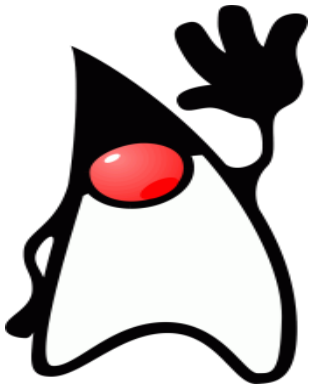
LAB PEMROGRAMAN I (JAVA FUNDAMENTAL)

PERTEMUAN 3

Dosen :

Bella Hardiyana S. Kom, M. Kom



BAB III

DASAR-DASAR PEMROGRAMAN

Tipe Data

Data adalah sekumpulan kejadian/fakta yang dipresentasikan dengan huruf, angka, gambar, simbol, dsb. Sedangkan, **tipe data** adalah media penyimpanan data berdasarkan jenis data. Sehingga satu tipe data menyimpan satu jenis data saja.

Contohnya adalah tipe data huruf hanya bisa digunakan untuk data berjenis huruf saja. Tipe data angka hanya bisa digunakan untuk data berjenis angka saja, dst.

Tipe Data Dasar/Primitif

Tipe data dasar adalah tipe data yang bukan dibentuk dari tipe data lain. Tipe data dasar merupakan tipe data yang sudah ada sejak bahasa pemrograman terdahulu (itulah mengapa disebut primitif).

Dalam bahasa pemrograman Java terdapat 8 (delapan) tipe data primitif, meliputi :

- boolean (tipe data logika)
- char (tipe data karakter)
- byte, short, int dan long (tipe data bilangan bulat)
- double dan float (tipe data bilangan pecahan/ floating point)

Tipe data boolean (tipe data logika)

Tipe data boolean adalah tipe data kebenaran, dimana tipe data ini hanya akan menghasilkan nilai benar dan salah.

```
boolean hasil; //awalnya hasil bernilai false
hasil = true;  //hasil sekarang bernilai true
hasil = (5<4); //hasil sekarang bernilai false
if(5+3 != 2*4) //pernyataan ini akan menghasilkan nilai false
  System.out.println("aksi true");
else
  System.out.println("aksi false"); //tulisan ini akan tampil
```

Karakter

Karakter adalah semua anggota/tombol yang ada pada keyboard, meliputi :

- Huruf kecil dan besar/kapital: ‘a’-‘z’ dan ‘A’-‘Z’
- Angka: ‘0’-‘9’
- Tombol fungsi: F1-F12
- Simbol-simbol khusus: ‘$’, ‘%’, ‘/’, ‘*’, ‘&’, ‘+’, dst.
- Tombol perintah: Space, Enter, Esc, Home, End, dst.

Masing-masing digit yang ada pada keyboard memiliki kode Ascii yang berbeda. Itulah sebabnya huruf kapital dan kecil berbeda, contoh: ‘A’ ≠ ‘a’, ‘B’ ≠ ‘b’, dst.

Tipe data char

Tipe data char adalah tipe data yang dapat menampung hanya satu digit tombol saja. Ciri khas dari tipe data char adalah selalu diapit dengan tanda petik satu/single quotes (‘).

```
char index;
index = 'a';  //index bernilai 'a'
index = '\n'; //index bernilai '\n' (pindah baris)
boolean cek = ('a'=='A'); //hasil perbandingan bernilai false
```

Tipe data bilangan bulat

Dalam ilmu matematika, **bilangan bulat** adalah bilangan yang bukan pecahan. Artinya bilangan tersebut tidak memiliki koma (,). Tapi dalam bahasa pemrograman koma (,) diganti dengan titik (.). Jadi, definisi **bilangan bulat** yang sudah diperbaharui adalah bilangan yang tidak memiliki titik (.).

Tipe Data	Panjang	Jangkauan
byte	8-bits	-2^7 s/d 2^7-1
short	16-bits	-2^{15} s/d $2^{15}-1$
int	32-bits	-2^{31} s/d $2^{31}-1$
long	64-bits	-2^{63} s/d $2^{63}-1$

Tipe data bilangan pecahan

Dalam definisi sederhana, **bilangan pecahan** dalam bahasa pemrograman adalah bilangan bertitik (dalam matematika bilangan pecahan adalah bilangan berkoma). Istilah dalam dunia komputer, bilangan pecahan disebut floating point.

Tipe Data	Panjang	Jangkauan
float	32-bits	-2^{31} s/d $2^{31}-1$
double	64-bits	-2^{63} s/d $2^{63}-1$

String

Definisi String adalah kumpulan dari karakter. Dalam bahasa pemrograman lain String adalah array dari karakter.

