

	DIKTAT MATA KULIAH PEMROGRAMAN I BAB VII FUNCTION (FUNGSI)	IF
---	---	-----------

Pendahuluan Fungsi

Dalam pemrograman C/C++ program merupakan kumpulan dari fungsi-fungsi baik yang didefinisikan langsung dalam program maupun yang disimpan di dalam file header. Sebelumnya juga ada sebuah fungsi yang pasti digunakan di dalam bahasa C/C++ yaitu fungsi main. Fungsi main ini harus ada di setiap program dalam bahasa C/C++ karena fungsi inilah yang akan dijalankan oleh compiler.

Fungsi merupakan blok dari kode program yang dirancang untuk melaksanakan tugas khusus. Sebuah fungsi berisi sejumlah pernyataan yang dikemas dalam sebuah nama. Nantinya, nama ini yang akan dipanggil dalam fungsi main. Tujuan pembuatan fungsi adalah sebagai berikut:

1. Memudahkan dalam mengembangkan program.
2. Menghemat ukuran program.
3. Program menjadi lebih terstruktur.
4. Dapat mengurangi pengulangan kode.

Yang harus dipahami, sebuah fungsi harus mempunyai satu nilai balikan (**return value**) oleh karena itu fungsi harus mempunyai tipe data. Bentuk umum fungsi adalah sebagai berikut:

```
tipe_data nama_fungsi(parameter)
{
    //badan_fungsi
    return nilai_return
}
```

Di dalam pemrograman C/C++ ada dua macam fungsi yaitu:

1. User-defined function
Fungsi-fungsi yang didefinisikan sendiri oleh seorang programmer.
2. Built-in function
Fungsi-fungsi "**siap pakai**" yang telah disediakan oleh compiler.

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Fungsi juga bisa dibedakan berdasarkan nilai balik yaitu:

1. Fungsi tanpa nilai balik (Procedure)
2. Fungsi dengan nilai balik.

Prototipe Fungsi

Ada dua cara menggunakan sebuah fungsi di dalam suatu program yaitu:

1. Menggunakan fungsi tanpa menyertakan prototipe fungsinya

Cara yang pertama ini bisa digunakan dengan syarat fungsi yang dibuat harus diletakkan di atas fungsi main. Apabila fungsi yang dibuat diletakkan setelah fungsi main maka fungsi tersebut dianggap tidak terdefinisi. Bentuk umumnya adalah sebagai berikut:

```
tipe_data nama_fungsi(parameter)
{
    //badan_fungsi
    return nilai_return
}

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

2. Menggunakan fungsi dengan menyertakan prototipe fungsinya

Prototipe fungsi adalah kode program yang digunakan untuk mendeklarasikan sebuah fungsi yang dibuat oleh programmer. Dengan menggunakan cara yang kedua ini maka seorang programmer dapat meletakkan fungsi yang dibuat di manapun. Bentuk umumnya adalah sebagai berikut:

```
//prototipe fungsi
tipe_data nama_fungsi1(parameter);
tipe_data nama_fungsi2(parameter);

tipe_data nama_fungsi1(parameter)
{
    //badan_fungsi
    return nilai_return
}

main()
{
    nama_fungsi1(parameter);
    nama_fungsi2(parameter);
    return 0;
}

tipe_data nama_fungsi2(parameter)
{
    //badan_fungsi
    return nilai_return
}
```

Fungsi Tanpa Nilai Balik (Procedure)

Pemrograman C/C++ **tidak** mengenal istilah **prosedur** seperti yang dipelajari di algoritma. Lalu bagaimana kalau seorang programmer ingin membuat sebuah fungsi yang tidak mengembalikan suatu nilai? Fungsi seperti ini dalam pemrograman C/C++ dikenal dengan fungsi yang bertipe **void**. Berarti dalam hal ini fungsi tersebut tidak memiliki nilai balik. Bentuk umum dari fungsi ini adalah sebagai berikut:

```
//prototipe fungsi
void nama_fungsi(parameter1,parameter2);

void nama_fungsi(parameter1,parameter2)
{
    //badan_fungsi
}

main()
{
    // Pemanggilan fungsi
    nama_fungsi(nilai_parameter1,nilai_parameter2);
    return 0;
}
```

Parameter pada fungsi ini bukanlah suatu harga mati maksudnya fungsi void ini tidak harus selalu menggunakan parameter. Tetapi alangkah lebih baiknya apabila dalam pembuatan dan pemanggilan sebuah fungsi disertai dengan parameternya agar memperjelas variabel yang akan diolah di dalam fungsi tersebut. Sebagai contoh buatlah program dengan sintaks seperti di bawah ini:

1. Fungsi void tanpa parameter

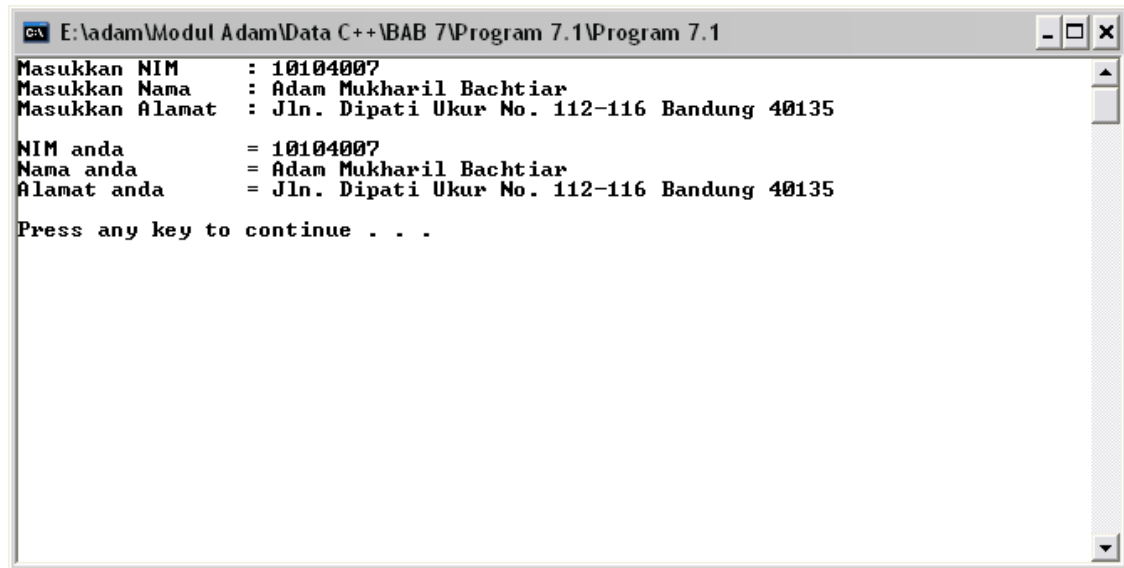
```
1  /*
2      Program 7.1
3      Nama File   : Lat-7.1.c
4      Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 //prototipe fungsi
11 void tampil_nama();
12
13 void tampil_nama()
14 {
15     int nim;
16     char nama[31],alamat[51];
17     printf("Masukkan NIM      : ");scanf("%i",&nim);
18     printf("Masukkan Nama    : ");fflush(stdin);
19     gets(nama);
```

```
20 printf("Masukkan Alamat : ");fflush(stdin);
   gets(alamat);
21 printf("\n");
22 printf("NIM anda          = %i\n\a",nim);
23 printf("Nama anda         = %s\n\a",nama);
24 printf("Alamat anda       = %s\n\n\a",alamat);
25 }
26
27 int main(int argc, char *argv[])
28 {
29     tampil_nama();
30     system("PAUSE");
31     return 0;
32 }
```

```
1  /*
2   Program 7.1
3   Nama File   : Lat-7.1.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5   */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 //prototipe fungsi
13 void tampil_nama();
14
15 void tampil_nama()
16 {
17     int nim;
18     char nama[31],alamat[51];
19     cout<<"Masukkan NIM      : ";cin>>nim;
20     cout<<"Masukkan Nama      : ";fflush(stdin);
   cin.get(nama,30);
21     cout<<"Masukkan Alamat    : ";fflush(stdin);
   cin.get(alamat,50);
22     cout<<endl;
23     cout<<"NIM anda           = "<<nim<<endl<<"\a";
24     cout<<"Nama anda          = "<<nama<<endl<<"\a";
25     cout<<"Alamat anda        = "<<alamat
   <<endl<<endl<<"\a";
26 }
27
```

