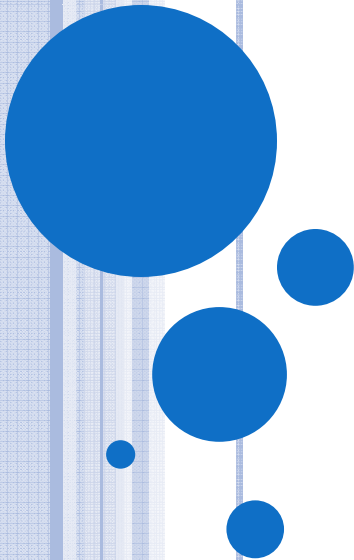


# OPERATOR C++



# PENGERTIAN

- Operator adalah simbol yang mengolah nilai pada operan dan menghasilkan satu nilai baru.
- Operator dapat dikelompokkan menjadi 4 bagian yaitu:
  1. Operator Assignment
  2. Operator Unary
  3. Operator Binary
  4. Operator Ternary



# 1. OPERATOR ASSIGNMENT

Adalah operator yang berfungsi untuk memasukkan (assign) nilai ke dalam suatu variabel ataupun konstanta.

Operator ini dilambangkan dengan tanda sama dengan (=)

Contoh:

```
MyChar = 'C';
```

```
MyString = "Rahasia C++";
```

```
MyInteger = 24;
```

```
MyDouble = 22.021;
```



## 2. OPERATOR UNARY

Adalah operator yang hanya melibatkan sebuah operand.

Yang termasuk ke dalam tabel operator unary antara lain:

Operator	Jenis Operasi	Contoh
+	Positif	+7
-	Negatif	-7
++	Increment	C++
--	decrement	C--



# Increment

*Increment* adalah suatu penambahan nilai yang terjadi pada sebuah variabel. Operator yang digunakan untuk melakukan increment adalah operator ++.

Ada dua jenis *increment* dalam C++ yaitu *pre-increment* dan *post-increment*.

*Pre-increment* artinya melakukan penambahan nilai sebelum suatu variabel itu diproses.

*Post-increment* artinya melakukan proses terlebih dahulu sebelum dilakukan penambahan nilai.



# Decrement

*Decrement* merupakan kebalikan dari proses *increment*, yaitu menurunkan (mengurangi) nilai dari suatu variabel.

Operator yang digunakan untuk melakukan decrement adalah operator --.

*Decrement* juga dibagi 2 yaitu *pre-decrement* dan *post-decremen*.



# CONTOH

```
#include <iostream.h>
int main ()
{
    int A; //mendeklarasikan variabel
    A = 5;
    cout<<"Contoh melakukan pre-increment \n";
    cout<<"Nilai A awal adalah : "<<A<<endl;
    cout<<"Nilai ++A adalah : "<<++A<<endl;
    cout<<"Nilai A akhir adalah:"<<A<<endl;
    cout<<'\n'<<endl;
    cout<<"Contoh melakukan pre-decrement \n";
    cout<<"Nilai A awal adalah : "<<A<<endl;
    cout<<"Nilai --A adalah : "<<--A<<endl;
    cout<<"Nilai A akhir adalah:"<<A<<endl;
    cout<<'\n'<<endl;
}
```

