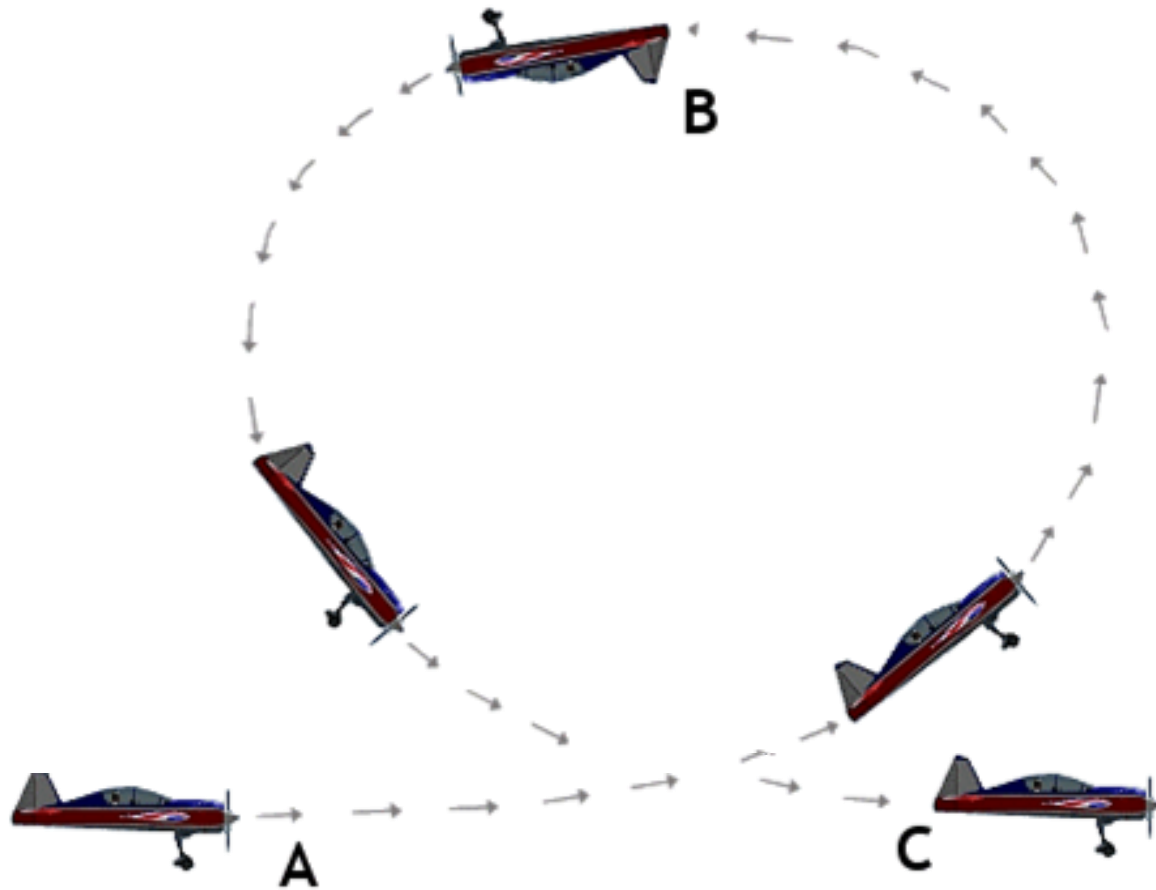


Pengulangan

Yan Mitha Djaksana

Pengulangan....



Struktur Pengulangan

- Kondisi Pengulangan
 - Ekspresi boolean yang harus dipenuhi untuk melaksanakan pengulangan.
- Badan Pengulangan
 - Bagian algoritma yang diulang.

Opsional:

- Inisialisasi
 - Aksi yang dilakukan sebelum pengulangan dilakukan pertama kali.
- Terminasi
 - Aksi yang dilakukan setelah pengulangan selesai dilaksanakan.

Struktur Umum

<inisialisasi>

Awal pengulangan

Badan pengulangan

Akhir pengulangan

<terminasi>

Keterangan:

Awal pengulangan dan akhir pengulangan dinyatakan sebagai kata kunci.

