

Abstrak

Pokok Pembahasan :

- Abstract Class

Tujuan Pembahasan :

- Agar dapat mengetahui dan mengimplementasi kan Abstrack Class

Class abstrak adalah class yang masih dalam bentuk abstrak. Karena bentuknya masih abstrak, dia tidak bisa dibuat langsung menjadi objek. Sebuah class agar dapat disebut class abstrak setidaknya memiliki satu atau lebih method abstrak. Method abstrak adalah method yang tidak memiliki implementasi atau tidak ada bentuk konkritnya.

Cara membuat class abstrak menjadi konkrit adalah dengan membuat implementasi dari method-method yang masih abstrak.

Ini bisa kita lakukan dengan pewarisan (inheritance).

Class abstrak biasanya digunakan sebagai class induk dari class-class yang lain. Class anak akan membuat versi konkrit dari class abstrak.

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.

Nah sekarang mari kita coba dalam kode program.

Buat Class Abstrak

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

Berikut ini kode class Triangle:

```
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 0.5 f * base * height;  
    }  
}
```

Berikut ini kode untuk class 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());
    }
}

```

Output :

```
/home/code-maniac/.sdkman/candidates/java/8.0.292.j9-adpt/bin/java ...
```

```
Luas Segitiga: 10.0
```

```
Luas Lingkaran: 314.15927
```

```
Process finished with exit code 0
```

```
|
```