

## BAB V

### KESIMPULAN

Struktur data Pohon-B banyak diterapkan untuk *me-manage* data yang tersimpan dalam *external storage* dengan jumlah yang cukup besar. Ada dua operasi penting yang menjadi pokok permasalahan dalam struktur data Pohon-B, yaitu bagaimana menghindari keadaan *overflow* dalam proses penyisipan kunci dan keadaan *underflow* dalam penghapusan kunci untuk menjaga pohon tetap memenuhi kriteria Pohon-B.

Keadaan *overflow* dapat ditangani dengan memecah simpul ( dimana kunci akan disisipkan ) menjadi dua dengan satu kunci tengah dibawa ke simpul ayah. Jika Pohon-B tersebut disusun berdasarkan *ascending order* ( urutan naik ) berorde  $m$ , maka  $m \div 2$  kunci terkecil pertama ditempatkan dalam simpul kiri, dan  $m \div 2$  kunci terbesar ditempatkan dalam simpul kanan. Jika  $m$  berorde genap, dilakukan metode *bias*.

Keadaan *underflow* dapat ditangani dengan cara memindah kunci dari simpul ayah ke simpul yang mengalami *underflow* dan satu kunci terdekat dari simpul saudara dipindah ke simpul ayah. Untuk kondisi dimana simpul saudara juga rawan *underflow*, maka dilakukan proses penggabungan dua simpul ( simpul yang mengalami *underflow* dan saudaranya yang terdekat ) menjadi satu simpul baru

dengan satu kunci yang dipindah dari simpul ayah. Penggabungan akan berlanjut terus ke akar, jika untuk kondisi paling buruk, simpul ayah juga rawan *underflow*.

Tinggi ( $h$ ) sebuah pohon telusur banyak cabang dengan  $n$  kunci dan  $m$  faktor cabang adalah  $O(\log n)$ . Jika  $n \geq 1$ , maka untuk sebarang  $n$ -kunci, tinggi

Pohon-B ( $h$ ) dengan derajat minimum  $t \geq 2$  tidak lebih dari  $\log_2 \frac{n+1}{t}$ . Banyaknya

rata-rata akses disk dalam proses penyisipan adalah  $h + 1$ .  $m$  yang ideal berada

dalam interval  $50 \leq m \leq 400$  (*flat region*) berdasarkan fungsi  $(35+0,06)/\lg m$ .

Interval harga  $m$  tidak mutlak, tetapi sangat tergantung pada kemampuan *hardware* pendukung seperti *disk drive*, kapasitas *buffer* dan sebagainya.

