

BAB I

PENDAHULUAN

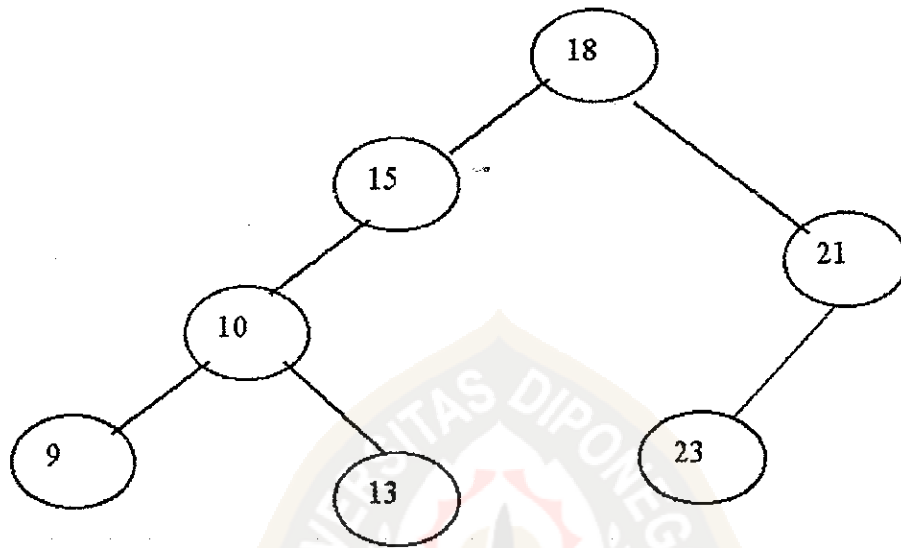
1.1 Latar Belakang

Salah satu cara yang sering digunakan dalam menyajikan suatu struktur data adalah dengan menggunakan struktur data pohon (*tree*). Sebuah pohon biner adalah himpunan berhingga elemen (simpul) yang kosong atau berisi sebuah simpul pohon dengan dua pohon biner saling terpisah yang disebut subpohon kiri (*left subtrees*) dan subpohon kanan (*right subtrees*). (Niklaus Wirth)

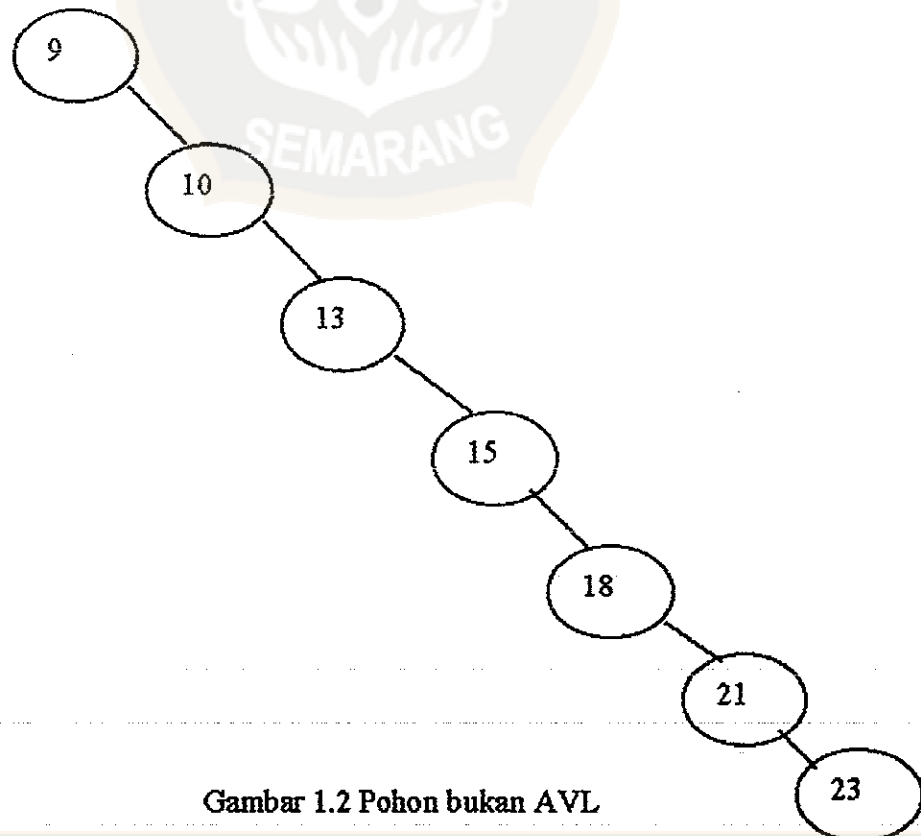
Pohon biner seimbang (*balanced binery trees*) sebagai salah satu jenis pohon merupakan jenis pohon sederhana, dengan setiap simpul yang mempunyai sifat bahwa perbedaan tinggi antara subpohon kiri dan subpohon kanan maksimal satu.

Dalam teori, pohon biner seimbang (*balanced binery trees*) menyediakan kamus struktur data dengan memberikan harga logaritmik untuk berbagai operasi. Dalam praktek, pohon biner seimbang dapat mencegah pencarian pohon dari tingkat kemrosotan (*degenerate*) ke dalam list di bawah tingkat penyisipan.

