

OPERASI DAN OPERATOR LOGIKA

Selain operator aritmatika yang sudah kalian pelajari pada pertemuan sebelumnya, terdapat juga operator logika. Operator logika digunakan dalam operasi-operasi logika, yaitu operasi yang menghasilkan nilai logika (*true* dan *false*). Untuk memperdalam wawasan mu tentang operator logika, mari kita mempelajari materi berikut.

1. Definisi Operasi Logika

Seperti yang sering kita dengar sebelumnya, dalam operasi aritmatika selalu berhubungan atau berkaitan dengan operasi logika. Logika adalah memberikan batasan yang pasti dari suatu keadaan, sehingga suatu keadaan tidak dapat berada dalam dua ketentuan sekaligus. Tahukah kamu apa yang dimaksud dengan operasi logika? Suatu fungsi logika atau operasi logika adalah suatu kombinasi variabel biner, *misalnya pada masukan dan keluaran dari suatu rangkaian digital yang mengikuti hukum aljabar Boolean*. Operasi logika merupakan konsep dasar yang digunakan oleh proses komputasi dalam mesin komputer. Operasi terhadap bilangan logika hanya dikenal dua nilai, yaitu *true* (1) dan *false* (0). Nilai ini disebut nilai *boolean*. *Boolean* ditemukan oleh seorang matematikawan Inggris bernama George Boole. Dalam aljabar *Boolean* semua hubungan logika antara variabel biner dapat dijelaskan oleh tiga operasi logika dasar, yaitu operasi NOT (*Negation*), operasi AND (*Conjunction*), dan operasi OR (*Disconjunction*).

Operasi-operasi logika dijelaskan dalam tiga bentuk antara lain sebagai berikut :

a. Tabel Fungsi (Tabel Kebenaran)

Tabel kebenaran menunjukkan keadaan semua variabel masukan dan keluaran untuk setiap kemungkinan.

b. Simbol Rangkaian

Simbol rangkaian digunakan untuk menjelaskan rangkaian digital.

c. Persamaan Fungsi

Persamaan fungsi digunakan untuk mengetahui operasi logika dengan menyamakan fungsi-fungsi.

2. Macam-Macam Operator Logika

Dalam suatu operasi aritmatika, logika maupun operasi lainnya pasti di dalamnya terdapat operator. Karena tanpa operator suatu operasi tidak ada fungsinya, sehingga tidak menghasilkan sebuah keluaran seperti yang kita harapkan.

Operator dasar yang ada dalam operasi logika dibagi menjadi tiga, yaitu sebagai berikut :

a. Operator AND (Conjunction)

Operasi AND hanya akan menghasilkan nilai 1 (*true*), jika semua *operand-operandnya* bernilai benar. Namun jika tidak, maka operasi tersebut akan menghasilkan nilai 0 (*false*). Operator AND disimbolkan dengan “&&”. Berikut ini tabel yang menunjukkan hasil dari operasi AND.

Tabel Kebenaran Operator AND

A	B	A&&B
1	1	1
1	0	0
0	0	0
0	1	0

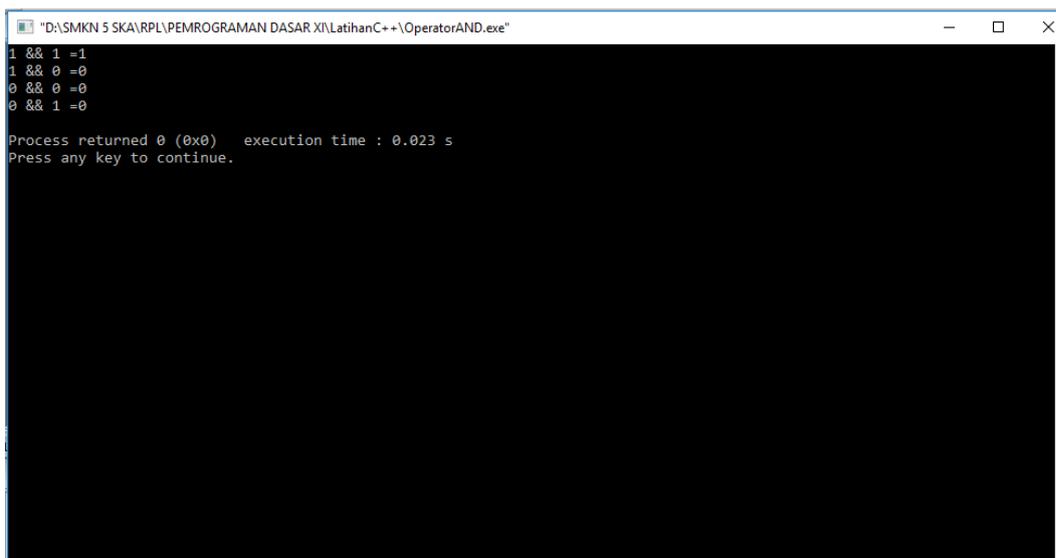
Untuk membuktikan pernyataan di atas dalam program C++, perhatikan contoh program berikut.

```
#include <iostream>
using namespace std;

int main(){
    //menampilkan operasi AND
    cout<<"1 && 1 ="<<(1 && 1)<<endl;
    cout<<"1 && 0 ="<<(1 && 0)<<endl;
    cout<<"0 && 0 ="<<(0 && 0)<<endl;
    cout<<"0 && 1 ="<<(0 && 1)<<endl;

    return 0;
}
```

