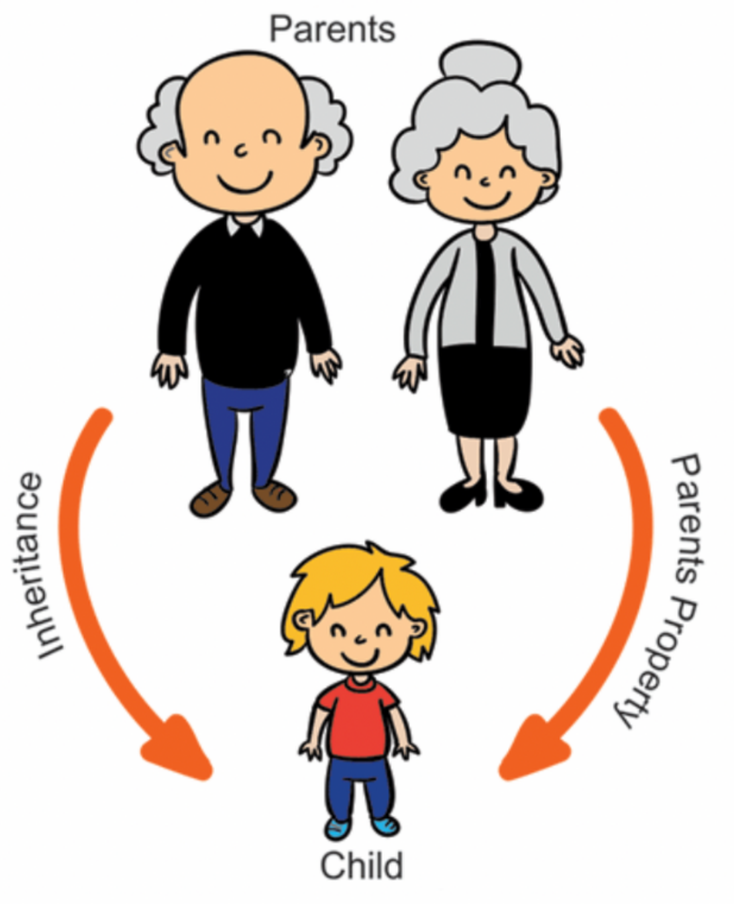
Bab 3
Inheritance

Pengertian Inheritance atau pewarisan

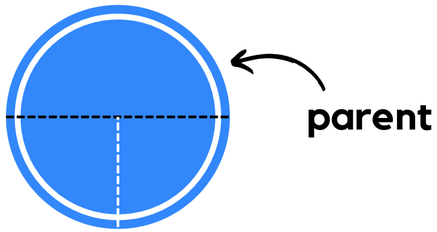
Inheritance atau Pewarisan/Penurunan adalah konsep pemrograman dimana sebuah class dapat '**menurunkan**" **properti** dan **method** yang dimilikinya kepada **class lain**. Konsep inheritance ialah membuat sebuah struktur atau '***hierarchy***' class dalam kode program, hal tersebut memungkinkan untuk melakukan **pewarisan attribute** atau **method** dari class yang umum ke class yang lebih **spesifik**. Konsep inheritance digunakan untuk memanfaatkan fitur ' **code reuse** ' untuk menghindari duplikasi kode program. Fungsi dari inheritance **memperluas** fungsi dari parent class.

Gambaran Konsep



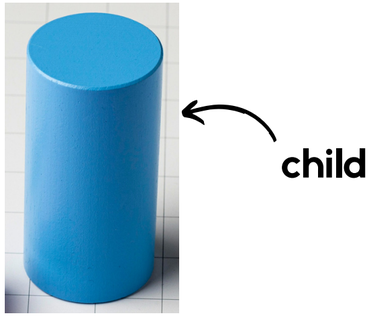
- Class yang akan '**diturunkan**' bisa disebut sebagai class induk (parent class), super class, atau base class.
- Sedangkan class yang '**menerima** penurunan' bisa disebut sebagai class anak (child class), sub class, derived class .

