

Tipe Data, Operator, Modifier

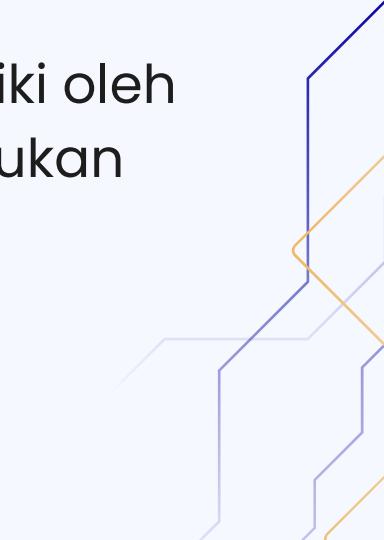
Ratna Mufidah, S.Kom., M.Kom.



Tipe Data

Klasifikasi data untuk memberitahu compiler atau interpreter tentang maksud programmer akan penggunaan data tersebut.

Tipe data akan menentukan tipe nilai yang dimiliki oleh sebuah variabel dan operasi yang dapat dilakukan terhadap variabel tersebut.



Tipe Data - Java

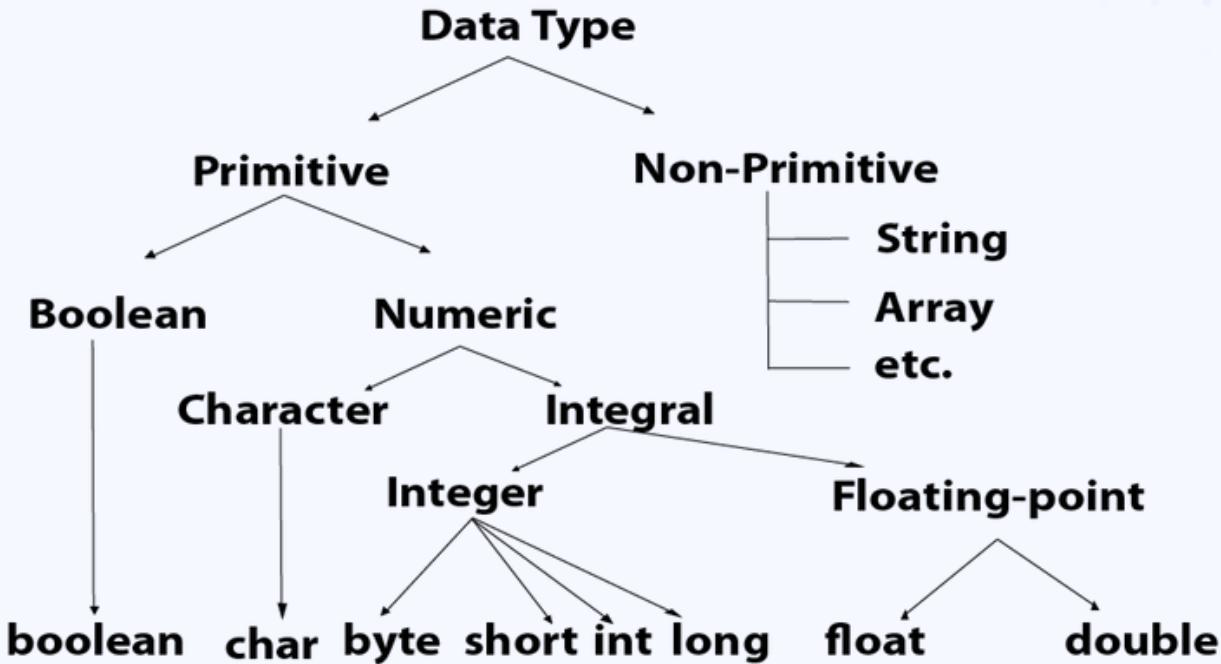
Tipe Data Primitif

- Tipe data yang merupakan kata kunci di Java (tertanam di compiler Java), sehingga pemrosesan jauh lebih cepat.
- Penulisannya menggunakan huruf kecil (*lowercase*)
- Contoh: int, double, float

Tipe Data Non-Primitif (Reference Data Type)

