

Modul Materi :

Struktur Kendali Perulangan

3

Obyektif

Setelah pada pertemuan sebelumnya telah mempelajari struktur kendali kondisi, pada pertemuan kali ini peserta akan mempelajari tentang struktur kendali perulangan yang juga berguna untuk mengontrol jalannya program yang akan dibuat. Pada pertemuan kali ini juga akan belajar tentang macam-macam struktur perulangan di QBasic dan bagaimana cara menuliskannya.

1. Pendahuluan

1.1 Pengenalan Struktur kendali Perulangan

Perulangan (*loop*) adalah bentuk yang sering ditemui di dalam suatu program aplikasi. Sebagai contoh, jika ingin menampilkan angka 1 sampai 10 pada komputer bagaimana cara menuliskannya, apakah menuliskan satu persatu angka- angka tersebut atau dengan sebuah program dengan struktur kendali perulangan. Pada contoh kasus tersebut dapat diterapkan dengan menggunakan struktur kendali perulangan tidak perlu lagi menuliskan angka-angka tersebut satu-per- satu, hanya perlu menuliskan sebuah program saja, dan dapat terhindar dari penulisan kode program yang sama. Dan jika ingin menuliskan kata “komputer” sebanyak 100 kali, tidak perlu menuliskan kata yang sama berulang-ulang.

2. Struktur kendali perulangan di QBasic

2.1 Perulangan counter

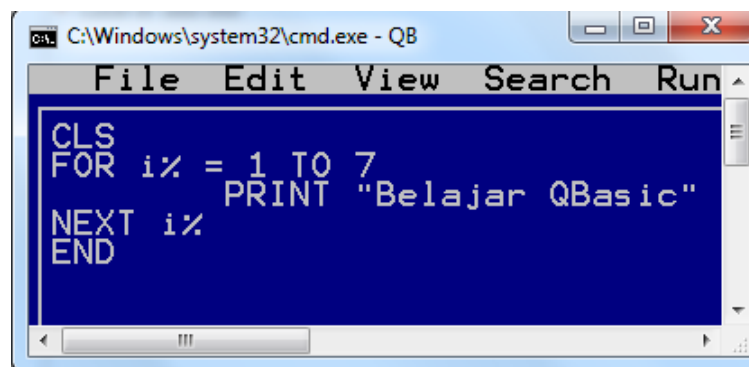
Perulangan counter ini digunakan untuk proses berulang yang jumlah perulangannya ditentukan berdasarkan perhitungan (sistem counter). Yang termasuk perulangan ini adalah perulangan FOR-NEXT.

Bentuk Umum :

```
FOR counter = mulai TO selesai [STEP selisih] [EXIT FOR]
    Statement 1
    ...
    Statement n
NEXT [counter]
```

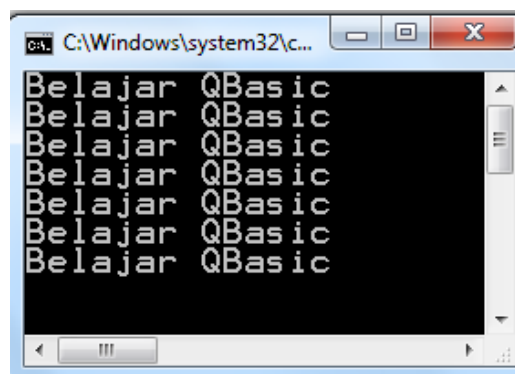
1. Pada perulangan FOR-NEXT jika ingin keluar dari program untuk kondisi tertentu sebelum kita menggunakan EXIT FOR.
2. STEP digunakan untuk selisih angka, misal ingin mencetak kelipatan 2, maka menuliskan STEP 2.

