

Materi Pertemuan 04

PEMILIHAN (*SELECTION*)/ PENGAMBILAN KEPUTUSAN (*DECISION*)

Disusun oleh :
Danang Junaedi

OBJEKTIF

Dalam bab ini mahasiswa mempelajari tentang :

1. Pengenalan proses Pemilihan (Selection)/Pengambilan Keputusan (Decision)
2. Penggunaan proses Pemilihan (Selection)/Pengambilan Keputusan (Decision)

TUJUAN & SASARAN

Setelah mempelajari materi dalam bab ini mahasiswa diharapkan dapat :

1. Menjelaskan mengenai proses Pemilihan (Selection)/Pengambilan Keputusan (Decision) , serta cara penulisannya dalam program
2. Menjelaskan penggunaan Pemilihan (Selection)/Pengambilan Keputusan (Decision)
3. Membedakan proses Pemilihan (Selection)/Pengambilan Keputusan (Decision)
4. Menggunakan proses Pemilihan (Selection)/Pengambilan Keputusan (Decision)

WAKTU & TEMPAT

1. 2 x 50 menit pertemuan di kelas
 - a. 50 menit materi
 - b. 50 menit tutorial (latihan soal)
2. 4 x 50 menit belajar di rumah

Pemilihan(*Selection*)/pengambilan keputusan (*decision*) merupakan salah satu komponen yang sangat penting dalam pembuatan program. Pemilihan(*Selection*)/ pengambilan keputusan (*decision*) merupakan suatu keadaan dalam dua nilai, yaitu *true* (ya) atau *false* (tidak). Tujuan pemilihan(*Selection*)/pengambilan keputusan (*decision*) adalah :

1. Untuk memilih salah satu alternatif jawaban atau pilihan yang ada
2. Mengontrol jalannya program agar dapat memilih salah satu dari sekian banyak pilihan yang ada
3. Memilih satu atau lebih statement yang akan diproses berdasarkan kondisi yang telah ditetapkan
4. Pemilihan solusi berdasarkan kriteria tertentu yang telah ditetapkan sebelumnya, untuk mendapatkan hasil yang optimal

Disini dilakukan perbandingan (dengan menggunakan operator perbandingan/relasi) antara dua hal atau lebih untuk kemudian diperoleh suatu keputusan berdasarkan hasil perbandingan tersebut. Dalam C/C++ terdapat empat jenis struktur pemilihan(*selection*)/pengambilan keputusan (*decision*), yaitu :

1. *One way selection* (if)
2. *Two way selection* (if- else)
3. *Multi ways selection* (if-else-if/*nested If* dan switch)
4. Struktur pemilihan(*selection*)/pengambilan keputusan (*decision*) case-of/depend-on(*switch*)

4.1 One Way Selection (if)

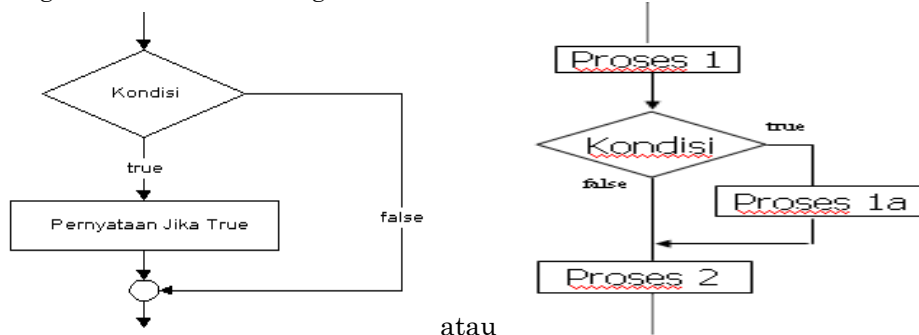
Bentuk ini merupakan bentuk paling sederhana dari keseluruhan struktur seleksi yang ada. Pada bentuk ini, hanya jika kondisi bernilai *true* (ya) maka proses atau statement didalam blok if tersebut yang akan dilaksanakan. Cara penulisannya adalah :

```

if ( expression )
    statement;
    atau
if ( expression )
    statement1;
    next_statement;

