

JOB SHEET 06

Flow Control: Loop (Perulangan)

*"Everybody should learn to program a computer,
because it teaches you how to think."*

TUJUAN PEMBELAJARAN

1. Mampu menjelaskan dan mengimplementasikan *Looping: Perulangan* dalam pemrograman menggunakan IDE.

POKOK MATERI

1. Perulangan For
2. Perulangan While
3. Perulangan Do While
4. Perulangan Bersarang (Nested Loop)

URAIAN MATERI

A. Perulangan

Pembahasan dalam bab ini kita akan membahas mengenai putaran atau perulangan. Loop adalah konstruksi aliran kontrol yang memungkinkan sepotong kode untuk dieksekusi secara berulang kali hingga beberapa kondisi terpenuhi. Loop ini memberikan kemudahan dalam pemrograman Anda, karena memungkinkan Anda melakukan banyak hal secara berulang tapi dengan program yang ringkas.

Misalnya, Anda ingin mencetak semua angka antara 1 hingga 10. Tanpa loop, mungkin Anda akan membuat program dengan banyak cout sampai 10 baris program bahkan lebih.

Misalnya lagi untuk menampilkan "Tutorial Pemrograman C!" sebanyak 10x ke layar! Mungkin kamu bisa menuliskannya dengan fungsi printf() sebanyak 10 kali seperti ini:

```
#include <stdio.h>

void main(){
    printf("Tutorial Pemrograman C!\n");
    printf("Tutorial Pemrograman C!\n");
    printf("Tutorial Pemrograman C!\n");
    printf("Tutorial Pemrograman C!\n");
    printf("Tutorial Pemrograman C!\n");
    printf("Tutorial Pemrograman C!\n");
    printf("Tutorial Pemrograman C!\n");
    printf("Tutorial Pemrograman C!\n");
    printf("Tutorial Pemrograman C!\n");
    printf("Tutorial Pemrograman C!\n");
}
```

Bagaimana kalau nanti mau menampilkan sebanyak 1000 kali?

Maka dari itu perlu penggunaan perulangan dalam program untuk efisiensi penulisan.

Perulangan akan membantu kita mengeksekusi kode yang berulang-ulang, berapapun yang kita mau. Ada 4 macam bentuk perulangan pada C. Secara umum, dibagi menjadi dua kelompok. Yaitu: counted loop dan uncounted loop. Perbedaanya: Counted Loop merupakan perulangan yang jelas dan sudah tentu banyak kali perulangannya. Sedangkan Uncounted Loop, merupakan perulangan yang tidak jelas berapa kali ia harus mengulang.



Perulangan yang termasuk dalam Counted Loop:

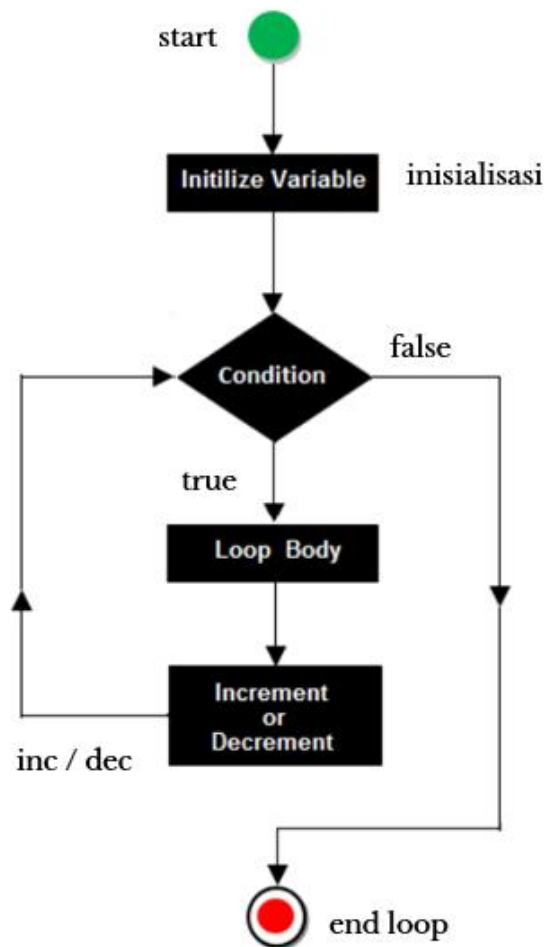
1. Perulangan For

Perulangan yang termasuk dalam Uncounted Loop:

