Rekursi dan Relasi Rekurens

Bahan Kuliah IF2120 Matematika Diskrit

Oleh: Rinaldi MUnir

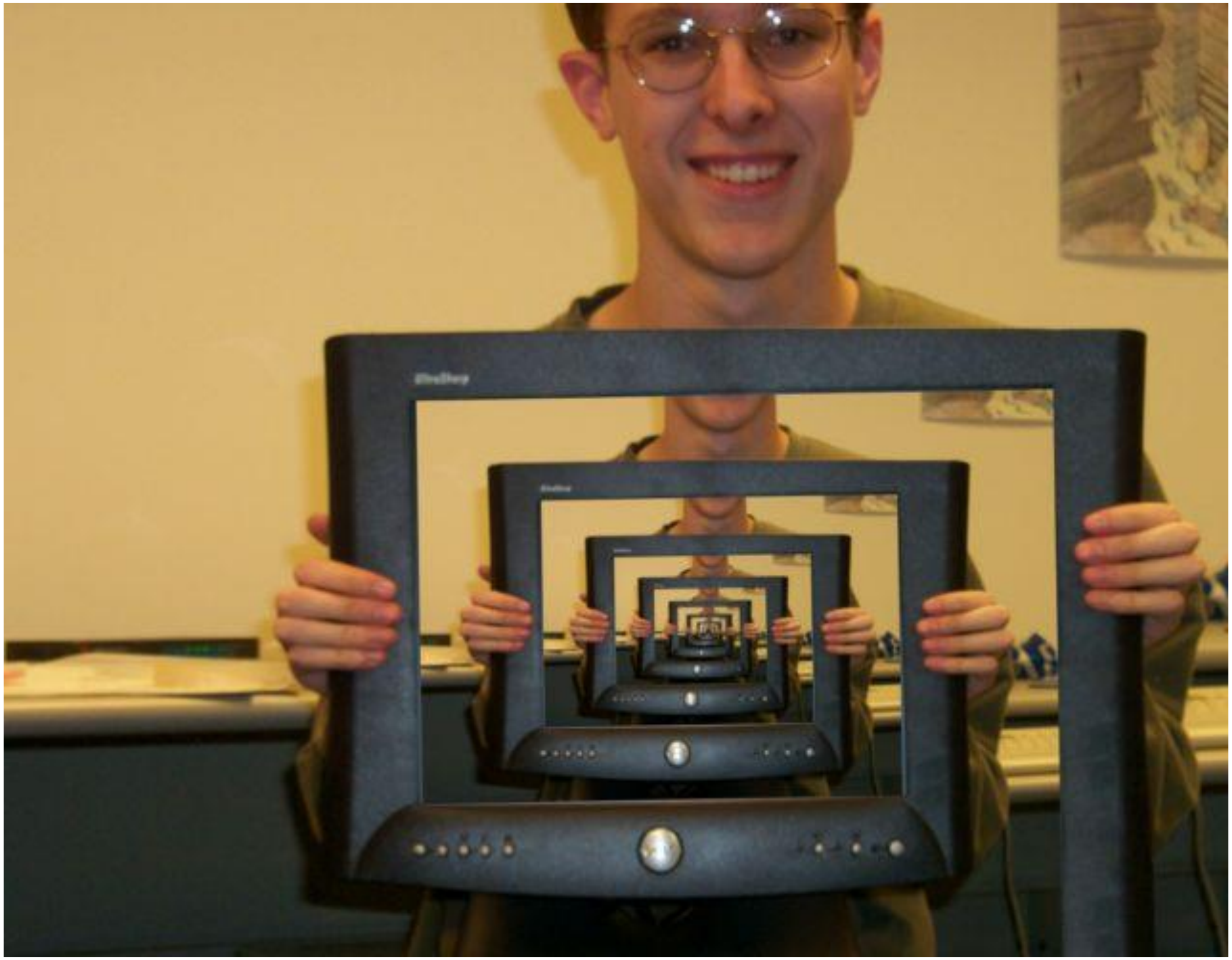
Program Studi Informatika
Sekolah Teknik Elektro dan Informatika (STEI)
ITB

