

MODUL

2

Operator

Modul **Praktikum**

C++

Dasar Pemrograman Komputer

Semester Genap 2017/2018

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK



MODUL 2 OPERATOR

A. Tujuan

Setelah mempelajari bab ini diharapkan mahasiswa akan mampu :

- Menjelaskan tentang jenis-jenis operator.
- Menjelaskan masing-masing fungsi operator.
- Memahami cara penggunaan masing-masing operator.
- Membuat program dengan mengaplikasikan operator.
- Mengevaluasi kesalahan pada program.

B. Dasar Teori

1. Pendahuluan

Operator merupakan simbol atau karakter yang biasa dilibatkan dalam program untuk melakukan sesuatu operasi atau manipulasi. Contohnya Penjumlahan, pengurangan, pembagian dan lain-lain.

Operator mempunyai sifat:

Unary

Sifat unary pada operator hanya melibatkan sebuah operand pada suatu operasi aritmatik. Contoh : -5

Binary

Sifat binary pada operator melibatkan dua buah operand pada suatu operasi aritmatik. Contoh : $4 + 8$

Ternary

Sifat ternary pada operator melibatkan tiga buah operand pada suatu operasi aritmatik. Contoh : $(10 \% 3) + 4 + 2$

2. Jenis-jenis operator

2.1 Operator Aritmatika

Operator untuk operasi aritmatika yang tergolong sebagai binary adalah:

Tabel 2.1.1 Operator Aritmatika

OPERATOR	KETERANGAN	CONTOH
*	Perkalian (Multiply)	4 * 5
/	Pembagian (Divide)	8 / 2
%	Sisa Bagi (Modulus)	5 % 2
+	Penjumlahan (Add)	7 + 2
-	Pengurangan (Substract)	6 - 2

Tabel 2.1.2 Operator Unary

OPERATOR	KETERANGAN	CONTOH
+	Tanda Plus	-4
-	Tanda Minus	+6

2.1.1 Hierarki Operator Aritmatika

Di dalam suatu akspresi arotmatika dapat kita jumpai beberapa operator aritmatika yang berbeda secara bersamaan. Urutan operator aritmatika adalah sebagai berikut:

Tabel 2.1.3 Tabel Hierarki Operator Aritmatika

Operator	Keterangan
* atau /	Tingkatkan operator sama, penggunaannya tergantung letak, yang di depan didahulukan
%	Sisa Pembagian
+ atau -	Tingkatkan operator sama, penggunaannya tergantung letak, yang di depan didahulukan

Contoh:

$$A = 8 + 2 * 3 / 6$$

Langkah perhitungannya :

$$A = 8 + 6/6 \text{ dimana } 6/6 = 1$$

$$A = 8 + 1$$

$$A = 9$$

Tingkatan operator ini dapat diabaikan dengan penggunaan tanda kurung (dan)

Contoh:

$$A = (8 + 2) * 3 / 6$$

Langkah perhitungannya :

$$A = 10 * 3/6 \text{ dimana } 8 + 2 = 10$$

$$A = 30 / 6$$

$$A = 5$$

2.2 Operator Penugasan

Operator penugasan (*Assignment operator*) dalam bahasa C++ berupa tanda sama dengan (“=”).

Contoh :

nilai = 80; A = x * y;

Penjelasan :

variable “nilai” diisi dengan 80 dan variable “A” diisi dengan hasil perkalian antara x dan y.

2.3 Operator Increment & Decrement

Pada pemrograman C++, menyediakan operator penambah dan pengurang (Increment & Decrement).

Tabel 1.4 Operator Increment & Decrement

Operator	Keterangan
++	Penambahan (Increment)
--	Pengurangan (Decrement)

A = A + 1 atau **A = A - 1**; dapat disederhanakan menjadi **A + = 1** atau **A - = 1**

Dan dapat disederhanakan menjadi **A++** atau **A--**. Notasi **++** atau **--** dapat diletakkan di depan atau di belakang variabel.

Contoh: A-- atau --A atau ++ A atau A++

Kedua bentuk penulisan di atas mempunyai arti yang berbeda:

- **Jika diletakkan di depan variabel.** Maka proses penambahan atau pengurangan akan dilakukan sesaat sebelum atau langsung pada saat menjumpai ekspresi ini sehingga nilai variabel tadi akan langsung berubah begitu ekspresi ini di temukan.