1. Perulangan While
2. Perulangan Do/While

1. Perulangan For pada C

Perulangan for merupakan perulangan yang termasuk dalam counted loop, karena sudah jelas berapa kali ia akan mengulang.



```
for(int i = 0; i < 10; i++){  
    printf("Perulangan ke-%i\n", i);  
}
```

Yang perlu diperhatikan adalah kondisi yang ada di dalam kurung setelah kata for. Kondisi ini akan menentukan:

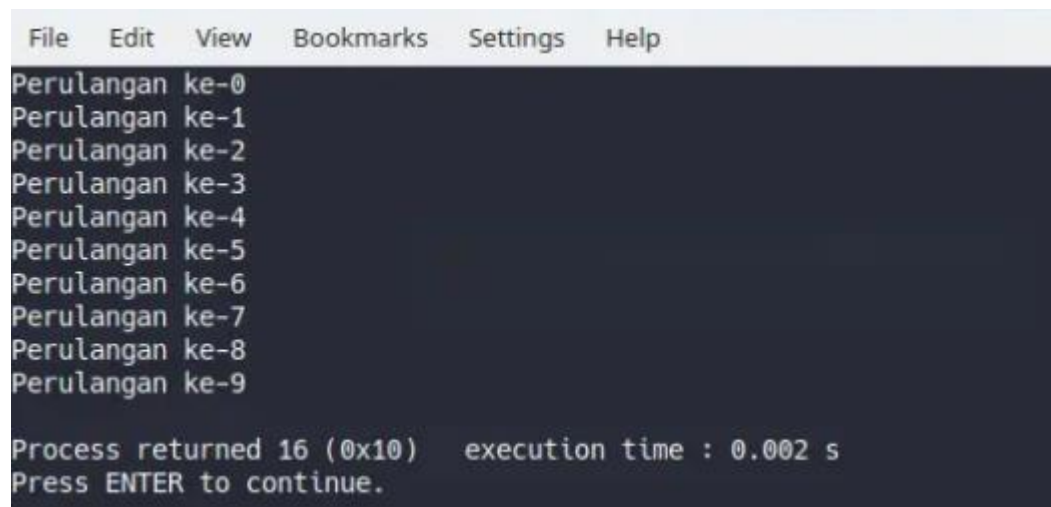
- Hitungan akan dimulai dari 0 ($i = 0$);
- Hitungannya sampai berapa? Sampai $i < 10$;
- Lalu di setiap perulangan i akan bertambah +1 ($i++$).

Variabel i pada perulangan for berfungsi untuk menyimpan nilai hitungan.

Jadi setiap perulangan dilakukan nilai *i* akan selalu bertambah satu. Karena kita menentukannya di bagian *i++*.

```
#include <stdio.h>

void main(){
    for(int i = 0; i < 10; i++){
        printf("Perulangan ke-%i\n", i);
    }
}
```



```
File Edit View Bookmarks Settings Help
Perulangan ke-0
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
Perulangan ke-5
Perulangan ke-6
Perulangan ke-7
Perulangan ke-8
Perulangan ke-9

Process returned 16 (0x10)   execution time : 0.002 s
Press ENTER to continue.
```

Apakah nama variabelnya harus selalu *i*? Tidak.

