PENGANTAR POHON (TREE)

PERTEMUAN 15

Egi Safitri

INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA

Bahasan

1. Definisi Pohon (Tree)

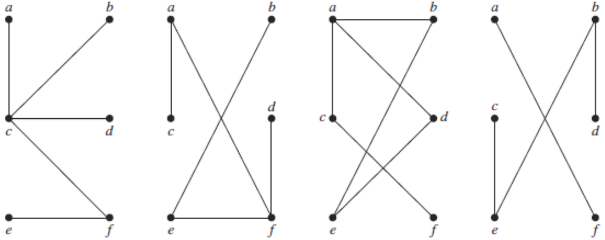
2. Pohon Berakar (Rooted Tree)

Definisi Pohon (*Tree*)

Pohon (*Tree*)

Pohon (*tree*) adalah graf tak berarah yang **terhubung** dan **tidak** memiliki sirkuit sederhana.

Yang manakah di antara graf-graf berikut yang merupakan pohon?



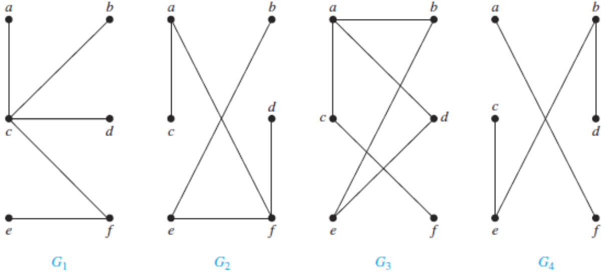
G_1 G_2 G_3 G_4

Definisi Pohon (*Tree*)

Pohon (*Tree*)

Pohon (*tree*) adalah graf tak berarah yang **terhubung** dan **tidak** memiliki sirkuit sederhana.

Yang manakah di antara graf-graf berikut yang merupakan pohon?



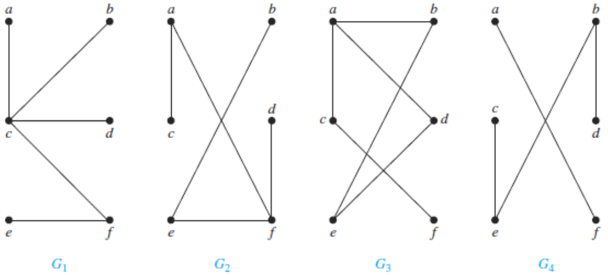
Graf G_1 dan G_2 adalah pohon, karena terhubung dan tidak memiliki sirkuit sederhana.

Definisi Pohon (*Tree*)

Pohon (*Tree*)

Pohon (*tree*) adalah graf tak berarah yang **terhubung** dan **tidak** memiliki sirkuit sederhana.

Yang manakah di antara graf-graf berikut yang merupakan pohon?



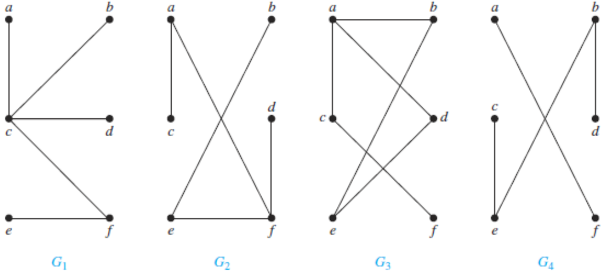
Graf G_1 dan G_2 adalah pohon, karena terhubung dan tidak memiliki sirkuit sederhana. Graf G_3 bukan pohon karena memiliki sirkuit $\langle a, b, e, d, a \rangle$.

Definisi Pohon (*Tree*)

Pohon (*Tree*)

Pohon (*tree*) adalah graf tak berarah yang **terhubung** dan **tidak** memiliki sirkuit sederhana.

Yang manakah di antara graf-graf berikut yang merupakan pohon?



