

PRAKTIKUM 6

6.1 JUDUL

PERULANGAN

6.2 TUJUAN

Pada akhir perkuliahan ini mahasiswa akan dapat:

- Dapat menjelaskan maksud dan tujuan perulangan
- Dapat memakai dengan tepat penggunaan perintah perulangan **do**, **do-while** atau **for**

6.3 DASAR TEORI

Perulangan ada dua konsep. Yaitu:

Perulangan terbatas.

Jika dalam program diperlukan perulangan yang **jumlahnya pasti**, atau perulangan dengan **batas yang jelas**, maka dapat digunakan perintah **for** untuk perulangan tersebut. Format **for** adalah sebagai berikut:

```
for (ungkapan1; ungkapan2; ungkapan3)
    pernyataan;
```

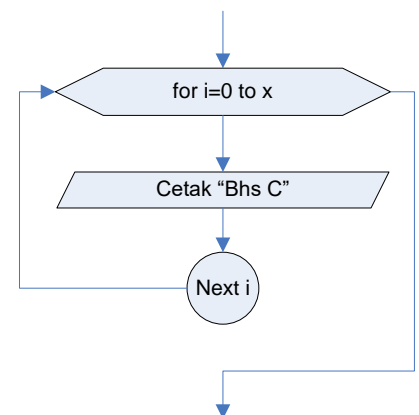
Ungkapan1 → nilai awal

Ungkapan2 → batas

Ungkapan3 → kelipatan

Contoh **for** untuk mencetak “bhs c” 5 kali adalah sbb:

```
for ( i=0 ; i<5 ; i++)
    puts("bhs C");
```



Flowchart for

Perulangan Tidak Pasti.

Perulangan tidak pasti adalah perulangan yang kita tidak tahu kapan akan berakhir. Misalkan perulangan untuk mengulang program yang telah dilakukan. Bisa saja user berkali-kali mencoba, hingga akhirnya menekan tombol ESC, sehingga program akan selesai.

Ada dua perulangan yang tidak pasti, yaitu

A. Dicek Dahulu (while)

Urutannya: kondisi dicek terlebih dahulu, jika kondisi while adalah “true” maka proses akan dijalankan. Jika tidak, proses tidak dijalankan.

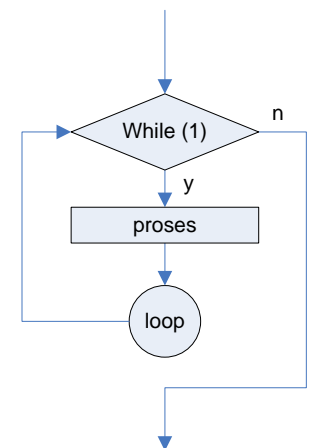
Dengan while, ada kemungkinan proses tidak dijalankan sama sekali.

Proses akan terus dijalankan selama while bernilai true

Contoh while untuk mencetak “Program dijalankan”, jika user menekan y.

Dan perulangan berhenti saat ditekan t.

```
#include<stdio.h>
#include<conio.h>
main()
{
    clrscr();
    char tekan;
    printf("Mau Mengulang y/t ");
    while (getch()=='y')
    {
        puts("Program Dijalankan");
        printf("Mau Mengulang y/t ");
    }
}
```



Flowchart while

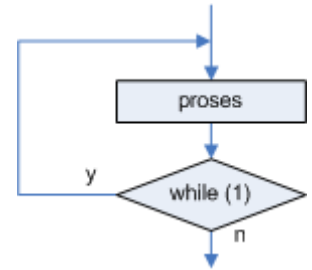
Perhatikan, jika kita menekan t saat pertama, maka tulisan “Program Dijalankan” tidak akan pernah dieksekusi.

B. Dijalankan Dahulu, baru di cek (do – while)

Urutannya: Proses akan dijalankan minimal sekali. Setelah itu dilakukan cek, jika **while** bernilai **true**, maka proses akan diulang, dan terus di ulang hingga while bernilai **false**.

Contoh while untuk mencetak “Program dijalankan”, jika user menekan y.

Dan perulangan berhenti saat ditekan t.



Flowchart do - while

```
#include<stdio.h>
#include<conio.h>
main()
{
    clrscr();
    char tekan;
    do
    {
        puts("Program Dijalankan");
        printf("Mau Mengulang y/t ");
    }
    while (getch()=='y');
}
```

Perhatikan, bedanya dengan while, pada do-while tampilan “Program Dijalankan” akan dijalankan minimal sekali. Baru muncul pertanyaan mengulang/ tidak.

Jika dijawab y, maka akan mengulang, jika t akan berhenti.