```

Atau dapat digambarkan dalam diagram berikut ini :



Penjelasan :

1. *Evaluate an expression and directs program execution depending on the result of that evaluation.*
2. *if expression evaluate as TRUE, statements is executed, if FALSE, statements is not executed, execution then passed to code follows the if statement, that is the next_statement.*
3. *So, execution of the statement depends on the result of expression.*

Contoh :

```
#include<stdio.h>
int main()
{
    int Nilai;
    printf("Masukan Nilai : ");
    scanf("%d",&Nilai);
    if (Nilai > 65)
        printf("Selamat anda lulus");
    return 0;
}
```

Outputnya adalah :

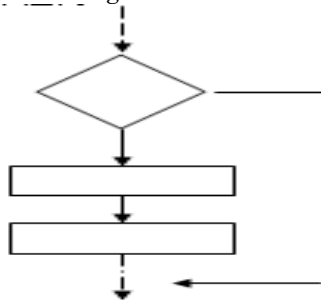
```
C:\WINDOWS\system32\cmd.exe
D:\ProgC++>ex4-01
Masukan Nilai : 60

D:\ProgC++>ex4-01
Masukan Nilai : 70
Selamat anda lulus
D:\ProgC++>
```

if statement also can control the execution of multiple statements through the use of a compound statement or a code block. A block is a group of two or more statements enclosed in curly braces, `{}`. Bentuknya adalah :

```
if ( expression )
{
    statement1;
    statement2;
    .
    .
    .
    statement-n;
}
```

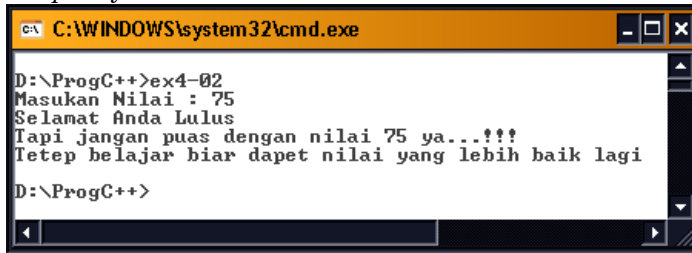
Atau dapat digambarkan dalam bentuk diagram berikut ini :



Contoh :

```
#include<stdio.h>
int main()
{
    int Nilai;
    printf("Masukan Nilai : ");
    scanf("%d",&Nilai);
    if (Nilai > 65)
    {
        printf("Selamat Anda Lulus\n");
        printf("Tapi jangan puas dengan nilai %d ya...!!!\n",Nilai);
        printf("Tetep belajar biar dapet nilai yang lebih baik lagi\n");
    }
    return 0;
}
```

Outputnya adalah :



4.2 Two Way Selection (if- else)

Bentuk ini merupakan penyempurnaan bentuk *one way selection*. Pada bentuk ini, proses atau statement didalam blok if tersebut akan dilaksanakan tergantung dari kondisi *true* (ya) atau *false* (tidak). Cara penulisannya adalah :