<inisialisasi> dan <terminasi> opsional

Konstruksi Pengulangan

➤ Pernyataan *FOR*

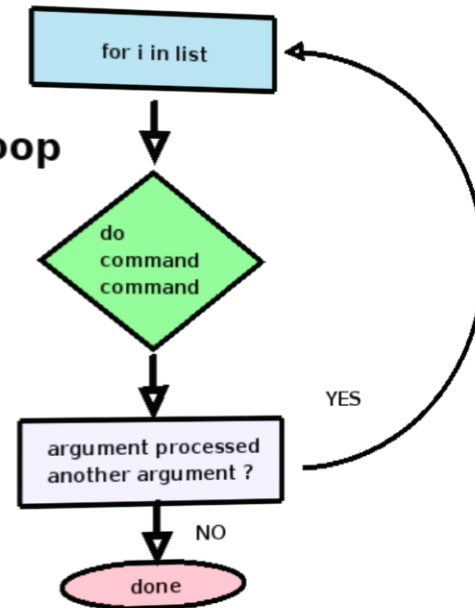
Konstruksi pengulangan
tanpa kondisi (*unconditional
looping*)

➤ Pernyataan *WHILE*

Konstruksi pengulangan
dengan kondisi (*conditional
looping*)

➤ Pernyataan *REPEAT*

for ...in Loop



ERNYATAAN FOR

Karakteristik

- Jumlah pengulangan diketahui atau dapat ditentukan sebelum eksekusi.
- Untuk mencacah berapa kali pengulangan dilakukan, diperlukan sebuah peubah (*variable*) pencacah (*counter*).
- Peubah nilainya bertambah (atau berkurang) satu setiap kali pengulangan dilakukan.
- Jika pengulangan sudah mencapai jumlah yang dispesifikasikan, maka proses pengulangan berhenti.

Bentuk Umum Pernyataan *FOR* menaik

```
for pencacah ← nilai_awal to nilai_akhir do  
    aksi  
endfor
```

Keterangan:

1. Pencacah haruslah tipe data yang memiliki *predecessor* dan *successor*, yaitu integer atau karakter.
2. Aksi dapat berupa satu atau lebih instruksi yang diulang.
3. nilai_awal harus lebih kecil dari nilai_akhir
4. Pencacah diinisialisasi dengan nilai awal. Nilai pencacah secara otomatis bertambah satu setiap kali badan pengulangan dimasuki. Sampai akhirnya nilai pencacah sama dengan nilai akhir.
5. Jumlah pengulangan yang terjadi adalah nilai_akhir – nilai_awal + 1

Contoh 1 : Tulis Hello World 10 x

PROGRAM CetakBanyak_HelloWorld
{Mencetak 'Hello world' sebanyak 10 x}

i adalah pencacah
bertipe integer

DEKLARASI

i: integer {pencacah pengulangan}

Jumlah pengulangan
sudah ditentukan : 10
kali

ALGORITMA:

for i \leftarrow 1 to 10 do {ulangi sebanyak 10 kali}

write('Hello, World')

endfor

Badan pengulangan
hanya berisi satu
pernyataan

Keluaran

```
Hello, world  
Hello, world  
Hello, world  
Hello, world  
Hello, world  
Hello, world  
Hello, world  
Hello, world  
Hello, world  
Hello, world
```

Contoh 2 : Penjumlahan Deret

Program PenjumlahanDeret

{Menjumlahkan deret $1+2+3+...+N$ dengan N adalah bilangan bulat positif. Nilai N dibaca terlebih dahulu}

DEKLARASI

N : integer	<i>{jumlah bilangan}</i>
i : integer	<i>{pencacah banyaknya pengulangan}</i>
jumlah : integer	<i>{pencatatat jumlah nilai}</i>

ALGORITMA:

```
read(N)
jumlah ← 0           {inisialisasi jumlah deret dengan 0}
for i ← 1 to N      {ulangi penjumlahan deret sebanyak N kali}
    jumlah ← jumlah + i
endfor
write(jumlah)
```

Bentuk Umum Pernyataan *FOR* menurun

```
for pencacah ← nilai_akhir downto nilai_awal do  
    aksi  
endfor
```

Keterangan:

1. Pencacah haruslah tipe data yang memiliki *predecessor* dan *successor*, yaitu integer atau karakter.
2. Aksi dapat berupa satu atau lebih instruksi yang diulang.
3. nilai_akhir harus lebih besar atau sama dengan nilai_awal
4. Pencacah diinisialisasi dengan nilai_akhir. Nilai pencacah secara otomatis berkurang satu setiap kali badan pengulangan dimasuki. Sampai akhirnya nilai pencacah sama dengan nilai_awal.
5. Jumlah pengulangan yang terjadi adalah nilai_akhir – nilai_awal + 1

Contoh : Peluncuran Roket

Program PeluncuranRoket

{Hitung mundur peluncuran roket}

DEKLARASI

i : integer

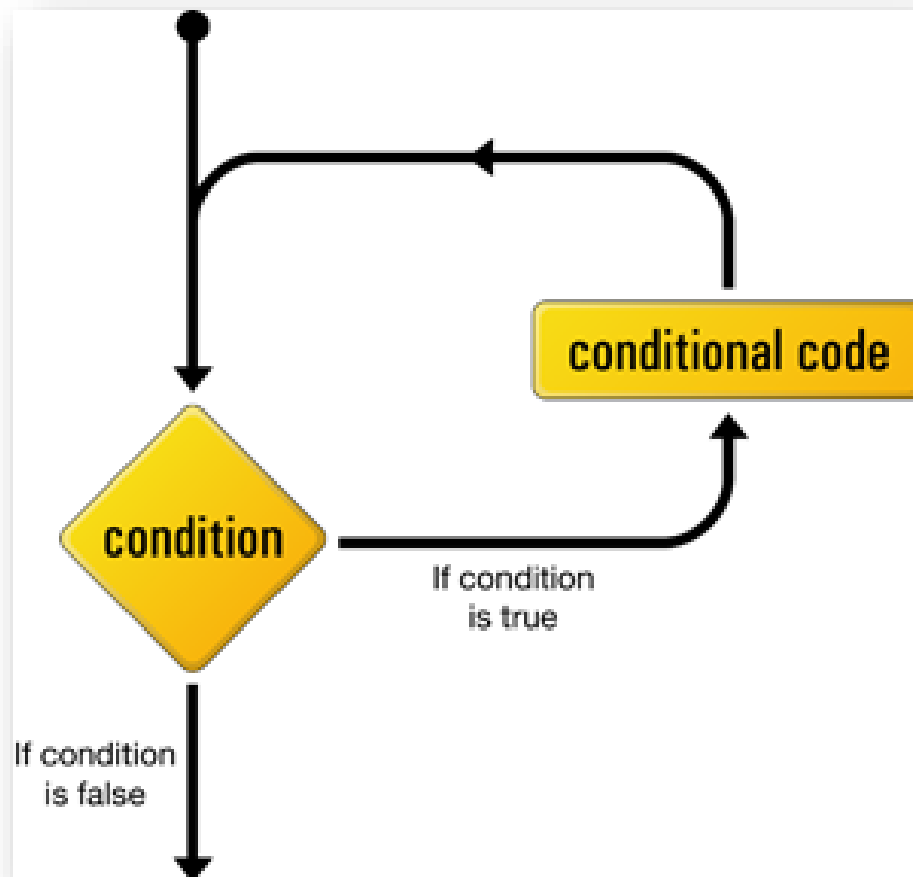
ALGORITMA:

for i \leftarrow 100 downto 0 do

write(i)

endfor

write('GO!') *{Roket meluncur pada saat hitungan 0}*



NYATAAN WHILE

Bentuk Umum

```
while kondisi do  
    aksi  
endwhile
```

Keterangan:

1. Aksi akan dilaksanakan selama kondisi bernilai true
2. Jika kondisi tidak dipenuhi maka aksi tidak akan dimasuki, dan pengulangan selesai.
3. Kondisi di akhir pengulangan (setelah *endwhile*) disebut *loop invariant* yaitu variable kondisi yang nilainya tidak berubah lagi.

