

Nama : MUHAMAD ADAM

NIM : 20210040016

Kelas : TI21A

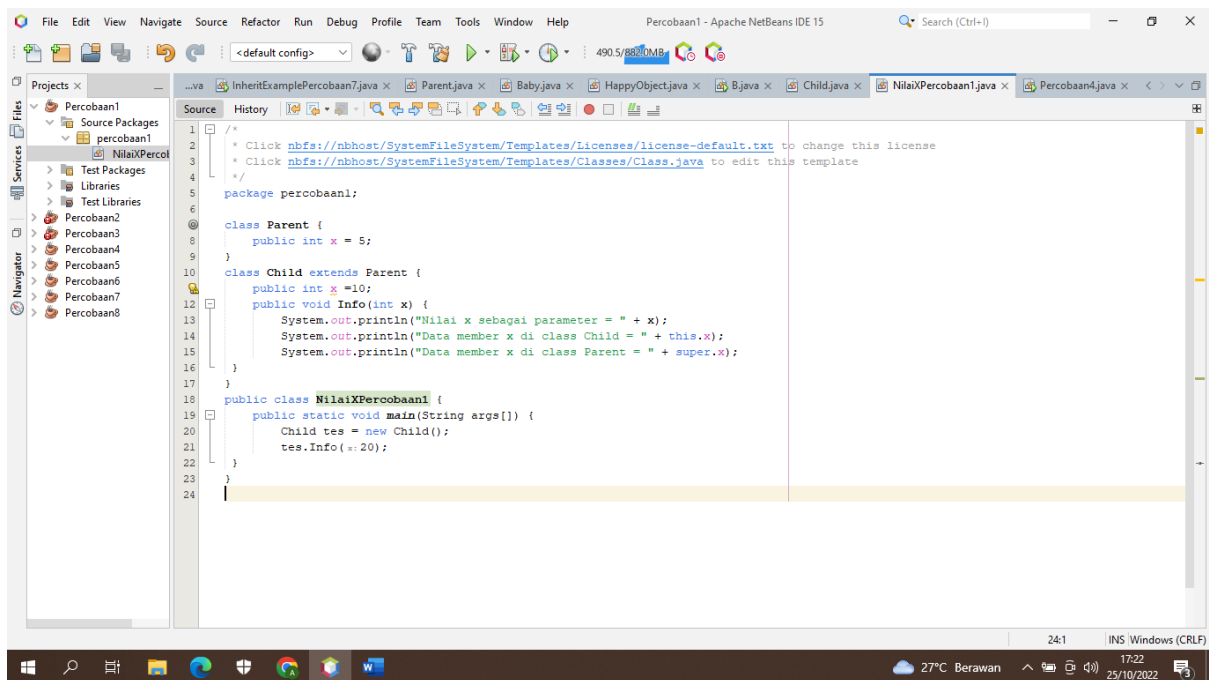
## TUGAS PRAKTIKUM

1. Kode Program upload ke github dengan nama repository praktikum-inheritance
2. Berikan analisa setiap percobaan dalam bentuk File teks pdf dan upload juga ke github praktikum-inheritance

## JAWABAN

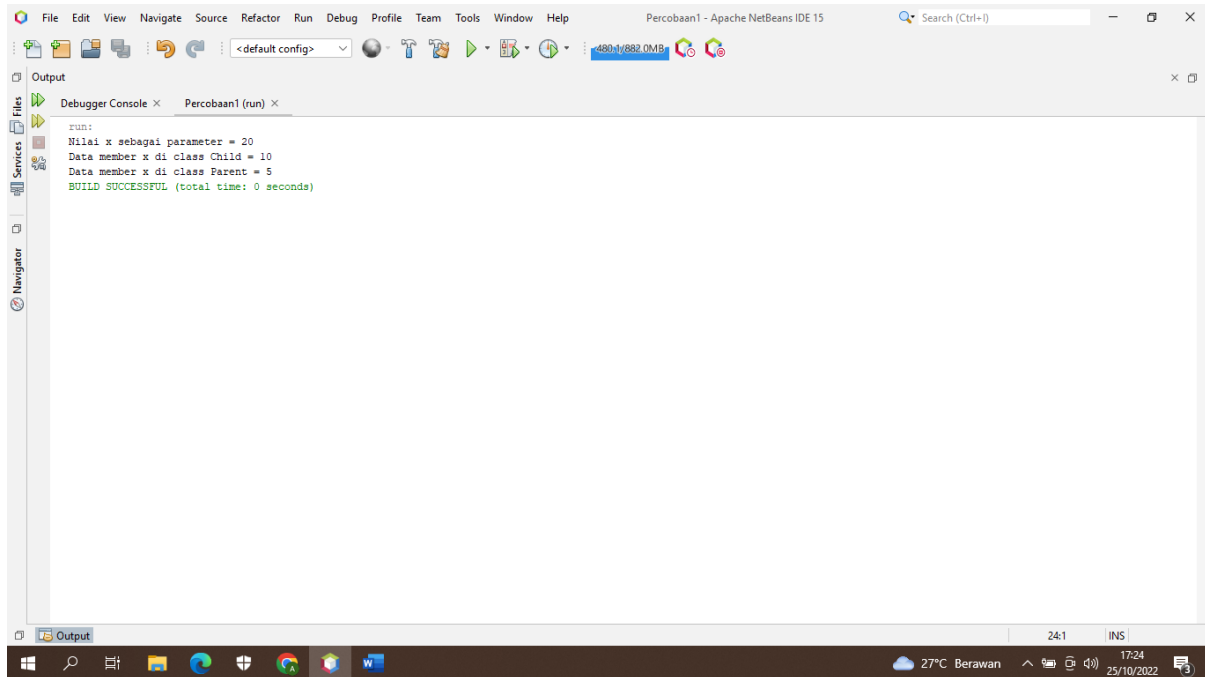
1. Percobaan berikut ini menunjukkan penggunaan kata kunci “super”.

### Input percobaan 1



```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package percobaan1;
6
7  class Parent {
8      public int x = 5;
9  }
10 class Child extends Parent {
11     public int x = 10;
12     public void Info(int x) {
13         System.out.println("Nilai x sebagai parameter = " + x);
14         System.out.println("Data member x di Class Child = " + this.x);
15         System.out.println("Data member x di Class Parent = " + super.x);
16     }
17 }
18
19 public class NilaiXPercobaan1 {
20     public static void main(String args[]) {
21         Child tes = new Child();
22         tes.Info(20);
23     }
24 }
```

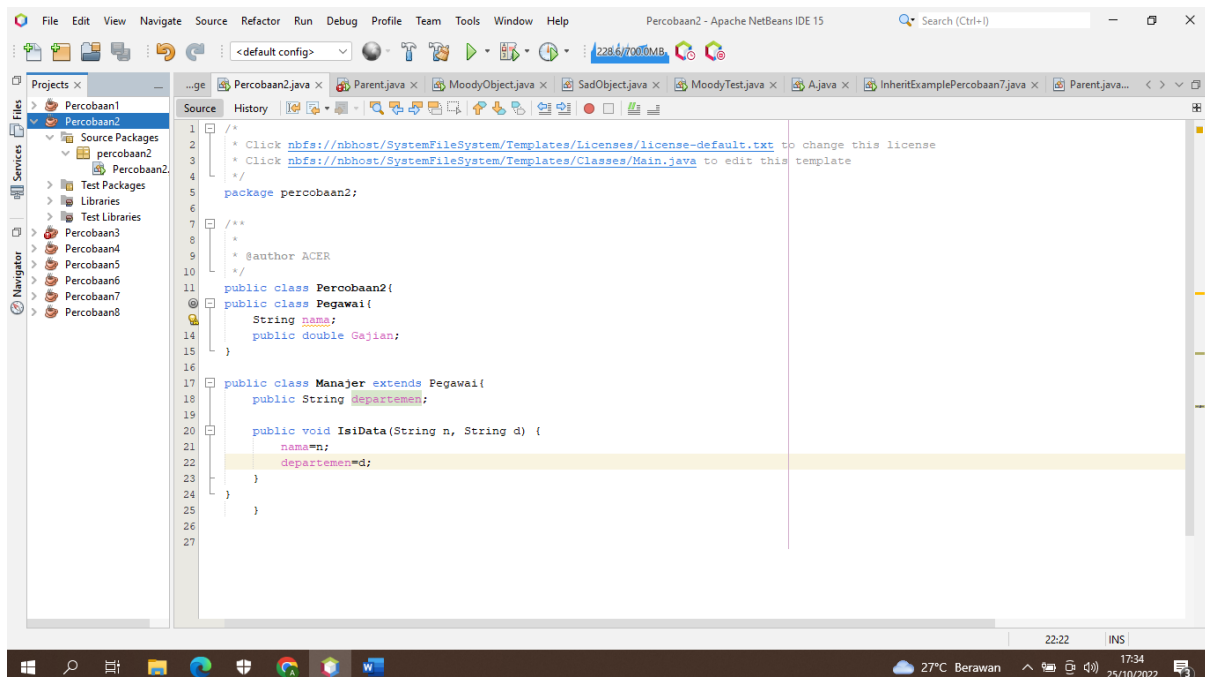
### Output percobaan 1



Pada percobaan pertama ini class parent sebagai induk class yang memiliki atribut integer = 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