Gambaran Konsep



circle

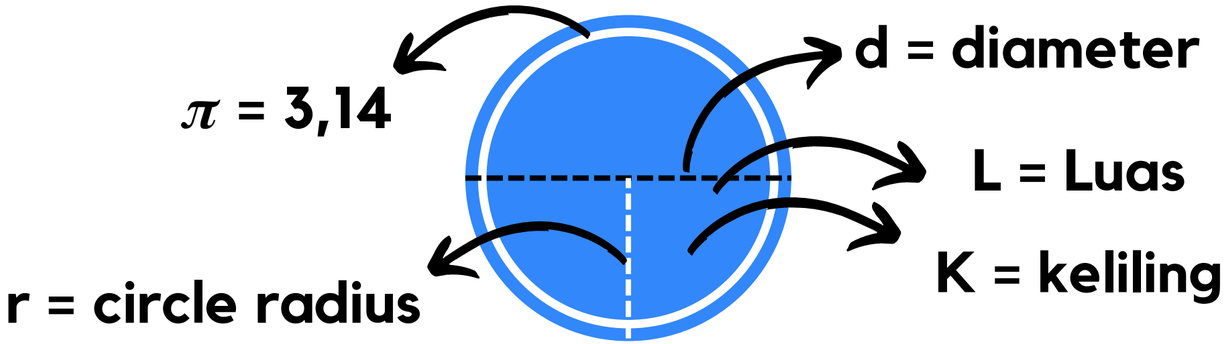
Inheritance

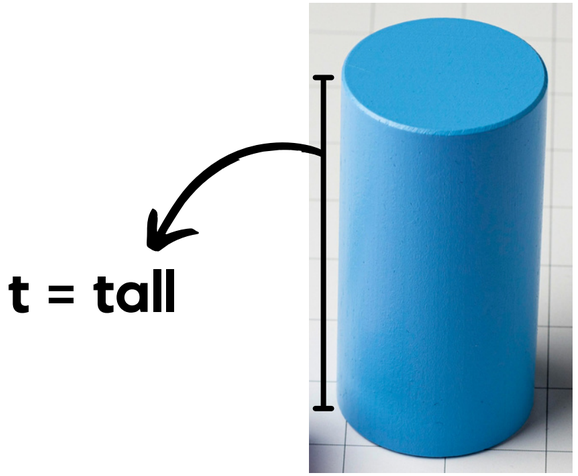


tube

- Class yang akan '**diturunkan**' bisa disebut sebagai class induk (parent class), super class, atau base class.
- Sedangkan class yang '**menerima** penurunan' bisa disebut sebagai class anak (child class), sub class, derived class .

Gambaran Konsep pada Lingkaran





circle

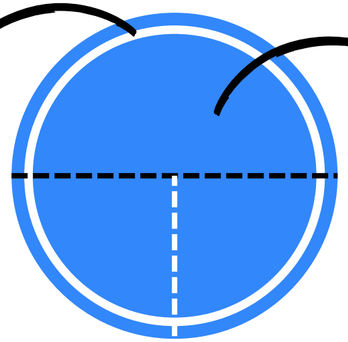
Inherit

tube

Lingkaran akan mewariskan
beberapa atributnya
π , r , d , luas,kelilng
yang akan digunakan pada bangun
ruang tabung

Gambaran Konsep pada Lingkaran

HitcircleAre
a = π * r^2



Hitkeliling = 2 * π *
r

HitVolume
=
HitcirleArea() * t

circle

Inherit

tube

Lingkaran akan mewariskan
beberapa method nya
circle area, keliling
yang akan digunakan pada bangun ruang
tabung untuk menghitung **Volume**

Istilah penting dalam konsep inheritance:

1. ***Super Class*** : kelas induk yang mewariskan atribut dan method kepada turunannya.
2. ***Sub Class* atau *Child Class*** : kelas turunan yang mewarisi atribut dan method. Sub Class dapat menambah atribut dan methodnya sendiri sebagai tambahan dari kelas yang memberi warisan.
3. ***Reusability*** : menggunakan kembali atribut dan method dari super class di sub class.

