



Dasar Teori

Fungsi adalah suatu bagian dari program yang dirancang untuk melaksanakan tugas tertentu dan letaknya dipisahkan dari program yang menggunakannya. Elemen utama dari program bahasa C dan C++ berupa fungsi-fungsi, dalam hal ini program dari bahasa C dan C++ dibentuk dari kumpulan fungsi pustaka (standar) dan fungsi yang dibuat sendiri oleh pemrogram. Fungsi banyak digunakan pada program C dengan tujuan :

- Program menjadi terstruktur, sehingga mudah dipahami dan mudah dikembangkan.
- Dapat mengurangi pengulangan (duplikasi) kode. Langkah-langkah program yang sama dan dipakai berulang-ulang diprogram dapat dituliskan sekali saja secara terpisah dalam bentuk fungsi-fungsi. Selanjutnya bagian program yang membutuhkan langkah-langkah ini tidak perlu selalu menuliskannya, tetapi cukup memanggil fungsi-fungsi tersebut.
- Dapat menghemat ukuran dari program.
- Dapat memudahkan dalam mencari kesalahan.
- Modifikasi program dapat dilakukan pada suatu modul tertentu saja tanpa mengganggu program keseluruhan.
- Reusability*, yaitu suatu fungsi dapat digunakan kembali oleh program atau fungsi lain.

Pada umumnya fungsi memerlukan nilai masukan atau **parameter**. Nilai parameter ini akan diolah oleh fungsi. Hasil akhir fungsi berupa sebuah nilai yang disebut sebagai **return value** atau nilai keluaran fungsi. Oleh karena itu, fungsi sering digambarkan sebagai “black box” seperti ditunjukkan pada gambar dibawah ini :



Gambar 7.1 Gambaran Fungsi

Penggambaran sebagai kotak hitam diantaranya menjelaskan bahwa bagian dalam fungsi bersifat pribadi bagi fungsi. Tidak ada suatu pernyataan diluar fungsi yang bisa mengakses bagian dalam fungsi, selain melalui parameter (atau variabel eksternal). Bentuk umum dari definisi sebuah fungsi adalah sebagai berikut :

```
tipedata_keluaran_fungsi namafungsi (deklarasi parameter)
{
    tubuh fungsi;
    return nilai_return;
}
```

Keterangan :

- *tipedata_keluaran_fungsi*, dapat berupa salah satu tipe data C dan C++, misalnya `char` atau `int`. Jika tipe data tidak disebutkan, maka dianggap bertipe `int` (secara default).
- *tubuh fungsi* berisi deklarasi variabel (jika ada) dan statemen-statemen yang akan melakukan tugas yang akan diberikan kepada fungsi yang bersangkutan. Tubuh fungsi ini ditulis didalam tanda kurung kurawal buka dan kurung kurawal tutup.

Sebuah fungsi yang sederhana bisa saja tidak mengandung parameter sama sekali dan tentu saja untuk keadaan ini deklarasi parameter juga tidak ada. Contoh :

```
inisialisasi()
{
    return(0);
}
```

Pada fungsi diatas :

- tipe keluaran fungsi tidak disebutkan, berarti keluaran fungsi bertipe `int`.
- Inisialisasi adalah nama fungsi
- Tanda `()` sesudah nama fungsi menyatakan bahwa fungsi tak memiliki parameter
- Tanda `{` dan `}` adalah awal dan akhir fungsi
- `return(0)` merupakan sebuah pernyataan dalam tubuh fungsi.

Didalam C dan C++ terdapat beberapa kategori Function, yaitu :

1. Standard Library Function

Yaitu fungsi-fungsi yang telah disediakan oleh C/C++ dalam file-file header atau librarynya.

Misalnya: `clrscr()`, `printf()`, `getch()`

Untuk function ini kita harus mendeklarasikan terlebih dahulu library yang akan digunakan.

2. Programmer Defined Function

Adalah function yang dibuat oleh programmer sendiri. Function ini memiliki nama tertentu yang unik dalam program, letaknya terpisah dari program utama, dan bisa dijadikan satu ke dalam suatu library buatan programmer itu sendiri yang kemudian juga di-include-kan untuk penggunaannya.

Prototipe Fungsi

Prototipe fungsi digunakan untuk menjelaskan kepada kompiler mengenai :

- Tipe keluaran fungsi
- Jumlah parameter
- Tipe dari masing-masing parameter

Bagi kompiler, informasi dalam prototipe akan dipakai untuk memeriksa keabsahan (validitas) parameter dalam pemanggilan fungsi. Salah satu keuntungannya adalah, kompiler akan melakukan konversi seandainya antara tipe parameter dalam fungsi dan parameter saat pemanggilan fungsi tidak sama, atau akan menunjukkan kesalahan bila jumlah parameter dalam definisi dan saat pemanggilan berbeda.

