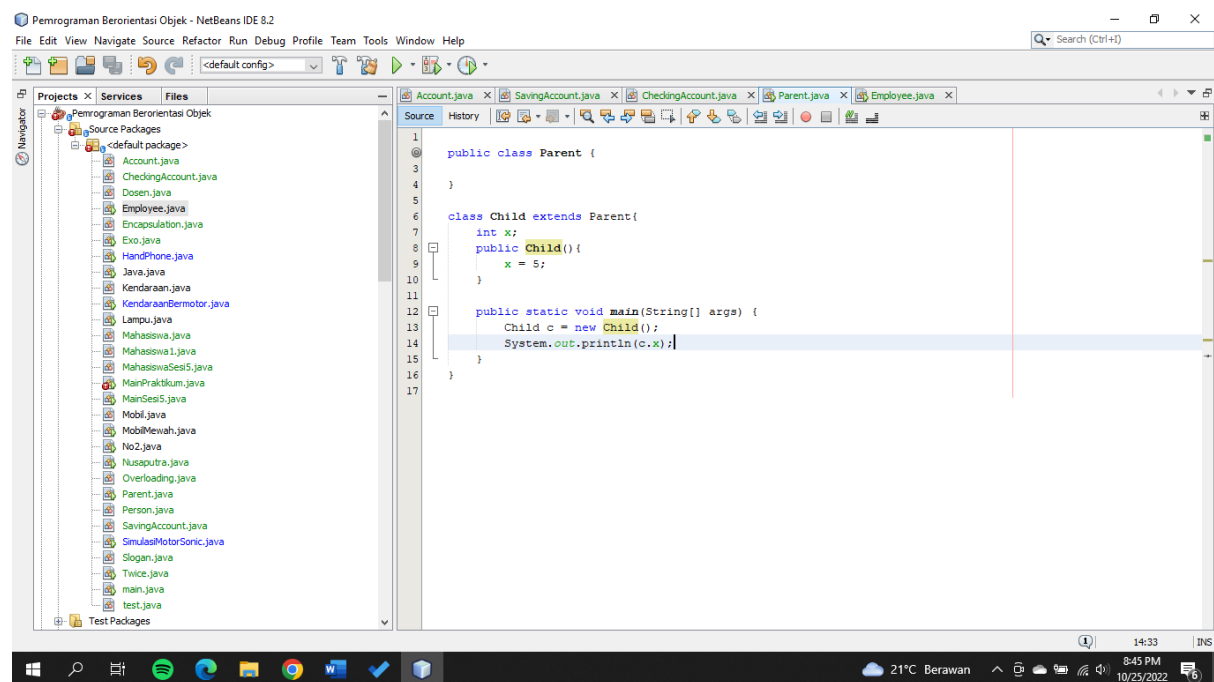
Percobaan 3:

Percobaan berikut ini menunjukkan penggunaan konstruktor yang tidak diwariskan.

