

## VARIABEL DAN MEMORI

Variabel merupakan komponen penting pada pemrograman, Variabel digunakan dalam program untuk menyimpan suatu nilai, dan nilai yang ada padanya dapat dirubah selama eksekusi program berlangsung.

Jika suatu variable diisi dengan nilai di luar jangkauannya maka nilai yang akan disimpan akan diubah sesuai dengan jangkauannya. Misalnya, bila suatu variable bertipe integer diberi nilai 75000, yang tersimpan pada variable tersebut berupa 9494. Sebab nilai positif terbesar pada tipe integer yaitu 32767. Hal ini bekerja sebagaimana speedometer pada kendaraan bermotor. Pada speedometer, apabila nilai maksimumnya terlampaui akan dimulai dari nilai terendahnya, yakni nol.

Perlu diketahui, pemrograman aritmatika yang menggunakan tipe seperti integer akan lebih cepat dibandingkan kalau menggunakan tipe long integer itulah sebabnya sedapat mungkin untuk menggunakan variable dengan memori berukuran kecil.

Pendefinisian variabel tergantung pada bahasa pemrograman yang dipakai ada yang pendefinisian variabel dapat diletakan dimana saja (contohnya : Basic, C++, Dbase, dll ) dan ada pula bahasa pemrograman yang sudah ditentukan pendefinisian variabelnya (contohnya : Pascal, Cobol, dll).

### Lingkup Variabel

Pemahaman terhadap lingkup variabel di dalam penulisan fungsi sangatlah penting, agar tidak salah dalam menggunakan suatu variabel. Lingkup variabel menentukan keberadaan suatu variabel tertentu didalam fungsi. Ada variabel yang hanya dikenal di suatu fungsi dan tidak dikenal pada fungsi lain. Namun ada juga variabel yang dapat diakses oleh semua fungsi.

Jenis variabel berdasarkan kelas penyimpanannya, yang berkaitan dengan lingkup variabel, yaitu:

- Variabel otomatis
- Variabel eksternal
- Variabel Statis
- Variabel otomatis

Variabel yang didefinisikan di dalam suatu fungsi berlaku sebagai variabel lokal bagi fungsi. Artinya, variabel tersebut hanya dikenal di dalam fungsi tempat variabel didefinisikan.

Suatu variabel otomatis mempunyai sifat :

- ✓ Variabel hanya akan diciptakan pada saat fungsi dipanggil.
- ✓ Pada saat fungsi berakhir (selesai dieksekusi), variabel otomatis menjadi sirna.
- ✓ Tidak ada inisialisasi secara otomatis (pada saat variabel diciptakan). Inisialisasi oleh pemrograman akan dikerjakan setiap kali fungsi dipanggil.
- ✓ Hanya dapat diakses di dalam fungsi yang mendefinisikannya.

Selang waktu antara penciptaan variabel hingga penyirnaannya sering disebut sebagai lifetime atau durasi. Durasi dari variabel otomatis hanya pada saat fungsi yang mendefinisikannya dieksekusi.

- Variabel eksternal

Variabel eksternal merupakan kebalikan dari variabel otomatis. Variabel eksternal adalah variabel yang didefinisikan diluar fungsi manapun. Variabel ini dikenal juga sebagai variabel global, sebab variabel ini dikenal disemua fungsi. Anda dapat mendeklarasikan bukan mendefinisikannya, karena tidak ada pengalokasian memori.

Sehingga sifat dari variabel eksternal kebalikan dari variabel otomatis.

Penggunaan variabel eksternal diusahakan sesedikit mungkin atau sedapat mungkin tidak usah digunakan. Tidak lain adalah karena variabel ini mudah sekali berubah oleh pernyataan penugasaan yang letaknya bisa dimana saja. Ini bisa menimbulkan efek samping yang sulit untuk melacaknya, terutama untuk program yang besar.

Variabel eksternal mempunyai durasi selama program dieksekusi. Dengan kata lain, memori yang digunakan untuk variabel ini tetap dipertahankan selama program belum berakhir.

- Variabel statis

Baik variabel eksternal maupun otomatis dapat berkedudukan sebagai variabel statis. Suatu variabel statis mempunyai sifat :

Jika variabel local berdiri sebagai variabel statis, maka :

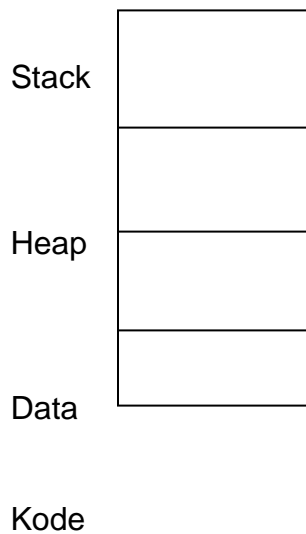
- Variabel tetap hanya dapat diakses pada fungsi yang mendefinisikannya
- Variabel tidak hilang saat dieksekusi fungsi berakhir nilainya akan tetap dipertahankan, sehingga akan dikenali pada pemanggilan fungsi untuk tahap berikutnya.
- Inisialisasi oleh pemrograman akan dilakukan sekali saja selama program dijalankan, jika tidak ada inisialisasi secara eksplisit, variabel diisi dengan nol.

Jika Variabel eksternal dijadikan sebagai variabel statis, variabel ini dapat diakses oleh semua file yang didefinisikan pada file yang sama dengan variabel eksternal tersebut (hal ini bermanfaat pada pemrograman file berganda atau kode program ditaruh pada beberapa file).

## MEMORI DINAMIS

## Konsep pengalokasian memori

Tataletak memori pada computer setelah suatu program dimuat ke memori computer. Seperti gambar berikut :



Pada saat program yang dibuat dijalankan, terdapat sejumlah memori yang tidak terpakai. Memori ini dikenal sebagai Heap. Memori inilah yang dapat dipakai untuk menciptakan variabel dinamis. Ukuran memori ini sangatlah bervariasi, bergantung pada model memori yang digunakan compiler dan juga system operasi.