

Praktikum 3 (1/3)

PENGAMBILAN KEPUTUSAN

A. TUJUAN

1. Menjelaskan tentang operator kondisi (operator relasi dan logika)
2. Menjelaskan penggunaan pernyataan if
3. Menjelaskan penggunaan pernyataan if-else

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*

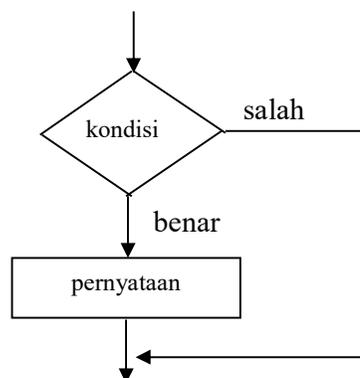
Pernyataan *if* mempunyai bentuk umum :

**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



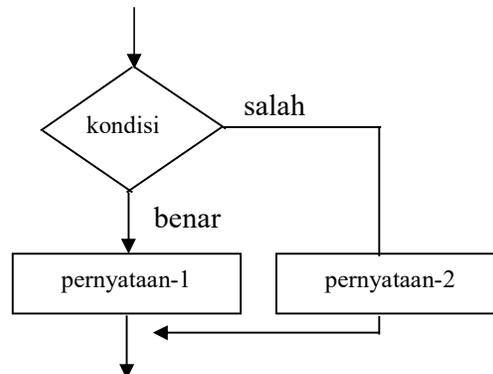
Gambar 3.1. Diagram alir *if*

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 **pernyataaan-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.

C. TUGAS PENDAHULUAN

Desainlah algoritma dan flowchart untuk semua soal pada percobaan

D. PERCOBAAN

1. Buat program untuk menginputkan sebuah bilangan, kemudian cetak ke layar bilangan tersebut dan beri komentar apakah bilangan tersebut ganjil atau genap.

Contoh input = 15

Output = Bilangan yang diinputkan adalah 15.

15 adalah bilangan ganjil.

2. Buat program menggunakan pernyataan *if* adalah untuk menentukan besarnya potongan harga yang diterima oleh seorang pembeli, berdasarkan kriteria :

- tidak ada potongan harga jika total pembelian kurang dari Rp. 100.000 (dalam hal ini potongan harga diinisialisasi dengan nol).
- bila total pembelian lebih dari atau sama dengan Rp. 100.000, potongan harga yang diterima dirubah menjadi sebesar 5% dari total pembelian.

Contoh input = 50.000

Output = Total pembelian adalah Rp. 50.000

3. Ulangi soal no.2. Output diganti dengan:

Total pembelian adalah Rp. 50.000

Anda tidak mendapat diskon.

4. Gunakan pernyataan *if...else* untuk menentukan sebuah bilangan masukan adalah bilangan prima atau bukan.

Input : x

Output : x adalah bilangan prima / bukan bilangan prima

5. Gunakan pernyataan *if...else* untuk membuat program yang menerima 2 buah bilangan bulat masukan. Tampilkan hasil dari pembagian bilangan pertama dengan bilangan kedua, dengan ketelitian 3 desimal.

Input : bil1 dan bil2

Output : hasil bagi bil1 dengan bil2

Nilai tambah : program bisa mengecek pembagian dengan nol, yaitu jika bilangan kedua adalah nol, maka tidak dilakukan proses pembagian, namun ditampilkan pesan kesalahannya (division by zero).

6. Buatlah program yang menerima input sembarang bilangan bulat. Cek apakah input tsb ada pada range 1-100, dan tampilkan hasilnya.

