|  |
| --- |
| DASAR BAHASA  PEMROGRAMAN BORLAND C++ |
|  |

**SUBSTANSI :**

**1. PROSES KOMPILASI**  
**2. STRUKTUR PROGRAM C++**

* #include,
* void, main(),
* Blok pernyataan,
* cout,
* clrscr();
* Komentar

**3. ELEMEN DASAR DALAM C++**

* Pengenal (Identifier),
* Tipe Data Dalam C++,
* Variabel dan Konstanta,
* Operator dan ungkapan,
* Input dalam C++
* Manipulator

**1. PROSES KOMPILASI**

Program C++ ditulis dengan ekstensi .cpp. Agar dapat dieksekusi atau dijalankan, maka program harus dikompilasi terlebih dahulu menggunakan compiler C++. Proses kompilasinya adalah; file sumber (.cpp) bersama dengan file header (.h) diterjemahkan dulu oleh kompiler C++ sehingga menjadi kode objek (.obj), format file objek ini adalah biner (berkode 0 dan 1). Selanjutnya semua file objek bersama dengan file pustaka (.lib) dikaitkan menjadi satu oleh linker dan hasilnya adalah file Executable.

|  |
| --- |
| [Gb. proses pembentukan file executable](http://1.bp.blogspot.com/-y-UvUXWyFrU/VNszFlbO8CI/AAAAAAAAEzg/ZOKHwSYfvOU/s1600/Gb.Proses+kompilasi.PNG) |
| Gb. proses pembentukan file executable |

**2. STRUKTUR PROGRAM C++**

#include <nama\_file>

**void** main()

{

<blok\_pernyataan>

}

**#include**

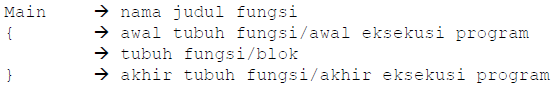
Adalah pengarah praprosesor yang berfungsi menginstruksikan kepada kompiler untuk menyisipkan file, saat program dikompilasi. Biasanya file yang disisipkan adalah file header.

void di depan main()

Dipakai untuk menyatakan bahwa fungsi **main()** tidak memiliki nilai balik.

**main()**

Merupakan fungsi utama yang menjadi awal dan akhir eksekusi program C++.

[](http://4.bp.blogspot.com/-pS6zyBXt8tE/VNs2zApEHJI/AAAAAAAAEzs/lU5v6WH5Yoc/s1600/fungsi+main.PNG)

**Tanda ()**

Digunakan untuk mengapit argumen fungsi, yaitu nilai yang akan dilewatkan ke fungsi.

**Blok pernyataan**

Blok pernyataan bisa terdiri dari satu atau beberapa statemen dan diakhiri dengan tanda **(;)**sebagai contoh saat mendeklarasikan 3 buah variabel dengan tipe data yang sama, seperti; **int a, b, c;**

**Contoh program :**

#include <iostream.h>

#include <conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Program yang mengandung blok pernyataan*

*\*/*

**void** main()

{

clrscr(); *//membersihkan layar*

cout << "Selamat Belajar C++\n";

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [Blok Pernyataan](https://2.bp.blogspot.com/-7WId8hZze-E/Vyg_EPKvYoI/AAAAAAAAKp8/RPXEV0P6PuM48I_m4qsVJyU9LnqGkhu6ACLcB/s1600/slamat+belajar+cpp.PNG) |
| **Blok Pernyataan** |

Untuk mengkompile program dengan menggunakan borland C++ dapat dilakukan dengan menekan tombol Ctrl + F9 atau tombol ini :

[https://1.bp.blogspot.com/-ISfxBAgE0wA/VNs8YekQx9I/AAAAAAAAE0E/CBxMha5wjHM/s1600/run%2Bbutton.PNG](http://1.bp.blogspot.com/-ISfxBAgE0wA/VNs8YekQx9I/AAAAAAAAE0E/CBxMha5wjHM/s1600/run+button.PNG)

**cout (dibaca "c out")**

Merupakan objek dalam C++ yang digunakan untuk mengarahkan data ke standar output (layar).

**Tanda <<**

Dua tanda kurang dari adalah operator “penyisipan/peletakan” yang akan mengarahkan operand (data) yang terletak di sebelah kanannya ke objek yang terletak di sebelah kirinya.

Pada contoh di atas, “Selamat Belajar C++\n” diarahkan ke cout, yang memberikan hasil berupa tampilan string tersebut ke layar. **\n** adalah karakter pindah baris (new line).

**#include < iostream.h >**

Menginstruksikan kepada kompiler untuk menyisipkan file iostream.h pada saat program dikompilasi tanpa diakhiri titik koma. File iostream.h perlu disertakan pada program yang melibatkan cout. Tanpa # include < iostream.h > akan terjadi kesalahan saat program dikompilasi. Sebab file iostream.h berisi deklarasi yang diperlukan oleh cout dan berbagai objek yang berhubungan dengan operasi masukan–keluaran.

