TABEL PADA COBOL

OBJEKTIF :

1. Mahasiswa Mengetahui Pengetahuan Dasar Tentang Bahasa COBOL.
2. Mahasiswa Mampu Menginstall dan Menjalankan Aplikasi OpenCobolIDE.
3. Mahasiswa Mampu Memahami Struktur Program Bahasa COBOL.

Array dalam COBOL dikenal sebagai tabel. Tabel adalah struktur data linier dan merupakan kumpulan item data individual dengan tipe yang sama. Item data dari tabel diurutkan secara internal.

Pendeklarasian tabel

Pendefinisian tabel berada pada DATA DIVISION di WORKING-STORAGE SECTION. Dalam pemrograman COBOL, klausa Occurs digunakan untuk mendefinisikan tabel. Klausa Occurs menunjukkan pengulangan definisi nama data. Ini dapat digunakan hanya dengan angka level mulai dari 02 hingga 49.

9.1 Tabel Berdimensi Satu

Dalam tabel satu dimensi, klausa OCCURS hanya digunakan sekali dalam deklarasi. Suatu tabel berdimensi satu, yang memiliki elemen bernama Nilai dengan N buah elemen, secara fisik dapat digambarkan sebagai berikut :

Nilai (1)	Nilai (2)	Nilai (3)	- - -	Nilai (N)
x	x	x		x

Indeks dari elemen suatu array menyatakan posisinya dalam urutan secara umum dalam suatu array.

Contoh program tabel berdimensi satu :

```
IDENTIFICATION DIVISION.
PROGRAM-ID. INPUT-NAMAMHS.
DATA DIVISION.
WORKING-STORAGE SECTION.
77 n PIC 99.
01 TABEL-NAMA-MHS.
  02 NAMA-MHS PIC X(20) OCCURS 5 TIMES.

PROCEDURE DIVISION.
PROGRAM-UTAMA.
            PERFORM INPUT-NAMA-MHS
                VARYING n FROM 1 BY 1 UNTIL n > 5.
            PERFORM TAMPILKAN-NAMA-MHS
                VARYING n FROM 1 BY 1 UNTIL n > 5.
STOP RUN.

INPUT-NAMA-MHS.
            DISPLAY 'MASUKKAN NAMA : '.
```

```
                    ACCEPT NAMA-MHS(n).

            TAMPILKAN-NAMA-MHS.
                    DISPLAY 'NAMA MAHASISWA KE-',n, ':' NAMA-MHS(n).
```

Pada program di atas akan mencetak output :

```
MASUKKAN NAMA :
ZIA
MASUKKAN NAMA :
RAHMA
MASUKKAN NAMA :
RINI
MASUKKAN NAMA :
CAHYA
MASUKKAN NAMA :
RASYID
NAMA MAHASISWA KE-01:ZIA
NAMA MAHASISWA KE-02:RAHMA
NAMA MAHASISWA KE-03:RINI
NAMA MAHASISWA KE-04:CAHYA
NAMA MAHASISWA KE-05:RASYID
```

Klausa Occurs 5 Times menunjukkan pengulangan datanames NAMA-MHS sebanyak 5 kali. Sehingga output yang dihasilkan yaitu berupa inputan untuk memasukkan nama mahasiswa dan menampikan nama mahasiswa sebanyak 5 mahasiswa. Program tersebut secara fisik dapat diilustrasikan sebagai berikut :

Nama Mahasiswa ke-01	Nama Mahasiswa ke-02	Nama Mahasiswa ke-03	Nama Mahasiswa ke-04	Nama Mahasiswa ke-05
ZIA	RAHMA	RINI	CAHYA	RASYID

9.2 Tabel Berdimensi Dua

Tabel berdimensi dua adalah sekumpulan nilai data yang membentuk tabel, yang elemen masing-masing data ditunjukkan oleh 2 subscript. Tabel ini membentuk sebuah matriks, dimana salah satu subscriptnya menunjukkan posisi kolom, dan subscript yang satunya menunjukkan posisi barisnya. Klausa OCCURS hanya digunakan dua kali dalam deklarasi tabel berdimensi dua.

