

Keg. Pembelajaran 3 : Operator dalam Bahasa C++

1. Tujuan Kegiatan Pembelajaran

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Mengetahui operator dalam bahasa pemrograman C++ secara baik.
- 2) Menjelaskan jenis-jenis operator dalam bahasa C++ secara benar.
- 3) Dapat menggunakan operator dalam bahasa C++ untuk memanipulasi data secara benar
- 4) Menghitung bilangan berdasarkan kenaikan dan penurunan secara tepat

2. Uraian Materi

Operator, dalam C++ merupakan simbol yang dilibatkan dalam program untuk melakukan operasi atau untuk memanipulasi, dalam hal : Penjumlahan dua buah nilai; Memberikan nilai ke variabel; Membandingkan kesamaan dua nilai.

Ditinjau berdasarkan sifatnya, operator dapat dibedakan menjadi tiga macam, yaitu :

- 1) unary → operator yang melibatkan sebuah operand
- 2) binary → operator yang melibatkan dua operand
- 3) ternary → operator yang melibatkan tiga operand Perhatikan :

✚ Penjelasan : Operator Unary

Operator unary adalah operator yang hanya melibatkan sebuah operand. Adapun yang termasuk dalam kategori operator unary ini adalah sebagai berikut:

Tabel 6. Operator unary

Operator	Jenis Operasi	Contoh
+	Membuat nilai positif	+7
-	Membuat nilai negatif	-7
++	Increment	C++
--	Decrement	C--

Operator Binary

Operator binary adalah operator yang digunakan dalam operasi yang melibatkan dua buah *operand*. Dalam bahasa C++, operator *binary* ini dikelompokkan menjadi empat jenis, yaitu operator aritmetika, logika, relasional dan bitwise.

1) Operator Aritmetika

Operator aritmatika adalah operator yang digunakan untuk melakukan operasi-operasi aritmetika. Adapun yang termasuk dalam operator aritmetika di dalam C++ adalah seperti berikut ini.

Tabel 7. Operator binary

Operator	Jenis Operasi	Contoh
+	Penjumlahan	$1 + 1 = 2$
-	Pengurangan	$2 - 1 = 1$
*	Perkalian	$1 * 1 = 1$
/	Pembagian	$9 / 3 = 3$
%	Sisa bagi (modulus)	$10 / 3 = 1$

Prioritas dalam penggunaan harus diperhatikan

Contoh : Menghitung Diskriminan

$$D = b^2 - 4 AC$$

Cara menulis di program :

$$D = b*b - 4 * A * C \text{ atau}$$

$$D = (b*b) - (4*A*C)$$

Penulisan Programnya menjadi :

```
# include <iostream.h> // tampilkan di layar
# include <conio.h> // melibatkan pembersihan layar
void main ( )
{
    int a, b, c, d;
    clrscr ( ); // digunakan untuk membersihkan layar →<conio.h>
    a = 5;
    b = 600;
    c = 8;
```

```

d = b*b - 4 * a * c;
cout << "Nilai d = " << d << '\n';
}

```

Gambar 11. Tampilan program menghitung diskriminan

2) Operator Logika

Operator logika adalah operator yang digunakan untuk melakukan operasi dimana nilai yang dihasilkan dari operasi tersebut hanya berupa nilai benar (*true*) dan salah (*false*). Adapun yang termasuk dalam operator logika adalah seperti berikut.