Contoh1 :



```
C:\Windows\system32\cmd.exe - QB
File Edit View Search Run
CLS
FOR i% = 1 TO 7
    PRINT "Belajar QBasic"
NEXT i%
END
```

Gambar 2.1. Contoh program FOR-NEXT



```
C:\Windows\system32\c...
Belajar QBasic
Belajar QBasic
Belajar QBasic
Belajar QBasic
Belajar QBasic
Belajar QBasic
Belajar QBasic
```

Gambar 2.2 Output program

Contoh2 :

```

CLS
PRINT "----- LUAS PERSEGI PANJANG -----"
PRINT "===== "
PRINT " PANJANG | LEBAR | LUAS "
PRINT "===== "
format$ = " ## | ## | ## "
lebar = 1
FOR panjang = 2 TO 12 STEP 2
    LUAS = panjang * lebar
    PRINT USING format$; panjang; lebar; LUAS
    lebar = lebar + 1
NEXT panjang
END
  
```

Gambar 2.3. Contoh program FOR-NEXT menggunakan counter “panjang”

----- LUAS PERSEGI PANJANG -----		
=====		
PANJANG	LEBAR	LUAS
=====		
2	1	2
4	2	8
6	3	18
8	4	32
10	5	50
12	6	72

Gambar 2.4 Output program

Contoh 3:

```

CLS
PRINT "----- PROGRAM EXIT FOR -----"
PRINT "===== "
PRINT " i | i | hasil "
PRINT "===== "
format$ = " ## | ## | ## "
FOR i = 1 TO 10
    bil = i * i
    PRINT USING format$; i; i; bil
    IF bil = 49 THEN EXIT FOR
NEXT i
PRINT "-- Proses Pengulangan Selesai --"
END
  
```

Gambar 2.5 Contoh program FOR-NEXT menggunakan EXIT FOR

```

----- PROGRAM EXIT FOR -----
=====
i | i | hasil
=====
1 | 1 | 1
2 | 2 | 4
3 | 3 | 9
4 | 4 | 16
5 | 5 | 25
6 | 6 | 36
7 | 7 | 49
-- Proses Pengulangan Selesai --

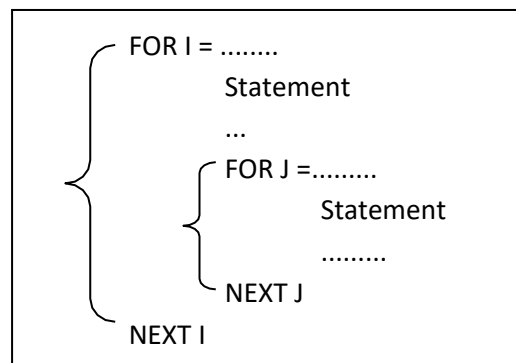
```

Gambar 2.6 Output program

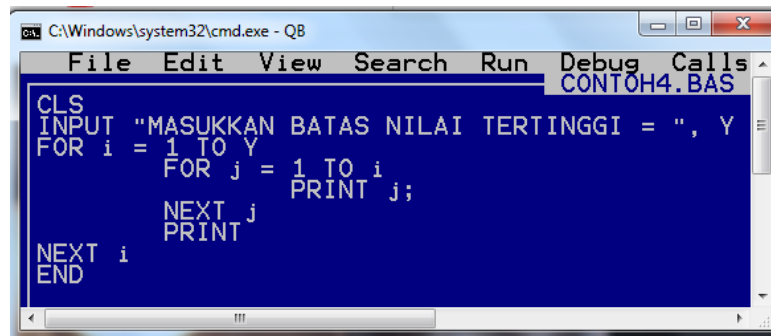
2.2 Perulangan jamak (nested looping)

Perulangan ini, dilakukan jika terdapat dua buah perulangan. Jadi terdapat perulangan di dalam perulangan. Contoh dari perulangan jamak ini paling mudah dipelajari dengan menggunakan *statement* FOR-NEXT.

Bentuk Umum :

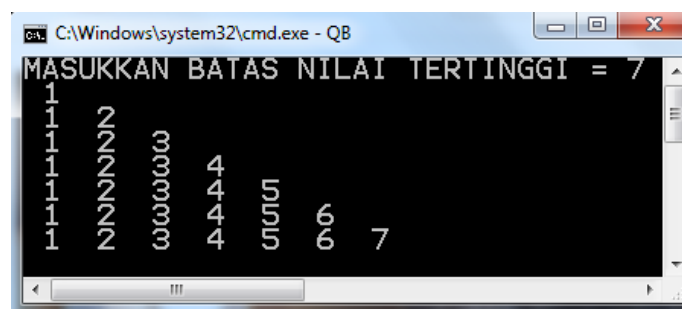


- Pada perulangan jamak ini, perulangan yang didalam harus diselesaikan dahulu.
- Dalam perulangan jamak ada 2 ketentuan yang harus dipenuhi, yaitu :
 1. Harus menggunakan perubah kendali yang berlainan
 2. Antara kalang yang satu dengan kalang yang lain tidak boleh saling berpotongan Contoh :



```
CLS
INPUT "MASUKKAN BATAS NILAI TERTINGGI = ", Y
FOR i = 1 TO Y
    FOR j = 1 TO i
        PRINT j;
    NEXT j
    PRINT
NEXT i
END
```

