**Algoritma dan Pemrograman 1**

**(IF1042)**

**Pertemuan Ke-11**

**Function**

****

**Disusun Oleh :**

**Wilis Kaswidjanti, S.Si.,M.Kom.**

**Jurusan Teknik Informatika**

**Fakultas Teknologi Industri**

**Universitas Pembangunan Nasional “Veteran”**

**Yogyakarta**

**2010Algoritma dan Pemrograman 1 (IF1042)**

**Judul Materi** : Function

**Deskripsi Materi** : Materi ini membahas function dan parameter-parameter yang terlibat.

**Tujuan Instruksional Khusus** :

1. Menjelaskan pemrograman modular
2. Menjelaskan perbedaan procedure dan function
3. Mendeskripsikan parameter pada procedure dan function
4. Membandingkan parameter input dan parameter output pada procedure dan function

**BAB IX**

**Function**

1. **PENDAHULUAN**

Bab ini lanjutan dari bab sebelumnya, yaitu membahas tentang function/fungsi.

**2. PENYAJIAN**

**2.1. Fungsi**

Fungsi adalah :

Sejumlah instruksi yang dikelompokkan menjadi satu, berdiri sendiri, yang berfungsi untuk menyelesaikan suatu pekerjaan tertentu.

Bahasa C++ adalah suatu bahasa yang struktur penulisannya merupakan kumpulan dari fungsi- fungsi.

* Fungsi terpisah denga program utama, namun merupakan bagian dari program utama.
* Blok program yang terdapat membantu program utama dalam menyelesaikan submasalah-submasalahnya.
* Dalam prosedur atau fungsi dimungkinkan adanya prosedur atau fungsi lain (nested)
* Terdiri dari dua (2) , yaitu pustaka (telah disediakan oleh pemrograman) dan dibuat sendiri.
* Nama prosedur atau fungsi dinyatakan dua kali.

Tujuan dari pembuatan fungsi adalah :

1. Program menjadi terstruktur. Program yang besar dapat dipecah menjadi beberapa program yang lebih kecil, yang setiap satu program kecil tersebut mempunyai tugas tertentu.
2. Dapat mengurangi duplikasi kode.
3. Fungsi dapat dipanggil dari program atau fungsi yang lain.

Pertama kali program dijalankan, komputer akan mencari Fungsi main() dan melaksanakan instruksi- instruksi yang ada didalamnya.

Suatu fungsi cukup didefinisikan sekali, tetapi dapat digunakan beberapa kali. Seandainya tubuh fungsi banyak mengandung pernyataan maka pemakaian fungsi dapat menghindari duplikasi kode dan menghemat penulisan program maupun kode dalam memori. Jadi fungsi merupakan semacam program.

Contoh sebuah program yang terdiri dari satu main program dan satu subprogram (fungsi).

Bagian ini disebut main program atau program induk, atau disebut juga fungsi induk main

#include<iostream.h>

**int** hitung(int A, int B);

**void** main ()

{ **int** A, B, T;

A=5; B=2; T=0;

T = hitung(A,B);

cout << T;

}

**int** hitung(**int** A, **int** B)

{ **int** T;

A = A \* 2;

T = A \* B;

return (T);

}

}

instruksi yang berguna untuk memanggil fungsi lain. Fungsi ini dapat kita buat sendiri ataupun fungsi dipustaka program C

 bagian ini memuat fungsi. Fungsi ini fungsi yang kita buat sendiri. Fungsi dapat ditulis diatas atau dibawah fungsi main.

**2.2. Mendeklarasikan Dan Mendefinisikan Fungsi**

Suatu fungsi mempunyai ‘judul’ yang minimal berisi Nama dan Tipe fungsi. Menulis ‘judul’ sebuah Fungsi sebagai awal dari suatu Fungsi disebut men-DEFINISIKAN Fungsi.

Pada umumnya fungsi memerlukan masukan yang dinamakan argument atau parameter. Hasil akhir fungsi akan berupa sebuah nilai ( nilai balik fungsi ).

