

# ISA- 105

# Kontrol Program: Pengulangan

Sofia Umaroh  
Prodi Sistem Informasi  
Itenas Bandung



# Review Tugas

---

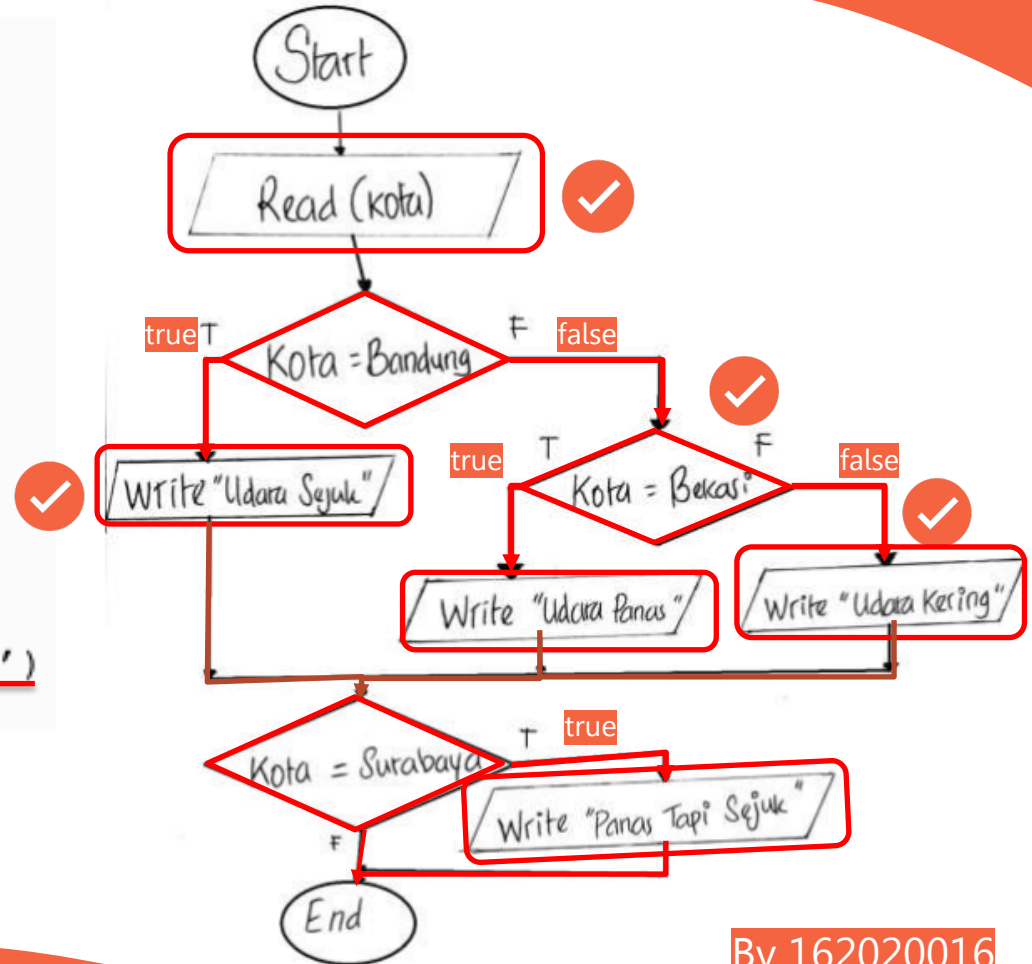
Tell us what you've gained from the self-study experience these week!

