PERTEMUAN 6

STRUKTUR REKURSIF

STRUKTUR REKURSIF

Rekursif adalah suatu proses yang bisa memanggil dirinya sendiri.

Contoh konsep penggunaan Rekursif

Masalah : Memotong Roti tawar tipis-tipis sampai habis

Algoritma :

1. Jika roti sudah habis atau potongannya sudah paling tipis maka pemotongan roti selesai.
2. Jika roti masih bisa dipotong, potong tipis dari tepi roti tersebut, lalu lakukan prosedur 1 dan 2 untuk sisa potongannya.

Contoh Fungsi Rekursif

a. Fungsi pangkat
b. Faktorial
c. Fibonancy
d. Menara Hanoi

Fungsi Pangkat

Menghitung 10 pangkat n dengan menggunakan konsep rekursif.

Secara Notasi pemrograman dapat ditulis :

$10^{0} = 1$ .....(1)

$10^{n} = 10 * 10^{n-1}$ .....(2)

Contoh :

$10^{3} = 10 * 10^{2}$

$10^{2} = 10 * 10^{1}$

$10^{1} = 10 * 10^{0}$

$10^{0} = 1$

Faktorial

0! = 1

N! = N x (N-1)! Untuk N > 0

Scr notasi pemrograman dapat ditulis sebagai :

FAKT (0) = 1 ..... (1)

FAKT(N) = N * FAKT (N-1)..... (2)

Contoh :

FAKT(5) = 5 * FAKT(4)

FAKT(4) = 4 * FAKT(3)

FAKT(3) = 3 * FAKT(2)

FAKT(2) = 2 * FAKT(1)

FAKT(1) = 1 * FAKT(0)

Nilai Awal

Misal :

hitung 5!, maka dapat dilakukan secara rekursif dgn cara :

5! = 5 * 4!

Scr rekursif nilai dr 4! Dpt dihitung kembali dgn 4 * 3!,

shg 5! Menjadi :5! = 5 * 4 * 3!

Scr rekursif nilai dr 3! Dpt dihitung kembali dgn 3 * 2!, shg 5! Menjadi : 5! = 5 * 4 * 3 * 2!

Scr rekursif nilai dr 2! Dpt dihitung kembali dgn 2 * 1, shg 5! Menjadi : 5! = 5 * 4 * 3 * 2 * 1 = 120.

Contoh Listing Faktorial

```
#include <iostream.h>
#include <iomanip.h>
Unsigned long faktorial (unsigned long);
Int main()
{
        for (int i=0; i:<=10; i++)
                cout << setw(2) << i << "! Faktorial(i) << endl;
        return 0;
}
// recursive definition of function factorial
Unsigned long factorial (unsigned long number)
{
        if (number <=1)    // base case
                return 1;
        else
                return number * factorial(number – 1);
}
```

Fibonancy

Deret Fibonancy : 0,1,1,2,3,5,8,13,.....

Secara notasi pemrograman dapat ditulis sebagai :

Fibo (1) = 0 & Fibo (2) = 1 ..... (1)

Fibo (N) = Fibo (N-1) + Fibo (N-2) ..... (2)

Contoh :

Fibo(5) = Fibo(4) + Fibo(3)

Fibo(4) = Fibo(3) + Fibo(2)

