

BAB III

KONSEP DASAR PEMROGRAMAN JAVA

Program Sederhana

```
/* Program Java Pertama */
public class Welcome
{
    public static void main(String[]args)
    {
        //Menampilkan Nama String
        System.out.println("Welcome To UMBY");
    }
}
```

Analisa Program

Code	Keterangan
/* Program Java Pertama */	Menandakan komentar
public class Welcome	Menandakan nama class <ul style="list-style-type: none">Dalam Java semua code program harus diletakkan di dalam deklarasi class dengan keyword "class".Suatu class dalam Java selalu diawali dengan hak akses, dalam code tersebut menggunakan hak akses "public" yang berarti dapat diakses oleh semua package maupun class yang lainnya.
{	Menandakan awal blok dari class
public static void main(String[]args) {	Merupakan sebuah titik awal program java (method utama)
//Menampilkan Nama String	Menandakan komentar
System.out.println("Welcome To UMBY"); }	Mencetak "Welcome To UMBY" di layar, yang di apit dengan tanda petik double
}	Menandakan akhir blok dari class

Keterangan :

- Bahasa java bersifat sangat Case Sensitive, yaitu penggunaan huruf besar dan huruf kecil sangat dibedakan.
- Space seperti karakter spasi, tab, pindah baris dan karakter lainnya yang berfungsi untuk memformat tampilan, tidak berpengaruh dalam bahasa Java sehingga dapat digunakan untuk memudahkan membaca kode yang tulis .

3.1 Identifier

- Identifier merupakan nama yang digunakan untuk menamai suatu class, interface, variable dan method.
- Syarat :

- Tidak boleh menggunakan keyword yang digunakan oleh java.
- Harus diawali dengan huruf (A, a), garis bawah (_) atau lambang dolar (\$).
- Tidak boleh menggunakan karakter yang digunakan sebagai operator oleh Java, seperti : +, -, *, /.
- Contoh :
 - Saya.Java
 - Saya1.java

3.2 Keyword

- Merupakan kata-kata yang memiliki arti khusus dan dikenali oleh compiler Java.

<code>abstract</code>	<code>continue</code>	<code>for</code>	<code>new</code>	<code>switch</code>
<code>assert</code>	<code>default</code>	<code>goto</code>	<code>package</code>	<code>synchronized</code>
<code>boolean</code>	<code>do</code>	<code>if</code>	<code>private</code>	<code>this</code>
<code>break</code>	<code>double</code>	<code>implements</code>	<code>protected</code>	<code>throw</code>
<code>byte</code>	<code>else</code>	<code>import</code>	<code>public</code>	<code>throws</code>
<code>case</code>	<code>enum</code>	<code>instanceof</code>	<code>return</code>	<code>transient</code>
<code>catch</code>	<code>extends</code>	<code>int</code>	<code>short</code>	<code>try</code>
<code>char</code>	<code>final</code>	<code>interface</code>	<code>static</code>	<code>void</code>
<code>class</code>	<code>finally</code>	<code>long</code>	<code>strictfp</code>	<code>volatile</code>
<code>const</code>	<code>float</code>	<code>native</code>	<code>super</code>	<code>while</code>

3.3 Tipe Data

- Java mempunyai dua buah tipe data, yaitu :
 - Tipe data Primitif
 - Merupakan tipe data dasar yang dikenal oleh Java.
 - Tipe data Primitif dapat digunakan untuk menyimpan nilai tertentu.
 - Contoh :
 - `Int Sisi = 2;`
 - `Int Keliling_Persegi = 6 * Sisi;`

Nama Tipe Data	Ukuran dalam Bit	Nilai	Standar
long	64	-2^{63} s/d $2^{63} - 1$	
int	32	-2^{31} s/d $2^{31} - 1$	
short	16	-2^{15} s/d $2^{15} - 1$	
byte	8	-2^7 s/d $2^7 - 1$	
double	64	Negatif : -1.7976931348623157E+308 s/d -4.94065645841246544e-324 Positif : 4.94065645841246544e-324 s/d 1.7976931348623157E+308	IEEE 745 floating point
float	32	Negatif : -3.4028234663852886E+38 s/d -1.40129846432481707e-45 Positif : 1.40129846432481707e-45 s/d 3.4028234663852886E+38	IEEE 745 Floating point
char	16	'\u0000' s/d '\uFFFF' (0 s/d 65535)	ISO Unicode Character Set
boolean	8	true atau false	

▪ Tipe data Referensi

- Merupakan tipe data yang digunakan untuk memegang referensi dari suatu object (instance dari class).
- Contoh :

```
class ContohReferensi
{
    int x = 0; //pendeklarasian variable x dengan tipe data int
    public static void main(String[] args)
    {
        ContohReferensi y; // pendeklarasian variable y dengan tipe data class
        y = new ContohReferensi(); //instatiate class ContohReferensi menjadi object
        y.x = 20; // Akses object ContohReferensi melalui variabel y
        System.out.println("Nilai x : " + y.x);
    }
}
```

3.3 Variable

- Merupakan lokasi yang digunakan untuk melakukan penyimpanan yang terdapat di memori.
- Deklarasi :
Tipe_Data Nama_Variabel ;
- Contoh :

Int Panjang;

3.5 Class

- Merupakan sekumpulan object yang mempunyai kesamaan keadaan dan perilaku.