Contoh : Masukkan sembarang bilangan : 50

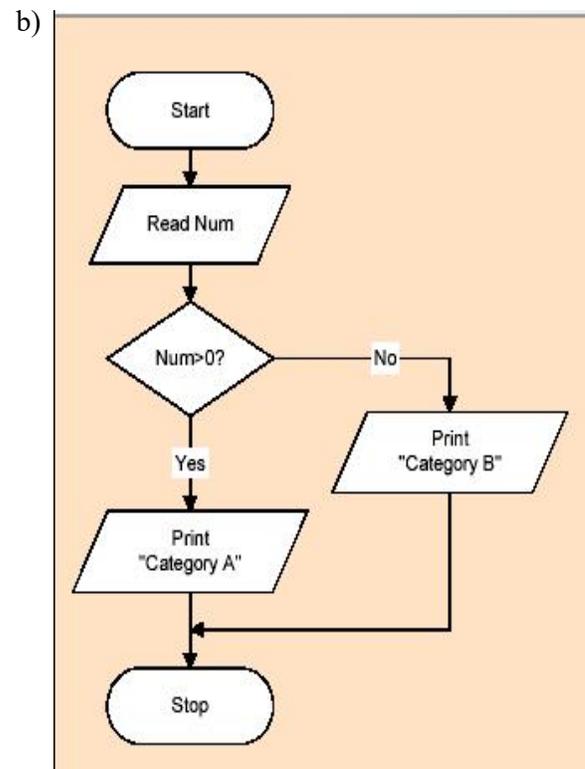
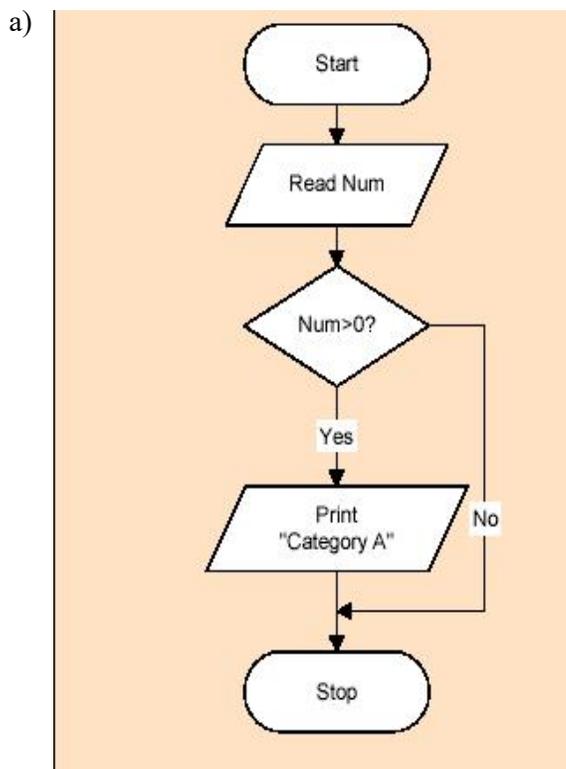
Output : 50 ada dalam range 1-100

Masukkan sembarang bilangan : 105

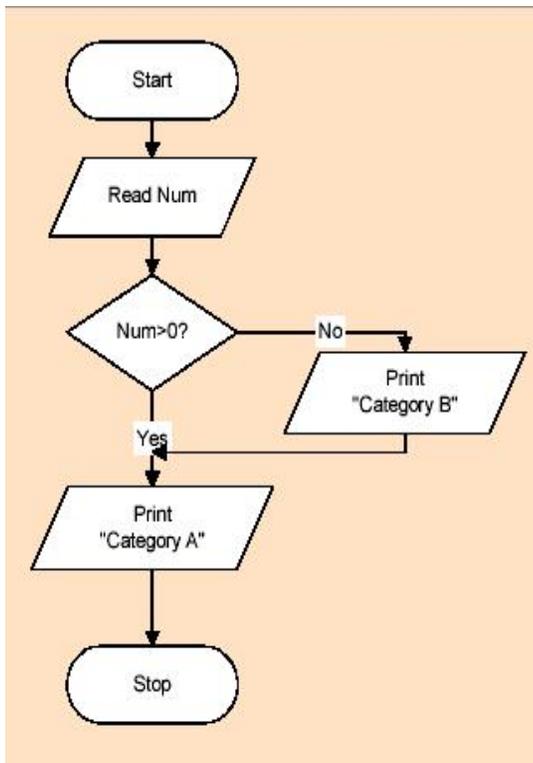
Output : 105 di luar range 1-100

E. LAPORAN RESMI

1. Listing program beserta contoh eksekusinya
2. Berikan ilustrasi tentang perbedaan pernyataan if dan if....else
3. Untuk 3 flow chart di bawah ini lakukan tracing flow chart untuk menemukan outputnya, jika input yang diberikan adalah :
 - (1) Num = 10
 - (2) Num = -10
 - (3) Num = 0



c)



Praktikum 3 (2/3)

PENGAMBILAN KEPUTUSAN

A. TUJUAN

1. Menjelaskan penggunaan pernyataan *if* dalam *if*
2. Menjelaskan penggunaan pernyataan *else-if*

B. DASAR TEORI

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** ang dimasukkan oleh user. Oleh karena itu program akan mengecek apakah **operator** berupa tanda ‘*’, ‘/’, ‘+’, ataukah 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 !”

C. TUGAS PENDAHULUAN

Buatlah desain algoritma dan flow chart untuk setiap soal dalam percobaan.

D. PERCOBAAN

1. Buatlah program untuk menghitung diskriminan dan mencari akar-akar dari persamaan kuadrat :

$ax^2 + bx + c = 0$, dengan ketentuan sbb :

$$D = b^2 - 4ac$$

- Jika $D = 0$, maka terdapat 2 akar real yang kembar, yaitu : $x_1 = x_2 = -b / 2a$
- Jika $D > 0$, maka terdapat 2 akar real yang berlainan, yaitu :

$$x_1 = (-b + \text{sqrt}(D)) / 2a$$

$$x_2 = (-b - \text{sqrt}(D)) / 2a$$

- Jika $D < 0$, maka terdapat 2 akar imaginair yang berlainan, yaitu :

$$x1 = -b / 2a + (\text{sqrt}(-D) / 2a) i$$

$$x2 = -b / 2a - (\text{sqrt}(-D) / 2a) i$$

Input : a, b, c (float)

Output : Nilai Diskriminan serta nilai akar-akar persamaan tsb (x1& x2).

