Analisis Rekurens

Oleh : Inggriani Liem
inge@informatika.org

Pelatnas I TOKI di ITB

Oktober 2014

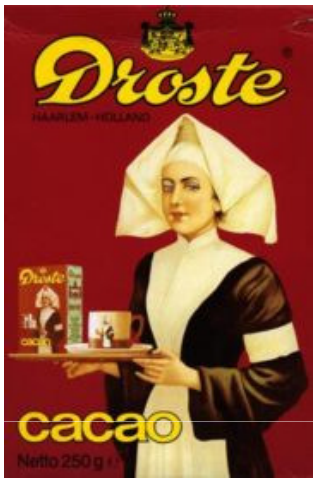
Tujuan

- Siswa memahami definisi rekurens, dan memakai analisis rekurens untuk konstruksi program
- Siswa mampu bemain dan mengimplementasi program rekursif :
  - fungsi rekursif
  - prosedur rekursif
  - struktur data rekursif

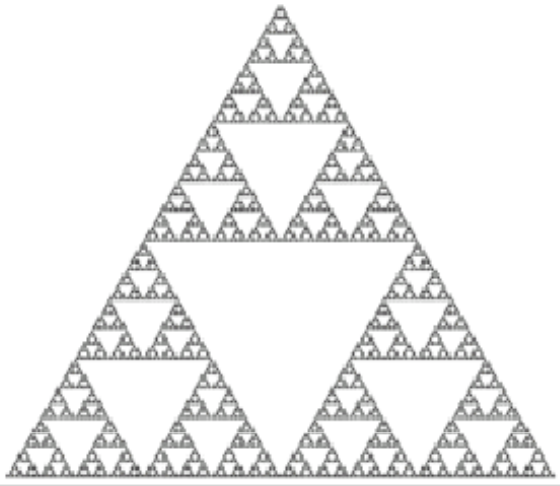
Definisi Rekurens

- Definisi entitas (type, fungsi) disebut rekursif jika definisi tersebut mengandung terminologi dirinya sendiri.
- Dalam pemrograman, realisasinya adalah dengan membuat prosedur atau fungsi rekursif.
- Program utama dalam bahasa Pascal tidak mungkin rekursif karena tidak punya parameter.

Gambar Rekursif



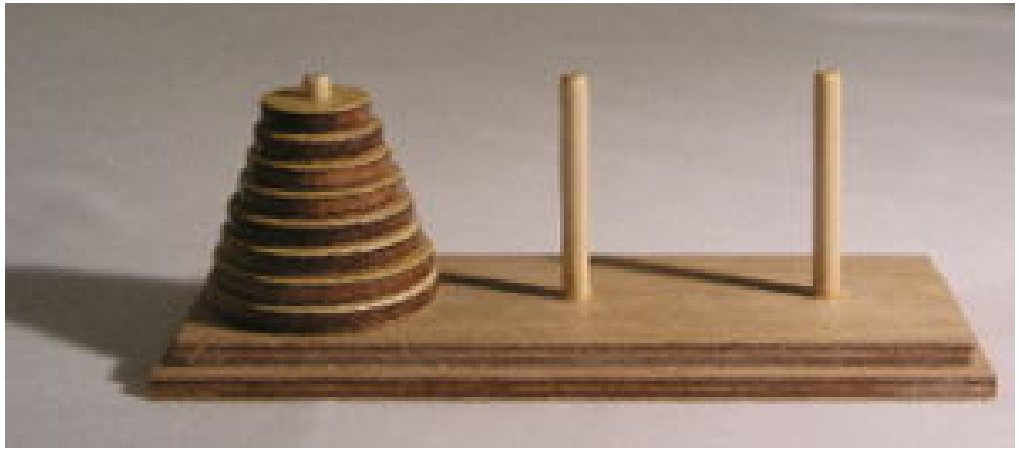
Rekursif dalam programming



A Sierpinski triangle—a confined recursion of triangles to form a geometric lattice.

