Algoritma dan Pemrograman

Pertemuan Ke-9
Statement Pengulangan 2

Disusun Oleh :
Wilis Kaswidjanti, S.Si.,M.Kom.

Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Pembangunan Nasional "Veteran"
Yogyakarta

Algoritma dan Pemrograman

Judul Materi : Statement Pengulangan 2

Deskripsi Materi : Materi ini membahas macam-macam statement pengulangan (while-do dan repeat-until).

Tujuan Instruksional Khusus :

1. Mendeskripsikan macam-macam statement perulangan
2. Menjelaskan perbedaan macam-macam statemen perulangan

BAB VII
STATEMENT PENGULANGAN 2

1. PENDAHULUAN

Statement pengulangan digunakan untuk mengerjakan suatu pernyataan yang dilakukan berulang-ulang sesuai jumlah pengulangan yang diinginkan atau sesuai dengan kondisi atau syarat yang diberikan. Materi pada pertemuan ini membahas sintak pengulangan sesuai dengan kondisi atau syarat yang diberikan.

2. PENYAJIAN

2.1. WHILE.. DO

Bentuk umum struktur WHILE..DO dalam algoritma adalah :

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

Bentuk umum struktur WHILE dalam bahasa C++ adalah :

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

Cara kerja loop dengan while..do

1. melakukan inisialisasi, yaitu memberikan nilai awal yang ada kaitannya dengan nilai condition (kondisi)
2. memeriksa nilai kondisi.
    a) Bila nilainya **true**, maka laksanakan loop yaitu mengerjakan instruksi yang ada dalam loop.
    Setelah melaksanakan loop, robah kondisi (change condision)
    Kemudian kembali memeriksa kondisi, dan seterusnya.

b) 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

Kondisi pengulangan diperiksa di awal pengulangan.

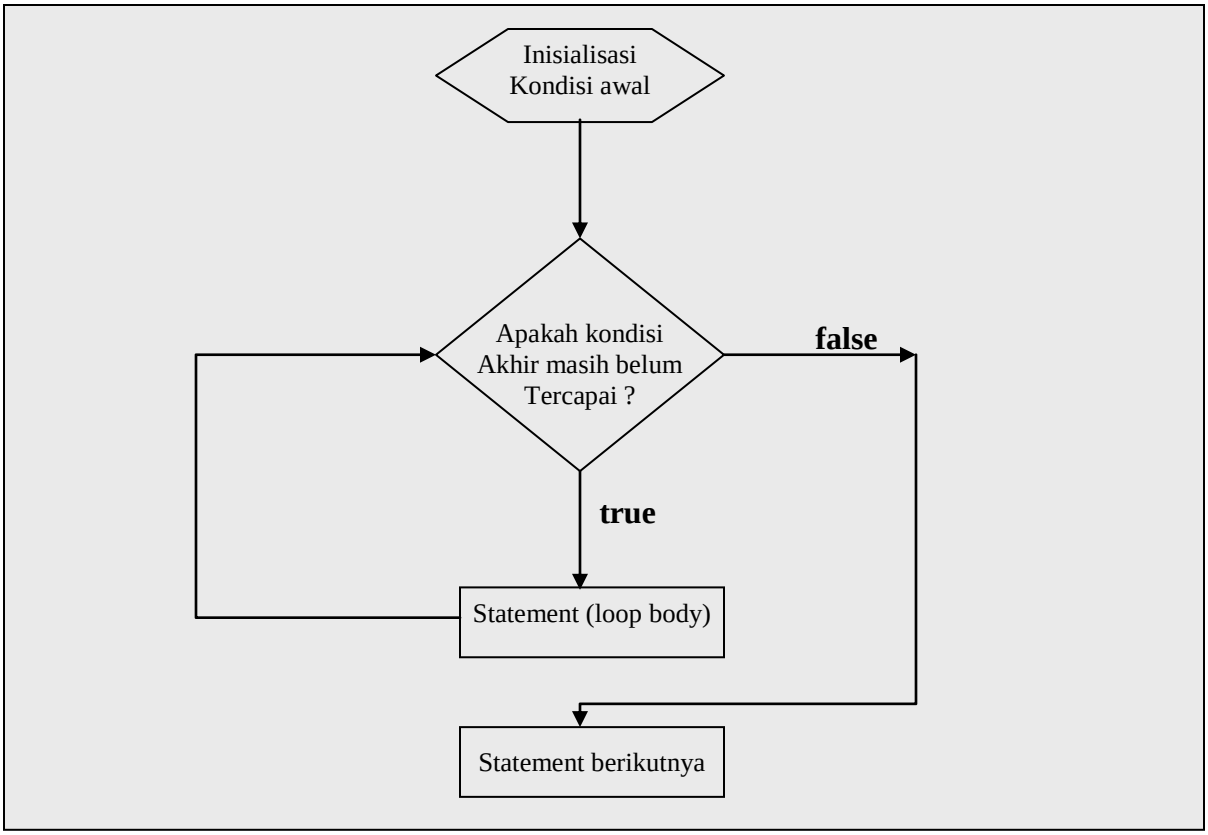
Jadi, instruksi didalam badan pengulangan hanya dapat dilaksanakan bila pengetesan kondisi menghasilkan nilai *true*.

Konsekuensinya, badan pengulangan mungkin tidak akan pernah dilaksanakan bila kondisi pengulangan pertama kali bernilai *false.*

Kapan kita dapat menggunakan WHILE

WHILE..DO

Gunakan struktur WHILE..DO pada kasus yang mengharuskan terlebih dahulu pemeriksaan kondisi objek sebelum objek tersebut dimanipulasi.