Ada dua cara menggunakan fungsi dalam suatu program, yaitu :

1. Menggunakan Fungsi tanpa menyertakan prototipe fungsinya

Jika membuat fungsi tanpa menyertakan prototipe fungsinya, maka fungsi yang dibuat harus diletakan diatas fungsi main. Jika tidak, maka tidak akan terdefinisi.

```
tipe data_keluaran_fungsi nama_fungsi (parameter)
{
    tubuh_fungsi;
    return nilai_return;
}

main()
{
    nama_fungsi(parameter);
    return 0;
}
```

2. Menggunakan Fungsi dengan menyertakan prototipe fungsinya

Jika membuat fungsi dengan menyertakan prototipe fungsinya terlebih dahulu, maka fungsi yang dibuat bebas diletakan dimanapun dalam program tersebut.

```
tipedata_keluaran_fungsi namafungsi1 (parameter);
tipedata_keluaran_fungsi namafungsi2 (parameter);

main()
{
    namafungsi1(parameter);
    namafungsi2(parameter);
    return 0;
}

tipedata_keluaran_fungsi namafungsi1 (parameter)
{
    tubuh fungsi;
    return nilai_return;
}

tipedata_keluaran_fungsi namafungsi2 (parameter)
{
    tubuh fungsi;
    return nilai_return;
}
```

Contoh lainnya adalah :

```
float jumlah (float x, float y);
```

Untuk fungsi yang tidak memiliki argumen (contoh program void.c), maka deklarasinya adalah :

```
void info_program(void);
```

↑ menyatakan bahwa **info_program()** tidak memiliki parameter

Catatan :

- Untuk fungsi-fungsi pustaka, prototipe dari fungsi-fungsi berada di file-file judulnya (header file). Misalnya fungsi pustaka `printf()` dan `scanf()` prototipenya berada pada file dengan nama `stdio.h`
- Untuk fungsi pustaka pencantuman pada prototipe fungsi dapat dilakukan dengan menggunakan `#include`

Procedure

Procedure adalah suatu fungsi yang tidak mengembalikan suatu nilai. Dalam bahasa C dan C++ istilah procedure dikenal dengan tipe void. Bentuk umum dari procedure adalah sebagai berikut :

```
void namafungsi1 (parameter_x, parameter_y);

void namafungsi1 (parameter_x, parameter_y);
{
    tubuh fungsi;
}

main()
{
    namafungsi1(nilai parameter_x, nilai parameter_y );
    return 0;
}
```

Contoh Program dari procedure adalah sebagai berikut :

```
1  /*
2   Program 7-1
3   Nama File : Program 7-1.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 void tampil_harga(); // Prototipe Fungsi
11
12 int main(int argc, char *argv[])
13 {
14     tampil_harga();
15     system("PAUSE");
16     return 0;
17 }
18
19 void tampil_harga()
20 {
21     char merk[20];
22     char type[25];
23     float harga;
24     printf("Inputkan MERK LAPTOP : ");fflush(stdin);gets(merk);
25     printf("Inputkan TYPE LAPTOP : ");fflush(stdin);gets(type);
26     printf("Inputkan HARGA LAPTOP : ");scanf("%f",&harga);
27     printf("\n===== \n");
```

```

28 printf("DATA YANG DIINPUTKAN\n");
29 printf("=====\n");
30 printf("MERK   : %s\n", merk);
31 printf("TYPE   : %s\n", type);
32 printf("HARGA   : %.2f\n\n", harga);
33 }

```

```

1  /*
2   Program 7-1
3   Nama File : Program 7-1.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9  #include <iomanip>
10
11 using namespace std;
12 void tampil_harga(); // prototipe Fungsi
13
14
15 int main(int argc, char *argv[])
16 {
17     tampil_harga();
18     system("PAUSE");
19     return EXIT_SUCCESS;
20 }
21
22 void tampil_harga()
23 {
24     char merk[21];
25     char type[26];
26     float harga;
27     cout<<"Inputkan MERK LAPTOP   : ";fflush(stdin);cin.get(merk,20);
28     cout<<"Inputkan TYPE LAPTOP   : ";fflush(stdin);cin.get(type,20);
29     cout<<"Inputkan HARGA LAPTOP   : ";fflush(stdin);cin>>harga;
30     cout<<endl;
31     cout<<setiosflags(ios::fixed);
32     cout<<"====="<<endl;
33     cout<<"DATA YANG DIINPUTKAN"<<endl;
34     cout<<"====="<<endl;
35     cout<<"MERK   : "<<merk<<endl;
36     cout<<"TYPE   : "<<type<<endl;
37     cout<<"HARGA   : "<<setprecision(2)<<harga<<endl<<endl;
38 }

```

```

F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman I\Kumpula...
Inputkan MERK LAPTOP      : ACER
Inputkan TYPE LAPTOP     : Aspire 4736Z
Inputkan HARGA LAPTOP    : 5250000

=====
DATA YANG DIINPUTKAN
=====
MERK      : ACER
TYPE     : Aspire 4736Z
HARGA    : 5250000.00

Press any key to continue . . .