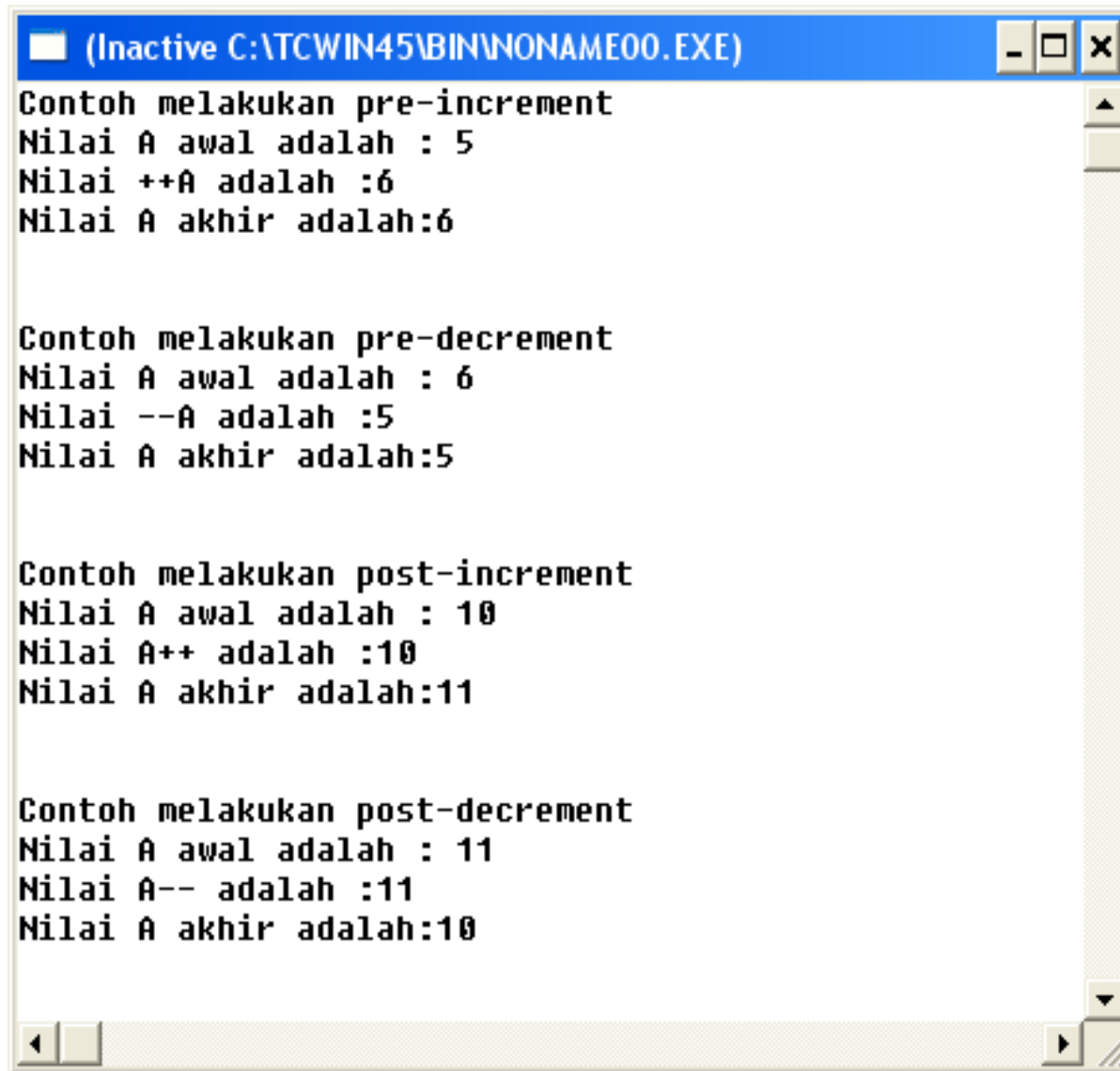


```
//mengubah nilai A menjadi 10
A = 10;
cout<<"Contoh melakukan post-increment \n";
cout<<"Nilai A awal adalah : "<<A<<endl;
cout<<"Nilai A++ adalah : "<<A++<<endl;
cout<<"Nilai A akhir adalah:"<<A<<endl;
cout<<"\n"<<endl;
cout<<"Contoh melakukan post-decrement \n";
cout<<"Nilai A awal adalah : "<<A<<endl;
cout<<"Nilai A-- adalah : "<<A--<<endl;
cout<<"Nilai A akhir adalah:"<<A<<endl;
cout<<"\n"<<endl;
return 0;
}
```





# HASIL EKSEKUSI



```
(Inactive C:\TCWIN45\BIN\NONAME00.EXE)
Contoh melakukan pre-increment
Nilai A awal adalah : 5
Nilai ++A adalah :6
Nilai A akhir adalah:6

Contoh melakukan pre-decrement
Nilai A awal adalah : 6
Nilai --A adalah :5
Nilai A akhir adalah:5

Contoh melakukan post-increment
Nilai A awal adalah : 10
Nilai A++ adalah :10
Nilai A akhir adalah:11

Contoh melakukan post-decrement
Nilai A awal adalah : 11
Nilai A-- adalah :11
Nilai A akhir adalah:10
```



### 3. Operator Binary

Adalah operator yang digunakan dalam operasi yang melibatkan dua buah operand.

