Praktikum 5 (1/5)

FUNGSI

A. TUJUAN

1. Memecah program dalam fungsi fungsi yang sederhana.
2. Menjelaskan tentang pemrograman terstruktur.

B. 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 berupa fungsi-fungsi, dalam hal ini program dari bahasa C dibentuk dari kumpulan fungsi pustaka (standar) dan fungsi yang dibuat sendiri oleh pemrogram. Fungsi banyak digunakan pada program C dengan tujuan :

a. Program menjadi terstruktur, sehingga mudah dipahami dan mudah dikembangkan. Dengan memisahkan langkah-langkah detail ke satu atau lebih fungsi-fungsi, maka fungsi utama (*main()*) menjadi lebih pendek, jelas dan mudah dimengerti.
b. dapat mengurangi pengulangan (duplikasi) kode. Langkah-langkah program yang sama dan dipakai berulang-ulang di program 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.

Dasar Fungsi

Fungsi standar C yang mengemban tugas khusus contohnya adalah ;

- *printf()* , yaitu untuk menampilkan informasi atau data ke layar.
- *scanf()* , yaitu untuk membaca kode tombol yang diinputkan.

Pada umumnya fungsi memerlukan nilai masukan atau parameter yang disebut sebagai argumen. Nilai masukan ini akan diolah oleh fungsi. Hasil akhir fungsi berupa sebuah nilai (disebut sebagai *return value* atau nilai keluaran fungsi). Oleh karena itu

fungsi sering digambarkan sebagai "kotak gelap" seperti ditunjukkan pada gambar di bawah ini.

Gambar 5.1 Fungsi sebagai sebuah kotak gelap

Penggambaran sebagai kotak gelap di antaranya menjelaskan bahwa bagian dalam fungsi bersifat pribadi bagi fungsi. Tak ada suatu pernyataan di luar fungsi yang bisa mengakses bagian dalam fungsi, selain melalui parameter (atau variabel eksternal yang akan dibahas belakangan). Misalnya melakukan *goto* dari pernyataan di luar fungsi ke pernyataan dalam fungsi adalah tidak diperkenankan.

Bentuk umum dari definisi sebuah fungsi adalah sebagai berikut ;

```
tipe-keluaran-fungsi  nama-fungsi (deklarasi argumen)
{
     tubuh fungsi
}
```

Keterangan :

- **tipe-keluaran-fungsi**, dapat berupa salah satu tipe data C, misalnya *char* atau *int* . Kalau penentu tipe tidak disebutkan maka dianggap bertipe *int* (secara *default*).
- **tubuh fungsi** berisi deklarasi variabel (kalau ada) dan statemen-statemen yang akan melakukan tugas yang akan diberikan kepada fungsi yang bersangkutan. Tubuh fungsi ini ditulis di dalam tanda kurung kurawal buka dan kurung kurawal tutup.

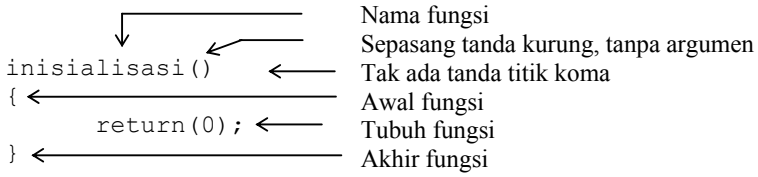
Sebuah fungsi yang sederhana bisa saja tidak mengandung parameter sama sekali dan tentu saja untuk keadaan ini deklarasi parameter juga tidak ada. Contoh :

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

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

Pada fungsi di atas :

- tipe keluaran fungsi tidak disebutkan, berarti keluaran fungsi ber tipe *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.



Gambar 5.2 Penjelasan definisi sebuah fungsi

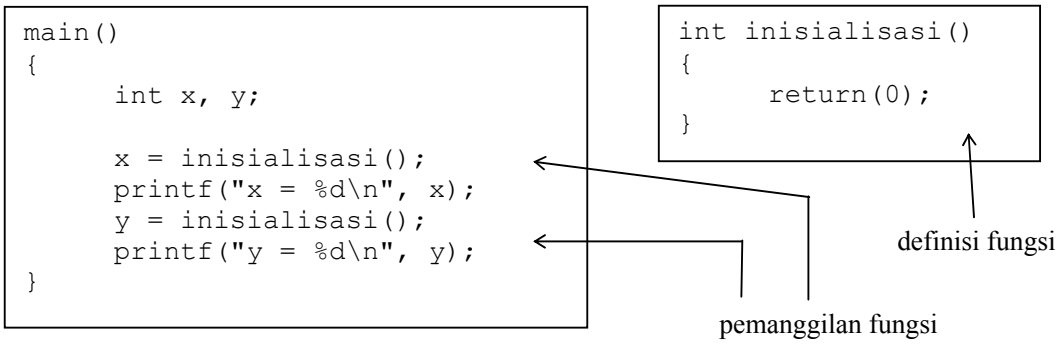
Memberikan Nilai Keluaran Fungsi

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 di layar.

