

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN KE-3



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-3 (METHOD)

TUJUAN

Dapat membuat dan menggunakan berbagai method

DASAR TEORI

Sebuah method menjelaskan behaviour dari sebuah object. Method juga dikenal sebagai fungsi atau prosedur. Pendeklarasian method biasa dituliskan seperti berikut ini :

```
<modifier> <returnType> <name>(<parameter>*) {  
    <statement>*  
}
```

dimana,

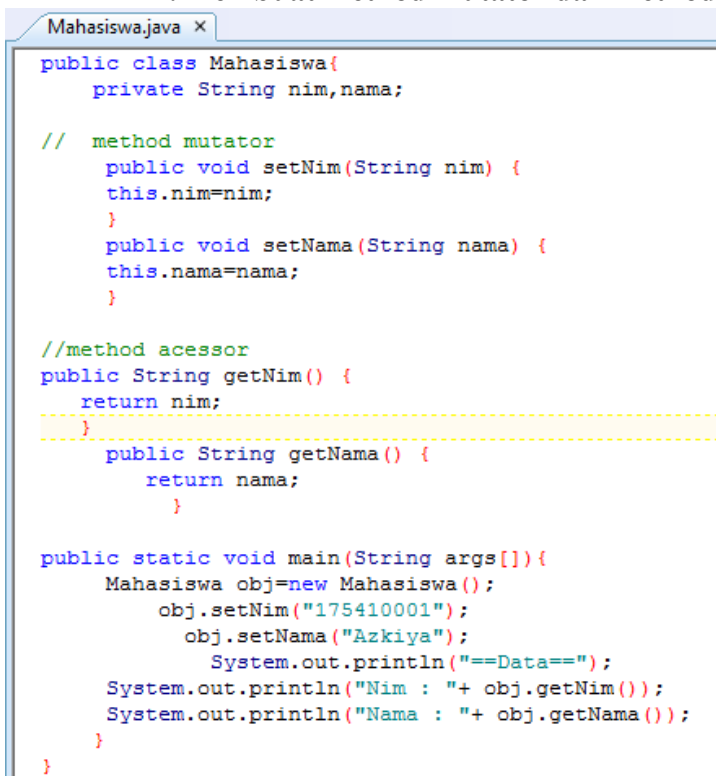
<modifier> dapat menggunakan beberapa modifier yang berbeda

<returnType> dapat berupa seluruh tipe data, termasuk void

<name> identifi er atas class

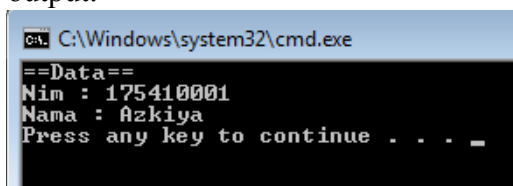
<parameter> ::= <tipe_parameter> <nama_parameter>[,]

PRAKTIK 1: membuat method mutator dan method acesor



```
Mahasiswa.java x  
public class Mahasiswa{  
    private String nim,nama;  
  
    // method mutator  
    public void setNim(String nim) {  
        this.nim=nim;  
    }  
    public void setName(String nama) {  
        this.nama=nama;  
    }  
  
    //method acesor  
    public String getNim() {  
        return nim;  
    }  
    public String getName() {  
        return nama;  
    }  
  
    public static void main(String args[]){  
        Mahasiswa obj=new Mahasiswa();  
        obj.setNim("175410001");  
        obj.setName("Azkiya");  
        System.out.println("==Data==");  
        System.out.println("Nim : "+ obj.getNim());  
        System.out.println("Nama : "+ obj.getName());  
    }  
}
```

output:



```
C:\Windows\system32\cmd.exe  
==Data==  
Nim : 175410001  
Nama : Azkiya  
Press any key to continue . . . _
```

penjelasan:

dalam program ini, public class Mahasiswa berfungsi untuk mendeklarasikan class bersifat public bernama Mahasiswa yang gunanya untuk mengumpulkan fungsi maupun variabel/atribut. Terdapat 2 atribut yang memiliki modifier private yaitu nim dan nama. Modifier private akan membuat member hanya bisa diakses oleh dari dalam class itu sendiri. Modifier private tidak bisa diberikan kepada class, enum, dan interface. Modifier private hanya bisa diberikan kepada member class.