Operator Binary dikelompokkan menjadi 4 jenis yaitu:

- a. **Operator Aritmatika**
- b. **Operator Logika**
- c. **Operator Relasional**
- d. **Operator Bitwise**



# Operator Aritmatika

Adalah operator yang digunakan untuk melakukan operasi-operasi aritmatika seperti penjumlahan, pengurangan, dsb.

## **Operator**

\*

/

%

+

-

## **Keterangan**

Perkalian

Pembagian

Modulus atau sisa bagi

Penjumlahan

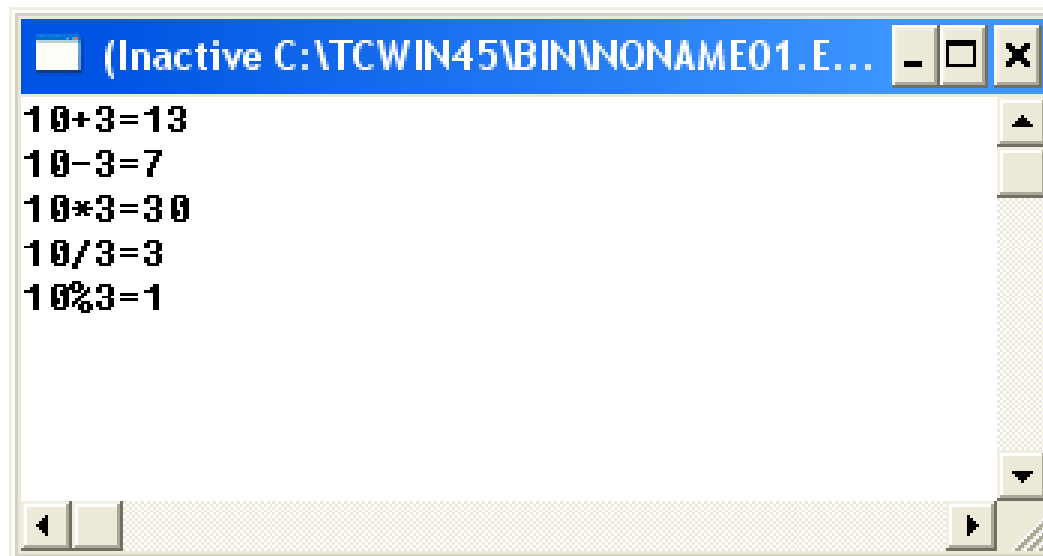
Pengurangan



```
#include<iostream.h>
int main ()
{
    int X = 10, Y = 3;
    int jumlah, kurang, kali, bagi, sisa_bagi;
    jumlah = X + Y;
    kurang = X - Y;
    kali = X * Y;
    bagi = X / Y;
    sisa_bagi= X % Y;
    cout<<X<<"+"<<Y<<"="<<jumlah<<endl;
    cout<<X<<"-"<<Y<<"="<<kurang<<endl;
    cout<<X<<"*"<<Y<<"="<<kali<<endl;
    cout<<X<<"/"<<Y<<"="<<bagi<<endl;
    cout<<X<<"%"<<Y<<"="<<sisa_bagi<<endl;
    return 0;
}
```



# HASIL EKSEKUSI



A screenshot of a Windows command prompt window. The title bar is blue and contains the text "(Inactive C:\TCWIN45\BIN\NONAME01.E...". The window contains the following text:

```
10+3=13  
10-3=7  
10*3=30  
10/3=3  
10%3=1
```

The window has standard Windows window controls (minimize, maximize, close) in the top right corner and scroll bars on the right and bottom.



# Operator Logika

- Adalah operator yang digunakan untuk melakukan operasi dimana nilai yang dihasilkan dari operasi tersebut hanya bernilai benar (true / 1) atau salah (false / 0). Nilai ini biasa disebut dengan boolean. Yang termasuk operator logika antara lain:

## 1. Operator AND (&&)

- hanya akan menghasilkan nilai 1 (benar) jika semua operand-nya bernilai benar, namun jika tidak maka operasi tersebut akan menghasilkan nilai 0 (salah).



## Contoh Pembuktian AND

```
#include<iostream.h>
int main()
{
    cout<<"1&&1="<<(1&&1)<<endl;
    cout<<"1&&0="<<(1&&0)<<endl;
    cout<<"0&&1="<<(0&&1)<<endl;
    cout<<"0&&0="<<(0&&0)<<endl;
    return 0;
}
```



## 2. Operator OR (| |)

- hanya akan menghasilkan nilai 0 (salah) jika semua operand-nya bernilai salah, namun jika tidak maka operasi tersebut akan menghasilkan nilai 1 (benar).