Contoh program tabel berdimensi dua :

```
        IDENTIFICATION DIVISION.
        PROGRAM-ID. TABEL-INPUT.
        DATA DIVISION.
        WORKING-STORAGE SECTION.
        01 TABEL-NILAI-MAHASISWA.
            02 MAHASISWA OCCURS 3 TIMES.
              03 NILAI OCCURS 2 TIMES.
              04 NILAI-MAHASISWA PIC 9(3).
```

```
       01 SUBSCRIPT-TABEL.
           02 SUBSCRIPT-MAHASISWA PIC 9.
           02 SUBSCRIPT-NILAI PIC 9.

       PROCEDURE DIVISION.
       PROGRAM-UTAMA.
                  PERFORM INPUT-TABEL.
                  PERFORM TAMPIL-TABEL.
       STOP RUN.

       INPUT-TABEL.
                  DISPLAY 'INPUT DATA NILAI MAHASISWA :'.
                  PERFORM ISI-DATA
                      VARYING SUBSCRIPT-MAHASISWA FROM 1 BY 1
                      UNTIL SUBSCRIPT-MAHASISWA > 3
                      AFTER SUBSCRIPT-NILAI FROM 1 BY 1
                      UNTIL SUBSCRIPT-NILAI > 2.

       ISI-DATA.
                  DISPLAY 'Mahasiswa ', SUBSCRIPT-MAHASISWA,
                  ' Nilai ', SUBSCRIPT-NILAI, ':'.
                  ACCEPT NILAI-MAHASISWA(SUBSCRIPT-MAHASISWA,SUBSCRIPT-NILAI).

       TAMPIL-TABEL.
                  DISPLAY 'MENAMPILKAN ISI TABEL NILAI MAHASISWA : '.
                  PERFORM DISPLAY-DATA
                      VARYING SUBSCRIPT-MAHASISWA FROM 1 BY 1
                      UNTIL SUBSCRIPT-MAHASISWA > 3
                      AFTER SUBSCRIPT-NILAI FROM 1 BY 1
                      UNTIL SUBSCRIPT-NILAI > 2.
       DISPLAY-DATA.
                  DISPLAY 'Mahasiswa ', SUBSCRIPT-MAHASISWA,
                  ' Nilai ', SUBSCRIPT-NILAI, ':',
                  NILAI-MAHASISWA(SUBSCRIPT-MAHASISWA, SUBSCRIPT-NILAI).
                  DISPLAY ' '.
```

Pada program di atas, terdapat penggunaan klausa OCCURS sebanyak 2 kali. Klausa OCCURS pertama digunakan untuk datanames MAHASISWA sebanyak 3 kali pengulangan item data, dan klausa OCCURS kedua digunakan untuk datanames NILAI sebanyak 2 kali pengulangan item data.

Output yang dihasilkan dari program tersebut yaitu berupa inputan memasukkan nilai dari tiap mahasiswa dan menampilkan isi tabel mahasiswa. Program tersebut dapat diilustrasikan sebagai berikut :

MAHASISWA 1		MAHASISWA 2		MAHASISWA 3	
NILAI 1	NILAI 2	NILAI 1	NILAI 2	NILAI 1	NILAI 2
90	80	80	70	90	100

9.3 Tabel Berdimensi Tiga

Tabel berdimensi tiga menyangkut penggunaan 3 buah subscript. Pada tabel berdimensi satu, digunakan klausa OCCURS satu tingkat. Pada tabel berdimensi dua, digunakan klausa OCCURS sebanyak 2 tingkat. Pada tabel berdimensi tiga digunakan klausa OCCURS sebanyak 3 tingkat. Tabel berdimensi tiga adalah tingkatan paling maksimum dalam dimensi pembuatan tabel dalam pemrograman COBOL.

