

Pemilihan (*Selection*)

Struktur kendali pemilihan digunakan untuk memilih langkah-langkah mana yang akan dikerjakan terlebih dahulu berdasarkan hasil pemeriksaan suatu kondisi.

Bahasa C menyediakan 5 langkah dalam pemilihan yaitu :

- *if*

If digunakan untuk memilih apakah proses tersebut akan dilakukan atau tidak. Berikut bentuk dari sintax *if* yang digunakan dalam bahasa C :

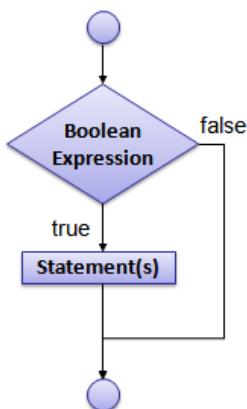
```
if(expression) { statement(s); }
```

Dimana *expression* adalah syarat yang akan dikerjakan, syarat ini memiliki nilai benar (*true*) atau salah (*false*). Bila *expression* tersebut bernilai benar (*true*), *statement* di dalam blok *if* tersebut akan dijalankan. Bila nilai *expression* bernilai salah (*false*), maka program akan berakhir. Berikut merupakan contoh program *if* pada bahasa C:

```
if(r>=0)
{
    area = r*r*pi;
    printf("Area dari r : %d adalah %f",r, area);
}
```

Pada contoh potongan program diatas, misalkan nilai dari radius adalah 5. Program akan mengeksekusi perintah yang ada di blok *if* tersebut bila syarat terpenuhi. Pertama-tama akan dilakukan pengecekan apakah nilai radius saat ini bernilai lebih dari atau sama dengan 0? Jawabannya adalah benar, $5 \geq 0$, maka program akan menghitung nilai dari area, dan kemudian akan dicetak nilai dari radius dan area tersebut. Bila tidak, maka tidak ada *statement* yang akan dieksekusi(dijalankan).

Secara *flowChart* bentuk dari *if* adalah :



- *if-else*

If-else digunakan untuk memilih satu dari 2 proses yang akan dieksekusi. Berikut bentuk dari sintax *if-else* yang digunakan dalam bahasa C :

```

if (expression){
    statement(s)-for-the-true-case;
}
else {
    statement(s)-for-the-false-case;
}

```

Statement untuk kondisi yang benar diletakkan didalam blok *if* yang dibatasi dengan tanda kurung kurawal buka dan tutup. Kemudian untuk statement kondisi yang salah diletakkan pada *else* blok yang diawali dan diakhiri dengan kurung kurawal buka dan tutup juga.

Berikut contoh program dari penggunaan *if-else* pada bahasa C:

```

#include <stdio.h>

int main()
{
    int angka;
    scanf("%d", &angka);
    if(angka<0 || angka>100)
        printf("Wrong Input");
    else

        printf("Input : %d", angka);

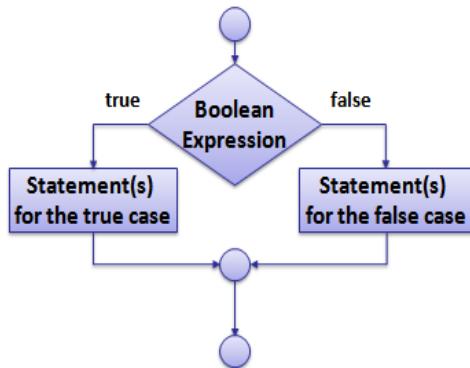
    getchar();
    return 0;
}

