

Pemrograman Dasar

IS - 104

Tipe Data, Casting & Operators

Review

- Sebutkan 3 tingkatan bahasa pemograman dan beri contohnya!
- Sebutkan dan jelaskan 3 struktur pemrograman!
- Sebutkan dan jelaskan isi dari programming tools!

Today's Menu

- Tipe Data
- Casting
- Operator

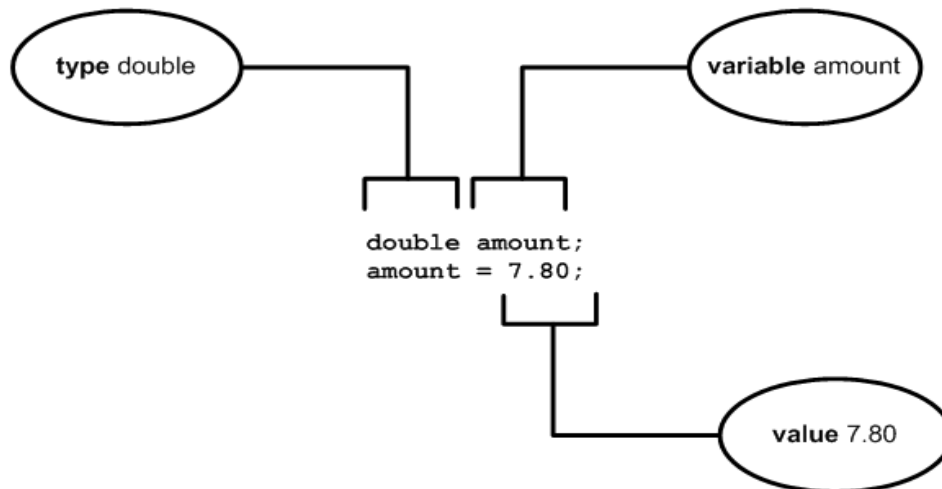
Variabel

- Berfungsi menyimpan data tertentu berdasarkan tipe data yang ditentukan.
- Data yang disimpan, disebut juga dengan value.
- Data yang disimpan, bisa juga diubah nilainya sesuai dengan kebutuhan
- Sebuah variabel pasti memiliki tipe, nama, dan value
- Variable bisa diberi nilai atau dirubah nilainya dengan menggunakan assignment statement (“=”)
- Sebelum dapat dipakai, harus dideklarasikan terlebih dahulu.

Deklarasi Variabel

<data tipe> <name> [=initial value];

- Nilainya berada diantara <> adalah nilai yang disyaratkan, sementara nilai dalam tanda [] bersifat optional.



A variable, its value and its type

Type Data Primitif

Data Type Group	Data Type in Java
Integers	byte, short, int, and long
Floating-point numbers	Float and double
Characters	Char
Boolean	boolean

Primitive Data Types Sizes

Type	Size (bits)
byte	8
int	32
float	32
char	16
short	16
long	64
double	64

Logika - Boolean

- tipe data boolean diwakili oleh dua pernyataan : true dan false. Sebagai contoh adalah,

```
boolean result = true;
```

- Contoh yang ditunjukkan diatas, mendeklarasikan variabel yang dinamai **result** sebagai tipe data boolean dan memberinya nilai **true**.

Tekstual – char

- Tipe data character (char), diwakili oleh karakter single Unicode. Tipe data ini harus memiliki ciri berada dalam tanda single quotes(' '). Sebagai contoh,

```
'a'      //Huruf a  
'\t'    // tab
```

- Untuk menampilkan karakter khusus seperti ' (single quotes) atau " (double quotes), menggunakan karakter escape \. Sebagai contoh,

```
\"      //untuk single quotes  
\"\"    //untuk double quotes
```

- Meskipun, String bukan merupakan tipe data primitif (namun merupakan suatu Class),kita akan memperkenalkan mengenai pada bagian ini.String mewakili tipe data yang terdiri atas beberapa karakter. mereka tidak termasuk tipe data primitif, melainkan suatu class. Mereka memiliki literal yang terdapat diantara tanda double quotes(“”).

Sebagai contoh,

```
String message=“Hello world!”
```

Integral – byte, short, int & long

- tipe data yang terintegrasi dalam java menggunakan tiga bentuk- yaitu desimal, oktal atau heksadesimal.

Sebagai contoh,

- `2` //nilai desimal 2
- `077` //angka 0 pada awal pernyataan mengindikasikan nilai oktal
- `0xBACC` //karakter 0x mengindikasikan nilai heksadesimal

Variable Type

Type Name	Kind of Value	Memory Used	Size Range
byte	integer	1 byte	-128 to 127
short	integer	2 bytes	-32768 to 32767
int	integer	4 bytes	-2147483648 to 2147483647
long	integer	8 bytes	-9223372036854775808 to 9223372036854775807
float	floating-point number	4 bytes	+ 3.40282347 x 10 +38 to + 1.40239846 x 10 -45
double	floating-point number	8 bytes	+ 1.767693134866231570 x 10 +308 to + 4.94065645841246544 x 10 -324
char	single character (unicode)	2 bytes	all unicode characters
boolean	true or false	1 bit	not applicable

Contoh 1:

```
1 public class Base
2 {
3     public static void main (String[] args )
4     {
5         boolean flag = true;
6         char ch = 'A';
7         byte b = 12;
8         short s = 24;
9         int i = 257;
10        long l = 890;
11        float f = 3.14f;
12        double d = 2.2222;
13        System.out.println("flag = " + flag);
14        System.out.println("ch = " + ch);
15        System.out.println("b= " + b);
16        System.out.println("s= " + s);
17        System.out.println("i= " + i);
18        System.out.println("l= " + l);
19        System.out.println("f= " + f);
20        System.out.println("d= " + d);
21    }
22 }
23
```

