

# BAB 3

## ELEMEN DASAR C++

# OPERATOR

- Operator merupakan simbol yang digunakan untuk melakukan suatu operasi atau manipulasi.

Misalnya untuk :

- Menjumlahkan dua buah nilai
  - Memberikan nilai ke suatu variabel
  - Membandingkan kesamaan dua buah nilai
- 
- Sebagian operator C++ tergolong sebagai operator binary, yaitu operator yang dikenakan terhadap dua buah nilai (operand). Contoh :  $a+b$

# OPERATOR (con't)

- Sifat operator

SIFAT	KETERANGAN	CONTOH
Unary	Operator ini hanya melibatkan sebuah operand	$-a$
Binary	Operator ini melibatkan 2 buah operand	$a + b$
Ternary	Operator ini melibatkan 3 buah operand	$(a > b) ? a : b$

# OPERATOR – *Operator Aritmatika*

- Operator dalam C++ untuk operasi aritmatika :

OPERATOR	KETERANGAN	CONTOH
*	Perkalian	$2 * 3$
/	Pembagian	$8 / 2$
%	Sisa pembagian (modulus)	$9 \% 2$
+	Penjumlahan	$3 + 5$
-	Pengurangan	$5 - 2$

- Contoh :  $\text{diskriminan} = b^2 - 4ac$   
Dalam C++ ditulis :  $\text{diskriminan} = (b*b) - (4*a*c)$

# OPERATOR – *Operator Aritmatika*

- PRIORITAS OPERATOR**

Operator yang mempunyai prioritas tinggi akan diutamakan dalam hal pengerjaannya dibandingkan dengan operator yang lebih rendah prioritasnya.

OPERATOR	PRIORITAS
- (unary minus)	Tertinggi
* , / , %	
+ , -	Terendah

- Tanda kurung bisa digunakan untuk mengubah urutan pengerjaan.

Misal :

$$x = 2 + 3 * 10 ; \quad \rightarrow x = 32$$

$$x = (2 + 3) * 10 ; \quad \rightarrow x = 50$$

# OPERATOR – *Operator Penugasan*

- Operator penugasan yang berupa simbol sama dengan (=), digunakan untuk memberikan nilai ke suatu variabel.

- ❖ Penugasan sederhana, contoh :

$a = 3;$

$a = 2 + b;$

- ❖ Penugasan dalam ungkapan, contoh :

$a = 2 * (b = 5);$

- ❖ Penugasan berganda, contoh :

$a = b = c = d = 6;$

# OPERATOR – *Operator Penaikan dan Penurunan*

OPERATOR	CONTOH	KETERANGAN
++ (Operator Penaikan)	a++	A dinaikkan nilainya sebanyak 1 <u>setelah</u> dilakukan operasi pada A
	++a	A dinaikkan nilainya sebanyak 1 <u>sebelum</u> dilakukan operasi pada A
-- (Operator Penurunan)	a--	A diturunkan nilainya sebanyak 1 <u>setelah</u> dilakukan operasi pada A
	--a	A diturunkan nilainya sebanyak 1 <u>sebelum</u> dilakukan operasi pada A

# OPERATOR – *Operator Penaikan dan Penurunan*

Contoh :

1.     A = 5;  
       B = 10 + A++;  
       Hasil akhir :

A = 6  
B = 15

2.     A = 5;  
       B = 10 + ++A;  
       Hasil akhir :

A = 6  
B = 16



# OPERATOR – *Operator Relasional*

- Digunakan untuk membandingkan 2 buah nilai.

OPERATOR	KETERANGAN	CONTOH
==	Sama dengan	a == b
!=	Tidak sama dengan	a != b
>	Lebih dari	a > b
>=	Lebih dari atau sama dengan	a >= b
<	Kurangnya dari	a < b
<=	Kurang dari atau sama dengan	a <= b

- Contoh ungkapan kondisi :  $x == y$

Hasil ungkapan ini bernilai :

- ❖ 1 , jika nilai x sama dengan nilai y
- ❖ 0 , jika nilai x tidak sama dengan y

# OPERATOR – *Operator Logika*

- Digunakan untuk menghubungkan 2 buah ungkapan kondisi menjadi sebuah ungkapan kondisi.