Adapun bentuk umum definisi sebuah fungsi adalah :

penentu-tipe nama\_fungsi(daftar parameter)

{

deklarasi variable lokal

tubuh fungsi

}

Penentu tipe adalah untuk menentukan tipe keluaran fungsi yang dapat berupa satu tipe data C++ yang berlaku, misalnya char atau int. Default tipe fungsi yang tidak disebutkan dianggap sebagai int.

Bila fungsi ditulis **dibawah fungsi main()**, maka fungsi tersebut  **harus ‘didaftarkan’ atau dideklarasikan terlebih dahulu** sebelum dapat digunakan. Dimana pen-deklarasian ini ditulis sebelum program induk main()

Contoh:

fungsi cetak dideklarasikan terlebih dahulu sebelum fungsi main. Tanda ‘ ; ‘ jika tidak dipakai maka dianggap mendefinisikan fungsi.

#include<iostream.h>

**void** CETAK();

**void** main()

{

CETAK();

}

**void** CETAK()

{

cout << ”Jakarta”;

}

Instruksi meng-call fungsi cetak.

Mendefinisikan fungsi cetak.

Fungsi ini adalah Instruksi untuk mencetak string “Jakarta”

Fungsi memerlukan tipe sesuai dengan tipe nilai yang dikirimnya atau dikembalikan *(return)* ke bagian program atau Fungsi yang memanggilnya.

Fungsi yang tidak mengirimkan nilai balik, tipenya tidak diperlukan sehingga dapat dibuat sebagai ***void*.**

Bila tipe tidak ditulis, bahasa C menganggap Fungsi tersebut menggunakan tipe ***default*** yaitu tipe ***int***.

Fungsi yang terletak sesudah fungsi main() dan tidak di-deklarasi-kan dahulu maka fungsi tersebut tidak dikenal dan akan menyebabkan ***error***.

Fungsi yang terletak sebelum fungsi main(), maka fungsi tersebut tidak perlu di-deklarasi-kan lagi.

Contoh :

#include<iostream.h>

void main()

{

CETAK(); instruksi meng-call fungsi CETAK.

}

void CETAK() men-definisikan fungsi.

{

cout << ”Jakarta”;

} Fungsi CETAK ditulis

dibawah Fungsi main().

Saat dijalankan, program diatas akan *error* karena fungsi CETAK tidak di-deklarasi-kan sebelumnya.

**Prosedur atau Fungsi.**

**Fungsi** digunakan apabila modul program mengembalikan sebuah nilai.

**Prosedur** digunakan bila modul menghasilkan efek netto dari satu atau sekumpulan aksi.

Dalam prakteknya, fungsi dan prosedur sulit dibedakan karena :

* Fungsi dapat dikonversi sebagai prosedur dengan cara menyatakan nilai yang dikembalikan *(return value)* oleh fungsi tersebut sebagai parameter keluaran pada prosedur.
* Prosedur yang memiliki satu buah parameter keluaran dapat ditulis sebagai fungsi dengan cara menyatakan parameter keluaran sebagai nilai yang dikembalikan oleh fungsi.

Suatu Fungsi atau sub-program, bersifat sebagai suatu program utuh, maksudnya fungsi tersebut dapat melakukan suatu pekerjaan secara utuh. Semua jenis atau macam instruksi dapat digunakan dalam sebuah fungsi.

**Calling Function** disebut juga main program / main function **yaitu** program yang memanggil fungsi tambahan.

**Called Function**  disebut juga fungsi tambahan karena merupakan fungsi yang dipanggil.

Calling function dapat mengirimkan ( passing ) suatu nilai ke Called function.

Bila nilai yang dikirim adalah ***nilai atau data yang akan diproses***, maka pengiriman nilai disebut **Passing by Value.**

Bila nilai yang dikirim berupa ***nilai pointer ( bukan data )***yang me-refer ke suatu data, maka pengiriman tersebut dinamakan **Passing by reference.**