Tabel 8. Operator logika

Operator	Jenis Operasi	Contoh
&&	AND (dan)	1 && 1 = 1
	OR (atau)	1 0 = 1
!	NOT (negasi)	!0 = 1

3) Operator Relasional

Operator relasional adalah operator yang digunakan untuk menentukan relasi atau hubungan dari dua buah *operand*. Operator ini ditempatkan di dalam sebuah ekspresi, yang kemudian akan menentukan benar atau tidaknya sebuah ekspresi. Yang termasuk dalam kategori operasi relasional adalah sebagai berikut:

Tabel 9. Operator Relasi

Operator	Jenis Operasi	Contoh
>	Lebih besar	(3 > 2) = 1
<	Lebih kecil	(3 < 2) = 0
>=	Lebih besar atau sama dengan	(3 >= 3) = 1
<=	Lebih kecil atau sama dengan	(3 <= 2) = 0
==	Sama dengan	(5 == 2) = 0
!=	Tidak sama dengan	(5 != 2) = 1

4) Operator Bitwise

Operator yang digunakan untuk melakukan operasi-operasi yang berhubungan dengan manipulasi bit. Operator bitwise ini hanya dapat dilakukan pada operand yang bertipe char dan int saja karena ini berkoresponden dengan tipe byte atau word di dalam bit. Adapun yang termasuk dalam operator bitwise dalam C++ adalah sebagai berikut:

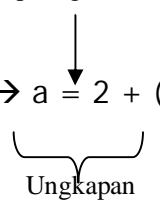
Tabel 10. Operator bitwise

Operator	Jenis Operasi	Contoh
&	AND	$1 \& 0 = 0$
	OR	$1 0 = 1$
^	EXCLUSIVE OR (XOR)	$1 \wedge 1 = 0$
~	NOT	$\sim 1 = 0$
>>	Shift Right	$16 \gg 1 = 8$
<<	Shift Left	$1 \ll 2 = 4$

5) Operator Penugasan (assignment)

Misal : $a = 5;$

$A = 2 + b;$

- Penugasan dalam ungkapan $\rightarrow a = 2 + (b+1)$

-
- Penugasan berganda $\rightarrow a = b = c = d = e = 1;$

6) Increment

Increment adalah suatu penambahan nilai yang terjadi pada sebuah variabel. Adapun yang digunakan untuk melakukan *increment* adalah operator $++$. Operator ini akan menambahkan nilai dari suatu variabel dengan nilai 1. Terdapat dua buah jenis *increment* yang

terdapat dalam bahasa C++, yaitu *pre-increment* dan *post-increment*. Arti *pre-increment* yaitu melakukan penambahan nilai sebelum suatu variabel itu diproses, sedangkan *post-increment* yaitu melakukan proses terlebih dahulu sebelum dilakukan penambahan nilai. Adapun bentuk umum dari *pre-increment* dan *post-increment* dapat dilihat pada Gambar12 di bawah ini.

```
// Melakukan pre-increment
++nama_variabel;

// Melakukan post-increment
Nama_variabel++;
```

Berikut diperlihatkan contoh program untuk melakukan pre-increment

```
#include <iostream>
void main()
{
    int C;           // Mendeklarsikan variabel C
                   // Mengisikan nilai ke dalam variabel C dengan nilai 5
    C = 5;

                   // Melakukan pre-increment
    cout<<"Nilai C awal      : "<<C<<endl;
    cout<<"Nilai ++C        : "<<++C<<endl;
    cout<<"Nilai C akhir    : "<<C<<endl;
    cout<<'\n';

                   // Mengubah nilai yang terdapat dalam variabel C dengan nilai 10
    C = 10;

                   // Melakukan post-increment
    cout<<"Nilai C awal      : "<<C<<endl;
    cout<<"Nilai C++        : "<<C++<<endl;
    cout<<"Nilai C akhir    : "<<C<<endl;
}
}
```

Gambar 12. Program pre-increment

Hasil dari program diatas adalah

<i>Pre increment</i>	Nilai C awal : 5	<i>Post-increment</i>	Nilai C awal : 10
	Nilai ++C : 6		Nilai C++ : 10
	Nilai C akhir : 6		Nilai C akhir : 11

7) Decrement

Decrement adalah suatu pengurangan nilai yang terjadi pada sebuah variabel. Adapun yang digunakan untuk melakukan *increment* adalah operator `--`. Operator ini akan mengurangi nilai dari suatu variabel dengan nilai 1. Terdapat dua buah jenis *decrement* yang terdapat dalam bahasa C++, yaitu *pre-increment* dan *post-increment*. Arti *pre-increment* yaitu melakukan pengurangan nilai sebelum suatu variabel itu diproses, sedangkan *post-increment* yaitu melakukan proses terlebih dahulu sebelum dilakukan pengurangan nilai. Adapun bentuk umum dari *pre-increment* dan *post-increment* dapat dilihat dibawah ini.