```

Pada program tersebut meminta inputan yang akan ditampung pada variabel angka. Fungsi seleksi dilakukan dimana terdapat dua buah kondisi yang dihubungkan dengan logical operator OR “||”, yang berarti bahwa bila salah satu kondisi terpenuhi atau benar maka *expression* tersebut akan bernilai benar. Andaikan **angka** bernilai 200, maka program akan mengecek apakah **angka** < 0? Jawabannya adalah salah (*false*), kemudian program akan mengecek kembali kondisi berikutnya apakah **angka** > 100? 200 bernilai lebih dari 100, maka kondisinya adalah benar. Karena relasi yang ada adalah OR, maka nilai akhir dari syarat ini adalah benar, sehingga program akan mencetak “*Wrong input*”. Tetapi sebaliknya bila kondisi **angka** yang dimasukkan adalah 50, maka kondisi pertama dan kedua bernilai salah, dan syarat tidak terpenuhi, maka program akan mencetak “*Input :* ” yang dilanjutkan dengan nilai dari **angka** tersebut.

Bila melihat kembali pada program yang ada, pada blok statement *if* ataupun *else* tidak terdapat tanda kurung kurawal buka dan tutup “{“ “}” yang biasanya digunakan untuk menandakan batasan blok. Hal ini dapat dilakukan jika hanya ada satu statement yang akan dieksekusi pada blok tersebut. Bila terdapat lebih dari satu statement yang ingin dilakukan pada blok statement *if*, maka perlu digunakan tanda kurung kurawal buka dan tutup “{“ “}” sebagai penanda.

Secara flow Chart bentuk dari *if-else* adalah :



- *nested if*

Nested if merupakan jenis variasi yang digunakan bila memiliki lebih dari satu syarat dan syarat tersebut saling berkaitan. Yang dimaksud dengan *nested* adalah di dalam *if* atau *if-else* statement terdapat *if* atau *if-else* statement lainnya. Tidak ada batasan maximum untuk *if* di dalam *if* tersebut. Berikut adalah contoh *nested if*:

```
if (score >= 90)
    grade = 'A';
else
    if (score >= 80)
        grade = 'B';
    else
        if (score >= 70)
            grade = 'C';
        else
            if (score >= 60)
                grade = 'D';
            else
                grade = 'F';
```

```
if (score >= 90)
    grade = 'A';
else if (score >= 80)
    grade = 'B';
else if (score >= 70)
    grade = 'C';
else if (score >= 60)
    grade = 'D';
else
    grade = 'F';
```

Potongan coding diatas adalah contoh penyeleksian dalam memberikan grade berdasarkan dari nilai yang ada. Bila kondisi tidak dipenuhi maka didalam blok *else* terdapat kondisi lainnya yang dicek kembali, dan seterusnya. Ini adalah salah satu contoh sederhana dari *nested if*. Potongan coding dikolom sebelah kiri dan kanan adalah sama, hanya lebih disarankan untuk mengikuti penulisan coding di kolom sebelah kanan.