```

Gambar 7.1 Tampilan Program 7-1

```

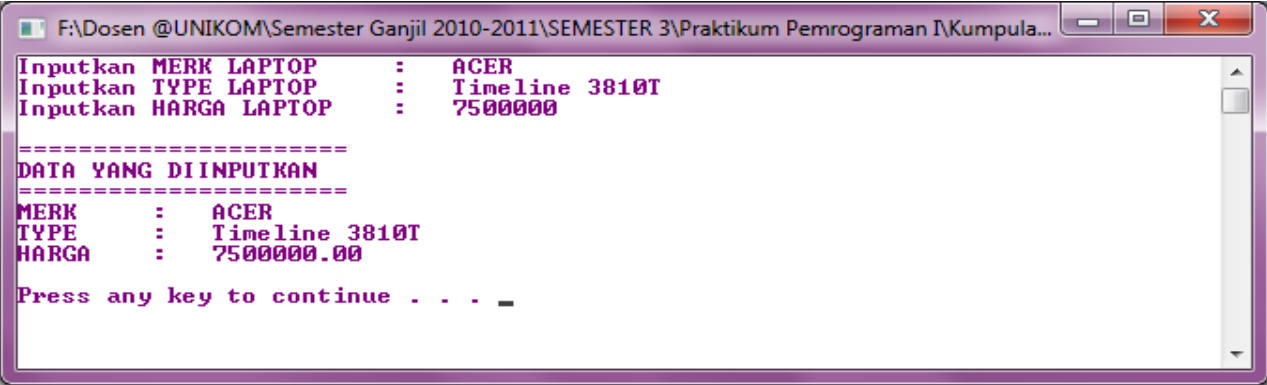
1  /*
2   Program 7-2
3   Nama File : Program 7-2.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 void tampil_harga(char merk[20], char type[25], float harga);
11
12 void tampil_harga(char merk[20], char type[25], float harga)
13 {
14     printf("\n===== \n");
15     printf("DATA YANG DIINPUTKAN \n");
16     printf("===== \n");
17     printf("MERK   : %s \n", merk);
18     printf("TYPE  : %s \n", type);
19     printf("HARGA : %.2f \n \n", harga);
20 }
21
22 int main(int argc, char *argv[])
23 {
24     char merk[20];
25     char type[25];
26     float harga;
27     printf("Inputkan MERK LAPTOP   : "); fflush(stdin); gets(merk);
28     printf("Inputkan TYPE LAPTOP    : "); fflush(stdin); gets(type);
29     printf("Inputkan HARGA LAPTOP   : "); scanf("%f",&harga);
30     tampil_harga(merk, type, harga);
31     system("PAUSE");
32     return 0;
33 }

```

```

1  /*
2   Program 7-2
3   Nama File : Program 7-2.cpp
4   Programmer : Eko Budi Setiawan
5  */
6  #include <cstdlib>
7  #include <iostream>
8  #include <iomanip>
9
10 using namespace std;
11 void tampil_harga(char merk[21], char type[26], float harga);
12
13 void tampil_harga(char merk[20], char type[25], float harga)
14 {
15     cout<<setiosflags(ios::fixed);
16     cout<<"======"<<endl;
17     cout<<"DATA YANG DIINPUTKAN"<<endl;
18     cout<<"======"<<endl;
19     cout<<"MERK   : "<<merk<<endl;
20     cout<<"TYPE   : "<<type<<endl;
21     cout<<"HARGA  : "<<setprecision(2)<<harga<<endl<<endl;
22 }
23
24 int main(int argc, char *argv[])
25 {
26     char merk[20];
27     char type[25];
28     float harga;
29     cout<<"Inputkan MERK LAPTOP   : ";fflush(stdin);cin.get(merk,20);
30     cout<<"Inputkan TYPE LAPTOP    : ";fflush(stdin);cin.get(type,20);
31     cout<<"Inputkan HARGA LAPTOP    : ";fflush(stdin);cin>>harga;
32     cout<<endl;
33     tampil_harga(merk,type,harga);
34     system("PAUSE");
35     return EXIT_SUCCESS;
36 }