Flowchart WHILE..DO

Catatan : Sama seperti pada **for** kondisi adalah expresi Boolean yang dapat menghasilkan nilai Boolean benar atau salah.

Catatan :

Kesalahan yang sering muncul adalah melupakan **inisialisasi**. Karena jika sebuah peubah tidak di-inisialisasi-kan, maka ia sudah otomatis berisi sebuah nilai tertentu.

Contoh dalam algoritma :

```
Algoritma Tulisan_10kali
{ Menampilkan tulisan Turbo C++ sepuluh kali }

DEKLARASI
     pencacah : integer

DESKRIPSI
     pencacah ← 0
     while(pencacah <10)
         output('Turbo C++');
         pencacah ← pencacah + 1
     endwhile
```

Contoh dalam bahasa C++ :

```
/* ------------------------------------------------------------------------
     while.cpp - Menampilkan tulisan Turbo C++ sepuluh kali
  ------------------------------------------------------------------------*/

#include <iostream.h>
main ()
{
      int pencacah;

      pencacah = 0;
      while (pencacah <10)
      {
          cout << ”Turbo C++” << endl;
          pencacah++;
      }
}
```

atau :

```
/* ------------------------------------------------------------------------
     while.cpp - Menampilkan tulisan Turbo C++ sepuluh kali
  ------------------------------------------------------------------------*/

#include <stdio.h>
main ()
{
      int pencacah;

      pencacah = 0;
      while (pencacah <10)
      {
          puts(”Turbo C++\n”);
          pencacah++;
      }
}
```

Perhatikan program *while.cpp* :

- Mula-mula variable *pencacah* diisi dengan 0 sewaktu pendeklarasian. Variable inilah yang dipakai sebagai penentu berakhir tidaknya pengulangan dalam **while.**
- Lalu dieksekusi pengulangan while. Dicek apakah kondisi benar atau salan dengan pertanyaan apakah pencacah < 10, jika benar maka pernyataan-pernyataan yang berada dalam *{}* dijalankan. Pertama, pernyataan

  cout << ”Turbo C++ << endl;
  yang menampilkan tulisan “Turbo C++”.

  Selanjutnya pernyataan :

pencacah++;

yang digunakan untuk menaikkan isi pencacah sebesar 1.

- Pada putaran kedua, kondisi pencacah < 10 diperiksa kembali. Karena isi pecacah adalah 1, kondisi tersebut menghasilkan nilai benar sehingga pernyataan yang berada dalam {} dieksekusi kembali.
- Pada saat nilai pencacah dinaikkan menjadi 10, kondisi pencacah < 10 memberikan hasil berupa false sehingga eksekusi while segera dihentikan. Dengan cara demikian, pada layar tampil sepuluh kali tulisan "Turbo C++".

3. REPEAT…UNTIL

Bentuk umum struktur REPEAT…UNTIL dalam algoritma :

```
[inisialisasi]
Repeat
   pernyataan
   {ada aksi thd var kondisi}
Until(kondisi_berhenti)
```

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

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

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

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.

Perbedaan Repeat…until pada algoritma dengan do…while pada bahasa C++ adalah : pada bagian keadaan/kondisi pengecekannya. Pada repeat…until loop akan dikerjakan kembali jika kondisi masih salah dan berhenti jika kondisi benar, tetapi kalau do…while kebalikannya.

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);
}
```

atau :

```
/* ------------------------------------------------------------------
     Dowhile.cpp - Menampilkan tulisan Turbo C sepuluh kali
  ------------------------------------------------------------------*/

#include <stdio.h>
main ()
{
      int pencacah ;

      pencacah = 0;
      do
      {
          puts(”Turbo C++\n”);
          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 ";
}
```

atau :

```
/* program:do while*/
#include<stdio.h>
#include<string.h>

int main()
{
     char checkword[80] = ”saya”;
     char password[80] = ””;
     do
     {
        puts(”Enter password: ”);
        scanf(”%s”,&password);
     }
     while(strcmp(password, checkword));
     puts("password benar ");
}
```

Penjelasan :

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

PENUTUP

Penulisan notasi pengulangan dengan menggunakan sintak while pada algoritma dan bahasa C++ tidak berbeda jauh. Sedangkan Repeat-until pada algoritma tidak dipunyai oleh bahasa C++, melainkan menggunakan sintak Do-while. Pengulangan dengan sintak ini akan berhenti dengan mengecek syaratnya, jika tidak sesuai maka berhenti.

SOAL-SOAL

1. Ubahlah *penggalan* algoritma di bawah ini ke dalam bahasa C++ (dengan menggunakan **tiga** macam sintaks **pengulangan**)

```
for p←1 to n do
    output(p)
    for q←1 to p do
        output(q)
    endfor
endfor
```

2. Buatlah program kalkulator dengan menggunakan menu pilihan (penjumlahan, pengurangan, perkalian) dan program akan diulang-ulang terus sampai memilih keluar dari program. (pakai pengulangan do-while)

Referensi :

- Buku Teks
  1. Munir, Rinaldi (2005), *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, Buku 1, Edisi Ketiga, Penerbit Informatika Bandung.
  2. Charibaldi, N. (2004), *Modul Kuliah Algoritma Pemrograman I*, Edisi Kedua, Yogyakarta

- Buku Acuan/Referensi
  1. Brassard, Gilles (1999), *Fundamentals of algorithma*, PrinteceHall.
  2. Jarne, Stroustrup B. (1997), *C++ Programming language,* AT &T.
  3. Kristanto, Andri (2003), *Algoritma pemrograman C++*, Graha Ilmu.
  4. Schildt,Herbert (2000), *The Complete Reference C++*, McGraw-Hill.
  5. Sedgewick, R. (2000), *Algoritma Third edition In C part 5*, Addison Wesley.