

MODUL PEMOGRAMAN JAVA

Oleh:

Chalifa Chazar

Pertemuan 3

TIPE DATA, OPERATOR, dan VARIABEL

Tujuan:

- » Mahasiswa mampu memahami konsep-konsep dasar tipe data, operator dan variabel dalam bahasa pemrograman Java.
- » Mahasiswa mampu menggunakan tools Netbeans sebagai IDE dalam pengembangan aplikasi Java

Pustaka:

- » Bambang Hariyanto (2014): **Esensi-Esensi Bahasa Pemrograman Java (Revisi Keempat)**. Informatika. Bandung.

1. Tipe Data

Tipe data mendefinisikan metode representasi informasi dan cara informasi diinterpretasikan. Tipe data berkaitan dengan penyimpanan variable di memori karena tipe data suatu variabel menentukan cara kompilator menginterpretasikan isian memori.

Bahasa Java dikatakan aman dan kuat karena:

- » Setiap variabel harus mempunyai tipe.
- » Semua penugasan, baik eksplisit atau via parameter di pemanggilan metode melewati pemeriksaan kompatibilitas tipe.
- » Kompilator Java memeriksa semua ekspresi dan parameter untuk menjamin tipe-tipenya kompatibel.

1.1. Tipe Data Bilangan Bulat

Terdapat 5 tipe data untuk bilangan bulat, antara lain:

1. byte
2. short
3. int

4. long
5. char

Masing-masing tipe memiliki kebutuhan memori yang berbeda-beda.

Tipe	Ukuran	Range
byte	8 byte	-128 s/d 127
short	16 byte	-32768 s/d 32767
int	32 byte	-2147483648 s/d 2147483647
long	64 byte	-9223372036854775808 s/d 9223372036854775807
char	16 byte	0 s/d 65535

1.2. Tipe Data Bilangan Mengambang

Bilangan mengambang atau disebut juga bilangan real memiliki 2 jenis tipe data, antara lain:

1. float
2. double

Masing-masing tipe memiliki kebutuhan memori yang berbeda-beda. Tipe *float* memerlukan 32 byte, sebagai *single-precision* dan tipe *double* memerlukan 64 byte sebagai *double-precision*.

Tipe	Ukuran	Range
float	32 byte	-3.4e038 s/d 3.4e+038
double	64 byte	1.7e-308 s/d 1.7e+308

1.3. Tipe Data Logika

Untuk nilai yang bersifat logis atau logika digunakan tipe data **boolean**. Variabel tipe ini memiliki salah satu dari dua nilai, yaitu *true* dan *false*.

1.4. Tipe Data Unicode

Tipe data **Unicode** direpresentasikan menggunakan tipe *char*. Tanda petik tunggal digunakan untuk menandai konstanta/literal *char*. Perhatikan contoh berikut ini:

'H' adalah satu karakter, sementara "H" adalah string yang berisi satu karakter. Keduanya memiliki nilai yang berbeda.

Java menggunakan **Unicode** untuk merepresentasikan karakter. **Unicode** mendefinisikan himpunan karakter internasional secara penuh, dapat merepresentasikan semua karakter dalam bahasa manusia di dunia. **Unicode** ini merupakan penyatuan seluruh himpunan karakter seperti Latin, Greek, Arabic, Cyrillic, Hebrew, Katakana, Hangul, dan banyak lagi.

2. Operator Bilangan Bulat

Terdapat 3 tipe operasi yang dapat dilakukan pada integer, yaitu:

- » Unary, berlaku pada satu angka integer
- » Binary, berlaku pada pasangan bilangan integer
- » Relasional, berlaku pada bilangan integer tapi menghasilkan hasil boolean bukan bilangan integer

2.1. Operator Bilangan Bulat Unary

Deskripsi	Operator
Increment	++
Decrement	--
Negasi	-
Bitwise complement	~

Perhatikan contoh berikut ini.

```
public static void main(String[] args) {  
    int m = 7;  
    int n = 2;  
    int x = 2 * ++m;  
    int y = 2 * n++;  
    System.out.println(x);  
    System.out.println(y);  
}
```

Apakah hasil output dari variabel x dan y?

Operator *increment* dan *decrement* (++) dan (--) menaikkan dan menurunkan variabel dengan 1. Operator ini dapat digunakan dalam bentuk prefiks atau postfiks.

Operator prefiks mempunyai efek sebelum evaluasi ekspresinya. Operator postfiks mempunyai efek setelah ekspresi dievaluasi. Operator prefiks unary segera sebelum variabel. Operator postfiks unary segera setelah variabel.