OPERATOR	KETERANGAN	CONTOH
<b>&amp;&amp;</b>	<b>Logika DAN/AND</b>	<b>(x &gt; 0) &amp;&amp; (x &lt; 5)</b>
<b>  </b>	<b>Logika ATAU/OR</b>	<b>(x == 2)    (y &lt; 5)</b>
<b>!</b>	<b>Logika BUKAN/NOT</b>	<b>! (a == 1)</b>

# OPERATOR – *Operator Logika (con't)*

- Hasil ungkapan pada operator || dan && :

UNGKAPAN 1	UNGKAPAN 2	HASIL	HASIL &&
Benar	Benar	Benar	Benar
Benar	Salah	Benar	Salah
Salah	Benar	Benar	Salah
Salah	Salah	Salah	Salah

- Hasil ungkapan pada operator !

UNGKAPAN	HASIL !
Benar	Salah
Salah	Benar

# OPERATOR – *Operator Kondisi*

- Digunakan untuk mendapatkan sebuah nilai dari 2 buah kemungkinan, berdasarkan suatu kondisi.
- Format pemakaiannya :  
*Ungkapan1 ? Ungkapan2 : Ungkapan3*
- Hasil ungkapan diatas :
  - ❖ Nilai dari ungkapan adalah *Ungkapan2* jika *Ungkapan1* bernilai benar.
  - ❖ Nilai dari ungkapan adalah *Ungkapan3* jika *Ungkapan1* bernilai salah.

# OPERATOR – *Operator Kondisi (con't)*

- Contoh penggunaan operator kondisi :

besar = bil1 > bil2 ? bil1 : bil2 ;

*Ungkapan1    Ungkapan2    Ungkapan3*

Jika nilai variabel bil1 dan bil2 diinisialisasi dengan 60 dan 135, maka nilai variabel besar = 135

# OPERATOR – *Operator Bitwise*

- Digunakan untuk keperluan memanipulasi data dalam bentuk bit.

OPERATOR	KETERANGAN	CONTOH
<<	Geser bit ke kiri (left shift)	25 << 2
>>	Geser bit ke kanan (right shift)	25 >> 2
&	Bitwise AND (dan)	25 & 2
	Bitwise OR (atau)	25   2
^	Bitwise XOR	25 ^ 2
~	Bitwise NOT (komplemen)	~25

- Prioritas operator bitwise :

<b>PRIORITAS</b>	<b>OPERATOR</b>
Tertinggi	~
	>> <<
	&
	^
Terendah	

# OPERATOR – *Operator Bitwise (con't)*

## ✧ Operator Geser Kiri (<<)


Digunakan untuk menggeser bit-bit dalam suatu bilangan bulat ke kiri. Pada operasi penggeseran ke kiri (<<), setiap penggeseran sebuah bit akan memberikan pengaruh seperti halnya perkalian dengan bilangan 2.

Bentuk pemakaian operator << :

*nilai << jumlah bit digeser ke kiri*

Contoh :       $25 \ll 1$

maka :      0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 = 25

 Digeser ke kiri 1 bit

0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 = 50

Di bagian kanan selalu disisipi 0 (sebanyak bit yang digeser)

# OPERATOR – *Operator Bitwise (con't)*

## ✧ Operator Geser Kanan (>>)

Digunakan untuk menggeser bit-bit dalam suatu bilangan bulat ke kanan. Pada operasi penggeseran ke kanan (>>), setiap penggeseran sebuah bit akan memberikan pengaruh seperti halnya pembagian dengan bilangan 2.