## 2. Percobaan 2

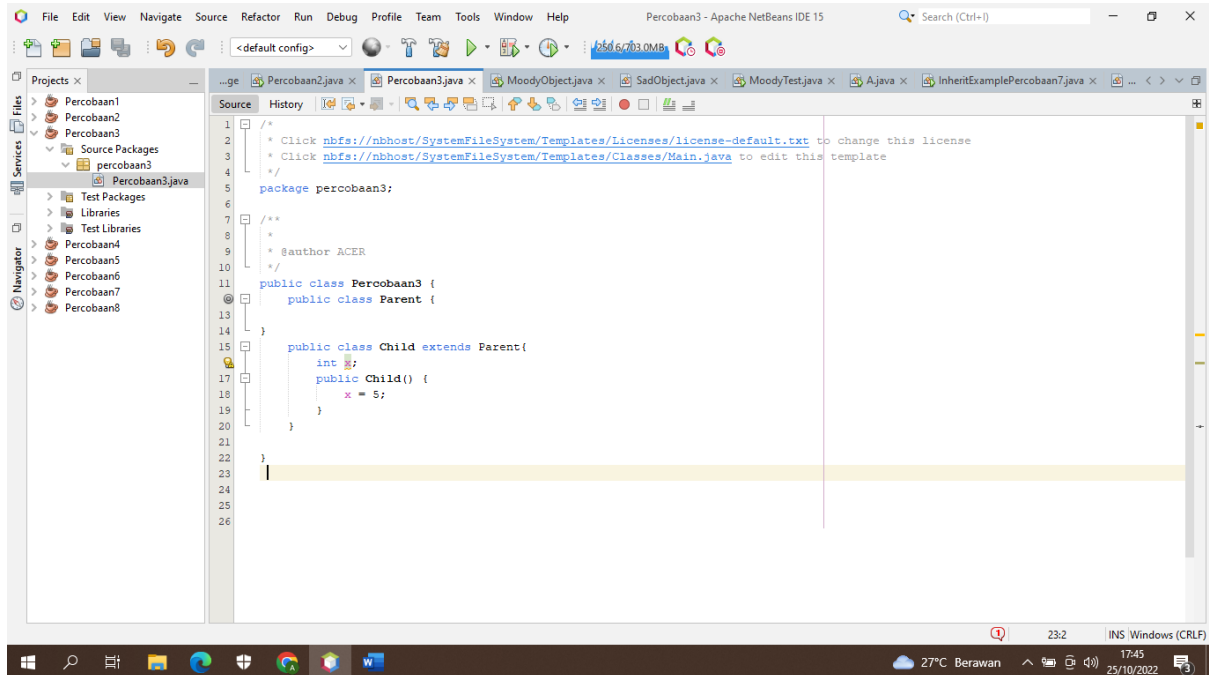
Percobaan berikut ini menunjukkan penggunaan kontrol akses terhadap atribut parent class. Mengapa terjadi error, dan bagaimana solusinya?



Solusi yang saya lakukan yaitu menghapus “public” sebelum kalimat class yang awalnya “public class Pegawai” menjadi class Pegawai begitupun dengan class Manajer dan saya juga menambahkan String nama; agar nama=n; di public void isiData tidak terjadi error.

### 3. Percobaan 3

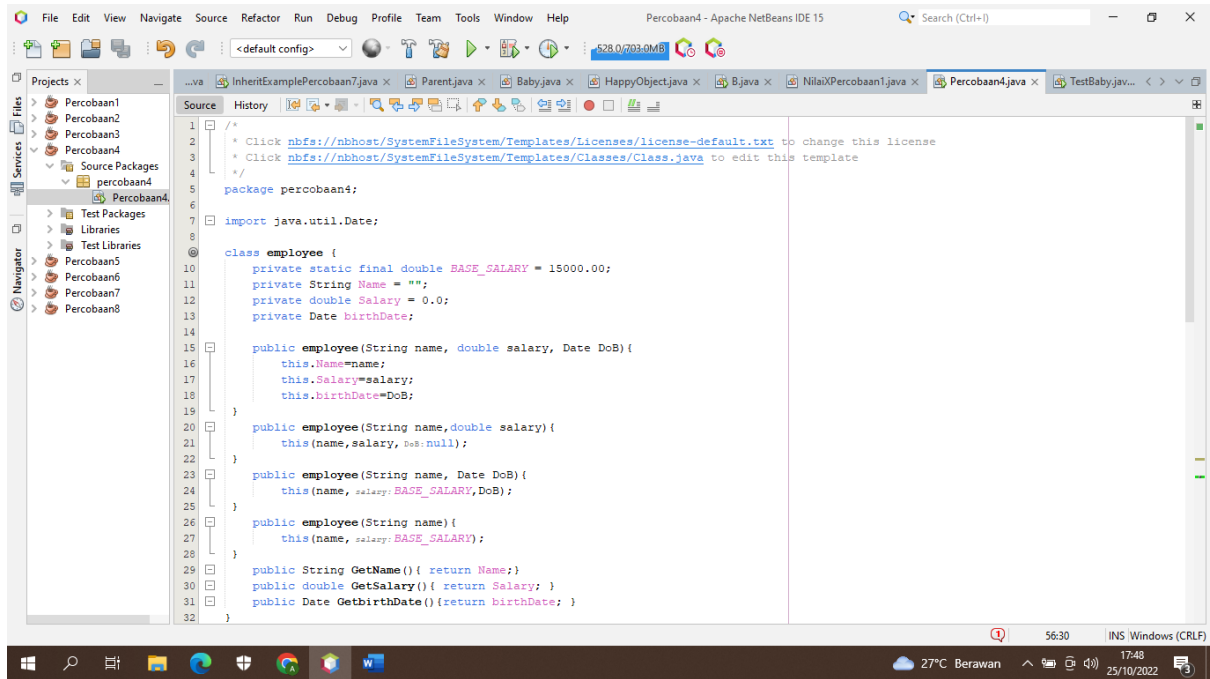
Percobaan berikut ini menunjukkan penggunaan konstruktor yang tidak diwariskan. Mengapa terjadi error, dan bagaimana solusinya?

