

BAB2

DIVISI-DIVISI PADA COBOL

Bentuk Umum dari IDENTIFICATION DIVISION

IDENTIFICATION DIVISION.
PROGRAM-ID. Nama program.
[AUTHOR. Nama programmer.]
[INSTALLATION. Nama instalasi.]
[DATE-WRITTEN. Tgl program dibuat.]
[DATE-COMPILED. Tgl program dikompilasi.]
[SECURITY. Sifat program.]

Bentuk Umum dari ENVIRONMENT DIVISION

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
[SOURCE- COMPUTER. Nama-komputer.]
[OBJECT-COMPUTER. Nama-komputer.]
SPECIAL-NAMES. PRINTER IS nama-mnemonik
CURRENCY SIGN IS literal
DECIMAL-POINT IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL. {file-control entry}

Bentuk umum paragraph FILE-CONTROL untuk organisasi file secara sequential

INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT nama-file ASSIGN TO DISK
 PRINTER
 ORGANIZATION IS LINE SEQUENTIAL
 ACCES MODE IS SEQUENTIAL
 FILE STATUS IS nama-data.

Bentuk umum paragraph FILE-CONTROL untuk organisasi file secara indexed

INPUT-OUTPUT SECTION.

```

FILE-CONTROL.
SELECT nama-file ASSIGN TO DISK
      ORGANIZATION IS INDEXED
                        SEQUENTIAL
      ACCESS MODE IS  RANDOM
                        DYNAMIC
      RECORD KEY IS  nama-data1
      FILE STATUS IS nama-data2.

```

Bentuk umum paragraph FILE-CONTROL untuk organisasi file secara relative

```

INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT nama-file ASSIGN TO DISK
      ORGANIZATION IS RELATIVE
      ACCESS MODE IS  SEQUENTIAL , RELATIVE KEY IS nama-data1
                        RANDOM
                        DYNAMIC , RELATIVE KEY IS nama-data1
      FILE STATUS IS nama-data2

```

**Bentuk Umum DATA DIVISION
DATA DIVISION.**

FILE SECTION.

```

[      file description entry]
[      record description entry] ...]

```

WORKING-STORAGE SECTION.

```

[      independent item ]
[      record description entry ]
[      data description entry ]

```

Penjelasan :

```

FILE SECTION.
FD  nama-file
    LABEL RECORD IS  STANDARD
                        OMITTED
    RECORD CONTAINS integer CHARACTERS
    DATA RECORD IS nama-record.
01  nama-record.
    level-number nama-field  PIC          simbol-picture
                        FILLER  PICTURE

```

Keterangan :

FILLER adalah suatu reserved word yang digunakan untuk pemberian nama suatu medan data, dimana medan data tsb tidak akan dipanggil dalam operasi/manipulasi data pada PROCEDURE DIVISION.

WORKING-STORAGE SECTION.

- mempersiapkan manipulasi/operasi terhadap data
- merancang bentuk hasil pemrosesan
- merancang bentuk 'Heading'
- menguraikan struktur record masukan dalam bentuk file, yang mana struktur record masukan belum didefinisikan dalam FILE SECTION atau dapat pada data masukan tsb bukan dalam bentuk file.

Independent Item :

- cara pendefinisian lokasi memori yang bukan dalam bentuk record terpisah struktur record lainnya.
misalnya : sebagai counter, indexmatriks, acuan dalam menentukan suatu kondisi
- indikator yang digunakan adalah level number 77.

Record description entry :

- pendefinisian data yang dikelompokkan dalam bentuk record

Data description entry :

- Suatu alat bantu dalam mengarahkan logika seorang pemrogram COBOL untuk mendapatkan suatu penyelesaian terhadap masalah yang dihadapi.
- Deskripsi data terbentuk dari beberapa clause :
 1. level number
 2. nama data
 3. FILLER
 4. PICTURE
 5. VALUE
 - untuk memberi harga pada suatu nama data ataupun FILLER, dapat berupa numeric literal, non numeric literal, figurative constant
 6. BLANK WHEN ZERO
 - sederetan reserved word yang dapat digunakan untuk menggantikan nilai nol pada suatu medan data dengan blank (kosong)
 7. *JUSTIFIED*
 - untuk menentukan apakah besaran yang dikandung oleh suatu nama data akan rapat kiri atau rapat kanan
 8. *OCCURS*
 - untuk mendefinisikan matriks atau tabel.

matriks adalah berupa sekumpulan medan data homogen yang dapat dicapai dengan suatu indeks (subscript).

