

File Indeks Majemuk

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

Struktur File Indeks Majemuk

- ▶ Terdiri dari main file dan file-file index file berindeks majemuk
- ▶ Tidak ada rantai overflow
- ▶ Tidak dikenal konsep atribut kunci → tidak ada keterurutan berdasarkan atribut kunci
- ▶ Pengubahan data langsung dilakukan terhadap main file

Struktur File Indeks Majemuk

- Index bersifat multiple index, dinamis, record anchored
- Entri index terdiri dari atribut dan TID
- Entri index terurut berdasarkan nilai atributnya
- Next record diakses berdasarkan keterurutan entri pada index-nya
- Tiap index dapat bersifat multilevel
- TID pada indeks berisi alamat block dan posisi record
- Perubahan pada main file berimplikasi terhadap indeks-nya
- Struktur indeks menggunakan BTree

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.

Contoh Insert Pada BTree

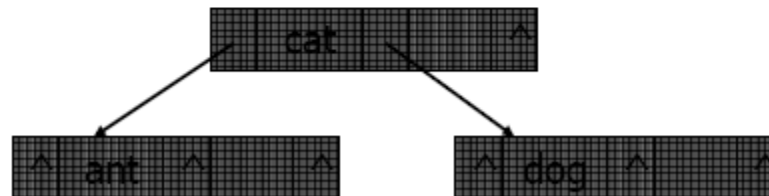
- Insert cat



- Insert ant

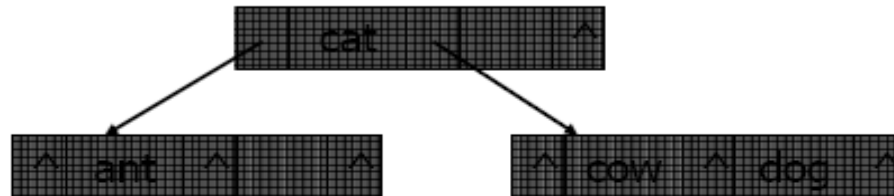


- Insert dog → overflow

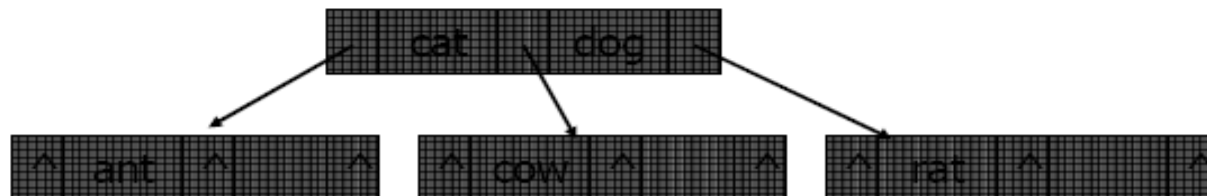


Contoh Insert Pada BTree

- Insert cow