- Tipe data berupa class yang ada di library Java (java.lang)
- Penulisannya diawali dengan huruf kapital.
- Contoh: String, Array

Tipe Data - Java



Tipe Data - Java

Primitif

Tipe Data	Ukuran	Deskripsi
byte	1 byte	Menyimpan bilangan bulat dari -128 hingga 127
short	2 bytes	Menyimpan bilangan bulat dari -32.768 hingga 32.767
int	4 bytes	Menyimpan bilangan bulat dari -2,147,483,648 hingga 2,147,483,647

Tipe Data - Java

Primitif

Tipe Data	Ukuran	Deskripsi
long	8 bytes	Menyimpan bilangan bulat dari -9,223,372,036,854,775,808 hingga 9,223,372,036,854,775,807
float	4 bytes	Menyimpan bilangan pecahan. Cukup untuk menyimpan 6 hingga 7 angka desimal
double	8 bytes	Menyimpan bilangan pecahan. Cukup untuk menyimpan 15 angka desimal

Tipe Data - Java

Primitif

Tipe Data	Ukuran	Deskripsi
boolean	1 bit	Menyimpan nilai True atau False
char	2 bytes	Menyimpan karakter tunggal (Unicode)

Tipe Data - Java

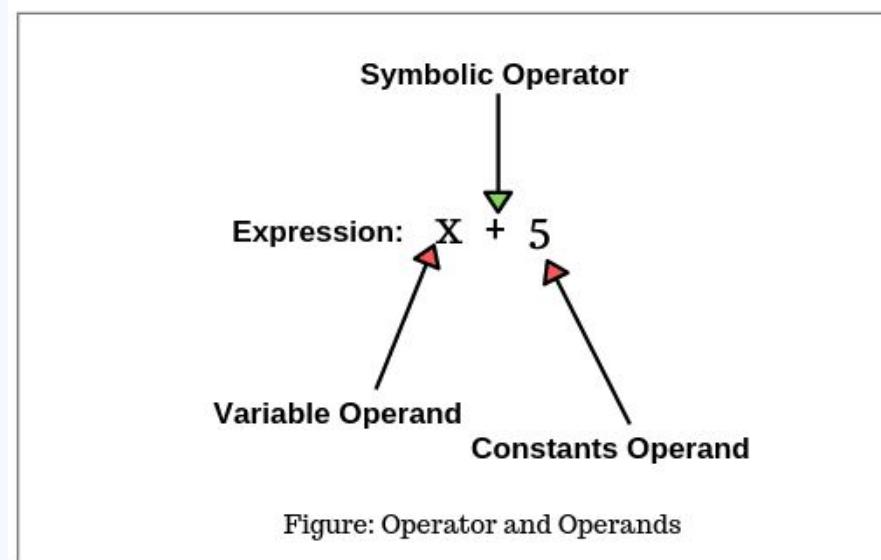
```
2oimport java.lang.*;  
3 import java.util.*;  
4  
5 public class tipedata {  
6  
7     public static void main(String[] args) {  
8         // TODO Auto-generated method stub  
9         System.out.println("TIPE DATA PRIMITIF\n");  
10        int x = 10;  
11        int y = x;  
12        System.out.print("Inisial: ");  
13        System.out.println("x = " + x + ", y = " + y);  
14  
15        y = 15;  
16  
17        System.out.print("Setelah y diubah menjadi 15: ");  
18        System.out.println("x = " + x + ", y = " + y);  
19  
20    }
```

```
21  
22  
23  
24  
25  
26 System.out.println("\n\nTIPE DATA NON-PRIMITIF\n");  
27 int[] c = { 10, 20, 30, 40 };  
28  
29 int[] d = c;  
30  
31 System.out.println("Inisial");  
32 System.out.println("Array c: "+ Arrays.toString(c));  
33 System.out.println("Array d: "+ Arrays.toString(d));  
34  
35 System.out.println("\nModifikasi nilai pada indeks ke-1 dalam array d\n");  
36 d[1] = 50;  
37  
38 System.out.println("Setelah diubah");  
39 System.out.println("Array c: "+ Arrays.toString(c));  
40 System.out.println("Array d: "+ Arrays.toString(d));  
41 }
```

Apa yang terjadi?