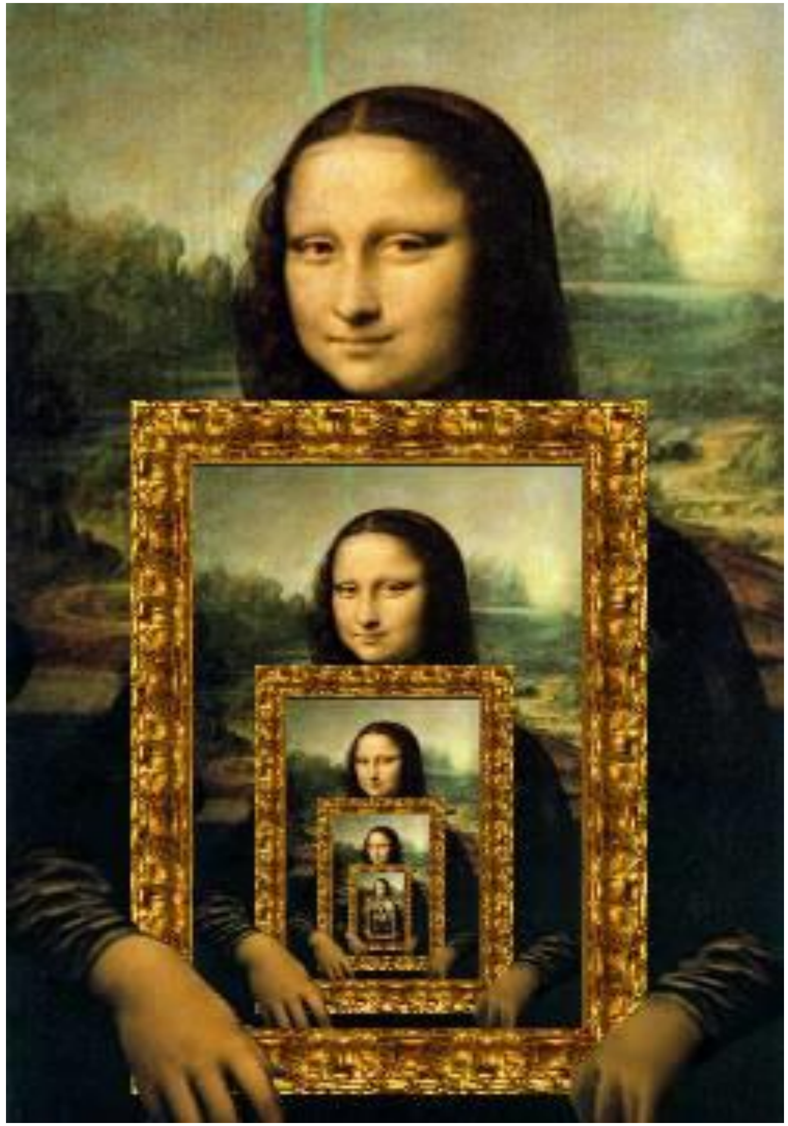
Rekursi

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

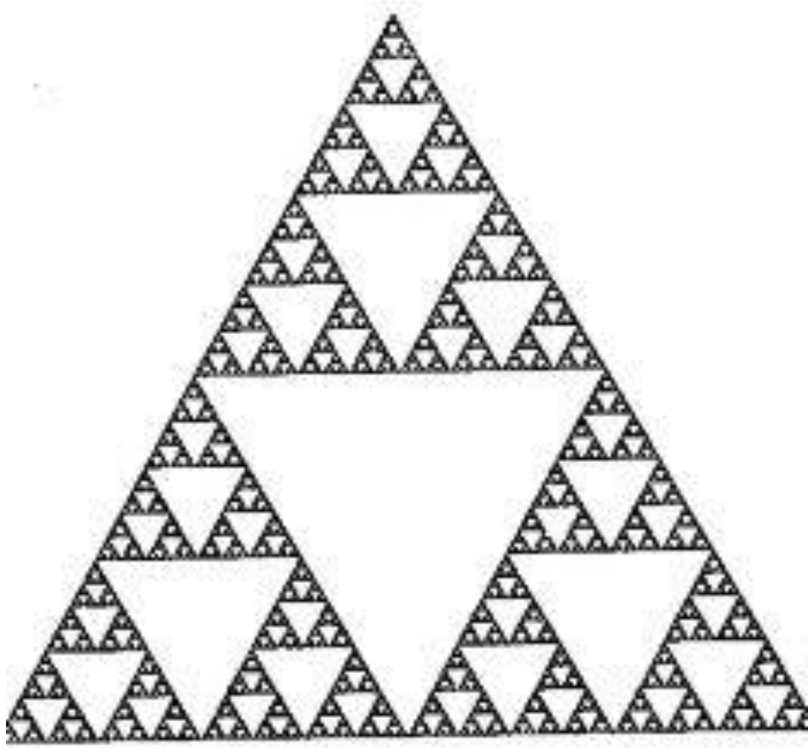
- Proses mendefinisikan objek dalam terminologi dirinya sendiri disebut **rekursi** (*recursion*).