- **Jika diletakkan di belakang variabel.** Maka proses penambahan atau pengurangan akan dilakukan setelah ekspresi ini di jumpai atau nilai variabel akan tetap pada saat ekspresi ini ditemukan

2.4 Operasi Relasi (Perbandingan)

Operator relasi ini digunakan untuk membandingkan dua buah nilai. Hasil dari perbandingan operator ini menghasilkan nilai numeric **1 (True)** atau **0 (False)**.

Tabel 1.5. Operator relasi (Perbandingan)

Operator	Keterangan
==	Sama dengan (bukan pemberi nilai)
!=	Tidak sama dengan
>	Lebih dari
<	Kurang dari
>=	Lebih dari sama dengan
<=	Kurang dari sama dengan

2.5 Operator Logika

Operator logika digunakan untuk menghubungkan dua atau lebih ungkapan **menjadi sebuah ungkapan berkondisi**.

Tabel 1.6 Operator Relasi Logika

Operator	Keterangan	Contoh
&&	Operator Logika AND	m && n
 	Operator Logika OR	m n
!	Operator Logika NOT	!m

2.6 Operator Bitwise

Operator Bitwise digunakan untuk memanipulasi data dalam bentuk bit

Tabel 1.7 Operator Bitwise

Operator	Keterangan
<<	Bitwise Shift Left
>>	Bitwise Shift Right
&	Bitwise AND
^	Bitwise XOR
	Bitwise OR
~	Bitwise NOT

2.6.1 Operator Bitwise << (Shift Left)

Operator Bitwise Shift Left digunakan untuk menggeser sejumlah bit ke kiri

Contoh : 0001 0101 = 21 jika di geser kekiri 1 bit 0010 1 010 = 42

2.6.2 Operator Bitwise >> Shift Right

Operator Bitwise >> Shift Right digunakan untuk menggeser sejumlah bit ke kanan

Contoh : 0001 0010 = 18 jika digeser ke kanan 2 bit 0000 0100 = 4

2.6.3 Operator Bitwise & (AND)

Operator Bitwise & (AND) digunakan untuk membandingkan bit dari dua operand. Akan bernilai benar (1) jika semua operand yang digabungkan bernilai benar (1).

Tabel 1.8 Operator Bitwise & (AND)

Bit Operand 1	Bit Operand 1	Hasil Operand
0	0	0
0	1	0
1	0	0
1	1	1

Contoh :

0001 0101

0001 0001 = **AND**

0001 0001

2.6.4 Operator Bitwise | (OR)

Operator Bitwise | (OR) digunakan untuk membandingkan bit dari dua buah operand. Akan bernilai benar jika ada salah satu operand yang digabungkan ada yang bernilai benar (1).

Tabel 1.9 Operator Bitwise | (OR)

Bit Operand 1	Bit Operand 1	Hasil Operand
0	0	0
0	1	1
1	0	1
1	1	1

Contoh :

0001 0101

0001 0001 = **OR**

0001 0101

2.6.5 Operator Bitwise ^ (XOR)

Operator Bitwise ^ (XOR) digunakan untuk membandingkan bit dari dua buah operand. Akan bernilai benar (1) jika ada salah dua bit operand yang dibandingkan hanya sebuah bernilai benar (1).

Tabel 1.1 Operator Bitwise ^ (XOR)

Bit Operand 1	Bit Operand 1	Hasil Operand
0	0	0
0	1	1
1	0	1
1	1	0

Contoh :

0001 0101

0001 0001 = **XOR**

0000 0100

Latihan

1. Latihan 1

Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama **Lat1**, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program **Lat1**!

```
#include <conio.h>
#include <iostream.h>
int main()
{
    char a, b, c = 0, d = 0, e = 0, f = 0; //Deklarasi
    cout<<"Masukan Nilai A : ";
    cin>>A;
    cout<<"Masukan Nilai B : ";
    cin>>b;
    C = a * b;
    d = a \ B;
    e = A + b;
    f = a - b;
    cout<<"\nHasil dari C = A * B = "<<c<<endl;
    cout<<"Hasil dari C = A / B = "<<D<<endl;
    cout>>"Hasil dari C = A + B = "<<e<<endl;
    cout<<"Hasil dari C = A - B = ">>F<<endl;
    getch();
}
```

2. Latihan 2

Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama **Lat2**, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program **Lat2**!

```
#include <iostream.h>
int main ();
{
    float A,b,c,d,E,f;
    cout<<"Masukan Nilai A:";
    cin<<a;
    cout<<"Masukan Nilai B:";
    cin>>B;
    cout<<"Masukan Nilai C:";
    cin<<c;
    cout<<"Masukan Nilai D:";
    cin>>D;
    e=a+b*c/d-b;
    f=(a+b)*c/d*(a-b);
    cout<<"Nilai dari e=a+b*c/d-b: "<<F<<endl;
    cout>>"Nilai dari f=(a+b)*c/d: "<<f<<endl;
}
```

