

PRAKTIKUM 7

7.1 JUDUL

FUNGSI DAN PROSEDUR

7.2 TUJUAN

Pada akhir perkuliahan ini mahasiswa akan dapat:

- Menyederhanakan perintah yang sama dengan subprogram
- Memahami konsep prosedur dan fungsi
- Memahami konsep mengirim variabel pada prosedur dan fungsi
- Memahami konsep menerima hasil balik dari fungsi

7.3 DASAR TEORI

Tujuan procedure dan fungsi adalah untuk memisahkan sekelompok listing program dari fungsi main() dalam sebuah nama. Sehingga sekelompok listing tadi dapat dipakai berulang-ulang dengan memanggil nama prosedur atau nama fungsinya saja, tanpa menuliskan seluruh listingnya.

Tujuan prosedur dan fungsi adalah

- Program menjadi terstruktur, sehingga mudah difahami dan mudah dikembangkan
- Dapat mengurangi pengulangan kode (duplikasi kode).

Sebenarnya, sejak memulai belajar bahasa C, anda telah menggunakan prosedur dan fungsi. Contohnya adalah perintah `getch()`, `printf()`, `scanf()`, bahkan `main()` semua itu adalah contoh prosedur dan fungsi. Kita tidak perlu tahu detail bagaimana (misalnya) `getch()` bekerja untuk menunggu penekanan tombol, yang cukup kita perlu tahu adalah cara memakai `getch()`. Nah, jika anda berencana membuat `getch()` sendiri, artinya anda akan membuat fungsi `getch()` sendiri.

Tidak ada keterangan baku mengenai perbedaan fungsi dan prosedur, namun menurut penulis, perbedaan utama keduanya adalah:

- Prosedur : Tidak mengembalikan nilai. Hanya melakukan perintah2x tertentu.
- Fungsi : Menembalikan nilai. Cirinya ada perintah `return()`.

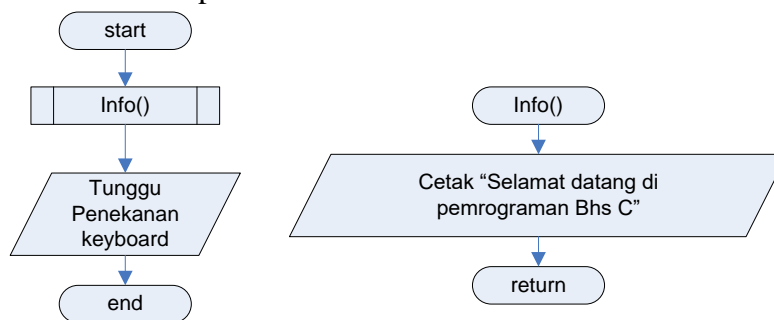
7.4 LANGKAH PERCOBAAN

1. Membuat Prosedur sederhana

a. Tuliskan listing Program berikut:

```
1 #include<stdio.h>
2 #include<conio.h>
3 void info()
4 {
5     clrscr();
6     puts("=====");
7     puts("Selamat Datang di Program Bhs C");
8     puts("=====");
9 }
10
11 main()
12 {
13     info();
14     getch();
15 }
```

b. Flowchart dari prosedur tersebut adalah sbb:



Perhatikan bentuk pemanggilan prosedur, dan prosedur itu sendiri.

Pada prosedur tidak memakai start, tapi **nama** prosedur/ fungsi itu sendiri.

c. Program diatas adalah prosedur yang sangat sederhana. **void** berarti tidak mengembalikan variabel.

- d. Jalankan Program, catat apa yang tampak di layar? (Jika tidak tampak, tekan **ALT+F5**, atau pilih menu windows → user screen.
- e. Prototipe prosedur. Ketikkan listing` berikut ini. (hanya membalik posisi main menjadi di atas)

```

1  #include<stdio.h>
2  #include<conio.h>
3  main()
4  = {
5      info();
6      getch();
7  }
8
9  void info()
10 = {
11     clrscr();
12     puts("=====");
13     puts("Selamat Datang di Program Bhs C");
14     puts("=====");
15 }
```

- f. Jalankan Program, catat apa yang tampak di layar? Mestinya mucul error...
- g. Kenapa bisa demikian?
- h. Coba tambahkan perintah menjadi...

```

.....
#include<conio.h>

void info(); ←
main()
{
.....
```

- i. Jalankan Program, catat apa yang tampak di layar?
 Dalam Hal ini, `void info();` disebut sebagai prototipe. Prototipe diperlukan jika prosedur atau fungsi yang ditulis diletakkan setelah main. Ingat, listing program dijalankan mulai dari atas, Sehingga jika tidak ada prototipe, maka main tidak pernah tahu kalo ada prosedur atau fungsi tersebut yang letaknya dibawah main.