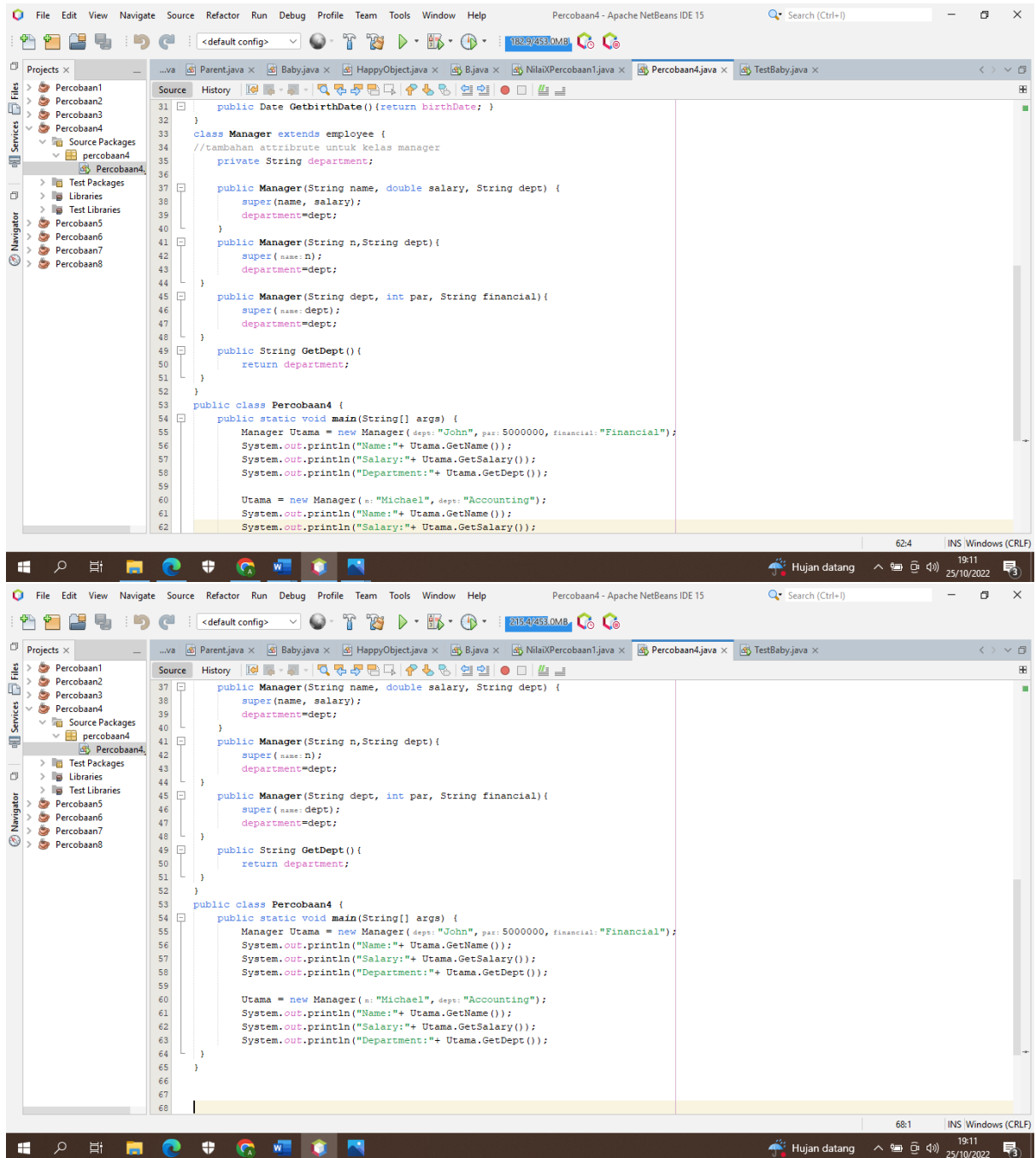


Solusinya saya menambahkan class utama yaitu percobaan3, kenapa costruktur terjadi error karena konstruktor tersebut berada di subclass

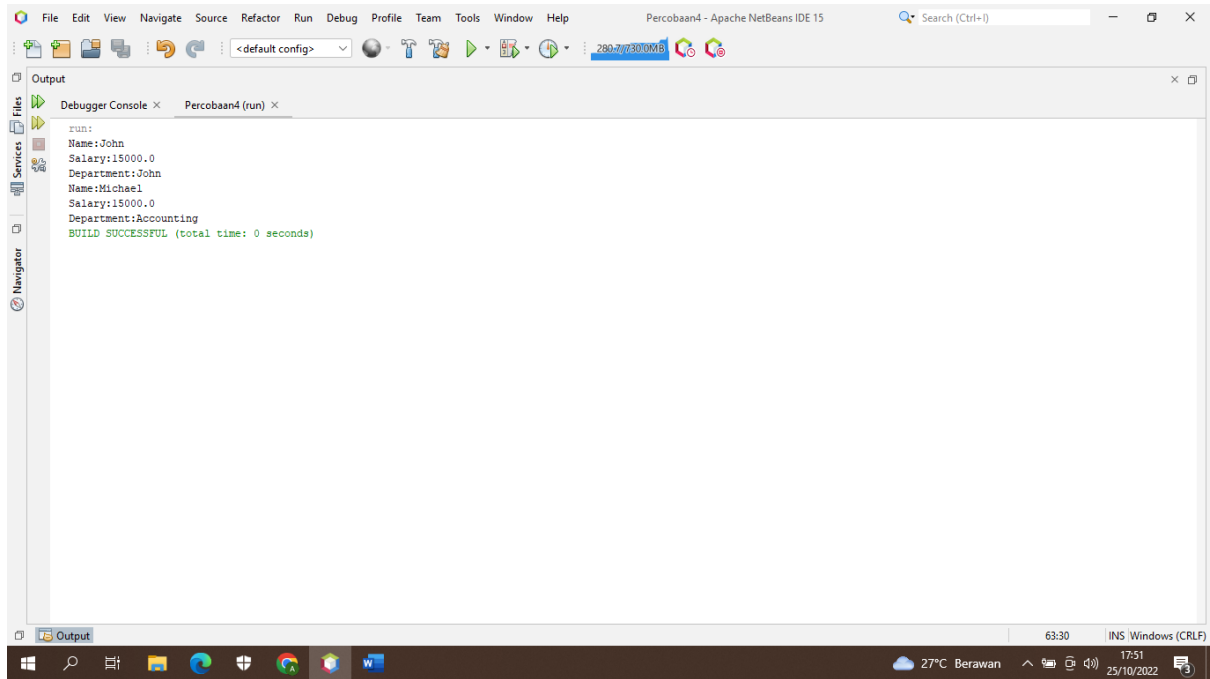
### 4. Percobaan 4

Percobaan berikut ini menunjukkan penggunaan kelas Employee dan subkelas Manager yang merupakan turunannya. Kelas TestManager digunakan untuk menguji Input





