

PRAKTIKUM 4

PENGAMBILAN KEPUTUSAN

A. TUJUAN

1. Menjelaskan tentang operator kondisi (operator relasi dan logika)
2. Menjelaskan penggunaan pernyataan if
3. Menjelaskan penggunaan pernyataan if-else
4. Menjelaskan penggunaan pernyataan if dalam if
5. Menjelaskan penggunaan pernyataan else-if
6. Menjelaskan penggunaan pernyataan switch

B. DASAR TEORI

Untuk keperluan pengambilan keputusan, C menyediakan beberapa jenis pernyataan, berupa

- Pernyataan *if*
- Pernyataan *if-else*, dan
- Pernyataan *switch*

Pernyataan-pernyataan tersebut memerlukan suatu kondisi, sebagai basis dalam pengambilan keputusan. Kondisi umum yang dipakai berupa keadaan benar dan salah.

Operator Relasi

Operator relasi biasa dipakai untuk membandingkan dua buah nilai. Hasil perbandingan berupa keadaan benar atau salah. Keseluruhan operator relasi pada C ditunjukkan pada Tabel 3-1

Tabel 3-1. Operasi relasi

Operator	Makna
>	Lebih dari
>=	Lebih dari atau sama dengan
<	Kurang dari
<=	Kurang dari atau sama dengan
==	Sama dengan
!=	

	Tidak sama dengan
--	-------------------

Operator Logika.

Operator logika biasa dipakai untuk menghubungkan ekspresi relasi. Keseluruhan operator logika ditunjukkan pada tabel 3-2.

Tabel 3-2. Operator logika

Operator	Makna
&&	dan (AND)
	atau (OR)
!	tidak (NOT)

Bentuk pemakaian operator && dan || adalah

operand1 operator operand2

Pernyataan *if*

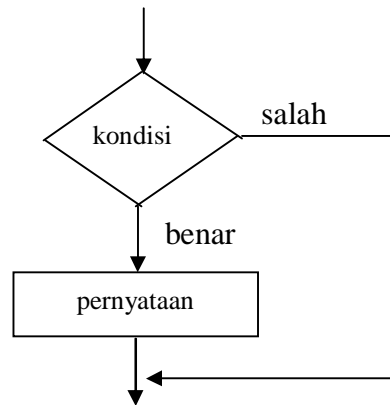
Pernyataan *if* mempunyai bentuk umum :

**if (kondisi)
pernyataan;**

Bentuk ini menyatakan :

- jika kondisi yang diseleksi adalah benar (bernilai logika = 1), maka pernyataan yang mengikutinya akan diproses.
- Sebaliknya, jika kondisi yang diseleksi adalah tidak benar (bernilai logika = 0), maka pernyataan yang mengikutinya tidak akan diproses.

Mengenai kondisi harus ditulis diantara tanda kurung, sedangkan pernyataan dapat berupa sebuah pernyataan tunggal, pernyataan majemuk atau pernyataan kosong. Diagram alir dapat dilihat seperti gambar 3.1



Gambar 3.1. Diagram alir *if*

Pernyataan *if-else*

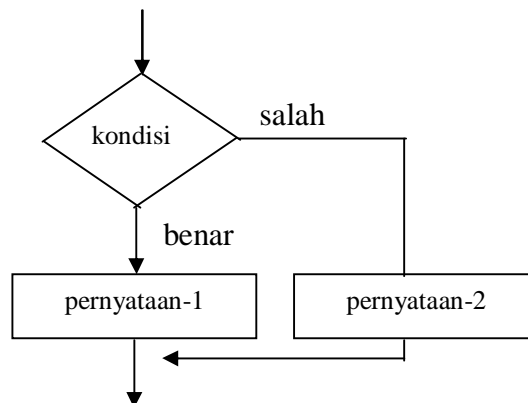
Pernyataan *if-else* memiliki bentuk :

```

if (kondisi)
  pernyataan-1;
else
  pernyataan-2;

```

Diagram alir dapat dilihat seperti gambar 3.2.



