

# ELEMEN JAVA

Dosen : Sulistyowati, ST., M.Kom.

# Identifier di Java

- Identifier adalah nama yang diberikan kepada variabel, method, kelas, paket, dan interface
- Adapun tatacara penamaan identifier:

1. Case sensitive, huruf kapital dan kecil dibedakan
2. Identifier yang diberikan oleh pemrogram tidak boleh sama dengan keyword yang ada di Java
3. Dimulai dengan huruf atau underscore (garis bawah) atau tanda (\$).

Namun sebisa mungkin diawali dengan huruf karena mungkin identifier dengan awalan underscore dan (\$) digunakan untuk pemrosesan internal dan file import

4. Karakter berikutnya dapat berupa huruf atau angka 0 sampai 9. Simbol-simbol seperti '+' dan spasi tidak dapat digunakan

# Keywords di Java

- Keyword adalah identifier yang digunakan Java untuk suatu tujuan khusus. Daftar keyword Java adalah sebagai berikut:

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

# Contoh Penulisan Nama Variabel yang Diizinkan

- @2var
- \_status
- tanggal
- jumlahBarang
- nama\_kecil
- final\_test
- int\_float

# Tipe Data di Java

- Tipe data dalam Java dibagi dalam dua kategori:
  1. Sederhana (primitif)
  2. Komposit (referensi)

# Tipe data sederhana (primitif)

- Tipe data sederhana merupakan tipe inti. Tipe sederhana tidak diturunkan dari tipe lain. Terdapat delapan tipe data primitif di Java:
  - Empat tipe adalah untuk bilangan bulat (integer) : byte, short, int, long
  - Dua untuk tipe angka pecahan (floating point): float, double
  - Satu untuk tipe karakter, yaitu char
  - Satu untuk tipe boolean yang berisi nilai logika: true/false

# Tipe data sederhana (primitif)

## Integer

Tipe Data	Panjang	Rentang Nilai	Contoh Nilai
byte	8 bit	$-2^7$ sampai $2^7 - 1$ (-128 sampai 127) (256 kemungkinan nilai)	5 -126
short	16 bit	$-2^{15}$ sampai $2^{15} - 1$ (-32.768 sampai 32.767) (65.535 kemungkinan nilai)	9 -23659
int	32 bit	$-2^{31}$ sampai $2^{31} - 1$ (-2.147.483.648 sampai 2.147.483.647) (4.294.967.296 kemungkinan nilai)	2067456397 -1456398567
long	64 bit	$-2^{63}$ sampai $2^{63} - 1$ (-9.223.372.036.854.775.808 sampai 9.223.372.036.854.775.807) (18.446.744.073.709.551.616 kemungkinan nilai)	3L - 2147483648L 67L

# Tipe data sederhana (primitif)

## Floating Point

Tipe data *floating point* merupakan tipe data untuk variabel yang nilai-nilainya adalah bilangan real (dapat mempunyai pecahan desimal).

Tipe Data	Panjang	Contoh Penulisan Nilai yang Diperbolehkan
float	32 bit	78F -34736.86F 6.4E4F ( sama dengan $6,4 \times 10^4$ )
double	64 bit	-2356 3.5E7 67564788965.567



# Tipe data sederhana (primitif)

## Char

Tipe data textual merupakan tipe data untuk variabel yang nilai-nilainya adalah karakter tunggal. Nilai variabel char ditulis dengan diberi tanda kutip tunggal '...'. Berikut contoh penggunaan tipe data char :

```
char alphabet = 'A' ;
```

## Logika (Boolean)

Tipe data logika adalah tipe data yang hanya memiliki 2 kemungkinan nilai, yaitu *true* atau *false*. Hanya satu tipe data logika pada teknologi Java, yaitu *boolean*.

# Tipe Data Komposit

Tipe data komposit merupakan tipe data yang disusun dari tipe data sederhana atau tipe komposit lain yang sudah ada. Tipe ini dapat berupa array, string, kelas, dan interface. Khusus untuk String pada Java dikenali sebagai kelas, bukan sebagai array of character. String pada Java diapit oleh tanda petik ganda ("....."), contoh:

```
String s="Saya makan nasi";
```

# Operator di Java

- **Operator Aritmatik**

Arti Operator	Operator	Contoh Pemakaian	Keterangan
Penjumlahan	+	$A = \text{num1} + \text{num2}$	
Pengurangan	-	$B = \text{num1} - \text{num2}$	
Perkalian	*	$C = \text{num1} * \text{num2}$	
Pembagian	/	$D = \text{num1} / \text{num2}$	jika num1 dan num2 adalah integer, pembagian akan menghasilkan nilai integer tanpa mengikutsertakan sisa, jika terdapat sisa.
Sisa (modulus)	%	$E = \text{num1} \% \text{num2}$	Hasil operasi modulus adalah sisa dari operasi $\text{num1} / \text{num2}$ . Hasil operasi modulus memiliki tanda ( + / - ) yang sama dengan operand pertama