- Factorial --
  $n! = n(n-1)! = n(n-1)\cdots 1$

  - Fibonacci numbers -- $f(n) = f(n-1) + f(n-2)$
  - Catalan numbers -- $C_0 = 1$, $C_{n+1} = (4n+2)C_n / (n+2)$



Dalam kehidupan sehari-hari

- The formula for the volume of a cylinder leads to the mathematical joke/self-description:
  - "What is the volume of a **pizza** of thickness **a** and radius **z** ?" Answer: **pi z z a**. This result is sometimes known as the second pizza theorem.
- Lihat gambar-gambar rekursif (Droste, Metro…)

Bahan Bacaan

- Akses Wikipedia dengan keyword : reccurence, recursive
- Akses semua link yang ada di situ

Analisis Rekurens

- Teks program rekursif terdiri dari dua bagian:
  - **Basis** (Basis-0 atau Basis-1), yang menyebabkan prosedur/fungsi berhenti
  - Bagian **rekurens** : mengandung call terhadap prosedur/fungsi tersebut, dengan parameter bernilai mengecil (menuju basis).
- Tuliskanlah secara eksplisit dalam teks program anda: mana bagian basis, mana rekurens

Contoh Definisi Rekursif Bilangan integer

bilangan integer

Basis : 0 adalah bilangan integer

Rekurens : if x adalah bilangan integer

then x+1 adalah bilangan integer

bilangan integer ganjil

Basis : 1 adalah bilangan integer ganjil

Rekurens: if x adalah bilangan integer ganjil

then x + 2 adalah bilangan integer ganjil

Type Rekursif

- Type rekursif :
  - Jika teks yang mendefinisikan type mengandung referensi terhadap diri sendiri, maka type disebut type rekursif.
  - Type dibentuk dengan komponen yang merupakan type itu sendiri.

Struktur Data Rekursif

- **List** :
  - list kosong adalah list
  - List tidak kosong :
    - elemen
    - Sisanya adalah List :
- **Pohon Biner**
  - Pohon Biner kosong adalah Pohon Biner
  - Pohon Biner tidak kosong :
    - Akar
    - SubPohon Kiri, adalah Pohon Biner
    - SubPohon Kanan , adalah Pohon Biner

Catatan :

Dua Struktur Data ini akan dibahas pada struktur Data Lanjut

Contoh Type Rekursif Dalam Bahasa C

```
/* definisi list linier */
typedef struct t_elmt * adress;
typedef struct t_elmt{
                int info;
                adress next } elmt;
adress L; /* L adalah list */
```

Contoh Type Rekursif Dalam Bahasa C

```
/* definisi binary tree */
typedef struct t_elmt * adress;
typedef struct t_elmt{
            int info;
            adress Left
            adress Right } Node;
adress P; /*P : binary tree */
```

Kerangka Fungsi Rekursif

```
int F (Param)
{
/* VAR lokal*/
  if (kondisi Basis)   /* Basis-0 atau Basis-1 */
     {
        return Ekspresi<berhenti>; }
  else{    /* Rekurens */
        return Ekspresi(F(Param mengecil))}
}
```

Contoh fungsi faktorial

```
int Faktorial (int N)
{
  if (n==0) /* Basis-0 */
   { return 1; }
  else {/* Rekurens */
     return n* Faktorial(n-1);}
}
```

Fungsi Rekursif

- Sedapat mungkin, jika harus membuat subprogram, realisasi dalam bentuk fungsi sebab lebih bebas side efek
- Fungsi rekursif selalu dapat ditransformasi menjadi prosedur rekursif. Akan dibahas pada contoh Faktorial.

