

Tipe Data Dasar

Variabel hanyalah lokasi memori yang dicadangkan untuk menyimpan nilai. Ini berarti bahwa ketika programmer membuat variabel, maka secara eksplisit programmer menyediakan beberapa ruang di memori.

Berdasarkan tipe data variabel, sistem operasi mengalokasikan memori dan memutuskan data apa yang dapat disimpan dalam memori yang dicadangkan tersebut. Oleh karena itu, dengan menetapkan tipe data yang berbeda ke variabel, programmer dapat menyimpan bilangan bulat, desimal, atau karakter dalam variabel ini.

Ada dua tipe data yang tersedia di Java -

- Tipe Data Primitif
- Jenis Data Referensi / Objek

Tipe Data Primitif

Ada delapan tipe data primitif yang didukung oleh Java. Tipe data primitif ditentukan sebelumnya oleh bahasa java dan dinamai dengan menggunakan kata kunci (keyword). Sekarang mari kita lihat delapan tipe data primitif secara mendetail.

byte

- Tipe data byte adalah bilangan bulat komplement dua bertanda 8-bit
- Nilai minimum adalah -128 (-2^7)
- Nilai maksimum adalah 127 (inklusif) ($2^7 - 1$)
- Nilai defaultnya adalah 0
- Tipe data byte digunakan untuk menghemat ruang dalam array besar, terutama sebagai pengganti integer, karena byte empat kali lebih kecil dari integer (int).
- Contoh: byte a = 100, byte b = -50

short

- Tipe data short adalah integer komplement dua bertanda(signed) 16-bit
- Nilai minimum adalah -32,768 (-2^{15})
- Nilai maksimum adalah 32.767 (inklusif) ($2^{15} - 1$)
- Tipe data short juga dapat digunakan untuk menghemat memori sebagai tipe data byte. Pendek adalah 2 kali lebih kecil dari integer (int)
- Nilai defaultnya adalah 0.
- Contoh: short s = 10000, short r = -20000

int

- Tipe data int adalah integer komplement dua bertanda 32-bit.

- Nilai minimum adalah - 2.147.483.648 (-2^{31})
- Nilai maksimum adalah 2.147.483.647 (inklusif) ($2^{31} - 1$)
- Integer umumnya digunakan sebagai tipe data default untuk nilai integral kecuali jika ada masalah tentang memori.
- Nilai defaultnya adalah 0
- Contoh: `int a = 100000`, `int b = -200000`

long

- Tipe data panjang adalah bilangan bulat (integer) komplemen dua yang bertanda dengan lebar data 64-bit
- Nilai minimum adalah -9.223.372.036.854.775.808 (-2^{63})
- Nilai maksimum adalah 9.223.372.036.854.775.807 (inklusif) ($2^{63} - 1$)
- Tipe ini digunakan menyimpan data yang lebih besar dari lebar yang dipunya oleh int.
- Nilai defaultnya adalah 0L
- Contoh: `long a = 100000L`, `long b = -200000L`

float

- Tipe data float adalah floating point IEEE 754 32-bit presisi tunggal
- Float terutama digunakan untuk menghemat memori dalam array besar pada bilangan floating point
- Nilai defaultnya adalah 0.0f
- Tipe data float tidak pernah digunakan untuk nilai yang tepat seperti mata uang
- Contoh: `float f1 = 234.5f`

double

- tipe data ganda adalah titik mengambang IEEE 754 64-bit presisi ganda
- Tipe data ini umumnya digunakan sebagai tipe data default untuk nilai desimal, umumnya pilihan default
- Tipe data ganda tidak boleh digunakan untuk nilai yang tepat seperti mata uang
- Nilai defaultnya adalah 0.0d
- Contoh: `double d1 = 123,4`

boolean

- tipe data boolean mewakili satu bit informasi
- Hanya ada dua kemungkinan nilai: benar dan salah
- Tipe data ini digunakan untuk penanda sederhana yang mendefinisikan kondisi benar / salah
- Nilai default salah / false
- Contoh: `boolean one = true`

char

- tipe data char adalah karakter Unicode 16-bit tunggal
- Nilai minimum adalah '\u0000' (atau 0)
- Nilai maksimum adalah '\uffff' (atau termasuk 65.535)
- Tipe data Char digunakan untuk menyimpan karakter apapun
- Contoh: char letterA = 'A'