**clrscr();**

Pernyataan yang diperlukan untuk menghapus layar. Apabila menggunakan pernyataan ini maka harus disertakan file header conio.h.

**Komentar**

Diperlukan untuk menjelaskan / memberikan anotasi terkait program atau bagian-bagian dalam program. Isi penjelasan biasanya berupa:

* Tujuan/fungsi program
* Saat program dibuat/direvisi
* Keterangan-keterangan lain tentang kegunaan sejumlah pernyataan dalam program

Prosedur komentar pada pemrograman C++ memiliki 2 cara yaitu;

1. **Diawali tanda // (dua tanda garis miring),**Semua tulisan setelah tanda // dianggap sebagai komentar dan tidak akan dieksekusi oleh C++.
2. **Diawali tanda /\* blok komentar dan diakhiri tanda \*/**, Untuk memberikan komentar lebih dari 1 baris, namun, biasanya bentuk ini sering dimanfaatkan untuk mengabaikan / menonaktifkan sejumlah baris program atau pernyataan yang telah dibuat karena suatu alasan.

**Contoh program :**

#include <iostream.h>

#include <conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Program yang mengandung komentar*

*\*/*

**void** main()

{

clrscr(); *//membarsihkan layar <--- ini adalah komentar*

*// teks ini sampai akhir baris tidak akan pernah dieksekusi <--- ini adalah komentar*

cout << "Selamat Belajar C++\n";

getch();

}

**3. ELEMEN DASAR DALAM C++**

**3.1 Pengenal (Identifier)**

Pengenal merupakan nama yang akan digunakan dalam pemrogrman yang biasa digunakan untuk menyatakan variabel, konstanta, tipe data, fungsi, label, obyek serta hal-hal lain yang dibuat oleh pemrogram.

Suatu pengenal merupakan kombinasi dari huruf, angka dan garis bawah (\_). Penamaan pengenal harus berawalan dengan huruf atau garis bawah dan menggunakan kata yang mudah dipahami dan dapat mewakili fungsi dari pengenal yang dibuat. Pengenal dalam C++ bersifat sensitive case atau dibedakan antara huruf kecil dan huruf besar. Misalkan pengenal gajipokok, GajiPokok, GAJIPOKOK, merupakan tiga buah pengenal yang berbeda.

**3.2 Tipe Data Dalam C++**

Mengenai tipe data, Anda bisa baca selengkapnya di sini :  
[**PENGERTIAN TIPE DATA, VARIABEL DAN KONSTANTA**](http://www.gatewan.com/2014/07/pengertian-tipedata-variabel-dan-konstanta.html)

**Contoh program :**

#include <iostream.h>

#include <conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Program yang mengandung tipe data dalam c++*

*\*/*

**void** main()

{

clrscr();

**char** huruf;

huruf = 'B';

cout<<"Isi huruf bertipe char = "<<huruf<<'\n';

**int** angka;

angka = 123;

cout<<"Isi angka bertipe int = "<<angka<<'\n';

**long** jumlah;

jumlah = 12345678;

cout<<"Isi jumlah bertipe long = "<<jumlah<<'\n';

**float** nilai;

nilai = 234.543;

cout<<"Isi nilai bertipe float = "<<nilai<<'\n';

**double** cacah;

cacah = 3453.345;

cout<<"Isi cacah bertipe double = "<<cacah<<'\n';

**long** **double** total;

total = 23456.3945;

cout<<"Isi total bertipe long double = "<<total<<'\n';

getch();

}

|  |
| --- |
| **Hasil Eksekusi :**  [Tipe Data C++](https://1.bp.blogspot.com/--NrHfdGO2Fs/VyhJJWdt-BI/AAAAAAAAKqM/yKsZiyWMGQk78ZvlcS0XvpTTpfnWMfV0gCLcB/s1600/tipe+data+c++.PNG) |
| Tipe Data C++ |

**3.3 Variabel dan Konstanta**

Dalam proses pemrograman tipe data biasa digunakan untuk mendefinisikan suatu variabel atau konstanta. Variabel adalah suatu memori yang dialokasikan dengan nama tertentu dan hanya bisa menampung data sesuai dengan tipe yang ditentukan. Sifat dari variabel adalah nilai yang dikandung akan mudah diubah sesuai dengan proses yang terjadi seperti contoh dibawah ini. Sedangkan konstanta adalah suatu memori yang dialokasikan dengan nama tertentu yang berisi suatu nilai yang memiliki sifat tetap yang tidak akan bisa berubah.

