

TREE (Pohon)

Dalam struktur data tree atau pohon memegang peranan yang cukup penting karena struktur ini biasanya digunakan terutama untuk menyajikan data yang mengandung hubungan hirarikal antara elemen-elemennya.

Tree atau Pohon adalah salah satu graph terhubung yang tidak mengandung sirkuit. Karena merupakan graph terhubung, maka pada pohon selalu terdapat path atau jalur yang menghubungkan setiap dua simpul dalam pohon.

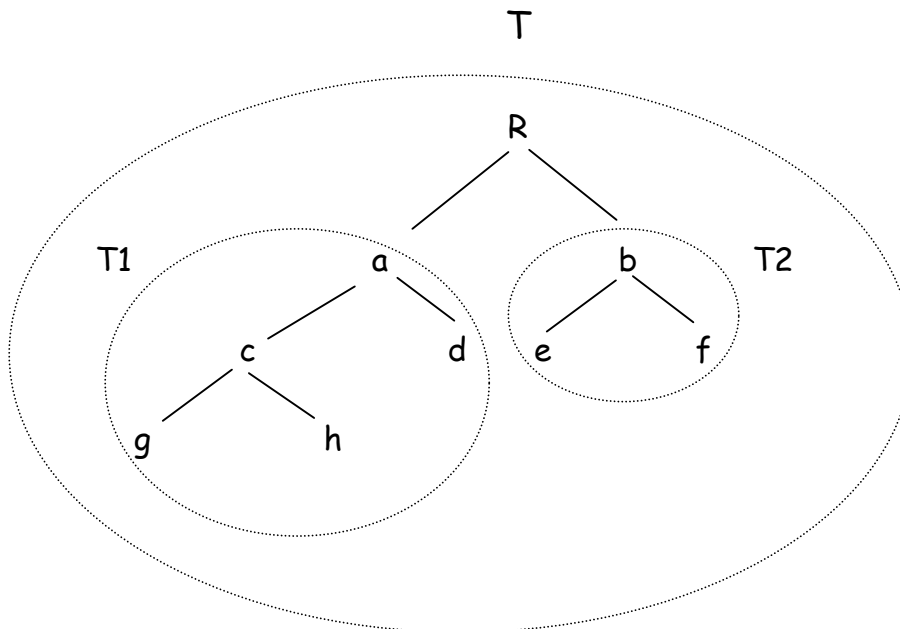
Sifat utama sebuah pohon biner :

1. Jika pohon mempunyai simpul sebanyak n , maka banyaknya ruas atau edge adalah $(n-1)$.
2. Mempunyai simpul khusus yang disebut AKAR atau ROOT, jika simpul tersebut memiliki derajat ke luar ≥ 0 , dan derajat masuk = 0.
3. Mempunyai simpul yang disebut sebagai DAUN atau LEAF, jika simpul tersebut berderajat keluar = 0 dan berderajat masuk = 1.
4. Setiap simpul mempunyai TINGKATAN atau LEVEL, yang dimulai dari root, yang levelnya = 0, sampai dengan level n pada daun paling bawah. Simpul yang mempunyai level yang sama disebut Bersaudara atau Brother atau Sibling.
5. Pohon mempunyai KETINGGIAN atau KEDALAMAN atau HEIGHT, yang merupakan level tertinggi + 1.
6. Pohon mempunyai BERAT atau WEIGHT yang merupakan banyaknya *daun* pada pohon.

Binary Tree

Definisi : Binary tree T didefinisikan sebagai himpunan hingga elemen yang disebut simpul atau node, sedemikian sehingga :

- (i) T adalah hampa (disebut Null tree atau Empty tree) atau
- (ii) T mengandung simpul R yang dinamakan akar (Root) dari T dan simpul-simpul sisanya membentuk pasangan binary tree yang disjoint (saling lepas) T_1 dan T_2 .



Jika T mengandung akar R , maka dua tree T_1 dan T_2 masing-masing disebut Subtree kiri dan Subtree kanan dari R .

Jika T , tidak kosong, maka akar dari T_1 dinamakan left Successor dari R dan jika T_2 tidak kosong, maka akar dari T_2 dinamakan right Successor dari R .

Dari gambar diatas :

a adalah left Successor dari R

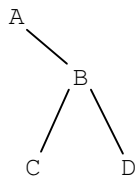
b adalah right Successor dari R

Sebarang node dalam binary tree T akan mempunyai 0 atau 1 atau 2 buah Successor. Untuk node yang tidak mempunyai Successor dinamakan terminal node.

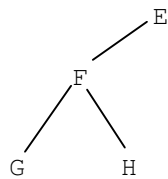
Istilah (terminologi) dalam Binary Tree T

1. Dua buah tree T dan T* dikatakan Similar jika keduanya mempunyai Struktur (bentuk) yang sama.

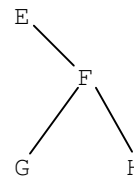
contoh :



(a)



(b)