Contoh Program 2.

```
#include <iostream>
void main()
{
    int C;                // Mendeklarsikan variabel C
                        // Mengisikan nilai ke dalam variabel C dengan nilai 5
    C = 5;
                        // Melakukan pre-increment
    cout<<"Nilai C awal    : "<<C<<endl;
    cout<<"Nilai --C      : "<<--C<<endl;
    cout<<"Nilai C akhir   : "<<C<<endl;
    cout<<'\n';

                        // Mengubah nilai yang terdapat dalam variabel C
                        // dengan nilai 10
    C = 10;
                        // Melakukan post-decrement
    cout<<"Nilai C awal    : "<<C<<endl;
    cout<<"Nilai C--      : "<<C--<<endl;
    cout<<"Nilai C akhir   : "<<C<<endl;
}
```

Gambar 13. Program Decrement

Hasil dari program diatas adalah

<i>Pre increment</i>	Nilai C awal : 5	<i>Post-increment</i>	Nilai C awal : 10
	Nilai ++C : 4		Nilai C++ : 10
	Nilai C akhir : 4		Nilai C akhir : 9

8) Operator Majemuk

Berikut program increment dan decrement majemuk:

$X = X + 2 \rightarrow X += 2;$

$Y = Y * 4 \rightarrow Y *= 4;$

Contoh yang lain :

$-- \rightarrow Y -= 2$ identik dengan $Y = Y - 2;$

$/= \rightarrow X /= 2$ identik dengan $X = X / 2;$

$\% = \rightarrow Y \% = 2$ identik dengan $Y = Y \% 2;$

9) Manipulator Setw(n)

Manipulator setw (n), digunakan u/ mengatur lebar tampilan data.

Misalnya : bilangan 145, dengan mengatur *setw(6)*; \rightarrow dampaknya akan diperoleh enam ruang spasi:

1	2	3	4	5	6
.	.	.	1	4	5

Contoh program :

```
// Menampilkan 3 buah barang menggunakan manipulator setw(4)
# include <iostream.h>
# include <iomanip.h>> //lab untuk manipulasi
# include <conio.h>
void main ()
{
    int jumbar1 = 150,
    jumbar2 = 23,
    jumbar3 = 1406;
    clrscr();
    cout << "barang 1 = " << setw (4)<<jumbar1<<endl;
    cout << "barang 2 = " << setw (4)<<jumbar2<<endl;
    cout << "barang 3 = " << setw (4)<<jumbar3<<endl;
}
```

Setw (4)

	1	5	0
		2	3
1	4	0	6

Cataan :

Dalam hal ini tampak rata kanan

Jika nilai parameter lebar setw() lebih kecil dari jml kar (spasi), data yg ditampilkan tidak berpengaruh.

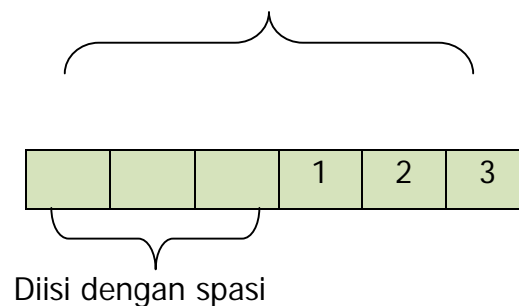
10) Manipulator Setfill ('*')

Manipulator ini digunakan untuk mengatur kar yang dipakai memenuhi bagian field yang ditentukan setw () tidak dipakai untuk menampilkan data.