Contoh Program tabel berdimensi tiga :

```
IDENTIFICATION DIVISION.
PROGRAM-ID. TABEL-BERDIMENSI-TIGA.
DATA DIVISION.
WORKING-STORAGE SECTION.
  01 DATA-DOSEN-TABLE.
     05 DOSEN OCCURS 2 TIMES.
       10 DOSEN PIC A(20) VALUE ' DOSEN'.
         10 DATADOSEN OCCURS 1 TIMES.
           15 KD-MK PIC X(20) VALUE ' KODE MATAKULIAH'.
           15 NAMA-MK PIC X(20) VALUE ' NAMA MATAKULIAH'.
           15 SKS OCCURS 1 TIMES.
             20 SKS PIC X(12) VALUE 'JUMLAH SKS'.
             20 KELAS PIC X(12) VALUE 'JUMLAH KELAS'.

PROCEDURE DIVISION.
MULAI.
        DISPLAY DATA-DOSEN-TABLE.
SELESAI.
 STOP RUN.
```

Pada tabel di atas, elemen DOSEN menggunakan OCCURS 2 TIMES dimana terjadi pengulangan item data sebanyak 2 kali. Didalam elemen DOSEN menggunakan klausa OCCURS 1 TIMES, artinya dalam data-item DOSEN hanya terjadi 1 kali pengulangan data-item KODE MATAKULIAH dan NAMA MATAKULIAH. Didalam item data NAMA-MATAKULIAH, menggunakan klausa OCCURS 1 TIMES untuk data-item JUMLAH SKS dan JUMLAH KELAS.

Program tersebut secara fisik dapat diilustrasikan sebagai berikut :

<table>
<tr><th colspan="3">DOSEN(1)</th><th colspan="3">DOSEN(2)</th></tr>
<tr><td rowspan="2">KODE MATAKULIAH</td><td colspan="2">NAMA MATAKULIAH</td><td rowspan="2">KODE MATAKULIAH</td><td colspan="2">NAMA MATAKULIAH</td></tr>
<tr><td>JUMLAH SKS</td><td>JUMLAH KELAS</td><td>JUMLAH SKS</td><td>JUMLAH KELAS</td></tr>
</table>

Penggunaan Klausa OCCURS ... DEPENDING ON

Jumlah elemen dalam tabel biasanya bervariasi. Jika hanya menggunakan klausa OCCURS, jumlah elemen didalam tabel akan bersifat tetap sesuai dengan nilai integer TIMES yang dideklarasikan pada WORKING-STORAGE Section. Tetapi dengan menggunakan klausa OCCURS ... DEPENDING ON maka jumlah elemen didalam tabel dapat dibuat bervariasi menyesuaikan dari suatu nilai nama-data tersebut.

Contoh :

```
01 RECORD-KARYAWAN.
    02 JUMLAH-DATA PIC 9(2).
    02 NILAI-DATA OCCURS 1 TO 100 TIMES
         DEPENDING ON JUMLAH-DATA.
        03 NAMA PIC A(20).
        03 GAJI PIC 9(6).
```

Klausa OCCURS ... DEPENDING ON menunjukkan meskipun jumlah elemen maksimum dalam tabel adalah 100 elemen, tetapi jumlah elemen tabel akan menyesuaikan dengan jumlah elemen nilai JUMLAH-DATA.

9.4 Subscript

Subscript adalah item data integer yang menjelaskan jumlah kemunculan elemen array dalam tabel. Subscript bukan lokasi memori aktual, melainkan merujuk pada posisi item data atau lokasi memori item data tertentu berada.

Untuk merujuk ke masing-masing elemen tabel diperlukan sebuah subscript yg dapat berupa literal numerik (bilangan bulat positif) ataupun suatu identifier yg berisi bilangan bulat positif.

Contoh :

```
 01 tabel-nilai-siswa.
    02 nilai-siswa  PIC 999 OCCURS 5 TIMES.
```

Pada contoh tabel-nilai-siswa, menyatakan array nilai-siswa yang menampung 5 data item numerik.

Pada array tersebut, memiliki nilai subscript antara 1 s/d 5. Subscript 1 menunjukkan elemen tabel yang pertama, subscript 2 menunjukkan elemen tabel yang kedua, dan seterusnya.

Contoh program pengaksesan subscript :

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SUBSCRIPT.