Graf G_1 dan G_2 adalah pohon, karena terhubung dan tidak memiliki sirkuit sederhana. Graf G_3 bukan pohon karena memiliki sirkuit $\langle a, b, e, d, a\rangle$. Graf G_4 bukan pohon karena tidak terhubung, simpul a tidak terhubung dengan simpul c.

Teorema

Sebuah graf merupakan pohon jika dan hanya jika setiap simpul pada graf tersebut dihubungkan oleh sebuah lintasan sederhana yang unik (tunggal).

Definisi (Jembatan (*bridge*))

Suatu sisi pada graf G dikatakan jembatan (*bridge*) apabila penghapusan sisi tersebut menyebabkan G menjadi tak terhubung.

Teorema

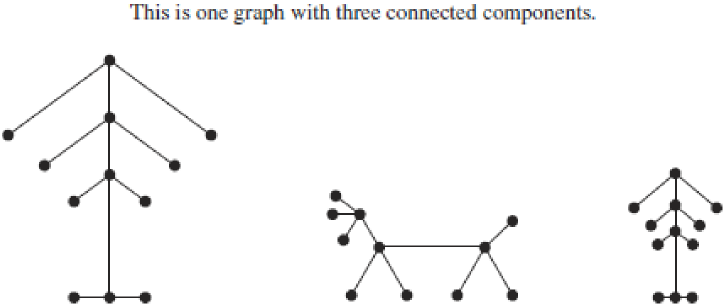
Sebuah graf merupakan pohon jika dan hanya jika setiap sisi pada graf tersebut merupakan jembatan.

Definisi Hutan (*Forest*)

Hutan (*Forest*)

Hutan (*forest*) adalah graf tak berarah yang **tidak** memiliki sirkuit sederhana. Hutan dapat memuat beberapa pohon.

Berikut adalah contoh hutan.



Sifat-sifat Pohon

Teorema

Misalkan $G = (V, E)$ adalah graf tak berarah sederhana, maka setiap pernyataan berikut ekuivalen:

1. G adalah pohon,
2. setiap pasang simpul $u, v \in V$ di G terhubung oleh sebuah lintasan sederhana yang unik (tunggal),
3. G terhubung dan memenuhi $|E| = |V| - 1$,
4. G tidak memiliki sirkuit sederhana dan memenuhi $|E| = |V| - 1$,
5. G tidak memiliki sirkuit sederhana dan penambahan sembarang sisi pada G menyebabkan G memiliki tepat satu sirkuit sederhana,
6. setiap sisi pada G adalah jembatan.

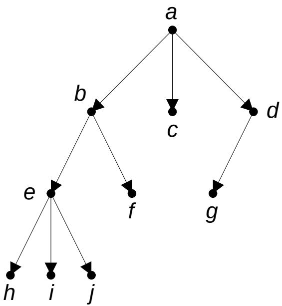
Bahasan

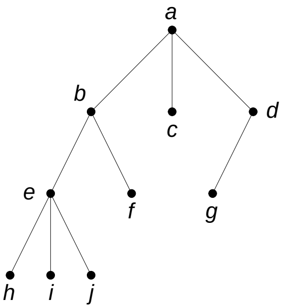
Pohon Berakar (*Rooted Tree*)

Pohon Berakar (*Rooted Tree*)

Pohon berarakar (*rooted tree*) adalah pohon yang sebuah simpulnya dijadikan akar dan setiap sisinya diberi arah sehingga menjauhi akar tersebut. Tanda arah/ panah pada pohon berakar dapat dibuang bila akar dari pohon tersebut sudah jelas.

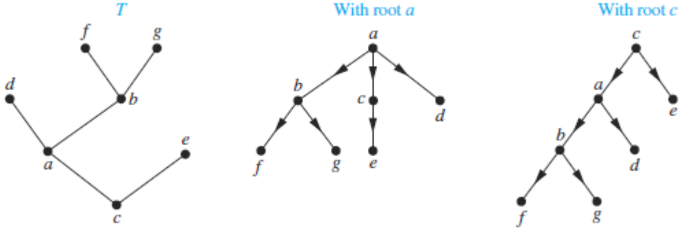
Berikut adalah ilustrasi pohon berakar dengan akar simpul a.



