Pertemuan 4

RELASI REKURENSI

Materi Pekuliahan Matematika Diskrit – September 2020

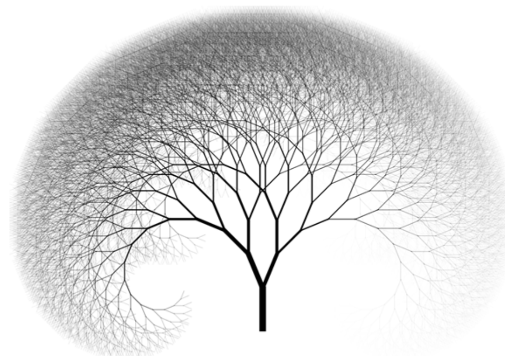
(Referensi: Rinaldi Munir, Matematika Diskrit: informatika.stei.itb.ac.id/~rinaldi.munir/Matdis)

AGENDA

1. Rekursi dan relasi Rekurensi
2. Fungsi rekursif
3. Sruktur rekursif
4. Relasi rekurensi
5. Pemodelan
6. Penyelesaian relasi rekurensi

1. Rekursi dan Relasi Rekurensi

- Beberapa pengamatan



(tensor-programming.com)

- Sebuah objek dikatakan **rekursif** (*recursive*) jika ia didefinisikan dalam terminologi dirinya sendiri.
- Proses mendefinisikan objek dalam terminologi dirinya sendiri disebut **rekursi** (*recursion*).

2. Fungsi Rekursif

Fungsi rekursif didefinisikan oleh dua bagian:

a. Basis

- Bagian basis berisi nilai fungsi yang terdefinisi secara eksplisit.

b. Rekurens

- Bagian ini mendefinisikan fungsi dalam terminologi dirinya sendiri.
- Berisi kaidah untuk menemukan nilai fungsi input dari nilai-nilai input lain yang lebih kecil.

Contoh 1

Misalkan fungsi f didefinisikan secara rekursif sebagai berikut:

$$f(n) = \begin{cases} 3 & , n = 0 \quad \text{basis} \\ 2f(n-1)+4 & , n > 0 \quad \text{rekurensi} \end{cases}$$

Tentukan nilai f(4).

Penyelesaian:

f(4) = 2f(3)+4
= 2(2f(2) + 4) + 4
= 2(2(2f(1)+4)+4)+4
= 2(2(2(2(f(0)+4)+4)+4)+4
= 2(2(2(2(3)+4)+4)+4)+4
= 108

Menghitung dengan cara lain:

f(0) = 3

f(1) = 2f(0) + 4 = 2 · 3 + 4 = 10

f(2) = 2f(1) + 4 = 2 · 10 + 4 = 24

f(3) = 2f(2) + 4 = 2 · 24 + 4 = 52

f(4) = 2f(3) + 4 = 2 · 52 + 4 = 108

Jadi, f(4) = 108.

Contoh 2

Nyatakan *n*! dalam definisi rekursif.

Penyelesaian:

$$n! = \underbrace{1\times 2\times 3\times \ldots \times (n-1)}_{(n-1)!}\times n = (n-1)!\times n$$

Misalkan f(n) = n!, maka

$$n! = \begin{cases} 1 & ,n=0 \\ n\cdot(n-1)! & ,n>0 \end{cases}$$

Menghitung 5! secara rekursif adalah:

5! = 5 · 4! = 5 · 4 · 3! = 5 · 4 · 3 · 2!
= 5 · 4 · 3 · 2 · 1! = 5 · 4 · 3 · 2 · 1 · 0!
= 5 · 4 · 3 · 2 · 1 · 1 = 120

Contoh 3

Barisan Fibonacci 0, 1, 1, 2, 3, 5, 8, 11, 19, …. Dapat dinyatakan secara rekursif sebagai berikut:

$$f_n = \begin{cases} 0 & ,n=0 \\ 1 & ,n=1 \\ f_{n-1}+f_{n-2} & ,n>1 \end{cases}$$