Lalu kita menggunakan method mutator. Merupakan sebuah metode yang digunakan untuk menetapkan nilai dari sebuah private field. Skema ini biasanya ditandai dengan awalan nama "set". Di dalam program tersebut, private field yang akan kita masukkan dalam method mutator adalah nim dengan **setNim(String nim)** dan nama dengan **setNama(String nama)**. Lalu untuk mengakses komponen dalam class tersebut, gunakan "this" seperti **this.nim=nim;** dan **this.nama=nama;**. Metode mutator tidak memiliki nilai balik (return), sebab metode ini merupakan pengaturan parameter yang akan digunakan oleh koresponden private fieldnya.

Yang kedua adalah method accessor. Method acesor merupakan sebuah metode yang digunakan untuk mengembalikan nilai dari sebuah private field. Skema ini biasanya ditandai dengan awalan nama "get". Sebagai contoh pada program tersebut, misalnya kita akan menambahkan metode accessor untuk nim dan nama dengan **getNim()** dan **getNama()**. Metode ini selalu mengembalikan value/nilai dengan tipe data yang sama sesuai korespondensi private fieldnya, maka dari itu selalu ada baris kode return di dalam method accessor seperti potongan kode berikut:

```
public String getNim() {  
    return nim;  
}  
public String getNama() {  
    return nama;  
}
```

Lalu deklarasikan **public static void main(String args[]){** untuk mengawali eksekusi . public, yang berarti metode ini bisa dipanggil dan digunakan didalam Class atau diluar Class. public, yang berarti metode ini bisa dipanggil dan digunakan didalam Class atau diluar Class. void, yang berarti bahwa metode ini tidak mengirimkan nilai balik. String[], adalah tipe data objek yang menangani serangkaian karakter-karakter yang berjenis array. args, adalah variabel objek.

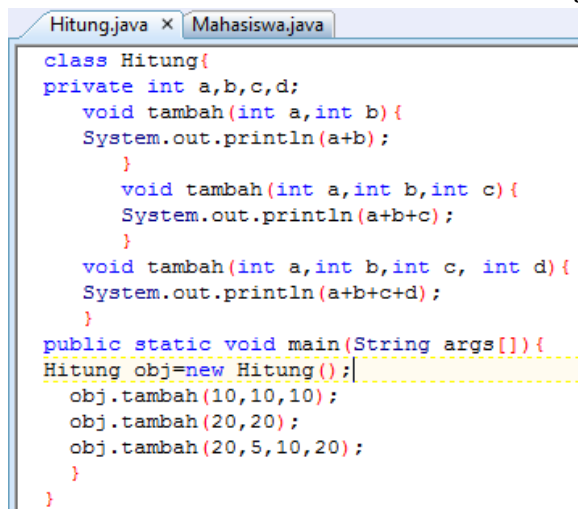
Kita buat objek bernama obj dengan **Mahasiswa obj=new Mahasiswa();**. Lalu di dalam objek tersebut kita gunakan kembali nim dan nama yang sudah melewati metode mutator tadi. Kedua atribut itu kita modifikasi nilainya supaya nim memiliki nilai 175410001 dan nama memiliki nilai Azkiya. Hal itu dilakukan dengan **obj.setNim("175410001");** dan **obj.setNama("Azkiya");**

Yang terakhir kita tinggal mencetak output dengan:

```
System.out.println("==Data==");  
System.out.println("Nim : "+ obj.getNim());  
System.out.println("Nama : "+ obj.getNama());
```

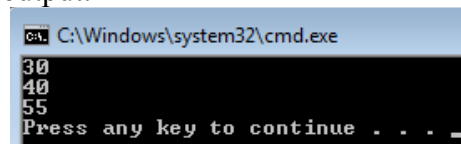
****PRAKTIK 2 DAN SETERUSNYA DI HALAMAN SELANJUTNYA***

PRAKTIK 2: Memuat method overloading



```
class Hitung{
private int a,b,c,d;
    void tambah(int a,int b){
        System.out.println(a+b);
    }
    void tambah(int a,int b,int c){
        System.out.println(a+b+c);
    }
    void tambah(int a,int b,int c, int d){
        System.out.println(a+b+c+d);
    }
public static void main(String args[]){
    Hitung obj=new Hitung();
    obj.tambah(10,10,10);
    obj.tambah(20,20);
    obj.tambah(20,5,10,20);
}
```

output:



```
C:\Windows\system32\cmd.exe
30
40
55
Press any key to continue . . .
```

penjelasan:

Method overloading terjadi bila dalam satu class memiliki method dengan nama yang sama namun memiliki parameter yang berbeda. Jadi di dalam sebuah program Java kita dapat membuat nama method yang sama tetapi dengan parameter harus ada perbedaan.

