



**MODUL #07**

**PEMROGRAMAN TERSTRUKTUR**

# OPERATOR PADA C++

NUR ALAMSYAH

# PENGERTIAN OPERATOR

**O**perator adalah simbol atau karakter khusus yang digunakan untuk melakukan suatu operasi membantu **kompilator** untuk memanipulasi secara matematis atau logis pada data yang diberikan.

Bahasa pemrograman C++ menyediakan banyak operator yang sangat berguna untuk melakukan operasi. pada artikel ini penulis akan membahas beberapa informasi tentang operator yang umum harus diketahui.

```
1 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 merupakan **A** dan **B**,

**operator** merupakan simbol atau tanda yang berfungsi untuk memanipulasi sebuah data atau beberapa data, dalam contoh di atas adalah tanda **+**.

Jadi **Operator** dapat dikatakan sebagai tanda yang digunakan untuk melakukan operasi-operasi tertentu didalam program. dengan operator, kita dapat melakukan operasi perhitungan, perbandingan, manipulasi bit, dan lain-lain.

Terdapat bagian - bagian yang harus diketahui dalam operator .

Contoh apabila terdapat statement seperti dibawah ini :

```
x = 7 + 9;
```

---

keterangan :

**x** disebut sebagai **variabel**

**=** disebut sebagai **operator assignment**

**7** dan **9** disebut **operand**

**7 + 9** disebut **ekspresi**

**+** disebut sebagai **operator aritmatika (penjumlahan)**

**x = 7 + 9** disebut **statement aritmatika**

# Sifat Operator Dalam C++

Dalam banyaknya macam-macam operator kita biasa menggolongkannya berdasarkan sifatnya, berikut sifat-sifat dari operator:

- **Unary :** Sifat *Unary* adalah operator yang hanya melibatkan sebuah *operand* pada suatu operasi. Ada dua operator *unary* utama yaitu *increment* dan *decrement operator*.

Bentuk Penulisan

```
1 -a;
```

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

Bentuk Penulisan

```
1 a + b;
```

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

Bentuk Penulisan

```
1 pernyataan: expresi?pilhan1:pilihan2;
```

## Kelompok Operator Dalam C++

Operator pada C++ dapat dibagi menjadi 4 (empat) kelompok , yaitu :

1. OPERATOR ASSIGNMENT
2. OPERATOR UNARY
3. OPERATOR BINARY
4. OPERATOR TERNARY

**PENJELASAN!**

## 1. OPERATOR ASSIGNMENT

Operator Assignment (pengisian) merupakan operator yang digunakan untuk memasukkan atau mengisi nilai ke dalam variabel . Dalam C++ , operator yang digunakan untuk keperluan ini adalah operator = (**sama dengan**).

Contoh :

**int a , b;**

**a = 70;**

**b = 50;**

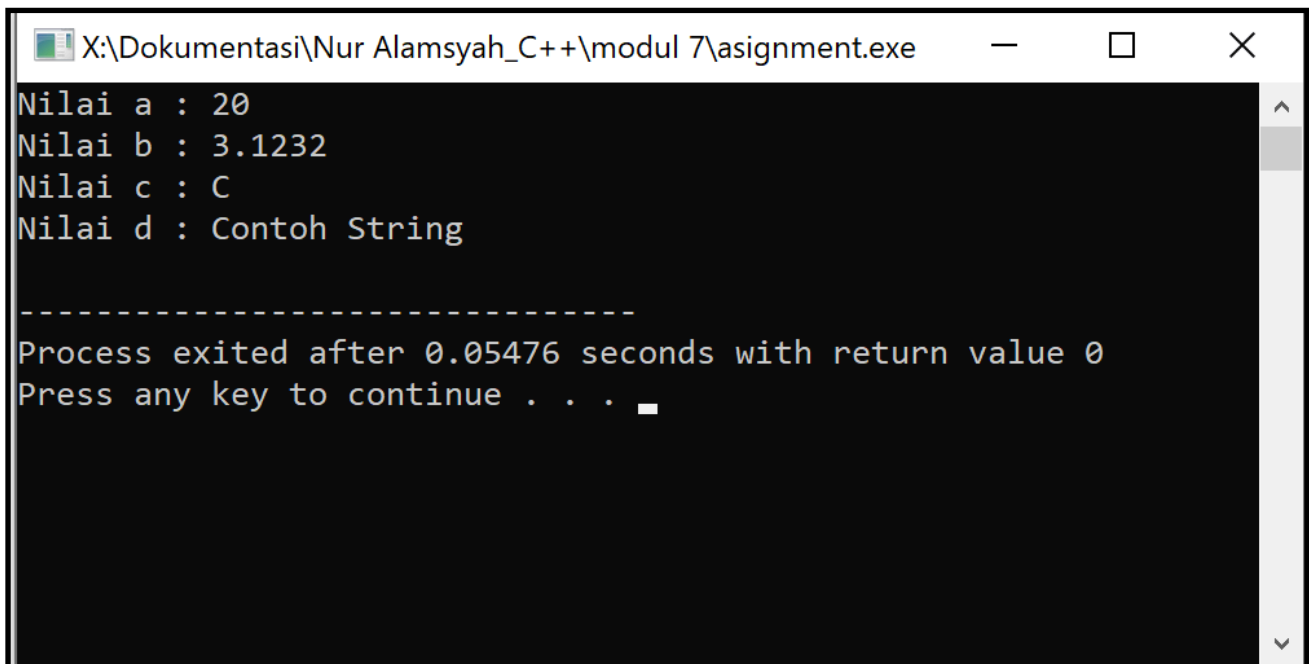
Pada contoh diatas , kita memasukkan atau memberi nilai pada variabel a dan b , nilai 70 untuk variabel a dan nilai 50 untuk variabel b.

Operator = (sama dengan) dapat digunakan untuk mengisi nilai dari berbagai macam tipe data , bisa berupa bilangan (bulat dan ril), karakter , boolean , string , maupun tipe data bentukan lainnya .

**Praktikkan: Buat file baru bernama "assignment.cpp" dan simpan difolder**

