

PRAKTIKUM 10

STRUKTUR PERULANGAN

10.1 TUJUAN PRAKTIKUM

Tujuan Umum

Mahasiswa dapat memahami:

1. Bentuk umum struktur perulangan FOR, WHILE, DO-WHILE.
2. Penggunaan struktur perulangan pada sebuah program.

Tujuan Khusus

Mahasiswa dapat :

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

10.2 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.

10.2.1 Struktur Perulangan FOR

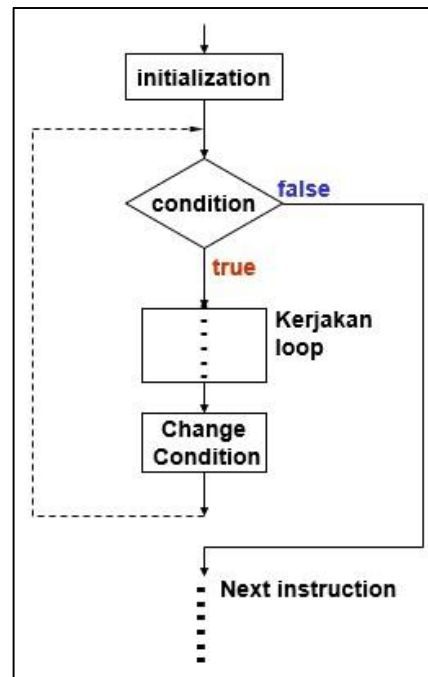
Struktur perulangan FOR biasanya digunakan untuk mengulang suatu proses yang telah diketahui jumlah perulangannya. Dari segi penulisannya, struktur perulangan FOR memiliki bentuk yang sederhana.

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.

**10.2.2 Struktur Perulangan WHILE**

Pada struktur pengulangan jenis ini kondisi akan diperiksa di bagian awal. Hal ini tentu menyebabkan kemungkinan bahwa apabila ternyata kondisi yang kita definisikan tidak terpenuhi (bernilai salah), maka proses pengulangan pun tidak akan pernah dilakukan. Adapun bentuk umum dari struktur while adalah seperti yang tampak di bawah ini.

```

while (ekspresi) {
    Statemen_yang_akan_diulang1;
    Statemen_yang_akan_diulang2;
    ...
}
  
```

Sama seperti pada struktur for, struktur pengulangan jenis ini juga memerlukan suatu inisialisasi nilai pada variabel yang akan digunakan sebagai pencacah, yaitu dengan menuliskannya di atas blok pengulangan. Selain itu kita juga harus melakukan penambahan ataupun pengurangan terhadap nilai dari variabel pencacah di dalam blok pengulangan tersebut. Hal ini bertujuan untuk menghentikan pengulangan sesuai dengan kondisi yang didefinisikan. Sebagai contoh apabila kita ingin melakukan pengulangan proses sebanyak 5 kali, maka kita akan menuliskannya sebagai berikut.

```
int j = 0; /* Melakukan inisialisasi terhadap variabel j dengan
           nilai 0 */
while (j<5) {
    /* Statemen yang akan diulang */
    ...
    j++;    /* Melakukan increment terhadap variabel j */
}
```

