

Universitas gunadarma

pascal

Bab 4- bab 10

Hana Pertiwi S.T

14

PASCAL

Struktur Perulangan WHILE-DO
Struktur Perulangan REPEAT-UNTIL

Bentuk umum :

WHILE ungkapan logika DO statemen

Digunakan untuk melakukan proses perulangan suatu statemen terus menerus selama kondisi ungkapan logika pada while masih bernilai logika benar.

Contoh program :

```
Var
  I : integer ;
Begin
  I := 0 ;
  While I < 5 do
    Begin
      Writeln (I);
      I := I + 1 ;
    End ;
End.
```

Output program : 0

1
2
3
4

REPEAT – UNTIL

Digunakan untuk mengulang statemen sampai kondisi yang diseleksi di *Until* tidak terpenuhi.

Bentuk umum :

REPEAT statemen UNTIL ungkapan

Contoh program :

Var

I : integer ;

Begin

I := 0 ;

Repeat

I := I + 1 ;

Writeln (I) ;

Until I = 5;

End.

Output program : 1

2

3

4

5

Perbedaan antara struktur “ repeat until “ dengan “ while do “ adalah :

- Paling sedikit statemen-statemen dalam repeat until diproses sekali, karena seleksi kondisi ada pada statemen until yang terletak dibawah.
- Pada while do paling sedikit dikerjakan nol kali, karena seleksi kondisi ada pada statemen while yang terletak diatas, sehingga apabila kondisi tidak terpenuhi maka tidak akan masuk ke dalam lingkungan perulangannya.
- Pada repeat until dapat tidak menggunakan blok statemen (BEGIN dan END) untuk menunjukkan batas perulangannya, karena batas perulangannya sudah ditunjukkan oleh repeat sampai dengan until.

1. Struktur Perulangan FOR
2. Penggunaan gabungan struktur perulangan antara WHILE-DO, REPEAT-UNTIL dan FOR.

Digunakan untuk mengulang statemen berulang kali sejumlah yang ditentukan.

1.1.1 Perulangan Positif

Dengan penghitung / counter dari kecil ke besar (penambahan positif).
Bentuk umum :

FOR variable control : = nilai awal TO nilai akhir DO statemen

Contoh program (1) :

```
Var
    I : integer ;
Begin
    For I := 1 to 5 do
        Write ( I ) ;
        Writeln ('Pascal');
    End.
```

Output program (1) :

12345Pascal

Contoh program (2) :

```
Var
    I : integer ;
Begin
    For I := 1 to 5 do
        Begin
            Write ( I ) ;
            Writeln ('Pascal');
        End ;
    End.
```

Output program (2) :

```
1Pascal
2Pascal
3Pascal
4Pascal
5Pascal
```

1.1.2 Perulangan Negatif

Dengan penghitung / counter dari besar ke kecil (pertambahannya negatif).

Bentuk umum :

FOR variable control : = nilai awal DOWN TO nilai akhir DO statemen

Contoh program :

```
Var
  I : integer ;
Begin
  For I := 5 down to 1 do
    Begin
      Write ( I ) ;
      Writeln ('Pascal');
    End ;
End.
```

Output program :

```
5Pascal
4Pascal
3Pascal
2Pascal
1Pascal
```

1.1.3 Perulangan Tersarang

Perulangan yang berada didalam perulangan yang lainnya. Perulangan yang lebih dalam akan diproses lebih dulu sampai habis, kemudian perulangan yang lebih luar baru akan bertambah, mengerjakan

perulangan yang lebih dalam lagi mulai dari nilai awalnya dan seterusnya.

Contoh program :

```
Var
    I, J : integer ;
Begin
    For I := 1 to 5 do
        Begin
            For J := 1 to 3 do
                Write ( I : 8, J : 3);
            Writeln ;
        End ;
    End.
```

- | |
|---|
| <ol style="list-style-type: none">1. Statemen IF<ol style="list-style-type: none">1.1. Struktur IF THEN1.2. Struktur IF THEN ELSE1.3. Struktur IF tersarang |
|---|

Bentuk Perintah If Tunggal

Bentuk paling sederhana pilihan tunggal adalah IF-THEN, yakni jika hanya ada satu pilihan yang disediakan.