Kita juga bisa menggunakan nama lain.

```
#include <stdio.h>

void main(){
    for(int counter = 0; counter < 50; counter+=2){
        printf("Perulangan ke-%i\n", counter);
    }
}
```

Pada contoh tersebut, kita melakukan perulangan dimulai dari nol 0. Lalu di setiap perulangan nilai variabel *counter* akan ditambah 2 (*counter+2*).

```

File Edit View Bookmarks Settings Help
Perulangan ke-32
Perulangan ke-34
Perulangan ke-36
Perulangan ke-38
Perulangan ke-40
Perulangan ke-42
Perulangan ke-44
Perulangan ke-46
Perulangan ke-48

Process returned 17 (0x11)   execution time : 0.003 s
Press ENTER to continue.

```

Bagaimana kalau counter perulangannya dimulai dari angka yang lebih besar sampai yang ke paling kecil? Ini biasanya kita buat saat ingin menghitung mundur.

Caranya gampang.

- Kita tinggal isi nilai counter dengan nilai terbesarnya.
- Misalnya kita akan mulai hitungan dari 10 sampai ke 0.
- Maka nilai counter, kita isi awalnya dengan 10.
- Lalu di kondisi perbandingannya, kita berikan `counter > 0`. Artinya perulangan akan dilakukan selama nilai counter lebih besar dari 0.
- Lalu kita kurangi (-1) nilai counter di setiap perulangan (`counter--`).

```

for(int counter = 10; counter > 0; counter--){
    printf("Perulangan ke-%i\n", counter);
}

```

```

File Edit View Bookmarks Settings Help
Perulangan ke-10
Perulangan ke-9
Perulangan ke-8
Perulangan ke-7
Perulangan ke-6
Perulangan ke-5
Perulangan ke-4
Perulangan ke-3
Perulangan ke-2
Perulangan ke-1

Process returned 16 (0x10)   execution time : 0.002 s
Press ENTER to continue.

```

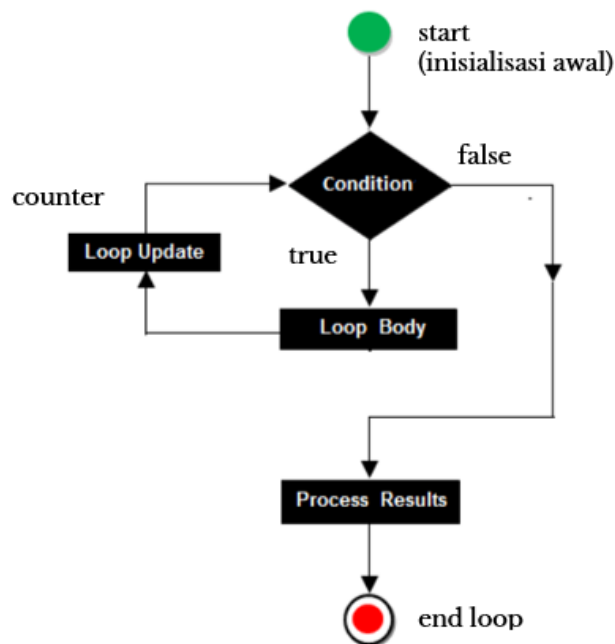
Mengapa tidak sampai nol (0)?

Karena kondisi yang kita berikan $\text{counter} > 0$. Apabila counter bernilai 0, maka kondisi ini akan menjadi false. Kecuali kita menggunakan operator lebih besar sama dengan (\geq), maka jika counter bernilai 0, kondisi akan menjadi true.

2. Perulangan While pada C

Perulangan while merupakan perulangan yang termasuk dalam perulangan uncounted loop. Perulangan while juga dapat menjadi perulangan yang counted loop dengan memberikan counter di dalamnya.

Bentuk flowchart-nya sama seperti flowchart for:



```
#include <stdio.h>

void main(){
    char ulang = 'y';
    int counter = 0;

    // perulangan while
    while(ulang == 'y'){
        printf("Apakah kamu mau mengulang?\n");
        printf("Jawab (y/t): ");
        scanf(" %c", &ulang);

        // increment counter
        counter++;
    }

    printf("\n\n-----\n");
    printf("Perulangan Selesai!\n");
    printf("Kamu mengulang sebanyak %i kali.\n", counter);
}
```

```
Jawab (y/t): y
Apakah kamu mau mengulang?
Jawab (y/t): y
Apakah kamu mau mengulang?
Jawab (y/t): y
Apakah kamu mau mengulang?
Jawab (y/t): t

-----
Perulangan Selesai!
Kamu mengulang sebanyak 4 kali.
```

