

LAPORAN
WORKSHOP SISTEM INFORMASI BERBASIS DESKTOP
(TIF120707)
SEMESTER 2



Inheritance, polymorphism, dan abstract class

ANGGA PRASETIO
E41231567
Teknik Informatika (B)

PROGRAM STUDI TEKNIK INFORMATIKA KAMPUS 3 NGANJUK
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
TAHUN 2024

a. Tugas Acara 9(a)

➤ Class DrawShapes

```
1 package acara9dan10;
2
3 import java.awt.*;
4 import java.applet.*;
5
6
7 // Deklarasi kelas DrawShapes yang merupakan turunan dari Applet
8 public class DrawShapes extends Applet {
9
10     // Deklarasi variabel font, redColor, blueColor, dan backgroundColor
11     Font font;
12     Color redColor;
13     Color blueColor;
14     Color backgroundColor;
15
16     // Metode init untuk inisialisasi
17     public void init() {
18
19         // Menginisialisasi variabel font dengan font Arial, style ITALIC, dan ukuran 18
20         font = new Font(name: "Arial", style: Font.ITALIC, size: 18);
21
22         // Menginisialisasi variabel redColor dengan objek Color yang merepresentasikan warna merah
23         redColor = Color.red;
24
25         // Menginisialisasi variabel backgroundColor dengan objek Color yang merepresentasikan warna orange
26         backgroundColor = Color.orange;
27
28         // Menginisialisasi variabel blueColor dengan objek Color yang merepresentasikan warna biru dengan komponen RGB (0, 0, 122)
29         blueColor = new Color(r: 0, g: 0, b: 122);
30     }
31 }
```

Activate Wir

```
31 // Metode stop
32 public void stop() {
33 }
34 // Metode paint() untuk menggambar
35 public void paint(Graphics graph) {
36
37     // Mengatur font yang akan digunakan
38     graph.setFont(font);
39
40     // Menggambar teks "Draw Shapes" pada posisi (90, 20)
41     graph.drawString(str: "Draw Shapes", x: 90, y: 20);
42
43     // Mengatur warna grafik menjadi blueColor
44     graph.setColor(c: blueColor);
45
46     // Menggambar persegi dengan batas (120, 120, 120, 120)
47     graph.drawRect(x: 120, y: 120, width: 120, height: 120);
48
49     // Menggambar persegi panjang dengan batas (115, 115, 90, 90)
50     graph.fillRect(x: 115, y: 115, width: 90, height: 90);
51
52     // Mengatur warna grafik menjadi redColor
53     graph.setColor(c: redColor);
54
55     // Menggambar lingkaran yang berada di dalam persegi dengan batas (110, 110, 50, 50)
56     graph.fillArc(x: 110, y: 110, width: 50, height: 50, startAngle: 0, arcAngle: 360);
57
58     // Mengatur warna grafik menjadi Color.CYAN
59     graph.setColor(c: Color.CYAN);
60 }
```

```

60
61 // Menggambar persegi dengan batas (50, 50, 50, 50)
62 graph.drawRect(x: 50, y: 50, width: 50, height: 50);
63
64 // Menggambar persegi panjang dengan batas (50, 50, 60, 60)
65 graph.fillRect(x: 50, y: 50, width: 60, height: 60);
66
67 }