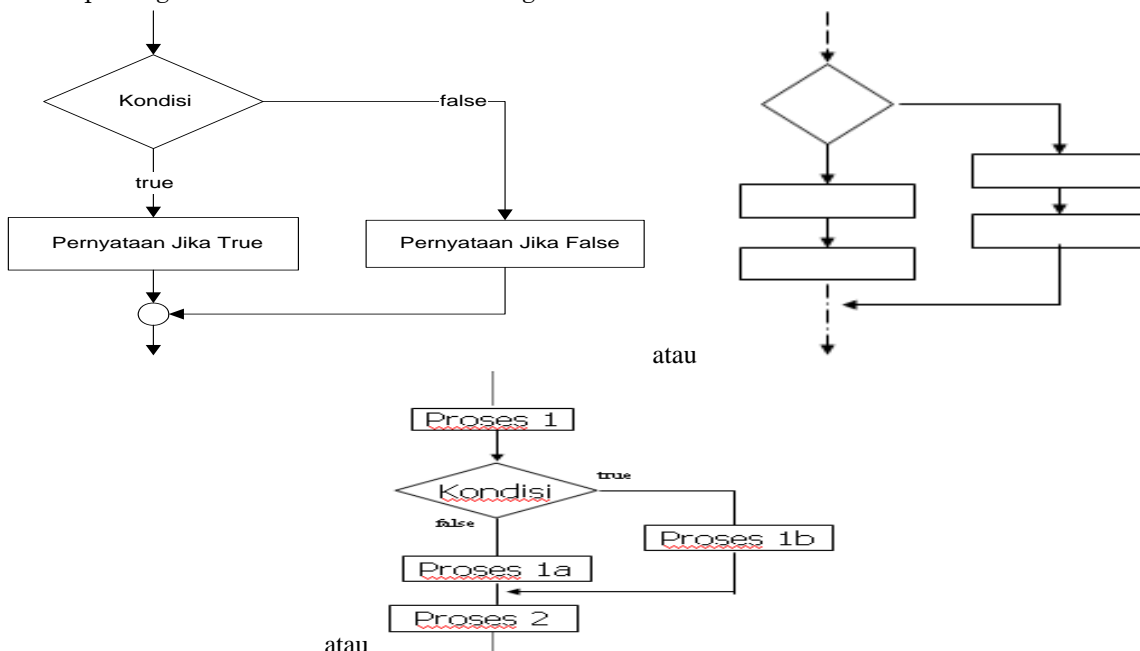
```

if (expression)
{
    statement a ke-1;
    ....
    statement a ke-n;
}
else
{
    statement b ke-1;
    ....
    statement b ke-n;
}
    
```

Semua **Statement a** dalam blok if di atas akan dijalankan jika **ekspresion bernilai True**

Semua **Statement b** dalam blok if di atas akan dijalankan jika **ekspresion bernilai False**

Atau dapat digambarkan dalam bentuk diagram berikut ini :



Contoh :

```
#include<stdio.h>
int main()
{
    int Nilai;
    printf("Masukan Nilai : ");
    scanf("%d",&Nilai);
    if (Nilai > 65)
    {
        printf("Selamat anda lulus\n");
    }
    else
    {
        printf("Maaf anda tidak lulus\n");
    }
    return 0;
}
```

Outputnya adalah :

```
D:\ProgC++>ex4-03
Masukan Nilai : 60
Maaf anda tidak lulus

D:\ProgC++>ex4-03
Masukan Nilai : 70
Selamat anda lulus

D:\ProgC++>
```

Syntax di atas dapat disederhanakan dengan menggunakan ternary operator. Syntaxnya adalah sebagai berikut :

(expression) ? statement a : statement b;

Statement a dalam blok di atas akan dijalankan jika **ekspresion** bernilai **True**

Statement b dalam blok di atas akan dijalankan jika **ekspresion** bernilai **False**

Contoh :

```
#include<stdio.h>
int main()
{
    int Nilai;
    printf("Masukan Nilai : ");
    scanf("%d",&Nilai);
    (Nilai > 65)? printf("Selamat anda lulus\n") : printf("Maaf anda tidak
lulus\n");
    return 0;
}
```

Outputnya adalah :

```
D:\ProgC++>ex4-04
Masukan Nilai : 60
Maaf anda tidak lulus

D:\ProgC++>ex4-04
Masukan Nilai : 70
Selamat anda lulus

D:\ProgC++>_
```

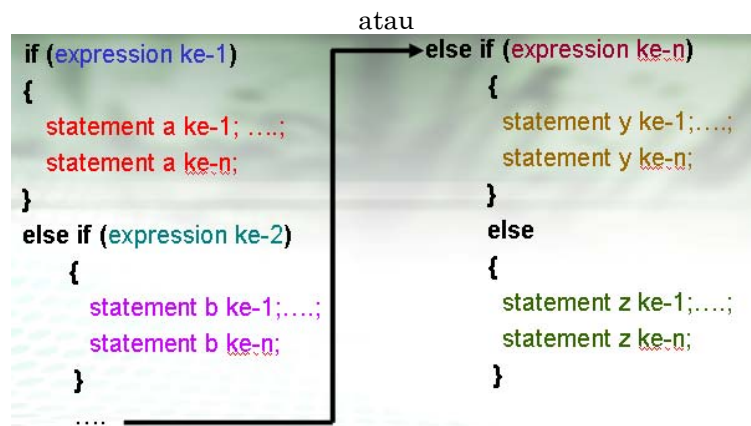
4.3 Multi Ways Selection (if-else-if/nested If)

