STRUKTUR REKURSIF

PERTEMUAN 6

DEFINISI REKURSIF

Rekursif adalah suatu proses yang bisa memanggil dirinya sendiri.

Dalam kehidupan sehari-hari banyak terdapat objek yang rekursif, contoh daun pakis atau pohon cemara.

Daun Pakis

Pohon Cemara

Lanjutan

Daun Pakis dibentuk oleh ranting-ranting daun yang mempunyai pola yang mirip dengan daun pakis itu sendiri.

Setiap ranting daun disusun lagi oleh ranting daun dengan pola yang mirip. Demikian juga dengan pohon cemara.

Objek yang mempunyai pola rekursif ini disebut **fraktal**.

Didalam bidang grafik dan seni, fraktal dimanfaatkan untuk membangkitkan gambar-gambar yang indah dan menawan.

STRUKTUR REKURSIF

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

- Fungsi ini digunakan untuk menghitung nilai: X^n dengan n berupa bilangan bulat positif. Solusi dari persoalan ini:
- JIKA n = 1 MAKA $X^n = X$
- SELAIN ITU: $X^n = X * X^{n-1}$

Bilangan Eksponensial

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 :

$10^3 = 10 * 10^2$

$10^2 = 10 * 10^1$

$10^1 = 10 * 10^0$

$10^0 = 1$

Algoritma Fungsi Pangkat:

Berikut adalah fungsi pangkat dengan menggunakan solusi di atas:

```
int pangkat(int a, int b){
    if (b==1){
        return a;
    }
    else {
        return a*pangkat(a,b-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

Penjelasan Langkah:

Faktorial (5) dapat juga dihitung sebagai berikut:

(1) 5! = 5 X 4!

(2) 4! = 4 x 3!

(3) 3! = 3 X 2!

(4) 2! = 2 X 1!

(5) 1! = 1 X 0

(6) 0! = 1

Runut balik dari baris 6 ke baris 1 maka didapat hasil

(6) 0! = 1

(5) 1! = 1 X 0! = 1 X 1 = 1

(4) 2! = 2 X 1! = 2 X 1 = 2

(3) 3! = 3 X 2! = 3 X 2 = 6

(2) 4! = 4 X 3! = 4 X 6 = 24

(1) 5! = 5 X 4! = 5 X 24 = 120

Contoh Mencari Faktorial 3

```
FAK(3)
If 3 = 0 then
   ...
Else
   FAK  3 * FAK(2)                                6

      If 2 = 0 then
         ...
      else
        FAK ←  2 * FAK(1)                         2

            If 1 = 0 then
                ...
            Else
              FAK ←1 * FAK(0)                     1

                  If 0 = 0 then                   1
                     FAK ←1
```

Contoh Soal :

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>
#include <conio.h>
unsigned long factorial (unsigned long number);
main()
{   for (int i=0; i<=10; i++)
            cout << setw(2) << i   << "! ="<<factorial(i) << endl;
    //return 0;
   getch();}
// recursive definition of function factorial
unsigned long factorial (unsigned long number)
{   if (number <=1)    // base case
            return 1;
    else
            return number * factorial(number-1);    }
```

Deret Fibonaccy

Deret Fibonaccy adalah deret dimana nilai suku ke n merupakan jumlah nilai suku ke n-1 dan suku ke n-2.

Untuk suku n > 2,

Suku pertama (n=1) nilainya adalah 1, dan suku ke dua (n=2) nilainya = 1.

Hasil deret fibonaccy adalah sebagai berikut:

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

Deret Fibonaccy Lanjutan

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) = $\underbrace{\text{Fibo(2) + Fibo(1)}}_{\textbf{Nilai Awal}}$

