

**TUGAS PEMROGRAMAN BERORIENTASI OBJEK
ENCAPSULATION dan
ABSTRACTION**



Dosen Pengampu :

Lukie Perdanasari S.Kom., M.T.

Disusun oleh :

Olla Nura Aura Fadillah

E31242387

Golongan A Internasional

**PROGRAM STUDI MANAJEMEN INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
2024**

1. Pengertian Encapsulation dan Abstraction

Encapsulation adalah konsep dalam pemrograman berorientasi objek yang menyembunyikan detail implementasi suatu objek dan hanya memperlihatkan bagian yang diperlukan melalui antarmuka publik. Hal ini membantu dalam menjaga keamanan data dan mencegah manipulasi langsung dari luar kelas. Sedangkan, **Abstraction** adalah konsep yang menyederhanakan kompleksitas dengan hanya menampilkan fitur esensial suatu objek dan menyembunyikan detail yang tidak relevan. Dengan abstraksi, pengguna hanya perlu memahami fungsi tanpa harus tahu bagaimana cara kerjanya secara internal.

2. Membuat Program Java Implementasi Encapsulation dan Abstraction

Program Java : Shape – Circle – Triangle

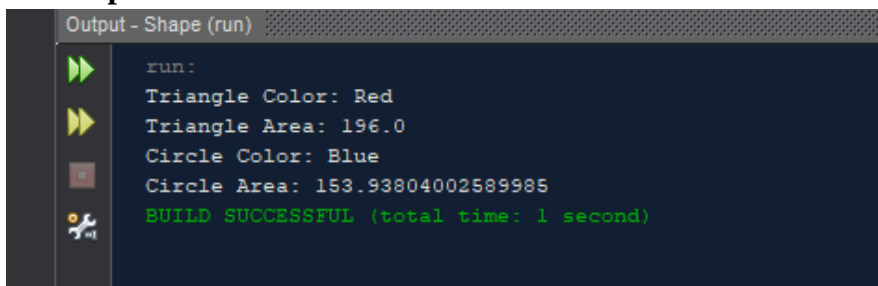
```
1. // Kelas abstrak Shape yang menjadi superclass untuk bentuk-bentuk geometri lainnya
2. abstract class Shape {
3.     protected String color; // Atribut untuk menyimpan warna bentuk
4.
5.     // Konstruktor untuk menginisialisasi warna bentuk
6.     public Shape(String color) {
7.         this.color = color;
8.     }
9.
10.    // Metode abstrak untuk menghitung luas, harus diimplementasikan di subclass
11.    public abstract double getArea();
12. }
13.
14. // Kelas Triangle sebagai subclass dari Shape, mewakili segitiga
15. class Triangle extends Shape {
16.     private final float base; // Alas segitiga
17.     private final float height; // Tinggi segitiga
18.
19.     // Konstruktor untuk menginisialisasi warna, alas, dan tinggi segitiga
20.     public Triangle(String color, float base, float height) {
21.         super(color); // Memanggil konstruktor dari superclass (Shape)
22.         this.base = base;
23.         this.height = height;
24.     }
25.
26.     // Implementasi metode getArea() untuk menghitung luas segitiga
27.     @Override
28.     public double getArea() {
29.         return 0.5 * base * height; // Rumus luas segitiga = 1/2 × alas × tinggi
30.     }
31. }
32.
33. // Kelas Circle sebagai subclass dari Shape, mewakili lingkaran
34. class Circle extends Shape {
35.     private final float radius; // Jari-jari lingkaran
```

```

36.
37. // Konstruktor untuk menginisialisasi warna dan jari-jari lingkaran
38. public Circle(String color, float radius) {
39.     super(color); // Memanggil konstruktor dari superclass (Shape)
40.     this.radius = radius;
41. }
42.
43. // Implementasi metode getArea() untuk menghitung luas lingkaran
44. @Override
45. public double getArea() {
46.     return Math.PI * radius * radius; // Rumus luas lingkaran =  $\pi \times r^2$ 
47. }
48. }
49.
50. // Kelas utama untuk menjalankan program
51. public class Main {
52.     public static void main(String[] args) {
53.         // Membuat objek Triangle dengan warna "Red", alas 14, dan tinggi 28
54.         Triangle triangle = new Triangle("Red", 14, 28);
55.
56.         // Membuat objek Circle dengan warna "Blue" dan jari-jari 7
57.         Circle circle = new Circle("Blue", 7);
58.
59.         // Menampilkan informasi segitiga
60.         System.out.println("Triangle Color: " + triangle.color); // Menampilkan warna segitiga
61.         System.out.println("Triangle Area: " + triangle.getArea()); // Menampilkan luas segitiga
62.
63.         // Menampilkan informasi lingkaran
64.         System.out.println("Circle Color: " + circle.color); // Menampilkan warna lingkaran
65.         System.out.println("Circle Area: " + circle.getArea()); // Menampilkan luas lingkaran
66.     }
67. }

```

2.1 Output



```

Output - Shape (run)
run:
Triangle Color: Red
Triangle Area: 196.0
Circle Color: Blue
Circle Area: 153.93804002589985
BUILD SUCCESSFUL (total time: 1 second)

```

2.2 Penjelasan Program

1. Abstract Class Shape

- Pada baris ke – 3 adalah Protected String Color Variabel warna (color) bisa diakses oleh subclass.
- Pada baris ke – 6 digunakan untuk memberi warna saat objek dibuat.

