

PRAKTIKUM 16

Fungsi : Penggolongan Variabel & Pemrograman Terstruktur

A. TUJUAN PEMBELAJARAN

1. Mengetahui perbedaan antara variabel lokal, eksternal, statis dan register
2. Memahami komunikasi antar sejumlah fungsi
3. Mengenal Konsep Pemrograman Terstruktur

B. DASAR TEORI

Penggolongan Variabel berdasarkan Kelas Penyimpanan

Suatu variabel, di samping dapat digolongkan berdasarkan jenis/tipe data juga dapat diklasifikasikan berdasarkan kelas penyimpanan (*storage class*). Penggolongan berdasarkan kelas penyimpanan berupa :

- variabel lokal
- variabel eksternal
- variabel statis
- variabel register

Variabel Lokal

Variabel lokal adalah variabel yang dideklarasikan dalam fungsi, dengan sifat :

- secara otomatis diciptakan ketika fungsi dipanggil dan akan sirna (lenyap) ketika eksekusi terhadap fungsi berakhir.
- Hanya dikenal oleh fungsi tempat variabel tersebut dideklarasikan
- Tidak ada inisialisasi secara otomatis (saat variabel diciptakan, nilainya tak menentu).

Dalam banyak literatur, variabel lokal disebut juga dengan variabel otomatis. Variabel yang termasuk dalam golongan ini bisa dideklarasikan dengan menambahkan

Variabel Statis

Variabel statis dapat berupa variabel internal (didefinisikan di dalam fungsi) maupun variabel eksternal. Sifat variabel ini :

- Kalau variabel statis bersifat internal, maka variabel hanya dikenal oleh fungsi tempat variabel dideklarasikan
- Kalau variabel statis bersifat eksternal, maka variabel dapat dipergunakan oleh semua fungsi yang terletak pada file yang sama, tempat variabel statis dideklarasikan
- Berbeda dengan variabel lokal, variabel statis tidak akan hilang sekeluanya dari fungsi (nilai pada variabel akan tetap diingat).
- Inisialisasi akan dilakukan hanya sekali, yaitu saat fungsi dipanggil yang pertama kali.

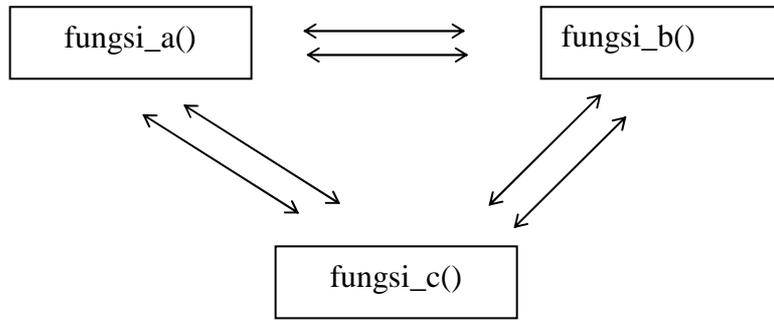
Kalau tak ada inisialisasi oleh pemrogram secara otomatis akan diberi nilai awal nol
Variabel statis diperoleh dengan menambahkan kata kunci *static* di depan tipe data variabel.

Variabel Register

Variabel register adalah variabel yang nilainya disimpan dalam register dan bukan dalam memori RAM. Variabel yang seperti ini hanya bisa diterapkan pada variabel yang lokal atau parameter formal, yang bertipe *char* atau *int*. Variabel register biasa diterapkan pada variabel yang digunakan sebagai pengendali *loop*. Tujuannya untuk mempercepat proses dalam *loop*. Sebab variabel yang dioperasikan pada register memiliki kecepatan yang jauh lebih tinggi daripada variabel yang diletakkan pada RAM.

Menciptakan Sejumlah Fungsi

Pada C, semua fungsi bersifat sederajat. Suatu fungsi tidak dapat didefinisikan di dalam fungsi yang lain. Akan tetapi suatu fungsi diperbolehkan memanggil fungsi yang lain, dan tidak tergantung kepada peletakan definisi fungsi pada program. Komunikasi antara fungsi dalam C ditunjukkan dalam gambar 16.2. Gambar tersebut menjelaskan kalau suatu fungsi katakanlah **fungsi_a()** memanggil **fungsi_b()**, maka bisa saja **fungsi_b()** memanggil **fungsi_a()**. Contoh program yang melibatkan fungsi yang memanggil fungsi yang lain ada pada program **kom_fung.c**, yaitu **fungsi_1()** dipanggil dalam **main()**, sedangkan **fungsi_2()** dipanggil oleh **fungsi_1()**.



