

LAPORAN PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK PERTEMUAN KE- 10



Nama : Riyan Risky Widya S
Nim : 185410103
Kelas : TI-2

STMIK AKAKOM YOGYAKARTA

A. TUJUAN

Dapat membuat dan menggunakan interface pada aplikasi.

B. DASAR TEORI

Interface adalah kelas yang benar-benar abstrak, artinya hanya berisi deklarasi method dan (jika ada) konstanta saja. Method-method tersebut nantinya harus diimplementasikan pada real class. Interface dapat dianalogikan seperti menandatangani kontrak kerja. Misalnya seorang dosen wajib mengajar, membuat soal, dsb, akan tetapi cara mengajar dan membuat soalnya diserahkan ke masing-masing dosen (tidak ditentukan dalam kontrak kerja).

Interface mendefinisikan suatu protokol perilaku tanpa dipusingkan dengan implementasinya. Suatu kelas yang mengimplementasikan interface maka ada kelas tersebut akan melekat perilaku yang didefinisikan interface.

Pewarisan merupakan proses penurunan field dan method dari suatu kelas induk kepada satu/lebih subkelas. Seringkali dibutuhkan suatu kelas yang field dan methodnya berasal dari lebih dari satu kelas induk (pewarisan jamak). Pewarisan jamak memang mempercepat dalam pembuatan kelas.

C. PEMBAHASAN

Praktik 1

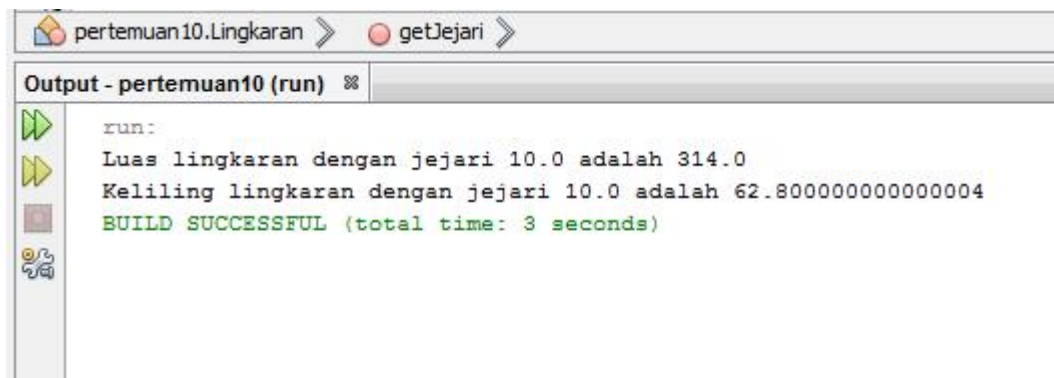
```
Start Page  x  Pertemuan10.java  x  TestLingkaran.java  x  TestInherInterface.java  x  MultiInterface
Source  History  [Icons]

1  package pertemuan10;
2
3  interface Bangun2D {
4
5      public double hitungLuas ();
6      public double hitungKeliling ();
7  }
8
9  class Lingkaran implements Bangun2D {
10
11      private double jejari;
12      public void setJejari(double jejari) {
13          this.jejari = jejari;
14      }
15
16      public double getJejari() {
17          return this.jejari;
18      }
19
20      public double hitungLuas() {
21          return (3.14 * this.jejari * this.jejari);
22      }
23
24      public double hitungKeliling() {
25          return (2 * 3.14 * this.jejari);
26      }
27  }
28
29  class TestLingkaran {
30      public static void main(String[] arg) {
31          Lingkaran bundar = new Lingkaran();
32          bundar.setJejari(10);
33          double luas = bundar.hitungLuas();
34          double keliling = bundar.hitungKeliling();
35          System.out.println("Luas lingkaran dengan jejari " +
36              bundar.getJejari() + " adalah "+luas);
37          System.out.println("Keliling lingkaran dengan jejari " +
38              bundar.getJejari() + " adalah "+keliling);
39      }
40  }
```

Membuat interface bangun2D dengan didalamnya ada 2 method yaitu hitungLuas dan hitungKeliling. Lalu class Lingkaran mengimplementasikan class bangun 2D, terdapat method setJejari(), method getJejari(), method hitungLuas(), dan hitung keliling(). masing masing method memiliki fungsinya sendiri, method setJejari digunakan untuk memasukkan nilai jari jari, method getJejari digunakan untuk mengembalikan nilai jari jari dan

