

# **STRUKTUR PERULANGAN**

## **Repeat - Until**

**Temu 10**

# Perulangan: Repeat - Until

- Bentuk umum :

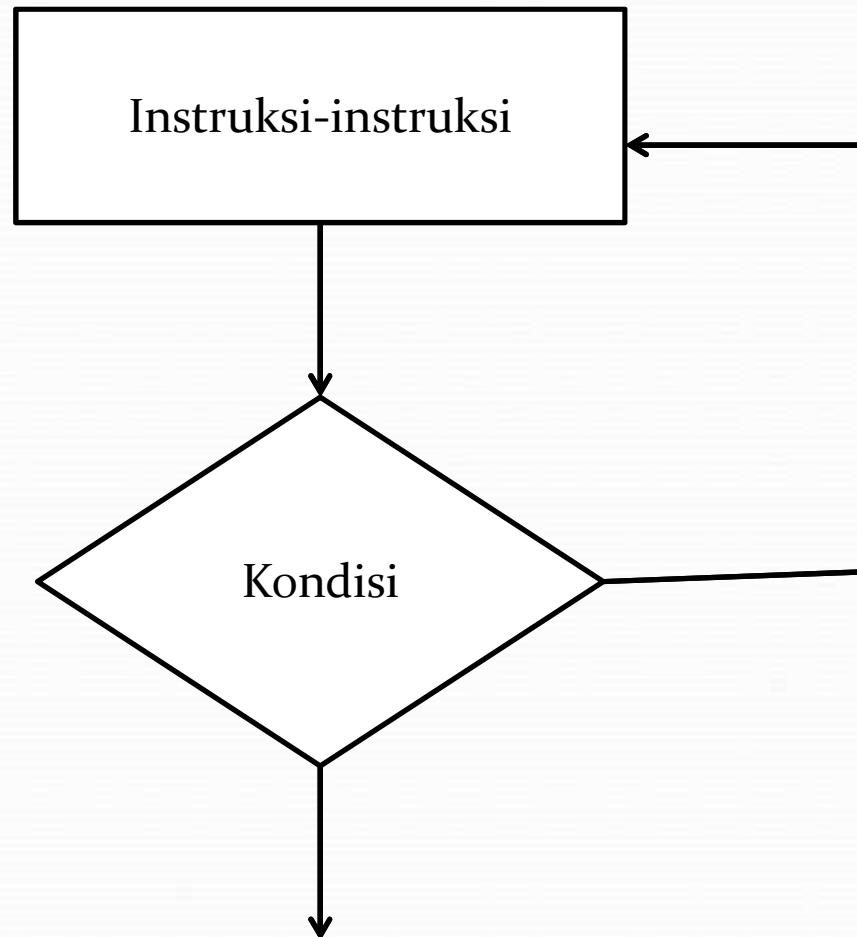
**Repeat**

.....

instruksi

.....

**Until ( kondisi )**



# Keterangan:

Intruksi ( atau runtunan instruksi) akan dilaksanakan berulang kali sampai kondisi bernilai true, jika kondisi bernilai false maka pengulangan masih terus dilakukan.

# Contoh :

- Algoritma Cetak\_Angka  
{mencetak 1, 2, ..., 10 ke piranti keluaran}

Deklarasi :

K: integer

Deskripsi :

K  $\leftarrow$  1 {inisialisasi}

repeat

    write (k)

