BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Secara sederhana algoritma merupakan prosedur untuk memecahkan suatu masalah. Algoritma adalah deretan instruksi yang jelas untuk memecahkan masalah, yaitu untuk memperoleh keluaran yang diinginkan dari suatu masukan dalam jumlah waktu yang terbatas [1]. Banyak cabang dari ilmu komputer yang diacu dalam terminologi algoritma, misalnya algoritma *Perutean* (*routing*) pesan di dalam jaringan komputer, algoritma *Brensenham* untuk menggambar garis lurus (bidang grafika komputer), algoritma *Knuth-Morris-Pratt* untuk mencari suatu pola di dalam teks atau *string mathcing* (bidang *information retrievel*), dan sebagainya. *String matching* atau pencocokkan *string* merupakan hal dasar yang sangat perlu dipelajari terutama dalam lingkup yang berkaitan dengan *text processing*. Secara umum, istilah yang terdapat dalam pencocokkan *string* antara lain teks dan *pattern*. Teks (*text*) adalah (*long*) *string* yang panjangnya n. Pattern yaitu *string* dengan panjang m karakter (m < n) yang akan di cari di dalam teks[2].

Terdapat beragam algoritma pencocokkan *string*, baik algoritma yang diciptakan dari awal maupun algoritma pengembangan, yang dapat digunakan untuk melakukan pencocokkan *string*. Salah satu algoritma pencocokkan *string* yang paling efisien adalah algoritma *Boyer-Moore* [2]. Menurut penelitian Kristanto, Rachmat dan Gunawan dalam jurnal *Implementasi Algoritma Boyer-Moore pada Permainan Word Search Puzzle* menyatakan bahwa algoritma *Boyer Moore* memiliki tingkat akurasi pada saat pencocokan *string* sebanyak 100%. Algoritma *Boyer-Moore* mempunyai empat konsep dasar di dalam proses pencocokkan *string*, yaitu *preporcessing, right-to-left-scan, bad-character-rule, good-suffix-rule*. Meskipun algoritma *Boyer-Moore* dipandang sebagai algoritma yang paling efisien untuk pencocokkan *string*, namun algoritma tersebut masih memiliki beberapa kekurangan, diantaranya yaitu tidak bagus untuk *binary string* dan lebih lambat untuk *pattern* yang pendek [3], sehingga muncul beberapa algorit-

ma pengembangan dari algoritma *Boyer-Moore*. Dua diantaranya adalah algoritma *Zhu-Takaoka* dan algoritma *Reverse Colussi*.

Algoritma *Zhu-Takaoka* memiliki karakteristik sama dengan algoritma *Boyer-Moore* yaitu *preprocessing, right-to-left-scan, bad-character-rule, good-suffix-rule*. Perbaikan yang dilakukan oleh algoritma *Zhu-Takaoka* adalah *bad-character-rule* yang menggunakan *array* dua dimensi untuk menghitung nilai pergeseran sedangkan algoritma *Boyer-Moore* hanya menggunakan *array* satu dimensi. Algoritma *Zhu-Takaoka* lebih cepat dalam pencarian kata dan kalimat dibandingkan dengan algoritma *Raita* [4].

Sementara itu, algoritma *Reverse Colussi* juga merupakan salah satu algoritma perbaikan dari algoritma *Boyer-Moore* yang idenya berasal dari algoritma *colussi*. Algoritma ini juga melakukan pencocokkan dari kanan ke kiri. Terdapat dua fase pencarian pada algoritma *Reverse Colussi*, yaitu fase pemrosesan awal dan fase pencarian. Pada fase pemrosesan awal dilakukan pencarian sepasang karakter, serta pencarian *special positions* dan *non-special position*. Sedangkan, pada fase pencarian dilakukan pencarian *pattern* terhadap teks. Algoritma *Reverse Colussi* menghasilkan jumlah perbandingan tiap karakter yang lebih sedikit dari pada algoritma *Boyer-Moore, Knuth-Morris-Pratt*, Dan *Colussi* [5]. Algoritma *Reverse Colussi* juga sangat cepat dalam melakukan pencarian *string*, tetapi membutuhkan waktu pemrosesan awal yang lambat. Hal ini dikarenakan harus menghasilkan dua tabel terlebih dahulu sebelum memulai pencocokkan [5].

