

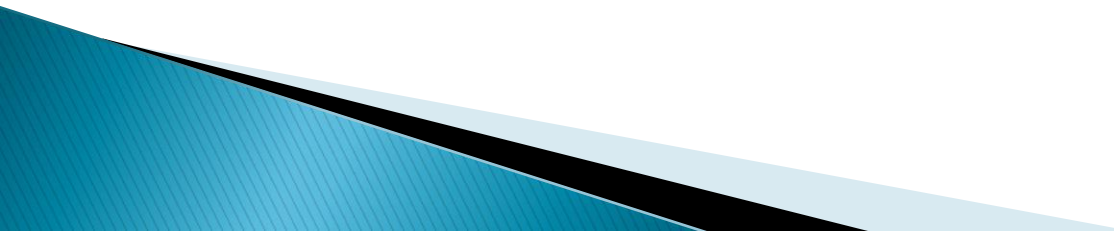
# PERULANGAN

For to do, While do dan Repeat until

# Struktur Perulangan (SP)

- ▶ SP terdiri dari
  - Kondisi pengulangan
    - Ekspresi *boolean* yang harus dipenuhi untuk melaksanakan pengulangan
  - Badan (*body*) pengulangan
    - Bagian algoritma yang diulang
- ▶ Struktur perulangan disertai :
  - Inisialisasi, yaitu aksi yang dilakukan sebelum pengulangan dilakukan pertama kali
  - Terminasi, aksi yang dilakukan setelah pengulangan selesai dilaksanakan

# Notasi struktur Perulangan

- ▶ Struktur FOR
  - ▶ Struktur WHILE
  - ▶ Struktur REPEAT
- 

# Struktur FOR

- ▶ Digunakan untuk menghasilkan pengulangan sejumlah (n) kali yang dispesifikasikan.
- ▶ Jumlah pengulangan diketahui (dapat ditentukan) sebelum eksekusi.
- ▶ Variabel pencacah
  - Nilainya selalu bertambah setiap kali perulangan dilakukan.
  - Jika nilainya sudah mencapai jumlah yang dispesifikasikan, maka proses perulangan akan berhenti
- ▶ Bentuk umum for :
  - Menaik (*ascending*)
  - Menurun (*descending*)

FOR menaik

```
for pencacah  $\leftarrow$  nilai_awal to nilai_akhir do  
    Aksi  
endfor
```

FOR menurun

```
for pencacah  $\leftarrow$  nilai_awal downto nilai_akhir do  
    Aksi  
endfor
```



# Struktur WHILE

- ▶ Bentuk umum struktur WHILE  
**while** kondisi **do**  
    aksi  
**endwhile**
- ▶ Keterangan :
  - Aksi akan dilakukan selama kondisi bernilai *true*.
  - Jika kondisi bernilai *false*, badan pengulangan tidak akan dilaksanakan, yang berarti perulangan selesai.

- ▶ Yang harus diperhatikan → pengulangan harus berhenti.
- ▶ Supaya kondisi bernilai *false*,
  - Di dalam badan pengulangan harus ada instruksi yang mengubah nilai peubah kondisi.

# Contoh 1

## Algoritma Cetak\_banyak\_Hallo

{Mencetak 'Hello Word' sebanyak 10 kali}

### DEKLARASI

k : integer { pencacah pengulangan }

### DESKRIPSI

k  $\leftarrow$  1

**while** k  $\leq$  10 **do**

    write ('Hello Word')

    k  $\leftarrow$  k + 1

**endwhile**

{ kondisi akhir pengulangan : k > 10 }



# Struktur REPEAT

- ▶ Bentuk umum struktur REPEAT  
    **repeat**  
        aksi  
    **Until** kondisi
- ▶ Keterangan :
  - Aksi akan dilakukan sampai kondisi *boolean* bernilai *true*.
  - Jika kondisi bernilai *false*, badan pengulangan akan dilaksanakan, yang berarti perulangan selesai jika kondisi bernilai *true*

