

## ☛\* Statemen RESTORE

digunakan untuk mengembalikan pointer ke awal DATA sehingga data dapat dibaca kembali.

Bentuk umum :

RESTORE [ no baris | label ]

Keterangan :

Jika tidak menggunakan no baris atau label, maka pointer kembali ke awal DATA, tetapi jika no baris atau label disebutkan, maka pointer akan kembali tepat pada DATA yang ditunjukkan oleh no baris atau label tersebut.

Contoh :

```
READ A, B      → A = 1, B = 2
C = A + B
READ A, B      → A = 3, B = 4
D = A * B
RESTORE        → pointer kembali ke DATA pada no 10
READ A, B      → A = 1, B = 2
E = A + B
PRINT C, D, E
END
10 DATA 1,2
20 DATA 3,4
```

Hasilnya :        3        12        3

Contoh :

```
10 DATA 1,2,3
20 DATA 5,6
30 DATA 7,8
READ A, B      → A = 1, B = 2
READ C, D      → C = 3, D = 5
PRINT A, B, C, D
READ A, B, C    → A = 6, B = 7, C = 8
RESTORE 20     → pointer ke DATA pada baris 20
READ D, E      → D = 5, E = 6
PRINT A, B, C, D, E
END
```

Hasil :

```
1        2        3        5
6        7        8        5        6
```

## ☛\* Statemen INPUT

digunakan untuk membaca nilai data yang diketikkan lewat keyboard dan memberikan nilai tersebut pada suatu variabel yang ditulis dibelakang statemen INPUT.

Bentuk Umum :

INPUT [:] [Prompt { ; | , } ] daftar

Keterangan :

[:] digunakan jika input berikutnya ditampilkan pada baris yang sama dengan Input sebelumnya.

[Prompt] digunakan untuk menampilkan tulisan atau keterangan tentang hak yang akan di input.

Daftar adalah variabel yang digunakan sebagai tempat bagi nilai yang di input.

Contoh :

```
INPUT bil1%, bil2, kalimatsatu$
```

```
INPUT kalimatdua$
```

```
PRINT bil1%, bil2, kalimatsatu$, kalimat2$
```

```
END
```

Hasilnya :

```
? 8.6,67,Microsoft
```

```
? "satu,dua,tiga"
```

```
9      67      Microsoft      satu,dua,tiga
```

Contoh :

```
INPUT "Nama anda = ", nama$
```

```
INPUT ; "Alamat = ", alamat$
```

```
INPUT "      Kota = ", kota$
```

```
END
```

Hasilnya :

```
Nama anda = Iwan
```

Alamat = Cinere      Kota = Jakarta

## STATEMEN KENDALI

Statemen kendali merupakan statemen yang digunakan untuk mengalihkan urutan pelaksanaan suatu instruksi.

Statemen kendali dalam QBASIC dibagi menjadi 4 kelompok :

1. Statemen kendali tak bersyarat :  
GOTO , GOSUB ... RETURN
2. Statemen kendali bersyarat :  
IF...THEN...ELSE , SELECT...CASE
3. Statemen kendali proses berulang :  
FOR...NEXT , DO...LOOP , WHILE...WEND
4. Statemen kendali terhitung :  
ON GOTO , ON GOSUB, ON ERROR GOTO , ON ERROR GOSUB

### ●\* Statemen GOTO

berguna untuk melompati proses eksekusi ke suatu baris yang diberi nomor atau label tanpa adanya kondisi.

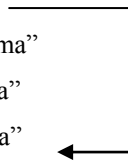
Baris yang dituju untuk dieksekusi ditandai dengan nomor baris atau label.

Bentuk Umum :

GOTO { no baris | label }

Contoh : Menggunakan nomor baris

```
CLS
PRINT "...mencoba statemen GOTO..."
PRINT
GOTO 10
PRINT "Data Pertama"
PRINT "Data Kedua"
10 PRINT "Data Ketiga"
END
```

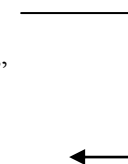


Hasilnya :

Data Ketiga

Contoh : menggunakan label

```
CLS
PRINT "...mencoba statemen GOTO..."
PRINT
GOTO Awal
PRINT "Data Pertama"
PRINT "Data Kedua"
Awal : PRINT "Data Ketiga"
END
```



Hasilnya :

..mencoba statemen GOTO...

