

Pohon (Bag. 2)

(Update 2026)

Bahan Kuliah

IF1220 Matematika Diskrit

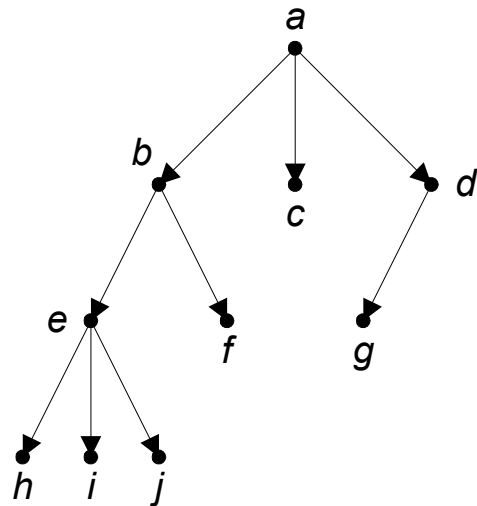
Oleh: Rinaldi M

**Program Studi Teknik Informatika
STEI- ITB**

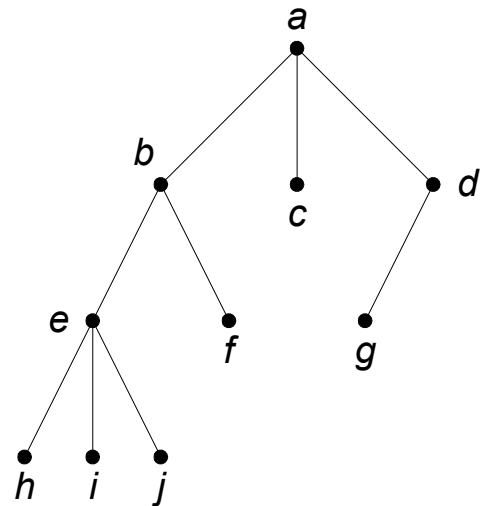


Pohon berakar (*rooted tree*)

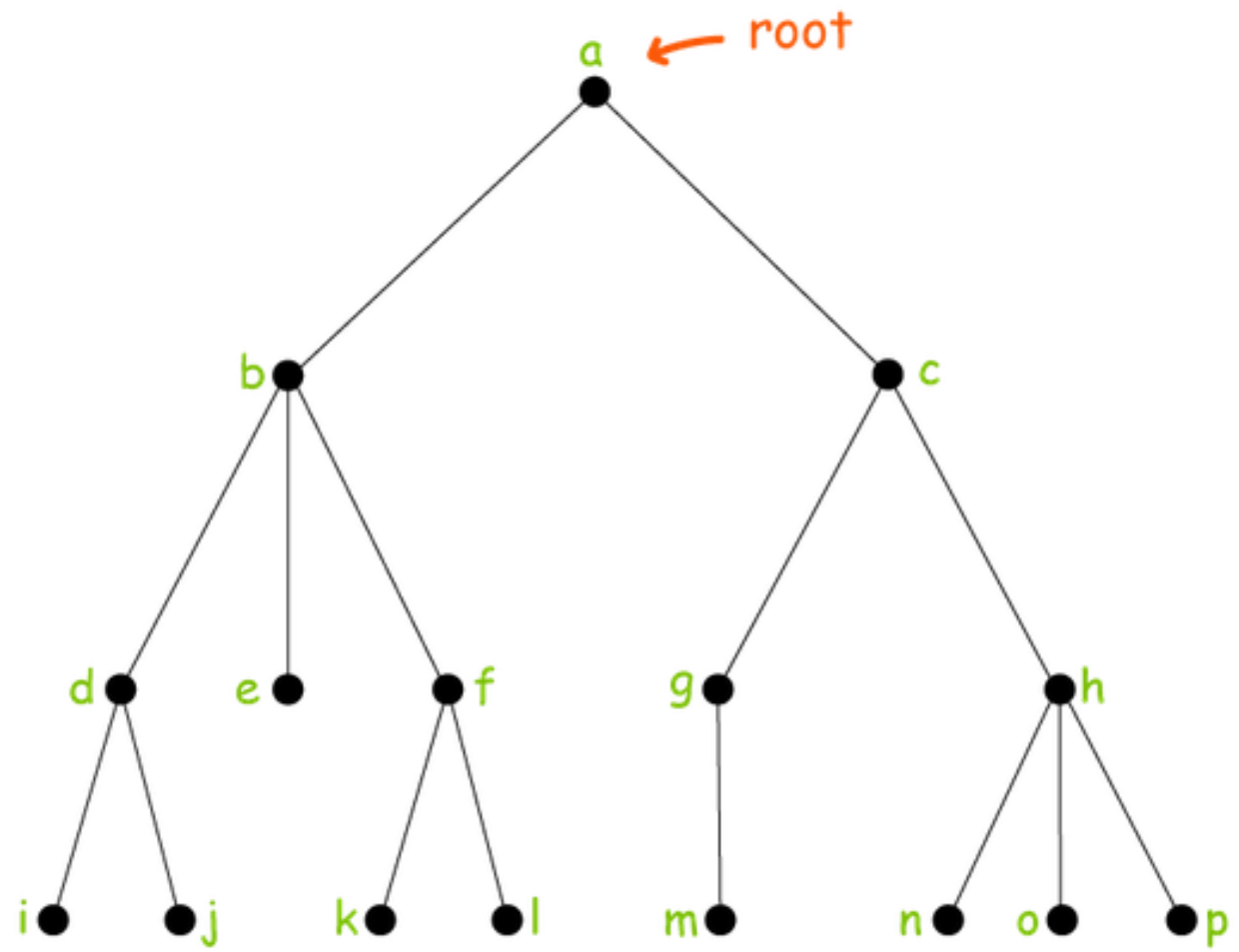
- Pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah dinamakan **pohon berakar** (*rooted tree*).

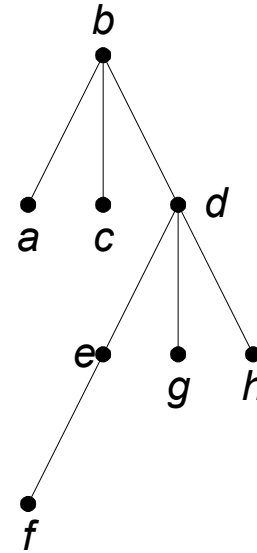
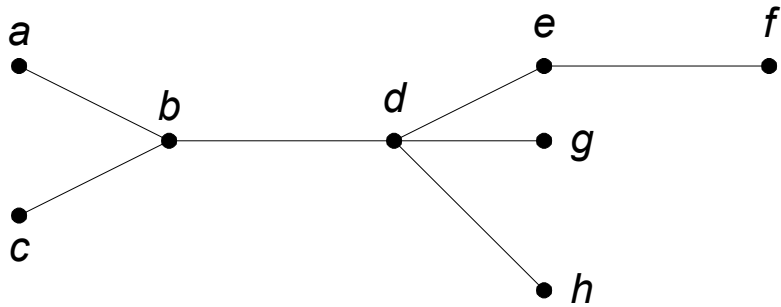


(a) Pohon berakar

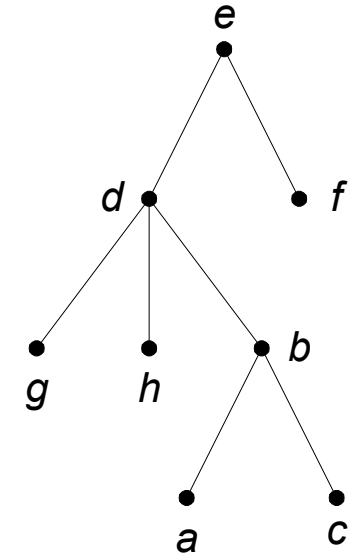


(b) sebagai perjanjian, tanda panah pada sisi dapat dibuang





b sebagai akar



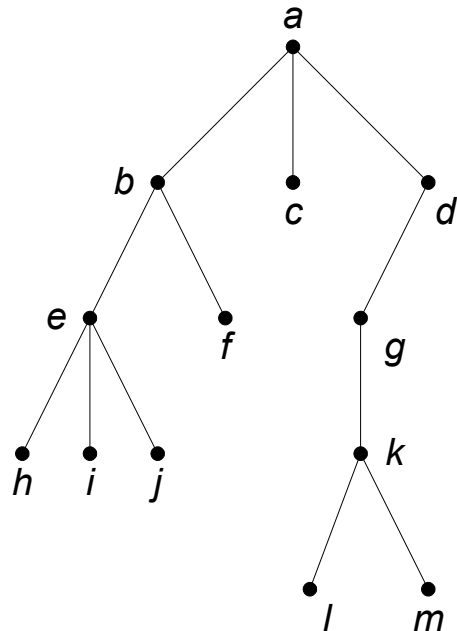
e sebagai akar

Pohon dan dua buah pohon berakar yang dihasilkan dari pemilihan dua simpul berbeda sebagai akar

Terminologi pada Pohon Berakar

Anak (*child* atau *children*) dan Orangtua (*parent*)

b , c , dan d adalah anak-anak simpul a ,
 a adalah orangtua dari anak-anak itu



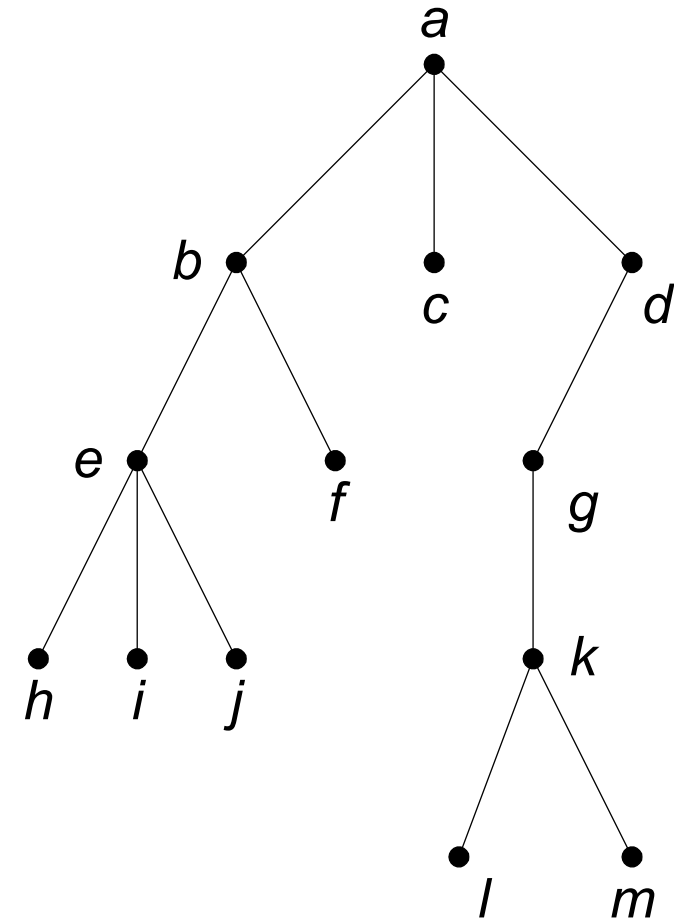
2. Lintasan (*path*)

Lintasan dari a ke j adalah a, b, e, j .

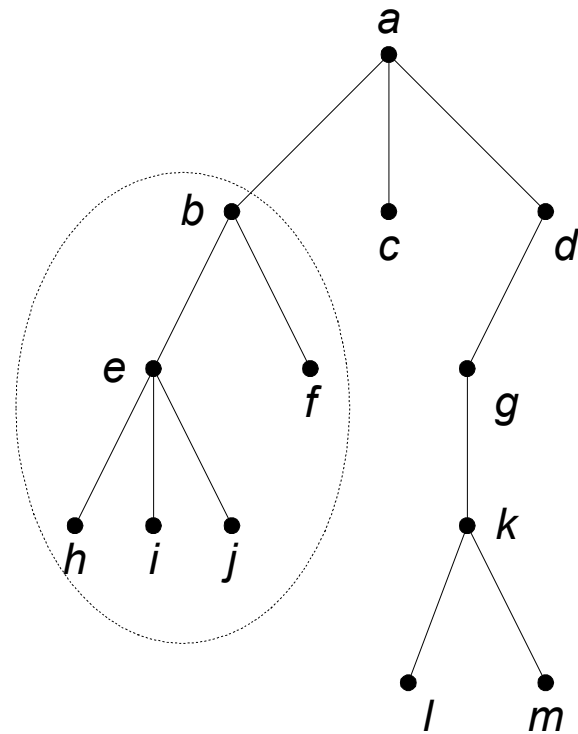
Panjang lintasan dari a ke j adalah 3.

3. Saudara kandung (*sibling*)

f adalah saudara kandung e , tetapi g bukan saudara kandung e , karena orangtua mereka berbeda.



4. Upapohon (*subtree*)



5. Derajat (*degree*)

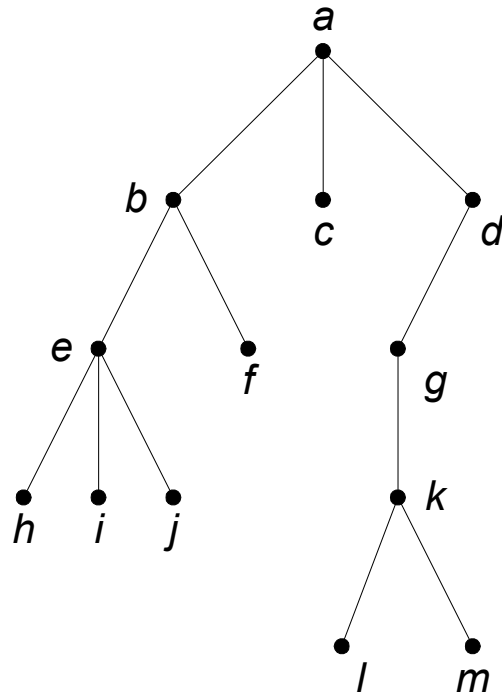
Derajat sebuah simpul adalah jumlah upapohon (atau jumlah anak) pada simpul tersebut.

Derajat a adalah 3, derajat b adalah 2,

Derajat d adalah satu dan derajat c adalah 0.

Jadi, derajat yang dimaksudkan di sini adalah derajat-keluar.

Derajat maksimum dari semua simpul merupakan derajat pohon itu sendiri. Pohon di bawah ini berderajat 3

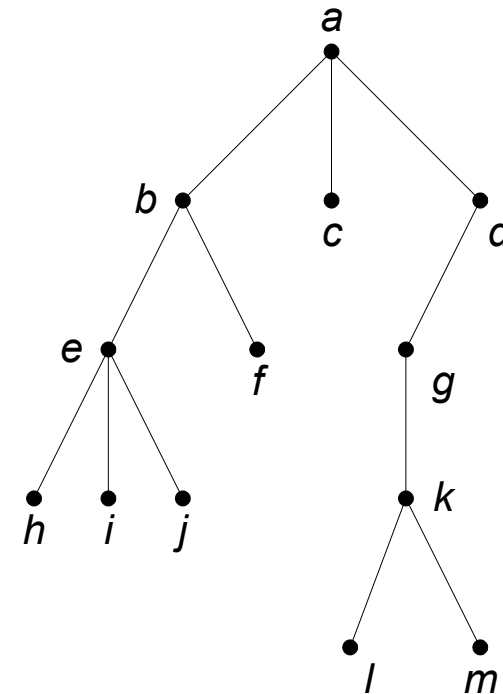


6. Daun (*leaf*)

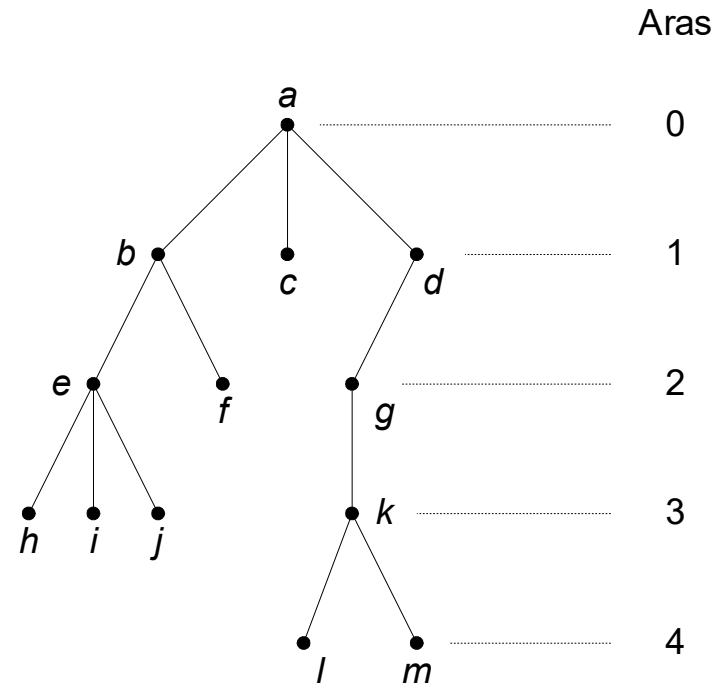
Simpul yang berderajat nol (atau tidak mempunyai anak) disebut **daun**. Simpul $h, i, j, f, c, l,$ dan m adalah daun.

7. Simpul dalam (*internal nodes*)

Simpul yang mempunyai anak disebut **simpul dalam** *atau* **simpul internal**. Simpul $a, b, d, e, g,$ dan k adalah simpul dalam.



8. Aras (*level*) atau Tingkat

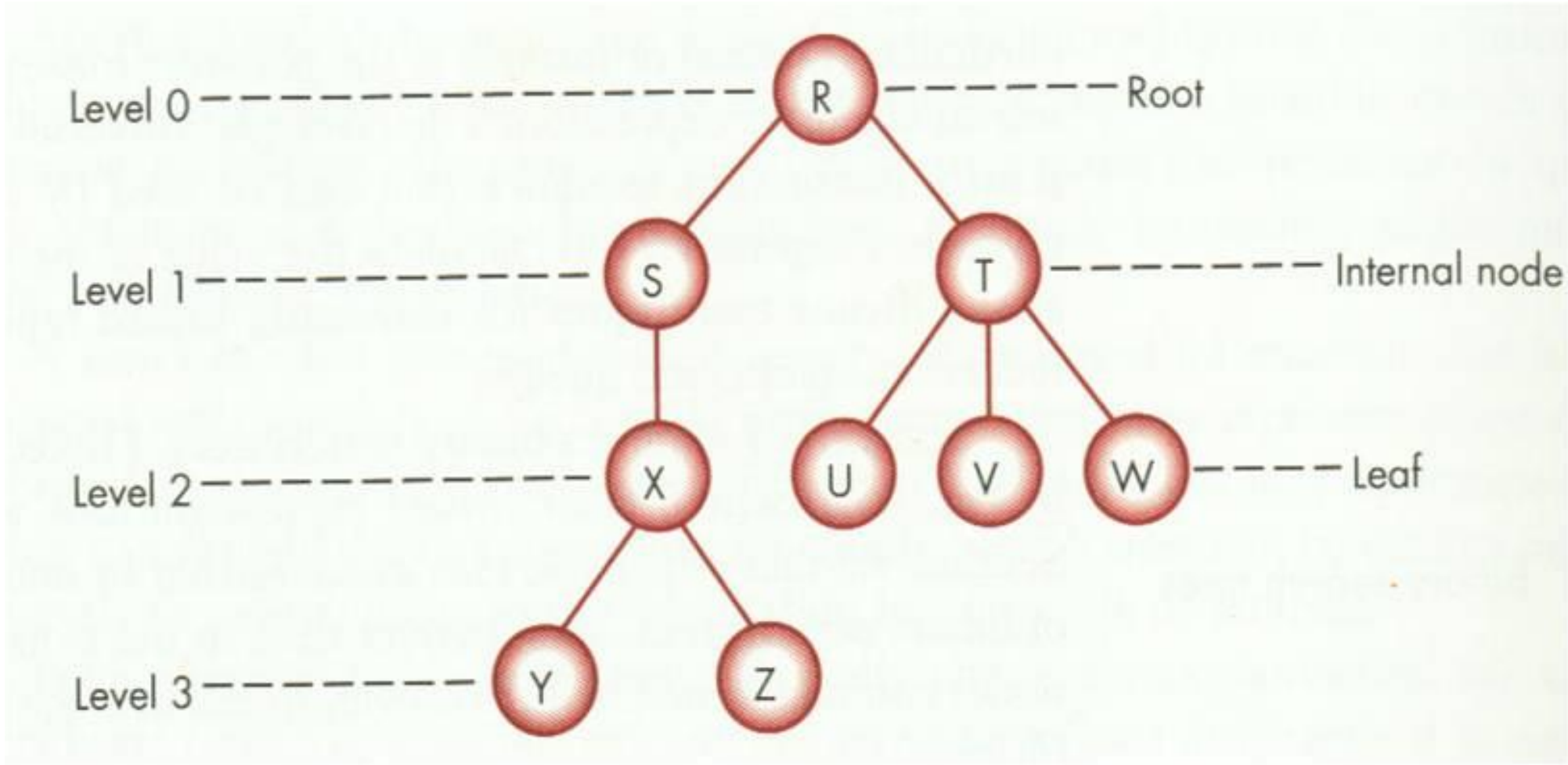


Simpul-simpul pada:

- aras 0: a
- aras 1: b, c, d
- aras 2: e, f, g
- aras 3: h, i, j, k
- aras 4: l, m

9. Tinggi (*height*) atau Kedalaman (*depth*)

Aras maksimum dari suatu pohon disebut **tinggi** atau **kedalaman** pohon tersebut. Pohon di atas mempunyai tinggi 4.