Algoritma Deret Fibonaccy

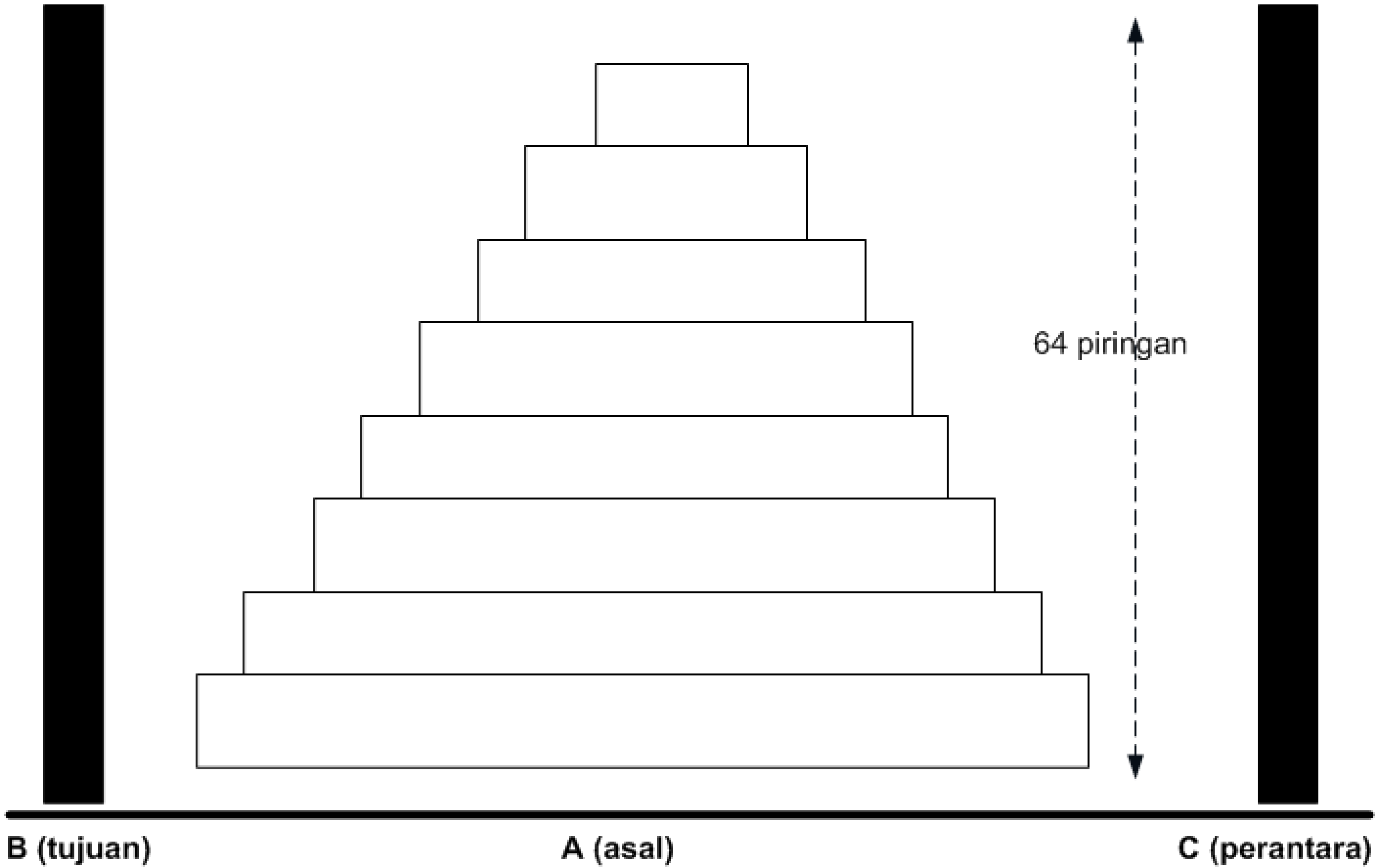
Deret Fibonancy

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

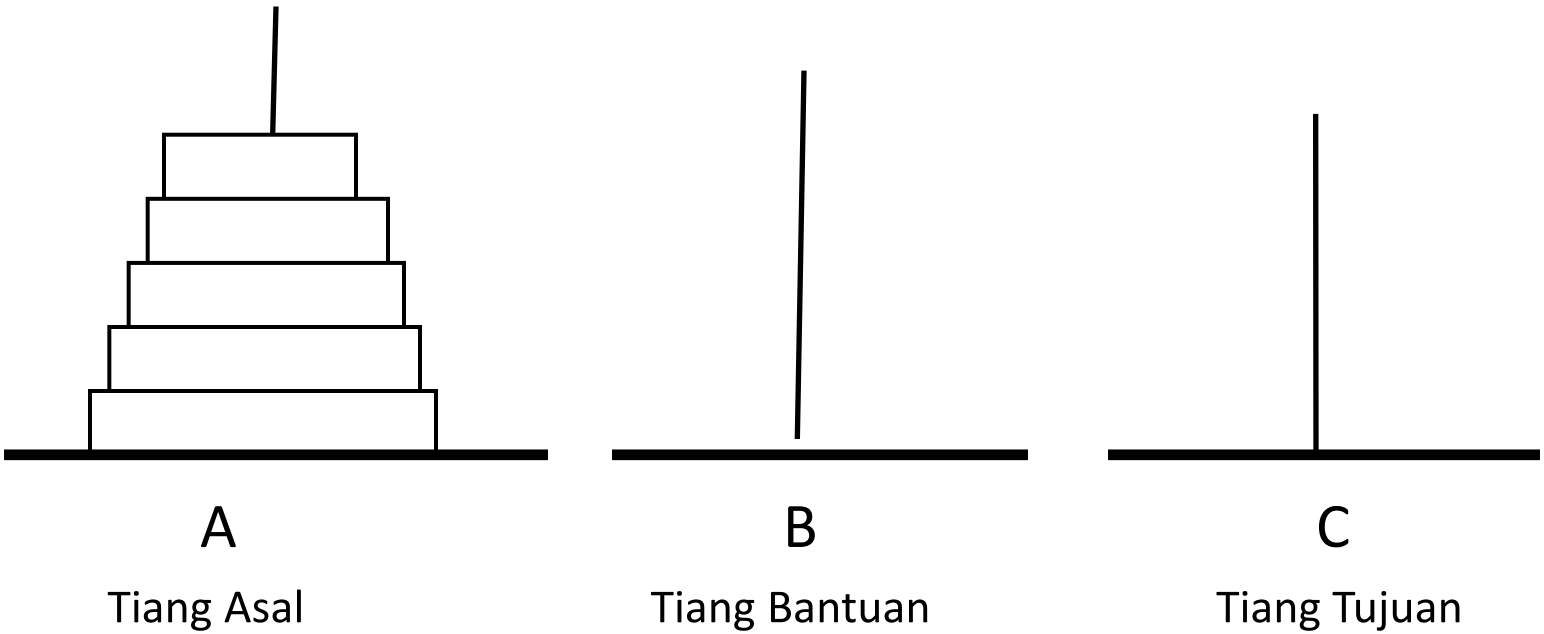
Menara Hanoi (*Tower of Hanoi*)

- Legenda klasik pendeta Budha di kota Hanoi (Vietnam)
- Terdapat tiga buah tiang tegak setinggi 5 meter dengan 64 buah piringan (disk).
- Setiap piringan mempunyai lubang ditengahnya agar dapat dimasukkan kedalam tiang.
- Bagaimana cara memindahkan seluruh piringan tersebut ke sebuah tiang yang lain, setiap kali hanya satu piringan yang boleh dipindahkan.
- Syarat: tidak boleh ada piringan besar diatas piringan yang lebih kecil.

