



Pemrograman 1

Pertemuan 10 : Repeat - Until

Repeat - Until

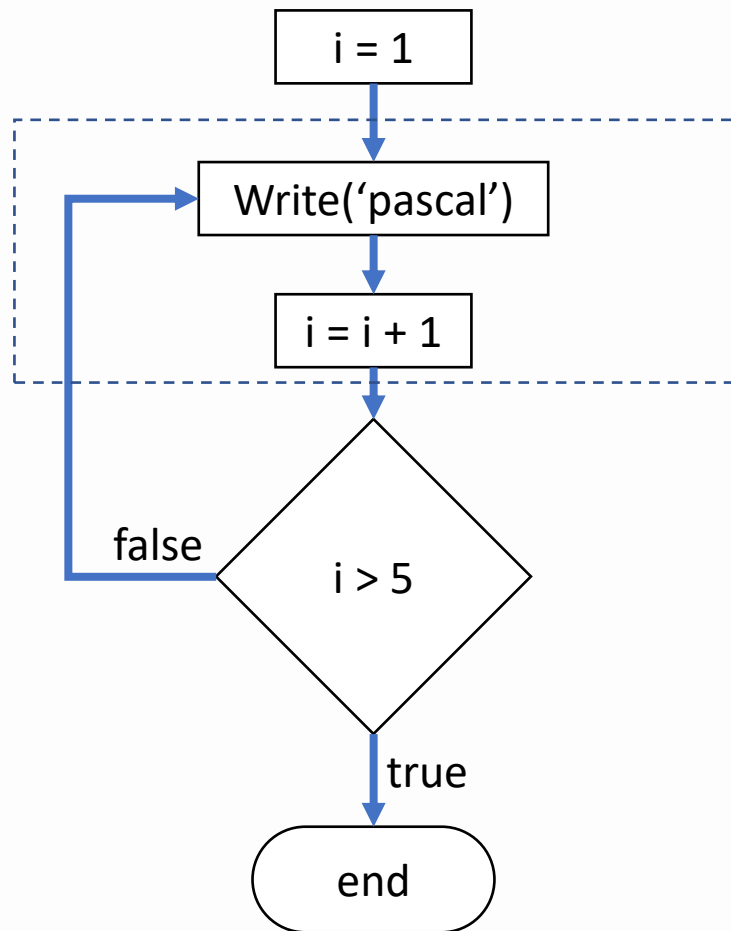
- Pada dasarnya, perulangan **REPEAT UNTIL** mirip seperti perulangan **WHILE DO**, dimana kita akan melakukan pengecekan dalam setiap iterasi apakah nilai variabel counter masih dipenuhi atau tidak.
- Perbedaanya, pada **REPEAT UNTIL**, pemeriksaan kondisi ini dilakukan di akhir perulangan, bukan di awal seperti **WHILE DO**.
- Format Penulisan perulangan Repeat – Until:

```
REPEAT
  begin
    (kode program yang ingin diulang disini...)
    (kode program yang untuk mengubah condition..)
  end;
UNTIL (condition)
```

- Dalam Bahasa sehari-hari kita dapat menyatakan bentuk diatas sebagai berikut:

Ulang Pernyataan Sampai kondisi benar

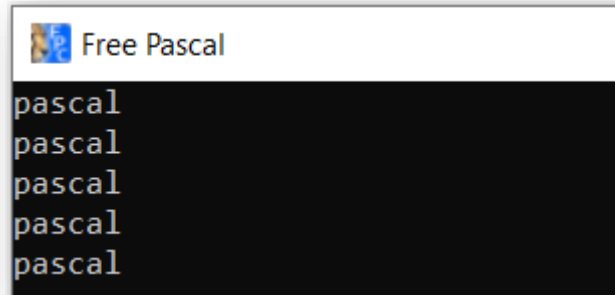
Struktur Repeat – Until dalam flowchart



- Flowchart tersebut menunjukkan struktur perulangan repeat-until
- Inisialisasi : (**i = 1**)
- Badan (body) Perulangan:
 - Write('pascal')
 - $i = i + 1$
- Kondisi perulangan (**$i > 5$**), artinya sebelum i bernilai > 5 (true) maka perulangan tidak akan berhenti

Contoh Program

```
1  program repeat_until;
2  uses crt;
3  var
4    i: integer;
5  begin
6    clrscr;
7    i:= 0;
8    repeat
9      begin
10         writeln('pascal');
11         i:= i + 1;
12       end;
13     until i = 5;
14     readln;
15 end.
```



```
Free Pascal
pascal
pascal
pascal
pascal
pascal
```

