

# ISA-105

## Kontrol Program: Pengulangan

Sofia Umaroh  
Prodi Sistem Informasi  
Itenas Bandung



# Struktur Program

---

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





# Kontrol Program

- Bahasa pemrograman mengharuskan kita menggunakan struktur kontrol tertentu untuk mengekspresikan algoritma sebagai source code
- Ada empat cara dasar untuk menyusun pernyataan (instruksi) di bahasa pemrograman:
  1. **Sequence (runtutan),**
  2. **Selection (pemilihan),**
  3. **Loop (pengulangan), dan**
  4. **Subprogram (prosedur).**



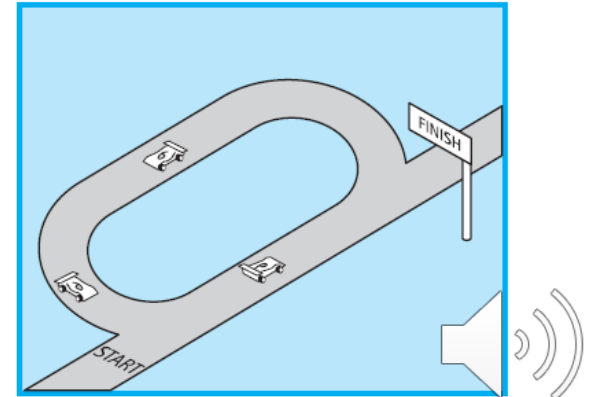
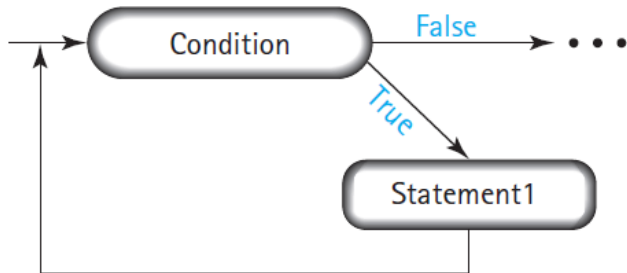


# Struktur Pengulangan

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

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

WHILE condition DO statement1





# Jenis Struktur Pengulangan

1. While ... do ...
2. Do ... while ...
3. Repeat ... until ...
4. For ... do ...



# Struktur Program While

Pengulangan **While** dapat digunakan pada struktur perulangan yang **diketahui** jumlah iterasinya, maupun **tidak diketahui**





# Instruksi 1

Cetak "Hello World"

## Bahasa C

```
int main(){  
    printf("Hello World");  
    return 0;  
}
```

Hello World

## Pascal

```
program Hello;  
begin  
    writeln ('Hello World')  
end.
```

Hello World





# Instruksi 2

Cetak "Hello World" sebanyak 5 kali!

## Bahasa C

```
int main(){  
    printf("Hello World");  
    printf("Hello World");  
    printf("Hello World");  
    printf("Hello World");  
    printf("Hello World");  
    return 0;  
}
```

## Output:

```
Hello World  
Hello World  
Hello World  
Hello World  
Hello World
```

## Pascal

```
program Hello;  
begin  
    writeln ('Hello World')  
    writeln ('Hello World')  
    writeln ('Hello World')  
    writeln ('Hello World')  
    writeln ('Hello World')  
end.
```

## Output:

```
Hello World  
Hello World  
Hello World  
Hello World  
Hello World
```







# Instruksi 3

Cetak "Hello World" sebanyak 150 kali!

## Bahasa C

```
1. int main(){
2.     printf("Hello World");
3.     printf("Hello World");
4.     printf("Hello World");
5.     printf("Hello World");
6.     printf("Hello World");
7.     ... ;
152.    printf("Hello World");
153.    printf("Hello World");
154.    printf("Hello World");
155.    printf("Hello World");
156.    return 0;
157.}
```

## Output:

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

Ada 150 baris kode  
yang sama?  
Serius?

Apakah ini efisien?