```
assignment.cpp
1  #include<iostream>
2  using namespace std;
3
4  int main (){
5      //Deklarasi variabel
6      int a;
7      double b;
8      char c;
9      char *d;
10
11     //mengisi nilai ke dalam variabel
12     a = 20;
13     b = 3.1232;
14     c = 'C';
15     d = (char*) "Contoh String";
16
17     //menampilkan nilai variabel
18     cout << "Nilai a : " << a << endl;
19     cout << "Nilai b : " << b << endl;
20     cout << "Nilai c : " << c << endl;
21     cout << "Nilai d : " << d << endl;
22
23     return 0;
24 }
```

### Hasil Tampilan:



```
X:\Dokumentasi\Nur Alamsyah_C++\modul 7\assignment.exe
Nilai a : 20
Nilai b : 3.1232
Nilai c : C
Nilai d : Contoh String

-----
Process exited after 0.05476 seconds with return value 0
Press any key to continue . . .
```

## 2. OPERATOR UNARY

Operator unary merupakan operator yang hanya melibatkan satu buah operand . Beberapa operator termasuk kedalam operator unary adalah seperti tampak pada tabel dibawah ini :

Operator	Jenis Operasi	Contoh
+	Membuat nilai positif	+7
-	Membuat nilai negatif	-7
++	<i>Increment</i>	C++
--	<i>Decrement</i>	C--

### # CONTOH KODE PROGRAM OPERATOR PLUS ( + ) DAN MINUS ( - ).

**Praktikkan:** Buat file baru bernama **“plusminus.cpp”** dan simpan difolder

```
plusminus.cpp
1  #include<iostream>
2  using namespace std;
3
4  int main(){
5      int x;
6      float y;
7
8      x = +7; //atau dapat ditulis dengan x = 7
9      y = -3.12; // memasukkan nilai negatif -3.12
10
11     // menampilkan nilai yang disimpan kedalam variabel x dan y
12     cout <<"Nilai x : " << x << endl;
13     cout <<"Nilai y : " << y << endl <<endl;
14
15     x = -x; // merubah nilai x menjadi negatif
16     y = -y; // merubah nilai y menjadi positif
17
18     //menampilkan kembali nilai variabel x dan y
19     cout <<"Nilai x yang baru : " << x << endl;
20     cout <<"Nilai y yang baru : " << y << endl <<endl;
21
22     return 0;
23
24 }
```

### Hasil Tampilan:

```
X:\Dokumentasi\Nur Alamsyah_C++\modul 7\plusminus.exe
Nilai x : 7
Nilai y : -3.12

Nilai x yang baru : -7
Nilai y yang baru : 3.12

-----
Process exited after 0.04865 seconds with return value 0
Press any key to continue . . .
```

## # INCREMENT

**Increment** adalah suatu penambahan nilai yang terjadi pada sebuah variabel . Adapun operator yang digunakan dalam melakukan proses increment adalah operator ++ (plus - plus). Operator ini akan menambahkan nilai dari suatu variabel dengan nilai 1 (satu) .

Terdapat dua tipe increment pada C++ ,yaitu pre-increment dan post-increment , arti dari pre-increment adalah melakukan penambahan nilai sebelum variabel itu diproses , sedangkan post-increment merupakan kebalikannya , yaitu melakukan proses terlebih dahulu sebelum dilakukan penambahan nilai .

Adapun bentuk umum dari pre-increment dan post-increment dapat dilihat dibawah ini :

//melakukan pre-increment

**++nama\_variabel;**

//melakukan post-increment

**nama\_variabel++;**

**Praktikkan: Buat file baru bernama “increment.cpp” dan simpan difolder**

```
increment.cpp
1  #include<iostream>
2  using namespace std;
3
4  int main(){
5      int x; //mendeklarasikan variabel x
6
7      //mengisi nilai kedalam variabel x dengan nilai 7
8      x = 7;
9
10     // melakukan pre-increment
11     cout <<"Nilai x nilai awal : " << x << endl;
12     cout <<"Nilai ++x      : " << ++x << endl;
13     cout <<"Nilai x nilai akhir: " << x << endl <<endl;
14
15     //merubah nilai x menjadi 10
16     x = 10;
17
18     //melakukan post-increment
19     cout <<"Nilai x nilai awal : " << x << endl;
20     cout <<"Nilai x++      : " << x++ << endl;
21     cout <<"Nilai x nilai akhir: " << x << endl <<endl;
22
23     return 0;
24
25 }
```



## Hasil Tampilan:

```
X:\Dokumentasi\Nur Alamsyah_C++\modul 7\increment.exe
Nilai x nilai awal : 7
Nilai ++x          : 8
Nilai x nilai akhir: 8

Nilai x nilai awal : 10
Nilai x++           : 10
Nilai x nilai akhir: 11

-----
Process exited after 0.05434 seconds with return value 0
Press any key to continue . . .
```

## # DECREMENT

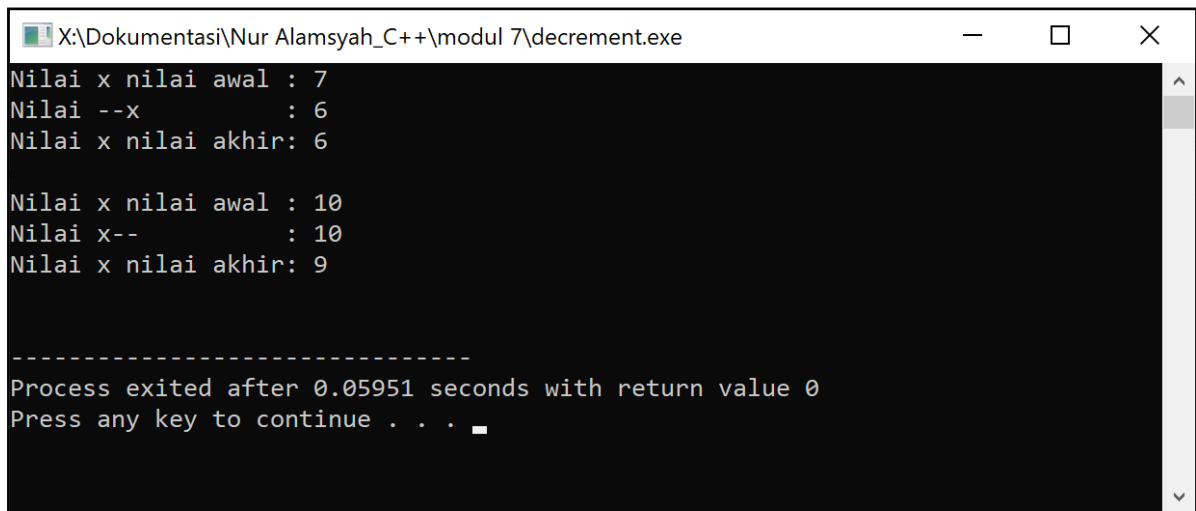
**Decrement** merupakan kebalikan atau lawan dari proses increment, yaitu menurunkan nilai(mengurangi) dari suatu variabel. Sama seperti pada increment pada proses decrement juga di bagi menjadi dua jenis yaitu pre-decrement dan post-decrement.

**Praktikkan: Buat file baru bernama "decrement.cpp" dan simpan difolder**

