

MODUL 2

OPERATOR DAN OPERASI KONDISI

A. Tujuan Praktikum :

Setelah mempelajari bab ini mahasiswa diharapkan mampu :

1. Menjelaskan tentang fungsi operator
2. Menjelaskan tentang fungsi operasi kondisi

B. Waktu Praktikum

4 X 1 jam

C. Petunjuk

1. Awali setiap kegiatan dengan berdoa, agar diberi kelanjutan dalam belajar dan mendapatkan ilmu yang bermanfaat.
2. Pahami tujuan praktikum, dasar teori, dan latihan – latihan praktikum dengan baik
3. Kerjakan tugas praktikum dengan baik, jujur, sabar dan teliti
4. Tanyakan kepada asisten laboratorium apabila ada hal-hal yang kurang jelas.

D. Dasar Toeri

Operator

Operator merupakan simbol atau karakter yang biasa dilibatkan dalam program untuk melakukan sesuatu operasi atau manipulasi

Contoh :

Penjumlahan, pengurangan, pembagian dan lain-lain

Sifat-sifat operator :

1. Unary

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

2. Binary

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

3. Ternary

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

Jenis-jenis operator:

1. Operator aritmetika

Macam – macam operator aritmetika adalah sebagai berikut :

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

Di dalam suatu akspresi arotmatika dapat kita jumpai beberapa operator aritmatika yang berbeda secara bersamaan. Ada hierarki (tingkatan) dalam penulisan operator aritmetika yakni :

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 :

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

$$A = 30 / 6$$

$$A = 5$$

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.

3. Operator Increment & Decrement

Pada pemrograman C++, menyediakan operator penambah(increment) dan pengurang (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 **dilakukan sesaat sebelum atau langsung** pada saat menjumpai ekspresi ini sehingga nilai **variabel 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

4. Operator 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

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	$!m$

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

a. Operator 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

b. Operator 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

c. Operator 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 :

21 & 17 = 17

0001 0101

AND

0001 0001 =

0001 0001

d. Operator 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 :

$$\begin{array}{r}
 21 \mid 17 = 21 \\
 0001\ 0101 \\
 \text{OR} \\
 \underline{0001\ 0001} = \\
 0001\ 0101
 \end{array}$$

e. Operator 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 :

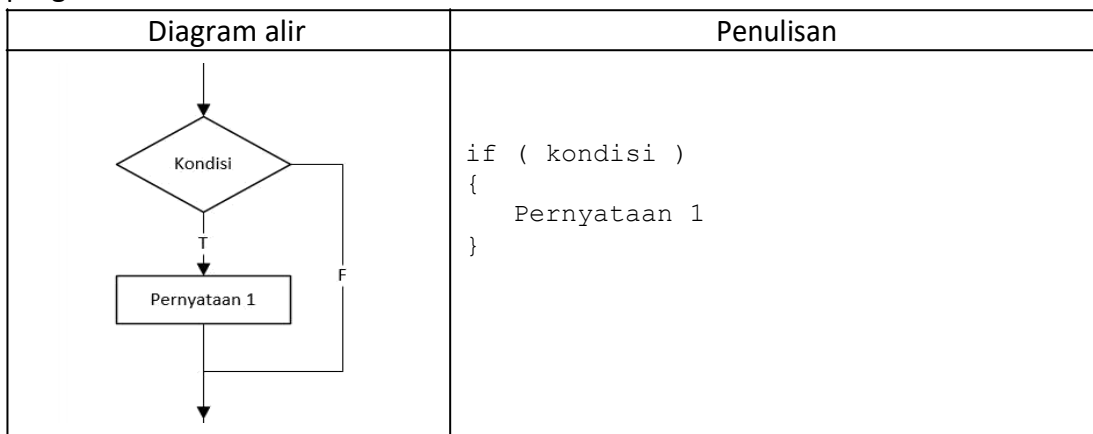
$$\begin{array}{r}
 21 \mid 17 = 4 \\
 0001\ 0101 \\
 \text{OR} \\
 \underline{0001\ 0001} = \\
 0000\ 0100
 \end{array}$$

Operasi Kondisi

Pernyataan percabangan pada pemrograman digunakan untuk memecahkan persoalan untuk mengambil suatu keputusan diantara sekian pernyataan yang ada, dimana sebuah instruksi (pernyataan) akan dilaksanakan jika sebuah kondisi/persyaratan terpenuhi.

