3) a.) Insert : 100, 20, 60, 50, 30 (M = 3)

b.) Insert : 40, 90, 45, 25, 65

c.) Delete : 90, 20, 60, 45, 50

d.) Delete : 100, 40, 25, 65, 30

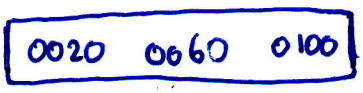
(Tiap merge, nodenya disort dari value terkecil ke value terbesar)

- B-Tree:

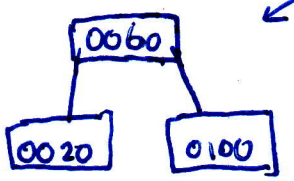
a.1.) Insert 100

a.2) Insert 20

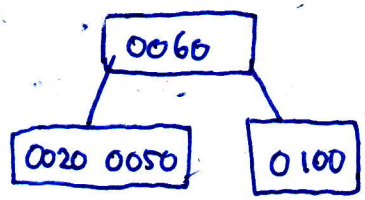
a.3.) Insert 60



→ karena melanggar Property ↓ Median di lempar ke atas



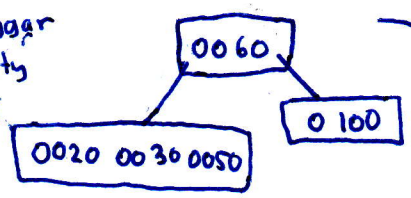
a.4.) Insert 50



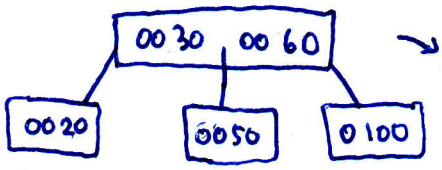
• Karena 50 < 60 dia diletakkan di kiri dan merge dengan 0020

a.5.) Insert 30

melanggar property ↓ median di lempar keatas

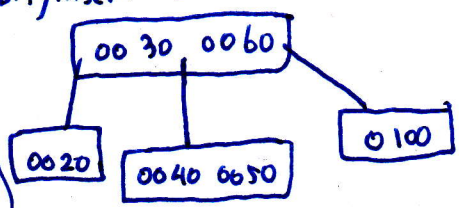


→ • Karena 30 < 60 dia diletakkan di kiri dan di merge dengan 0020 0050



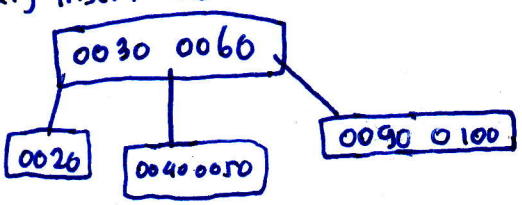
→ 30 merge dengan 60. Lalu angka 20 dan 50 di split

b.1.) Insert 40



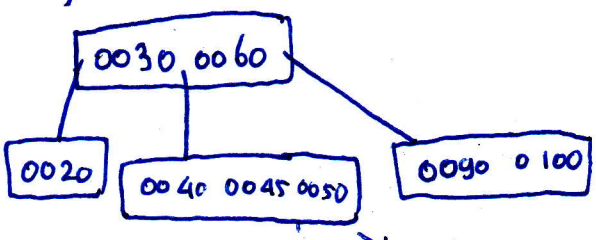
(• karena 30 < 40 < 60, 40 diletakkan di tengah
• kemudian dimerge dengan 50)

b.2.) Insert 90

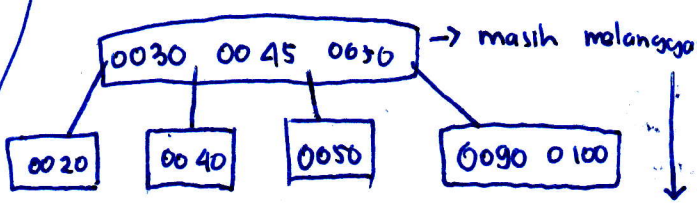


(• karena 90 > 60, 90 diletakkan dikanan. Kemudian di merge dengan 100)