    k  $\leftarrow$  k + 1

until k > 10

# Contoh 1

- Mencetak kata ‘Unindra’ sebanyak 10 kali

Program Cetakbanyak\_Unindra

Deklarasi

i : integer {Pencacah pengulangan}

Algoritma

i  $\leftarrow$  1 {inisialisasi}

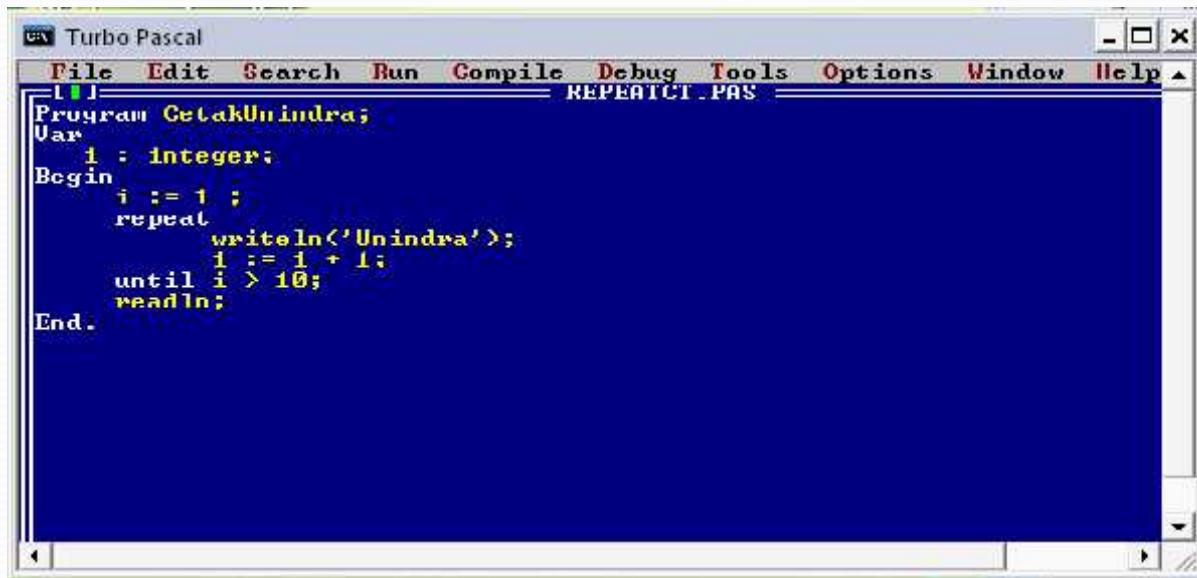
repeat

    write('Unindra')

    i  $\leftarrow$  i + 1

until i > 10

# Contoh 1 Pada Pascal



The screenshot shows the Turbo Pascal IDE interface. The title bar reads "Turbo Pascal" and the file name is "REPEATCT.PAS". The menu bar includes File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The main window displays the following Pascal code:

```
Program CetakUnindra;
Var
  i : integer;
Begin
  i := 1 ;
  repeat
    writeln('Unindra');
    i := i + 1;
  until i > 10;
  readln;
End.
```



The screenshot shows the Turbo Pascal terminal window. The title bar reads "Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International". The window displays the repeated output of the word "Unindra" ten times, corresponding to the loop in the code above.

```
Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International
Unindra
```

## Contoh 2

- Mencetak angka 1 sampai N

Program Cetak1sampaIN

Deklarasi :

N, i : integer

Algoritma :

read(N)

i  $\leftarrow$  1

repeat

    write(i)

    i  $\leftarrow$  i + 1

until i > N

## Contoh 2 Pada Pascal

The screenshot shows the Turbo Pascal IDE interface. The menu bar includes File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The title bar displays "NONAME00.PAS". The code window contains the following Pascal program:

```
Program CetakisampaiN;
Var
  N, i : integer;
Begin
  write('Berapa Banyak Bilangan ?');
  readln(N);
  i := 1;
  repeat
    write(i, ' ');
    i := i + 1;
  until i > n;
  readln;
End.
```

The status bar at the bottom shows "13:5" and a set of keyboard shortcuts: F1 Help, F2 Save, F3 Open, Alt+F9 Compile, F9 Make, Alt+F10 Local menu.