Bentuk ketiga merupakan bentuk yang tidak memiliki struktur khusus, artinya keadaannya sangat tergantung dari kasus atau permasalahan yang terjadi. Yang dapat digunakan sebagai sandaran adalah apabila salah satu pohon keputusan, baik *true* (ya) ataupun *false* (tidak) diikuti oleh struktur seleksi lainnya, maka bentuk itulah yang dikatakan struktur seleksi bersarang (*nested if*). Cara penulisannya adalah :

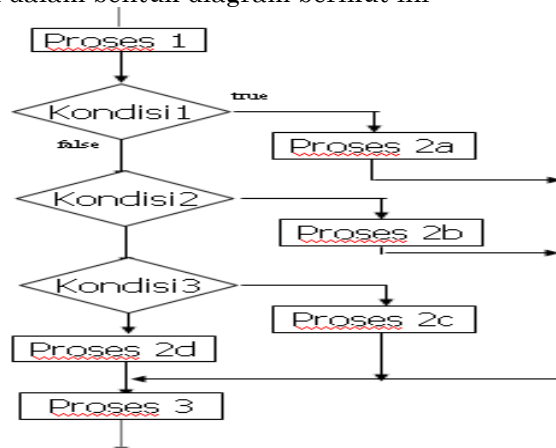
```

if ( expression )
    statement1;
else if ( expression )
    statement2;
else if ( ... )
    statement3;
...
...
...
else
    statementN;
next_statement;

```



Atau dapat digambarkan dalam bentuk diagram berikut ini



Penjelasan :

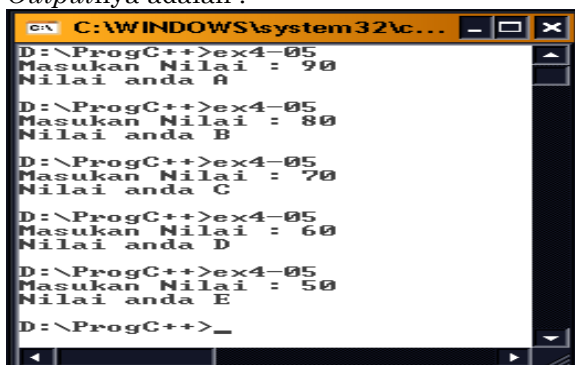
1. Semua **statement a** dalam blok if di atas akan dijalankan jika **expression ke-1** bernilai True
2. Jika **expression ke-1** bernilai False, maka **expression ke-2** akan dicek. Jjika **expression ke-2** bernilai True Semua **statement b** dalam blok if di atas akan dijalankan.

3. Dan begitu seterusnya sampai ke **statement z** atau akhir dari blok if

Contoh :

```
#include<stdio.h>
int main()
{
    int Nilai;
    printf("Masukan Nilai : ");
    scanf("%d",&Nilai);
    if (Nilai > 85)
    {
        printf("Nilai anda A\n");
    }
    else
    {
        if (Nilai > 75)
        {
            printf("Nilai anda B\n");
        }
        else
        {
            if (Nilai > 65)
            {
                printf("Nilai anda C\n");
            }
            else
            {
                if (Nilai > 55)
                {
                    printf("Nilai anda D\n");
                }
                else
                {
                    printf("Nilai anda E\n");
                }
            }
        }
    }
    return 0;
}
```

Outputnya adalah :



```
C:\WINDOWS\system32\c...
D:\ProgC++>ex4-05
Masukan Nilai : 90
Nilai anda A

D:\ProgC++>ex4-05
Masukan Nilai : 80
Nilai anda B

D:\ProgC++>ex4-05
Masukan Nilai : 70
Nilai anda C

D:\ProgC++>ex4-05
Masukan Nilai : 60
Nilai anda D

D:\ProgC++>ex4-05
Masukan Nilai : 50
Nilai anda E

D:\ProgC++>_
```