- 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

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

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**)$\rightarrow$**integer**
{ mengembalikan nilai n!;
basis : jika n = 0, maka 0! = 1
rekurens: jika n > 0, maka n! = n $\times$ (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:

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

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

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

$$= b + (a-1)b$$

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

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: Σ

  Contoh: Σ = {0, 1}, Σ = {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

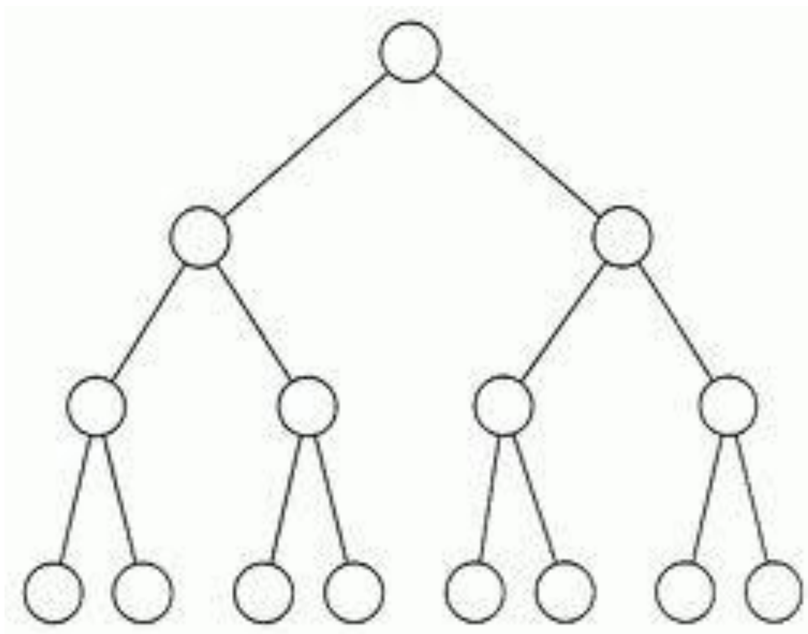
- Panjang string (disimbolkan dengan L) dapat didefinisikan secara rekursif:

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

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

Struktur Rekursif

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



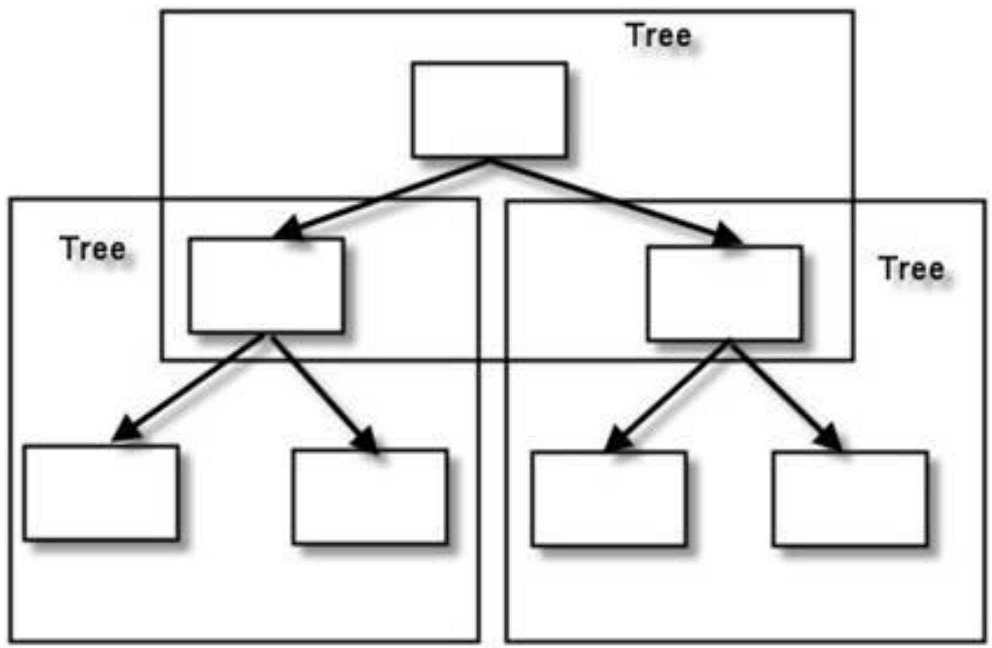
- Simpul (*node*) pada pohon biner mempunyai paling banyak dua buah anak.

- 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*).