Output dari program di atas adalah :



Kesimpulan yang diperoleh dari kedua bentuk (tabel kebenaran dan kode program) di atas bahwa dari tabel kebenaran yang di implementasikan dalam program C++ menghasilkan nilai atau keluaran yang sama. Operasi menghasilkan nilai 1, jika semua *operand* bernilai 1.

b. *Operator OR (Disconjunction)*

Operator OR hanya akan menghasilkan nilai 0 (*false*), jika semua *operand-operandnya* bernilai salah. Namun jika tidak, maka operasi tersebut akan menghasilkan nilai 1 (*true*). Operator OR disimbolkan dengan tanda "||". Berikut ini tabel yang menunjukkan hasil dari operasi OR.

Tabel Kebenaran Operator OR

A	B	A B
1	1	1
1	0	1
0	0	0
0	1	1

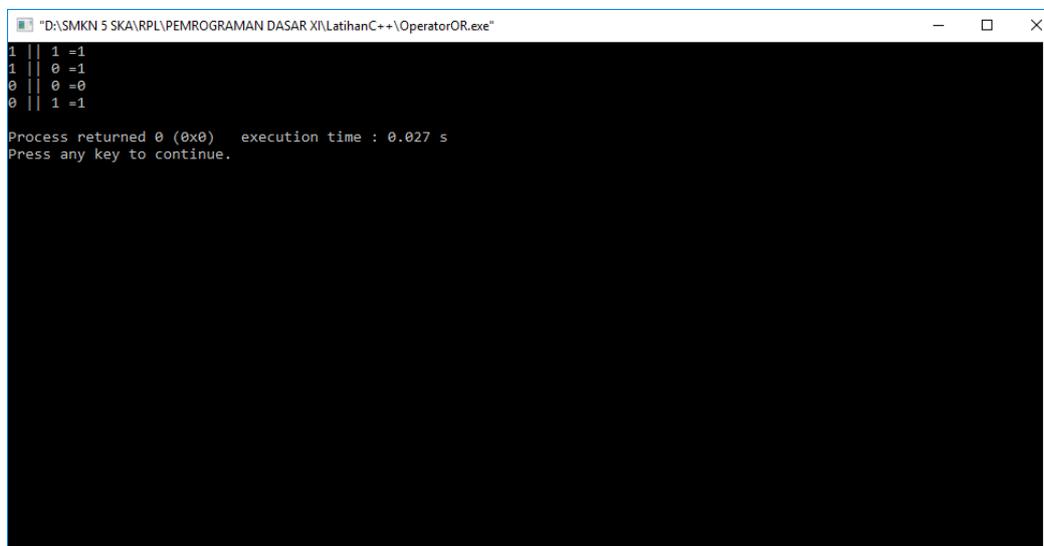
Untuk membuktikan pernyataan di atas dalam program C++, perhatikan contoh program berikut.

```
#include <iostream>
using namespace std;

int main(){
    //menampilkan operasi OR
    cout<<"1 || 1 ="<<(1 || 1)<<endl;
    cout<<"1 || 0 ="<<(1 || 0)<<endl;
    cout<<"0 || 0 ="<<(0 || 0)<<endl;
    cout<<"0 || 1 ="<<(0 || 1)<<endl;

    return 0;
}
```

Output dari program di atas adalah :



```
"D:\SMKN 5 SKA\RPL\PEMROGRAMAN DASAR XIN\LatihanC++\OperatorOR.exe"
1 || 1 =1
1 || 0 =1
0 || 0 =0
0 || 1 =1
Process returned 0 (0x0)   execution time : 0.027 s
Press any key to continue.
```

Kesimpulan yang diperoleh dari kedua bentuk (tabel kebenaran dan kode program) di atas bahwa dari tabel kebenaran yang di implementasikan dalam program C++ menghasilkan nilai atau keluaran yang sama. Operasi menghasilkan nilai 0, jika semua *operand* bernilai 0.

c. *Operator NOT (Negation)*

Nilai yang dihasilkan dari operasi NOT adalah kebalikan dari nilai yang terkandung di dalamnya. Jika nilai awal adalah 1 (*true*), maka setelah operasi NOT nialinya menjadi 0 (*false*). Begitu juga sebaliknya. Operator NOT disimbolkan dengan lambang "!". Berikut ini tabel yang menunjukkan hasil dari operasi NOT.

