

MODUL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN I

PERULANGAN  
FOR



TIM ASISTEN PEMROGRAMAN  
ANGKATAN 11  
ILMU KOMPUTER FPMIPA UPI

## STRUKTUR PERULANGAN FOR

Struktur perulangan (atau dalam bahasa inggris disebut dengan loop) adalah instruksi kode program yang bertujuan untuk mengulang beberapa baris perintah. For ini berguna untuk **mempersingkat codingan perintah yang sama**.

```
printf("error...\n");  
printf("error...\n");  
printf("error...\n");  
printf("error...\n");  
printf("error...\n");  
printf("error...\n");  
printf("error...\n");  
printf("error...\n");  
printf("error...\n");
```

```
int i;  
for(i = 0; i < 10; i++) {  
    printf("error...\n");  
}
```

Output:

```
error...  
error...  
error...  
error...  
error...  
error...  
error...  
error...  
error...  
error...
```

Dapat dilihat bahwa struktur perulangan for bisa mengulang beberapa baris perintah.

## FORMAT DASAR PERULANGAN FOR

```
for (inisialisasi; kondisi; iterasi) {  
    ...  
    proses  
    ...  
}
```

### 1. Inisialisasi

Inisialisasi adalah **kondisi pada saat awal perulangan**. Biasanya kondisi awal ini berisi perintah untuk memberikan nilai kepada **variabel counter** (biasanya menggunakan variabel *i*).

Contoh:

```
for(i = 0; /*kondisi*/; /*iterasi*/) {  
    /*..  
    proses  
    ..*/  
}
```

```
int i, n;  
n = 31;  
for(i = n; /*kondisi*/; /*iterasi*/) {  
    /*..  
    proses  
    ..*/  
}
```

### 2. Kondisi

Bagian dimana menjadi **kondisi yang harus dipenuhi agar perulangan berjalan**. Selama kondisi ini terpenuhi, maka compiler akan terus melakukan perulangan.

Contoh:

```
for(/*inisialisasi*/; i <= 5; /*iterasi*/) {
    /*..
    proses
    ..*/
}
```

```
int n = 8;
for(/*inisialisasi*/; i > n; /*iterasi*/) {
    /*..
    proses
    ..*/
}
```

```
x < y
x > y
x <= y
x >= y
```

### 3. Iterasi

Bagian yang dipakai untuk **memproses variabel counter** agar bisa memenuhi kondisi akhir perulangan. Bagian ini akan selalu di eksekusi di setiap perulangan.

```
for(i = 0; i < 1000; i++) {
    /*..
    proses
    ..*/
}
```

a. Menaikan nilai  
(increment)

```
//menaikan nilai 1
i++
i = i + 1
//menaikan nilai 2
i = i + 2
i+= 2
//menaikan nilai 3
i+= 3
```

b. Menurunkan nilai  
(decrement)

```
//menurunkan nilai 1
i--
i = i - 1
//menurunkan nilai 2
i = i - 2
i-= 2
//menurunkan nilai 3
i-= 3
```

#### 4. Proses

Tahap proses adalah bagian yang terjadi di dalam bagian perulangan di mana berisi semua **proses yang perlu dilakukan secara berulang ulang**.

### CONTOH PENGGUNAAN PERULANGAN FOR

#### 1. Program menampilkan angka yang berurutan

```
int n = 3;
for(i = 0; i < n; i++) {
    printf("no. urut = %d\n", i+1);
}
```

Output

```
no. urut = 1
no. urut = 2
no. urut = 3
```

#### 2. Program faktorial

```
#include <stdio.h>

int main() {
    int i, j; //iterator
    int n; //jumlah perulangan yang diinginkan
    scanf("%d", &n); //meminta user mengisi variabel n

    int hasil_faktorial = 1;
    for(i = n; i > 0; i--) { //perulangan sebanyak n
        hasil_faktorial*=i;
        //hasil_faktorial = hasil_faktorial*i;
    }

    printf("%d\n", hasil_faktorial);
    return 0;
}
```

Output

```
4
24
```

3. Program menampilkan perkalian sampai 10 dari angka masukan

```
int num;
scanf("%d", &num);
printf("\n");

int n = 10;

for(i = 1; i <= n; i++) {
    printf("%d\n", num*i);
}
```

Output

```
3
3
6
9
12
15
18
21
24
27
30
```

#### 4. Program membuat pola

```
#include <stdio.h>

int main() {
    int i; //iterator
    int n; //jumlah perulangan yang diinginkan
    scanf("%d", &n); //meminta user mengisi variabel n

    for(i = 0; i < n; i++) { //perulangan sebanyak n
        printf("*");
    }
    printf("\n");

    return 0;
}
```

a.

```
5
*****
```

```
for(i = 0; i < n; i++) { //perulangan sebanyak n
    printf("*\n");
}
```

b.

```
5
*
*
*
*
*
*
```

## FORMAT FOR DALAM FOR (FOR BERSARANG)

```
#include <stdio.h>

int main() {
    int i, j; //iterator
    int n; //jumlah perulangan yang diinginkan
    scanf("%d", &n); //meminta user mengisi variabel n

    for(i = 0; i < n; i++) { //perulangan sebanyak n
        for(j = 0; j < n; j++) {
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

Output

5

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*



Pola lain:

```
#include <stdio.h>

int main() {
    int i, j; //iterator
    int n; //jumlah perulangan yang diinginkan
    scanf("%d", &n); //meminta user mengisi variabel n

    for(i = 0; i < n; i++) { //perulangan sebanyak n
        for(j = 0; j <= i; j++) {
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
5
*
**
***
****
*****
*****
```

## LATIHAN

Buatlah program untuk menampilkan output berikut!

1.

```
6
*****
*****
*****
***
**
*
```

2.

```
6
      *****
     *****
    *****
   *****
  *****
 *****
*****
```

3.

```
6
      *****
     *****
    *****
   *****
  *****
 *****
*****
 *****
      *****
     *****
    *****
```

## AKHIR KATA

Penulis mengucapkan terimakasih yang sebesar-besarnya kepada semua elemen yang telah mendukung berjalannya sesi praktikum pada mata kuliah Algoritma dan Pemrograman I kali ini. Semoga apa yang kita dapatkan kali ini bisa menjadi berkah bagi kita semua.

## DAFTAR PUSTAKA

Tim Asisten Pemrograman Algoritma dan Pemrograman 1 Angkatan 10. (2019). Switch dan Tipe Data Terstruktur. Modul Praktikum Algoritma dan Pemrograman 1. Bandung, Jawa Barat, Indonesia

<https://www.duniailkom.com/tutorial-belajar-c-perulangan-for-bahasa-c/>