**MODUL 2**

Operator

Modul **Praktikum**

C++

**Dasar Pemrograman Komputer**

**JURUSAN TEKNIK ELEKTRO**

FAKULTAS TEKNIK

MODUL 2

# OPERATOR

1. **Tujuan**

Setelah mempelajari bab ini diharapkan siswa 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.

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

# Jenis-jenis operator

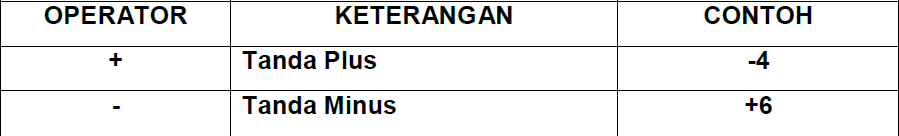
* 1. **Operator Aritmatika**

Operator untuk operasi aritmatika yang tergolong sebagai binary adalah:

# Tabel 2.1.1 Operator Aritmatika



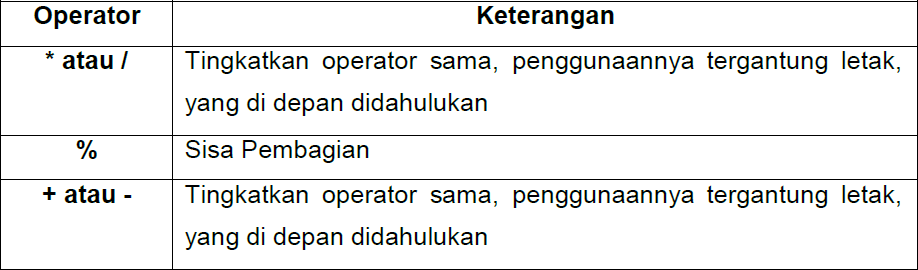
**Tabel 2.1.2 Operator Unary**



# 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



**Contoh:**

A = 8 + 2 \* 3 / 6

Langkah perhitungannya :

A = 8 + 6/6 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 dimana 8 + 2 = 10

A = 30 / 6

A = 5

# Operator Penugasan

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

# Contoh :

**nilai = 80; A = x \* y;**

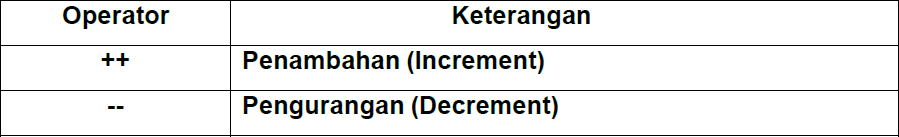
Penjelasan :

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

# Operator Increment & Decrement

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

# Tabel 1.4 Operator Increment & 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++

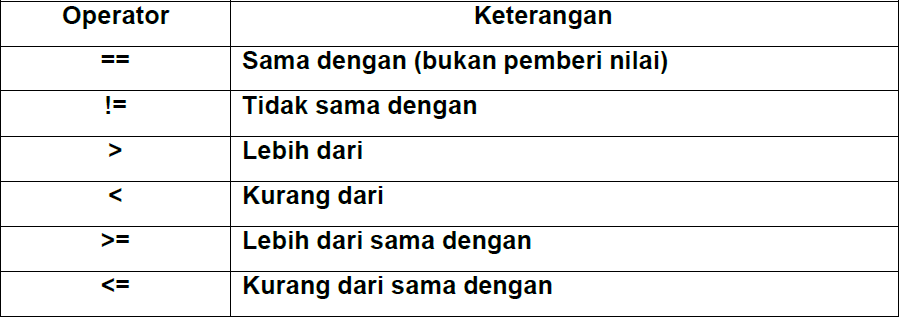
Kedua bentuk penulisan di atas mempunyai arti yang berbeda:

* **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

# 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)



* 1. **Operator Logika**

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

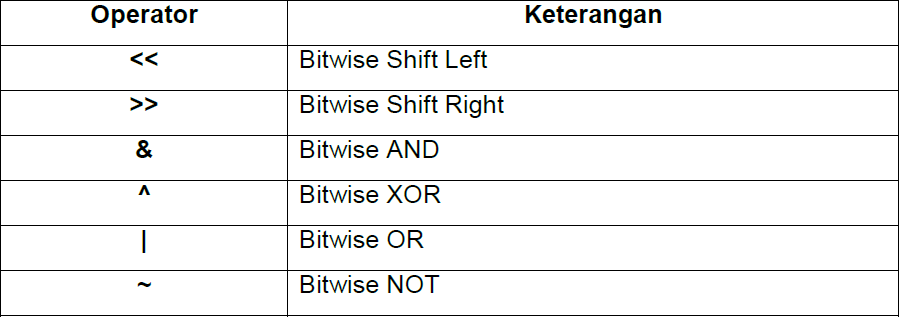
# Tabel 1.6 Operator Relasi Logika



* 1. **Operator Bitwise**

Operator Bitwise digunakan untuk memanipulasi data dalam bentuk bit

# Tabel 1.7 Operator Bitwise



* + 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

# Operator Bitwise >> Shift Right

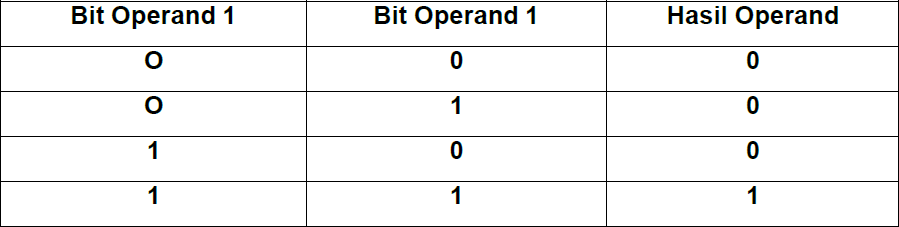
Operator Bitwise >> Shift Right digunakan untuk menggeser sejumlah bit ke kanan