Sebelum variabel digunakan maka variabel tersebut harus didefinisikan terlebih dahulu. Pendefinisian variabel dapat dimana saja sebelum variabel itu digunakan dengan bentuk :

tipevariabel namavariabel;

Untuk memasukkan nilai kedalam variabel digunakan bentuk :

variabel = nilai;

**Contoh program :**

#include <iostream.h>

#include <conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Program yang mengandung pemasukkan nilai pada variabel*

*\*/*

**void** main()

{

**int** bilangan; *//Mendefinisikan variabel bilangan bertipe int*

bilangan = 10; *//Memberikan nilai 10 pada variabel bilangan*

clrscr();

cout<<"Isi bilangan = "<<bilangan<<'\n';

bilangan = 50;

cout<<"Isi bilangan = "<<bilangan<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [Variabel](https://4.bp.blogspot.com/-OW-ulxBJT9Y/VyhKv8ADB-I/AAAAAAAAKqY/kdw3OHKBMjc2ZC0cwflJhu_na8sORb9fQCLcB/s1600/variabel+konstanta.PNG) |
| **Variabel** |

Untuk mendeklarasikan konstanta digunakan bentuk :  
const tipedata namakonstanta = nilaikonstanta;

**Contoh program :**

#include <iostream.h>

#include <conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Program yang mengandung konstanta*

*\*/*

**void** main()

{

**const** **float** phi = 3.14;

**int** jari = 11;

**float** luas;

clrscr();

luas = phi \* jari \* jari;

cout<<"Luas lingkaran = "<<luas<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [Konstanta](https://3.bp.blogspot.com/-YAxPZXFvdoM/VyhLv9RSDEI/AAAAAAAAKqg/MZpvhyAymtUx9ZRViC5XlyHDF5wWw9QuACLcB/s1600/konstanta.PNG) |
| **Konstanta** |

Dalam C++ dikenal adanya pemodifikasian variabel menggunakan unsigned dan signed. Variabel yang ditambahi unsigned akan menyebabkan nilai yang dikandung variabel benilai positif, sedangkan signed tidak menyebabkan perubahan nilai dari data yang dikandung (sama dengan nilai data dasar).

|  |  |  |
| --- | --- | --- |
| **Modifikasi Tipe Data** | **Persamaan** | **Jangkauan Nilai** |
| unsigned char | tidak ada | 0 s/d 225 |
| unsigned int | unsigned | 0 s/d 65.535 |
| unsigned short int | unsigned short | 0 s/d 65.535 |
| unsigned long int | unsigned long | 0 s/d 4.294.967.295 |
| signed char | char | -128 s/d 127 |
| signed int | int | -32.768 s/d32.767 |
| signed short int | short, signed short | -32.768 s/d32.767 |
| signed long int | long, long int, signed long | -2.147.483.648 s/d 2.147.483.687 |

Untuk memahami lebih dalam, simak obeservasi  di bawah ini;

* [**Cara Mengetahui Jangkauan Tipe Data Integer dan Unsign Integer pada C++**](http://www.gatewan.com/2014/04/mengetahui-jangkauan-tipe-data-integer-dan-unsign-integer.html)

**3.4 Operator dan ungkapan**

Operator merupakan simbol yang akan digunakan untuk melakukan suatu operasi atau manipulasi. Untuk menghasilkan suatu nilai operator harus menerima operand. Sedangkan ungkapan merupakan gabungan antara operator dengan operand yang menghasilkan nilai ungkapan, contoh :

**6 + 10 – 4**

Pada ungkapan diatas terdiri dari dua operator yaitu + (penjumlahan) dan - (pengurangan) dan tiga operand yaitu 6, 10 dan 4, nilai ungkapan diatas adalah 12.

**Operator Aritmatika**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Keterangan** | **Contoh** |
| \* | Perkalian | 10 \* 5 |
| / | Pembagian | 20 / 4 |
| % | Sisa pembagian (modulus) | 22 % 3 |
| + | Penjumlahan | 5 + 6 |
| - | Pengurangan | 8 - 4 |

**Contoh program :**

#include<iostream.h>

#include<conio.h

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan operator aritmatika*

*\*/*

**void** main()

{

**int** d,b,a,c;

b = 7;

a = 5;

c = 2;

d = b \* b - 4 \* a \* c;

cout<<"Nilai d = "<<d<<'\n';

*//mencari sisa bagi*

**int** sisa;

sisa = 20 % 3;

cout<<"Sisa 20 % 3 = "<<sisa<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [Operator Aritmatika](https://4.bp.blogspot.com/--W2F4VNy4dw/VyhP5dU1Q4I/AAAAAAAAKqs/zOhORWi3Vj4kPeVgmTyIaI75Beof6d5KACLcB/s1600/operator.PNG) |
| **Operator Aritmatika** |

