

Struktur Program C++

Bahasa C++ merupakan bahasa yang terstruktur, sehingga struktur penulisan program dalam bahasa c++ dirangkai dari kumpulan sejumlah fungsi, setidaknya memiliki sebuah fungsi utama, yang bernama main ().

Struktur Bahasa C++

Preprocessor directive : perintah dalam bahasa c++ yang akan membuat dan mengkompilasi file perantara dengan program utama

Deklarasi global : merupakan bagian program c++ tempat pendeklarasian konstanta, variabel, fungsi atau header fungsi, tipe data baru, atau struktur data yang memiliki sifat global

Fungsi utama : fungsi utama yang akan dieksekusi oleh compiler c++. dan compiler c++ hanya akan mengeksekusi fungsi-fungsi lain yang dipanggil pada fungsi utama atau yang memiliki hubungan dengan fungsi utama secara tidak langsung,

Fungsi : tempat pendefinisian header fungsi yang telah di deklarasikan pada bagian deklarasi global.

Setiap fungsi didefinisikan dengan diawali dengan nama -fungsi dan diikuti dengan memuat di dalamnya perintah /instruksi berupa statement-statement untuk melakukan pekerjaan pengolahan data yang dapat diberikan kepadanya, ditulis dengan menggunakan tanda awal { dan diakhiri dengan }

Berikut merupakan contoh sederhana pemrograman dengan bahasa C++

```
#include<file-include>
main()
{
    pernyataan;
}
```

Contoh-1

```
#include <conio.h>

main()
{
    int a = 7;
    char b = 'G';
    clrscr();

    printf("%c Merupakan Abjad Yang Ke - %d", b, a);
}
```

Pengenalan Tipe Data

Penjelasan C++ memiliki 7 tipe data dasar

Tabel 2.1. Tipe Data

Tipe Data	Ukuran Memori	Jangkauan Nilai	Format	Jumlah Digit
Char	1 Byte	-128 s.d 127		
Int	2 Byte	-32768 s.d 32767		
Short	2 Byte	-32768 s.d 32767		
Long	4 Byte	-2,147,435,648 s.d 2,147,435,647		
Float	4 Byte	3.4 x 10 ⁻³⁸ s.d 3.4 x 10 ⁺³⁸		5 – 7
Double	8 Byte	1.7 x 10 ⁻³⁰⁸ s.d 1.7 x 10 ⁺³⁰⁸		15 – 16
Long Double	10 Byte	3.4 x 10 ⁻⁴⁹³² s.d 1.1 x 10 ⁺⁴⁹³²		19

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh computer. Misalnya saja 5 dibagi 2 bisa saja menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe integer maka akan menghasilkan nilai 2, namun jika keduanya bertipe float maka akan menghasilkan nilai 2.5000000. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.

Dalam bahasa C terdapat lima tipe data dasar, yaitu :

No	Tipe Data	Ukuran	Range	Format	Keterangan
1	char	1 byte	-128 s/d 127	%c	Karakter/string
2	int	2 byte	- 32768 s/d 32767	%i , %d	Integer/bilangan bulat
3	float	4 byte	- 3.4E-38 s/d 3.4E+38	%f	Float/bilangan pecahan
4	double	8 byte	- 1.7E-308 s/d 1.7+308	%lf	Pecahan presisi ganda
5	void	0 byte	-	-	Tidak bertipe
6	String	-	-	%s	String

Deklarasi Konstanta

Penjelasan Bentuk deklarasi konstanta diawali dengan reserved word **const**.

Bentuk penulisannya :

```
const nama-konstanta = nilai konstanta;
```

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Konstanta nilainya selalu tetap. Konstanta harus didefinisikan terlebih dahulu di awal program. Konstanta dapat bernilai integer, pecahan, karakter dan string. Contoh konstanta : 50; 13; 3.14; 4.50005; 'A'; 'Bahasa C'. Selain itu, bahasa C juga menyediakan beberapa karakter khusus yang disebut karakter escape, antara lain :

\a : untuk bunyi bell (alert)
\b : mundur satu spasi (backspace)
\f : ganti halaman (form feed)
\n : ganti baris baru (new line)
\r : ke kolom pertama, baris yang sama (carriage return)
\v : tabulasi vertical
\0 : nilai kosong (null)
\' : karakter petik tunggal
\" : karakter petik ganda
\\ : karakter garis miring

Variabel

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variable bisa diubah-ubah sesuai kebutuhan.