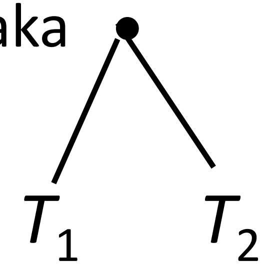


Binary tree consisting of 3 binary trees

- 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

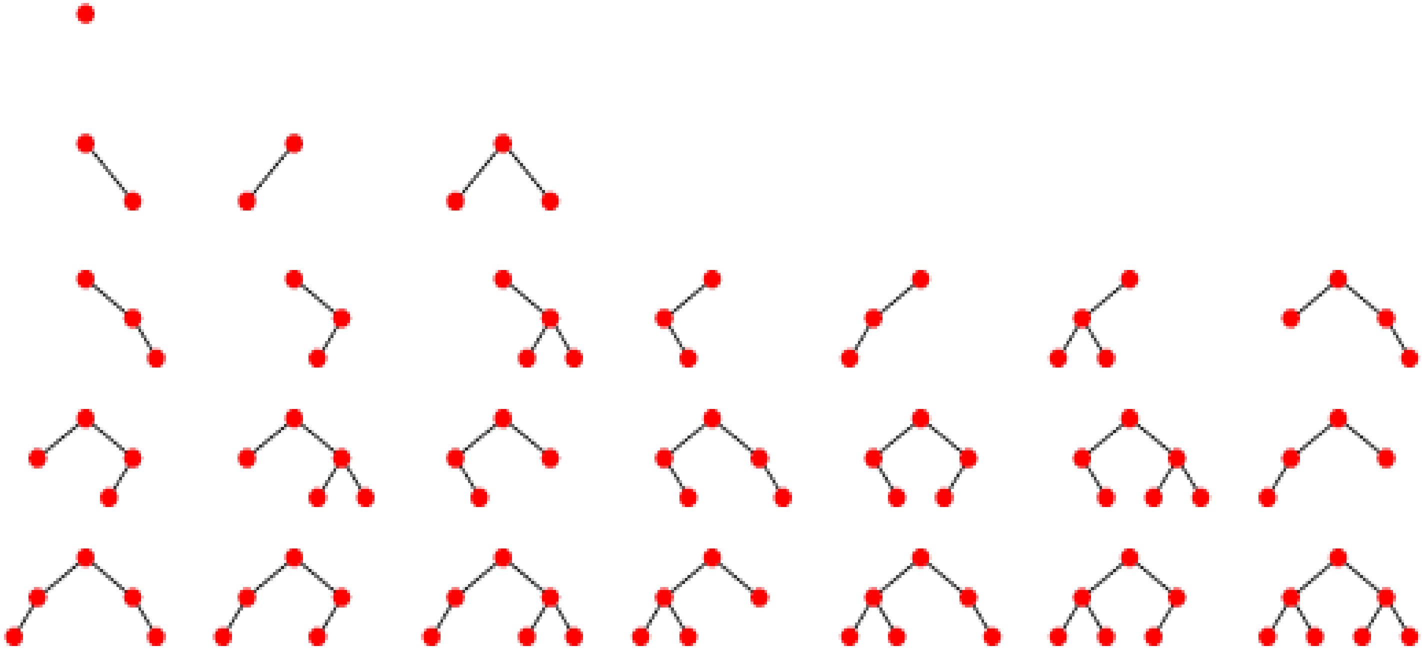


adalah pohon biner

Proses pembentukan pohon biner secara rekursif:

(i) ϕ

(ii)



Barisan Rekursif

- Perhatikan barisan bilangan berikut ini:

  1, 2, 4, 8, 16, 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

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 8**. 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 9**: Misalkan $\{a_n\}$ adalah barisan yang memenuhi relasi rekurens berikut:

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

  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

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

  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.

Pemodelan dengan Relasi Rekurens

1. **Bunga majemuk.**

Contoh 10. Misalkan uang sebanyak Rp10.000 disimpan di bank dengan sistem bunga berbunga dengan besar bunga 11% per tahun. Berapa banyak uang setelah 30 tahun?

Misalkan P_n menyatakan nilai uang setalah *n* tahun. Nilai uang setelah *n* tahun sama dengan nilai uang tahun sebelumnya ditambah dengan bunga uang:

$$P_n = P_{n-1} + 0{,}11\, P_{n-1} \; ; P_0 = 10.000$$

- Solusi relasi rekurens $P_n = P_{n-1} + 0{,}11\ P_{n-1}$; $P_0 = 10.000$ dapat dipecahkan sebagai berikut:

$$
\begin{aligned}
P_n &= P_{n-1} + 0{,}11\ P_{n-1} = (1{,}11)\ P_{n-1} \\
&= (1{,}11)\ [(1{,}11)P_{n-2}] = (1{,}11)^2 P_{n-2} \\
&= (1{,}11)^2\ [(1{,}11)\ P_{n-3}] = (1{,}11)^3 P_{n-3} \\
&= \ldots \\
&= (1{,}11)^n P_0
\end{aligned}
$$