Statement IF-THEN digunakan untuk menyeleksi suatu kondisi, bila kondisi yang diseleksi terpenuhi, maka statement yang mengikuti THEN akan diproses. Sebaliknya bila kondisi tidak terpenuhi, maka yang akan diproses adalah statement berikutnya. Bentuk umum statemen kendali IF untuk pilihan tunggal

IF kondisi THEN

Contoh:

Program Suhu1;

USES Crt;

```
VAR
Suhu : Real;

BEGIN
ClrScr;

Write ('Suhu Tubuh : ');

Readln (Suhu);

IF Suhu > 37 THEN

WriteLn('Suhu Tinggi !');

WriteLn('Selesai');

END.
```

Bentuk Perintah If Ganda

Bentuk yang digunakan adalah IF pilihan ganda, yakni dengan memilih salah satu dari 2 alternatif yang disediakan . Bentuk Umum:

```
IF kondisi THEN

Statement1

ELSE

Statement 2
```

Bentuk Perintah If Bertingkat

Bentuk statement IF bertingkat merupakan variasi statement IF yang lengkap. Dalam pengambilan keputusan disediakan lebih dari 2 alternatif. Bentuk Umum:

```
IF kondisi THEN

Statement1
```

ELSE IF kondisi2 THEN

Statement2

ELSE

Statement selanjutnya

2. Statemen CASE

2.1. Struktur CASE OF

Struktur CASE OF ELSE

Case Of

Pernyataan CASE OF Digunakan untuk pengambilan keputusan jika terdapat lebih dari dua alternatif jawaban yang tersedia.

Bentuk umum :

CASE ungkapan OF

Label 1 : Statemen 1

Label 2 : Statemen 1

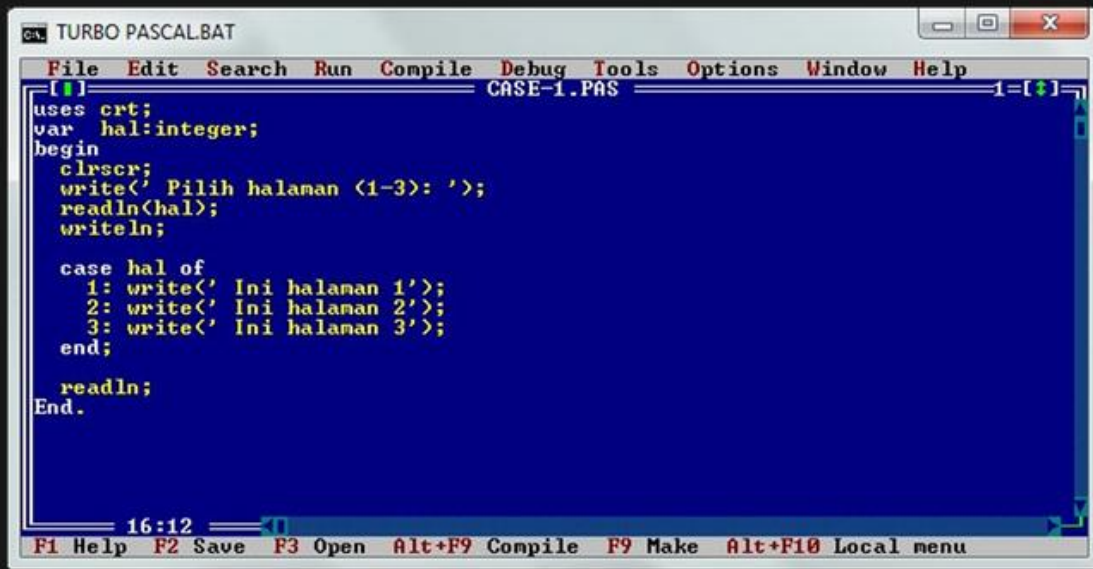
.....
.....
.....

Label n : Statemen n

END;

Pernyataan CASE OF boleh diikuti ELSE. Perlu diketahui, ELSE dalam pernyataan CASE OF Hanya bersifat pilihan, dalam artian boleh disertakan boleh tidak. Jika ELSE disertakan, pernyataan yang mengikuti ELSE akan dieksekusi hanya bila nilai dari syarat-pemilih tidak ada dalam semua label.

Contoh program:



```
uses crt;
var hal:integer;
begin
  clrscr;
  write(' Pilih halaman <1-3>: ');
  readln(hal);
  writeln;

  case hal of
    1: write(' Ini halaman 1');
    2: write(' Ini halaman 2');
    3: write(' Ini halaman 3');
  end;

  readln;
End.
```


Hasil eksekusi program:

Contoh 1



```
Pilih halaman <1-3>: 1
Ini halaman 1
```

Contoh program:



```
uses crt;
var hal:integer;
begin
  clrscr;
  write(' Pilih halaman <1-3>: ');
  readln(hal);
  writeln;

  case hal of
    1: write(' Ini halaman 1');
    2: write(' Ini halaman 2');
    3: write(' Ini halaman 3');
  else
    writeln(' Maaf, halaman yang tersedia hanya 1-3');
  end;

  readln;
End. -
```