Tabel Kebenaran Operator NOT

X	!X
1	0
0	1

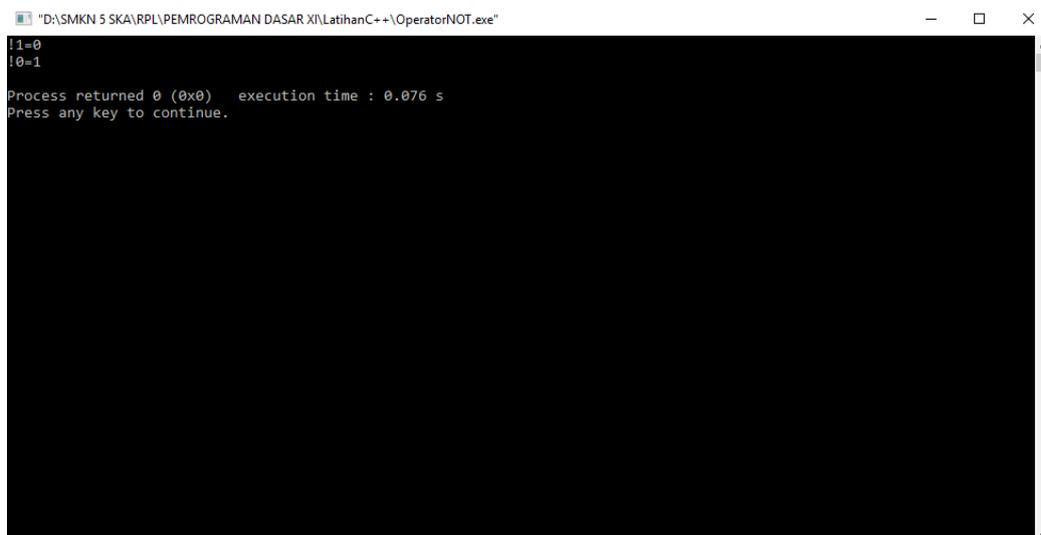
Untuk membuktikan pernyataan di atas dalam program C++, perhatikan contoh program berikut.

```
#include <iostream>
using namespace std;

int main(){
    //menampilkan operasi NOT
    cout<<"!1="<<(!1)<<endl;
    cout<<"!0="<<(!0)<<endl;

    return 0;
}
```

Output dari program di atas adalah :



The screenshot shows a terminal window titled "D:\SMKN 5 SKA\RPL\PEMROGRAMAN DASAR XI\LatihanC++\OperatorNOT.exe". The output displayed is:

```
!1=0
!0=1
Process returned 0 (0x0)   execution time : 0.076 s
Press any key to continue.
```

Kesimpulan yang diperoleh dari kedua bentuk (tabel kebenaran dan kode program) di atas bahwa dari tabel kebenaran yang di implementasikan dalam program C++ menghasilkan nilai atau keluaran yang sama. Nilai yang dihasilkan dari operasi NOT adalah kebalikan dari nilai yang terkandung di dalamnya. Jika nilai awal adalah 1 (*true*), maka setelah operasi NOT nilainya menjadi 0 (*false*).

TUGAS!

1. Perhatikan contoh program di bawah ini dengan teliti !

```
#include <iostream>
using namespace std;

int main(){
    int a=6, b=0;
    //menampilkan operasi AND
    cout<<"a && a ="<<(6 && 6)<<endl;
    cout<<"a && b ="<<(6 && 5)<<endl;
    cout<<"b && b ="<<(5 && 5)<<endl;
    cout<<"b && a ="<<(5 && 6)<<endl;

    return 0;
}
```

Coba praktikkan contoh program di atas ! Cermati !!!

Screen shoot kode program yang sudah kalian tulis di **codeblocks** kemudian masukkan ke dalam ms.word selanjutnya **Build and Run** pada program yang sudah kalian buat dan screen shot hasilnya pula, kemudian masukkan ke dalam ms.word juga !

2. Buatlah program sederhana, jika diketahui *output* sebagai berikut !

Masukkan bilangan ke -1=85

Masukkan bilangan ke-2=14

Hasil penjumlahannya yaitu =99

Hasil pengurangannya yaitu =71

Hasil perkaliannya yaitu = 1190

Hasil persentase dari pembagian yaitu =1

(Screen shoot kode program yang telah kalian buat beserta *output run program nya* ! Serta masukan ke dalam ms.word)

Hasil tugas dari setiap siswa berbentuk 1 file MS.Word dengan format **nama_no.absen.doc** kemudian kumpulkan kepada teman mu **Lambang Wahyu Mahardika** untuk dijadikan 1 folder **rar** dengan nama **TUGAS PEMROGRAMAN DASAR XI**

Kirim ke email : rina.smkn5ska@gmail.com

Pengiriman paling lambat Selasa, 14 November 2017. Pukul 17.00 WIB.