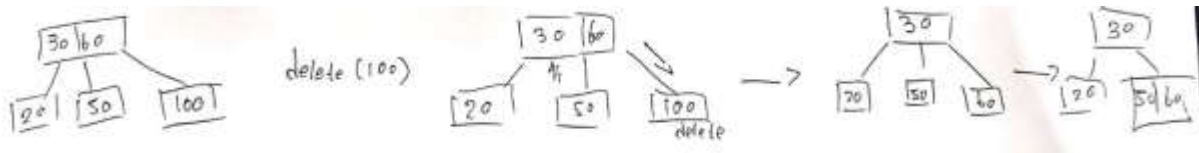


B-Tree

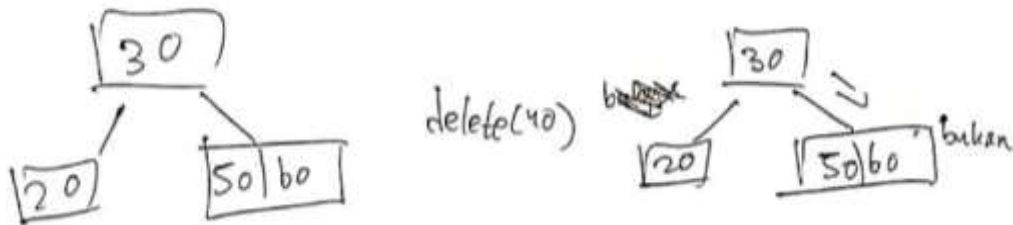
Delete(100 40 25 65 30)



Delete 100

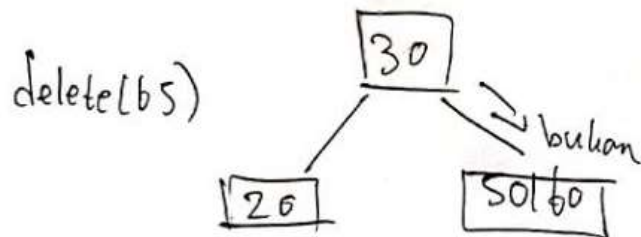
Setelah diremove 100, maka 60 akan menggantikan tempatnya

Dan setelah menggantikan posisi 100, node 50 dan 60 akan bergabung



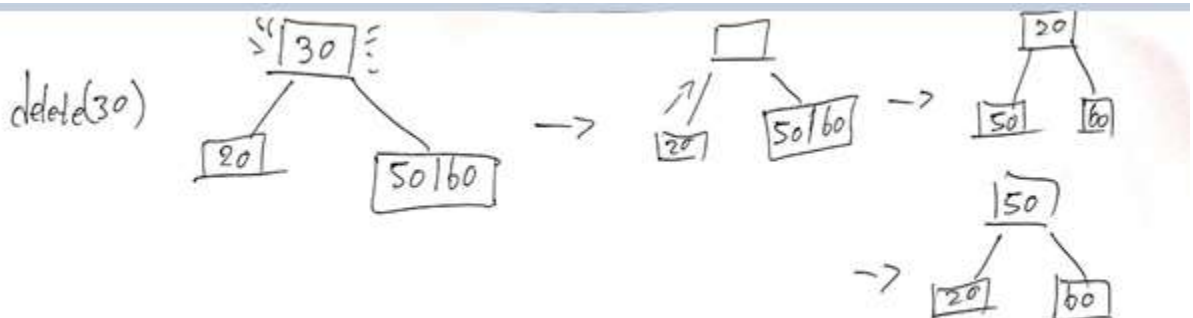
Delete 40

Tidak ada, dicari kesebelah kanan karena lebih besar



Delete 65

Tidak ada, dicari kesebelah kanan karena jika ke arah kiri hanya lebih kecil



Delete 30

Setelah 30 dihapus, maka 20 akan menggantikan posisinya

Namun 50 bernilai > 20 dan < 60 sehingga 50 harus ditengah

50 bertukar posisi dengan 20

Red-Black-Tree

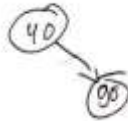
Insert(40, 90, 45, 25, 65)

insert(40)

(40)

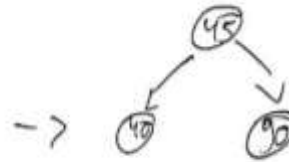
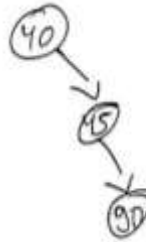
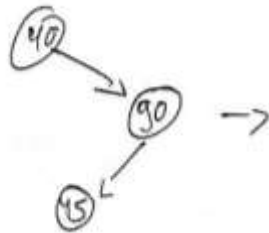
Insert 40, masih kosong, insert pada umumnya seperti biasa

insert(90)



Insert 90, $90 > 40$ sehingga posisi sebelah kanan dan dibawah

insert(45)

