

DIVISION PADA COBOL

OBJEKTIF :

1. Mahasiswa Mengetahui Pengetahuan Dasar Tentang Division pada COBOL.
 2. Mahasiswa Mampu Mengetahui Fungsi Setiap Division pada COBOL.
 3. Mahasiswa Mampu Memahami Struktur Division pada COBOL.
-

Program COBOL bersifat hierarkis dalam sebuah struktur. Setiap elemen hierarki terdiri dari satu atau lebih elemen bawahan. Hierarki program COBOL terdiri dari program, division, section, paragraph, sentences, statements dan character.

Division merupakan elemen structural utama dalam COBOL. Terdapat empat division dalam pemrograman COBOL antara lain: Identification division, Environment division, Data division, dan Procedure division.

2.1 IDENTIFICATION DIVISION

Identification Division merupakan division yang pertama dalam pemrograman COBOL. Division ini harus ada dalam program. Identification division bertujuan untuk memberikan identifikasi atau pengenalan, baik bagi computer maupun pemakai program tersebut. Perlu diketahui bahwa division ini tidak memiliki satu section-pun, tetapi langsung dengan paragraph.

Pada Identification Division, terdiri dari beberapa paragraph antara lain : Identification division, Program-ID, Author, Installation, Date-written, Date-Compiled, Security, dan Remarks.

IDENTIFICATION DIVISION

Paragraph ini merupakan kata kunci jadi harus dituliskan dan tidak boleh salah dalam penulisannya, karena program tidak akan menjalankan program tersebut jika masih terdapat sebuah kesalahan.

PROGRAM – ID

Paragraph ini harus ada dan langsung dituliskan sesudah IDENTIFICATION DIVISION. Paragraph ini menunjukkan nama program. Untuk program-name, aturan penulisannya mengikuti aturan yang berlaku pada programmer's word. Program-name inilah yang merupakan identifikasi program yang dapat dikenali oleh komputer. Pada PROGRAM-ID, untuk menentukan nama program dapat terdiri dari 1 hingga 30 karakter.

AUTHOR

Author digunakan untuk menuliskan nama pemrogram yang telah membuat program tersebut. Paragraph ini hanya dianggap sebagai komentar oleh kompilator, karena itu paragraph inipun sifatnya optional.

INSTALLATION

Installation dianggap sebagai komentar saja oleh kompilator dan digunakan untuk memberi keterangan untuk siapa program ini dibuat, paragraph ini juga bersifat optional.

DATE - WRITTEN

Date-written merupakan paragraph dimana seorang programmer menuliskan tanggal pembuatan program. Paragraph ini bersifat optional.

DATE - COMPILED

Paragraph ini dapat digunakan untuk menuliskan waktu kompilasi program. Karena paragraph ini bersifat optional, maka hanya akan dianggap sebagai komentar oleh kompilator.

SECURITY

Security digunakan untuk menunjukkan tingkat pengamanan komputer. Paragraph ini bersifat optional. Paragraph ini hanya dianggap sebagai komentar oleh kompilator.

REMARKS

Remarks digunakan untuk memberikan keterangan yang ada hubungannya dengan program. Jika penulisan keterangan tersebut melebihi satu baris, maka sambungannya harus dimulai dalam margin B di baris berikutnya. Ada juga kompilator yang tidak menyediakan tempat bagi keterangan semacam ini. Penulisannya cukup dengan memberikan tanda * (asterisk) pada kolom 7.

Bentuk umum dari Identification Division :

IDENTIFICATION	DIVISION_
PROGRAM-ID_	Nama-program_
AUTHOR_	Nama-programmer_
INSTALLATION_	Nama-instalasi_
DATE-WRITTEN_	Tanggal program dibuat_
DATE-COMPILED_	Tanggal program dikompilasi_
SECURITY_	Sifat program_
REMARKS_	Keterangan_

Contoh penggunaann Identification Division pada COBOL:

IDENTIFICATION	DIVISION_
PROGRAM-ID_	NILAI_
AUTHOR_	Rini_
INSTALLATION_	Universitas Gunadarma_
DATE-WRITTEN_	15-07-2020_
DATE-COMPILED_	16-07-2020_
SECURITY_	Private_
REMARKS_	Penginputan nilai UTS dan UAS_