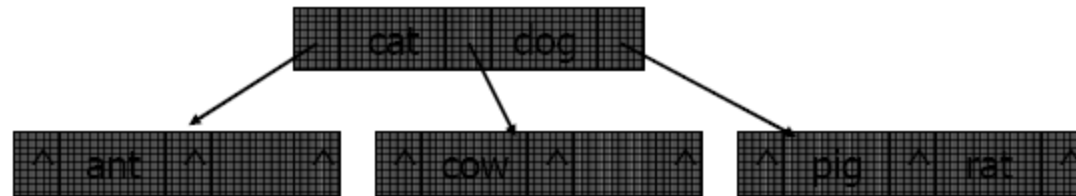


- Insert rat → overflow

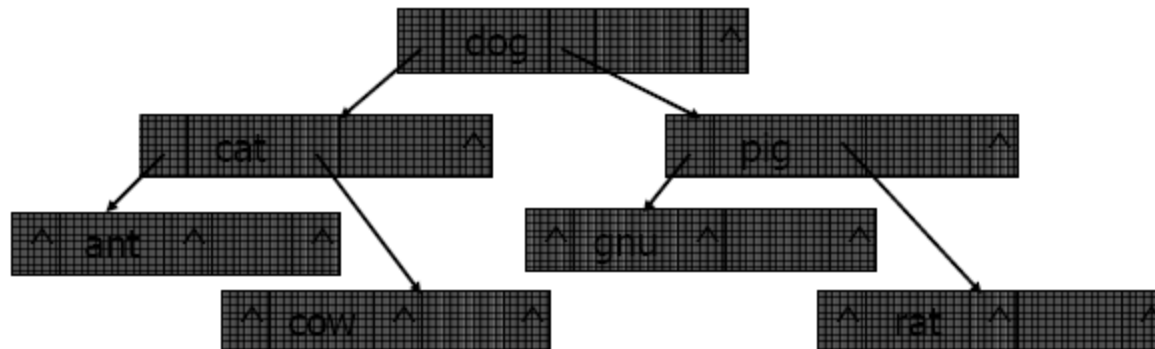


Contoh Insert Pada BTree

- Insert pig



- Insert gnu → overflow

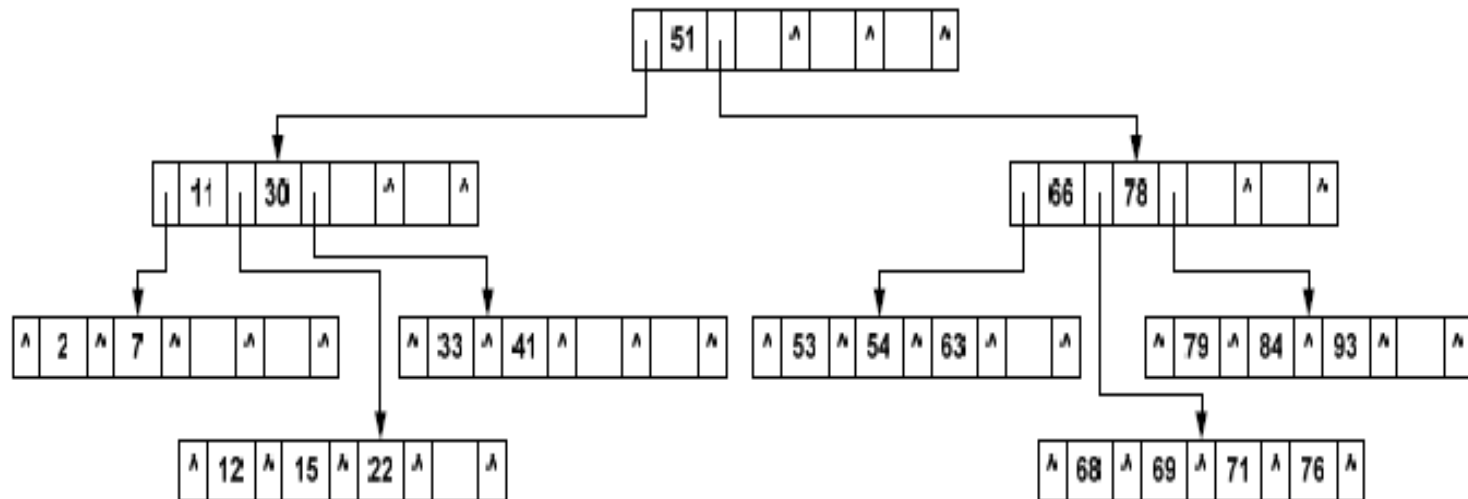


BTree Deletion Algorithm

- 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

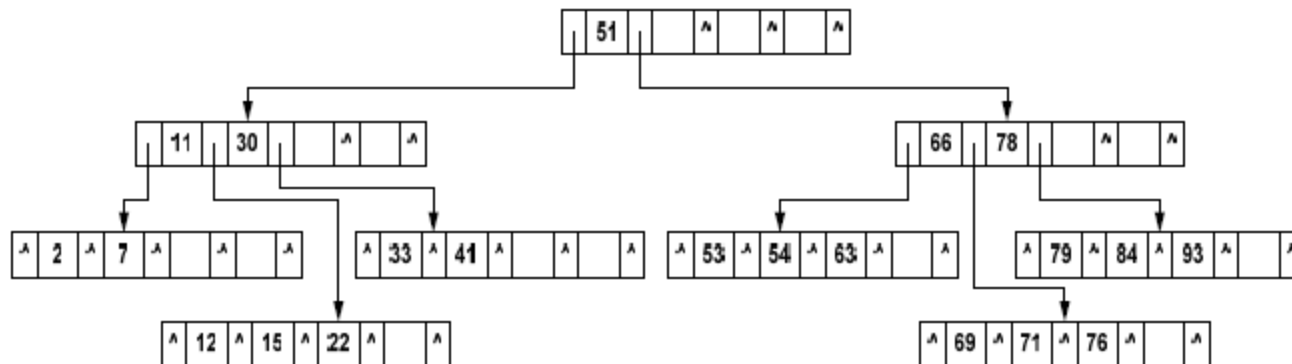
Diketahui suatu BTree sbb :



Contoh Delete Node Pada BTree

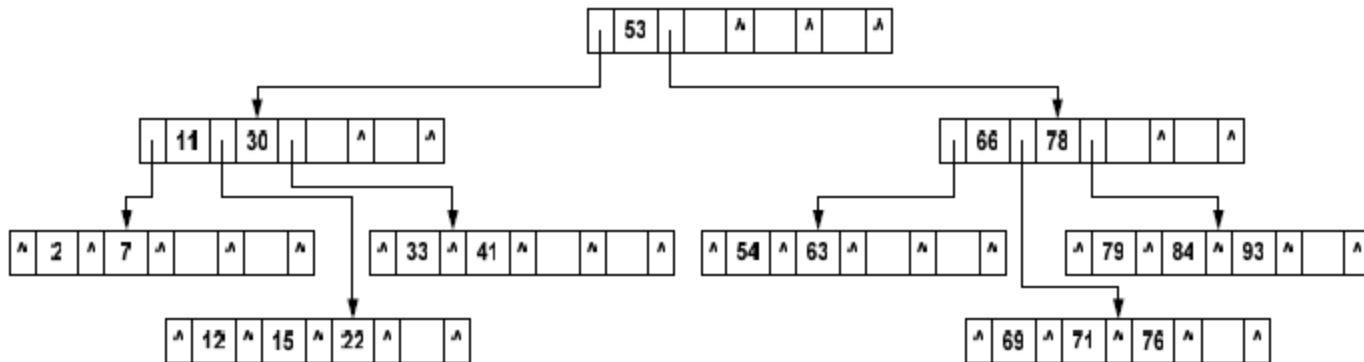
Kasus 1 : Menghapus node daun (leaf node), tidak melanggar kapasitas minimum

Misal hapus node 68, maka dihasilkan BTree sbb :



