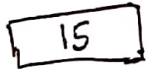


B-Tree, Max Degree = 3

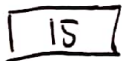
no. 2

Insert 15



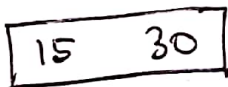
Tree kosong maka 15 dapat langsung masuk ke dalam tree. sebagai root.

Insert 30



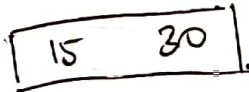
Dicek dahulu apakah sudah melewati maksimum key dari node yang ada, karena ^{key} node hanya 1. Brarti belum maksimum.

Angka / key 30 dapat ditaruh disebelah kanan 15 karena node bersifat sorted



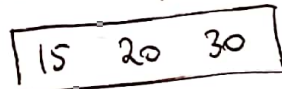
Properti yg tdk boleh dilanggar
Setiap node punya (maks 3 child)
Maximum key dalam node 2
Minimal " " " 1

Insert 20

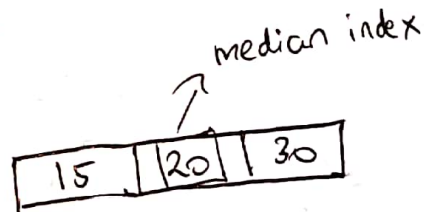


Karena yang mau di insert 20. Jadi posisi 20 lebih besar dari 15 dan lebih kecil dari 30.

Node akan seperti ini :



Tetapi key nya sudah > 2 (besar dari 2) / melewati maksimum, maka Median di split ke atas dan menjadi root.



15 menjadi anak kiri dan 30 menjadi anak kanan. Agar property B-tree tdk terganggu.

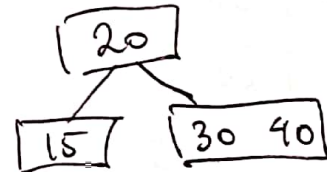


Insert 40

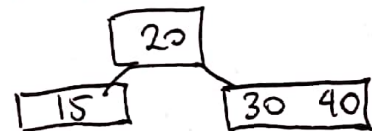
Pertama akan dilihat dari Root, jika lebih < maka ke kiri jika lebih > maka ke kanan.



Dicek apakah node anak kanan key nya sudah maksimal jika belum langsung di insert.



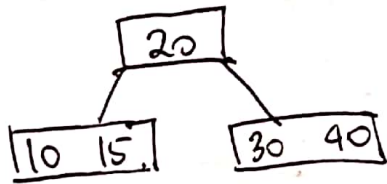
Insert 10



10 < 20, dan node key < max



Delete 30



Sebelum di hapus /delete kita harus mencari dulu posisi yang mau dihapus menggunakan root, karena $30 > \text{root}(20)$ maka posisi 30 berada di child kanan.

Setelah ditemukan tdk dapat langsung dihapus, dicek dahulu apakah key yang tersisa akan $<$ dari minimum key, karena nanti yang tersisa adalah 1, maka masih dalam minimum key, 30 dihapus.



Delete 15

$15 < 20$, berada di node kiri ketemu 15, apakah akan $<$ dari 1. Ternyata tidak 15 dihapus di node kiri.

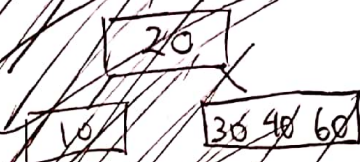


Insert 60



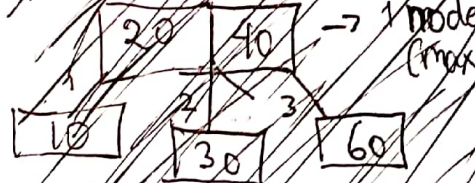
$60 > \text{root}(20)$ maka terletak di sebelah kanan. Serta key $\nleq \text{max}$