Nama dari suatu variable dapat ditentukan sendiri oleh pemrogram dengan aturan sebagai berikut :

- **Terdiri dari gabungan huruf dan angka** dengan karakter pertama harus berupa huruf. Bahasa C bersifat case-sensitive artinya huruf besar dan kecil dianggap berbeda. Jadi antara **nim**, **NIM** dan **Nim** dianggap berbeda.
- **Tidak boleh mengandung spasi.**
- **Tidak boleh mengandung symbol-simbol khusus**, kecuali garis bawah (underscore). Yang termasuk symbol khusus yang tidak diperbolehkan antara lain : \$, ?, %, #, !, &, *, (,), -, +, = dsb
- **Panjangnya bebas**, tetapi **hanya 32** karakter pertama yang terpakai.

Contoh penamaan variabel yang benar : NIM, a, x, nama_mhs, f3098, f4, nilai, budi, dsb.

Contoh penamaan variable yang salah : %nilai_mahasiswa, 80mahasiswa, rata-rata, ada spasi, penting!, dsb
variabel, dibagi menjadi dua jenis kelompok, yaitu :

- **Variabel Numerik**
- **Variabel Teks**

Variabel Numerik

Variabel numerik ini dibagi menjadi menjadi 3 (tiga) macam :

- Bilangan **Bulat** atau **Integer**
- Bilangan **Desimal Berpresisi Tunggal** atau **Floating Point**.
- Bilangan **Desimal Berpresisi Ganda** atau **Double Precision**.

Variabel Text

- Character (Karakter Tunggal)
- String (Untuk Rangkaian Karakter)

Deklarasi Variabel

Penjelasan

Adalah proses memperkenalkan variabel kepada C++ dan pendeklarasian tersebut bersifat mutlak karena jika tidak diperkenalkan terlebih dulu maka Borland C++ tidak menerima variabel tersebut.

Deklarasi Variabel ini meliputi tipe variabel, seperti : integer atau character dan nama variabel itu sendiri. Setiap kali pendeklarasian variabel harus diakhiri oleh tanda titik koma (;).

Tabel 2.3. Deklarasi Variabel

TIPE VARIABEL	SIMBOL DEKLARASI
Integer	int
Floating Point	float
Double Precision	double
Karakter	char
Unsigned Integer	unsigned int
Unsigned Character	unsigned char
Long Integer	long int
Unsigned Long Integer	unsigned long int

Bentuk penulisannya :

```
Tipe data nama variabel;
```

Contoh Deklarasi

```
int nama_mahasiswa;  
char grade;  
float rata_rata ;  
int nilai;
```

KODE PENENTU FORMAT

- . %c : Membaca sebuah karakter
- . %s : Membaca sebuah string
- . %i, %d : Membaca sebuah bilangan bulat (integer)
- . %f, %e : Membaca sebuah bilangan pecahan (real)
- . %o : membaca sebuah bilangan octal
- . %x : Membaca sebuah bilangan heksadesimal
- . %u : Membaca sebuah bilangan tak bertanda.

Perintah Keluaran

```
#include <stdio.h> //untuk mengaktifkan perintah printf dan scanf  
#include <conio.h> //untuk mengaktifkan perintah getch()
```

Penjelasan

Perintah standar output yang disediakan oleh C++, diantaranya adalah :

- `printf()`
- `puts()`
- `putchar()`
- `cout()`

printf()

Penjelasan

Fungsi *printf()* merupakan fungsi keluaran yang paling umum digunakan untuk menampilkan informasi kelayar.

Bentuk Penulisan

```
printf("string-kontrol", argumen-1, argumen-2, ...);
```

String-Kontrol dapat berupa keterangan yang akan ditampilkan pada layar beserta penentu format. Penentu format dipakai untuk memberi tahu kompilernya mengenai jenis data yang dipakai dan akan ditampilkan. Argumen ini dapat berupa variabel, konstanta dan ungkapan.