Contoh :

```
cout << setw(6) << 123 <<endl;
```

Setw (6)

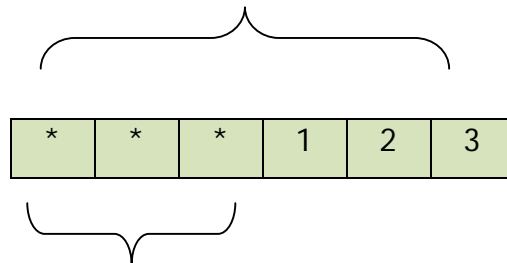


Penulisannya dalam program menjadi :

```
cout << setfill('*') ;
```

```
cout << setw(6) << 123 <<endl;
```


Setw (6)



Diisi dengan `setfill('*')`

11) Manipulator `dec`, `oct`, `hex`

Digunakan untuk menampilkan data dalam bentuk :

- a) desimal (bil. basis 10) → `dec`
- b) oktal (bil. basis 8), dan → `oct`
- c) hexadesimal (bil. Basis 16) → `hex`

Contoh Programnya :

```
// Mengkonversi ke dec, okt, hex*
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
void main ()
{
    int nilai = 250;
    clrscr();
    cout << nilai<<endl;
    cout << oct<< nilai<<endl;
    cout << hex<< nilai<<endl;
    cout << dec<< nilai<<endl;
}
```

250
372
F9
250

12) Manipulator `setbase ()`,

Manipulator `setbase ()` ini adalah sama dengan manipulator `dec`, `oct`, dan `hex`.

contoh :

```
cout << nilai<<endl;
cout << setbase (8) << nilai<<endl;
cout << setbase (10) << nilai<<endl;
cout << setbase (16)<< nilai<<endl;
```

13) Manipulator setprecision ()

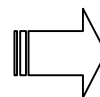
Manipulator ini digunakan untuk bekerja di bilangan pecahan.

Bentuk penulisannya :

```
setprecision ( n )
```

```
// contoh manipulator setprecision*
```

```
# include <iostream.h>
# include <iomanip.h>
# include <conio.h>
void main ()
{
    float nilai = 123.45;
    clrscr();
    cout << setprecision(0) << nilai<<endl;
    cout << setprecision(1) << nilai<<endl;
    cout << setprecision(2) << nilai<<endl;
    cout << setprecision(3) << nilai<<endl;
    cout << setprecision(5) << nilai<<endl;
}
```



123
123.4
123.45
123.450
123.45000

3. Rangkuman

Operator dalam bahasa pemrograman merupakan hal yang sangat penting dalam pemanfaatannya. Dengan memanipulasi tampilan data di layar akan menghasilkan tampilan data yang menarik. Untuk melakukan itu semua diperlukan apa yang disebut dengan operator. Dalam hal ini operator dapat dikelompokkan menjadi 3 jenis, yaitu unary, binary dan ternary.

Adapun jenis operator unary antara lain meliputi $+x$, $-x$, $++y$ dan $-z$. Yang termasuk operator binary, antara lain meliputi operator aritmatika, logika, relasi dan bitwise. Yang harus diperhatikan dalam penggunaan operator ini adalah masalah tingkatan prioritas. Untuk menghitungnya dimulai dari tingkatan yang tertinggi ke yang terendah. Di samping itu, ada operator untuk memanipulasi tampilan data, rata kanan, rata kiri, ketelitian dan konversi ke bilangan lain, misalnya ke basis 10 (decimal), basis 8 (octal), dan basis 16 (hexadecimal).

4. Latihan :

Tentukan ungkapan ini salah atau benar :

- 1) $(3+7)/2 < 6+7*4 \ || \ 9/3-2*1>6 \ \&\& \ 3<=8 \ \rightarrow T(1)$
- 2) $!((2+4<2)||((3+4) < (12/3+6))) \ || \ (4*3>=3) \ \&\& \ (3+8-2*3)>= 2$
- 3) $! (7*2/3-7>8/2*2\&\&3>2||!((3+3)<(8/2)) \ \&\& \ (7>=6))$