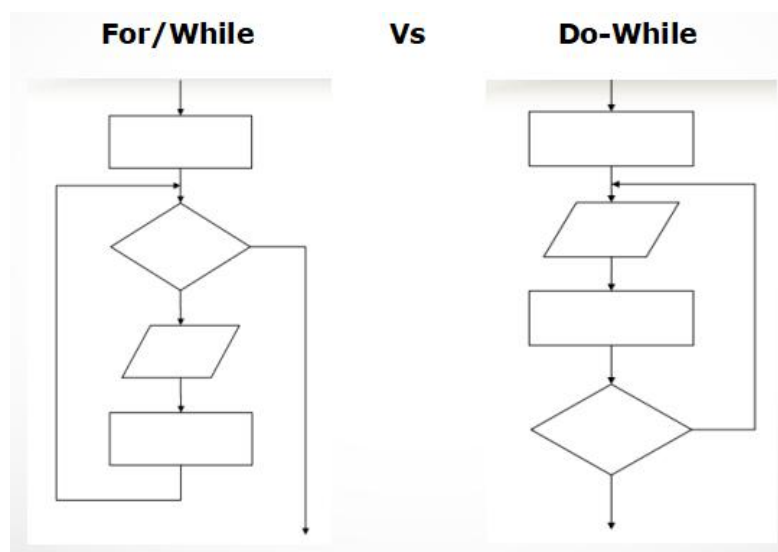
10.2.3 Struktur Perulangan DO-WHILE

Berbeda dengan struktur while dimana kondisinya terletak di awal blok pengulangan, pada struktur do-while kondisi diletakkan di akhir blok pengulangan. Hal ini menyebabkan bahwa statemen yang terdapat di dalam blok pengulangan ini pasti akan dieksekusi minimal satu kali, walaupun kondisinya bernilai salah sekalipun. Maka dari itu struktur do-while ini banyak digunakan untuk kasus-kasus pengulangan yang tidak mempedulikan benar atau salahnya kondisi pada saat memulai proses pengulangan.

Adapun bentuk umum dari struktur pengulangan do-while adalah seperti yang tertulis di bawah ini.

```
do {
    Statemen_yang_akan_diulang;
    ...
} while (ekspresi);    /* Ingat tanda semicolon (;) */
```

Dibawah ini perbedaan While dan Do-While jika digambarkan dalam sebuah flowchart.



10.3 PELAKSANAAN PRAKTIKUM

1. Tuliskan Program 10.1 berikut ini pada editor Dev-C++ (program ini merupakan program untuk mencari nilai terbesar dari 3 buah bilangan yang diinput).

Program 10.1 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
20 }
```

2. Program 10.1 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 10.1 menjadi Program 10.2 berikut ini.

Program 10.2 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 11.2 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 11.3 berikut ini, tuliskan pada editor dan jalankan!

Program 11.3 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 }

```

5. Ujicoba program 10.3 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 10.3 di atas, jawablah beberapa pertanyaan berikut ini!

NO	PERTANYAAN	JAWABAN
1	Pada struktur FOR, tuliskan perintah bagian inisialisasi !	
2	Pada struktur FOR, tuliskan perintah yang menunjukkan kondisi 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 10.3 dan data inputan pada langkah ke-5, berapa nilai variabel i setelah keluar dari perulangan?	

7. Tuliskan dan jalankan program 10.4 berikut ini! Lalu tuliskan hasilnya pada tempat yang sudah tersedia.

Program 10.4 deret_ganjil_cara1.cpp

```

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

Tuliskan hasilnya

8. Program 10.4. merupakan program untuk menampilkan 10 bilangan ganjil yang dimulai dari 1. Program 10.5 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 10.5 berikut ini. Catat hasilnya!

Program 10.5 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 }
```

Tuliskan hasilnya

9. Jika program 10.5 diubah menjadi program 10.6 berikut ini, apa yang terjadi? Tuliskan dan jalankan programnya, lalu catat apa yang terjadi. Mengapa?

Program 10.6 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 }
```

Tuliskan hasil program dan komentar Anda!

10. Bagaimana menggunakan struktur perulangan WHILE?

Program 10.7 jumlah_15_bil.cpp

```
1  #include <stdio.h>
2  main()
3  {
4      int j = 1;
5      int jumlah = 0;
6
7      while (j <= 5) {
8          jumlah += j;
9          j++;
10     }
11
12     printf("Jumlah = %d", jumlah);
13 }
```

Program 10.8 perulangan_bersarang.cpp

```
1  #include <stdio.h>
2  main()
3  {
4      int j=1;
5      int k;
6      /* Melakukan pengulangan ke-1 */
7      while (j <= 10) {
8          k = 1;
9          /* Melakukan pengulangan ke-2 */
10         while(k <=10 ) {
11             printf("%3d" , j*k);
12             k++;
13         }
14         printf("\n");
15         j++;
16     }
17 }
```

Program 10.9 perulangan bersarang kombinasi.cpp

```
1  #include <stdio.h>
2  main(void)
3  {
4      int j;
5      int k;
6      for (j=1; j<=5; j++) {
7          k = 1;
8          while(k<=j ) {
9              printf("%2d ", j*k);
10             k++;
11         }
12         printf("\n");
13     }
14     j = 4;
15     while (j>=1) {
16         for (k=1; k<=j; k++ ) {
17             printf("%2d ", j*k);
18         }
19         printf("\n");
20         j--;
21     }
22 }
```

11. Untuk mengetahui perbedaan dari penggunaan perulangan WHILE dan DO-WHILE, cobalah menuliskan dua program dibawah ini (10.10 dan 10.11) kemudian selidiki perbedaan output diantara keduanya?

Program 10.10 perulangan dg while.cpp