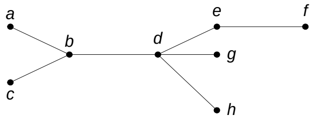


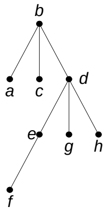
Pemilihan Akar (*Root*)

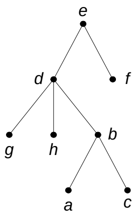
Diberikan sebuah pohon, pohon berakar dari pohon tersebut dapat berbeda, bergantung pada simpul yang dipilih sebagai akar.



Pada gambar tengah, simpul yang dipilih menjadi akar adalah simpul a, sedangkan pada gambar kanan, simpul yang dipilih menjadi akar adalah simpul c.

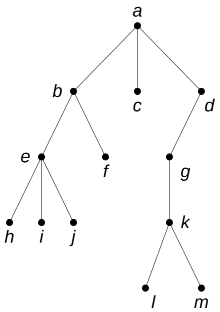






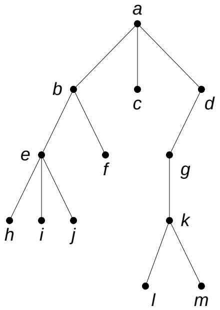
Pada gambar tengah, simpul yang dipilih menjadi akar adalah simpul b, sedangkan pada gambar kanan, simpul yang dipilih menjadi akar adalah simpul e.

Orangtua (*Parent*), Anak (*Child/ Children*), dan Lintasan



Pada pohon berakar di atas, kita memiliki:

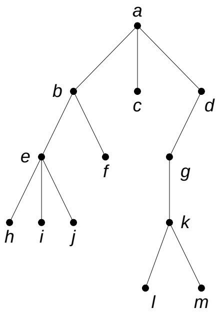
Orangtua (*Parent*), Anak (*Child/ Children*), dan Lintasan



Pada pohon berakar di atas, kita memiliki:

1. simpul b, c, dan d adalah anak-anak (*children*) dari a, masing-masing dari b, c, dan d adalah anak (*child*) dari a;

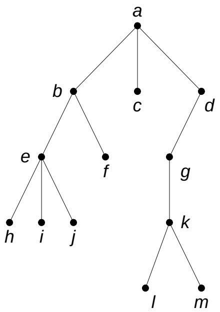
Orangtua (*Parent*), Anak (*Child/ Children*), dan Lintasan



Pada pohon berakar di atas, kita memiliki:

1. simpul b, c, dan d adalah anak-anak (*children*) dari a, masing-masing dari b, c, dan d adalah anak (*child*) dari a;
2. simpul a adalah orangtua (*parent*) dari b, c, maupun d;

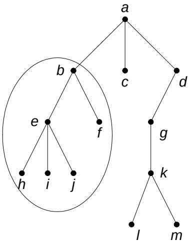
Orangtua (*Parent*), Anak (*Child/ Children*), dan Lintasan



Pada pohon berakar di atas, kita memiliki:

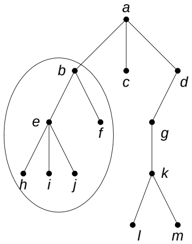
1. simpul b, c, dan d adalah anak-anak (*children*) dari a, masing-masing dari b, c, dan d adalah anak (*child*) dari a;
2. simpul a adalah orangtua (*parent*) dari b, c, maupun d;
3. lintasan dari a ke j hanya ada satu (tunggal), yaitu $\langle a, b, e, j \rangle$ dengan panjang 3.