b.3.) Insert 45



melanggar property ↓ median ditendang keatas

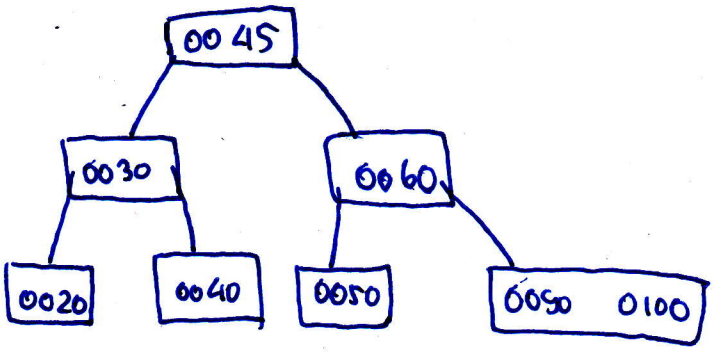


→ masih melanggar

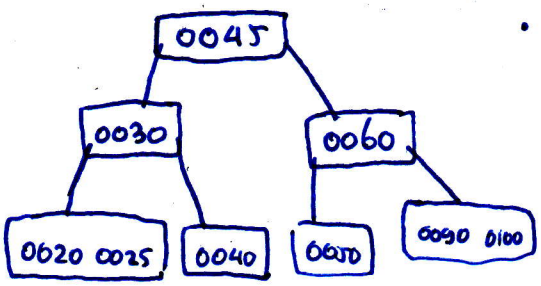
• median tendang keatas

• lalu, 30 dengan 50 di split.

b.3) (lanjutan)

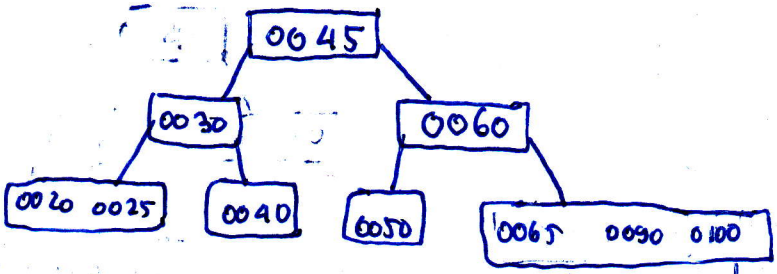


b.4) Insert 25



- karena 25 < 45, 25 diletakan di node kiri dan karena 25 < 30, 25 diletakan di node kiri lagi dan merge dengan 20