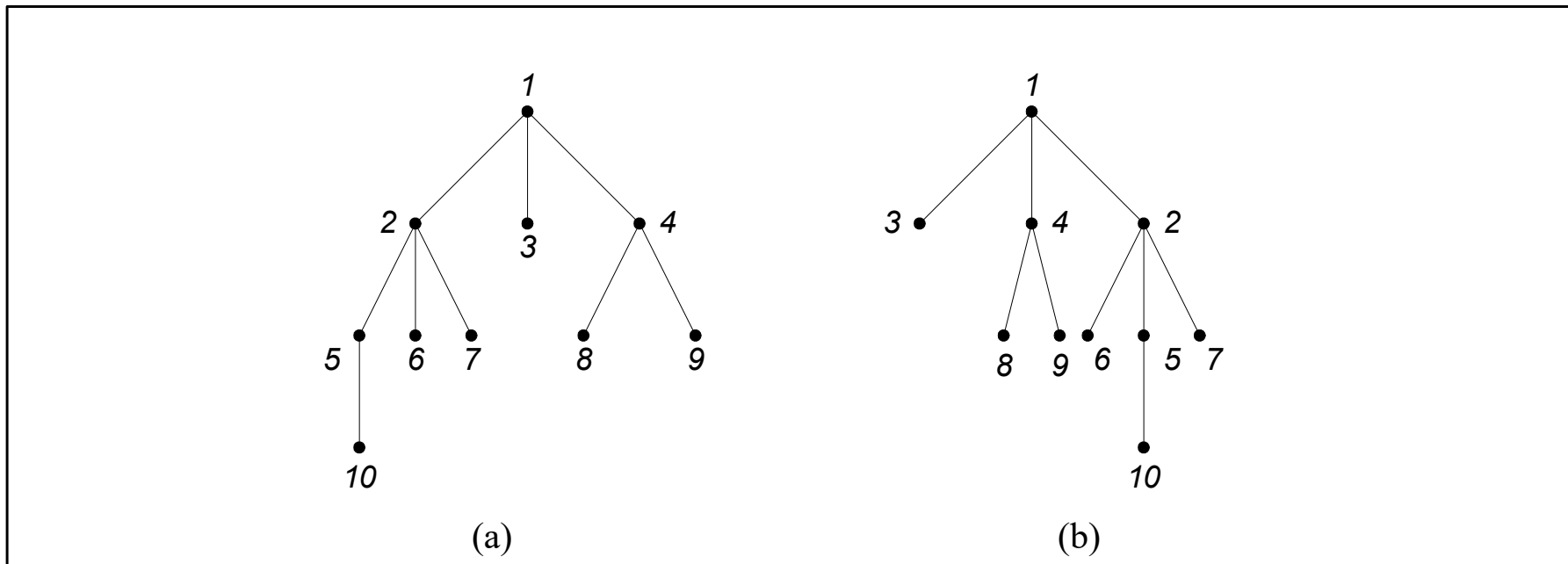
Ketemu juga pohon (*tree*) yang sesuai dengan teori graf. *) 😊

*) Akarnya di atas



Pohon Terurut (*ordered tree*)

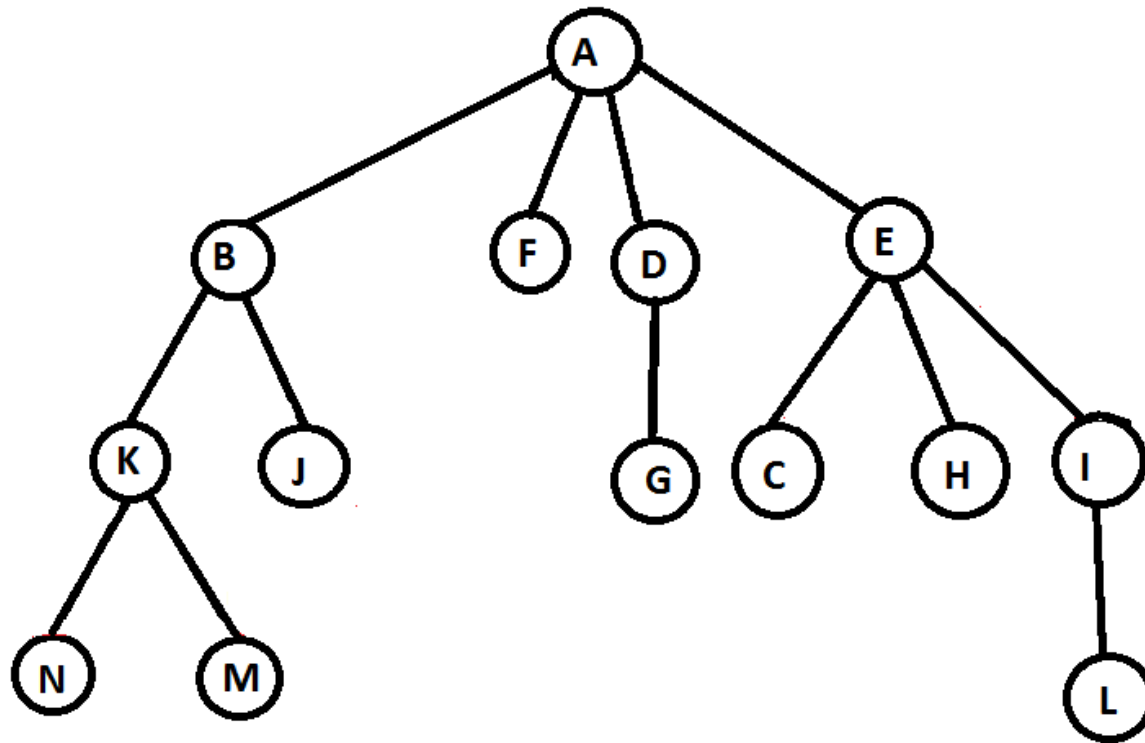
Pohon berakar yang urutan anak-anaknya penting disebut **pohon terurut** (*ordered tree*).

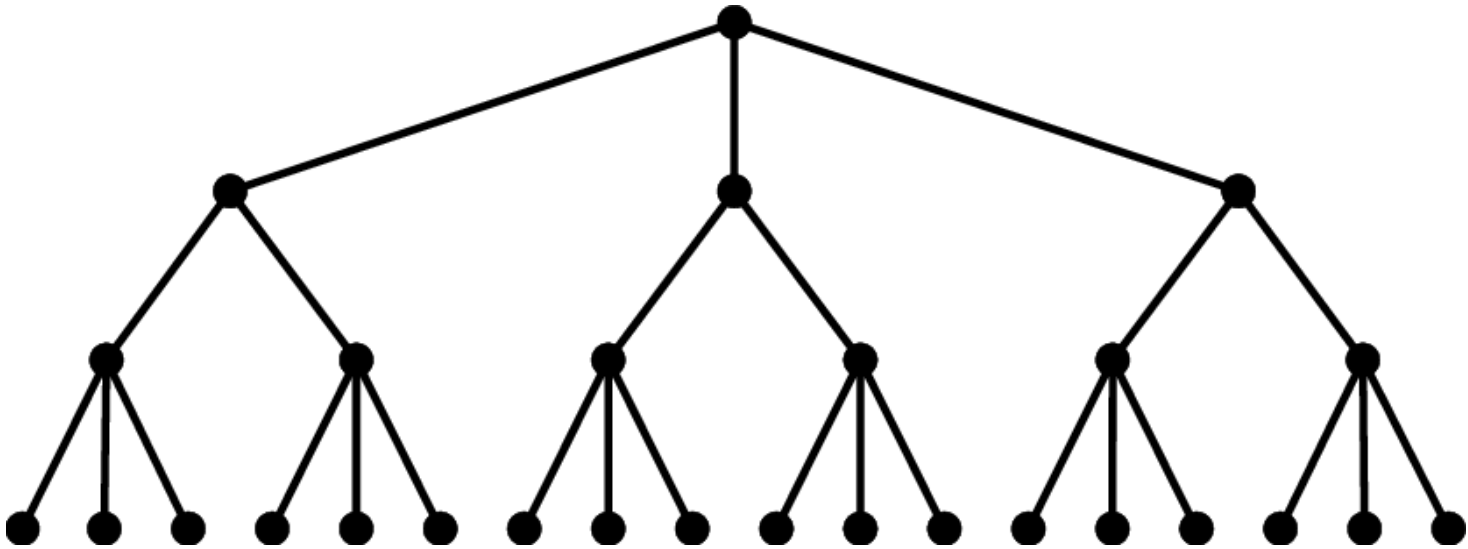


(a) dan (b) adalah dua pohon terurut yang berbeda

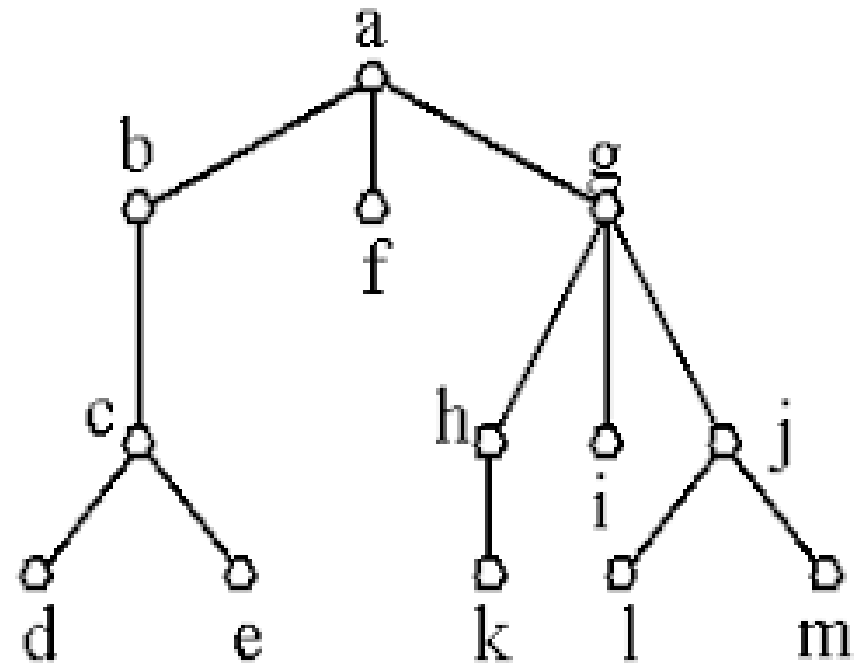
Pohon *m*-ary

- Pohon berakar yang setiap simpul cabangnya mempunyai **paling banyak** m buah anak disebut **pohon *m*-ary**.



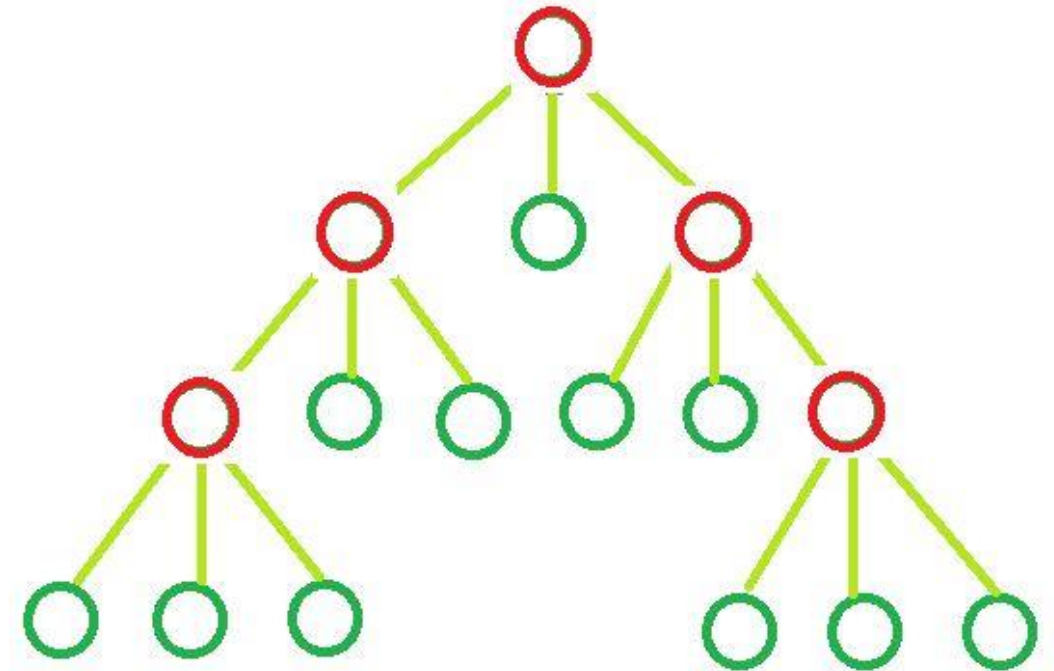
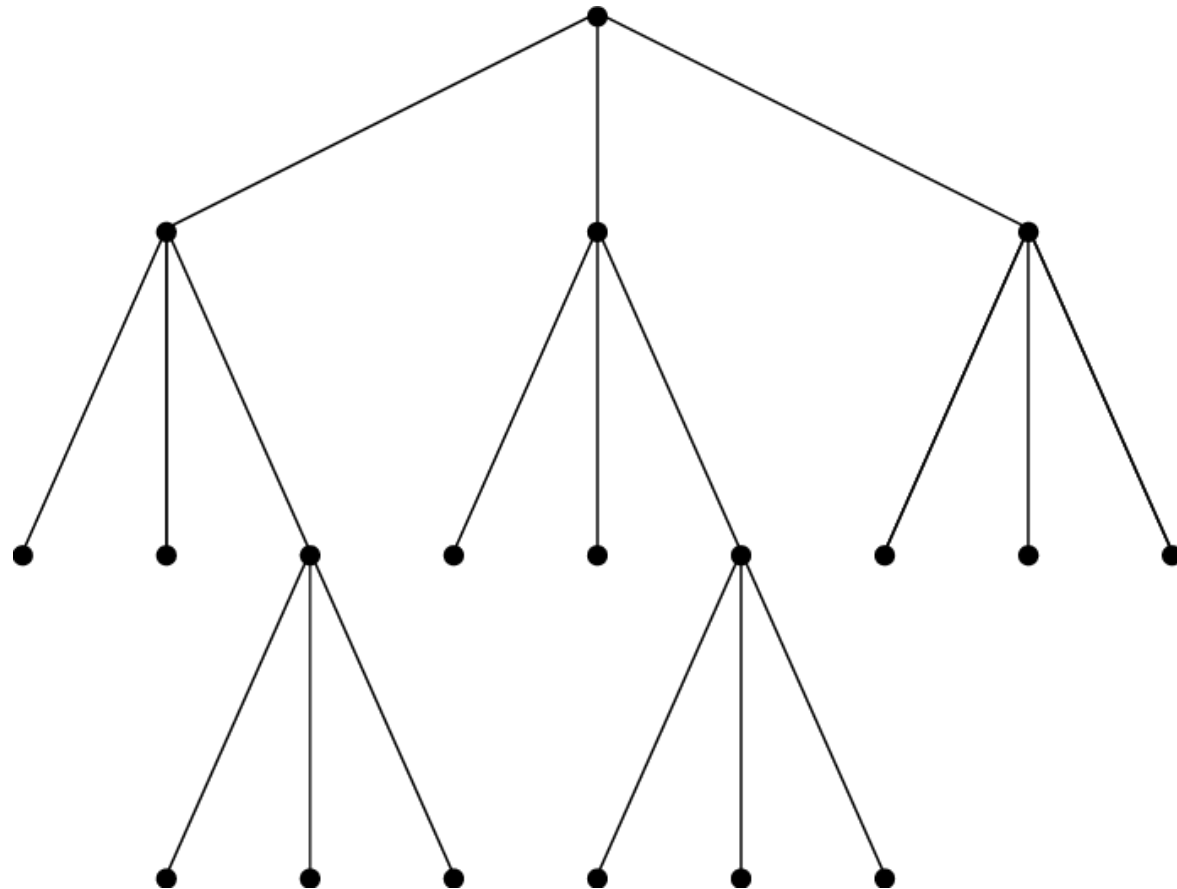


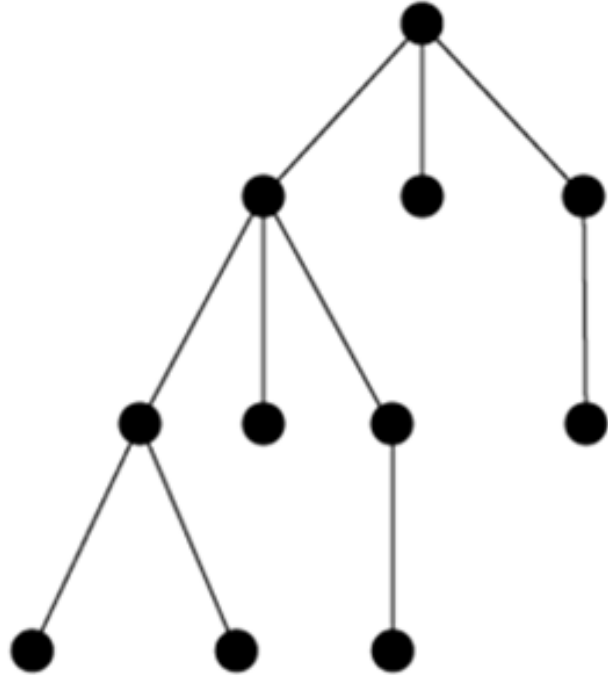
Pohon 3-ary



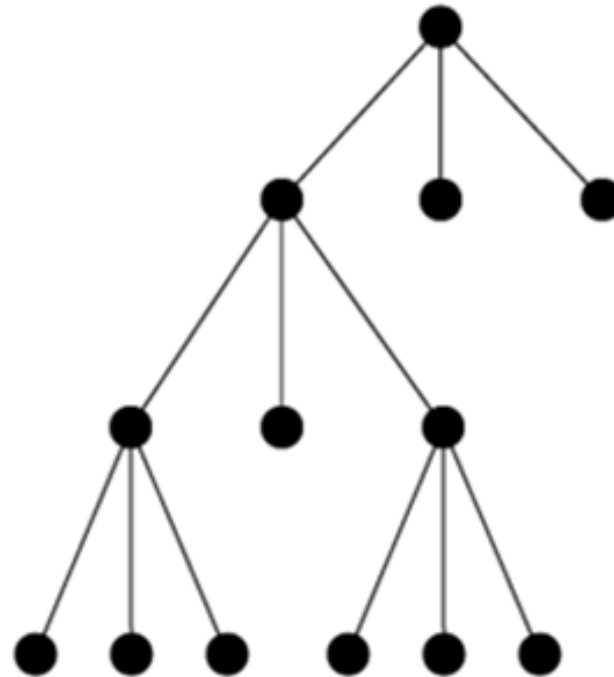
Pohon 3-ary

- Pohon *m*-ary dikatakan **penuhi** (*full m-ary tree*) jika setiap simpul, kecuali simpul pada aras daun, memiliki tepat *m* anak.
- Di bawah ini pohon 3-ary penuh:



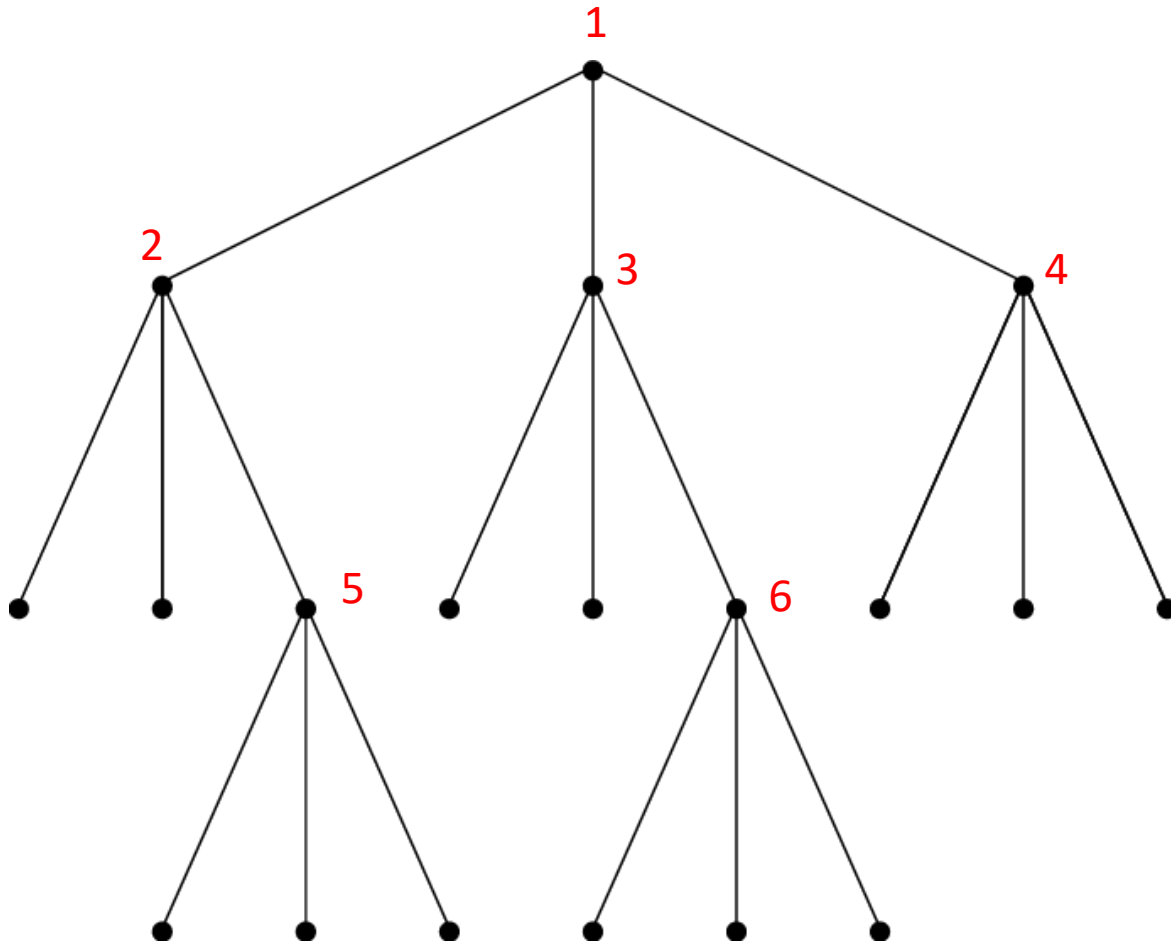


3-ary tree
(each internal vertex has
no more than 3 children)



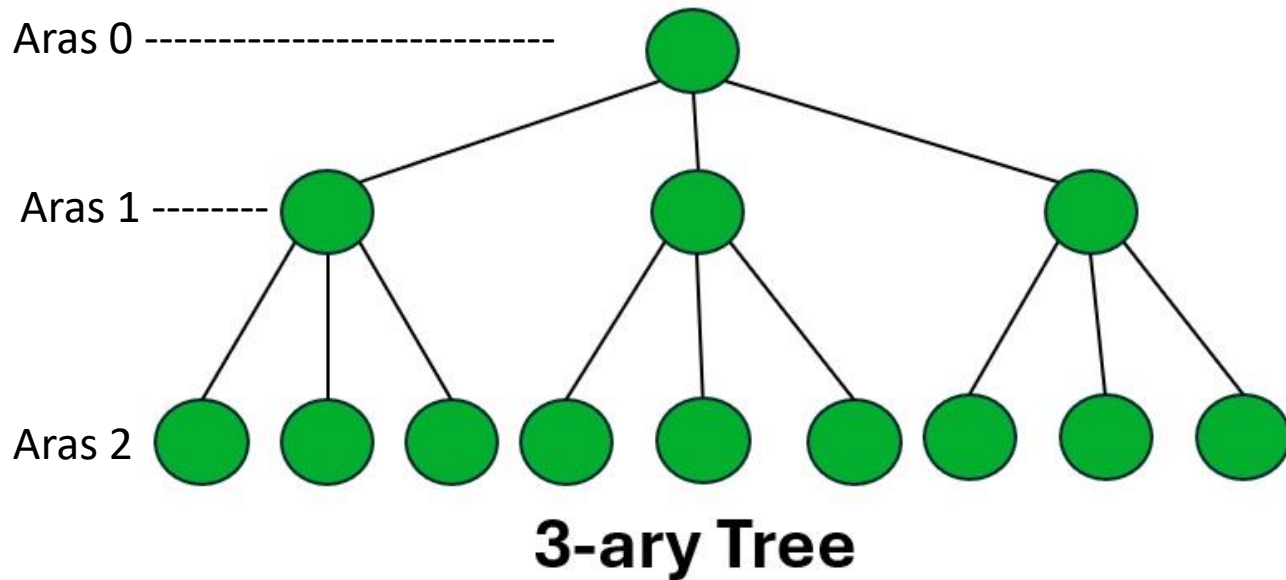
Full 3-ary tree
(each internal vertex
has exactly 3 children)

- Jumlah simpul pada pohon *m*-ary penuh dengan *k* buah simpul internal adalah $n = mk + 1$



Pohon 3-ary penuh di samping ini memiliki $k = 6$ buah simpul internal. Setiap simpul internal memiliki $m = 3$ buah anak. Jumlah simpul seluruhnya adalah:
 $n = (3)(6) + 1 = 19$

- Sebuah pohon m -ary penuh disebut sempurna (*perfect m -ary tree*) jika:
 - setiap simpul memiliki tepat m anak, kecuali simpul daun
 - semua aras terisi penuh
- Jika tinggi pohon m -ary sempurna adalah h , maka jumlah simpul seluruhnya di dalam pohon adalah: $n = m^0 + m^1 + m^2 + \dots + m^h = \sum_{i=0}^h m^i$

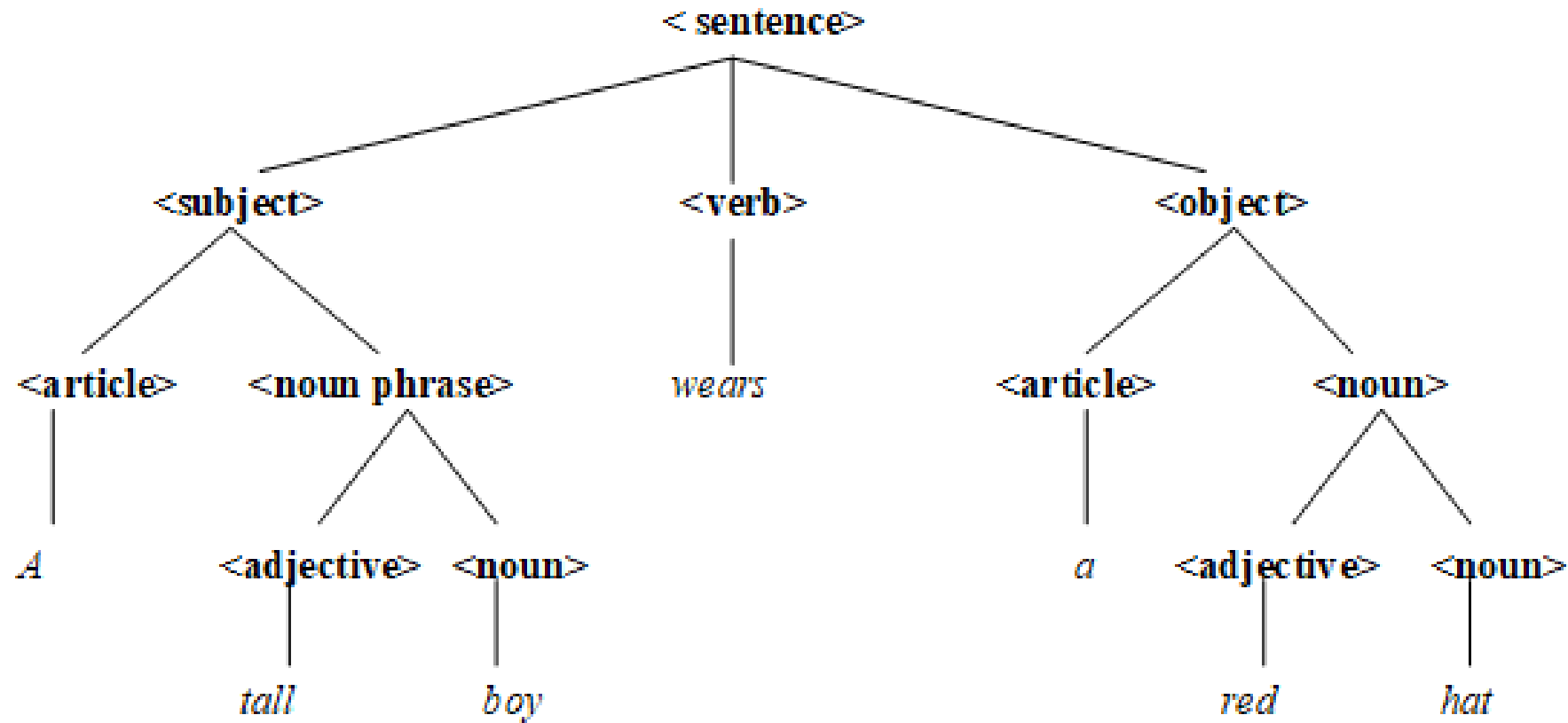


Pohon 3-ary sempurna di samping ini memiliki tinggi $h = 2$.

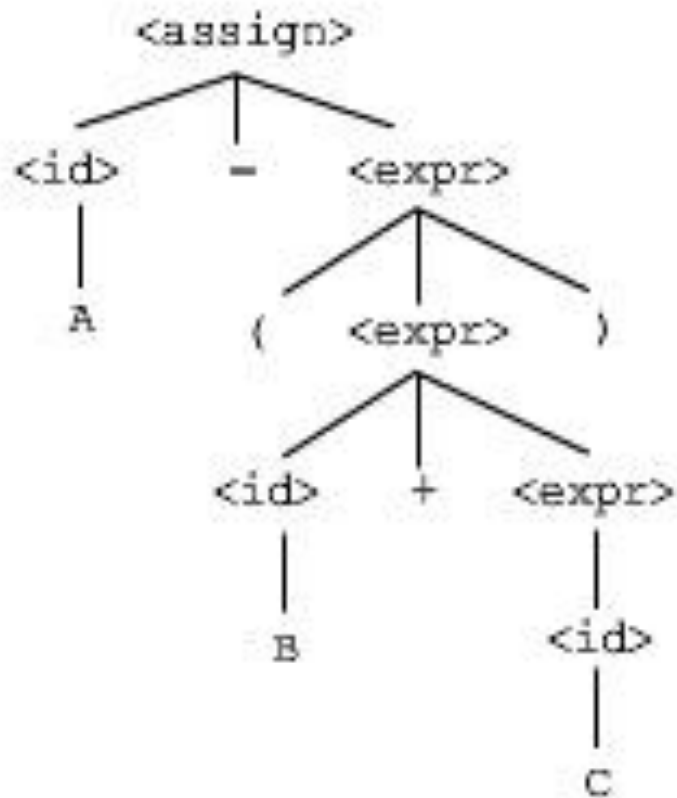
Jumlah simpul seluruhnya adalah:

$$\begin{aligned}
 n &= 3^0 + 3^1 + 3^2 \\
 &= 1 + 3 + 9 \\
 &= 13
 \end{aligned}$$

Salah satu kegunaan pohon n-ary: *parsing* kalimat



Gambar Pohon parsing dari kalimat *A tall boy wears a red hat*



Parse Tree for the Statement A = (B + C)

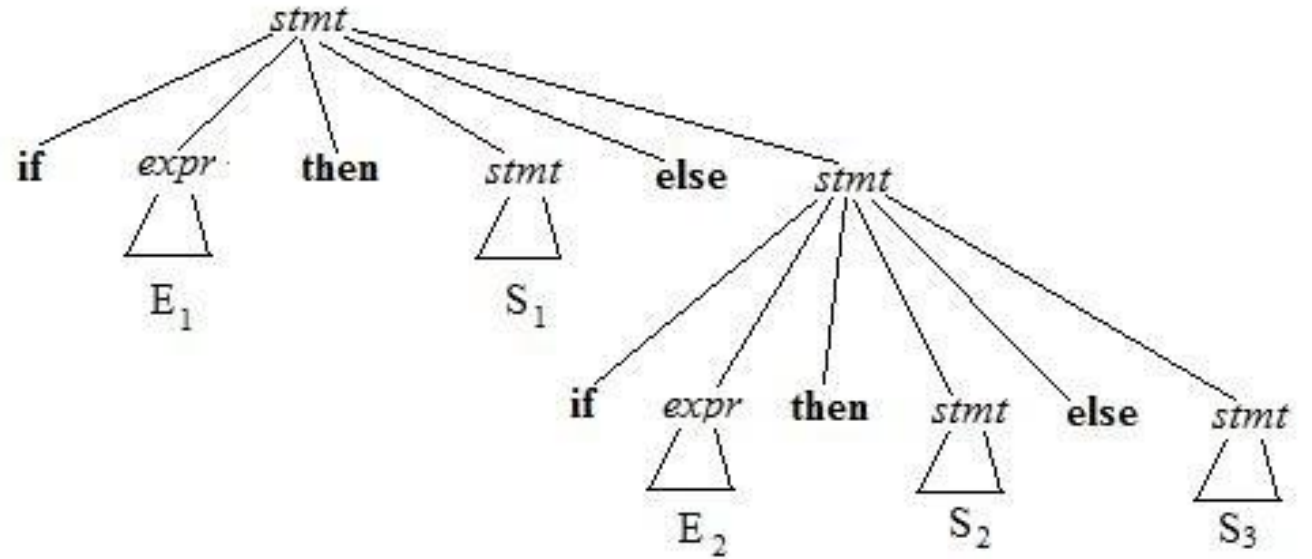
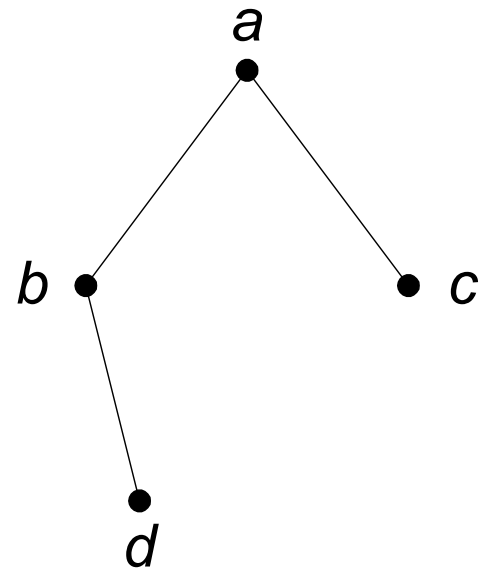
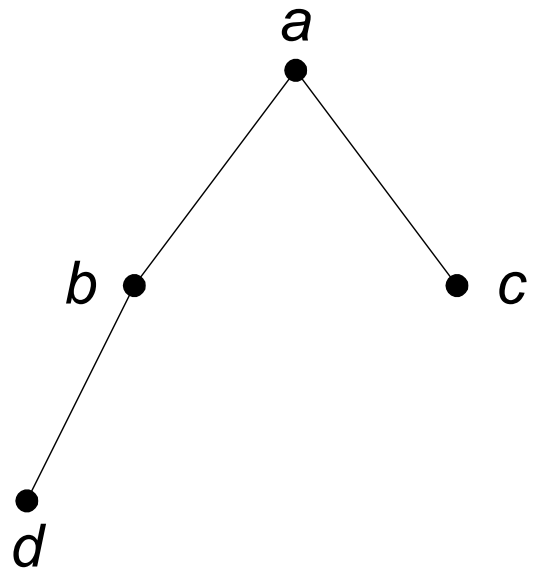


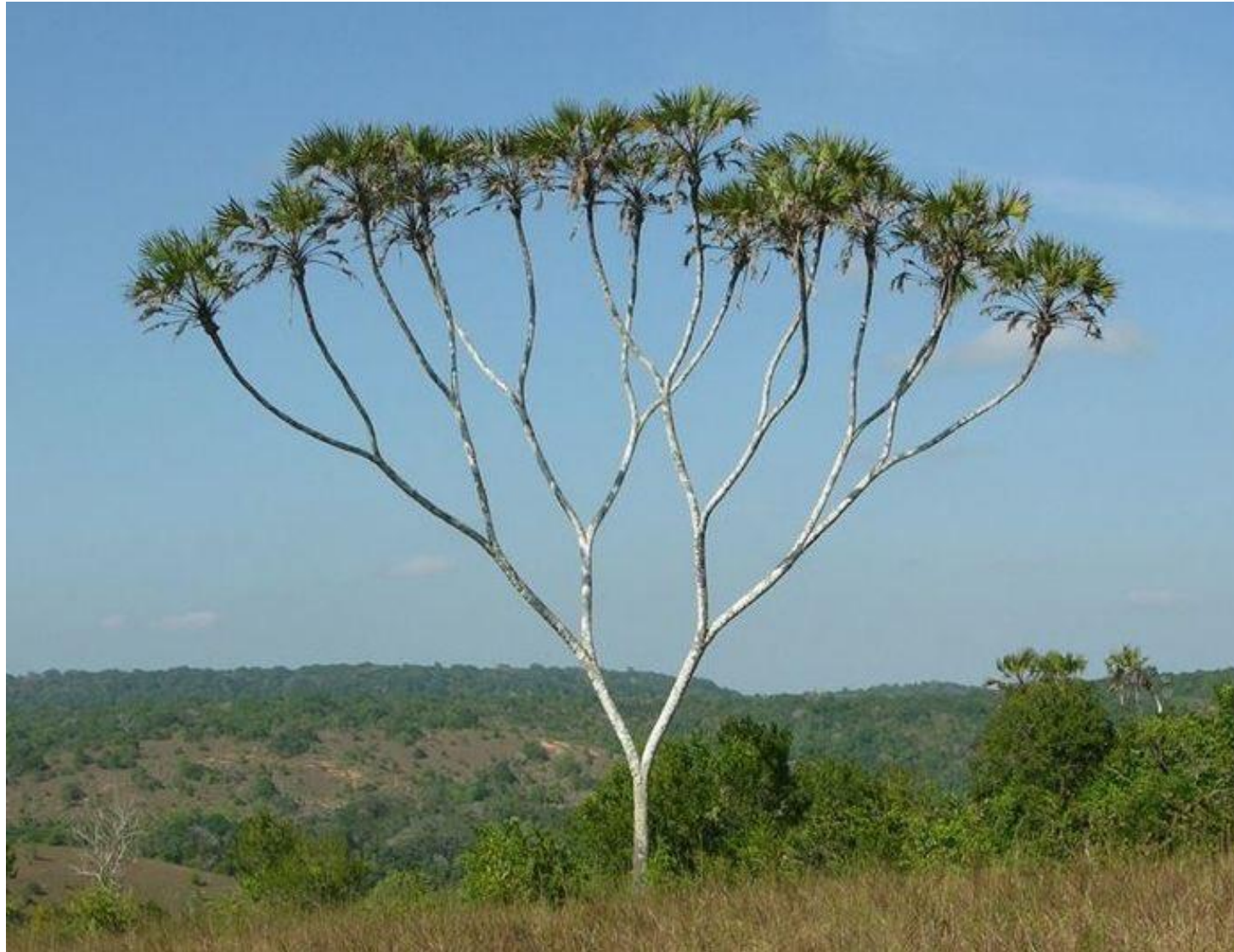
Fig 2.3 Parse tree for conditional statement