```
decrement.cpp
3
4 int main(){
5     int x; //mendeklarasikan variabel x
6
7     //mengisi nilai kedalam variabel x dengan nilai 7
8     x = 7;
9
10    // melakukan pre-decrement
11    cout <<"Nilai x nilai awal : " << x << endl;
12    cout <<"Nilai --x          : " << --x << endl;
13    cout <<"Nilai x nilai akhir: " << x << endl <<endl;
14
15    //merubah nilai x menjadi 10
16    x = 10;
17
18    //melakukan post-decrement
19    cout <<"Nilai x nilai awal : " << x << endl;
20    cout <<"Nilai x--           : " << x-- << endl;
21    cout <<"Nilai x nilai akhir: " << x << endl <<endl;
22
23    return 0;
24
25 }
```



### Hasil Tampilan:



```
X:\Dokumentasi\Nur Alamsyah_C++\modul 7\decrement.exe
Nilai x nilai awal : 7
Nilai --x          : 6
Nilai x nilai akhir: 6

Nilai x nilai awal : 10
Nilai x--          : 10
Nilai x nilai akhir: 9

-----
Process exited after 0.05951 seconds with return value 0
Press any key to continue . . .
```

## 3. OPERATOR BINARY

Operator binary adalah operator yang digunakan dalam operasi yang melibatkan dua buah operand.

Dalam bahasa pemrograman C++, operator binary ini dikelompokkan menjadi 4 jenis, yaitu:

1. **Operator Aritmatika,**
2. **Operator Logika,**
3. **Operator Relasional dan**
4. **Operator Bitwise.**

### PENJELASAN!

#### 1. Operator Aritmatika

**Operator aritmetika** adalah operator yang digunakan untuk melakukan operasi-operasi aritmetika seperti penjumlahan, pengurangan, dan sebagainya.

Operator	Jenis Operasi	Contoh
+	Penjumlahan	$2 + 3 = 5$
-	Pengurangan	$5 - 3 = 2$
*	Perkalian	$2 * 3 = 6$
/	Pembagian	$10.0/3.0 = 3.3333$
%	Sisa bagi (modulus)	$10\%3 = 1$

**Praktikkan:** Buat file baru bernama **"aritmatika.cpp"** dan simpan difolder

```

aritmatika.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      //Operator + - * / %
7      int a = 10, b = 4;
8
9      cout<<"a + b = "<<a + b<<endl;
10     cout<<"a - b = "<<a - b<<endl;
11     cout<<"a * b = "<<a * b<<endl;
12     cout<<"a / b = "<<a / b<<endl;
13     cout<<"a % b = "<<a % b<<endl;
14     return 0;
15 }
```

**Hasil Tampilan:**

```
X:\Dokumentasi\Nur Alamsyah_C++\modul 7\aritmatika.exe
a + b = 14
a - b = 6
a * b = 40
a / b = 2
a % b = 2