## 3. Operator NOT (!)

- Nilai yang dihasilkan oleh nilai NOT adalah kebalikan dari nilai yang dikandung di dalamnya. Jika nilai awal 1 (benar) maka nilai operasi NOT menjadi 0 (salah).

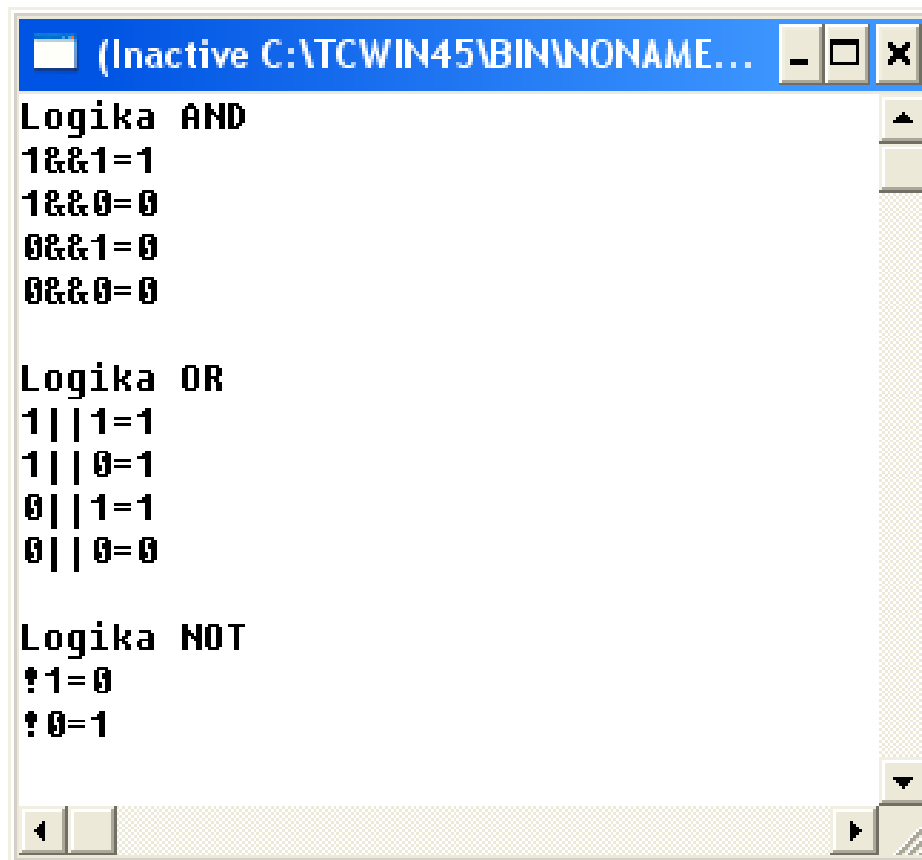




```
#include<iostream.h>
int main()
{
    cout<<"Logika AND"<<endl;
    cout<<"1&&1="<<(1&&1)<<endl;
    cout<<"1&&0="<<(1&&0)<<endl;
    cout<<"0&&1="<<(0&&1)<<endl;
    cout<<"0&&0="<<(0&&0)<<endl;
    cout<<endl;
    cout<<"Logika OR"<<endl;
    cout<<"1 | 1="<<(1 | 1)<<endl;
    cout<<"1 | 0="<<(1 | 0)<<endl;
    cout<<"0 | 1="<<(0 | 1)<<endl;
    cout<<"0 | 0="<<(0 | 0)<<endl;
    cout<<endl;
    cout<<"Logika NOT"<<endl;
    cout<<"!1="<<(!1)<<endl;
    cout<<"!0="<<(!0)<<endl;
    return 0;
}
```



# HASIL EKSEKUSI



```
(Inactive C:\TCWIN45\BIN\NONAME... - [ ] [X])
Logika AND
1&1=1
1&0=0
0&1=0
0&0=0

Logika OR
1||1=1
1||0=1
0||1=1
0||0=0

Logika NOT
!1=0
!0=1
```



# Operator Relasional

- Adalah operator yang digunakan untuk menentukan relasi atau hubungan dari dua buah operand. Operator ini banyak digunakan untuk melakukan pengecekan sebuah ekspresi (kondisi) dalam struktur percabangan.