```



Gambar 7.1 Tampilan Program 7-2

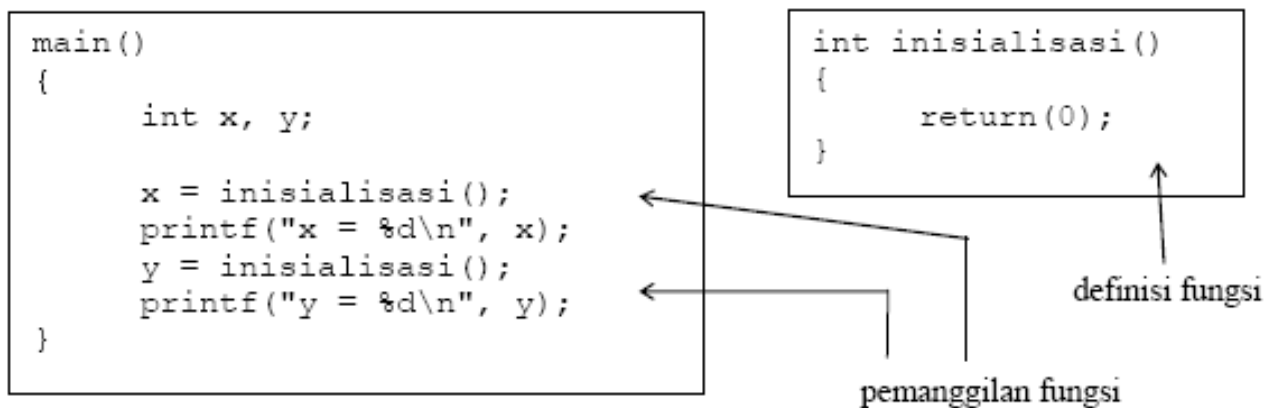
Fungsi dengan Nilai Balik

Suatu fungsi dibuat untuk maksud menyelesaikan tugas tertentu. Suatu fungsi dapat hanya melakukan suatu tugas saja tanpa memberikan suatu hasil keluaran atau melakukan suatu tugas dan kemudian memberikan hasil keluaran. Fungsi yang hanya melakukan suatu tugas saja tanpa memberikan hasil keluaran misalnya adalah fungsi untuk menampilkan hasil dilayar.

Dalam tubuh fungsi, pernyataan yang digunakan untuk memberikan nilai keluaran fungsi berupa `return`. Struktur dari Fungsi dengan Nilai Balik adalah sebagai berikut :

```
void namafungsi1 (parameter_x, parameter_y);  
  
void namafungsi1 (parameter_x, parameter_y);  
{  
    tubuh fungsi;  
    return nilai_balik;  
}  
  
main()  
{  
    namafungsi1(nilai parameter_x, nilai parameter_y );  
    return 0;  
}
```

Contoh :



Hasil Eksekusi :

x = 0

y = 0

Program diatas sekaligus menjelaskan bahwa suatu fungsi cukup didefinisikan satu kali tetapi bisa digunakan beberapa kali. Pada keadaan semacam ini, seandainya tubuh fungsi mengandung

banyak pernyataan, maka pemakaian fungsi dapat menghindari duplikasi kode dan tentu saja menghemat penulisan program maupun kode dalam memori.

Misalnya pada saat pernyataan

```
x = inisialisasi ();
```

dijalankan, mula-mula eksekusi akan diarahkan ke fungsi inisialisasi (), selanjutnya suatu nilai keluaran (hasil fungsi) akhir fungsi diberikan ke x. Proses yang serupa, dilakukan untuk pernyataan

```
y = inisialisasi ();
```

Bagi suatu fungsi, jika suatu pernyataan return dieksekusi, maka eksekusi terhadap fungsi akan berakhir dan nilai pada parameter return akan menjadi keluaran fungsi. Untuk fungsi yang tidak memiliki pernyataan return, tanda } pada bagian akhir fungsi akan menyatakan akhir eksekusi fungsi.

Fungsi dengan Nilai Balik Bilangan

Berikut ini contoh dari suatu fungsi yang mengembalikan nilai bertipe bilangan integer dan float.

```
1  /*
2   Program 7-3
3   Nama File : Program 7-3.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int minimum(int x, int y);
11
12 int minimum(int x, int y)
13 {
14     if (x<y)
15         return x;
16     else
17         return y;
18 }
19
20 int main(int argc, char *argv[])
21 {
22     int hasil;
23     int a=20;
24     int b=44;
25     hasil = minimum (a,b);
```

```

26     printf("Nilai minimum = %d\n",hasil);
27     printf("Nilai minimum = %d\n",minimum(12,15));
28     system("PAUSE");
29     return 0;
30 }

```

```

1  /*
2   Program 7-3
3   Nama File : Program 7-3.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9  using namespace std;
10 int minimum(int x, int y);
11 int minimum(int x, int y)
12 {
13     if (x<y)
14         return x;
15     else
16         return y;
17 }
18
19 int main(int argc, char *argv[])
20 {
21     int hasil;
22     int a=20;
23     int b=44;
24     hasil = minimum (a,b);
25     cout<<"Nilai Minimum  = "<<hasil<<endl;
26     cout<<"Nilai Minimum  = "<<minimum(12,15);
27     cout<<endl<<endl;
28     system("PAUSE");
29     return EXIT_SUCCESS;
30 }

```

```

F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman I\Kumpula...
Nilai Minimum = 20
Nilai Minimum = 12
Press any key to continue . . . -

```

Gambar 7.3 Tampilan Program 7-3

```

1  /*
2   Program 7-4
3   Nama File : Program 7-4.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 float maksimum (float a, float b);
11 float minimum (float a, float b);
12
13 float maksimum (float a, float b)
14 {
15     if (a>b)
16         return a;
17     else
18         return b;
19 }
20
21 int main(int argc, char *argv[])
22 {
23     float hasil_maksimum, hasil_minimum;
24     float a = 20.5;
25     float b = 37.5;
26     hasil_maksimum = maksimum (a,b);
27     hasil_minimum = minimum (a,b);
28     printf("Nilai maksimum1 = %g\n",hasil_maksimum);
29     printf("Nilai maksimum2 = %g\n",maksimum(3.5,2.5));
30     printf("\n===== \n");
31     printf("Nilai minimum1 = %g\n",hasil_minimum);
32     printf("Nilai minimum2 = %g\n",minimum(3.5,2.5));
33
34     system("PAUSE");
35     return 0;
36 }
37
38 float minimum (float a, float b)
39 {
40     if (a<b)
41         return a;
42     else
43         return b;
44 }
45

