

# Class Abstract

# Abstract

- Class abstrak adalah class yang masih dalam bentuk abstrak. Karena bentuknya masih abstrak, dia tidak bisa dibuat langsung menjadi objek.
- Abstract class adalah class yang mempunyai setidaknya satu **abstract method**.
- Abstract method adalah method yang tidak memiliki body (hanya deklarasi method).
- Abstract class tidak bisa dibuat obyeknya.
- Obyek hanya bisa dibuat dari non-abstract class (concrete class).

# Contoh Class Abstract

```
package Teori;  
  
public abstract class Parent {  
    public abstract void mAbstract();  
  
    public static void main(String args[]){  
        Parent p = new Parent();  
    }  
}
```

Tidak bisa membuat  
object dari class abstract

Teori.Parent is abstract; cannot be instantiated

Parent p = new Parent();  
1 error

# Contoh Class Abstract

```
public abstract class Parent {  
    public abstract void mAbstract();  
}
```

```
class Child extends Parent{  
    public void mAbstract(){}  
}
```

```
package Teori;  
  
public abstract class Parent {  
    public abstract void mAbstract();  
}  
  
abstract class Child extends Parent{
```

# Contoh Class Abstract

“Method abstrak itu adalah method yang tidak memiliki isi.”

```
// ini abstrak method
void sayHello();

// ini bukan abstrak method karena
// punya implementasi di body method
void greeting(){
    System.out.println("Hello Java");
}
```

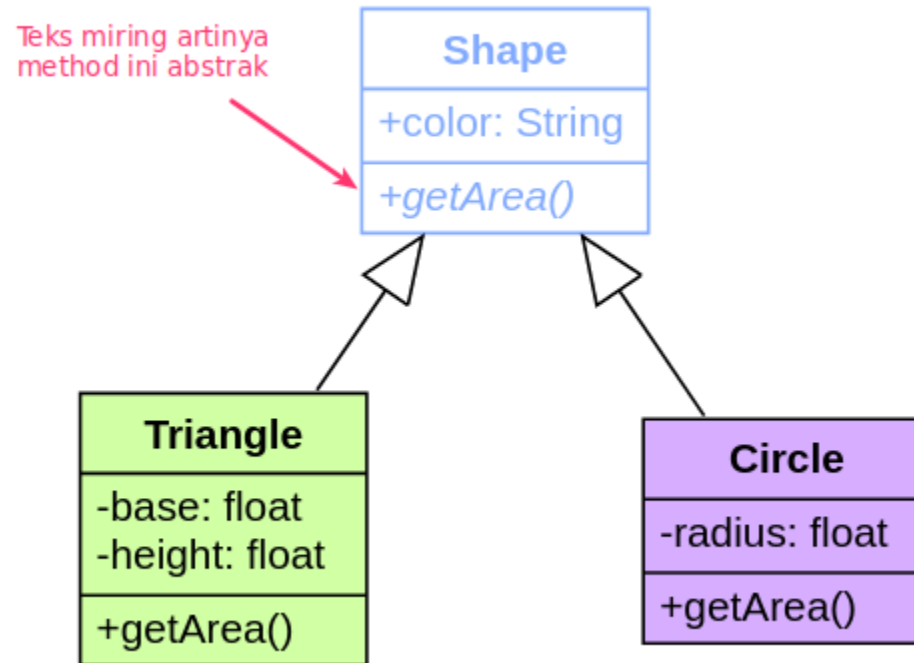
# Kegunaan Class Abstract

- Class Abstract berisi beberapa method dan beberapa method abstract. Class Abstract berisi sebagian implementasi, dan subclass yang melengkapi implementasinya. Dengan kata lain Class Abstract memiliki beberapa kesamaan (Bagian yang diimplementasikan oleh subclass) dan memiliki perbedaan (method yang dimiliki sendiri) abstract)
- Deklarasikan method abstract, jika ada satu atau lebih subclass yang diharapkan mempunyai fungsionalitas yang sama tapi implementasi berbeda.
- Gunakan class abstract untuk mendefinisikan behavior secara umum sebagai superclass, sedangkan subclass menyediakan implementasi detail.
- Jika class abstract semua method merupakan method abstract, sebaiknya class abstract tersebut diganti menjadi Interface.