Kerangka Prosedur Rekursif

```
void P (Param){
/* VAR lokal */
  if (kondisi Basis) /* Basis-0 atau Basis-1 */
  {
                <berhenti>
   else {
          /* Rekurens */
          P (Param  mengecil)}
}
```

Contoh Faktorial

```
int fak(int N) {

    if (N==0)   {
        return 1;
    } else {
        return (N*fak(N-1));
    }
}
```

Faktorial salah fatal

```
int fak(int N) {
    int tmp;
    if (N==0)  {
       return 1;
    } else {
       tmp = tmp * fak(N-1);
       return (tmp);
    }
}
```

/* kenapa salah fatal ??? */

Contoh Faktorial (2)

```
int fact0  (int N) {
    int HslTemp;
 /*kurang baik,variabel lokal tak perlu*/
    if (N == 0) { /* basis - 0*/
        HslTemp  = 1;
    } else {
        HslTemp = N * FSalah (N-1);
    }
   return HslTemp;
}
```

Contoh prosedur faktorial

```
void Faktorial(int N, int* Hsl)
{
  if (n==0) /* Basis-0 atau Basis-1 */
  {*Hsl = 1; }
   else {
     /* Rekurens */
    Faktorial (N-1,Hsl);
    /* Faktorial (N-1, &(*Hsl) */
      *Hsl = N * (*Hsl); }
}
```

Prosedur Rekursif

- Untuk Prosedur rekursif, perhatikan Initial State dan Final State.
- Perlu diperhatikan pula cara pemanggilannya (tidak se-*natural* seperti fungsi yang dipakai dalam ekspresi)
- Akan ditunjukkan pada contoh-contoh program C yang dibahas

Variabel Global

- Hati-hati dengan variable global. Walaupun demikian, dalam beberapa kasus, akan dibutuhkan variabel global.
- Untuk tabel : Apakah seluruh variable perlu dipassing sebagai parameter, atau cukup indeksnya.

Basis Nol atau Satu ?

- Jika menangani kasus kosong, maka gunakan basis-0. Karena “kosong” adalah konsep “ciptaan” kita, maka hati-hati dengan nilai yang dihasilkan oleh kasus kosong.
- Jika persoalan hanya ada artinya kalau tidak kosong, maka harus memakai basis-1. Contoh : nilai maksimum elemen tabel tidak akan terdefinisi kalau tabelnya kosong (tidak ada elemennya).

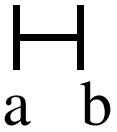
Studi Kasus Faktorial

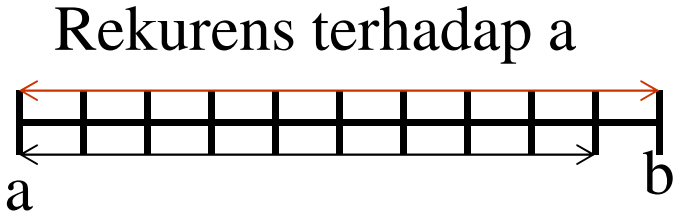
- Program faktorial berdasar definisi rekursif
- Program Faktorial dengan loop, yang kemudian dijadikan rekursif :
  - parameter output
  - variabel lokal
  - parameter input/output

Summation

- Bagaimana menghitung :

  Sigma (a, b) : menghitung penjumlahan nilai setiap titik i dengan i= 1+2+3+..N… untuk interval [a,b]





Basis :

b<a: tidak ada interval (kasus kosong)

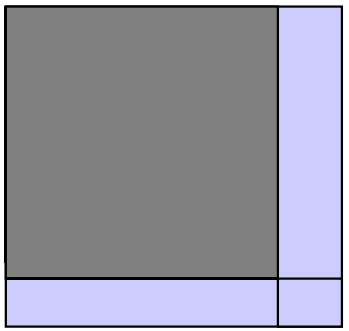
Pemrosesan Tabel secara rekursif

- Dengan berinspirasi ke Sigma (i, a,b) dituliskan beberapa proses tabel secara rekursif :
  - mengisi tabel
  - menulis isi tabel
  - menjumlahkan elemen
  - menghitung banyaknya elemen positif
- Latihan : hitunglah maximum dan Occurrence nilai maksimum secara rekursif

Tabel sebagai Var global

```
#include<iostream>
using namespace std;
int T[] = {1,2,3,4};
void printRec(int a, int b) {
  if (b<a) {/* do nothing */ }
  else { printf ("%d", T[a], "\n"); printRec
(a+1,b); }
}
int main () {
  printRec(0,3);
  return 0;
}
```