```

```

1  /*
2   Program 7-4
3   Nama File : Program 7-4.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11 float maksimum (float a, float b);
12 float minimum (float a, float b);
13
14 float maksimum (float a, float b)
15 {
16     if (a>b)
17         return a;
18     else
19         return b;
20 }
21
22 int main(int argc, char *argv[])
23 {
24     float hasil_maksimum, hasil_minimum;
25     float a = 20.5;
26     float b = 37.5;
27     hasil_maksimum = maksimum (a,b);
28     hasil_minimum = minimum (a,b);
29     cout<<"Nilai Maksimum1 = "<<hasil_maksimum<<endl;
30     cout<<"Nilai Maksimum2 = "<<maksimum(3.5,2.5)<<endl;
31     cout<<"====="<<endl;
32     cout<<"Nilai Minimum1 = "<<hasil_minimum<<endl;
33     cout<<"Nilai Minimum2 = "<<minimum(3.5,2.5)<<endl<<endl;
34     system("PAUSE");
35     return EXIT_SUCCESS;
36 }
37
38 float minimum (float a, float b)
39 {
40     if (a<b)
41         return a;
42     else
43         return b;
44 }

```

```
F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman I\Kumpula...
Nilai Maksimum1 = 37.5
Nilai Maksimum2 = 3.5
-----
Nilai Minimum1 = 20.5
Nilai Minimum2 = 2.5
Press any key to continue . . .
```

Gambar 7.4 Tampilan Gambar 7-4

Fungsi dengan Nilai Balik String

Selain mengembalikan nilai balik berupa bilangan, fungsi juga dapat mengembalikan nilai balik berupa String.

```
1  /*
2   Program 7-5
3   Nama File : Program 7-5.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 char *kampus();
11 char *kampus()
12 {
13     return "Dep.Teknik Informatika \n" "Universitas Komputer Indonesia \n"
14         "Jl.Dipatiukur 102-116 Bandung \n";
15 }
16
17 int main(int argc, char *argv[])
18 {
19     printf("%s \n",kampus());
20     system("PAUSE");
21     return 0;
22 }
```

```
1  /*
2   Program 7-5
3   Nama File : Program 7-5.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
```

```

7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11 char *kampus();
12 char *kampus()
13 {
14     return "Dep.Teknik Informatika \n" "Universitas Komputer Indonesia \n"
15         "Jl.Dipatiukur 102-116 Bandung \n";
16 }
17
18 int main(int argc, char *argv[])
19 {
20     cout<<kampus()<<endl;
21     system("PAUSE");
22     return EXIT_SUCCESS;
23 }

```

```

F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman \Kumpula...
Dep.Teknik Informatika
Universitas Komputer Indonesia
Jl.Dipatiukur 102-116 Bandung
Press any key to continue . . . _

```

Gambar 7.5 Tampilan proram 7-5

Fungsi dengan Parameter Masukan

Fungsi dengan Parameter Masukan ini menggunakan parameter sebagai nilai masukan dalam fungsi. Kemudian nilai masukan tersebut akan diproses untuk menghasilkan suatu nilai balik. Contoh dari Fungsi dengan Parameter Masukan adalah sebagai berikut :

```

1  /*
2   Program 7-6
3   Nama File : Program 7-6.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9  float luas_segitiga(float alas, float tinggi);
10
11 float luas_segitiga(float alas, float tinggi)
12 {

```

```

13     float luas;
14     luas=0.5*alas*tinggi;
15     return luas;
16 }
17
18 int main(int argc, char *argv[])
19 {
20     float isi_alas, isi_tinggi,luas;
21     printf("Alas Segitiga      : ");scanf("%f",&isi_alas);
22     printf("Tinggi Segitiga    : ");scanf("%f",&isi_tinggi);
23     luas=luas_segitiga(isi_alas,isi_tinggi);
24     printf("=====\n");
25     printf("Luas segitiga adalah = %.2f",luas);
26     printf("\n");
27     printf("\n");
28     system("PAUSE");
29     return 0;
30 }

```

```

1  /*
2  Program 7-6
3  Nama File : Program 7-6.cpp
4  Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9  #include <iomanip>
10
11 using namespace std;
12 float luas_segitiga(float alas, float tinggi);
13
14 float luas_segitiga(float alas, float tinggi)
15 {
16     float luas;
17     luas=0.5*alas*tinggi;
18     return luas;
19 }
20
21 int main(int argc, char *argv[])
22 {
23     float isi_alas, isi_tinggi, luas;
24     cout<<"Alas Segitiga = ";cin>>isi_alas;
25     cout<<"Tinggi Segitiga = ";cin>>isi_tinggi;
26     luas=luas_segitiga(isi_alas,isi_tinggi);
27     cout<<"=====\n";
28     cout<<setiosflags(ios::fixed);
29     cout<<"Luas Segitiga adalah "<<setprecision(2)<<luas;
30     cout<<endl<<endl;

```



```

31
32     system("PAUSE");
33     return EXIT_SUCCESS;
34 }

```

```

F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman \Kumpula...
Alas Segitiga = 10
Tinggi Segitiga = 3
=====
Luas Segitiga adalah 15.00
Press any key to continue . . .

