

Statement Kondisi & GO TO

I. Pernyataan GO TO

Pernyataan GO TO akan mengarahkan proses program ke suatu nama paragraph yg ditunjuk oleh pernyataan GO TO tersebut. Ada 2 bentuk :

Bentuk 1 :

GO TO nama-paragraph.

Contoh 1 :

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. go-to1.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
PROCEDURE DIVISION.  
Tampilkan.  
    DISPLAY ` Tampilan contoh, hentikan dg CTRL-BREAK `  
    GO TO Tampilkan.  
SELESAI.  
STOP RUN.
```

Contoh 2 :

IDENTIFICATION DIVISION.

PROGRAM-ID. goto2.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 JawabPIC A.

PROCEDURE DIVISION.

Tanya.

DISPLAY ` Masukkan Jawab berupa huruf (D/E/F/G) ? `

ACCEPT Jawab.

Seleksi-Jawab.

IF Jawab = 'D' GO TO Kode-D.

IF Jawab = 'E' GO TO Kode-E.

IF Jawab = 'F' GO TO Kode-F.

IF Jawab = 'G' GO TO Kode-G.

Kode-D.

DISPLAY ` Kampus Depok `

GO TO Selesai.

Kode-E.

DISPLAY ` Kampus Kelapa Dua `

GO TO Selesai.

Kode-F.

```
DISPLAY ` Kampus Cikini `  
GO TO Selesai.
```

Kode-G.

```
DISPLAY ` Kampus Kelapa Dua Perempatan `  
GO TO Selesai.
```

Selesai.

```
STOP RUN.
```

Tampilan output :

Masukkan Jawab berupa huruf (D/E/F/G) ?

E

Kampus Kelapa Dua

Masukkan Jawab berupa huruf (D/E/F/G) ?

D

Kampus Depok

Bentuk 2 :

GO TO nama-paragraph-1 [, nama-paragraph-2 ,] DEPENDING ON variabel

Bentuk ini dipakai untuk mengatur eksekusi ke nama-paragraph tertentu sesuai dengan nilai dari variabel (harus bertipe numerik).

Contoh :

IDENTIFICATION DIVISION.

PROGRAM-ID. hari.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 pilihan PIC 9.

PROCEDURE DIVISION.

pilih-hari.

DISPLAY (,) ` Pilih salah satu kode daerah waktu di Indonesia (1 .. 3) = `

ACCEPT (,) pilihan WITH AUTO SKIP.

DISPLAY SPACES.

GO TO waktu-1, waktu-2, waktu-3 DEPENDING ON pilihan.

GO TO pesan-salah.

waktu-1.

DISPLAY ` Waktu Indonesia Bagian Barat `.

GO TO akhir-program.

waktu-2.

DISPLAY ` Waktu Indonesia Bagian Tengah `.

GO TO akhir-program.

waktu-3.

DISPLAY ` Waktu Indonesia Bagian Timur `.

GO TO akhir-program.

pesan-salah.

DISPLAY ` Kode yang dimasukkan salah !!! `.

akhir-program.

STOP RUN.

Hasil eksekusi :

Pilih salah satu kode daerah waktu di Indonesia (1 .. 3) = 1
Waktu Indonesia Bagian Barat

Pilih salah satu kode daerah waktu di Indonesia (1 .. 3) = 5
Kode yang dimasukkan salah !!!

Catatan :

Penggunaan WITH AUTO SKIP untuk memindahkan kursor secara otomatis ke baris selanjutnya bila panjang data yg dimasukkan sama dengan lebar field pada PICTURE nya. Bila panjang data kurang yg dimasukkan dari lebar field, diperlukan penekanan ENTER untuk mengakhiri input data tersebut.

Bentuk **GO TO waktu-1, waktu-2, waktu-3 DEPENDING ON pilihan** dalam program di atas identik dengan :

IF pilihan = 1 GO TO waktu-1

IF pilihan = 2 GO TO waktu-2

IF pilihan = 3 GO TO waktu-3

II. Pernyataan Alter

Digunakan utk mengubah arah tujuan proses dari statement GOTO yg telah ada di dalam program.

Bentuknya:

```
ALTER nama-paragraph-1 TO {PROCEED TO} nama-paragraph-2.
```

Keterangan:

Nama-paragraph-1 menunjukkan suatu paragraph yang hanya berisi satu statement GO TO saja. sedang nama-paragraph-2 menunjukkan suatu paragraph lain yang ada di PROCEDURE DIVISION.

Contoh:

```
IDENTIFICATION DIVISION  
PROGRAM-ID ALTER.  
AUTHOR. JOGIYANTO HM.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
PROCEDURE DIVISION.  
PROCEDURE DIVISION.  
TAMPILKAN.  
    DISPLAY 'KOMPUTER'.  
    PARA-1  
        GO-TO PARA-2.  
    PARA-2.  
        ALTER PARA-1 TO PROCEED TO SELESAI  
        GO TO TAMPILKAN.  
SELESAI.  
STOP RUN.
```

III. Pernyataan PERFORM

Perintah ini dipakai untuk membuat suatu program meloncat ke suatu paragraph tertentu dan mengerjakan seluruh isi paragraph tersebut sampai selesai lalu kembali lagi ke statement setelah statement PERFORM.

Pernyataan PERFORM digunakan untuk :

- ⇒ Proses perulangan
- ⇒ Pemrograman terstruktur

Bentuk1 :

PERFORM *nama-paragraph*

PERFORM *nama-paragraph-1* { THRU / THROUGH } *nama-paragraph-2*

- ⇒ Dari ilustrasi di atas terlihat bahwa PERFORM digunakan untuk mengarahkan proses program menuju ke suatu paragraph tertentu, mengerjakan isi dari paragraph tersebut & setelah selesai kembali melanjutkan ke pernyataan-1 (pernyataan yg ada setelah pernyataan PERFORM tersebut).
- ⇒ Jika digunakan PERFORM paragraph-A THRU paragraph-C , maka proses akan diarahkan untuk menjalankan paragraph-A hingga paragraph-C (paragraph-A, paragraph-B, paragraph-C)

Bentuk2 :

PERFORM *nama-paragraph-1* { THRU / THROUGH } *nama-paragraph-2*
{ *nama-data / integer* } TIMES

Digunakan untuk melakukan suatu proses berulang-ulang sebanyak yg ditunjukkan oleh nilai dari *nama-data* atau *integer*.

contoh :

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. ulang1.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
77 nomor-urut PIC 9.  
PROCEDURE DIVISION.  
Ulang-proses.  
    PERFORM Cetak-tampilan 5 TIMES.  
    STOP RUN.  
Cetak-tampilan.  
    ADD 1 to nomor-urut.  
    DISPLAY ` Tampilan ke = `, nomor-urut.
```

Hasil keluaran :

```
Tampilan ke = 1  
Tampilan ke = 2  
Tampilan ke = 3  
Tampilan ke = 4  
Tampilan ke = 5
```


Bentuk3 :

PERFORM *nama-paragraph-1* { THRU / THROUGH } *nama-paragraph-2*
UNTIL *kondisi*

Bentuk ini digunakan untuk mengulang satu atau beberapa proses. Banyaknya perulangan ditentukan oleh kondisi. Jika kondisi *benar*, maka proses perulangan akan diakhiri.

Contoh :

IDENTIFICATION DIVISION.

PROGRAM-ID. ulang2.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 pilihan PIC X.

88 pilihan-absah VALUE 'A' THRU 'D'
' a' THRU 'd'.

SCREEN SECTION.

01 layar-menu.

02 LINE 5 COLUMN 25 ' ** MENU PILIHAN ** '.

02 LINE 7 COLUMN 25 ' [A] Memasukkan data '.

02 LINE 8 COLUMN 25 ' [B] Mengedit data '.

02 LINE 9 COLUMN 25 ' [C] Menghapus data '.

02 LINE 10 COLUMN 25 ' [D] Selesai '.

02 LINE 11 COLUMN 25 ' Pilihan [A .. D] = '.

02 PIC X USING pilihan.

PROCEDURE DIVISION.

Program-utama.

MOVE SPACES TO pilihan.

PERFORM input-pilihan UNTIL pilihan-absah.

DISPLAY ' Input pilihan selesai !!! '.

STOP RUN.

input-pilihan.

DISPLAY SPACE.

DISPLAY layar-menu.

ACCEPT layar-menu.