Operator

Simbol yang digunakan untuk melakukan operasi tertentu

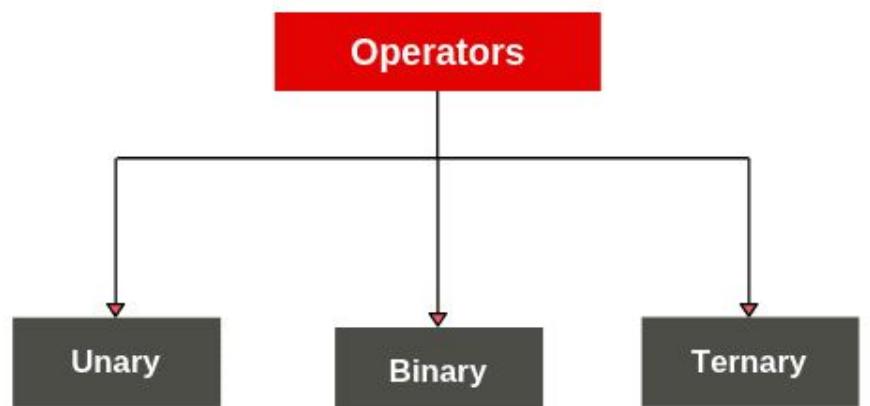


Operator

$d = a * c + b$

Sebutkan manakah
operator dan manakah
operand?

Operator



Jenis operator berdasarkan jumlah operand:

1. **Operator Unary**: operator yang melibatkan satu operand
2. **Operator Binary**: operator yang melibatkan dua operand
3. **Operator Ternary**: operator yang melibatkan tiga operand

Operator

Jenis Operator secara Umum

1. Operator Aritmatika
2. Operator Penugasan
3. Operator Penggabungan
4. Operator Increment dan Decrement
5. Operator Bit
6. Operator Pembanding
7. Operator Logika

Operator

Operator Aritmatika

- digunakan untuk **melakukan operasi aritmatika dasar** seperti penjumlahan, pengurangan, perkalian, dan pembagian pada tipe data numerik.
- hasil operasi matematika akan mengikuti tipe data dari operand

Operators	Meaning	Description
1. +	Addition or unary plus	Performs addition operation.
2. –	Subtraction or unary minus	Performs subtraction operation.
3. *	Multiplication	Performs multiplication operation.
4. /	Division	Performs division operation.
5. %	Modulo division (Remainder)	Performs remainder after division operation.

<https://www.scientecheeasy.com/2020/05/java-operators-arithmetic.html/>



Operator

Operator Penugasan

- Operator penugasan berguna untuk memberi nilai ke suatu variabel
- Operator penugasan menggunakan tanda sama dengan (=)
- Operator penugasan digabungkan dengan operator aritmatika membentuk operator penugasan gabungan (*compound assignment*)

Contoh:

- **Penugasan sederhana:**
`a = 10`
`usia = 20`
- **Penugasan gabungan:**
`a += 10`
yang artinya
`a = a+10`

Operator

Operator Penugasan

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
&=	x &= 3	x = x & 3
=	x = 3	x = x 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