```

Gambar 7.6 Tampilan Program 7-6

Fungsi dengan Parameter Keluaran

Fungsi dengan Parameter keluaran ini digunakan untuk menampung nilai yang telah dihasilkan dari suatu proses dari fungsi yang dilakukan. Parameter keluaran ini biasanya digunakan untuk fungsi bertipe **void**. Contohnya adalah :

```

1  /*
2   Program 7-7
3   Nama File : Program 7-7.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 void luas_lingkaran(float diameter, float *luas);
11 void luas_lingkaran(float diameter, float *luas)
12 {
13     *luas=0;
14     *luas=3.14*(0.5*diameter);
15 }
16
17 int main(int argc, char *argv[])
18 {
19     float diameter, luas;
20     diameter = 0;
21     printf("Inputkan Diameter = ");scanf("%f",&diameter);
22     luas_lingkaran(diameter, &luas);
23     printf("=====\n");
24     printf("Luas Lingkaran = %.2f \n", luas);

```

```

25     printf("\n\n");
26     system("PAUSE");
27     return 0;
28 }

```

```

1  /*
2   Program 7-7
3   Nama File : Program 7-7.cpp
4   Programmer : Eko Budi Setiawan
5  */
6  #include <cstdlib>
7  #include <iostream>
8  #include <iomanip>
9
10 using namespace std;
11 void luas_lingkaran(float diameter, float *luas);
12 void luas_lingkaran(float diameter, float *luas)
13 {
14     *luas=0;
15     *luas=3.14*(0.5*diameter);
16 }
17
18 int main(int argc, char *argv[])
19 {
20     float diameter, luas;
21     diameter = 0;
22     cout<<"Inputkan Diameter = ";cin>>diameter;
23     luas_lingkaran(diameter, &luas);
24     cout<<"=====\n";
25     cout<<setiosflags(ios::fixed);
26     cout<<"Luas Lingkaran = "<<setprecision(2)<<luas<<endl;
27     cout<<endl;
28     system("PAUSE");
29     return EXIT_SUCCESS;
30 }

```

```

F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman I\Kumpula...
Inputkan Diameter = 10
=====
Luas Lingkaran = 15.70
Press any key to continue . . . _

```

Gambar 7.7 Tampilan Gambar 7-7

Fungsi dengan Parameter Masukan & Keluaran

Fungsi ini digunakan sebagai masukan untuk diproses dalam tubuh fungsi, kemudian hasil dari proses tersebut akan disimpan kedalam parameter sebagai suatu nilai keluaran. Contoh :

```
1  /*
2   Program 7-8
3   Nama File : Program 7-8.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9  #include <math.h>
10
11 void akar(float *akar);
12 void akar(float *akar)
13 {
14     *akar=sqrt(*akar);
15 }
16
17 int main(int argc, char *argv[])
18 {
19     float isi_akar,temp;
20     printf("Input Bilangan      = ");scanf("%f", &isi_akar);
21     temp=isi_akar;
22     akar(&isi_akar);
23     printf("=====\n");
24     printf("Akar dari %.0f adalah = %.2f ", temp, isi_akar);
25     printf("\n\n");
26     system("PAUSE");
27     return 0;
28 }
```

```
1  /*
2   Program 7-8
3   Nama File : Program 7-8.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9  #include <math.h>
10
11 using namespace std;
12 void akar(float *akar);
13 void akar(float *akar)
14 {
```

```

15     *akar=sqrt(*akar);
16 }
17
18 int main(int argc, char *argv[])
19 {
20     float isi_akar, temp;
21     cout<<"Input Bilangan    = ";cin>>isi_akar;
22     temp=isi_akar;
23     akar(&isi_akar);
24     cout<<"=====\n";
25     cout<<"Akar dari "<<temp<<" adalah "<<isi_akar;
26     cout<<endl<<endl;
27     system("PAUSE");
28     return EXIT_SUCCESS;
29 }

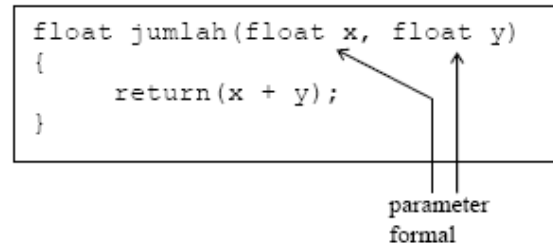
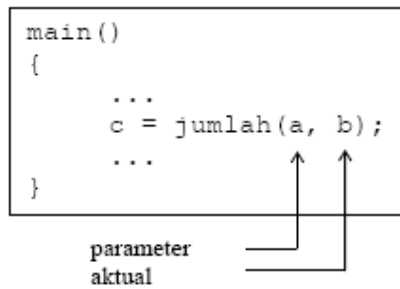
```

Gambar 7.8 Tampilan Program 7-8

Parameter Formal dan Aktual

Adapun perbedaan dari parameter Formal dan parameter aktual adalah sebagai berikut :

1. Parameter Formal : Adalah suatu fungsi yang variabelnya ada pada daftar Parameter didalam definisi fungsi. Atau dengan kata lain parameternya dideklarasikan sewaktu pembuatan prototipe fungsi.
2. Parameter Aktual : Adalah parameter yang tidak selalu berupa variabel yang digunakan sewaktu pemanggilan fungsi.