```
28 int main(int argc, char *argv[])
29 {
30     tampil_nama();
31     system("PAUSE");
32     return EXIT_SUCCESS;
33 }
```

Hasil eksekusi:



```
E:\adam\Modul Adam\Data C++\BAB 7\Program 7.1\Program 7.1
Masukkan NIM      : 10104007
Masukkan Nama     : Adam Mukharil Bachtiar
Masukkan Alamat   : Jln. Dipati Ukur No. 112-116 Bandung 40135

NIM anda          = 10104007
Nama anda         = Adam Mukharil Bachtiar
Alamat anda       = Jln. Dipati Ukur No. 112-116 Bandung 40135
Press any key to continue . . .
```

Gambar 7.1 Hasil eksekusi program Lat 7.1

2. Fungsi void menggunakan parameter

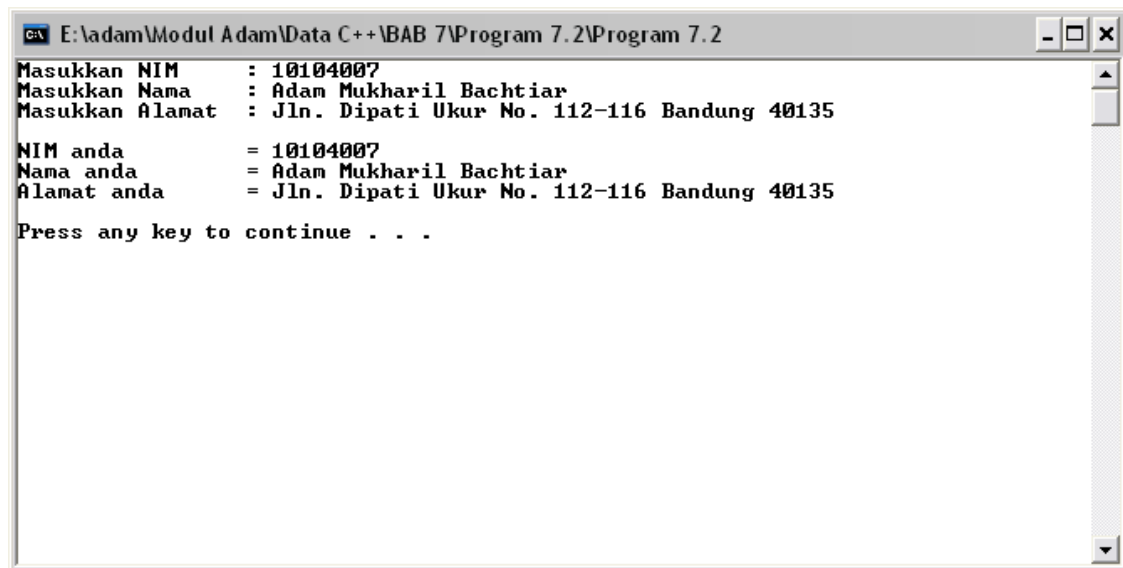
```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  //prototipe fungsi
5  void tampil_nama(int nim,char nama[31],char alamat[51]);
6
7  void tampil_nama(int nim,char nama[31],char alamat[51])
8  {
9      printf("NIM anda          = %i\n\a",nim);
10     printf("Nama anda         = %s\n\a",nama);
11     printf("Alamat anda       = %s\n\n\a",alamat);
12 }
13
14 int main(int argc, char *argv[])
15 {
16     int nim_isi;
17     char nama_isi[31],alamat_isi[51];
18     printf("Masukkan NIM      : ");scanf("%i",&nim_isi);
```

```
19     printf("Masukkan Nama      : ");fflush(stdin);
    gets(nama_isi);
20     printf("Masukkan Alamat   : ");fflush(stdin);
    gets(alamat_isi);
21     printf("\n");
22     tampil_nama(nim_isi,nama_isi,alamat_isi);
23     system("PAUSE");
24     return 0;
25 }
```

```
1  /*
2   Program 7.2
3   Nama File   : Lat-7.2.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 //prototipe fungsi
13 void tampil_nama(int nim,char nama[31],char alamat[51]);
14
15 void tampil_nama(int nim,char nama[31],char alamat[51])
16 {
17     cout<<"NIM anda           = "<<nim<<endl<<"\a";
18     cout<<"Nama anda          = "<<nama<<endl<<"\a";
19     cout<<"Alamat anda        = "<<alamat
        <<endl<<endl<<"\a";
20 }
21
22 int main(int argc, char *argv[])
23 {
24     int nim_isi;
25     char nama_isi[31],alamat_isi[51];
26     cout<<"Masukkan NIM      : ";cin>>nim_isi;
27     cout<<"Masukkan Nama      : ";fflush(stdin);
        cin.get(nama_isi,30);
28     cout<<"Masukkan Alamat   : ";fflush(stdin);
        cin.get(alamat_isi,50);
29     cout<<endl;
30     tampil_nama(nim_isi,nama_isi,alamat_isi);
31     system("PAUSE");
32     return EXIT_SUCCESS;
33 }
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Hasil eksekusi:



```
E:\adam\Modul Adam\Data C++\BAB 7\Program 7.2\Program 7.2
Masukkan NIM      : 10104007
Masukkan Nama     : Adam Mukharil Bachtiar
Masukkan Alamat   : Jln. Dipati Ukur No. 112-116 Bandung 40135
NIM anda          = 10104007
Nama anda         = Adam Mukharil Bachtiar
Alamat anda       = Jln. Dipati Ukur No. 112-116 Bandung 40135
Press any key to continue . . .
```

Gambar 7.2 Hasil eksekusi program Lat 7.2

Contoh lain untuk fungsi bertipe void adalah fungsi untuk membuat kotak yang sintaksnya sebagai berikut:

```
1  /*
2   Program 7.3
3   Nama File   : Lat-7.3.c
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9  #include <conio2.h>
10
11 //prototipe fungsi
12 void Kotak(int X1,int Y1, int X2,int Y2,int Bingkai,
13           int Latar);
14 void Kotak(int X1,int Y1, int X2,int Y2,int Bingkai,
15           int Latar)
16 {
17     int i;
18     textcolor(Bingkai);
19     textbackground(Latar);
20     gotoxy(X1,Y1);printf("Ú");//untuk simbol Ú cari di
    simbol M.Word dengan Character Code 218 from
    ASCII(Decimal) lalu copy-kan
    gotoxy(X1,Y2);printf("%c",192);//ASCII Decimal 192
```

```
21 gotoxy(X2,Y1);printf("%c",191);
22 gotoxy(X2,Y2);printf("%c",217);
23 for (i=X1+1;i<=X2-1;i++)
24 {
25     gotoxy(i,Y1);printf("%c",196);
26     gotoxy(i,Y2);printf("%c",196);
27 }
28
29 for(i=Y1+1;i<=Y2-1;i++)
30 {
31     gotoxy(X1,i);printf("%c",179);
32     gotoxy(X2,i);printf("%c",179);
33 }
34 }
35
36 int main(int argc, char *argv[])
37 {
38     //warna background
39     textbackground(WHITE);
40     clrscr();
41     Kotak(1,1,80,24,WHITE,BLUE);
42     Kotak(2,2,15,23,WHITE,RED);
43     textbackground(WHITE);
44     textcolor(BLACK);
45     gotoxy(1,25);system("PAUSE");
46     return 0;
47 }
```