# Contoh Class Abstract

- Kita dapat menggambar lingkaran, persegi panjang garis, kurva Bezier dan object-object graphic lainnya.
- Object tersebut mempunyai state tertentu (seperti: position, orientation, line color, fill color) dan behaviour secara umum(contoh: moveTo, rotate, resize, draw).
- Beberapa state dan behavior ini sama untuk semua object graphic contoh: position, fill color, and moveTo.
- Implementasi yang berbeda sebagai contoh resize or draw. Semua Object Graphic harus mengetahui bagaimana cara draw dan resize

# Contoh Program Class Abstrak





- Class Shape adalah class abstrak, karena dia punya method abstrak `getArea()` (hitung luas).
- Jika kita membuat objek dengan class Shape dan memanggil method `getArea()`, maka si class Shape akan bingung..
- Shape atau bentuk bangun datar yang mau dihitung luasnya seperti apa?
- Rumusnya gimana?
- Karena itu, class ini kita jadikan abstrak. Soalnya belum jelas cara ngitung luasnya.
- Dan kita juga semua tahu, kalau Shape atau bangun datar pasti memiliki luas. Tapi cara ngitungnya berbeda-beda.

<default package>  
buat class baru bernama Shape

```
public abstract class Shape {  
  
    String color;  
  
    void setColor(String color){  
        this.color = color;  
    }  
  
    String getColor(){  
        return this.color;  
    }  
  
    abstract float getArea();  
}
```

Berikutnya, buat class Triangle. Class ini merupakan turunan dari class Shape.

```
public class Triangle extends Shape {  
  
    private float base;  
    private float height;  
  
    public Triangle(int base, int height) {  
        this.base = base;  
        this.height = height;  
    }  
  
    @Override  
    float getArea() {  
        return 0.5f * base * height;  
    }  
}
```

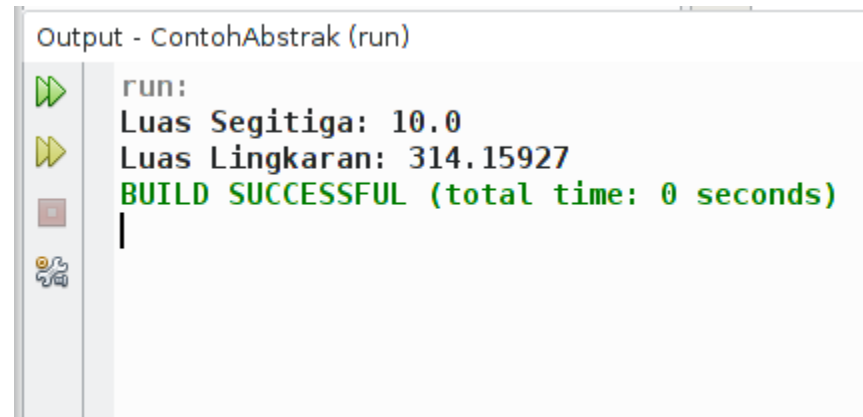
Setelah itu, buat satu lagi turunan dari class Shape, yakni Circle.

```
public class Circle extends Shape {  
  
    private float radius;  
  
    public Circle(float radius) {  
        this.radius = radius;  
    }  
  
    @Override  
    float getArea() {  
        return (float) (Math.PI * radius * radius);  
    }  
}
```

# Terakhir, buat class Main untuk mencoba membuat objek dengan class yang sudah dibuat.

```
public class Main {  
    public static void main(String[] args) {  
        // membuat objek dari class Triangle  
        Shape segitiga = new Triangle(4, 5);  
  
        // membuat objek dari class Circle  
        Shape lingkaran = new Circle(10);  
  
        System.out.println("Luas Segitiga: " + segitiga.getArea());  
        System.out.println("Luas Lingkaran: " + lingkaran.getArea());  
    }  
}
```

- Coba perhatikan objek segitiga dan lingkaran! Kedua objek ini memiliki tipe yang sama yakni Shape, tapi mereka dibuat dari class yang berbeda.



The screenshot shows an IDE output window titled "Output - ContohAbstrak (run)". On the left side of the window, there is a vertical toolbar with icons for running (a green play button), stepping through code (a yellow play button), stopping (a red square), and debugging (a magnifying glass over a bug). The main area of the window displays the following text:

```
run:  
Luas Segitiga: 10.0  
Luas Lingkaran: 314.15927  
BUILD SUCCESSFUL (total time: 0 seconds)  
|
```

# Apa Bedanya Class Abstrak dengan Interface?

- Class abstrak dan interface sama-sama digunakan untuk membuat abstraksi.
- Keduanya hampir memiliki sifat yang sama.
- Tapi ada bedanya.
  1. di class abstrak kita bisa buat properti atau variabel sedangkan di interface kita cuma bisa buat konstanta saja
  2. di class abstrak kita bisa implementasikan kode method seperti class biasa, sedangkan di interface harus menggunakan default
  3. di class abstrak dapat memiliki member private dan protected sedangkan interface harus publik semua
  4. Class abstrak diimplementasikan dengan pewarisan (extends) sedangkan interface menggunakan implements

# Sumber

- <https://www.petanikode.com/java-oop-abstract/>
- Materi Kuliah PBO ITS Surabaya