- Deklarasi :
Class Nama_Class
{
Body Class
}

3.6 Object (Instance)

- Merupakan suatu entiti yang memiliki keadaan, behavior, dan identitas yang tugasnya dirumuskan dalam suatu lingkup masalah dengan baik.
- Contoh :

```
class persegi panjang //persegi panjang adalah class
{
    int panjang; //panjang dan lebar merupakan attribute
    int lebar;
}

class luas_persegi panjang
{
    public static void main(String[] args)
    {
        double luas;
        persegi panjang p = new persegi panjang(); //p adalah object
        p.panjang=10;
        p.lebar=3;
        luas=p.panjang * p.lebar;
        System.out.println("Luas persegi panjang : "+luas);
    }
}
```

3.7 Attribute

- Attribute merupakan elemen data dari suatu object.
- Contoh :
Attribute dari seorang mahasiswa :
 - NIM
 - Nama

3.8 Method

- Merupakan sub program yang dihubungkan pada definisi class tertentu.
- Contoh :

```
class penjumlahan
{
    double a; // pendeklarasian attribute a dan b
    double b;
```

```

void hasil() //hasil merupakan method
{
    System.out.println("Hasil penjumlahan = "+(a+b));
}
}

class cetak
{
    public static void main(String[]args)
    {
        penjumlahan jum; //pembuatan object baru jum
        jum = new penjumlahan(); //pendeklarasian object
        jum.a = 10; //mengisi data object
        jum.b = 1;
        jum.hasil(); // memanggil method
    }
}

```

3.9 Modifier

- Merupakan kata tambahan yang berada di depan suatu unsur seperti kelas, data, method.
- Digunakan untuk menentukan hubungan suatu unsur kelas dengan kelas lainnya, misalnya suatu hubungan antara kelas dengan object.

Modifier	Keterangan
Public	Menyatakan kelas/method/attribute tersebut dapat diakses oleh kelas yang lain
Private	Menyatakan kelas tersebut tidak dapat diturunkan dan diakses oleh kelas yang lain
Protected	Menyatakan kelas/method/attribute tersebut dapat diakses oleh kelas lain yang berada dalam satu paket dan dapat diakses oleh kelas lain yang merupakan turunannya
Abstract	
Final	Menyatakan suatu kelas tidak dapat diturunkan (extend/inheritance/pewarisan) pada kelas lain.
Extends	Pada saat terjadi suatu inheritance : <ul style="list-style-type: none"> • Kelas yang mewariskan method dan attributenya disebut super class • Kelas yang mewarisi disebut subclass
Implements	Kelas yang mengimplementasikan satu atau lebih interface.

3.10 Constructor

- Merupakan method yang terdapat dalam suatu kelas yang digunakan untuk membuat dan menginisialisasikan sebuah object baru.
- Syarat :
 - Nama method harus sama dengan nama class
 - constructor tidak memverifikasi tipe data yang dikirim
- Contoh :

```

class constructor // constructor merupakan nama class
{
    int panjang; // pendeklarasian attribute
    int lebar;
    constructor() // method sebagai constructor = nama class
    {
        panjang = 5;
        lebar = 4;
    }
    int luas ()
    {
        return(panjang*lebar);
    }
}

class persegipanjang2
{
    public static void main(String[]args)
    {
        constructor c; //pembuatan object baru c
        c = new constructor();
        System.out.println("Luas persegi panjang = "+c.luas());
    }
}

```

3.11 Komentar Pada Java

- Merupakan suatu pernyataan yang tidak berpengaruh terhadap suatu proses.
- Simbol :
 - // Komentar >> Digunakan jika komentar hanya satu baris
 - /* Komentar */ >> Digunakan untuk komentar dengan beberapa baris sekaligus
 - /** Komentar */ >> Khusus digunakan untuk keperluan dokumentasi.

3.12 System.out.println() VS System.out.print()

- System.out.print() = Tidak ganti baris
- System.out.println() = Menyatakan ganti baris
- Contoh :

```

public class Welcome
{
    public static void main(String[]args)
    {
        System.out.println("Welcome");
        System.out.println("To");
        System.out.println("Universitas Mercu Buana Yogyakarta");
    }
}

```

```

public class Welcome
{
    public static void main(String[]args)
    {
        System.out.print(" Welcome ");
        System.out.print(" To ");
        System.out.print(" Universitas Mercu Buana Yogyakarta ");
    }
}

```

```
geany_run_script.sh
Welcome
To
Universitas Mercu Buana Yogyakarta

-----
(program exited with code: 0)
Press return to continue
```

```
geany_run_script.sh
Welcome To Universitas Mercu Buana Yogyakarta

-----
(program exited with code: 0)
Press return to continue
```

Tugas :

1. Buatlah program java untuk menampilkan yang menghasilkan output seperti gambar berikut :

Status	dnd@dnd-laptop:~/Program/Java\$ javac segitiga.java
Compiler	dnd@dnd-laptop:~/Program/Java\$ java segitiga
Messages	Nama : A. SidiQ P. NIM : 0511078701
Scribble	=====
Terminal	Program menghitung luas segitiga Alas = 2 cm Tinggi = 3 cm Rumus = (alas*tinggi)/2 Luas segitiga = 3 cm dnd@dnd-laptop:~/Program/Java\$

2. Analisa program yang telah dibuat tersebut.