Gambar 2.7 Contoh program Nested looping



```
MASUKKAN BATAS NILAI TERTINGGI = 7
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
```

Gambar 2.8 Output program

2.3 Perulangan Trapping

Perulangan jenis ini, merupakan perulangan yang proses berulangnya ditentukan berdasarkan kondisi tertentu. Pada perulangan ini dibagi lagi menjadi 2 jenis perulangan yaitu :

- Perulangan yang kondisinya berada diawal, dan
- Perulangan yang kondisinya berada di akhir.

Macam-macam perulangan yang kondisinya berada di awal :

- DO WHILE-LOOP

Bentuk umum :

```
DO WHILE ekspresi_logika
    Statement
    ...
LOOP
```

- Pada perulangan Do While-Loop terdapat juga statement EXIT DO yang digunakan untuk keluar dari perulangan DO WHILE sebelum perulangan tersebut selesai. Contoh :

```

CLS
PRINT "    i  +  i  =  HASIL  "
PRINT "===== "
format$ = "  ##  |  ##  |  ##  "
hitung = 10
DO WHILE hitung > 0
    hasil = hitung + hitung
    PRINT USING format$; hitung; hitung; hasil
    hitung = hitung - 1
LOOP
PRINT "Perulangan Selesai"
END
  
```

Gambar 2.9 Contoh perulangan menggunakan DO While-Loop

```

    i  +  i  =  HASIL
=====
10    10    20
 9     9    18
 8     8    16
 7     7    14
 6     6    12
 5     5    10
 4     4     8
 3     3     6
 2     2     4
 1     1     2
Perulangan Selesai
  
```

Gambar 2.10 Output program

b. DO UNTIL-LOOP

Bentuk umum :

```

DO UNTIL ekspresi_kondisi
    Statement
    ...
LOOP
  
```

- Pada perulangan DO UNTIL, kondisi merupakan kondisi yang pasti berapa banyak ingin melakukan perulangan.
- Pada perulangan DO UNTIL-Loop terdapat juga statement EXIT DO yang digunakan untuk keluar dari perulangan DO UNTIL sebelum perulangan tersebut selesai.

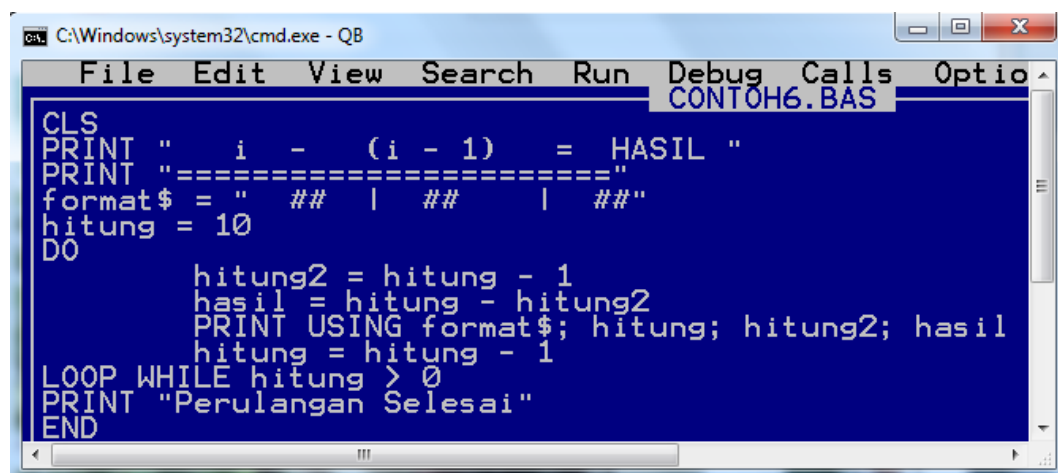
Perulangan yang kondisinya berada di akhir:

a. DO-LOOP WHILE

Bentuk umum :

