Algoritma dan Pemrograman

Pertemuan Ke-6
Statement Pengendalian 1

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 Pengendalian 1

Deskripsi Materi : Materi ini membahas macam-macam statement pengendalian (if-then/if-then-else).

Tujuan Instruksional Khusus :

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

BAB VI
STATEMENT PENGENDALIAN 1

1. PENDAHULUAN

Bab ini memiliki kompetensi dasar untuk menggunakan dan membandingkan statement-statement pengendalian dengan sintak, semantik dan logika yang benar. Statement kendali digunakan untuk mengambil suatu keputusan atau memilih bagian program yang akan dikerjakan sesuai dengan kondisi atau syarat yang diberikan. Statemen-statemen di atas memerlukan suatu kondisi atau syarat sebagai dasar pengambilan keputusan. Salah satu kondisi yang umum digunakan adalah berupa keadaan benar atau salah (*true or false).*

2. PENYAJIAN

Ketika menyelesaikan masalah, kadang kita menemui berbagai masalah yang memerlukan pengecekan apakah suatu pekerjaan akan dikerjakan atau tidak harus sesuai dengan syarat. Ada beberapa format yang berbeda untuk setiap kasus ketika diimplementasikan di algoritma pemrograman.

2.1. Satu Kasus

Menggunakan konstruksi IF-THEN (jika-maka) dalam bentuk pernyataan :

if *kondisi* then
    pernyataan
endif

Pernyataan sesudah kata then (dapat berupa satu atau lebih pernyataan) adalah aksi yang hanya akan dilaksanakan bila *kondisi* bernilai benar (*true*). Bila *kondisi* bernilai salah (*false*), tidak ada pernyataan apapun yang dikerjakan. Kata endif sengaja ditambahkan untuk mempertegas awal dan akhir struktur IF-THEN.

Format bahasa C++ :

If *(**kondisi****)***
    pernyataan;

misal : Algoritma
    x : integer
    x ← 10

```
if (x > 5 ) then x ← x + 5
endif
output (x) = 15
```

Contoh Masalah :

Buatlah algoritma yang membaca sebuah bilangan bulat dari suatu papan ketik, lalu mencetak pesan "genap" jika bilangan tersebut adalah genap.

Penyelesaian :

Bilangan genap adalah bilangan yang habis dibagi dengan 2 (sisa pembagian = 0). Oleh karena itu, kita perlu membagi data masukan dengan 2. Jika data masukan habis dibagi 2, maka dapat ditulis bahwa bilangan tersebut bilangan genap.

```
ALGORITMA Genap
{ Mencetak pesan "bilangan genap" jika sebuah bilangan bulat
yang dibaca dari piranti masukan merupakan bilangan genap }

DEKLARASI
   x  :  integer

DESKRIPSI :
   read(x)
   if x mod 2 = 0 then
      write('genap')
   endif
```

Program Bahasa C++ :

```
#include <iostream.h>
main()
{
  int x;
  cout << "Masukkan bilangan : ";
  cin >> x;
  if (x % 2 == 0)
    cout << "genap";
}
```

atau

```
#include <stdio.h>
main()
{
  int x;
  puts("Masukkan bilangan : ");
  scanf("%d",&x);
  if (x % 2 == 0)
    puts("genap");
```

```
}
```

2.2. Dua Kasus

Menggunakan konstruksi IF-THEN-ELSE (jika-maka-kalau tidak) dalam bentuk pernyataan :

if *kondisi* then
    pernyataan$_1$
else
    pernyataan$_2$
endif

Pernyataan$_1$ dilaksanakan jika *kondisi* bernilai benar, sebaliknya jika *kondisi* bernilai salah, maka *pernyataan*$_2$ yang akan dilaksanakan. else menyatakan ingkaran (negation) dari *kondisi*.

Format bahasa C++ :

If ***(****kondisi****)***
    pernyataan$_1$;
else
    pernyataan$_2$;

Contoh Masalah :

Tulislah algoritma yang membaca sebuah bilangan bulat, lalu menuliskan pesan "genap" jika bilangan tersebut adalah genap atau "ganjil" jika bilangan tersebut adalah bilangan ganjil.

Penyelesaian :

Misalkan bilangan yang dibaca adalah x. Hanya ada dua kemungkinan jenis untuk x, yaitu bilangan genap atau bilangan ganjil. Bilangan genap adalah bilangan yang habis dibagi dengan 2 (sisa pembagian = 0), sedangkan bilangan ganjil bersisa 1 bila dibagi dengan 2. Contohnya, 4 adalah bilangan genap karena 4 mod 2 = 0, tapi 3 adalah bilangan ganjil karena 3 mod 2 =1.