Pada contoh kasus di atas, nama program NILAI dituliskan pada bagian PROGRAM-ID. Dengan nama pemrogram ditulis pada paragraph AUTHOR yaitu Rini. Program yang ditujukan kepada Universitas Gunadarma akan ditulis dibagian INSTALLATION. DATE-WRITTEN akan ditulis sesuai tanggal program tersebut dibuat, yaitu 15 Juli 2020 atau dapat ditulis 15-07-2020. Dan DATE-COMPILED dapat ditulis 16-07-2020 sebagai tanggal kompilasi program. Pada bagian SECURITY, dapat ditulis sesuai sifat program tersebut yaitu private. Dan dapat menuliskan keterangan yang berkaitan dengan program bahwa program dibuat untuk penginputan nilai UTS dan UAS.

2.2 ENVIRONMENT DIVISION

Environment Division merupakan division kedua dalam pemrograman COBOL. Environment Division menyediakan informasi mengenai peralatan yang dipergunakan dalam program. Dalam division ini akan menunjukkan atau menerangkan bagian mana dari sistem komputer yang dipakai untuk mengerjakan program yang bersangkutan, file dan device yang digunakan, serta karakteristik dari file yang digunakan tersebut, dan keterangan lain yang berlaku dalam program tersebut.

Environment Division digunakan untuk menentukan file input dan output ke program. Ini terdiri dari dua bagian, antara lain :

CONFIGURATION SECTION

Section ini terdiri atas 3 paragraph, yang menjelaskan mengenai komputer yang digunakan untuk mengkompilasi dan mengeksekusi program yang bersangkutan. Disamping itu juga digunakan untuk menunjukkan atau memberi nama khusus untuk function-name.

Bentuk umum dari Configuration Section :

```
CONFIGURATION SECTION.  
SOURCE-COMPUTER_      computer-name  
OBJECT-COMPUTER_      computer-name  
SPECIAL-NAMES_       special-names-entry
```

Dalam beberapa mesin komputer, CONFIGURATION SECTION ini bersifat optional. Akan tetapi di beberapa mesin komputer lainnya mengharuskan untuk menuliskannya.

- **SOURCE-COMPUTER**

Source-computer digunakan untuk menunjukkan nama komputer yang dipakai untuk membuat dan mengkompilasi program yang bersangkutan. Computer-name digunakan untuk nama setiap mesin komputer. Entry dalam paragraph ini akan dianggap sebagai komentar. Namun jika menyertakan WITH DEBUGGING MODE, maka akan diproses untuk menelusuri kesalahan dan tidak dianggap sebagai komentar.

Contoh penulisan SOURCE-COMPUTER untuk UNIVAC (Sebagai komputer yang digunakan untuk membuat dan mengkompilasi program) :

```
SOURCE-COMPUTER_      UNIVAC_
```

- **OBJECT-COMPUTER**

Object-name digunakan untuk menunjukkan nama komputer yang digunakan untuk menjalankan program yang telah dikompilasi. Paragraph ini bersifat sebagai dokumentasi atau hanya dianggap sebagai komentar.

Contoh penulisan OBJECT-COMPUTER untuk EDVAC (Sebagai komputer yang digunakan untuk mengeksekusi program yang telah dikompilasi) :

```
OBJECT-COMPUTER_      EDVAC_
```

INPUT-OUTPUT SECTION

INPUT-OUTPUT SECTION digunakan untuk menyebutkan nama file yang akan digunakan dalam program dan menghubungkan dengan alat pembaca dan perekam data (Input/Output Device) sebagai peralatan penyimpanan sekunder. Selain itu, section ini digunakan sebagai proses input dan output file tersebut.

Bentuk umum Input-Output Section :

```
INPUT-OUTPUT SECTION_
FILE-CONTROL_
    file-control-entry_
I-O-CONTROL_
    input-output-control-entry_
```

- **FILE-CONTROL Paragraph**

FILE-CONTROL paragraph digunakan untuk memberikan nama suatu file yang digunakan dalam unit program tersebut (internal file name) dan dihubungkan dengan device yang dipakai.