**TIDAK**  
Kita membutuhkan  
**Pengulangan**





# Struktur Pengulangan: **While**

- Pengulangan **While** dapat digunakan pada struktur perulangan yang **diketahui** jumlah iterasinya, maupun **tidak diketahui**
- Harus selalu ada kondisi untuk menghentikan pengulangan, **misal:** Roket diluncurkan setelah hitungan ke 10
- **Counter** adalah variable pengendali pengulangan yang harus **diinisialisasi, dicek dalam kondisi, dan nilainya terus berubah** setiap pengulangan dilakukan





# Komponen Penting: **While**

1. **Counter**, variable yang mengatur banyaknya pengulangan dilakukan. Counter start dan end?
2. **Inisialisasi counter**, tentukan penghitung dimulai dari berapa?
3. **Kondisi berhenti**, tentukan kondisi kapan pengulangan harus berhenti (counter maksimal berapa?)
4. **Penambahan counter**, tambahkan nilai counter untuk pengulangan berikutnya



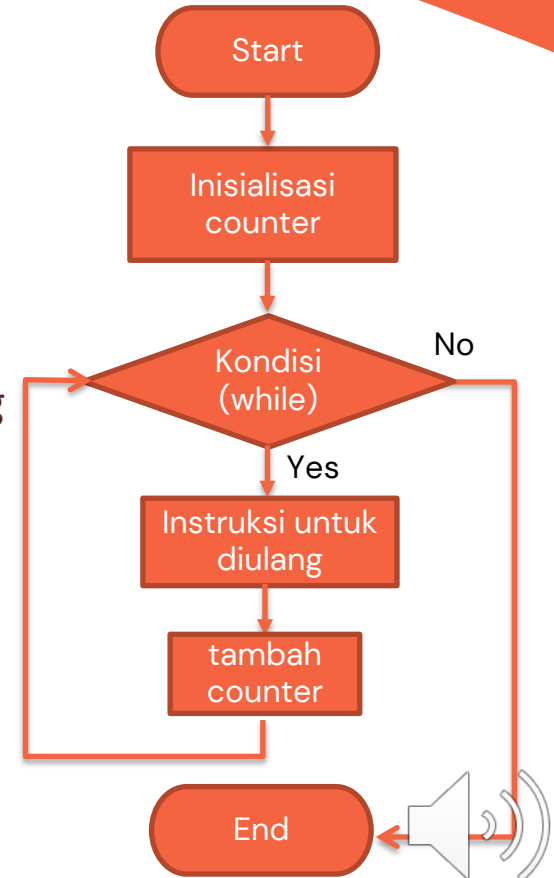


# Instruksi: **While**

Memberi tahu komputer untuk

- Menentukan posisi awal counter
- menguji <kondisi>** dan jika kondisi itu benar
- mengulangi semua instruksi** antara tanda kurung (mulai) "{" dan (akhir) "}"
- jika kondisi salah, **pengulangan berhenti**

```
set counter mulai;  
while (kondisi (s)) {  
    instruksi ...  
    tambah counter  
}
```





# Studi kasus:

Cetak "Hello World" sebanyak 5 kali!

**Untuk mencetak "Hello World" menggunakan pengulangan, pertama kita tentukan komponen pentingnya.**

**Berapa kali pengulangan dilakukan?**

Sebanyak 5 kali, maka  $n = 5$

**Dimulai dari berapa? (inisialisasi counter)**

Pengulangan dimulai dari 0, maka counter = 0

**Syarat pengulangan dilakukan**

Selama counter  $< 5$ , cetak Hello World

**Berapa penambahan counternya setiap kali diulang?**

Penambahan 1 setiap kali instruksi akan diulang

**Instruksi yang ingin diulang?**

Cetak "Hello World"  
`printf("Hello World");`





# Studi kasus:

Cetak "Hello World" sebanyak 150 kali!

## Algoritma

1. *counter* dimulai dari 0
2. Apakah *counter* belum mencapai 5
3.     jika ya: cetak "Hello World"
4. *counter* tambah 1
5. Kembali ke baris-2

~~True~~ false

~~Dilakukan berulang~~

## Pseudocode

```
Deklarasi
    counter : integer {counter}

Algorithm
begin
    counter <- 0 {inisialisasi}

    while (counter < 5)
    begin
        write('Hello World')
        counter = counter+1
    endwhile
end.
```





# Studi Kasus

Problem:

Cetak "Hello World" sebanyak 150 kali!

