



PENGANTAR PEMROGRAMAN TERSTRUKTUR

Pendahuluan

Pada era tahun 1950 –1960, kecepatan komputer sangat rendah dan disertai juga dengan keterbatasan dari media penyimpan, sehingga tentunya berakibat juga dengan keterbatasan dalam penulisan program-program komputer. Namun di era saat ini, dengan kecepatan komputer yang cukup handal demikian juga ketersediaan dari media penyimpan yang cukup handal dan besar, serta didukung juga dengan perkembangan bahasa pemrograman yang ada sehingga kita dapat dengan mudah membuat suatu program. Permasalahan yang timbul dalam pembuatan program tersebut adalah bagaimana kita dapat memahaminya, sehingga apabila terdapat perubahan yang akan dilakukan kita dapat memperbaikinya secara mudah. Hal ini tentu saja harus kita perhatikan terutama apabila kita melihat dari biaya yang harus dikeluarkan dalam pembuatan program tersebut.

Istilah Pemrograman Terstruktur (Structured Programming) mengacu dari suatu kumpulan tehnik yang dikemukakan oleh Edsger Dijkstra. Dengan tehnik ini akan meningkatkan produktifitas programmer, dengan mengurangi waktu yang dibutuhkan dalam penulisan (write), pengujian (test), penelusuran kesalahan (debug) dan pemeliharaan(maintain) suatu program. Pada pembahasan berikut ini kita akan melihat bagaimana tehnik ini yang pendekatan yang dilakukan secara modular, dapat membantu kita dalam membangun suatu program.

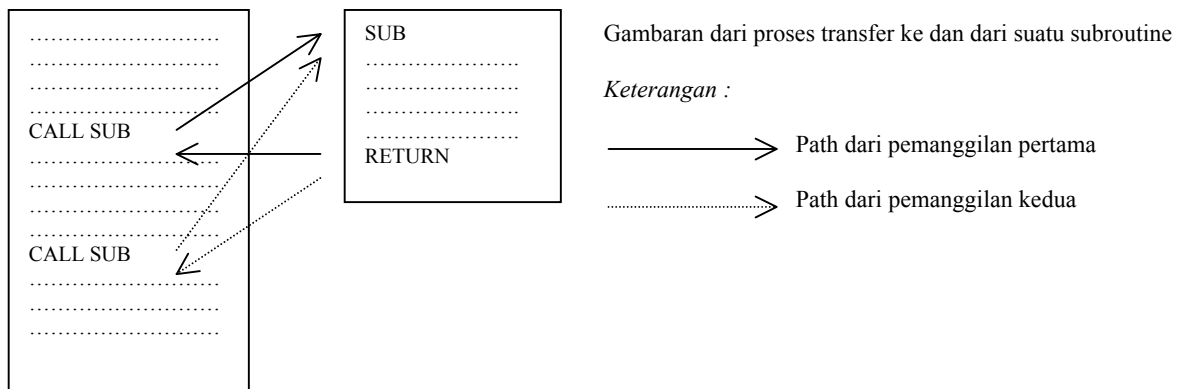
Pemrograman Secara Modular



Dalam pemrograman secara modular, suatu program akan dipilah kedalam sejumlah modul, dimana setiap modul menjalankan fungsinya sendiri. Tentunya fungsi yang dijalankan oleh setiap modul sangat terbatas sesuai dengan ruang lingkup yang akan dikerjakan. Dengan adanya sejumlah modul program ini tentu saja kesalahan yang timbul dapat dikurangi. Setiap program tentu akan memiliki program utamanya, yang kemudian akan memanggil sejumlah modul-modul yang ada.

Implementasi dari pendekatan secara modular

Pemrograman secara modular ini dapat diimplementasikan dengan penggunaan *subroutine*, suatu kelompok instruksi yang menjalankan suatu pengolahan yang sifatnya terbatas seperti pencetakan, pembacaan untuk proses input atau untuk proses penghitungan.





Subroutine dapat dikelompokkan menjadi *internal subroutine* dan *external subroutine*, berikut ini penjelasannya:

Internal Subroutines

Adalah bagian dari suatu program yang digunakan. Dideklarasikan cukup sekali saja, untuk sejumlah proses yang sama akan dilakukan oleh program tersebut. Program akan memanggil subroutines tersebut jika diperlukan dan apabila telah selesai, kontrol selanjutnya dikembalikan ke instruksi berikutnya.

Instruksi yang mengendalikan kontrol transfer ke suatu subroutine umumnya dikenal sebagai *call* dan *return*.