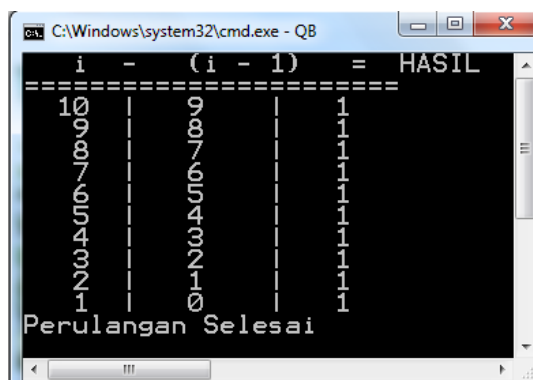
```
DO
    Statement
    ...
LOOP WHILE ekspresi_logika
```

- Perlu di ingat pada perulangan DO-LOOP WHILE ini, paling tidak pasti akan terjadi 1 kali perulangan.
- Pada perulangan Do While-Loop terdapat juga statement EXIT DO yang digunakan untuk keluar dari perulangan DO WHILE sebelum perulangan tersebut selesai. Contoh :



```
CLS
PRINT " i - (i - 1) = HASIL "
PRINT "====="
format$ = " ## | ## | ##"
hitung = 10
DO
    hitung2 = hitung - 1
    hasil = hitung - hitung2
    PRINT USING format$; hitung; hitung2; hasil
    hitung = hitung - 1
LOOP WHILE hitung > 0
PRINT "Perulangan Selesai"
END
```

Gambar 2.11 Contoh program penggunaan DO-LOOP WHILE



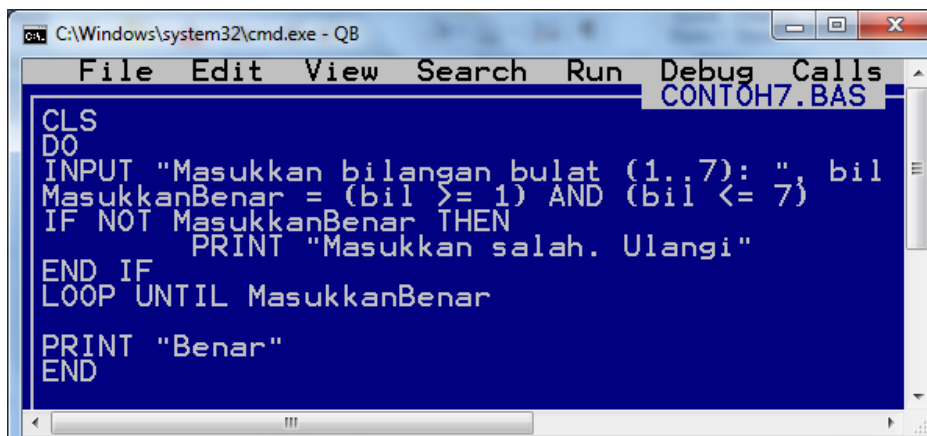
```
 i - (i - 1) = HASIL
=====
10 | 9 | 1
 9 | 8 | 1
 8 | 7 | 1
 7 | 6 | 1
 6 | 5 | 1
 5 | 4 | 1
 4 | 3 | 1
 3 | 2 | 1
 2 | 1 | 1
 1 | 0 | 1
Perulangan Selesai
```

Gambar 2.12 Output program

b. DO- LOOP UNTIL

```
DO
    Statement
    ...
LOOP UNTIL ekspresi_Kondisi
```

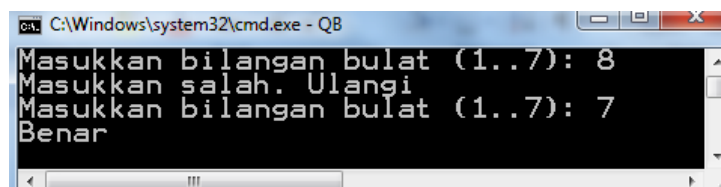
- Pada perulangan Do UNTIL, kondisi merupakan kondisi yang pasti berapa banyak ingin melakukan perulangan.
- Pada perulangan Do UNTIL-Loop terdapat juga statement EXIT DO yang digunakan untuk keluar dari perulangan DO UNTIL sebelum perulangan tersebut selesai Contoh :