The screenshot shows the Turbo Pascal terminal window. The title bar displays "Turbo Pascal Version 7.0 Copyright <c> 1993.92 Borland International". The window displays the following text:

```
Berapa Banyak Bilangan ?20
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

# Contoh 3

Menghitung nilai rata-rata N buah bilangan bulat

Program HitungRataRata

Deklarasi

n : integer	{banyaknya data, > 0}
x : integer	{data bilangan bulat}
i : integer	{pencacah banyak data}
jumlah : integer	{ pencatat jumlah data}
rerata : real	{nilai rata-rata seluruh data}

Algoritma

```
Read(n)
Jumlah ← 0
i ← 1
repeat
    read(x)
    jumlah ← jumlah + x
    i ← i + 1
until i > n
rerata ← jumlah/n
write(rerata)
```

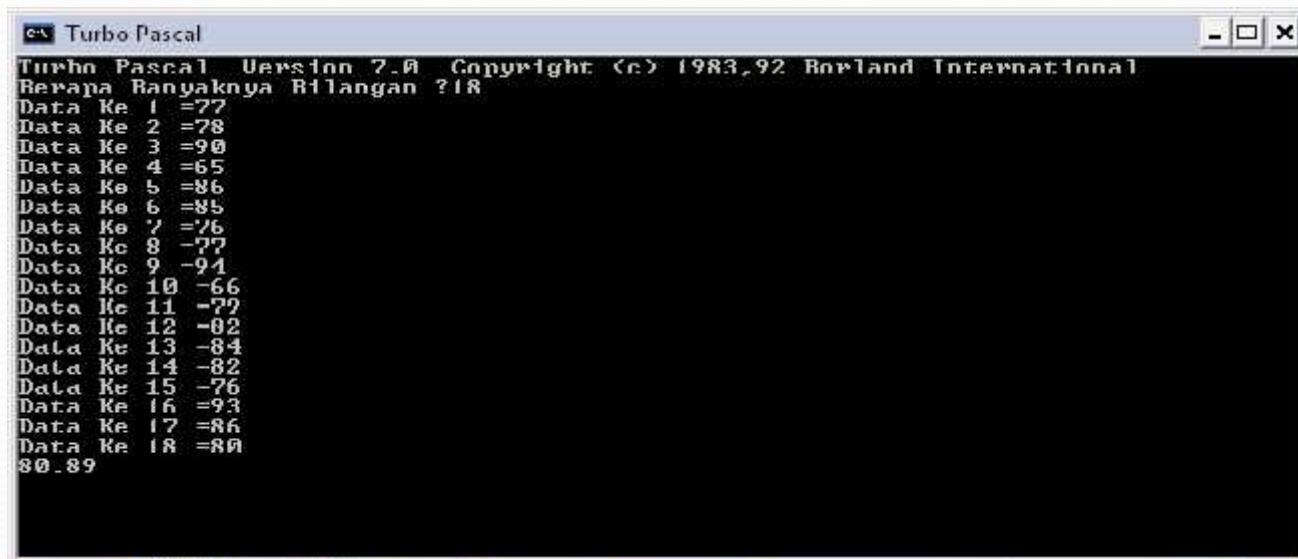
# Contoh 3# Pada Pascal



The screenshot shows the Turbo Pascal IDE interface. The menu bar includes File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The title bar displays 'NONAME00.PAS'. The code window contains the following Pascal program:

```
Program HitungRataRata;
Var
  n, x, i, jumlah : integer;
  rerata : real;
Begin
  write('Berapa Banyaknya Bilangan ?');
  readln(n);
  jumlah := 0;
  i := 1;
  repeat
    write('Data Ke ',i,' =');
    readln(x);
    jumlah := jumlah + x;
    i := i + 1;
  until i > n;
  rerata := jumlah/n;
  write(rerata : 0 : 2);
  readln;
End.
```

The status bar at the bottom shows the time as 17:26 and various keyboard shortcuts like F1 Help, F2 Save, F3 Open, Alt+F9 Compile, F9 Make, and Alt+F10 Local menu.



The screenshot shows the terminal window of the Turbo Pascal IDE displaying the execution results of the program. The output shows the user inputting 18 numbers and the program calculating their average.