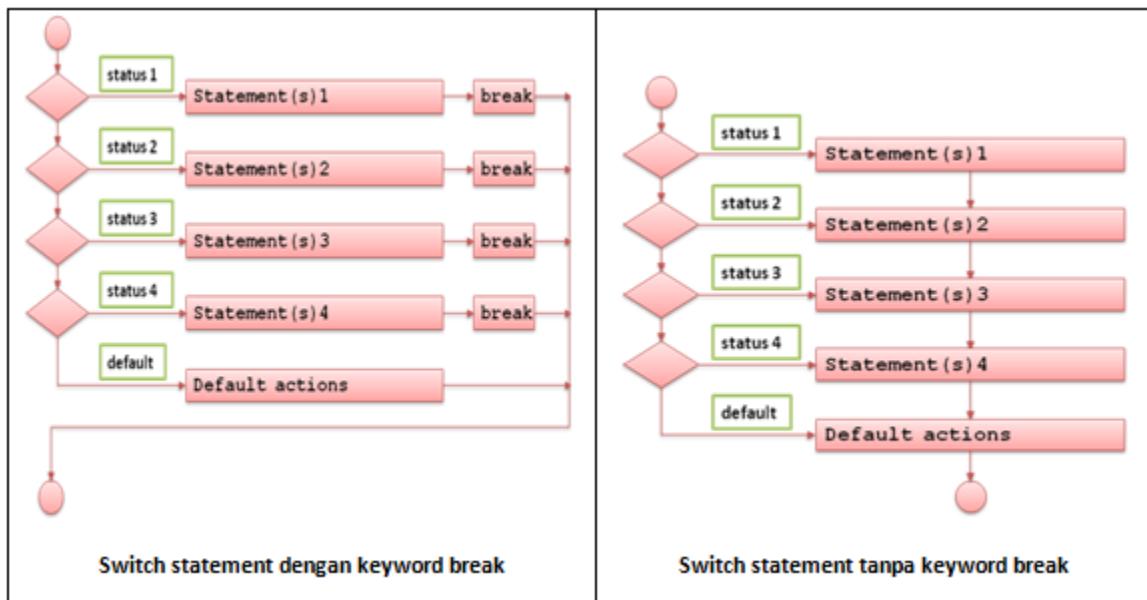
- *switch-case*

Switch – case adalah salah satu cara seleksi yang digunakan untuk memilih satu dari sejumlah alternatif. Seleksi yang dilakukan pada *switch* statement berdasarkan pada

sebuah status. Format penulisan statement *switch* itu sendiri dapat dilihat dibawah ini :

```
switch (switch-expression)
{
    case value1:    statement(s) 1;
        break;
    case value2:    statement(s) 2;
        break;
    ...
    case valueN:    statement(s) N;
        break;
default:    statement(s)-for-default;
}
```

switch-expression hanya boleh menggunakan sebuah nilai dengan tipe karakter(*char*), *byte*, *short* atau *int*. Kemudian untuk nilai *value1*, *value2*, *valueN* harus mempunyai tipe data yang sama seperti tipe data yang ada di *switch-expression*. Statement-statement pada setiap pilihan (*case*) akan dieksekusi sampai program membaca keyword *break* (**akan dijelaskan lebih rinci di topik selanjutnya**), atau hingga *switch* statement telah berakhir. Keyword *break* dan *default case* bersifat optional, boleh digunakan boleh juga tidak. *Case* statement tersebut juga dieksekusi secara berurut (*sequential*).



Gunakan keyword *break* bila memang diperlukan. Karena bila tidak ada keyword *break*, misalnya status 2 terpenuhi, maka statement 2 dan seterusnya akan dieksekusi juga selama blok *switch* belum berakhir. Untuk lebih kelas dapat dilihat pada *flow chart* diatas.

Dibawah ini adalah contoh program sederhana yang menerapkan *switch statement* dalam melakukan seleksi. Seleksi dilakukan terhadap hasil input user berupa bilangan integer yang disimpan ke dalam variable number. Yang menjadi *switch expression* adalah nilai sisa dari number dibagi dua (modulus). Statement di case 0 akan dijalankan bila sisa bagi bernilai 0, dan statement di case 1 akan dijalankan bila sisa bagi tersebut bernilai 1. Di dalam setiap statement *case* juga diberikan *keyword break*. Program sederhana ini merupakan contoh dari penentuan apakah sebuah angka merupakan bilangan ganjil atau genap.

```
#include <stdio.h>

int main()
{
    int number;
    scanf("%d", &number);
    switch(number%2)
    {
        case 0 : printf("%d is even number",number);
                   break;
        case 1 : printf("%d is odd number",number);
                   break;
    }
    getchar();
    return 0;
}
```

Berikut adalah contoh *input* dan *output* dari program tersebut:

9
9 is odd number

10
10 is even number

- *Conditional expression*

Jenis terakhir dari seleksi statement dapat juga dilakukan dengan bentuk *conditional expression*. Format yang digunakan adalah sebagai berikut:

Boolean-expression ? expression1 : expression2;

Lihatlah contoh *if* sederhana yang dituliskan dalam bentuk *conditional expression* berikut ini:

<pre>if (x > 0) y = 1; else y = -1;</pre>	Use the conditional Expression	<pre>y = (x > 0) ? 1 : -1;</pre>
--	--------------------------------	-------------------------------------