Call by Value

Pemanggilan dengan nilai (*call by value*) merupakan cara yang dipakai untuk seluruh fungsi buatan yang telah dibahas pada praktikum sebelumnya. Pada menanggilan dengan nilai, nilai dari parameter aktual akan disalin ke parameter formal. Dengan cara ini nilai parameter aktual tidak bisa dirubah sekalipun nilai parameter formal berubah. Untuk lebih jelasnya lihat pada fungsi pada contoh berikut ini.

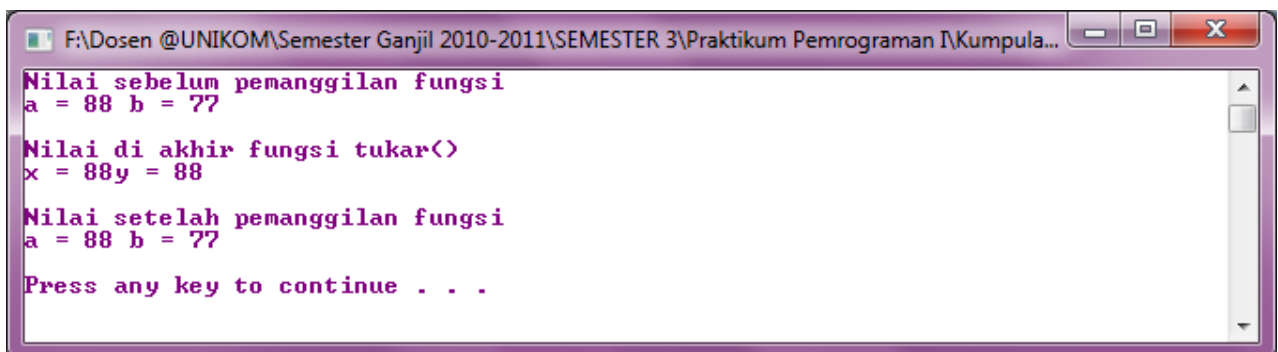
```

1  /*
2  Program 7-9
3  Nama File : Program 7-9.c
4  Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 void tukar (int, int);
11 void tukar (int x, int y)
12 {
13     int z;
14     z = x;
15     y = z;
16     printf("\nNilai di akhir fungsi tukar()\n");
17     printf("x = %d y = %d\n", x, y);
18 }
19
20 int main(int argc, char *argv[])
21 {
22     int a = 88, b = 77;
23     printf("Nilai sebelum pemanggilan fungsi\n");
24     printf("a = %d b = %d\n", a, b);
25     tukar(a,b);
26     printf("\nNilai setelah pemanggilan fungsi\n");
27     printf("a = %d b = %d\n", a, b);
28     printf("\n");
29     system("PAUSE");

```

```
30     return 0;
31 }
```

```
1  /*
2   Program 7-9
3   Nama File : Program 7-9.cpp
4   Programmer : Eko Budi Setiawan
5   */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11 void tukar (int, int);
12 void tukar (int x, int y)
13 {
14     int z;
15     z = x;
16     y = z;
17     cout<<"\nNilai di akhir fungsi tukar()\n";
18     cout<<"x = "<<x<< "y = "<<y<<endl;
19 }
20
21 int main(int argc, char *argv[])
22 {
23     int a = 88, b = 77;
24     cout<<"Nilai sebelum pemanggilan fungsi\n";
25     cout<<"a = "<<a<< " b = "<<b<<endl;
26     tukar(a,b);
27     cout<<"\nNilai setelah pemanggilan fungsi\n";
28     cout<<"a = "<<a<< " b = "<<b<<endl;
29     cout<<"\n";
30     system("PAUSE");
31     return EXIT_SUCCESS;
32 }
33
```



```
F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman I\Kumpula...
Nilai sebelum pemanggilan fungsi
a = 88 b = 77
Nilai di akhir fungsi tukar()
x = 88y = 88
Nilai setelah pemanggilan fungsi
a = 88 b = 77
Press any key to continue . . .
```

Gambar 7.9 Tampilan Program 7-9

Tampak bahwa sekeduanya dari pemanggilan fungsi `tukar()`, variabel `a` dan `b` yang dilewatkan ke fungsi `tukar()` tidak berubah, walaupun pada fungsi `tukar()` telah terjadi penukaran antara parameter `x` dan `y`. Mengapa hal ini bisa terjadi? sebab `x` hanyalah salinan dari `a` dan `y` adalah salinan dari `b`. Pada saat pemanggilan fungsi, maka

- `x` bernilai 88 (nilai `a`)
- `y` bernilai 77 (nilai `b`)