-----
Process exited after 0.05025 seconds with return value 0
Press any key to continue . . .
```

## 2. Operator Logika

**Operator Logika** adalah operator yang digunakan untuk melakukan operasi dimana nilai yang dihasilkan dari operasi tersebut hanya berupa nilai benar (true) dan salah (false). Nilai ini disebut dengan nilai boolean.

Boolean sendiri ditemukan oleh seorang matematikawan Inggris yang bernama George Bool.

Adapun yang termasuk kedalam operator logika dalam C++ adalah seperti berikut:

Operator	Jenis Operasi	Contoh
&&	AND (dan)	1 && 1 = 1
	OR (atau)	1    0 = 1
!	NOT (negasi)	! 0 = 1

### # Operator && (AND)

Operasi AND hanya akan menghasilkan nilai 1 (benar) jika semua operandnya bernilai benar. Namun jika tidak, maka operasi tersebut akan menghasilkan nilai 0 (salah).

Berikut tabel yang menunjukkan hasil dari operasi AND.

X	Y	X&&Y
1	1	1
1	0	0
0	0	0
0	1	0

### # Operator || (OR)

Operator OR hanya akan menghasilkan nilai 0 (salah) jika semua operator nya bernilai salah, namun jika tidak, maka operator tersebut akan menghasilkan nilai 1 (benar).

Berikut Tabel yang menunjukkan hasil dari operasi OR

X	Y	X    Y
1	1	1
1	0	1
0	0	0
0	1	1

### # Operator ! (NOT)

Nilai yang dihasilkan dari operator NOT adalah kebalikan dari nilai yang dikandung didalamnya.

X	!X
1	0
0	1

**Praktikkan: Buat file baru bernama “logika.cpp” dan simpan difolder**

```
logika.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      bool a, b;
7      a = (7==5);
8      b = (5>4);
9
10     bool x, y, z;
11     x = a && b;
12     y = a || b;
13     z = !(a&&b);
14     cout<< x << y << z <<endl;
15     return 0;
16 }
```

**Hasil Tampilan:**

```
X:\Dokumentasi\Nur Alamsyah_C++\modul 7\logika.exe
011
-----
Process exited after 0.03471 seconds with return value 0
Press any key to continue . . .
```

### 3. Operator Relational

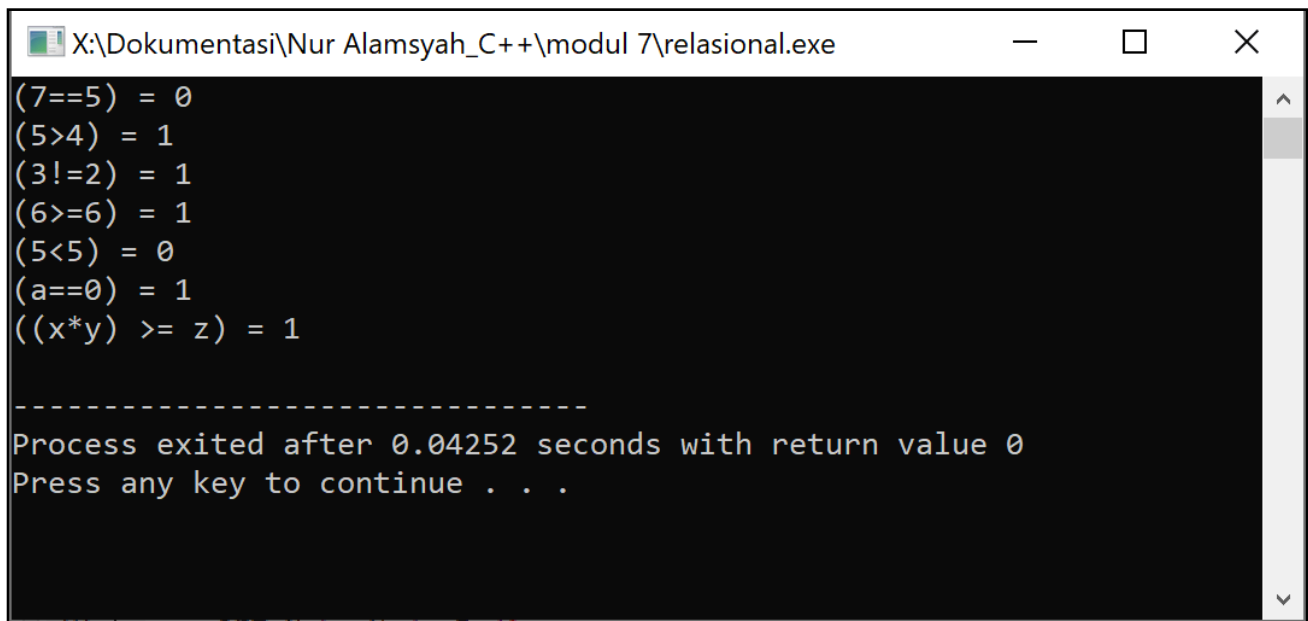
**Operator Relasional** adalah operator yang digunakan untuk menentukan relasi atau hubungan dari dua buah operand. Operator ini ditempatkan didalam sebuah ekspresi, yang kemudian akan menentukan benar atau tidaknya sebuah ekspresi.

Operator	Jenis Operasi	Contoh
>	Lebih besar	(5 > 2) = 1
<	Lebih kecil	(5 < 2) = 0
>=	Lebih besar atau sama dengan	(5 >= 5) = 1
<=	Lebih kecil atau sama dengan	(5 <= 2) = 0
==	Sama dengan	(5 == 2) = 0
!=	Tidak sama dengan	(5 != 2) = 1

**Praktikkan:** Buat file baru bernama **“relasional.cpp”** dan simpan difolder