Contoh 1: Penjumlahan Deret

PROGRAM PenjumlahanDeret

{ Menjumlahkan deret $1 + 2 + 3 + \dots + N$ dengan N adalah bilangan bukat positif. Nilai N dibaca terlebih dahulu. }

DEKLARASI

N	: <u>integer</u>	<i>{banyaknya suku deret, >0}</i>
i	: <u>integer</u>	<i>{suku deret}</i>
jumlah	: <u>integer</u>	<i>{jumlah deret}</i>

ALGORITMA:

<u>read</u> (N)	<i>{banyaknya suku deret}</i>
jumlah \leftarrow 0	<i>{inisialisasi jumlah deret}</i>
i \leftarrow 1	<i>{suku deret yang pertama}</i>
<u>while</u> i \leq N <u>do</u>	
jumlah \leftarrow jumlah + i	<i>{jumlah deret sekarang}</i>
i \leftarrow i + 1	<i>{suku deret berikutnya}</i>
<u>endwhile</u>	
{i > N}	<i>{kondisi setelah pengulangan}</i>
berhenti}	
write (jumlah)	

Contoh 2 : Hitung Rata-rata

PROGRAM HitungRataRata

{Menghitung rata-rata N buah bilangan bulat yang dibaca dari papan ketik. N>0.}

DEKLARASI

```
N      : integer           {banyaknya data, >0}
x      : integer           {data bilangan bulat yang dibaca dari papan
ketik}
i      : integer           {pencacah banyak data}
jumlah : integer           {pencatat jumlah data}
rerata  : real             {nilai rata-rata seluruh data}
```

ALGORITMA:

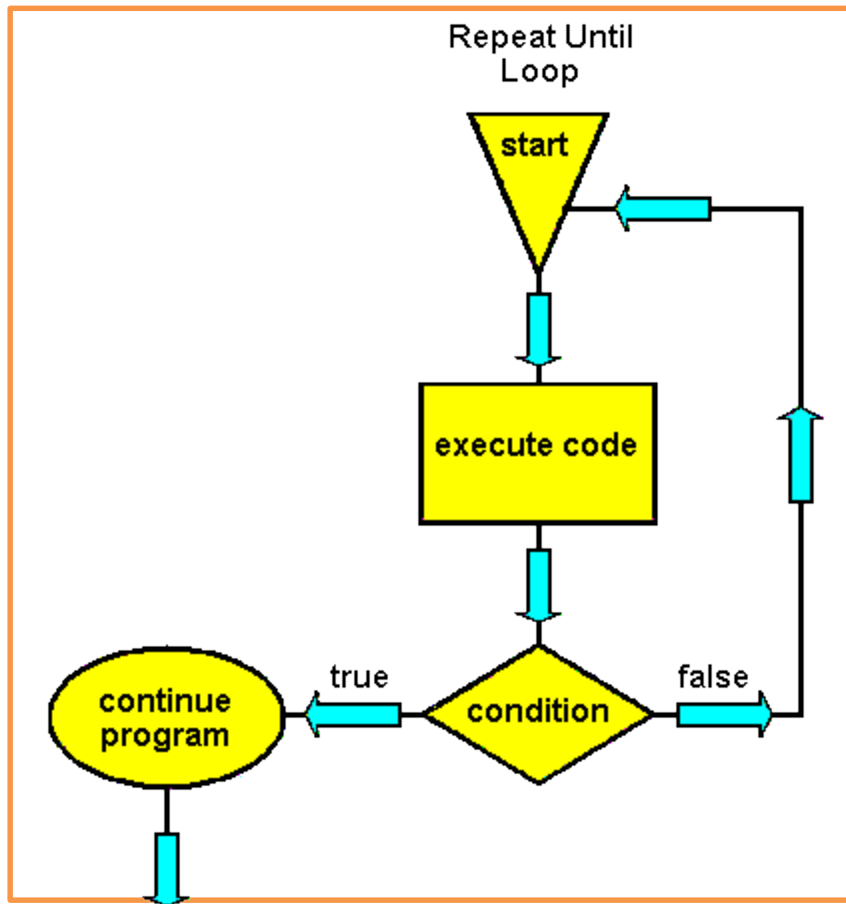
```
read (N)
jumlah ← 0 {inisialisasi}
i ← 1      {inisialisasi pencacah}
while i ≤ N do
    read (x)
    jumlah ← jumlah + x
    i ← i + 1
endwhile
{i > N}
rerata ← jumlah/N
{hitung rata-rata}
```

