PERNYATAAN PERULANGAN COBOL

OBJEKTIF :

1. Mahasiswa Mengetahui Pengetahuan Pernyataan Perulangan Bahasa COBOL.
2. Mahasiswa Mampu Membuat dan Menjalankan Pernyataan Perulangan Bahasa COBOL.
3. Mahasiswa Mampu Memahami Struktur Pernyataan Perulangan Bahasa COBOL.

7.1 PERNYATAAN PERFORM

Program COBOL akan memproses statement-statement secara sequence (berurutan). Dimulai dari pernyataan pertama pada PROCEDURE DIVISION dan sampai dengan STOP RUN. Dengan menggunakan PERFORM, kita dapat mengubah aliran proses sekuensial didalam program COBOL. Terdapat dua tujuan utama penggunaan PERFORM verb, yaitu :

- Untuk mentransfer control ke blok kode yang ditentukan
- Untuk mengeksekusi blok code secara iterative

Setiap jenis PERFORM terdapat 2 tipe eksekusi yang memungkinkan, yaitu :

- Eksekusi out-of-line blok kode

  Loop body adalah paragraph atau bagian yang terpisah. Pada Bahasa program lain setara dengan memiliki procedure, function, methods dalam loop body dari konstruksi while atau for

- Eksekusi in-line blok kode

  Eksekusi Inline akan sangat familiar bagi programmers yang sudah pernah menggunakan iterasi (while, do/repeat,for) dari kebanyakan bahasa program.

Ketika sebuah loop diperlukan, tetapi hanya beberapa statement yang terlibat maka sebaiknya menggunakan inline PERFORM. Sedangkan, ketika sebuah loop diperlukan dan loop body menjalankan beberapa tugas dan fungsi tertentu maka out-of-line perform sebaiknya digunakan

Pernyataan perform terbagi menjadi 4, yaitu :

- Pernyataan PERFORM THRU
- Pernyataan PERFORM UNTIL
- Pernyataan PERFORM TIMES
- Pernyataan PERFORM VARYING

PERFORM THRU

Digunakan untuk menjalankan serangkaian paragraf dengan memberikan nama paragraf diawal dan diakhir secara berurutan. Setelah menjalankan paragraf terakhir, akan mengembalikan proses. Dan statement didalam PERFORM akan di *execute* sampai tercapainya END-PERFORM. Perform thru biasanya digunakan untuk mengeksekusi sekumpulan besar pernyataan atau blok program yang dipisahkan dengan paragraf.

Contoh Inline Perform:

```
       PERFORM
          DISPLAY 'HELLO WORLD'
       END-PERFORM.
```

Inline perform dalam COBOL akan bertindak sebagai akhir dari perform paragraf. Tidak adanya inline perform dalam perform thru sehingga inline perform tidak melakukan sebuah routines. Maka dari itu dalam perform thru hanya menggunakan outline perform.

Bentuk umum Out Line Perform Thru :

```
PERFORM PARAGRAF1 THRU PARAGRAF2
```

PERFORM UNTIL

Format perform ini mengimplementasikan antara pre-test dan post-test iterasi dalam Cobol. Ini sama dengan penggunaan while dan do..while pada java, atau while dan repeat..until pada pascal.

Dalam perform ini, sebuah paragraph akan di eksekusi sampai kondisi menjadi true. **WITH TEST BEFORE** yang merupakan kondisi default yang berindikasi bahwa kondisi akan diperiksa sebelum pernyataan didalam paragraph dieksekusi. Jika menggunakan **WITH TEST BEFORE**, maka PERFORM akan berperilaku seperti while loop dan kondisi diuji sebelum masuk kedalam *loop body*. Jika menggunakan **WITH TEST AFTER**, perform akan berperilaku seperti do..while loop dan kondisi akan diuji setelah masuk kedalam loop body.