DATA DIVISION.
WORKING-STORAGE SECTION.
   01 SUBSCRIPT-TABLE.
     05 WS-A OCCURS 3 TIMES.
        10 ANGKA PIC A(2).
        10 WS-B OCCURS 2 TIMES.
          15 HURUF PIC X(3).

PROCEDURE DIVISION.
       MOVE '12ABCDEF34GHIJKL56MNOPQR' TO SUBSCRIPT-TABLE.
```

```
            DISPLAY 'SUBSCRIPT-TABLE : ' SUBSCRIPT-TABLE.
            DISPLAY 'WS-A(1)    : ' WS-A(1).
            DISPLAY 'WS-B(1,1)  : ' WS-B(1,1).
            DISPLAY 'WS-B(1,2)  : ' WS-B(1,2).
            DISPLAY 'WS-A(2)    : ' WS-A(2).
            DISPLAY 'WS-B(2,1)  : ' WS-B(2,1).
            DISPLAY 'WS-B(2,2)  : ' WS-B(2,2).
            DISPLAY 'WS-A(3)    : ' WS-A(3).
            DISPLAY 'WS-B(3,1)  : ' WS-B(3,1).
            DISPLAY 'WS-B(3,2)  : ' WS-B(3,2).
        STOP RUN.
```

Output yang dihasilkan yaitu :

```
SUBSCRIPT-TABLE : 12ABCDEF34GHIJKL56MNOPQR
WS-A(1)    : 12ABCDEF
WS-B(1,1)  : ABC
WS-B(1,2)  : DEF
WS-A(2)    : 34GHIJKL
WS-B(2,1)  : GHI
WS-B(2,2)  : JKL
WS-A(2)    : 56MNOPQR
WS-B(3,1)  : MNO
WS-B(3,2)  : PQR
```

9.5 Index

Bila subscript digunakan untuk menunjukan suatu elemen dalam tabel, maka compiler COBOL harus menghasilkan suatu instruksi bahasa mesin untuk merubah nilai dari subscript ke alamat memori yang sebenarnya dari elemen tabel yang dikehendaki.

Suatu index dapat digunakan menggantikan subscript untuk membuat perhitungan alamat memori sebenarnya dari elemen tabel yang lebih efisien.

Index pada tabel tidak bisa dimanipulasi dengan cara yang sama dengan menggunakan subscript. Statement khusus yang digunakan pada index ini adalah SET dan SEARCHElemen tabel juga dapat diakses menggunakan index. Index adalah perpindahan elemen dari awal tabel. Index dinyatakan dengan klausa OCCURS dan menggunakan klausa INDEXED BY.

Bentuk Umum :

```
        OCCURS TableSize TIMES
            [ INDEXED BY Index-name ]
```

Pemberian nama index harus bersifat unik, tidak boleh sama dengan nama index lainnya. Nilai index tidak dapat dimanipulasi dengan menggunakan kata kerja COBOL biasa seperti MOVE, ADD, dan SUBSTRACT. Untuk mengubah nilai index hanya dapat diubah menggunakan pernyataan SEARCH, SEARCH ALL, SET dan opsi PERFORM Varying.

Contoh program index :

```
        IDENTIFICATION DIVISION.
        PROGRAM-ID. PROGRAM-INDEX.
        DATA DIVISION.
        WORKING-STORAGE SECTION.
```

```
       01 INDEX-TABLE.
           05 WS-A OCCURS 3 TIMES INDEXED BY I.
             10 WS-B OCCURS 2 TIMES INDEXED BY J.
                15 ISI-TABEL PIC X(3).