## Operator Jenis

## Operasi

>

Lebih besar

<

Lebih kecil

>=

Lebih besar atau sama dengan

<=

Lebih kecil atau sama dengan

==

Sama dengan

!=

Tidak sama dengan

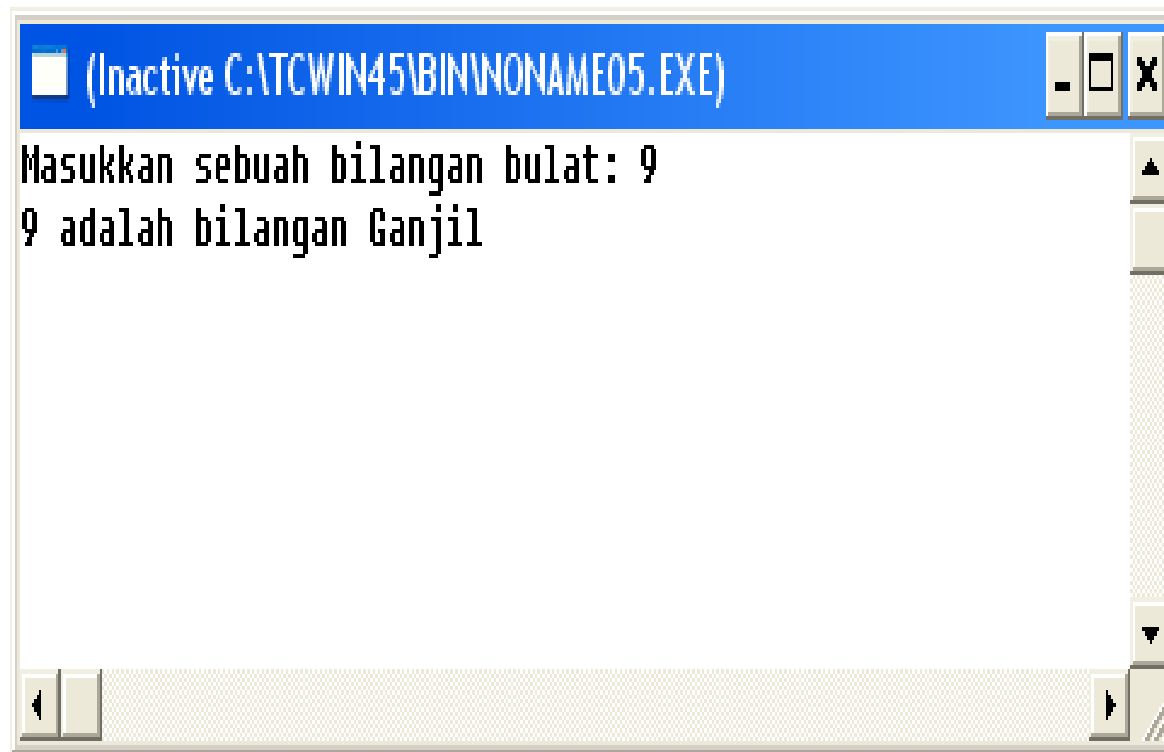


## CONTOH

```
#include <iostream.h>
int main()
{
    int nilai;
    //memasukkan bilangan bulat
    cout<<"Masukkan sebuah bilangan bulat: ";
    cin>>nilai;
    if (nilai % 2 == 0)
        cout<<nilai<<" adalah bilangan Genap"<<endl;
    else
        cout<<nilai<<" adalah bilangan Ganjil"<<endl;
    return 0;
}
```



# HASIL EKSEKUSI



## Operator Bitwise

Adalah operator yang digunakan untuk melakukan operasi-operasi yang berhubungan dengan pemanipulasian bit.

Yang termasuk dalam operator bitwise antara lain:

<b>Operator</b>	<b>Keterangan</b>
<<	Shift left
>>	shift right
&	operasi bit AND
	Operasi bit OR
^	Operasi bit XOR
~	Operasi bit NOT



- Operator **&** berguna untuk melakukan operasi **"DAN"** pada tataran bit.

1001 → 9

1010 → 10

1000 → 8

- Operator **|** berguna untuk melakukan operasi **"ATAU"** pada tataran bit.