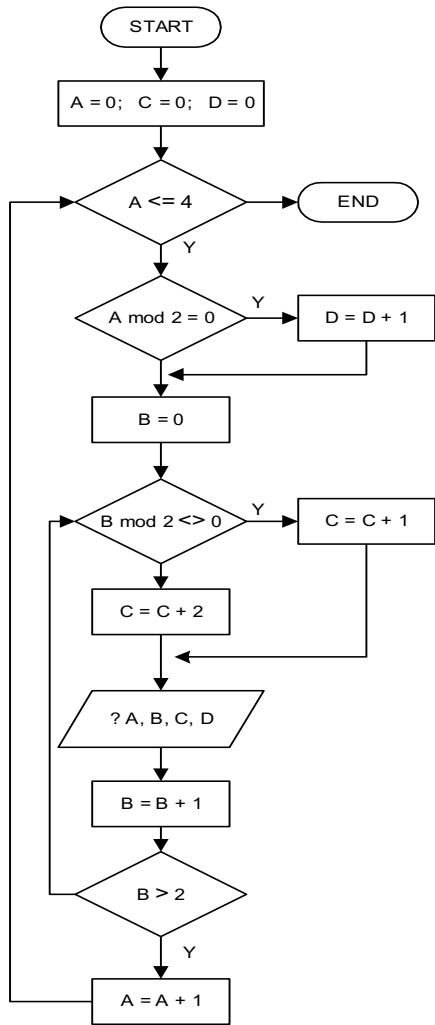


Deklarasi  
kota : string

Algoritma

```
read(kota);  
if kota = 'Bandung' then  
    write('Udara sejuk')  
else if kota = 'Bekasi' then  
    write('Udara panas')  
else  
    write('Udara kering')  
endif  
  
if kota = 'Surabaya' then  
    write ('Panas tapi sejuk')  
endif
```





A  
5

B  
3

C  
25

D  
3

A	B	C	D
0	0	2	1
0	1	3	1
0	2	5	1
1	0	7	1
1	1	8	1
1	2	10	1
2	0	12	2
2	1	13	2
2	2	15	2
3	0	17	2
3	1	18	2
3	2	20	2
4	0	22	3
4	1	23	3
4	2	25	3

# Struktur Pengulangan

---

Dalam menyelesaikan masalah, algoritma memiliki 4 kontrol program dengan struktur yang berbeda



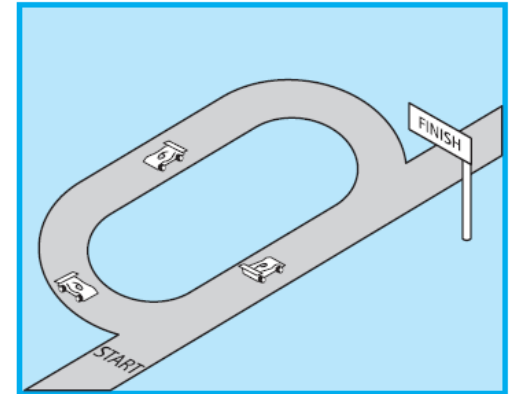
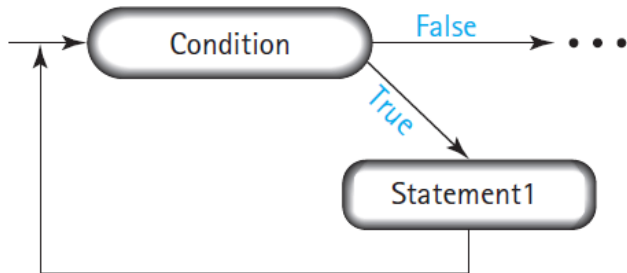


# Struktur Pengulangan

- Struktur kontrol berulang (loop), mengulangi pernyataan selama kondisi (syarat) tertentu terpenuhi.

LOOP (also called *repetition* or *iteration*)

WHILE condition DO statement1





# Struktur Pengulangan

- Struktur kontrol berulang (loop), mengulangi pernyataan selama kondisi (syarat) tertentu terpenuhi.
- Struktur logika pengulangan memungkinkan urutan instruksi dieksekusi terus menerus selama kondisi terpenuhi.
- Gunakan tiga jenis instruksi Loop dalam pemecahan masalah (sementara, lakukan-sementara, untuk instruksi)
- Mis. masalah loop: Menghitung, mengumpulkan jumlah



# Jenis Struktur Pengulangan

1. **While ... do ...**
2. **For ... do ...**
3. Do ... while ...



# Struktur Program Do - While

---

Pengulangan **Do - While** dilakukan setidaknya satu kali, pengecekan kondisi dilakukan setelah pengulangan pertama dilakukan





# Struktur Pengulangan: Do - While

- Pernyataan do – while (atau dalam Bahasa Pascal repeat – until) mirip dengan pernyataan while.
- Dalam pernyataan while, kondisi kelanjutan loop diuji **diawal** pengulangan sebelum badan pengulangan dilakukan.
- Pernyataan do – while (repeat – until) menguji kondisi pengulangan setelah badan pengulangan dilakukan.
- Karena itu, badan pengulangan akan **dilakukan setidaknya sekali**.



