

BAB V
PENGULANGAN / LOOPING

Kompetensi Dasar :

1. Mahasiswa memahami kontrol pengulangan pada bahasa Pascal.

Indikator :

1. Mahasiswa mampu membedakan penggunaan tiap-tiap kontrol pengulangan sesuai dengan kebutuhan kasus.
2. Mahasiswa mampu mendeteksi kebutuhan dasar dalam pendukung kontrol pengulangan yaitu nilai awal dan akhir sebagai pembatas perulangan.
3. Mahasiswa dapat membaca proses dan alur dari sebuah perulangan.
4. Mahasiswa mampu membuat program pengulangan atau looping.

Pengulangan digunakan untuk menjalankan satu atau beberapa pernyataan sebanyak beberapa kali. Dengan kata lain, pengulangan memungkinkan kita untuk menjalankan beberapa pernyataan hanya dengan menuliskan pernyataan tersebut satu kali saja.

Iteration Statement	Bahasa Pascal	Bahasa C
Repeat/do loop: Evaluasi kondisi dilakukan di bagian akhir.	<pre>repeat (statement); ... (statement); until (kondisi);</pre> <p>Contoh: <i>i</i> := 1; repeat <i>i</i> := <i>i</i> + 2; writeln(<i>i</i>); until <i>i</i> >=10;</p>	<pre>do (statement) while (kondisi);</pre> <p>Contoh: <i>i</i> = 1; do { <i>i</i> = <i>i</i> + 2; printf("%i\n", <i>i</i>); } while (<i>i</i><10);</p>
while do: Evaluasi kondisi dilakukan di bagian awal	<pre>while (kondisi) do (statement)</pre> <p>Contoh: <i>i</i> := 1; while (<i>i</i><10) do begin <i>i</i> := <i>i</i> + 2; writeln(<i>i</i>); end;</p>	<pre>while (kondisi) do (statement)</pre> <p>Contoh: <i>i</i> = 1; while (<i>i</i><10) { <i>i</i> := <i>i</i> + 2; printf("%i\n", <i>i</i>); }</p>
for..loop: Perulangan dengan increment nilai	<pre>for counter := lower to upper do (statement)</pre> <p>Atau</p> <pre>for counter := upper downto lower do (statement)</pre> <p>Contoh: for <i>i</i> := 1 to 10 do writeln(<i>i</i>);</p>	<pre>for (init;exp1;exp2) (statement)</pre> <p>Contoh: for (<i>i</i>=1;<i>i</i><=10;<i>i</i>++) printf("%i\n", <i>i</i>);</p>
	<pre>for i := 1 to 10 do begin write(<i>i</i>); for j := 10 downto i do write(<i>j</i>); writeln; end;</pre>	<pre>for (i=1;i<=10;i++) { printf("%i", <i>i</i>); for(j=10;j>=i;j--) printf("%i", <i>j</i>); printf("\n"); }</pre>

A. Pernyataan for

Pernyataan pengulangan yang paling sering digunakan adalah for. Pernyataan ini digunakan jika kita telah mengetahui berapa kali kita akan mengulang satu atau beberapa pernyataan. Bentuk umum pernyataan for adalah sebagai berikut:

for pencacah := nilai_awal to nilai_akhir do
Begin
(pernyataan-pernyataan yang akan diulang)
end;

Sekedar tambahan, blok berisi pernyataan-pernyataan yang harus diulang sering disebut juga kalang (*loop*).

1. Pernyataan For Positif

Contoh :

Perulangan positif untuk satu statement :

uses wincrt;
var
i : integer;
begin
for i := 1 to 5 do
writeln ('Belajar Pemrograman');
readln;
end.

Maka bila program diatas dicompile

Hasilnya :

Belajar pemrograman
Belajar pemrograman
Belajar pemrograman
Belajar pemrograman
Belajar pemrograman

Penjelasan : Berati statemen Belajar pemrograman akan diulang sebanyak 5 kali yaitu dengan

menghitung nilai i dari i ke 1 sampai nilai i terakhir yaitu i ke 5.

2. Pernyataan For Negatif

Perulangan negatif adalah perulangan dengan menghitung (counter) dari besar ke kecil.

Statement yang digunakan adalah **For-DownTo-Do**

contoh:

<code>uses wincrt;</code>
<code>var</code>
<code> i : integer;</code>
<code>begin</code>
<code> for i := 10 downto 1 do</code>
<code> write (i:3);</code>
<code>end.</code>

Hasil :

10987654321

B. Pernyataan for bersarang

Kita juga bisa menuliskan pernyataan for di dalam pernyataan for. Kontruksi semacam ini sering disebut dengan pernyataan for bersarang. Perhatikan contoh berikut:

<code>for x := 1 to 3 do</code>
<code> for y :=1 to 2 do</code>
<code> writeln (x, ' ' y);</code>

Kalang for yang luar (dengan pencacah variabel x) akan menjalankan kalang yang dalam (dengan pencacah

y) sebanyak 3 kali. Dan pada setiap pengulangan di layar akan dituliskan nilai x dan y. Berikut ini adalah keluaran dari program di atas:

```
1 1
1 2
2 1
2 2
3 1
3 2
```

Pada saat x bernilai 1, y diulang sebanyak 2 kali. Jadi pada layar akan tertulis 1 1 dan 1 2. Demikian juga pada saat x bernilai 2, y diulang sebanyak 2 kali. Jadi pada layar akan tertulis 2 1 dan 2 2. Hal yang sama terjadi pada saat x bernilai 3, y diulang sebanyak 2 kali. Sehingga pada layar tertulis 3 1 dan 3 2.