1. Pernyataan If

Pernyataan if mempunyai pengertian” jika kondisi bernilai benar, maka (pernyataan) perintah akan dikerjakan dan jika tidak memenuhi syarat maka program akan diabaikan”



2. Pernyataan If Else

Pernyataan if mempunyai pengertian “ jika kondisi bernilai benar, maka pernyataan1 akan dikerjakan dan jika tidak memenuhi syarat maka akan mengerjakan pernyataan2”. Dari pengertian tersebut dapat dilihat di diagram alir pada gambar 2. diagram alir if else

Diagram alir	Penulisan
<pre> graph TD Start(()) --> Kondisi{Kondisi} Kondisi -- T --> Pernyataan1[Pernyataan 1] Kondisi -- F --> Pernyataan2[Pernyataan 2] Pernyataan1 --> Exit(()) Pernyataan2 --> Exit </pre>	<pre> if (kondisi) { Pernyataan 1 } else { Pernyataan 2 } </pre>

3. Pernyataan NESTED If (If bersarang)

Nested if merupakan pernyataan yang berada di dalam pernyataan yang lain.

Diagram alir	Penulisan
<pre> graph TD Start(()) --> K1{Kondisi 1} K1 -- T --> K2{Kondisi 2} K1 -- F --> K3{Kondisi 3} K2 -- T --> P1[Pernyataan 1] K2 -- F --> P2[Pernyataan 2] K3 -- T --> P3[Pernyataan 3] K3 -- F --> P4[Pernyataan 4] P1 --> Exit(()) P2 --> Exit P3 --> Exit P4 --> Exit </pre>	<pre> if (kondisi 1) { if (kondisi 2) { Pernyataan 1 } else { Pernyataan 2 } } else { if (kondisi 3) { Pernyataan 3 } else { Pernyataan 4 } } </pre>

4. Pernyataan If Majemuk

Bentuk dari if-else bertingkat sebenarnya mirip dengan nested if. Keuntungan penggunaan if-else bertingkat dibanding dengan nested if adalah bentuk penulisan yang lebih sederhana

Diagram alir	Penulisan
<pre> graph TD Start(()) --> K1{Kondisi 1} K1 --> P1[Pernyataan 1] K1 --> K2{Kondisi 2} K2 --> P2[Pernyataan 2] K2 --> K3{Kondisi 3} K3 --> P3[Pernyataan 3] K3 --> P4[Pernyataan 4] P1 --> Exit(()) P2 --> Exit P3 --> Exit P4 --> Exit </pre>	<pre> if (kondisi 1) { Pernyataan 1 } else if (kondisi 2) { Pernyataan 2 } else if (kondisi 3) { Pernyataan 3 } else { Pernyataan 4 } </pre>

5. Pernyataan Switch Case

Bentuk switch – case merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah atau banyak alternative. Pernyataan switch – case ini memiliki kegunaan yang sama seperti if – else bertingkat, tetapi untuk memeriksa data yang bertipe karakter atau integer.

Penulisan

```

Switch ( Ekspresi )
{
case Konstata 1:
    pernyataan 1;
    break;
case Konstata 2:
    pernyataan 2;
    break;
case Konstata 3:
    pernyataan 3;
    break;
default:
    pernyataan lain;
}

```

E. Latihan Praktikum

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

```
// Program Penggunaan Operator Aritmatika
#include <iostream.h>
#include <conio.h>
void main()
{
    int a = 82, b = 26 c = 10;
    cout<<"-a = "<<-a<<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;
    cout<<a<<" % "<<b<<" = "<<a%b<<endl;
    cout<<a<<" - "<<b<<" * "<<c<<" = "<<a-b*c<<endl;
    cout<<" ("<<a<<" + "<<b<<") / "<<c<<" = "<<(a+b)/c<<endl;
    getch();
}
```