Instanceof

Instanceof adalah salah satu keyword pada Java, yang digunakan untuk **membandingkan** suatu objek, apakah instansiasi dari suatu class atau tidak, hasil dari perbandingan tersebut akan menghasilkan nilai boolean berupa nilai true atau false.

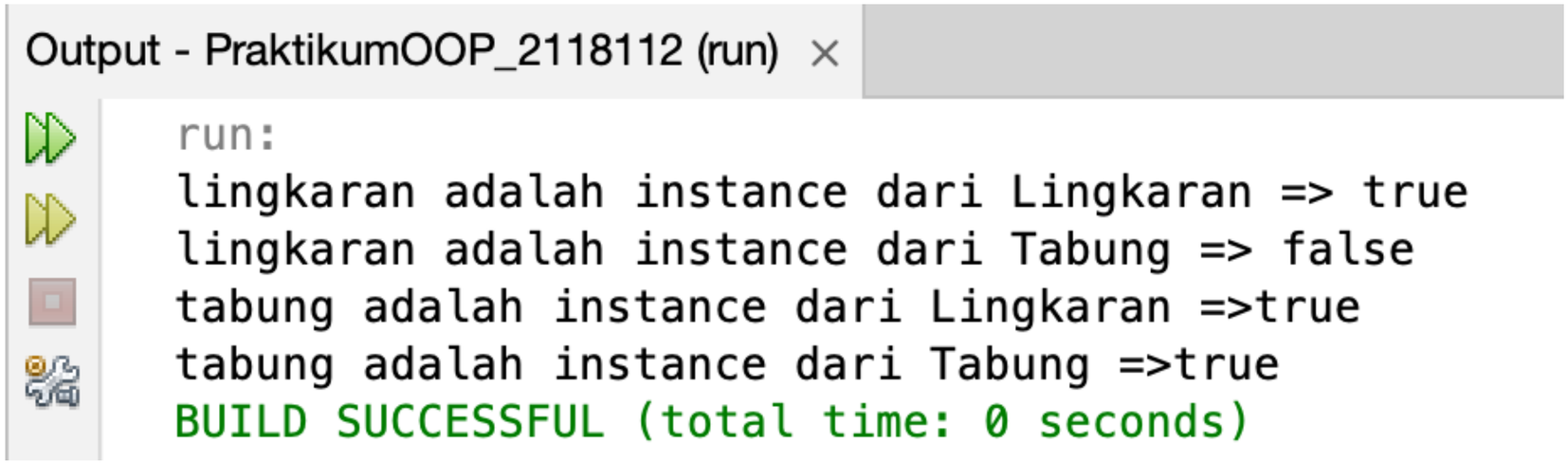
```java
public class lingkaran {
    double jariJari;
    public lingkaran(){
        this.jariJari=7.;
    }
    public double hitungLuas() {
        return 3.14 * jariJari * jariJari;
    }
}
```

```java
public class Tabung extends lingkaran {
    double tinggi;
    public Tabung() {
        this.tinggi = 10;
    }
    public double hitungVolume() {
        return hitungLuas() * tinggi;
    }
}
```

```java
public class main {
    public static void main(String[] args) {
        lingkaran lingkaran = new lingkaran();
        Tabung tabung = new Tabung();
        System.out.println("lingkaran adalah instance dari Lingkaran => " + (lingkaran instanceof lingkaran));
        System.out.println("lingkaran adalah instance dari Tabung => " + (lingkaran instanceof Tabung));
        System.out.println("tabung adalah instance dari Lingkaran =>" + (tabung instanceof lingkaran));
        System.out.println("tabung adalah instance dari Tabung =>" + (tabung instanceof Tabung));

    }
}
```

Hasil :



Analisa :

Class lingkaran merupakan parent class, Class tabung merupakan turunan (inheritance) dari Class lingkaran. Ketika membuat instans objek tabung, constructor dari class lingakran akan tetap dipanggil.