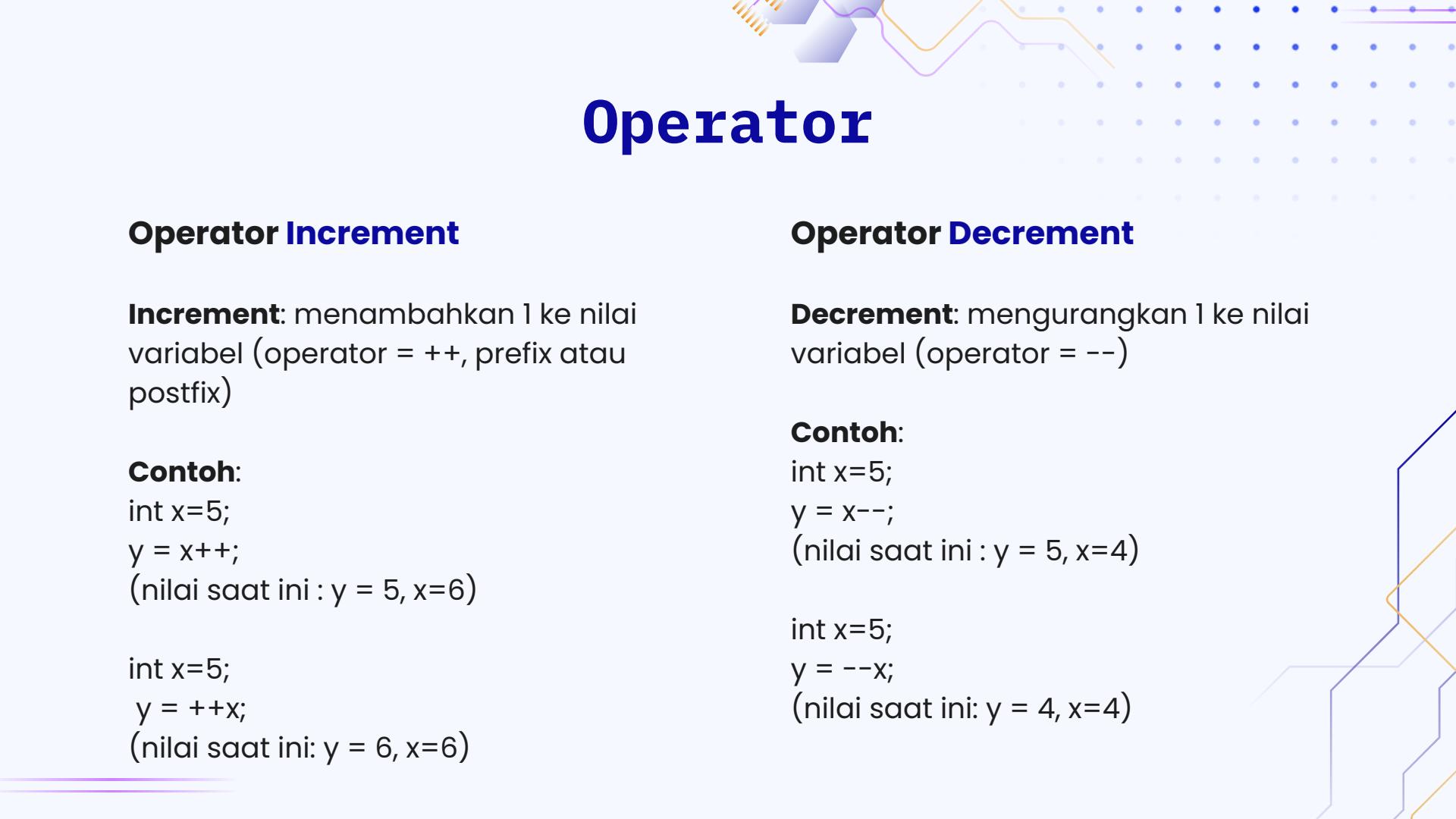
Operator

Operator Penggabungan

Operator (+) dapat digunakan untuk penggabungan String dan String maupun String dan Bilangan

Contoh

```
System.out.println("Nama saya" + "Budi");  
int usia= 19;  
System.out.println("Usia saya" + usia);
```



Operator

Operator Increment

Increment: menambahkan 1 ke nilai variabel (operator = ++, prefix atau postfix)

Contoh:

```
int x=5;  
y = x++;  
(nilai saat ini : y = 5, x=6)
```

```
int x=5;  
y = ++x;  
(nilai saat ini: y = 6, x=6)
```

Operator Decrement

Decrement: mengurangkan 1 ke nilai variabel (operator = --)

Contoh:

```
int x=5;  
y = x--;  
(nilai saat ini : y = 5, x=4)
```

```
int x=5;  
y = --x;  
(nilai saat ini: y = 4, x=4)
```



Operator

Operator Increment

```
int x, y, z;  
x = 19;  
y = x++;  
z = ++x;
```

Berapakah nilai dari x, y, z?

Operator Decrement

```
int x, y, z;  
x = 19;  
y = x--;  
z = --x;
```

Berapakah nilai dari x, y, z?