Pohon Biner (*binary tree*)

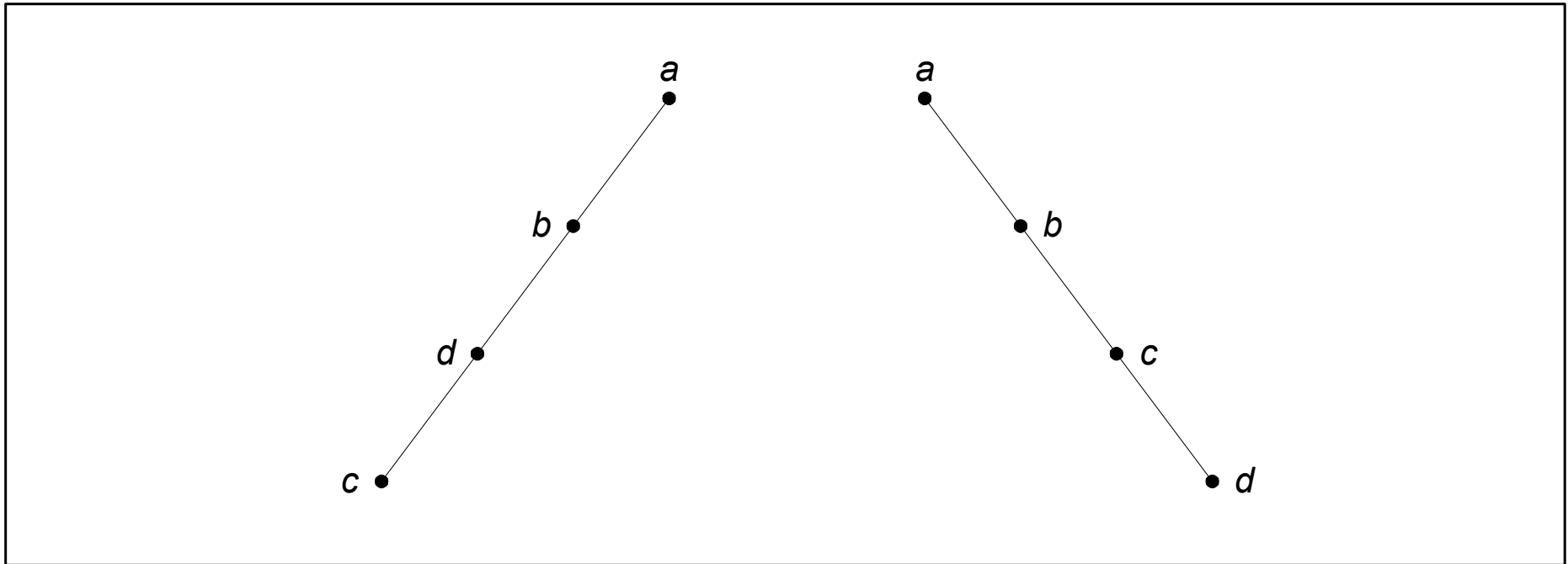
- Adalah pohon *m-ary* dengan $m = 2$.
- Pohon yang paling penting karena banyak aplikasinya.
- Setiap simpul di dalam pohon biner mempunyai paling banyak 2 buah anak.
- Dibedakan antara anak kiri (*left child*) dan anak kanan (*right child*)
- Karena ada perbedaan urutan simpul, maka pohon biner adalah pohon terurut.



Gambar Dua buah pohon biner yang berbeda

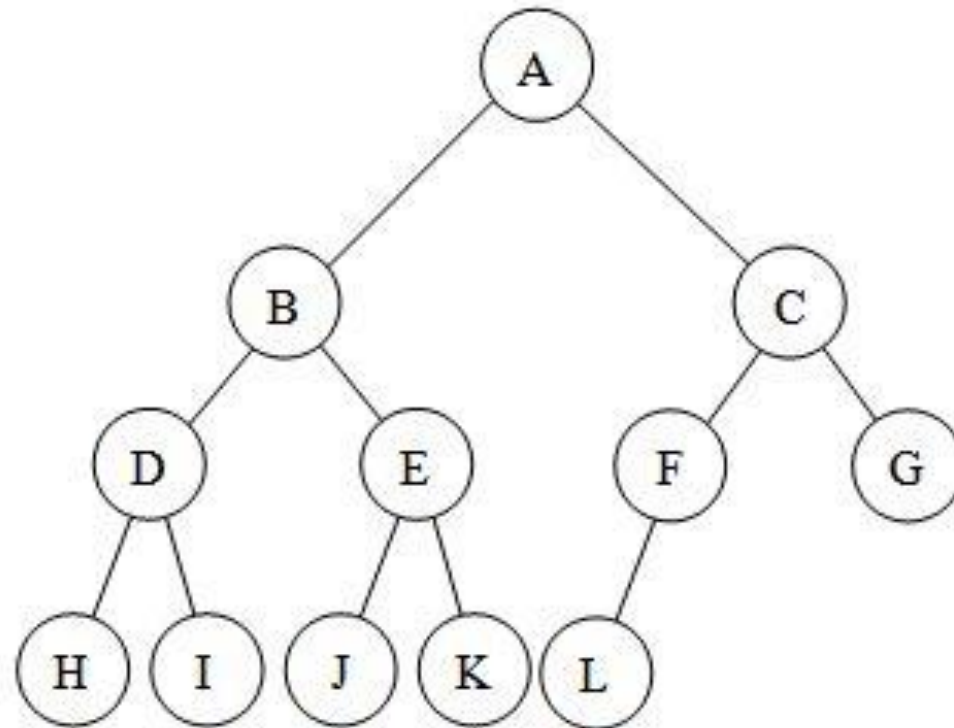


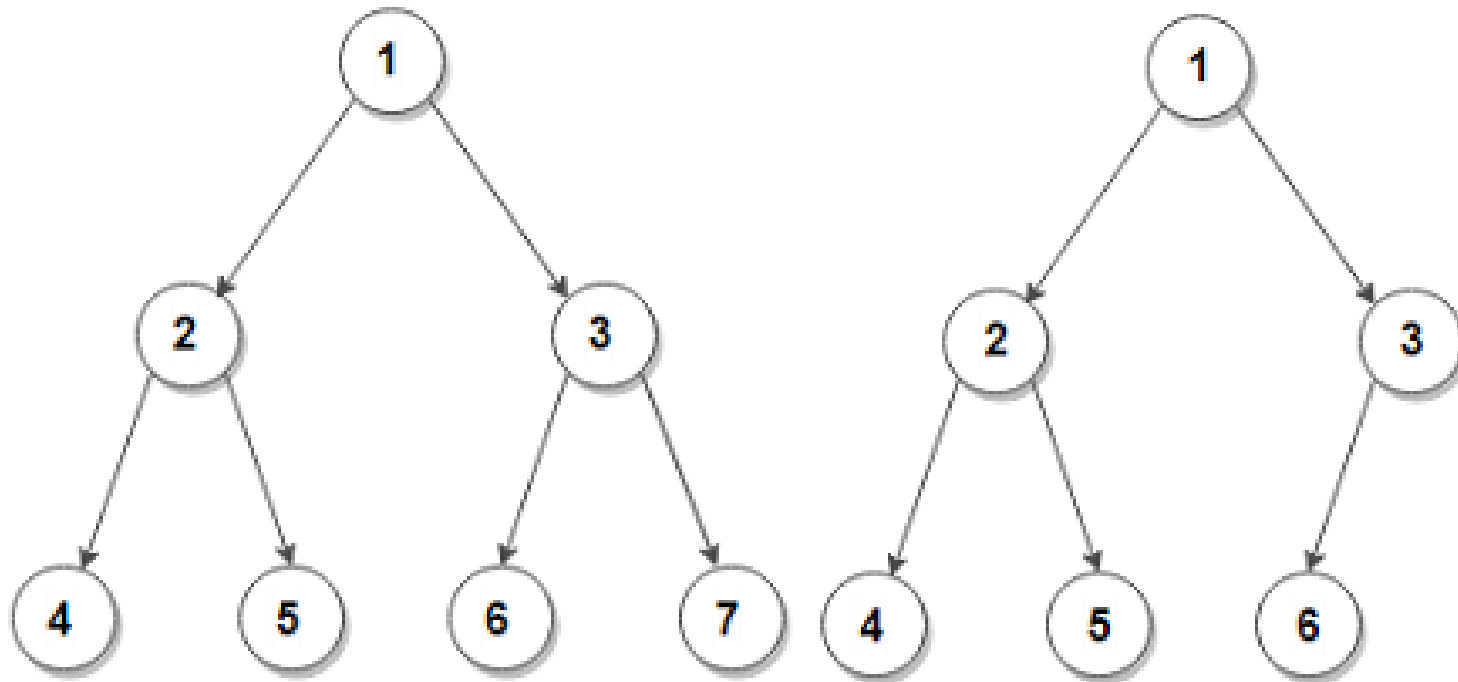
Real binary tree



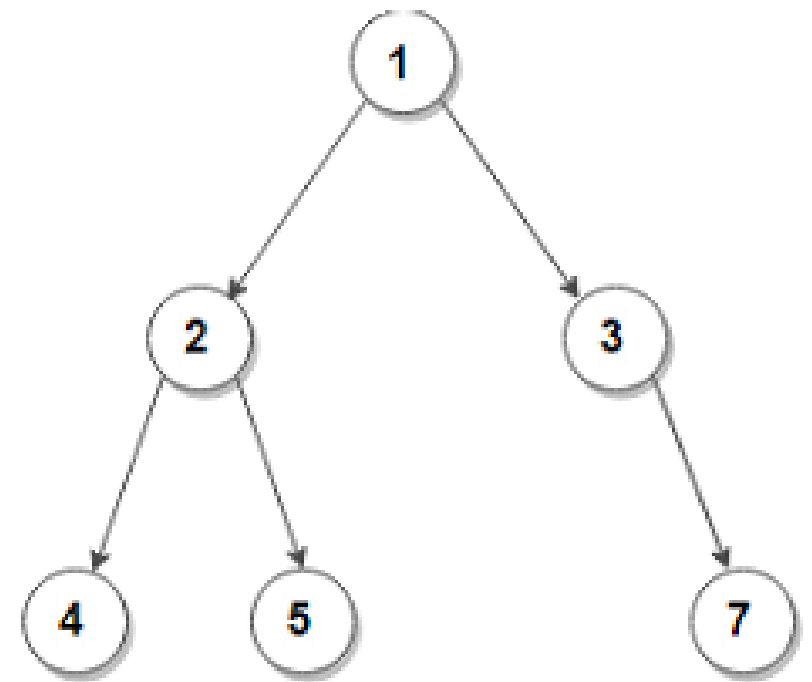
Gambar (a) Pohon condong-kiri, dan (b) pohon condong kanan

Complete binary tree: setiap aras, mungkin kecuali pada aras terakhir, terisi lengkap, dan semua simpul dipadatkan sejauh mungkin ke bagian kiri (pengisian simpul dari kiri ke kanan)

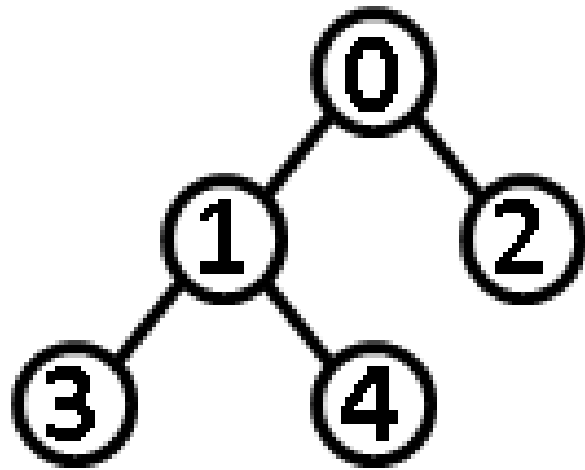




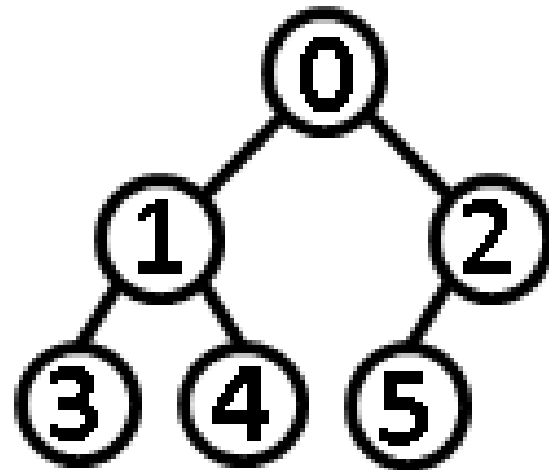
Complete Binary Tree



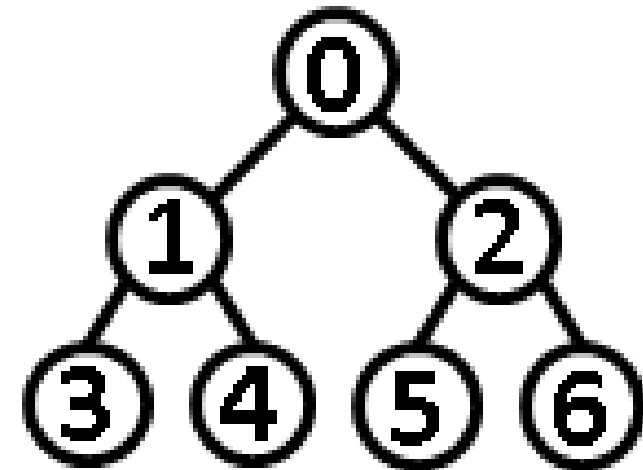
Not a Complete Binary Tree



**full
binary tree**



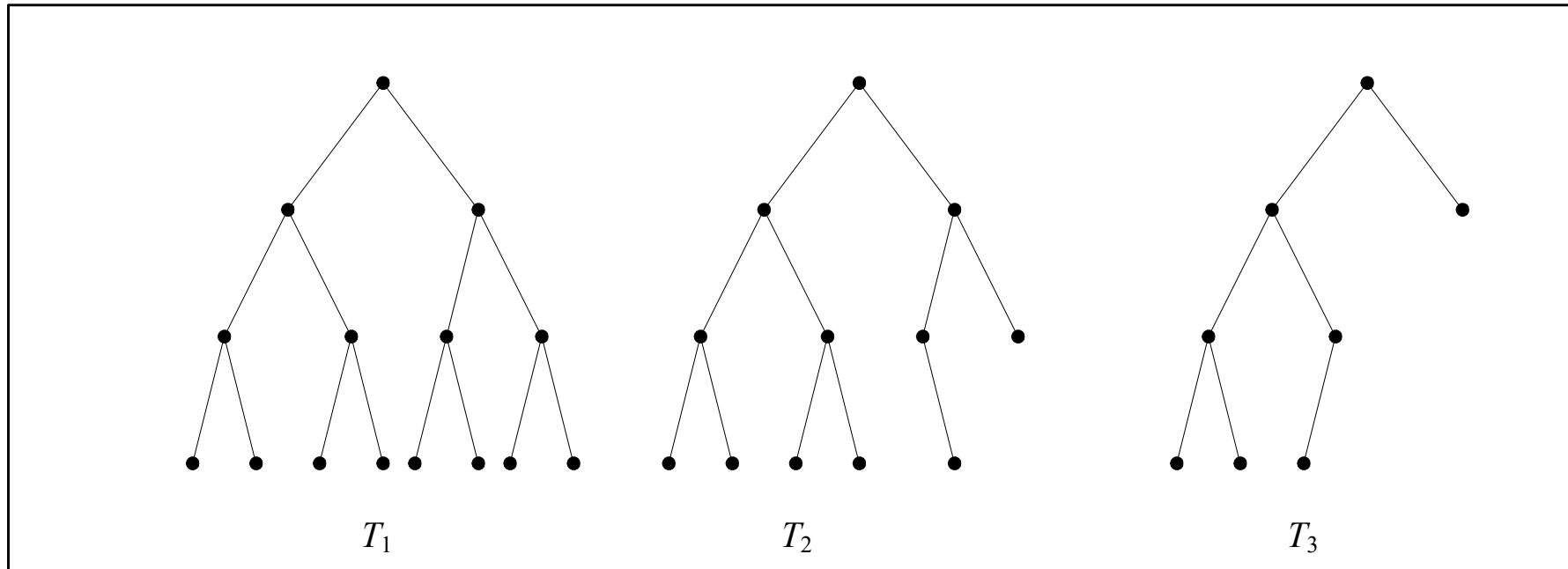
**complete
binary tree**



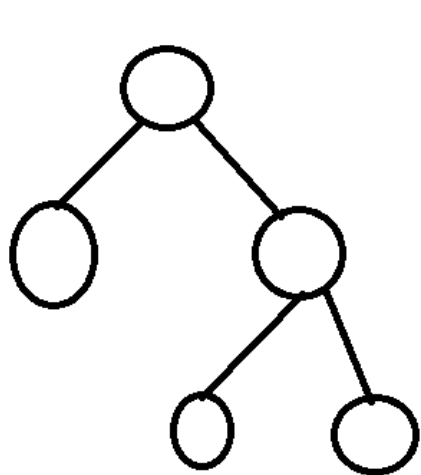
**perfect
binary tree**

Pohon Biner Seimbang

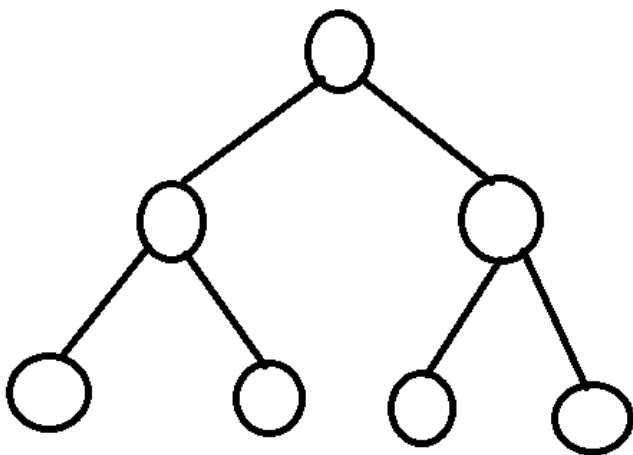
Pada beberapa aplikasi, diinginkan tinggi upapohon kiri dan tinggi upapohon kanan yang seimbang, yaitu berbeda maksimal 1.



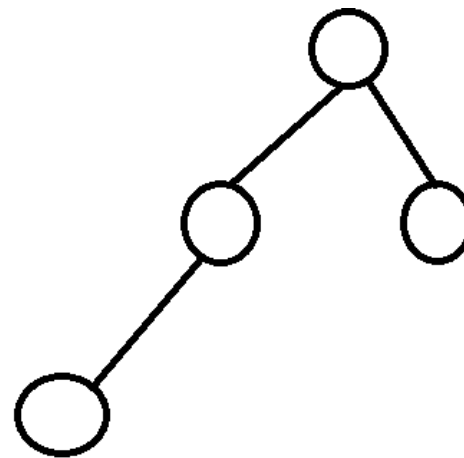
Gambar T_1 dan T_2 adalah pohon seimbang, sedangkan T_3 bukan pohon seimbang.