Pertama-tama, kita buat Class bernama Hitung. class Hitung berfungsi untuk mendeklarasikan class bernama Hitung yang gunanya untuk mengumpulkan berbagai fungsi maupun variabel/atribut. Di dalam program di atas, di dalam class ada beberapa atribut dengan tipe data integer yang memiliki modifier private yaitu a, b, c, d yang ditulis dengan **private int a,b,c,d**;

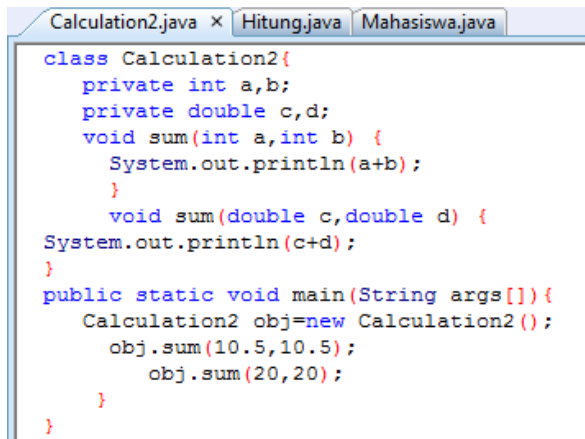
Selanjutnya, di dalam class Hitung kita membuat 3 method yang memiliki nama dan tipe data yang sama tetapi memiliki perbedaan pada parameternya. Pada contoh program di atas, kita membuat method tambah dengan parameter a dan b yaitu **void tambah(int a,int b)**. Selanjutnya kita melakukan overload method tambah dengan parameter a, b, dan c dengan **void tambah(int a,int b,int c)**. Lalu kita overload lagi dengan satu method terakhir yaitu method tambah dengan parameter a, b, c, dan d dengan **void tambah(int a,int b,int c, int d)**. Di dalam setiap method tambah tersebut, program mencetak hasil penjumlahan dari setiap parameter di dalamnya misalnya method tambah **void tambah(int a,int b)** akan menghasilkan hasil tambah a dan b. Hal itu dilakukan dengan **System.out.println(a+b);**. Pencetakan penjumlahan ini juga berlaku untuk 2 method tambah selanjutnya

Kita deklarasikan public static void main(String args[]) untuk mengawali eksekusi program. Setelah itu, kita membuat objek bernama obj dengan **Hitung obj=new Hitung()**;

Yang terakhir adalah kita hubungan objek obj dengan method tambah yaitu dengan menuliskan **obj.tambah**. Lalu kita beri nilai pada setiap parameter nya yaitu dengan:

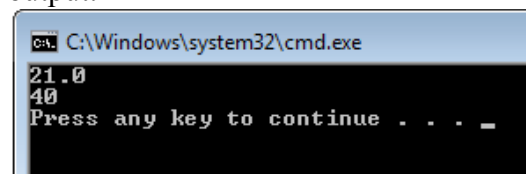
obj.tambah(10,10,10); : merupakan method overload pertama karena memiliki 3 parameter
obj.tambah(20,20); : merupakan method pertama karena memiliki 2 parameter
obj.tambah(20,5,10,20); : merupakan method overload kedua karena memiliki 4 parameter

PRAKTIK 3: membuat method overloading dengan perbedaan tipe parameter



```
Calculation2.java x Hitung.java Mahasiswa.java
class Calculation2{
    private int a,b;
    private double c,d;
    void sum(int a,int b) {
        System.out.println(a+b);
    }
    void sum(double c,double d) {
        System.out.println(c+d);
    }
    public static void main(String args[]){
        Calculation2 obj=new Calculation2();
        obj.sum(10.5,10.5);
        obj.sum(20,20);
    }
}
```

output:



```
ca. C:\Windows\system32\cmd.exe
21.0
40
Press any key to continue . . . _
```

penjelasan:

Pada method overloading kita juga bisa menggunakan tipe data parameter yang berbeda dari method yang pertama kita buat. Dalam kasus di atas, kita memiliki 4 atribut yaitu a, b, c, d. Atribut a dan b memiliki tipe data integer yang modifier nya private (terbatas). Sementara atribut c dan d memiliki tipe data double. Semua atribut itu ada di dalam sebuah class yang bernama Calculation2. Cara mendeklarasikan a,b,c,d adalah dengan **private int a,b;** dan **private double c,d;**

Setelah itu, kita buat method pertama nya yaitu **void sum(int a,int b)** yang akan dicetak dengan menjumlahkan a dan b yaitu dengan **System.out.println(a+b);**. Setelah method pertama dibuat, kita langsung membuat method overloading dengan tipe data parameter yang berbeda yaitu tipe data double. Kita buat method overloading tersebut dengan **void sum(double c,double d)** yang akan dicetak dengan menjumlahkan c dan d yaitu dengan menuliskan **System.out.println(c+d);**

Lalu tulis **public static void main(String args[])** sebelum kita membuat objek. Lalu kita buat objek bernama obj dengan cara menuliskan **Calculation2 obj=new Calculation2();**. Setelah itu kita masukkan nilai di dalam objek tersebut dengan:

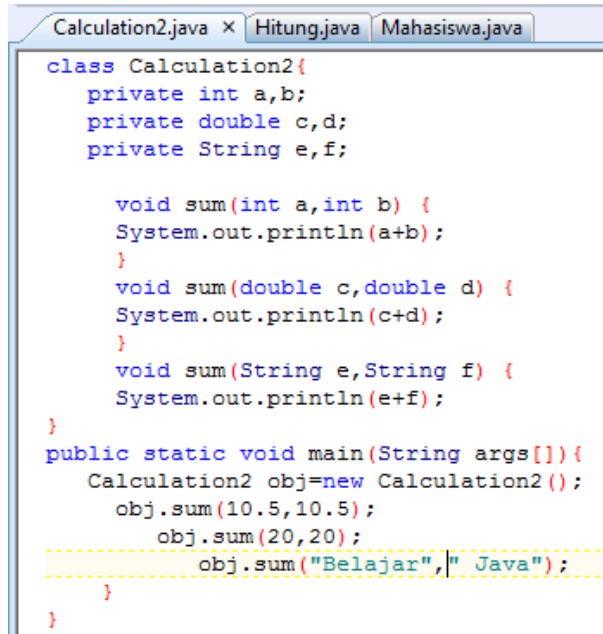
obj.sum(10.5,10.5); : method overloading bertipe data double karena angka nya desimal

obj.sum(20,20); : method pertama bertipe data integer karena angka nya bulat

****LATIHAN 1 DAN SETERUSNYA DI HALAMAN SELANJUTNYA***

LATIHAN

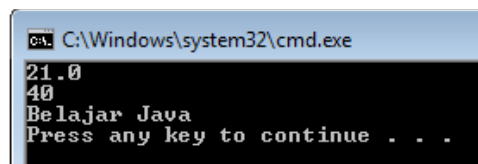
1. Modifikasi praktik 3 dengan menambahkan method sum dengan tipe parameter yang berbeda!



```
class Calculation2{
    private int a,b;
    private double c,d;
    private String e,f;

    void sum(int a,int b) {
        System.out.println(a+b);
    }
    void sum(double c,double d) {
        System.out.println(c+d);
    }
    void sum(String e,String f) {
        System.out.println(e+f);
    }
}
public static void main(String args[]){
    Calculation2 obj=new Calculation2();
    obj.sum(10.5,10.5);
    obj.sum(20,20);
    obj.sum("Belajar"," Java");
}
```

Output:



```
C:\Windows\system32\cmd.exe
21.0
40
Belajar Java
Press any key to continue . . .
```

penjelasan:

Pada program di atas, saya menambahkan 2 atribut yaitu e dan f yang bertipe data String dan memiliki modifier **private**. Atribut e dan f ada di dalam class Calculation2. Kedua atribut itu dideklarasikan dengan **private String e,f;**. Untuk membuat method pertama dan method overloading nya sudah dijelaskan di praktik 3, jadi langsung saja kita membuat method overloading selanjutnya yaitu dengan **void sum(String e,String f)** yang akan dicetak dengan menjumlahkan nilai dari e dan f yaitu dengan menuliskan **System.out.println(e+f);**

Lalu tulis public static void main(String args[]) sebelum kita membuat objek. Lalu kita buat objek bernama obj dengan cara menuliskan Calculation2 obj=new Calculation2();. Setelah itu kita masukkan nilai di dalam objek tersebut dengan:

obj.sum(10.5,10.5); : method overloading bertipe data double karena angka nya desimal

obj.sum(20,20); : method pertama bertipe data integer karena angka nya bulat

obj.sum("Belajar"," Java"); : method overloading baru bertipe data String

****LATIHAN 2 DAN SETERUSNYA DI HALAMAN SELANJUTNYA***

2. Modifikasi praktik 1 dengan menambahkan method TampilkanData!

```
Mahasiswa.java x
public class Mahasiswa{
    private String nim,nama,TampilkanData;

    // method mutator
    public void setNim(String nim) {
        this.nim=nim;
    }
    public void setNama(String nama) {
        this.nama=nama;
    }
    public void setTampilkanData(String TampilkanData) {
        this.TampilkanData=TampilkanData;
    }

    //method acessor
    public String getNim() {
        return nim;
    }
    public String getNama() {
        return nama;
    }
    public String getTampilkanData() {
        return TampilkanData;
    }
}

public static void main(String args[]){
    Mahasiswa obj=new Mahasiswa();
    obj.setTampilkanData("===Menampilkan Data===");
    obj.setNim("175410001");
    obj.setNama("Azkiya");
    System.out.println(obj.getTampilkanData());
    System.out.println("Nim : "+ obj.getNim());
    System.out.println("Nama : "+ obj.getNama());
}
}
```

Output:

```
C:\Windows\system32\cmd.exe
===Menampilkan Data===
Nim : 175410001
Nama : Azkiya
Press any key to continue . . . _
```

penjelasan:

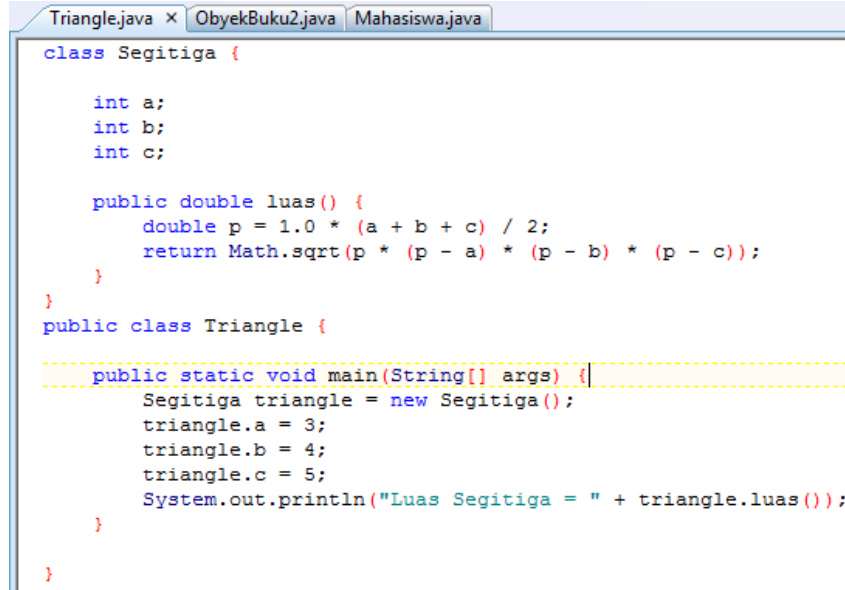
Pertama-tama, kita buat public class bernama Mahasiswa. public class Mahasiswa berfungsi untuk mendeklarasikan class bernama Mahasiswa yang gunanya untuk mengumpulkan berbagai fungsi maupun variabel/atribut. Di dalam program di atas, di dalam class ada beberapa atribut memiliki modifier private yaitu nim, nama, dan TampilkanData yang ditulis dengan **private String nim,nama,TampilkanData;**