Program disamping menghasilkan output kata pascal sebanyak 5x, hal ini terjadi karena inisialisasi yang dilakukan dimulai dari nol, perhatikan kembali kondisi yang menyebabkan repeat-until berhenti, yaitu jika kondisi bernilai **true**

Kondisi True pada Repeat-Until

- Pada dasarnya, perulangan Repeat-until akan berhenti jika kondisi ekspresi perulangan bernilai true. Lalu bagaimana jika diawal inialisasi nilai tersebut sudah true?. Perulangan repeat-until tetap melakukan perulangan karena pengecekan dilakukan diakhir
- Perhatikan source code program berikut

```
1 program repeat_until;
2 uses crt;
3 var
4   i: integer;
5 begin
6   clrscr;
7   i:= 6;
8   repeat
9     begin
10      writeln('pascal');
11      i:= i + 1;
12    end;
13  until i >= 5;
14  readln;
15 end.
```



Program disamping tetap menghasilkan output kata pascal sebanyak 1x, hal ini terjadi karena walaupun nilai inialisasi yang dilakukan sudah memenuhi syarat perhentian perulangan repeat-until akan tetapi pengecekan dilakukan diakhir, sehingga pekerjaan perulangan tetap dilakukan.



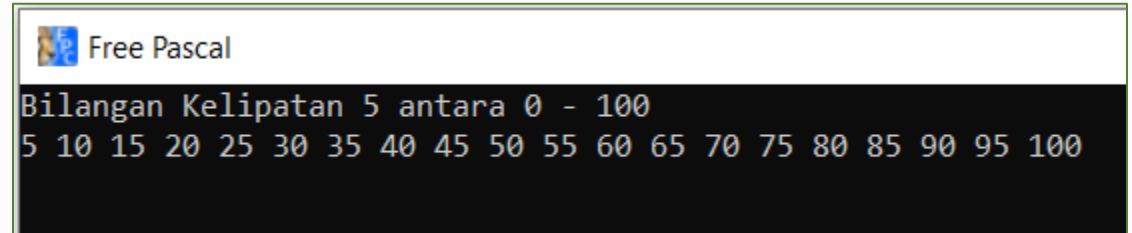
While atau Repeat

- Contoh-contoh sebelumnya menunjukkan bahwa penggunaan while dan repeat sama saja. Namun ada beberapa pertimbangan dalam memilih untuk menggunakan while atau 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 Program Lainnya

- Buatlah program untuk menampilkan deret bilangan kelipatan 5 yang berada antara nilai 0 - 100

```
1 program repeat_until;
2 uses crt;
3 var
4   i: integer;
5 begin
6   clrscr;
7   i:= 5;
8   writeln('Bilangan Kelipatan 5 antara 0 - 100');
9   repeat
10  begin
11    write(i, ' ');
12    i:= i + 5;
13  end;
14 until i > 100;
15 readln;
16 end.
```

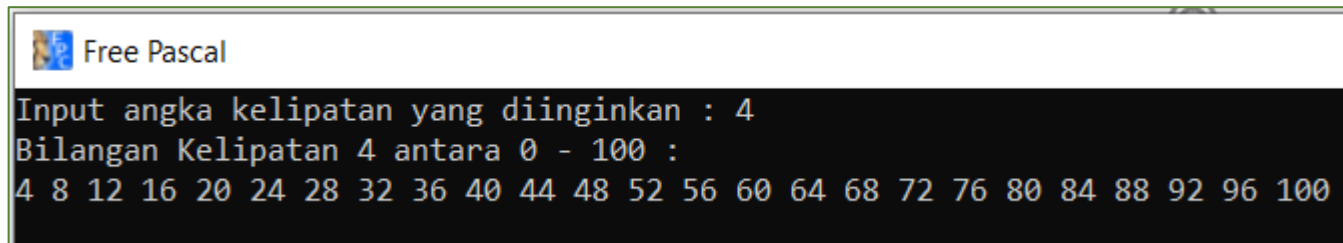


The screenshot shows the Free Pascal IDE's output window. The title bar reads "Free Pascal". The output text is as follows:

```
Bilangan Kelipatan 5 antara 0 - 100
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
```

Latihan

- Pada contoh program sebelumnya, yaitu menampilkan kelipatan 5. penentuan kelipatan 5 langsung didalam kode program. Kembangkan program tersebut sehingga program tersebut meminta inputan dari console angka kelipatan yang ingin ditampilkan dan juga meminta Batasan dari bilangan yang ditampilkan.
 - Output yang diharapkan



```
Free Pascal
Input angka kelipatan yang diinginkan : 4
Bilangan Kelipatan 4 antara 0 - 100 :
4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84 88 92 96 100
```


THANK YOU