# Struktur Pengulangan: Do - While

lakukan suatu aksi  
dan akan diulang  
kembali...

```
do {  
    // aksi  
} while(kondisi)
```

selama (*while*)  
kondisi terpenuhi  
(bernilai True)



# Komponen Penting: Do

## - while

1. **Counter**, yang mengendalikan banyaknya pengulangan dilakukan
2. **Inisialisasi counter**, tentukan pencacah dimulai dari berapa?
3. **Kondisi berhenti**, tentukan kondisi pengulangan harus berhenti
4. **Penambahan counter**, tambahkan nilai counter untuk pengulangan berikutnya



# Instruksi: **While**

Memberi tahu komputer untuk

- a) **melakukan suatu aksi**
- b) **kemudian menguji <kondisi>** dan jika kondisi itu benar
- c) **maka aksi tersebut diulang**
- d) jika kondisi salah, **pengulangan berhenti**

```
Inisialisasi counter;  
do {  
    aksi ...  
    aksi ...  
    :  
    update aksi  
} while (kondisi (s))
```

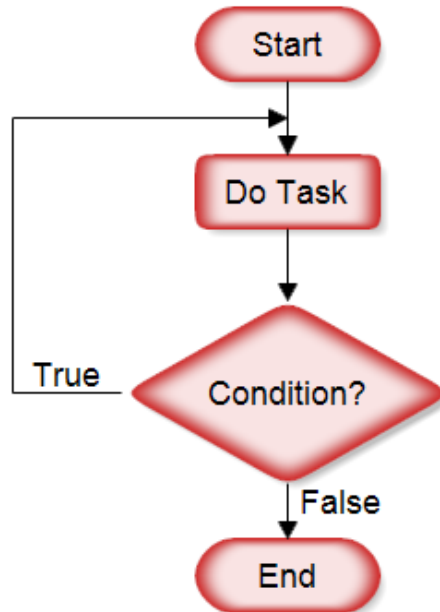
Badan pengulangan  
di dalam kurung  
kurawal { ... }



# Flowchart: While or Do-While?

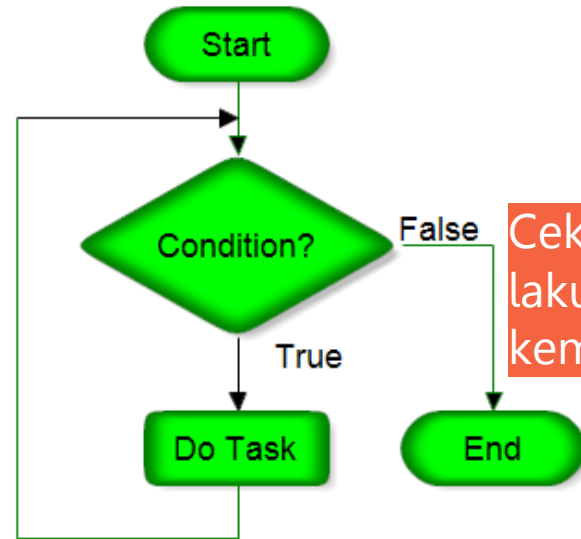
Do While Loop

Lakukan dulu  
cek kemudian



While Loop

Cek dulu  
lakukan  
kemudian





## Contoh 1:

- Memakan satu pack oreo Supreme
- Aksi dilakukan dulu
- kemudian setelah aksi pertama dilakukan, kondisi dicek
- Jika bernilai True, ulangi aksi
- jika tidak, selesai makan oreo (sudah habis)





## Contoh 2:

### Problem:

Anton diminta untuk menerima input angka acak secara terus-menerus dan akan berhenti jika bilangan yang diinputkan adalah -99 (nilai sentinel)

### Deklarasi

nilai : integer

### Algorithm

1. begin
2. read (nilai)
3. if nilai = -99, go to step2
4. if nilai != -99, go to step 5
5. end

aksi yang diulang

```
Masukkan nilai:8
Masukkan nilai:12
Masukkan nilai:3
Masukkan nilai:43
Masukkan nilai:21
Masukkan nilai:-99
```

```
...Program finished with exit code 0
console.█
```

Kondisi pengulangan, jika benar kembali ke step2





## Contoh 2:

### Problem:

Anton diminta untuk menerima input angka acak secara terus-menerus dan akan berhenti jika bilangan yang diinputkan adalah -99

