

B-Tree $m=3$

3. (40 points)

Insert : 15, 30, 20, 40, 10

Delete : 30, 15

Insert : 60, 30, 18, 25, 45

Delete : 20, 10, 40

Insert : 65, 20, 35

1. Insert angka 15 sebagai root

```

graph TD
    15[15]
  
```

2. Insert angka 30, karena node 15 masih bisa diisi, maka 30 masuk ke sebelah kanan 15

```

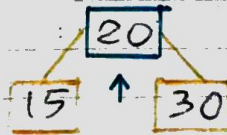
graph TD
    15[15 | 30]
  
```

3. Insert angka 20, terletak diantara 15 dan 30

```

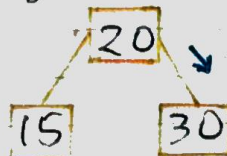
graph TD
    15[15 | 20 | 30]
  
```

4. Karena melanggar aturan (maksimal 2 key) maka lakukan split, 20 ke atas

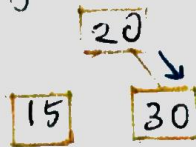


5. Insert 40.

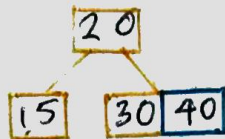
a. Bandingkan dengan 20, lebih besar dari, ke kanan



6. bandingkan dengan 30, lebih besar, ke kanan.

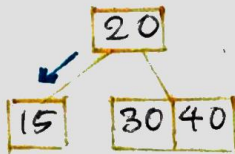


Tapi sudah null, maka cek apakah node key 30 masih bisa diisi? karena masih bisa 40 di sebelah kanan 30

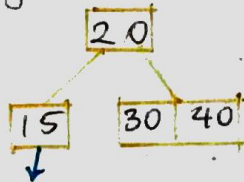


6. Insert 10,

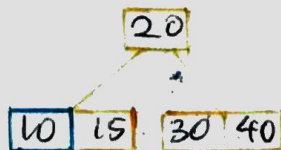
a. bandingkan dengan 20, lebih kecil dari, maka ke kiri



6. bandingkan dengan 15, lebih kecil, ke kiri

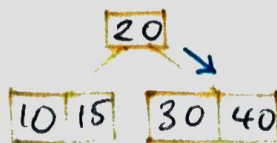


Tapi sudah null maka cek Tempat 15, masih kosong 10 di sebelah kiri 15.

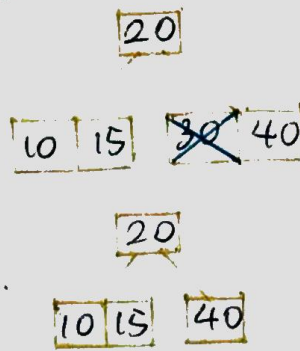


7. delete 30

a. cari 30, lebih besar dari 20, ke kanan.

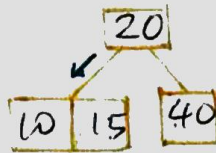


6 traverse array nya - ketemu 30, delete, cek kondisi.

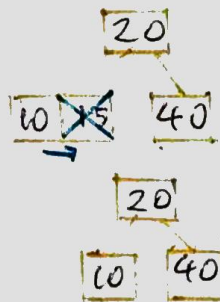


7. delete 15

a. cari 15, lebih kecil dari 20, keluarkan

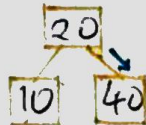


b traverse ke 10, lalu 15, angkut 15 ketemu, delete, cek kondisi.

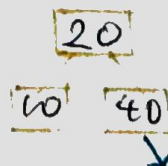


8. Insert 60

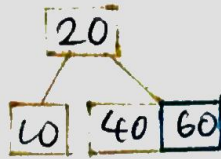
a. bandingkan dengan 20, lebih besar, ke kanan



b bandingkan dengan 40, lebih besar, ke kanan

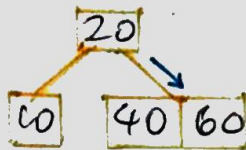


Tetapi null, lalu cari tempat key 40. masih kosong
Maka 60 disebelah kanan 40

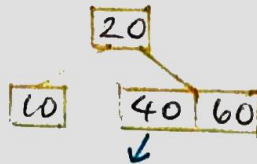


9. Insert 30

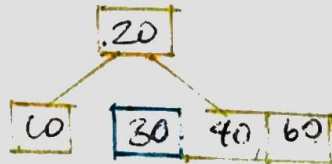
a. bandingkan dengan 20, lebih besar, ke kanan



b. traverse. bertemu dengan 40, lebih kecil, tapi null.



