Rekursi dan Relasi Rekurens
Bagian 1

Bahan Kuliah IF2120 Matematika Diskrit

Oleh: Rinaldi Munir

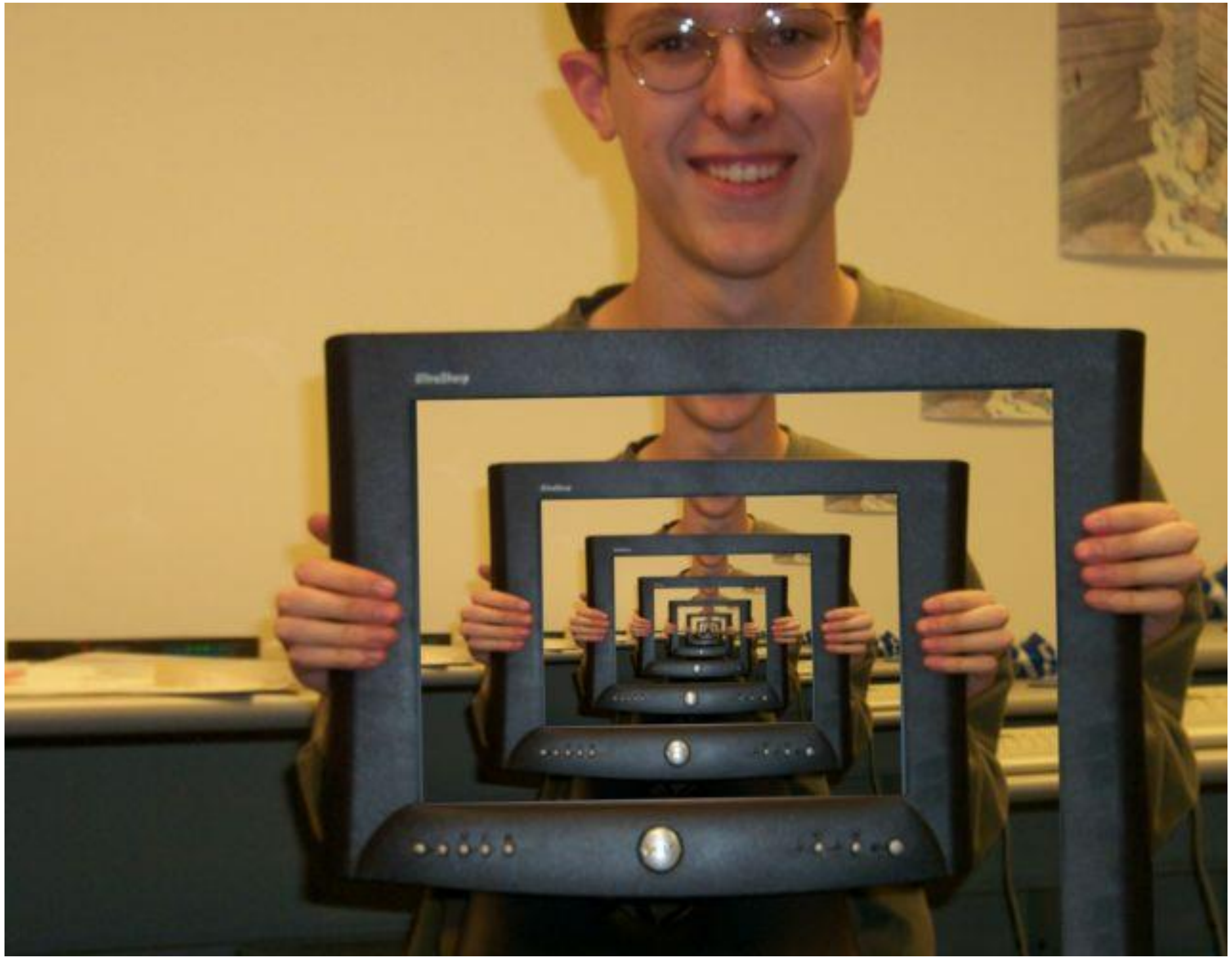
Program Studi Informatika
Sekolah Teknik Elektro dan Informatika (STEI)
ITB