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

# 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)



**Contoh :**

0001 0101

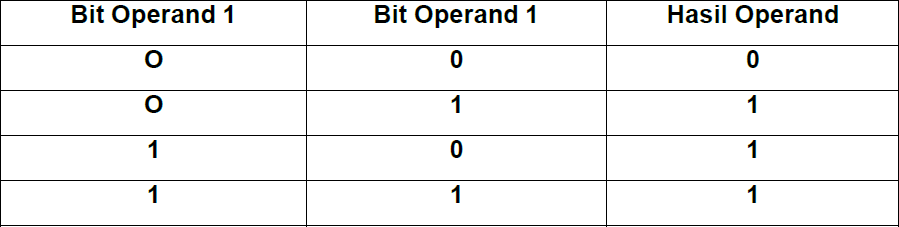
0001 0001 **= AND**

0001 0001

# Operator Bitwise | (OR)

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

# Tabel 1.9 Operator Bitwise | (OR)



**Contoh :**

0001 0101

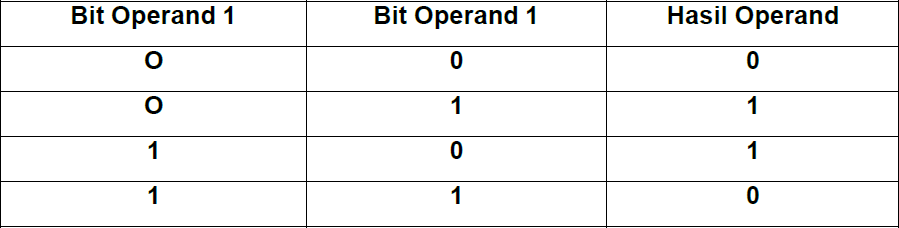
0001 0001 **= OR**

0001 0101

# Operator Bitwise ^ (XOR)

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

# Tabel 1.1 Operator Bitwise ^ (XOR)



**Contoh :**

0001 0101

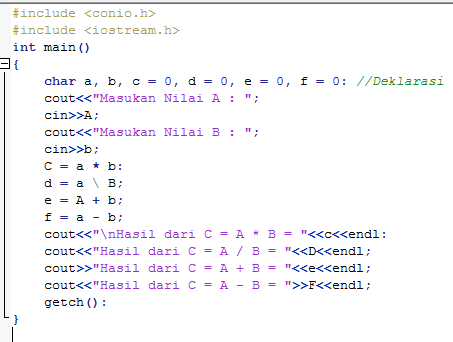
0001 0001 **= XOR**

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**!



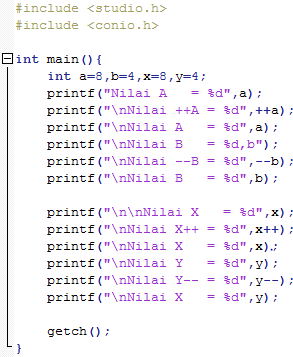
# Latihan 2

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**!



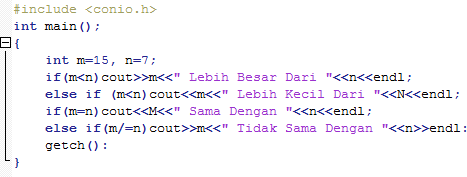
# Latihan 3

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**!



# Latihan 4

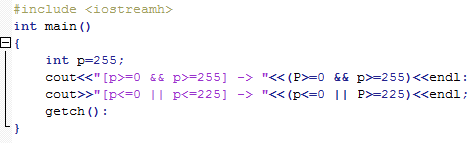
Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama **Lat4**, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program **Lat4**!



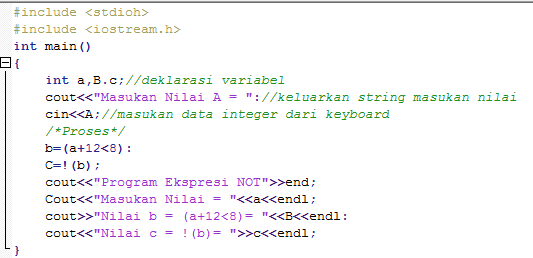
# Latihan 5

* 1. 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**!

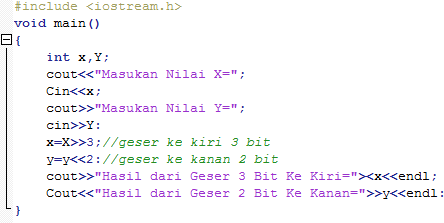


* 1. Tuliskan kode program di bawah ini dengan menggunakan program MinGW Developer Studio, kemudian compile dan simpan dengan nama **Lat5b**, execute kode program tersebut., jika ada error perbaiki program tersebut. Jelaskan & simpulkan hasil execute program **Lat5b**!



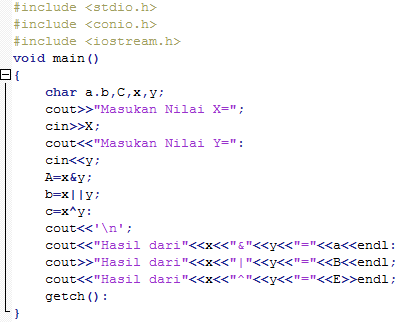
# Latihan 6

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**!



# 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**!



# Tugas Rumah

Buat suatu program yang menghasilkan data seperti di bawah ini !