```
relasional.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      bool a,b,c,d,e;
7      a = (7==5);
8      b = (5>4);
9      c = (3!=2);
10     d = (6>=6);
11     e = (5<5);
12
13     cout<<"(7==5) = "<<a<<endl;
14     cout<<"(5>4) = "<<b<<endl;
15     cout<<"(3!=2) = "<<c<<endl;
16     cout<<"(6>=6) = "<<d<<endl;
17     cout<<"(5<5) = "<<e<<endl;
18
19     int x=5, y=3, z=2;
20     a = (a==0);
21     b = ((x*y) >= z);
22
23     cout<<"(a==0) = "<<a<<endl;
24     cout<<"((x*y) >= z) = "<<b<<endl;
25
26     return 0;
27 }
```

### Hasil Tampilan:



```
X:\Dokumentasi\Nur Alamsyah_C++\modul 7\relasional.exe
(7==5) = 0
(5>4) = 1
(3!=2) = 1
(6>=6) = 1
(5<5) = 0
(a==0) = 1
((x*y) >= z) = 1

-----
Process exited after 0.04252 seconds with return value 0
Press any key to continue . . .
```

## 4. Operator Bitwise

**Operator bitwise** berguna untuk melakukan operasi-operasi yang berhubungan dengan manipulasi Bit. Hanya bisa dilakukan pada operand yang bertipe char dan int saja karena ini berkoresponden dengan tipe byte atau word didalam bit.

Operator	Jenis Operasi	Contoh
&	AND	1 & 0 = 0
	OR	1   0 = 1
^	Exclusive OR (XOR)	1 ^ 1 = 0
~	NOT	~ 1 = 0
>>	Shift Right	5 << 1 = 10
<<	Shift Left	10 >> = 5

**Praktikkan:** Buat file baru bernama **“bitwise.cpp”** dan simpan difolder



```
bitwise.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a = 60, b = 13;
7      // a = 00111100
8      // b = 00001101
9
10     cout<<"a&b = "<<(a&b)<<endl;
11     // a&b = 00001100
12
13     cout<<"a|b = "<<(a|b)<<endl;
14     // a|b = 00111101
15
16     cout<<"a^b = "<<(a^b)<<endl;
17     // a^b = 00110001
18
19     cout<<"~a = "<<(~a)<<endl;
20     // ~a = 11000011
21
22     cout<<"a << 2 = "<<(a << 2)<<endl;
23     // a = 11110000
24
25     cout<<"a >> 2 = "<<(a >> 2)<<endl;
26     // a = 00001111
27
28     return 0;
29 }
```

### Hasil Tampilan:

```
X:\Dokumentasi\Nur Alamsyah_C++\modul 7\bitwise.exe
a&b = 12
a|b = 61
a^b = 49
~a = -61
a << 2 = 240
a >> 2 = 15

-----
Process exited after 0.04261 seconds with return value 0
Press any key to continue . . .
```

4.

