

Pengulangan (Loop) merupakan sebuah konsep yang penting dalam pemrograman. Dengan struktur pengulangan, program dapat berjalan beberapa kali sesuai inisialisasi, jumlah iterasi dan kondisi berhenti yang ditentukan. Hal ini dapat menyederhanakan program yang kompleks menjadi lebih sederhana. Dalam C disediakan berbagai perintah Loop, dimana setiap perintah loop memiliki keunikan tersendiri. Di dalam bab ini akan dijelaskan tentang struktur pengulangan dalam algoritma serta implementasi struktur pengulangan dengan perintah loop yang ada di dalam C.

Program yang efisien adalah program yang memungkinkan pengguna bekerja sesedikit mungkin dan komputer bekerja sebanyak mungkin. Salah satu cara melakukan hal tersebut adalah dengan menggunakan kembali sekumpulan baris program yang terdapat pada bagian lain dari program tersebut atau baris program yg terdapat di dalam program lain.

Pengulangan merupakan sebuah konsep pemrograman yang penting karena konsep ini memungkinkan pengguna menggunakan sekumpulan baris program berulang kali dengan tiga komponen yang mengendalikannya, yaitu: Inisialisasi; menentukan kondisi awal dilakukannya pengulangan. Jumlah iterasi; menunjukkan berapa kali pengulangan akan dilakukan. Kondisi berhenti; menentukan kondisi yang dapat mengakhiri pengulangan.

Ketika mengimplementasikan dalam program, ketiga komponen ini tidak selalu dapat didefinisikan dalam struktur pengulangan. Mungkin saja salah satu komponen tersebut tidak didefinisikan. Pengulangan tetap dapat berjalan, asal komponen yang tidak didefinisikan tersebut dapat diketahui secara tersirat berdasarkan komponen lain yang didefinisikan. Hal lain yang perlu diperhatikan adalah bahwa pengulangan harus berhenti. Jika pengulangan tidak pernah berhenti, maka logika program salah. Pengulangan akan berhenti jika jumlah iterasi yang diminta sudah tercapai atau kondisi berhenti bernilai benar. Maka, dalam setiap pengulangan, pemrogram perlu menentukan jumlah iterasi atau kondisi berhenti dan langkah pencapaian menuju kondisi berhenti tersebut.

Pada bab ini akan dijelaskan 3 struktur pengulangan dan implementasinya di dalam C, yaitu struktur pengulangan While, For, dan Do While.

Pengulangan dengan menggunakan WHILE merupakan sebuah pengulangan yang dikendalikan oleh suatu kondisi tertentu, dimana kondisi tersebut yang akan menentukan apakah pengulangan itu akan terus dilaksanakan atau dihentikan. Kondisi tersebut akan dicek disetiap awal iterasi, apakah sebuah kondisi terpenuhi atau tidak. Jika kondisi terpenuhi (bernilai benar), maka iterasi akan dilanjutkan. Jika kondisi tidak terpenuhi, maka iterasi dihentikan.

Perulangan dengan WHILE dapat digunakan pada struktur perulangan yang diketahui jumlah iterasinya dan juga pada struktur perulangan yang tidak diketahui jumlah iterasinya, tetapi harus selalu terdapat kondisi berhenti.

Pencacah adalah variabel pengendali iterasi yang harus diinisialisasi, dicek dalam kondisi, dan terus berubah nilainya setiap iterasi dilakukan. Pencacah inilah yang akan membuat sebuah kondisi berhenti tercapai. Pada struktur pengulangan dengan sintaks WHILE, nilai pencacah akan diubah di akhir aksi pengulangan.

Telah dikatakan di awal bahwa perulangan dengan WHILE dapat digunakan untuk struktur pengulangan yang belum diketahui jumlah iterasinya, tetapi tetap mempunyai kondisi berhenti. Bagian pernyataan yang mengikuti while akan dieksekusi selama ungkapan pada while bernilai benar (tidak sama dengan nol). Pengujian terhadap ungkapan while dilakukan sebelum bagian pernyataan.