C. Pernyataan while-do

Pernyataan pengulangan ini biasanya digunakan bila kita belum tahu pasti berapa banyak kita akan mengulang pernyataan-pernyataan. Berakhirnya pengulangan ini ditentukan oleh suatu kondisi. Bila kondisi sudah tidak terpenuhi maka pengulangan akan berakhir. Dengan kata lain, selama kondisi masih terpenuhi, pengulangan akan terus dilakukan. Bentuk umum pernyataan pengulangan while adalah sebagai berikut:

<code>while kondisi do</code>
<code>begin</code>
<code> (pernyataan-pernyataan yang akan diulang)</code>
<code>end;</code>

kondisi sendiri merupakan suatu ekspresi bertipe Boolean, artinya hanya dapat bernilai benar (true) atau salah (false).

Contoh:

while (x > 0) do
begin
x := x - 1;
y := y - 1;
end;

Program ini akan mengulang dua pernyataan :

(x := x - 1) dan (y := y - 1)

selama nilai x masih positif.

D. Perulangan While - Do tersarang

Perulangan While - Do tersarang (nested While - Do) merupakan perulangan While - Do yang satu di dalam perulangan While - Do yang lainnya.

Contoh :

uses wincrt;
var
a, b : integer;
begin
clrscr;
a:=1;
b:=1;
while a < 4 do {loop selama a masih lebih kecil dari 4}
begin
a := a + 1;
while b < 3 do {loop selama b masih lebih kecil dari 3}
begin
write (a:3,b:2);
b:=b+1;
end;
end;
readln;
end.

E. Pernyataan `repeat..until`

Pernyataan pengulangan ini hampir sama dengan pernyataan pengulangan `while`, dan biasanya digunakan bila jumlah pengulangan belum dapat ditentukan pada saat program ditulis.

Perbedaan pernyataan `repeat..until` dan `while` terletak pada letak pengecekan kondisi. Jika pada pernyataan `while`, kondisi dicek pada awal kalang, sedangkan pada pernyataan `repeat..until`, kondisi dicek pada akhir kalang.

Perbedaan yang lain, bila pernyataan `while` mengulang pernyataan selama kondisi masih terpenuhi, pernyataan `repeat..until` mengulang pernyataan selama kondisi belum terpenuhi. Bentuk umum pernyataan `repeat..until` adalah sebagai berikut:

<code>repeat</code>
<code> (pernyataan-pernyataan yang akan diulang)</code>
<code>until kondisi</code>

Contoh :

<code>repeat</code>
<code> x := x - 1;</code>
<code> y := y - 1;</code>
<code>until (x <= 0);</code>

Program ini akan mengulang dua pernyataan (`x := x - 1`) dan (`y := y - 1`) sampai nilai `x` lebih kecil atau sama dengan 0.

F. Repeat - Until tersarang

Repeat - Until tersarang adalah suatu perulangan

Repeat - Until yang satu berada didalam perulangan

Repeat - Until yang lainnya.

Contoh :

<code>uses wincrt;</code>
<code>var</code>
<code>a,b,c : real;</code>
<code>begin</code>
<code> writeln('=====');</code>
<code> writeln(' sisi a sisi b sisi c ');</code>
<code> writeln('=====');</code>
<code> a := 1;</code>
<code> repeat { perulangan luar }</code>
<code> b := 0;</code>
<code> repeat { perulangan dalam }</code>
<code> c := sqrt (a * a + b * b);</code>
<code> writeln (a:6:2, b:9:2, c:9:2);</code>
<code> b := b + 5;</code>
<code> until b > 25; { berhenti jika b lebih besar dari 5 untuk</code>
<code> perulangan dalam }</code>
<code> a := a + 1;</code>
<code> until a > 3; { berhenti jika a lebih besar dari 3 untuk</code>
<code> perulangan luar }</code>
<code> writeln('=====');</code>
<code>end.</code>

G. Rangkuman

1. Pengulangan digunakan untuk menjalankan satu atau beberapa pernyataan sebanyak beberapa kali. Dengan kata lain, pengulangan memungkinkan kita untuk menjalankan beberapa pernyataan hanya dengan menuliskan pernyataan tersebut satu kali saja.
2. Ada beberapa pernyataan perulangan :
 - a. Pernyataan for
 - b. Pernyataan for bersarang
 - c. Pernyataan while do
 - d. Pernyataan while do bersarang
 - e. Pernyataan repeat until
 - f. Pernyataan repeat until bersarang

H. Tugas Praktikum (dikumpulkan akhir sesi)

Buatlah algoritma dan flowchart untuk permasalahan dibawah ini.

1. Buat deret hitung 3, 7,, 15

program Deret_Hitung;
uses wincrt;
var
i, n, deret : integer;
begin
write ('Batas pengulangan mencetak deret = '); readln (n);
deret := 3;
for i := 1 to n do
begin
write (deret);
write (' ');
deret := deret + 4;
end;
end.

2. Buat deret kuadrat 1, 4, 9,

program Deret_Kuadrat;
uses wincrt;
var
i, n, deret : integer;
begin
write ('Batas pengulangan mencetak deret = '); readln (n);
deret := 0;
for i := 1 to n do
begin
deret := deret + 1;
write (sqr(deret));
write (' ');
end;
end.

I. Tugas Rumah (dikumpulkan pada pertemuan berikutnya)

Buatlah algoritma, flowchart, dan program untuk permasalahan deret ukur 3, 9, 27,

Daftar Pustaka

Pranata, Anthony., *Algoritma Pemrograman*, J&J Learning, Yogyakarta

Sutopo, Anton, *Pemrograman Komputer Dasar*, 2005, Yogyakarta

Hadi S., Chendra, *Memahami Struktur dan Elemen pada bahasa Pemrograman*, 2003, www.ilmukomputer.com