Tabel 2.4. Penentu Format Printf()

TIPE DATA	Penentu Format Untuk <i>printf()</i>
Integer	%d
Floating Point	
Bentuk Desimal	%f
Bentuk Berpangkat	%e
Bentuk Desimal dan Pangkat	%g
Double Precision	%lf
Character	%c
String	%s
Unsigned Integer	%u
Long Integer	%ld
Long Unsigned Integer	%lu
Unsigned Hexadecimal Integer	%x
Unsigned Octal Integer	%o

```
printf("%c merupakan abjad yang ke - %d", 'b', 2);
```

b typenya char

Contoh-1

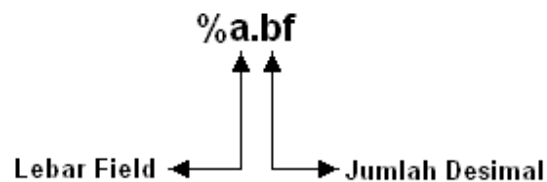
```
#include <stdio.h>
#include <conio.h>

main()
{
    int a = 7;
    char b = 'G';
    clrscr();

    printf("%c Merupakan Abjad Yang Ke - %d", b, a);
}
```

Penggunaan Penentu Lebar Field

Penjelasan Bila ingin mencetak atau menampilkan data yang bertipe data FLOAT atau pecahan, tampilan yang tampak biasanya kurang bagus. Hal tersebut dapat diatur lebar field-nya dan jumlah desimal yang ingin dicetak. Berikut bentuk penulisannya :



Contoh-2

```
#include <stdio.h>
#include <conio.h>

main()
{
    float a = 7.50, b = 243.21;
    clrscr();
    printf("Bilangan A = %f \n", a);
    printf("Bilangan B = %f", b);
}
```

Output yang akan dihasilkan, jika tidak menggunakan panentu lebar field adalah

```
Bilangan A = 7.500000
Bilangan B = 243.210007
```

Contoh-3

```
#include <stdio.h>
#include <conio.h>
main()
{
    float a = 7.50, b = 243.21;
    clrscr();
    printf("Bilangan A = %4.1f \n", a);
    printf("Bilangan B = %4.1f", b);
}
```

Output yang akan dihasilkan, jika menggunakan panentu lebar field adalah

Bilangan A = 7.5
Bilangan B = 243.2

puts()

Penjelasan

Perintah **puts()** sebenarnya sama dengan **printf()**, yaitu digunakan untuk **mencetak string** ke layar. **puts()** berasal dari kata **PUT STRING**.

Perbedaan antara **printf()** dengan **puts()** adalah :

Tabel 2.6. Perbedaan fungsi puts() dengan printf()

printf()	puts()
Harus menentukan tipe data untuk data string, yaitu %s	Tidak Perlu penentu tipe data string, karena fungsi ini khusus untuk tipe data string.
Untuk mencetak pindah baris, memerlukan notasi '\n'	Untuk mencetak pindah baris tidak perlu notasi '\n', karena sudah dibeikan secara otomatis.

Contoh-4

```
#include <stdio.h>
#include <conio.h>

main()
{
    char a[4] = "TEUB";
    clrscr();
    puts("Saya Kuliah di. ");
    puts(a);
}
```

putchar()

Penjelasan

Perintah **putchar()** digunakan untuk menampilkan **sebuah karakter ke layar**. Penampilan karakter tidak diakhiri dengan pindah baris.

Contoh-5

```
#include <stdio.h>
#include <conio.h>

main()
{
    clrscr();
    putchar('T');
    putchar('E');
    putchar('U');
    putchar('b');
}
```

cout()

Penjelasan

Fungsi **cout()** merupakan sebuah object didalam C++ digunakan untuk menampilkan suatu data kelayar. Untuk menggunakan fungsi cout() ini, harus menyertakan file header **iostream.h** .

Contoh-6

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>

main()
{
    float a, b, c;

    a=7.5; b=8.4; c=0
    clrscr();
    cout<<"Masukan Nilai A : "<<a;
    cout<<"Masukan Nilai B : "<<b;

    c = a + b;

    cout<<"Masukan Nilai C : "<<c;
    getch();
}
```

Fungsi Manipulator

Penjelasan

Manipulator pada umumnya digunakan untuk mengatur tampilan layar, untuk mengguakan manipulator ini file header yang harus disertakan file header **omanip.h** . Ada beberapa fungsi manipulator yang disediakan oleh C++, antara lain.

- endl
- end
- flush()
- dec()
- hex()
- oct()
- setbase()
- setw()
- setfill()
- setprecision()
- setosflags()

Berikut akan dibahas beberapa fungsi manipulator, diantaranya :

a. endl

Penjelasan

endl merupakan suatu fungsi manipulator yang digunakan untuk menyisipkan karakter NewLine atau mengatur pindah baris. Fungsi ini sangat berguna untuk piranti keluaran berupa file di disk. File header yang harus disertakan adalah file header **iostream.h** .

Contoh-7

