

# STRUKTUR PERULANGAN FOR

## 6.1 TUJUAN PRAKTIKUM

### Tujuan Umum

Mahasiswa dapat memahami:

1. Bentuk umum struktur perulangan FOR.
2. Penggunaan struktur perulangan FOR pada sebuah program.

### Tujuan Khusus

Mahasiswa dapat :

1. Menuliskan program yang menggunakan struktur perulangan FOR.
2. Menulis program untuk alur yang dinyatakan dalam bentuk Flowchart atau pseudocode.

## 6.2 TEORI SINGKAT

Struktur Perulangan digunakan untuk menyelesaikan persoalan yang melibatkan suatu proses yang dikerjakan beberapa kali sesuai pola tertentu. Dengan kata lain, melalui struktur perulangan memungkinkan pemrogram untuk menjalankan satu atau beberapa perintah yang ada di dalam blok perulangan secara berulang sesuai dengan nilai yang ditentukan atau sampai mencapai sebuah batas tertentu.

Sebagai contoh, jika diminta membuat program untuk menginput 3 buah nilai dan mencetak total dari ketiga buah nilai tersebut, tentunya dengan mudah kita cukup mendeklarasikan 3 buah variabel untuk menampung masing-masing nilai yang diinput untuk selanjutnya dihitung totalnya. Bagaimana jika yang diminta adalah 100 atau 1000 buah nilai? Apakah kita harus mendeklarasikan variabel dan menuliskan 100 atau 1000 perintah untuk menginput nilai tersebut? Proses tersebut dapat dilakukan dengan lebih mudah menggunakan perulangan.

### 6.2.1 Struktur Perulangan FOR

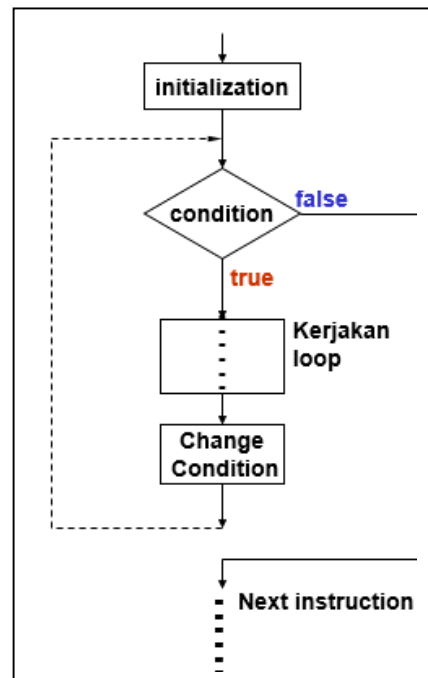
Struktur perulangan FOR biasanya digunakan untuk mengulang suatu proses yang telah diketahui jumlah perulangannya. Dari segi penulisannya, struktur perulangan FOR memiliki bentuk yang sederhana.

Bentuk umum struktur perulangan FOR adalah :

```
for (inisialisasi; kondisi; perubahan-kondisi) {  
    blok-perintah-for;  
}
```

### Catatan:

- **inisialisasi:** Instruksi pemberian suatu nilai yang mempengaruhi nilai kondisi. Pada proses yang normal, pemberian nilai awal ini akan menyebabkan kondisi bernilai TRUE. Instruksi ini hanya pernah satu kali dilaksanakan, yaitu hanya pada saat awal struktur FOR dijalankan.
- **kondisi:** Suatu kondisi yang bernilai TRUE atau FALSE, dan akan membatasi proses perulangan. Blok perintah pada struktur perulangan akan dijalankan selama **kondisi** masih bernilai TRUE.
- **perubahan-kondisi:** Suatu instruksi yang dapat mempengaruhi nilai kondisi. Pada proses yang normal, perubahan nilai disini suatu saat akan membuat **kondisi** bernilai FALSE.



## 6.3 PELAKSANAAN PRAKTIKUM

1. Tuliskan Program 6.1 berikut ini pada editor Dev-C++ (program ini merupakan program untuk mencari nilai terbesar dari 3 buah bilangan yang diinput).

### Program 6.1 terbesar3bil.cpp

```
1  #include <stdio.h>
2  main()
3  {
4      int A, max;
5      printf("Input 3 buah bilangan\n");
6      printf("Bilangan 1: "); scanf("%i", &A);
7      max = A;
8      printf("Bilangan 2: "); scanf("%i", &A);
9      if (A > max) {
10         max = A;
11     }
12     printf("Bilangan 3: "); scanf("%i", &A);
13     if (A > max) {
14         max = A;
15     }
16
17     printf("\nBilangan terbesar: %i", max);
18
19     return 0;
20 }
```

2. Program 6.1 di atas digunakan untuk mencari nilai terbesar dari 3 buah bilangan yang diinput. Bagaimana jika bilangan yang diinput ada 5? Kita cukup menduplikasi baris 8 sampai 11 atau 12 sampe 15, sedemikian hingga akan menginput 5 buah bilangan. Ubahlah program 6.1 menjadi Program 6.2 berikut ini.