```

bitwise.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a = 60, b = 13;
7      // a = 00111100
8      // b = 00001101
9
10     cout<<"a&b = "<<(a&b)<<endl;
11     // a&b = 00001100
12
13     cout<<"a|b = "<<(a|b)<<endl;
14     // a|b = 00111101
15
16     cout<<"a^b = "<<(a^b)<<endl;
17     // a^b = 00110001
18
19     cout<<"~a = "<<(~a)<<endl;
20     // ~a = 11000011
21
22     cout<<"a << 2 = "<<(a << 2)<<endl;
23     // a = 11110000
24
25     cout<<"a >> 2 = "<<(a >> 2)<<endl;
26     // a = 00001111
27
28     return 0;
29 }

```

## OPERATOR TERNARY

**Operator Ternary** adalah operator yang digunakan dalam operasi yang melibatkan tiga buah operand.

Adapun operator yang digunakan untuk menyatakannya adalah **operator ? : ;**

Bentuk umum dari penggunaan operator ternary:

pernyataan: **ekspresi?pilhan1:pilihan2;**

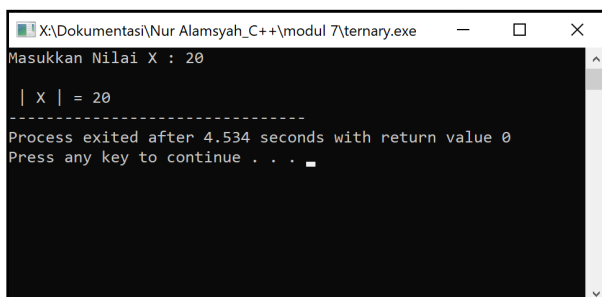
Jika Ekspresi1 bernilai benar, maka program akan mengeksekusi Ekspresi2. Sedangkan jika Ekspresi1 bernilai salah maka yang dieksekusi adalah Ekspresi3.

**Praktikkan: Buat file baru bernama “ternary.cpp” dan simpan difolde**

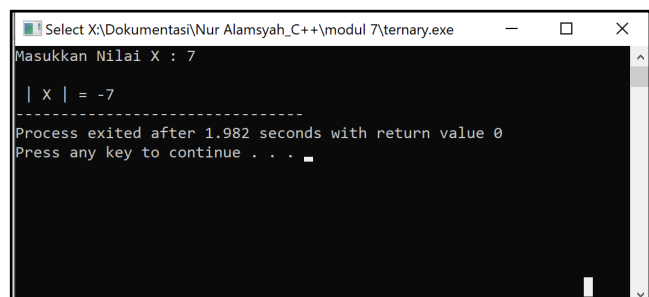
```
ternary.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int X;
7
8      //meminta user untuk memasukkan nilai X dari keyboard
9      cout <<"Masukkan Nilai X : ";
10     cin >> X;
11     cout <<endl;
12
13     //melakukan pemeriksaan terhadap nilai X
14     X = (X<10) ? -X : X;
15
16     //menampilkan nilai X setelah proses pemeriksaan
17     cout<<" | X | = "<<X;
18
19     return 0;
20 }
```

Jika nilai yang dimasukkan lebih dari 10 misal 20 maka program akan membaca pilihan 2 dan jika nilai yang diinputkan kurang dari 10 misal 7 maka program akan menjalankan pilihan 1.

**Hasil Tampilan:**



```
X:\Dokumentasi\Nur Alamsyah_C++\modul 7\ternary.exe
Masukkan Nilai X : 20
| X | = 20
-----
Process exited after 4.534 seconds with return value 0
Press any key to continue . . .
```



```
Select X:\Dokumentasi\Nur Alamsyah_C++\modul 7\ternary.exe
Masukkan Nilai X : 7
| X | = -7
-----
Process exited after 1.982 seconds with return value 0
Press any key to continue . . .
```