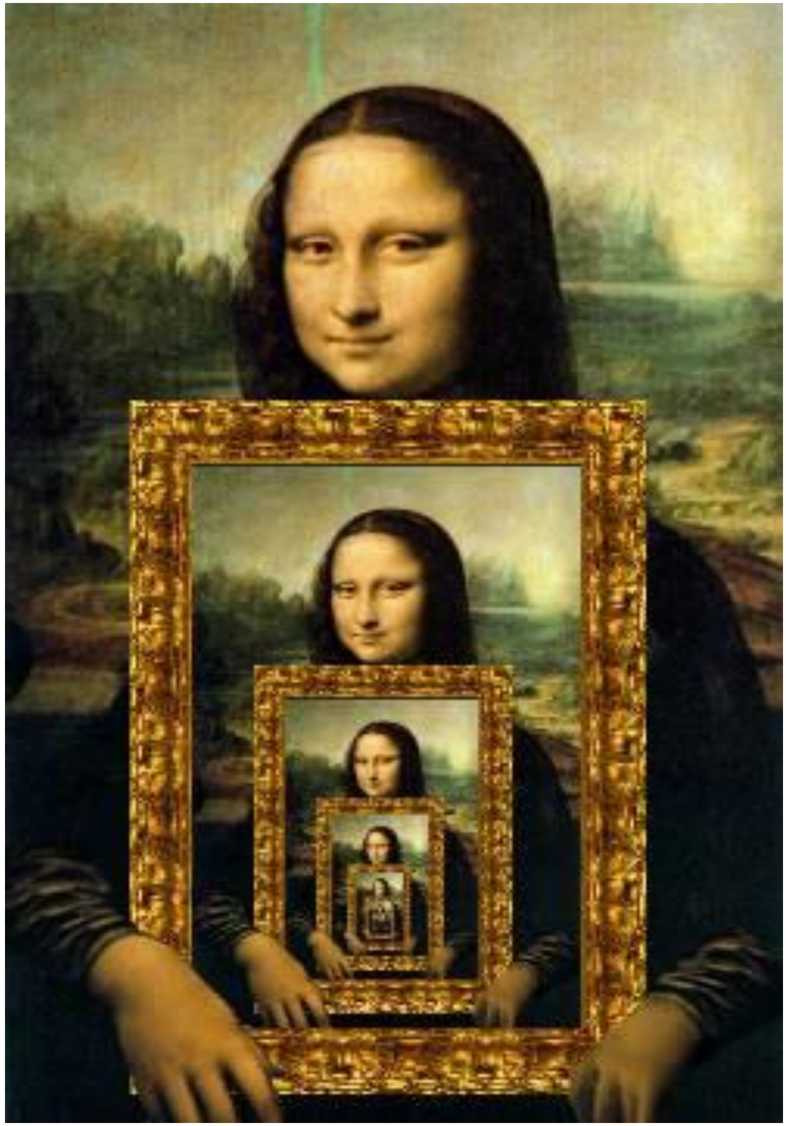
Rekursi

- Sebuah objek dikatakan **rekursif** (*recursive*) jika ia didefinisikan dalam terminologi dirinya sendiri.