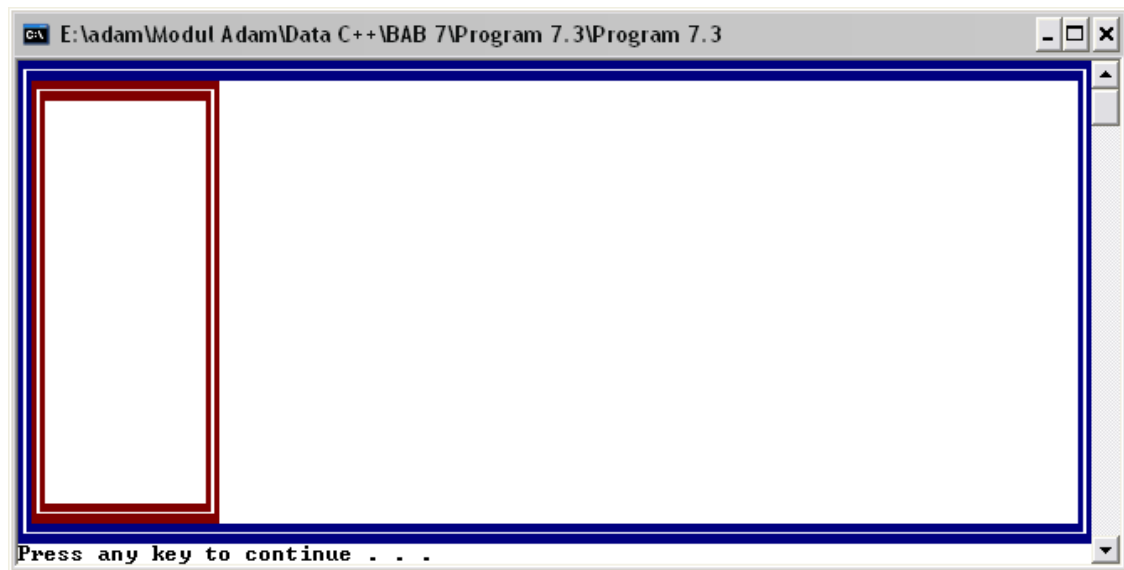
```
1  /*
2      Program 7.3
3      Nama File   : Lat-7.3.cpp
4      Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9  #include <conio2.h>
10
11 using namespace std;
12
13 //prototipe fungsi
14 void Kotak(int X1,int Y1, int X2,int Y2,int Bingkai,
15           int Latar);
16 void Kotak(int X1,int Y1, int X2,int Y2,int Bingkai,
```

```

                                int Latar)
17 {
18     int i;
19     textcolor(Bingkai);
20     textbackground(Latar);
21     gotoxy(X1,Y1);cout<<"Ú";//untuk simbol Ú cari di
    simbol M.Word dengan Character Code 218 from
    ASCII(Decimal) lalu copy-kan
22     gotoxy(X1,Y2);cout<<"\xC0";
23     gotoxy(X2,Y1);cout<<"\xBF";
24     gotoxy(X2,Y2);cout<<"\xD9";
25     for (i=X1+1;i<=X2-1;i++)
26     {
27         gotoxy(i,Y1);cout<<"\xC4";
28         gotoxy(i,Y2);cout<<"\xC4";
29     }
30
31     for(i=Y1+1;i<=Y2-1;i++)
32     {
33         gotoxy(X1,i);cout<<"\xB3";
34         gotoxy(X2,i);cout<<"\xB3";
35     }
36 }
37
38 int main(int argc, char *argv[])
39 {
40     //warna background
41     textbackground(WHITE);
42     clrscr();
43     Kotak(1,1,80,24,WHITE,BLUE);
44     Kotak(2,2,15,23,WHITE,RED);
45     textbackground(WHITE);
46     textcolor(BLACK);
47     gotoxy(1,25);system("PAUSE");
48     return EXIT_SUCCESS;
49 }
```


DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Hasil eksekusi:



Gambar 7.3 Hasil eksekusi program Lat 7.3

Void kotak merupakan sebuah fungsi yang akan membuat kotak di layar sesuai dengan koordinat yang diberikan di bagian parameter. Koordinat tersebut adalah koordinat kiri atas (X1,Y1) dan koordinat titik kanan bawah (X2,Y2). Selain itu ada parameter Bingkai yang berfungsi untuk menentukan warna bingkai serta parameter Layar untuk menentukan warna latar belakang kotak yang dibuat.

<p>Catatan</p> 	<p>Parameter dalam sebuah fungsi memang buka sebuah harga mati akan tetapi sangat disarankan penggunaan parameter dalam pembuatan dan pemanggilan fungsi untuk memperjelas penggunaan variabel dari sebuah fungsi. Untuk parameter tetap ada yang bersifat global maupun lokal seperti yang dijelaskan di Bab III.</p>
--	--

Fungsi dengan Nilai Balik

Berbeda dengan fungsi yang bertipe **void**, fungsi ini berguna untuk melakukan suatu proses yang dapat mengembalikan suatu nilai. Dalam membuat fungsi ini kita perlu untuk memilih tipe data yang akan menjadi tipe data untuk nilai balik dari fungsi yang dibuat. Bentuk umum fungsi tersebut adalah sebagai berikut:

```
//prototipe fungsi
tipe_data nama_fungsi(parameter1,parameter2);

tipe_data nama_fungsi(parameter1,parameter2)
{
    //badan_fungsi
    return nilai_yang_dikembalikan;
}

main()
{
    // pemanggilan fungsi
    nama_fungsi(nilai_parameter1,nilai_parameter2);
    return 0;
}
```

Fungsi yang Mengembalikan Tipe String

Berikut ini adalah contoh sebuah fungsi yang mengembalikan data bertipe string.

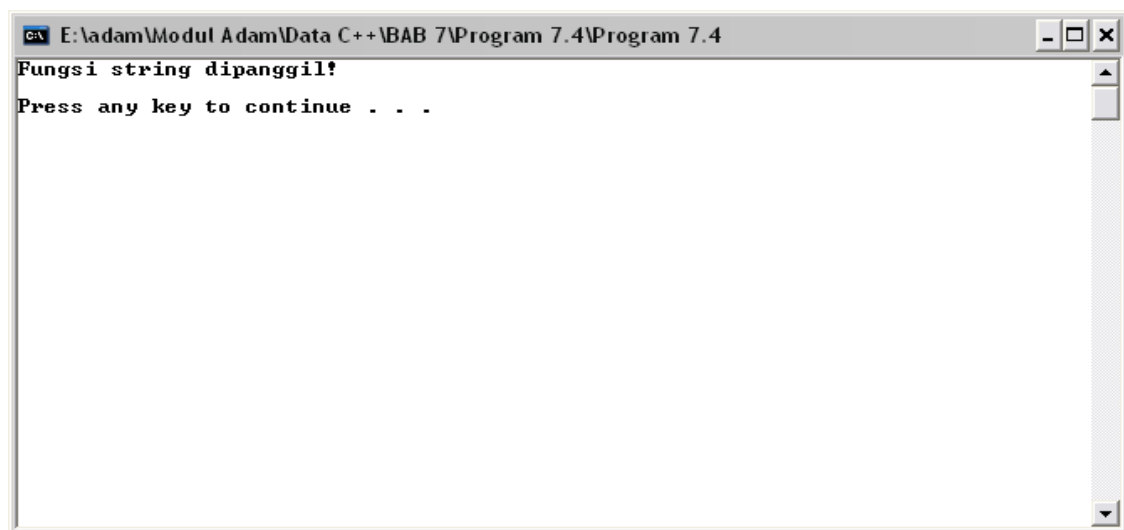
```
1  /*
2      Program 7.4
3      Nama File   : Lat-7.4.c
4      Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 char *nama();
11
12 char *nama()
13 {
14     return "Fungsi string dipanggil!";
15 }
16
17 int main(int argc, char *argv[])
18 {
19     printf("%s\n\n",nama());
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

```
20    system("PAUSE");
21    return 0;
22 }
```

```
1  /*
2   Program 7.4
3   Nama File   : Lat-7.4.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 char *nama();
13
14 char *nama()
15 {
16     return "Fungsi string dipanggil!";
17 }
18
19 int main(int argc, char *argv[])
20 {
21     cout<<nama()<<endl<<endl;
22     system("PAUSE");
23     return EXIT_SUCCESS;
24 }
```

Hasil eksekusi:



```

E:\adam\Modul Adam\Data C++\BAB 7\Program 7.4\Program 7.4
Fungsi string dipanggil!
Press any key to continue . . .
```

Gambar 7.4 Hasil eksekusi program Lat 7.4

Fungsi yang Mengembalikan Tipe Bilangan

Berikut ini adalah contoh sebuah fungsi yang mengembalikan data bertipe bilangan.

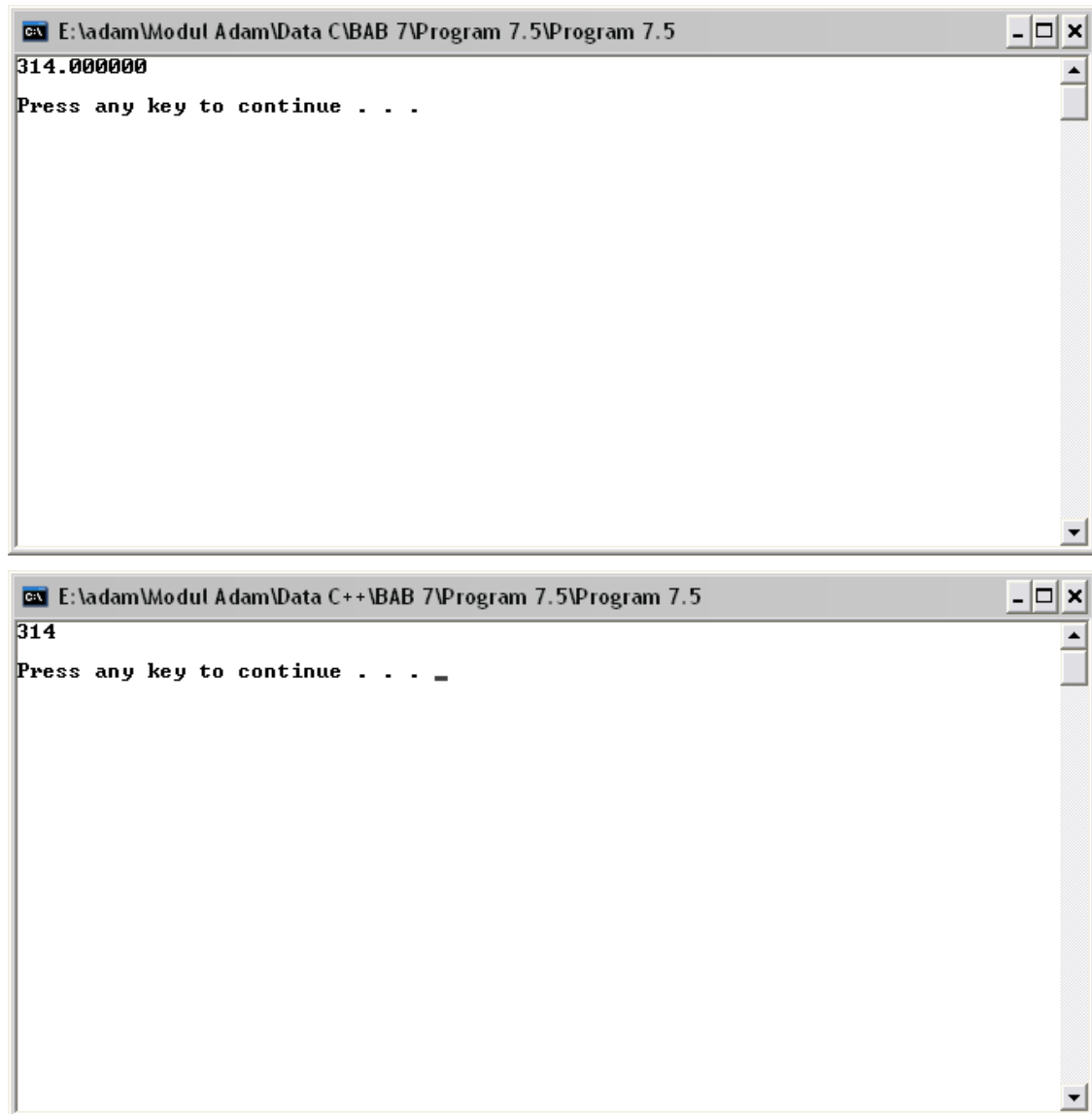
```
1  /*
2     Program 7.5
3     Nama File   : Lat-7.5.c
4     Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 float luas_lingkaran();
11
12 float luas_lingkaran()
13 {
14     return(3.14*10*10); //jari2 = 10
15 }
16
17 int main(int argc, char *argv[])
18 {
19     printf("%f\n\n", luas_lingkaran());
20     system("PAUSE");
21     return 0;
22 }
```

```
1  /*
2     Program 7.5
3     Nama File   : Lat-7.5.cpp
4     Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 float luas_lingkaran();
13
14 float luas_lingkaran()
15 {
16     return(3.14*10*10); //jari2 = 10
17 }
18
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

```
19 int main(int argc, char *argv[])
20 {
21     cout<<luas_lingkaran()<<endl<<endl;
22     system("PAUSE");
23     return EXIT_SUCCESS;
24 }
```

Hasil eksekusi:



Gambar 7.5 Hasil eksekusi program Lat 7.5 dalam bahasa C dan C++

Fungsi Bertipe dengan Parameter

Kedua buah fungsi sebelum sub bab ini adalah contoh dari fungsi bertipe tanpa parameter. Dalam sub bab ini akan dibuat fungsi bertipe dengan menggunakan parameter tetapi sebelum itu ada tiga jenis parameter yang bisa digunakan di dalam pembuatan fungsi, yaitu:

1. Parameter masukan
2. Parameter keluaran
3. Parameter masukan/keluaran.

Parameter Masukan

Sesuai dengan namanya, parameter ini digunakan sebagai nilai masukan dalam sebuah fungsi. Nilai tersebut kemudian akan diproses oleh fungsi untuk menghasilkan sebuah nilai balik. Untuk lebih jelasnya buatlah program dengan sintaks di bawah ini:

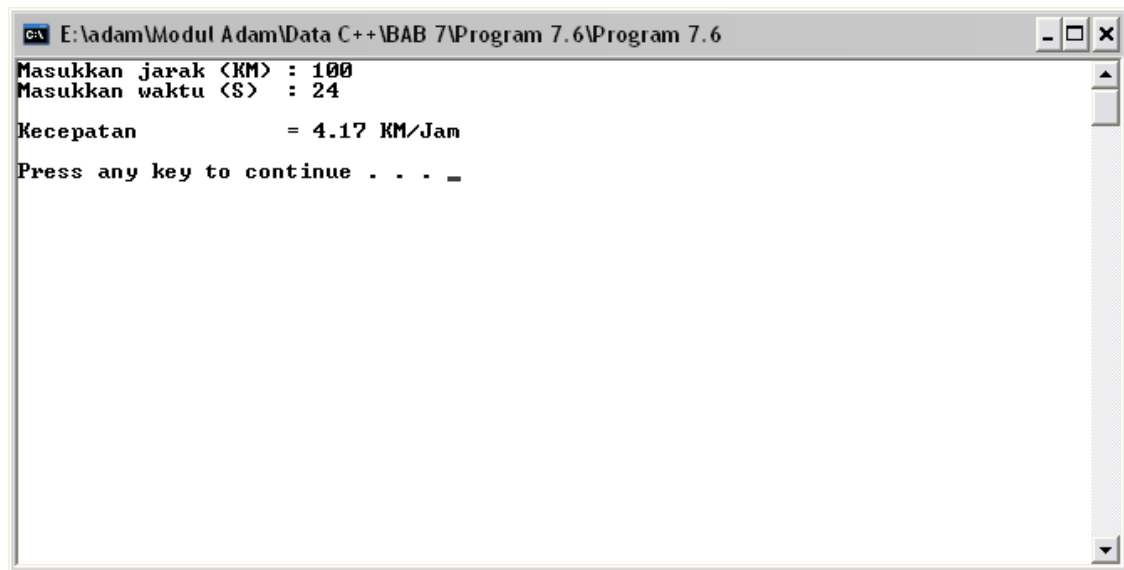
```
1  /*
2   Program 7.6
3   Nama File   : Lat-7.6.c
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 float kecepatan(float s, float t);
11
12 float kecepatan(float s, float t)
13 {
14     float v;
15     v=s/t;
16     return v;
17 }
18
19 int main(int argc, char *argv[])
20 {
21     float s_isi,t_isi,v;
22     printf("Masukkan jarak (KM) : ");scanf("%f",&s_isi);
23     printf("Masukkan waktu (S)   : ");scanf("%f",&t_isi);
24     v=kecepatan(s_isi,t_isi);
25     printf("\nKecepatan           = %.2f KM/Jam\n\n",v);
```

```
26     system("PAUSE");
27     return 0;
28 }
```

```
1  /*
2   Program 7.6
3   Nama File   : Lat-7.6.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5   */
6
7  #include <cstdlib>
8  #include <iostream>
9  #include <iomanip>
10
11 using namespace std;
12
13 float kecepatan(float s, float t);
14
15 float kecepatan(float s, float t)
16 {
17     float v;
18     v=s/t;
19     return v;
20 }
21
22 int main(int argc, char *argv[])
23 {
24     float s_isi,t_isi,v;
25     cout<<"Masukkan jarak (KM) : ";cin>>s_isi;
26     cout<<"Masukkan waktu (S)   : ";cin>>t_isi;
27     v=kecepatan(s_isi,t_isi);
28     cout<<endl;
29     cout<<setiosflags(ios::fixed);
30     cout<<"Kecepatan              = "<<setprecision(2)
31          <<v<<" KM/Jam"<<endl<<endl;
32     system("PAUSE");
33     return 0;
34 }
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Hasil eksekusi:



```
E:\adam\Modul Adam\Data C++\BAB 7\Program 7.6\Program 7.6
Masukkan jarak (KM) : 100
Masukkan waktu (S) : 24
Kecepatan          = 4.17 KM/Jam
Press any key to continue . . . _
```

Gambar 7.6 Hasil eksekusi program Lat 7.6

Variabel s dan variabel t pada fungsi kecepatan berfungsi sebagai parameter masukan.

Parameter Keluaran

Parameter keluaran adalah parameter yang berfungsi untuk menampung nilai yang dihasilkan dari proses di dalam fungsi. Parameter ini harus dilewatkan berdasarkan alamat atau referensinya bukan nilainya. Parameter keluaran biasanya digunakan untuk fungsi bertipe **void**. Untuk lebih jelasnya buatlah program dengan sintaks di bawah ini:

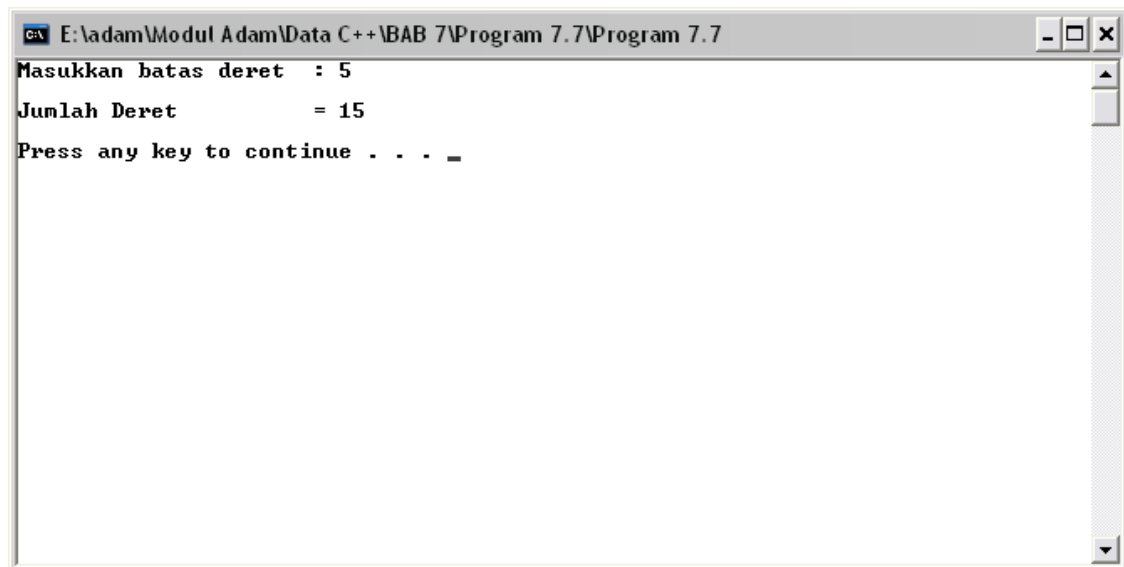
```
1  /*
2   Program 7.7
3   Nama File   : Lat-7.7.c
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 void jumlah_deret(int batas, int *jumlah);
11
```

```
12 void jumlah_deret(int batas,int *jumlah)
13 {
14     int x;
15     *jumlah=0;
16     for(x=1;x<=batas;x++)
17     {
18         *jumlah=*jumlah+x;
19     }
20 }
21
22 int main(int argc, char *argv[])
23 {
24     int batas,jumlah;
25     printf("Masukkan batas deret  : ");scanf("%i",&batas);
26     jumlah_deret(batas,&jumlah);
27     printf("%i\n\n",jumlah);
28     system("PAUSE");
29     return 0;
30 }
```

```
1  /*
2      Program 7.7
3      Nama File   : Lat-7.7.cpp
4      Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 void jumlah_deret(int batas, int *jumlah);
13
14 void jumlah_deret(int batas,int *jumlah)
15 {
16     int x;
17     *jumlah=0;
18     for(x=1;x<=batas;x++)
19     {
20         *jumlah=*jumlah+x;
21     }
22 }
23
24 int main(int argc, char *argv[])
25 {
26     int batas,jumlah;
27     cout<<"Masukkan batas deret  : ";cin>>batas;
```

```
28     jumlah_deret(batas,&jumlah);  
29     cout<<endl;  
30     cout<<"Jumlah Deret          = "<<jumlah  
        <<endl<<endl;  
31     system("PAUSE");  
32     return EXIT_SUCCESS;  
33 }
```

Hasil eksekusi:

A screenshot of a Windows command prompt window titled "E:\adam\Modul Adam\Data C++\BAB 7\Program 7.7\Program 7.7". The window shows the output of a C++ program. The first line is "Masukkan batas deret : 5". The second line is "Jumlah Deret = 15". The third line is "Press any key to continue . . . _". The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Gambar 7.7 Hasil eksekusi program Lat 7.7

Parameter jumlah yang ada di dalam fungsi jumlah_deret adalah parameter keluaran. Apabila lebih diperhatikan lagi ketika pemanggilan fungsi jumlah_deret, parameter yang diisikan untuk posisi int *jumlah berisikan variabel yang diawali dengan tanda &. Hal itu digunakan untuk mengakses alamat dari suatu variabel yang menyimpan nilai(akan dipelajari lebih lanjut di Bab **Pointer**).

Parameter Masukan/Keluaran

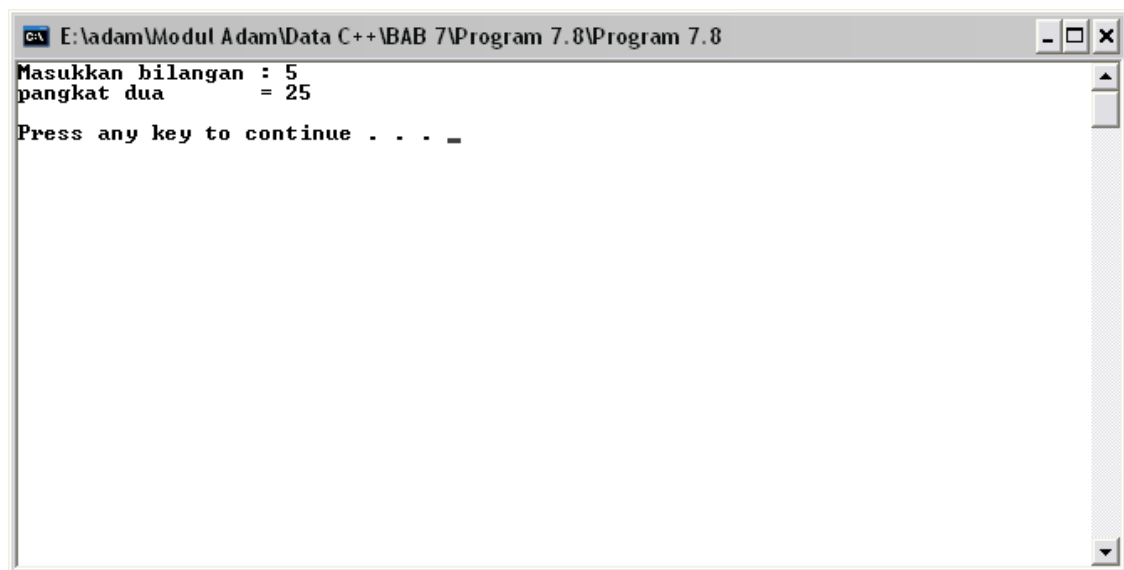
Parameter ini bisa digunakan sebagai parameter masukan ataupun parameter keluaran. Artinya parameter ini akan digunakan sebagai masukan untuk proses di dalam fungsi dan setelah proses tersebut selesai hasilnya akan disimpan kembali ke dalam parameter tersebut sebagai nilai keluaran. Untuk lebih jelasnya buat program dengan sintaks di bawah ini:

```
1  /*
2      Program 7.8
3      Nama File   : Lat-7.8.c
4      Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9  #include <math.h>
10
11 void pangkat_dua(float *bil);
12
13 void pangkat_dua(float *bil)
14 {
15     *bil=pow(*bil,2);
16 }
17 int main(int argc, char *argv[])
18 {
19     float bilangan;
20     printf("Masukkan bilangan : ");scanf("%f",&bilangan);
21     pangkat_dua(&bilangan);
22     printf("pangkat dua          = %.0f\n\n",bilangan);
23     system("PAUSE");
24     return 0;
25 }
```

```
1  /*
2      Program 7.8
3      Nama File   : Lat-7.8.cpp
4      Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9  #include <math.h>
10 #include <iomanip>
11
12 using namespace std;
13
14 void pangkat_dua(float *bil);
15
16 void pangkat_dua(float *bil)
17 {
18     *bil=pow(*bil,2);
19 }
20
21 int main(int argc, char *argv[])
```

```
22 {  
23     float bilangan;  
24     cout<<"Masukkan bilangan : ";cin>>bilangan;  
25     pangkat_dua(&bilangan);  
26     cout<<setiosflags(ios::fixed);  
27     cout<<"pangkat dua      = "<<setprecision(0)  
28         <<bilangan<<endl<<endl;  
29     system("PAUSE");  
30     return EXIT_SUCCESS;  
31 }
```

Hasil eksekusi:



```
E:\adam\Modul Adam\Data C++\BAB 7\Program 7.8\Program 7.8  
Masukkan bilangan : 5  
pangkat dua      = 25  
Press any key to continue . . . _
```

Gambar 7.8 Hasil eksekusi program Lat 7.8

Parameter Formal dan Aktual

Jenis-jenis Parameter terbagi menjadi dua yaitu:

1. Parameter Formal

Parameter yang dideklarasikan pada saat pembuatan sebuah prototipe fungsi atau fungsi. Parameter inilah yang akan diisi dengan nilai, variabel, atau alamat ketika pemanggilan fungsinya.

2. Parameter Aktual

Parameter yang dideklarasikan dan digunakan ketika pemanggilan fungsi. Penamaan variabel untuk parameter ini bisa sama dengan nama variabel

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

parameter formalnya atau boleh juga berbeda dengan nama variabel parameter formalnya.

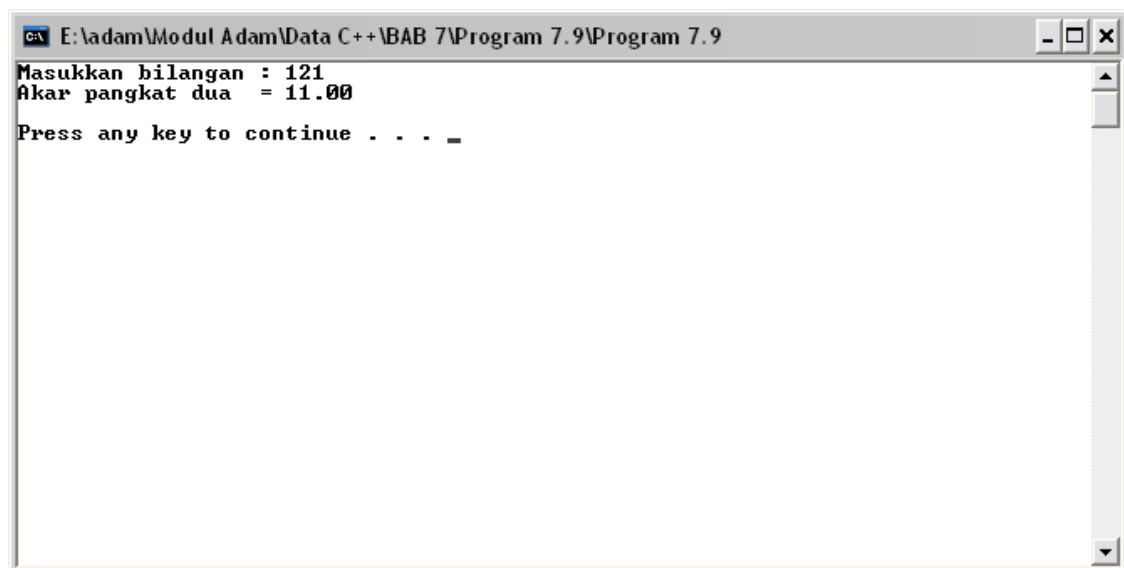
Untuk lebih jelasnya buat program dengan sintaks di bawah ini:

```
1  /*
2     Program 7.9
3     Nama File   : Lat-7.9.c
4     Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9  #include <math.h>
10
11 float akar_pangkat(int bil);
12
13 float akar_pangkat(int bil)//bil di sini parameter
                             formal
14 {
15     return (sqrt(bil));
16 }
17
18 int main(int argc, char *argv[])
19 {
20     int bil;
21     printf("Masukkan bilangan : ");scanf("%i",&bil);
22     printf("Akar pangkat dua = %.2f\n\n",
           akar_pangkat(bil));//bil di sini parameter
                             aktual
23     system("PAUSE");
24     return 0;
25 }
```

```
1  /*
2     Program 7.9
3     Nama File   : Lat-7.9.cpp
4     Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9  #include <math.h>
10 #include <iomanip>
11
12 using namespace std;
13
14 float akar_pangkat(int bil);
```

```
15
16 float akar_pangkat(int bil)//bil di sini parameter
    formal
17 {
18     return (sqrt(bil));
19 }
20
21 int main(int argc, char *argv[])
22 {
23     int bil;
24     cout<<"Masukkan bilangan : ";cin>>bil;
25     cout<<setiosflags(ios::fixed);
26     cout<<"Akar pangkat dua = "<<setprecision(2)
        <<akar_pangkat(bil)<<endl<<endl;
        //bil di sini parameter aktual
27     system("PAUSE");
28     return EXIT_SUCCESS;
29 }
```

Hasil eksekusi:

A screenshot of a Windows command prompt window titled "E:\adam\Modul Adam\Data C++\BAB 7\Program 7.9\Program 7.9". The window shows the output of the program: "Masukkan bilangan : 121", "Akar pangkat dua = 11.00", and "Press any key to continue . . . _". The window has standard Windows controls (minimize, maximize, close) in the top right corner.

Gambar 7.9 Hasil eksekusi program Lat 7.9

Keterangan:

1. Variabel bil yang ada di prototipe fungsi dan pembuatan fungsi akar_akar_pangkat merupakan parameter formal.
2. Variabel bil yang ada di fungsi main ketika pemanggilan fungsi akar_pangkat merupakan parameter aktual.

Melewatkan Parameter Di Fungsi

Parameter bisa dilewatkan di dalam sebuah fungsi dengan dua cara yaitu:

1. Berdasarkan nilai (pass by value)

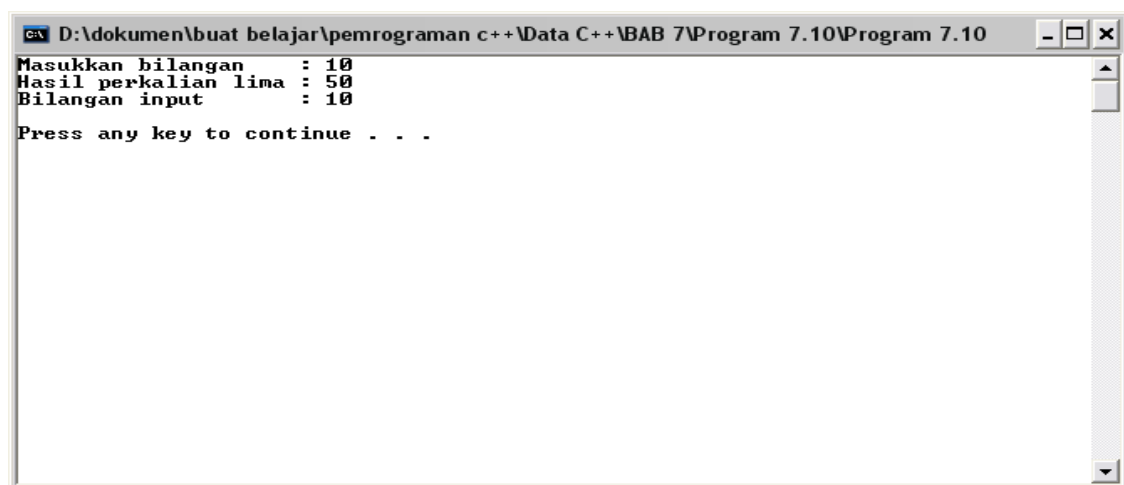
Nilai yang dihasilkan oleh suatu fungsi tidak akan mempengaruhi nilai di dalam program (di luar fungsi tersebut). Hal ini disebabkan karena dalam pemanggilan fungsi, compiler hanya membuat salinan dari nilai yang ada di parameter aktual ke parameter formal. Untuk lebih jelasnya buat program dengan sintaks di bawah ini.