2. Tuliskan kode program di bawah ini dengan menggunakan CodeBlocks, kemudian compile dan simpan dengan nama LATIH2, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

```
// Penggunaan Notasi di Depan Variabel
#include <stdio.h>
#include <conio.h>
void main()
{
    int a = 10, b = 6, c = 4, d = 8;
    printf("Penggunaan notasi di depan variabel \n")
    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);
    printf("Penggunaan notasi di depan variabel \n")
    printf(" Nilai C   = %d", c);
    printf("\n Nilai C++ = %d", c++);
    printf("\n Nilai C   = %d", c);
    printf("\n Nilai D   = %d", d);
    printf("\n Nilai D-- = %d", d--);
    printf("\n Nilai D   = %d", d);
    getch();
}
```


3. Tuliskan kode program di bawah ini dengan menggunakan CodeBlocks, kemudian compile dan simpan dengan nama LATIH3, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

```
// Penggunaan operator relasi dan if, if else
#include <iostream.h>
void main()
{
    int m , n ;
    cout<<"Masukkan nilai m : ";
    cin>>m;
    cout<<"/n"
    cout<<"Masukkan nilai n : ";
    cout>>n;
    cout<<"/n"
    if (m == n) cout<<m<<" Sama Dengan "<<n<<endl;
    else if (m != n) cout<<m<<" Tidak Sama Dengan "<<n<<endl;
    cout<<"/n"
    if (m > n) cout<<m<<" Lebih Besar Dari "<<n<<endl;
    else if (m < n) cout<<m<<" Lebih Kecil Dari "<<n<<endl;
}
}
```

4. Tuliskan kode program di bawah ini dengan menggunakan CodeBlocks, kemudian compile dan simpan dengan nama LATIH4, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

```
/* Penggunaan Operasi Logika NOT */
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
void main()
    int a, b, c;//deklarasi Variabel
    cout<<"Masukan Nilai a = ";
    //Keluarkan String Masukan Nilai A = ke Layar
    cin>>a;//Masukan Data integer dari keyboard
    /* Proses */
    b = (a + 4 < 10);
    c = !(b);
    cout<<endl<<"Program Ekspresi NOT "<<endl;
    cout<<"Nilai a = "<<a<<endl;
    cout<<"Nilai b = (a + 4 < 10) = "<<b<<endl;
    cout<<"Nilai c = !(b) = "<<c;
    getch();
}
```

5. Tuliskan kode program di bawah ini dengan menggunakan CodeBlocks, kemudian compile dan simpan dengan nama LATIH5, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
int main()
{
    int x, a, b, c, d, e, y;
    cout<<"Masukan Nilai X = ";
    cin>>x;
    a=x;
    b=x;
    a = a << 1; //geser ke kiri 1 bit
    cout<<"\n Hasil dari Geser 1 Bit Kekiri = "<<x<<endl;
    b = b >> 1; //geser ke kanan 1 bit
    cout<<"\n Hasil dari Geser 1 Bit Kekiri = "<<x<<endl;
    cout<<"Masukan Nilai y = ";
    cin>>y;
    c = x & y;
    d = x | y;
    e = x ^ y;
    cout<<"\n Hasil dari X Bitwise AND Y = "<<c<<endl;
    cout<<"\n Hasil dari X Bitwise OR Y = "<<d<<endl;
    cout<<"\n Hasil dari X Bitwise XOR Y = "<<e<<endl;
    getch();
}
```

6. Tuliskan kode program di bawah ini dengan menggunakan CodeBlocks, kemudian compile dan simpan dengan nama LATIH6, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
int main()
{
    char x;
    int y;
    cout<<"Apakah kamu laki - laki (Y/N)?? ";
    cin>>x;
}
```

```
if (x == 'y' | 'Y')
{
    cout<<"Berapakah usia kamu ?";
    cin>> y;
    if (y <= 20)
        cout<<"Hallo Mas Bro";
    else
        cout<<"Selamat siang Bapak";
}
else
{
    cout<<"Berapakah usia kamu ?";
    cin>> y;
    if (y <= 20)
        cout<<"Hallo Mbak Bro";
    else
        cout<<"Selamat siang Bu";
}

getch();
}
```

7. Tuliskan kode program di bawah ini dengan menggunakan CodeBlocks, kemudian compile dan simpan dengan nama LATIH7, jalankan kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil program yang telah anda jalankan !