Call by Reference

Pemanggilan dengan referensi (`call by reference`) merupakan upaya untuk melewatkan alamat dari suatu variabel kedalam fungsi. Cara ini dapat dipakai untuk mengubah isi suatu variabel diluar fungsi dengan pelaksanaan perubahan dilakukan didalam fungsi.

```
1  /*
2   Program 7-10
3   Nama File : Program 7-10.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 void tukar (int *px, int *py);
11 void tukar(int *px, int *py)
12 {
13     int z;
14     z = *px;
15     *px = *py;
16     *py = z;
17     printf("\nNilai di akhir fungsi tukar()\n");
18     printf("x = %d y = %d\n", *px, *py);
19 }
20 int main(int argc, char *argv[])
21 {
22     int a= 88,b = 77;
23     printf("Nilai sebelum pemanggilan fungsi\n");
24     printf("a = %d b = %d\n", a, b);
25     tukar(&a,&b); //alamat a dan alamat b
26     printf("\nNilai setelah pemanggilan fungsi\n");
27     printf("a = %d b = %d\n", a, b);
28     printf("\n");
29     system("PAUSE");
30     return 0;
31 }
```

```

1  /*
2   Program 7-10
3   Nama File : Program 7-10.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11 void tukar (int *px, int *py);
12 void tukar(int *px, int *py)
13 {
14     int z;
15     z = *px;
16     *px = *py;
17     *py = z;
18     cout<<"\nNilai di akhir fungsi tukar()\n";
19     cout<<"x = "<<*px<<" y = "<<*py<<endl;
20 }
21
22 int main(int argc, char *argv[])
23 {
24     int a = 88, b = 77;
25     cout<<"Nilai sebelum pemanggilan fungsi\n";
26     cout<<"a = "<<a<<" b = "<<b<<endl;
27     tukar(&a,&b);
28     cout<<"\nNilai setelah pemanggilan fungsi\n";
29     cout<<"a = "<<a<<" b = "<<b<<endl;
30     cout<<"\n";
31     system("PAUSE");
32     return EXIT_SUCCESS;
33 }

```

```

F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman I\Kumpula...
Nilai sebelum pemanggilan fungsi
a = 88 b = 77
Nilai di akhir fungsi tukar()
x = 77 y = 88
Nilai setelah pemanggilan fungsi
a = 77 b = 88
Press any key to continue . . .

```

Gambar 7.10 Tampilan Program 7-10

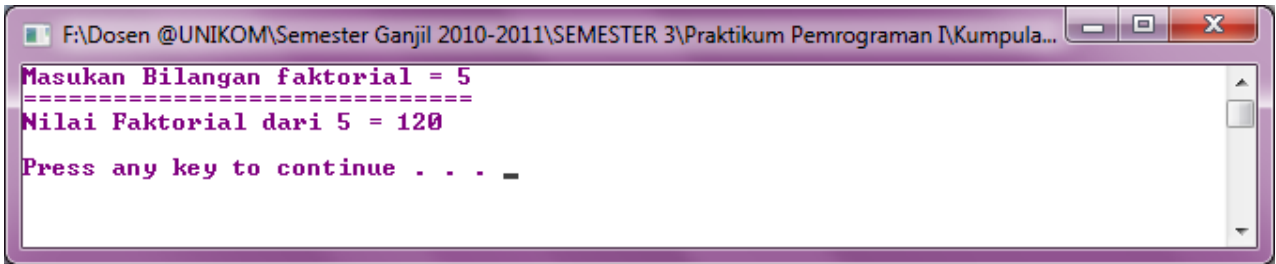
Fungsi Rekursif

Fungsi yang telah dibahas sebelumnya dipanggil dari bagian lain diluar tubuh fungsi yang bersangkutan. Fungsi rekursif adalah suatu fungsi yang memanggil dirinya sendiri, artinya fungsi tersebut dipanggil didalam tubuh fungsi itu sendiri.

```
1  /*
2   Program 7-11
3   Nama File : Program 7-11.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int faktorial (int z);
11 int faktorial (int z)
12 {
13     if(z==1)
14         return(1);
15     else
16         return (z*faktorial(z-1));
17 }
18
19 int main(int argc, char *argv[])
20 {
21     int fak;
22     printf("Masukan Bilangan faktorial = ");scanf("%d",&fak);
23     printf("=====\n");
24     printf("Nilai Faktorial dari %d = %d ", fak, faktorial(fak));
25     printf("\n \n");
26     system("PAUSE");
27     return 0;
28 }
```

```
1  /*
2   Program 7-11
3   Nama File : Program 7-11.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11 int faktorial (int z);
12 int faktorial (int z)
```

```
13 {
14     if(z==1)
15         return(1);
16     else
17         return (z*faktorial(z-1));
18 }
19
20 int main(int argc, char *argv[])
21 {
22     int fak;
23     cout<<"Masukan Bilangan faktorial = ";cin>>fak;
24     cout<<"=====\n";
25     cout<<"Nilai Faktorial dari "<<fak<<" = "<< faktorial(fak);
26     cout<<"\n \n";
27     system("PAUSE");
28     return EXIT_SUCCESS;
29 }
```



Gambar 7.11 Tampilan Program 7-11