Ilustrasi Menara Hanoi



Konsep Menara Hanoi



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.

Lanjutan

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

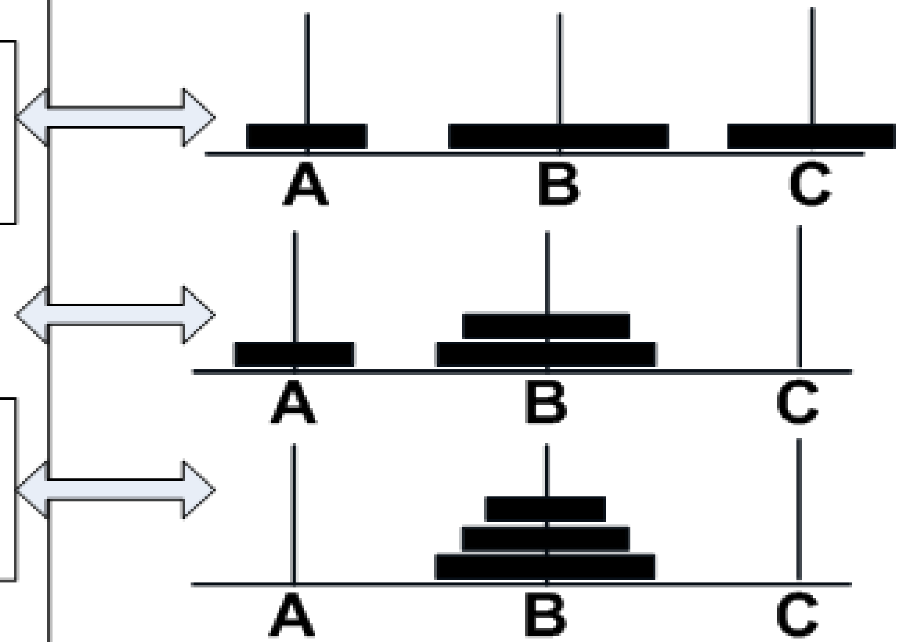
MENARA(1,A,C,B) .... A → B

MENARA(2,A,B,C) A → C .... A → C

MENARA(1,B,A,C) .... B → C

Contoh Ilustrasi Menara Hanoi dengan tiga piringan (N=3)

```
If 2 = 1 then
   …
else
   Hanoi (1, C, A, B)
  If 1 =1 then
     Pindahkan piringan dari C ke A
  else ...
    Pindahkan piringan dari C ke B
    Hanoi (1, A, B, C)
  If 1 =1 then
     Pindahkan piringan dari A ke B
  else ...
```