4.4 Struktur pemilihan(selection)/pengambilan keputusan (decision) case-of/depend-on(switch)

Merupakan bentuk lain dari model seleksi, dimana pada bentuk ini tidak dilakukan pengujian *true* (ya) atau *false* (tidak), tetapi suatu proses sudah dikelompokkan berdasarkan pilihan yang sudah ditetapkan. Tujuan dari model ini adalah :

1. Menyederhanakan bentuk multi ways selection
2. Mengurangi kemungkinan error karena tidak perlu menulis syntax switch lagi

Cara penulisannya adalah :

```
switch(expression)
{
    case template_1 : statement(s);
                    break;

    case template_2 : statement(s);
                    break;

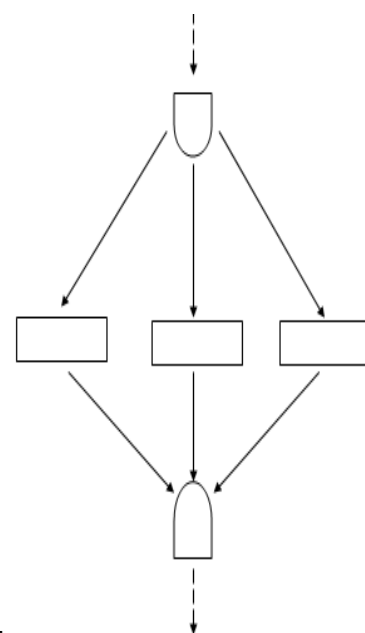
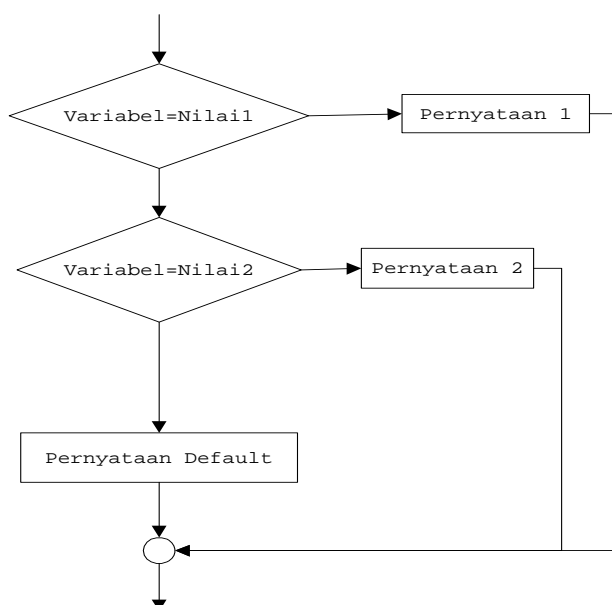
    ...
    ...
    ...
    case template_n : statement(s);
                    break;

    default : statement(s);
}
next_statement;
```

atau

```
switch (expression)
{
    case const-expression ke-1 : {statement a ke-1;...; statement a ke-n; break;}
    case const-expression ke-2 : {statement b ke-1;...; statement b ke-n; break;}
    ...
    case const-expression ke-n : {statement y ke-1;...; statement y ke-n; break;}
    default : {statement z ke-1;...;statement z ke-n; break;}
}
```

Atau dapat digambarkan dalam bentuk diagram berikut ini :



atau

Contoh :

```
#include<stdio.h>
int main()
{
    char Grade;
    printf("Masukan Grade Nilai : ");
    scanf("%c",&Grade);
    switch(Grade)
    {
        case 'A' : printf("Bobot nilai : 4\n");break;
        case 'B' : printf("Bobot nilai : 3\n");break;
        case 'C' : printf("Bobot nilai : 2\n");break;
        case 'D' : printf("Bobot nilai : 1\n");break;
        case 'E' : printf("Bobot nilai : 0\n");break;
        default : printf("Grade nilai yang anda masukan tidak dikenal\n");break;
    }
    return 0;
}
```