Jadi, $P_n = (1{,}11)^n\ P_0 = 10.000\ (1{,}11)^n$

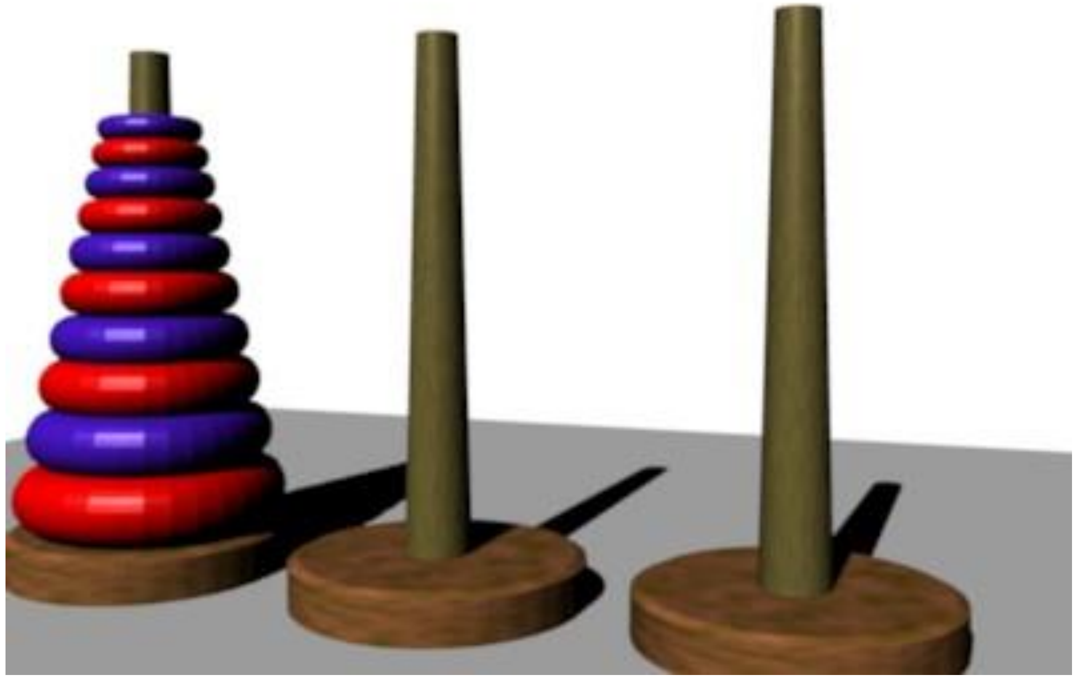
Setelah 30 tahun, banyaknya uang adalah

$$P_{30} = 10.000\ (1{,}11)^{30} = \text{Rp}228.922{,}97$$

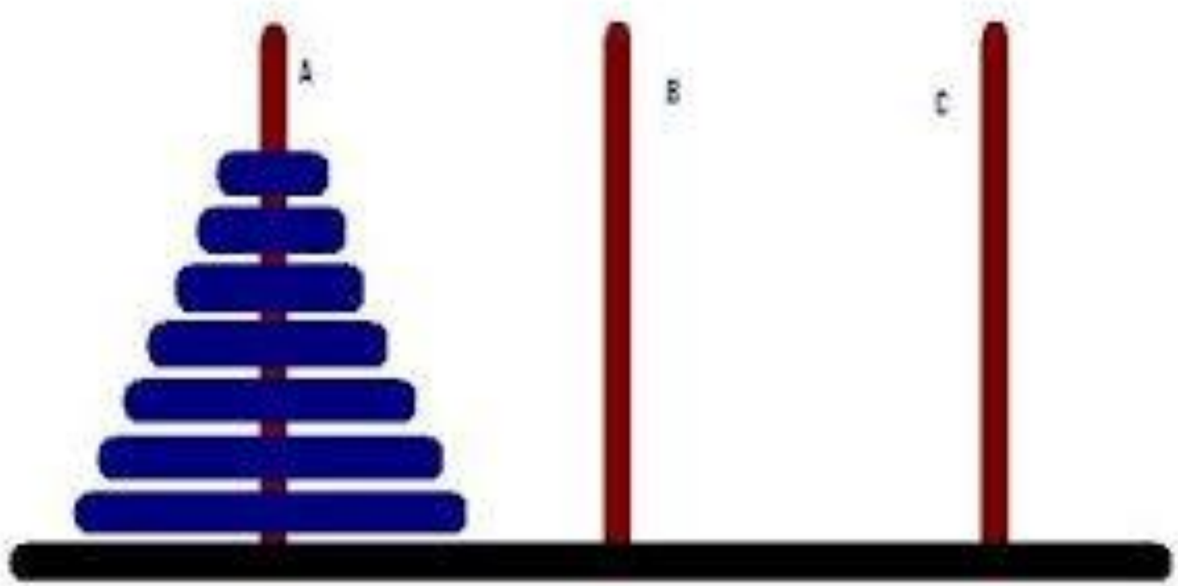
2. Menara Hanoi (*The Tower of Hanoi*)

Contoh 11. Menara Hanoi adalah sebuah *puzzle* yang terkenal pada akhir abad 19. Puzzle ini ditemukan oleh matematikawan Perancis, Edouard Lucas.