Konstanta

- Kita dapat membuat variabel menjadi konstanta yang tidak dapat diubah nilainya dengan menambahkan keyword “final” sebelum tipe data dari variabel.

```
final int konstantainteger = 10;
```

```
final float pajak = 15.5;
```

Operator

- Dalam Java, ada beberapa tipe operator. Ada operator aritmatika, operator relasi, operator logika, dan operator kondisi.
- Operator ini mengikuti macam-macam prioritas yang pasti jadi compilernya akan tahu yang mana operator untuk dijalankan lebih dulu dalam kasus beberapa operator yang dipakai bersama-sama dalam satu pernyataan

Operator Arithmatika

Operator	Use	Description
+	$op1 + op2$	Adds $op1$ and $op2$
*	$op1 * op2$	Multiplies $op1$ by $op2$
/	$op1 / op2$	Divides $op1$ by $op2$
%	$op1 \% op2$	Computes the remainder of dividing $op1$ by $op2$
-	$op1 - op2$	Subtracts $op2$ from $op1$

Operator Relasi

- Operator Relasi **membandingkan** dua nilai dan menentukan keterhubungan diantara nilai-nilai tersebut. Hasil keluarannya berupa nilai boolean yaitu **true** atau **false**

Operator	Use	Description
>	op1 > op2	op1 is greater than op2
>=	op1 >= op2	op1 is greater than or equal to op2
<	op1 < op2	op1 is less than op2
<=	op1 <= op2	op1 is less than or equal to op2
==	op1 == op2	op1 and op2 are equal
!=	op1 != op2	op1 and op2 are not equal

Operator Logika

- Operator logika memiliki satu atau lebih operand boolean yang menghasilkan nilai boolean.
- Ada enam operator logika yaitu:
 - **&& (logika AND)**
 - & (boolean logika AND)
 - **|| (logika OR)**
 - | (boolean logika inclusive OR)
 - ^ (boolean logika exclusive OR)
 - **! (logika NOT)**

Operator Logika

- Pernyataan dasar untuk operasi logika adalah

$x1 \text{ op } x2$

- Dimana $x1$, $x2$ dapat menjadi pernyataan boolean. Variabel atau konstanta, dan op adalah salah satu dari operator **&&**, **&**, **||**, **|** atau **^**.
- Tabel kebenaran yang akan ditunjukkan selanjutnya, merupakan kesimpulan dari hasil dari setiap operasi untuk semua kombinasi yang mungkin dari $x1$ dan $x2$.

AND (&&)

x1	x2	Result
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

OR (||)

x1	x2	Result
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

XOR (^)

x1	x2	Result
True	True	False
True	False	True
False	True	True
False	False	False

NOT (!)

x1	Result
TRUE	FALSE
FALSE	TRUE

Specialized Assignment Operators

- Kombinasi antara operator = dan operator aritmatik lainnya seperti -, *, /, dan %.

amount += 5; sama dengan **amount = amount + 5;**

Operator Increment dan Decrement

`count = count + 1;` //increment nilai count dengan nilai 1

- Pernyataan di atas sama dengan `count++;`

Initial value of x	Expression	Final Value of y	Final value of x
5	<code>y = x++</code>	5	6
5	<code>y = ++x</code>	6	6
5	<code>y = x--</code>	5	4
5	<code>y = --x</code>	4	4

Contoh 2:

```
public class Main {  
  
    public static void main(String args[]) {  
        int a = 1;  
        int b = 2;  
        int c = ++b;  
        int d = a++;  
  
        System.out.println("a = " + a);  
        System.out.println("b = " + b);  
        System.out.println("c = " + c);  
        System.out.println("d = " + d);  
  
    }  
}
```

Operator Precedence

- Operator precedence didefinisikan sebagai perintah yang dilakukan compiler ketika melakukan evaluasi terhadap operator, untuk mengajukan perintah dengan hasil yang tidak ambigu/ hasil yang jelas.

.	[]	()	
++	--	!	~
*	/	%	
+	-		
<<	>>	>>>	<<<
<	>	<=	>=
==	!=		
&			
^			
&&			
?:			
=			

Conversion

1. Assignment Compatibilities

Jika kita ingin merubah nilai suatu variable dengan nilai yang dimiliki oleh suatu variabel lain yang memiliki tipe data berbeda, maka tipe data yang baru harus memiliki ukuran yang lebih besar (widening conversions).



byte → short → int → long → float → double

Assignment Compatibilities..cont

```
doble d;  
short s;  
d = 1.2345;  
s = d;
```

Pertukaran yang illegal
Error: possible loss of
precision

```
int i;  
double d;  
i = 10;  
d = i;
```

Pertukaran yang legal
Nilai d= 10.0

- Jika hal ini tidak memungkinkan, maka kita bisa melakukan proses casting.

Casting

Casting digunakan untuk melakukan konversi paksa dari tipe variabel yang satu ke tipe variabel yang lain. Dengan menggunakan casting, java akan merubah nilai asal menjadi nilai target yang dianggap sama/hampir sama. Dan hal ini memungkinkan terjadinya perubahan nilai (information loss)

General Syntax : **(Type_Name) Expression**

Contoh 3:

```
public class MainClass {  
  
    public static void main(String[] args) {  
        long a = 10;  
        int b = (int) a; // narrowing conversion  
  
        System.out.println(a);  
        System.out.println(b);  
    }  
  
}
```

Next Meeting :

- Struktur Kontrol Keputusan

THAT'S ALL FOR TODAY!