## Tanpa Pengulangan

### Algorithm

1. begin
2. write ('Selamat Datang')
3. write ('Selamat Datang')
4. write ('Selamat Datang')
5. write ('Selamat Datang')
6. write ('Selamat Datang')
7. end

## Menggunakan Pengulangan

### Deklarasi

counter : integer

harus ada variabel pencacah

### Algorithm

1. begin
2. counter <- 0
3. while (pencacah diinisialisasi)
4.     write ('Hello World')
5.     counter++
6. (endwhile)
7. end

variabel counter

harus

diinisialisasi

harus ada kondisi berhenti





# Studi kasus:

Cetak "Hello World" sebanyak 5 kali!

Deklarasi

```
counter : integer {counter}
```

Algorithm

```
begin
```

```
counter <- 0 {inisialisasi}
```

```
while (counter < 5)
```

```
begin
```

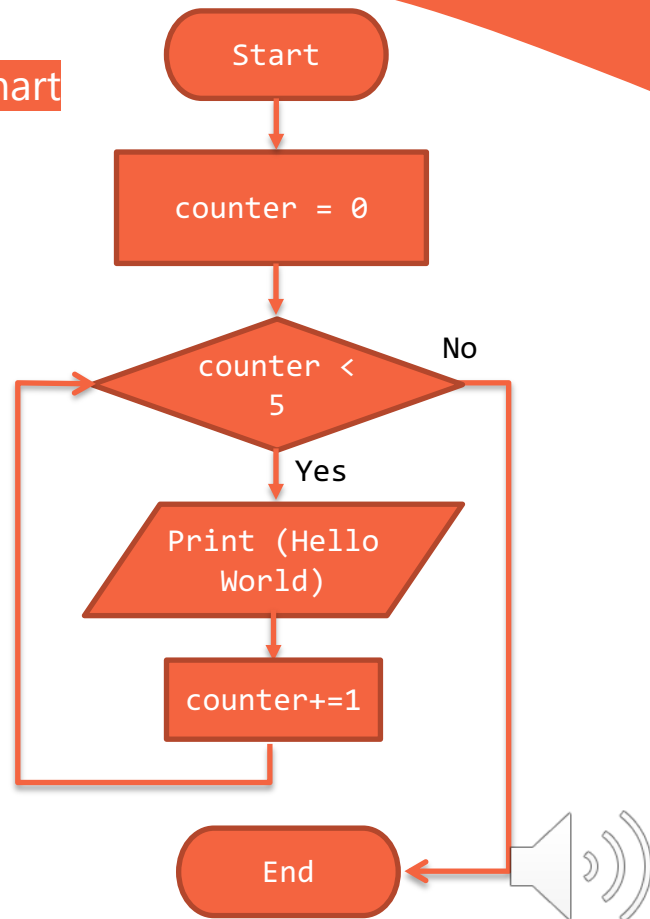
```
write('Hello World')
```

```
counter = counter+1
```

```
endwhile
```

```
end.
```

Flowchart







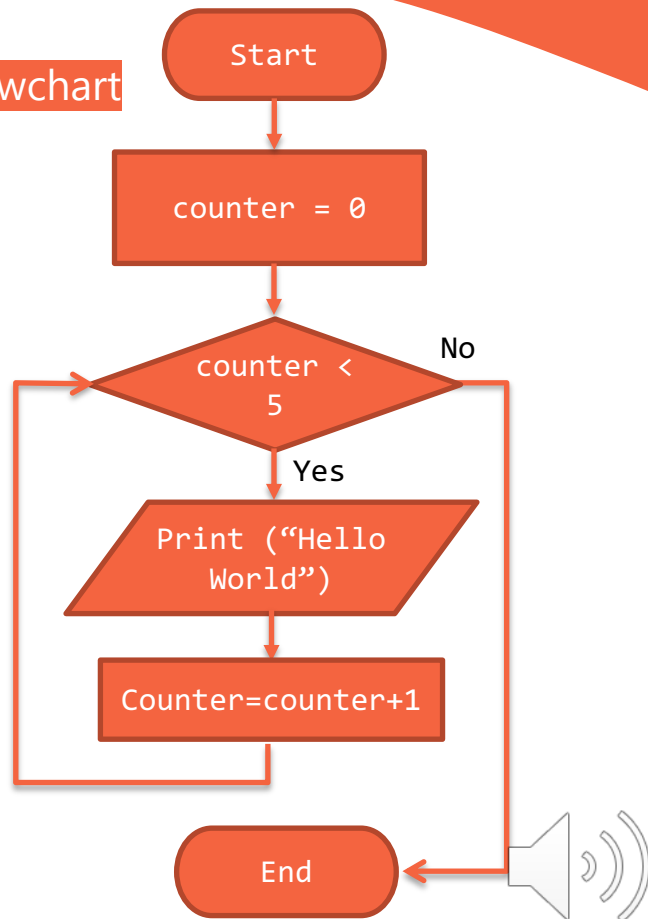
# Cara Kerja Loop:

Cetak "Hello World" sebanyak 5 kali!

| Counter | Counter<5? | Output      | Counter++         |
|---------|------------|-------------|-------------------|
| 0       | yes        | Hello World | Counter = 0+1 = 1 |
| 1       | yes        | Hello World | Counter = 1+1 = 2 |
| 2       | yes        | Hello World | Counter = 2+1 = 3 |
| 3       | yes        | Hello World | Counter = 3+1 = 4 |
| 4       | yes        | Hello World | Counter = 4+1 = 5 |
| 5       | No         |             |                   |

EndLoop

Flowchart



# Studi kasus:

Problem: Cetak "Hello World" sebanyak 5 kali!

## Pseudocode

```
Deklarasi
  counter : integer
{counter}

Algorithm
begin
  counter <- 0
{inisialisasi}
  while (counter < 5)
  begin
    write('Hello World')
    counter = counter+1
  endwhile
end.
```

## Bahasa Pascal

```
Program cetakHello;
var
  counter: integer;

Begin
  counter := 0;
  while counter < 5
  begin
    write('Hello World');
    counter = counter + 1;
  end;
End.
```

## Bahasa C

```
#include <stdio.h>;

int main(){
  //deklarasi
  int counter;
  counter = 0;
  while(counter < 5){
    printf("Hello World");
    counter++;
  }
  return 0;
}
```



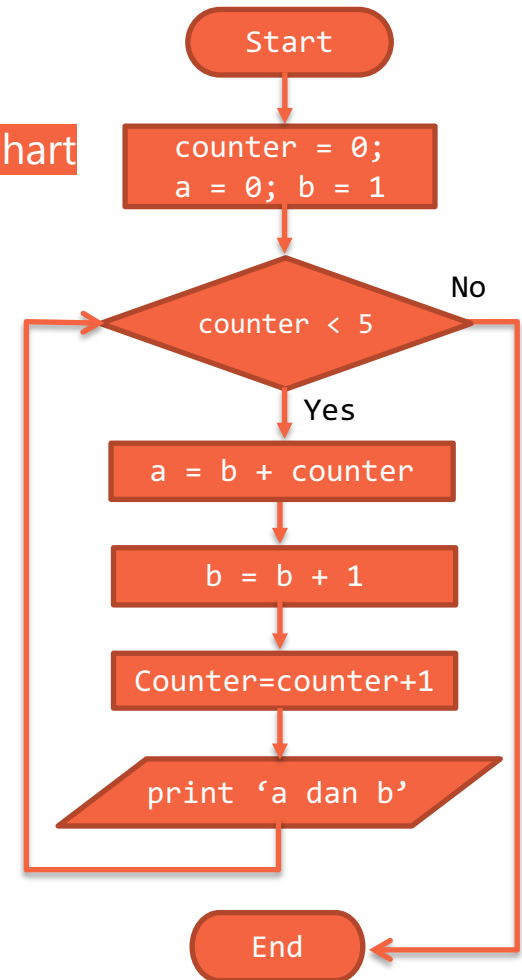


# Cara Kerja Loop:

Apa output dari flowchart ini?

