



Julian Chandra W

- Nama : Julian Chandra W
- Telp : 085647155605
- Email : maetzro_87@yahoo.co.id

Referensi :

1. Benny Hermawan. 2004. *Menguasai Java 2 & Object Oriented Programming*. Andi.
2. Bambang Hariyanto. 2007. *esensi-esensi bahasa pemrograman JAVA*. Informatika.
3. Sariadin Siallagan. 2009. *Pemrograman Java Dasar-dasar Pengenalan & pemahaman*. Andi

UNIVERSITAS KOMPUTER INDONESIA
JURUSAN MANAJEMEN INFORMATIKA

Julian Chandra W

ATURAN PERKULIAHAN

- ABSENSI < 80% atau Alpha > 3
Dinyatakan **TIDAK LULUS**
- Komponen Penilaian

Quiz	10%	80 - 100	A
Tugas	20%	68 - 79	B
UTS	30%	56 - 57	C
UAS	40%	45 - 55	D
		0 - 44	E

UNIVERSITAS KOMPUTER INDONESIA
JURUSAN MANAJEMEN INFORMATIKA

Julian Chandra W

SILABUS

1. Pengenalan Bahasa Pemrograman Java
2. Struktur Dasar Pemrograman Java
3. Percabangan (If dan Switch)
4. Looping (For, While, Do While)
5. Array
6. Object Oriented Programming
7. Method
8. Pilar OOP (inheritance, polymorphism, encapsulation)

UNIVERSITAS KOMPUTER INDONESIA
JURUSAN MANAJEMEN INFORMATIKA

Apakah Java itu ?

- **Java adalah suatu bahasa pemrograman Object Oriented dengan unsur-unsur seperti bahasa C++ dan bahasa-bahasa lainnya dengan libraries yang cocok untuk lingkungan internet.**
- Java diciptakan oleh James Gosling developer dari Sun Microsystems pada tahun 1991.
- **Tujuan java adalah menjadi perangkat lunak yang kecil, efisien dan portable untuk berbagai perangkat keras.**

Karakteristik Java

1. Sederhana (Simple)

Bahasa pemrograman java menggunakan Sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan *multiple inheritance*. Java juga menggunakan *automatic memory allocation* dan *memory garbage collection*.

2. Berorientasi objek (Object Oriented)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

3. Terdistribusi (Distributed)

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries networking* yang terintegrasi pada java.

Karakteristik Java

4. Interpreted

Program java dijalankan menggunakan interpreter yaitu *Java Virtual Machine* (JVM). Hal ini menyebabkan *source code* Java yang telah dikompilasi menjadi java *bytecodes* dapat dijalankan pada platform yang berbeda-beda.

5. Robust

Java mempunyai reliabilitas yang tinggi. Compiler pada java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception handling* untuk membantu mengatasi error pada pemrograman.

6. Architecture Neutral

Program java merupakan *platform independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform berbeda dengan *Java Virtual Machine*.

Karakteristik Java

7. Portable

Source code maupun program java dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.

8. Multithreaded

Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

9. Dynamic

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan *properties* ataupun *method* dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.

Instalasi

- Instalasi Software
 - JDK untuk pengembangan aplikasi
 - JRE untuk menjalankan aplikasi
- Setting Environment
 - PATH
- Editor
 - Notepad, Notepad ++, UltraEdit dsb.

Program Sederhana

Simpan program dengan nama selamat.java

```
public class selamat {  
    public static void main(String [] args) {  
        System.out.println("Selamat Datang");  
    }  
}
```

Latihan

Buat Program untuk menampilkan Biodata Anda :

Nim :
Nama :
Tempat Tanggal Lahir :
Hobi :
Alamat :
No Telepon :
E-mail :

Simpan dengan nama anda

Struktur Program Java

Struktur Program Java

- *Comments*
- *Modifiers*
- *Statements*
- *Blocks*
- *Classes*
- *Methods*
- *Reserved words*

Comments

1. Comments

- ✓ *Comments* digunakan untuk memberikan keterangan/penjelasan suatu baris program.
- ✓ *Comments* tidak akan dikompilasi oleh *compiler*.

Lambang untuk comments :

- ✓ double slash (//) untuk memberikan penjelasan satu baris
- ✓ /* dan */ untuk memberikan penjelasan satu baris atau lebih.

Contoh Comments :

```
public class tes {  
    public static void main(Strings [] args) {  
        /*  
        System.out.println akan  
        mencetak 'akan mencetak'  
        */  
        //System.out.println("tidak akan mencetak");  
        System.out.println("akan mencetak");  
    }  
}
```

Modifiers, Statements, Blocks, Classes

2. Modifiers

Modifiers digunakan untuk menentukan penggunaan dari suatu data, methods dan *class*. Contoh *modifiers* adalah public, static, private, final, abstract dan protected.