Output

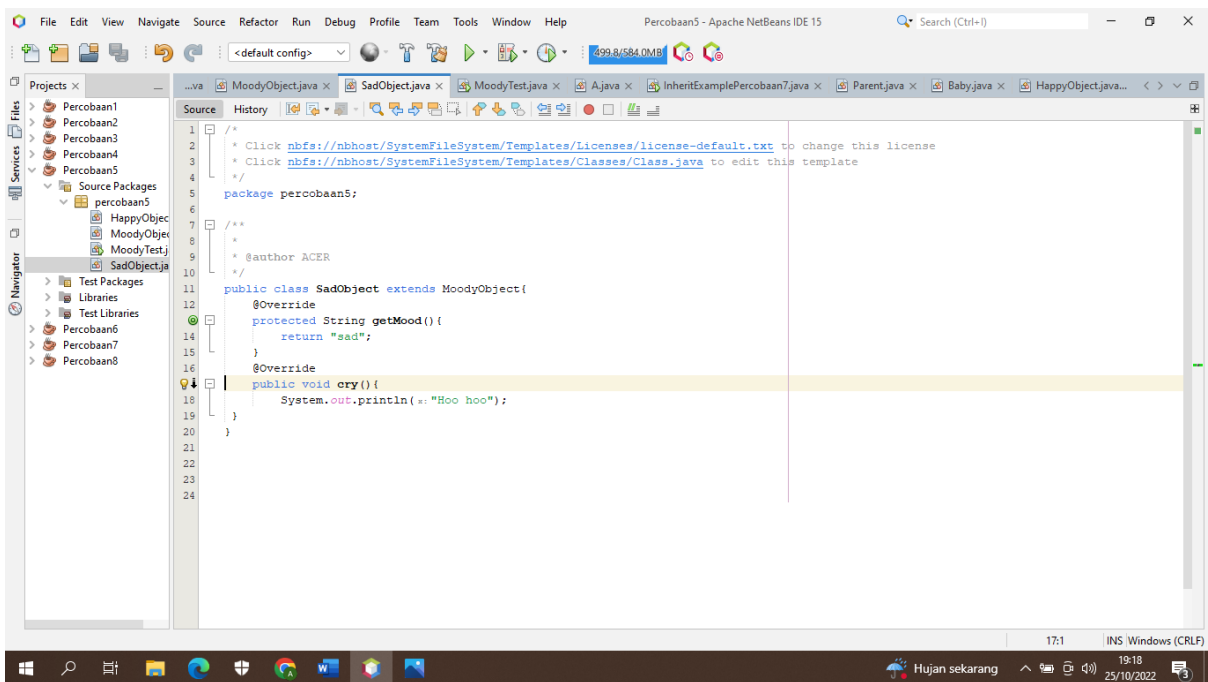
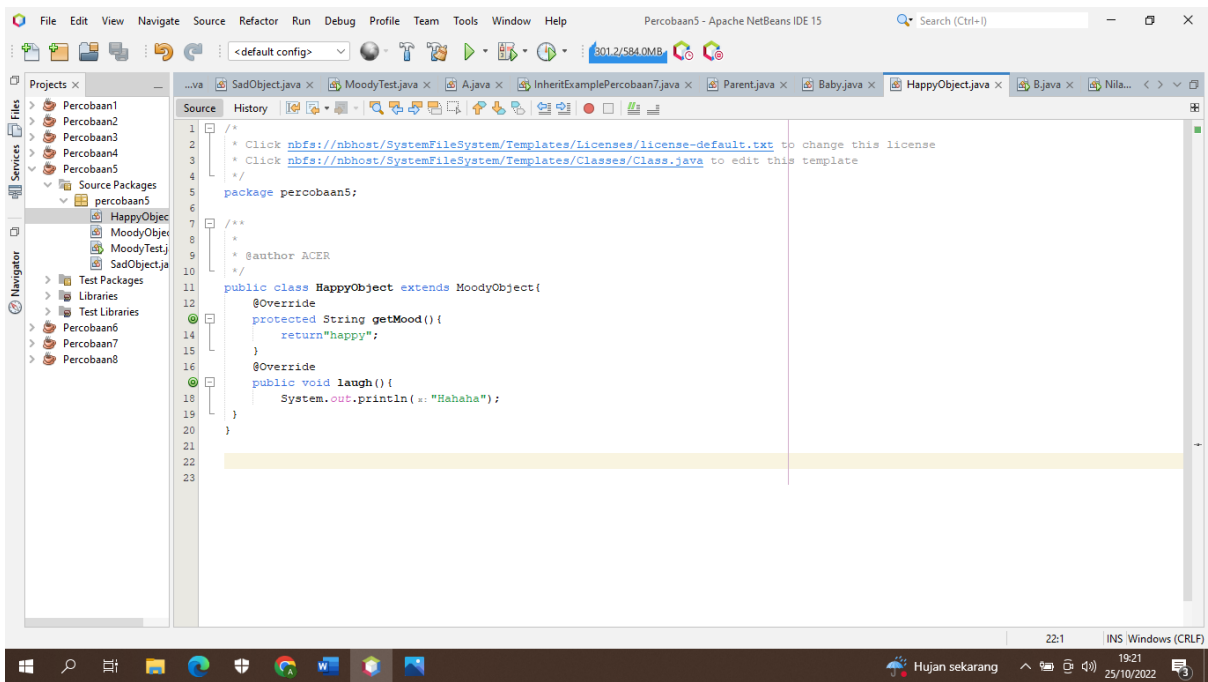


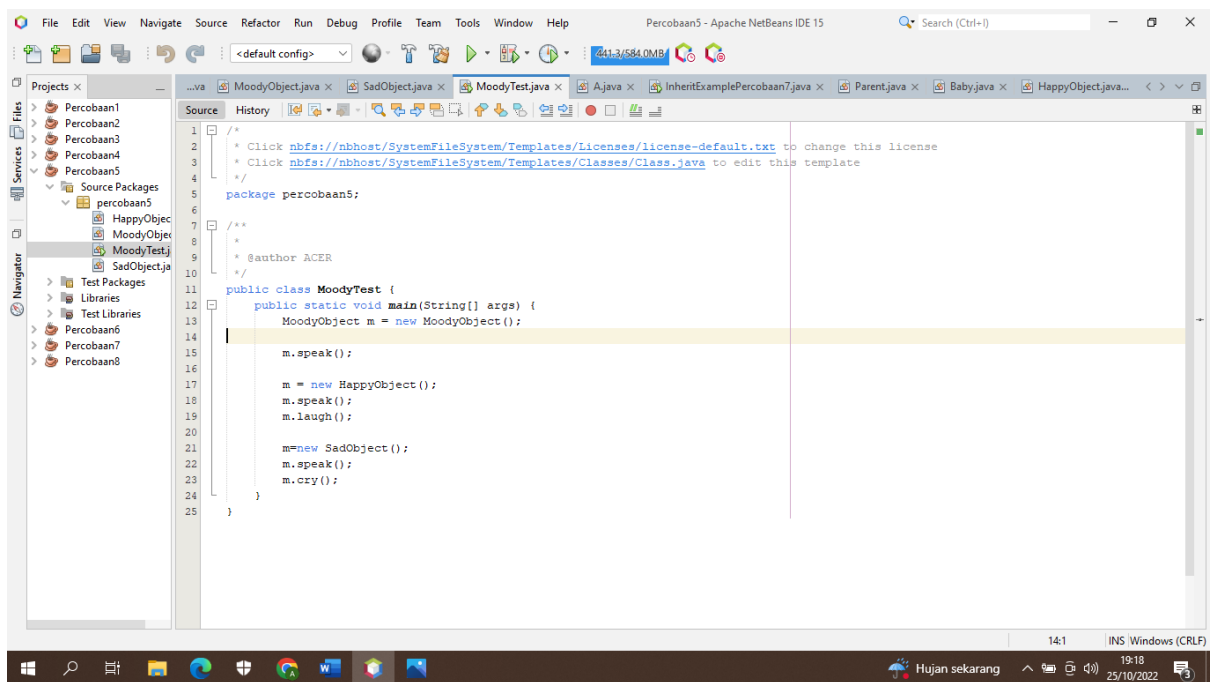
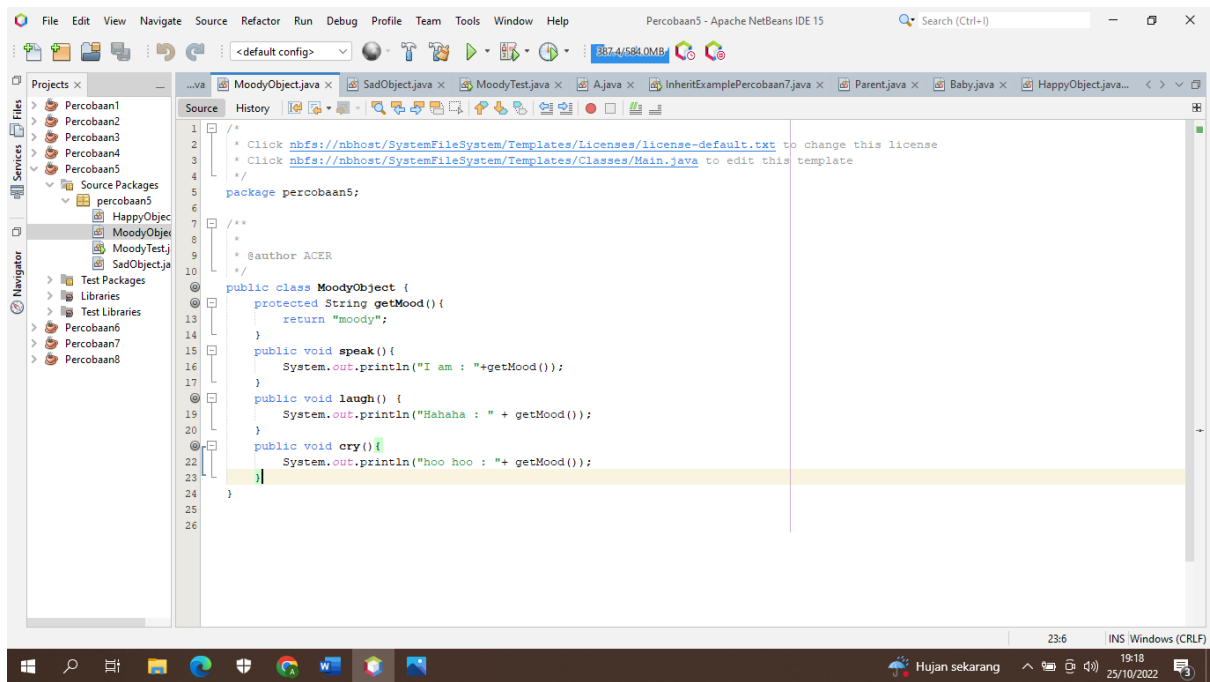
## 5. Percobaan 5

Percobaan berikut ini menunjukkan penggunaan kelas MoodyObject dengan subkelas HappyObject dan SadObject. Kelas MoodyTest digunakan untuk menguji kelas dan subkelas.

