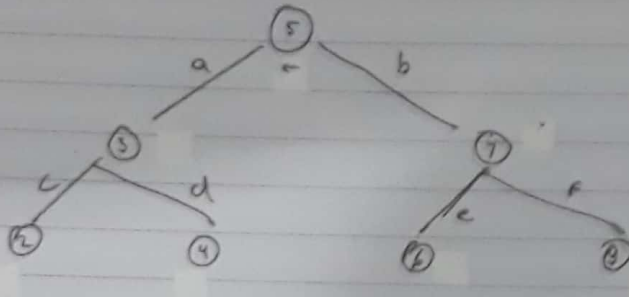


Nicholas (Night Stalker)

1. Linear data structure = untuk mengisi dan mengakses elemen satu hingga seterusnya
 Pengaksesan hingga ke akhir elemen (Lurus DS)
 non-linear = Tidak Perlu mengakses seluruh elemen untuk ke elemen terakhir (Tree, Graph)

2.



5 = Base root

key = atribut untuk membagi tree, dalam hal ini value dari node

edge = garis penghubung (a, b, c, d, e, f)

Siblings = node dengan height sama dan Parents sama. (2 Siblings 4) (6 Siblings 9)
 (3 Siblings 7)

Parent = node yang memiliki keturun. (3 Parent 2 & 4) (7 Parent 6 & 9) (5 Parent 3 & 7)

Child = node hasil percabangan (2 & 4 Child 3) (6 & 9 Child 7) (3 & 7 Child 5)

Leaf = node paling bawah (2, 4, 6, 9)

3. Full binary tree : masing-masing elemen punya 0 atau 2 anak kecuali level paling bawah

Complete Binary tree = Semua level wajib terisi node kecuali level paling bawah

Perfect Binary tree = Semua node wajib terisi 2 node kecuali level paling bawah

4. Balanced binary tree akan terbentuk apabila perbedaan tinggi level paling bawah tidak ada yang lebih dari 1.

5. Maximum element di level k : 2^k

Maximum total element dengan tree k level : $2^{k+1} - 1$

Minimum tinggi tree dgn n node : $\lceil \log_2(n) \rceil$

Maximum tinggi dengan n node : $n - 1$

6. Buat array. Root ada di index 0

anak kiri : $2k+1$

anak kanan : $2k+2$

Cari parents : $(p-1)/2$

7. In Order Successor : Setelah diurutkan seluruh elemen dalam BST, Inorder Successor adalah elemen paling besar terdahulu

In order Predecessor : Setelah diurutkan seluruh elemen BST, Inorder Predecessor adalah elemen paling kecil terdahulu

8

