

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN KE-9



Disusun Oleh :

NAMA : Raden Isnawan Argi Aryasatya

NIM : 195410257

JURUSAN : Teknik Informatika

JENJANG : S1

KELAS : TI-5

Laboratorium Terpadu
Sekolah Tinggi Manajemen Informatika Komputer
AKAKOM
YOGYAKARTA

2021

PERTEMUAN KE-9 (KELAS ABSTRAK)

TUJUAN

Dapat membuat dan menggunakan kelas abstrak

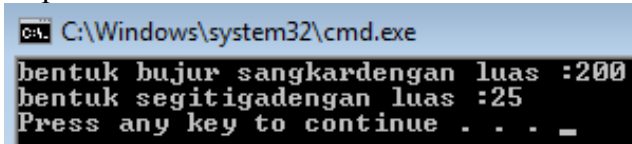
DASAR TEORI

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 turunannya 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.

PRAKTIK 1: Penggunaan kelas abstrak berikut:

```
TestAbstrak2.java x
1  abstract class Bentuk {
2      protected int panjang;
3      protected int lebar;
4      public String getBentuk(){
5          return "bentuk dasar";
6      }
7      public abstract int HitungLuas();
8  }
9
10 class BujurSangkar extends Bentuk{
11     public BujurSangkar(int panjang, int lebar){
12         this.panjang = panjang;
13         this.lebar = lebar;
14     }
15     public String getBentuk(){
16         return "bentuk bujur sangkar";
17     }
18     public int HitungLuas(){
19         return panjang * lebar;
20     }
21 }
22
23 class Segitiga extends Bentuk{
24     public Segitiga(int panjang, int lebar){
25         this.panjang = panjang;
26         this.lebar = lebar;
27     }
28     public String getBentuk(){
29         return "bentuk segitiga";
30     }
31
32     public int HitungLuas(){
33         return (panjang * lebar)/2;
34     }
35 }
36
37 class TestAbstrak2{
38     public static void cetakLuasBentuk(Bentuk btk){
39         System.out.println(btk.getBentuk()+"dengan luas :"+ btk.HitungLuas());
40     }
41
42     public static void main (String[] args){
43         BujurSangkar bs = new BujurSangkar(10,20);
44         Segitiga st = new Segitiga(5,10);
45         cetakLuasBentuk(bs);
46         cetakLuasBentuk(st);
47     }
48 }
49
50
```

output:



```
C:\Windows\system32\cmd.exe
bentuk bujur sangkardengan luas :200
bentuk segitigadengan luas :25
Press any key to continue . . . _
```

penjelasan:

Pada program di atas, kita membuat kelas abstrak beserta kelas-kelas turunannya. Lalu, apa itu kelas abstrak? Kelas abstrak adalah kelas yang masih dalam bentuk abstrak. Karena bentuknya masih abstrak, dia tidak bisa dibuat langsung menjadi objek. Kelas abstrak tersebut tidak dapat dibuat objeknya.

Untuk pembuatan programnya, pertama kita membuat kelas abstrak bernama **Bentuk** yang di dalamnya ada beberapa atribut yaitu panjang dan lebar yang bertipe data integer dan memiliki modifier **protected**. Modifier **protected** akan membuat member dan class hanya bisa diakses dari: class itu sendiri; sub class atau class anak; package (class yang berada satu package dengannya). Kemudian kita buat accessor method dengan menuliskan **public String getBentuk()** yang didalamnya di return sebuah kalimat "bentuk dasar". Setelah itu kita buat sebuah method abstrak. Method abstrak adalah method yang tidak memiliki implementasi atau tidak ada bentuk konkritnya. Kita deklarasikan method tersebut dengan menuliskan **public abstract int HitungLuas();**