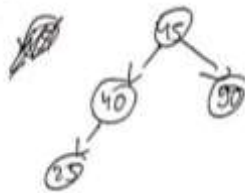


Insert 45, $45 < 90$, sebelah kiri 90

Namun $45 > 40$, inser sebelah kiri 40

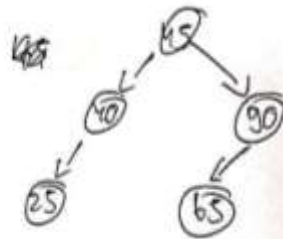
Namun $40 < 45$, 40 sebelah kiri 45

insert(25)



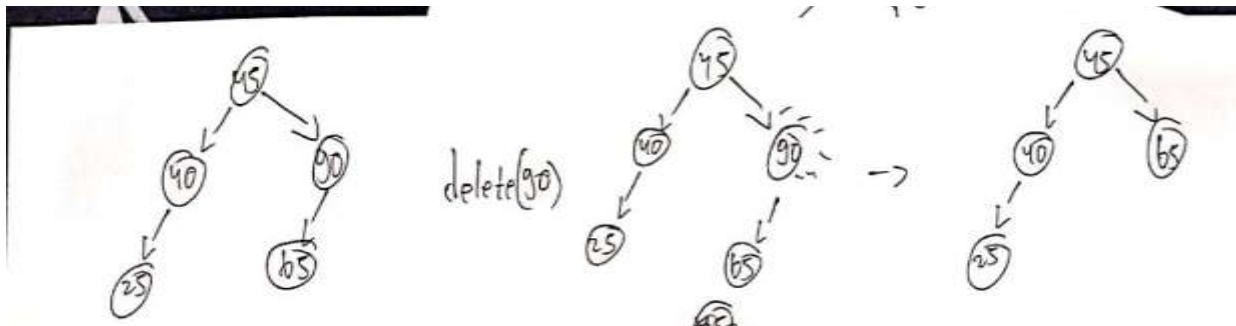
Insert 25, $25 < 40$, sehingga diletakkan pada sebelah kirinya

insert(65)

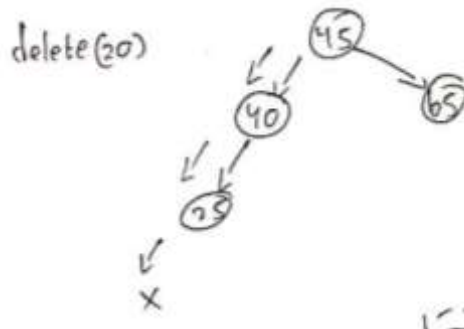


Insert 65, $65 < 95$, letakkan pada sebelah kiri 90

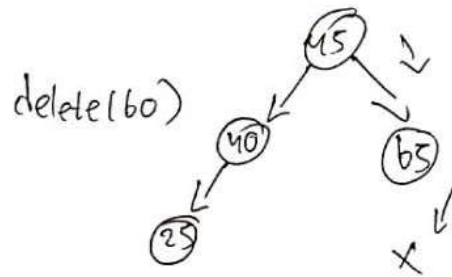
Delete(90 20 60 45 50)



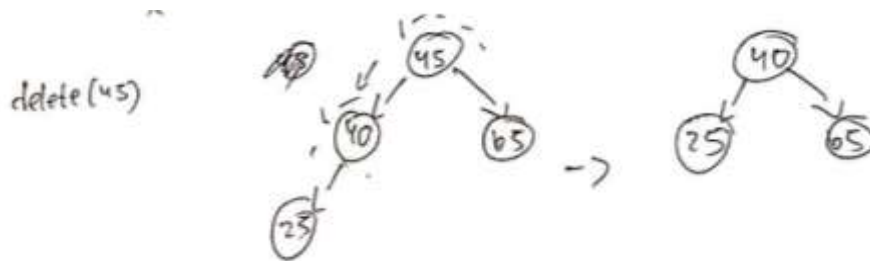
Delete 90, setelah 90 dihapus, maka 65 menempati posisinya



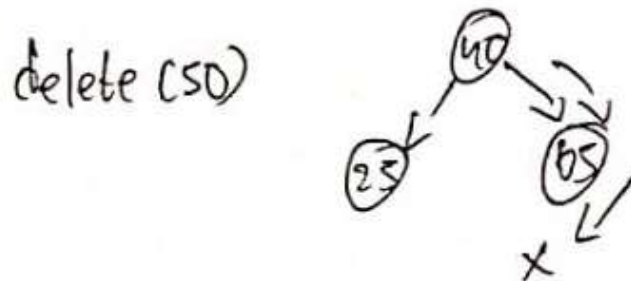
Delete 20, tidak ada yang dihapus, pencarian dilakukan pada sebelah kiri dikarenakan nilai angka lebih kecil



Delete 60, tidak terjadi apapun, pencarian dilakukan pada sebelah kanan dikarenakan sisi kanan lebih besar daripada sisi kiri



Delete 45, setelah 45 dihapus, 40 menempati posisinya



Delete 50, tidak terjadi apapun, pencarian dilakukan pada sebelah kanan dikarenakan sisi kanan lebih besar daripada sisi kiri