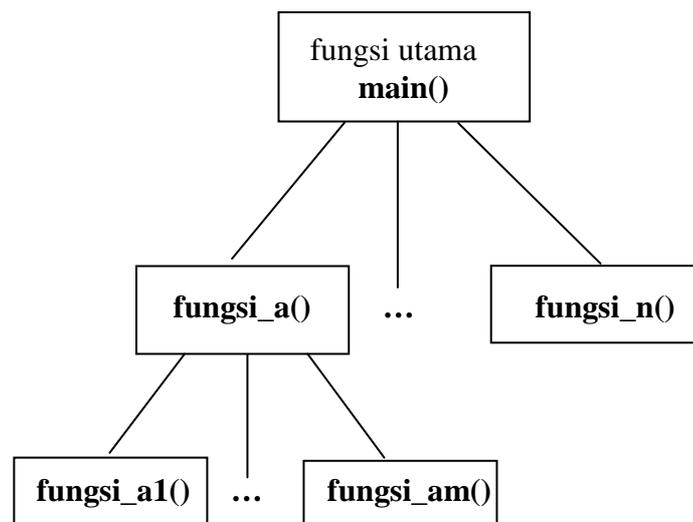
Gambar 16.2 Komunikasi antar fungsi dalam C

Pengenalan Konsep Pemrograman Terstruktur

Fungsi sangat bermanfaat untuk membuat program yang terstruktur. Suatu program yang terstruktur dikembangkan dengan menggunakan “*top-down design*” (rancang atas bawah). Pada C suatu program disusun dari sejumlah fungsi dengan tugas tertentu. Selanjutnya masing masing fungsi dipecah-pecah lagi menjadi fungsi yang lebih kecil. Pembuatan program dengan cara ini akan memudahkan dalam pencarian kesalahan ataupun dalam hal pengembangan dan tentu saja mudah dipahami/ dipelajari.

Dalam bentuk diagram, model suatu program C yang terstruktur adalah seperti yang tertera pada bagan berikut ini. Namun sekali lagi perlu diketahui, bahwa pada C semua fungsi sebenarnya berkedudukan sederajat.

Fungsi **main()** terdiri dari **fungsi_a()** sampai dengan **fungsi_n()**, menegaskan bahwa dalam program fungsi **main()** akan memanggil **fungsi_a()** sampai dengan **fungsi_n()**. Adapun fungsi-fungsi yang dipanggil oleh fungsi **main()** juga bisa memanggil fungsi-fungsi yang lain. Model terstruktur Program C ditunjukkan pada gambar 16.3



Gambar 16.3 Model terstruktur Program C

5. Definisikanlah fungsi `main()`, `masukan()` dan `average()`, sebagai berikut :
- Fungsi `masukan()` menerima satu parameter berupa jumlah data yang akan dimasukkan dan memberikan return value berupa nilai total dari seluruh data yang dimasukkan. Fungsi ini bertugas menerima masukan data sebanyak n kali dan sekaligus menghitung total nilai seluruh data.
 - Fungsi `average()` menerima dua parameter berupa jumlah data yang telah dimasukkan dan nilai total seluruh data. Fungsi ini memberikan return value berupa nilai rata-rata dari seluruh data yang dimasukkan.
 - Pada fungsi `main()` mintalah masukan jumlah data yang akan diinputkan. Selanjutnya lakukan pemanggilan fungsi `masukan()` dan `average()`, kemudian tampilkan nilai rata-rata dari seluruh datanya.

6. Definisikanlah fungsi-fungsi sebagai berikut :

- Fungsi `f_to_i()` untuk mengubah ukuran dari satuan kaki (*feet*) ke inci
- Fungsi `i_to_cm()` untuk mengubah ukuran dari satuan inci ke centimeter
- Fungsi `c_to_m()` untuk mengubah ukuran dari satuan centimeter ke meter

Dalam `main()` mintalah masukan ukuran dalam satuan kaki (*feet*) kemudian lakukan konversi sampai mendapatkan keluaran berupa ukuran dalam meter. Tentukan jumlah dan tipe parameter dan return value yang dibutuhkan

Keterangan : 1 kaki = 12 inchi, 1 inchi = 2.54 cm, 100 cm = 1 meter

D. LAPORAN RESMI

1. Untuk setiap listing program dari percobaan-percobaan di atas, ambil *capture* outputnya.
2. Tuliskan kesimpulan dari percobaan yang telah anda lakukan.