Kemudian kita buat child class atau sub class bernama **BujurSangkar** yang diwarisi atribut dan method dari parent class **Bentuk**. Untuk itu kita tulis dengan baris kode class **BujurSangkar extends Bentuk**. **Extends** pada baris kode tersebut berfungsi untuk memanggil fungsi, method, dan atribut dari class parent (**Bentuk**), sehingga kita tidak perlu lagi membuat script yang sama pada class yang akan kita buat dengan class yang kita buat sebelumnya. Di dalamnya kita buat konstruktor yang memiliki parameter **int panjang** dan **int lebar** yang ditulis dengan **public BujurSangkar(int panjang, int lebar)** lalu di dalam konstruktor tersebut kita akses variabel milik bentuk yaitu panjang dan lebar dengan **this.panjang = panjang;** dan **this.lebar = lebar;**. Kemudian kita buat method accessor yang memiliki return seperti berikut:

```
public String getBentuk(){
    return "bentuk bujur sangkar";
```

kemudian dengan method **HitungLuas()** milik class **Bentuk**, kita beri isi di dalamnya dengan mengembalikan/return berupa panjang dikali dengan lebar. Berikut baris kode nya:

```
public int HitungLuas(){
    return panjang * lebar;
```

untuk class **Segitiga** intinya sama saja seperti class **BujurSangkar**, bedanya hanya di return value method accessor dan perhitungan dalam method **HitungLuas()**. Metode accessor nya ditulis seperti berikut:

```
public String getBentuk(){
    return "bentuk bujur sangkar";
```

Kemudian di dalam class **Segitiga**, method **HitungLuas** ditulis seperti berikut:

```
public int HitungLuas(){
    return(panjang * lebar)/2;
```

Setelah membuat kedua class tersebut, terakhir kita membuat script main method untuk membuat atau meng-instantiasi objek yang bernama **TestAbstrak2**. Di dalamnya kita buat method bermodifikasi **public** yang tidak memiliki return value dengan menuliskan baris kode **public static void cetakLuasBentuk (Bentuk btk)**. **Bentuk btk** artinya **Bentuk** akan diwakilkan dengan atribut bernama **btk**. Lalu kita cetak output seperti berikut:

```
System.out.println(btk.getBentuk()+"dengan luas :"+ btk.HitungLuas());
```

Lalu kita buat objeknya. Kita buat objek bs untuk class BujurSangkar, lalu kita buat objek st untuk class Segitiga. Kemudian diberi nilai/argumen. Berikut penulisan kode nya:

```
BujurSangkar bs = new BujurSangkar(10,20);
```

```
Segitiga st = new Segitiga(5,10);
```

Terakhir cetak output dengan:

```
cetakLuasBentuk(bs);
```

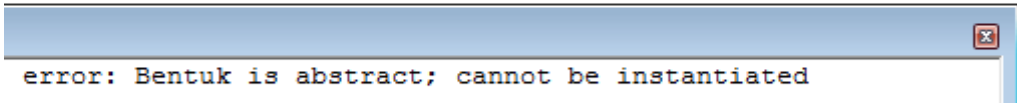
```
cetakLuasBentuk(st);
```

1. Buat objek dari kelas Bentuk, berikan penjelasan mengapa terjadi kesalahan

```
class TestAbstrak2{
    public static void cetakLuasBentuk(Bentuk btk){
        System.out.println(btk.getBentuk()+"dengan luas :"+ btk.HitungLuas());
    }

    public static void main (String[] args){
        Bentuk bt = new Bentuk (4,8);
        BujurSangkar bs = new BujurSangkar(10,20);
        Segitiga st = new Segitiga(5,10);
        cetakLuasBentuk(bt);
        cetakLuasBentuk(bs);
        cetakLuasBentuk(st);
    }
}
```

error:

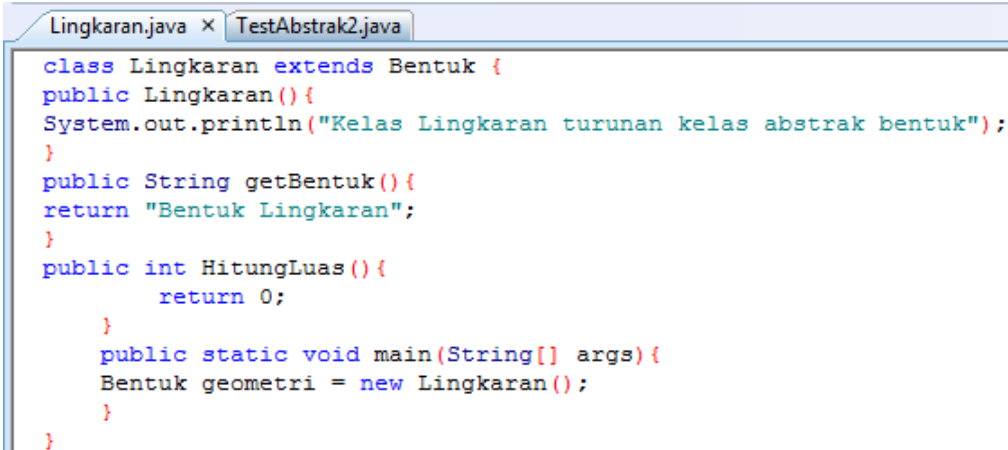


```
error: Bentuk is abstract; cannot be instantiated
```

penjelasan:

Kesalahan tersebut terjadi karena kita tidak dapat menginstansiasi atau membuat objek baru dari class abstrak. Dengan kata lain, class abstrak tidak dapat dibuatkan instan atau objeknya menggunakan keyword new. Untuk membuat objek dan mengimplementasikannya, kita harus menggunakan sub class (pada program tersebut, contohnya adalah subclass BujurSangkar dan subclass Segitiga).

2. Berikan penjelasan kenapa tidak terjadi kesalahan pada pembuatan objek geometri



```
Lingkaran.java x TestAbstrak2.java

class Lingkaran extends Bentuk {
    public Lingkaran(){
        System.out.println("Kelas Lingkaran turunan kelas abstrak bentuk");
    }
    public String getBentuk(){
        return "Bentuk Lingkaran";
    }
    public int HitungLuas(){
        return 0;
    }
    public static void main(String[] args){
        Bentuk geometri = new Lingkaran();
    }
}
```

output:

```
C:\Windows\system32\cmd.exe
Kelas Lingkaran turunan kelas abstrak bentuk
Press any key to continue . . . _
```

penjelasan:

Program tersebut bisa dijalankan karena kita menggunakan subclass Lingkaran untuk membuat dan mengimplementasikan objeknya. Kita tidak langsung membuat objek dari kelas Bentuk, melainkan dengan kelas Bentuk yang diinisialisasi dengan kelas Lingkaran yaitu dengan baris kode Bentuk geometri = new Lingkaran();. Maka dari itu, dalam program tersebut kita dapat menginstansiasi objek menggunakan keyword new.

LATIHAN

step by step pengerjaan:

1. Buatlah sebuah class abstract bernama Mahasiswa memiliki method sbb
 - isiBiodata(String nama, int nilai) sebagai method abstract
 - registrasi(), menampilkan tulisan ke layar : “nama.....telah melakukan registrasi”
 - testMasuk(), menampilkan tulisan ke layar : “nilai test anda”
2. Tambahkan variabel nama dan nilai pada Mahasiswa
3. Buatlah class MahasiswaBaru yang melakukan extending dari class Mahasiswa. Deklarasi ulang method isiBiodata, bodi method akan menampilkan ke layar biodata anda (“nama:....., nilai :,pilihan jurusan:.....asal sekolah :”)
4. Tambahkan main method pada MahasiswaBaru

