

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada masa kini komputer mengalami perkembangan yang sangat cepat untuk memberikan kemudahan dalam mengakses, menyimpan dan mendapatkan berbagai informasi. Seiring meningkatnya kecanggihan teknologi komputer tersebut, maka jumlah informasi yang perlu disimpan ke dalam sebuah komputer semakin bertambah. Dengan bertambahnya jumlah informasi tersebut, maka perangkat komputer pun harus memiliki daya tampung dan daya kelola data yang lebih baik.

Di dalam sebuah komputer, terdapat salah satu media penyimpanan data sementara yang dinamakan *Random Access Memory* (RAM). Proses penyimpanan data di dalam memori tersebut menggunakan aturan yang disebut struktur data. Memori komputer diakses langsung oleh *Central Processing Unit* (CPU) sehingga dapat memproses data dengan sangat cepat, akan tetapi daya tampung di dalamnya sangat terbatas. Maka dari itu, diperlukan konsep struktur data yang tepat, agar data dapat disimpan dan diproses dengan baik oleh memori dan CPU di dalam komputer. Struktur data mempunyai peranan penting di dalam proses pengembangan aplikasi komputer, terutama aplikasi yang berguna untuk memproses data dalam jumlah yang banyak. Konsep struktur data banyak diterapkan pada beberapa bahasa pemrograman komputer, salah satunya adalah Java.

Di dalam bahasa pemrograman Java, struktur data diterapkan di dalam *Java Collections Framework*. Di dalam *framework* tersebut terdapat sebuah *interface* `Map` yang berguna untuk menyimpan data berupa pasangan *key-value* atau disebut *entry*. Pasangan *key-value* sering digunakan di dalam berbagai aplikasi, salah satu contohnya yaitu untuk mengelola data mahasiswa. *Key* digunakan sebagai nomor induk mahasiswa dan *value* berupa data lengkap mahasiswa tersebut, sehingga untuk mengakses data seorang mahasiswa, hanya

diperlukan *key* sebagai acuan. Salah satu kelas yang mengimplementasi *interface* `Map` di dalam Java adalah kelas `TreeMap`. Pada kelas `TreeMap`, masing-masing *entry* atau pasangan *key-value* disimpan di dalam *node* pada yang disusun ke dalam struktur data *red-black tree* yang berbasis *binary search tree*, sehingga jumlah *node* sama dengan jumlah pasangan *key-value*. (Collins, 2011).

Hal ini mengakibatkan semakin banyak jumlah *key-value* yang disimpan, maka semakin banyak jumlah *node* pada struktur pohon tersebut. Oleh karena itu di dalam penelitian ini akan dibuat sebuah tipe data abstrak *map* yang mempunyai cara kerja mirip dengan `TreeMap`. Perbedaan dari tipe data abstrak *map* yang akan dibuat ini adalah *node* dapat menyimpan lebih dari satu pasang *key-value*. Pasangan *key-value* di dalam *node* tersebut akan disimpan menggunakan struktur *array*. Masing-masing *node* tersebut selanjutnya dibentuk ke dalam struktur data *right-threaded AVL tree* yang merupakan gabungan dari struktur data *AVL tree* dan *right-threaded tree* (Pfaff, 2007). *AVL tree* sendiri merupakan salah satu jenis struktur pohon seimbang selain *red-black tree* (McAllister, 2010).

Tipe data abstrak *map* yang telah dibuat tersebut kemudian akan diuji coba bersama dengan kelas `TreeMap` pada Java. Proses uji coba akan dilakukan dengan mengukur kecepatan waktu dalam memproses penambahan, pencarian dan penghapusan data. Selain itu juga dilakukan pengukuran penggunaan memori, sehingga dapat dilakukan analisis untuk melakukan perbandingan dari kedua tipe *map* tersebut dilihat dari waktu proses dan kebutuhan memori yang digunakan.

1.2 Rumusan Masalah

Melihat latar belakang yang telah diuraikan di atas, terdapat beberapa permasalahan yang dihadapi, diantaranya:

1. Bagaimana pengembangan struktur data *right-threaded AVL tree* beserta implementasi program simulasi dari struktur data tersebut.
2. Bagaimana pengembangan sebuah tipe data abstrak *map* berbasis *right-threaded AVL tree* dengan konsep satu *node* di dalam struktur *tree* tersebut dapat menyimpan lebih dari satu pasang *key-value* beserta program simulasi dari tipe data abstrak *map* tersebut.