**Program 6.2 terbesar5bil.cpp**

```

1  #include <stdio.h>
2  main()
3  {
4      int A, max;
5      printf("Input 5 buah bilangan\n");
6      printf("Bilangan 1: "); scanf("%i", &A);
7      max = A;
8      printf("Bilangan 2: "); scanf("%i", &A);
9      if (A > max) {
10         max = A;
11     }
12     printf("Bilangan 3: "); scanf("%i", &A);
13     if (A > max) {
14         max = A;
15     }
16     printf("Bilangan 4: "); scanf("%i", &A);
17     if (A > max) {
18         max = A;
19     }
20     printf("Bilangan 5: "); scanf("%i", &A);
21     if (A > max) {
22         max = A;
23     }
24
25     printf("\nBilangan terbesar: %i", max);
26
27     return 0;
28 }

```

3. Jalankan dan ujilah program 6.2 di atas dengan beberapa data. Tuliskan pada tabel di bawah ini.

Bil1	Bil2	Bil3	Bil4	Bil5	KELUARAN

4. Dengan tujuan yang sama (mencari nilai terbesar), bagaimana jika yang diinput 10 bilangan, 100 bilangan atau lebih? Dengan cara sebelumnya tentunya akan sangat repot, dan program menjadi sangat panjang. Perhatikan Program 6.3 berikut ini, tuliskan pada editor dan jalankan!

**Program 6.3 terbesar\_n\_bil.cpp**

```

1  #include <stdio.h>
2  main()
3  {
4      int A, max, i;
5      printf("Input 10 buah bilangan\n");
6      printf("Bilangan pertama: ");
7      scanf("%i", &A);
8      max = A;
9      for(i=1; i<=9; i++) {
10         printf("Bilangan ke-%i: ", (i+1));
11         scanf("%i", &A);
12         if (A > max) {
13             max = A;
14         }
15     }
16
17     printf("\nBilangan terbesar: %i", max);
18
19     return 0;
20 }

```

5. Ujicoba program 6.3 dengan data masukan berikut ini dan tuliskan perubahan nilai masing-masing variabel / kondisi pada tabel.

Data masukan: 10 7 15 12 9 5 7 18 3 11

i	i<=9	Input A	A > max	max	Keterangan
-	-	10	-	10	Inisialisasi
1	TRUE	7	FALSE	10	Perulangan ke-1
2	TRUE	15	TRUE	15	Perulangan ke-2
3	TRUE				
4					
5					
6					
7					
8					
9					
10	FALSE				Keluar dari perulangan

6. Berdasarkan Program 6.3 di atas, jawablah beberapa pertanyaan berikut ini!

NO	PERTANYAAN	JAWABAN
1	Pada struktur FOR, tuliskan perintah bagian <b>inisialisasi</b> !	
2	Pada struktur FOR, tuliskan perintah yang menunjukkan <b>kondisi</b> akhir perulangan!	
3	Berapa kali perintah-perintah dalam blok FOR dijalankan?	
4	Jika diinginkan bilangan yang diinput menjadi 20 bilangan, bagian perintah mana yang harus diganti?	
5	Berdasarkan program 6.3 dan data inputan pada langkah ke-5, berapa nilai variabel i setelah keluar dari perulangan?	

7. Tuliskan dan jalankan program 6.4 berikut ini! Lalu tuliskan hasilnya pada tempat yang sudah tersedia.

**Program 6.4 deret\_ganjil\_cara1.cpp**

```
1  #include <stdio.h>
2  main()
3  {
4      int i;
5      printf("10 Bilangan          Pertama\n");
6      Ganjil=1; i<=19; i=i+2) {
7          printf("%4i", i);
8      }
9
10     return 0;
11 }
```

Tuliskan hasilnya

8. Program 6.4. merupakan program untuk menampilkan 10 bilangan ganjil yang dimulai dari 1. Program 6.5 merupakan program cara kedua untuk tujuan yang sama. Walaupun keduanya menghasilkan keluaran yang sama, namun cara kedua lebih mudah dipahami dan lebih disarankan.

Tuliskan dan jalankan program 6.5 berikut ini. Catat hasilnya!

**Program 6.5 deret\_ganjil\_cara2.cpp**

```
1  #include <stdio.h>
2  main()
3  {
4      int i, N=1;
5      printf("10 Bilangan          Pertama\n");
6      for(i=1; i<=10; i++) {
7          printf("%4i", N);
8          N = N + 2;
9      }
10
11     return 0;
12 }
```

Tuliskan hasilnya

9. Jika program 6.5 diubah menjadi program 6.6 berikut ini, apa yang terjadi? Tuliskan dan jalankan programnya, lalu catat apa yang terjadi. Mengapa?