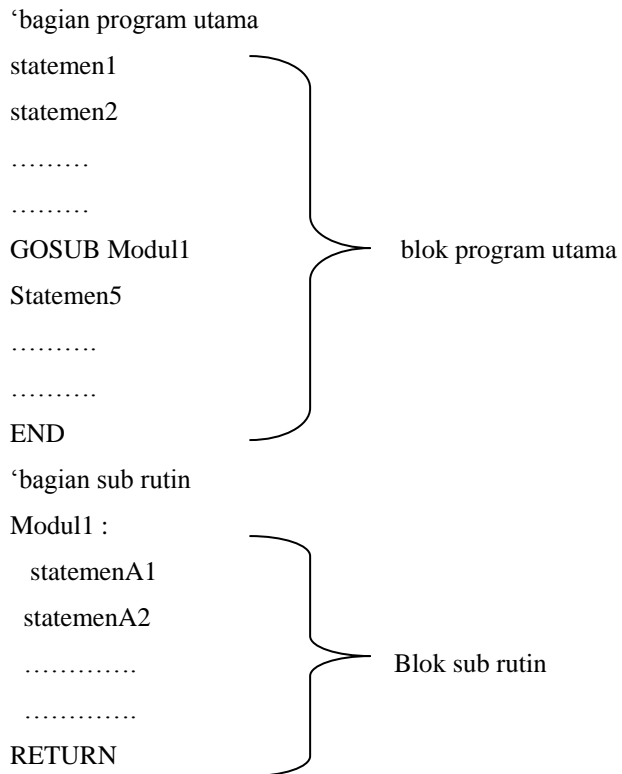
Data Ketiga

Catatan : Statemen GOTO menyebabkan proses program melompat ke statemen PRINT “Data Ketiga” dan proses berlanjut ke statemen END.

### ❖\* Statemen GOSUB...RETURN

kegunaannya sama dengan statemen GOTO, tetapi pada GOSUB...RETURN program terpecah menjadi 2 bagian, yaitu program utama (main program) dan program bagian (sub rutin).

Bagan GOSUB...RETURN :



Pada bagan di atas, program terbagi 2, bagian pertama merupakan program utama dan bagian kedua merupakan sub rutin. Nama sub rutin yang digunakan adalah Modul1. Nama sub rutin dapat ditentukan secara bebas.

Instruksi didalam sub rutin dilaksanakan setelah adanya pemanggilan dari program utama. Pemanggilan sub rutin dituliskan dengan menuliskan nama sub rutin pada statemen GOSUB.

Statemen RETURN menyebabkan pelaksanaan kembali ke program utama. Proses selanjutnya yang akan dikerjakan adalah statemen yang terletak sesudah statemen GOSUB.

Contoh :

```
‘program utama.....’
CLS
PRINT “Hasil dari program utama “ : PRINT
GOSUB Modul1
PRINT “Hasil dari program utama “ : PRINT
GOSUB Modul2
PRINT “Hasil dari program utama “ : PRINT
GOSUB Modul1
END
‘program sub rutin.....’
Modul1 :
  PRINT “-----“
  PRINT “Hasil dari subrutin Modul1”
  PRINT “-----“ : PRINT
RETURN
```

Modul2 :

```
PRINT "*****"  
PRINT "Hasil dari subrutin Modul2"  
PRINT "*****" : PRINT  
RETURN
```

Hasil :

Hasil dari program utama

-----

Hasil dari sub rutin Modul1

-----

Hasil dari program utama

\*\*\*\*\*

Hasil dari sub rutin Modul2

\*\*\*\*\*

Hasil dari program utama

-----

Hasil dari sub rutin Modul1

-----

Contoh :

Nilai = 2

PRINT "Nilai pada program utama = "; Nilai

GOSUB Hitung

Nilai = 7

GOSUB Hitung

END

Hitung :