c. setelah dibandingkan, 30 terletak di kiri 40



melanggar kondisi. cari median, lalu ke atas



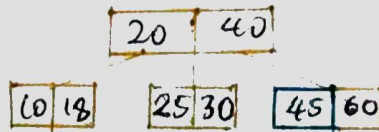
Cek kondisi, masih belum melanggar

12. Insert 45

- a. traverse. bandingkan dengan 20, lebih besar,
traverse kekanan, bandingkan dengan 40, lebih besar. turun
kelan

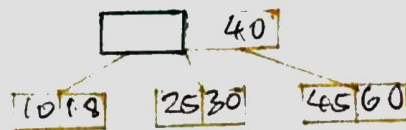
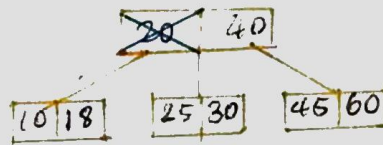


- b. bandingkan dengan 60 lebih besar tapi null. maka insert di
kiri 60

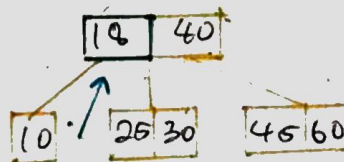


13. Delete 20

- a. Cari 20. Delete

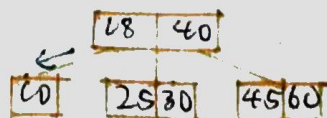


- b. melanggar aturan, cari predecessor = 18. 18 naik

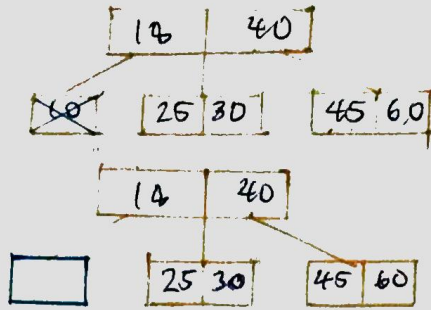


14. Delete 10

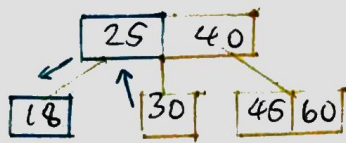
- a. Cari 10. lebih kecil dari 18, ke kiri.



b. ketemu 10.

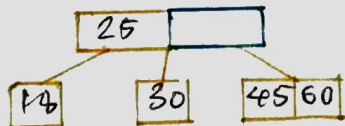
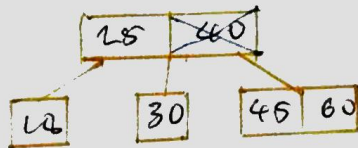


mengikuti aturan, maka cari successor nya 18 = 25. 25 naik
18 turun

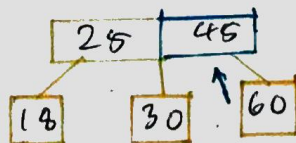


15. Delete 40

a. cari 40, delete

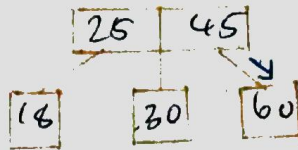


b. Cari predecessor tapi tidak bisa, lalu cari successor
45 naik

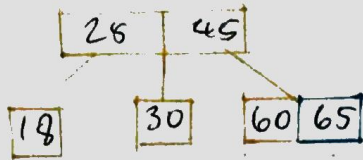


16. Insert 65

- a. 65 lebih besar dari 25 dan 45, ke kanan.



- b. 65 lebih besar dari 60 ke kanan bawah yang sudah null maka tambah disebelah kanan 60

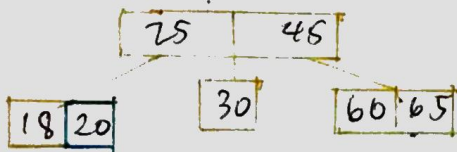


17. Insert 20

- a. 20 lebih kecil dari 25, ke kiri.

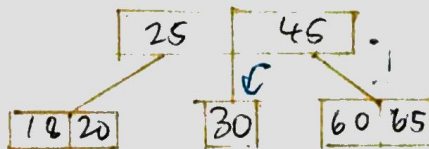


- b. 20 lebih besar dari 18, ke kanan.



18. Insert 35

- a. 35 lebih besar dari 25 dan lebih kecil dari 45, ke kiri, 45.



- b. 35 lebih besar dari 30, ke kanan bawah yang sudah null, maka disebelah 30