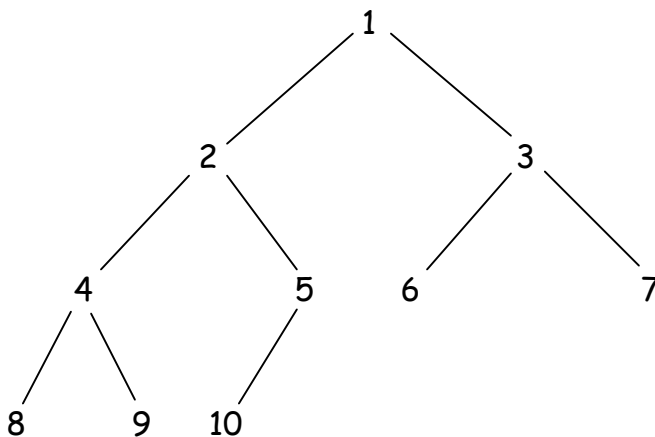
(c)

(a) dan (c) Similar

2. Complete Binary Tree

Definisi : Sebuah tree dikatakan *Complete Binary tree* jika semua level (kecuali level terakhir) mempunyai jumlah node maksimum (2^r) dan bila semua simpul pada tingkat terakhir muncul dibagian kiri pohon.

Catatan : Untuk setiap level r mempunyai paling banyak 2^r node



Dengan memberi label seperti diatas, maka untuk node K mempunyai left child : $2*K$ dan Rightchild : $2*K+1$, serta node K mempunyai father $[K/2]$.

Ketinggian/Kedalaman atau Depth (D_n) dari Complete Binary tree T_n dengan n node adalah

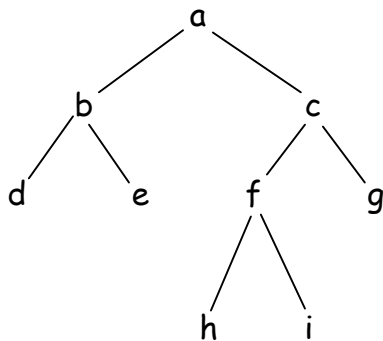
$$D_n = \text{INT}(\sqrt{2} \log N + 1)$$

3. Extended Binary tree : 2 - tree

Definisi Sebuah binary tree T dikatakan sebagai 2-tree atau extended binary tree jika setiap node N mempunyai 0 atau 2 buah Child.

Catatan : - Node dengan 2 buah Child dikatakan internal node

- Node dengan 0 Child dikatakan External node



d, e, h, i, g : adalah external node

Aplikasi penting dari 2-tree adalah untuk menyajikan suatu ekspresi aritmetik yang mengandung operasi biner. External node digunakan untuk menyajikan operand dan Internal node disajikan sebagai operator yang bekerja terhadap 2 subpohonnya.

Contoh : ekspresi aritmetik = $(a-b) / ((c+d) * e)$

Penyajian Binary tree dalam memory

a. Penyajian dengan Double Linked List

Penyajian Linked List untuk Binary tree menggunakan array paralel :
DATA, LEFT, RIGHT, dan Variabel Pointer ROOT.

Misalkan Node N berada pada lokasi K, maka

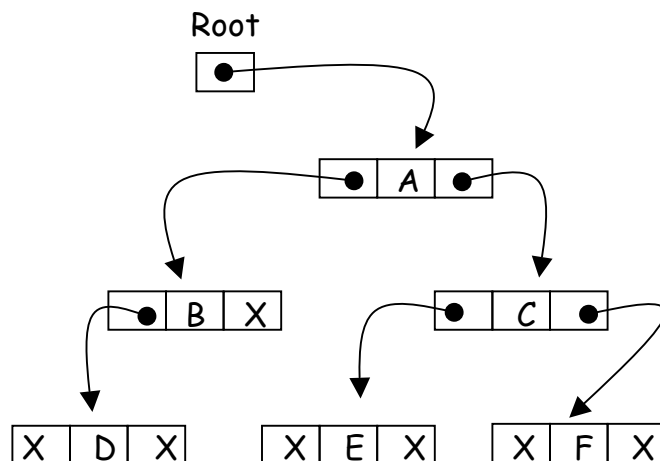
- DATA [K] adalah isi data dari N
- LEFT [K] adalah lokasi left Child dari N
- RIGTH [K] adalah lokasi right Child dari N

contoh :

ARRAY PARALEL

	Data	Left	Right
1	D	0	0
2			
3	B	1	0
4			
5	A	3	7
6			
7	C	8	9
8	E	0	0
9			
10			
11			

BINARY TREE



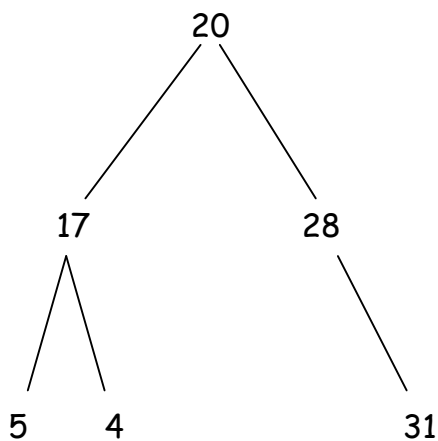
b. Penyajian Secara Sequential

Andaikan T adalah sebuah binary tree, maka penyajian T yang efisien adalah penyajian sequential. Penyajian ini hanya menggunakan single-linear array TREE.

(a) ROOT R dari T disimpan dalam TREE[1]

(b) Jika node N berada pada TREE[K] maka left Child disimpan dalam TREE[2*K] dan right Child disimpan dalam TREE[2*K+1]

contoh :



1	20
2	17
3	28
4	5
5	4
6	
7	31
8	
9	
10	