

[Home](#) / [Tutorial](#) / [C++](#) / [Operator](#)

Operator

Posted on Januari 22, 2018 | Last Modified Januari 22, 2018

Operators



Pengertian Operator

Operator adalah simbol atau karakter khusus yang digunakan untuk melakukan suatu operasi membantu compiler untuk membantu memanipulasi secara matematis atau logis pada data yang diberikan.

Bahasa pemrograman C++ menyediakan banyak operator yang sangat berguna pada penggunaannya. Dan karena C++ memiliki banyak operator pada artikel ini penulis akan membahas beberapa informasi tentang operator yang umum harus diketahui.

```
A + B;
```

Di atas adalah contoh bentuk dari operasi, dalam suatu operasi umumnya mempunyai istilah “operand” dan “operator”. operand merupakan data yang akan dimanipulasi oleh operator, dalam contoh di atas adalah A dan B, untuk operator merupakan simbol atau tanda yang

berfungsi untuk memanipulasi sebuah data atau beberapa data, dalam contoh di atas adalah tanda “+”.

Sifat-Sifat Operator C++

Dalam banyaknya macam-macam operasi kita biasa menggolongkannya berdasarkan sifatnya, hal ini akan mempermudah anda dalam membedakan sifat-sifat operator.

Unary : Sifat Unary pada operator adalah operator yang hanya melibatkan sebuah operand pada suatu operasi. Ada dua operator unary utama yaitu increment and decrement operator. Bentuk Penulisan :

```
- a;
```

Binary : Sifat Binary pada operator adalah operator yang melibatkan dua buah operand dan satu operator di tengah dua operand pada suatu operasi .

Bentuk Penulisan :

```
a + b;
```

Ternary : Sifat Ternary pada operator adalah operator kondisi (Conditional Operator) yang melibatkan tiga buah operand pada suatu operasi.

Bentuk Penulisan:

```
pernyataan: ekspresi?pilihan1:pilihan2;
```

Macam-macam Operator

Operator Aritmetika (*, /, %, +, -)

Operator untuk operasi aritmetika tergolong sebagai operator binary yang artinya operator-operator ini memerlukan dua operand, Operator aritmetika terdiri dari :

Operator	Keterangan	Contoh
*	Perkalian	a * b
/	Pembagian	a / b
%	Sisa pembagian (Modulo)	a % b
+	Penjumlahan	a + b
–	Pengurangan	a – b

Dan operator aritmetika juga mempunyai beberapa operator yang bersifat unary, yaitu :

Operator	Keterangan	Contoh
+	Penjumlahan	+a
–	Pengurangan	-a

Compound Assignment

(+=, -=, *=, /=, %=, >>=, <<=, &=, ^=, |=)

Compound Assignment jika di artikan ke dalam bahasa Indonesia adalah “Senyawa Tugas”. Tapi di Indonesia orang biasa menyebut sebagai “Pemberi Nilai Aritmetika”. Operasi ini mirip seperti Operator aritmetika hanya saja berbeda cara penulisanya (Lebih singkat).

Operator	Keterangan	Contoh
+=	Penambahan	X += Y
-=	Pengurangan	X -= Y
*=	Perkalian	X *= Y
/=	Pembagian	X /= Y
%=	Sisa Pembagian (Modulo)	X %= Y
>>=	Pergeseran bit ke kiri	X >>= Y
<<=	Pergeseran bit ke kanan	X <<= Y
&=	Bitwise DAN	X &= Y
^=	Bitwise OR Eksklusif	X ^= Y
=	Bitwise OR Inklusif	X = Y

Increment Dan Decrement Operator (++ , -)

Increment dan Decrement Operator atau Operasi Peningkatan dan Penurunan. Increment adalah peningkatan atau penambahan operator dengan tanda (++) berfungsi untuk meningkatkan nilai operand sebanyak "1". Dan Decrement adalah Penurunan atau pengurangan operator dengan tanda (--) berfungsi untuk menurunkan nilai operand sebanyak "1".

Tabel Operator Penambahan dan Pengurangan

Operator	Keterangan	Penjabaran
++	Increment / Peningkatan / Penambahan	$X = X + 1$
--	Decrement / Penurunan / Pengurangan	$X = X - 1$

Peletakan operator ini akan mempengaruhi fungsi dari operator tersebut. Dalam peletakan operator, Operator increment dan decrement dapat diletakan pada awal atau akhir variable, seperti dibawah ini :

++X (Prefix) adalah Nilai variabel X dinaikan dahulu sebelum diproses.

X++ (Postfix) adalah Nilai variabel X diproses dahulu sebelum dinaikan.

Relational and Comparison Operators (==, !=, <, >, <=, >=)

Dua Ekspresi dapat dibandingkan dengan menggunakan Relational and Comparison Operators. Misalnya, untuk menentukan apakah suatu variable memiliki nilai lebih besar atau lebih kecil dan lain-lain, hal itu dapat ditentukan dengan menggunakan operator Relational and Comparison Operators. dan Hasil operasi tersebut akan bernilai Benar (1) atau salah (0) yaitu merupakan bilangan Boolean.