External Subroutines

Diletakkan secara terpisah dari program yang menggunakan subroutine tersebut. Subroutine ini dideklarasikan supaya bisa dipakai oleh program yang lain. Untuk menggunakannya tentu seorang programmer harus mengetahui dimana ?, apa namanya ?, bagaimana pengiriman datanya ?, bagaimana jawaban yang akan diperoleh ?. Subroutine ini biasanya digunakan untuk pemrosesan yang kompleks, yang dibutuhkan oleh banyak user.

Masalah Yang dihadapi dalam Pendekatan Modular



Masalah yang timbul misalnya tidak mengetahui modul mana yang harus digunakan, data apa yang harus dikirimkan dsb-nya. Hal ini dapat dimaklumi karena tugas seorang programmer tentu akan berbeda dengan programmer yang lainnya. Oleh karena itu dalam tahap perencanaan fungsi-fungsi dari suatu modul harus dapat dimengerti secara jelas, dan tentu saja harus mengikuti suatu standar yang telah ditentukan.

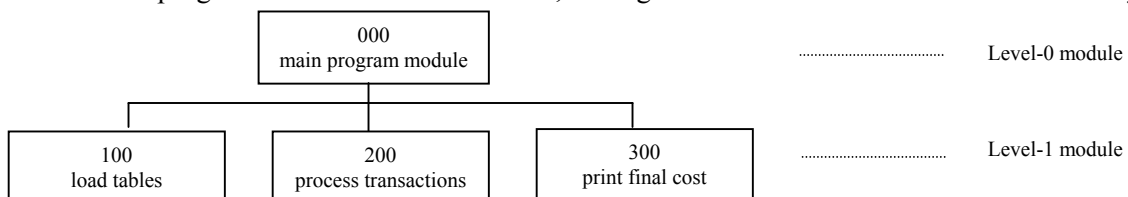
TOP-DOWN PROGRAMMING

Pendekatan ini sangat berguna sekali dalam perencanaan suatu program bersifat modul. Dalam pendekatan ini pertama-tama kita mendefinisikan modul untuk program utama, yang merupakan program yang pertama kali dieksekusi, memanggil modul yang lain dan kemudian menghentikan eksekusi program. Jika fungsi yang akan dikerjakan terlalu kompleks, maka modul ini harus dipecah kedalam sejumlah modul-modul yang ada dibawahnya.

Structure Charts

Digunakan sebagai alat bantu perencanaan dalam top-down programming. Sering juga disebut sebagai *hierarchy/hierarchical/chart/visual table of content (VTOC)*. Tidak ada standar untuk structure chart dan digunakan untuk menggambarkan seluruh komponen yang ada.

Bujur sangkar menggambarkan modul dan diidentifikasi dengan sebuah angka, dimana angka nol untuk menandakan program utama (main program). Modul ini menggambarkan keseluruhan program dan ditandai oleh level-0, sedangkan dibawah adalah level-1 dan seterusnya.

