19

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN KE-10

****

**Disusun oleh :**

**NAMA : FERDI DIRGANTARA**

**NIM : 175410039**

**JURUSAN : TEKNIK INFORMATIKA**

**JENJANG : S1**

LABORATORIUM TERPADU

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTERAKAKOMYOGYAKARTA

2019

1. **TEORI SINGKAT**

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 pada kelas tersebut akan melekat perilaku yang didefinisikan interface.

Declarasi Interface :

[modifier] interface NamaInterface {

deklarasi konstanta;

deklarasi method-method;

}

Implementasi kemudian dilakukan pada class lainnya:

[modifier] class NamaKelas implements NamaInterface{

definisi method-method;

…

}

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. Tetapi mempunyai beberapa kelemahan diantaranya adalah terjadi ambiguitas karena adanya method yang sama yang diturunkan dari beberapa kelas induk.

1. **PEMBAHASAN LISTING**

Praktik 1



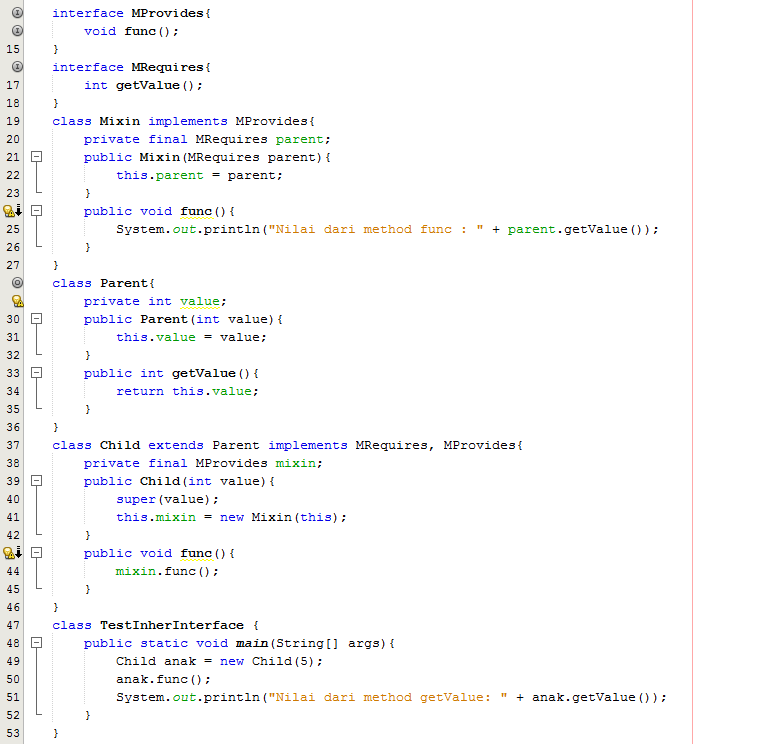


Kode program diatas merupakan contoh penggunaan interface yaitu interface Bangun2D yang memiliki 2 method abstract yaitu hitungLuas() dan hitungKeliling(). Dalam interface semua merupakan bentuk abstract tidak seperti kelas abstract yang mana kita harus mendefinisikan suatu kelas atau method abstract sesuai keinginan kita.

Pada kelas Lingkaran dilakukan override pada kedua method yang terdapat pada interface Bangun2D untuk melakukan perhitungan luas dan keliling dari suatu lingkaran. Selain itu, terdapat method setJejari() yang digunakan untuk menentukan nilai jari-jari dari suatu lingkaran dan method getJejari() yang digunakan untuk mengambil nilai jari-jari yang telah ditentukan tersebut. Nilai jari-jari tersebut digunakan untuk menghitung luas dan keliling lingkaran.

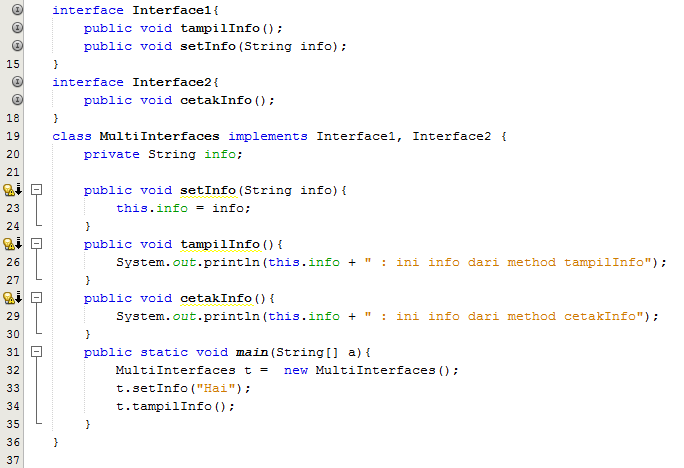
Pada kelas TestLingkaran kemudian dibuat objek bundar untuk mengakses kelas Lingkaran dan kemudian menggunakan method setJejari() dengan nilai argument 10 (nilai jejari) kemudian nilai tersebut dihitung dengan menggunakan method hitungLuas() dan hitungKeliling() baru hasilnya dapat ditampilkan ke layar dengan bantuan System.out.println().

