

## BAB II

# PERCABANGAN, PERULANGAN, REKURSIF, PROSEDUR DAN FUNGSI

### TUJUAN PRAKTIKUM

1. Mengerti operasi percabangan dan statemen yang digunakan.
2. Mengerti operasi perulangan dan rekursif dan statemen yang digunakan.
3. Mengetahui penggunaan Procedure dan Function.

### TEORI PENUNJANG

#### Percabangan

Ada 2 macam statemen yang digunakan yaitu: *If* dan *case*

##### *Statemen If*

Struktur dari statemen If dapat berupa If-then atau If-Then-Else sintak :

```
If ungkapan Then statemen
```

Ungkapan adalah kondisi yang akan diseleksi oleh statemen If. Bila kondisi yang diseleksi terpenuhi, maka statemen yang mengikuti then akan diproses, jika kondisi tidak terpenuhi maka yang akan diproses adalah statemen berikutnya.

```
If kondisi Then statemen1 else statemen2
```

Statemen1 atau dapat berupa blok statemen akan diproses bilamana kondisi yang diseleksi benar (terpenuhi) sedang statemen2 atau dapat berupa blok statemen akan diproses bilamana kondisi yang diseleksi tidak terpenuhi.

##### *Statemen Case*

Statemen Case dapat berbentuk struktur *Case-of* atau *Case-of...Else*.

Bentuk dari struktur case-of :

```
Case ungkapan Of
  Daftar case-label1: statemen1;
  Daftar case-label2: statemen2;
  Daftar case-label3: statemen3;
  . . .
  Daftar case-labeln : statemenn;
Else
  Daftar case-labelx : statemenx;
End;
```

Bila tidak ada kondisi yang terpenuhi maka statemen yang akan diproses di dalam lingkungan *Case-of* adalah statemen yang ada di *Else*.

## **Perulangan**

Bahasa Pascal memberikan 3 jenis struktur program untuk perulangan dan sebuah statemen goto.

*Statemen For ... to / downto ... do ...*

Pernyataan ini akan melakukan perulangan dari suatu nilai awal sampai nilai akhir yang telah ditentukan. Statemen downto merupakan perulangan negatif dari nilai paling tinggi sampai nilai paling rendah.

Sintak :

Perulangan positif :

```
For variabel-kontrol := nilai-awal To nilai-akhir Do statemen
```

Perulangan negatif :

```
For variabel-kontrol:=nilai-awal Down-to nilai-akhir Do statemen
```

Variabel-kontrol, nilai-awal, nilai-akhir mempunyai tipe yang sama yaitu bertipe integer.

### *Statemen While ... do ...*

Pernyataan while akan melakukan pengujian kondisi bila kondisi tersebut terpenuhi, maka proses dilakukan, bila tidak maka proses dilanjutkan ke pernyataan berikutnya.

Sintak :        While ungkapan-logika Do statemen

### *Statemen Repeat ... Until ...*

Pernyataan repeat akan memproses pernyataannya dulu sampai klausa until tidak terpenuhi. Proses dilakukan selama kondisi salah.

Sintak :        Repeat statemen Until ungkapan

## **Rekursif**

Prosedur atau fungsi dipanggil lewat namanya baik itu di luar sub program maupun di dalam sub program. Pemanggilan prosedur atau fungsi di dalam prosedur atau fungsi itu sendiri disebut dengan rekursif. Struktur rekursif baik dipakai untuk beberapa algoritma yang sulit. Perbandingan rekursif ini dengan perulangan(iterasi) adalah pada rekursif setiap pemanggilan akan menggunakan memori.

## **Procedure dan fungsi**

Dalam bahasa pemrograman yang terstruktur, program yang besar dapat dipecah menjadi bagian-bagian yang lebih kecil yang dinamakan dengan subprogram. Biasanya subprogram berisi suatu fungsi atau prosedur tugas tertentu.

## **Procedure**

Suatu program terpisah dalam blok sendiri yang berfungsi sebagai subprogram (program bagian). Prosedur diawali dengan kata cadangan Procedure di dalam bagian deklarasi prosedur. Prosedur dipanggil dan digunakan di dalam blok program yang lainnya dengan menyebutkan judul dan prosedurnya.

Sintak :

```
Program judul-program;  
    Procedure judul-prosedur;  
        Begin  
            . . .  
        End;  
Begin  
    . . .  
End.
```

### **Function**

Blok fungsi hampir sama dengan blok prosedur, hanya fungsi harus dideklarasikan dengan tipenya. Tipe deklarasi ini menunjukkan tipe hasil dari fungsi. Tipe tersebut ditulis pada akhir deklarasi fungsi yang didahului dengan titik koma, sebagai berikut.

```
Function identifier(dafta-parameter) : type;
```

Contoh :

```
Function faktorial(var Fak, hasil : integer) : integer;
```