IF Nilai > 5 THEN RETURN

Pangkat = Nilai ^ 2

PRINT "Nilai pada sub rutin = "; Pangkat

RETURN

Hasil :

Nilai pada program utama = 2

Nilai pada sub rutin = 4

## ☛ \* Statemen IF...THEN...ELSE

Terbagi menjadi 2 bentuk :

1. IF...THEN...ELSE satu baris
2. IF...THEN...ELSE banyak baris
- 3.

### ☛ IF...THEN...ELSE satu baris

Bentuk umum :

IF kondisi THEN {statemen|nobrs1|GOTO lbl1} [ELSE {statemen|nobrs2|GOTO lbl2}]

Catatan :

nobrs = nomor baris

lbl = label

Keterangan :

Bentuk umum di atas berarti :

- jika kondisi bernilai **benar**, maka salah satu dari 3 pilihan dibelakang statemen THEN akan dikerjakan,
- tetapi jika salah, ada 2 kemungkinan yang bisa terjadi :
  - Kemungkinan 1 : jika dalam statemen IF terdapat statemen ELSE, maka salah satu dari 3 pilihan dibelakang statemen ELSE akan dikerjakan.
  - Kemungkinan 2 : jika statemen ELSE tidak ditulis, maka eksekusi langsung akan melompat ke baris dibawah statemen IF.

Gambar ;;;;;;

Contoh :

```
IF I >= kali THEN 20
```

Contoh :

```
IF I >= kali THEN GOTO 20 ELSE kali = kali + 1
```

Contoh :

```
IF I >= kali THEN
    GOTO 20
ELSE
    kali = kali + 1
```

} statemen salah, akan dianggap bentuk IF..THEN..ELSE banyak baris, bentuk ini harus diakhiri dengan END IF

Contoh :

```
IF (I <= 100) AND (I >= 80) THEN PRINT I
```

### ☛ IF...THEN...ELSE banyak baris

disebut juga dengan **BLOK IF..THEN..ELSE**

Bentuk Umum :

```
IF kondisi1 THEN
    Statemen_1
[ ELSE kondisi2 THEN
    [statemen_2] ]
.....
.....
[ ELSE
    [statemen_n] ]
```

END IF

Aturan penggunaan Blok IF...THEN...ELSE :

1. Dibelakang statemen THEN tidak boleh ada statemen apapun selain baris komentar. Jika ditulis suatu statemen, maka compiler akan menganggap sebagai IF..THEN..ELSE satu baris.
2. Kata ELSE, ELSE IF dan END IF hanya boleh diawali dengan nomor baris atau label baris. Jika tidak, maka kata ini harus merupakan kata awal dari baris tersebut.
3. Blok IF harus terletak sebagai statemen pertama dalam suatu baris.
4. Blok harus diakhiri dengan kata END IF.

Gambar;;;;;

Kondisi BENAR atau SALAH dalam IF..THEN..ELSE adalah menggunakan suatu operator. Operator dapat berupa operator logika, relasi maupun gabungan keduanya.

Di bawah ini adalah Tabel Kebenaran untuk operasi logika :

X	Y	AND	OR	XOR	EQV	IMP
T	T	T	T	F	T	T
T	F	F	T	T	F	F
F	T	F	T	T	F	T
F	F	F	F	F	T	T

Contoh program IF...THEN...ELSE banyak baris

Persoalan kondisi :

Nilai angka lebih besar dari 90, maka konversi le nilai A

Nilai angka antara 75 sampai 90, maka konversi ke nilai B

Nilai angka antara 65 sampai 74, maka konversi ke nilai C

Nilai angka antara 55 sampai 64, maka konversi ke nilai D

Nilai angka kecil dari 55, maka konversi ke nilai F

Programnya :

```
IF NilaiAngka > 90 THEN
```

```
    NilaiHuruf$ = "A"
```

```
ELSE IF NilaiAngka >= 75 THEN
```

```
    NilaiHuruf$ = "B"
```

```
ELSE IF NilaiAngka >= 65 THEN
```

```
    NilaiHuruf$ = "C"
```

```
ELSE IF NilaiAngka >= 55 THEN
```

```
    NilaiHuruf$ = "D"
```

```
ELSE
```

```
    NilaiHuruf$ = "E"
```

```
END IF
```