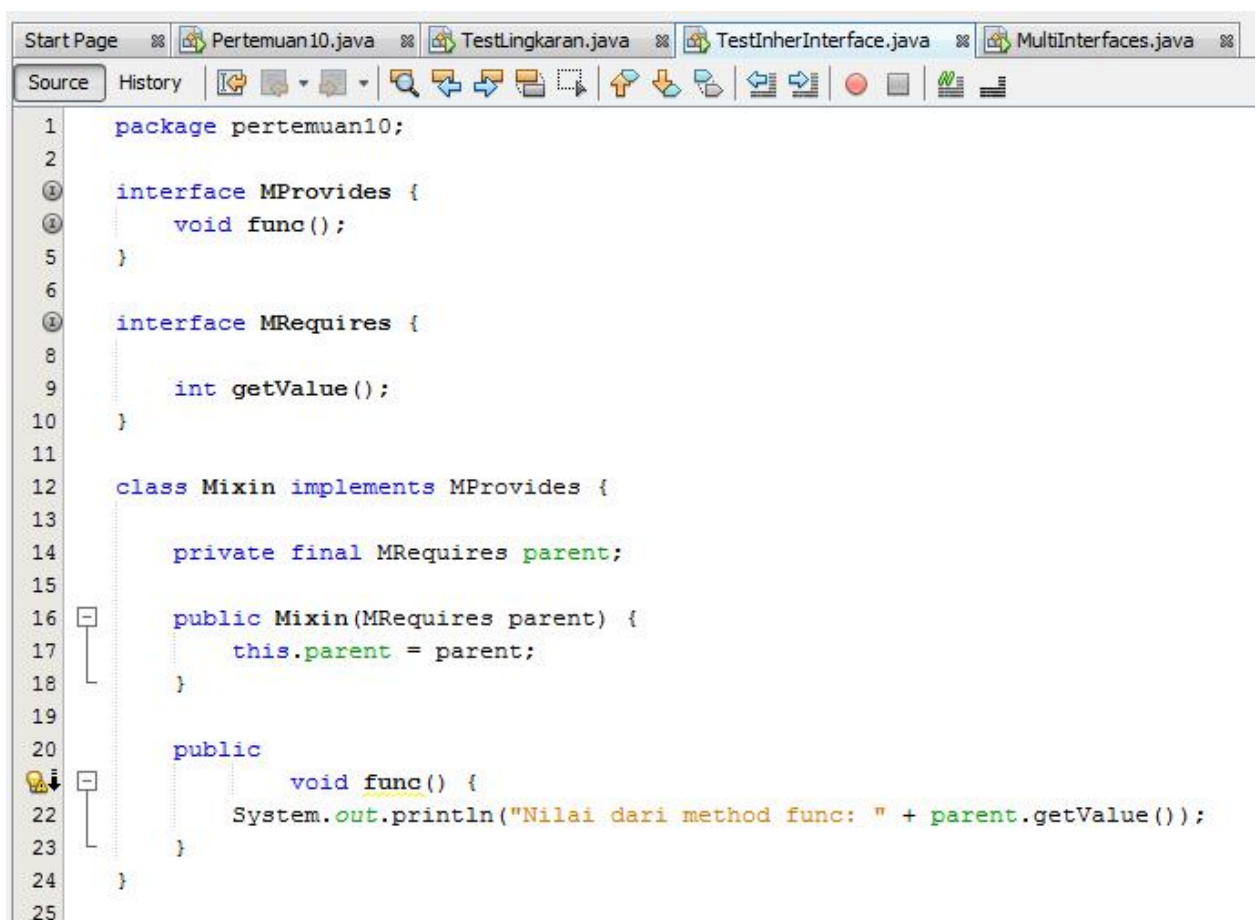
hitung luas digunakan untuk menghitung luas dan hitung lingkaran digunakan untuk menghitung lingkaran.

Class yang kedua adalah testLingkaran terdapat method main. Didalamnya terdapat objek bundar yang merupakan implementasi dari class lingkaran. Lalu memasukkan jari jari 10 dan menghitung luas dan keliling lalu menampilkanya.



```
run:
Luas lingkaran dengan jejari 10.0 adalah 314.0
Keliling lingkaran dengan jejari 10.0 adalah 62.800000000000004
BUILD SUCCESSFUL (total time: 3 seconds)
```