Dalam tubuh fungsi, pernyataan yang digunakan untuk memberikan nilai keluaran fungsi berupa *return*. Sebagai contoh, pada fungsi **inisialisasi()** di atas terdapat pernyataan

```
return(0);
```

merupakan pernyataan untuk memberikan nilai keluaran fungsi berupa nol.



Gambar 5.3 Proses pemanggilan fungsi

Program di atas 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.

Di bawah ini diberikan contoh sebuah fungsi yang mengandung dua buah pernyataan *return*. Fungsi digunakan untuk memperoleh nilai minimum di antara 2 buah nilai yang menjadi parameternya.

```
int minimum(int x, int y)
{
     if (x < y)
          return(x);
     else
          return(y);
}
```

Pada fungsi di atas terdapat dua buah parameter berupa **x** dan **y**. Oleh karena itu fungsi juga mengandung bagian untuk mendeklarasikan parameter, yang menyatakan **x** dan **y** bertipe *int*. Adapun penentuan nilai keluaran fungsi dilakukan pada tubuh fungsi, berupa pernyataan

```
if (x < y)
     return(x);
else
     return(y);
```

yang menyatakan :

- jika x < y maka nilai keluaran fungsi adalah sebesar nilai x.
- untuk keadaan lainnya (x >= y) maka keluaran fungsi adalah sebesar y.

Fungsi Dengan Keluaran Bukan Integer

Untuk fungsi yang mempunyai keluaran bertipe bukan integer, maka fungsi haruslah didefiniskan dengan diawali tipe keluaran fungsinya (ditulis di depan nama fungsi). Sebagai contoh untuk menghasilkan nilai terkecil di antara dua buah nilai real, maka definisinya berupa :

```
float minimum(float x, float y)
{
     if (x < y)
          return(x);
     else
          return(y);
}
```

Perhatikan, di depan nama **minimum** diberikan tipe keluaran fungsi berupa *float*. Seluruh parameter sendiri juga didefinisikan dengan tipe *float*.

Khusus untuk fungsi yang dirancang tanpa memberikan nilai keluaran (melainkan hanya menjalankan suatu tugas khusus) biasa didefinisikan dengan diawali kata kunci *void* (di depan nama fungsi).

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 menunjukan kesalahan bila jumlah parameter dalam definisi dan saat pemanggilan berbeda.

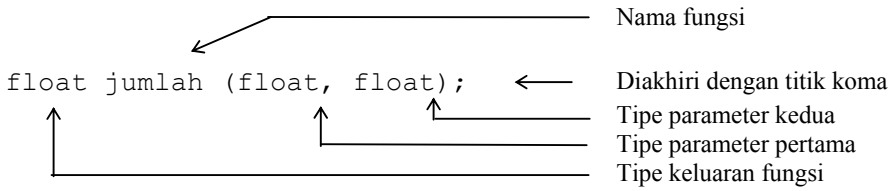
Contoh prototipe fungsi;

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

atau

```
float jumlah (float, float);
```

Penjelasannya adalah sbb :



Gambar 5.4 Prototipe fungsi

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

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

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

<u>Catatan :</u>

- 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 *preprocessor directive* **#include**.

C. TUGAS PENDAHULUAN

Buatlah desain flowchart untuk setiap soal dalam percobaan

D. PERCOBAAN

1. a. Buatlah sebuah fungsi yang berfungsi untuk menampilkan sebuah string (di layar) = "`Pilihan Menu`" (misalkan nama fungsinya = **menu**). Fungsi tersebut tidak memiliki nilai kembalian *(return value)* dan juga tidak menerima parameter masukan apapun.
   b. Tulislah prototipe fungsi untuk fungsi tersebut.
   c. Buat function main untuk memanggil function *menu()* secara berulang-ulang, dengan jumlah perulangan yang merupakan input dari user.

2. a. Buatlah sebuah fungsi untuk menghitung jumlah triangular n (misal nama fungsinya = **triangular**). Fungsi tersebut memiliki sebuah parameter berupa bilangan int (n) yang akan dicari triangularnya serta tidak memiliki nilai kembalian *(return value)*
   b. Tulislah prototipe fungsi untuk fungsi tersebut.
   c. Buat function main untuk memanggil function *triangular()* tersebut dengan nilai n yang merupakan input dari user.

3. a. Buatlah sebuah fungsi untuk menghitung nilai bilangan kuadrat (misal nama fungsinya = **kuadrat**). Fungsi tersebut memiliki sebuah parameter bertipe float, yaitu bilangan yang akan dikuadratkan serta memiliki sebuah *return value* bertipe float, yaitu hasil kuadratnya
   b. Tulislah prototipe fungsi untuk fungsi tersebut.
   c. Buat function main untuk memanggil function *kuadrat()* tersebut dengan bilangan x yang akan dicari kuadratnya merupakan input dari user.

E. LAPORAN RESMI

1. Tulis listing program dari semua percobaan yang dilakukan.

2. Kemudian tuliskan outputnya. Terangkan mengapa demikian.