1001 → 9

1010 → 10

1011 → 11

- Operator **^** berguna untuk melakukan operasi **"ATAU EKSKLUSIF"** pada tataran bit.

1001 → 9

1010 → 10

0011 → 3



- Operator  $\sim$  memberikan hasil dengan masing-masing bit berupa kebalikan dari bit operand.

1011 1001

0100 0110

- Operator  $\ll$  (geser kiri) berfungsi untuk menggeser bit-bit ke kiri.

contoh:

$29 \ll 1$  memberikan nilai hasil 58

0001 1101  $\longrightarrow$  29

0011 1010  $\longrightarrow$  58

- Operator  $\gg$  (geser kanan) berfungsi untuk menggeser bit-bit ke kanan.

contoh:  $29 \gg 1$  memberikan nilai 14

0001 1101  $\longrightarrow$  29

0000 1110  $\longrightarrow$  14



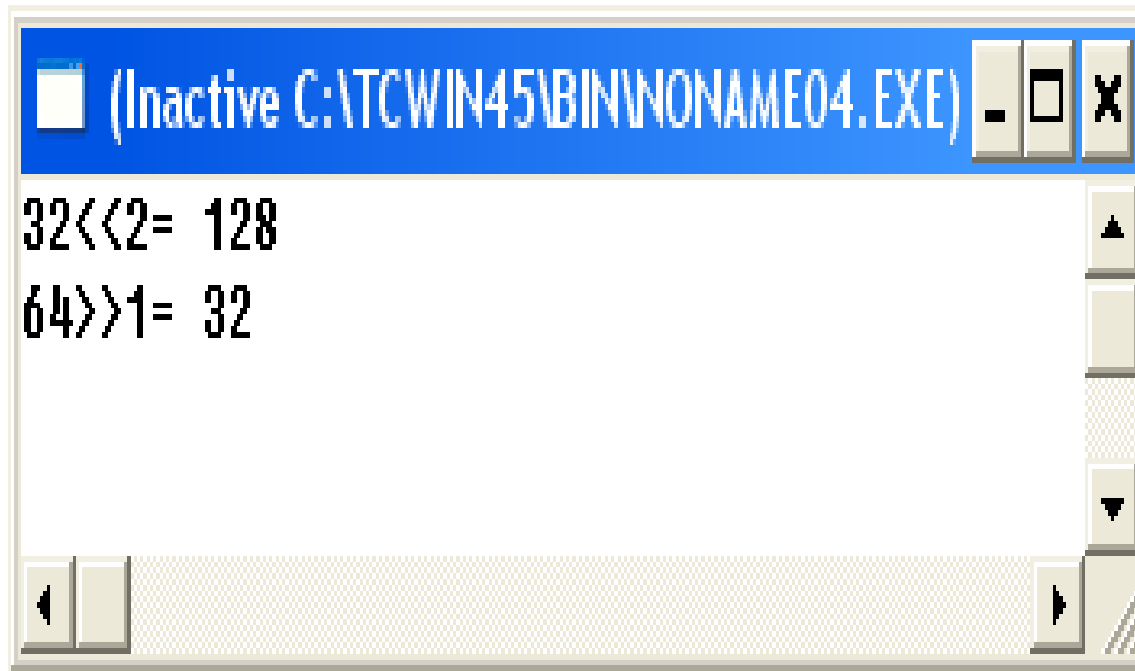


## Contoh 1

```
#include<iostream.h>
int main()
{
    int X, Y;
    //geser ke kiri 2 bit
    X = 32<<2;
    //geser ke kanan 1 bit
    Y = 64>>1;
    //menampilkan hasil
    cout<<"32<<2= "<<X<<endl;
    cout<<"64>>1= "<<Y<<endl;
    return 0;
}
```



## Hasil Eksekusi



The image shows a screenshot of a Windows command prompt window. The title bar is blue and contains the text "(Inactive C:\TCWIN45\BIN\NONAME04.EXE)". The main area of the window is white and contains two lines of text: "32<<2= 128" and "64>>1= 32". The window has standard Windows window controls (minimize, maximize, close) in the top right corner and a scroll bar on the right side. The bottom of the window has a greyed-out area with a scroll bar on the left and a small icon on the right.

```
(Inactive C:\TCWIN45\BIN\NONAME04.EXE)
32<<2= 128
64>>1= 32
```

