

# 1 R E K U R S I

**REKURSI** berarti suatu proses yang bisa memanggil dirinya sendiri.

Dalam rekursi sebenarnya terkandung pengertian prosedur atau fungsi.

**Perbedaannya:** - dalam rekursi bisa memanggil dirinya sendiri

- prosedur atau fungsi harus dipanggil lewat pemanggil prosedur atau fungsi
- dalam pemakaian sehari-hari, rekursi merupakan teknik pemrograman yang berdayaguna untuk digunakan pada pekerjaan pemrograman dengan mengekspresikannya ke dalam suku-suku dari program lain dengan menambahkan langkah-langkah sejenis
- Dalam beberapa hal, rekursi kurang efisien dibanding proses iterasi

**Contoh:** 1) Menghitung nilai faktorial

2) Bilangan Fibonacci

3) Menyusun permutasi

## PROSES REKURSIF

1) {\*\*\*\*\*  
\* Fungsi untuk menghitung bilangan faktorial \*  
\* dengan cara rekursif \*  
\*\*\*\*\*}

**Function FAKT(N : integer) : integer;**

**Begin**

**If N = 0 then**

**FAKT := 1**

**Else**

**FAKT := N \* FAKT(N - 1)**

**End;**

**Catatan:** Notasi FAKT(N-1) yang digunakan untuk memanggil nilai fungsi sebelumnya, sering disebut dengan pemanggil rekursi atau *recursion call*.

2) {\*\*\*\*\*  
 \* **Prosedur untuk menghitung bilangan Fibonacci** \*  
 \* **dengan cara rekursif** \*  
 \*\*\*\*\*}

**Function FIBO(N : integer) : integer;**

**Begin**

**If (N = 1) or (N = 2) then**

**FIBO := 1**

**Else**

**FIBO(N) := FIBO(N - 1) + FIBO(N - 2)**

**End;**

3) {\*\*\*\*\*  
 \* **Fungsi untuk menghitung bilangan Fibonacci** \*  
 \* **dengan cara iterasi** \*  
 \*\*\*\*\*}

**Function FIBO\_ITERASI(N : integer) : integer;**

**Var F, Akhir, Bantu, I : integer**

**Begin**

**I := 1; F := 1; Akhir := 0;**

**If N = 0 then F := 0;**

**While I <> N do**

**Begin**

**Bantu := F; I := I + 1;**

**F := F + Akhir;**

**Akhir := Bantu**

**End;**

**FIBO\_ITERASI := F**

**End;**