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

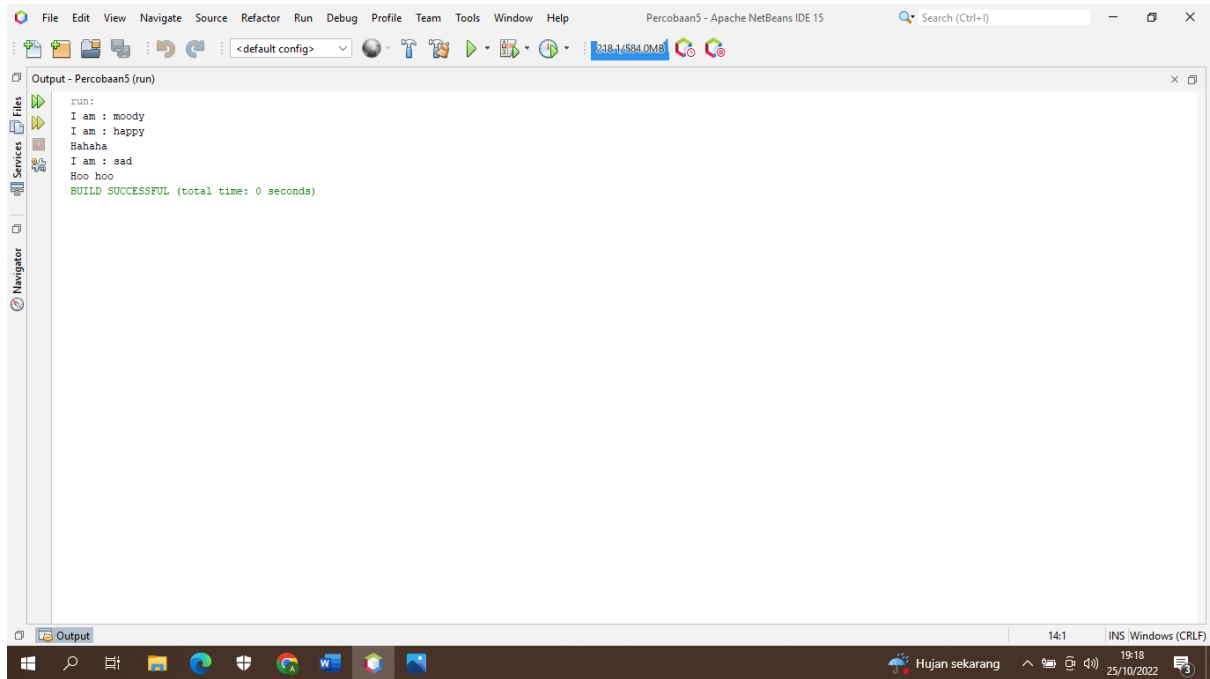
Input





Output



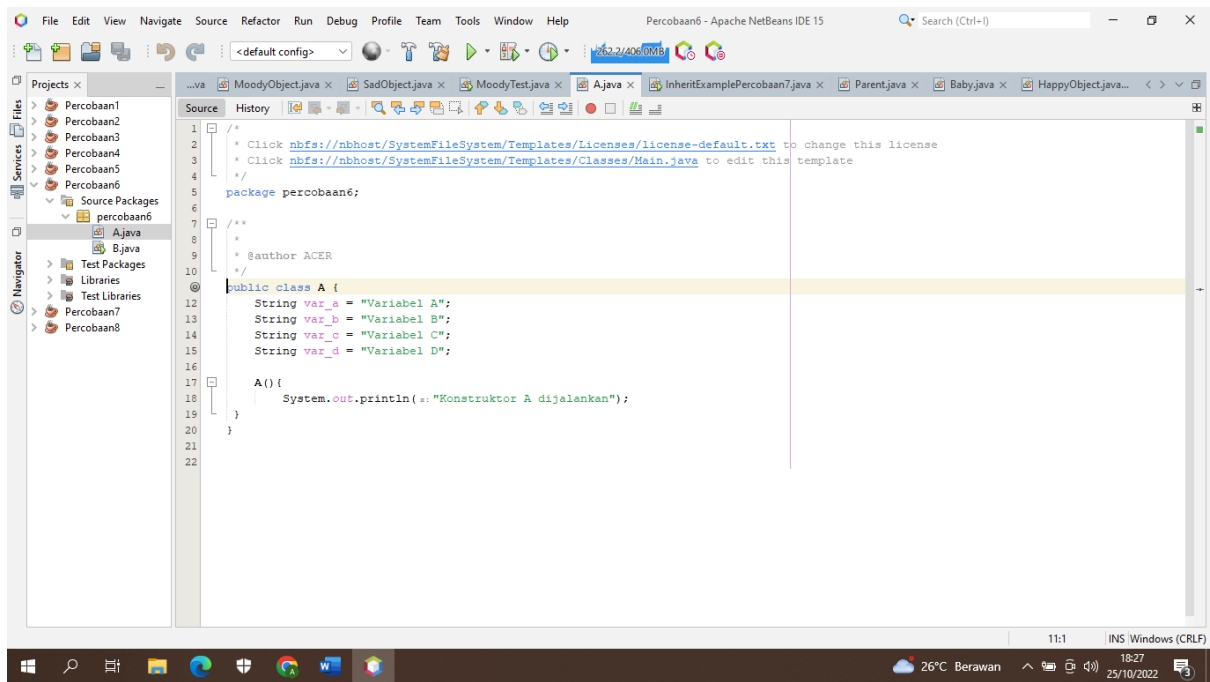


saya memperbaikinya hanya dengan menghapus public di tiap subclass.

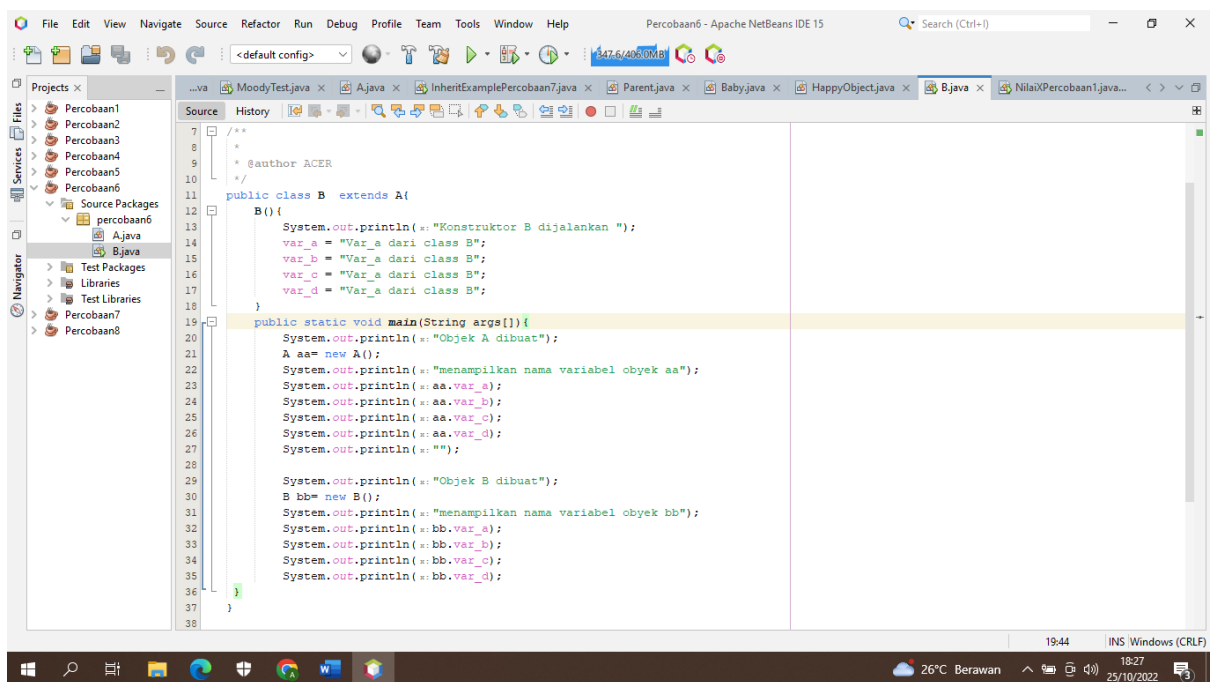
## 6. Percobaan 6

Percobaan berikut ini menunjukkan penggunaan kelas A dan dengan subkelas B. Simpan kedua kelas ini dalam 2 file yang berbeda (A.java dan B.java) dan dalam satu package. Perhatikan proses pemanggilan konstruktor dan pemanggilan variabel

