

File Indeks Majemuk

File Indeks Majemuk

- Pendahuluan
- Struktur File Indeks Majemuk
- BTree Insertion Algorithm
- BTree Deletion Algorithm

Pendahuluan

- File berindeks majemuk dimungkinkan membuat beberapa indeks lebih dari satu atribut
- Indeks dibolehkan pada sembarang atribut, bahkan semua atribut
- Pada file berindeks majemuk, pembaruan dilakukan terhadap file utama bukan file overflow
- Karena record dicari lewat indeks, maka indeks harus dinamis
- Begitu terjadi pembaruan maka indeks diperbarui mengikuti perubahan di file utama
- Pada file berindek majemuk semua indeks dianggap sama dan tidak ada kebutuhan pengelolaan ke wilayah overflow

Struktur File Indeks Majemuk

- Pada struktur ini terdapat indeks sebanyak atribut di file, bahkan kita dapat membuat indeks dengan kunci gabungan beberapa atribut sekaligus
- Pengaksesan record yang digunakan adalah *record anchor* (indeks menunjuk ke record).
- Tiap indeks dapat di indeks lagi seperti pada file indeks sequensial
- Struktur indeks menggunakan Btree
- Blok-blok indeks di B-tree harus dijaga setidaknya memuat setengah effective fanout (y_{eff}) bernilai antara y dan $y/2$. Untuk analisis diasumsikan kepadatan B-tree sebesar 0.69 atau
$$x = \log y_{\text{eff}} n'$$
- Dimana : Satu indeks mengacu n' record dengan n' ditentukan oleh jumlah record yang mempunyai nilai atribut yang dapat diindeks.

BTree Insertion Algorithm

- Cari posisi yang sesuai bagi rekord baru, mulai dari root BTree.
- Jika tersedia space, insert new record sesuai urutan, jika tidak terjadi overflow
- Jika terjadi overflow :
 - split menjadi dua node
 - Pilih node tengah untuk naik ke level berikutnya
 - set pointer dari parent node ke child node

Contoh Insert Pada BTree

- Diketahui BTree dengan kapasitas order $d = 1$
- Gambarkan hasil Insert data :
cat, ant, dog, cow, rat, pig dan gnu pada BTree tersebut.