Penulisan FILE-CONTROL Paragraph :

```
FILE-CONTROL_
    SELECT [OPTIONAL] internal-file-name
    ASSIGN TO {DISK/PRINTER}
```

- **SELECT Clause**

Clause ini digunakan untuk menyebutkan internal-file-name yang akan dipakai dalam program tersebut. Setiap file-name tersebut harus mempunyai satu file description atau sort file description (FD atau SD) dalam FILE SECTION di DATA DIVISION.

- **ASSIGN Clause**

File-name yang sudah disebutkan sebelumnya (dalam SELECT Clause) harus dihubungkan dengan keadaan fisik yang diterangkan dalam system-name. System-name menjelaskan class dari device yang digunakan, organisasi data dan external-name dari file tersebut.

Bentuk umum penulisan Environment Division :

```
ENVIRONMENT DIVISION_
CONFIGURATION SECTION_
SOURCE-COMPUTER_           Computer-name
OBJECT-COMPUTER_           Computer-name
INPUT-OUTPUT SECTION_
FILE-CONTROL
    SELECT [internal-file-name]
    ASSIGN TO {DISK/PRINTER}
```

Contoh penulisan Environment Division pada COBOL :

```
ENVIRONMENT DIVISION_  
CONFIGURATION SECTION_  
SOURCE-COMPUTER_ LENOVO330_  
OBJECT-COMPUTER_ LENOVO320_  
INPUT-OUTPUT SECTION_  
FILE-CONTROL_  
    SELECT NilaiUjian ASSIGN TO DISK_
```

Pada contoh di atas, terdapat dua section pada ENVIRONMENT DIVISION. Pada CONFIGURATION SECTION, menjelaskan tentang informasi komputer LENOVO330 yang digunakan dalam pembuatan dan kompilasi program yang ditulis dalam SOURCE-COMPUTER, dan menjelaskan informasi komputer LENOVO320 yang digunakan untuk mengeksekusi program tersebut yang ditulis dalam OBJECT-COMPUTER.

Kemudian pada bagian INPUT-OUTPUT SECTION, menyebutkan nama file NilaiUjian yang digunakan dalam unit program dan ditulis dalam SELECT Clause pada FILE-CONTROL Paragraph. File tersebut akan dihubungkan ke device pemakai menggunakan ASSIGN Clause dan menyertakan class dari device yang digunakan yaitu DISK.

2.3 DATA DIVISION

Data division adalah division ketiga dalam pemrograman COBOL. Division ini digunakan memberikan informasi mengenai struktur file, record format beserta dengan item yang ada dalam perhitungan, atau parameter yang akan dimasukkan ke subprogram, dan informasi lain yang dianggap perlu.

Pada umumnya, data division ini terbagi atas 4 section, tetapi ada juga beberapa kompilator yang menyediakan beberapa section lainnya. Disini akan dibahas lebih dahulu 2 section pokok dalam data division, yaitu File Section dan Working-Storage Section.

FILE SECTION

File Section digunakan untuk menerangkan karakteristik data file beserta recordnya. Perlu diingat bahwa file yang sudah disebutkan pada FILE-CONTROL dalam ENVIRONMENT DIVISION harus selalu dijelaskan karakteristiknya dalam FILE SECTION.

FILE SECTION terdiri atas File Description Entry dan Record Description Entry.

- **File Description**

File Description berisi keterangan mengenai karakteristik suatu file.

- **Record Description**

Record Description digunakan untuk menjelaskan struktur suatu record dan field yang membentuknya.

WORKING-STORAGE SECTION

Section ini digunakan untuk menjelaskan atau mendefinisikan internal data, baik berupa konstanta maupun work-area yang disediakan untuk keperluan khusus, misalnya untuk table dan lain-lain.

Seperti yang dituliskan pada bentuk Data Division, Working-Storage Section terdiri atas Data-item Description Entry dan Record Description Entry.

Data-item Description Entry selalu dimulai dengan level-number 77, dan mereka dianggap sebagai independent-data-item. Untuk beberapa kompilator, semua 77 level-entry harus dituliskan dibagian pertama dari section ini.

Sedangkan Record Description Entry mempunyai bentuk yang sama dengan yang ditulis dalam File Section, yakni selalu dimulai dengan level-number 01.

Beberapa fungsi WORKING-STORAGE SECTION, yaitu :

- 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.