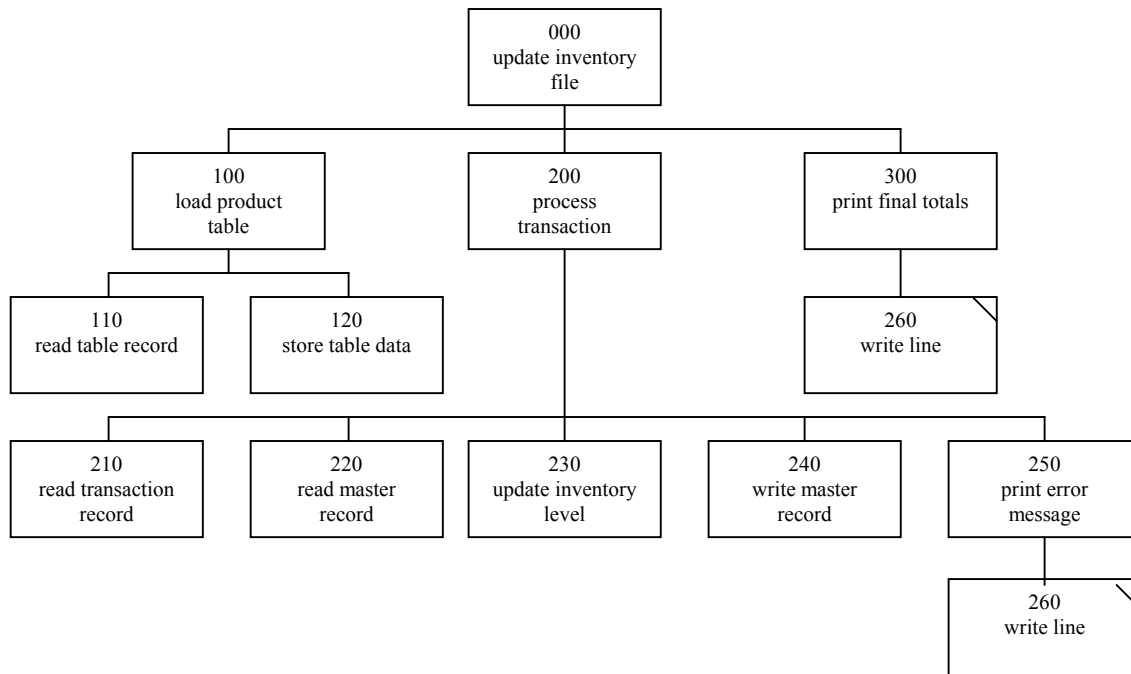




Meng-identifikasi Modul

Nama modul adalah deskripsi singkat dari modul tersebut. Identifikasi dari sebuah modul juga memuat angkanya. Sebagai contoh pada gambar modul “200 process transaction” mempunyai 5 buah fungsi subordinat yaitu :

- Reading transaction record
- Reading master record
- Updating inventory level
- Writing master record
- Printing error message



Contoh gambar structure chart untuk program pengupdate-an file inventory

Modul 210 – 250 juga termasuk level-2 seperti juga modul 110 dan 120. Modul 260 adalah modul perkecualian. Sebuah modul dapat memiliki sebuah subordinat modul, jika modul itu juga menjadi subordinat paling sedikit satu buah modul.

Me-review Structure Chart



Sesudah sebuah structure chart dipersiapkan, kemudian direview kembali untuk memastikan kelengkapan dan strukturnya. Review ini dilakukan mulai dari level teratas hingga level terbawah. Peranan user (pemakai program tersebut) harus disertakan untuk dapat melihat segala sesuatu yang dibutuhkan.

Structure Chart dan Flowchart

Dalam structure chart digambarkan fungsi yang akan dilakukan dan relasi antar modul-modul, sedangkan informasi yang diperlukan untuk peng-coding-an sangat sedikit. Didalamnya tidak digambarkan langkah-langkah setiap pemrosesan maupun kondisi-kondisinya. Sehingga flowchart umumnya dipersiapkan juga. Namun karena logika dari modul-modul itu sangat sederhana, sehingga memungkinkan untuk melakukan peng-coding-an tanpa harus menggunakan flowchart. Demikian juga untuk user lebih mudah untuk mengertinya.



Programming Structure

Salah satu sasaran dalam men-desain program terstruktur adalah mengurangi kebingungan yang dihasilkan dari penggunaan percabangan atau perintah go-to.

Dalam pemrograman terstruktur tidak digunakan lagi perintah go-to. Ada tiga bentuk yang digunakan yaitu : *Sequence Structure*, *Loop Structure* dan *Selection Structure*.

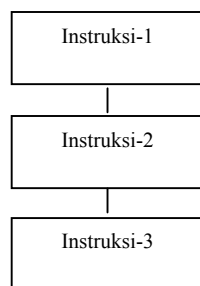
Sequence Structure

Dalam sequence structure, instruksi dieksekusi berdasarkan urutannya. Dimulai dari bagian atas dan diakhiri di bagian bawahnya.

Bujur sangkar dapat menggambarkan operasi :

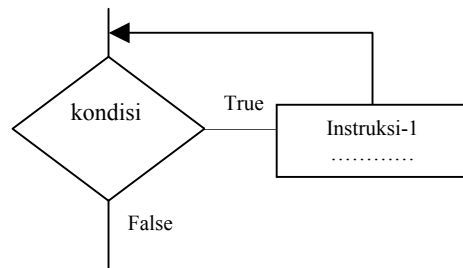
- Input dan Output
- Operasi aritmatika
- Operasi pemindahan data dalam memori komputer

Dalam sequence structure tidak diperkenankan penggunaan kotak keputusan.



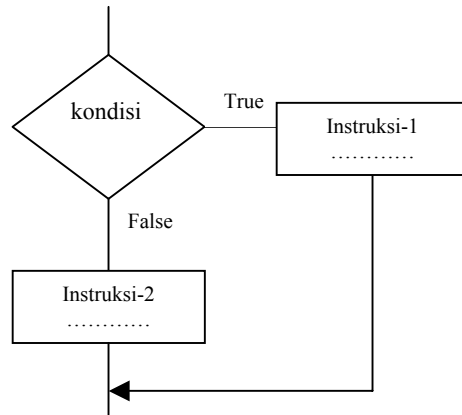
Loop Structure

Loop (iteration) structure menggambarkan perulangan dari satu atau lebih instruksi.



Selection Structure

Dalam struktur ini terdapat sejumlah perintah yang dikerjakan tergantung dari kondisi yang dipenuhinya. Seperti juga dengan sequence dan loop structure, terdapat single entry point dan single exit point.



Referensi diambil dari :

La Budde, Keith, Structured Programming Concept, MCGraw-Hill Book Company.