Referensi Jenis Data

- Variabel referensi dibuat menggunakan konstruktor kelas yang ditentukan. Mereka digunakan untuk mengakses objek. Variabel ini dideklarasikan sebagai jenis tertentu yang tidak dapat diubah. Misalnya Employee, puppy, dll.
- Objek kelas dan berbagai tipe variabel array berada di bawah tipe data referensi.
- Nilai default dari variabel referensi apa pun adalah nol.
- Variabel referensi dapat digunakan untuk merujuk objek apa pun dari tipe yang dideklarasikan atau tipe apa pun yang kompatibel.
- Contoh: Animal animal = new Animal ("giraffe");

Literal Java

Literal adalah representasi kode sumber dari nilai tetap. Mereka direpresentasikan langsung dalam kode tanpa perhitungan apa pun.

Literal dapat ditetapkan ke variabel tipe primitif apa pun. Misalnya -

```
byte a = 68;  
char a = 'A';
```

byte, int, long, dan short dapat diekspresikan dalam sistem bilangan desimal (basis 10), heksadesimal (basis 16) atau oktal (basis 8) juga.

Awalan 0 digunakan untuk menunjukkan oktal, dan awalan 0x menunjukkan heksadesimal saat menggunakan sistem bilangan ini untuk literal. Misalnya -

```
int decimal = 100;  
int octal = 0144;  
int hexa = 0x64;
```

String literal di Java ditentukan seperti di sebagian besar bahasa lain dengan mengapit urutan karakter di antara sepasang tanda kutip ganda. Contoh literal string adalah -

Contoh

```
"Hello World"  
"two\nlines"
```

```
"\"This is in quotes\""
```

Jenis string dan karakter literal dapat berisi karakter Unicode apa pun. Misalnya -

```
char a = '\u0001';  
String a = "\u0001";
```

Bahasa Java mendukung beberapa escape sequence khusus untuk String dan literal char juga. Mereka adalah -

Notasi	Karakter terwakili
\n	Baris baru (0x0a)
\r	return (0x0d)
\f	Formfeed (0x0c)
\b	Spasi mundur (0x08)
\s	Spasi (0x20)
\t	tab
\"	Kutipan ganda
\'	Kutipan tunggal
\\	garis miring terbalik
\ddd	Karakter oktal (ddd)
\uxxxx	Karakter UNICODE heksadesimal (xxxx)

--	--

Java has several arithmetic operators to perform math operations.

Symbol	Operator Description
+	Addition operator
-	Subtraction operator
*	Multiplication operator
/	Division operator (finds the quotient)
%	Modular operator (finds the remainder)
++	Increment operator (adds one). Is a unary operator.
--	Decrement operator (subtracts one). Is a unary operator.

Arithmetic Operators Precedence (Kemunculan operator aritmatik)

- All math expressions are evaluated following the order of precedence:
 - Expressions in parenthesis “()” are handled first.
 - All multiplication, division, and modular operations are handled next, working from left to right.
 - Finally, all addition and subtraction operations are handled, working from left to right.

Increments and Decrements

- Increments and decrements are handled first for preincrement notation and last for post-increment notation.
- Increment in Java means to add one to the variable value.
- Decrement in Java means to subtract one from the variable.
- Pre-increment notation:
 - ++x;
- Post-increment notation:
 - x++;

Catatan :

Pemetaan Tipe data :

boolean

byte -- short -- int -- long -- float -- double

/

/

char

int a = 10;

double b = a; ==> promotion ==> double b = (double)a;

\

\==> implicit

double c = 8.99;

int d = c; XXX ==> casting ==> int d = (int)c;

\

\==> explicit

byte e = 10;

byte f = 20;

byte g = e + f; XXX ==> byte g = (byte)(e + f);

float h = 8.5; XXX ==> float h = 8.5F;

float h = 8.5f;

float i = 8;