6.4 LANGKAH PERCOBAAN

1. Perulangan for

a. Tuliskan listing Program berikut:

```
#include <stdio.h>
#include <conio.h>
main()
{
    clrscr();
    int i;
    for (i=0;i<5;i++)
        puts("Bisa Bahasa C");
}
```


b. Jalankan Program, catat apa yang tampak di layar? (Jika tidak tampak, tekan **ALT+F5**, atau pilih menu windows → user screen). Gunakan cara ini untuk menampilkan hasil pada langkah2x selanjutnya.

- c. Ganti perintah for menjadi `for (i=0;i<=5;i++)`.
- d. Jalankan Program, catat apa yang tampak di layar?
- e. Apa beda hasil b dan d? kenapa demikian.

2. Perulangan for lebih dari 1 baris

- a. Ketikkan listing berikut.

```
#include <stdio.h>
#include <conio.h>
main()
{
    clrscr();
    int i;
    for (i=0;i<5;i++)
    {
        puts("Aku Bisa Bahasa C");
        puts("Kamu Juga");
    }
}
```




- b. Jalankan Program, catat apa yang tampak di layar?
- c. Hilangkan tanda kurung siku buka “{” dan tutup “}” yang ditunjukkan pada panah. Jalankan program, catat apa yang tampak di layar?
- d. Kenapa no c bisa muncul seperti itu?

3. For didalam for

- a. Kadang kita memerlukan perulangan didalam perulangan yang sering disebut for didalam for. Contoh berikut adalah membuat tabel perkalian 1 sd. 5

```
#include <stdio.h>
#include <conio.h>
main()
{
    clrscr();
    int i,j;
    for (i=1;i<6;i++)
    {
        for (j=1;j<6;j++)
            printf("%3i",i*j);
        printf("\n");
    }
}
```

- b. Jalankan Program, catat apa yang tampak di layar?
- c. Hilangkan perintah `printf("\n");` Jalankan Program, catat apa yang tampak di layar? Jadi, untuk apa perintah `printf("\n");`

- d. Kembalikan listing seperti listing 3.a. Ganti perintah `printf("%3i",i*j);` dengan perintah `printf(" * ");` Jalankan, catat hasilnya. Perhatikan ada spasinya di perintah ini. 
- e. Buatlah flowchart for didalam for 3.a.

4. Membuat segitiga

- a. Berikut ini akan membuat animasi angka sehingga membentuk formasi segitiga.

```
#include <stdio.h>
#include <conio.h>
main()
{
    clrscr();
    int i,j;
    for (i=0;i<6;i++)
    {
        for (j=0;j<i;j++)
            printf("%3i",i);
        printf("\n");
    }
}
```

- b. Jalankan Program, catat apa yang tampak di layar?
- c. Ganti perintah `printf("%3i",i);` dengan perintah `printf("%3i",j);`
- d. Jalankan Program, catat apa yang tampak di layar?

5. Perulangan dengan while

- a. Berikutnya akan dicoba menggunakan while. Setelah dijalankan, Program akan diam, menunggu penekan tombol. Jika kita menekan huruf "a" (a kecil, bukan A besar), maka akan menampilkan "aku bisa bahasa C". dan akan terus menampilkan jika kita menekan huruf a. Dan akan berhenti jika kita menekan selain itu.

```
#include <stdio.h>
#include <conio.h>
main()
{
    clrscr();
    int i;
    while (getch()=='a')
        printf("Aku Bisa Bahasa C\n");
}
```

- b. Jalankan Program, tekan huruf a beberapa kali. Selanjutnya tekan sembarang tombol. catat apa yang tampak di layar?

- c. Jalankan Program, langsung tekan selain huruf a. Catat apa yang tampak di layar? Apa beda no b dan c.
- d. Kenapa setelah perintah `while ()` tidak pakai `{ }` ?
- e. Ketikkan dan coba program berikut.

```
#include <stdio.h>
#include <conio.h>
main()
{
    clrscr();
    char tombol;
    puts("Tekan sembarang tombol, akhiri dengan tombol x");
    while((tombol=getch())!='x')
    {
        printf("\nAnda Menekan %c ",tombol);
    }
}
```

Jalankan program, tekan beberapa tombol selain x. Catat apa yang tampak di layar. Akhiri dengan menekan x.