3. Bagaimana implementasi perangkat lunak sederhana untuk mengukur kecepatan waktu eksekusi dan penggunaan memori pada tipe data abstrak *map* yang telah dikembangkan dan juga pada kelas `TreeMap` di dalam Java serta analisis perbandingan terhadap kedua tipe data *map* tersebut berdasarkan hasil pengukuran yang telah dilakukan.

1.3 Tujuan Penelitian

Berikut ini merupakan tujuan dari penelitian yang akan dicapai di dalam tugas akhir ini untuk menjawab rumusan masalah yang ada, diantaranya:

1. Mendesain dan mengimplementasikan struktur data *right-threaded AVL tree* beserta program simulasinya.
2. Mendesain dan mengembangkan sebuah tipe data abstrak *map* yang dapat menyimpan dan memproses data berupa *key-value* berbasis struktur data *right-threaded AVL tree* beserta dengan program simulasinya.
3. Mengukur penggunaan waktu dan memori antara kelas `TreeMap` di dalam Java dan tipe data abstrak *map* yang telah dibuat di dalam tugas akhir ini.
4. Menganalisis hasil pengukuran waktu dan memori untuk memperoleh perbandingan dari tipe data abstrak *map* yang telah dibuat dengan kelas `TreeMap` pada Java.

1.4 Batasan Masalah

Terdapat beberapa batasan masalah yang akan diterapkan di dalam penelitian ini, diantaranya:

1. *Key* yang digunakan pada program simulasi struktur data *right-threaded AVL tree* bertipe *integer*.
2. *Key* yang digunakan pada tipe data abstrak *map* bertipe *integer*.
3. *Value* yang digunakan pada tipe data abstrak *map* bertipe *generic*.
4. Aplikasi simulasi untuk menguji coba struktur *right-threaded AVL tree* dan tipe data abstrak *map* berbasis *Graphical User Interface* atau GUI

5. Aplikasi untuk menghitung waktu eksekusi dan memori pada komputer berbasis *console*.
6. Pengukuran waktu eksekusi dilakukan pada fungsi penyimpanan, penghapusan dan pengambilan *key-value*.
7. Pengukuran kebutuhan memori pada komputer dilakukan pada fungsi penyimpanan *key-value*.
8. *Sample* data yang digunakan oleh program simulasi pada tipe data abstrak *map* berupa data *customer* dengan *volume* maksimal 10000 data.
9. *Sample* data yang digunakan pada aplikasi pengukuran waktu dan memori terdiri dari 3 jenis data yang dibedakan berdasarkan tipe *value*, yaitu: *Customer*, *String* dan *Double*, sedangkan *key* dari masing-masing *value* seluruhnya bertipe *integer*.
10. Semua *sample* data dibuat ke dalam beberapa file yang berekstensi *.csv*.
11. Batasan perangkat keras yang digunakan untuk mengukur kecepatan waktu eksekusi dan penggunaan memori yaitu:
 - Processor Intel Core i5 M 520 2.40 GHz
 - Memory (RAM) 4.00 Gb

1.5 Sistematika Pembahasan

Sistematika penyajian dalam laporan tugas akhir ini adalah sebagai berikut:

- BAB I PENDAHULUAN
Bab 1 terdiri atas Latar Belakang, Rumusan Masalah, Tujuan, Batasan Masalah dan Sistematika Pembahasan.
- BAB II LANDASAN TEORI
Bab 2 membahas teori-teori yang mendasari dan mendukung di dalam pembuatan tugas akhir ini. Teori-teori ini diambil dari pustaka yang diperlukan.

- **BAB III ANALISIS DAN DESAIN**
Bab 3 berisi analisis perangkat lunak yang sudah ada dan desain perancangan tipe data abstrak.
- **BAB IV PENGEMBANGAN PERANGKAT LUNAK**
Bab 4 berisi implementasi *class/modul* yang telah diidentifikasi dari hasil analisis dan desain.
- **BAB V TESTING DAN EVALUASI SISTEM**
Bab 5 berisi rencana dan pelaksanaan pengujian program yang berupa *unit testing* dan *black box*.
- **BAB VI KESIMPULAN DAN SARAN**
Bab 6 mencakup kesimpulan dan saran yang dirumuskan dari hasil penelitian.