| Counter | Counter<5? | a | Counter++ |
|---------|------------|---|-----------|
|         |            |   |           |
|         |            |   |           |
|         |            |   |           |
|         |            |   |           |
|         |            |   |           |

Flowchart



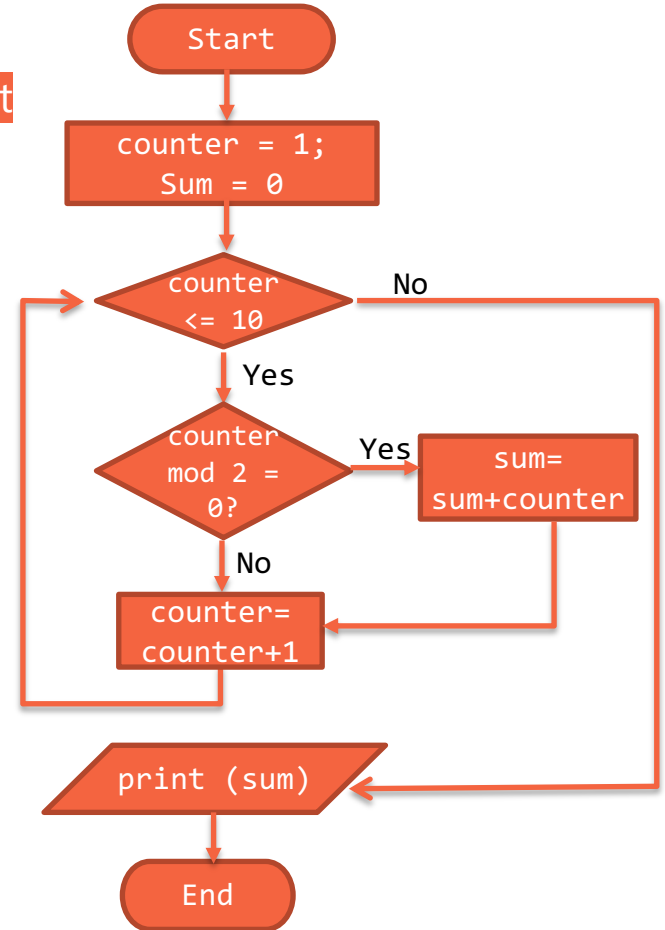


# Latihan:

Apa output dari flowchart ini?

Flowchart

| Counter | Counter $\leq$ 10? | Counter mod 2 = 0? | sum | counter++ | Output |
|---------|--------------------|--------------------|-----|-----------|--------|
|         |                    |                    |     |           |        |
|         |                    |                    |     |           |        |
|         |                    |                    |     |           |        |
|         |                    |                    |     |           |        |
|         |                    |                    |     |           |        |
|         |                    |                    |     |           |        |





## Studi kasus 2:

Kinan diminta untuk mencetak angka mundur dari 10 - 1

*Untuk mencetak angka mundur dari 10 - 1 menggunakan pengulangan, pertama kita tentukan komponen pentingnya.*

**Berapa kali pengulangan dilakukan?**

Sebanyak 10 kali, maka  $n = 10$

**Dimulai dari berapa? (inisialisasi counter)**

Pengulangan dimulai dari 10, maka counter = 10

**Syarat pengulangan dilakukan?**

Selama counter  $> 0$

**Perubahan counter? Penambahan/ pengurangan?**

Hitung mundur dari 10 - 1, maka terdapat pengurangan 1

**Instruksi yang ingin diulang?**

Cetak angka 10 - 1  
`printf( ? );`





## Studi kasus: 2

### Problem:

Kinan diminta untuk mencetak angka mundur dari 10 - 1

### Algoritma

1. deklarasikan variable counter
2. inisialiasai counter = 10
3. selama counter lebih dari 0
4.     cetak counter
5.     counter kurangi 1
6. Kembali ke poin 3
7. jika kondisi 3 salah, selesai.

