

PRAKTIKUM 1.1

Tipe Data

Terdapat beberapa tipe data primitive yang ada di Java yaitu :

Tipe Data	Keterangan
boolean	true atau false
char	Karakter
byte	-128 - 127
short	-32768 - 32767
int	-2147483648 - 2147483647
long	-9223372036854775808 - 9223372036854775807
double	4.9E-324 - 1.7976931348623157E308
float	1.4E-45 - 3.4028235E38

String bukanlah merupakan tipe data di Java, String merupakan Object. Namun string memiliki keunikan yaitu String dapat langsung dibuat tanpa harus membuat Object.

Variabel

Variabel merupakan sesuatu yang digunakan untuk menampung sebuah data. Sebuah variabel harus ada dalam sebuah kelas atau metode. Pembuatan sebuah variabel di Java terlihat pada kode dibawah ini.

```
Tipe namaVariabel;      // mendeklarasikan variabel  
Tipe namaVariabel= nilai; //inisialisasi variabel
```

Contoh 1.1 Pendeklarasian variabel:

```
Int nilai;  
Int nilai=0;
```

Setelah membahas tentang variabel, kita lanjut pengenalan operator di java. Operator di Java tidak beda dengan di C++. Berikut beberapa operator dalam pemrograman Java.

Operator

Operator merupakan sebuah karakter khusus yang digunakan untuk menghasilkan suatu nilai. Berikut beberapa macam operator yang ada dalam java.

Operator Logika

Operator	Keterangan
+	Penjumlahan
-	Pembagian
*	Perkalian
/	Pembagian
%	Sisa

Operator Penugasan

Operator	Keterangan
=	Pemberian nilai
+=	Penambahan bilangan
-=	Pengurangan bilangan
*=	Perkalian bilangan
/=	Pembagian bilangan
%=	Pemerolehan sisa bagi

Operator Pembanding

Operator	Keterangan
==	sama
!=	Tidak sama
>=	Lebih dari sama dengan
<=	Kurang dari sama dengan
>	Lebih dari
<	Kurang dari

Operator Logika

Operator	Keterangan
&&	Dan
	Atau

STRUKUR PERCABANGAN PADA JAVA

Percabangan IF...ELSE

Sintaks if	Contohnya:
<pre>if (boolean expression) { statement or block } else if (boolean expression) { statement or block } else { statement or block }</pre>	<pre>public Class Percabangan { public static void main(String[] args) { int a = 2; if (a < 5) { System.out.println("Nilai a lebih kecil dari 5"); } else { System.out.println("Nilai a lebih besar dari 5"); } } }</pre>

Percabangan SWITCH

Sintaks switch	Contohnya:
<p>Syntax switch sebagai berikut :</p> <pre>switch (expression) { case constant1 : statements; break; case constant2 : statements; break; default : statements; break; }</pre>	<pre>switch(x){ case 1: System.out.println("Senin"); break; case 2: System.out.println("Selasa"); break; case 3: System.out.println("Rabu"); break; case 4: System.out.println("Kamis"); break; case 5: System.out.println("Jum'at"); break; case 6: System.out.println("Sabtu"); break; case 7: System.out.println("Minggu"); break; default: System.out.println("Salah input hari"); break; }</pre>

Percabangan dengan Ekspresi bersyarat

Digunakan untuk mengantikan sebuah bentuk if else. Sintaks adalah sebagai berikut :

Sintaks ekspresi bersyarat	Contoh :
<pre>exp1 ? exp2 : exp3</pre> <p>Dalam hal ini, jika exp1 adalah benar, exp2 dieksekusi.Jika exp2 adalah salah, exp3 dieksekusi.</p>	<pre>int a = 50; int b = 100; int nilai = 3 > 2 ? a : b; Outputnya : nilai = 50</pre>

Perulangan for

Syntax for	Contoh
for (init_expr;boolean testexpr;alter_expr) { statement or block }	public Class PerulanganFOR { public static void main(String[] args) { for (int i = 1; i <=5; i++) { System.out.println(i+"=Belajar Java itu mudah"); } } }

Perulangan while

Sintaks while	Contoh
while (boolean testexpr) { statement or block }	public Class PerulanganWHILE { public static void main(String[] args) { int i = 0; while (i<10) { System.out.println(i); i++; } } }

Perulangan do....while

Sintaks do/while	Contoh
do { statement or block } while (boolean testexpr)	public Class PerulanganDOWHILE { public static void main(String[] args) { int i = 0; do { System.out.println(i); i++; } while (i<10); } }

Membuat Array

Array dibuat dengan menggunakan keyword new. Contoh di bawah ini membuat sebuah Array dengan tipe primitif char:

```
s = new char[26];
```

Indeks Array dimulai dengan angka 0. Batasan legal untuk nilai indeks ini adalah dari nol hingga jumlah elemen Array – 1. Apabila program berusaha mengakses Array di luar batas yang legal, maka akan muncul runtime exception.

Untuk membuat Array dengan elemen objek, gunakan cara penulisan berikut

```
p = new point[10];
```

Pernyataan tersebut tidak membuat 10 objek Point. Untuk membuat objek Point, gunakan pernyataan berikut :

```
p[0] = new point(0, 1);
p[1] = new point(1, 2);
```

Menginisialisasi Array

Java memiliki cara singkat untuk membuat sebuah objek Array dengan elemen-elemennya memiliki nilai awal :

```
String names[] = {"Joko", "Joni", "Jujuk"};
Kode di atas adalah sama dengan kode berikut ini:
String names[];
names[0] = "Joko";
names[1] = "Joni";
names[2] = "Jujuk";
```