Latihan

1. Definisikan a^n secara rekursif ; dengan a adalah bilangan riil tidak-nol dan n adalah bilangan bulat tidak-negatif.

2. Nyatakan $a \times b$ secara rekursif; dengan a dan b adalah bilangan bulat positif.

11

Penyelesaian

1.
$$a^n = \underbrace{a \cdot a \cdot a \cdot \ldots a}_{n\text{ kali}} = a \cdot \underbrace{a \cdot a \cdot a \cdot \ldots a}_{n-1\text{ kali}} = a \cdot a^{n-1}$$

sehingga:

$$a^n = \begin{cases} 1 & , n = 0 \\ a \cdot a^{n-1} & , n > 0 \end{cases}$$

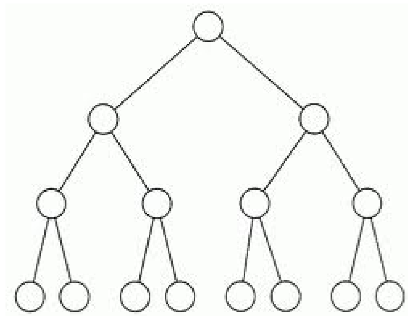
2.
$$a \cdot b = \underbrace{b + b + b + \ldots + b}_{a\text{ kali}}$$
$$= b + \underbrace{b + b + \ldots + b}_{a-1\text{ kali}}$$
$$= b + (a-1)b$$

$$\Rightarrow \quad a \cdot b = \begin{cases} b & , a = 1 \\ b + (a-1)b & , a > 1 \end{cases}$$

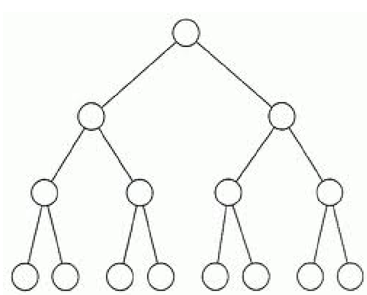
12

3. Struktur Rekursif

Di dalam ilmu computer, ada struktur data penting. Salah satunya adalah pohon biner (binary tree) sebagai berikut:

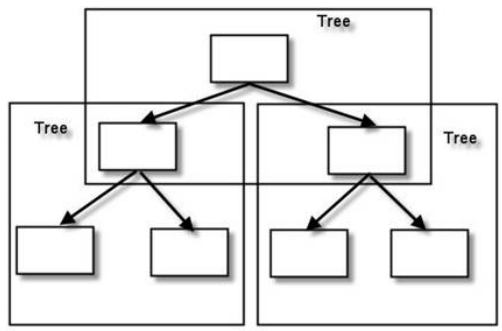


Struktur pohon biner



a. Simpul (***node***) pada pohon biner mempunyai paling banyak dua buah anak.
b. Jumlah anak pada setiap simpul bisa 1, 2, atau 0.
c. Simpul yang mempunyai anak disebut simpul cabang (***branch node***) atau simpul dalam (*internal node*)
d. Simpul yang tidak mempunyai anak disebut simpul daun (***leaf***).

Pohon biner adalah struktur yang rekursif, sebab setiap simpul mempunyai cabang yang juga berupa pohon. Setiap cabang disebut *subtree*.

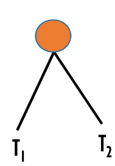


Binary tree consisting of 3 binary trees

15

Oleh karena itu, pohon dapat didefinisikan secara rekursif sebagari berikut:

a. Basis: kosong adalah pohon biner
b. Rekurens: Jika T_1 dan T_2 adalah pohon biner maka struktur berikut adalah pohon biner

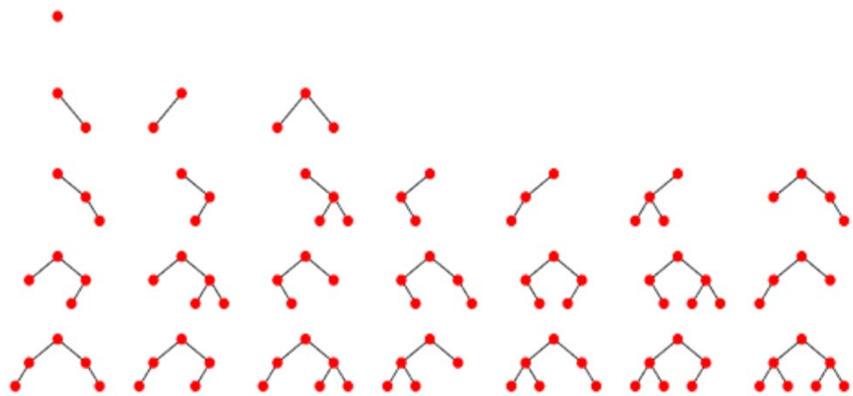


16

Proses pembentukan pohon biner secara rekursif:

(i) ϕ

(ii)



4. Barisan Rekursif

- Perhatikan barisan bilangan berikut ini: 1, 2, 4, 8, 16, 32, 64, …
- Setiap elemen ke-n untuk n = 0, 1, 2, … merupakan hasil perpangkatan 2 dengan n, atau

$$a_n = 2^n.$$

- Secara rekursif, setiap elemen ke-n merupakan hasil kali elemen sebelumnya dengan 2, atau

$$a_n = 2a_{n-1}.$$

Basis: $a_0 = 1$

Rekurens: $a_n = 2a_{n-1}$.