2. Mengirim variabel ke Prosedur.

- a. Berikut ini akan ditunjukkan cara mengirimkan variabel ke prosedur (*diberi nama prosedur karena tidak mengembalikan nilai*). Prosedur yang dibuat akan

memilih 1 diantara 2 variabel yang dikirim, dan menampilkan nilai yang lebih besar.

Ketikkan listing berikut ini

```
1 #include<stdio.h>
2 #include<conio.h>
3
4 void besar(int x, int y);
5
6 main()
7 {
8     int a,b;
9     clrscr();
10    printf("Masukkan dua bilangan dipisah spasi");
11    scanf("%i %i",&a,&b);
12    besar(a,b);
13    getch();
14 }
15
16
17 void besar(int x, int y)
18 {
19     printf("Nilai x = %i\r\n",x);
20     printf("Nilai y = %i\r\n",y);
21     if (x>y)
22         printf("Yang besar = %i",x);
23     else
24         printf("Yang besar = %i",y);
25 }
```

- b. Jalankan Program, masukkan angka 15 3, catat apa yang tampak di layar?
- c. Coba ganti 15,20) . Jalankan Program, catat apa yang tampak di layar?
- d. Coba sendiri dengan mengganti-ganti nilai didalam kurung beberapa kali, jalankan dan catat hasilnya.
- e. Perhatikan baik-baik, apa hubungan antara posisi variabel sebelum dan sesudah koma dengan x dan y.

3. Mengirim variabel ke Prosedur, dan mengembalikannya (FUNGSI).

- a. Berikutnya, bukan lagi bernama prosedur, tetapi diberi nama fungsi karena mengembalikan nilai hasil proses ke `main()` dengan menggunakan perintah `return()`, maka variabel yang ada didalam kurung akan dikirim balik ke

pemanggil. Dan bisa dipastikan, fungsi TIDAK AKAN menggunakan deklarasi void, MELAINKAN bisa int atau float sesuai nilai yang dikembalikan.

- b. Contoh program sederhana tentang return dari sebuah fungsi. Ketikkan listing berikut :

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  int coba_fungsi();
5
6  main()
7  = {
8      clrscr();
9      printf("Nilai yang dikembalikan fungsi = %i\n",coba_fungsi());
10     getch();
11 }
12
13 int coba_fungsi()
14 = {
15     return(10);
16 }
```

- c. Jalankan program, lihat dan catat hasilnya.
- d. Sama seperti di atas, nilai fungsi juga dapat dimasukkan ke dalam variabel terlebih dahulu sebelum di cetak di fungsi utama / di fungsi main()

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  int coba_fungsi();
5
6  main()
7  = {
8      clrscr();
9      int nilai;
10     nilai=coba_fungsi();
11     printf("Nilai yang dikembalikan fungsi = %i\n",nilai);
12     getch();
13 }
14
15 int coba_fungsi()
16 = {
17     return(10);
18 }
```

- e. Jalankan program, lihat dan catat hasilnya.

- f. Berikutnya akan dicoba mengirim variabel dan menampilkan kembali variabel tersebut dalam bentuk kuadrat (perkalian bilangan tersebut dengan dirinya).

```
1 #include<stdio.h>
2 #include<conio.h>
3
4 int kuadrat(int i);
5
6 main()
7 {
8     clrscr();
9     int nilai,hasil;
10    nilai=4;
11    hasil=kuadrat(nilai);
12    printf("Nilai kuadrati dari %i adalah %i",nilai,hasil);
13    getch();
14 }
15
16 int kuadrat(int i)
17 {
18     return(i*i);
19 }
```

- g. Jalankan program, lihat dan catat hasilnya.
- h. Apakah hasilnya sudah sesuai?
- i. Coba ganti nilai dengan angka yang anda sukai. Apakah hasilnya sesuai?...
- j. Berikut ini adalah sama dengan praktek 9.4.2, bedanya adalah di listing berikut, bukan lagi prosedur. Tapi fungsi karena mengembalikan nilai ke fungsi utama.