Outputnya adalah :

```
D:\ProgC++>ex4-06
Masukan Grade Nilai : A
Bobot nilai : 4
D:\ProgC++>ex4-06
Masukan Grade Nilai : B
Bobot nilai : 3
D:\ProgC++>ex4-06
Masukan Grade Nilai : C
Bobot nilai : 2
D:\ProgC++>ex4-06
Masukan Grade Nilai : D
Bobot nilai : 1
D:\ProgC++>ex4-06
Masukan Grade Nilai : E
Bobot nilai : 0
D:\ProgC++>ex4-06
Masukan Grade Nilai : e
Grade nilai yang anda masukan tidak dikenal
```

4.5 Operasi Logika AND dan OR

Operand1	Operand2	Output
0	0	0 (F)
0	1	0 (F)
1	0	0 (F)
1	1	1 (T)

Table 3.10: Logical AND Operation

Operand1	Operand2	Output
0	0	0 (F)
0	1	1 (T)
1	0	1 (T)
1	1	1 (T)

Table 3.11: Logical OR Operation

4.6 Referensi

1. Deitel, H.M. and Deitel, P.J., "C++ How to Program, 2nd Edition", Prentice Hall, 2003 (Bab 2)
2. Deitel, H.M. and Deitel, P.J., "C How to Program, 4nd Edition", Prentice Hall, 2004 (bab 3)
3. Junaedi, EP, Fajar,"Algoritma dan Pemrograman", Salemba Infotek, Jakarta,2007 [Bab 2]

4. Sekarwangi, Guntari;Kurniandi, Ocke, “ Diktat Kuliah Pemrograman I (Terstuktur)”, Teknik Informatika – Universitas Widyatama, Bandung, 2004 [Bab 4]
5. Kristanto, Andi”Struktur Data dengan C++”, Graha Ilmu, Yogyakarta, 2003 [Bab 4]
6. <http://www.geocities.com/paylook/module3.html>, Tanggal Akses : 6 Oktober 2004
7. <http://www.geocities.com/paylook/module6.html>, Tanggal Akses : 6 Oktober 2004
8. Herianto, “Presentasi Pemrograman Terstruktur.ppt”,2004
9. Yatini B., Indra;Nasution, Erliansyah, “Algoritma & Struktur Data dengan C++”, Graha Ilmu,Yogyakarta, 2005 [bab 2]

4.7 Bahan Renungan (Semoga bisa menjadi tambahan motivasi)

TUJUH KEAJAIBAN DUNIA

Sekelompok siswa kelas geografi sedang mempelajari "Tujuh Keajaiban Dunia". Pada awal dari pelajaran, mereka diminta untuk membuat daftar apa yang mereka pikir merupakan "Tujuh Keajaiban Dunia" saat ini. Walaupun ada beberapa ketidaksesuaian, sebagian besar daftar berisi;

1. Piramida
2. Taj Mahal
3. Tembok Besar Cina
4. Menara Pisa
5. Kuil Angkor
6. Menara Eiffel
7. Kuil Parthenon

Ketika mengumpulkan daftar pilihan, sang guru memperhatikan seorang pelajar, seorang gadis yang pendiam, yang belum mengumpulkan kertas kerjanya. Jadi, sang guru bertanya kepadanya apakah dia mempunyai kesulitan dengan daftarnya.

Gadis pendiam itu menjawab, "Ya, sedikit. Saya tidak bisa memilih karena sangat banyaknya." Sang guru berkata, "Baik, katakan pada kami apa yang kamu miliki, dan mungkin kami bisa membantu memilihnya."

Gadis itu ragu sejenak, kemudian membaca, "Saya pikir, "Tujuh Keajaiban Dunia" adalah,

1. Bisa melihat
2. Bisa mendengar
3. Bisa menyentuh
4. Bisa menyayangi

Dia ragu lagi sebentar, dan kemudian melanjutkan,

5. Bisa merasakan
6. Bisa tertawa
7. Dan, bisa mencintai

Ruang kelas tersebut sunyi seketika. Alangkah mudahnya bagi kita untuk melihat pada eksploitasi manusia dan menyebutnya "keajaiban". Sementara kita lihat lagi semua yang telah Tuhan karuniakan untuk kita, kita menyebutnya sebagai "biasa". Semoga anda hari ini diingatkan tentang segala hal yang betul betul ajaib dalam kehidupan anda.

Sumber : Tidak Diketahui