18

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN KE-9

****

**Disusun oleh :**

**NAMA : FERDI DIRGANTARA**

**NIM : 175410039**

**JURUSAN : TEKNIK INFORMATIKA**

**JENJANG : S1**

LABORATORIUM TERPADU

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTERAKAKOMYOGYAKARTA

2019

1. **TEORI SINGKAT**

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.

Biasanya suatu kelas dijadikan abstrak karena kelas tersebut terlalu umum. Sebagai contoh kelas makhluk hidup berisi method-method seperti : bernafas, makan, dan masih banyak lagi. Method-method tersebut terlalu abstrak untuk dibuat definisinya sehingga perlu dibuat kelas khusus yang merupakan turunan dari kelas makhluk hidup, yang mengimplementasikan method-method  
diatas.

Deklarasi kelas abstrak :

[modifier] abstract class NamaKelasAbstract {

*daftar field*

deklarasi method-method;

}

Implementasi kemudian dilakukan pada class turunannya:

class Turunan extends NamaKelasAbstract {

definisi method-method;

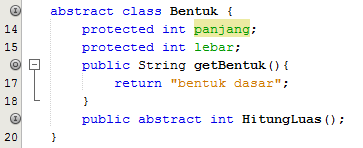
…

}

Beberapa hal yang perlu diperhatikan adalah sebagai berikut :

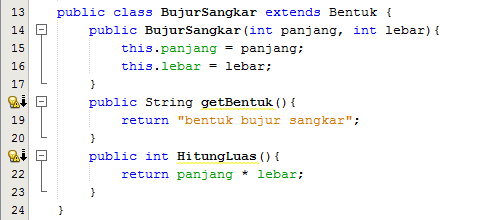
1. Class abstrak tidak dapat dibuatkan instan atau objeknya menggunakan keyword new.
2. Sebuah class dapat dideklarasikan sebagai class abstrak walaupun tidak memiliki method abstrak.
3. Variabel dengan tipe class abstrak tetap bisa diciptakan, tetapi harus mereferensi ke subclass dari lass abtrak tersebut yang tentunya tidak abstrak.
4. **PEMBAHASAN LISTING**

Kelas Bentuk



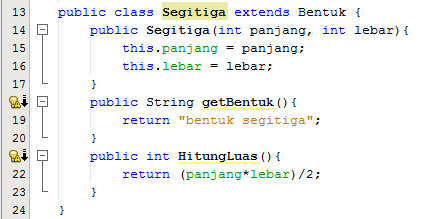
Kelas Bentuk diatas merupakan kelas abstract yang memiliki attribute tipe integer dengan nama “panjang” dan “lebar”, dengan method *getBentuk()* yang digunakan untuk memberikan nilai balik berupa *String* sebagai penanda bentuk dasar dan sebuah method abstract dengan *return value* bertipe *integer* yang yaitu method *HitungLuas()* yang nantinya akan digunakan untuk meletakkan rumus/formula dari masing-masing bentuk dalam menghitung luas.

Kelas BujurSangkar



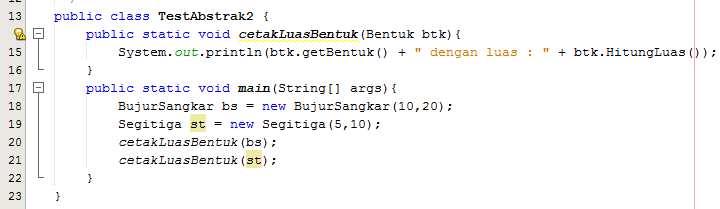
Kelas BujurSangkar turunan dari kelas Bentuk diatas memiliki konstruktor untuk menerima nilai panjang dan lebar, nilai parameter yang diterima ditampung kedalam variabel panjang dan variabel lebar yang terdapat pada kelas Bentuk. Dilakukan suatu method *override* pada method *getBentuk()* dengan nilai balik berupa *String* yaitu pesan “bentuk bujur sangkar” dan method *HitungLuas()* yang digunakan untuk menghitung luas bujur sangkar yaitu panjang x lebar.

