

1) Linear \rightarrow Struktur data yang memiliki urutan / dalam bentuk berurut seperti array, linked list

Non-linear \rightarrow Struktur data yang tidak memiliki urutan / tidak dalam bentuk berurut seperti tree

2) Base root \rightarrow satu node yang berada di paling atas atau node pertama pada sebuah tree yg tidak kosong

Key \rightarrow atribut dari node yang dijadikan perbandingan nilai seperti integer, double, char [], dll

Edge \rightarrow garis antara dua node yang menghubungkan kedua node tersebut

Parent \rightarrow node yang berada tepat di atas dari node lainnya dan terhubung dengan edge \rightarrow satu level

Child \rightarrow node yang berada tepat satu level di bawah node parent dan terhubung dengan edge

Siblings \rightarrow node yang memiliki parent yang sama

Leaf \rightarrow node yang berada di paling bawah / tidak memiliki child atau edge menuju ke node lain dibawahnya

3) Full \rightarrow setiap node memiliki nol / dua child

Complete \rightarrow setiap node memiliki ~~not / dua~~ child kecuali node yang berada satu level di atas leaf

Perfect \rightarrow setiap node memiliki dua child, kecuali leaf

4) Tree balanced ketika setiap subnode di kiri & kanan memiliki selisih tinggi ~~tdk~~ tidak lebih besar dari satu

5) maks node pada level $k \rightarrow \text{maks} = 2^k$

maks node pada suatu tree dengan level $k \rightarrow \text{maks} = 2^{k+1} - 1$

min. level pada n node $\rightarrow \text{banyak node} = 2^{k+1} - 1$

max level pada n node $\rightarrow n = k+1$

6) Dimulai dari root (node paling atas) sebagai index pertama. Untuk setiap node dibawah root, index nya dapat dituliskan sebagai berikut,

Child kiri = $2p+1$ & Child kanan = $2p+2$

→ p = index parent & jika hanya ada satu child, gunakan child kiri

7) Inoder predecessor → node yang berada di paling kanan dari subtree sebelah kiri

Inoder successor → node yang berada di paling kiri dari subtree sebelah kanan

8) Urutan

① Insert 80

② Insert 30

③ insert 60

④ insert 50

⑤ insert 75

⑥ ~~insert~~ delete 60

⑦ delete 30

⑧ delete 75

⑨ insert 65


⑩ Insert 30

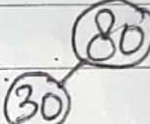
⑪ insert 35

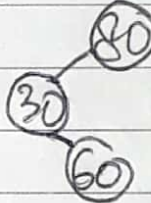
⑫ delete 80

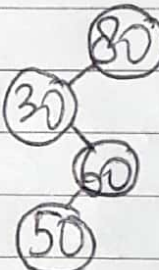
⑬ delete 65

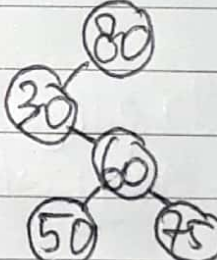
⑭ delete 35

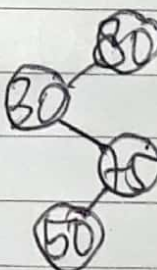
8) ① km tree kosong, maka 80 jadi
base root dari tree → 

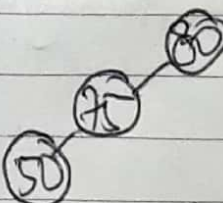
② km $30 < 80$, maka 30 jadi
child kiri dari 80 → 

③ km $60 < 80$, kita cek
 $30 & 60$. km $60 > 30$, maka
60 jadi child kanan 30 → 

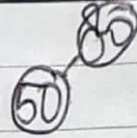
④ $50 < 80$, cek ke kiri (30)
 $50 > 30$, cek ke kanan (60) →
 $50 < 60$, child kiri 60 adalah
50 

⑤ $75 < 80$, cek ke kiri (30)
 $75 > 30$, cek ke kanan (60) →
 $75 > 60$, 75 jadi child kanan
60 

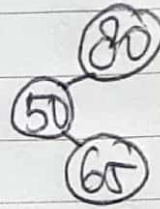
⑥ km 60 punya 2 child,
maka ~~75~~ child kanan menggantikan
posisi parentnya → 

⑦ km 30 punya 1 child,
child menggantikan parent → 

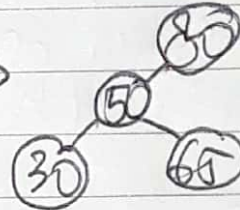
⑧ km 75 punya 1 child,
child menggantikan parent →



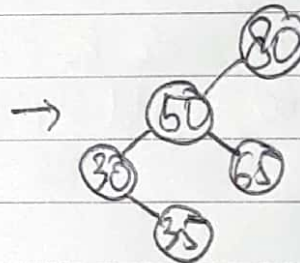
⑨ $65 < 80$, cek kiri (50)
 $65 > 50$, 65 jadi child kanan 50 →



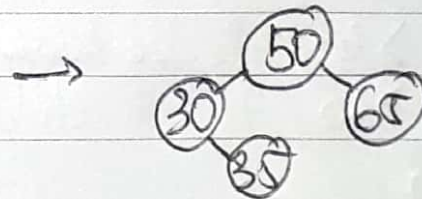
⑩ $30 < 80$, cek kiri (50)
 $30 < 50$, 30 jadi child kiri 50 →



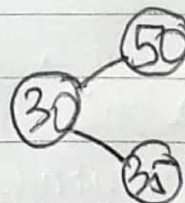
⑪ $35 < 80$, cek kiri (50)
 $35 < 50$, ~~cek~~ cek kiri (30)
 $35 > 30$, 35 jadi child kanan 30 →



⑫ km 80 punya 1 child,
child menggantikan parent →



⑬ km 60 tidak punya child →



⑭ km 35 tidak punya child →