**Operator Penugasan**

Operator penugasan menggunakan simbol sama dengan (=) yang berfungsi untuk memberikan nilai pada suatu variabel. Contoh penggunaan operator penugasan dalam C++ :

|  |  |
| --- | --- |
| **Contoh** | **Keterangan** |
| y = 4 + ( a = 7 ); | Variabel a diberi nilai 7, lalu y diisi nilai ungkapan 4 + 7 |
| a = b = c = d = e = 10; | Pertama **e** diisi dengan 10, lalu **d** diisi dengan nilai **e**, lalu **c** diisi dengan nilai **d**, dan **a** diisi dengan nilai **b**. |

**Contoh program :**

#include<iostream.h>

#include<conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan operator penugasan*

*\*/*

**void** main()

{

**int** a,b,c,d,e;

a = b = c = d = e = 100;

cout<<"Isi a = "<<a<<'\n';

cout<<"Isi b = "<<b<<'\n';

cout<<"Isi c = "<<c<<'\n';

cout<<"Isi d = "<<d<<'\n';

cout<<"Isi e = "<<e<<'\n';

**int** x,y;

x = 10 \* (y = 3);

cout<<"Isi x = "<<x<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [Operator Penugasan](https://2.bp.blogspot.com/-4wXOuQB77bs/VyhShS5QivI/AAAAAAAAKq4/zIXqMe0MGpAYYD_-PCAi0cmflEEplMMSACLcB/s1600/operator+penugasan.PNG) |
| **Operator Penugasan** |

**Operator penaikkan (increment) dan penurunan (decrement)**

Digunakan untuk menaikkan dan menurunkan nilai variabel yang bertipe bilangan bulat. Bentuk operator ini adalah ++ (penaikkan) dan -- (penurunan). Sebagai contoh :

x = x + 1;

y = y - 1;

**Dapat ditulis :**

y--;

x++;

**atau**

--y;

++x;

**Contoh program:**

#include<iostream.h>

#include<conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan operator penaikkan dan penurunan*

*\*/*

**void** main()

{

**int** a,b,c;

a = 15;

*//variabel a dinaikkan 1 baru dijumlahkan dengan 10*

b = 10 + ++a;

cout<<"Isi b = "<<b<<'\n';

*//variabel b dijumlahkan dengan 10 baru b dinaikkan 1*

c = 10 + b++;

cout<<"Isi b = "<<b<<'\n';

cout<<"Isi c = "<<c<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ Operator Penaikan dan Penurunan](https://1.bp.blogspot.com/-8_GAaI_BykM/Vy-EI8w156I/AAAAAAAAKrY/SF9ZMk29BnQTI6lkccK1mzu2uJfv0LEwgCLcB/s1600/operator+penaikan+penurunan.PNG) |
| **C++ Operator Penaikan dan Penurunan** |

**Operator bitwise**Digunakan untuk memanipulasi bilangan biner

|  |  |  |
| --- | --- | --- |
| **Operator** | **Keterangan** | **Contoh** |
| << | Geser bit ke kiri | 10 << 3 |
| >> | Geser bit ke kanan | 10 >> 3 |
| & | Bitwise and | 10 & 3 |
| | | Bitwise or | 10 | 3 |
| ^ | Bitwise xor | 10 ^ 3 |
| ~ | Bitwise not | ~ 10 |

**Prioritas operator bitwise**

|  |  |
| --- | --- |
| **Prioritas** | **Operator** |
| Tinggi | ~ |
| Normal | <<  >> |
| & |
| ^ |
| Rendah | | |

**Contoh program :**

#include<iostream.h>

#include<conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan operator bitwise*

*\*/*

**void** main()

{

**unsigned** **long** a,b,x;

a = 50;

b = 3;

x = a << b;

cout<<"Hasil "<<a<<" << "<<b<<" = "<<x<<'\n';

x = a >> b;

cout<<"Hasil "<<a<<" >> "<<b<<" = "<<x<<'\n';

x = a & b;

cout<<"Hasil "<<a<<" & "<<b<<" = "<<x<<'\n';

x = a | b;

cout<<"Hasil "<<a<<" | "<<b<<" = "<<x<<'\n';

x = a ^ b;

cout<<"Hasil "<<a<<" ^ "<<b<<" = "<<x<<'\n';

x = ~a;

cout<<"Hasil ~"<<a<<" = "<<x<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ Operator Bitwise](https://2.bp.blogspot.com/-WGvE50vx2ig/Vy-LyqKyQJI/AAAAAAAAKro/cQSujn79rXUT5wgvLdj78yyE-Xp8SbgVgCLcB/s1600/penggunaan+operator+bitwise.PNG) |
| **C++ Operator Bitwise** |

**Operator majemuk**  
Operator majemuk digunakan untuk memendekkan penulisan suatu penugasan.  
x = x + 5;  
y = y – 8;  
  
Dapat ditulis :  
x += 5;  
y -= 8;

