

BAB IV

PERULANGAN

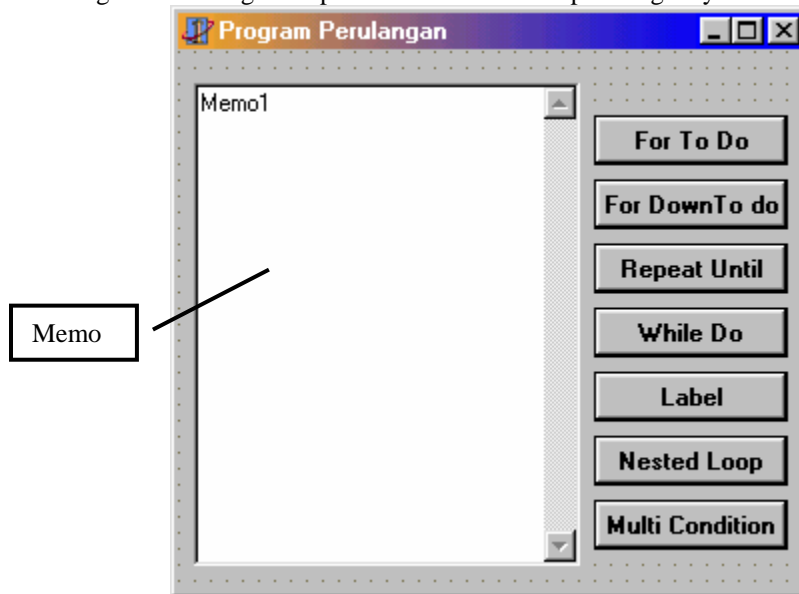
(For To Do, For Downto Do, Repeat Until, While Do)

Perulangan adalah suatu cara untuk mengulang satu atau sekumpulan perintah sampai mencapai kondisi tertentu.


Dalam Delphi terdapat beberapa perulangan yaitu :

1. Perulangan menggunakan For To Do
2. Perulangan menggunakan For DownTo Do
3. Perulangan menggunakan Repeat Until
4. Perulangan menggunakan While Do
5. Perulangan menggunakan Label
6. Perulangan tersarang (perulangan dalam perulangan)
7. Perulangan dengan banyak kondisi

Untuk lebih jelasnya, buatlah form seperti berikut dengan menggunakan 1 buah memo dan 7 buah BitBtn. Memo digunakan sebagai tempat untuk melihat hasil perulangannya.



Catatan :

Komponen Memo () umumnya digunakan untuk memasukan data string yang bisa menampung data dalam bentuk beberapa paragraf (banyak baris). Komponen ini dapat kita samakan dengan Notepad. Isi dari komponen ini ada di properti **Lines**.

Method-method yang banyak dipakai yang dimiliki oleh memo adalah :

- **Clear**, gunanya untuk mengosongkan memo.

Contoh :

```
Memo1.Clear;
```

- **Lines.Add**, gunanya untuk memasukan/menambahkan baris di Memo.

Inputan untuk method add harus berupa string. Jika yang ingin ditampilkan berupa angka, maka harus dikonversikan dulu.

Contoh :

```
Memo1.Lines.Add('Text Yang Ditambahkan');
A:=100;
Memo1.Lines.Add('Nilai A adalah : '+inttostr(A));
```

- **Lines.Delete**, gunanya untuk menghapus suatu baris.

Contoh :

```
Memo1.Lines.Delete(0); // Menghapus baris ke-1
```

1. For To Do

Perulangan ini berjalan dengan menggunakan suatu variabel counter yang akan bertambah secara otomatis ketika perintah yang diulang telah selesai dikerjakan.

Bentuk umum dari perulangan ini adalah :

```
For counter := nilai awal To Nilai akhir Do  
Begin  
    Perintah;  
    Perintah;  
End;
```

Perulangan ini akan berulang selama nilai variabel counter masih lebih kecil atau sama dengan nilai akhir.