Praktik 2



```
1 package pertemuan10;
2
3 interface MProvides {
4     void func();
5 }
6
7 interface MRequires {
8
9     int getValue();
10 }
11
12 class Mixin implements MProvides {
13
14     private final MRequires parent;
15
16     public Mixin(MRequires parent) {
17         this.parent = parent;
18     }
19
20     public
21     void func() {
22         System.out.println("Nilai dari method func: " + parent.getValue());
23     }
24 }
25
```

```

25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

class Parent {

    private int value;

    public Parent(int value) {
        this.value = value;
    }

    public int getValue() {
        return this.value;
    }
}

class Child extends Parent implements MRequires, MProvides{
    private final MProvides mixin;

    public Child(int value) {
        super(value);
        this.mixin = new Mixin(this);
    }

    public void func() {
        mixin.func();
    }
}

class TestInherInterface {

    public static void main(String[] arg) {
        Child anak = new Child(5);
        anak.func();
        System.out.println("nilai dari method getValue:" + anak.getValue());
    }
}

```

Membuat interface mprovides dan mrequire dengan didalamnya masing masing ada 1 method. Lalu class mixing mengimplementasikan class mprovides, dan class parent. Class child mengextend class parent dan mengimplement class mrequire dan mprovides sehingga terdapat 2 method yaitu child dan func.

Class terhir adalah class tesIntefave yang terdapat method main. Didalamnya ada objek yang terbuat dari class child tadi dan memanggil method func yang ternyata berhasil.

```
18 |         mixin func();
pertemuan10.Child >
Output - pertemuan10 (run) %
run:
Nilai dari method func: 5
nilai dari method getValue:5
BUILD SUCCESSFUL (total time: 0 seconds)
```

Praktik 3

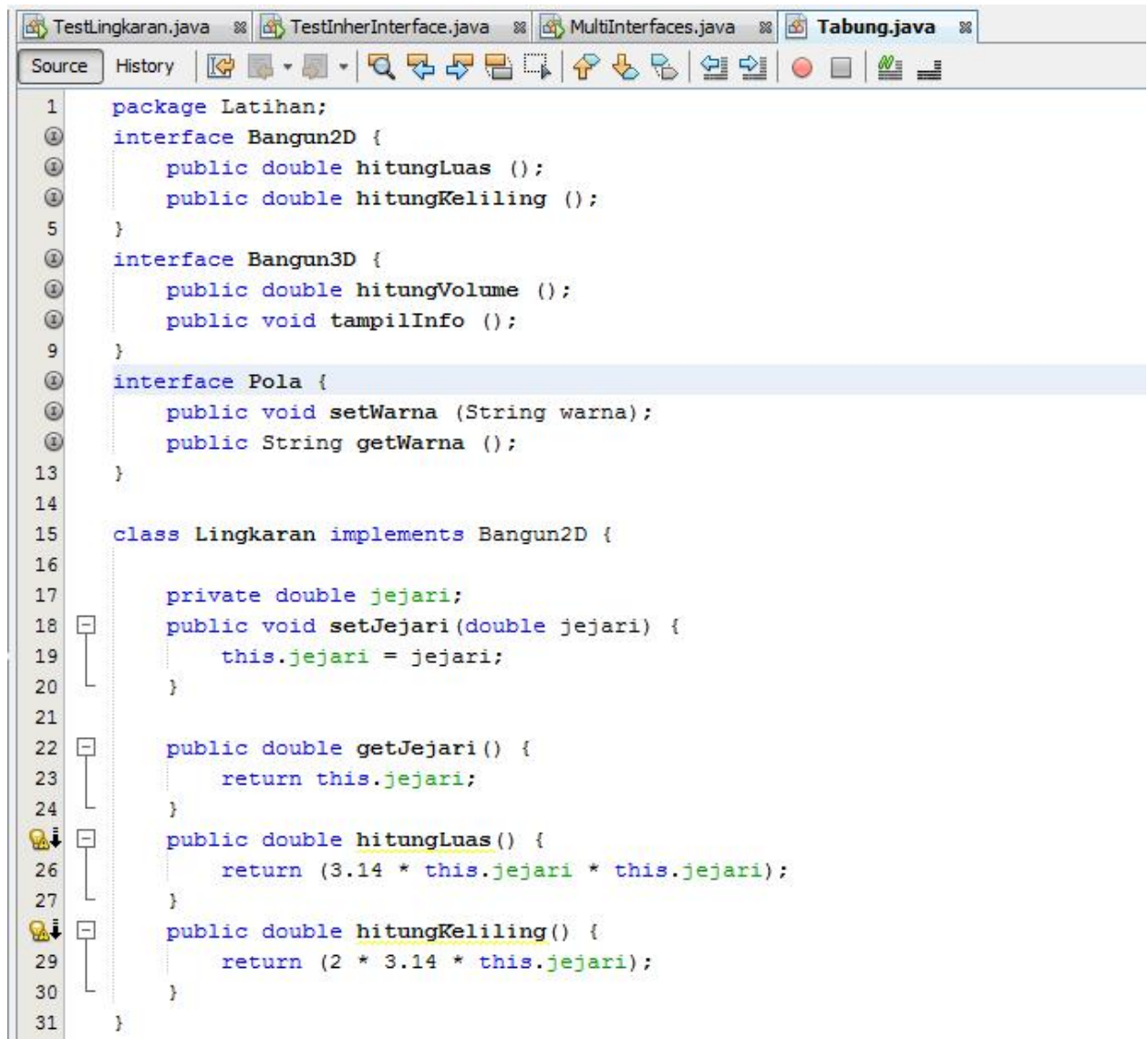
```
1  package pertemuan10;
2
3  interface Interface1 {
4      public void tampilInfo();
5      public void setInfo(String info);
6  }
7
8  interface Interface2 {
9      public void cetakInfo();
10 }
11
12 public class MultiInterfaces implements Interface1, Interface2 {
13
14     private String info;
15
16     public void setInfo(String info) {
17         this.info = info;
18     }
19
20     public void tampilInfo() {
21         System.out.println(this.info + ": ini info dari method tampilInfo");
22     }
23     public void cetakInfo() {
24         System.out.println(this.info + ": ini info dari method cetakInfo");
25     }
26     public static void main(String[] a) {
27         MultiInterfaces t = new MultiInterfaces();
28         t.setInfo("Hai");
29         t.tampilInfo();
30     }
31 }
```