Kesimpulan:

lingkaran adalah instance dari kelas **Lingkaran**, **lingkaran** bukan instance dari kelas **Tabung**, **tabung** adalah instance dari kelas **Lingkaran** karena kelas **Tabung** adalah subclass dari kelas **Lingkaran**, **tabung** juga adalah instance dari kelas **Tabung.**

Aturan dalam konsep Inheritance

1. Penggunaan inheritance diawali dengan pembuatan super class

```
public class lingkaran {

}
```

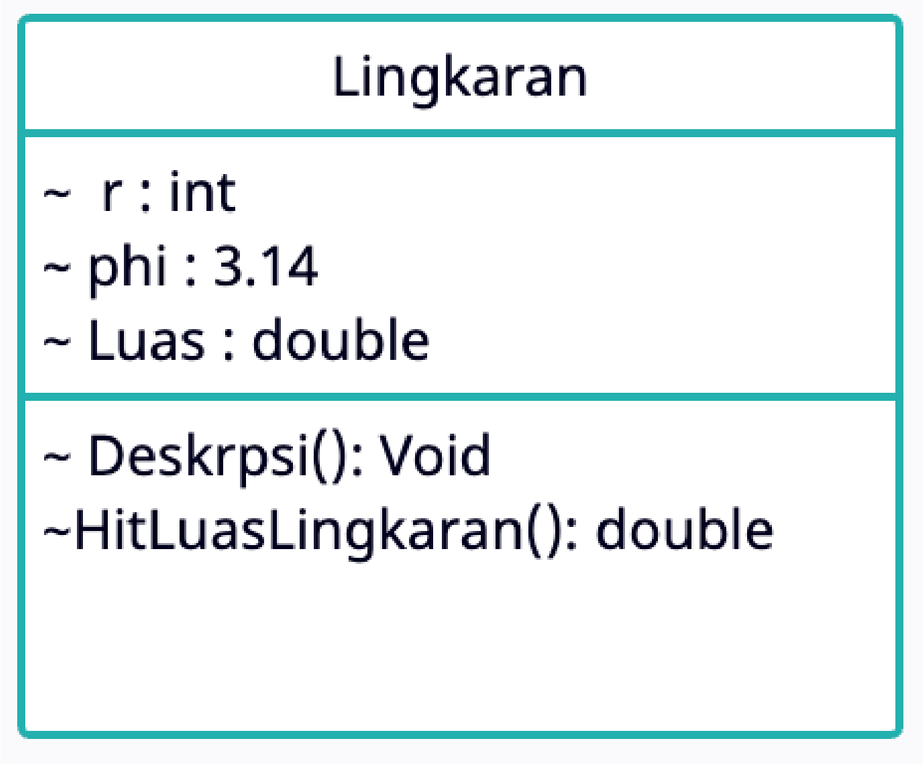
Agar kelas turunan / subclass dapat mewarisi atribut dan method dari super class harus menggunakan keyword " ***extends*** " dan diikuti dengan "nama_super_class".

```
public class Tabung extends lingkaran {

}
```

Class Diagram

Class diagram lingkaran



Implementasi Lingkaran

```
14      //attribut
15      int r;
16      double phi,luas;
17
18      //construtor
19      public Lingkaran(){
20          r = 7;
21          phi = 3.14;
22      }
23      //method
24      void Deskripsi(){
25          System.out.println( x: "Ini adalah hasil menghitung");
26      }
27      //method untuk menghitung luas
28      double HitluasLingkaran(){
29      //L = πr2
30          luas=(phi*r*r);
31          return luas;
32      }
33  }
```