The screenshot shows a QBASIC window titled 'C:\Windows\system32\cmd.exe - QB' with a menu bar (File, Edit, View, Search, Run, Debug, Calls) and a toolbar. The code in the window is as follows:

```
CLS
DO
INPUT "Masukkan bilangan bulat (1..7): ", bil
MasukkanBenar = (bil >= 1) AND (bil <= 7)
IF NOT MasukkanBenar THEN
    PRINT "Masukkan salah. Ulangi"
END IF
LOOP UNTIL MasukkanBenar
PRINT "Benar"
END
```

Gambar 2.13 Contoh penggunaan DO-LOOP UNTIL



The screenshot shows the output of the program in the same QBASIC window. The text displayed is:

```
Masukkan bilangan bulat (1..7): 8
Masukkan salah. Ulangi
Masukkan bilangan bulat (1..7): 7
Benar
```

Gambar 2.14 Output program

3. Rangkuman

1. Dengan menggunakan struktur kendali perulangan dapat terhindar dari penulisan data yang sama.
2. Perulangan counter digunakan untuk proses berulang yang jumlah perulangannya ditentukan berdasarkan perhitungan.
3. Perulangan jamak (*nested looping*) dilakukan jika terdapat dua buah perulangan. Jadi terdapat perulangan di dalam perulangan.
4. Ada dua macam perulangan trapping yaitu yang kondisinya berada diawal (*DO – WHILE – LOOP*) dan yang kondisinya berada diakhir (*DO – LOOP – WHILE*).

4. Tuntunan Latihan!!

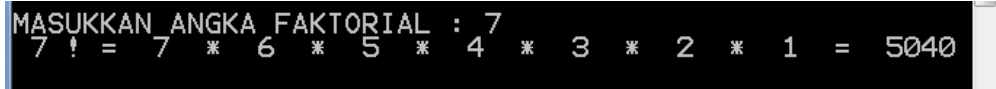
1. Pertama-tama buka aplikasi Qbasic.
2. Lalu setelah itu tekan tombol ESC pada keyboard anda.
3. Sekarang ketikkan sebuah program dengan sintak seperti di bawah ini

```
CLS
PRINT "----- PROGRAM EXIT FOR ----- "
PRINT "===== "
PRINT " i      /   i   | HASIL "
PRINT "===== "
format$ = " ## | ##      | ## "
FOR i = 1 TO 10
    bil = 1 * i
    PRINT USING format$; i; i; bil
    IF bil = 49 THEN EXIT FOR
NEXT i
PRINT "— Proses Pengulangan Selesai —"
```

4. Setelah itu klik menu RUN kemudian START atau tekan tombol F5.
5. Kemudian simpan program yang telah anda buat.

5. LATIHAN SOAL!!

1. Coba buat sebuah program pada QBasic dengan output :



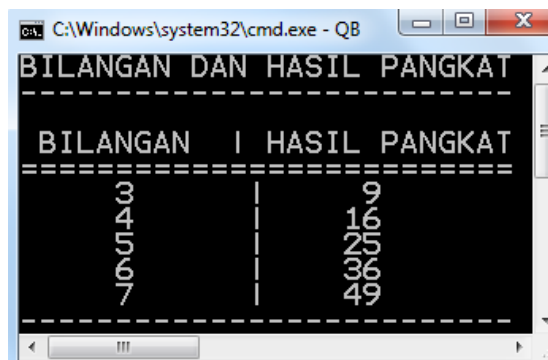
MASUKKAN ANGKA FAKTORIAL : 7
7 ! = 7 * 6 * 5 * 4 * 3 * 2 * 1 = 5040

Gambar 2.12 Output program

2. Apakah output dari potongan program berikut :

```
CLS
PRINT "PROGRAM JUMLAH DERET"
PRINT "DERET 1/2 + 2/3 + ... + 99/100"
PRINT
DERET = 0
FOR I = 1 TO 99
    DERET = DERET + ( I / ( I + 1 ) )
NEXT I
PRINT "HASILNYA ADALAH : "; DERET
END
```

3. Coba buatlah sebuah program dengan output seperti berikut ini!



C:\Windows\system32\cmd.exe - QB

BILANGAN DAN HASIL PANGKAT	
BILANGAN	HASIL PANGKAT
3	9
4	16
5	25
6	36
7	49

Gambar 2.13 Output Program Bilangan dan Hasil Pangkat