**Program 6.6 deret\_ganjil\_cara2.cpp**

```
1  #include <stdio.h>
2  main()
3  {
4      int i, N=1;
5      printf("10 Bilangan Ganjil Pertama\n");
6      for(i=1; i<=10; i--) {
7          printf("%4i", N);
8          N = N + 2;
9      }
10
11     return 0;
12 }
```

Tuliskan hasil program dan komentar Anda!

## 6.4 LATIHAN

1. Tuliskan dan jalankan program berikut ini dan tuliskan hasilnya di tempat yang sudah disediakan.

**Program 6.7 deret\_genap.cpp**

```
1  #include <stdio.h>
2  main()
3  {
4      int i, N=10;
5      printf("Deret          Genap\n");
6      for(i=1; i<=10; i++)
7      {   printf("%4i", N);
8          N = N - 2;
9      }
10
11     return 0;
12 }
```

Hasil Program 6.7

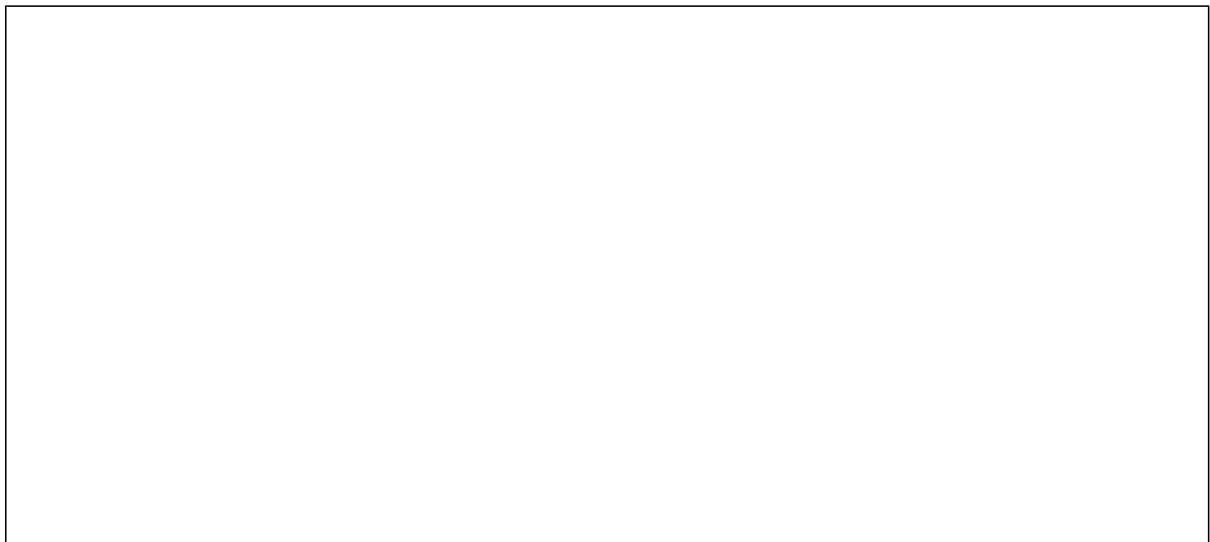
Gambarkan Flowchart dari Program 6.7



#### Program 6.8 cari\_bilangan.cpp

```
1  #include "stdio.h"
2  main()
3  {
4      int A[10] = {10,5,7,15,12,20,9,7,11,14};
5      int N, I, flag = 0;
6      printf("Bilangan\n");
7      for (I=0; I<10; I++)
8          printf("%3i", A[I]);
9      printf("\n");
10
11     printf("Input bilangan yang dicari: ");
12     scanf("%i", &N);
13
14     for (I=0; I<10; I++) {
15         if (A[I] == N) {
16             flag = 1;
17         }
18     }
19     if (flag == 1) {
20         printf("Bilangan %i ADA ditemukan.", N);
21     } else {
22         printf("Bilangan %i TIDAK ditemukan.", N);
23     }
24
25     return 0;
26 }
```

#### Hasil Program 6.8



## 6.5 TUGAS MANDIRI

Kerjakan soal-soal berikut ini:

1. Buatlah program Bahasa C untuk mencetak deret berikut ini:  
10 20 30 40 50 60 70 80 90 100
2. Buatlah program Bahasa C untuk mencetak deret berikut ini:  
100 95 90 85 80 75 70 65 60 55
3. Buatlah program Bahasa C untuk mencetak deret berikut ini:  
1 2 4 8 16 32 64 128 256 512 1024