```
# include <stdio.h>
# include <conio.h>
# include <iostream.h>
main()
```



```

{
    float a, b, c;

    a=7.5; b=8.4; c=0
    clrscr();
    cout<<"Masukan Nilai A : "<<a<<endl;
    cout<<"Masukan Nilai B : "<<b<<endl;

    c = a + b;

    cout<<"Masukan Nilai C : "<<c<<endl;
    getch();
}

```

b. ends

Penjelasan

ends merupakan suatu fungsi manipulator yang digunakan untuk menambah karakter null (nilai ASCII NOL) keteretan suatu karakter. Fungsi ini akan berguna untuk mengirim sejumlah karakter kefile didisk atau modem dan mangakhirinya dengan karakter NULL.. File header yang harus disertakan adalah file header **iostream.h** .

Contoh-8

```

# include <stdio.h>
# include <conio.h>
# include <iostream.h>

main()
{
    int a, b, c, d;
    clrscr();
    cout<<"Masukan Nilai A : "; cin>>a;
    cout<<"Masukan Nilai B : "; cin>>b;

    c = a % b;
    d = a * b;

    cout<<"Hasil dari C = A % B adalah "<<c<<ends;
    cout<<"Hasil dari D = A * B adalah "<<d<<ends;
    getch();
}

```

c. dec, oct dan hex

Penjelasan

dec, oct dan hex merupakan suatu fungsi manipulator yang digunakan untuk menampilkan data dalam bentuk desimal, oktal dan hexadesimal. File header yang harus disertakan adalah file header **iomanip.h** .

Contoh-9

```

# include <stdio.h>
# include <conio.h>
# include <iostream.h>
# include <iomanip.h>
void main()

```

```

{
  int nilai = 10;
  clrscr();

  cout<<"Nilai = "<<nilai<<endl;
  cout<<"Nilai ke Octal = "<<oct<<nilai<<endl;
  cout<<"Nilai ke Hexadesimal = "<<hex<<nilai<<endl;
  cout<<"Nilai ke Desimal = "<<dec<<nilai<<endl;

  getch();
}

```

d. setprecision ()

Penjelasan

Fungsi **setprecision()** merupakan suatu fungsi manipulator yang digunakan untuk mengatur jumlah digit desimal yang ingin ditampilkan. Fungsi ini biasa pada fungsi **cout()**, file header yang harus disertakan adalah file header **iomanip.h**.

Contoh-10

```

# include <stdio.h>
# include <conio.h>
# include <iostream.h>
# include <iomanip.h>

void main()
{
  float a, b, c;
  a = 25.77;
  b = 23.45;
  clrscr();

  c = a * b;

  cout<<setiosflags(ios::fixed);
  cout<<setprecision(1)<<c<<endl;
  cout<<setprecision(2)<<c<<endl;
  cout<<setprecision(3)<<c<<endl;
  cout<<setprecision(4)<<c<<endl;
  cout<<setprecision(5)<<c<<endl;
  getch();
}

```

Perintah Masukan

Penjelasan

- `scanf()`
- `gets()`
- `cout()`
- `getch`
- `getche()`

Perintah standar input yang disediakan oleh C++, diantaranya adalah :

scanf()

Penjelasan

Fungsi ***scanf()*** digunakan untuk memasukkan **berbagai jenis data**. Bentuk Umum dari fungsi ini adalah :

```
scanf("penentu format", &nama-variabel);
```

Keterangan

simbol **&** merupakan pointer yang digunakan untuk menunjuk kealamat variabel memori yang dituju.

Tabel 2.7. Penentu Format scanf()

TIPE DATA	Penentu Format Untuk <i>scanf()</i>
Integer	%d
Floating Point	
Bentuk Desimal	%e atau %f
Bentuk Berpangkat	%e atau %f
Double Precision	%lf
Character	%c
String	%s
Unsigned Integer	%u
Long Integer	%ld
Long Unsigned Integer	%lu
Unsigned Hexadecimal Integer	%x
Unsigned Octal Integer	%o

Contoh-11

```
# include <stdio.h>
# include <conio.h>
main()
{
    int a, b, c = 0 ;
    clrscr();
    printf("Masukan Nilai A = "); scanf("%d",&a);
    printf("Masukan Nilai B = "); scanf("%d",&b);

    c = a + b;
    printf("Hasil Penjumlahan = %d",c);
}
```

gets()

Penjelasan

Fungsi **gets()** digunakan untuk memasukkan **data string**. Bentuk Umum dari fungsi ini adalah :