Implementasi Lingkaran

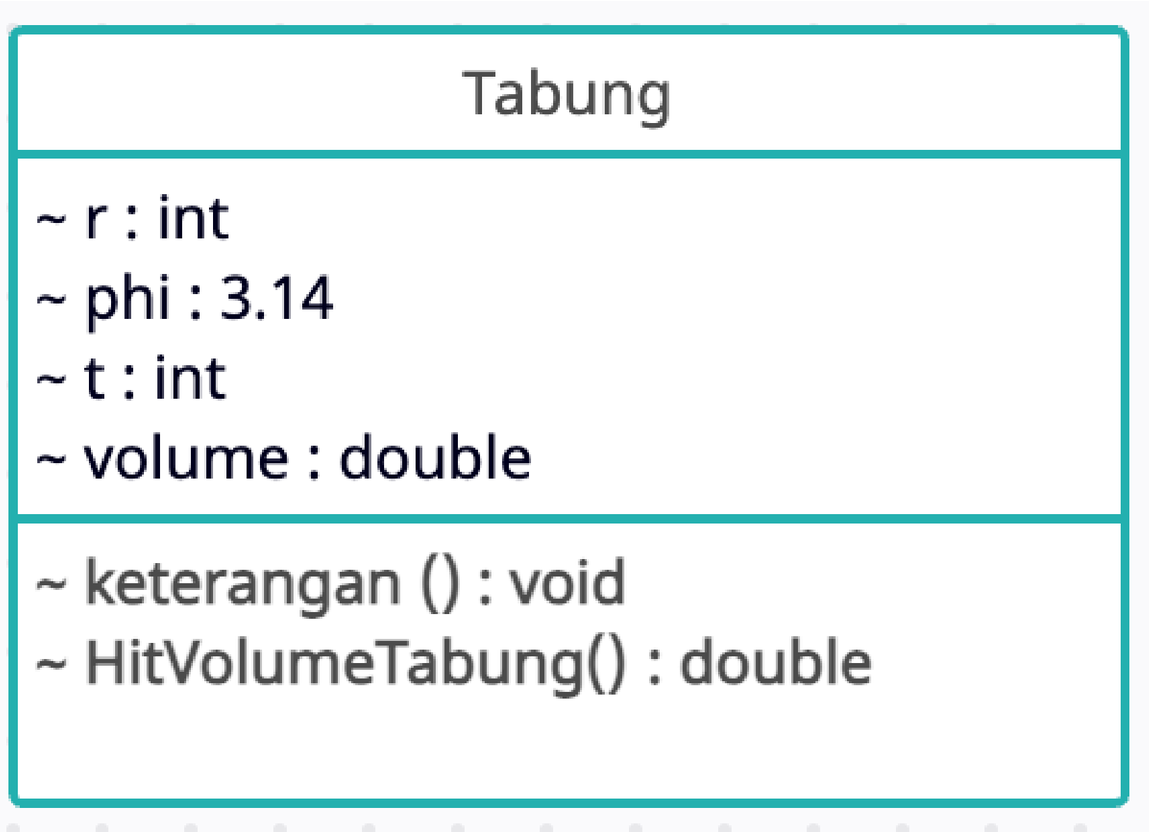
```
11  public class mainLingkaran {
12
13      public static void main(String[] args) {
14          Lingkaran lkr = new Lingkaran(); // memanggil constructor
15
16          lkr.Deskripsi();
            System.out.println("Hasilnya adalah :" + lkr.HitluasLingkaran());
18      }
19  }
```

Output - PraktikumOOP_2118112 (run)

```
run:
Ini adalah hasil menghitung
Hasilnya adalah :153.86
BUILD SUCCESSFUL (total time: 0 seconds)
```

Class Diagranm

Class diagram Tabung



Implementasi Tabung

```
11  public class Tabung extends Lingkaran {
12
13      int t;
        double volume, luasPermukaan;
15
16      public Tabung() {
17  //      s
18          t = 20;
19      }
20
21      void Keterangan() {
22          Deskripsi();
23          System.out.println( x:"mengitung Volume Tabung");
24      }
25
        double HitvolumeTabung() {
27          //π x r2 x t.
28          volume = ((phi * r * r) * t);
29          return volume;
30      }
31  }
```