Kelas Segitiga



Kelas Segitiga turunan dari kelas Bentuk diatas memiliki konstruktor untuk menerima nilai panjang dan lebar, nilai parameter yang diterima ditampung kedalam variabel panjang dan variabel lebar yang terdapat pada kelas Bentuk. Dilakukan suatu method *override* pada method *getBentuk()* dengan nilai balik berupa *String* yaitu pesan “bentuk segitiga” dan method *HitungLuas()* yang digunakan untuk menghitung luas segitiga yaitu (panjang x lebar) / 2.

Kelas TestAbstrak2

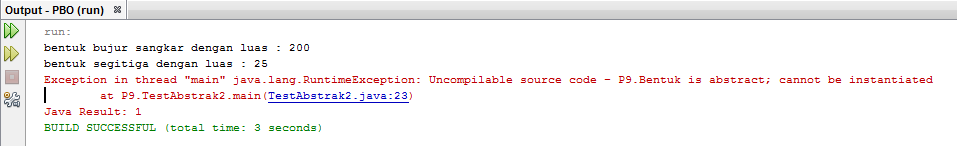


Pada kelas TestAbstrak2 kita melakukan pengujian pada masing-masing kelas turunan yang ada yaitu menghitung luas bujur sangkar dan luas segitiga. Nilai masing-masing bentuk dimasukkan pada saat instansiasi objek dengan menggunakan operator *new* yaitu pada objek “bs” untuk kelas BujurSangkar dan objek “st” untuk kelas Segitiga.

Untuk menampilkan hasil perhitungan luas masing-masing bentuk dibuatkan method *cetakInfoBentuk(Bentuk btk)* yang akan mengambil nama bentuk dengan memanggil method *getBentuk()* dan nilai balik perhitungan luas dengan memanggil method *HitungLuas()* yang terdapat pada masing-masing kelas. Method tersebut dipanggil pada main method dan menerima nilai parameter berupa objek dari masing-masing kelas.

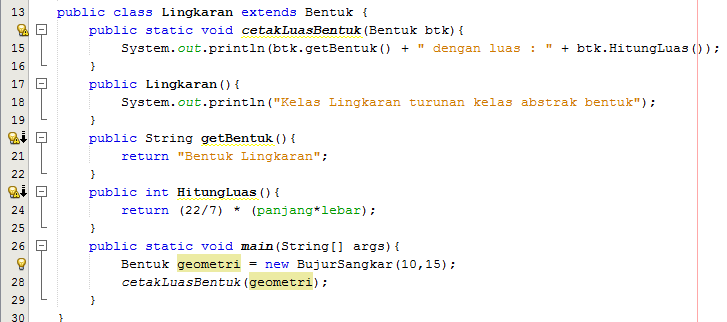
Output yang diperoleh berupa luas bujur sangkar dan luas segitiga yang mana diperoleh dari masing-masing perhitungan nilai parameter yang diberikan pada saat instansiasi objek.

Buat Objek dari kelas Bentuk, berikan pejelasan mengapa terjadi kesalahan



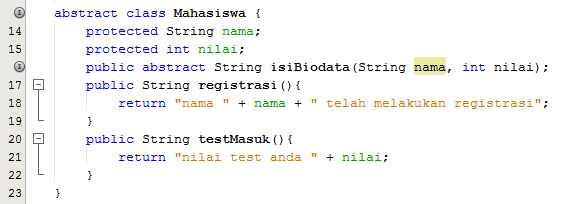
Kelas Bentuk tidak dapat di instansiasi karena merupakan suatu kelas *abstract* yang menyebabkan program tidak dapat melakukan kompilasi program.

Kelas Lingkaran

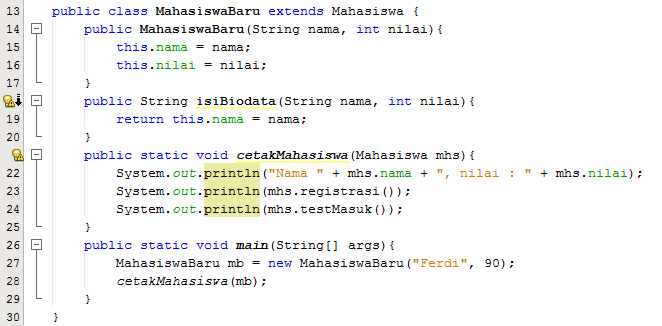


Kelas Lingkaran turunan dari kelas Bentuk diatas memiliki konstruktor untuk menerima nilai panjang dan lebar, nilai parameter yang diterima ditampung kedalam variabel panjang dan variabel lebar yang terdapat pada kelas Bentuk. Dilakukan suatu method *override* pada method *getBentuk()* dengan nilai balik berupa *String* yaitu pesan “bentuk lingkaran” dan method *HitungLuas()* yang digunakan untuk menghitung luas segitiga yaitu (22/7) x (panjang x lebar).

