ABSTRAK

Map adalah struktur data umum dan bagian yang penting dari pemrograman komputer. *Map* memungkinkan kita untuk menyimpan data berupa pasangan kunci-nilai, sehingga pengguna dapat memasukkan data ke dalam *map* dengan menentukan pasangan kunci-nilai dan kemudian dapat mengambil nilai yang sesuai dengan kunci tertentu dari dalam *map*. Terdapat bahasa pemrograman yang berbeda mendukung map dalam berbagai cara, dalam hal ini bahasa pemrograman Java memiliki implementasi tersendiri dari *map* seperti `TreeMap`. `TreeMap` menyimpan semua pasangan kunci-nilai ke dalam struktur data pohon yang disebut pohon *red-black* dan struktur tersebut diurutkan oleh kunci. `TreeMap` menyimpan setiap pasangan kunci-nilai ke setiap *node* di pohon sehingga jumlah *node* dan pasangan kunci-nilai bernilai sama. Ada cara yang berbeda untuk menempatkan pasangan kunci-nilai dalam *node* yang setiap node di pohon menyimpan lebih dari satu pasangan kunci-nilai sehingga dapat mengurangi jumlah *node* dan dengan menggunakan struktur data pohon lain yang *right-threaded* AVL *tree*. Terakhir adalah memeriksa kinerja kecepatan eksekusi dan penggunaan memori menggunakan aplikasi sederhana.

Kata Kunci : Map, TreeMap, AVL tree

ABSTRACT

Map is common data structures and important part of computer programming. A map allows us to store data as key-value pairs, so user can put data into the map by specifying a key-value pair and then can retrieve the value corresponding to a particular key from the map. There are different programming languages supported map in a different ways, in this case the Java Programming language had its own implementation of maps such as TreeMap. TreeMap stores all key-value pairs into a tree data structure called red-black tree and it is sorted by keys. TreeMap stores each key-value pairs into a single node in the tree thus the number of nodes and key-value pairs are equals. There are different way to put the key-value pairs in the nodes which is each node in the tree hold more than one key-value pairs so it can reduce the number of nodes and by using other tree data structures which is right-threaded AVL tree. Finally check the performance by the execution speed and memory usage by a simple application.

Keyword : Map, TreeMap*,* AVL *tree*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR PROGRAM