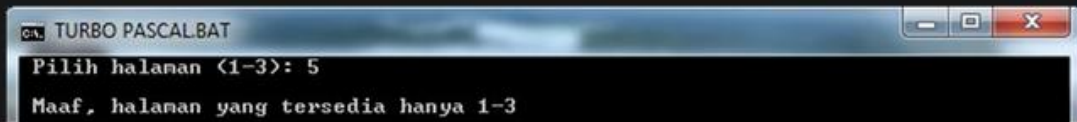
Hasil eksekusi program:

Contoh 1



```
GA. TURBO PASCAL.BAT
Pilih halaman <1-3>: 1
Ini halaman 1_
```

Contoh 2



```
GA. TURBO PASCAL.BAT
Pilih halaman <1-3>: 5
Maaf, halaman yang tersedia hanya 1-3
```

1. Parameter Dalam Procedure

- a. Pengiriman parameter secara nilai
- b. Pengiriman parameter secara acuan

2. Procedure memanggil procedure yang lain

3. Procedure Tersarang

4. Procedure memanggil dirinya sendiri.

- 1. Fungsi tanpa parameter
- 2. Parameter dalam fungsi
- 3. Fungsi Standart aritmatika
- 4. Fungsi tersarang
 - 1. Fungsi Memanggil dirinya sendiri.

Prosedur diawali dengan kata cadangan Procedure di dalam bagian deklarasi prosedur. Prosedur dipanggil dan digunakan di dalam blok program yang lainnya dengan menyebutkan judul prosedurnya.

Prosedur banyak digunakan pada program yang terstruktur, karena :

- 1. Merupakan penerapan konsep program modular, yaitu memecah-mecah program yang rumit menjadi program-program bagian yang lebih sederhana dalam bentuk prosedur-prosedur.
- 2. Untuk hal-hal yang sering dilakukan berulang-ulang, cukup dituliskan sekali saja dalam prosedur dan dapat dipanggil atau dipergunakan sewaktu-waktu bila diperlukan. Sebagaimana halnya sebuah program, suatu procedure juga memiliki header dan block. Perbedaan bentuknya dengan program hanyalah pada bagian header-nya saja.

bentuk umum procedure dan function

Tuliskan bentuk umum procedure dan function

➤ Procedure terbagi 2 yaitu

- Header procedure tanpa parameter

Bentuk umum : **Procedure**<Namaprocedure>;

contoh : **Procedure** TambahKali;

- Header procedure dengan parameter

Bentuk umum : **Procedure**<Namaprocedure>(<Daftarparameter>)

contoh : **Procedure** hitung (a,b : integer);

➤ Function terbagi 2 yaitu

- Header Function tanpa parameter

Bentuk umum : **Function** <Namafunction>:TipeData;

contoh : **Function** Hitung : integer;

Function Nama : String;

- Header Function dengan Parameter

Bentuk umum : **Function** <Namafunction>(<Daftarparameter>):TipeData;

contoh : **Function** Hitung (a,b:byte): integer;

Function Cekposisi (x,y:integer): boolean;

Pendeklarasian tipe data array
Array satu dimensi beserta Input dan outputnya
Array dua dimensi beserta Input dan outputnya

Larik atau array adalah tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe yang sama. Komponen-komponen ini disebut dengan tipe komponen (component type) atau tipe basis (base type). Larik mempunyai jumlah komponen yang banyaknya tetap. Banyaknya komponen dalam suatu larik ditunjukkan oleh suatu indeks yang disebut dengan tipe indeks (index type). Tipe indeks ini berbentuk ungkapan tipe ordinal. Tiap-tiap komponen di larik atau array dapat diakses dengan menunjukkan nilai indeksnya (index value) atau disebut juga dengan istilah subscript .

Ada beberapa cara mendefinisikan larik. Salah satu bentuk harus didefinisikan bisa digunakan secara konsisten dalam algoritma, antara lain :

a. nama_array: array [1..n] of tipe data;

 contoh : A:Array[1..10] of integer;

b. tipe_data nama_array[n];

 contoh : integer A[10];

c. type larik : array[1..n] of tipe data;

 nama_array : larik;

 contoh : type larik : array [1..10] of integer ;

 A : larik;

A. Array Satu dimensi.

 Array satu dimensi adalah array yang memiliki satu indeks saja. Bentuk umumnya adalah :
 tipe_data nama_variabel [ukuran];

B. Array dua dimensi

 Array dua dimensi adalah yang memiliki dua indeks . Bentuk umum :
 tipe_data nama_variabel[index1][index2];.