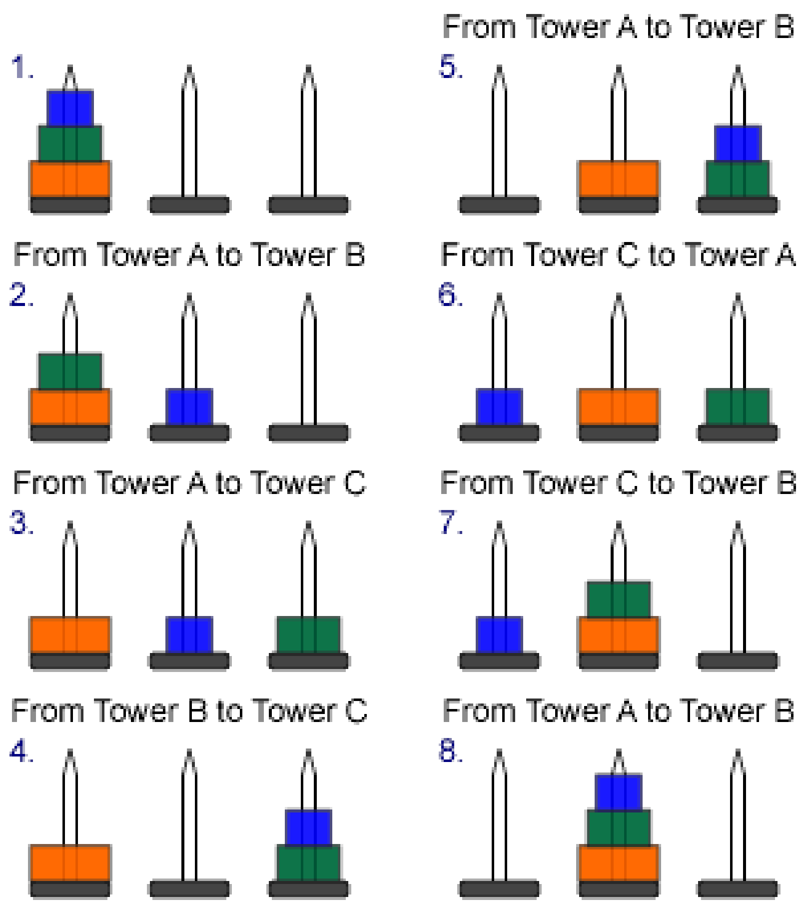
Dikisahkan bahwa di kota Hanoi, Vietnam, terdapat tiga buah tiang tegak setinggi 5 meter dan 64 buah piringan (*disk*) dari berbagai ukuran. Tiap piringan mempunyai lubang di tengahnya yang memungkinkannya untuk dimasukkan ke dalam tiang. Pada mulanya piringan tersebut tersusun pada sebuah tiang sedemikian rupa sehingga piringan yang di bawah mempunyai ukuran lebih besar daripada ukuran piringan di atasnya. Pendeta Budha memberi pertanyaan kepada murid-muridnyanya: bagaimana memindahkan seluruh piringan tersebut ke sebuah tiang yang lain; setiap kali hanya satu piringan yang boleh dipindahkan, tetapi tidak boleh ada piringan besar di atas piringan kecil. Tiang yang satu lagi dapat dipakai sebagai tempat peralihan dengan tetap memegang aturan yang telah disebutkan. Menurut legenda pendeta Budha, bila pemindahan seluruh piringan itu berhasil dilakukan, maka dunia akan kiamat!