**Contoh program :**

#include<iostream.h>

#include<conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan operator majemuk*

*\*/*

**void** main()

{

**int** a;

a = 20;

cout<<"Nilai a sekarang = "<<a<<'\n';

a += 5;

cout<<"Nilai a += 5 sekarang = "<<a<<'\n';

a -= 3;

cout<<"Nilai a -= 3 sekarang = "<<a<<'\n';

a \*= 2;

cout<<"Nilai a \*= 2 sekarang = "<<a<<'\n';

a %= 3;

cout<<"Nilai a %= 3 sekarang = "<<a<<'\n';

a <<= 1; *//a = a << 1;*

cout<<"Nilai a <<= 1 sekarang = "<<a<<'\n';

a >>= 1; *//a = a >> 1;*

cout<<"Nilai a >>= 1 sekarang = "<<a<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ Operator Majemuk](https://2.bp.blogspot.com/-DUoDoqAboiM/Vy-NUPPMyEI/AAAAAAAAKr0/czKwqlaKn8o5O4NqheRzSs3enf6GLxoDACLcB/s1600/operator+majemuk.PNG) |
| **C++ Operator Majemuk** |

**Operator Relasi**

Operator relasi digunakan untuk membandingkan dua buah nilai, yang hasilnya berupa nilai 1 jika benar dan nalai 0 jika salah.

|  |  |
| --- | --- |
| **Operator** | **Keterangan** |
| == | Sama dengan (bukan penugasan) |
| != | Tidak sama dengan |
| < | Lebih kecil |
| > | Lebih besar |
| <= | Lebih kecil atau sama dengan |
| >= | Lebih besar atau sama dengan |

**Contoh program :**

#include<iostream.h>

#include<conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan operator relasi*

*\*/*

**void** main()

{

**int** a,b,hasil;

a = 45;

b = 25;

hasil = a == b;

cout<<"Hasil relasi "<<a<<" == "<<b<<" = "<<hasil<<'\n';

hasil = a != b;

cout<<"Hasil relasi "<<a<<" != "<<b<<" = "<<hasil<<'\n';

hasil = a > b;

cout<<"Hasil relasi "<<a<<" > "<<b<<" = "<<hasil<<'\n';

hasil = a < b;

cout<<"Hasil relasi "<<a<<" < "<<b<<" = "<<hasil<<'\n';

hasil = a >= b;

cout<<"Hasil relasi "<<a<<" >= "<<b<<" = "<<hasil<<'\n';

hasil = a <= b;

cout<<"Hasil relasi "<<a<<" <= "<<b<<" = "<<hasil<<'\n';

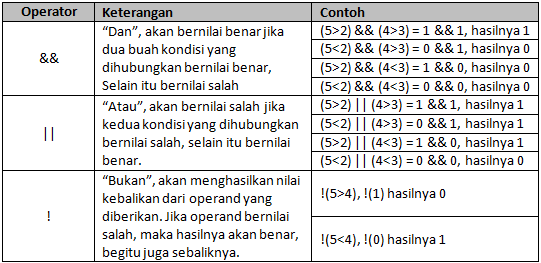
getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ Operator Relasi](https://2.bp.blogspot.com/-MHmWYu3hAlg/VzBdfx3SJYI/AAAAAAAAKsI/qGKXdopO1yo5zRz9B6naJ9h-z9StS-9qQCLcB/s1600/op+relasi.PNG) |
| **C++ Operator Relasi** |

**Operator Logika**

Operator logika digunakan untuk menghubungkan dua buah kondisi menjadi sebuah kondisi.

[](https://1.bp.blogspot.com/-pXyD_tQvhbc/VzBqp-d2mnI/AAAAAAAAKsY/Tum3Qk6aTbkzDgBhRfwscZ2QtQa7DixRwCLcB/s1600/operator+logika.PNG)

**Contoh program :**

#include<iostream.h>

#include<conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan operator logika*

*\*/*

**void** main()

{

**int** hasil;

cout<<"Operator logika && (dan)"<<'\n';

hasil = (5 > 2) && (4 > 3);

cout<<"Hasil logika (5 > 2) && (4 > 3)= "<<hasil<<'\n';

hasil = (5 < 2) && (4 > 3);

cout<<"Hasil logika (5 < 2) && (4 > 3)= "<<hasil<<'\n';

hasil = (5 > 2) && (4 < 3);

cout<<"Hasil logika (5 > 2) && (4 < 3)= "<<hasil<<'\n';

hasil = (5 < 2) && (4 < 3);

cout<<"Hasil logika (5 > 3) && (4 > 3)= "<<hasil<<'\n';

cout<<"\nOperator logika || (atau)"<<'\n';

hasil = (5 > 2) || (4 > 3);

cout<<"Hasil logika (5 > 2) || (4 > 3)= "<<hasil<<'\n';