Bentuk umum Data Division :

```
DATA DIVISION_  
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  
WORKING-STORAGE SECTION_  
77 level data description entry  
record description
```

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.
- **Independent Item** merupakan cara pendefinisian lokasi memori yang bukan dalam bentuk record terpisah struktur record lainnya. Misalnya : sebagai counter, index matriks, acuan dalam menentukan suatu kondisi. Indikator yang digunakan adalah level number 77.
- **Record description entry** merupakan pendefinisian data yang dikelompokkan dalam bentuk record.
- **Data description entry** merupakan suatu alat bantu dalam mengarahkan logika seorang pemrogram COBOL untuk mendapatkan suatu penyelesaian terhadap masalah yang dihadapi.

- **Ommited** digunakan untuk card-file atau print-file yang tidak mempunyai file.
- **Standard** digunakan untuk disk-file yang mempunyai label.

Deskripsi data terbentuk dari beberapa clause :

1. Level Number

2. Nama Data

3. FILLER

4. PICTURE

5. VALUE OF FILE-ID

(Berisi informasi mengenai file yang mempunyai label di disk. Label di disk ini berupa suatu nama-file yang berisi data dan direkam di dalam disk tersebut.)

6. BLANK WHEN ZERO merupakan sederetan reserved word yang dapat digunakan untuk menggantikan nilai nol pada suatu medan data dengan blank (kosong)

7. JUSTIFIED digunakan untuk menentukan apakah besaran yang dikandung oleh suatu nama data akan rapat kiri atau rapat kanan

8. OCCURS Untuk mendefinisikan matriks atau tabel.

9. Label Record atau Label Records menunjukkan apakah file yang dipergunakan mempunyai label atau tidak.

Contoh penulisan Data Division :

```
DATA DIVISION_.  
FILE SECTION_.  
FD NilaiUjian_.  
01 NILAI PIC 999V99_.  
  
WORKING-STORAGE SECTION_.  
01 NILAI-UTS PIC 9(8)_.  
01 NILAI-UAS PIC 9(8)_.  
01 RATA-RATA PIC 9(8)_.  

```

Berdasarkan contoh di atas, terdapat nama file NilaiUjian. Record pada file NilaiUjian dapat menampung data ke dalam datanames NILAI dengan menggunakan Picture Clause 999V999.

Pada Working-Storage Section, terdapat datanames yang akan menampung data numerik yaitu bernama NILAI-UTS, NILAI-UAS, dan RATA-RATA.

2.4 PROCEDURE DIVISION

Procedure Division merupakan division yang memperinci setiap langkah instruksi. Mulai dari instruksi untuk menyiapkan peralatan atau device yang akan digunakan baik input maupun output, instruksi untuk menyiapkan input data yang akan diolah, instruksi untuk melakukan hal-hal seperti pengolahan aritmatika dan logika, manipulasi data, atau memasukkan (store) data ke output device. Data-name yang muncul dalam division ini harus sudah dinyatakan dalam division sebelumnya.

Section dan paragraph merupakan user defined section atau paragraph. Section dan paragraph tersebut selalu mempunyai nama yang sering disebut dengan procedure-names. procedure name mengikuti aturan yang sudah ditetapkan untuk programmer supplied names, tetapi khusus untuk pemakaian dalam division ini diberikan sedikit kelonggaran dengan memperbolehkan penggunaan numeric character untuk nama tersebut. Jadi tidak harus dimulai dengan alphabetic character.

Untuk section, setelah nama tersebut (procedure-name) diikuti dengan minimal satu space atau blank, kata SECTION dan titik. Tidak boleh disambung dengan kata-kata lainnya. Kecuali untuk DECLARATIVES dan untuk segmentations. Sedangkan untuk paragraph, setelah nama diikuti dengan sebuah titik dan minimal sebuah blank. Perlu diingat bahwa untuk suatu unit program tidak diperbolehkan untuk menggunakan procedure name yang sama.

Untuk penulisan section dan paragraph-name pada PROCEDURE DIVISION akan selalu dituliskan dalam margin A. PROCEDURE DIVISION bersifat hierarki dalam struktur yang terbagi dalam beberapa bagian, antara lain :

- **Section**

Suatu section dapat terdiri atas suatu paragraph atau lebih yang pembagiannya dapat disesuaikan oleh kehendak programmer. Section ini dibatasi oleh section lainnya atau suatu EXIT Statement.