9. REDEFINES

untuk mendefinisikan kembali satu/beberapa item (nama data) yang berarti memberikan identitas satu lokasi memori dengan dua nama.

10. RENAMES

dengan indikator level number 66 yang digunakan untuk pendefinisikan suatu lokasi memori berdasarkan nama data atau struktur yang mendahuluinya, hanya dapat dilakukan dalam satu record.

CONTOH :

penggunaan independent item :

WORKING-STORAGE SECTION.

77 NOMOR PIC 999 VALUE ZERO.

77 TOTAL PIC 999 VALUE 0.

77 AKHIR PIC 9.

88 DATA-MHS VALUE 0.

88 MASIH-ADA VALUE 1

penggunaan BLANK WHEN ZERO :

77 TOTAL PIC 999 VALUE ZERO, BLANK [WHEN] ZERO.

77 PEMBAYARAN PIC 9(5) BLANK ZERO.

Jika TOTAL bernilai 0 maka sepanjang 3 karakter kosong.

Jika PEMBAYARAN bernilai nol maka sepanjang 5 karakter kosong.

penggunaan OCCURS :

01 STATUS-PERKAWINAN.

02 STATUS PIC X(15) OCCURS 4 TIMES.

berarti STATUS-PERKAWINAN nama tabel/matriks yang mempunyai 4 elemen,

yaitu STATUS(1), STATUS(2), STATUS(3), STATUS(4).

penggunaan REDEFINES :

01 TABEL-BULAN.

10 FILLER PIC X(10) VALUE "JANUARI ".

10 FILLER PIC X(10) VALUE "FEBRUARI ".

10 FILLER PIC X(10) VALUE "MARET ".

01 MATRIKS-BULAN REDEFINES TABEL-BULAN.

10 BULAN PIC X(10) OCCURS 3 TIMES.

pendefinisian suatu lokasi memori dengan panjang 30 character yang digunakan untuk TABEL-BULAN. Kemudian lokasi tsb didefinisikan lagi dengan

suatu nama MATRIKS-BULAN. Dengan OCCURS clause, maka pencapaian elemen tabel dapat dilakukan dengan suatu matriks. Misalnya, untuk mendapatkan data "MARET" maka pencapaian data adalah dengan BULAN(3), untuk bulan "FEBRUARI" dengan BULAN(2).

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 :

MOVE nama-data-1
 besaran TO nama-data2

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 :

```
88  kondisi          VALUE IS          THROUGH          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
```



```

atau
ELSE MOVE "SLTA" TO KET.
IF S1 MOVE "SARJANA" TO KET
ELSE
IF D3 MOVE "D3" TO KET
ELSE
MOVE "SLTA" TO KET.

```

Level Number & Picture

PICTURE EDITING & PEMBULATAN LEVEL NUMBER

Adalah suatu nomor yang menunjukkan jenjang dari data/item dalam suatu record. Makin besar nomor levelnya, makin rendah jenjang tingkatannya.

LEVEL NUMBER YANG DIPAKAI :

- a. 01 = Untuk mengawali record description
- b. 02 - 49 = Untuk mengawali data item
- c. 66 = Untuk RENAMES
- d. 77 = Untuk mengawali item yang berdiri sendiri
- e. 88 = Untuk mengawali nama kondisi / pilihan

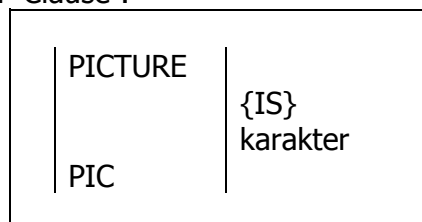
PICTURE CLAUSE

Picture clause memberikan informasi tentang bagaimana compiler COBOL mengatur memory untuk data yg dimasukkan.