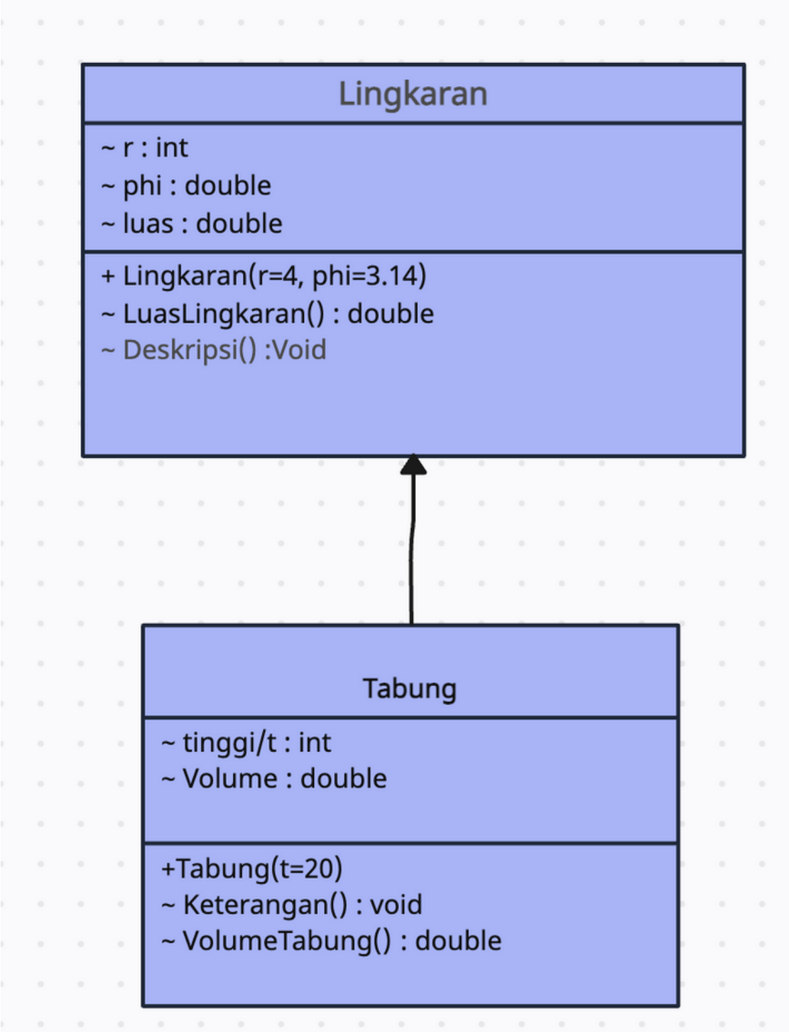
Implementasi Tabung

```
11  public class mainTabung {
12
13      public static void main(String[] args) {
14          Tabung tbg = new Tabung();
15          tbg.Keterangan();
16          System.out.println("volumenya adalah :" + tbg.HitvolumeTabung());
17  //          System.out.println("Luas Permukaan Tsabung adalah :"+tbg.luasPermukaanTabung());
18
19      }
20  }
```

Output - PraktikumOOP_2118112 (run)

```
run:
mengitung Volume Tabung
volumenya adalah :3077.2000000000003
BUILD SUCCESSFUL (total time: 0 seconds)
```