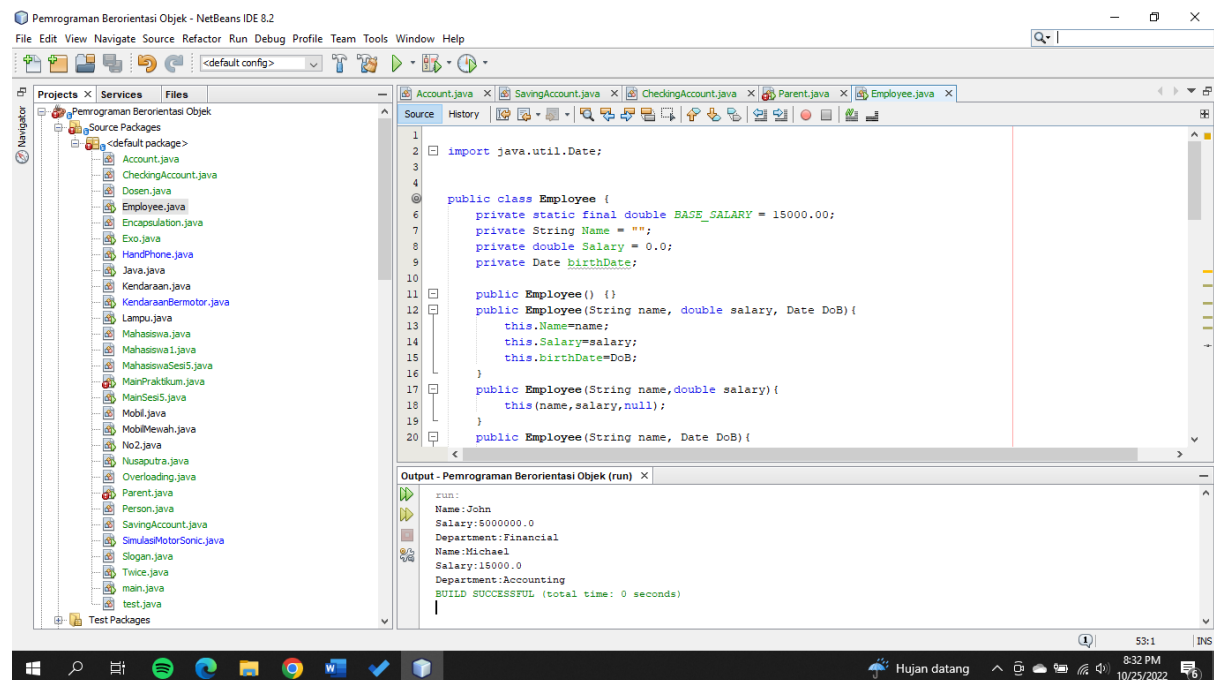
Mengapa terjadi error, dan bagaimana solusinya?

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

Analisa : Tidak terjadi error walaupun parent class tidak mempunyai konstruktor.

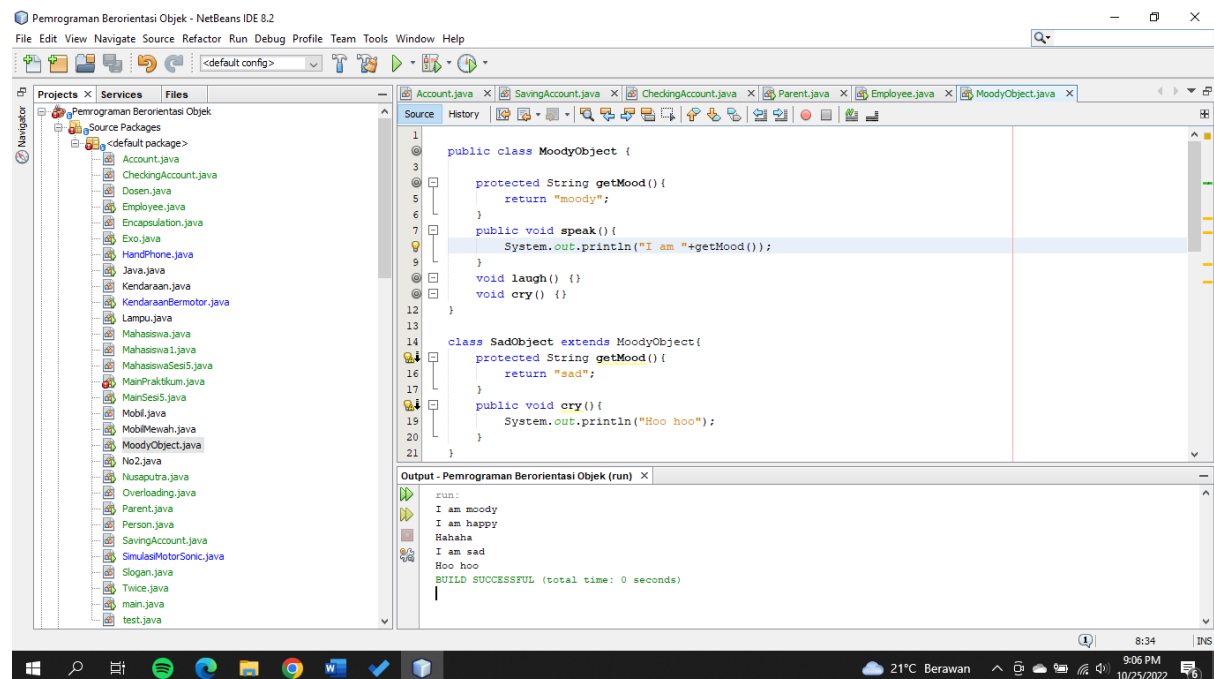


Percobaan 4 :



Analisa : Ada beberapa kesalahan dalam penulisan yang seharusnya “department=dept;” menjadi “department-dept;” dan juga kita menambahkan “import java.until.Date” agar “private Date birthdate;” terbaca. Pemanggilan objek pertama menggunakan konstruktor dengan 3 parameter yaitu, name, salary dan dept, sedangkan objek kedua menggunakan konstruktor dengan 2 parameter yaitu name dan dept.