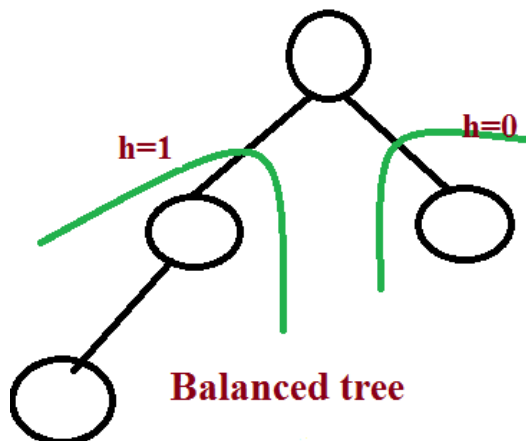
Binary Tree



Fully (perfect) Binary Tree

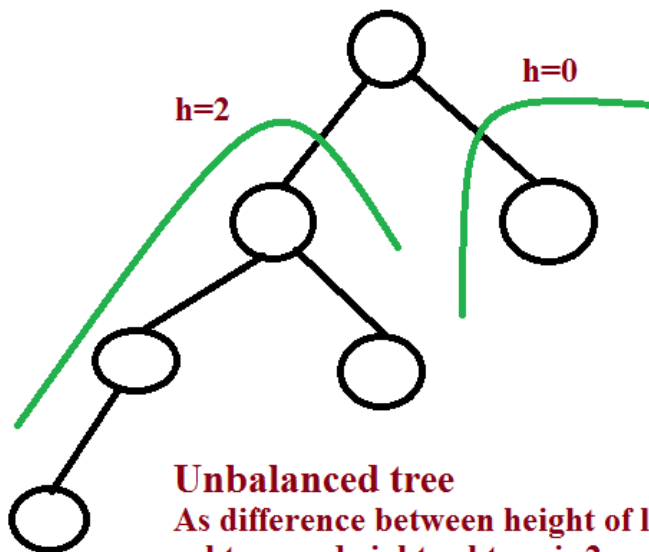


Complete Binary Tree



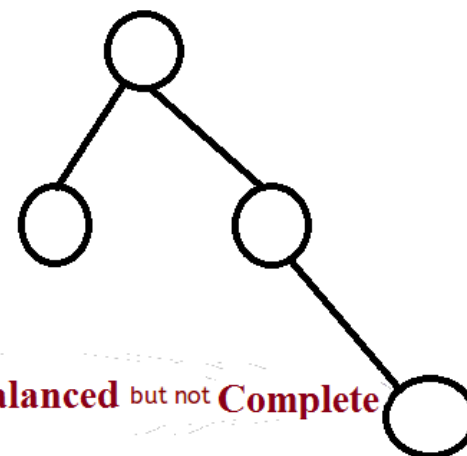
Balanced tree

As difference between height of left subtree and right subtree is 1

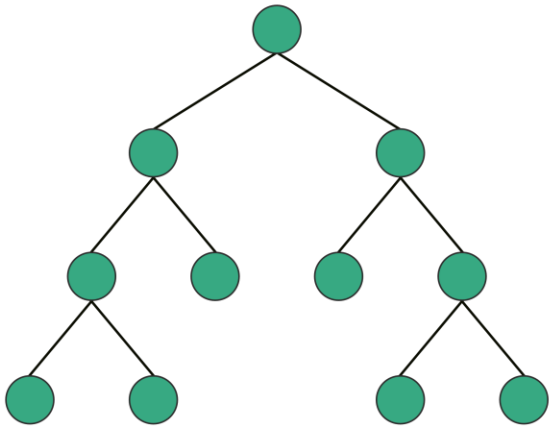


Unbalanced tree

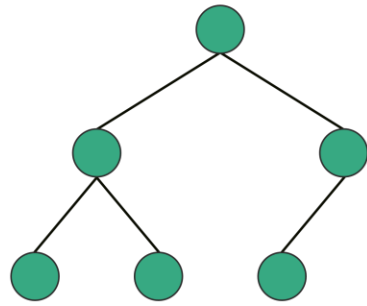
As difference between height of left subtree and right subtree is 2



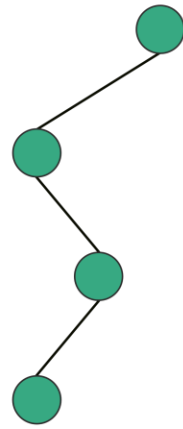
Balanced but not **Complete**



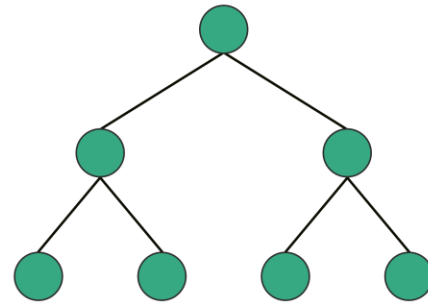
Full



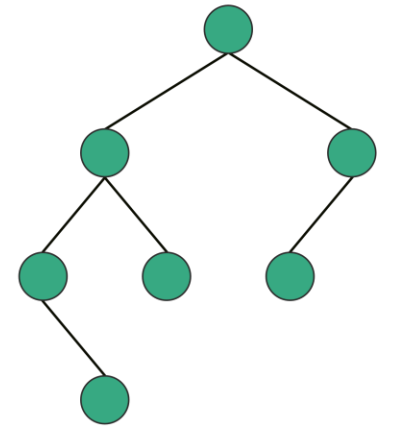
Complete



Degenerate



Perfect

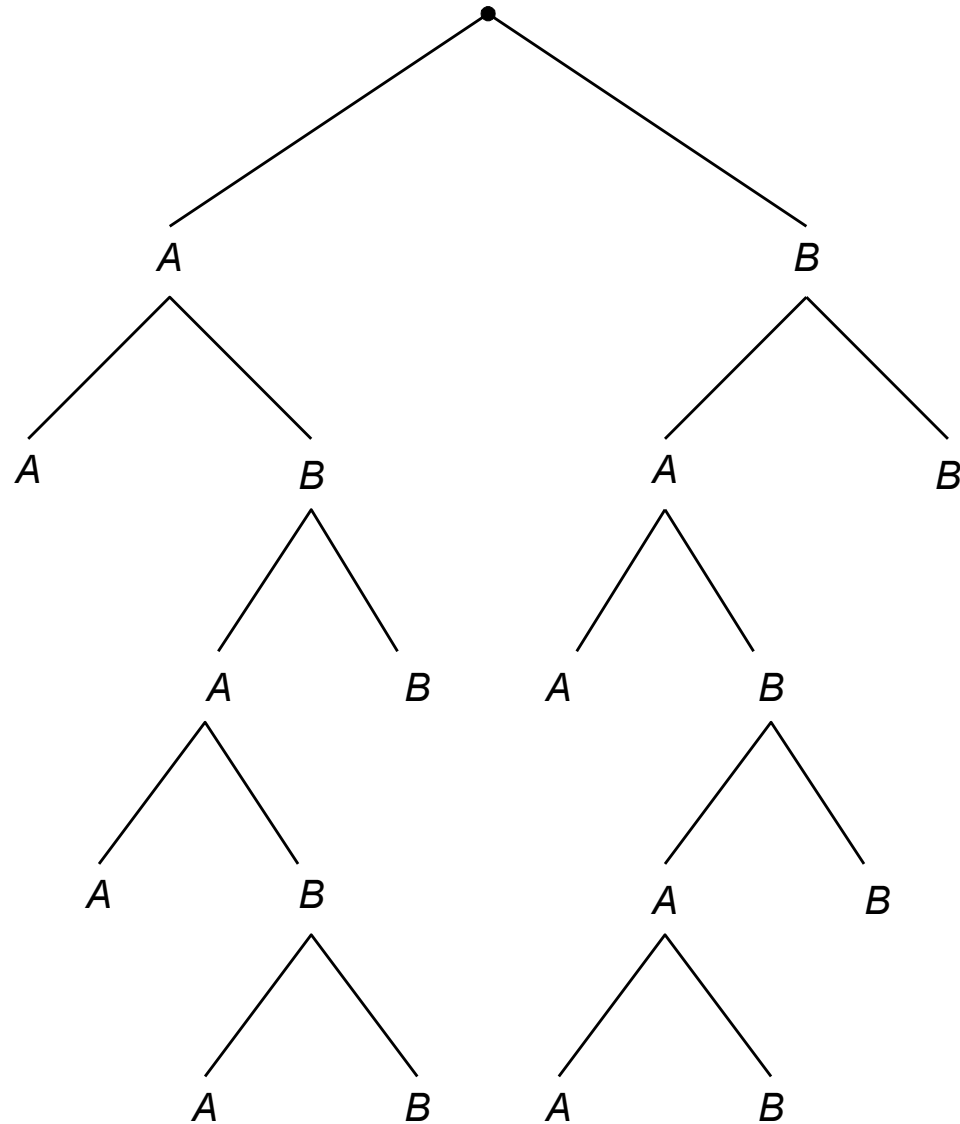


Balanced

Latihan

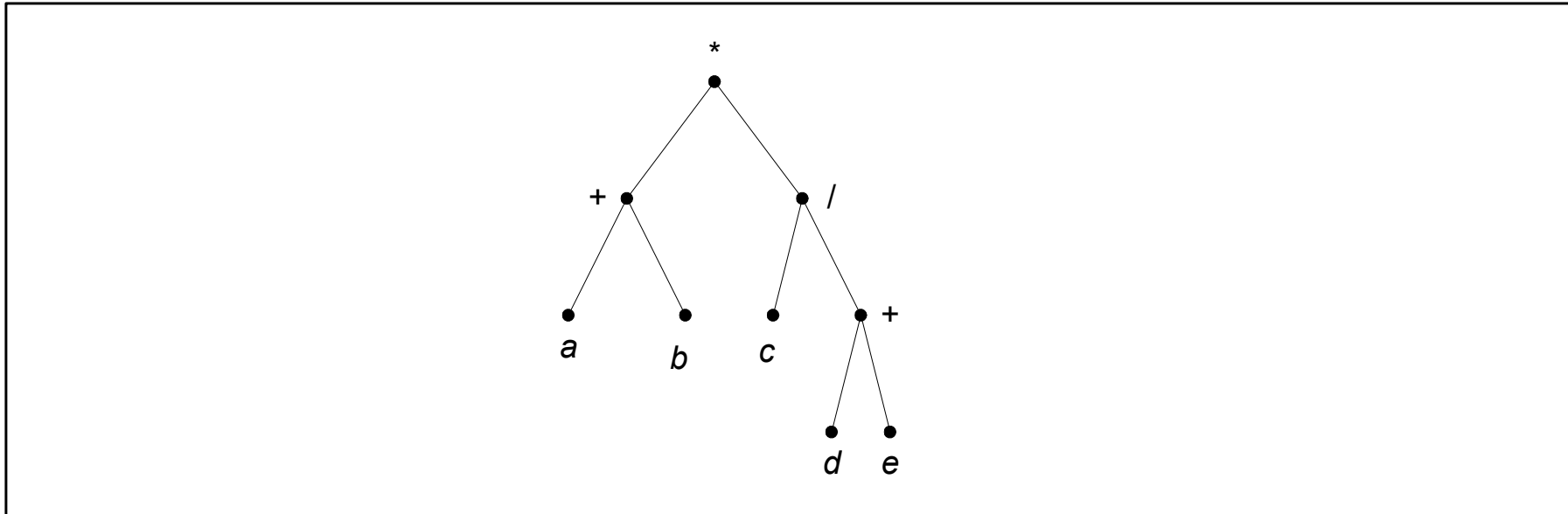
Gunakan pohon berakar untuk menggambarkan semua kemungkinan hasil dari pertandingan tenis antara dua orang pemain, Anton dan Budi, yang dalam hal ini pemenangnya adalah pemain yang pertama memenangkan dua set berturut-turut atau pemain yang pertama memenangkan total tiga set.

Jawaban:



Aplikasi Pohon Biner

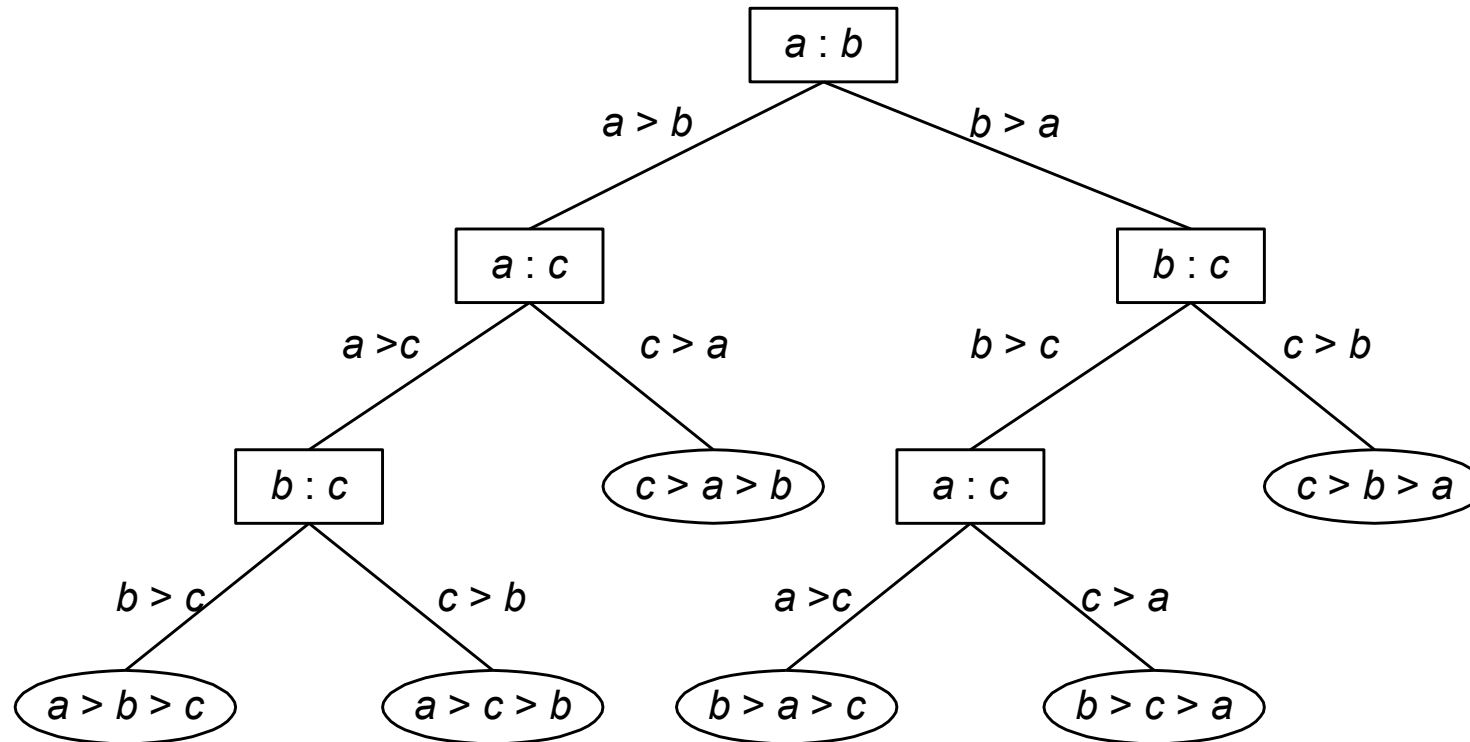
1. Pohon Ekspresi



Gambar Pohon ekspresi dari $(a + b) * (c / (d + e))$

daun \rightarrow operand
simpul dalam \rightarrow operator

2. Pohon Keputusan



Gambar Pohon keputusan untuk mengurutkan 3 buah elemen

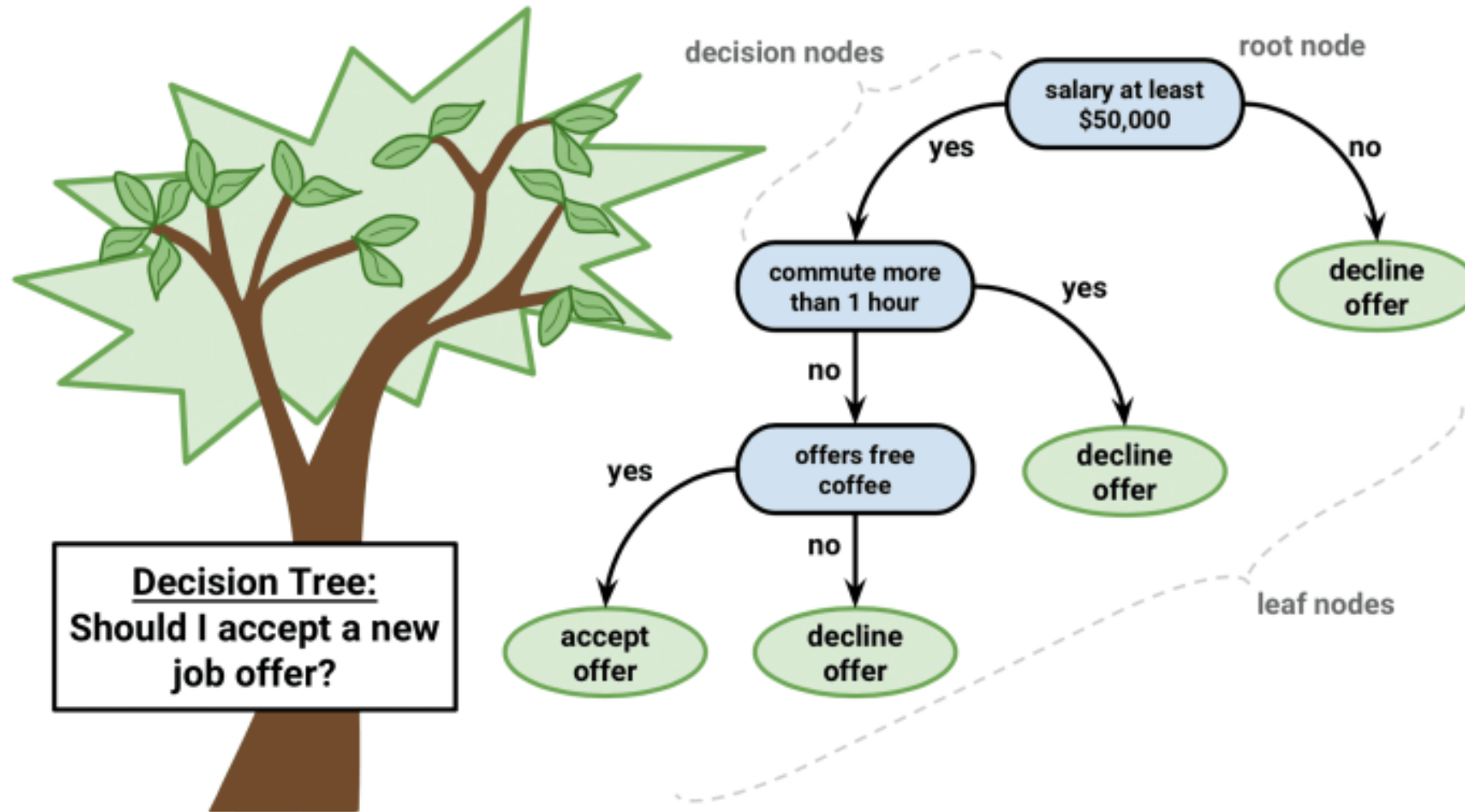
Latihan

Diketahui 8 buah koin uang logam. Satu dari delapan koin itu ternyata palsu. Koin yang palsu mungkin lebih ringan atau lebih berat daripada koin yang asli. Misalkan tersedia sebuah timbangan neraca yang sangat teliti. Buatlah pohon keputusan untuk mencari uang palsu dengan cara menimbang paling banyak hanya 3 kali saja.