3. Latihan 3

Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama **Lat3**, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program **Lat3**!

```
#include <studio.h>
#include <conio.h>

int main() {
    int a=8,b=4,x=8,y=4;
    printf("Nilai A    = %d",a);
    printf("\nNilai ++A = %d",++a);
    printf("\nNilai A    = %d",a);
    printf("\nNilai B    = %d",b);
    printf("\nNilai --B = %d",--b);
    printf("\nNilai B    = %d",b);

    printf("\n\nNilai X    = %d",x);
    printf("\nNilai X++ = %d",x++);
    printf("\nNilai X    = %d",x);
    printf("\nNilai Y    = %d",y);
    printf("\nNilai Y-- = %d",y--);
    printf("\nNilai X    = %d",y);

    getch();
}
```

4. Latihan 4

Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama **Lat4**, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program **Lat4**!

```
#include <conio.h>
int main();
{
    int m=15, n=7;
    if(m<n)cout<<m<<" Lebih Besar Dari "<<n<<endl;
    else if (m<n)cout<<m<<" Lebih Kecil Dari "<<n<<endl;
    if(m=n)cout<<m<<" Sama Dengan "<<n<<endl;
    else if(m/=n)cout<<m<<" Tidak Sama Dengan "<<n<<endl;
    getch();
}
```

5. Latihan 5

- a. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama **Lat5a**, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program **Lat5a**!

```
#include <iostreamh>
int main()
{
    int p=255;
    cout<<"[p]>=0 && p>=255] -> "<<(P>=0 && p>=255)<<endl;
    cout<<"[p<=0 || p<=225] -> "<<(p<=0 || P>=225)<<endl;
    getch();
}
```

- b. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama **Lat5b**, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program **Lat5b**!

```
#include <stdioh>
#include <iostream.h>
int main()
{
    int a,B,c;//deklarasi variabel
    cout<<"Masukan Nilai A = ";//keluarkan string masukan nilai
    cin<<A;//masukan data integer dari keyboard
    /*Proses*/
    b=(a+12<8);
    C=! (b);
    cout<<"Program Ekspresi NOT">>end;
    Cout<<"Masukan Nilai = "<<a<<endl;
    cout<<"Nilai b = (a+12<8)= "<<B<<endl;
    cout<<"Nilai c = !(b)= ">>c<<endl;
}
```

6. Latihan 6

Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama **Lat6**, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program **Lat6**!

```
#include <iostream.h>
void main()
{
    int x,Y;
    cout<<"Masukan Nilai X=";
    Cin<<x;
    cout>>"Masukan Nilai Y=";
    cin>>Y;
    x=X>>3;//geser ke kiri 3 bit
    y=y<<2;//geser ke kanan 2 bit
    cout>>"Hasil dari Geser 3 Bit Ke Kiri="<<x<<endl;
    Cout<<"Hasil dari Geser 2 Bit Ke Kanan=">>y<<endl;
}
```

7. Latihan 7

Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama **Lat7**, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program **Lat7**!

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
void main()
{
    char a,b,C,x,y;
    cout>>"Masukan Nilai X=";
    cin>>X;
    cout<<"Masukan Nilai Y=":
    cin<<y;
    A=x&y;
    b=x||y;
    c=x^y;
    cout<<"\n";
    cout<<"Hasil dari "<<x<<"&"<<y<<"="<<a<<endl;
    cout>>"Hasil dari "<<x<<"|"<<y<<"="<<B<<endl;
    cout<<"Hasil dari "<<x<<"^"<<y<<"="<<E>>endl;
    getch();
}
```

Tugas Rumah

Buat suatu program yang menghasilkan data seperti di bawah ini !

```
Masukan nilai a : 2
Masukan nilai b : 5
Masukan nilai c : 1

Nilai dari d = (a*b)-c/(b+a) = 10
Hasil dari Geser 2 Bit Kekiri = 40

Masukan nilai j : 6
Masukan nilai k : 2
Masukan nilai l : 9

Nilai dari m = j*(k+l) = 66
Hasil dari Geser 3 Bit Kekananan = 8

40 Tidak Sama Dengan 8
40 Lebih Besar Dari 8

Terminated with return code 0
Press any key to continue ...
```