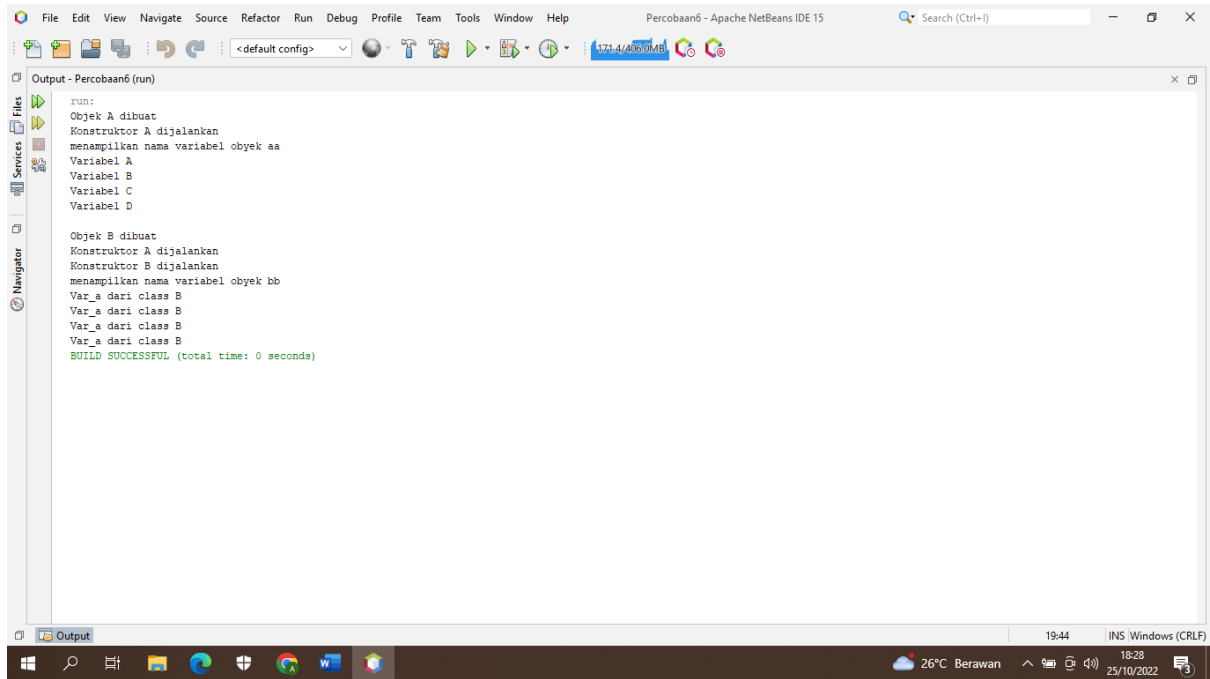
Input class A.java



## Input class B.java



## Output



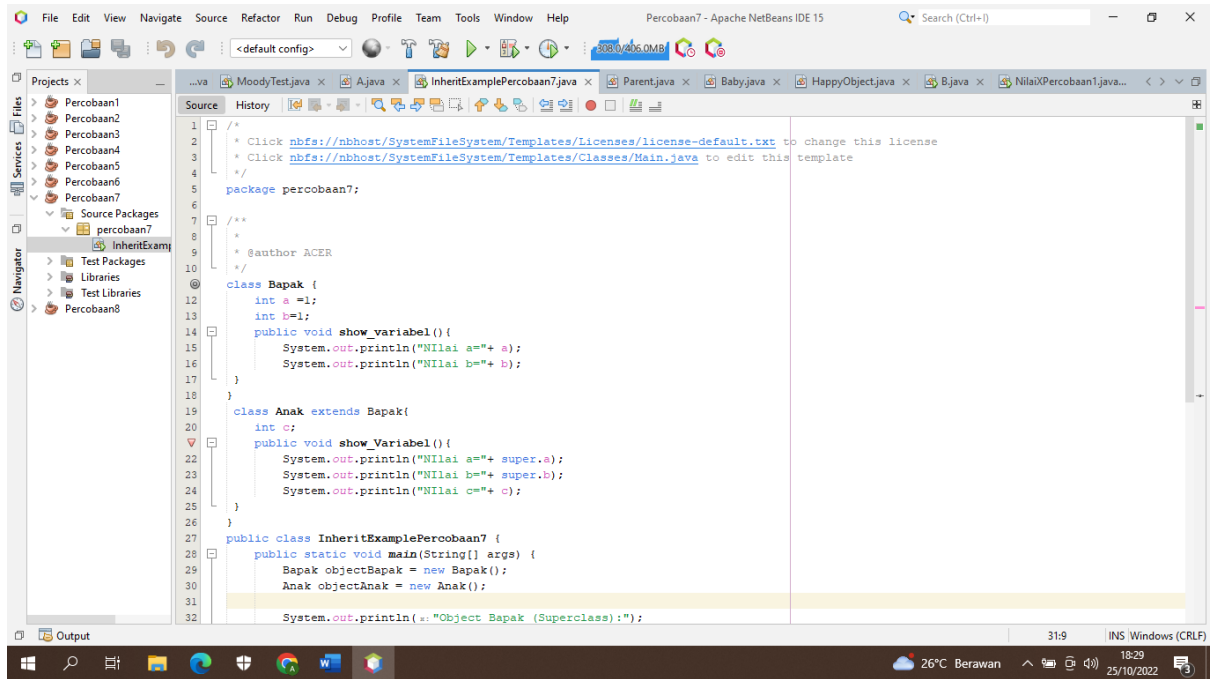
Saya memisahkan subclass B dari kelas A tanpa mengubah apapun dan Cuma sekedar drag and drop dan saya juga sudah memperhatikan konstruktornya.

## 7. Percobaan 7

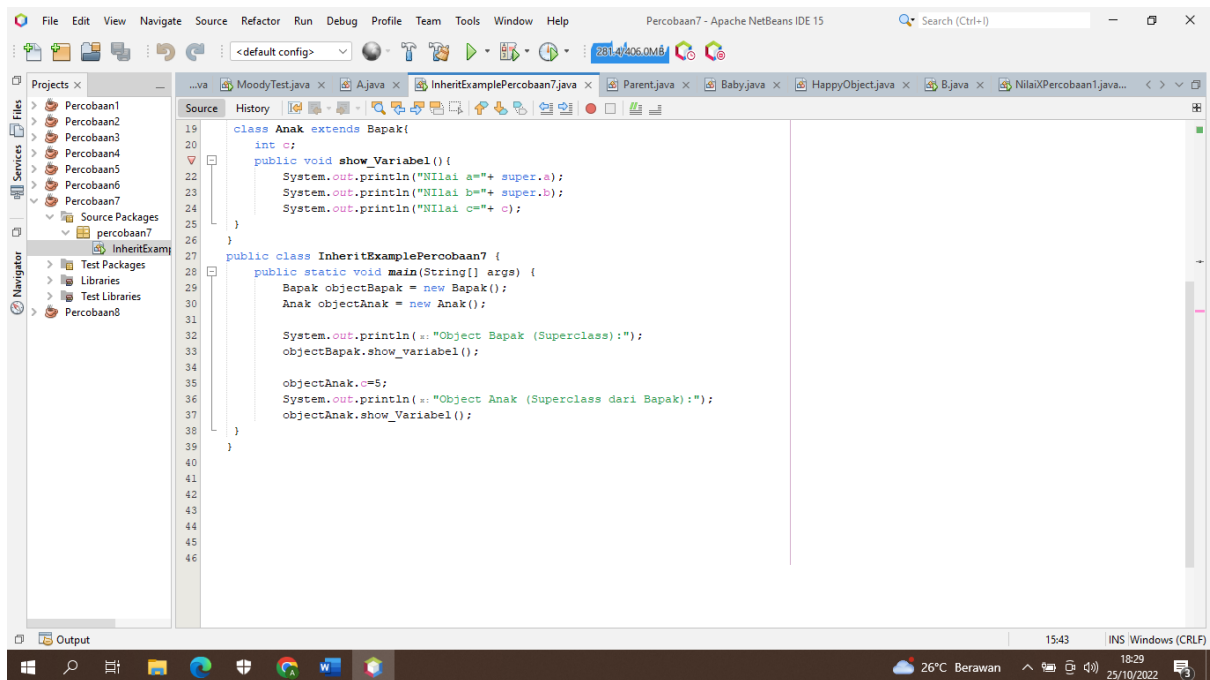
Percobaan berikut ini menunjukkan penggunaan Inheritance dan Overriding method pada kelas Bapak dan subkelas Anak. Terjadi override pada method show\_variabel. Perhatikan perubahan nilai pada variabel a, b, dan c.