- **Operator Unary**

Arti Operator	Operator	Contoh Pemakaian
Pre-Increment	++operand	int i = 8; int j = ++i; i bernilai 9, j bernilai 9
Post-Increment	operand++	int i = 8; int j = i++; i bernilai 9, j bernilai 8
Pre-Decrement	--operand	int i = 8; int j = --i; i bernilai 7, j bernilai 7
Post-Decrement	operand--	int i = 8; int j = i--; i bernilai 7, j bernilai 8

# Prioritas Operator

- operator yang berada dalam tanda kurung “( ... )” atau disebut juga *parantheses*;
- operator-operator *increment* atau *decrement*;
- operator - operator perkalian atau pembagian, yang urutan operasinya dari kiri ke kanan;
- operator-operator penjumlahan atau pengurangan, yang urutan operasinya dari kiri ke kanan;
- Contoh : `int c = 12 * 3 + 5 / (8 - 3) ;`

Maka urutan operasinya adalah sebagai berikut :

```
int c = 12 * 3 + 5 / 5 ;
```

```
int c = 36 + 5 / 5 ;
```

```
int c = 36 + 1 ;
```

```
int c = 37 ;
```

- **Operator Shorthand / Pemendekan**

Operator ini digunakan dengan menyingkat penulisan.

Operator	Nama operator	Contoh	Ekuivalen
<code>+=</code>	Addition assignment	<code>i += 8</code>	<code>i = i + 8</code>
<code>-=</code>	Subtraction assignment	<code>f -= 8.0</code>	<code>f = f - 8.0</code>
<code>*=</code>	Multiplication assignment	<code>i *= 8</code>	<code>i = i * 8</code>
<code>/=</code>	Division assignment	<code>i /= 8</code>	<code>i = i / 8</code>
<code>%=</code>	Remainder assignment	<code>i %= 8</code>	<code>i = i % 8</code>

# • Operator Relasi

Arti operator	Operator	Contoh pemakaian
Is equal to (sama dengan)	<code>==</code>	<pre>int i = 1; System.out.println(i==1);  // (output : true)</pre>
Is not equal to (tidak sama dengan)	<code>!=</code>	<pre>int i = 1; System.out.println(i!=1);  // (output : false)</pre>
Is less than (lebih kecil)	<code>&lt;</code>	<pre>int i = 1; System.out.println(i&lt;1);   // (output : false)</pre>
Is less than or equal to (lebih kecil atau sama dengan)	<code>&lt;=</code>	<pre>int i = 1; System.out.println(i&lt;=1);  // (output : true)</pre>
Is greater than (lebih besar)	<code>&gt;</code>	<pre>int i = 1; System.out.println(i&gt;1);   // (output : false)</pre>
Is greater than or equal to (lebih besar atau sama dengan)	<code>&gt;=</code>	<pre>int i = 1; System.out.println(i&gt;=1);  // (output : true)</pre>

- **Operator kondisional**

Operator	Keterangan	Contoh pemakaian
&&	AND	<pre>int i = 1; int j = 2; System.out.println((i&lt;1) &amp;&amp; (j&gt;0)); // (output : false)</pre>
	OR	<pre>int i = 1; int j = 2; System.out.println((i&lt;1)    (j&gt;0)); // (output : true)</pre>
!	NOT	<pre>int i = 1; System.out.println(!(i&lt;3)); // (output : false)</pre>



# Komentar

- Komentar adalah bagian kode program yang tidak dibaca saat program dijalankan. Komentar sangat penting untuk memperjelas kode program agar lebih mudah dipahami. Cara penulisan komentar :

Jenis Komentar	Algoritma	Java
Komentar untuk satu baris	<i>{komentar}</i>	<i>//komentar</i>
Komentar untuk beberapa baris	<i>{</i> <i>.....</i> <i>komentar</i> <i>.....</i> <i>}</i>	<i>/*</i> <i>.....</i> <i>komentar</i> <i>.....</i> <i>*/</i>

# Kata Kunci Final

Dalam aplikasi terkadang dibutuhkan suatu variabel yang dicegah untuk dimodifikasi oleh program lainnya. Hal ini dilakukan untuk menjaga agar logika program tetap terjaga. Untuk membuat konstanta, digunakan sebuah modifier final.

- **Contoh**

```
class Lingkaran{  
    final double pi = 3.14; /*konstanta yg nilainya  
                               tidak dapat diubah lagi*/  
    double jari2;    double luas;  
    public double hitungLuas(){  
        luas = pi*jari2*jari2;  
        return luas;  
    }  
    public double hitungKeliling(){  
        return 2*pi*jari2;  
    }  
}
```