3. Statements

Statements merupakan baris perintah atau kumpulan perintah. Setiap *statements* pada java selalu diakhiri dengan titik koma (;).

4. Blocks

Blocks digunakan untuk membentuk suatu grup *statements*. *Blocks* diawali dengan kurung kurawal buka ({) dan kurung kurawal tutup (}).

5. Classes

Classes merupakan inti dari program Java. Suatu *class* merupakan "blueprint" untuk menciptakan suatu object.

Identifiers

Identifiers

Identifiers merupakan penamaan dalam pemrograman untuk variabel, konstanta, *method*, *class*, dan *package*. Java merupakan bahasa pemrograman yang *case-sensitive* (membedakan antara huruf besar dan kecil). Aturan penamaan *identifiers* pada java adalah :

1. Dapat dimulai dengan huruf, *underscore* (_) atau tanda \$
2. Tidak dapat menggunakan simbol operator seperti : +, -, *, / dan lain- lain
3. Tidak dapat menggunakan *reserved word*
4. Dapat mempunyai panjang berapa saja.

Variables

Variables

Variables digunakan untuk input,output atau menyimpan hasil suatu proses pada program. Untuk melakukan deklarasi suatu variables digunakan :

Datatype variableName;

Contoh : int angka;

untuk mendeklarasi sekaligus melakukan inisialisasi digunakan :

Datatype variableName = value;

Contoh : int angka = 5;

Aturan Konvensi :

1. Bila terdiri satu abjad atau satu kata, semua abjad dituliskan sebagai huruf kecil.
Contoh: a , nama.
2. Bila nama terdiri dari dua kata atau lebih, huruf pertama dari kata pertama sebaiknya kecil, dan huruf pertama kata kedua dan seterusnya, sebaiknya huruf besar.

Constants (konstanta)

Constants (konstanta)

Constants merupakan data yang tidak berubah selama program berjalan.

Pendeklarasian konstanta menggunakan :

static final Datatype CONSTANTNAME = Value;

Contoh : static final double PI = 3.14159;

area = radius*radius*PI;

Aturan konvensi:

1. Semua Abjad dari nama konstanta dituliskan dalam huruf kapital.
2. Bila Konstanta lebih dari satu kata, maka semuanya disambung menjadi satu dan semua abjadnya juga dituliskan dalam bentuk kapital.

TIPE DATA -- Numerik

Tipe data primitif pada java yaitu :

1. Numerik (bulat dan pecahan)

Numerik bulat ada 4 macam yaitu :

Tipe Data	Besar(bits)	Jangkauan
• byte	8	-128 ke 127
• short	16	-32,768 ke 32,767
• int	32	-2,147,483,648 ke 2,147,483,647
• long	64	-9,223,372,036,854,775,808 ke – 9,223,372,036,854,775,807

Numerik pecahan ada 2 macam yaitu :

- float 32 bits -3.4E38 ke 3.4E38
- double 64 bits -1.7E308 ke 1.7E308

TIPE DATA -- Character, Boolean, Array

2. Character

merupakan tipe data yang meyimpan satu buah karakter.

tipe	Besar (bits)	jangkauan
char	16	Unicode Character

3. Boolean

boolean merupakan tipe data yang hanya berisi *true* atau *false*

4. Tipe data array

digunakan untuk menampung beberapa tipe data yang sama dalam sebuah nama *variable* yang mempunyai indeks. Array pada java merupakan object yang dapat berisi tipe data primitif maupun class.

Ekspresi dan Operator, Operator Aritmatika

Ekspresi merupakan suatu statement yang mengembalikan suatu nilai
 Operator merupakan Simbol yang digunakan dalam ekspresi

Operator Aritmatika pada Java :

Operator	Arti	Contoh
+	Penambahan	3 + 4
-	Pengurangan	5 - 7
*	Perkalian	5 * 5
/	Pembagian	14 / 7
%	Modulus (Sisa bagi)	20 % 7

Operator kondisi (Conditional)

Operator	Arti	Contoh
==	sama dengan	x == 3
!=	tidak sama dengan	x != 3
<	lebih kecil	x < 3
>	lebih besar	x > 3
<=	lebih kecil sama dengan	x <= 3
>=	lebih besar sama dengan	x >= 3

Catatan :
 operator ini sering digunakan pada kondisi if dan kondisi looping.

Operator logika (Logical)

Operator	Arti	Contoh
&&	and	x >= 2 && x <= 4
	or	x > 2 x < 5
!	not	x != 5
^	xor	x == 2 ^ x == 3

Operator &&

Kondisi1	Kondisi2	Hasil
False	False	False
False	True	False
True	False	False
True	True	True

TERIMA KASIH