Gambar 3.2. Diagram alir *if-else*

Arti dari pernyataan *if-else* :

- Jika kondisi benar, maka **pernyataan-1** dijalankan.
- Sedangkan bila kondisi bernilai salah, maka **pernyataan-2** yang dijalankan.

Masing-masing **pernyataan-1** dan **pernyataan-2** dapat berupa sebuah pernyataan tunggal, pernyataan majemuk ataupun pernyataan kosong.

Contoh penggunaan pernyataan *if-else* adalah untuk menyeleksi nilai suatu bilangan pembagi. Jika nilai bilangan pembagi adalah nol, maka hasil pembagian dengan

nilai nol akan mendapatkan hasil tak berhingga. Jika ditemui nilai pembaginya nol, maka proses pembagian tidak akan dilakukan.

Pernyataan *if* di dalam *if*

Di dalam suatu pernyataan *if* (atau *if-else*) bisa saja terdapat pernyataan *if* (atau *if-else*) yang lain. Bentuk seperti ini dinamakan sebagai *nested if*. Secara umum, bentuk dari pernyataan ini adalah sebagai berikut :

```
if (kondisi-1)
    if (kondisi-2)
        .
        .
        if(kondisi-n)
            pernyataan;
        else
            pernyataan;
        .
        .
    else
        pernyataan;
else
    pernyataan;
```

- Kondisi yang akan diseleksi pertama kali adalah kondisi yang terluar (**kondisi-1**). Jika **kondisi-1** bernilai salah, maka statemen *else* yang terluar (pasangan *if* yang bersangkutan) yang akan diproses. Jika *else* (pasangannya tsb) tidak ditulis, maka penyeleksian kondisi akan dihentikan.
- Jika **kondisi-1** bernilai benar, maka kondisi berikutnya yang lebih dalam (**kondisi-2**) akan diseleksi. Jika **kondisi-2** bernilai salah, maka statemen *else* pasangan dari *if* yang bersangkutan yang akan diproses. Jika *else* (untuk **kondisi-2**) tidak ditulis, maka penyeleksian kondisi akan dihentikan.
- Dengan cara yang sama, penyeleksian kondisi akan dilakukan sampai dengan **kondisi-n**, jika kondisi-kondisi sebelumnya bernilai benar.

Pernyataan *else-if*

Contoh implementasi *nested if* ini misalnya pembuatan sebuah program kalkulator sederhana. User memberikan masukan dengan format :

operand1 operator operand2

Jenis operasi yang dikenakan bergantung pada jenis **operator** yang dimasukkan oleh user. Oleh karena itu program akan mengecek apakah **operator** berupa tanda '*', '/', '+', atau tanda '-'.

- Jika operator berupa tanda '*' maka **operand1** akan dikalikan dengan **operand2**.
- Jika operator berupa tanda '/' maka **operand1** akan dibagi dengan **operand2**.
- Jika operator berupa tanda '+' maka **operand1** akan dijumlahkan dengan **operand2**.
- Jika operator berupa tanda '-' maka **operand1** akan dikurangi dengan **operand2**.
- Kalau operator yang dimasukkan bukan merupakan salah satu dari jenis operator di atas, maka ekspresi tersebut tidak akan diproses, dan user akan mendapatkan pesan berupa: "Invalid operator !"

Pernyataan *switch*

Pernyataan ***switch*** merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah alternatif, misalnya untuk menggantikan pernyataan ***if*** bertingkat.