Saudara Kandung (*Sibling*), Sepupu (*Cousin*), dan Subpohon (*Subtree*)



Pada pohon berakar di atas, kita memiliki:

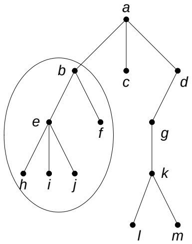
Saudara Kandung (*Sibling*), Sepupu (*Cousin*), dan Subpohon (*Subtree*)



Pada pohon berakar di atas, kita memiliki:

1. e adalah saudara kandung (*sibling*) dari f, dan sebaliknya, karena e dan f memiliki orangtua yang sama, yaitu b;

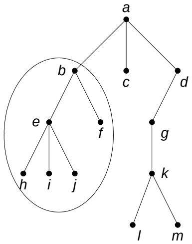
Saudara Kandung (*Sibling*), Sepupu (*Cousin*), dan Subpohon (*Subtree*)



Pada pohon berakar di atas, kita memiliki:

1. e adalah saudara kandung (*sibling*) dari f, dan sebaliknya, karena e dan f memiliki orangtua yang sama, yaitu b;
2. e bukan saudara kandung dari g (dan sebaliknya), karena orangtua mereka berbeda;

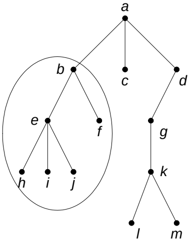
Saudara Kandung (*Sibling*), Sepupu (*Cousin*), dan Subpohon (*Subtree*)



Pada pohon berakar di atas, kita memiliki:

1. e adalah saudara kandung (*sibling*) dari f, dan sebaliknya, karena e dan f memiliki orangtua yang sama, yaitu b;
2. e bukan saudara kandung dari g (dan sebaliknya), karena orangtua mereka berbeda;
3. meskipun e bukan saudara kandung g, e adalah sepupu (*cousin*) dari g, karena orangtua e dan orangtua g memiliki orangtua yang sama, yaitu a;

Saudara Kandung (*Sibling*), Sepupu (*Cousin*), dan Subpohon (*Subtree*)



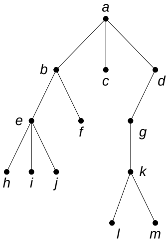
Pada pohon berakar di atas, kita memiliki:

1. e adalah saudara kandung (*sibling*) dari f, dan sebaliknya, karena e dan f memiliki orangtua yang sama, yaitu b;
2. e bukan saudara kandung dari g (dan sebaliknya), karena orangtua mereka berbeda;
3. meskipun e bukan saudara kandung g, e adalah sepupu (*cousin*) dari g, karena orangtua e dan orangtua g memiliki orangtua yang sama, yaitu a;
4. subgraf yang dilingkari dengan "akar" b dikatakan subpohon (*subtree*) dari pohon dengan akar a.

Derajat (*Degree*) pada Pohon Berakar

Definisi (Derajat pada pohon berakar)

Derajat dari sebuah simpul pada pohon berakar adalah banyak anak atau banyak subpohon pada simpul tersebut. Derajat dari sebuah pohon adalah derajat terbesar dari seluruh derajat simpul yang ada.



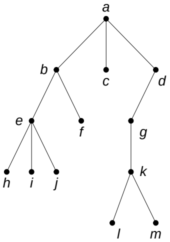
Pada pohon berakar di atas, kita memiliki:

1. $\deg(a) =$

Derajat (*Degree*) pada Pohon Berakar

Definisi (Derajat pada pohon berakar)

Derajat dari sebuah simpul pada pohon berakar adalah banyak anak atau banyak subpohon pada simpul tersebut. Derajat dari sebuah pohon adalah derajat terbesar dari seluruh derajat simpul yang ada.



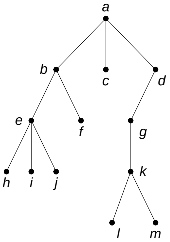
Pada pohon berakar di atas, kita memiliki:

1. $\deg(a) = 3$, $\deg(b) =$

Derajat (*Degree*) pada Pohon Berakar

Definisi (Derajat pada pohon berakar)

Derajat dari sebuah simpul pada pohon berakar adalah banyak anak atau banyak subpohon pada simpul tersebut. Derajat dari sebuah pohon adalah derajat terbesar dari seluruh derajat simpul yang ada.