SCRIPT PROGRAM:

```
LatihanAbstrak.java x
1 //-----batas awal nomor 1-----//
2 abstract class Mahasiswa{
3     protected String nama; //nomor 2//
4     protected int nilai; //nomor 2//
5
6     public abstract void isiBiodata();
7     public void registrasi(){
8         System.out.println("nama "+ nama +" telah melakukan registrasi");
9     }
10    public void testMasuk(){
11        System.out.println("nilai test anda "+ nilai);
12    }
13 }
14 //-----batas akhir nomor 1-----//
15
16
17 //-----batas awal nomor 3-----//
18 class MahasiswaBaru extends Mahasiswa{
19     public MahasiswaBaru(String nama, int nilai){
20         this.nama = nama;
21         this.nilai = nilai;
```

```

22     }
23     public void isiBiodata(){
24         System.out.println("nama: "+nama+" ,nilai : "+nilai);
25         super.registrasi();
26         super.testMasuk();
27     }
28 }
29 //-----batas akhir nomor 3-----//
30
31
32 //-----batas awal nomor 4-----//
33 public class LatihanAbstrak{
34     public static void main(String[] args){
35         Mahasiswa mb = new MahasiswaBaru("ahmad",80);
36         mb.isiBiodata();
37     }
38 }
39 //-----batas akhir nomor 4-----//

```

output:

```

C:\Windows\system32\cmd.exe
nama: ahmad ,nilai : 80
nama ahmad telah melakukan registrasi
nilai test anda 80
Press any key to continue . . . _

```

penjelasan:

Pada program di atas, pada nomor 1 kita membuat sebuah kelas abstrak bernama Mahasiswa yang ditulis dengan **abstract class Mahasiswa{**. Di dalamnya ada tiga method utama yang salah satunya adalah method abstrak. Apa itu abstract method? Abstract method adalah method yang tidak memiliki body (hanya deklarasi method) sehingga method tersebut harus dibuatkan di dalam sebuah subclass supaya bisa dimodif untuk digunakan atau diimplementasikan.

Pertama, kita membuat method abstrak tersebut terlebih dahulu dengan nama isiBiodata() yang tidak memiliki parameter maupun body, kita tulis dengan baris kode **public abstract void isiBiodata();**.

Kedua, kita buat method bernama registrasi dengan **public void registrasi()**. Method tersebut nantinya jika akan mencetak sebuah output yang ditulis seperti berikut:
System.out.println("nama "+ nama +" telah melakukan registrasi");

Ketiga, kita buat method bernama testMasuk dengan **public void testMasuk()**. Method tersebut nantinya jika dipanggil akan mencetak sebuah output yang ditulis seperti berikut:
System.out.println("nilai test anda "+ nilai);

pada nomor 2, kita membuat dua atribut/variabel yang memiliki modifier protected yaitu String nama dan int nilai. Modifier protected akan membuat member dan class hanya bisa diakses dari: class itu sendiri; sub class atau class anak; package (class yang berada satu package dengannya). Berikut penulisan baris kode deklarasi kedua variabel tersebut:
protected String nama;
protected int nilai;

Pada nomor 3, kita membuat class MahasiswaBaru yang akan diwarisi method dan atribut dari parent class Mahasiswa. Extends pada baris kode tersebut berfungsi untuk memanggil fungsi, method, dan atribut dari class parent (Mahasiswa), sehingga kita tidak perlu lagi membuat script yang sama pada class yang akan kita buat dengan class yang kita buat sebelumnya. Di dalam class

tersebut kita buat konstruktor yang memiliki parameter String nama dan int nilai yang ditulis dengan **public MahasiswaBaru(String nama, int nilai)** lalu kedua variabel tersebut kita akses dengan **this.nama = nama** dan **this.nilai = nilai**.

Lalu kita buat sebuah method yang meng-override isiBiodata yang dituliskan dengan **public void isiBiodata()**. Di dalamnya kita munculkan pencetak kalimat yaitu dengan menuliskan `System.out.println("nama: "+nama+" ,nilai : "+nilai);`. Kemudian kita panggil method registrasi dan method testMasuk dengan keyword super (karena mengambil atau mengakses dari kelas parent) untuk memunculkan output yang dimiliki oleh kedua method tersebut.