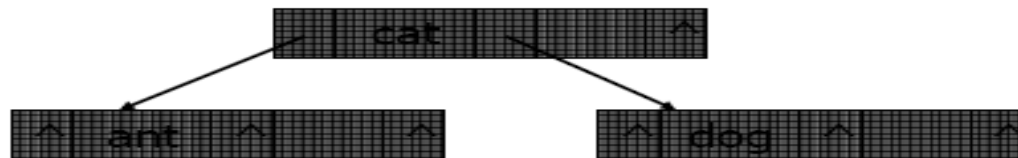
- Insert cat



- Insert ant

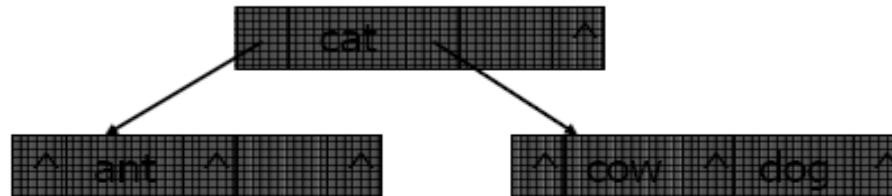


- Insert dog → overflow



Contoh Insert Pada BTree

- Insert cow

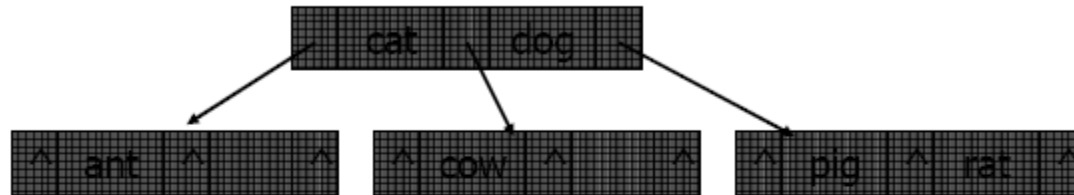


- Insert rat → overflow

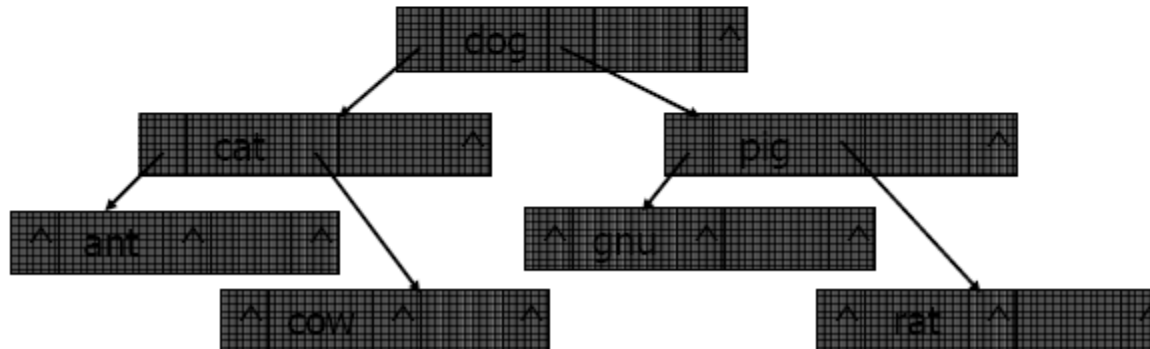


Contoh Insert Pada BTree

- Insert pig



- Insert gnu → overflow

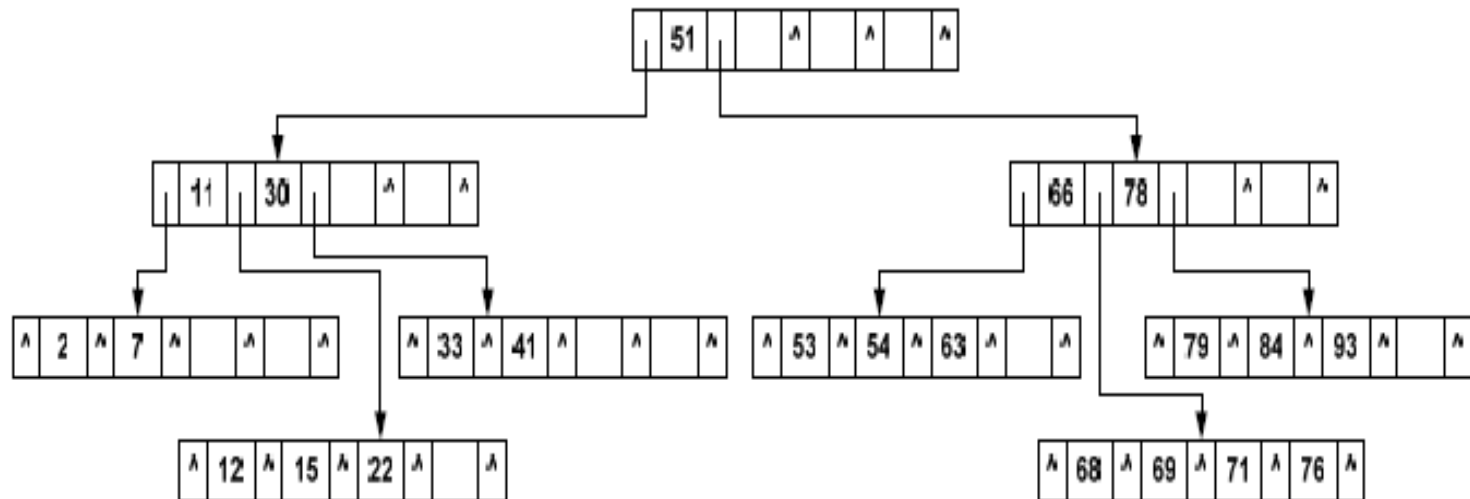


BTree Deletion Algorithm

- Menghapus node daun (leaf node), tidak melanggar kapasitas minimum
- Menghapus non leaf node, ganti dengan satu record dari daun, tidak melanggar kapasitas minimum
- Menghapus leaf node, melanggar kapasitas, perbaiki dengan redistribusi record
- Menghapus leaf node, melanggar kapasitas, perbaiki dengan coalescing node

Contoh Delete Node Pada BTree

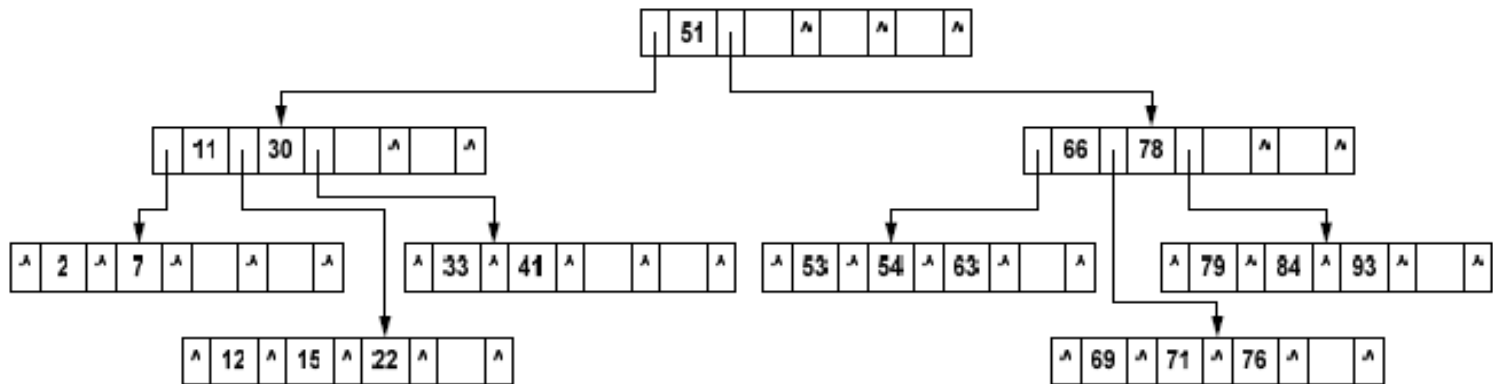
Diketahui suatu BTree sbb :