Coba perhatikan blok While:

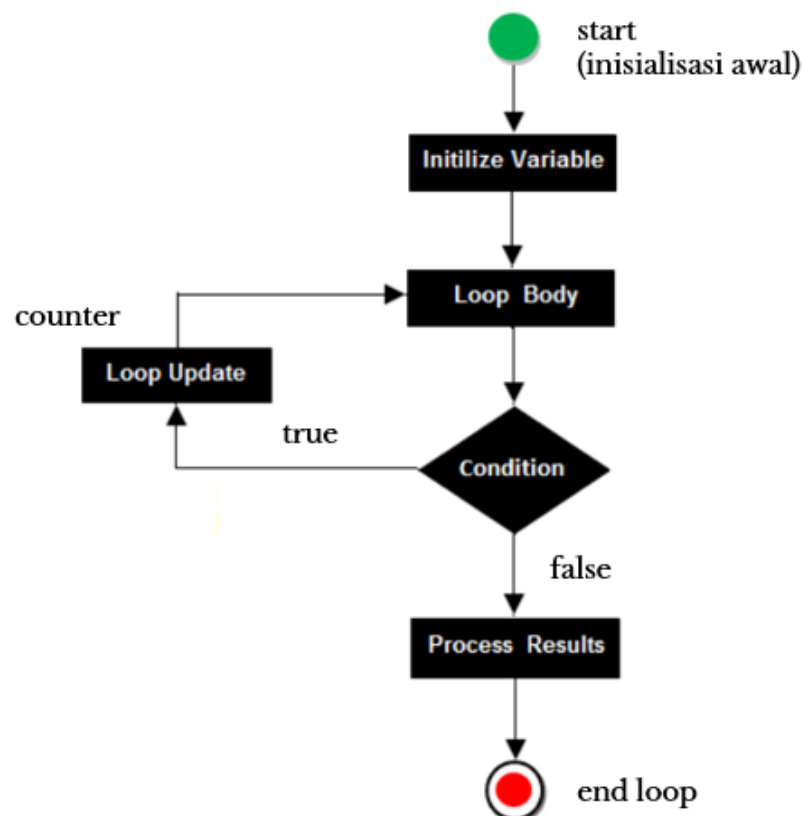
```
// perulangan while
while(ulang == 'y'){
    printf("Apakah kamu mau mengulang?\n");
    printf("Jawab (y/t): ");
    scanf(" %c", &ulang);
    // increment counter
    counter++;
}
```


Perulangan akan terjadi selama variabel ulangi bernilai y. Lalu kita menggunakan fungsi scanf() untuk mengambil input. Selama kita menjawab y pada input, maka perulangan akan terus dilakukan. Tapi kalau kita jawab yang lain, maka perulangan akan dihentikan. Karena kondisi perulangannya tidak terpenuhi.

3. Perulangan Do While pada C

Perulangan do while sama seperti perulangan while. Perbedaannya: Perulangan do while akan melakukan perulangan sebanyak 1 kali terlebih dahulu, lalu mengecek kondisi yang ada di dalam kurung while.

Flowchart perulangan do while:



```
do {
    // blok kode yang akan diulang
} while (<kondisi>);
```

Jadi perbedaanya: Perulangan do while akan mengecek kondisi di belakang (sesudah mengulang), sedangkan while akan mengecek kondisi di depan atau awal (sebelum mengulang).

```
#include <stdio.h>

void main(){

    char ulang = 'y';
    int counter = 0;

    do {
        printf("Apakah kamu mau mengulang?\n");
        printf("Jawab (y/t): ");
        scanf(" %c", &ulang);

        // increment counter
        counter++;
    } while(ulang == 'y');

    printf("\n\n-----\n");
    printf("Perulangan Selesai!\n");
    printf("Kamu mengulang sebanyak %i kali.\n", counter);
}
```