Perhatikan contoh berikut ini.

```
public static void main(String[] args) {  
    int a = 10;  
    int b = -a;  
    System.out.println(b);  
}
```

Apakah hasil output dari variabel b?

Catatan:

Operator *negasi* unary (-) merubah tanda nilai bilangan bulat.

Perhatikan contoh berikut ini.

```
public static void main(String[] args) {  
    int i = 8;  
    int j = ~i;  
    System.out.println(j);  
}
```

Apakah hasil output dari variabel j?

Catatan:

Operator *bitwise complement* (~) melakukan negasi *bitwise* nilai bilangan bulat. *Bitwise negation* berarti masing-masing bit di angka maka semua bilangan biner 0 menjadi 1 dan semua bilangan biner 1 menjadi 0.

2.2. Operator Bilangan Bulat Biner

Operator bilangan bulat biner beroperasi pada pasangan bilangan bulat. Berikut ini adalah tabel operator bilangan bulat biner.

Deskripsi	Operator
Penambahan	+
Pengurangan	-
Perkalian	*
Pembagian	/
Sisa bagi	%
Bitwise AND	&
Bitwise OR	
Bitwise XOR	^
Left-shift	<<
Right-shift	>>
Zero-fill-right-shift	>>>

2.3. Operator Relasional Bilangan Bulat

Operator relasional yang beroperasi pada bilangan bulat menghasilkan nilai **boolean**. Berikut ini adalah tabel operator bilangan bulat relasional.

Deskripsi	Operator
Kurang dari	<
Lebih besar dari	>
Lebih kecil atau sama dengan	<=
Lebih besar dari atau sama dengan	>=
Sama dengan	==
Tidak sama dengan	!=

3. Operator Bilangan Mengambang

Sama halnya dengan operator bilangan bulat, terdapat 3 tipe operasi bilangan mengambang, yaitu unary, biner dan relasional.

3.1. Operator Unary Bilangan Mengambang

Operator bilangan mengambang unary beroperasi pada angkat titik mengambang tunggal. Berikut ini adalah tabel operator bilangan mengambang unary.

Deskripsi	Operator
Increment	++
Decrement	--

3.2. Operator Biner Bilangan Mengambang

Operator ini beroperasi pada pasangan bilangan mengambang. Berikut ini adalah tabel operator biner bilangan titik mengambang.

Deskripsi	Operator
Penambahan	+
Pengurangan	-
Perkalian	*
Pembagian	/
Sisa bagi	%

3.3. Operator Relasional Bilangan Mengambang

Operator relasional titik mengambang membandingkan dua operan berupa bilangan titik mengambang dan menghasilkan tipe **boolean**. Operator relasional bilangan titik mengambang sama dengan operator relasional untuk bilangan bulat.

4. Operator Boolean

Operator **boolean** beroperasi pada tipe **boolean** dan menghasilkan tipe **boolean**. Berikut ini adalah daftar operator **boolean**.

Deskripsi	Operator
Evaluation AND	&
Evaluation OR	
Evaluation XOR	^
Logical AND	&&
Logical OR	
Negation	!
Equal-to	==
Not-equal-to	!=
Conditional	?:

5. Variabel

Variabel adalah suatu tempat untuk menampung data yang nilainya selalu berubah. Data yang disimpan dalam variabel akan hilang setelah program selesai dieksekusi. Pada pemograman

Java, setiap variabel harus mempunyai tipe. Anda dapat mendeklarasikan variabel dengan menempatkan tipe data terlebih dahulu, diikuti nama variabel. Perhatikan contoh mendeklarasikan suatu variabel di bawah ini.

```
double salary;  
  
int number;  
  
int i, j;  
  
String name;
```

Catatan:

Tanda titik koma (;) digunakan untuk mengakhiri masing-masing deklarasi. Sedangkan tanda koma (,) digunakan untuk mendeklarasikan suatu variabel yang merupakan satu pernyataan lengkap.

6. Konstanta

Konstanta adalah suatu tempat untuk menampung data yang nilainya tetap dan tidak pernah berubah. Pada pemograman Java, untuk mendeklarasikan konstanta digunakan kata kunci (keyword) **final**. Perhatikan contoh mendeklarasikan suatu variabel di bawah ini.

```
package dasar_pemograman;  
  
/**  
 *  
 * @author FAE  
 */  
public class matematika {  
    public static double keliling_lingkaran(double jari2){  
        final double phi = 3.14;  
        return 2*phi*jari2;  
    }  
}
```


Latihan!

1. Buatlah program untuk menghitung bentuk penjumlahan, pengurangan, perkalian dan pembagian dengan Java
2. Buatlah program untuk menghitung luas dan keliling suatu bangunan

----- **GOOD LUCK** -----