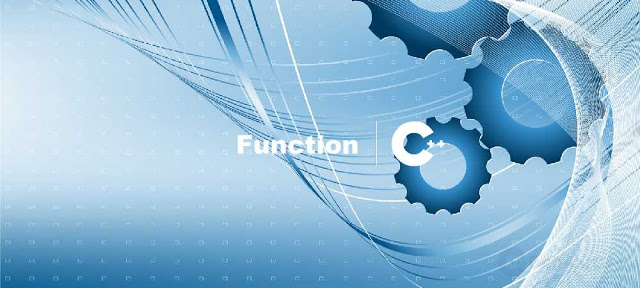
[](https://4.bp.blogspot.com/-slQOIPpZeVs/VrWrh9fuXxI/AAAAAAAADFE/4kSyRsoWuFs/s1600/function.jpg)

Pengertian Function

Fungsi (Function, dalam bahasa inggris), merupakan blok dari

**Pengertian Function**

Fungsi (Function) merupakan blok dari kode yang dirancang untuk melaksanakan tugas khusus Atau Satu blok instruksi yang dieksekusi ketika dipanggil dari bagian lain dalam suatu program. Kegunaan dari fungsi ini adalah untuk:

* Mengurangi pengulangan penulisan program yang berulangan atau sama.
* Program menjadi lebih terstruktur, sehingga mudah dipahami dan dapat lebih dikembangkan.

Bentuk Umum Penulisan

Tipedata nama\_fungsi (tipeparamenter,...)

Keterangan

* *Tipedata*adalah tipe data/nilai yang dikembalikan dengan statemen ‘return’. Tipe default nya : ‘int’. Untuk menyatakan fungsi yang tidak mengembalikan nilai balik, dideklarasikan sebagai : ‘void’
* *nama\_fungsi*adalah nama fungsi tersebut
* *tipeparamenter*,... adalah tipe parameter, bila parameter lebih dari satu (1), masing-masing dipisahkan dengan tanda koma ( , ) untuk menyatakan fungsi tanpa parameter dispesifikasikan : ‘void’. Bila tipe parameter tidak dispesifikasikan, defaultnya : ‘void’

Baca juga : [Pengertian dan Cara Pembuatan File Header C++](http://belajarcpplus.blogspot.com/2016/01/pengertian-dan-cara-pembuatan-file-header-cplusplus.html)

Fungsi harus dideklarasikan terlebih dahulu sebelum didefinisikan. Maksudnya adalah memberitahu compiler jumlah dan tipe parameter yang diterima dan nilai balik fungsi (bila ada) agar compiler dapat memeriksa ketepatannya. Definisi fungsi itu sendiri adalah menspesifikasikan tugas fungsi tersebut.

Fungsi-fungsi yang sudah kita kenal sebelumnya adalah fungsi main(), yang bersifat mutlak, karena fungsi ini program akan dimulai, sebagai contoh yang lainnya fungsi printf(), cout() yang mempunyai tugas untuk menampilkan informasi atau data kelayar dan masih banyak lainnya.

**Struktur Function**

Sebuah fungsi sederhana mempunyai bentuk penulisan sebagai berikut:

nama\_fungsi(argumen)

{

… pernyataan / perintah;

… pernyataan / perintah;

… pernyataan / perintah;

}

Keterangan:

* *Nama fungsi*, boleh dituliskan secara bebas dengan ketentuan, tidak menggunakan spasi dan nama-nama fungsi yang mempunyai arti sendiri.
* *Argumen*, diletakan diantara tanda kurung “( )” yang terletak dibelakang nama fungsi. Argumen boleh diisi dengan suatu data atau dibiarkan kosong.
* *Pernyataan / perintah*, diletakan diantara tanda kurung ‘{ }’.

Baca juga : [Pengertian dan Macam-macam Preprocessor Directives C++](http://belajarcpplus.blogspot.co.id/2016/01/pengertian-dan-macam-macam-preprocessor-directives-cplusplus.html)

Pada pemanggilan sebuah fungsi biasanya cukup dengan menuliskan nama fungsinya. Contoh pembuatan fungsi sederhana

Contoh pembuatan fungsi garis()

#include<conio.h>

#include<stdio.h>

#include<iostream.h>

garis( )

{

   printf("\n----------------------\n");

}

main( )

{

   clrscr( );

   garis( ); //memanggil fungsi garis

   cout<<"Belajar Cpp"<<endl;

   garis( ); //memanggil fungsi garis

   getche( );

}

Baca juga :

* [Pengertian Prototipe dan Parameter Function C++](http://belajarcpplus.blogspot.com/2016/01/pengertian-prototipe-dan-parameter-function-cplusplus.html)
* [Pengertian dan Contoh Function Inline C++](http://belajarcpplus.blogspot.com/2016/01/pengertian-dan-contoh-function-inline-cplusplus.html)
* [Pengertian dan Contoh Function Overloading C++](http://belajarcpplus.blogspot.com/2016/01/pengertian-dan-contoh-function-overloading-cplusplus.html)
* [Cara dan Contoh Structure dengan Array pada Function C++](http://belajarcpplus.blogspot.co.id/2016/01/cara-dan-contoh-structure-dengan-array-pada-function-cplusplus.html)

Cukup sekian dari belajarcpp.com semoga artikel ini bisa bermanfaat untuk kita semua. Mohon maaf atas kekuranganya dan Terima Kasih atas dukungan dan kunjungan anda ke BelajarCPP. Have a nice day.

Last update : 6 Februari 2016