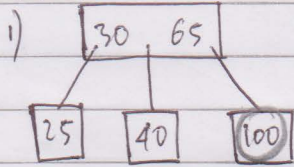
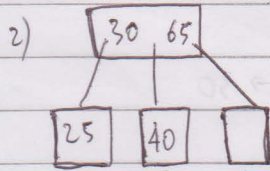


3) Delete : 100, 40, 25, 65, 30

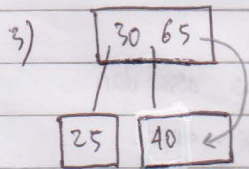
• Delete 100



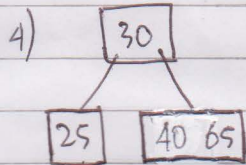
- Mencari key 100
- Traverse dari 65 → 100



- 100 terletak di leaf node dengan kondisi node memiliki key yang berjumlah = minimum key.
- 100 dihapus dan node tsb kosong sekarang.

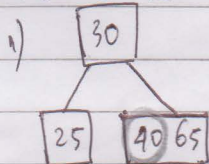


- node kosong akan di merge dengan node kiri / kanan nya.
- node tsb di merge dengan node kirinya.

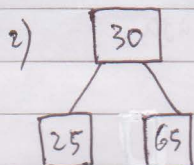


- 65 mengisi node kosong tsb.
- tree sekarang memenuhi property B-Tree.
- dengan maksimal key adalah 2

• Delete 40



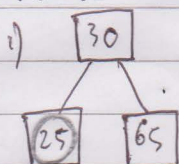
- Mencari key 40
- Traverse dari 30 → 40



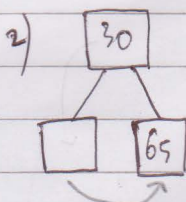
Node ini masih memiliki 1 key

- 40 terletak di leaf node dengan kondisi node memiliki key lebih dari minimum key
- 40 dihapus begitu saja.
- tree sekarang memenuhi property B-Tree dengan minimum key adalah 1.

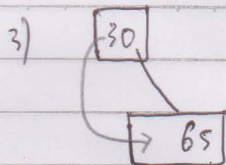
• Delete 25



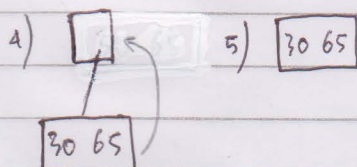
- Mencari key 25
- Traverse dari 30 → 25



- 25 terletak di leaf node dengan kondisi node memiliki key yang berjumlah = minimum key.
- 25 dihapus dan node tsb kosong sekarang

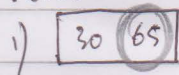


- node kosong akan di merge dengan node kiri / kanannya.
- node tsb di merge dengan node kanannya.

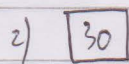


- 30 mengisi node kosong tsb dan node 30 sebelumnya menjadi kosong.
- node kosong ini merupakan root.
- node dengan key 30 dan 65 akan terlempar ke atas dan menjadi root.

• Delete 65

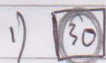


- Mencari key 65



- 65 dihapus begitu saja.
- Tree sekarang memenuhi property B-Tree dengan minimum key adalah 1.

• Delete 30



- Mencari key 30
- 30 dihapus begitu saja