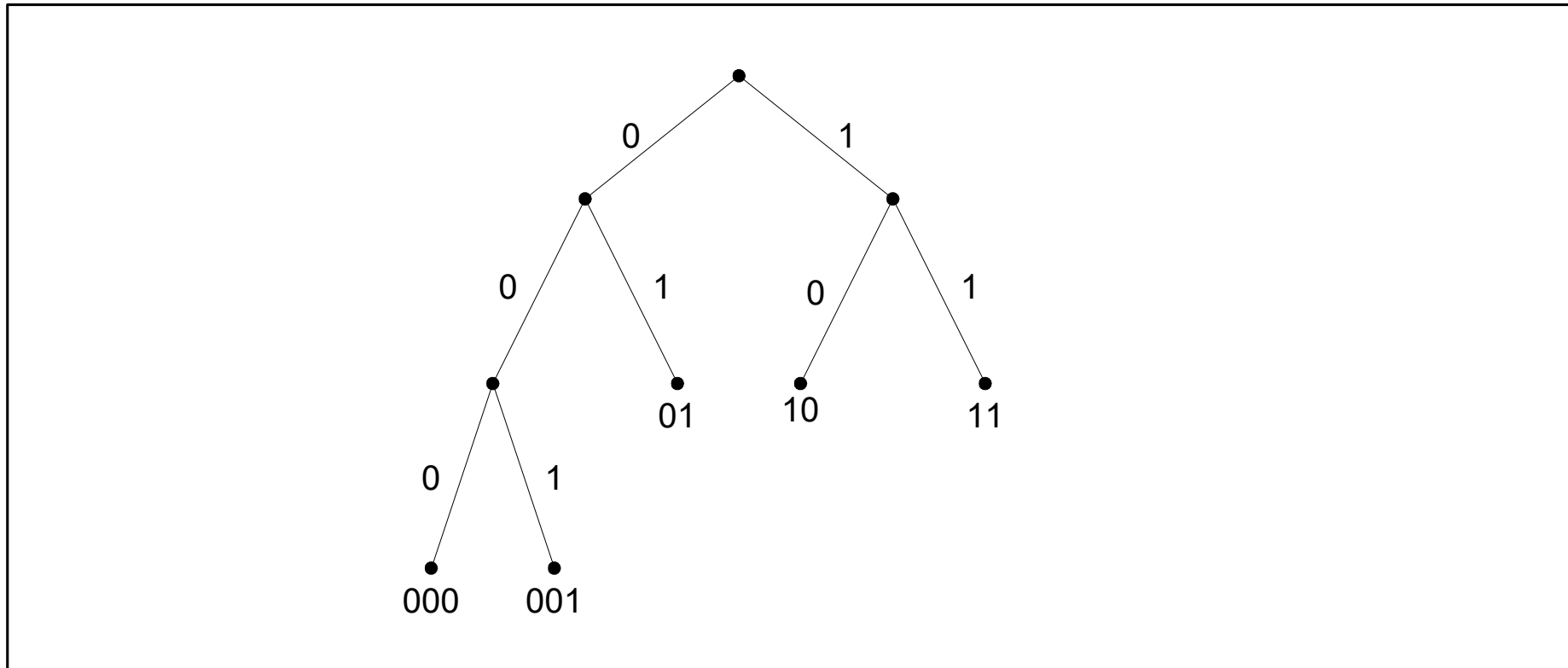
(Petunjuk: misalkan 8 koin tersebut adalah a, b, c, d, e, f, g, dan h)

Pohon keputusan di dalam *Machine Learning (ML)*



3. Kode Awalan (Prefix Code)

Setiap kode bukan merupakan awalan dari kode yang lain



Gambar Pohon biner dari kode prefiks { 000, 001, 01, 10, 11 }

4. Kode Huffman (Aplikasi kode awalan)

Tabel Kode ASCII

Simbol	Kode ASCII
<i>A</i>	01000001
<i>B</i>	01000010
<i>C</i>	01000011
<i>D</i>	01000100

rangkaian bit untuk string '*ABACCD A*':

01000001010000010010000010100000110100000110100010001000001

atau $7 \times 8 = 56$ bit (*7 byte*).

Tabel kekerapan (frekuensi) dan kode Huffman untuk *string* *ABACCD*

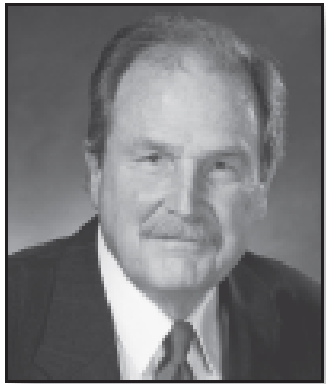
Simbol	Kekerapan	Peluang	Kode Huffman
<i>A</i>	3	$3/7$	0
<i>B</i>	1	$1/7$	110
<i>C</i>	2	$2/7$	10
<i>D</i>	1	$1/7$	111

Dengan kode Huffman, rangkaian bit untuk '*ABACCD*':

0110010101110

hanya 13 bit!

Bagaimana cara memperoleh kode Huffman?



DAVID A. HUFFMAN (1925–1999) David Huffman grew up in Ohio. At the age of 18 he received his B.S. in electrical engineering from The Ohio State University. Afterward he served in the U.S. Navy as a radar maintenance officer on a destroyer that had the mission of clearing mines in Asian waters after World War II. Later, he earned his M.S. from Ohio State and his Ph.D. in electrical engineering from MIT. Huffman joined the MIT faculty in 1953, where he remained until 1967 when he became the founding member of the computer science department at the University of California at Santa Cruz. He played an important role in developing this department and spent the remainder of his career there, retiring in 1994.

Huffman is noted for his contributions to information theory and coding, signal designs for radar and for communications, and design procedures for asynchronous logical circuits. His work on surfaces with zero curvature led him to develop original techniques for folding paper and vinyl into unusual shapes considered works of art by many and publicly displayed in several exhibits. However, Huffman is best known for his development of what is now called Huffman coding, a result of a term paper he wrote during his graduate work at MIT.

Huffman enjoyed exploring the outdoors, hiking, and traveling extensively. He became certified as a scuba diver when he was in his late 60s. He kept poisonous snakes as pets.

Sumber: Rosen

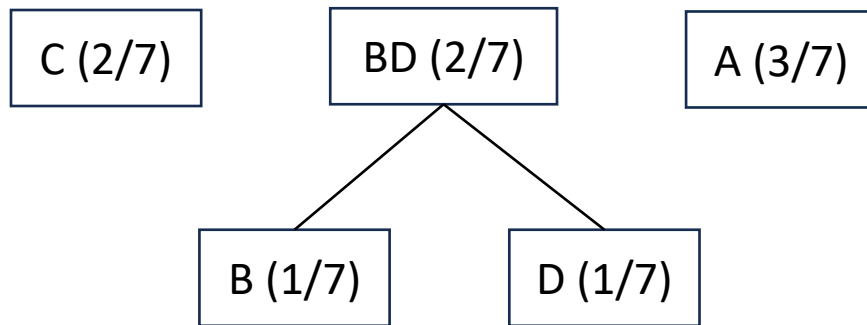
Algoritma Kompresi Huffman

Algoritma pembentukan kode Huffman

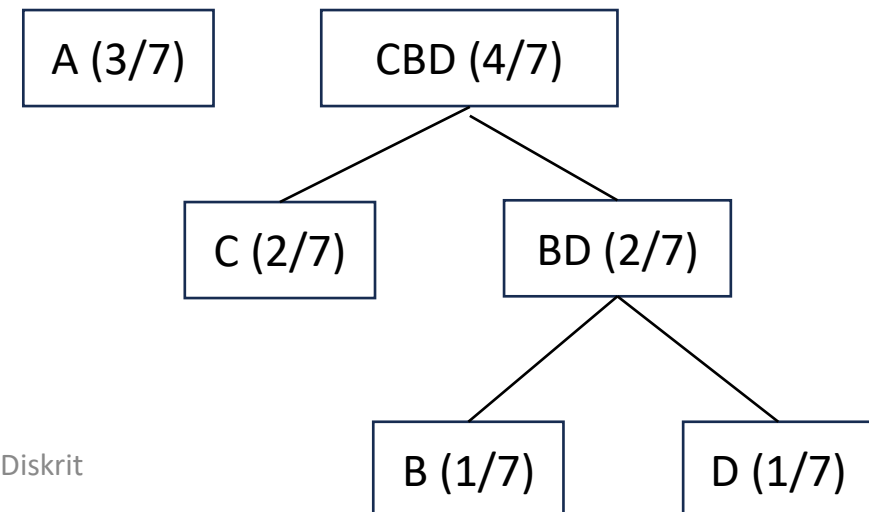
1. Urutkan simbol berdasarkan peluangnya (atau jumlah kemunculan) dari kecil ke besar
2. Pilih dua simbol dengan peluang paling kecil (Pada contoh di atas simbol B dan D). Kedua simbol tadi dikombinasikan sebagai simpul orangtua. Simbol orangtua adalah gabungan (*concatenation*) dari kedua simbol tersebut. Peluang simbol hasil penggabungan adalah jumlah peluang dari kedua simbol (Pada contoh di atas simbol B dan D digabung menjadi simbol BD dengan peluang $1/7 + 1/7 = 2/7$). Tempatkan simbol hasil gabungan sesuai dalam urutan menaik.
3. Selanjutnya, pilih dua simbol berikutnya, termasuk simbol baru, yang mempunyai peluang terkecil.
4. Ulangi langkah 3 sampai seluruh simbol habis.
5. Beri label secara konsisten sisi kiri dengan 0 dan sisi kanan dengan 1 (atau 1 di kiri dan 0 kanan).
6. Lintasan dari akar ke daun berisi sisi-sisi pohon yang deretan labelnya menyatakan kode Huffman untuk simbol daun tersebut

1. B (1/7) D (1/7) C (2/7) A (3/7) (terurut membesar)

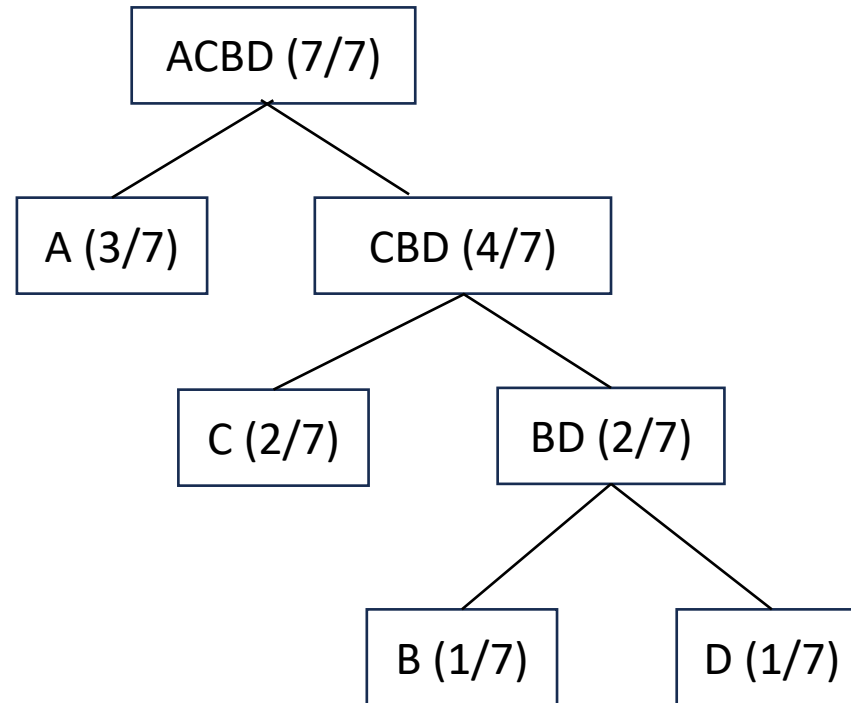
2. Gabungkan B dan D menjadi BD, peluang $BD = 1/7 + 1/7 = 2/7$, tempatkan simbol baru BD di dalam urutan yang terurut menaik



3. Gabungkan C dan BD menjadi CBD, peluang $CBD = 2/7 + 2/7 = 4/7$, tempatkan simbol baru CBD di dalam urutan yang terurut menaik

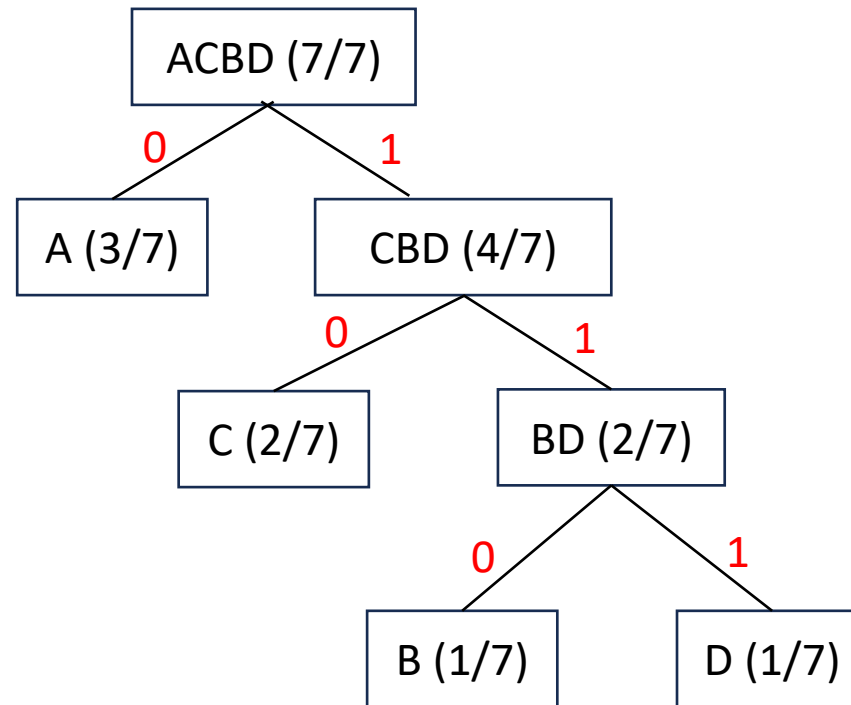


3. Gabungkan A dan CBD menjadi ACBD, peluang $CCBD = 3/7 + 4/7 = 7/7$



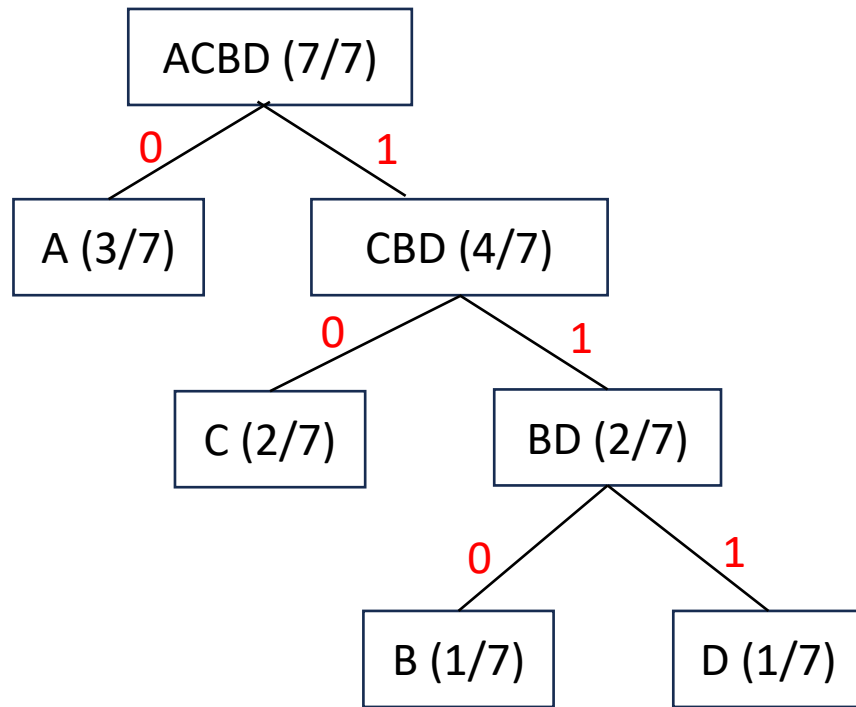
4. Simbol sudah habis. Stop. Pohon Huffman sudah terbentuk. Selanjutnya ke Langkah 5.

5. Sisi-sisi kiri di dalam pohon Huffman diberi label 0, sisi-sisi kanan diberi label 1.



6. Traversal pohon dari akar ke daun. Lintasan dari akar ke daun berisi rangkaian bit yang menyatakan kode Huffman untuk setiap simbol di dalam daun.

Kode Huffman: A = 0, C = 10, B = 110, D = 111



- Kode Huffman: A = 0, C = 10, B = 110, D = 111

Encoding:

- Kodekan setiap simbol di dalam pesan tadi dengan kode Huffman:
- Contoh: ABACCCDA = 0110010101110

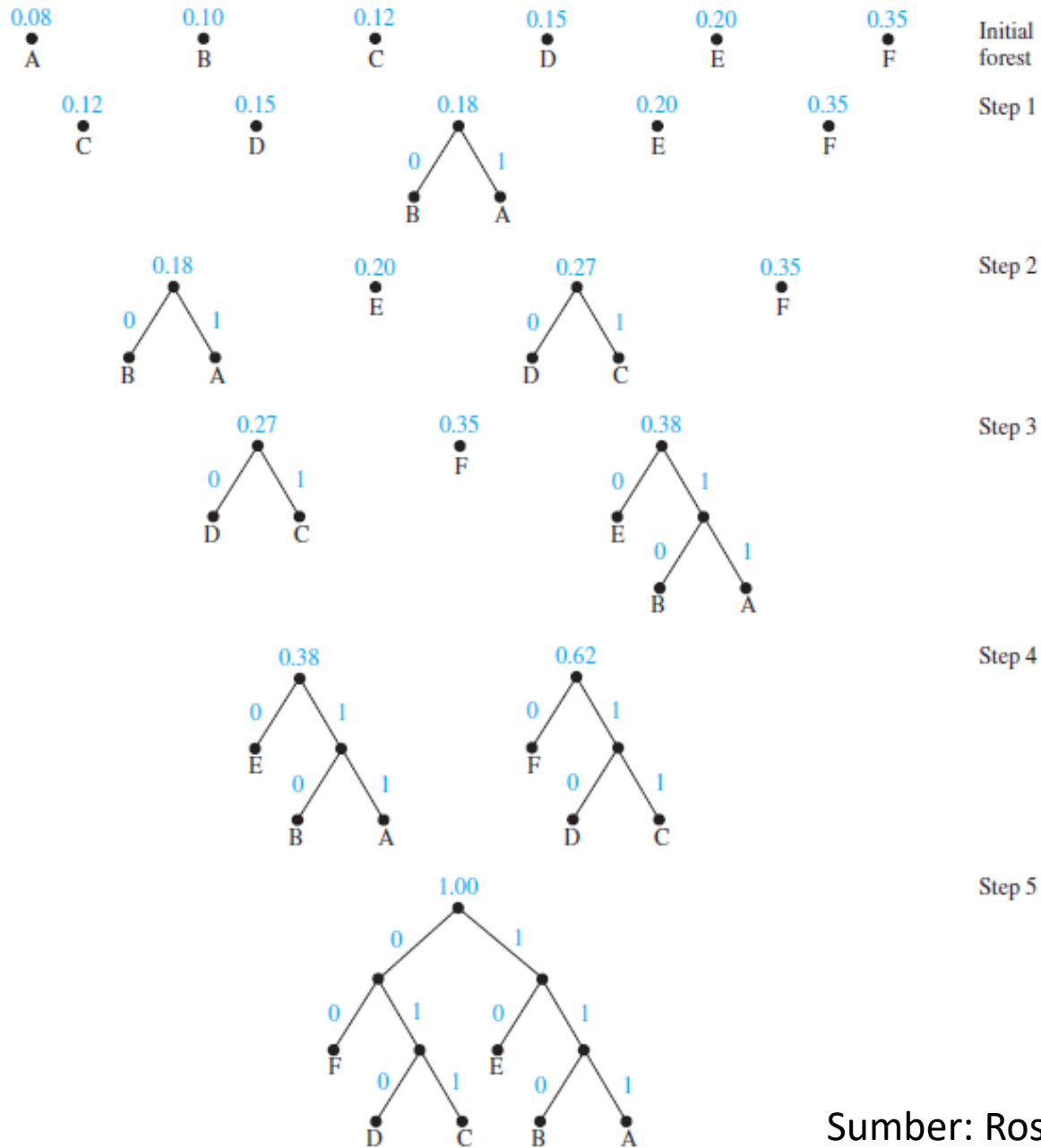
Decoding:

1. Baca simbol biner pertama
2. Mulai traversal dari akar mengikuti sisi yang sesuai dengan simbol biner tersebut
3. Baca terus simbol biner dan traversal sisi yang bersesuaian sampai ketemu daun. Kodekan barisan biner yang sudah dibaca dengan simbol daun
4. Baca simbol biner berikutnya dan ulangi traversal dari akar. Ulangi langkah 4 sampai semua simbol biner habis

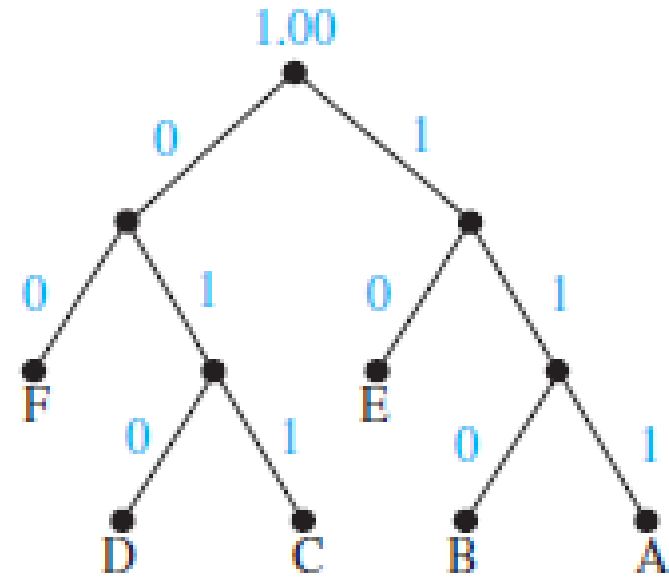
Contoh 2: Sebuah pesan terdiri dari huruf a, b, c, d, e, dan f . Peluang kemunculan setiap huruf di dalam pesan adalah sebagai berikut:

Karakter	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Peluang	0.08	0.10	0.12	0.15	0.20	0.35

Tentukan kode Huffman untuk setiap huruf tersebut!