Terakhir kita membuat sebuah main method yaitu public class LatihanAbstrak. Eksekusi kita awali dengan **public static void main(String[]args)**. Lalu kita buat objek bernama mb sekaligus memberi argummen/nilai yaitu dengan menuliskan:

Mahasiswa mb = new MahasiswaBaru("ahmad",80);

kemudian kita tampilkan output dengan memanggil **mb.isiBiodata();**

TUGAS

step by step pengerjaan:

1. Buatlah kelas abstrak Manusia yang berisi deskripsi umum dari manusia. Kemudian buatlah kelas Mahasiswa dan kelas Dosen yang merupakan turunan dari kelas Manusia, dan masing-masing kelas menangani data mahasiswa dan dosen.
2. Buat objek dari kelas yang dibuat

SCRIPT PROGRAM:

```
TugasAbstrak.java x
1  //=====batas awal nomor 1=====//
2  abstract class Manusia {
3      protected String nama;
4      protected String jk;
5      protected String asal;
6      protected String tgl_lahir;
7
8      public Manusia(String nama,String jk,
9                      String asal,String tgl_lahir){
10         this.nama = nama;
11         this.jk = jk;
12         this.asal = asal;
13         this.tgl_lahir = tgl_lahir;
14     }
15
16     public void dataMhs(){
17         System.out.println("=====DATA MAHASISWA=====");
18     }
19
20     public void dataDsn(){
21         System.out.println("=====DATA DOSEN=====");
22     }
23
24     public String getNama(){
25         return nama;
26     }
27     public String getJk(){
28         return jk;
29     }
```

```

30     public String getAsal(){
31         return asal;
32     }
33     public String getTgl_lahir(){
34         return tgl_lahir;
35     }
36     public abstract void DataDiri();
37 }
38
39 class Mahasiswa extends Manusia{
40     protected String prodi;
41     protected int nim;
42
43     public Mahasiswa(String nama,String jk,String asal,
44         String tgl_lahir, String prodi, int nim){
45         super(nama,jk,asal,tgl_lahir);
46         this.prodi = prodi;
47         this.nim = nim;
48     }
49
50     public String getProdi(){
51         return prodi;
52     }
53     public int getNim(){
54         return nim;
55     }
56
57     @Override
58     public void DataDiri() {
59         super.dataMhs();
60         System.out.println("nama: "+getNama());
61         System.out.println("jenis kelamin: "+getJk());
62         System.out.println("daerah asal: "+getAsal());
63         System.out.println("tanggal lahir: "+getTgl_lahir());
64         System.out.println("progam studi: "+getProdi());
65         System.out.println("NIM: "+getNim());
66         System.out.println("");
67     }
68
69 }
70
71 class Dosen extends Manusia{
72     protected String gelar;
73     protected String matkul;
74     public Dosen(String nama,String jk,String asal,
75         String tgl_lahir, String gelar, String matkul){
76         super(nama,jk,asal,tgl_lahir);
77         this.gelar = gelar;
78         this.matkul = matkul;
79     }
80     protected String getGelar(){
81         return gelar;
82     }
83     protected String getMatkul(){
84         return matkul;
85     }
86
87     @Override
88     public void DataDiri() {
89         super.dataDsn();
90         System.out.println("nama: "+getNama());
91         System.out.println("jenis kelamin: "+getJk());
92         System.out.println("daerah asal: "+getAsal());
93         System.out.println("tanggal lahir: "+getTgl_lahir());
94         System.out.println("gelar: "+getGelar());
95         System.out.println("matkul: "+getMatkul());
96     }
97 }
98 //====batas akhir nomor 1====//
99
100 //====batas awal nomor 2====//
101 class TugasAbstrak{