Menerangkan data item yg digunakan :

- ukuran field
- type data
- numerik(9)
- alphanumeric(X)
- alphabet(A)

Bentuk dari Clause :



Data item yg berbentuk REPORT-EDITED-ITEM dapat menggunakan PICTURE karakter :

Z " DB * \$ - CR O . + B /

Contoh Program :

```

:
:
DATA DIVISION.
FILE SECTION.
FD      REKAMAN.
        LABEL RECORD IS STANDARD
        VALUE OF FILE-ID IS "MHS.DAT"
01      DATA-MHS.
        02      NPM          PIC X(8).
        02      NAMA         PIC X(20).
        02      NILAI-1      PIC 9(3).
        02      BAYARAN PIC 9(6)V99.
WORKING-STORAGE SECTION.
01      BENTUK-LAPORAN.
        02      NPM-O        PIC X(8).
        02      NAMA-O      PIC X(20).
        02      NILAI-1O    PIC Z(2)9.
        02      BAYARAN-O   PIC ZZ,Z(3).99.

```

Misal salah satu record DATA-MHS adalah :

10289382
Maria
80
25000,80

Data tersebut akan tersimpan di MEMORI dan digunakan dalam PROCEDURE DIVISION dalam bentuk :

Nama Data item	Picture	Nilai data	Tersimpan di memori										Digunakan dalam Prosedur										
NPM	X(8)	10289382	1	0	2	8	9	3	8	2			1	0	2	8	9	3	8	2			
NAMA	X(20)	Maria	M	a	r	i	a						M	a	r	i	a						
NILAI-1	9(3)	80	0	8	0								0	8	0								
BAYARAN	9(6)V99	25000,80	0	2	5	0	0	0	.	8	0		0	2	5	0	0	0	.	8	0		

PICTURE karakter 9

- Hanya mengandung nilai NUMERIK.
- Ukuran dari panjang item ditunjukkan dengan banyaknya angka 9
- Mempunyai sifat RATA KANAN
- Bila nilai data < lokasi yg ditunjukkan pada picture = sisanya mulai dari kiri diisi dengan angka 0
- Bila nilai data > lokasi yg ditunjukkan pada picture = data akan dipotong mulai dari kiri.

PICTURE karakter V

- Menunjukkan letak anggapan dari titik desimal
- Tidak termasuk panjang dari field, tetapi informasi ini disimpan disuatu tempat dikomputer, sehingga perhitungan tetap benar
- Hanya boleh sebuah pada suatu field.

PICTURE karakter P

- Digunakan dgn gabungan PICTURE karakter V, utk menimbulkan 0
- Jarang digunakan

PICTURE karakter S

- Untuk memberi tanda negatif
- Digunakan bersamaan dengan PICTURE karakter 9

PICTURE karakter A

- Untuk menyimpan nilai data huruf (alphabet)
- Sifatnya rata kiri
- Bila nilai data < field maka lokasi field yg kosong diisi blank
- Bila nilai data > field maka nilai data terpotong mulai dari kanan

PICTURE karakter X

- Untuk menunjukkan posisi nilai alphanumeric (gabungan angka, huruf atau spesial karakter) & BERSIFAT RATA KIRI

PICTURE EDITING

- Berguna untuk perubahan bentuk data dari data yg tersimpan di storage.
- Dengan adanya perubahan, maka data yg dicetak akan tampak lebih mudah dibaca.
- Hanya Nilai numerik saja yang dapat diedit KECUALI Picture editing karakter B yg bisa meng-edit nilai huruf.
- Picture editing tdk boleh digunakan utk field penerima data langsung sbg input data
- Yang dapat digunakan sbg penerima langsung data input adalah : PICTURE karakter yaitu : 9, V, P, S, A atau X

PICTURE EDITING Z

Digunakan untuk menggantikan angka 0 dimuka dgn blank

PICTURE EDITING \$

- Digunakan untuk pencetakan \$ pada posisi paling kiri.
- Apabila lebih dari 1 karakter \$ maka digunakan utk pencetakan \$ pd posisi paling kiri dari angka selain 0.
- Bisa diganti dengan F, tetapi harus dideklarasikan dahulu pada CONFIGURATION SECTION.