Contoh 3 : Peluncuran Roket

```
PROGRAM Peluncuran Roket  
{Hitung mundur peluncuran roket}
```

```
DEKLARASI  
    i : integer
```

```
ALGORITMA:  
    i ← 100  
    while i ≥ 0 do  
        write (i)  
        i ← i - 1  
    endwhile  
    {i < 0}  
    write ('Go!')           {roket meluncur pada
```



NYATAAN *REPEAT*

Bentuk Umum

```
repeat  
    aksi  
until kondisi
```

Keterangan:

1. Aksi akan dilaksanakan sampai kondisi bernilai true
2. Jika kondisi dipenuhi maka pengulangan selesai.
3. Karena proses pengulangan suatu saat harus berhenti, maka di dalam badan pengulangan harus ada pernyataan yang mengubah nilai peubah kondisi.

Contoh 1 : Penjumlahan Deret

PROGRAM PenjumlahanDeret

{ Menjumlahkan deret $1 + 2 + 3 + \dots + N$ dengan N adalah bilangan bukat positif. Nilai N dibaca terlebih dahulu.}

DEKLARASI

N : integer {banyaknya suku deret, >0 }
 i : integer {suku deret}
jumlah : integer {jumlah deret}

ALGORITMA:

read (N) {banyaknya suku deret}
jumlah $\leftarrow 0$ {inisialisasi jumlah deret}
 $i \leftarrow 1$ {suku deret yang pertama}
repeat
 jumlah \leftarrow jumlah + i {jumlah deret sekarang}
 $i \leftarrow i + 1$ {suku deret berikutnya}
until $i > N$
write(jumlah)

Contoh 2 : Hitung Rata-Rata

PROGRAM HitungRataRata

{ Menghitung rata-rata N buah bilangan bulat yang dibaca dari papan ketik. N>0.}

DEKLARASI

N	:	<u>integer</u>	{banyaknya data, >0}
x	:	<u>integer</u>	{data bilangan bulat yang dibaca dari papan ketik}
i	:	<u>integer</u>	{pencacah banyak data}
jumlah	:	<u>integer</u>	{pencatat jumlah data}
rerata	:	<u>real</u>	{nilai rata-rata seluruh data}

ALGORITMA:

```
read (N)
jumlah ← 0 {inisialisasi}
i ← 1      {inisialisasi pencacah}
repeat
    read (x)
    jumlah ← jumlah + x
    i ← i + 1
until i > N
rerata ← jumlah/N
write(rerata)
```

Contoh 3 : Peluncuran Roket

```
PROGRAM Peluncuran Roket  
{ Hitung mundur peluncuran roket}
```

```
DEKLARASI  
  i : integer
```

```
ALGORITMA:  
i ← 100  
repeat  
  write (i)  
  i ← i - 1  
until i < 0  
write('Go!')    {roket meluncur pada  
hitungan 0}
```



WHILE ATAU REPEAT?

WHILE

- Kondisi pengulangan diperiksa di awal pengulangan.
- Instruksi dalam badan pengulangan hanya dapat dilaksanakan bila pengetesan kondisi menghasilkan nilai true.
- Badan pengulangan mungkin tidak akan pernah dilaksanakan bila kondisi pengulangan pertama kali bernilai false.

REPEAT

- Kondisi pengulangan diperiksa pada akhir pengulangan.
- Instruksi dalam badan pengulangan dilaksanakan dulu, barulah pengetesan kondisi dilakukan.
- Badan pengulangan dilaksanakan paling sedikit satu kali.

WHILE atau REPEAT? (1)

1. Penjumlahan Deret

{Menghitung jumlah seluruh nilai dari sejumlah data bilangan. Nilai bilangan (x) dibaca dari keyboard. Akhir pembacaan adalah bila nilai x yang dibaca = 0}

WHILE atau REPEAT? (2)

2. Simulasi Menu Program

{menampilkan menu, membaca pilihan menu,
dan menampilkan nomor menu yang dipilih
pengguna, sampai pengguna memilih menu
nomor 5 (keluar)}

WHILE atau REPEAT? (3)

3. Penghitungan Upah Karyawan

{Menentukan upah karyawan. Masukan yang dibaca dari keyboard adalah nama karyawan, golongan, dan jumlah jam kerja. Keluaran program adalah nama karyawan dan upahnya}