

PROCEDURE DIVISION

INPUT/OUTPUT statement

ACCEPT statement

untuk memasukkan data, dari layarsewaktu program tsb dijalankan. Pada umumnya data tsb dalam jumlah relatif kecil dan tidak didefinisikan sebagai file.

Bentuk umum : ACCEPT nama-data.

Contoh :

```
ACCEPT DATA-NILAI.  
ACCEPT TANGGAL FROM DATE.
```

DISPLAY statement

untuk melakukan transfer data terhadap suatu hasil pemrosesan dari memori komputer ke dalam media keluaran. Dimana hasil tsb tidak didefinisikan sebagai file.

Bentuk umum :

```
DISPLAY besaran [UPON nama-alat-output].  
nama-data
```

besaran dapat berupa numerik literal atau non numerik literal

Contoh :

```

DISPLAY "JUDUL".
DISPLAY NAMA_MHS NPM "nILAI :" NILAI UPON ALAT-CETAK.
DISPLAY (15, 5) "LAPORAN"
    
```

OPEN statement

untuk mempersiapkan/membuka file yang digunakan dalam program.

Bentuk umum :

```

OPEN      INPUT      nama-file
          OUTPUT     nama-file
          I-O        nama-file
    
```

Contoh :

```

OPEN      INPUT FILE-MHS
          OUTPUT REPORT-MHS.

OPEN      I-O  INDUK-MHS.
    
```

READ statement

untuk membaca data/record yang ada dalam suatu file.

Bentuk umum :

```

READ nama-file [RECORD [INTO nama-record]
               AT END
               [          tindakan ] ]
               INVALID KEY
    
```

Contoh :

```
READ FILE-MHS AT END MOVE 0 TO AKHIR.
READ FILE-MHS AT END GO TO HABIS.
```

WRITE statement

untuk melakukan transfer hasil pemrosesan yang didefinisikan sebagai record ke dalam media keluaran dalam bentuk file.

Bentuk umum :

```
WRITE nama-record [FROM nama-data-1]
```

```
BEFORE                nama-data-2    LINES
[  ADVANCING          integer    LINES  ]
  AFTER                PAGE
```

```
[  INVALID KEY tindakan  ]
```

Contoh :

```
WRITE REPORT-MHS FROM JUDUL-1.
WRITE REPORT-MHS FROM JUDUL-2 AFTER ADVANCING 2 LINES.
```

CLOSE statement.

untuk mengakhiri suatu proses input/output dari suatu file (semua file yang sudah tidak digunakan harus ditutup).

Bentuk umum :
CLOSE nama-file

Contoh :
CLOSE FILE-MHS, REPORT-MHS.

BRANCHING statement.

STOP statement

untuk mengakhiri pelaksanaan program.

Bentuk umum :
STOP RUN.

Contoh :
CLOSE FILE-MHS, REPORT-MHS.
STOP RUN.

GO TO statement

untuk mengalihkan pelaksanaan program, yang ditujukan pada suatu nama paragraph.

Bentuk umum :
GO TO nama-paragraph.

Contoh :

```

PARAGRAPH-1.
  ADD 1 TO HAL.
  GO TO PARAGRAPH-3.
PARAGRAPH-2.
  MOVE ZERO TO NOM.
PARAGRAPH-3.
  CLOSE FILE-MHS, REPORT-MHS.
  STOP RUN.
    
```

MANIPULATION statement.

***MOVE* statement.**

memindahkan suatu besaran/data dari satu field ke lokasi field yang lain sehingga input data dapat dimanipulasi untuk menghasilkan output.

Bentuk umum :

```

          nama-data-1
MOVE      TO  nama-data2 .....
          besaran
    
```

Contoh :

```

01  RECORD-MHS.
02  NPM      PIC X(7).
02  UJIAN-1  PIC 99.
.....
    
```

```

.....
MOVE ZERO TO UJIAN-1.
MOVE NPM TO NPM-O.
    
```

CONDITIONAL statement.

digunakan untuk melakukan pengetesan terhadap suatu kondisi tertentu.

Bentuk umum :

```

          stmt-1          ELSE stmt-2
IF kondisi
          NEXT SENTENCE ELSE NEXT SENTENCE
    
```

```

GREATER THAN >
operand-1    EQUAL TO    operand-2
              =
              LESS THAN
              <
    
```

Hubungan antar kondisi:

AND : tindakan akan dilaksanakan bila *semua* kondisi /syarat dipenuhi.

OR : tindakan akan dilaksanakan bila *salah satu* kondisi/syarat dipenuhi.

Contoh :

- *Jika JAM-KERJA lebih besar dari 40 ,
maka UANG-LEMBUR = (JAM-KERJA - 40) x UPAH
Kalau tidak UANG-LEMBUR = 0*
- ```

IF JK IS GREATER THAN 40
 COMPUTE UL = (JK - 40) * UPAH
ELSE
 MOVE ZERO TO UL.

```

*Kondisi Name Kondisi*

*Penggunaan level number 88 :*

|    |         |           |         |      |
|----|---------|-----------|---------|------|
|    |         | VALUE IS  | THROUGH |      |
| 88 | kondisi |           | It-1    | It-2 |
|    |         | VALUE ARE | THRU    |      |

Contoh :

- Jika Kode = 1 maka pendidikan terakhir Sarjana
- Jika Kode = 2 maka pendidikan terakhir D3
- Jika Kode = 3 maka pendidikan terakhir SLTA

```

01 REC-PEC.
 02
 02 KODE PIC 9.
 88 S1 VALUE IS 1.
 88 D3 VALUE IS 2.
 88 SLTA VALUE IS 3.
 IF KODE = 1
 MOVE "SARJANA" TO KET
 ELSE
 IF KODE = 2 MOVE "D3" TO KET
 ELSE MOVE "SLTA" TO KET.
atau
 IF S1 MOVE "SARJANA" TO KET
 ELSE
 IF D3 MOVE "D3" TO KET
 ELSE
 MOVE "SLTA" TO KET.

```