- Pada baris ke-7 untuk menyimpan warna yang diterima sebagai parameter ke dalam atribut color.
- Pada baris ke-11 adalah metode abstrak yang **harus diimplementasikan** oleh subclass.

Class Shape adalah kelas abstrak yang menjadi dasar bagi bentuk lainnya. Kelas ini menyimpan atribut color untuk merepresentasikan warna suatu bentuk. Konstruktor dalam kelas ini digunakan untuk menginisialisasi warna saat objek dibuat. Selain itu, terdapat metode abstrak `getArea()` yang harus diimplementasikan oleh setiap subclass yang mewarisi kelas ini.

2. Class Triagle

- Pada baris ke-15 adalah Triangle adalah subclass dari Shape
- Pada baris ke-16 adalah Variabel untuk menyimpan **alas segitiga**.
- Pada baris ke-17 adalah Variabel untuk menyimpan **tinggi segitiga**.
- Pada baris ke-20 adalah menginisialisasi objek segitiga.
- Pada baris ke-21 adalah Memanggil konstruktor Shape untuk mengatur warna.
- Pada baris ke-22 adalah Menyimpan nilai alas segitiga.
- Pada baris ke-23 adalah Menyimpan nilai tinggi segitiga.
- Pada baris ke-27 adalah Menandakan bahwa metode `getArea()` adalah implementasi dari metode abstrak Shape.
- Pada baris ke-29 adalah rumus luas segitiga: $\frac{1}{2} \times \text{alas} \times \text{tinggi}$.

Class Triangle adalah subclass dari Shape yang merepresentasikan bentuk segitiga. Kelas ini memiliki atribut tambahan berupa base untuk menyimpan panjang alas dan height untuk menyimpan tinggi segitiga. Konstruktor dalam kelas ini digunakan untuk menginisialisasi warna, alas, dan tinggi dari segitiga yang dibuat. Metode `getArea()` diimplementasikan dengan rumus luas segitiga, yaitu $\frac{1}{2} \times \text{alas} \times \text{tinggi}$.

3. Class Circle

- Pada baris ke- 34 adalah Circle adalah subclass dari Shape.
- Pada baris ke- 35 adalah Variabel untuk menyimpan **jari-jari lingkaran**.
- Pada baris ke- 38 adalah menginisialisasi objek lingkaran.
- Pada baris ke- 39 adalah Memanggil konstruktor Shape untuk mengatur warna.
- Pada baris ke- 40 adalah Menyimpan nilai jari-jari lingkaran.
- Pada baris ke- 44 adalah Menandakan bahwa metode `getArea()` adalah implementasi dari metode abstrak Shape.
- Pada baris ke- 45 adalah Menggunakan rumus luas lingkaran: $\pi \times r^2$.

Class Circle adalah subclass dari Shape yang merepresentasikan bentuk lingkaran. Kelas ini memiliki atribut radius yang menyimpan ukuran jari-jari lingkaran. Konstruktor digunakan untuk menginisialisasi warna dan jari-jari saat objek dibuat. Metode `getArea()` diimplementasikan dengan rumus luas lingkaran, yaitu $\pi \times r^2$.

4. Class Main

- Pada baris ke- 51 adalah Kelas utama untuk menjalankan program.

- Pada baris ke- 52 adalah Metode main tempat eksekusi program dimulai, sebagai pondasi awal.
- Pada baris ke- 54 adalah Membuat objek segitiga dengan warna "**Red**", alas **14**, dan tinggi **28**.
- Pada baris ke- 57 adalah Membuat objek lingkaran dengan warna "**Blue**" dan jari-jari **7**.
- Pada baris ke- 60 adalah Cetak warna segitiga.
- Pada baris ke- 61 adalah Cetak luas segitiga.
- Pada baris ke- 64 adalah Cetak warna lingkaran.
- Pada baris ke- 65 adalah Cetak luas lingkaran.

Class Main adalah kelas utama yang menjalankan program. Di dalamnya terdapat metode main() yang digunakan sebagai titik awal eksekusi program. Kelas ini membuat objek Triangle dan Circle dengan warna serta ukuran tertentu. Setelah itu, program akan mencetak warna dan luas dari masing-masing bentuk sebagai output.

3. LINK GITHUB

<https://github.com/ollafadillah/PBO-Tugas-Encapsulation-and-Abstraction.git>