~~Insert 30~~



~~Jumlah key di Node kanan sudah $>$ (lebih) dari max / batas~~

~~maka akan dilakukan split median~~



~~* NB: melakukan split tidak boleh melanggar properti B-tree~~

Insert 18



$18 < 20 (\text{root})$ maka terletak dikiri root. Key di kiri $< \text{max}$

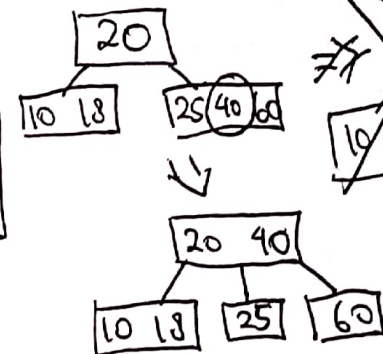


Insert 25

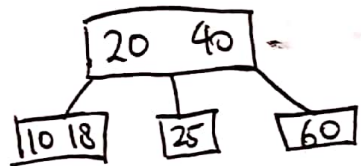
$25 > 20 (\text{root})$ maka disebelah kanan



Tapi keynya $> \text{max}$, jadi kita lakukan split median dan tidak boleh melanggar aturan B-tree



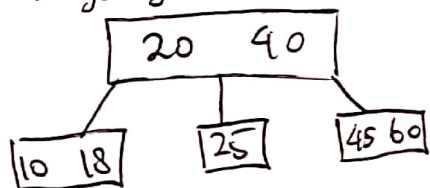
Insert 45



Karena key di node root ada 2 maka Kita Cari sesuai yang mau di insert, jadi ada 3 pilihan:
Kecil dr 20, diantara 20 - 40, besar dari 40.

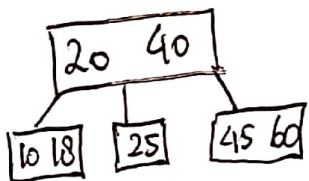
Karena $45 > 40$, maka di letak di kanan 40.

Dicek apakah itu leaf. Jika key \leq max maka boleh langsung insert.

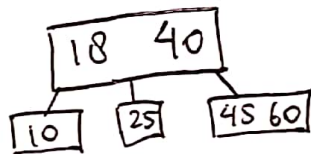


Delete 20

Yang ingin di delete terdapat di root node maka harus kita replace dgn predecessor atau successor, disini akan digunakan pre.

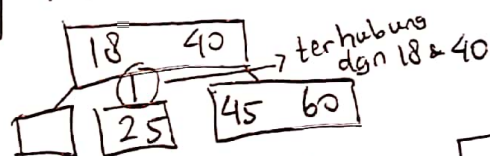


predecessor dari 20 adalah 18, maka nilai 18 akan me-replace 20 dan key 18 akan dihapus



Delete 10

$10 < 18$, disebelah kiri.

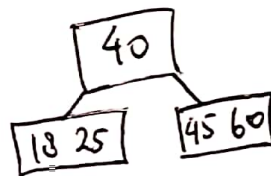


Tapi, node menjadi 0, dimana key min = 1. key

Disini kita melakukan, mengambil key root dan melakukan proses merge



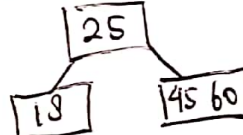
Delete 40



Sama seperti saat delete 20, kita cari predecessor dari yg mau di delete



replace dan hapus key lama

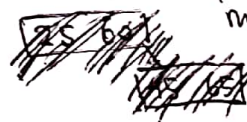


Insert 65

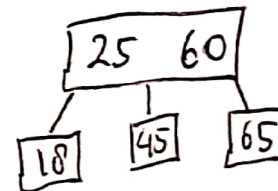
ketika di insert,



key node kanan $>$ max, maka dilakukan split median

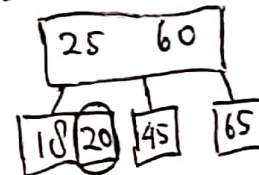


Insert 20



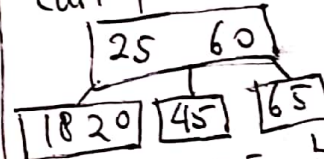
$20 <$ root 25, terletak di node kiri dari root 25.

Karena jumlah key di node kiri $<$ max langsung insert



Insert 35

Cari posisi 35,



Karena 35 berada diantara 25 - 60, maka 35 terletak di kanan 25 dan kiri 60. Node nya $<$ Max Key