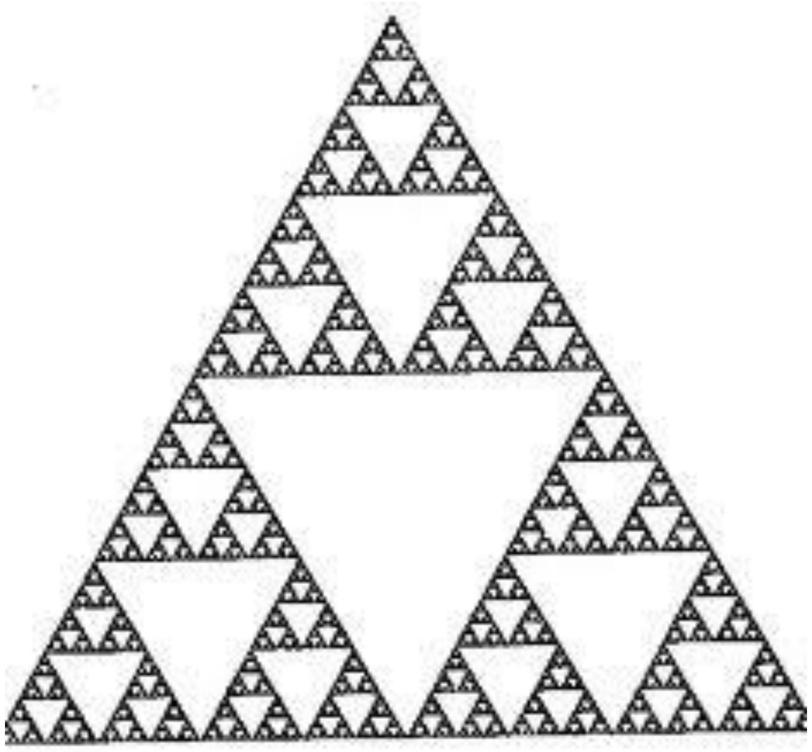
- Proses mendefinisikan objek dalam terminologi dirinya sendiri disebut **rekursi** (*recursion*).

- Perhatikan tiga buah gambar pada tiga *slide* berikut ini.





- Objek fraktal adalah contoh bentuk rekursif.



Fraktal di alam

Fungsi Rekursif

- Fungsi rekursif didefinisikan oleh dua bagian:

(i) *Basis*

- Bagian yang berisi nilai fungsi yang terdefinisi secara eksplisit.
- Bagian ini juga sekaligus menghentikan rekursif (dan memberikan sebuah nilai yang terdefinisi pada fungsi rekursif).

(ii) *Rekurens*

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

- **Contoh 1**: Misalkan *f* didefinsikan secara rekusif sbb

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

basis

rekurens

Tentukan nilai f(4)!

Solusi: f(4) = 2f(3) + 4

= 2(2f(2) + 4) + 4

= 2(2(2f(1) + 4) + 4) + 4

= 2(2(2(2f(0) + 4) + 4) + 4) + 4

= 2(2(2(2·3 + 4) + 4) + 4) + 4

= 2(2(2(10) + 4) + 4) + 4

= 2(2(24) + 4) + 4

= 2(52) + 4

= 108

Cara lain menghitungnya:

$f(0) = 3$

$f(1) = 2f(0) + 4 = 2 \cdot 3 + 4 = 10$

$f(2) = 2f(1) + 4 = 2 \cdot 10 + 4 = 24$

$f(3) = 2f(2) + 4 = 2 \cdot 24 + 4 = 52$

$f(4) = 2f(3) + 4 = 2 \cdot 52 + 4 = 108$

Jadi, $f(3) = 108$.

- **Contoh 2**: Nyatakan *n*! dalam definisi rekursif

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

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

- Algoritma menghitung faktorial:

```
function Faktorial (input  n :integer)→integer
{ mengembalikan nilai n!;
  basis   : jika n = 0, maka 0! = 1
  rekurens: jika n > 0, maka n! = n × (n-1)!
}
DEKLARASI
      -
ALGORITMA:
   if n = 0 then
     return 1                          { basis }
   else
     return  n * Faktorial(n – 1)        { rekurens }
   end
```

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

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

- **Contoh 4**: Fungsi (polinom) Chebysev dinyatakan sebagai

$$T(n, x) = \begin{cases} 1, & n = 0 \\ x, & n = 1 \\ 2x \cdot T(n-1, x) - T(n-2, x), & n > 1 \end{cases}$$

- **Contoh 5**: Deret $\sum_{k=0}^{n} a_k$ didefinisikan secara rekursif sebagai berikut:

$$\sum_{k=0}^{n} a_k = a_0 + a_1 + a_2 + \ldots + a_{n-1} + a_n$$
$$= (a_0 + a_1 + a_2 + \ldots + a_{n-1}) + a_n$$
$$= (\sum_{k=0}^{n-1} a_k) + a_n$$

sehingga

$$\sum_{k=0}^{n} a_k = \begin{cases} a_0 & , n = 0 \\ (\sum_{k=0}^{n-1} a_k) + a_n & , n > 0 \end{cases}$$

- **Latihan**

1. Definisikan a^n secara rekursif , yang dalam hal ini a adalah bilangan riil tidak-nol dan n adalah bilangan bulat tidak-negatif.