Hasil keluaran :

```
** MENU PILIHAN **
```

```
[ A ] Memasukkan data
```

```
[ B ] Mengedit data
```

```
[ C ] Menghapus data
```

```
[ D ] Selesai
```

```
Pilihan [ A .. D ] = X
```

```
** MENU PILIHAN **
```

```
[ A ] Memasukkan data
```

```
[ B ] Mengedit data
```

```
[ C ] Menghapus data
```

```
[ D ] Selesai
```

```
Pilihan [ A .. D ] = B
```

```
Input pilihan selesai !!!
```

Catatan :

- USING clause digunakan sebagai pengganti FROM dan TO clause, bila pada PROCEDURE DIVISION menggunakan statement DISPLAY dan ACCEPT.
- Pernyataan MOVE SPACES TO pilihan digunakan untuk memberi nilai-awal dari kondisi agar bernilai salah (bukan ` A ` .. ` D ` atau ` a ` .. ` d `), sehingga paragraph *input-pilihan* akan dijalankan.
- Kondisi ini akan bernilai benar, jika salah satu diantara tombol A sampai D atau a sampai d ditekan pd saat ACCEPT layar-menu dijalankan.

Bentuk4 :

```
PERFORM nama-paragraph-1 { THRU / THROUGH } nama-paragraph-2
VARYING nama-data-1
FROM {nama-data-2 / integer-1} BY {nama-data-3 / integer-2}
UNTIL kondisi-1
[ AFTER nama-data-4
FROM {nama-data-5 / integer-3} BY {nama-data-6 / integer-4 }
UNTIL kondisi-2 ]
[ AFTER nama-data-7
FROM {nama-data-8 / integer-5} BY {nama-data-9 / integer-6 }
UNTIL kondisi-3 ]
```

Contoh PERFORM - VARYING 1 :

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. ulang3.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
77 bilangan PIC 99.  
PROCEDURE DIVISION.  
Program-utama.  
    PERFORM Cetak-bilangan  
        VARYING bilangan FROM 1 BY 1  
        UNTIL bilangan > 6.  
STOP RUN.  
Cetak-bilangan.  
    DISPLAY bilangan.
```

Hasil eksekusi :

```
01  
02  
03  
04  
05  
06
```

Contoh PERFORM - VARYING 2 :

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. ulang4.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
77 bilangan PIC 99.  
PROCEDURE DIVISION.  
Program-utama.  
    PERFORM Cetak-bilangan  
        VARYING bilangan FROM 0 BY 2  
        UNTIL bilangan > 10.  
STOP RUN.  
Cetak-bilangan.  
    DISPLAY bilangan.
```

Hasil eksekusi :

```
00  
02  
04  
06  
08  
10
```

Catatan :

Jika diinginkan untuk menampilkan suatu bilangan secara decrease (urutan dari besar ke kecil) , misalnya 20, 18, 164, 2 .

Penulisan statement PERFORM nya adl :

```
PERFORM cetak-bilangan  
  VARYING bilangan FROM 20 BY -2  
  UNTIL bilangan < 2
```

Contoh PERFORM - VARYING - AFTER 1 :

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. ulang5.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
77 X PIC 9.  
77 Y PIC 9
```

PROCEDURE DIVISION.

Program-utama.

PERFORM Cetak-nilai -XY

VARYING X FROM 1 BY 1 UNTIL X > 3

AFTER Y FROM 1 BY 1 UNTIL Y > 2.

STOP RUN.

Cetak-nilai-XY.

DISPLAY ` nilai X = `, X, ` Y = `, Y.

Hasil eksekusi :

nilai X = 1 Y = 1

nilai X = 1 Y = 2

nilai X = 2 Y = 1

nilai X = 2 Y = 2

nilai X = 3 Y = 1

nilai X = 3 Y = 2

Keterangan :

Dari program di atas terlihat bahwa nilai X = 1 belum bertambah sampai nilai Y bertambah terlebih dulu hingga kondisi-2 (Y > 2) terpenuhi, baru nilai X nya bertambah (X = 2) dan nilai Y dimulai lagi dari awal (Y = 1).

Bentuk ini serupa dg LOOP di dalam LOOP (Nested Loop).

IV. EXIT Verb

Di dalam suatu paragraph yang diproses oleh statement PERFORM, jika ada loncatan ke akhir paragraph, maka harus ada suatu paragraph yang dituju, karena statement loncatan (GOTO atau IF) selalu diikuti nama paragraph.

Padahal loncatan ini maksudnya adalah ke akhir paragraph yang diproses oleh PERFORM, maka harus dibuat suatu paragraph dummy yang berisi statement EXIT, karena loncatan tidak boleh keluar dari paragraph yang diproses oleh PERFORM