**Cara pemanggilan fungsi pada C++ yaitu :**

1. Cara tak langsung (untuk fungsi yang mengembalikan nilai)

*type varhasil;*

*deklarasi parameter aktual;*

*varhasil=namafungsi(daftar\_parameter\_aktual);*

*cout << hasil;*

1. Cara langsung (untuk fungsi bertipe void/ tidak mengembalikan nilai)

*deklarasi parameter aktual;*

*namafungsi(daftar\_parameter\_aktual);*

*cout << ”...”<<namafungsi(daftar\_parameter\_aktual;*

**Cara pendefinisian fungsi pada C++ :**

*tipe\_hasil namafungsi()*

*{*

*deklarasi variabel lokal*

*daftar\_pernyataan;*

*}*

/\* dilakukan bila tidak ada parameter yang dilewatkan fungsi ini \*/

/\* input data ada pada program yang memanggil \*/

**Atau**

*tipe\_hasil namafungsi(daftar\_nama\_parameter\_formal);*

*deklarasi parameter formal*

*{*

*deklarasi variabel lokal*

*daftar\_pernyataan;*

*return(…);*

*}*

/\* dilakukan bila ada parameter yang dilewatkan fungsi ini \*/

**Atau**

*tipe\_hasil namafungsi(daftar\_tipe\_parameter\_formal)*

*{*

*deklarasi variabel lokal*

*daftar\_pernyataan;*

*return(…);*

*}*

**Atau**

*tipe\_hasil namafungsi(daftar\_tipedannama\_parameter\_formal)*

*{*

*deklarasi variabel lokal*

*daftar pernyataan;*

*return(…);*

*}*

**3. Translasi Notasi Algoritma untuk fungsi ke dalam Notasi bahasa C++.**

**ALGORITMA**

Function *NamaFungsi*(*deklarasi parameter,jika ada*) 🡪 *tipe*

*{spesifikasi fungsi,menjelaskan apa yang dilakukan dan yang dikembalikan oleh fungsi.}*

DEKLARASI

*{semua nama yang dipakai didalam fungsi dan hanya berlaku local didalam prosedur didefinisikan disini }*

DESKRIPSI

*{badan fungsi,berisi instruksi–instruksi untuk menghasilkan nilai yang akan dikembalikan oleh fungsi }*

return *ekspresi { pengembalian nilai yang dihasilkan fungsi }*

**Bahasa C++ :**

tipe NamaFungsi (deklarasi parameter,jika ada);

*/\* spesifikasifungsi,menjelaskan apa yang dilakukan dan yang dikembalikan oleh fungsi \*/*

\*/

{

*/\* DEKLARASI \*/*

*/\* semua nama yang dipakai didalam fungsi dan hanya berlaku*

*local didalam prosedur didefinisikan disini\*/*

*/\* DESKRIPSI \*/*

*/\* badan fungsi,berisi instruksi-instruksi untuk menghasilkan*

*nilai yang akan dikembalikan oleh fungsi \*/*

**return** *ekspresi; /\*pengembalian nilai yang dihasilkan fungsi\*/* }

Catatan :

1. Dalam bahasa C++, fungsi dapat mengembalikan nilai bertipe sederhana ( *integer, riil, Boolean, char,* dan *string* ) maupun bertipe bentukan.
2. Apabila fungsi tidak memiliki daftar parameter formal, maka tanda “(” dan “)” tetap harus ditulis. Hal yang sama juga berlaku pada waktu pemanggilannya.
3. Sebagaimana halnya pada prosedur, fungsi di-definisi-kan di luar blok main(), dan purwarupa fungsi dideklarasikan sebelum blok main().

**Beberapa Contoh translasi :**

1. Fungsi *F(x) = 2x2 + 5x – 8* dan program pemanggilnya.

**ALGORITMA**

Fungsi :

function F(input x: riil)->riil

*{ mengembalikan nilai F(x)=2x2+5x-8, x € R }*

DEKLARASI

*{ tidak ada }*

DESKRIPSI

Return 2\*x\*x + 5\*x - 8

Program utama :

Algoritma TabelFungsi

*{program utama yang memperagakan cara pemanggilan fungsi F.Program ini menampilkan table nilai-nilai x dan f(x) didalam selang[10,15]dengan ∆x = 0.2 }*

DEKLARASI

x : riil

function F(input x: riil)🡪 riil

*{ mengembalikan nilai F(x)=2x2+5x-8, x R }*

DESKRIPSI:

*{ buat header table }*

write(‘------------------------’)

write(‘ x f(x) ’)

write(‘------------------------’)

x 🡨 10.0

while x ≤ 15.0 do

write(x,’ ‘,F(x))

x 🡨 x + 0.2

endwhile

*{ buat garis penutup table }*

write(‘------------------------’)

**Bahasa C / C++ dengan include <stdio.h> :**

*/\* Program TabelFungsi \*/*

*/\* Prog.utama yang memperagakan cara pemanggilan fungsi F.Prog.ini menampilkan table nilai-nilai x dan f(x) didalam selang [10,15] dengan ∆x = 0.2 \*/*

**#include** <stdio.h>

*/\* purwarupa fungsi \*/*

**float** F(**float** x);

*/\* mengembalikan nilai F(x)=2x2+5x-8, x bertipe riil \*/*

main()

{ */\* DEKLARASI \*/*

**float** x;

*/\* DESKRIPSI \*/*

*/\* buat header table \*/*

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

printf(" x f(x) \n");

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

x = 10.0;

while (x <= 15.0)

{

printf("%4.2f %10.4f \n",x,F(x));

x = x + 0.2;

} */\*endwhile\*/*

*/\* buat garis penutup table \*/*

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

}

**float** F(**float** x)

*/\* mengembalikan nilai F(x)=2x2+5x-8, x bertipe riil \*/*

{ */\*DEKLARASI\*/*

*/\* tidak ada \*/*

*/\*DESKRIPSI\*/*

return 2\*x\*x + 5\*x - 8;

}

**Bahasa C++ dengan include <iostream.h> :**

*/\* Program TabelFungsi \*/*

*/\* Prog.utama yang memperagakan cara pemanggilan fungsi F.Prog.ini menampilkan table nilai-nilai x dan f(x) didalam selang [10,15] dengan ∆x = 0.2 \*/*

#include <iostream.h>

#include <iomanip.h>

*/\* purwarupa fungsi \*/*

**float** F(**float** x);

*/\* mengembalikan nilai F(x)=2x2+5x-8, x bertipe riil \*/*

void main()

{ */\* DEKLARASI \*/*

**float** x;

*/\* DESKRIPSI \*/*

*/\* buat header table \*/*

cout <<"---------------------\n";

cout <<" x f(x) \n";

cout <<"---------------------\n";

x = 10.0;

while (x <= 15.0)

{

cout << x << setw(15) << F(x) << endl;

x = x + 0.2;

} */\*endwhile\*/*

*/\* buat garis penutup table \*/*

cout <<"---------------------\n";

}

**float** F(**float** x)

*/\* mengembalikan nilai F(x)=2x2+5x-8, x bertipe riil \*/*

{ */\*DEKLARASI\*/*

*/\* tidak ada \*/*

*/\*DESKRIPSI\*/*

return 2\*x\*x + 5\*x - 8;

}

1. Fungsi yang mengembalikan hasil bertipe bentukan.

**ALGORITMA**

Fungsi :

function TitikTengah(input P1,P2 :titik)🡪 titik

*{ mengembalikan Titik Tengah dari P1 dan P2 }*

DEKLARASI

Pt : titik

DESKRIPSI

Pt.x 🡨 (P1.x + P2.x)/2

Pt.y 🡨 (P1.y + P2.y)/2

Return pt

Program pemanggil :

Algoritma HitungTitikTengah

*{program untuk menghitung titik tengah dari dua buah titik dibidang datar }*

DEKLARASI

Type Titik : record < x : riil, y : riil >

P1, P2 : Titik

Pt : Titik *{ titik tngah P1 dan P2 }*

Function TitikTengah( input P1,P2 :titik)->titik

*{ mengembalikan Titik Tengah dari P1 dan P2 }*

DESKRIPSI

Read(P1.x, P1.y)

Read(P2.x, P2.y)

Pt 🡨 TitikTengah(P1,P2)

write(Pt.x, Pt.y)