b.5.) Insert 65

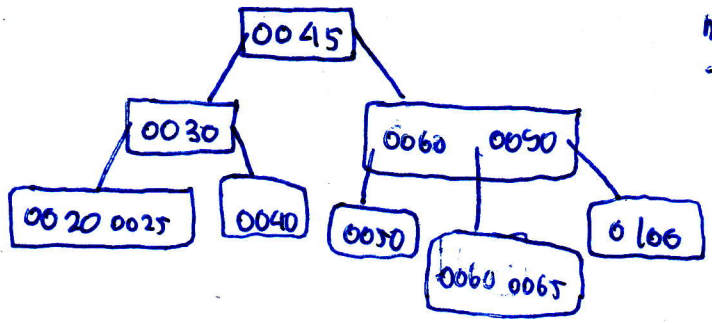


melanggar Property

↓

Median di lempar keatas

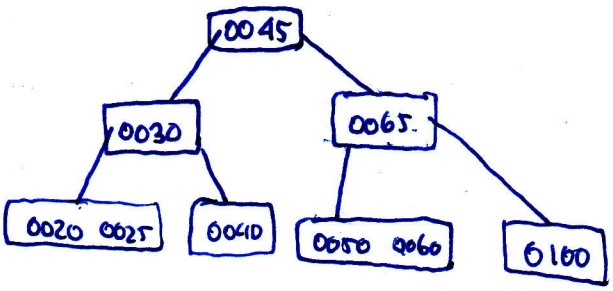
↓



Nanti 65 dan 100 dr split

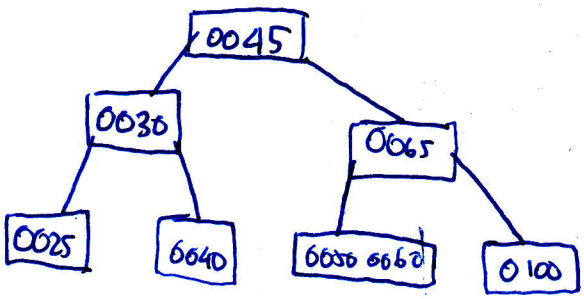
C.1) Delete 90

- Saat 90 dihapus, posisinya digantikan oleh 65 dan 60 diletakan dibawah dan dimerge dengan 50 (50 dan 60 < 65)



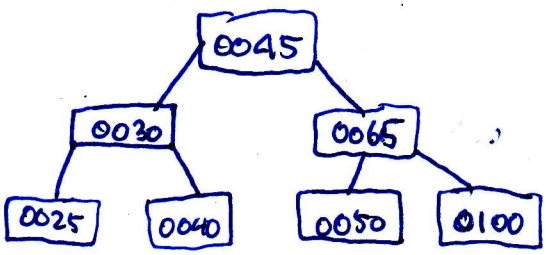
C.2) Delete 20

- karena Node yang terdapat 20 dan 25 adalah leaf, Proses yang terjadi adalah menghapus 20 nya saja.



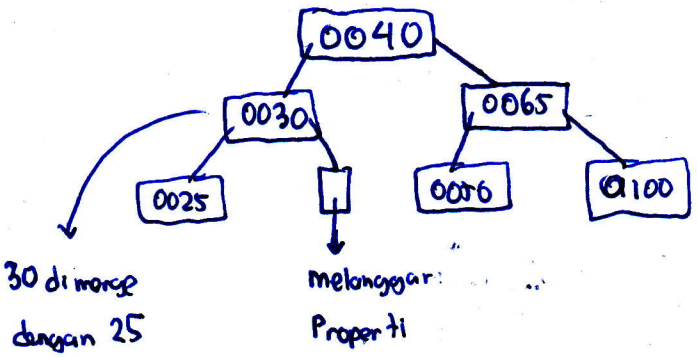
C.3) Delete 60

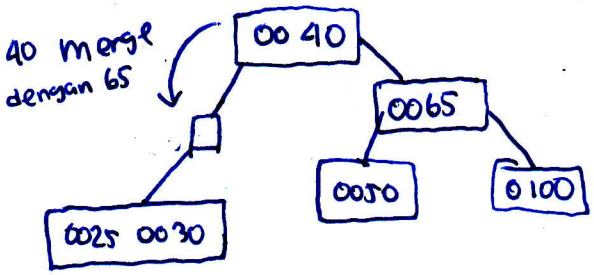
- Hapus 60 saja karena node-nya termasuk leaf.

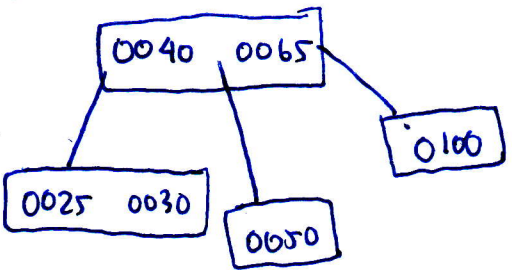


C.4.) Delete 45 (Root)

- Setelah 45 Dihapus, karena 45 root, Cari angka terbesar di node kiri dan Jadikan root terbaru.