# Contoh 2

## Algoritma Cetak\_banyak\_Hallo

{Mencetak 'Hello Word' sebanyak 1 kali}

### DEKLARASI

k : integer { pencacah pengulangan }

### DESKRIPSI

k  $\leftarrow$  6

**repeat**

    write ('Hello Word')

    k  $\leftarrow$  k + 1

**until** k > 5

{ kondisi akhir pengulangan : k > 5 }

# Latihan

1. Buat algoritma untuk menampilkan nilai bilangan ganjil dengan input  $x$  (masukan nilai sembarang) sampai ke  $n$  kali (banyaknya pengulangan) menggunakan for to do. dengan catatan nilai awal dalam for jangan dijadikan output,  
contoh1  $x=4$   $n=2$  outputnya 5 dan 7  
contoh2  $x=7$   $n=2$  outputnya 7 dan 9

# FOR

▶ Pascal  
for pencacah:=a to b do  
begin  
aksi1;  
aksi2;  
.  
.  
.  
End;

▶ C  
for (pencacah=a;  
a<=b; a++)  
{  
aksi1;  
aksi2;  
.  
.  
.  
}

# WHILE

▶ PASCAL  
while kondisi do  
begin  
aksi1;  
aksi2;  
.  
.  
.  
end

▶ C  
while kondisi  
{  
aksi1;  
aksi2;  
.  
.  
.  
}

# REPEAT

## ▶ PASCAL

**repeat**

aksi1;

aksi2;

.

.

.

**until kondisi;**

## ▶ C

**do {**

aksi1;

aksi2;

.

.

.

**} while**

# Contoh 1 -- FOR

(Algoritma menghitung deret  $1 + 2 + 3 + \dots + N$ )

## Algoritma PENJUMLAHAN\_DERET

{ Menjumlahkan deret

$1 + 2 + 3 + \dots + N$

dengan N adalah bilangan bulat positif yang dibaca dari piranti masukan.

Dan mencetak jumlah deret }

## DEKLARASI :

N : integer {banyak suku deret}

k : integer {suku deret}

jumlah : integer {jumlah deret}

## DESKRIPSI :

read(N)

jumlah  $\leftarrow$  0

for k  $\leftarrow$  1 to N

    jumlah  $\leftarrow$  jumlah + k

endfor

write (jumlah)



# PASCAL

# C

Program PENJUMLAHAN\_DERET

```
var
  N : integer;
  k : integer;
  Jumlah : integer;

begin
  write('Berapa N?');
  readln(N);
  jumlah:=0;
  for k:=1 to N
    begin
      jumlah:=jumlah + k;
    end;
  writeln('Jumlah deret = ',
    jumlah);
end
```

```
#include <stdio.h>
```

```
main()
{
  int N, k, jumlah;
  printf ("Berapa N? ");
  scanf("%d",&N);
  jumlah=0;
  for (k=1; k<N; k++)
  {
    jumlah=jumlah + k;
  }
  printf ("Jumlah deret = %d",
    jumlah)
}
```



# Contoh 1 -- WHILE

(Algoritma menghitung deret  $1 + 2 + 3 + \dots + N$ )

## Algoritma PENJUMLAHAN\_DERET

{ Menjumlahkan deret

$1 + 2 + 3 + \dots + N$

dengan N adalah bilangan bulat positif yang dibaca dari piranti masukan.

Dan mencetak jumlah deret }

## DEKLARASI :

N : integer {banyak suku deret}

k : integer {suku deret}

jumlah : integer {jumlah deret}

## DESKRIPSI :

read(N)

jumlah  $\leftarrow$  0

k  $\leftarrow$  1

**while** k < N **do**

    jumlah  $\leftarrow$  jumlah + k

    k  $\leftarrow$  k + 1

**endwhile**

**write** (jumlah)

# PASCAL

# C

## Program PENJUMLAHAN\_DERET

```
var
  N : integer;
  k : integer;
  Jumlah : integer;

begin
  write('Berapa N?');
  readln(N);
  jumlah:=0;   k:=1;
  while k <= N do
    begin
      jumlah:=jumlah + k;
      k:=k+1;
    end;
  writeln('Jumlah deret = ', jumlah);
end
```

```
#include <stdio.h>
```

```
main()
{
  int N, k, jumlah;
  printf("Berapa N? ");
  scanf("%d",&N);
  jumlah=0;
  k=1;
  while (k <= N)
  {
    jumlah=jumlah + k;
    k++;
  }
  printf("Jumlah deret = %d", jumlah)
}
```