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
int main()
{
    int x, y, a, z;
    cout<<"Masukan Nilai X = ";
    cin>>x;
    cout<<"Masukan Nilai y = ";
    cin>>y;
    cout<<"Operasi yang dapat dilakukan \n";
    cout<<"1. Penjumlahan \n";
    cout<<"2. Pembagian \n";
    cout<<"3. Pengurangan \n";
    cout<<"4. perkalian \n";
    cout<<"Masukan pilihan operasi = ";
    cin>>a;
    switch (a)
    case 1 :
        z = x+y;
        cout<<"Nilai x + y = "<<z;
        break;
```

```
case 2 :
    z = x/y;
    cout<<"Nilai x : y = "<<z;
    break;
case 3 :
    z = x-y;
    cout<<"Nilai x - y = "<<z;
    break;
case 4 :
    z = x*y;
    cout<<"Nilai x * y = "<<z;
    break;
default :
    cout<<"Pilihan tidak diketahui"
    getch();
}
```

F. Tugas Praktikum

1. Buatlah sebuah program untuk menghitung tagihan biaya listrik untuk satu bulan dengan ketentuan sebagai berikut :

- a. Memasukkan besar pemakaian kwh yang digunakan.
- b. Melakukan konversi nilai angka pemakaian listrik ke nilai huruf untuk menentukan golongan pengguna dengan ketentuan sebagai berikut :
 - 1) Pemakaian 0 – 100 kwh = golongan C
 - 2) Pemakaian 100 – 200 kwh = golongan B
 - 3) Pemakaian 200 – 300 kwh = golongan A

Lalu Melakukan perhitungan tagihan yang harus dibayar dengan ketentuan sebagai berikut :

G. Golongan A

Abonemen = 50000
Harga per kwh = 2250/kwh
Pajak = 15% dari biaya

H. Golongan B

Abonemen = 35000
Harga per kwh = 1750/kwh
Pajak = 10% dari biaya

I. Golongan C

Abonemen = 20000
Harga per kwh = 1500/kwh
Pajak = 5% dari biaya

Biaya = Abonemen + (harga per kwh*pemakaian)

Total = biaya + pajak

2. Buatlah diagram alir dan program dengan menggunakan logika percabangan Switch Case untuk melakukan pemilihan menu penghitungan rumus-rumus :

- 1 Rumus Menghitung Volume Bola;
- 2 Rumus Menghitung Volume Prisma;
- 3 Rumus Menghitung konversi suhu;

4 Rumus Menghitung Volume Tabung.

3. Sebuah toko bangunan menjual marmer dengan ketentuan sebagai berikut :
- a. Jika kode marmer=1 maka jenis marmer = oca, dengan ukuran panjang sisi marmer =40 m, maka harganya 200.000/m², Jika ukuran marmer=50 m, maka harganya 220.000/m², selain itu harganya = 250.000/m².
 - b. Jika kode marmer =2 maka jenis marmer = pearl, dengan ukuran marmer=40 m, maka harganya 150.000, Jika ukuran Marmer=50 m, maka harganya 160.000, selain itu harganya = 170.000.
 - c. Jika kode marmer =3 maka jenis marmer = shade, dengan ukuran marmer=40 m maka harganya 200.000, Jika ukuran Marmer=50 m, maka harganya 200.000, selain itu harganya = 200.000.
 - d. Jika kode marmer =4 maka jenis marmer = laterite, dengan ukuran marmer=40 m , maka harganya 300.000, Jika ukuran Marmer=50 m, maka harganya 300.000, selain itu harganya = 350.000.
 - e. Jika kode marmer =5 maka jenis marmer = coal, dengan ukuran marmer=40 m, maka harganya 100.000, Jika ukuran Marmer=50 m, maka harganya 120.000, selain itu harganya = 130.000.
 - f. Selain kode marmer diatas, maka salah kode