```
1  #include <stdio.h>
2  main()
3  {
4      int j;
5      printf("Statemen sebelum blok pengulangan\n");
6      j = 10;
7      while (j < 5) {
8          printf("Statemen di dalam blok pengulangan\n");
9          j++;
10     }
11     printf("Statemen setelah blok pengulangan\n");
12 }
```

Program 10.11 perulangan_dg_dowhile.cpp

```
1  #include <stdio.h>
2  main()
3  {
4      int j;
5      printf("Statemen sebelum blok pengulangan\n");
6      j = 10;
7      do {
8          printf("Statemen di dalam blok pengulangan\n");
9          j++;
10     } while (j < 5);
11     printf("Statemen setelah blok pengulangan\n");
12 }
```

10.4 LATIHAN

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

Program 10.12 deret_genap.cpp

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

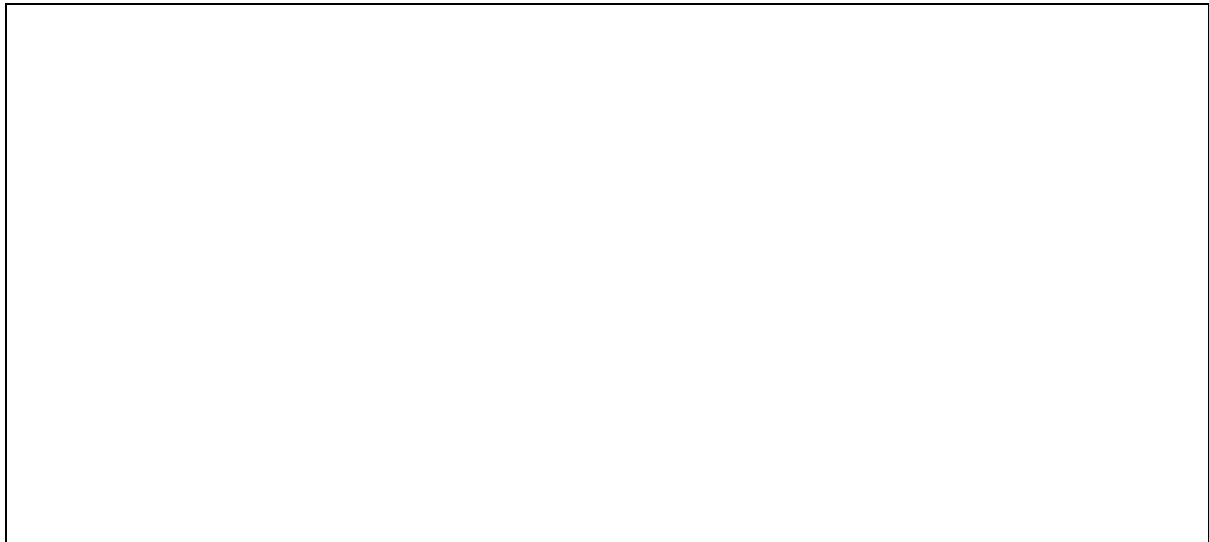
Hasil Program 10.12

Gambarkan Flowchart dari Program 10.12

Program 10.13 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 }
```

Hasil Program 10.13



10.5 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
4. Seseorang mengendarai sepeda dengan kecepatan tetap 2 meter/detik. Susun program untuk mencetak berapa meter yang dia tempuh setelah bersepeda selama 100 detik.
5. Seseorang menyimpan uang Rp. 1.000.000 di bank dengan bunga ber-bunga 2% perbulan. Jadi setelah satu bulan uangnya menjadi Rp. 1.020.000. Satu bulan berikutnya uang Rp. 1.020.000 ini mendapat bunga lagi 2%, yaitu Rp.20.400 sehingga setelah 2 bulan uangnya menjadi $\text{Rp. } 1.020.000 + \text{Rp. } 20.400 = \text{Rp. } 1.040.400$. Demikian seterusnya (bunga bulan ini ditambahkan ke saldo uangnya dan mendapatkan bunga lagi pada bulan berikutnya) . Susun program untuk menghitung dan mencetak jumlah uangnya setelah 10 bulan.