       PROCEDURE DIVISION.
              MOVE 'ABCDEFGHIJKLMNOPQR' TO INDEX-TABLE.
              PERFORM A VARYING I FROM 1 BY 1 UNTIL I>1.
           STOP RUN.
              A.
              PERFORM C VARYING J FROM 1 BY 1 UNTIL J>1.
              C.
                  DISPLAY WS-B(2,1).
```

Output yang dihasilkan yaitu GHI, karena pada statement display menampilkan index (2,1) pada isi tabel.

9.6 SET & SEACRH

SET Verb

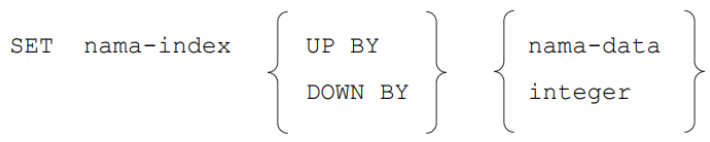
Dalam Bahasa pemrogaman COBOL, Set verb digunakan untuk mengubah, menginisialisasi, menambah, atau mengurangi nilai indeks. Set verb dapat digunakan bersamaan dengan SEARCH atau SEARCH ALL untuk menemukan elemen dalam tabel.

Bentuk umum SET statement untuk penanganan tabel dasar :

Bentuk set statement di atas akan memberikan nilai awal index-name2 atau identifier2 atau integer ke field penerima yang ditulis setelah SET verb.

Bila nama-data2 digunakan, maka nama ini harus sudah disebutkan sebagai elementary data-item yang bernilai numerik integer atau sebagai USAGE IS INDEX. Bila index-name2 digunakan, maka nama index ini harus disebutkan oleh INDEXED BY di dalam OCCURS clause. Bila integer digunakan, maka harus berbentuk nilai integer positif.

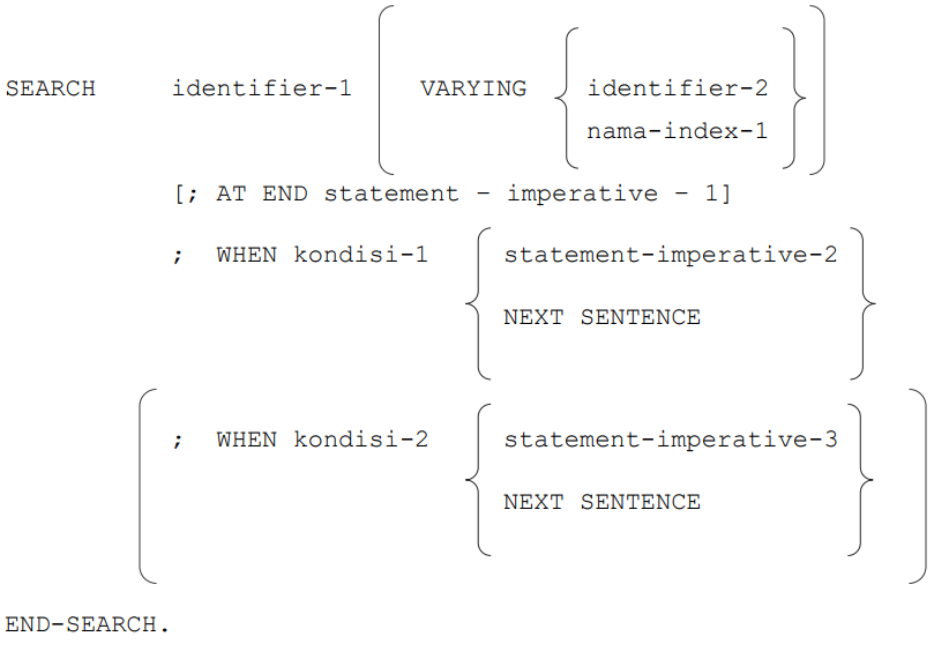
Bentuk penggunaan SET verb untuk menambah atau mengurangi nilai dari index :



Bentuk di atas digunakan untuk menambah (UP BY) atau mengurangi (DOWN BY) nilai dari nama index dengan nilai nama-data atau nilai integer.

SEARCH Verb

Pencarian nilai elemen tabel tertentu dapat dilakukan dengan statement SEARCH. Pencarian ini dapat dilakukan pada tabel disortir maupun tidak disortir. Ini hanya digunakan untuk tabel yang dideklarasikan oleh frase Index. Dimulai dengan nilai awal index. Jika elemen yang dicari tidak ditemukan, maka index secara otomatis bertambah dengan 1 dan berlanjut hingga akhir tabel.



Identifier-1 merupakan nama data-item yang telah disebutkan di dalam DATA DIVISION pada klausa OCCURS yang diikuti oleh klausa INDEXED BY. VARYING menunjukkan urutan pencariannya. AT END menunjukkan jika elemen yang dicari tidak ditemukan, maka statement-imperative-1 akan diproses. WHEN menunjukkan jika kondisi pencarian yang diinginkan terpenuhi atau pencarian ditemukan, maka statement-imperative yang mengikutinya atau NEXT SENTENCE yang diproses. Bentuk statemen SEARCH ini akan melakukan pencarian dengan metode linear search.

SEARCH ALL Verb

Search all verb adalah metode pencarian biner, yang digunakan untuk menemukan elemen di dalam tabel. Tabel harus dalam urutan untuk penggunaan search all verb. Indeks tidak memerlukan inisialisasi. Dalam pencarian biner, tabel dibagi menjadi dua bagian dan menentukan di mana setengah elemen yang akan dicari. Proses ini berulang hingga elemen ditemukan atau akhirnya tercapai.

Bentuk umum SEACH ALL statement :