Bentuk umum pernyataan ***switch*** adalah :

```
switch (ekspresi)
{
    case konstanta-1:
        pernyataan-1;
        .....
        break;
    case konstanta-2:
        .
        .
        .
    case konstanta-n:
        pernyataan-n;
        .....
        break;
    default:
        .....
        .....
        break;
}
```

dengan **ekspresi** dapat berupa ekspresi bertipe integer atau bertipe karakter. Demikian juga **konstanta-1**, **konstanta-2**, ..., **konstanta-n** dapat berupa konstanta integer atau

karakter. Setiap pernyataan-i (**pernyataan-1**, ... , **pernyataan-n**) dapat berupa pernyataan tunggal ataupun pernyataan jamak. Dalam hal ini urutan penulisan pernyataan *case* tidak berpengaruh. Proses penyeleksian berlangsung sebagai berikut :

- pengujian pada *switch* akan dimulai dari **konstanta-1**. Kalau nilai **konstanta-1** cocok dengan ekspresi maka **pernyataan-1** dijalankan. Kata kunci *break* harus disertakan di bagian akhir setiap pernyataan *case*, yang akan mengarahkan eksekusi ke akhir *switch*.
- Kalau ternyata **pernyataan-1** tidak sama dengan nilai **ekspresi**, pengujian dilanjutkan pada **konstanta-2**, dan berikutnya serupa dengan pengujian pada **konstanta-1**.
- Jika sampai pada pengujian *case* yang terakhir ternyata tidak ada kecocokan, maka pernyataan yang mengikuti kata kunci *default* yang akan dieksekusi. Kata kunci *default* ini bersifat opsional.
- Tanda kurung kurawal tutup (}) menandakan akhir dari proses penyeleksian kondisi *case*.

C. TUGAS PENDAHULUAN

1. Buatlah program dengan menggunakan pernyataan if-else untuk menentukan kelulusan mahasiswa yang ditentukan berdasarkan nilai ujian yang didapatkan. Jika nilai ujian lebih dari atau sama dengan 70 mahasiswa dinyatakan lulus dan jika kurang dari 70 maka dinyatakan tidak lulus.

Tampilan:

Masukkan nilai ujian: 65
Anda tidak lulus

Tampilan 1

Masukkan nilai ujian: 80
Anda lulus

Tampilan 2

1. Buatlah program yang membaca sebuah integer dari 1 sampai dengan 7, dan menuliskan nama hari yang bersesuaian dengannya di layar. Integer 1 bersesuaian dengan hari minggu , integer 2 dengan hari senin, dan seterusnya.
Gunakan perintah if bertingkat(else-if)

Dimana tampilan yang dihasilkan sebagai berikut:

Hari ke...:1 } Tampilan 1
Minggu

Hari ke...:6 } Tampilan 2
Jum'at

D. PERCOBAAN

1. Dengan menggunakan pernyataan if-else bertingkat, buatlah program untuk memilih penghitungan luas segitiga, persegi panjang atau bujur sangkar.

Tampilan :

Masukkan pilihan anda : 1

Masukkan alas dan tinggi : 10 20

Jadi luasnya adalah 100

Masukkan pilihan anda : 2

Masukkan panjang dan lebar : 10 20

Jadi luasnya adalah 200

2. Dengan menggunakan pernyataan switch-case, buatlah program untuk mengkonversikan nilai huruf ke nilai angka. Huruf A memiliki nilai angka 4, huruf B memiliki nilai angka 3, huruf C memiliki nilai angka 2, huruf D memiliki nilai angka 1, sedangkan huruf E memiliki nilai angka 0.

Tampilan:

Masukkan nilai huruf:A } Tampilan 1
Nilai angka 4

.

.

Masukkan nilai huruf:E } Tampilan 4
Nilai angka 0

3. Dengan menggunakan nested if, buatlah program untuk mencari nilai terkecil dari 3 angka yang diinputkan. Adapun potongan programnya adalah

```
if(a<b)
    if(a<c)
        result=a;
    else
        result=c;
else if(b<c)
    result=b;
else
    result=c;
```

Tampilan:

Masukkan 3 angka:4 5 7

Angka yang terkecil adalah 4

4. Kerjakan soal no 2 dengan menggunakan pernyataan if else.

E. LAPORAN RESMI

1. kerjakan program pada percobaan 1 dengan perintah switch-case
2. Buatlah flowchart dari percobaan 1 sampai percobaan 4