hasil = (5 < 2) || (4 > 3);

cout<<"Hasil logika (5 < 2) || (4 > 3)= "<<hasil<<'\n';

hasil = (5 > 2) || (4 < 3);

cout<<"Hasil logika (5 > 2) || (4 < 3)= "<<hasil<<'\n';

hasil = (5 < 2) || (4 < 3);

cout<<"Hasil logika (5 > 3) || (4 > 3)= "<<hasil<<'\n';

cout<<"\nOperator logika !"<<'\n';

hasil = !(5 > 2);

cout<<"Hasil logika !(5 > 2) = "<<hasil<<'\n';

hasil = !(5 < 2);

cout<<"Hasil logika !(5 < 2) = "<<hasil<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++  Operator Logika](https://2.bp.blogspot.com/-ao3ohjmuIl0/VzBuqUgwkaI/AAAAAAAAKsk/WNEGzIyWVLkVsfOIDJMnlumIzTPZSa0mQCLcB/s1600/hasil+op+logika.PNG) |
| **C++  Operator Logika** |

**Operator Kondisi**

Operator kondisi digunakan untuk mendapatkan sebuah nilai dari dua kemungkinan berdasarkan suatu kondisi. Bentuk operator kondisi :

*Ungkapan1?ungkapan2:ungkapan3*  
Jika ungkapan1 diuji bernilai benar maka hasilnya adalah ungkapan2, jika salah maka hasilnya adalah ungkapan3.

**Contoh program:**

#include<iostream.h>

#include<conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan operator kondisi*

*\*/*

**void** main()

{

**int** angka1, angka2, maks;

angka1 = 70;

angka2 = 90;

maks = (angka1 > angka2)?angka1:angka2;

cout<<"Angka terbesar = "<<maks<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ Operator Kondisi](https://4.bp.blogspot.com/-qyMxnu0spNs/VzBv7oKsZCI/AAAAAAAAKsw/NE1TDcLS9-QLP9f8oObcrdqdveeOxDeBwCLcB/s1600/op+kondisi.PNG) |
| **C++ Operator Kondisi** |

**3.5 Input dalam C++**

1. cout (dibaca C out)

Merupakan obyek dalam C++ yang berfungsi untuk menampilkan data ke standar output (layar). Objek cout sudah banyak digunakan pada contoh-contoh didepan yang digunakan untuk menampilkan hasil ke layar.  
Bentuk :  
cout << var;

1. cin (dibaca C in)  
   Merupakan obyek dalam C++ yang berfungsi untuk membaca data dari keyboard.  
   Bentuk :  
   cin >> var;

Berfungsi membaca data dari keyboard dan memasukkan dalam variabel bernama var.

**Contoh program :**

#include<iostream.h>

#include<conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan cin*

*\*/*

**void** main()

{

**int** angka;

**char** huruf;

cout<<"Masukkan angka : ";

cin>>angka;

cout<<"Masukkan sebuah huruf : ";

cin>>huruf;

cout<<"\nAngka yang anda masukkan : "<<angka<<'\n';

cout<<"Huruf yang anda masukkan : "<<huruf<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ Cin](https://4.bp.blogspot.com/-VVAcb4aTKGY/VzBxXubfgCI/AAAAAAAAKs8/AoiqfY6sV_sbtbE8wPDT3rmFi8WofiFlwCLcB/s1600/cin.PNG) |
| C++ Cin |

cin >> tidak dapat digunakan untuk membaca karakter Spasi atau Tab, jika akan dibaca tombol tersebut dunakan fungsi getch().  
**3. Fungsi getch() dan getche()**   
Fungsi getch() dan getche() digunakan untuk membaca tanpa perlu menekan enter, selain itu fungsi ini juga dapat membaca tombol Spasi dan Enter.  
**Bentuk :**  
karakter = getch();  
karakter = getche();  
Apabila fungsi getch() dan getche() disertakan maka header conio.h perlu disertakan. Fungsi getch() tidak menampilkan karakter dari tombol yang ditekan, sedangkan getche() menampilkan karakter dari tombol yang ditekan.

**Contoh program :**

#include<iostream.h>

#include<conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan getch() dan geche()*

*\*/*

**void** main()

{

**char** kar1, kar2;

cout<<"Tekan sembarang karakater : ";

kar1 = getch();

cout<<"\nTekan sembarang karakater : ";

kar2 = getche();

cout<<"\n\nKarakter pertama : "<<kar1<<'\n';

cout<<"Karakter kedua : "<<kar2<<'\n';

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ getch dan getche](https://1.bp.blogspot.com/-GFZeCtu-DQM/VzBy4niL1mI/AAAAAAAAKtI/fAbZGGO3nIc4r1CywVRxmGDYuc7DRxpTACLcB/s1600/getch-getche.PNG) |
| C++ getch dan getche |

**3.6 Manipulator**

Manipulator digunakan untuk mengatur tampilan data pada layar. Beberapa manipular dalam C++ :  
**1. Manipulator endl**

Manipulator endl digunakan untuk menyisipkan karakter ganti baris (newline). Manipulator ini identik dengan karakter ’\n’.