```
Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International
Berapa Banyaknya Bilangan ?18
Data Ke 1 =77
Data Ke 2 =78
Data Ke 3 =90
Data Ke 4 =65
Data Ke 5 =86
Data Ke 6 =85
Data Ke 7 =76
Data Ke 8 =77
Data Ke 9 =94
Data Ke 10 =66
Data Ke 11 =72
Data Ke 12 =82
Data Ke 13 =84
Data Ke 14 =82
Data Ke 15 =76
Data Ke 16 =93
Data Ke 17 =86
Data Ke 18 =80
80.89
```

## Contoh 4

Algoritma Peluncuran roket dengan hitung mundur

Program PeluncuranRoket ;

Deklarasi

i : integer

Algoritma

i ← 100

Repeat

    write(i)

    i ← i - 1

until i < 0

    write('Go!')

# Contoh 4# Pada Pascal

The screenshot shows the Turbo Pascal 7.0 IDE interface. The menu bar includes File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The title bar displays "REPROKET.PAS". The code window contains the following Pascal program:

```
Program PeluncuranRoket;
Var
    i : integer;
Begin
    i := 100;
    Repeat
        write(i,' ');
        i := i - 1;
    Until i < 0;
    write('GO !');
    readln;
End.
```

The status bar at the bottom shows the time as 7:19 and various keyboard shortcuts: F1 Help, F2 Save, F3 Open, Alt+F9 Compile, F9 Make, and Alt+F10 Local menu.

The screenshot shows the Turbo Pascal 7.0 command window. It displays the copyright information for Turbo Pascal Version 7.0, followed by a sequence of numbers from 100 down to 1, separated by spaces, and ending with "GO !".

```
Turbo Pascal Version 7.0 Copyright (c) 1983-92 Borland International
100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 7
4 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48
47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 GO !
```

# WHILE atau REPEAT

Kapan menggunakan WHILE dan kapan menggunakan REPEAT ?

- Gunakan konstruksi while pada kasus yang mengharuskan terlebih dahulu pemeriksaan kondisi objek sebelum objek tersebut dimanipulasi
- Gunakan konstruksi repeat pada kasus yang terlebih dahulu memanipulasi objek baru kemudian memeriksa kondisi objek tersebut.

# Contoh

- Dibaca sejumlah nilai bilangan bulat  $x$  dari proses masukan.
- Diminta menghitung jumlah seluruh nilai pecahannya ( $1/x$ )
- Akhir proses penjumlahan adalah bila nilai  $x$  yang dibaca = 0. Sebagai contoh, jika nilai  $x$  yang dibaca berturut-turut adalah 3, 5, 8, 0, maka jumlah nilai pecahannya adalah  $1/3 + 1/5 + 1/8 = 0.65833$
- Misalkan jumlah deret tersebut adalah  $S$ .
- Sebelum penjumlahan  $S$  diinisialisasikan dengan 0
- Setiap kali membaca  $x$ , harus diperiksa apakah  $x = 0$ , jika  $x = 0$  maka proses pengulangan berhenti, tetapi jika tidak maka penjumlahan  $S$  dengan  $1/x$  :  $S \leftarrow S + 1/x$

# WHILE benar

Program PenjumlahanDeretPecahan

Deklarasi

```
x : integer      {data bilangan bulat yang dibaca}  
s : real         {jumlah deret}
```

Algoritma

```
s ← 0      {inisialisasi jumlah deret}  
read(x)  
while x ≠ 0 do  
    s ← s + 1/x  
    read(x)  
endwhile  
{x=0}  
write(s)
```

# REPEAT Salah

Program PenjumlahanDeretPecahan

Deklarasi

x : integer

s : real

Algoritma

s  $\leftarrow$  0

repeat

    read(x)

    s  $\leftarrow$  s + 1/x

until x = 0

write(s)

# Kesimpulan

- Penggunaan struktur REPEAT pada contoh diatas berakibat fatal bila x yang pertama kali dibaca berharga nol sebab pembagian dengan nol (1/0) tidak terdefinisi (error)