Analisis kasus :

Kasus 1 : jika x mod 2 = 0, maka tulis pesan "bilangan genap"

Kasus 2 : jika x mod 2 ≠ 0, maka tulis pesan "bilangan ganjil"

```
ALGORITMA GenapGanjil
{ Mencetak pesan "genap" jika sebuah bilangan bulat yang
dibaca merupakan bilangan genap atau "ganjil" jika bilangan
tersebut ganjil }

DEKLARASI
    x : integer

DESKRIPSI
   read(x)
   if x mod 2 = 0 then
       write('genap')
   else
       write('ganjil')
   endif
```

Program Bahasa C++ :

```
#include <iostream.h>
main()
{
  int x;
  cout << "Masukkan bilangan : "; cin >> x;
  if (x % 2 == 0)
      cout << "genap";
  else
      cout << "ganjil";
}
```

atau :

```
#include <stdio.h>
main()
{
  int x;
  puts("Masukkan bilangan : "); scanf("%d",&x);
  if (x % 2 == 0)
      puts("genap");
  else
      puts("ganjil");
}
```

2.3. Tiga Kasus atau Lebih

Menggunakan konstruksi IF-THEN-ELSE (jika-maka-kalau tidak) bertingkat-tingkat dalam bentuk pernyataan :

Tiga Kasus :

```
if kondisi1 then
      pernyataan1
else
  if kondisi2 then
       pernyataan2
 else
       if kondisi3 then
          pernyataan3
       endif
 endif
endif
```

Empat Kasus :

```
   if kondisi1 then
      pernyataan1
   else
     if kondisi2 then
        pernyataan2
 else
       if kondisi3 then
          pernyataan3
       else
             if kondisi4 then
          pernyataan4
          endif
   endif
 endif
endif
```

Tiga kasus dalam format bahasa C++ :

```
if kondisi1
      pernyataan1;
else
  if kondisi2
       pernyataan2;
 else
       if kondisi3
          pernyataan3;
```

Contoh Masalah :

Tulislah algoritma yang membaca sebuah bilangan bulat, lalu menentukan apakah bilangan tersebut positif, negatif atau nol.

Penyelesaian :

Misalkan bilangan bulat itu adalah x.

Analisis Kasus

Kasus 1 : jika x > 0, maka x adalah bilangan positif

Kasus 2 : jika x < 0, maka x adalah bilangan negatif

Kasus 3 : jika x = 0, maka x adalah bilangan nol

```
ALGORITMA JenisBilanganBulat
{ Menentukan apakah sebuah bilangan bulat merupakan bilangan
positif, negatif atau nol }

DEKLARASI
   x : integer

DESKRIPSI :
   read(x)
   if x > 0 then
      write('positif')
   else
      if x < 0 then
         write('negatif')
      else
          if x = 0
             write('nol')
          endif
      endif
  endif
```

Program Bahasa C++ :

```
#include <iostream.h>
main()
{
  int x;
  cout << "Masukkan bilangan : "; cin >> x;
  if (x > 0)
      cout << "positif";
  else
      if (x < 0)
        cout << "negatif";
      else
        if (x == 0)
           cout << "nol";
}
```

atau :

```
#include <stdio.h>
main()
{
  int x;
  puts(”Masukkan bilangan : ”); scanf(”%d”,&x);
  if (x > 0)
       puts(”positif”);
  else
       if (x < 0)
         puts(”negatif”);
       else
         if (x == 0)
            puts(”nol”);
}
```

PENUTUP

Pada dasarnya ada dua macam statement pengendalian yaitu IF-THEN-ELSE dan CASE-OF. Pada prakteknya dua macam notasi tersebut dapat dimodifikasi sesuai kebutuhan.

SOAL-SOAL

1. Buatlah algoritma yang membaca sebuah bilangan bulat dari suatu papan ketik, lalu mencetak pesan “genap” jika bilangan tersebut adalah genap.
2. Susun program untuk menginput sebuah bilangan bulat lebih besar dari 1. Kemudian periksa apakah bilangan tersebut adalah bilangan PRIMA atau bukan. Bila bilangan tersebut bilangan prima, maka cetak perkataan “PRIMA”, sedangkan bila bilangan tersebut bukan bilangan prima maka cetak perkataan “BUKAN PRIMA”.

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.