

# PART 41

---

PERULANGAN FOR DO DALAM PASCAL

# PENGERTIAN PERULANGAN FOR DO DALAM PASCAL

---

Perulangan, atau **looping** dalam bahasa inggris, adalah konsep pemrograman dimana kita mengulang baris program beberapa kali. "Beberapa kali" disini bisa dikatakan tidak terbatas, selama komputer masih bisa mengolahnya.

Sebagai contoh, saya ingin anda menulis teks "**Hello World**" sebanyak 1000 kali. Tentu sangat melelahkan mengetik semua ini (walaupun dengan di-copy paste). Menggunakan struktur **looping**, kita bisa membuatnya dalam waktu singkat dan cepat, hanya butuh beberapa baris kode program.

Salah satu struktur perulangan di dalam Pascal adalah **FOR DO**, berikut format penulisannya:

```
FOR (variabel_counter) := (nilai_awal) TO (nilai_akhir) DO  
begin  
  (kode program yang ingin diulang disini...)  
end;
```

---

`variabel_counter` adalah variabel yang berfungsi sebagai **counter**, atau penghitung di dalam perulangan. Variabel ini otomatis menaik dari `nilai_awal` hingga `nilai_akhir`. Dalam setiap kenaikan, blok kode program yang berada di dalam `begin` dan `end`; akan dijalankan. `variabel_counter` ini bisa digunakan sepanjang perulangan (jika diperlukan).

# CONTOH KODE PROGRAM PASCAL UNTUK PERULANGAN FOR DO

Sebagai contoh pertama, saya ingin membuat 1000 teks "Hello World". Berikut kode programnya:

```
1  program for_do_loop;  
2  uses crt;  
3  var  
4    i: integer;  
5  begin  
6    clrscr;  
7  
8    for i := 1 to 1000 do  
9      begin  
10       writeln('Hello World');  
11     end;  
12  
13    readln;  
14  end.
```

Di bagian **var**, saya membuat sebuah variabel **i** dengan tipe data **integer**. Variabel ini saya siapkan sebagai variabel **counter**.

Untuk membuat perulangan sebanyak 1000 kali, saya menggunakan perintah **for i := 1 to 1000 do**. Ini bisa dibaca dengan "*Untuk variabel i, jalankan perulangan mulai dari 1 hingga 1000*".

Apa yang akan diulang? Adalah blok program yang diawali perintah **begin**, dan diakhiri perintah **end**;. Di dalam blok ini saya membuat sebuah perintah: **writeln('Hello World')**.

Dengan demikian, pascal akan menjalankan 1000 kali baris **writeln('Hello World')**. Mari kita coba:

- 
- Mari masuk ke contoh selanjutnya. Kali ini saya ingin membuat kode program yang merupakan daftar perkalian **7**, yakni  **$7 \times 1 = 7$** ,  **$7 \times 2 = 14$** ,  **$7 \times 3 = 21$** , dst hingga  **$7 \times 10 = 70$** . Disini kita harus menggunakan variabel counter *i* agar setiap perkalian naik dari 1 hingga 10. Berikut kode program yang saya gunakan:



---

```
1  program for_do_loop;
2  uses crt;
3  var
4      i: integer;
5  begin
6      clrscr;
7
8      for i := 1 to 10 do
9          begin
10             writeln('7 x ',i,' = ',7 * i);
11         end;
12
13     readln;
14 end.
```

- 
- Sebagai contoh terakhir, mari kita coba modifikasi kode program diatas agar lebih interaktif. Kali ini saya ingin membuat daftar perkalian tapi angka perkaliannya diinput oleh pengguna. Misalnya diinput angka **9**, tampilkan daftar perkalian **9 x 1**, **9 x 2**, hingga **9 x 10**. Jika diinput **14**, tampilkan perkalian **14 x 1**, **14 x 2**, hingga **14 x 10**.
  - Sebagai latihan, silahkan anda coba buat sendiri.

```
1  program for_do_loop;
2  uses crt;
3  var
4      i,angka: integer;
5  begin
6      clrscr;
7      writeln('=====');
8      writeln('Program Daftar Perkalian');
9      writeln('=====');
10     writeln;
11
12     write('Input perkalian yang diinginkan: ');
13     readln(angka);
14
15     writeln;
16     for i := 1 to 10 do
17     begin
18         writeln( i , ' x ', angka , ' = ', i * angka);
19     end;
20
21     readln;
22 end.
```



---

Disini saya menggunakan 2 buah variabel: `i` dan `angka`. Variabel `angka` digunakan untuk menampung inputan, sedangkan variabel `i` sebagai **variabel counter**.

Selain tambahan kode `readln(angka)`, kode program diatas mirip seperti sebelumnya. Tentu saja di dalam blok perulangan saya harus membuat baris `writeln( i , ' x ' , angka , ' = ' , i * angka)` agar variabel `angka` di proses sebagai bilangan pengali.