Hasil akhir:



Kode Huffman:

A = 111,

B = 110,

C = 011,

D = 010,

E = 10

F = 00.

Panjang rata-rata jumlah bit untuk mengkodekan setiap simbol adalah

$$3 \cdot 0.08 + 3 \cdot 0.10 + 3 \cdot 0.12 + 3 \cdot 0.15 + 2 \cdot 0.20 + 2 \cdot 0.35 = 2.45.$$

Sumber: Rosen

Latihan (Kuis 2021)

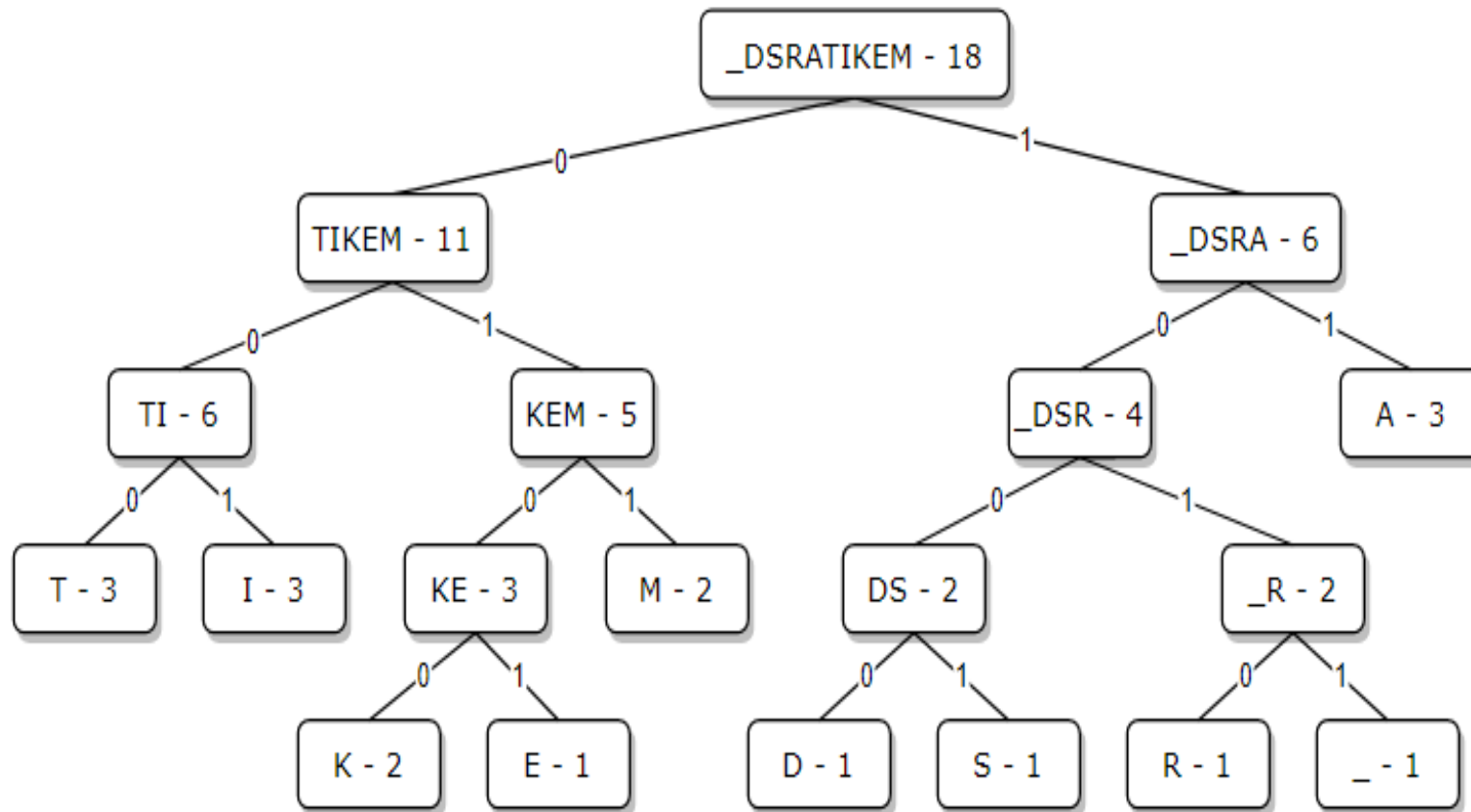
Terdapat sebuah pesan “MATEMATIKA DISKRIT” dalam sebuah *script*. Berapakah panjang kode pesan tersebut jika dikodekan dengan kode Huffman (termasuk spasi)!

Jawaban:

Hitung tiap frekuensi huruf unik di dalam “MATEMATIKA DISKRIT”

M	: 2	K	: 2
A	: 3	D	: 1
T	: 3	S	: 1
E	: 1	R	: 1
I	: 3	_	: 1

Pohon Huffman dan kode Huffman yang terbentuk adalah seperti berikut



Simbol	Frekuensi	Kode
A	3	11
T	3	000
I	3	001
M	2	011
K	2	0100
E	1	0101
D	1	1000
S	1	1001
R	1	1010
_	1	1011

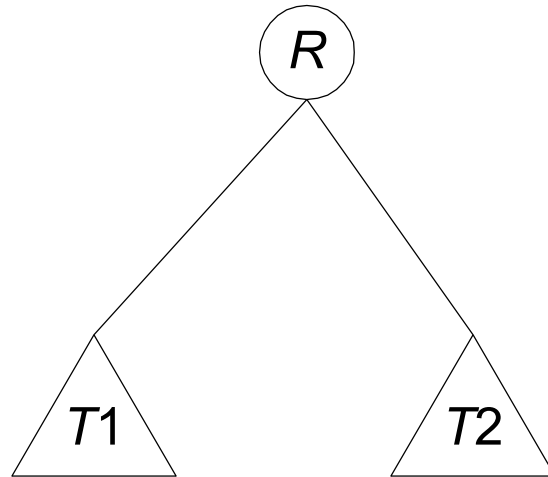
Jadi panjang kode Huffman dari "MATEMATIKA DISKRIT" adalah,
 $3*2 + 3*3 + 3*3 + 2*3 + 2*4 + 1*4 + 1*4 + 1*4 + 1*4 + 1*4$
 $= 6 + 9 + 9 + 6 + 8 + 4 + 4 + 4 + 4 + 4 = 58$
 bit

Latihan

Lakukan kompresi pesan ADA APA DENGAN CINTA (termasuk spasi) dengan menggunakan kode Huffman. Gambarkan pohonnya, lalu kodekan pesan tersebut dengan kode Huffman, lalu hitung rasio kompresinya. Rasio kompresi dihitung dengan rumus:

$$\text{rasio} = (\text{jumlah bit setelah kompresi} / \text{jumlah bit sebelum kompresi}) \times 100\%$$

5. Pohon Pencarian Biner (*Binary Search Tree*)

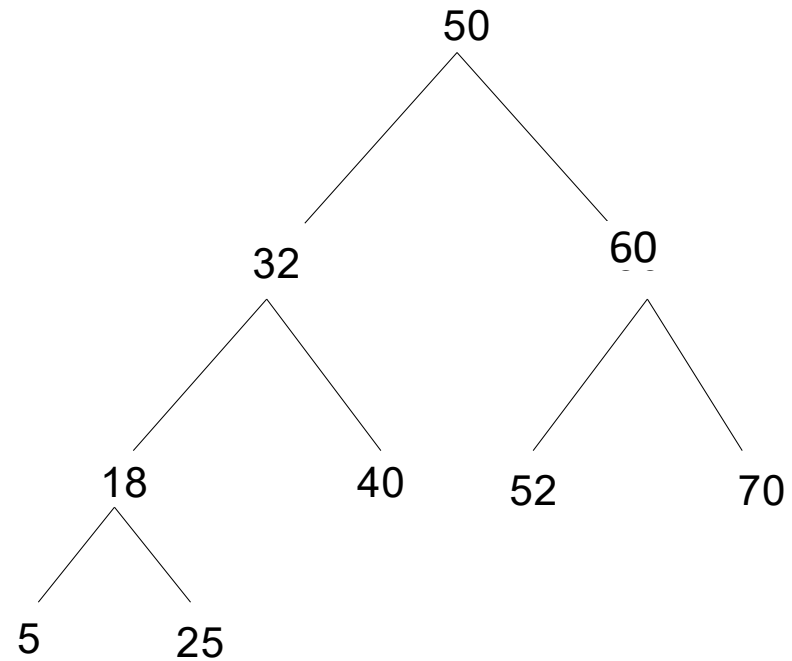


$\text{Kunci}(T1) < \text{Kunci}(R)$

$\text{Kunci}(T2) > \text{Kunci}(R)$



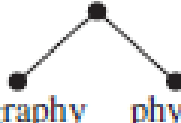
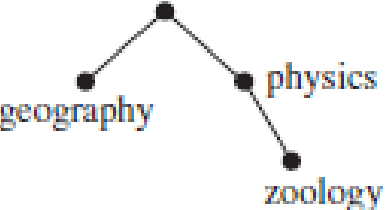
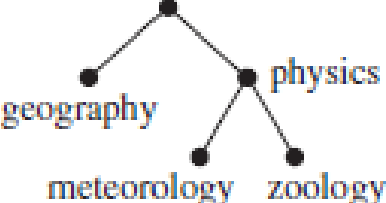
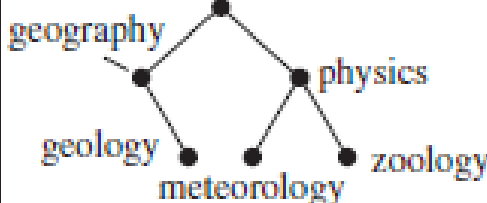
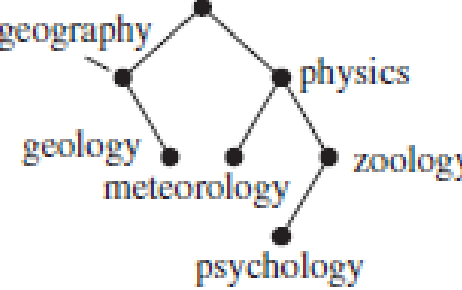
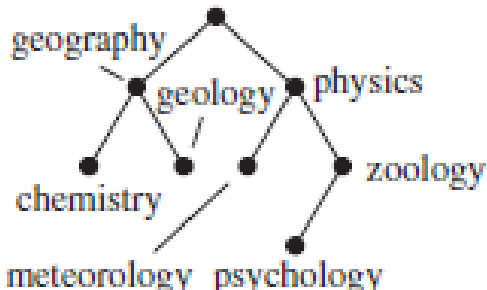
Semua nilai di upa-pohon T1 selalu lebih kecil dari nilai root
Semua nilai di upa-pohon T2 selalu lebih ebsar dari nilai root

Data: 50, 32, 18, 40, 60, 52, 5, 25, 70



Berapa banyak perbandingan yang dilakukan untuk menemukan 25?
Berapa banyak perbandingan yang dilakukan untuk menemukan 100?

Latihan: Bentuklah pohon pencarian biner untuk kata-kata: *mathematics*, *physics*, *geography*, *zoology*, *meteorology*, *geology*, *psychology*, dan *chemistry* (menggunakan urutan alfabet)

<p>mathematics</p> 	<p>mathematics</p>  <p>physics</p> <p>physics > mathematics</p>	<p>mathematics</p>  <p>geography physics</p> <p>geography < mathematics</p>	<p>mathematics</p>  <p>geography physics</p> <p>zoology</p> <p>zoology > mathematics zoology > physics</p>
<p>mathematics</p>  <p>geography physics</p> <p>meteorology zoology</p> <p>meteorology > mathematics meteorology < physics</p>	<p>mathematics</p>  <p>geography physics</p> <p>geology meteorology zoology</p> <p>geology < mathematics geology > geography</p>	<p>mathematics</p>  <p>geography physics</p> <p>geology meteorology zoology</p> <p>psychology</p> <p>psychology > mathematics psychology > physics psychology < zoology</p>	<p>mathematics</p>  <p>geography physics</p> <p>chemistry geology zoology</p> <p>meteorology psychology</p> <p>chemistry < mathematics chemistry < geography</p>

Penelusuran (traversal) Pohon Biner

1. *Preorder* : R, T_1, T_2

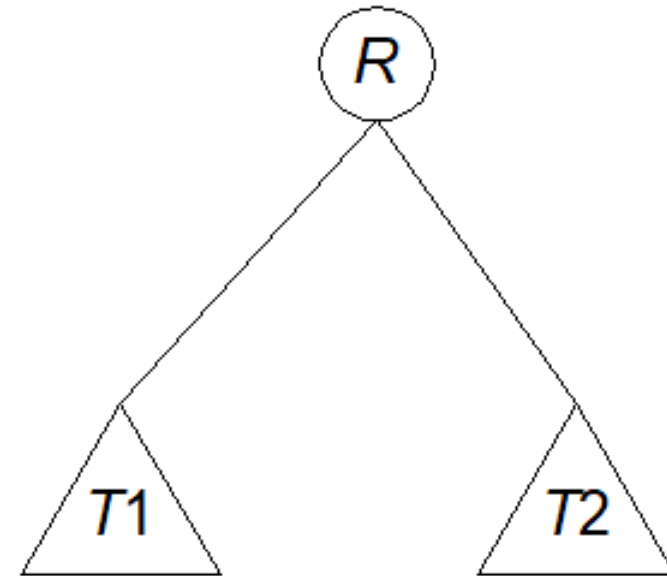
- kunjungi R
- kunjungi T_1 secara *preorder*
- kunjungi T_2 secara *preorder*

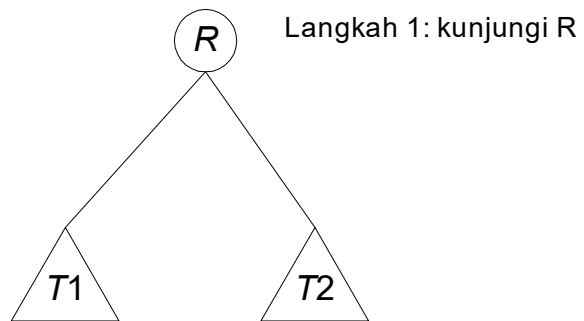
2. *Inorder* : T_1, R, T_2

- kunjungi T_1 secara *inorder*
- kunjungi R
- kunjungi T_2 secara *inorder*

3. *Postorder* : T_1, T_2, R

- kunjungi T_1 secara *postorder*
- kunjungi T_2 secara *postorder*
- kunjungi R

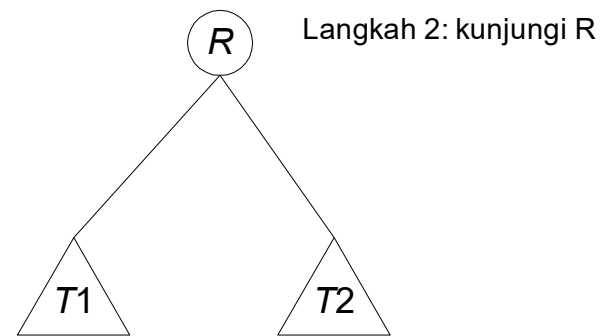




Langkah 2: kunjungi $T1$
secara *preorder*

Langkah 3: kunjungi $T2$
secara *preorder*

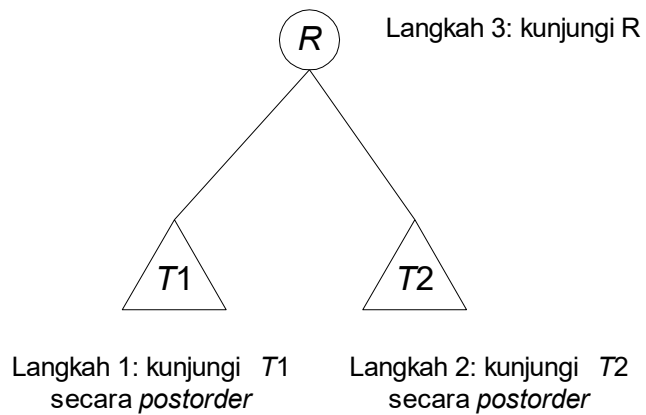
(a) *preorder*



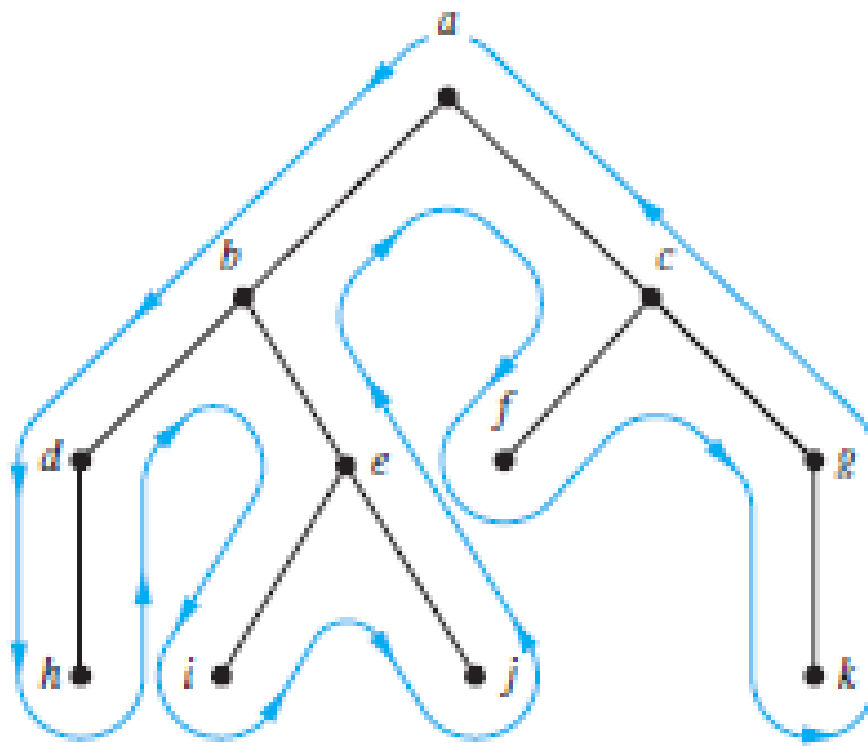
Langkah 1: kunjungi $T1$
secara *inorder*

Langkah 3: kunjungi $T2$
secara *inorder*

(b) *inorder*



(c) *postorder*



Sumber: Rosen

FIGURE 9 A Shortcut for Traversing an Ordered Rooted Tree in Preorder, Inorder, and Postorder.