Contoh Delete Node Pada BTree

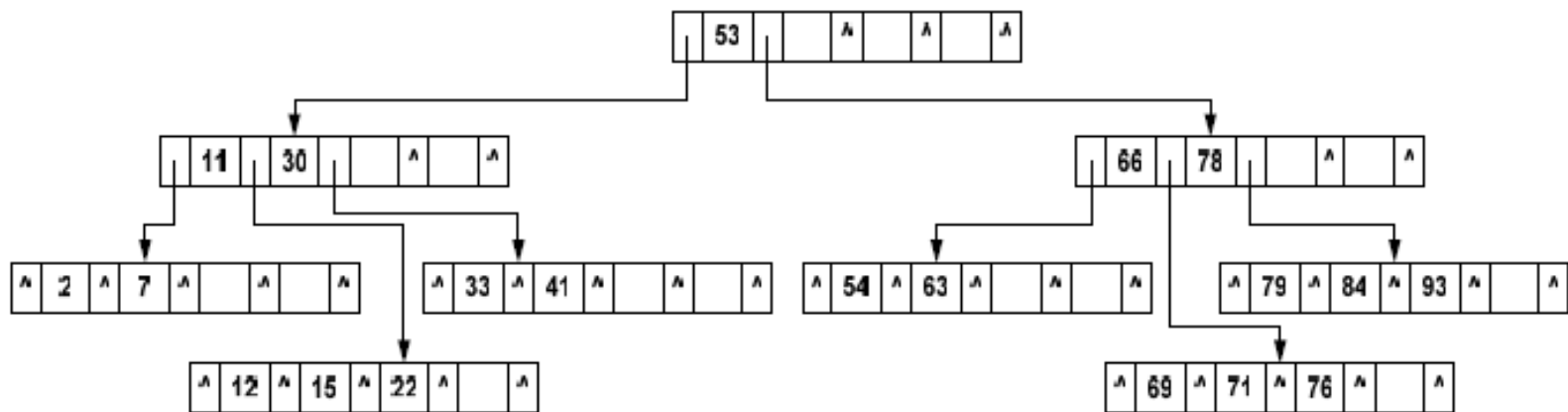
Kasus 1 : Menghapus node daun (leaf node), tidak melanggar kapasitas minimum

Misal hapus node 68, maka dihasilkan BTree sbb :



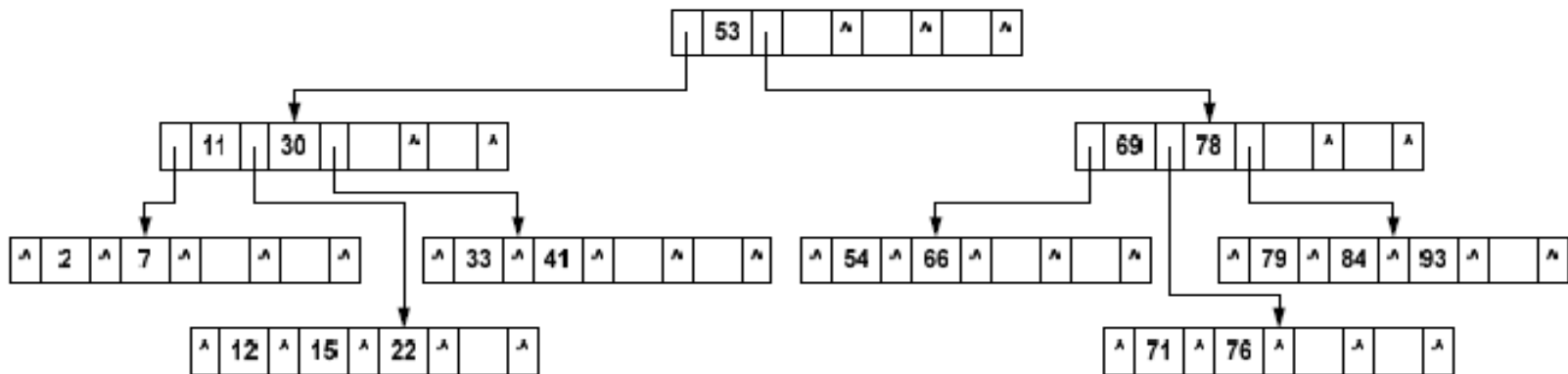
Contoh Delete Node Pada BTree

Kasus 2 : Menghapus non leaf node, ganti dengan satu record dari daun, tidak melanggar kapasitas minimum. Misal hapus node 51