Pemodelan:



- Kasus untuk n = 3 piringan



- Secara umum, untuk *n* piringan, penyelesaian dengan cara berpikir rekursif adalah sebagai berikut:

  Kita harus memindahkan piringan paling bawah terlebih dahulu ke tiang *B* sebagai alas bagi piringan yang lain. Untuk mencapai maksud demikian, berpikirlah secara rekursif: pindahkan *n* – 1 piringan teratas dari *A* ke *C,* lalu pindahkan piringan paling bawah dari *A* ke *B,* lalu pindahkan *n* – 1 piringan dari *C* ke *B*.

  pindahkan n – 1 piringan dari A ke C
  pindahkan 1 piringan terbawah dari A ke B
  pindahkan n – 1 piringan dari C ke B

  Selanjutnya dengan tetap berpikir rekursif-pekerjaan memindahkan *n* – 1 piringan dari sebuah tiang ke tiang lain dapat dibayangkan sebagai memindahkan *n* – 2 piringan antara kedua tiang tersebut, lalu memindahkan piringan terbawah dari sebuah tiang ke tiang lain, begitu seterusnya.

- Misalkan H_n menyatakan jumlah perpindahan piringan yang dibutuhkan untuk memecahkan teka-teki Menara Hanoi.

  pindahkan n – 1 piringan dari A ke C → H_{n-1} *kali*
  pindahkan 1 piringan terbawah dari A ke B → *1 kali*
  pindahkan n – 1 piringan dari C ke B → H_{n-1} *kali*

  Maka jumlah perpindahan yang terjadi adalah:

  $$H_n = 2H_{n-1} + 1$$

  dengan kondisi awal $H_1 = 1$

- Penyelesaian relasi rekurens:

$$
\begin{aligned}
H_n &= 2H_{n-1} + 1 \\
&= 2(2H_{n-2} + 1) + 1 = 2^2 H_{n-2} + 2 + 1 \\
&= 2^2 (2H_{n-3} + 1) + 2 + 1 = 2^3 H_{n-3} + 2^2 + 2 + 1 \\
&\quad \ldots \\
&= 2^{n-1} H_1 + 2^{n-2} + 2^{n-3} + \ldots + 2 + 1 \\
&= 2^{n-1} + 2^{n-2} + 2^{n-3} + \ldots + 2 + 1 \quad \rightarrow \text{deret geometri} \\
&= 2^n - 1
\end{aligned}
$$

- Untuk n = 64 piringan, jumlah perpindahan piringan yang terjadi adalah

$$H_{64} = 2^{64} - 1 = 18.446.744.073.709.551.615$$

- Jika satu kali pemindahan piringan membutuhkan waktu 1 detik, maka waktu yang diperlukan adalah

  18.446.744.073.709.551.615 detik

  atau setara dengan 584.942.417.355 tahun atau sekitar 584 milyar tahun!

- Karena itu, legenda yang menyatakan bahwa dunia akan kiamat bila orang berhasil memindahkan 64 piringan di menara Hanoi ada juga benarnya, karena 584 milyar tahun tahun adalah waktu yang sangat lama, dunia semakin tua, dan akhirnya hancur. *Wallahualam*

Penyelesaian Relasi Rekurens

- Relasi rekurens dapat diselesaikan secara iteratif atau dengan metode yang sistematis.

- Secara iteratif misalnya pada contoh bunga majemuk (Contoh 10) dan Menara Hanoi (Contoh 11).

- Secara sistematis adalah untuk relasi rekurens yang berbentuk homogen lanjar (*linear homogeneous*).

- Relasi rekurens dikatakan homogen lanjar jika berbentuk

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k}$$

  yang dalam hal ini $c_1, c_2, \ldots, c_k$ adalah bilangan riil dan $c_k \neq 0$.

- **Contoh 12**. $P_n = (1,11)\, P_{n-1}$ → homogen lanjar

  $f_n = f_{n-1} + f_{n-2}$ → homogen lanjar

  $a_n = 2a_{n-1} - a^2_{n-2}$ → tidak homogen lanjar

  $H_n = 2H_{n-1} - 1$ → tidak homogen lanjar

  $a_n = na_{n-1}$ → tidak homogen lanjar

Penjelasan:

$H_n = 2H_{n-1} - 1$ tidak homogen lanjar karena *term* -1 tidak dikali dengan nilai H_j untuk sembarang j

$a_n = na_{n-1}$ tidak homogen lanjar karena koefisiennya bukan konstanta.

- Solusi relasi rekurens yang berbentuk homogen lanjar adalah mencari bentuk

$$a_n = r^n$$

  yang dalam hal ini r adalah konstanta.

- Sulihkan $a_n = r^n$ ke dalam relasi rekuren homugen lanjar:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k}$$

  menjadi

$$r^n = c_1 r^{n-1} + c_2 r^{n-2} + \ldots + c_k r^{n-k}$$

- Bagi kedua ruas dengan r^{n-k} , menghasilkan

$$r^k - c_1 r^{k-1} - c_2 r^{k-2} - \ldots - c_{k-1} r - c_k = 0$$

- Persamaan di atas dinamakan **persamaan karakteristik** dari relasi rekurens.