2. Nyatakan $a \times b$ secara rekursif, yang dalam hal ini a dan b adalah bilangan bulat positif.

(Solusinya ada setelah slide berikut!)

- Solusi:

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. $$a \cdot b = \underbrace{b + b + b + \ldots + b}_{a\ \text{kali}}$$

$$= b + \underbrace{b + b + \ldots + b}_{a-1\,\text{kali}}$$

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

Himpunan Rekursif

- String adalah rangkaian sejumlah karakter

  Contoh:

  ‘itb’ disusun oleh karakter i, t,dan b

  ‘informatika’ disusun oleh karakter i, n, f, o, r, m, a, t, i, k, a

- String kosong (*null string*) atau ‘’ adalah string dengan panjang nol . Notasi: λ

- Alfabet adalah himpunan karakter yang elemen-elemennya adalah penyusun string. Notasi: $\sum$

  Contoh: $\sum$ = {0, 1}, $\sum$ = {a, b, c, …, z}

- Misalkan Σ^* adalah himpunan string yang dibentuk dari alfabet Σ, maka Σ^* dapat didefinisikan secara rekursif sebagai berikut:

  (i) Basis: $\lambda \in \Sigma^*$

  (ii) Rekurens: Jika $w \in \Sigma^*$ dan $x \in \Sigma$, maka $wx \in \Sigma^*$

- **Contoh 6**: Misalkan $\Sigma = \{0, 1\}$, maka elemen-elemen Σ^* dibentuk sebagai berikut:

  (i) λ (basis)

  (ii) $0 + \lambda = 0$, $1 + \lambda = 1$

  $0 + 1 = 01$, $0 + 0 = 00$, $1 + 0 = 10$, $0 + 0 = 00$, $1 + 1 = 11$

  $00 + 1 = 001$,

  010, 110, 1110, 110001, ….dst

- Sebuah *string* dibentuk dari penyambungan (*concatenation*) sebuah string dengan string lain.

  Contoh: 'a' · 'b' = 'ab'

  'w' · 'xyz' = 'wxyz'

  'itb' · ' ' = 'itb ' (tanda · menyatakan *concatenation*)

- Penggabungan dua buah string dapat didefinisikan secara rekursif sebagai berikut:

  (i) Basis: Jika $w \in \Sigma^*$, maka $w \cdot \lambda = w$, yang dalam hal ini λ adalah string kosong

  (ii) Rekurens: Jika $w_1 \in \Sigma^*$ dan $w_2 \in \Sigma^*$ dan $x \in \Sigma$, maka

  $w_1 \cdot w_2 \cdot x = (w_1 \cdot w_2) \cdot x$

- Panjang sebuah string adalah banyaknya karakter di dalam string tersebut.

  Contoh:

  ‘itb’ panjangnya 3

  ‘informatika’ panjangnya 11

  λ (string kosong) panjangnya 0

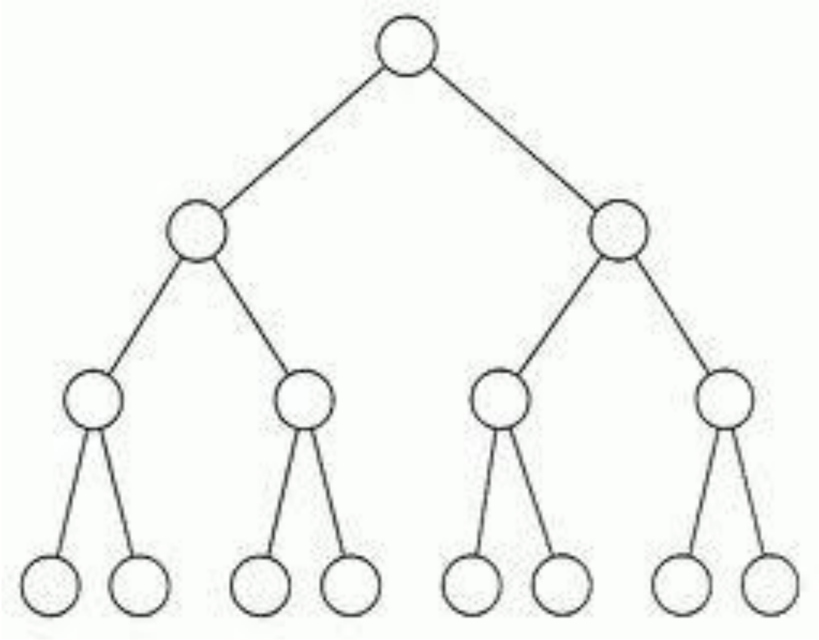
- Panjang string (disimbolkan dengan L) dapat didefinisikan secara rekursif:

  (i) Basis: $L(\lambda) = 0$

  (ii) Rekurens: $L(wx) = L(w) + 1$ jika $w \in \Sigma^*$ dan $x \in \Sigma$

Struktur Rekursif

- Struktur data yang penting dalam komputer adalah pohon biner (*binary tree*).



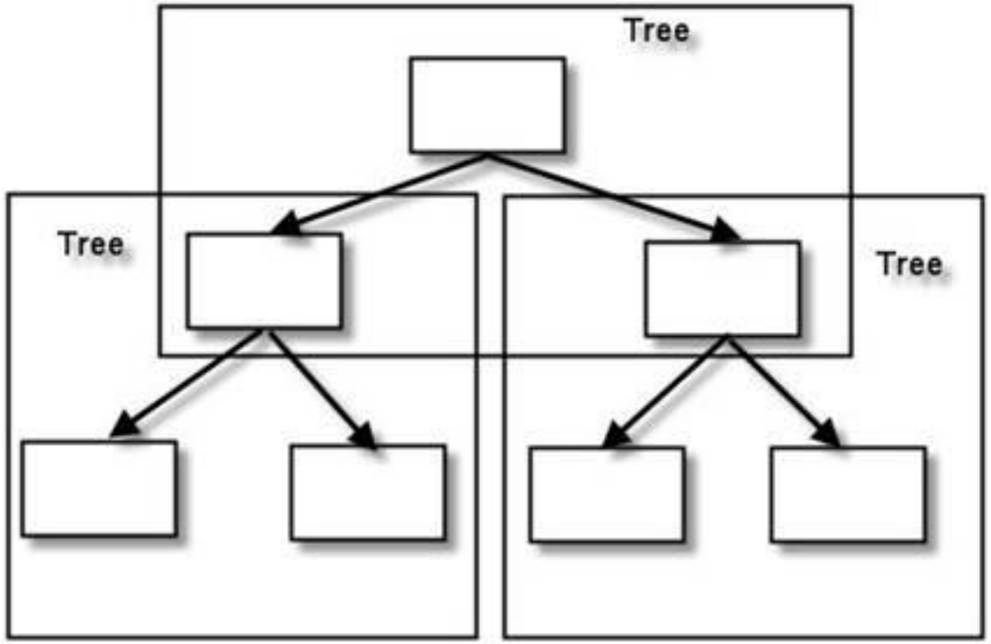
- Simpul (*node*) pada pohon biner mempunyai paling banyak dua buah anak.

- Jumlah anak pada setiap simpul bisa 1, 2, atau 0.

- Simpul yang mempunyai anak disebut simpul cabang (*branch node*) atau simpul dalam (*internal node*)

- Simpul yang tidak mempunyai anak disebut simpul daun (*leave*).