```

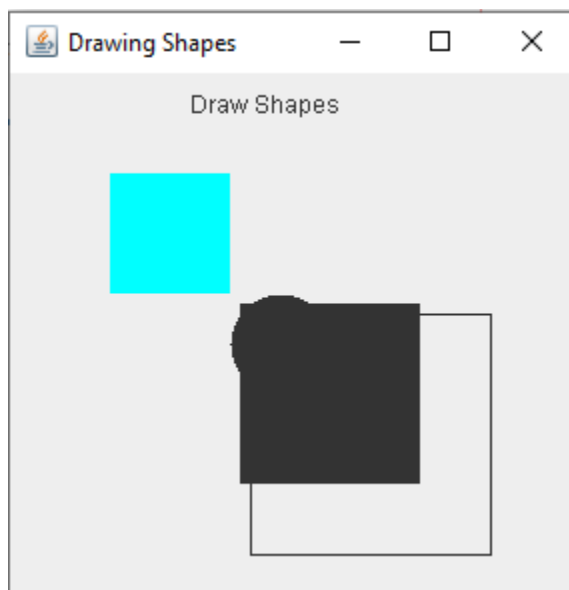
- Class NewMain untuk menjalankan class program DrawShapes

```

1 package acara9dan10;
2 // Mengimport kelas DrawShapes dari package acara9dan10
3 import acara9dan10.DrawShapes;
4 import javax.swing.*; // Mengimport kelas JFrame dari paket javax.swing
5
6 public class NewMain extends JFrame { // Deklarasi kelas NewMain yang merupakan turunan dari JFrame
7
8     public NewMain() {
9         // Mengatur judul frame menjadi "Drawing Shapes"
10        setTitle(title: "Drawing Shapes");
11        // Membuat objek DrawShapes
12        DrawShapes drawshapes = new DrawShapes();
13        // Menambahkan objek DrawShapes ke dalam frame
14        add(comp: drawshapes);
15        // Mengatur ukuran frame menjadi 300 x 300 piksel
16        setSize(width: 300, height: 300);
17        // Mengatur operasi default ketika frame ditutup
18        setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
19        // Menjadikan frame terlihat
20        setVisible(b: true);
21    }
22
23    public static void main(String[] args) {
24        // Membuat objek NewMain
25        NewMain main = new NewMain();
26    }
27
28 }

```

- Hasil dari kode program diatas



b. Tugas Acara 9(b)

```
1 package Acara9;
2
3
4 class A {
5     void callthis() {
6         System.out.println(x: "Inside Class A's Method!");
7     }
8 }
9 class B extends A {
10
11     void callthis() {
12         System.out.println(x: "Inside Class B's Method!");
13     }
14 }
15
16 class C extends A {
17
18     void callthis() {
19         System.out.println(x: "Inside Class C's Method!");
20     }
21 }
22
23 class DynamicDispatch {
24
25     public static void main(String args[]) {
26         A a = new A();
27         B b = new B();
28         C c = new C();
29         A ref;
30         // Dynamic Dispatching: ref diatur ke objek dari kelas B
31         ref = b;
32         ref.callthis();// Memanggil metode callthis() dari kelas B
33
34         // Dynamic Dispatching: ref diatur ke objek dari kelas C
35         ref = c;
36         ref.callthis();// Memanggil metode callthis() dari kelas C
37
38         // Dynamic Dispatching: ref diatur ke objek dari kelas A
39         ref = a;
40         ref.callthis();// Memanggil metode callthis() dari kelas A
41     }
42 }
```

Output kode program diatas

```
Output - Acara9Dan10 (run) #2
run:
Inside Class B's Method!
Inside Class C's Method!
Inside Class A's Method!
BUILD SUCCESSFUL (total time: 1 second)
```

c. Tugas Acara 9(c)

➤ Class Animal

```
1 package Acara9;
2
3
4 public void makeNoise() {
5
6     System.out.println(x: "talk");
7
8 }
9
10 }
```