- **Paragraph**

Suatu paragraph merupakan suatu kesatuan logika yang terdiri atas satu atau beberapa sentence.

- **Sentence**

Yang dimaksud dengan sentence adalah suatu kesatuan yang terdiri atas satu atau lebih statement yang diakhiri dengan sebuah titik dan minimal satu space atau blank. Tanda koma, titik koma (semicolon) atau kata THEN dapat digunakan sebagai batas pemisah antara statement-statement yang membentuk sentence tersebut. Setiap sentence harus dituliskan dalam margin B.

- **Statement**

Suatu statement terdiri atas satu COBOL verb diikuti oleh operands (data-names, literals, dan lain-lain) dan COBOL words lainnya digunakan untuk melengkapi statement tersebut. Kombinasi dari data-names, literals, COBOL words dan atau operator (aritmatika dan logika) yang mengandung arti tertentu disebut juga dengan expression.

Jenis-jenis statement meliputi :

INPUT/OUTPUT statement

1. **Accept Statement** digunakan untuk memasukkan data, dari layar sewaktu program tersebut dijalankan. Pada umumnya data tersebut dalam jumlah relatif kecil dan tidak didefinisikan sebagai file.

Bentuk umum :

```
ACCEPT nama-data_.
```

Contoh :

```
ACCEPT DATA-NILAI_.  
ACCEPT TANGGAL FROM DATE_.
```


2. **Display Statement** digunakan untuk melakukan transfer data terhadap suatu hasil pemrosesan dari memori komputer ke dalam media keluaran. Dimana hasil tersebut tidak didefinisikan sebagai file.

Bentuk umum :

```
DISPLAY nama-record atau string
```

Contoh :

```
DISPLAY 'Hello world!'
```

3. **Open Statement** digunakan untuk mempersiapkan/membuka file yang digunakan dalam program.

Bentuk Umum :

```
OPEN INPUT nama-file OUTPUT nama-file
```

Contoh :

```
OPEN INPUT FILE-MHS OUTPUT REPORT-MHS_.
```

4. **Read Statement** digunakan untuk membaca data/record yang ada dalam suatu file.

Bentuk Umum :

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

Contoh :

```
READ FILE-MHS AT END MOVE 0 TO AKHIR_  
READ FILE-MHS AT END GO TO HABIS_.
```

5. **Write Statement** digunakan 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]
```

Contoh :

```
WRITE REPORT-MHS FROM JUDUL-1_.
```

6. **Close Statement** digunakan 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.
```

SCREEN SECTION

Section ini berguna untuk mengatur bentuk format dari layar input maupun output. Pada section ini juga anda dapat menuliskan level number, tetapi hanya level number 01 sampai 49 saja yang dapat digunakan, sedangkan level number 77 hanya dapat ditulis di WORKING-STORAGE SECTION. SCREEN SECTION berhubungan dengan statement DISPLAY dan ACCEPT pada PROCEDURE DIVISION. Berikut ini beberapa perintah display formatting pada SCREEN SECTION :

- LINE clause untuk menempatkan kursor pada baris yang ditentukan.
- COLUMN clause untuk menempatkan kursor pada kolom yang ditentukan.

Bentuk Umum Procedure Division :

```
PROCEDURE DIVISION.  
[Nama paragraph].  
[Nama section].  
    Statement.  
STOP RUN.
```

Contoh penggunaan Procedure Division pada COBOL :

```
PROCEDURE DIVISION.  
MULAI.  
    DISPLAY "NILAI UTS : ".  
    ACCEPT NILAI-UTS.  
    DISPLAY "NILAI UAS : ".  
    ACCEPT NILAI-UAS.  
    COMPUTE RATA-RATA = (NILAI-UTS + NILAI-UAS)/2.  
    DISPLAY "RATA-RATA NILAI : ", RATA-RATA.  
SELESAI.  
    STOP RUN.
```

Pada procedure division di atas, terdapat penamaan section mulai dan selesai. Dalam procedure tersebut, akan menampilkan text input untuk memasukkan data nilai. Kemudian akan dilakukan perhitungan terhadap kedua datanames yaitu datanames NILAI-UTS dan NILAI-UAS yang hasilnya akan ditampung ke dalam datanames RATA-RATA.