Class diagram



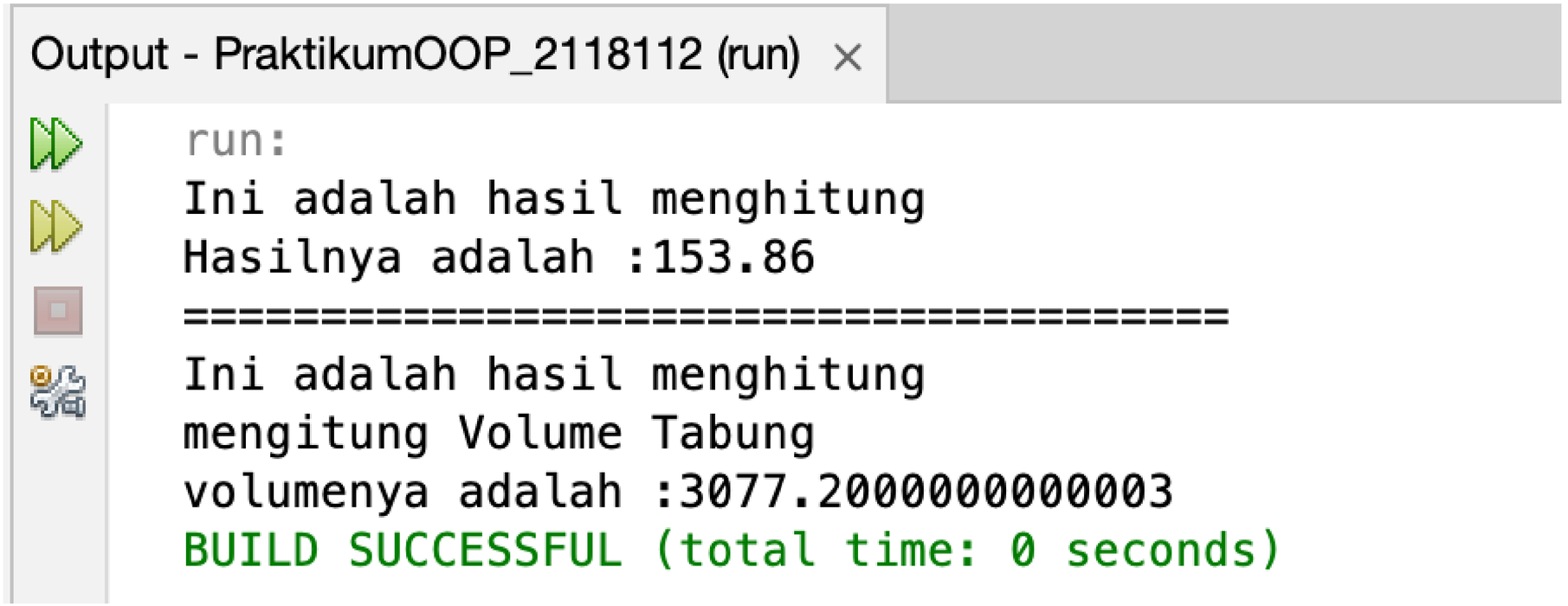
Implementasi extends

```
11  public class Tabung extends Lingkaran {
12
13      int t;
        double volume, luasPermukaan;
15
16      public Tabung() {
17          t = 20;
18      }
19
20      void Keterangan() {
21          Deskripsi();
22          System.out.println( x:"mengitung Volume Tabung");
23      }
24
25      double HitvolumeTabung() {
26          //π x r2 x t.
27          volume = (HitluasLingkaran() * t);
28          return volume;
29      }
30  }
```

Implementasi extends

```
11  public class mainTabung {
12
13      public static void main(String[] args) {
14          Lingkaran lkr = new Lingkaran(); // memanggil constructor
15
16          lkr.Deskripsi();
17          System.out.println("Hasilnya adalah :" + lkr.HitluasLingkaran());
18          System.out.println( x:"=====================================");
19          Tabung tbg = new Tabung();
20          tbg.Keterangan();
21          System.out.println("volumenya adalah :" + tbg.HitvolumeTabung());
22      }
23  }
```

Hasil implementasi



Aturan dalam konsep Inheritance

2. Dalam inheritance, suatu konstruktor tidak dapat diwariskan.

Konsturktor dari super class tidak dapat diwariskan kepada subclass, namun bisa dipanggil dengan menggunakan keyword **super**().

Super() adalah sebuah kata kunci super mengacu pada objek superclass (induk) yang digunakan untuk memanggil method superclass, dan untuk mengakses konstruktor superclass.

Implementasi :

```
public class Lingkaran {

    int r;
    double phi,luas;

    public Lingkaran(int r){
        this.r = r;
        phi = 3.14;
    }
    void Deskripsi(){
        System.out.println( x:"Ini adalah hasil menghitung");
    }
    double HitluasLingkaran(){
        luas=(phi*r*r);
        return luas;
    }
}
```