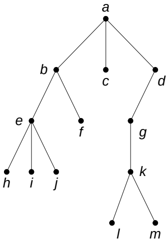
Pada pohon berakar di atas, kita memiliki:

1. $\deg(a) = 3$, $\deg(b) = 2$, $\deg(c) =$

Derajat (*Degree*) pada Pohon Berakar

Definisi (Derajat pada pohon berakar)

Derajat dari sebuah simpul pada pohon berakar adalah banyak anak atau banyak subpohon pada simpul tersebut. Derajat dari sebuah pohon adalah derajat terbesar dari seluruh derajat simpul yang ada.



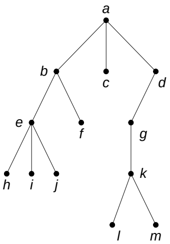
Pada pohon berakar di atas, kita memiliki:

1. $\deg(a) = 3$, $\deg(b) = 2$, $\deg(c) = 0$, dan $\deg(d) =$

Derajat (*Degree*) pada Pohon Berakar

Definisi (Derajat pada pohon berakar)

Derajat dari sebuah simpul pada pohon berakar adalah banyak anak atau banyak subpohon pada simpul tersebut. Derajat dari sebuah pohon adalah derajat terbesar dari seluruh derajat simpul yang ada.



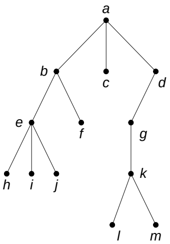
Pada pohon berakar di atas, kita memiliki:

1. $\deg(a) = 3$, $\deg(b) = 2$, $\deg(c) = 0$, dan $\deg(d) = 1$, jadi derajat di sini adalah derajat keluar (ke bawah);

Derajat (*Degree*) pada Pohon Berakar

Definisi (Derajat pada pohon berakar)

Derajat dari sebuah simpul pada pohon berakar adalah banyak anak atau banyak subpohon pada simpul tersebut. Derajat dari sebuah pohon adalah derajat terbesar dari seluruh derajat simpul yang ada.



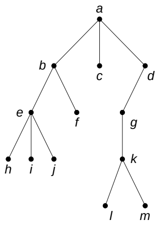
Pada pohon berakar di atas, kita memiliki:

1. $\deg(a) = 3$, $\deg(b) = 2$, $\deg(c) = 0$, dan $\deg(d) = 1$, jadi derajat di sini adalah derajat keluar (ke bawah);
2. derajat pohon di atas adalah 3 karena derajat terbesar dari seluruh derajat simpul yang ada adalah 3 (untuk simpul a dan simpul e).

Daun (*Leaf*) dan Simpul Dalam (*Internal Node*)

Definisi (Daun (*leaf*) dan simpul dalam (*internal node*))

Daun adalah simpul yang berderajat nol (simpul yang tidak memiliki anak). Simpul dalam (*internal node/ internal vertex*) adalah simpul bukan akar yang memiliki anak (simpul tersebut memiliki anak dan orangtua).

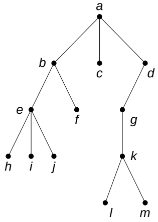


Pada graf berakar di atas:

Daun (*Leaf*) dan Simpul Dalam (*Internal Node*)

Definisi (Daun (*leaf*) dan simpul dalam (*internal node*))

Daun adalah simpul yang berderajat nol (simpul yang tidak memiliki anak).
Simpul dalam (*internal node/ internal vertex*) adalah simpul bukan akar yang memiliki anak (simpul tersebut memiliki anak dan orangtua).