```
1  /*
2   Program 7.10
3   Nama File   : Lat-7.10.c
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 void perkalian_lima(int bil);
11
12 int main(int argc, char *argv[])
13 {
14     int bil_isi;
15     printf("Masukkan bilangan      : ");
16     scanf("%i",&bil_isi);
17     perkalian_lima(bil_isi);
18     printf("Bilangan input          : %i\n\n",bil_isi);
19     system("PAUSE");
20     return 0;
21 }
22 void perkalian_lima(int bil)
23 {
24     bil=bil*5;
25     printf("Hasil perkalian lima : %i\n",bil);
26 }
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

```
1  /*
2    Program 7.10
3    Nama File   : Lat-7.10.cpp
4    Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 void perkalian_lima(int bil);
13
14 int main(int argc, char *argv[])
15 {
16     int bil_isi;
17     cout<<"Masukkan bilangan      : ";cin>>bil_isi;
18     perkalian_lima(bil_isi);
19     cout<<"Bilangan input        : "<<bil_isi
20         <<endl<<endl;
21     system("PAUSE");
22     return EXIT_SUCCESS;
23 }
24 void perkalian_lima(int bil)
25 {
26     bil=bil*5;
27     cout<<"Hasil perkalian lima : "<<bil<<endl;
28 }
```

Hasil eksekusi:



```

D:\dokumen\buat belajar\pemrograman c++\Data C++\BAB 7\Program 7.10\Program 7.10
Masukkan bilangan : 10
Hasil perkalian lima : 50
Bilangan input : 10
Press any key to continue . . .
```

Gambar 7.10 Hasil eksekusi program Lat 7.10

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Pada kasus tersebut variabel `bil_isi` dan variabel `bil` menempati alamat yang berbeda sehingga pada saat pemanggilan fungsi nilai keduanya akan berbeda. Variabel `bil_isi` akan bernilai tetap (sesuai yang diinput) karena memang program ini tidak melakukan apa-apa terhadap variabel tersebut.

2. Berdasarkan alamat (pass by reference)

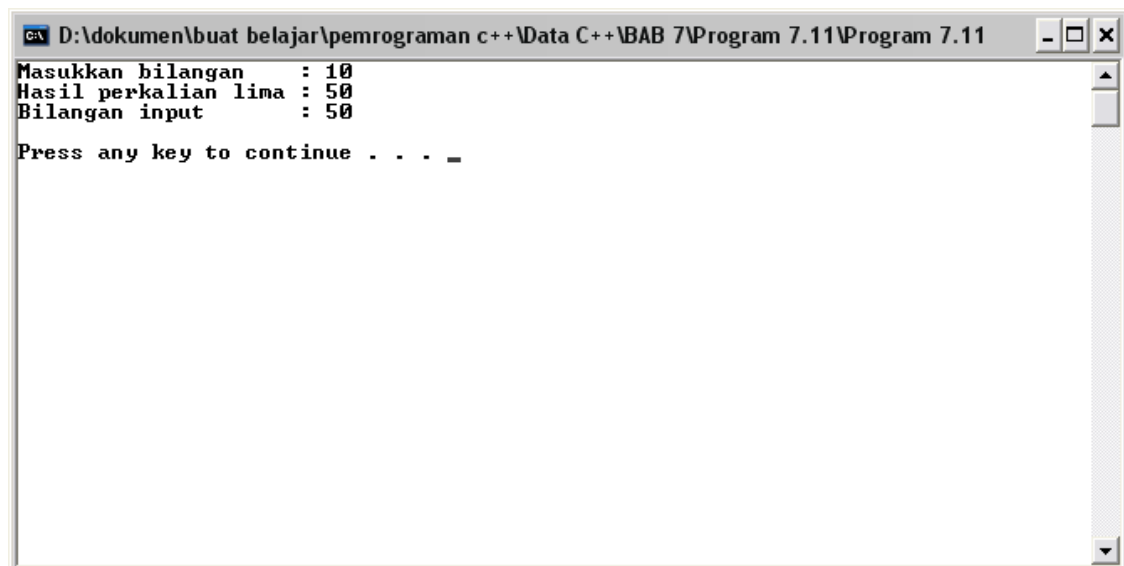
Berbeda dengan parameter pass by value, pass by reference akan melewati parameter ke sebuah fungsi berdasarkan alamatnya. Hal ini akan menyebabkan nilai di dalam fungsi akan sama dengan nilai yang ada di luar fungsi. Untuk lebih jelasnya buat program dengan sintaks di bawah ini.

```
1  /*
2   Program 7.11
3   Nama File   : Lat-7.11.c
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 void perkalian_lima(int *bil);
11
12 int main(int argc, char *argv[])
13 {
14     int bil_isi;
15     printf("Masukkan bilangan      : ");
16     scanf("%i",&bil_isi);
17     perkalian_lima(&bil_isi);
18     printf("Bilangan input          : %i\n\n",bil_isi);
19     system("PAUSE");
20     return 0;
21 }
22 void perkalian_lima(int *bil)
23 {
24     *bil=(*bil)*5;
25     printf("Hasil perkalian lima : %i\n",*bil);
26 }
```

```
1  /*
2   Program 7.11
3   Nama File   : Lat-7.11.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
```

```
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 void perkalian_lima(int *bil);
13
14 int main(int argc, char *argv[])
15 {
16     int bil_isi;
17     cout<<"Masukkan bilangan      : ";cin>>bil_isi;
18     perkalian_lima(&bil_isi);
19     cout<<"Bilangan input        : "<<bil_isi
20         <<endl<<endl;
21     system("PAUSE");
22     return EXIT_SUCCESS;
23 }
24 void perkalian_lima(int *bil)
25 {
26     *bil=(*bil)*5;
27     cout<<"Hasil perkalian lima : "<<*bil<<endl;
28 }
```

Hasil eksekusi:



```

D:\dokumen\buat belajar\pemrograman c++\Data C++\BAB 7\Program 7.11\Program 7.11
Masukkan bilangan : 10
Hasil perkalian lima : 50
Bilangan input : 50
Press any key to continue . . . _
```

Gambar 7.11 Hasil eksekusi program Lat 7.11

Variabel `bil_isi` dan variabel `bil` akan menempati memori yang sama sehingga apabila satu berubah maka yang lain akan berubah juga.