### Deklarasi

counter : integer

harus ada variabel pencacah

### Algorithm

1. begin
2. counter <- 10
3. while (counter > 0)
4.     write (counter)
5.     counter=counter-1
6. (endwhile)
7. end

variabel counter

harus

diinisialisasi

harus ada kondisi berhenti



## Studi kasus: 2

### Problem:

Kinan diminta untuk mencetak angka mundur dari 10 - 1

| Pseudocode  | Bahasa Pascal  | Bahasa C  |
|---|--|---|
| <b>Deklarasi</b><br>counter : integer   | Program isiBensin;<br>Var<br>counter: integer;   | #include <stdio.h>;<br><br>int main(){  |
| <b>Algorithm</b><br>begin<br>counter <- 10<br>while (counter > 0)<br>write (counter)<br>counter <- counter-1<br>(endwhile)<br>end | Begin<br>counter := 10;<br>while counter > 0<br>begin<br>write(counter);<br>counter = counter-1;<br>end;<br>End. | //deklarasi<br>int counter;<br>counter = 10;<br>while(counter > 0){<br>printf("%d",counter);<br>counter--;<br>}<br>return 0;<br>} |





## Contoh 3:

### Problem:

Kinan diminta untuk mencetak angka mundur dari 10 - 1

```
1.int main(){
2.   int counter; //declare counter
3.   counter = 10; //inisialisasi counter/pencacah = 1
3.
4.   while (counter > 0){           //kondisi loop berkelanjutan
5.       printf("%d ", counter);
6.       counter--; //penambahan variable counter
9.   } //end of while
10.} //end of program
```

10 9 8 7 6 5 4 3 2 1





# Struktur Program For... do...

---

Pengulangan **For** dapat digunakan pada struktur perulangan yang **diketahui** jumlah iterasinya





# Struktur Pengulangan: For

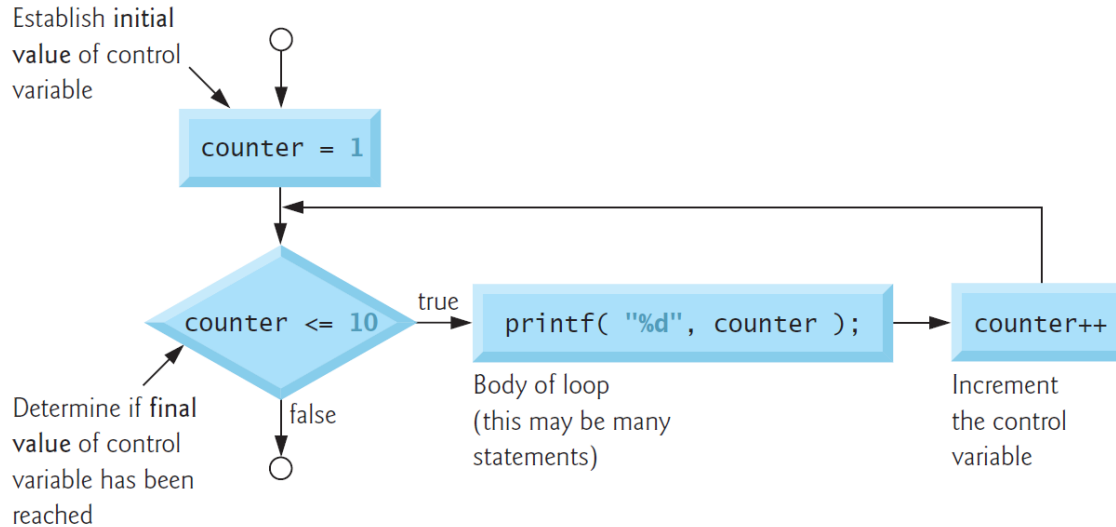
- Pernyataan while dapat digunakan untuk mengimplementasikan setiap loop kontrol balik.
- Terdapat juga pernyataan pengulangan for, yang menentukan detail control pengulangan di dalam satu baris

```
for (initialization; Loop Continuation Condition; increment)  
    statement
```