Kemudian lakukan modifikasi pada method show\_variabel() pada class Anak. Gunakan super untuk menampilkan nilai a dan b (memanfaatkan method yang sudah ada pada superclass).

input

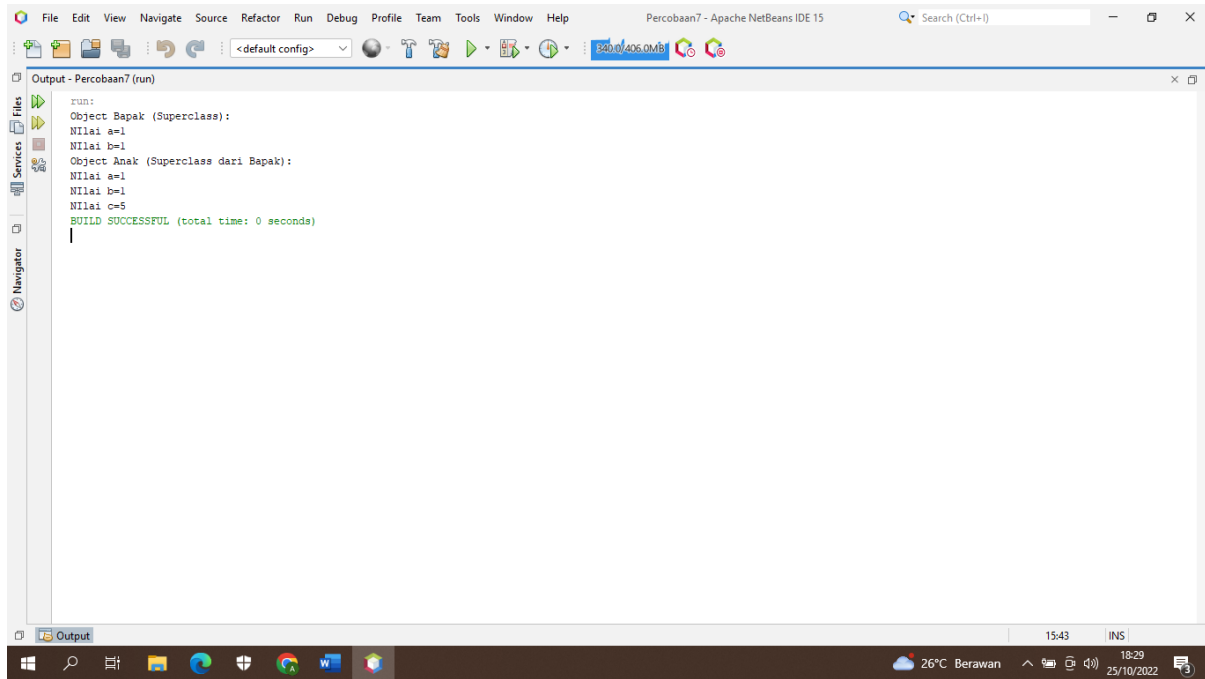


```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package percobaan7;
6
7  /**
8   *
9   * @author ACER
10  */
11  @
12  class Bapak {
13      int a = 1;
14      int b = 1;
15      public void show_variabel() {
16          System.out.println("Nilai a=" + a);
17          System.out.println("Nilai b=" + b);
18      }
19  }
20
21  class Anak extends Bapak {
22      int c;
23      public void show_Variabel() {
24          System.out.println("Nilai a=" + super.a);
25          System.out.println("Nilai b=" + super.b);
26          System.out.println("Nilai c=" + c);
27      }
28  }
29
30  public class InheritExamplePercobaan7 {
31      public static void main(String[] args) {
32          Bapak objectBapak = new Bapak();
33          Anak objectAnak = new Anak();
34
35          System.out.println("Object Bapak (Superclass):");
```



```
19  class Anak extends Bapak {
20      int c;
21      public void show_Variabel() {
22          System.out.println("Nilai a=" + super.a);
23          System.out.println("Nilai b=" + super.b);
24          System.out.println("Nilai c=" + c);
25      }
26  }
27
28  public class InheritExamplePercobaan7 {
29      public static void main(String[] args) {
30          Bapak objectBapak = new Bapak();
31          Anak objectAnak = new Anak();
32
33          System.out.println("Object Bapak (Superclass):");
34          objectBapak.show_variabel();
35
36          objectAnak.c = 5;
37          System.out.println("Object Anak (Superclass dari Bapak):");
38          objectAnak.show_Variabel();
39      }
40  }
41
42
43
44
45
46
```

Output

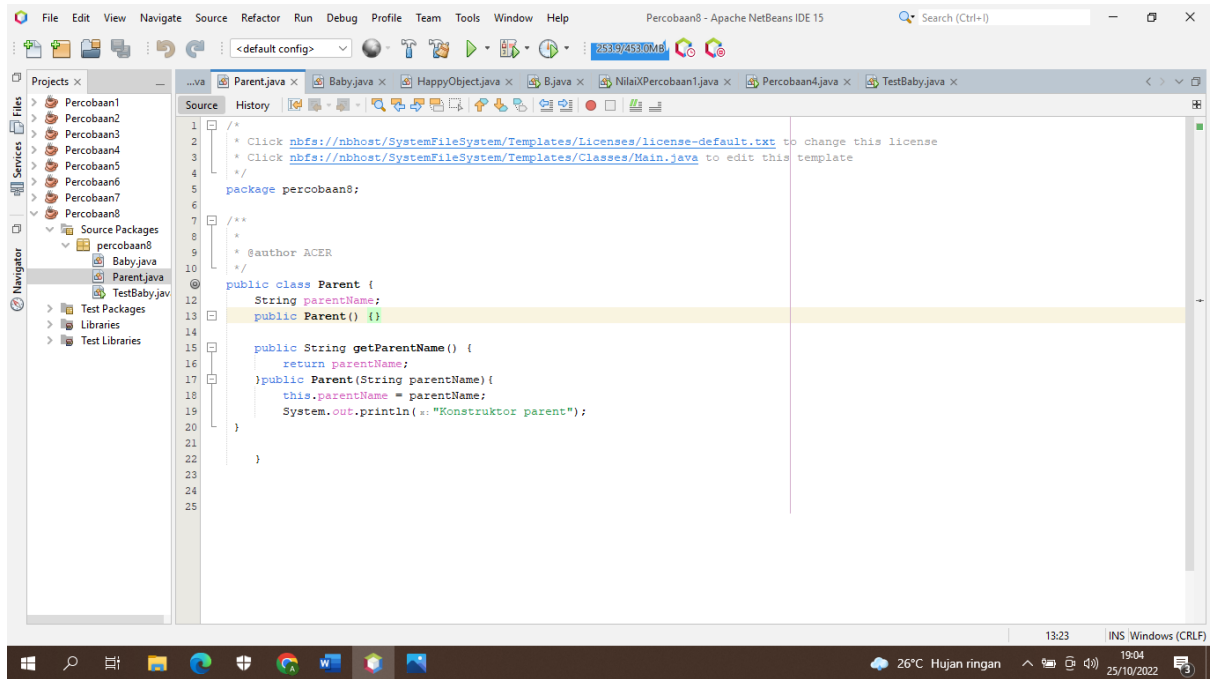


Perbedaan dengan yang sebelum diubah yaitu pada subclass anak nilai a,b yang mewarisi nilai bapak dan c yaitu nilai dari objek sianak atau buka nilai warisan.