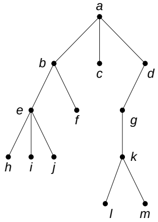
Pada graf berakar di atas:

1. simpul h, i, j, c, l, dan m adalah daun (*leaf*) karena simpul-simpul tersebut tidak memiliki anak,

Daun (*Leaf*) dan Simpul Dalam (*Internal Node*)

Definisi (Daun (*leaf*) dan simpul dalam (*internal node*))

Daun adalah simpul yang berderajat nol (simpul yang tidak memiliki anak). Simpul dalam (*internal node/ internal vertex*) adalah simpul bukan akar yang memiliki anak (simpul tersebut memiliki anak dan orangtua).



Pada graf berakar di atas:

1. simpul h, i, j, c, l, dan m adalah daun (*leaf*) karena simpul-simpul tersebut tidak memiliki anak,
2. simpul b, d, e, g, dan k adalah simpul dalam (*internal node/ internal vertex*) karena simpul-simpul tersebut memiliki anak dan orangtua.

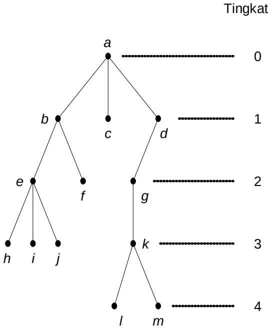
Tingkat (*Level*) dan Tinggi (*Height*)/ Kedalaman (*Depth*)

Definisi (Tingkat/ *level*)

Tingkat/ *level* dari sebuah simpul adalah banyaknya sisi pada sebuah lintasan unik (tunggal) antara simpul tersebut dan akar.

Definisi (Tinggi (*height*) atau kedalaman (*depth*))

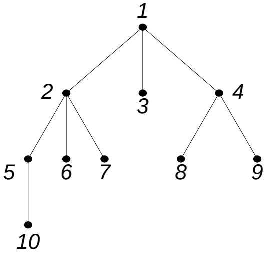
Tinggi (*height*) atau kedalaman (*depth*) dari suatu pohon adalah nilai tingkat (*level*) terbesar yang mungkin ada di pohon tersebut.

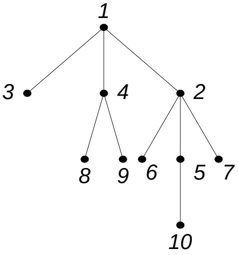


Pohon berakar di atas memiliki empat tingkat.

Pohon Terurut (*Ordered Tree*)

Pohon terurut (*ordered tree*) merupakan pohon yang urutan anak-anak dari setiap simpulnya diperhatikan. Berikut adalah contoh ilustrasi pohon terurut.





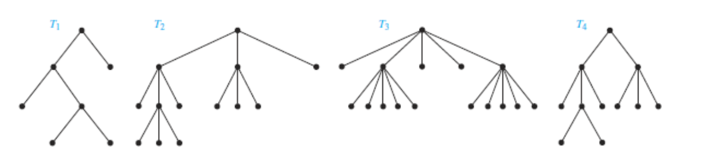
Pohon m-*ary* dan Pohon m-*ary* Teratur

Definisi (Pohon m-*ary* dan m-*ary* penuh/ teratur)

Sebuah pohon dikatakan pohon m-*ary* bila akar dan setiap simpul dalamnya memiliki paling banyak m anak. Sebuah pohon m-*ary* disebut pohon m-*ary* penuh/ teratur (*full/ regular* m-*ary tree*) bila akar dan setiap simpul dalamnya memikiki tepat m anak. Pohon 2-*ary* selanjutnya disebut pohon biner (*binary tree*).

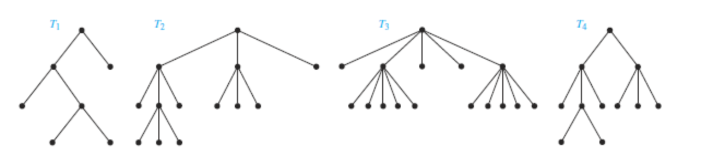
Latihan

Manakah di antara pohon-pohon berikut yang merupakan pohon m-*ary*? Tentukan nilai m-nya!