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

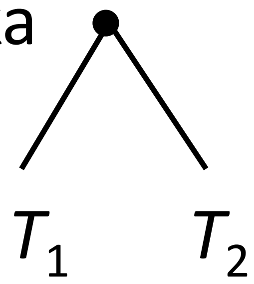


Binary tree consisting of 3 binary trees

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

  (i) Basis: kosong adalah pohon biner

  (ii) Rekurens: Jika T_1 dan T_2 adalah pohon biner, maka

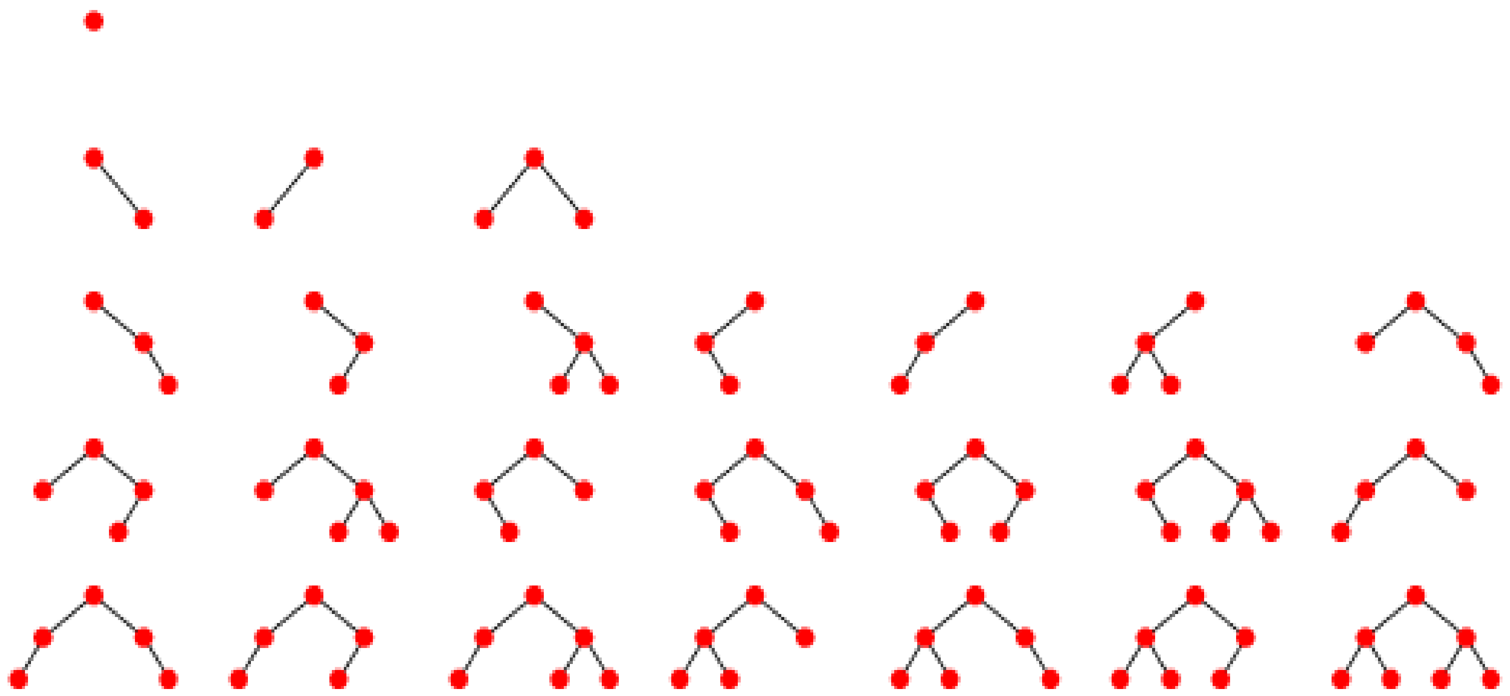
  

  adalah pohon biner

Proses pembentukan pohon biner secara rekursif:

(i) ϕ

(ii)



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 7**: 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