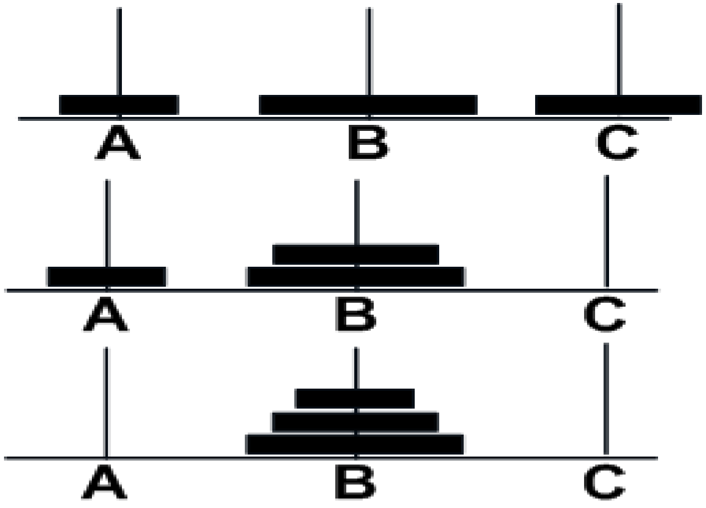
Lanjutan…

```
If 2 = 1 then
   ...
else
   Hanoi (1, C, A, B)
```

```
If 1 =1 then
   Pindahkan piringan dari C ke A
else ...
```

```
   Pindahkan piringan dari C ke B
   Hanoi (1, A, B, C)
```

```
If 1 =1 then
   Pindahkan piringan dari A ke B
else ...
```



Rumus Langkah Pemindahan :

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

Rumus Langkah Pemindahan :

$$\boxed{2^N - 1}$$

N = Jumlah Piringan

LATIHAN SOAL

PERTEMUAN 6

1. Suatu proses yang bisa memanggil dirinya sendiri, adalah :
   a. Branching
   b. Looping
   c. Iteratif
   d. Rekursif
   e. searching

2. Pada Menara Hanoi banyaknya pemindahan untuk N buah piringan ke menara tujuannya adalah :
   a. 2n +1
   b. 2n*1
   c. 2n-1
   d. 2^n -1
   e. 2+1

2. Pada Menara Hanoi banyaknya pemindahan untuk N buah piringan ke menara tujuannya adalah :

   a. 2n +1
   b. 2n*1
   c. 2n-1
   d. 2^n -1
   e. 2+1

3. FAKT(4) adalah :

   a. 1*2*3
   b. 2*FAKT(1)
   c. 4*FAKT(3)
   d. 3*3
   e. 3*FAKT(2)

3. FAKT(4) adalah :

   a. 1*2*3 d. 3*3

   b. 2*FAKT(1) e. 3*FAKT(2)

   c. 4*FAKT(3)

4. Untuk menyelesaikan masalah menara Hanoidengan banyaknya piringan ialah 5 buah, maka diperlukan pemindahan sebanyak :

   a. 30 kali c. 32 kali e. 28 kali

   b. 70 kali d. 31 kali

4. Untuk menyelesaikan masalah menara Hanoi dengan banyaknya piringan ialah 5 buah, maka diperlukan pemindahan sebanyak :

   a. 30 kali
   b. 70 kali
   c. 32 kali
   d. 31 kali
   e. 28 kali

5. Diketahui deret Fibonancy

   0,1,1,2,3,5,8,13, 21,x dan y

   maka nilai x dan y dari deret tersebut adalah :

   a. 34 dan 55
   b. 55 dan 34
   c. 21 dan 34
   d. 34 dan 21
   e. 13 dan 21

5. Diketahui deret Fibonancy

   0,1,1,2,3,5,8,13, 21,x dan y

   maka nilai x dan y dari deret tersebut adalah :

   a. 34 dan 55
   b. 55 dan 34
   c. 21 dan 34
   d. 34 dan 21
   e. 13 dan 21

1. Suatu proses yang bisa memanggil dirinya sendiri, adalah :

   a. Branching
   b. Looping
   c. Iteratif
   d. Rekursif
   e. searching