Contoh 4

Koloni bakteri dimulai dari lima buah bakteri. Setiap bakteri membelah diri menjadi dua bakteri baru setiap satu jam. Berapa jumlah bakteri baru sesudah *4* jam?

Misalkan a_n = jumlah bakteri setelah *n* jam, yang dapat dinyatakan dalam relasi rekursif sebagai berikut:

$$a_n = \begin{cases} 5 & , n = 0 \\ 2a_{n-1} & , n > 0 \end{cases}$$

n = 1 → jumlah bakteri = $a_1 = 2a_0 = 2 \cdot 5 = 10$

n = 2 → jumlah bakteri = $a_2 = 2a_1 = 2 \cdot 10 = 20$

n = 3 → jumlah bakteri = $a_3 = 2a_2 = 2 \cdot 20 = 40$

n = 4 → jumlah bakteri = $a_4 = 2a_3 = 2 \cdot 40 = 80$

Jadi, setelah 4 jam terdapat 80 buah bakteri

6. Relasi Rekurens

- Barisan (*sequence*) $a_0, a_1, a_2, \ldots, a_n$ dilambangkan dengan $\{a_n\}$
- Elemen barisan ke-*n*, yaitu a_n, dapat ditentukan dari suatu persamaan.
- Bila persamaan yang mengekspresikan a_n dinyatakan secara rekursif dalam satu atau lebih *term* elemen sebelumnya, yaitu $a_0, a_1, a_2, \ldots, a_{n-1}$, maka persamaan tersebut dinamakan **relasi rekurens**.

Contoh:

$a_n = 2a_{n-1} + 1$

$a_n = a_{n-1} + 2a_{n-2}$

$a_n = 2a_{n-1} - a_{n-2}$

- **Kondisi awal** (*initial conditions*) suatu barisan adalah satu atau lebih nilai yang diperlukan untuk memulai menghitung elemen-elemen selanjutnya.

  Contoh: $a_n = 2a_{n-1} + 1$; $a_0 = 1$

  $a_n = a_{n-1} + 2a_{n-2}$; $a_0 = 1$ dan $a_1 = 2$

- Karena relasi rekurens menyatakan definisi barisan secara rekursif, maka kondisi awal merupakan langkah basis pada definisi rekursif tersebut.

- **Contoh 5**. Barisan Fibonacci 0, 1, 1, 2, 3, 5, 8, 13, …

  dapat dinyatakan dengan relasi rekurens

  $f_n = f_{n-1} + f_{n-2}$; $f_0 = 0$ dan $f_1 = 1$

- Kondisi awal secara unik menentukan elemen-elemen barisan. Kondisi awal yang berbeda akan menghasilkan elemen-elemen barisan yang berbeda pula.

- Solusi dari sebuah relasi rekurens adalah sebuah formula yang tidak melibatkan lagi *term* rekursif. Formula tersebut memenuhi relasi rekurens yang dimaksud.

- **Contoh 6**: Misalkan $\{a_n\}$ adalah barisan yang memenuhi relasi rekurens berikut:

  $a_n = 2a_{n-1} - a_{n-2}$; $a_0 = 1$ dan $a_1 = 2$

  Periksa apakah $a_n = 3n$ merupakan solusi relasi rekurens tersebut.

  Penyelesaian: $2a_{n-1} - a_{n-2} = 2[3(n-1)] - 3(n-2)$

  $= 6n - 6 - 3n + 6$

  $= 3n = a_n$

  Jadi, $a_n = 3n$ merupakan solusi dari relasi rekurens tersebut.

- Apakah $a_n = 2^n$ merupakan solusi relasi rekurens

  $a_n = 2a_{n-1} - a_{n-2}$; $a_0 = 1$ dan $a_1 = 2$?

  Penyelesaian: $2a_{n-1} - a_{n-2} = 2 \cdot 2^{n-1} - 2^{n-2}$

  $= 2^{n-1+1} - 2^{n-2}$

  $= 2^n - 2^{n-2} \neq 2^n$

  Jadi, $a_n = 2^n$ bukan merupakan solusi relasi rekurens tsb.

  Cara lain: Karena $a_0 = 1$ dan $a_1 = 2$, maka dapat dihitung

  $a_2 = 2a_1 - a_0 = 2 \cdot 2 - 1 = 3$

  Dari rumus $a_n = 2^n$ dapat dihitung $a_0 = 2^0 = 1$,

  $a_1 = 2^1 = 2$, dan $a_2 = 2^2 = 4$

  Karena $3 \neq 4$, maka $a_n = 2^n$ bukan merupakan solusi dari relasi rekurens tsb.