Operator	Keterangan	Contoh
==	Sama dengan	$(a==a)$ atau $(a==5)$ atau $(5==5)$
!=	tidak sama dengan	$(a!=a)$ atau $(a!=5)$ atau $(5!=5)$
<	kurang dari	$(a<a)$ atau $(a<5)$ atau $(5<5)$
>	lebih besar dari	$(a>a)$ atau $(a>5)$ atau $(5>5)$
<=	kurang dari sama dengan	$(a<=a)$ atau $(a<=5)$ atau $(5<=5)$
>=	lebih dari sama dengan	$(a>=a)$ atau $(a>=5)$ atau $(5>=5)$

Logical Operator (&&, ||, !)

Logical operator biasanya digunakan untuk memeriksa kesamaan nilai dari dua data atau lebih dan juga dapat mengembalikan nilai dari bilangan boolean.

Asumsikan jika kita memiliki dua variable atau operand A dengan nilai 1 dan B dengan nilai 0, akan penulis coba demonstrasikan pada contoh di dalam table di bawah.

Operator	Keterangan	Contoh
&&	Disebut sebagai Logical AND operator. Jika kedua operand tidak memiliki nilai yang sama maka akan bernilai false	A && A = false
	Disebut sebagai Logical OR operator. Jika keduanya atau salah satu data bernilai true (1) maka akan menghasilkan nilai true (1)	A B = true
!	Disebut sebagai Logical NOT operator. Untuk mengembalikan nilai, jika A adalah true (1) maka akan bernilai false (0)	!(A && B) = true

Bitwise Operators (&, |, ^, ~, <<, >>)

Operator Bitwise adalah operasi matematika yang mengoperasikan pada level bilangan biner berbasis 2. Operator bitwise berfungsi untuk mengoperasikan sebuah data angka bilangan biasa yang akan diubah menjadi bilangan biner terlebih dahulu dan dioperasikan sesuai operasi yang diperintahkan, untuk hasil akhirnya akan diubah kembali menjadi bilangan biasa.

Operator	Asm equivalent	Keterangan
&	AND	Bitwise DAN
	OR	Bitwise OR Inklusif
^	XOR	Bitwise OR Eksklusif
~	NOT	Akan membalikan nilai
<<	SHL	Pergeseran Bit ke kiri
>>	SHR	Pergeseran Bit ke kanan

Conditional Ternary Operator (?)

Conditional ternary operator akan mengevaluasi ekspresi, menjadi memiliki "nilaiSatu" jika ekspresi yang mengevaluasi adalah benar(true) dan memiliki "nilaiDua" jika ekspresi bernilai sebagai false.

```
Ekspresi ? nilaiSatu : nilaiDua
```

Atau gampangnya

```
Ekspresi ? true : false
```


Comma Operator (,)

Comma Operator atau Operator koma (,) digunakan untuk memisahkan dua atau lebih ekspresi yang disertakan di mana hanya satu ekspresi yang diharapkan. Ketika set ekspresi harus dievaluasi untuk nilai, hanya ekspresi paling kanan dianggap.

Sebagai contoh :


```
a = (b=3, b+2);
```


pertama akan menetapkan nilai 3 untuk b dan kemudian menetapkan b+2 untuk variabel a. Jadi, pada akhirnya, variabel a akan berisi nilai 5 sedangkan variabel b akan berisi nilai 3. Dan juga biasanya operator koma digunakan untuk memisahkan argument pada function.

 Share

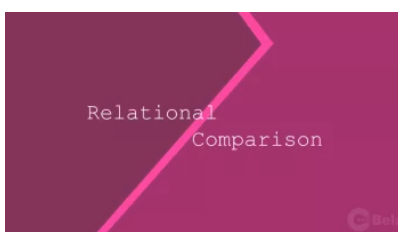
 Tweet

 Share

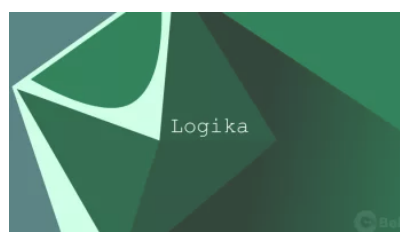
 Pin

 Share

Terkait



Operator Relasi dan Perbandingan
Januari 23, 2018
dalam "C++"



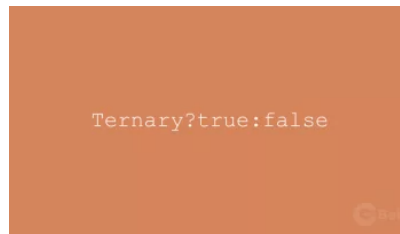
Operator Logika
Januari 23, 2018
dalam "C++"



Operator Peningkatan dan Penurunan
Januari 23, 2018
dalam "C++"



Operator Aritmetika
Januari 23, 2018
dalam "C++"



Ternary
Januari 23, 2018
dalam "C++"



Assignment Operator
Januari 22, 2018
dalam "C++"