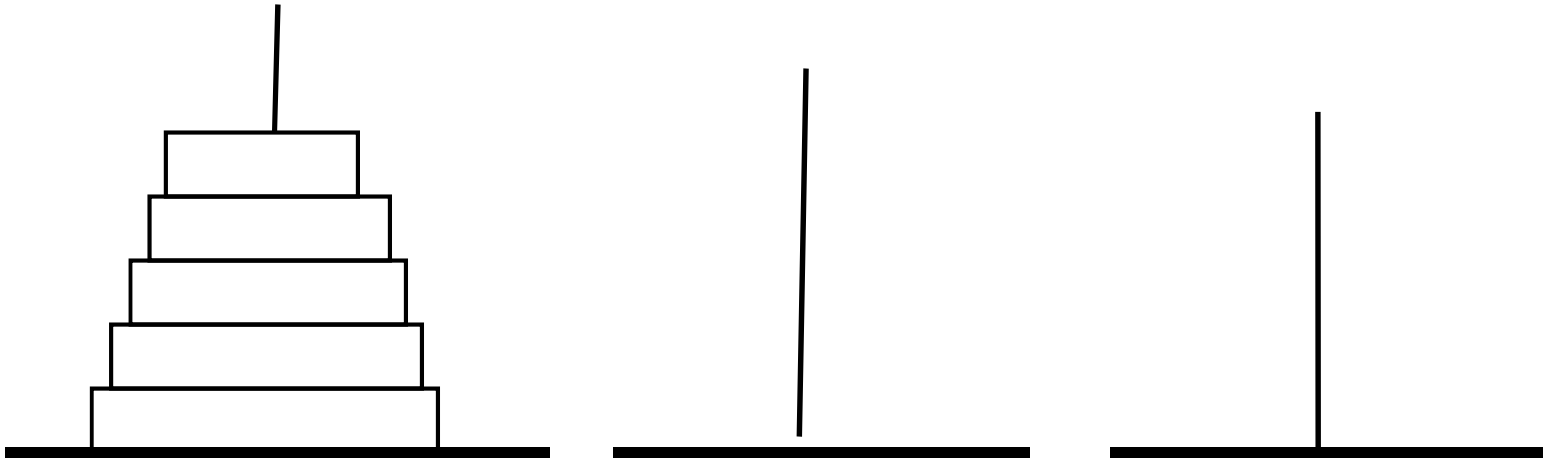
Fibo(3) = Fibo(2) + Fibo(1)

Nilai Awal

Deret Fibonancy

```
A[1] = 1;
A[2] = 2;
For (i=3; i<=10; i++)
 {
   A[i] = A[i-1] + A[i-2];
 }
```

Konsep Menara Hanoi



A

Tiang Asal

B

Tiang Bantuan

C

Tiang Tujuan

- Jika n=1, maka langsung pindahkan saja piringan dr tiang A ke tiang C & selesai.
- Pindahkan n-1 piringan yg paling atas dr tiang A ke tiang B.
- Pindahkan piringan ke n (piringan terakhir) dr tiang A ketiang C
- Pindahkan n-1 piringan dari tiang B ke tiang C.

Langkah pemindahan tsb diatas dpt diubah dengan notasi sbb:

Menara (n,asal,bantu,tujuan)

- Utk jml piringan n>1 dpt dibagi menjadi 3 notasi penyelesaian
- Menara (n-1, Asal,Tujuan, Bantu);
- Menara (n, Asal, Bantu, Tujuan); atau Asal → Tujuan;
- Menara (n-1, Bantu, Asal, Tujuan);

Langkah Pemindahan Piringan

MENARA(1,A,C,B) ..... A → B

MENARA(2,A,B,C) A → C ..... A → C

MENARA(1,B,A,C) .....B → C

MENARA(3,A,C,B) A→B ..... A → B

MENARA(1,C,B,A) .....C → A

MENARA(2,C,A,B)C → B ..... C → B

MENARA(1,A,C,B) ..... A → B

MENARA (4,A,B,C) A→C .....A → C

MENARA(1,B,A,C) ..... B → C

MENARA(2,B,C,A) B → A .....B → A

MENARA(1,C,B,A) ..... C → A

MENARA(3,B,A,C) B →C ..... B → C

MENARA(1,A,C,B) ..... A → B

MENARA(2,A,B,C) A → C ..... A → C

MENARA(1,B,A,C) ..... B → C

Ilustrasi diatas menghasilkan 15 langkah penyelesaian dari permasalahan konsep menara Hanoi dgn jumlah piringan sebanyak 4 buah18

Rumus Langkah Pemindahan :

$$2^N - 1$$

N = Jumlah Piringan