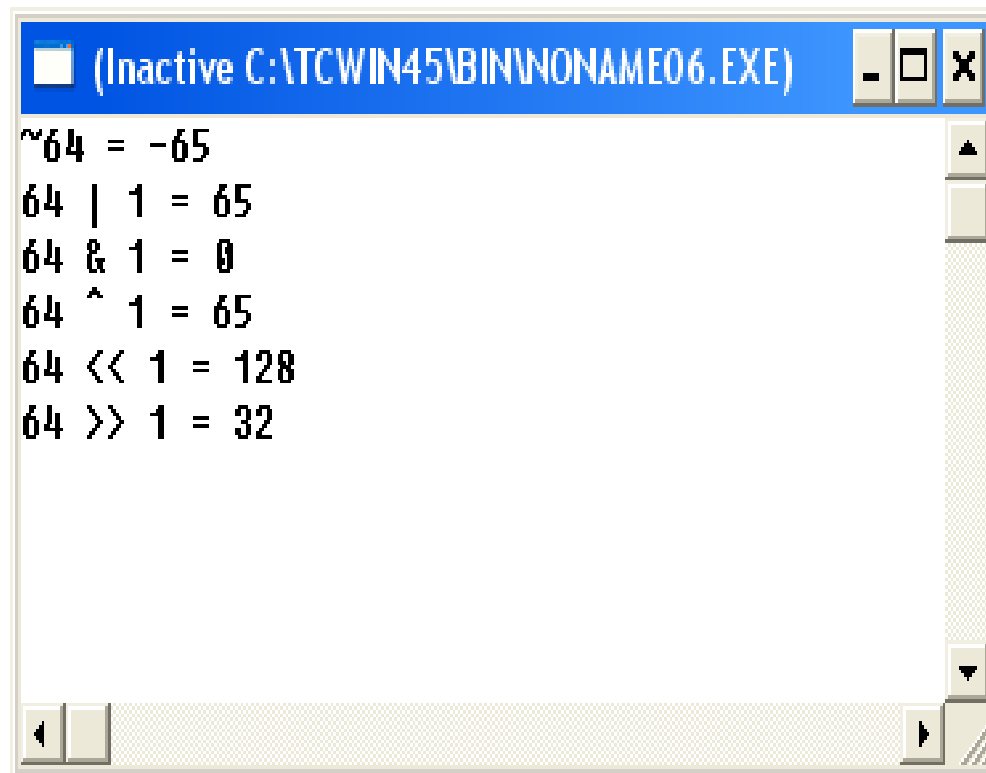


## CONTOH 2

```
#include<iostream.h>
int main()
{
    int x=64;
    int y= 1;
    cout<<'~'<<x<<" = "<< ~x<<endl;
    cout<<x<<" | "<<y<<" = "<<(x | y)<<endl;
    cout<<x<<" & "<<y<<" = "<<(x&y)<<endl;
    cout<<x<<" ^ "<<y<<" = "<<(x^y)<<endl;
    cout<<x<<" << "<<y<<" = "<<(x<<y)<<endl;
    cout<<x<<" >> "<<y<<" = "<<(x>>y)<<endl;
    return 0;
}
```



# HASIL EKSEKUSI



The screenshot shows a Windows command prompt window with the title bar "(Inactive C:\TCWIN45\BIN\NONAME06.EXE)". The window contains the following text:

```
~64 = -65  
64 | 1 = 65  
64 & 1 = 0  
64 ^ 1 = 65  
64 << 1 = 128  
64 >> 1 = 32
```



## 4. Operator Ternary

- Adalah operator yang digunakan dalam operasi yang melibatkan tiga buah operand. **Bentuk umum:**

Ekspresi1? Ekspresi2: Ekspresi3;

- Jika ekspresi1 bernilai benar, maka program akan mengeksekusi ekspresi2. Sedangkan jika ekspresi1 salah maka yang dieksekusi adalah ekspresi3.



## Contoh Listing Program

```
#include<iostream.h>
int main()
{
    int angka;
    cout<<"Masukkan bilangan: ";
    cin>>angka;
    (angka %2 ==0) ? (cout<<angka<<" adalah
    bilangan Genap \n") : (cout<<angka<<" adalah
    bilangan Ganjil \n");
    return 0;
}
```



# Hasil Eksekusi Program