```
gets(nama-variabel-array);
```

Perbedaan antara **scanf()** dengan **gets()** adalah :

Tabel 2.8. Perbedaan scanf() dengan gets()

scanf()	gets()
Tidak dapat menerima string yang mengandung spasi atau tab dan dianggap sebagai data terpisah	Dapat menerima string yang mengandung spasi atau tab dan masing dianggap sebagai satu kesatuan data.

Contoh-12

```
# include <stdio.h>
# include <conio.h>

main()
{
    char nm1[20];
    char nm2[20];

    clrscr();

    puts("Masukan nama ke - 1 = ");
    gets(nm1);
    printf("Masukan nama ke - 2 = ");
    scanf("%s",&nm2);

    printf("\n\n");

    puts("Senang Berkenalan Dengan Anda ..");
    puts(nm1);
    printf("Senang Berkenalan Dangan Anda ..%s", nm1);
    puts("Senang Berkenalan Dangan Anda ..");
    puts(nm1);

    printf("\n\n");

    puts("Senang Berkenalan Dangan Anda ..");
    puts(nm2);
    printf("Senang Berkenalan Dangan Anda ..%s", nm2);
}
```

cin ()

Penjelasan

Fungsi **cin()** merupakan sebuah object didalam C++ digunakan untuk memasukkan suatu data. Untuk menggunakan fungsi cin() ini, harus menyertakan file header **iostream.h** .

Contoh-13

```
# include <stdio.h>
# include <conio.h>
# include <iostream.h>

main()
{
    float a, b, c;

    clrscr();
    cout<<"Masukan Nilai A : ";
    cin>>a;
    cout<<"Masukan Nilai B : ";
    cin>>b;

    c = a + b;

    cout<<"Masukan Nilai C : "<<c<<endl;
}
```

getch ()

Penjelasan

Fungsi **getch()** (*get character and echo*) dipakai untuk membaca sebuah karakter dengan sifat karakter yang dimasukkan tidak perlu diakhiri dengan menekan tombol ENTER, dan karakter yang dimasukan tidak akan ditampilkan di layar. **File header** yang harus disertakan adalah **conio.h**.

Contoh-14

```
# include <stdio.h>
# include <conio.h>

main()
{
    char kar;

    clrscr();

    printf("Masukan Sebuah Karakter Bebas = ");
    kar = getch();
    printf("\nTadi Anda Memasukan karakter %c", kar);
    getch();
}
```

getche()

Penjelasan

Fungsi **getche()** dipakai untuk membaca sebuah karakter dengan sifat karakter yang dimasukkan tidak perlu diakhiri dengan menekan tombol ENTER, dan karakter yang dimasukan ditampilkan di layar. File header yang harus disertakan adalah **conio.h**.

Contoh-15

```
# include <stdio.h>
# include <conio.h>

main()
{
    char kar;

    clrscr();

    printf("Masukan Sebuah Karakter Bebas = ");
    kar = getche();
    printf("\nTadi Anda Memasukan karakter %c", kar);
    getch ();
}
```

Selain itu kedua fungsi ini dapat digunakan untuk menahan agar tidak langsung balik kembali kedalam listing program dan hasil dari program yang di eksekusi dapat dilihat tanpa menekan tombol ALT – F5. Karena fungsi *getch()* merupakan fungsi masukan, jadi sebelum program keluar harus menginputkan satu buah karakter.

Operator Aritmatika

Penjelasan

Operator untuk operasi aritmatika yang tergolong sebagai operator binary adalah :

Tabel 3.1. Operator Aritmatika

Operator	Keterangan	Contoh
*	Perkalian	4 * 5
/	Pembagian	8 / 2
%	Sisa Pembagian	5 % 2
+	Penjumlahan	7 + 2
-	Pengurangan	6 - 2

Operator yang tergolong sebagai operator **Unary**, adalah :

Tabel 3.2. Operator Unary

Operator	Keterangan	Contoh
+	Tanda Plus	-4
-	Tanda Minus	+6

Contoh-1

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
main()
{
    int a, b, c = 0, d = 0;
```

```
clrscr();
cout<<"Masukan Nilai A : "; cin>>a;
cout<<"Masukan Nilai B : "; cin>>b;
c = a % b;
d = a * b;
cout<<" Hasil dari C = A % B = "<<c<<endl;
cout<<" Hasil dari D = A * B = "<<d<<endl;
getch();
}
```