**Contoh program :**

#include<iostream.h>

#include<conio.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan endl*

*\*/*

**void** main()

{

**int** angka1 = 100;

**int** angka2 = 5000;

**float** total = 12323.8;

cout<<"Isi angka 1 : "<<angka1<<endl;

cout<<"Isi angka 2 : "<<angka2<<endl;

cout<<"Isi angka 2 : "<<total<<endl;

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ endl](https://3.bp.blogspot.com/-eb_0-itDyzc/VzHXzCcXsVI/AAAAAAAAKts/Ue9WgnwphFc20gO41R1Llbb22y_y2EQUgCLcB/s1600/manipulator+endl.PNG) |
| **C++ endl** |

**Manipulator setw()**  
Manipulator setw() digunakan untuk mengatur lebar tampilan data pada layar. Jika anda akan mengatur tampilan data pada layar maka perlu disertakan header iomanip.h.

**Contoh program :**

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan setw()*

*\*/*

**void** main()

{

**int** angka = 1005;

cout<<setw(0)<<angka<<endl;

cout<<setw(4)<<angka<<endl;

cout<<setw(7)<<angka<<endl;

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ setw](https://1.bp.blogspot.com/-adrLwDMtDFw/VzHb26EQnrI/AAAAAAAAKt4/YagGw_m0sdcA6Yami5qt7pxDYKBxTN2IwCLcB/s1600/setw.PNG) |
| **C++ setw** |

**Manipulator setfill()**  
Manipulator setfill() digunakan untuk mengatur karakter yang diisikan pada field yang ditentukan oleh setw() yang tidak digunakan untuk menampilkan data.

**Contoh program :**

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan setw() dan setfill()*

*\*/*

**void** main()

{

**int** angka = 1005;

cout<<setw(0)<<setfill('\*')<<angka<<endl;

cout<<setw(4)<<setfill('\*')<<angka<<endl;

cout<<setw(6)<<setfill('\*')<<angka<<endl;

cout<<setw(8)<<setfill('\*')<<angka<<endl;

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ setfill](https://1.bp.blogspot.com/-zkaflyLqf6c/VzHha0LYIUI/AAAAAAAAKuI/PnaPdrvDAD0PWvoU3E1ZX3sO1H-Q4tA8QCLcB/s1600/setfill.PNG) |
| **C++ setfill** |

**4. Manipulator dec, oct dan hex**  
Manipulator dec, oct dan hex digunakan untuk menampilkan data dalam bentuk desimal (basis 10), oktal (basis 8), dan heksadesimal (basis 16).

**Contoh program :**

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan dec, oct dan hex*

*\*/*

**void** main()

{

**int** x = 150;

cout<<"Nilai x : "<<x<<endl;

cout<<"Nilai x dalam oktal : "<<oct<<x<<endl;

cout<<"Nilai x dalam heksadesimal : "<<hex<<x<<endl;

cout<<"Nilai x dalam desimal : "<<dec<<x<<endl;

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ dec, oct dan hex](https://2.bp.blogspot.com/-hQX9MyBdINU/VzHtL5MHb2I/AAAAAAAAKuY/Jb4DKPUHkGIrSwIb0WpWgzq7IrLpRCaowCLcB/s1600/doc,+oct,+hex.PNG) |
| **C++ dec, oct dan hex** |

**5. Manipulor setiosflags()**Manipulor setiosflags() merupakan manipulator yang dapat dipakai untuk mengontrol sejumlah tanda format sebagai berikut :

|  |  |
| --- | --- |
| **Tanda Format** | **Keterangan** |
| ios::left | Menyetel rata kiri terhadap lebar field yang diatur melalui **setw()** |
| ios::right | Menyetel rata kanan terhadap lebar field yang diatur melalui **setw()** |
| ios::scientifie | Memformat keluaran dalam notasi eksponensial |
| ios::fixed | Memformat keluaran dalam bentuk notasi desimal |
| ios::dec | Memformat keluaran dalam basis 10 (desimal) |
| ios::oct | Memformat keluaran dalam basis 8 (oktal) |
| ios::hex | Memformat keluaran dalam basis 16 (heksadesimal) |
| ios::uppercase | Memformat huruf pada notasi heksadesimal dalam bentuk huruf kapital |
| ios::showbase | Menampilkan awalan 0x untuk bilangan heksadesimal atau 0 (nol) untuk bilangan oktal |
| ios::showpoint | Menampilkan titik desimal pada bilangan pecahan yang tidak memiliki bagian pecahan |
| ios::showpos | Untuk menampilkan tanda + pada bilangan positif |