Pseudocode	Bahasa Pascal	Bahasa C
<pre>Deklarasi   nilai : integer  Algorithm begin read (nilai) if nilai = -99, go to step2 if nilai != -99, go to step 5 end</pre>	<pre>Program inputNilai; Var   nilai: integer;  Begin   repeat     printf("Masukan nilai:");     read(nilai);   until (nilai &lt;&gt; -99); End.</pre>	<pre>#include &lt;stdio.h&gt;;  int main(){   //deklarasi   int nilai;    do{     printf("Masukkan nilai; ");     scanf("%d",&amp;nilai);   } while(nilai != -99);   return 0; }</pre>



## Contoh 3:

### Problem:

Melanjutkan kasus ada Contoh 2, Anton diminta **menjumlahkan semua** nilai yang diinputkan oleh user sampai user menginputkan nilai -99.

### Deklarasi

nilai, jumlah : integer

### Algorithm

1. begin
2. jumlah <- 0
3. read (nilai)
4. jumlah <- jumlah + nilai
5. if nilai <- -99, go to step3
6. if nilai <> -99, cetak jumlah
7. end

aksi yang diulang

```
Masukkan nilai:1
Masukkan nilai:2
Masukkan nilai:3
Masukkan nilai:4
Masukkan nilai:5
Masukkan nilai:-99
Total jumlah nilai yang diinputkan = 15
Press .Program finished with exit code 0
Press
```

Kondisi pengulangan, jika benar kembali ke step2



## Contoh 3:

### Problem:

Melanjutkan kasus ada Contoh 2, Anton diminta **menjumlahkan semua** nilai yang diinputkan oleh user sampai user menginputkan nilai -99.

Pseudocode	Bahasa Pascal	Bahasa C
<p><b>Deklarasi</b>     nilai, jumlah : <u>integer</u></p> <p><b>Algorithm</b></p> <ol style="list-style-type: none"><li><u>begin</u></li><li>jumlah &lt;- 0</li><li>read (nilai)</li><li>jumlah &lt;- jumlah + nilai</li><li>if nilai &lt;- -99, go to step3</li><li>if nilai &lt;&gt; -99, cetak jumlah</li><li><u>End</u></li></ol>	<pre>Program inputNilai; Var     nilai, jum: integer;  Begin     jum := 0;     repeat         printf("Masukan nilai:");         read(nilai);         jum:=jum=nilai;     until (nilai &lt;&gt; -99);     write(jumlah);  End.</pre>	<pre>#include &lt;stdio.h&gt;; int main(){     //deklarasi     int nilai, jum;     jum = 0;     do{         printf("Masukkan nilai; ");         scanf("%d",&amp;nilai);         jum+=nilai;     } while(nilai != -99);     printf(jum);     return 0; }</pre>



## LATIHAN:

### **Problem:**

Kelas Alpro mengikuti kuis. Skor (bilangan bulat dalam rentang 0 hingga 100) untuk kuis ini tersedia untuk Anda dan baris data terakhir ditandai dengan **nilai sentinel -1**. Buatlah Bahasa C/Pascal dan flowchart untuk menentukan rata-rata kelas menggunakan struktur pengulangan **do-while**!



# Struktur Pengulangan: For

```
1. #include <stdio.h>;
2.
3. int main(){
4.     //deklarasi
5.     int counter; //declare counter
6.     for(counter=0;counter<100;counter++) //Header pengulangan
7.     {
8.         printf("Hello World\n"); //Aksi yang diulang
9.     }
10.    return 0;
11. }
```

```
Hello World
Hello World
...
```



# Weekly Activity

1. Kerjakan contoh 2 dan 3 menggunakan compiler C, kemudian screenshot hasil run dan jelaskan apa yang dipahami dari kode masing-masing program
2. Kelas Alpro mengikuti kuis. Skor (bilangan bulat dalam rentang 0 hingga 100) untuk kuis ini tersedia untuk Anda dan baris data terakhir ditandai dengan **nilai sentinel -1. Tuliskan kode program** Bahasa C/Pascal **dan flowchart** untuk menentukan rata-rata kelas menggunakan struktur pengulangan **do-while!**



# Terima Kasih

---

**NEXT:**

- KUIS