- f. Jelaskan maksud $\rightarrow ((tombol=getch())!='x')$

6. Perulangan dengan do - while

- a. Selanjutnya akan dicoba menggunakan do-while dengan kasus yang sama seperti while di atas (percobaan 5a). Program akan menampilkan **terlebih dahulu** “aku bisa bahasa C”, dan jika kita menekan huruf “a” (a kecil, bukan A besar), maka akan muncul lagi “aku bisa bahasa C” terus dan terus selama kita menekan tombol ‘a’ kecil.

```
#include <stdio.h>
#include <conio.h>
main()
{
    clrscr();
    do
    {
        printf("Aku Bisa Bahasa C\n");
    }while (getch()=='a');
}
```

- b. Jalankan Program, tekan huruf a beberapa kali. Selanjutnya tekan sembarang tombol selain huruf a. catat apa yang tampak di layar?
- c. Bisakah anda menekan tombol karakter tertentu agar saat pertama dijalankan program langsung selesai?

d. Apakah bisa perintah do tanpa tanda { } ?

7. TERNARY

1. Kembali ke perintah **IF-ELSE**.

Dalam bahasa C kita dapat menyederhanakan perintah if-else hanya dalam 1 (satu) baris yang disebut operator ternary.

Formatnya: Kondisi ? true : false ;

2. Berikut ini adalah coding untuk if-else dalam format ternary untuk menentukan manakah antara dua bilangan yang nilainya lebih besar.

```
#include<stdio.h>
#include<conio.h>
main()
{
    clrscr();
    float nilai1, nilai2, mak;

    printf("Masukkan dua buah nilai = ");
    scanf("%f %f",&nilai1, &nilai2);

    mak = (nilai1>nilai2) ? nilai1 : nilai2;

    printf("Nilai terbesar adalah =%f\n",mak);
    getch();
}
```

3. Jalankan, dengan cara memasukkan dua bilangan dengan dipisahkan spasi atau enter. Dan lihat tampilan di layar. Anda juga bisa memasukkan nilai pecahan dengan pemisah titik (pengganti koma).
4. Cobalah beberapa kali. Catat hasilnya.

8. Latihan

1. Buatlah program untuk menampilkan bilangan ganjil atau genap dengan batas atas dan bawah yang sudah ditentukan.

Jadi, inputnya adalah:

- ganjil atau genap
- nilai awal loop
- nilai akhir

Proses harus dapat:

- Menampilkan bilangan ganjil atau genap sesuai input dan batas
- Jika diminta ganjil, tapi nilai awal genap, mulai dari setelah nilai awal. Tapi jika diminta ganjil dan nilai awal ganjil, normal.
- Jika diminta genap, tapi nilai awal ganjil, tampilan dimulai dari nilai setelah nilai ganjil. Tapi jika diminta genap dan nilai awal genap, samakan.
-

Contoh tampilan:

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: BC
Program Menampilkan bilangan ganjil atau genap
Masukkan 1 (ganjil) atau 2(genap) = 1
Masukkan awal = 11
Masukkan akhir = 23
Bilangan =
11 12 13 14 15 16 17 18 19 20 21 22 23
```

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: BC
Program Menampilkan bilangan ganjil atau genap
Masukkan 1 (ganjil) atau 2(genap) = 1
Masukkan awal = 2
Masukkan akhir = 14
Bilangan =
3 4 5 6 7 8 9 10 11 12 13 14
```

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: BC
Program Menampilkan bilangan ganjil atau genap
Masukkan 1 (ganjil) atau 2(genap) = 2
Masukkan awal = 111
Masukkan akhir = 124
Bilangan =
112 113 114 115 116 117 118 119 120 121 122 123 124 _
```


6.5 DATA

<Isikan data langkah percobaan di lembar kerja praktikum>

6.6 Analisa Data

<Buatlah analisa berdasar data yang diperoleh>

6.7 Kesimpulan

<Ambil kesimpulan dari hasil praktikum, data dan analisa data anda>

6.8 Soal

1. Perhatikan tampilan berikut ini:

```
Program Menghitung total dan rata2
Dibuat oleh : Agus SBN, Nim : 1234

Masukkan banyaknya data = 5
Input data ke 1 = 80.5
Input data ke 2 = 75.0
Input data ke 3 = 90.3
Input data ke 4 = 65.8
Input data ke 5 = 78.9

Banyaknya data =      5
Total nilai    =    390.50
Rata-rata      =    78.10
```

Tampilkan judul.

Tampilkan nama dan nim sesuai dengan nama dan nim anda

Banyaknya data bebas. Input data juga bebas.

Tampilan rapi

- Buatlah flowchart nya. Tulis tangan.
- Hitung manual menggunakan kalkulator mulai banyaknya data hingga data terakhir. Tulis tangan. Ingat, jangan ada contoh hitungan yang sama dengan teman anda.
- Buat codingnya (di cetak)
- Jalankan. Capture hasilnya (di cetak)
- Apakah hasil d sudah sesuai dengan no b. (jika sama, program anda sudah berjalan dengan baik)