Selanjutnya kita membuat method mutator dan method accessor TampilkanData dengan public void setTampilkanData(String TampilkanData). Lalu untuk mengakses komponen dalam class tersebut, gunakan “this” yaitu **this.TampilkanData=TampilkanData;**. Lalu kita buat method accessor nya dengan public **String getTampilkanData()** lalu di return dengan **return TampilkanData;**

Lalu deklarasikan **public static void main(String args[]){** untuk mengawali eksekusi. Kita buat objek bernama **obj** dengan **Mahasiswa obj=new Mahasiswa();**. Lalu di dalam objek tersebut kita gunakan kembali **TampilkanData** yang sudah melewati metode mutator tadi. Kedua atribut itu kita modifikasi nilainya supaya memiliki nilai “===Menampilkan Data===”. Hal itu dilakukan dengan **obj.setTampilkanData("===Menampilkan Data===");**

Yang terakhir kita mencetak output dengan **System.out.println(obj.getTampilkanData());**

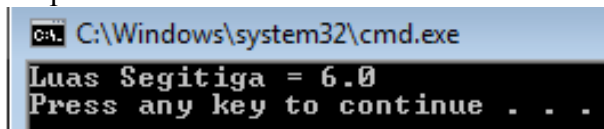
3. Buatlah kelas lengkap dengan atribut dan operasi untuk menentukan luas segitiga !



```
Triangle.java x ObyekBuku2.java Mahasiswa.java
class Segitiga {
    int a;
    int b;
    int c;

    public double luas() {
        double p = 1.0 * (a + b + c) / 2;
        return Math.sqrt(p * (p - a) * (p - b) * (p - c));
    }
}
public class Triangle {
    public static void main(String[] args) {
        Segitiga triangle = new Segitiga();
        triangle.a = 3;
        triangle.b = 4;
        triangle.c = 5;
        System.out.println("Luas Segitiga = " + triangle.luas());
    }
}
```

Output:



```
C:\Windows\system32\cmd.exe
Luas Segitiga = 6.0
Press any key to continue . . .
```

penjelasan:

Pertama-tama, kita buat Class bernama Segitiga. Di dalam program di atas, di dalam class ada beberapa atribut yaitu a, b, dan c. Lalu kita buat method luas() yang isinya adalah perhitungan luas segitiga tersebut. Lalu mewakili apa a, b, dan c tersebut? Di program tersebut, saya membuat a untuk mewakili 3 (alas), b untuk mewakili 4 (tinggi), dan c untuk mewakili 5 (sisi miring). Dari situ bisa kita lihat bahwa program yang saya buat ini adalah untuk menghitung segitiga siku-siku karena setiap sisinya berbeda panjangnya.

Perhitungan di dalam method luas() adalah dengan berikut:

```
double p = 1.0 * (a + b + c) / 2;
return Math.sqrt(p * (p - a) * (p - b) * (p - c));
```

yang intinya adalah menambahkan nilai a, b, dan c lalu dibagi 2 atau bisa juga dikali 0,5. Lalu di return menggunakan Math.sqrt. Math.sqrt digunakan untuk mencari hasil dari akar kuadrat sebuah angka. Fungsi ini membutuhkan 1 argumen, yaitu angka yang akan dihitung.

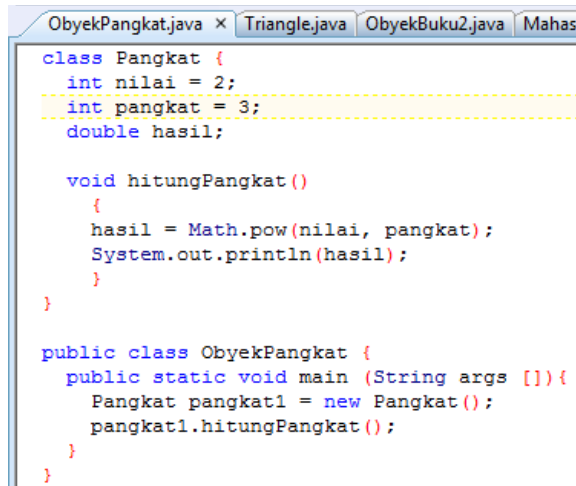
Lalu kita buat objek bernama triangle dengan cara **Segitiga triangle = new Segitiga();**. Lalu kita beri value di setiap atributnya seperti:

```
triangle.a = 3;
triangle.b = 4;
triangle.c = 5;
```

deklarasikan public static void main(String args[]){ untuk mengawali eksekusi. Yang terakhir kita cetak outputnya dan tampilkan hasil perhitungannya dengan **System.out.println("Luas Segitiga = " + triangle.luas());** yang akan menghasilkan angka 6.0 karena 3+4+5 adalah 12, setelah itu dibagi 2 maka hasilnya 6.