Percobaan 5 :

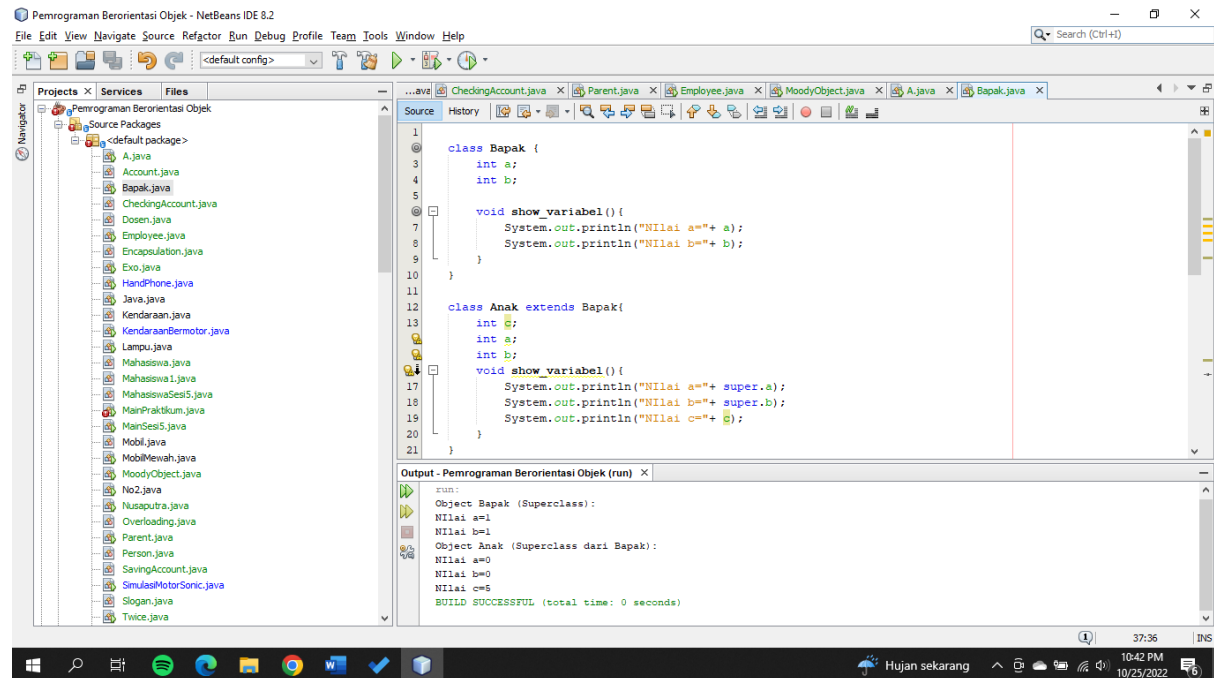


Analisis : Disini kita menghapus “public” di tiap class yang bukan sesuai nama file, karena jika memakai “public” maka harus di dalam file yang berbeda. Program ini akan menjalankan kelas yang dibuat menjalankan fungsi-fungsinya.

Percobaan 6 :

Analisis : Terapat 2 class yaitu class A sebagai parent class dan class B sebagai sub class dari A, sub class A akan mengganti nilai var_a dan var_b dari parent class nya. Ketika objek B dibuat, konstruktor A akan tetap dijalankan.

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 nya bernilai 0. Jadi objek Anak tidak akan melakukan “overirride” pada objek Bapak, selama dalan bentuk 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 mengoverride kelas parentnya. this.babyName = babyName untuk passing nilai babyName pada objek dengan parameter konstukturanya babyName.