1. **PEMBAHASAN LATIHAN**



Kelas Mahasiswa diatas merupakan kelas abstract yang memiliki attribute nama dan nilai, kelas diatas merupakan bentuk umum dari mahasiswa, dari kelas diatas terdapat method abstract yaitu method *isiBiodata()* yang digunakan untuk menerima biodata mahasiswa nantinya dan juga terdapat method *registrasi* yang digunakan untuk memberikan nilai balik berupa informasi bahwa nama mahasiswa yang ditulis telah melakukan registrasi dan menampilkan nilai test mahasiswa denga menggunakan method *testMasuk()* yang kedua method tersebut merupakan method dengan *return value* berupa *String*.

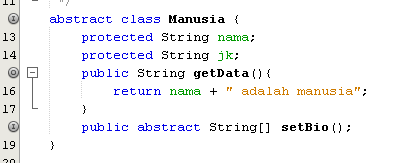


Kelas MahasiswaBaru digunakan untuk menguji kelas abstract dari kelas Mahasiswa dimana dari kelas diatas terdapat konstruktor yang akan menerima data pada saat instansiasi objek untuk kelas MahasiswaBaru dan juga terdapat method override dari method isiBiodata() serta terdapat method cetakMahasiswa juga yang digunakan untuk menampilkan data dari nilai yang telah diberikan pada saat instansiasi objek MahasiswaBaru diatas dengan nama objek yaitu “mb” dan memiliki parameter nama “Ferdi” dengan nilai 90.

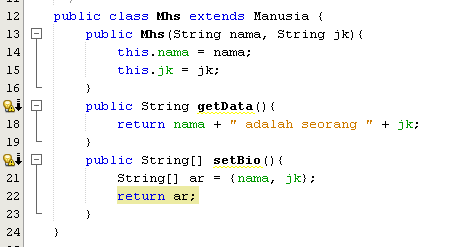
1. **TUGAS**

Membuat Kelas dan Method abstract untuk manusia dan melakukan pengujian

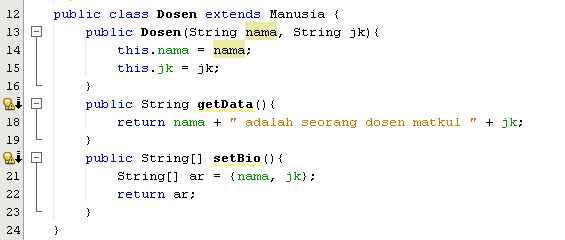
Kelas Manusia



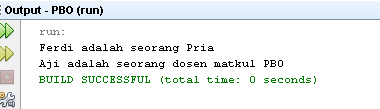
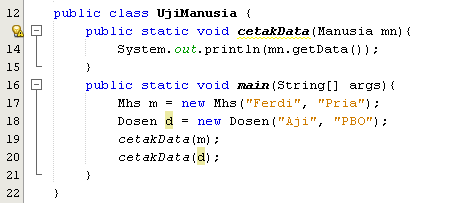
Kelas Mhs



Kelas Dosen



Kelas UjiManusia



Pengujian diatas menampilkan pesan “Ferdi adalah seorang Pria” dan juga “Aji adalah seorang dosen matkul PBO”. Nilai-nilai atau output tersebut diperoleh dari instansiasi dari kedua objek yang dilakukan pada kelas UjiManusia diatas dengan menambahkan method cetakData() yang digunakan untuk menampilkan data yang terdapat pada objek kelas Mhs dan kelas Dosen.

1. **KESIMPULAN**

Kelas abstrak memudahkan pembuatan program dengan memanfaatkan inheritance dan instansiasi dari suatu kelas umum. Kelas tersebut kemudian dapat memiliki berbagai macam kelas turunan yang sesuai dengan kelas umum yang dibuat dalam artian kelas turunan tersebut masih sejenis dengan kelas umum namun lebih spesifik disbanding kelas umum sebelumnya.

1. **LISTING**

*Terlampir*