Sintaks DO... WHILE... melakukan pengulangan serupa dengan sintaks WHILE. Penggunaan sintaks ini juga tidak harus menyebutkan jumlah pengulangan yang harus dilakukan, karena dapat digunakan untuk perulangan dengan jumlah iterasinya yang belum diketahui, tetapi harus mempunyai kondisi berhenti.

Bedanya, jika pada sintaks WHILE kondisi dievaluasi / diuji sebelum aksi pengulangan dilakukan, sedangkan pada sintaks DO...WHILE pengujian kondisi dilakukan setelah aksi pengulangan dilakukan.

Pada struktur pengulangan dengan sintaks DO... WHILE..., aksi akan terus dilakukan hingga kondisi yang dicek di akhir pengulangan, bernilai benar. Dengan sintaks ini, pengulangan pasti dilakukan minimal satu kali, yakni pada iterasi pertama sebelum pengecekan kondisi. WHILE dengan DO WHILE seringkali memberikan hasil yang sama, tetapi ada kalanya hasilnya akan berbeda, sehingga harus berhati-hati dalam penggunaan kondisi antara WHILE dengan DO WHILE. Dengan kata lain Bagian

pernyataan1 hingga pernyataanN dijalankan secara berulang sampai ungkapan bernilai salah (sama dengan nol). Namun berbeda dengan while, pengujian ungkapan dilakukan dibelakang (setelah bagian pernyataan).

Pada algoritma ini juga ditunjukkan bahwa iterasi pada pengulangan dengan sintaks DO... WHILE... dapat dilakukan dengan pencacah mundur. Pengguna terus diminta memasukkan sebuah bilangan yang kemudian dijumlahkan dengan bilanganbilangan sebelumnya (pada iterasi pertama, bilangan dijumlahkan dengan nol) hingga pencacah bernilai 0. Program akan memberikan output berupa hasil perhitungan rerata dari ketiga bilangan yang diinputkan. Tidak semua implementasi kondisi pada DO...WHILE sama dengan implementasi pada WHILE. Sama seperti pada penggunaan sintaks WHILE, sintaks DO...WHILE dapat digunakan untuk struktur pengulangan yang belum diketahui jumlah iterasinya, tetapi tetap mempunyai kondisi berhenti. Karena pada struktur DO...WHILE tidak terdapat pengecekan kondisi di awal, maka isi dari DO...WHILE akan langsung dijalankan, pengecekan akan dilakukan setelah user memasukkan pilihan. Bagian pernyataan yang mengikuti while akan dieksekusi selama ungkapan pada while bernilai salah (tidak sama dengan nol). Pengujian terhadap ungkapan while dilakukan sebelum bagian pernyataan.

Sintaks pengulangan FOR merupakan sintaks yang relative paling mudah digunakan. Sintaks ini serupa dengan sintaks WHILE... DO... dalam hal pengecekan kondisi dilakukan di awal. Dalam menggunakan struktur pengulangan dengan sintaks FOR, pemrogram harus mendefinisikan nilai awal dan nilai akhir pencacah yang menunjukkan jumlah iterasi. Setiap kali iterasi berlangsung, nilai pencacah akan diubah. Jika pencacah sudah mencapai nilai akhir yang ditentukan, maka pengulangan akan berhenti.

Serupa dengan struktur pengulangan dengan sintaks lain, struktur pengulangan dengan sintaks FOR juga dapat dilakukan dengan pencacah mundur.

Struktur pengulangan memungkinkan program melakukan satu atau lebih aksi beberapa kali. Tiga komponen pengendali aksi adalah inisialisasi, jumlah iterasi, dan kondisi berhenti. Tiga struktur pengulangan yang dapat digunakan adalah struktur pengulangan dengan sintaks WHILE, DO...WHILE, dan FOR. Struktur pengulangan dapat dibuat bersarang dengan sintaks pengulangan yang sama atau berbeda, bahkan dapat digabungkan dengan struktur pemilihan. Untuk keluar dari struktur

pengulangan sebelum kondisi berhenti, kita dapat menggunakan sintaks BREAK. Hal yang terpenting dari struktur pengulangan adalah kondisi berhenti yang akan memberikan kondisi apakah pengulangan dilakukan atau tidak.