PERTEMUAN 9-11
STATEMENT PENGULANGAN

Algoritma dan Pemrograman I

Teknik Informatika

UNIKOM



Secara umum struktur pengulangan dibagi dua bagian :



- Kondisi pengulangan
  Yaitu expresi *Boolean* yang harus dipenuhi untuk melaksanakan pengulangan. Kondisi ini dinyatakan secara ***Explisit (oleh pemrogram)*** dan ***Implisit (dikelola oleh computer).***
- Badan (body) pengulangan
  Yaitu bagian algoritma yang diulang.

Format Syntax Algoritma

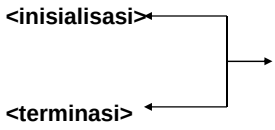
Stuktur pengulangan biasanya disertai dengan :

- **Inisialisasi**

  Aksi yang dilakukan sebelum pengulangan dilakukan pertama kali.
- **Terminasi**

  Aksi yang dilakukan setelah pengulangan selesai dilaksanakan.

Struktur pengulangan secara umum :



awal dan akhir pengulangan dinyatakan sebagai kata kunci yang bergantung pada struktur pengulangan yang dipakai. Inisialisasi dan terminasi adalah bagian yang opsional.

Tiga macam notasi struktur pengulangan :

- Struktur **FOR atau TRAVERSAL** *(Pengulangan tanpa kondisi / Unconditional Looping).*
- Struktur **WHILE..DO** *(Pengulangan dengan kondisi / Conditional Looping).*
- Struktur **REPEAT..UNTIL**

Tiga macam notasi struktur pengulangan dalam **bahasa C++** :

- **FOR**
- **WHILE**
- **DO..WHILE**

FOR

- Statement for digunakan untuk mengulang eksekusi suatu ungkapan dengan jumlah pengulangan yang sudah tertentu.
- Bentuk sintaksnya dalam algoritma :

FOR *var* ← *awal* to/downto *akhir* step counter DO
pernyataan
ENDFOR

atau

FOR *var* ← *awal* to/downto *akhir* DO
pernyataan
ENDFOR

atau

namavar TRAVERSAL [*awal..akhir*]
pernyataan

FOR

Bentuk sintaksnya dalam bahasa C++ sebagai berikut :

For(*ungkapan1*;*ungkapan2*;*ungkapan3*)
{
pernyataan;
}

Contoh penggunaan for pada C++

```
/* program:for*/
#include<iostream.h>
main()
{
  int n;
  for(n=1;n<=10;n++)
  {
    cout << n << endl;
  }
}
```

WHILE.. DO

- Bentuk umum struktur WHILE..DO dalam algoritma adalah :

  [*inisialisasi*]
  while (*kondisi*) do
  pernyataan
  {ada aksi thd var kondisi}
  endwhile

- Bentuk umum struktur WHILE dalam bahasa C++ adalah :

  [inisialisasi]
  while (*kondisi*)
  {
  pernyataan;
  */*ada aksi thd var kondisi*/*
  }

Cara kerja loop dengan while..do

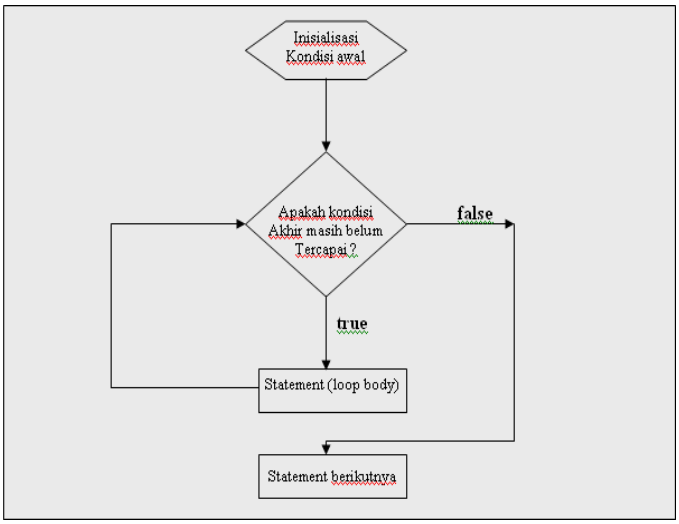
- melakukan inisialisasi, yaitu memberikan nilai awal yang ada kaitannya dengan nilai condition (kondisi)
- memeriksa nilai kondisi.
  1. Bila nilainya **true**, maka laksanakan loop yaitu mengerjakan instruksi yang ada dalam loop.
  2. Bila kondisi nilainya **false**, maka langsung keluar, melaksanakan instruksi selanjutnya.

  Loop selesai.

Perbedaan FOR dengan WHILE..DO

- FOR digunakan untuk proses pengulangan yang **jumlah pengulangannya dapat diketahui di awal**.
- WHILE..DO selain dapat berfungsi seperti FOR juga dapat digunakan untuk proses yang **jumlah pengulangannya tidak dapat diketahui.**

Struktur WHILE..DO



REPEAT…UNTIL

Bentuk umum struktur REPEAT…UNTIL dalam algoritma :

[inisialisasi]
Repeat
  pernyataan
  {ada aksi thd var kondisi}
Until(*keadaan*)

DO…WHILE

Bentuk umum struktur DO…WHILE dalam bahasa C++ :

[inisialisasi]
do
{
  pernyataan;
  */*ada aksi thd var kondisi*/*
}
while(*keadaan*)

Perbedaan Repeat…until pada algoritma dengan do…while pada bahasa C++



repeat…until : loop akan dikerjakan kembali jika kondisi masih salah dan **berhenti jika kondisi benar**.

do…while : Setiap loop atau pengulangan dikerjakan, maka kondisi akan di-cek. Jika masih benar, proses loop dilakukan lagi, dan **jika salah maka proses loop berhenti** dan berlanjut pada perintah selanjutnya.



Contoh dalam algoritma :

```
Algoritma Tulisan_10kali
{-------------------------------------------------
  Menampilkan tulisan Turbo C++ sepuluh kali
-------------------------------------------------}
DEKLARASI
        pencacah : integer
DESKRIPSI
        pencacah ← 0
        repeat
                output('Turbo C++');
         pencacah ← pencacah + 1
        until (pencacah >= 10)
```

Contoh dalam bahasa C++ :

```
/*
Dowhile.cpp - Menampilkan tulisan Turbo C++ sepuluh kali
                                                        */
#include <iostream.h>
main ()
{
        int pencacah ;
        pencacah = 0;
        do
        {
  cout << ”Turbo C++” << endl;
  pencacah ++;
        }
        while (pencacah <10);
}
```

Pada contoh tersebut kata “Turbo C++” akan dicetak sampai pencacah mencapai nilai = 10 atau lebih. Pencacah akan bertambah 1 setiap kali *body loop* dijalankan karena perintah pencacah++.

Contoh lain :

```
/* program:do while*/
#include<iostream.h>
#include<string.h>
int main()
{
        char checkword[80] = ”saya”;
        char password[80] = ””;
        do
        {
                cout << ”Enter password: ”;
                cin >> password;
        }
        while(strcmp(password, checkword));
        cout << "password benar ";
}
```

- Penjelasan :

  Pada contoh diatas kata “Enter Password :” akan diulangi terus sampai kita memasukkan password yang benar yaitu kata ‘saya’.