PICTURE EDITING . DAN ,

- Karakter . digunakan utk menunjukkan letak posisi dari titik desimal pada pencetakan.
- Hanya boleh SATU . (titik) pada suatu picture editing
- Boleh lebih dari SATU , (koma) pada suatu picture editing
- Karakter , (koma) berfungsi memberikan bentuk koma pada tempat-tempat tertentu.

PICTURE EDITING -

- Untuk menampilkan tanda negatif bila data aslinya negatif
- Bila data aslinya tidak negatif maka digantikan blank

PICTURE EDITING +

- Untuk menampilkan tanda negatif atau tanda positif
- Bila data aslinya tidak bertanda, maka dianggap positif

PICTURE EDITING DB dan CR

- Untuk DEBIT dan KREDIT pada akuntansi
- DB dan CR hanya ditulis mulai dari ujung sebelah kanan dari PICTURE
- Hasilnya hanya tampak pada hasil edit bila nilai datanya negatif

PICTURE EDITING B

- Khusus untuk mengedit nilai data BUKAN NUMERIK
- Akan menyebabkan blank pada posisi dimana B berada.

PICTURE EDITING *

- Digunakan utk mengganti nilai 0 pd nilai data disebelah kanan dgn tanda * .
- Biasanya untuk maksud pencegahan terhadap nilai rupiah pd penulisan dokumen, sehingga tidak bisa disisipi angka sendiri.

PICTURE EDITING 0

- Digunakan untuk menyisipkan angka 0 pada posisi dimana karakter 0 itu berada.

PICTURE EDITING /

- sama dgn karakter 0, hanya saja karakter yg digunakan adlh / (slash)

BENTUK DARI DATA

Terdapat 2 data yaitu : DATA VARIABEL
DATA KONSTANTA

Data VARIABEL

Adalah data yang nilainya dapat berubah didalam program.

Contoh :

ACCEPT **jam-kerja**

Statemen ACCEPT digunakan utk memasukkan nilai data ke suatu lokasi storage yg diwakili oleh suatu nama-data

Isi dari jam-kerja tergantung dari masukkan, jadi selalu berubah-ubah.

Data KONSTANTA

Adalah data yang nilainya tidak bergantung dari input yang dibaca.

Contoh :

MULTIPLY jam-kerja BY **5000** GIVING gaji-o
5000 adalah data konstanta.

Ada 3 bentuk data konstanta yg dipakai pd PROCEDURE DIVISION dalam COBOL :

1. Literal Numerik

Untuk Operasi aritmatika. Aturan penggunaannya :

- Maksimum panjang 18 digit
- Boleh ada plus dan minus, dgn syarat letaknya pada posisi paling kiri tanpa ada spasi dgn angka pertama.
- Boleh ada titik desimal dimana saja kecuali yg terkanan

Contoh :

BENAR	SALAH
555	73.
-15.16	- 16
18260	25,5
-.80	\$5

2. Literal Bukan Numerik

Bukan untuk operasi aritmatika.

Aturan penggunaannya :

- Maksimum panjang 120 karakter
- Boleh terdiri dari kumpulan karakter, kecuali karakter petik.
- Harus dibatasi dengan tanda petik.

Contoh :

BENAR	SALAH
'Nilai IPS'	Nilai IPS
'23-12-94'	'neraca 'IP"

3. Figurative Constant

Termasuk dalam reserved words

Contoh : ZERO,ZEROS,ZEROES (artinya nol)
SPACE,SPACES(artinya blank)
QUOTE,QUOTES(artinya tanda petik)
ALL literal

Contoh penggunaan :
MOVE ZERO TO jumlah

MOVE SPACES TO jarak
DISPLAY QUOTE 'PT AKU ' QUOTE
MOVE ALL 'A' TO HEADER