Operator

Operator Bitwise

digunakan untuk melakukan manipulasi bit individual dari suatu angka

&	operasi AND untuk bit
	operasi OR untuk bit
^	operasi Ex OR untuk bit
~	operasi NOT untuk bit
<<	geser kiri (geser 1 bit = *2)
>>	geser kanan (geser 1 bit = /2)
>>>	geser kanan tak bertanda

Operator

Operator Bitwise

Contoh:

a = 5 = 0101 (Biner)

b = 7 = 0111 (Biner)

Operasi Bitwise AND untuk 5 dan 7

0101

& 0111

0101 = 5

Contoh:

a = 5 = 0101 (Biner)

b = 7 = 0111 (Biner)

Operasi Bitwise OR untuk 5 dan 7

0101

| 0111

0111 = 7

Operator

Operator Pembanding

Operator pembanding menggunakan ekspresi dengan nilai balik boolean (true or false)

Operator	Name	Example
<code>==</code>	Equal to	<code>x == y</code>
<code>!=</code>	Not equal	<code>x != y</code>
<code>></code>	Greater than	<code>x > y</code>
<code><</code>	Less than	<code>x < y</code>
<code>>=</code>	Greater than or equal to	<code>x >= y</code>
<code><=</code>	Less than or equal to	<code>x <= y</code>

https://www.w3schools.com/java/java_operators.asp

Operator

Operator Pembanding

```
1 package kode;
2
3 public class compare {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int x = 5;
8         int y = 3;
9         System.out.println(x > y);
10    }
11}
```

Operator

Operator Logika

- digunakan untuk menentukan logika antar variabel atau nilai
- menggunakan ekspresi dengan nilai balik Boolean (true or false)

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	$x < 5 \&\& x < 10$
	Logical or	Returns true if one of the statements is true	$x < 5 x < 4$
!	Logical not	Reverse the result, returns false if the result is true	$!(x < 5 \&\& x < 10)$

Operator

Operator Logika

```
1 package kode;
2
3 public class logic {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int x = 5;
8         System.out.println(x > 3 && x < 10);
9     }
10
11 }
```

Modifier

Modifier

Modifier adalah keyword yang diletakkan di depan class, interface, variabel atau method

Jenis Modifier:

1. Access Modifier:

Pengaturan pengaksesan dari variable dan method

2. Static Modifier:

Membuat method dan variable menjadi milik class, bukan object

Tidak perlu membuat object untuk penggunaan variable (field) dan method

3. Final Modifier:

Menyatakan bahwa sesuatu (class, method, variable) sudah final dan tidak dapat diubah

Modifier

```
3 public class bilangan {  
4     int pencacah = 0;  
5     int nilai;  
6  
7     public bilangan(int nilai){  
8         this.nilai = nilai;  
9         pencacah++;  
10    }  
11  
12    public void info(){  
13        System.out.println("Nilai:" + nilai);  
14        System.out.println("Pencacah:" + pencacah);  
15        System.out.println("");  
16    }  
17 }
```

bilangan.java

```
3 public class bilanganmain {  
4  
5     public static void main(String[] args) {  
6         // TODO Auto-generated method stub  
7         bilangan b1 = new bilangan(50);  
8         b1.info();  
9  
10        bilangan b2 = new bilangan(15);  
11        b2.info();  
12  
13        bilangan b3 = new bilangan(30);  
14        b3.info();  
15    }  
16  
17 }
```

bilanganmain.java

Jika variabel pencacah pada kelas bilangan diubah menjadi static, apa output yang dihasilkan? mengapa?

Modifier

Acces Modifier pada Variabel dan Method

Modifier	Dalam Class yang Sama	Dalam Package yang Sama	Dalam SubClass	Dalam Package Lain
private	✓			
tanpa tanda	✓	✓		
protected	✓	✓	✓	
public	✓	✓	✓	✓

Modifier

Static Modifier

- Sama seperti *static* variabel, ketika method ditambahkan *static* modifier, maka method tersebut dikontrol oleh class, dan bukan oleh object lagi
- Pemanggilan method dapat dilakukan tanpa membuat object
- *Static method* biasanya digunakan pada method yang hanya melakukan perhitungan matematika

Thanks !

Ada pertanyaan?

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