Contoh tersebut sama seperti contoh pada perulangan while. Saat perulangan pertama, cobalah untuk membatalkan perulangannya dengan menjawab t. maka hasilnya:

```
Apakah kamu mau mengulang?
Jawab (y/t): t

-----
Perulangan Selesai!
Kamu mengulang sebanyak 1 kali.

Process returned 0 (0x0)   execution time : 2.506 s
Press ENTER to continue.
█
```

4. Perulangan Bersarang (Nested Loop)

Di dalam blok perulangan, kita juga dapat membuat perulangan. Ini disebut dengan nested loop atau perulangan bersarang atau perulangan di dalam perulangan.

```
#include <stdio.h>

void main(){
    for(int i = 0; i < 10; i++){
        for(int j = 0; j < 10; j++){
            printf("Perulangan ke (%d, %d)\n", i, j);
        }
    }
}
```

```
Perulangan ke (9, 1)
Perulangan ke (9, 2)
Perulangan ke (9, 3)
Perulangan ke (9, 4)
Perulangan ke (9, 5)
Perulangan ke (9, 6)
Perulangan ke (9, 7)
Perulangan ke (9, 8)
Perulangan ke (9, 9)

Process returned 21 (0x15)   execution time : 0.003 s
Press ENTER to continue.
```

Pada perulangan tersebut, kita menggunakan dua perulangan for.

Perulangan pertama menggunakan variabel i sebagai counter, sedangkan perulangan kedua menggunakan variabel j sebagai counter.

Untuk modul tambahan, pelajari:

- Perulangan:

<https://www.youtube.com/watch?v=HhrhtYaUrz8&t=545s>

<https://www.youtube.com/watch?v=X7jAPOHx71A>

LATIHAN

1. Buatlah sebuah program untuk menghasilkan deret bilangan genap dan ganjil dari 0 – 50.
2. Buatlah sebuah program untuk menghasilkan program dengan keluaran sebagai berikut:

```
*  
**  
***  
****  
*****
```

3. Buatlah sebuah program untuk menampilkan keluaran sebagai berikut:

```
1  
2 4  
3 6 9  
4 8 12 16  
5 10 15 20 25
```

4. Buatlah program ATM dimana Hattori dapat melakukan setoran dan penarikan saldo. Menu yang ada pada ATM:

ATM

- 1) Cek Saldo
- 2) Setoran
- 3) Penarikan Tunai
- 4) Exit

Setiap terdapat transaksi setoran ataupun penarikan maka nilai total saldo akan berubah (bisa dicek melalui menu cek saldo). Kemudian kebijakan

bank menyatakan saldo minimal yang harus disisakan dalam saldo adalah Rp. 50.000. jika saldo kurang dari atau sama dengan Rp. 50.000 maka tidak bisa dilakukan penarikan.

Kasus:

No Rek : 0123
Nama Akun : Hattori
Saldo ATM : Rp. 175.000

Skenario:

- a) Hattori menyetorkan uang sebesar Rp. 25.000, kemudian setoran kedua sebesar Rp. 45.000. Berapakah saldo Hattori saat ini?
- b) Hattori melakukan penarikan pertama sebesar Rp. 50.000, kemudian penarikan kedua sebesar Rp. 45.000. Berapakah sisa saldo Hattori setelah itu?
- c) Hattori melakukan penarikan ketiga sebesar Rp. 120.000. Berapakah sisa saldo Hattori?

Berikan jawaban dengan ilustrasi dan bukti screenshot testing program yang anda buat!

REFERENCES

1. The C Programming Language. 2nd Edition
2. [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language))
3. https://en.wikipedia.org/wiki/Imperative_programming
4. <https://www.petanikode.com/tutorial/c/>
5. <https://www.learn-c.org/>
6. <https://www.tutorialspoint.com/cprogramming/index.htm>
7. <https://www.programiz.com/>
8. <https://www.dicoding.com/>
9. <https://data-flair.training/blogs/c-tutorials-home/>