- Solusi persamaan karakteristik disebut **akar-akar karakteristik**, dan merupakan komponen solusi relasi rekurens yang kita cari ($a_n = r^n$).

- Untuk relasi rekurens homogen lanjar derajat k = 2,

$$a_n = c_1 a_{n-1} + c_2 a_{n-2}$$

persamaan karakteristiknya berbentuk:

$$r^2 - c_1 r - c_2 = 0$$

- Akar persamaan karakteristik adalah r_1 dan r_2.

- **Teorema 1**: Barisan $\{a_n\}$ adalah solusi relasi rekurens $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ jika dan hanya jika $a_n = \alpha_1 r^n{}_1 + \alpha_2 r^n{}_2$ untuk n = 0, 1, 2, … dengan α_1 dan α_2 adalah konstan.

- **Contoh 13**. Tentukan solusi relasi rekurens berikut:

$$a_n = a_{n-1} + 2a_{n-2} \;;\; a_0 = 2 \text{ dan } a_1 = 7?$$

Penyelesaian:

Persamaan karakteristik: $r^2 - r - 2 = 0$.

Akar-akarnya: $(r - 2)(r + 1) = 0 \rightarrow r_1 = 2$ dan $r_2 = -1$

$a_n = \alpha_1 r^n_1 + \alpha_2 r^n_2 \rightarrow a_n = \alpha_1 2^n + \alpha_2(-1)^n$

$a_0 = 2 \rightarrow a_0 = 2 = \alpha_1 2^0 + \alpha_2(-1)^0 = \alpha_1 + \alpha_2$

$a_1 = 7 \rightarrow a_1 = 7 = \alpha_1 2^1 + \alpha_2(-1)^1 = \alpha_1 - \alpha_2$

Diperoleh dua persamaan: $\alpha_1 + \alpha_2 = 2$ dan $\alpha_1 - \alpha_2 = 7$, solusinya adalah $\alpha_1 = 3$ dan $\alpha_2 = -1$

Jadi, solusi relasi rekurens adalah:

$a_n = 3 \cdot 2^n - (-1)^n$

- Jika persamaan karakteristik memiliki dua akar yang sama (akar kembar, $r_1 = r_2$), maka Teorema 1 tidak dapat dipakai. Terapkan Teorema 2 berikut ini.

- **Teorema 2**: Misalkan $r^2 - c_1 r - c_2 = 0$ mempunyai akar kembar r_0. Barisan $\{a_n\}$ adalah solusi relasi rekurens $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ jika dan hanya jika $a_n = \alpha_1 r^n{}_0 + \alpha_2 n r^n{}_0$ untuk $n = 0, 1, 2, \ldots$ dengan α_1 dan α_2 adalah konstan.

- **Contoh 14**. Tentukan solusi relasi rekurens berikut:

$$a_n = 6a_{n-1} - 9a_{n-2} \;;\; a_0 = 1 \text{ dan } a_1 = 6?$$

Penyelesaian:

Penyelesaian:

Persamaan karakteristik: $r^2 - 6r + 9 = 0$.

Akar-akarnya: $(r - 3)(r - 3) = 0 \rightarrow r_1 = r_2 = 3 \rightarrow r_0$

$a_n = \alpha_1 r^n{}_0 + \alpha_2 n r^n{}_0 \rightarrow a_n = \alpha_1 3^n + \alpha_2 n 3^n$

$a_0 = 1 \rightarrow a_0 = 1 = \alpha_1 3^0 + \alpha_2 \cdot 0 \cdot 3^0 = \alpha_1$

$a_1 = 6 \rightarrow a_1 = 6 = \alpha_1 3^1 + \alpha_2 \cdot 1 \cdot 3^1 = 3\alpha_1 + 3\alpha_2$

Diperoleh dua persamaan: $\alpha_1 = 1$ dan $3\alpha_1 + 3\alpha_2 = 6$,

solusinya adalah $\alpha_1 = 1$ dan $\alpha_2 = 1$

Jadi, solusi relasi rekurens adalah:

$a_n = 3^n + n3^n$

- **Latihan**. Selesaikan relasi rekurens berikut:

  (a) $a_n = 2a_{n-1}$; $a_0 = 3$

  (b) $a_n = 5a_{n-1} - 6a_{n-2}$; $a_0 = 1$ dan $a_1 = 0$?

  (c) Barisan Fibonacci: $f_n = f_{n-1} + {}_{fn-2}$

- (UTS 2013) Selesaikan relasi rekurens berikut: $T(n) = 7T(n-1) - 6T(n-2)$; $T(0) = 2$, $T(1) = 7$ (Catatan: T_n ditulis $T(n)$, T_{n-1} ditulis $T(n-1)$, dst).