Bentuk pemakaian operator >> :

*nilai >> jumlah bit digeser ke kanan*

Contoh :  $25 \gg 1$

maka : 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 = 25

 Digeser ke kanan 1 bit

0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 = 12



Di bagian kiri selalu disisipi dengan 0 (sebanyak bit yang digeser)



# OPERATOR – *Operator Bitwise (con't)*

## ✧ Operator ATAU (|)

Bentuk pemakaian operator | : *operand1* | *operand2*

Hasil operasi per bit :

Bit operand1	Bit operand2	Hasil
1	1	1
1	0	1
0	1	1
0	0	0

Contoh :

81 | 99

maka :

0000	0000	0101	0001	← 81
0000	0000	0110	0011	← 99
<hr/>				
0000	0000	0111	0011	← 115

## OPERATOR – *Operator Bitwise (con't)*

Bentuk pemakaian operator & : *operand1* & *operand2*

## Hasil operasi per bit :

Bit operand <sub>1</sub>	Bit operand <sub>2</sub>	Hasil &
1	1	1
1	0	0
0	1	0
0	0	0

## Contoh :

81 &amp; 99

maka :

0000 0000 010 1 0001

← 81

0000 0000 0110 0011

← 99

0000 0000 0100 000 1

&

← 65



# OPERATOR – *Operator Bitwise (con't)*

## ✧ Operator KOMPLEMEN (~)

Bentuk pemakaian operator :  $\sim \text{operand1}$

Hasil operasi per bit :

Bit	Operand
1	0
0	1

Contoh :

$\sim 81$

maka :

0000	0000	0101	0001		← 81
<hr/>					~
1111	1111	1010	1110		← 65454

# OPERATOR – *Operator Majemuk*

- Dimaksudkan untuk memendekkan penulisan operasi penugasan.

OPERATOR	CONTOH	KETERANGAN
+=	x += 2	Kependekan dari x = x + 2
-=	x -= 2	Kependekan dari x = x - 2
*=	x *= 2	Kependekan dari x = x * 2
/=	x /= 2	Kependekan dari x = x / 2
%=	x %= 2	Kependekan dari x = x % 2
<<=	x <<= 2	Kependekan dari x = x << 2
>>=	x >>= 2	Kependekan dari x = x >> 2
&=	x &= 2	Kependekan dari x = x & 2
=	x  = 2	Kependekan dari x = x   2
^=	x ^= 2	Kependekan dari x = x ^ 2

# KOMENTAR

- Komentar merupakan bagian dari sintaks program namun komentar tidak ikut di baca atau di eksekusi atau kata lain komentar tidak akan mempengaruhi jalannya program.
- Komentar sangat membantu programmer atau orang lain dalam memahami program, karena berupa penjelasan-penjelasan mengenai program/bagian-bagian program.
- Penjelasan bisa berupa :
  - Tujuan/fungsi program
  - Keterangan saat program dibuat/direvisi
  - Keterangan lain tentang kegunaan sejumlah pernyataan dalam program

# KOMENTAR (con't)

- Ada 2 jenis komentar dalam C++ :

1. **Menggunakan tanda //**

digunakan untuk menuliskan komentar yang terdiri dari 1 baris saja, gaya penulisan ini hanya di kenal di C++.

Contoh : `int a,b,c; //pendeklarasian variabel`

2. **Menggunakan tanda /\*...\*/**

digunakan untuk menuliskan komentar sebanyak satu baris atau lebih. komentar dimulai dengan tanda /\* dan akan berakhir sampai di temukan tanda \*/.

Contoh : `/*mengulang mencetak`

```
    pesan sebanyak 5 x */  
    for (i=1; i<=5; i++)  
        cout<<"SELAMAT";
```

# TUGAS

- Membuat makalah tentang manipulator dalam C++
- Dikerjakan secara individu
- Pengumpulan tugas yang melebihi batas waktu, **tidak diterima (dianggap tidak mengerjakan tugas)**