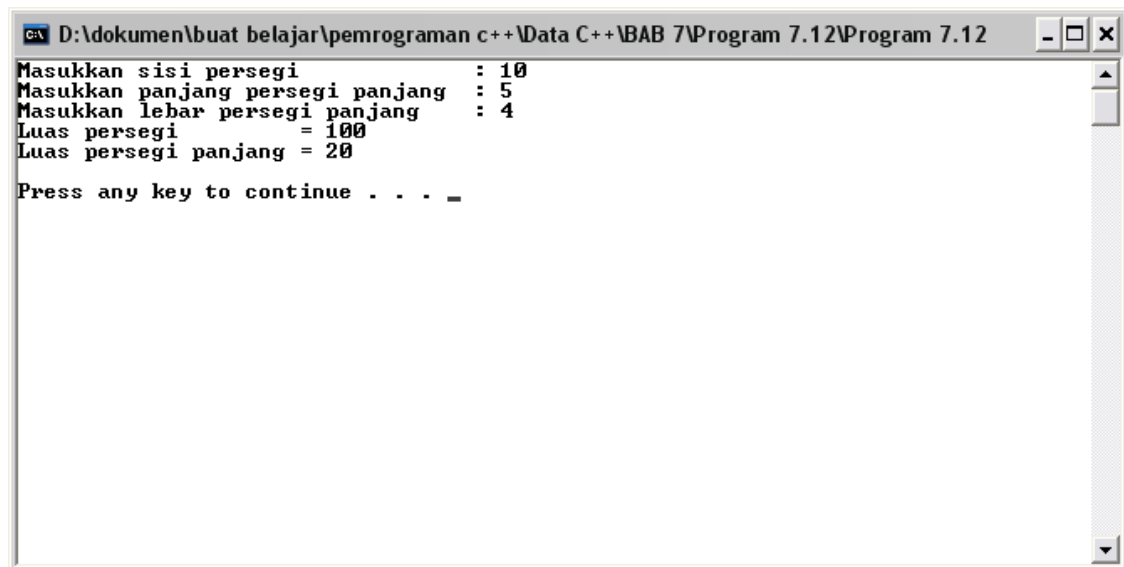
Overloading Function

Overloading function adalah fungsi yang mempunyai nama yang sama akan tetapi mempunyai parameter yang berbeda. Berbeda di sini dapat dilihat dari tiga sisi yaitu secara jumlah, tipe data, maupun gabungan dari keduanya. Salah satu contoh dari overloading function bisa dilihat dari penggunaan fungsi printf dan cout. Kedua fungsi tersebut bisa digunakan untuk menampilkan teks saja ataupun gabungan teks dengan variabel walaupun cara pemanggilannya sama yaitu tinggal memanggil nama fungsi tersebut lalu tinggal dibedakan berdasarkan parameter aktualnya. Tetapi kelebihan ini hanya bisa digunakan untuk bahasa C++. Untuk lebih jelasnya buat program dengan sintaks di bawah ini.

```
1  /*
2   Program 7.12
3   Nama File   : Lat-7.12.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int luas(int sisi);
13 int luas(int panjang,int lebar);
14
15 int main(int argc, char *argv[])
16 {
17     int sisi_isi;
18     int panjang_isi,lebar_isi;
19     cout<<"Masukkan sisi persegi           : ";
20     cin>>sisi_isi;
21     cout<<"Masukkan panjang persegi panjang : ";
22     cin>>panjang_isi;
23     cout<<"Masukkan lebar persegi panjang   : ";
24     cin>>lebar_isi;
25     cout<<"Luas persegi           = "<<luas(sisi_isi)
26     <<endl;
27     cout<<"Luas persegi panjang = "
28         <<luas(panjang_isi,lebar_isi)<<endl<<endl;
29     system("PAUSE");
30     return EXIT_SUCCESS;
31 }
```

```
28 int luas(int sisi)
29 {
30     int hasil;
31     hasil=sisi*sisi;
32     return hasil;
33 }
34
35 int luas(int panjang,int lebar)
36 {
37     int hasil;
38     hasil=panjang*lebar;
39     return hasil;
40 }
```

Hasil eksekusi:



The screenshot shows a Windows command prompt window titled "D:\dokumen\buat belajar\pemrograman c++\Data C++\BAB 7\Program 7.12\Program 7.12". The program prompts the user for input and displays the following output:

```
Masukkan sisi persegi : 10
Masukkan panjang persegi panjang : 5
Masukkan lebar persegi panjang : 4
Luas persegi = 100
Luas persegi panjang = 20
Press any key to continue . . . _
```

Gambar 7.12 Hasil eksekusi program Lat 7.12

Program di atas memiliki dua buah fungsi yang memiliki nama yang sama yaitu fungsi luas. Untuk fungsi luas yang pertama memiliki parameter sisi digunakan untuk menghitung luas persegi sedangkan fungsi luas yang kedua memiliki parameter panjang dan lebar digunakan untuk menghitung luas persegi panjang.

Rekursi Fungsi

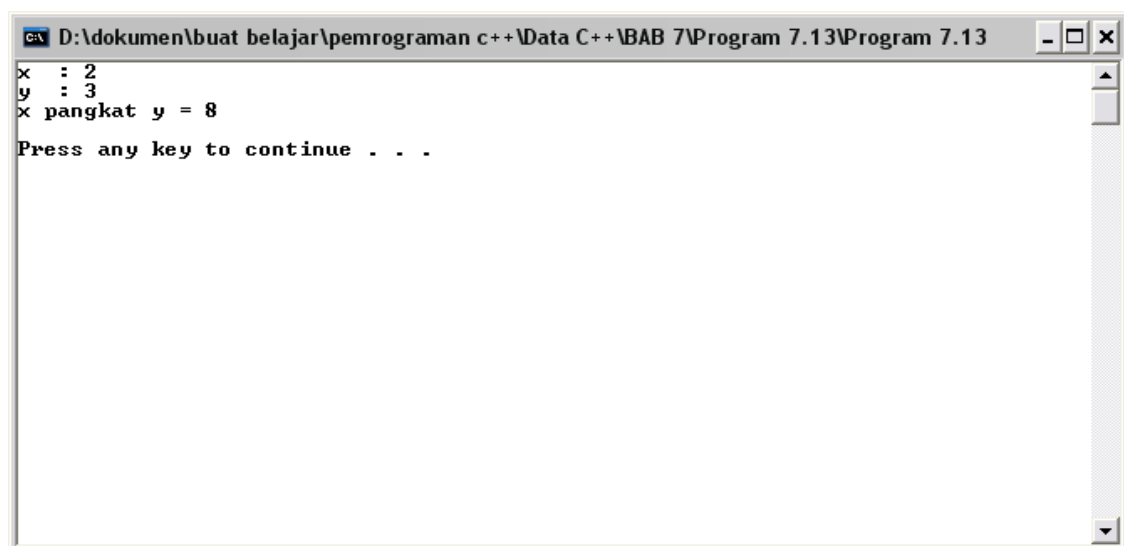
Di dalam pemrograman C/C++ dimungkinkan untuk suatu fungsi memanggil fungsi itu sendiri. Hal tersebut disebut sebagai rekursi fungsi. Sebagai contoh adalah program untuk menghitung x^y . Programnya sebagai berikut:

```
1  /*
2    Program 7.13
3    Nama File   : Lat-7.13.c
4    Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int pangkat(int x,int y);
11
12 int main(int argc, char *argv[])
13 {
14     int a,b;
15     printf("x  : ");scanf("%i",&a);
16     printf("y  : ");scanf("%i",&b);
17     printf("x pangkat y = %i\n\n",pangkat(a,b));
18     system("PAUSE");
19     return 0;
20 }
21
22 int pangkat(int x,int y)
23 {
24     if(y==1)
25     {
26         return x;
27     }
28     else
29     {
30         return (x*pangkat(x,y-1));
31     }
32 }
```

```
1  /*
2    Program 7.13
3    Nama File   : Lat-7.13.cpp
4    Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
```

```
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int pangkat(int x,int y);
13
14 int pangkat(int x,int y)
15 {
16     if(y==1)
17     {
18         return x;
19     }
20     else
21     {
22         return (x*pangkat(x,y-1)); //rekursi fungsi
23     }
24 }
25
26 int main(int argc, char *argv[])
27 {
28     int a,b;
29     cout<<"x  : ";cin>>a;
30     cout<<"y  : ";cin>>b;
31     cout<<"x pangkat y = "<<pangkat(a,b)<<endl<<endl;
32     system("PAUSE");
33     return EXIT_SUCCESS;
34 }
```

Hasil eksekusi:



```

D:\dokumen\buat belajar\pemrograman c++\Data C++\BAB 7\Program 7.13\Program 7.13
x : 2
y : 3
x pangkat y = 8
Press any key to continue . . .
```

Gambar 7.13 Hasil eksekusi program Lat 7.13