```



```

TugasAbstrak.java x
102 public static void main(String[] args) {
103     Mahasiswa mhs = new Mahasiswa("Joe Frazier","L","Arizona",
104                                   "9 Maret 2000","Informatika",195410257);
105     Dosen dsn = new Dosen("Etha Sintha","P","Los Angeles",
106                           "3 Januari 1993","Magister Ilmu Komputer","PBO");
107     mhs.DataDiri();
108     dsn.DataDiri();
109 }
110 }
111 //====batas akhir nomor 2====//

```

output:

```

C:\Windows\system32\cmd.exe
====DATA MAHASISWA====
nama: Joe Frazier
jenis kelamin: L
daerah asal: Arizona
tanggal lahir: 9 Maret 2000
program studi: Informatika
NIM: 195410257

====DATA DOSEN====
nama: Etha Sintha
jenis kelamin: P
daerah asal: Los Angeles
tanggal lahir: 3 Januari 1993
gelar: Magister Ilmu Komputer
matkul: PBO
Press any key to continue . . .

```

penjelasan singkat:

pertama kita buat kelas abstrak bernama Manusia yang memiliki 4 variabel yaitu String nama, String jk, String asal, String tgl_lahir. Semuanya bermodifler protected. Lalu buat konstruktor Manusia yang memiliki parameter 4 variabel tersebut. Kemudian kita akses 4 variabel tersebut dengan this.nama = nama; this.jk = jk; this.asal = asal; this.tgl_lahir = tgl_lahir;

buat dua method untuk memunculkan "====DATA MAHASISWA====" sebelum memunculkan data mahasiswa dan "====DATA DOSEN====" sebelum memunculkan data dosen.

Kemudian buat method accessor untuk semua variabel. Contoh salah satunya:

```

public String getNama(){
    return nama;
}

```

Lalu kita buat method abstrak nya yaitu **public abstract void DataDiri();** yang nantinya akan berisi kalimat-kalimat yang akan ditampilkan di output.

Kemudian buata kelas Mahasiswa yang diwariskan method dan atribut dari kelas Manusia. Tambahkan dua variabel baru yaitu prodi dan nim. Setelah itu buat konstruktor yang memiliki parameter berupa variabel milik Manusia sekaligus milik Mahasiswa. Lalu seluruh variabel itu diakses. Khusus yang punya manusia, kita gunakan super. Yang lain gunakan this.

Buat method accessor untuk prodi dan nim. Contoh:

```

public String getProdi(){
    return prodi;
}
public int getNim(){
    return nim;
}

```

Kemudian kita override method `DataDiri` dan panggil method `dataMhs`, kemudian method `DataDiri` diisi dengan kalimat yang diinisialisasi dengan seluruh variabel manusia + mahasiswa

Untuk class `Dosen` konsepnya sama saja, hanya penambahan variabel nya adalah gelar dan matkul.

Terakhir kita buat main method yaitu **class TugasAbstrak**. Eksekusi kita awali dengan menuliskan **public static void main(String[] args)**.

kita buat objek `mhs` sekaligus menambahkan argumen/nilai dengan cara menuliskan baris kode:

```
Mahasiswa mhs = new Mahasiswa("Joe Frazier", "L", "Arizona",  
                                "9 Maret 2000", "Informatika", 195410257);
```

kita buat objek `dsn` sekaligus menambahkan argumen/nilai dengan cara menuliskan baris kode:

```
Dosen dsn = new Dosen("Etha Sintha", "P", "Los Angeles",  
                      "3 Januari 1993", "Magister Ilmu Komputer", "PBO");
```

Lalu kita tampilkan output dengan:

```
mhs.DataDiri();  
dsn.DataDiri();
```

KESIMPULAN

Pada pertemuan kali ini, kita belajar tentang kelas abstrak. 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.

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 class abtrak tersebut yang tentunya tidak abstrak.