- k. Cara kerjanya sama dengan sebelumnya, yaitu mengembalikan nilai yang lebih besar.

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  int besar(int x, int y);
5
6  main()
7  = {
8      Int a,b,hasil;
9      clrscr();
10     printf("Masukkan dua bilangan dipisah spasi");
11     scanf("%i %i",&a,&b);
12     hasil=besar(a,b);
13     printf("Nilai yang lebih besar = %i",hasil);
14     getch();
15 }
16
17 int besar(int x, int y)
18 = {
19     printf("Nilai x = %i\r\n",x);
20     printf("Nilai y = %i\r\n",y);
21     if (x>y)
22         return(x);
23     else
24         return(y);
25 }
```

- l. Jalankan Program, catat apa yang tampak di layar?
- m. Apa perbedaan `besar()` pada nomor 3 dengan program sebelumnya 9.4.2.
- n. Apakah yang harus kita tuliskan didalam tanda kurung buka tutup pada `return()`.

4. Melewatkan parameter pada fungsi

- a. Perlu diperhatikan, bahwa metode pengiriman variabel pada contoh2x di atas menggunakan pemanggilan dengan nilai. Tidak ada hubungan antara nama variabel pemanggil dengan fungsi yang dipanggil. Nama boleh sama, tapi tidak akan berpengaruh. Perhatikan listing berikut. Ketiklah listing berikut.

```

1  #include<stdio.h>
2  #include<conio.h>
3  void tukar(int x, int y)
4  = {
5      int z;
6      z=x;
7      x=y;
8      y=z;
9      printf("Nilai pada fungsi tukar\r\n");
10     printf("a = %i, b = %i \r\n",x,y);
11 }
12 main()
13 = {
14     clrscr();
15     int a,b;
16     a=88;
17     b=77;
18     printf("Nilai sebelum pemanggilan fungsi\r\n");
19     printf("a = %i, b = %i \r\n",a,b);
20     tukar(a,b);
21     printf("Nilai setelah pemanggilan fungsi\r\n");
22     printf("a = %i, b = %i \r\n",a,b);
23     getch();
24 }

```

- b. Jalankan Program, catat apa yang tampak di layar?
- c. Untuk apa sebenarnya program diatas?
- d. Apakah nilai a dan b dapat ditukar?
- e. Analisalah tentang nama variabel, scope (jangkauan) variabel dan nama variabel itu sendiri (bandingkan isi variabel d main dan di prosedur).
- f. Cobalah untuk mengganti nama variabel pada tukar, yaitu x menjadi a, dan y menjadi b. Sehingga menjadi spt ini

```

void tukar(int a, int b)
{
    int z;
    z=a;
    a=b;
    b=z;
    printf("Nilai pada fungsi tukar\r\n");
    printf("a = %i, b = %i \r\n",a,b);
}

```

- g. Jalankan Program, catat apa yang tampak di layar?
- h. Untuk apa sebenarnya program diatas?

- i. Apakah nilai a dan b dapat ditukar?
- j. Analisalah tentang nama variabel, scope (jangkauan) variabel dan nama variabel itu sendiri.

7.5 DATA

<Isikan data langkah percobaan di lembar kerja praktikum>

7.6 Analisa Data

<Buatlah analisa berdasar data yang diperoleh>

7.7 Kesimpulan

<Ambil kesimpulan dari hasil praktikum, data dan analisa data anda>

7.8 Soal

Untuk lebih memahami konsep prosedur dan fungsi, desainlah sebuah menu yang memiliki 3 (tiga) item. Yaitu

1. Menghitung Luas dan Keliling lingkaran?
2. Menghitung luas dan keliling persegi panjang?
3. keluar

Buatlah menu biasa dengan menggunakan switch case. Dan Panggilah perhitungan tersebut dengan menggunakan prosedur/ fungsi yang anda desain.