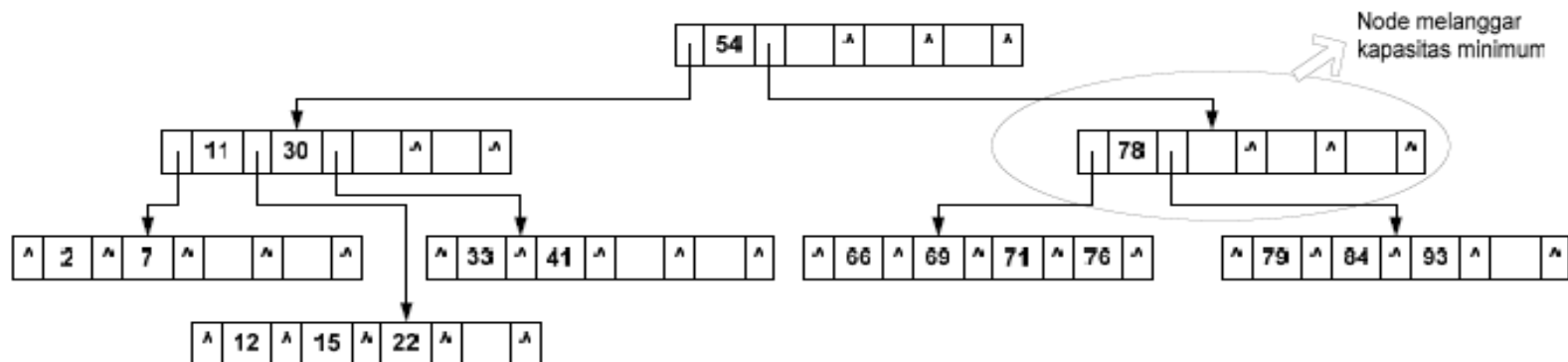
Contoh Delete Node Pada BTree

Kasus 3 : Menghapus leaf node, melanggar kapasitas, perbaiki dengan redistribusi rekord
Misal hapus node 63



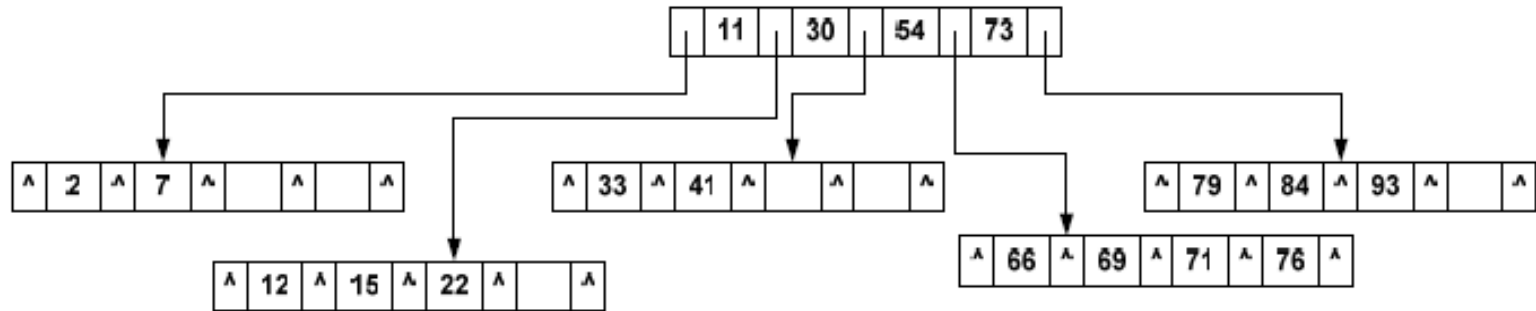
Contoh Delete Node Pada BTree

Kasus 4 : Menghapus leaf node, melanggar kapasitas, perbaiki dengan coalescing node
Misal hapus node 53



Pada BTree tsb, dilakukan coalescing node sehingga akan dihasilkan BTree sbb :

Contoh Delete Node Pada BTree



Perhatikan bahwa pada kasus delete ini telah menyebabkan penurunan level Btree