**Bahasa C / C++ dengan include <stdio.h> :**

*/\* Program HitungTitikTengah \*/*

*/\* Program untuk menghitung titik tengah dari dua buah titik di bidang. \*/*

#include <stdio.h>

*/\* DEKLARASI GLOBAL \*/*

**typedef struct** { **float** x; **float** y; } Titik;

Titik TitikTengah(Titik P1, Titik P2);

*/\*mengembalikan true,Pt adalah titik tengah dari P1 dan P2 \*/*

main()

{ */\* DEKLARASI \*/*

Titik P1, P2;

Titik Pt; /\* titik tengah P1 dan P2 \*/

*/\* DESKRIPSI \*/*

printf(" Titik P1 : \n");

printf(" x = "); scanf("%f", &P1.x);

printf(" y = "); scanf("%f", &P1.y);

printf(" Titik P2 : \n");

printf(" x = "); scanf("%f", &P2.x);

printf(" y = "); scanf("%f", &P2.y);

Pt = TitikTengah(P1,P2);

printf(" Titik Tengah : (%2.1f, %2.1f) \n",Pt.x, Pt.y);

}

Titik TitikTengah(Titik P1, Titik P2)

*/\* mengembalikan titik tengah dari P1 dan P2 \*/*

{ /\*DEKLARASI\*/

Titik Pt;

/\*DESKRIPSI\*/

Pt.x = (P1.x + P2.x)/2;

Pt.y = (P1.y + P2.y)/2;

return Pt;

}

**Bahasa C++ dengan include <iostream.h> :**

/\* Program HitungTitikTengah \*/

/\* Program untuk menghitung titik tengah dari dua buah titik di bidang. \*/

#include <iostream.h>

*/\* DEKLARASI GLOBAL \*/*

**typedef struct** { float x; float y; } Titik;

Titik TitikTengah(Titik P1, Titik P2);

*/\* mengembalikan true,Pt adalah titik tengah dari P1 dan P2 \*/*

**void** main()

{ */\* DEKLARASI \*/*

Titik P1, P2;

Titik Pt; */\* titik tengah P1 dan P2 \*/*

*/\* DESKRIPSI \*/*

cout << " Titik P1 : " << endl;

cout << " x = ";

cin >> P1.x ;

cout << " y = ";

cin >> P1.y ;

cout << " Titik P2 :" << endl;

cout << " x = ";

cin >> P2.x ;

cout << " y = " ;

cin >> P2.y ;

Pt = TitikTengah(P1,P2);

cout << " Titik Tengah : "<< Pt.x << ","<< Pt.y ;

}

Titik TitikTengah(Titik P1, Titik P2)

*/\* mengembalikan titik tengah dari P1 dan P2 \*/*

{ */\*DEKLARASI\*/*

Titik Pt;

*/\*DESKRIPSI\*/*

Pt.x = (P1.x + P2.x)/2;

Pt.y = (P1.y + P2.y)/2;

return Pt;

}

**PENUTUP**

Fungsi adalah sejumlah instruksi yang dikelompokkan menjadi satu, berdiri sendiri, yang berfungsi untuk menyelesaikan suatu pekerjaan tertentu. Bahasa C++ adalah suatu bahasa yang struktur penulisannya merupakan kumpulan dari fungsi- fungsi.

**SOAL-SOAL**

Buat program kalkulator dengan fungsi menjumlahkan, membagi, mengurangi dan mengalikan dua bilangan.

**REFERENSI** :

* Buku Teks

1. Munir, Rinaldi (2005), *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, Buku 1, Edisi Ketiga, Penerbit Informatika Bandung.
2. Charibaldi, N. (2004), *Modul Kuliah Algoritma Pemrograman I*, Edisi Kedua, Yogyakarta

* Buku Acuan/Referensi

1. Brassard, Gilles (1999), *Fundamentals of algorithma*, PrinteceHall.
2. Jarne, Stroustrup B. (1997), *C++ Programming language,* AT &T.
3. Kristanto, Andri (2003), *Algoritma pemrograman C++*, Graha Ilmu.
4. Schildt,Herbert (2000), *The Complete Reference C++*, McGraw-Hill.
5. Sedgewick, R. (2000), *Algoritma Third edition In C part 5*, Addison Wesley.