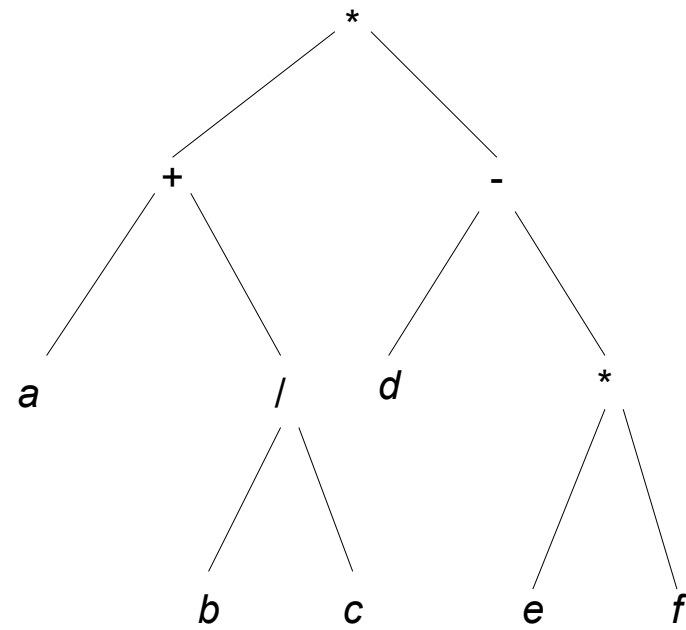
Preorder: a, b, d, h, e, i, j, c, f, g, k
 Inorder: h, d, b, i, e, j, a, f, c, k, g
 Postorder: h, d, i, j, e, b, f, k, g, c, a

Traversal pada pohon ekspresi:

preorder : * + a / b c - d * e f (prefix)

inorder : a + b / c * d - e * f (infix)

postorder : a b c / + d e f * - * (postfix)



Latihan

Diberikan masukan berupa rangkaian karakter dengan urutan sebagai berikut:

P, T, B, F, H, K, N, S, A, U, M, I, D, C, W, O

- (a) Gambarkan pohon pencarian (*search tree*) yang terbentuk.
- (b) Tentukan hasil penelusuran *preorder*, *inorder*, dan *postorder*, dari pohon jawaban (a) di atas.

Traversal pada Pohon m -ary

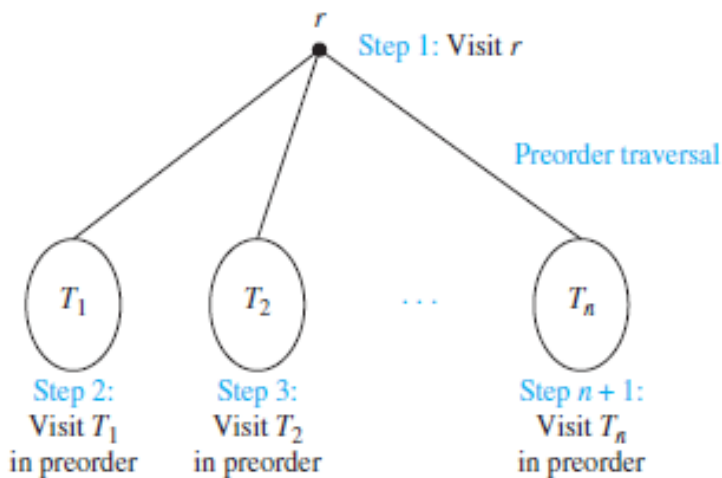


FIGURE 2 Preorder Traversal.

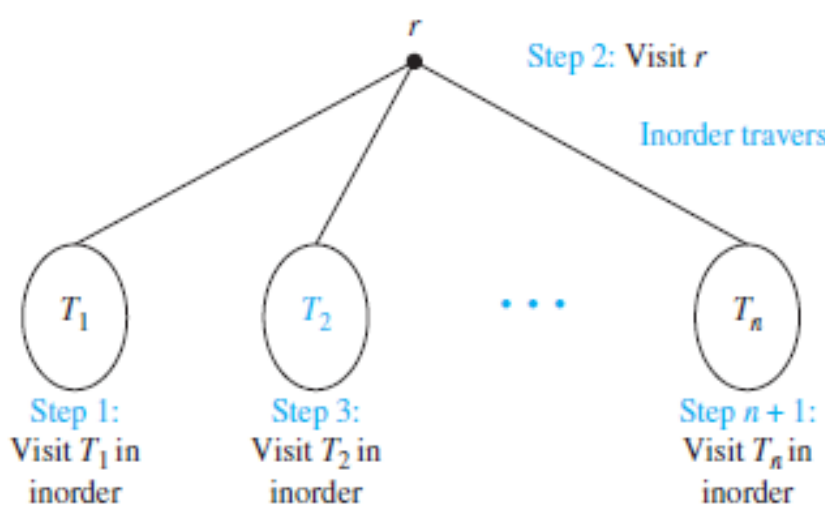


FIGURE 5 Inorder Traversal.

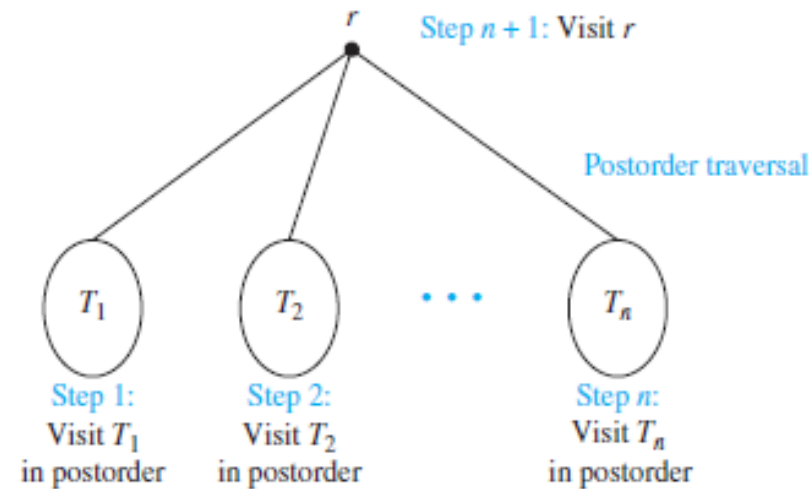
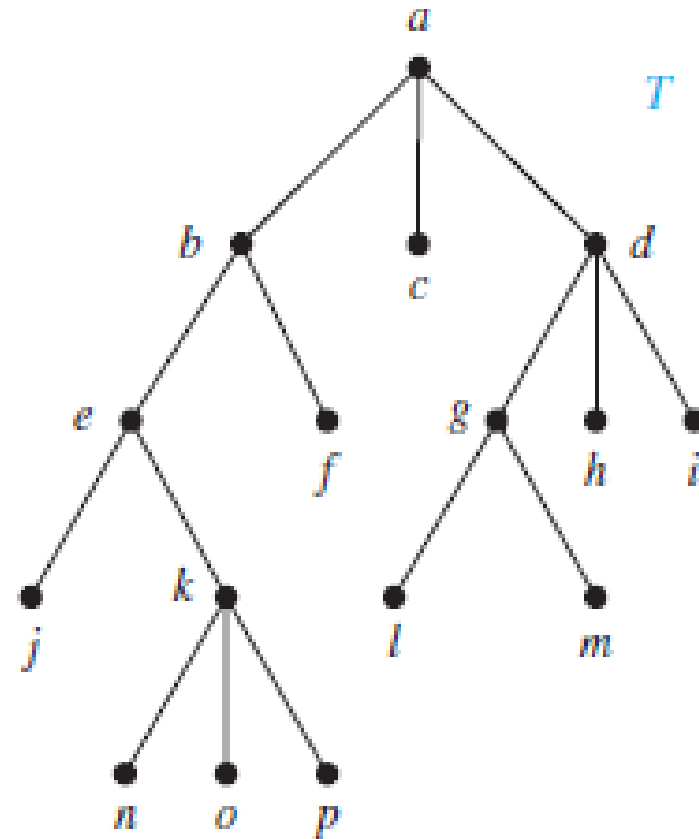


FIGURE 7 Postorder Traversal.

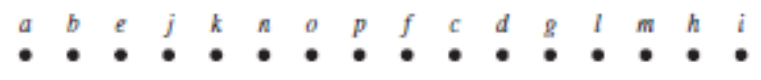
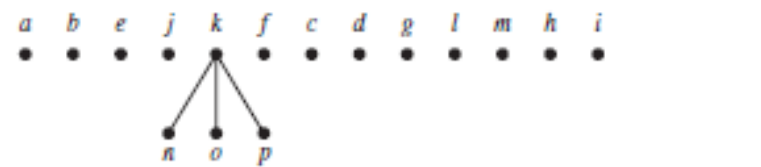
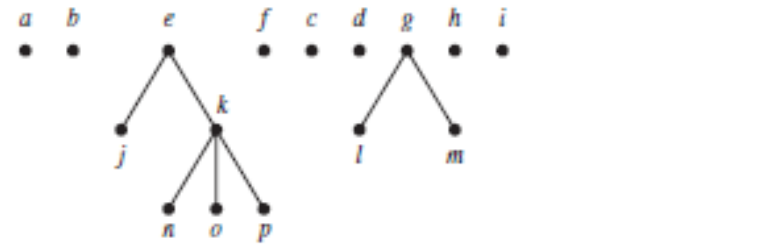
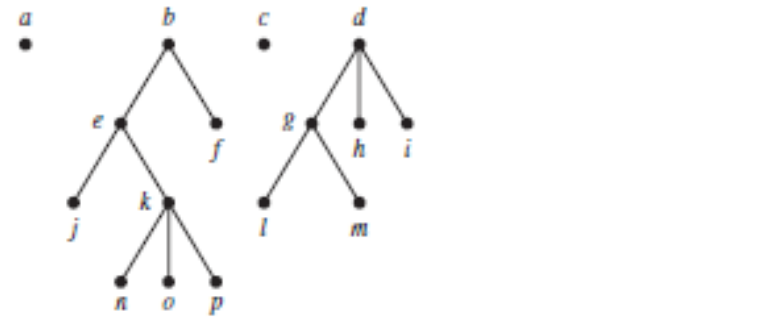
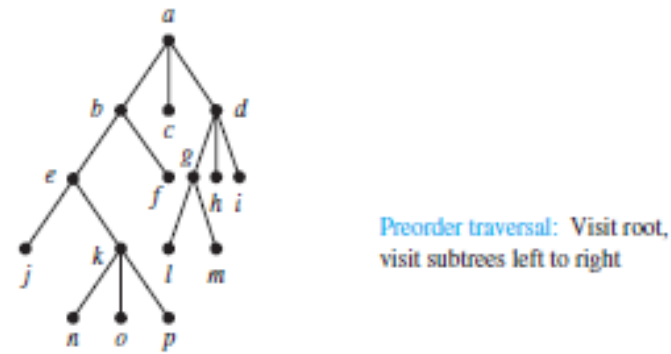
Sumber: Rosen

Contoh: Lakukan traversal *preorder*, *inorder*, dan *postorder* pada pohon 3-ary berikut:



Sumber: Rosen

Preorder:



Sumber: Rosen

FIGURE 4 The Preorder Traversal of T .

Inorder:

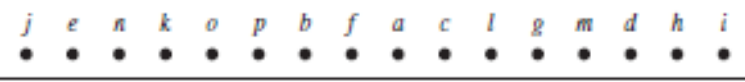
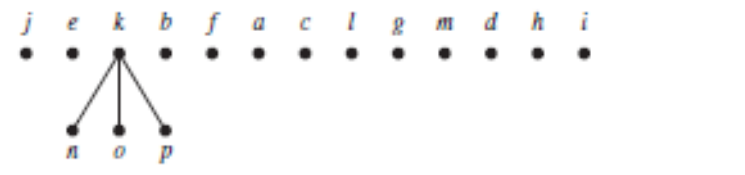
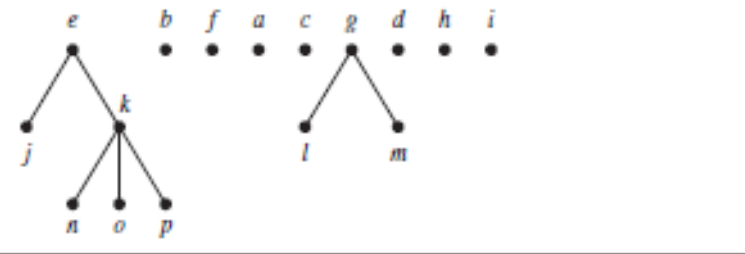
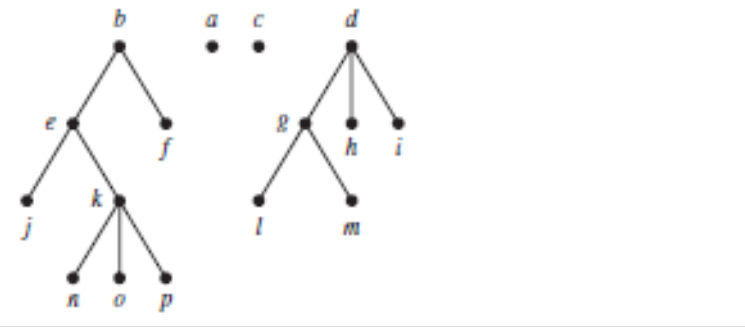
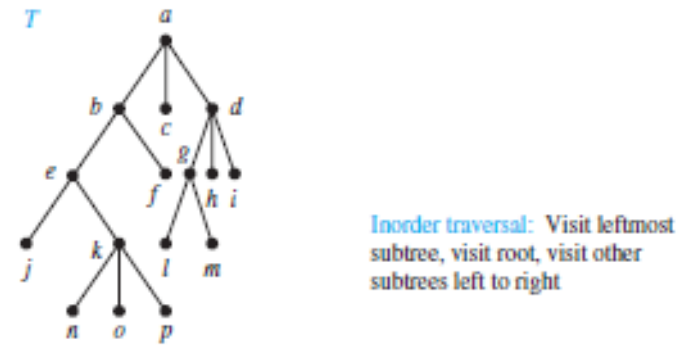
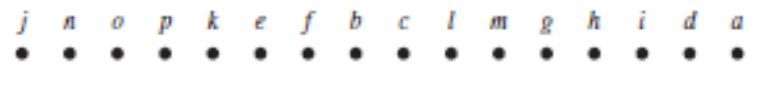
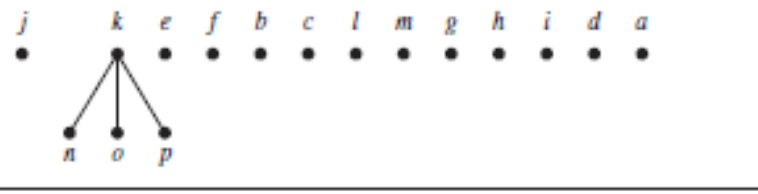
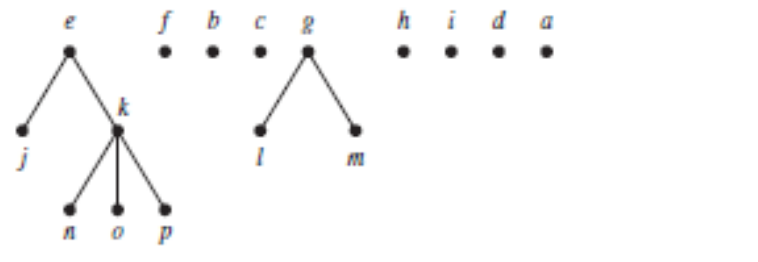
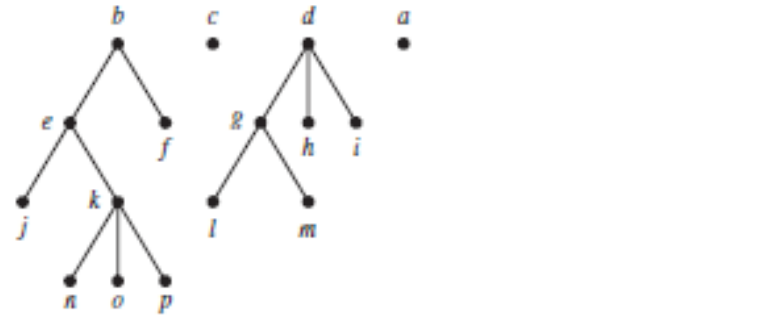
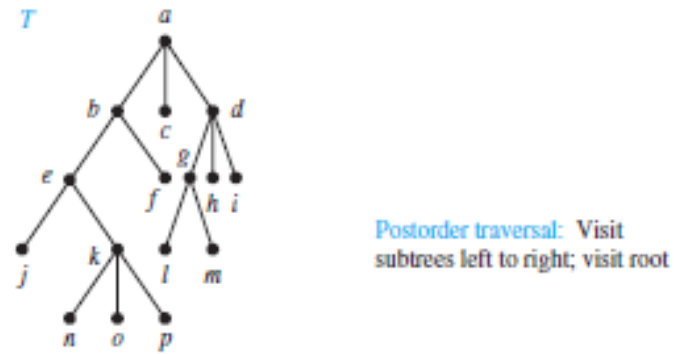


FIGURE 6 The Inorder Traversal of *T*.

Sumber: Rosen

Postorder:

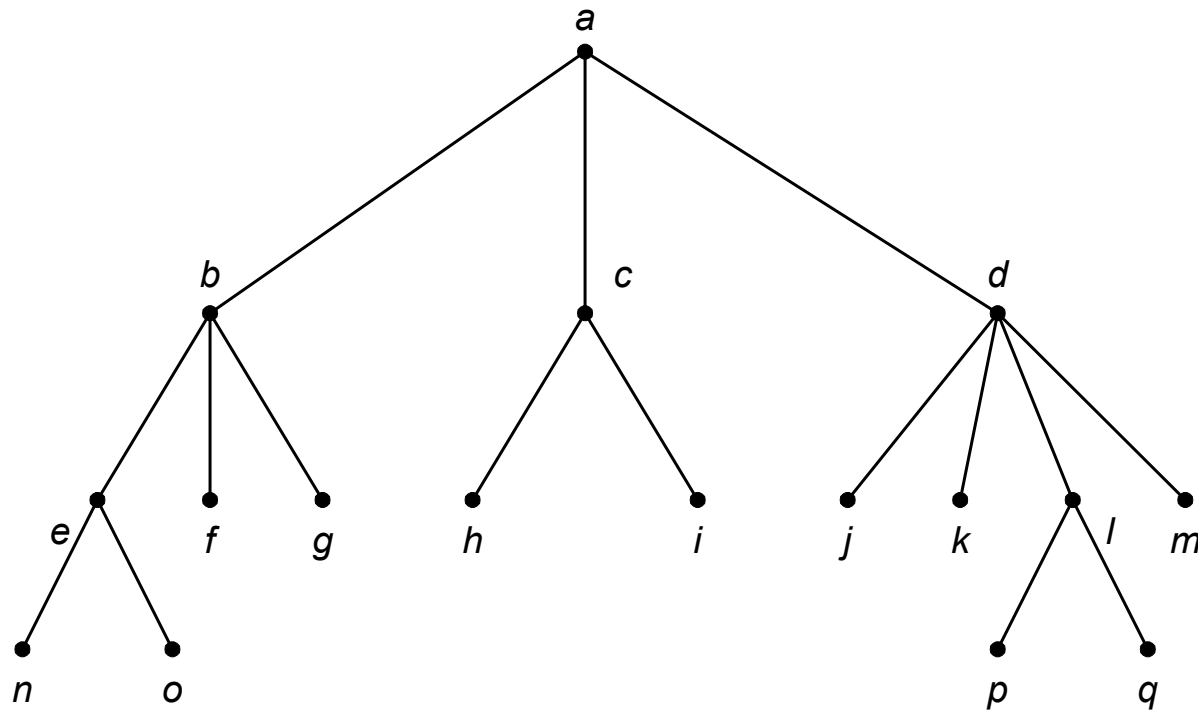


Sumber: Rosen

FIGURE 8 The Postorder Traversal of *T*.

Latihan

Tentukan hasil kunjungan *preorder*, *inorder*, dan *postorder* pada pohon 4-*ary* berikut ini:



TAMAT