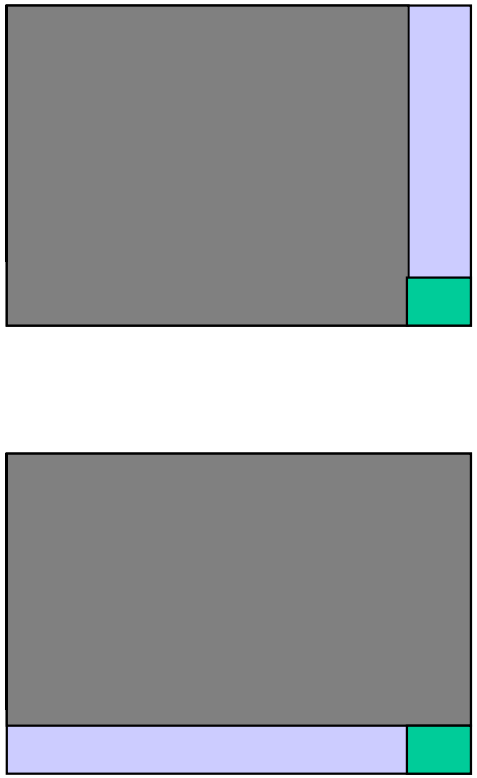
Pemrosesan Matriks Bujur Sangkar Secara Rekursif



Untuk Matriks Bujur Sangkar (NxN)

Basis : Matriks (1x1)
Rekurens, perhatikan Area :
- Matriks dengan area N-1 x N-1 +
- Area tabel horizontal (1 .. N-1) +
- Area tabel vertikal (1 .. N-1)

Pemrosesan Matriks Secara Rekursif



Untuk Matriks Bujur Sangkar Bukan Bujur Sangkar (Nx M) : ada 3 pilihan definisi rekursif:

a. Seperti matriks Bujur sangkar
b. Matriks (N x M-1), elemen (N,M), Matriks kolom (ke-M)
c. Matriks (N-1xM), elemen (N,M), Matriks Baris (ke-N)

Kenapa Membuat Program Rekursif?

- Persoalan yang ingin diselesaikan memang mengandung definisi “rekursif”.
- Program yang dibuat mengolah data bertype rekursif.
- Karena programmernya ingin menulis solusi rekursif:) Persoalan yang tidak rekursif tidak perlu diselesaikan secara rekursif, tapi bisa diselesaikan secara rekursif. Ada banyak jalan ke Roma…...

Mencari Solusi Rekursif

- Solusi Rekursif yang berangkat dari definisi rekursif :Baca baik-baik persoalan, dan buat definisi rekursif sesuai definisi persoalan
- Solusi rekursif karena persoalan diwakili oleh struktur data rekursif
- Buat prosedur/fungsi rekursif (anda harus membuat prosedur/fungsi dengan parameter)

Persoalan Program Rekursif

- Harus dijamin berhenti.
- Sulit dibayangkan eksekusinya tanpa terikat definisinya.
- Performansi, keterbatasan implementasi mesin pengeksekusi.

Call rekursif sbg Mekanisme "mengulang"

- Mekanisme eksekusi pengulangan , dengan loop while, repeat,...
- Mekanisme eksekusi call rekursif
- Akan diberikan contoh:
  - prosedur rekursif yang merupakan translasi dari loop
  - Translasi dari program rekursif menjadi loop

Penutup
Program Rekursif

- Program - fungsi - prosedur rekursif
- Definisi rekursif: basis dan rekurens
- Rekurens karena definisi, atau karena mekanisme eksekusi.
- Bedakan kasus khusus dengan kasus basis
- Akan diberikan contoh-contoh ...

Recurrence Relation

- Contoh penggunaan recurrence relation pada perhitungan deret (lihat contoh iteratif yang diberikan)
- Jika hanya linier, biasanya dilakukan dengan Loop
- Contoh yang akan diberikan : deret hitung, deret sinus, dan sejenisnya
- Akan diberikan contoh konstruksi secara iteratif yang diprogram secara rekursif