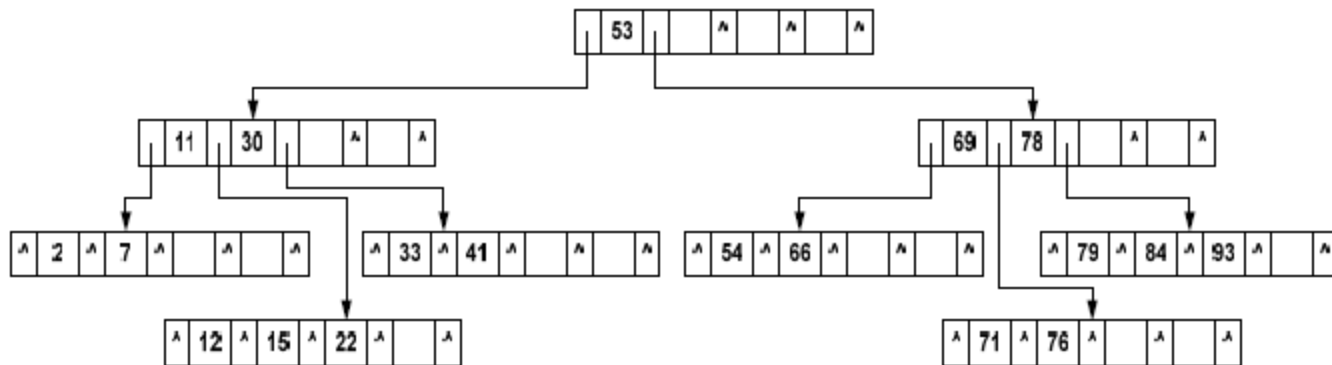
Contoh Delete Node Pada BTree

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



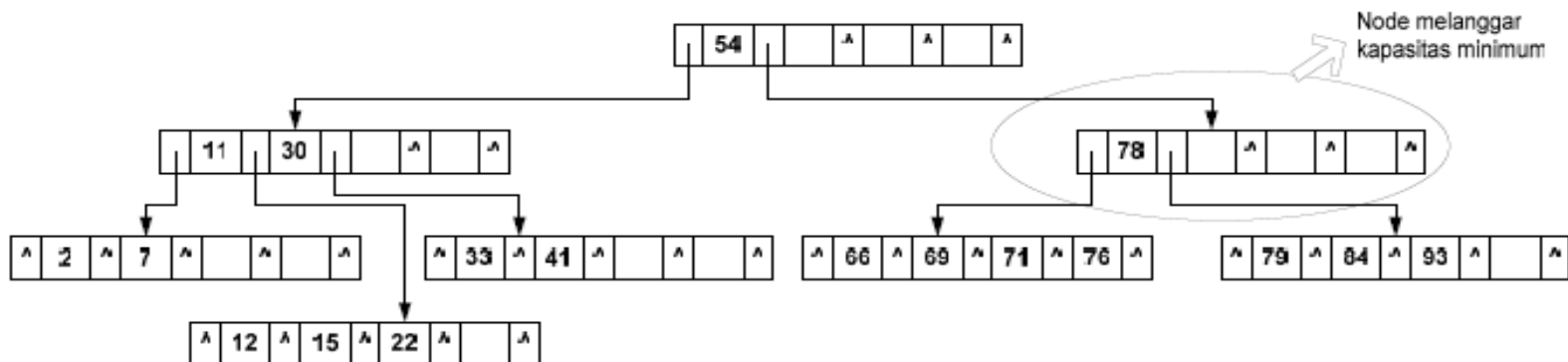
Contoh Delete Node Pada BTree

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



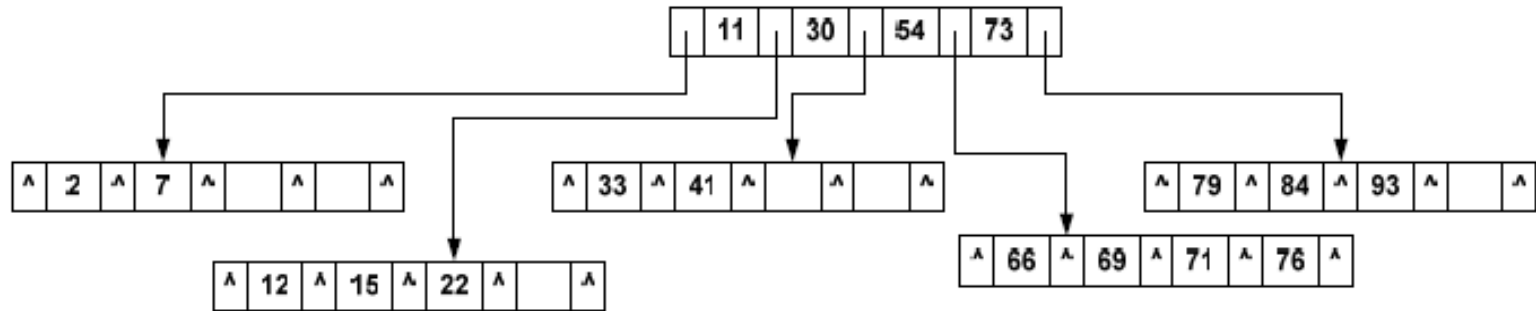
Contoh Delete Node Pada BTree

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