Praktik 2



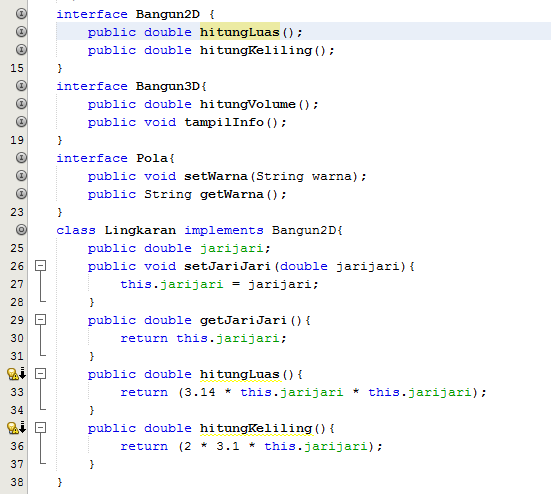
Kombinasi penggunaan turunan suatu kelas (inheritance) dengan interface dapat dilakukan dalam pemrograman berorientasi objek yang mana untuk mengakses suatu interface dapat menggunakan klausa implements seperti yang dicontohkan pada kelas Child diatas yaitu merupakan kelas turunan dari kelas Parent dan mengimplementasikan interface MRequires dan interface MProvides. Selain kelas Child terdapat juga kelas Mixin yang mengimplemen interface MProvides.

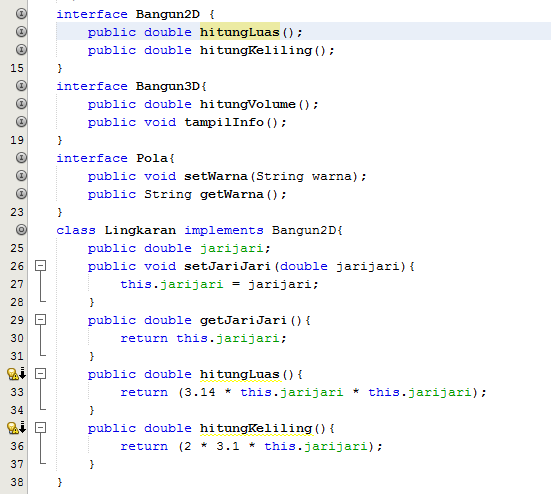
Praktik 3

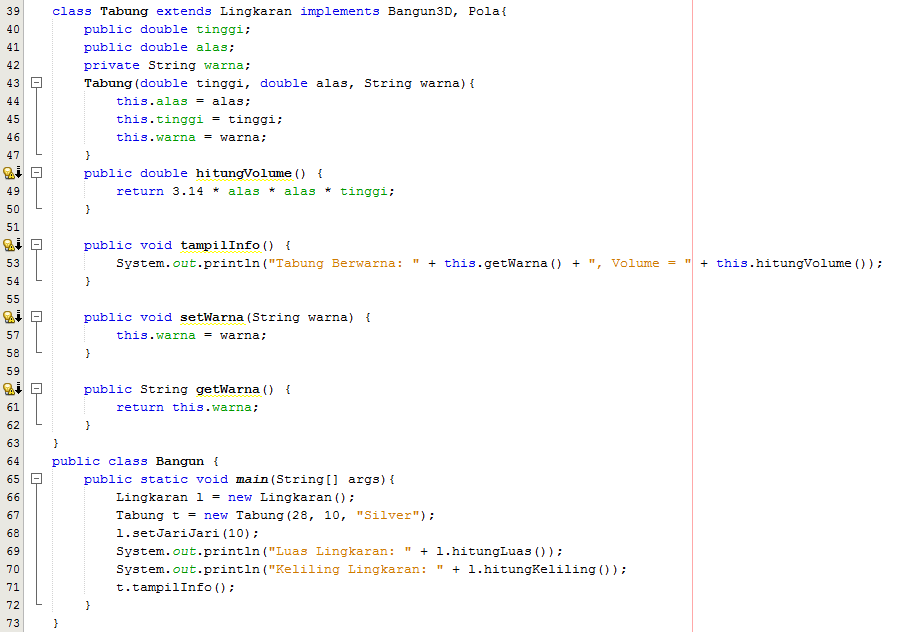


Berbeda dengan inheritance (Kelas Turunan) pada interface kita dapat melakukan multiple interface (mengimplementasikan banyak interface) dalam suatu kelas seperti contoh pada kelas MultiInterfaces diatas dan juga pada kelas Child pada praktikum sebelumnya juga.

1. **PEMBAHASAN LATIHAN**







Latihan yang diberikan yaitu membuat suatu program untuk meghitung luas dan keliling suatu lingkaran seperti pada kelas lingkaran sebelumnya, tetapi terdapat suatu interface baru yaitu interface Bangun3D untuk menghitung dan menampilkan volume suatu bangun ruang dan juga interface Pola yang digunakan untuk menentukan warna dari suatu bangun raung.

1. **TUGAS**

Jelaskan perbedaan antara kelas abstrak dan interface.

Jawab :

**Kelas abstrak** adalah kelas yang tidak dapat dibuat obyeknya (diinstantiasi). Hal ini dikarenakan kelas tersebut belum lengkap karena memiliki minimal satu buah method yang tidak mempunyai definisi, hanya berupa deklarasi saja. Agar dapat diinstantiasi maka kelas turunannyalah yang harus membuat implementasinya (membuat definisi dari method yang belum lengkap tadi). Sebelum semua method dibuat definisinya, maka kelas yang mewarisi kelas abstrak juga tidak dapat diinstantiasi, artinya menjadi kelas abstrak juga.

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

1. **KESIMPULAN**

Penggunaan interface dapat dimanfaatkan untuk membuat suatu pola atau kelas yang berbentuk umum atau dapat digunakan dalam berbagai hal. Dengan interface yang miliki method dan konstanta yang abstract yang mana dapat berubah-ubah sesuai situasi dan kondisi yang dialami program dapat memudahkan pembentukan program menjadi suatu bentuk yang lebih sederhana dan maintenance yang lebih baik.

1. **LISTING**

*Terlampir*