Contoh Program yang ditulis dalam event OnClick dari tombol **For To Do**

```
procedure TForm1.BitBtn1Click(Sender: TObject);  
var  
    i:integer;  
    c:char;  
begin  
    Mem1.Clear;  
    for i:=1 to 10 do  
        Mem1.Lines.Add(IntToStr(i));  
    for c:='A' to 'Z' do  
        Mem1.Lines.Add(c);  
end;
```

2. For DownTo Do

Perulangan ini sama seperti perulangan dengan menggunakan For To Do tetapi arah isi counter nya berkurang (mundur/turun).

Bentuk umum dari perulangan ini adalah :

```
For counter := nilaiawal DownTo Nilaiakhir Do  
Begin  
    Perintah;  
    Perintah;  
End;
```

Perulangan ini akan berulang selama nilai variabel counter masih lebih besar atau sama dengan nilai akhir. Nilai awal suatu counter harus lebih besar atau sama dengan daripada nilai akhirnya.

Contoh Program yang ditulis dalam event OnClick dari tombol **For To Do**

```
procedure TForm1.BitBtn2Click(Sender: TObject);  
var  
    i:integer;  
    c:char;  
begin  
    Mem1.Clear;  
    for i:=10 downto 1 do  
        Mem1.Lines.Add(IntToStr(i));  
    for c:='Z' downto 'A' do  
        Mem1.Lines.Add(c);  
end;
```

3. Repeat Until

Perulangan ini bekerja sampai kondisi yang diinginkan tercapai. Perulangan ini bisa digunakan sebagai perulangan yang menggunakan banyak kondisi keluar (multi condition loop).

Bentuk umum :

```
Repeat
  Perintah;
  Perintah;
Until Kondisi;
```

```
procedure TForm1.BitBtn3Click(Sender: TObject);
var
  i:real;
begin
  Memo1.Clear;
  i:=0;
  repeat
    i:=i+0.5;
    Memo1.Lines.Add(FloatToStr(i));
  until i=10;
end;
```

Perulangan ini berulang sampai kondisi yang ada setelah until mempunyai nilai true. Jadi repeat until berjalan selama kondisi salah sampai kondisi benar.

Pengecekan kondisi berada di bagian bawah yaitu setelah until sehingga semua program yang memakai perulangan ini akan melakukan minimal 1 kali proses perulangan.

4. While Do

Perulangan ini mirip dengan perulangan Repeat Until tetapi pengecekannya berada di awal sebelum melakukan proses yang berulang. Perulangan ini berjalan selama kondisi pengecekan bernilai true. Jika kondisi telah mempunyai nilai false maka perulangan tidak dilakukan lagi.

Bentuk umum dari While Do adalah seperti berikut :

```
While kondisi do
Begin
  Perintah;
  Perintah;
End;
```

Contoh program while do yang ditulis pada event onclick dari tombol While Do.

```
procedure TForm1.BitBtn4Click(Sender: TObject);
var
  i:real;
begin
  Memo1.Clear;
  i:=0;
  while i<10 do
  begin
    i:=i+0.5;
    Memo1.Lines.Add(FloatToStr(i));
  end;
end;
```

5. Label

Label adalah suatu perintah untuk memandai suatu baris program. Untuk pindah ke posisi yang ditandai oleh label, maka digunakan perintah Goto.

Untuk lebih jelasnya lihatlah program berikut yang ditulis dalam event OnClick dari tombol Label.

```
procedure TForm1.BitBtn5Click(Sender: TObject);
label
  A;
var
  i:integer;
begin
  Mem1.Clear;
  i:=0;
  A:
    i:=i+1;
    Mem1.Lines.Add(IntToStr(i));
    if i<10 then
      Goto A;
End;
```

6. Perulangan Tersarang (perulangan dalam perulangan)

Perulangan tersarang (nested loop) adalah suatu perulangan dimana di dalam perulangan tersebut ada perulangan lain. Perulangan yang di ada di bagian terdalam, maka akan dikerjakan dulu sampai selesai baru mengerjakan perulangan yang ada diluarnya.