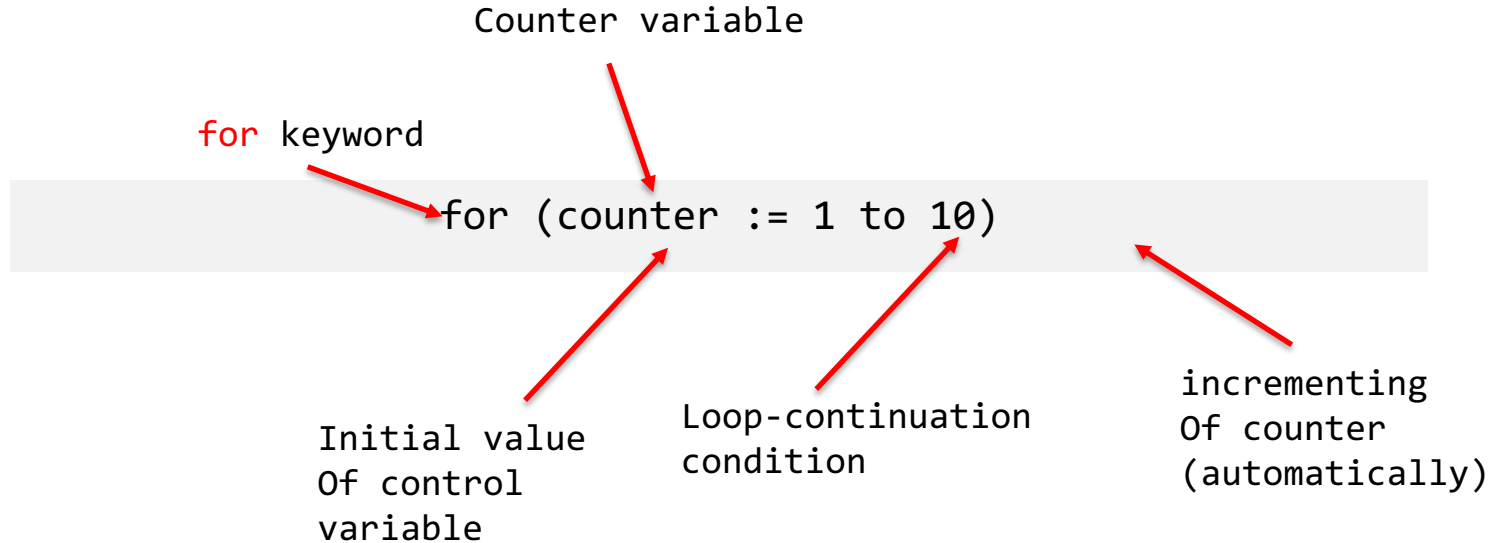


# Flowchart Pengulangan: For



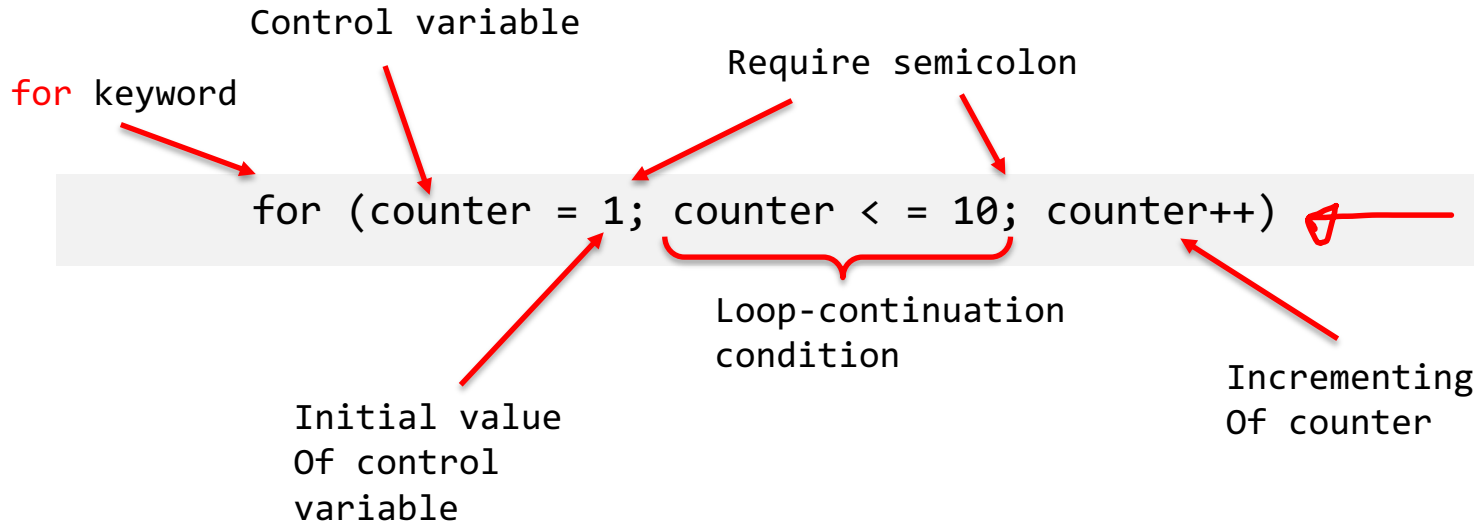


# Struktur Pengulangan: For – Pascal





# Struktur Pengulangan: For – C





# For and While Repetition Statement

```
for (initialization; Loop Continuation Condition; increment)  
    statement
```

Bentuk Pengulangan **for** menggunakan pengulangan **while**

```
initialization;  
while (Loop Continuation Condition) {  
    statement  
    increment;  
}
```





# Studi Kasus 3

Cetak "Hello World" sebanyak 5 kali!

```
//Bahasa C
for (counter=0; counter<5; counter++){
    printf("Hello World);
}
```

```
//Bahasa Pascal
for counter:=0 to 5 do
    write('Hello World');
}
```





# Studi Kasus 4

Kinan diminta untuk mencetak angka mundur dari 10 - 1

```
//Bahasa C
for (counter=0; counter<10; counter--){
    printf("Hello World);
}
```

untuk pengulangan  
mundur counter  
dikurangi 1

```
//Bahasa Pascal
for counter:=10 downto 1 do
    write('Hello World');
}
```

untuk pengulangan  
mundur menggunakan  
**downto**







# Examples Using the for Statement

Vary the control variable from 100 to 1 in increments of -1 (decrements of 1).

```
for counter:=100 downto 1 do
```

```
for (counter=100; counter>0; counter--)
```





# Examples Using the for Statement

Vary the control variable from 7 to 77 in steps of 7 in C.

```
for counter:=7 to 77 do
```

```
for (counter=7; counter<=77; counter+=7)
```





# Examples Using the for Statement

Vary the control variable from 20 to 2 in steps of -2.

```
