

## PERTEMUAN 4

### Perulangan

Salah satu keunggulan komputer adalah dapat melakukan tugas yang sama secara terus-menerus/berulang-ulang tanpa mengenal lelah. Untuk dapat menginstruksikan computer melakukan tugas yang berulang-ulang terdapat 3 sintaks yang umum digunakan, yaitu :

1. FOR-DO
2. WHILE-DO
3. DO-WHILE

Perulangan **FOR-DO** digunakan ketika kita mengetahui jumlah perulangan yang akan terjadi.

Perulangan **WHILE-DO** digunakan ketika kita tidak mengetahui berapa jumlah perulangan yang harus dijalankan untuk mencapai goal. Pengecekan tercapai atau tidaknya goal (kondisi) dicek sebelum statement dijalankan.

Perulangan **DO-WHILE** digunakan ketika tidak mengetahui berapa jumlah perulangan yang harus dijalankan untuk mencapai goal. Pengecekan tercapai atau tidaknya goal (kondisi) dicek setelah statement dijalankan. Dengan kata lain statement akan dijalankan minimal sekali.

PSEUDOCODE	
	<b>bentuk umum FOR</b>
1	// deklarasi counter
2	FOR ( <i>/*nilai counter start*/</i> ) TO ( <i>/*kondisi stop*/</i> ) DO
3	// statement
4	// increment counter
5	ENDFOR
	<b>contoh 1-FOR singular</b>
1	INT i
2	FOR (i := 0) TO (i < 10) DO
3	// statement
4	i := i + 1
5	ENDFOR
	<b>contoh 2-FOR bersarang (nested Loop)</b>
1	INT i, j
2	FOR (i := 0) TO (i < 10) DO
3	// statement
4	FOR (j := 0) TO (j < 10) DO
5	// statement
6	j := j + 1

7	ENDFOR
8	i := i + 1
9	ENDFOR
<b>bentuk umum WHILE-DO</b>	
1	// deklarasi kondisi
2	WHILE (kondisi is true) DO
3	// statement
4	// statement untuk mengubah kondisi menjadi false
5	ENDWHILE
<b>contoh 1-WHILE-DO</b>	
1	INT x := 0
2	WHILE (x < 3) DO
3	// statement
4	x := x + 1
5	ENDWHILE
<b>contoh 2-WHILE-DO</b>	
1	BOOL cond := TRUE
2	INT x := 0
3	WHILE (cond) DO
4	// statement
5	x := x + 1
6	IF (x > 3)
7	cond := FALSE
8	ENDWHILE
<b>bentuk umum DO-WHILE</b>	
1	// deklarasi kondisi
2	DO
3	// statement
4	WHILE (kondisi is true)
5	// statement untuk mengubah kondisi menjadi false
6	ENDWHILE

<b>C++</b>	
<b>FOR singular</b>	
1	for (int i = 0; i < 10; i++)
2	{
3	// statement
4	}

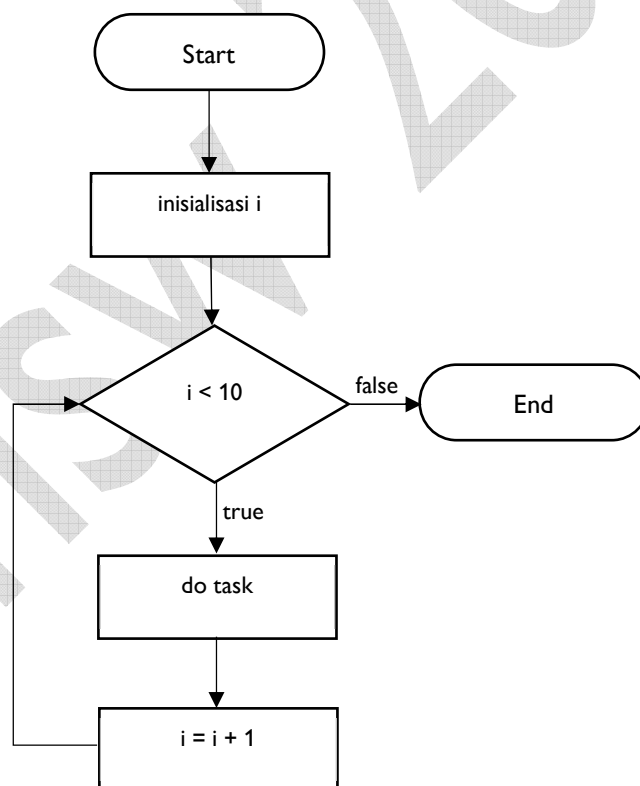
	<b>FOR bersarang (nested Loop)</b>  <pre> 1  for (int i = 0; i &lt; 10; i++) 2  { 3      // statement 4      for (int j = 0; j &lt; 10; j++) 5      { 6          // statement 7      } 8      // statement 9  }</pre>
	<b>contoh 1-WHILE-DO</b>  <pre> 1  int x = 0; 2  while (x &lt; 3) 3  { 4      // statement 5      x++; 6  }</pre>
	<b>contoh 2-WHILE-DO</b>  <pre> 1  bool cond = true; 2  int x = 0; 3  while (cond) 4  { 5      // statement 6      x++; 7      if (x &gt; 3) 8      { 9          cond = false; 10     } 11 } 12 }</pre>
	<b>contoh 1-DO-WHILE</b>  <pre> 1  int x = 0; 2  do 3  { 4      // statement 5      x++; 6  } 7  while (x &lt; 3);</pre>

**contoh 2-DO-WHILE**

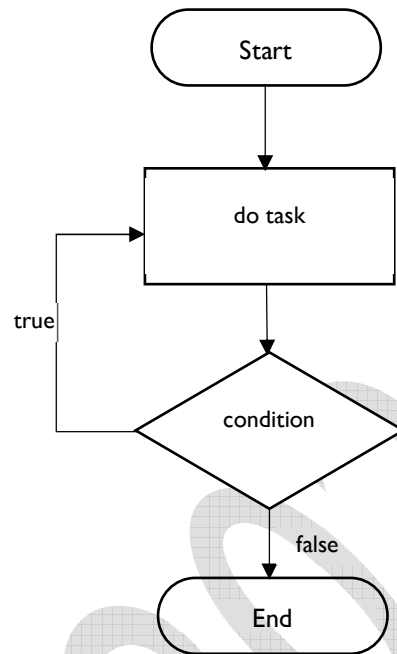
```
1 bool cond = true;
2 int x = 0;
3 do
4 {
5     // statement
6     x++;
7     if (x > 3)
8     {
9         cond = false;
10    }
11 }
12 while (cond);
```

**FLOWCHART**

*FOR singular*



### **WHILE-DO**



### **DO-WHILE**