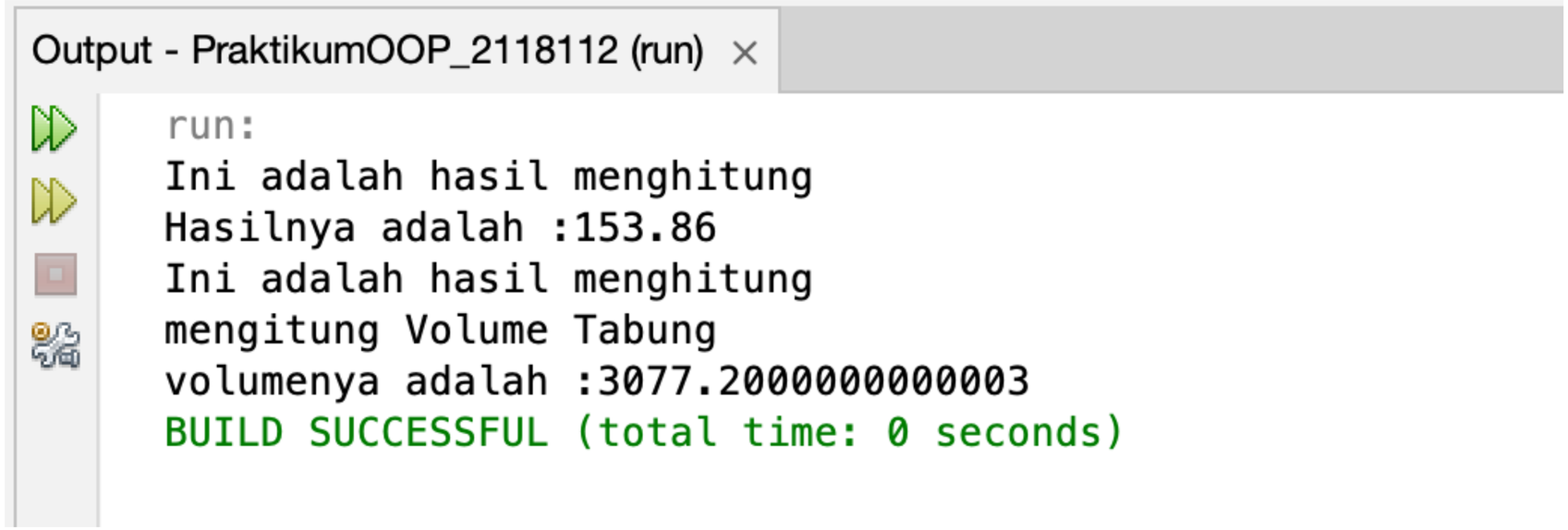
Implementasi :

```
11  public class Tabung extends Lingkaran {
12
13      int t;
        double volume, luasPermukaan;
15
        public Tabung(int r , int t) {
17          super(r);
18          this.t = t;
19      }
20
21      void Keterangan() {
22          Deskripsi();
23          System.out.println( x:"mengitung Volume Tabung");
24      }
25
26      double HitvolumeTabung() {
27          volume = (HitluasLingkaran() * t);
28          return volume;
29      }
30
31  }
```

Implementasi :

```
11  public class main {
12      public static void main(String[] args) {
13          Lingkaran lkr = new Lingkaran( r:7); // memanggil constructor
14          lkr.Deskripsi();
15          System.out.println("Hasilnya adalah :" + lkr.HitluasLingkaran());
16          Tabung tbg = new Tabung( r:7, t:20);
17          tbg.Keterangan();
18          System.out.println("volumenya adalah :" + tbg.HitvolumeTabung());
19      }
20  }
```

Hasil :



Analisa :

Pada source code diatas class tabung mewarisi attribut dan method dari class lingkaran ,namun pada **constroctor berparameter** tidak bisa **menginisialisasi** nilai awal pada jari-jari/r . Keyword super() pada program diatas digunakan untuk menginisialisasi nilai jari-jari pada class lingkaran **melewati** class tabung. Maka dari itu dibutuhkan keyword super untuk mengakses constructor dari class induk.

Aturan dalam konsep Inheritance

3. Acces modifier pada inheritance

Pada inheritance, member dengan acces modifier private tidak dapat diwariskan. Sedangkan member default dapat diwariskan asalkan berada dalam package yang sama, jadi di luar itu tidak dapat diwariskan. Untuk member protected hanya dapat diakses oleh subclass melalui inheritance baik itu pada package yang sama ataupun berbeda. Terakhir adalah member public yang dapat diwariskan di mana saja.(Akan dibahas pada bab selanjutnya).

"Open NetBeans, and let's write code Java"