Tidak seperti char, String bisa menampung lebih banyak digit dibandingkan dengan char yang hanya bisa menampung satu digit saja. Jumlah digit yang bisa ditampung oleh String yaitu sebanyak >= 1 (lebih besar sama dengan satu) digit.

Cara membedakan char dengan String cukup mudah. Ciri dari **char** adalah selalu **diapit** oleh tanda **single quotes/petik satu(').** Sedangkan **String** selalu **diapit** oleh tanda **double quotes/petik dua(").**

Penggunaan String

```
String nim="10506357"; //cara pertama
String nama=new String("Hardiyana"); //cara kedua
System.out.println(nim);  //output: 10506357
System.out.println(nama); //output: Hardiyana
```

Catatan: Baik cara pertama maupun cara kedua adalah sama. Cara yang paling umum dan paling mudah adalah cara pertama.
Mengingat String adalah sebuah kelas, maka cara kedua adalah cara yang sebenarnya direkomendasikan oleh Java, alasannya adalah cara kedua menggunakan konsep instance objek.

Java Identifier

Java identifier adalah nama yang mewakili class, interface, atribut, method, argument, dsb. Pendeklarasian indentifier di Java bersifat case-sensitive.

Penamaan (identifier) akan dibedakan menjadi 2 macam, yakni : penamaan bersifat umum dan khusus

Aturan penamaan yang bersifat umum

1. Tidak boleh sama dengan daftar keyword.
2. Tidak boleh menggunakan space dan simbol-simbol khusus.
3. Harus selalu diawali dengan huruf.
4. Tidak boleh menggunakan indentifier yang sudah pernah dideklarasikan.
5. Boleh menggunakan karakter underscore (_), tapi ini tidak direkomendasikan.
6. Boleh menggunakan karakter angka, tapi tidak boleh digunakan diawal.
7. Tidak direkomendasikan menggunakan identifier dengan huruf kapital semua.
8. Jika identifier memiliki dua atau lebih kata, maka huruf awal pada kata tersebut adalah huruf besar.

Contoh penggunaan Identifier

nim (BENAR)

namaMahasiswa (BENAR)

jenis_Obat (BENAR, TAPI TIDAK DIREKOMENDASIKAN)

1Kelas (SALAH)

_1 (BENAR JIKA INDENTIFIER BUKAN CLASS, TAPI TIDAK DIREKOMENDASIKAN)

_ (BENAR JIKA IDENTIFIER BUKAN CLASS, TAPI TIDAK DIREKOMENDASIKAN)

LUAS (BENAR, TAPI TIDAK DIREKOMENDASIKAN)

Tinggi$ (SALAH)

Aturan penamaan yang bersifat khusus

1. *Identifier* class

Aturan khusus untuk *identifier* class.

a) Semua aturan umum, ditambah aturan poin **b dan c**.
b) Huruf awal harus diawali dengan huruf kapital/besar.
c) Nama class yang dideklarasikan sebagai public harus sama dengan nama file.

Contoh: NamaKelas, PraktikumPertama, dsb.

Aturan penamaan yang bersifat khusus

***2. Identifier* atribut, *variable* dan *argument*/parameter**

Aturan khusus untuk *identifier* atribut, *variable* dan parameter.

a) Semua aturan umum, ditambah aturan poin **b**.
b) Huruf awal harus diawali dengan huruf kecil.

Contoh: x, tinggiPersegi, phi, dsb.

Aturan penamaan yang bersifat khusus

***3. Identifier* method**

Aturan khusus untuk *identifier* method.

a) Semua aturan umum, ditambah aturan poin **b dan c**.
b) Huruf awal harus diawali dengan huruf kecil.
c) Nama method selalu diakhiri dengan tanda kurung buka dan tutup (‘namaMethod()’).

Contoh: methodUtama(), cetak(), inputData(), dsb.

Daftar Keyword dalam bahasa Java

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

Catatan: true, false dan null bukan termasuk *keyword,* tetapi kita tidak dapat menggunakan *keyword* tersebut.

Komentar Dalam Bahasa Java

Komentar adalah statement berupa catatan/pesan yang tidak akan diproses oleh komputer.

1) Penulisan komentar bergaya C++ (// …)
Komentar dalam gaya C++ hanya berlaku untuk satu baris saja.

2) Penulisan komentar bergaya C (/* … */)
Komentar dalam gaya C bisa berlaku untuk lebih dari satu baris. Diawali dengan /* dan diakhiri dengan */.

3) Penulisan komentar bergaya javadoc (/ … */)**
Sama hanya dengan komentar gaya C, komentar dalam gaya javadoc pula bisa berlaku untuk lebih dari satu baris. Diawali dengan /** dan diakhiri dengan */.