pertemuan10.MultiInterfaces > setInfo >

```
Output - pertemuan10 (run) %
run:
Hai: ini info dari method tampilInfo
BUILD SUCCESSFUL (total time: 0 seconds)
```

Membuat interface interface1 didalamnya terdapat 2 method yaitu tampilInfo dan setInfo. Interface kedua adalah interface2 terdapat method cetakInfo. Pada class multiInterface mengimplement kan 2 class interface tadi. Sehingga ketika ditampilkan akan terlihat seperti dibawah.

Latihan



The screenshot shows an IDE with four tabs: TestLingkaran.java, TestInherInterface.java, MultiInterfaces.java, and Tabung.java. The 'Source' tab is active, displaying the code for TestLingkaran.java. The code defines two interfaces, Bangun2D and Bangun3D, and a class Lingkaran that implements Bangun2D. The Lingkaran class has a private attribute jejari and implements methods setJejari, getJejari, hitungLuas, and hitungKeliling. The hitungLuas method calculates the area of a circle, and hitungKeliling calculates the circumference. The code is as follows:

```
1 package Latihan;
2 interface Bangun2D {
3     public double hitungLuas ();
4     public double hitungKeliling ();
5 }
6
7 interface Bangun3D {
8     public double hitungVolume ();
9     public void tampilInfo ();
10 }
11
12 interface Pola {
13     public void setWarna (String warna);
14     public String getWarna ();
15 }
16
17 class Lingkaran implements Bangun2D {
18     private double jejari;
19     public void setJejari(double jejari) {
20         this.jejari = jejari;
21     }
22     public double getJejari() {
23         return this.jejari;
24     }
25     public double hitungLuas() {
26         return (3.14 * this.jejari * this.jejari);
27     }
28     public double hitungKeliling() {
29         return (2 * 3.14 * this.jejari);
30     }
31 }
```

```

33 public class Tabung implements Bangun3D, Pola {
34     private double tinggi;
35     private Lingkaran alas;
36     private String warna;
37
38     public void setTinggi(double tg){
39         this.tinggi = tg;
40     }
41
42     public double hitungVolume (){
43         alas = new Lingkaran();
44         alas.setJari(10);
45         alas.hitungLuas();
46         double volum = alas.hitungLuas()*tinggi;
47         return volum;
48     }
49
50     public void tampilInfo (){
51         System.out.println("Volume tabung adalah "+this.hitungVolume());
52     }
53
54     public void setWarna(String warna){
55         this.warna = warna;
56     }
57
58     public String getWarna (){
59         return warna;
60     }
61 }

```

D. TUGAS

-

E. KESIMPULAN

Interface hanya digunakan untuk membuat abstraksi tentang objek yang akan dibuat, class yang interface hanya sebagai class induk tidak dapat dijadikan objek secara langsung dan pewarisan jamak tidak bisa dilakukan didalam java.