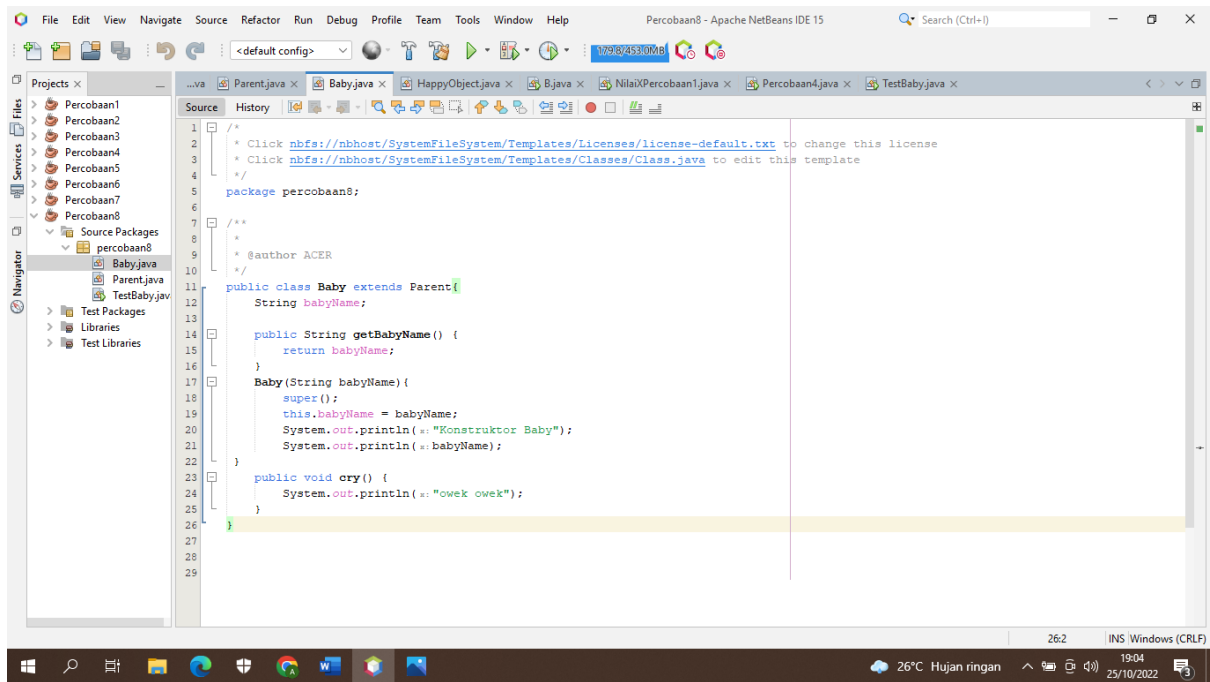
## Percobaan 8

Percobaan berikut ini menunjukkan penggunaan overriding method pada kelas Parent dan subkelas Baby, saat dilakukan pemanggilan konstruktor superclass dengan menggunakan super.

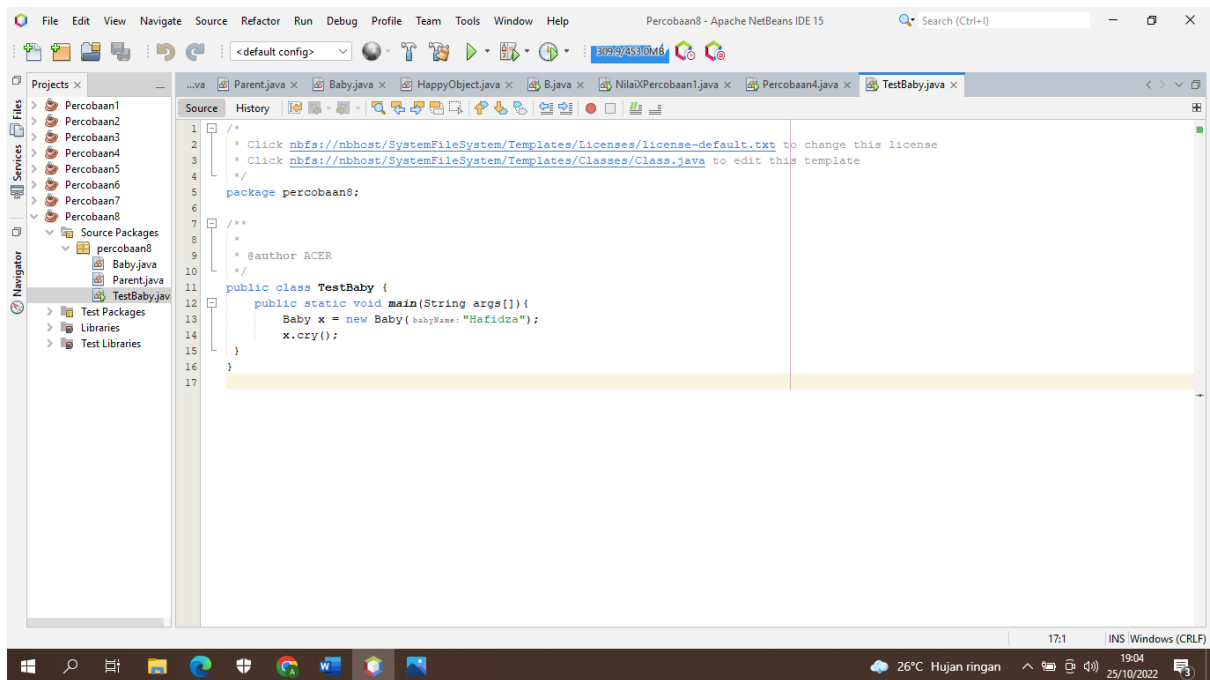
Input parent



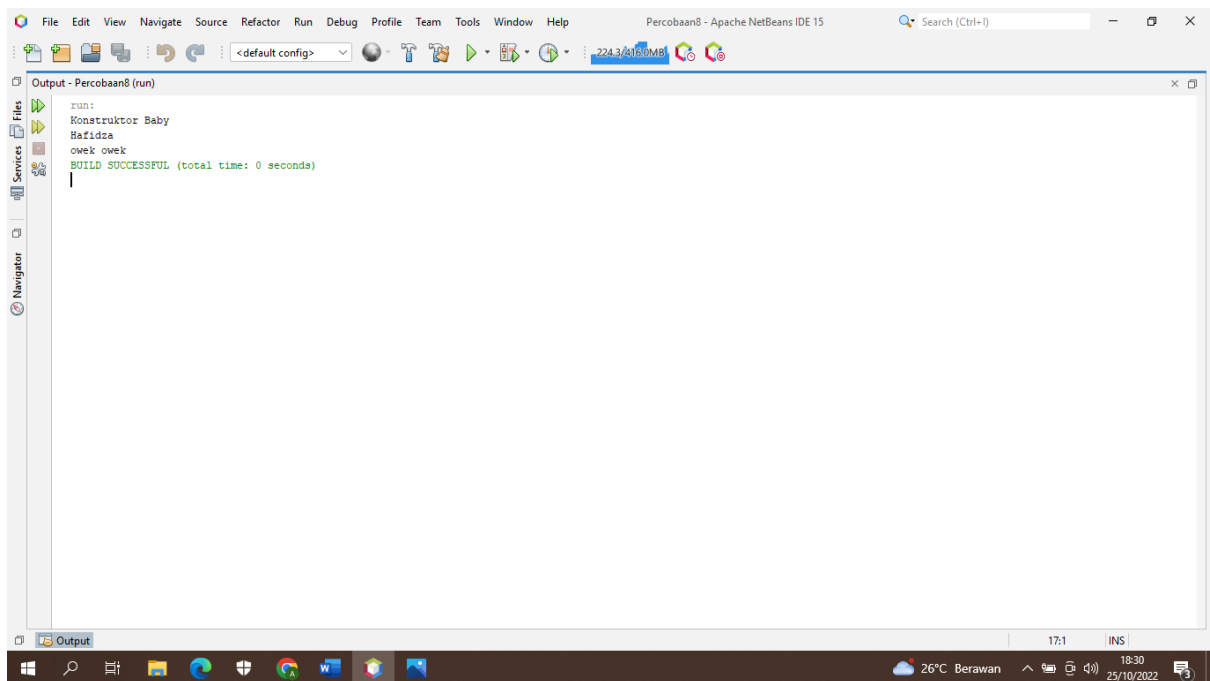
## Input baby



## input testbaby



## Output



Dengan begitu programnya berjalan dengan lancar