Alternatif lain :

```
IF NilaiAngka > 90 THEN
```

```
    NilaiHuruf$ = "A"
```

```
ELSE IF (NilaiAngka <= 90) AND (NilaiAngka >= 75) THEN
```

```
    NilaiHuruf$ = "B"
```

```
ELSE IF (NilaiAngka <= 74) AND (NilaiAngka >= 65) THEN
```

```
    NilaiHuruf$ = "C"
```

```
ELSE IF (NilaiAngka <= 64) AND (NilaiAngka >= 55) THEN
```

```
    NilaiHuruf$ = "D"
```

```
ELSE
```

```
NilaiHuruf$ = "E"
```

```
END IF
```

### ●\* **Statemen SELECT...CASE**

mempunyai kegunaan yang hampir sama dengan statemen IF...THEN...ELSE banyak baris.

Perbedaannya :

Pada statemen SELECT...CASE kondisi yang di test hanya sebuah, dan proses eksekusi akan diteruskan ke bagian tertentu dari suatu program berdasar nilai kondisi yang di test. Pada statemen IF...THEN...ELSE banyak baris dapat men test lebih dari sebuah kondisi yang satu sama lain saling berbeda.

Bentuk Umum :

```
SELECT CASE unkapam
  CASE hasil1
    Statemen_1
  CASE hasil2
    Statemen_2
  .....
  .....
  CASE hasiln
    Statemen_n
  CASE ELSE
    Statemenm
END SELECT
```

Gambarf...

Contoh : Perbandingan program menggunakan IF..THEN..ELSE dengan SELECT..CASE

```
INPUT X%
IF X% = 1 THEN
  PRINT "SATU"
IF X% = 2 THEN
  PRINT "DUA"
IF X% = 3 THEN
  PRINT "TIGA"
ELSE
  PRINT "SALAH"
END IF
```

```
INPUT X%
SELECT CASE X%
  CASE 1
    PRINT "SATU"
  CASE 2
    PRINT "DUA"
  CASE 3
    PRINT "TIGA"
  CASE ELSE
    PRINT "SALAH"
END SELECT
```



Untuk menentukan ungkapan yang mempunyai jangkauan tertentu bisa digunakan bentuk umum sebagai berikut :

CASE ungkapan TO ungkapan

CASE IS oprelasi ungkapan

Oprelasi : sembarang ungkapan relasi ( <, <=, >, >=, =, <>)

Contoh :

```
INPUT NilAngka
```

```
SELECT CASE NilAngka
```

```
  CASE IS > 90
```

```
    NilaiHuruf$ = "A"
```

```
  CASE 75 TO 90
```

```
    NilaiHuruf$ = "B"
```

```
  CASE 65 TO 74
```

```
    NilaiHuruf$ = "C"
```

```
  CASE 55 TO 64
```

```
    NilaiHuruf$ = "D"
```

```
  CASE ELSE
```

```
    NilaiHuruf$ = "E"
```

```
END SELECT
```

```
PRINT "Nilai Huruf: "; NilaiHuruf$
```

## ●\* Statemen KALANG

digunakan untuk proses yang berulang, yaitu memproses sejumlah statemen beberapa kali sampai suatu kondisi dipenuhi atau sebaliknya.

Kombinasi yang digunakan adalah IF...THEN...ELSE dan GOTO

Contoh :

```
JumData% = 0
```

```
INPUT "Banyak data : "; N%
```

```
I% = 1
```

```
IsiData :
```

```
  PRINT "Isikan data ke "; I%; ":"
```

```
  INPUT DataKe%
```

```
  JumData% = JumData% + DataKe%
```

```
  IF I% = N% THEN GOTO CetakHasil
```

```
  I% = I% + 1
```

```
  GOTO IsiData
```

```
CetakHasil :
```

```
  PRINT "Jumlah semua : "; JumData%
```

```
  PRINT "Selesai"
```

```
END
```