

MODUL 3 OPERATOR

A. TUJUAN

Setelah mempelajari bab ini diharapkan mahasiswa akan mampu:

- Menjelaskan tentang fungsi operator

B. PETUNJUK

1. Awali setiap aktivitas anda dengan doa, agar anda lancer dalam belajar.
2. Pahami tujuan, dasar teori dan latihan- latihan praktikum dengan baik.
3. Kerjakan tugas-tugas praktikum dengan baik, jujur dan sabar.
4. Tanyakan kepada asisten praktikum apabila ada hal-hala yang kurang jelas.

C. ALAT dan BAHAN

- PC/Laptop yang memenuhi kapasitas untuk pemrograman Bahasa C++
- Software C++
- Modul/handout materi yang akan di praktikkan.

D. DASAR TEORI

1. Pendahuluan

Operator merupakan simbol atau karakter yang biasa dilibatkan dalam program untuk melakukan sesuatu operasi atau manipulasi Contoh : 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. Operator Aritmatika

Operator untuk operasi aritmatika yang tergolong sebagai binary adalah

Tabel 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 1.2 Operator Unary

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

3. Operator Penugasan

Operator penugasan (*Assignment operator*) dalam Bahasa C++ berupa tanda sama dengan

("="). Contoh:

```
Nilai = 80;
```

```
A = x * y;
```

Penjelasan:

Variabel "nilai" diisi dengan 80 dan variabel "A" diisi dengan hasil perkalian antara x dan y.

4. Hierarki Operator Aritmatika

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

Tabel 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 :

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

$$A = 8 + 1$$

$$A = 9$$

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

Contoh:

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

Langkah perhitungannya :

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

$$A = 30 / 6$$

$$A = 5$$

5. Operator Increment dan 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.

6. Operasi Relasi (Perbandingan)

Operasi 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

7. 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

8. 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

9. Operator Bitwise << Shift Left

Operasi bitwise shift left digunakan untuk menggeser sejumlah bit ke kiri

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

10. 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

11. 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

12. Operator Bitwise | (OR)

Operator bitwise or digunakan untuk membandingkan bit dari dua buah operan. Akan bernilai benar jika 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

13. 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

E. LATIHAN

1. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama LATIH1, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda execute !


```
// Program Penggunaan Operator Aritmatika

#include <iostream.h>

#include <conio.h>

void main()

{

int a = 82, b = 26;

cout<<a<<" + "<<b<<" = "<<a+b<<endl;
cout<<a<<" - "<<b<<" = "<<a-b<<endl;
cout<<a<<" * "<<b<<" = "<<a*b<<endl;
cout<<a<<" / "<<b<<" = "<<a/b<<endl;
cout<<a<<" % "<<b<<" = "<<a%b<<endl;
getch();
}
```

2. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama LATIH2, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program LATIH2!

```
// Program Penggunaan Operator Aritmatika
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
void main()
{
float a, b, c = 0, d = 0, e = 0, f = 0; //Deklarasi
Variabel cout<<"Masukan Nilai A : ";
cin>>a;
cout<<"Masukan Nilai B : ";
cin>>b;
c = a* b;
d = a / b;
e = a + b;
```

3. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama LATIH3, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program LATIH3!

```

#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. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama LATIH4, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program LATIH4 !

```

#include <iostream.h>
#include <conio.h>

void main() {
    int m=5,n=7;
    if(m==n) cout<<m<<" Sama Dengan " <<n<<endl;
    else if(m!=n) cout<<m<<" Tidak sama dengan " <<n<<endl;
    else if(m>n) cout<<m<<" Lebih besar dari " <<n<<endl;
    else if(m<n) cout<<m<<" Lebih besar dari " <<n<<endl;
    getch();
}

```

5. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama LATIH5, execute

kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program LATIH5 !

```
//Program Operator Logika
#include <iostream.h>

#include <conio.h>

int main()
{
    int m = 166;

    cout<<"(m)>=0 && m<=150) -> "<<(m)>=0 & m<=150)<<endl;
    cout<<"(m)>=0 || m<=150) -> "<<(m)>=0 || m<=150)<<endl;
    getch();
}
```

6. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama LATIH6, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program LATIH6 !

```
#include <stdio.h>
#include <iostream.h>

void 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 + 4 < 10);
    c = !(b);
    cout<<endl<<"Program Ekspresi NOT "<<endl;
    cout>>"Masukan Nilai = "<<a<<endl;
    cout<<"Nilai b = (a + 4 < 10) = "<<b<<endl;
    cout<<"Nilai c = !(b) = "<<c;
    getch();
}
```

7. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama LATIH7, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program LATIH7 !

```
#include <iostream.h>
void main()
{
    int x, y;
    cout<<"Masukan Nilai X = ";
    cin>>x;
    cout<<"Masukan Nilai Y = ";
    cin>>y;
    x = x << 1; //geser ke kiri 1 bit
    y = y >> 2; //geser ke kanan 2 bit
    cout<<"Hasil dari Geser 1 Bit Kekiri = "<<Y<<endl;
    cout<<"Hasil dari Geser 2 Bit Kekanan = "<<X<<endl;
    getch();
}
```

8. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama LATIH8, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program LATIH8 !

JOBSHREE

```

#include <stdio.h>
#include <conio.h>
#include <iostream.h>
void main()
{
    int 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<<" = "<<c<<endl;
    getch();
}

```

F. TUGAS PRAKTIKUM

1. TGP1

Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama TGP1, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program TGP1 !

```

// Penggunaan Notasi di Depan Variabel
#include <stdio.h>
#include <conio.h>
void main()
{
    int a = 8, b = 4;
    printf(" Nilai A = %d", a);
    printf("\n Nilai ++A = %d", ++a);
    printf("\n Nilai A =%d", a);
    printf("\n Nilai B=%d", b);
    printf("\n Nilai --B =%d", --b);
    printf("\n Nilai B=%d", b);
    getch();
}

```

2. TGP2

Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama TGP2, execute kode program tersebut., jika ada error perbaiki program tersebut.

Jelaskan & simpulkan hasil execute program TGP2 !

```
// Penggunaan Notasi di Belakang Variabel
#include <stdio.h>
#include <conio.h>
void main()
{
int a = 8, b = 4;
printf(" Nilai A   = %d", a);
printf("\n Nilai A++ = %d", a++);
printf("\n Nilai A   =%d", a);
printf("\n Nilai B=%d", b);
printf("\n Nilai B-- =%d", B--):
printf("\n Nilai B=%d", b);
getch();
}
```

G. TUGAS RUMAH

```
D:\Z. ASISTEN PRAKTIKUM TE UM\ASISTEN PEMROGRAMAN KOMPUTER 2016\...
Masukan nilai a : 8
Masukan nilai b : 9
Masukan nilai c : 10

Nilai dari d = <a*b>-c/<b+a> = 72
Hasil dari Geser 2 Bit Kekiri = 288

Masukan nilai j : 18
Masukan nilai k : 20
Masukan nilai l : 11

Nilai dari m = j*<k+1> = 558
Hasil dari Geser 3 Bit Kekanannya = 69

288 Tidak Sama Dengan 69
288 Lebih Besar Dari 69
```