Statement dan Blok Dalam Bahasa Java

Statement adalah suatu aksi yang dijalankan oleh komputer. Semua bahasa pemrograman yang mengadopsi semantiks dari bahasa C pasti menandai statement/aksi dengan tanda **semicolon (;).** Sedangkan **Blok** adalah satu/lebih statement yang ditulis dalam **kurung kurawal (‘{}’).**

```
public static void main(String[] args){
 System.out.print("Halo-Halo ");
 System.out.print("Bandung");
}
```

Struktur Dasar Bahasa Java

```
package alamatPaket;
//blok package ditulis (0-1)

import alamatPaketYgDiImport;
//blok import ditulis (0-N)

<modifier> class <NamaClass>{
 <modifier> <tipeData> <namaAtribut> [=inisialisasi];
 //blok deklarasi atribut ditulis (0-N)

 <modifier> <NamaClass>(<argumen>){
 }
 //blok deklarasi constructor ditulis (0-N)

 <modifier> <tipeData> <namaMethod>(<argumen>){
 }
 //blok deklarasi method ditulis (0-N)

 public static void main(String[] args){
 }
 //blok deklarasi method main ditulis (0-1)
}
```

A

B

C

D

E

F

G

Variabel

Variabel adalah wadah/tempat untuk menyimpan data. Pendeklarasian variabel terikat dengan aturan penamaan/identifier. Dalam suatu variabel kita wajib menentukan tipe data, karena dalam bahasa komputer variabel yang hanya bisa menyimpan satu jenis data saja.

<tipeData> <namaVar> [=inisialisasi];

Catatan: untuk <tipeData> dan <namaVar> bersifat wajib, sedangkan [=inisialisasi] bersifat opsional (boleh ditulis, boleh tidak).

Operator Aritmetika

Operator	Penggunaan	Keterangan
+	bil1 + bil2	bil1 ditambah bil2
-	bil - bil2	bil1 dikurangi bil2
*	bil1 * bil2	bil1 dikali bil2
/	bil1 / bil2	bil1 dibagi bil2
%	bil1 % bil2	mencari sisa hasil bagi dari bil1 dan bil2

Operator Increment & Decrement dan Operator Khusus

Operator	Penggunaan	Keterangan
++	bil1++	Setara dengan bil1=bil1+1 atau bil1+=1. Penambahan dilakukan setelah proses (post)
	++bil1	Setara dengan bil1=bil1+1 atau bil1+=1. Penambahan dilakukan sebelum proses (pre)
--	bil1--	Setara dengan bil1=bil1-1 atau bil1-=1. Pengurangan dilakukan setelah proses (post)
	--bil1	Setara dengan bil1=bil1-1 atau bil1-=1. Pengurangan dilakukan sebelum proses (pre)
+=	bil1+=bil2	Setara dengan bil1=bil1+bil2
-=	bil1-=bil2	Setara dengan bil1=bil1-bil2
*=	bil1*=bil2	Setara dengan bil1=bil1*bil2
/=	bil1/=bil2	Setara dengan bil1=bil1/bil2
%=	bil%=bil2	Setara dengan bil1=bil1%bil2

Operator Perbandingan

Operator perbandingan adalah operator yang berfungsi untuk membandingkan dua buah blok (kiri dan kanan) dan menghasilkan nilai boolean (true/false).

Operator	Penggunaan	Keterangan
>	bil1>bil2	Apakah bil1 lebih besar dari bil2
>=	bil1>=bil2	Apakah bil1 lebih besar sama dengan bil2
<	bil1<bil2	Apakah bil1 lebih kecil dari bil2
<=	bil1<=bil2	Apakah bil1 lebih kecil sama dengan bil2
==	bil1==bil2	Apakah bil1 sama dengan bil2
!=	bil2!=bil2	Apakah bil1 tidak sama dengan bil2

Operator Logika

A	B	AND(&&)	OR(||)	XOR(^)	NOT(!) A
true	true	true	true	false	false
true	false	false	true	true	
false	true	false	true	true	true
false	false	false	false	false	

AND: jika ada false maka hasil false.
OR: jika ada true maka hasil true.
XOR: jika kondisi bernilai sama, maka false.
NOT: kebalikan dari nilai awal.

Operator Kondisi (?:)

Operator Kondisi (?:) adalah operator yang menghasilkan nilai boolean, konsepnya mirip dengan seleksi if. Berikut ini adalah format penulisan.

```
(kondisi)? aksiTrue: aksiFalse;
```

```
(kondisi1)?AksiTrue1: (kondisi2)?AksiTrue2:AksiFalse2, dst;
```

Keterangan:

- **Kondisi** adalah pernyataan perbandingan yang akan menghasilkan nilai boolean.
- **aksiTrue** adalah pernyataan yang dijalankan jika hasil perbandingan (kondisi) bernilai true.
- **aksiFalse** adalah pernyataan yang dijalankan jika hasil perbandingan (kondisi) bernilai false.