TUGAS

1. Buatlah program lengkap dengan atribut dan operasi yang digunakan untuk menghitung nilai pangkat dari suatu bilangan !

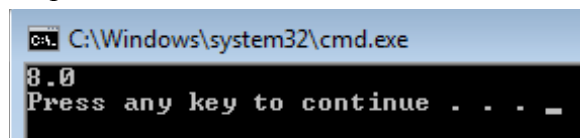


```
class Pangkat {
    int nilai = 2;
    int pangkat = 3;
    double hasil;

    void hitungPangkat()
    {
        hasil = Math.pow(nilai, pangkat);
        System.out.println(hasil);
    }
}

public class ObyekPangkat {
    public static void main (String args []){
        Pangkat pangkat1 = new Pangkat ();
        pangkat1.hitungPangkat ();
    }
}
```

Output:



```
C:\Windows\system32\cmd.exe
8.0
Press any key to continue . . . _
```

penjelasan:

Pertama-tama, kita buat Class bernama Pangkat. class Pangkat berfungsi untuk mendeklarasikan class bernama Pangkat yang gunanya untuk mengumpulkan berbagai fungsi maupun variabel/atribut. Di dalam program di atas, di dalam class ada beberapa atribut yaitu nilai, pangkat, dan hasil yang ditulis dengan **int nilai = 2; int pangkat = 3; dan double hasil;**

Lalu buat method void hitungPangkat() yang berfungsi untuk menghitung bilangan yang dipangkatkan yang sudah kita deklarasikan di dalam tempat atribut tadi. Di dalamnya kita tulis value dari hasil yaitu dengan **Math.pow(nilai, pangkat);** yang akan menghubungkan nilai dan pangkat lalu dikalkulasi. Math.pow berfungsi untuk menghitung bilangan berpangkat. Lalu kita cetak hasilnya dengan **System.out.println(hasil);**

Setelah itu buatlah public class ObyekPangkat yang di dalamnya merupakan proses membuat objek bernama pangkat 1 yang dibuat dengan cara **Pangkat pangkat1 = new Pangkat();**. Yang terakhir adalah memberi value dan menghubungkan pangkat1 dengan void hitungPangkat dengan **pangkat1.hitungPangkat();**. Hasil outputnya adalah 8.0 yang merupakan hasil dari 2 pangkat 3.

****KESIMPULAN ADA DI HALAMAN SELANJUTNYA***

KESIMPULAN LAPORAN

Di dalam laporan ini dibahas 3 jenis method yaitu:

1. Method Tanpa Return Value

Jenis method ini ditandai dengan return type yang berupa void dan pada bagian statement tidak terdapat keyword return. Pada pemrograman berorientasi obyek method jenis ini digunakan untuk membuat method mutator. Nama untuk method mutator sebaiknya diawali dengan kata set, misalnya: setNama(), setAlamat().

2. Method Dengan Return Value

Jenis method ini ditandai dengan return type selain void dan pada bagian statement terdapat keyword return. Pada pemrograman berorientasi obyek method jenis ini digunakan untuk membuat method acessor, dimana method acessor fungsinya adalah untuk membaca/mendapatkan nilai suatu atribut. Nama untuk method acessor sebaiknya diawali dengan kata get, misalnya: getNama(), getAlamat().

3. Method Overloading

Bahasa java mendukung method overloading , java dapat membedakan beberapa method dengan nama yang sama di dalam sebuah kelas namun parameternya berbeda. Hal ini sangat menguntungkan karena memudahkan kita dalam mengingat nama method, bayangkan bila program pada class Gambar harus diberi nama drawInterger(int i), drawString(String s), drawDouble(doubled). Method overloading dibedakan oleh jumlah dan jenis tipe data parameternya.

Terima Kasih.