Latihan

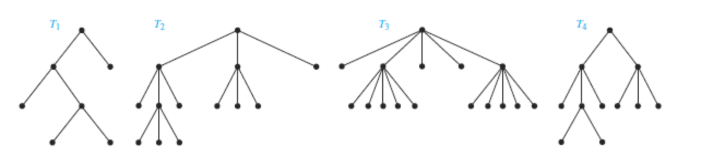
Manakah di antara pohon-pohon berikut yang merupakan pohon m-*ary*? Tentukan nilai m-nya!



1. T_1 adalah pohon biner penuh/ teratur (*full/ regular binary tree*) karena akar dan setiap simpul dalamnya memiliki tepat 2 anak.

Latihan

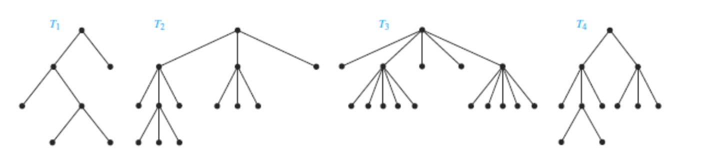
Manakah di antara pohon-pohon berikut yang merupakan pohon m-*ary*? Tentukan nilai m-nya!



1. T_1 adalah pohon biner penuh/ teratur (*full/ regular binary tree*) karena akar dan setiap simpul dalamnya memiliki tepat 2 anak.
2. T_2 adalah pohon 3-*ary* penuh/ teratur karena akar dan setiap simpul dalamnya memiliki tepat 3 anak.

Latihan

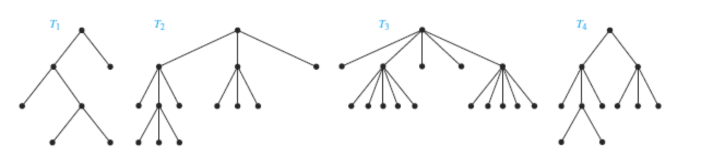
Manakah di antara pohon-pohon berikut yang merupakan pohon m-*ary*? Tentukan nilai m-nya!



1. T_1 adalah pohon biner penuh/ teratur (*full/ regular binary tree*) karena akar dan setiap simpul dalamnya memiliki tepat 2 anak.
2. T_2 adalah pohon 3-*ary* penuh/ teratur karena akar dan setiap simpul dalamnya memiliki tepat 3 anak.
3. T_3 adalah pohon 5-*ary* penuh/ teratur karena akar dan setiap simpul dalamnya memiliki tepat 5 anak.

Latihan

Manakah di antara pohon-pohon berikut yang merupakan pohon m-*ary*? Tentukan nilai m-nya!



1. T_1 adalah pohon biner penuh/ teratur (*full/ regular binary tree*) karena akar dan setiap simpul dalamnya memiliki tepat 2 anak.
2. T_2 adalah pohon 3-*ary* penuh/ teratur karena akar dan setiap simpul dalamnya memiliki tepat 3 anak.
3. T_3 adalah pohon 5-*ary* penuh/ teratur karena akar dan setiap simpul dalamnya memiliki tepat 5 anak.
4. T_4 adalah pohon 3-*ary*, namun bukan pohon 3-*ary* penuh/ teratur karena akarnya hanya memiliki 2 anak dan terdapat simpul dalam yang hanya memiliki 2 anak.

Pohon m-*ary* Seimbang (*Balanced* m-*ary Tree*)

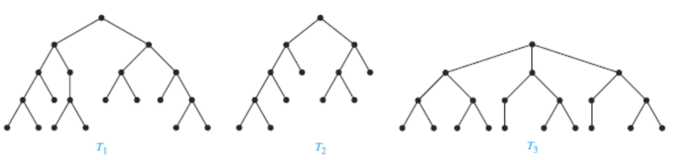
Definisi

Sebuah pohon berakar m-*ary* dengan h tingkat dikatakan pohon m-*ary* seimbang (*balanced* m-*ary tree*) bila semua daunnya berada di tingkat h atau $h-1$.