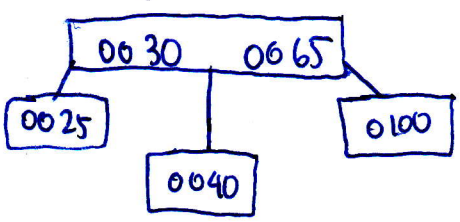




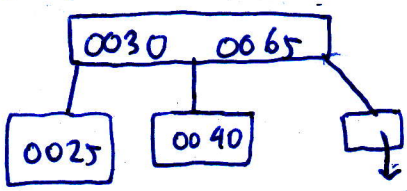


C.5) Delete 50

- Walaupun 50 adalah leaf, jika 50 dihapus, Akan melanggar properti karena kekurangan key. Jadi 40 (left sibling) menggantikan posisi awalnya 50 dan 30 menggantikan posisi awalnya 30



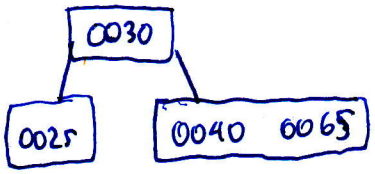
D.1.) Delete 100



melanggar Properti

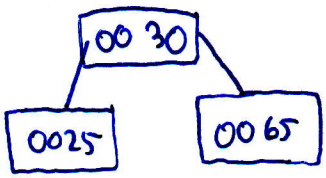
↓

65 merge dengan 40

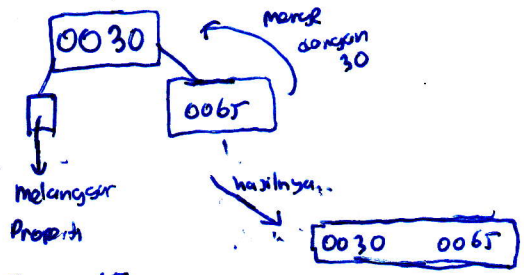


D.2.) Delete 40

- Karena 40 adalah leaf dan jika 40 dihapus tidak melanggar properti, maka proses yang terjadi adalah menghapus 40 saja



D.3) Delete 25



D.4.) Delete 65

- Langsung hapus 65 saja

D.5) Delete 30

Langsung hapus 30 saja karena 30 berada di node leaf

(kosong)

Pseudocode:

```
B-Tree-Search(x,k)
  i = 1
While i ≤ x.n and k > x.key[i]
    i = i+1
IF i ≤ x.n and k == key[i]
    return (x,i)
  if x.leaf
    return nil
  else Disk-Read(x.c[i])
      return B-Tree-Search(x.c[i],k)

B-Tree-Create(T)
   x = Allocate-Node()
   x.leaf = true
   x.n = 0
   Disk-Write(x)
   T.root = x

B-Tree-Split-Child(x,i)
    z = Allocate-Node()
    y = x.c[i]
    z.leaf = y.leaf
    z.n = t-1
    for j = 1 to t-1
        z.key[j] = y.key[j+t]
    IF NOT y.leaf
        for j = 1 to t
            z.c[j] = y.c[j+t]
    y.n = t-1
    for j = x.n+1 down to i+1
        x.c[j+1] = x.c[j]
    x.c[i+1] = z
    for j = x.n down to i
        x.key[j+1] = x.key[j]
    x.key[i] = y.key[t]
    x.n = x.n+1
    Disk-Write(y)
    Disk-Write(z)
    Disk-Write(x)
```

(Langutan Pseudocode)

```
B-Tree-Insert (T,k)
  r = T.root
  IF r.n == 2t-1
        S = Allocate-Node()
        T.root = S
        S.leaf = false
        S.n = 0
         S.c[1] = r
         B-Tree-Splitchild (S,1)
         B-Tree-NonFull (S,k)
  else B-Tree-Insert-Nonfull (r,k)

B-Tree-Insert-Non Full (x,k)
  i = x.n
  IF x.leaf
     while i ≥ 1 and k < x.key[i]
          x.key[i+1] = x.key[i]
           i = i-1
      x.key[i+1] = k
      x.n = x.n+1
      Disk-Write (x)
   else
     while i ≥ 1 and k < x.key[i]
         i = i-1
     i = i+1
     Disk Read (x.c[i])
     IF x.c[i].n == 2t-1
         B-Tree-Split-Child (x,i,x.c[i])
              IF k > x.key[i]
                   i = i+1
      B-Tree-Insert-NonFull (x.c[i],k)
```