for counter:=20 downto 2 do
```

```
for (counter=20; counter>2; counter-=2)
```





# Contoh :

## Problem:

Anton diminta untuk mencetak kata "Selamat Datang" sebanyak 100 kali di layar.

## Algorithm

1. begin
2. write ('Selamat Datang')
3. write ('Selamat Datang')
4. write ('Selamat Datang')
- .
- .
101. write ('Selamat Datang')
102. end

## Deklarasi

counter : integer

harus ada  
variabel  
pencacah

## Algorithm

1. begin
2. for counter 1 to 100 do
3. write ('Selamat Datang')
4. (endfor)
5. end

Header  
pengulangan for



# Contoh :

## Problem:

Anton diminta untuk mencetak kata "Hello World" sebanyak 100 kali di layar.

### Pseudocode

```
Deklarasi
  counter : integer

Algorithm
begin
for counter 0 to 100 do
  write ('Hello World')
(endfor)
end
```

### Bahasa Pascal

```
Program cetakHello;
Var
  counter: integer;

Begin
  for counter := 0 to 100 do
    begin
      writeln ('Hello World');
    end;
End.
```

### Bahasa C

```
#include <stdio.h>;

int main(){
  //deklarasi
  int counter;

  for(counter=0;counter<100;counter++)
  {
    printf("Hello World\n");
  }
  return 0;
}
```





# 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 6

1. Tuliskan Bahasa Pascal/C dan flowchart untuk mencetak lagu anak ayam menggunakan pengulangan for.
2. Tuliskan Bahasa Pascal/C dan flowchart untuk menghitung nilai factorial dari N. N diinputkan oleh user. Misalnya nilai **masukan N = 5**, maka **outputnya adalah 120** , didapatkan dari hasil perhitungan  $5*4*3*2*1=120$



# Terima Kasih

---

**NEXT:**

Kontrol pengulangan Repeat Until