Artinya pada pohon m-ary seimbang semua daunnya berada pada tingkat yang sama atau maksimal berbeda satu tingkat.

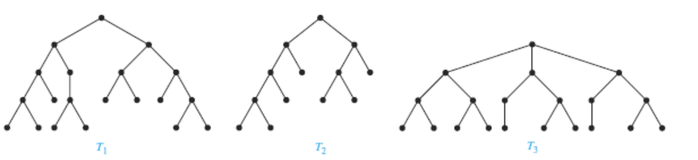
Latihan

Manakah di antara pohon-pohon berikut yang merupakan pohon m-*ary* seimbang?



Latihan

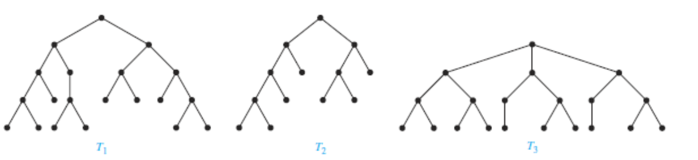
Manakah di antara pohon-pohon berikut yang merupakan pohon m-*ary* seimbang?



1. T_1 adalah pohon biner seimbang karena setiap daunnya berada pada tingkat 4 atau 3.

Latihan

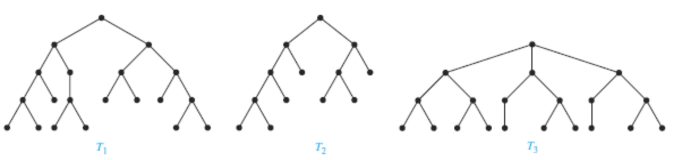
Manakah di antara pohon-pohon berikut yang merupakan pohon m-*ary* seimbang?



1. T_1 adalah pohon biner seimbang karena setiap daunnya berada pada tingkat 4 atau 3.
2. T_2 bukan pohon biner seimbang karena daunnya berada di tingkat 4, 3, atau 2.

Latihan

Manakah di antara pohon-pohon berikut yang merupakan pohon m-*ary* seimbang?



1. T_1 adalah pohon biner seimbang karena setiap daunnya berada pada tingkat 4 atau 3.
2. T_2 bukan pohon biner seimbang karena daunnya berada di tingkat 4, 3, atau 2.
3. T_3 adalah pohon 3-*ary* seimbang karena semua daunnya berada pada tingkat 3.

Beberapa Teorema Terkait Pohon m-*ary*

Teorema

Paling banyak terdapat m^h daun pada sebuah pohon m-*ary* dengan tinggi h. Jika pohon tersebut adalah pohon penuh/ teratur, banyaknya daun maksimal juga m^h.

Teorema

Banyak maksimal simpul pada tingkat/ kedalaman t dari sebuah pohon biner m-*ary* penuh/ teratur dan seimbang adalah m^t.

Teorema

Banyak maksimal seluruh simpul dari sebuah pohon biner m-*ary* penuh/ teratur dan seimbang dengan tinggi h adalah

$$m^0 + m^1 + m^2 + \cdots + m^{h-1} + m^h =$$

Beberapa Teorema Terkait Pohon m-*ary*

Teorema

Paling banyak terdapat m^h daun pada sebuah pohon m-*ary* dengan tinggi h. Jika pohon tersebut adalah pohon penuh/ teratur, banyaknya daun maksimal juga m^h.

Teorema

Banyak maksimal simpul pada tingkat/ kedalaman t dari sebuah pohon biner m-*ary* penuh/ teratur dan seimbang adalah m^t.

Teorema

Banyak maksimal seluruh simpul dari sebuah pohon biner m-*ary* penuh/ teratur dan seimbang dengan tinggi h adalah

$$m^0 + m^1 + m^2 + \cdots + m^{h-1} + m^h = \frac{m^{h+1} - 1}{m - 1}.$$