Nested loop bisa merupakan kombinasi antara For To Do, For DownTo Do, Repeat Until, While Do.

Contoh Program

```
procedure TForm1.BitBtn6Click(Sender: TObject);
var
  i,j:integer;
begin
  Mem1.Clear;
  for i:=1 to 5 do
  begin
    j:=0;
    while j<5 do
    begin
      j:=j+1;
      Mem1.Lines.Add(IntToStr(i)+' X '+IntToStr(j)
        +' = '+IntToStr(i*j));
    end;
    Mem1.Lines.Add('-----');
  end;
end;
```

7. Perulangan dengan banyak kondisi

Perulangan dengan banyak kondisi adalah suatu perulangan dimana berjalannya suatu perulangan tidak hanya berdasarkan suatu kondisi saja tetapi bisa lebih dari satu kondisi. Operator yang bisa digunakan biasanya seperti And, Or, Not dan lain-lain.

Contoh Program :

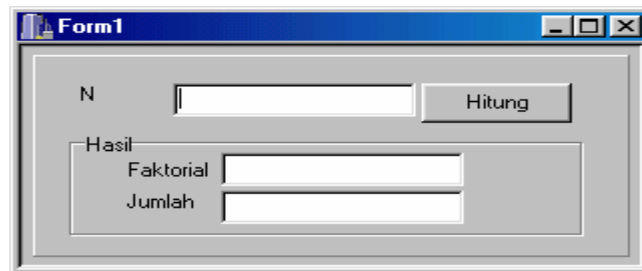
```

procedure TForm1.BitBtn7Click(Sender: TObject);
var
  i:integer;
  total:integer;
begin
  Mem1.Clear;
  total:=0;
  i:=0;
  repeat
    i:=i+1;
    Total:=Total+i;
    Mem1.Lines.Add('Isi I = '+IntToStr(i));
    Mem1.Lines.Add('Isi Total = '+IntToStr(Total));
  until (total>150)or(i>25);
end;

```

Kasus :

1. Buat suatu program untuk menentukan faktorial dan jumlah deret sampai ke N. Buat form sebagai berikut :



2. Buat suatu program untuk menghitung saldo akhir dari suatu tabungan dengan bunga dan jangka waktu tertentu :

Contoh hasil :

Saldo : 100000

Bunga : 10

Waktu : 3

Ketika user menekan tombol Hitung, maka akan muncul hasil seperti berikut di memonya.

Saldo Bulan 1 = Rp. 110000

Saldo Bulan 2 = Rp. 121000

Saldo Bulan 3 = Rp. 133100



3. Mirip dengan program ke-dua hanya input yang diberikan adalah : Saldo awal, bunga dan saldo akhir. Dari ketiga input tersebut maka akan didapatkan lamanya jangka waktu yang diinginkan agar saldo mencapai saldo akhir.

Contoh :

Saldo Awal : 100000

Bunga : 10

Saldo Akhir : 150000

Maka hasil pada memonya adalah :

Saldo Bulan 1 = Rp. 110,000

Saldo Bulan 2 = Rp. 121,000

Saldo Bulan 3 = Rp. 133,100

Saldo Bulan 4 = Rp. 146,410

Saldo Bulan 5 = Rp. 161,051

Form yang diinginkan :

The image shows a screenshot of a Delphi form window titled "Form1". The form has a title bar with standard Windows window controls (minimize, maximize, close). The main area of the form has a header section with the text "Perhitungan Bunga" in a large, bold, black font. Below the header, there is a section labeled "Data Input" which contains three input fields: "Saldo Awal" with the value "100000", "Bunga" with the value "10" and a percentage sign "%", and "Saldo akhir" with the value "150000". Below these input fields is a button labeled "Hitung". To the right of the input fields is a large, empty rectangular area, likely intended for displaying the results of the calculation.