Petunjuk :

- untuk mencari akar dari x, gunakan : $\text{sqrt}(x)$ yang didefinisikan pada `<math.h>`.
- gunakan else if

2. Dengan menggunakan pernyataan else..if , buatlah program kalkulator sederhana, untuk mendapatkan tampilan hasil sebagai berikut :

Masukkan bilangan pertama : 5

Masukkan bilangan kedua : 3

Menu Matematika

1. Penjumlahan

2. Pengurangan

3. Pembagian

4. Perkalian

Masukkan pilihan anda : 4

Hasil operasi tersebut = 15

3. Buat program untuk mengkonversikan nilai angka ke nilai huruf.

Petunjuk : nilai_angka <= 40 = E

40 < nilai_angka <= 55 = D

55 < nilai_angka <= 60 = C

60 < nilai_angka <= 80 = B

80 < nilai_angka <= 100 = A

Input : nilai_angka = 62

Output : Nilai huruf adalah B

E. LAPORAN RESMI

1. Tulis listing dari kedua program diatas.
2. Bila pada pilihan menu diinputkan angka selain 1, 2, 3 dan 4 pada soal no. 2, apa yang terjadi ?
Jelaskan .

Praktikum 3 (3/3)

PENGAMBILAN KEPUTUSAN

A. TUJUAN

1. Menjelaskan penggunaan pernyataan switch

B. DASAR TEORI

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

Buatlah desain algoritma dan flow chart untuk setiap soal dalam percobaan.

D. PERCOBAAN

1. Tulislah kembali pernyataan-pernyataan di bawah ini dengan menggunakan pernyataan *switch*

```

if( letter == 'X' )
    sum = 0;
else if ( letter == 'Z' )
    valid_flag = 1;
else if( letter == 'A' )
    sum = 1;
else
    printf("Unknown letter -->%c\n", letter );

```

2. Ubahlah program di bawah ini, implementasikan kembali dengan menggunakan pernyataan *switch*

```

main()
{
    int  valid_operator = 1;
    //valid_operator diinisialisasi dengan logika 1
    char operator;
    float  number1, number2, result;

    printf("Masukkan 2 buah bilangan & sebuah operator\n");

```

```

printf("dengan format : number1 operator number2\n\n");
scanf("%f %c %f", &number1, &operator, &number2);

if(operator == '*')
    result = number1 * number2;
else if(operator == '/')
    result = number1 / number2;
else if(operator == '+')
    result = number1 + number2;
else if(operator == '-')
    result = number1 - number2;
else
    valid_operator = 0;

if(valid_operator)
    printf("\n%g %c %g is %g\n", number1, operator,
        number2, result );
else
    printf("Invalid operator!\n");
}

```

3. Buatlah program untuk menampilkan menu dan melakukan proses sbb :

Menu : 1. Menghitung volume kubus
 2. Menghitung luas lingkaran
 3. Menghitung volume silinder.

Input : pilihan user (1, 2 atau 3)

Jika pilihan = 1, maka :

Input : panjang sisi kubus

Output : Volume kubus ($vol = sisi^3$)

Jika pilihan = 2, maka :

Input : panjang jari-jari lingkaran

Output : Luas lingkaran ($luas = 3.14 * r^2$)

Jika pilihan = 3, maka :

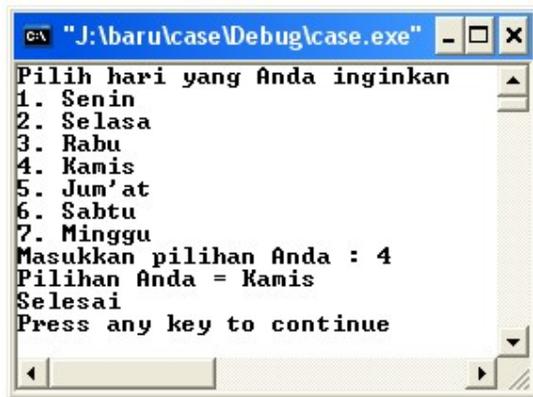
Input : panjang jari-jari lingkaran & tinggi silinder

Output : Volume silinder ($vol = 3.14 * r^2 * t$)

Jika pilihan selain 1, 2 & 3 (default) : Tampilkan pesan kesalahan.

Petunjuk : gunakan switch-case

4. Buatlah program untuk menampilkan pilihan hari : 1 s/d 7 untuk pilihan Senin s/d Minggu. Selanjutnya, minta user utk memasukkan salah satu pilihan 1-7. Tampilkan nama hari yang terpilih (lihat contoh output) Implementasikan dengan menggunakan else-if dan switch case



```
CA "J:\baru\case\Debug\case.exe"
Pilih hari yang Anda inginkan
1. Senin
2. Selasa
3. Rabu
4. Kamis
5. Jum'at
6. Sabtu
7. Minggu
Masukkan pilihan Anda : 4
Pilihan Anda = Kamis
Selesai
Press any key to continue
```

E. LAPORAN RESMI

1. Listing program beserta contoh eksekusinya
2. Buat kesimpulan tentang else-if dan switch-case, apakah ada perbedaan atau persamaan antara dua instruksi diatas ?.