Untuk memberikan gambaran tentang pentingnya pohon biner seimbang, berikut disajikan salah satu contoh pencarian data yang menggunakan pohon AVL dan pohon bukan AVL.



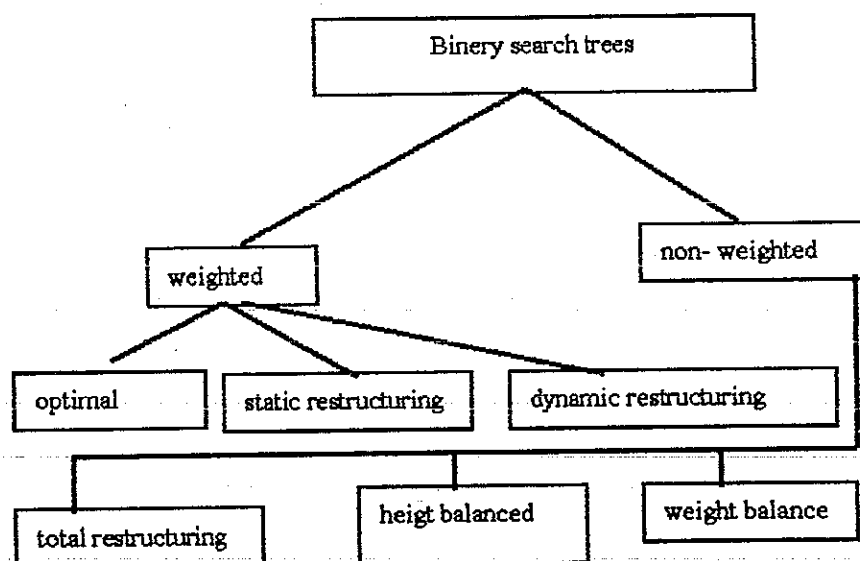
Gambar 1.1 Pohon AVL



Gambar 1.2 Pohon bukan AVL

Dari contoh gambar 1. 1 tersebut dapat dilihat bahwa proses pencarian data secara umum akan lebih cepat dilaksanakan pada pohon pertama (Gb 1.1) dibanding pada pohon kedua (Gb 1. 2). Sebagai contoh, untuk mencari bilangan 18 pohon pertama membuntuhkan satu kali perbandingan tetapi pohon kedua perlu 6 kali perbandingan. Tetapi sebaliknya untuk mencari bilangan 9, pada pohon pertama butuh 4 kali perbandingan sedang pohon kedua cukup satu kali perbandingan. Dari kedua contoh tersebut dapat dilihat bahwa pencarian data yang diharapkan pada pohon biner yang pertama akan lebih cepat dibanding pohon kedua. perbedaannya akan semakin nyata apabila banyaknya data yang diketahui semakin banyak.

Ada beberapa metoda yang dapat dipertimbangkan untuk pencarian pohon biner seimbang (*balanced binery search trees*), bergantung pada kriteria untuk menyeimbangkannya. Di antara metoda tersebut adalah height balance (*AVL tree*), weight balance dan total restructuring. Secara garis besar dapat digambarkan sebagai berikut :



Gambar 1.3 Teknik Pohon pencarian biner

Dalam tugas akhir ini akan dibahas metoda height balance (AVL) untuk menyeimbangkan pohon pencarian biner dari bentuk tak seimbang. Dilengkapi dengan operasi-operasi penyeimbangannya.

1.2 Perumusan Masalah

Masalah yang akan disajikan dalam tugas akhir ini adalah :

1. Bagaimana menggunakan metoda height balance (AVL) untuk pohon pencarian biner seimbang.
2. Bagaimana analisa algoritma (running time) dari algoritma AVL tersebut yang digunakan untuk menyeimbangkan pohon pencarian biner seimbang.
3. Bagaimana algoritma tersebut diimplementasikan dalam program pascal.

1.3 Metoda Pembahasan

Pertama yang akan dibahas adalah pengertian pohon pencarian biner seimbang, pembentukannya, penggunaan metoda AVL untuk menyeimbangkan pohon biner tersebut, penyisipan simpul baru dan penghapusan simpul.

Pembahasan selanjutnya adalah mengenai sifat-sifat pohon pencarian biner seimbang tersebut. Kemudian dilakukan analisa running time dari algoritma dari metoda tersebut.

1.4 Sistematika Pembahasan

Bab I berisi tentang latar belakang dan perumusan masalah dari tugas akhir ini disertai metoda pembahasannya.

Bab II berisi tentang koefisien binomial, relasi rekursif, fungsi pembangkit dan penyelesaian relasi rekursif dengan fungsi pembangkit . Di samping itu dibahas pula tentang konsep dasar pohon yang terdiri atas graph dan pohon .

Bab III berisi tentang pembentukan pohon pencarian biner seimban, penambahan simpul baru, penghapusan simpul yang disertai dengan algoritma-algoritma penyeimbangannya, running time dari pohon AVL tersebut.

Bab IV berisi Implementasi Program penyeimbangan pohon AVL dan kesimpulan dari tugas akhir ini.