➤ Class Dog

```
1 package Acara9;
2
3 public class Dog extends Animal {
4
5     // Override metode makeNoise dari kelas Animal
6     public void makeNoise() {
7         System.out.println(x: "Bark");
8     }
9
10     public static void main(String[] args) {
11         Animal animal = new Animal(); // Membuat objek Animal
12         animal.makeNoise(); // Memanggil metode makeNoise() dari kelas Animal
13         Dog dog = new Dog(); // Membuat objek dog
14         dog.makeNoise(); // Memanggil metode makeNoise() dari kelas Dog (override)
15         Animal animaldog = new Dog(); // Membuat objek Animal yang merujuk ke objek Dog
16         animaldog.makeNoise(); // Dynamic Dispatching: Memanggil metode makeNoise() dari kelas Dog (override)
17
18         // Pengujian instanceof untuk mengecek hubungan antara objek dan kelas
19         if (animal instanceof Animal) {
20             System.out.println(x: "animal is Animal");
21         }
22         if (dog instanceof Animal) {
23             System.out.println(x: "dog is Animal");
24         }
25         if (animaldog instanceof Animal) {
26             System.out.println(x: "animal dog is Animal");
27         }
28         if (animal instanceof Dog) {
29             System.out.println(x: "animal is Dog"); // Output tidak akan tereksekusi
30         }
31     }
32 }
```

➤ Hasil output

```
Output - Acara9Dan10 (run)
run:
talk
Bark
Bark
animal is Animal
dog is Animal
animal dog is Animal
BUILD SUCCESSFUL (total time: 1 second)
```

Tugas Acara 10

1. Class BangunDatar

```
1 package Acara10;
2
3 public class BangunDatar {
4
5     float luas() {
6         System.out.println(x: "Menghitung luas bangun datar");
7         return 0;
8     }
9
10    float keliling() {
11        System.out.println(x: "Menghitung keliing bangun datar");
12        System.out.println(x: "=====");
13        return 0;
14    }
15 }
16
```

2. Class Persegi

```
1 package Acara10;
2
3 public class Persegi extends BangunDatar {
4
5     int sisi;
6
7     public Persegi(int sisi) {
8         this.sisi = sisi;
9     }
10
11    public float luas() {
12        return this.sisi * this.sisi;
13    }
14
15    public float keliling() {
16        return this.sisi * 4;
17    }
18 }
19
```

3. Class Segitiga

```
1 package Acara10;
2
3
4
5 public class Segitiga extends BangunDatar{
6     int alas;
7     int tinggi;
8
9
10    public Segitiga(int alas, int tinggi){
11        this.alas=alas;
12        this.tinggi=tinggi;
13    }
14    public float luas(){
15        return this.alas * this.tinggi;
16    }
17 }
```

4. Class Lingkaran

```
1 package Acara10;
2
3 public class Lingkaran extends BangunDatar {
4
5     int r;
6
7     public Lingkaran(int r) {
8         this.r = r;
9     }
10
11    public float luas() {
12        return (float) (Math.PI * r * r);
13    }
14
15    public float keliling() {
16        return (float) (2 * Math.PI * r * r);
17    }
18 }
```

5. Class NewMain

```
1 package Acara10;
2
3 public class NewMain {
4
5     public static void main(String[] args) {
6         BangunDatar bangunDatar = new BangunDatar();
7         Persegi persegi = new Persegi(sisi: 4);
8         Segitiga segitiga = new Segitiga(alas: 6, tinggi: 3);
9         Lingkaran lingkaran = new Lingkaran(r: 50);
10
11         //memanggil method luas dan keliling
12         bangunDatar.luas();
13         bangunDatar.keliling();
14
15         System.out.println("Luas persegi: " + persegi.luas());
16         System.out.println("Keliling persegi: " + persegi.keliling());
17         System.out.println("Luas segitiga: " + segitiga.luas());
18         System.out.println("Luas lingkaran: " + lingkaran.luas());
19         System.out.println("Keliling lingkaran: " + lingkaran.keliling());
20     }
21 }
22 }
```

6. Hasil output

Output - Acara9Dan10 (run)

```
run:
Menghitung luas bangun datar
Menghitung keliing bangun datar
=====
Luas persegi: 16.0
Keliling persegi: 16.0
Luas segitiga: 18.0
Luas lingkaran: 7853.9814
Keliling lingkaran: 7853.9814
BUILD SUCCESSFUL (total time: 0 seconds)
```