```
SEARCH ALL      identifier-1     [; AT END statement-imperative-1]

      ;WHEN    { nama-data-1     { IS EQUAL TO }  { identifier-2          }  }
               { Nama-kondisi-1  {      IS     }  {      literal-1         }  }
                                                 { Ungkapan aritmatika-1  }

    [ AND      { nama-data-2     { IS EQUAL TO }  { identifier-3          }  }
               { Nama-kondisi-2  {      IS     }  { literal-2            }  }
                                                 { Ungkapan aritmatika-2  }  ]

    { Statement-imperative-2 }
    { NEXT SENTENCE          }
END-SEARCH.
```

Bentuk SEARCH ALL statement dipergunakan pada tabel yang telah urut, yaitu tabel yang dibentuk oleh klausa OCCURS yang mengandung klausa KEY sebagai berikut :

```
[ ; OCCURS      integer-1  TIMES
  [ { ASCENDING  }
    { DESCENDING } KEY  IS nama-data-1 [ , nama-data-2 ] ….

    [ INDEXED BY  nama-index-1  [ , nama-index-2 ] ……. ]  ]
```

Pada bentuk statement ini, hanya sebuah WHEN option yang diperbolehkan, tetapi kondisi dapat berbentuk jamak dengan mempergunakan operator logika AND (operator logika lainnya tidak dijinkan).

Juga pada WHEN option, hanya diperbolehkan kondisi yang menyeleksi kesamaan, menggunakan relational operator = atau IS EQUAL TO saja (relational operator yang lain seperti misalnya >, < đan lainnya tidak diijinkan) atau diperbolehkan menggunakan kondisi dan nama-kondisi (yang disebutkan di DATA DIVISION pada level number 88), yang hanya mengandung satu nilai saja.

Nama-nama data setelah WHEN option hanya telah disebutkan KEY IS clause pada OCCURS clause. Dan nama-kondisi setelah WHEN option juga harus dihubungkan sebagai nama-data di KEY IS clause. Metode pencarian yang dipergunakan pada bentuk statement ini yaitu binary search methods.

Contoh program set dan search :

```
        IDENTIFICATION DIVISION.
        PROGRAM-ID. SET-SEARCH.
        DATA DIVISION.
        WORKING-STORAGE SECTION.
        01 WS-TABLE.
```

```
           05 WS-RECORD PIC A(1) OCCURS 6 TIMES INDEXED BY I.
       01 WS-SEARCH PIC A(1) VALUE 'A'.

       PROCEDURE DIVISION.
              MOVE 'ABCDEF' TO WS-TABLE.
              SET I TO 1.
              SEARCH WS-RECORD
                  AT END DISPLAY 'A TIDAK DITEMUKAN DI DALAM TABEL'
                  WHEN WS-RECORD(I) = WS-SEARCH
                  DISPLAY 'HURUF A DITEMUKAN DI DALAM TABEL'
              END-SEARCH.

       STOP RUN.
```

Output yang dihasilkan yaitu HURUF A DITEMUKAN DI DALAM TABEL'.