**Contoh program :**

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan setiosflags()*

*\*/*

**void** main()

{

**int** x = 520;

cout<<"x ditampilkan dengan ios::left "<<endl;

cout<<setiosflags(ios::left)<<setw(10)<<x<<endl<<endl;

cout<<"x ditampilkan dengan ios::right "<<endl;

cout<<setiosflags(ios::right)<<setw(10)<<x<<endl<<endl;

**float** y = 123.456;

cout<<"y ditampilkan dengan ios::fixed "<<endl;

cout<<setiosflags(ios::fixed)<<y<<endl<<endl;

cout<<resetiosflags(ios::fixed);

cout<<"y ditampilkan dengan ios::scientific "<<endl;

cout<<setiosflags(ios::scientific)<<y<<endl<<endl;

cout<<resetiosflags(ios::scientific);

**int** bil = 51;

cout<<"Tanpa ios::showbase : "<<endl;

cout<<oct<<bil<<endl;

cout<<dec<<bil<<endl;

cout<<hex<<bil<<endl;

cout<<"Dengan ios::showbase : "<<endl;

cout<<setiosflags(ios::showbase);

cout<<oct<<bil<<endl;

cout<<hex<<bil<<endl;

cout<<dec<<bil<<endl;

cout<<"\nDengan ios::uppercase untuk heksadesimal : "<<endl;

cout<<setiosflags(ios::uppercase)<<bil<<endl;

cout<<resetiosflags(ios::showbase);

cout<<resetiosflags(ios::uppercase);

**float** a = 234.00;

cout<<"\nTanpa ios::showpoint : "<<endl;

cout<<a<<endl<<endl;

cout<<"Dengan ios::showpoint : "<<endl;

cout<<setiosflags(ios::showpoint)<<a<<endl;

cout<<resetiosflags(ios::showpoint);

**int** b = 27;

cout<<"\nTanpa ios::showpos : "<<endl;

cout<<b<<endl<<endl;

cout<<"Dengan ios::showpos : "<<endl;

cout<<setiosflags(ios::showpos)<<b<<endl;

cout<<resetiosflags(ios::showpos);

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [setiosflags()](https://1.bp.blogspot.com/-OprKl18xc6g/VzHy21HBcaI/AAAAAAAAKuo/eC4g2tXiQpAFOR6vjGXZV6HtaipVdKTGwCLcB/s1600/setiosflags.PNG) |
| **C++ setiosflags()** |

**6. Manipulator resetiosflags()**Manipulator resetiosflags() digunakan untuk mengembalikan ke bentuk format awal setelah adanya penggunaan manipulator, misalnya :  
setiosflags(ios::left)  
telah digunakan , maka untuk kembali ke bentuk awal bisa digunakan :  
resetiosflags(ios::left)

**Contoh program :**

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan resetiosflags()*

*\*/*

**void** main()

{

**int** x = 12340;

cout<<"Ditampilkan dengan ios::right "<<endl;

cout<<setiosflags(ios::left)<<setw(10)<<x<<endl<<endl;

cout<<"Setelah dilakukan resetiosflags(ios::right)"<<endl;

cout<<resetiosflags(ios::left);

cout<<setw(10)<<x<<endl<<endl;

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ resetiosflags()](https://1.bp.blogspot.com/-Y0jdbIg9q04/VzH3U3p6HNI/AAAAAAAAKu0/SfGcJvzOUjcJC024qbLWl7rdD_sVypxcwCLcB/s1600/resetiosflag.PNG) |
| **C++ resetiosflags()** |

**7. Manipulator setprecision()**

Manipulator setprecision() digunakan untuk mengatur jumlah digit pecahan yang akan ditampilkan pada bilangan pecahan.

Bentuk :

setprecision(n)

Dengan n adalah jumlah digit yang ingin ditampilkan

**Contoh program:**

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

*/\*\**

*\*gatewan.com*

*\*Wawan Beneran*

*\*Penggunaan setprecision()*

*\*/*

**void** main()

{

**float** y = 12.340;

cout<<setiosflags(ios::fixed);

cout<<setprecision(0)<<y<<endl;

cout<<setprecision(1)<<y<<endl;

cout<<setprecision(2)<<y<<endl;

cout<<setprecision(3)<<y<<endl;

cout<<setprecision(4)<<y<<endl;

cout<<setprecision(5)<<y<<endl;

getch();

}

|  |
| --- |
| **Hasil eksekusi :**  [C++ setprecision()](https://3.bp.blogspot.com/-loGOD5tEQrw/VzH7BqEBWVI/AAAAAAAAKvA/dpUp0v6CPrksvtuF2OiBbQmKz9J04wPOwCLcB/s1600/setprecision.PNG) |
| C++ setprecision() |