Bentuk umum inline perform until :

```
PERFORM WITH TEST {BEFORE/AFTER}
    UNTIL Condition
    Statements
END-PERFORM.
```

Bentuk umum outline perform until :

```
PERFORM {paragraph/section}[{thru}{paragraph/section}
WITH TEST {BEFORE/AFTER}
   UNTIL Condition
```

Gambaran sederhana penggunaan pernyataan perform until yaitu :

```
PERFORM PARA-NAME UNTIL COUNT=5
PERFORM PARA-NAME WITH TEST BEFORE UNTIL COUNT=5
PERFORM PARA-NAME WITH TEST AFTER UNTIL COUNT=5
```

PERFORM TIMES

Dalam Perform ini, sebuah paragraph akan di eksekusi sesuai yang ditentukan.

Bentuk umum inline perform times :

```
PERFORM {N} TIMES
  Statements
END-PERFORM
```

Bentuk umum outline perform times :

```
PERFORM {paragraph/section} [THRU] {N} TIMES
  Statements
{paragraph/section}
```

Maksud dari **N** pada bentuk umum diatas adalah untuk menunjukkan berapa kali prosedur tersebut dilaksanakan. **N** harus berupa numerik, Jika nilainya sama dengan 0 atau bernilai negative, maka prosedur tersebut tidak dilaksanakan.

Gambaran sederhananya :

```
PERFORM nama-paragraf 5 TIMES.
```

PERFORM VARYING

Digunakan untuk mengimplementasikan perhitungan iterasi. Pernyataan ini mirip dengan pernyataan FOR pada Bahasa program lain seperti Pascal, C dan Java. Pada perform varying sebuh paragraph akan dieksekusi sampai kondisi pada statement Until terpenuhi (True).

Bentuk umum inline perform varying :

```
PERFORM WITH TEST {BEFORE/AFTER}VARYING {index/counter}
FROM {start value}BY {step value}
  AFTER {start value} BY {step value}UNTIL Condition]
     Statements
END-PERFORM.
```

Bentuk umum outline perform varying :

```
PERFORM {paragraph/section}[{THRU}{paragraph/section}]
WITH TEST {BEFORE/AFTER} VARYING {index/counter}
FROM {start value}BY {step value}
  UNTIL Condition
[AFTER {start value} BY {step value}UNTIL Condition]
```

Statement ini digunakan untuk mengerjakan suatu prosedur di luar urutan yang ada. Setelah melaksanakan prosedur tersebut, maka statement sesudah PERFORM tersebut yang dilaksanakan. Gambaran sederhana :

```
PERFORM nama-paragraf VARYING A FROM 1 BY 1 UNTIL A=5
```

7.2 PERNYATAAN GO TO

Pernyataan GO TO digunakan untuk mengubah alur eksekusi dalam suatu program dan memindahkan proses program dari suatu bagian PROCEDURE DIVISION ke bagian lainnya. Dalam pernyataan GO TO, Program yang sedang diproses di transfer secara *forward direction*. Dan pernyataan ini juga dapat digunakan untuk keluar dari suatu paragraf.

UNCONDITIONAL GO TO

Pernyataan ini memindahkan proses ke pernyataan paragraph pertama atau bagian yang diidentifikasi dengan procedure-name , kecuali pernyataan GO TO dimodifikasi dengan pernyataan ALTER. Secara sederhananya, proses pogram berpindah secara langsung ke paragraf yang ditunjuk tanpa adanya ketentuan kondisi tertentu.

Bentuk umum :

```
        GO TO nama-paragraf.
```

CONDITIONAL GO TO

Pernyataan ini akan memindahkan proses ke salah satu serangkaian procedure, tergantung pada nilai item oleh identifier. Jadi, dengan statement ini akan membawa proses berpindah (GOTO) ke suatu nama paragraf yang ditunjuk, tergantung (DEPENDING) dari nilai data name pada statement .

Bentuk umum :

```
        GO TO nama-para1,nama-para2,... Nama-paragraph-n
            DEPENDING ON x
```

Jika ‘x’ sama dengan 1, proses akan dipindahkan ke paragraf pertama dan jia ‘x’ sama dengan 2 maka proses akan berpindah ke paragraf kedua. Begitu seterusnya.