Sebagaimana yang telah dijelaskan bahwa algoritma *Boyer-Moore* masih memiliki beberapa kekurangan. Sementara itu, algoritma *Reverse Colussi* dan algoritma *Zhu-Takaoka* merupakan pengembangan dari algoritma *Boyer-Moore* yang keduanya memiliki keunggulan dan kelemahan masing-masing, yakni dalam proses pencarian, pencocokkan, kecepatan, ketepatan, dan lain-lain. Namun demikian, belum dapat dipastikan apakah kedua algoritma pengembangan tersebut sudah memperbaiki kekurangan yang dimiliki algoritma *Boyer-Moore*. Selain itu, di antara kedua algoritma tersebut belum dapat dipastikan juga algoritma manakah yang lebih optimal dalam melakukan pencocokan *pattern*. Oleh karena itu, analisis perbandingan performansi terhadap kedua algoritma dapat dilakukan untuk

mengetahui tingkat akurasi dan waktu yang dibutuhkan dalam satu kali proses pencocokkan *string* yang digunakan oleh kedua algoritma tersebut.

Bahasa sunda memiliki sistem penulisan *aksara* sunda baku yaitu *aksara swara, aksara ngalagena, rarangken* dan angka. Setiap kata dalam tulisan bahasa sunda pasti mengandung keempat sistem penulisan tersebut. Karateristik dari *aksara ngalagena* memiliki pola kebanyakan dua huruf seperti ka, ca, ta, pa, ya, wa, ga, ja, da, ba, ra, sa, na, ha, ma, la, dan lain-lain. Selain itu dalam *aksara* sunda biasanya setiap kata memiliki karakter yang berulang seperti kata *awewe, lalaki, jelema, papanggih, babaturan, badag, baham, balangsak* dan lain-lain [6].

Berdasarkan permasalahan yang telah dipaparkan di atas, untuk menganalisis perbandingan performansi dari algoritma *Zhu-Takaoka* dan *Reverse Colussi* dalam melakukan pencocokan *string*, maka kedua algoritma tersebut akan diimplementasikan pada pencocokkan *string* menggunakan data sinopsis buku berbahasa sunda yang diambil dari Rumah Baca Buku Sunda. Pola dua karakter dari *aksara ngalagena* akan memperlihatkan kelebihan dari algoritma *Zhu Takaoka* pada saat pencocokan *string*, sedangkan kata dalam bahasa sunda yang memiliki karakter sering berulang akan menampilkan kelebihan dari algoritma *Reverse Colussi* dalam mencari *sepecial position*, analisis perbandingan performansi tersebut dilakukan untuk menentukan algoritma pengembangan *Boyer Moore* manakah yang lebih optimal berdasarkan hasil perbandingan akurasi dan kecepatan waktu proses pencocokan *string* untuk melakukan pencocokan karakter yang terdapat dalam sinopsis dari buku berbahasa sunda.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan masalah dari penelitian ini adalah bagaimana membandingkan algoritma *Zhu-Takaoka* dan algoritma *Reverse Colussi* untuk mengetahui tingkat akurasi yang dihasilkan dan waktu yang diperlukan pada proses pencocokkan *string* bahasa sunda.

1.3 Maksud dan Tujuan

Maksud dari penulisan skripsi ini adalah melakukan analisis terhadap kinerja algoritma *Zhu-Takaoka* dan algoritma *Reverse Colussi* dalam pencocokkan *string* bahasa sunda.

Tujuan yang akan dicapai pada penelitian ini adalah mengetahui akurasi dan waktu yang diperlukan pada saat melakukan pencocokkan *string* bahasa sunda oleh algoritma *Reverse Colussi* dan algoritma *Zhu-Takaoka*.

1.4 Batasan Masalah

Batasan masalah dalam melakukan perbandingan algoritma *Reverse Colussi* dan *Zhu-Takaoka* pada pencocokkan *string* ini adalah sebagai berikut.

1. Membandingkan performansi algoritma *Reverse Colussi* dan algoritma *Zhu-Takaoka* mengenai tingkat akurasi yang dihasilkan dan waktu yang digunakan dalam satu *pattern*.
2. *Pattern* atau kata yang dicari memiliki relevansi dengan sumber *string* yang ada pada sistem.
3. Panjang *pattern* yang digunakan pada penelitian ini, yaitu berjumlah 2 sampai 20 karakter karena menurut observasi dilapangan diketahui bahwa nama orang atau nama tempat yang diingat oleh pengunjung yang dijadikan kata kunci atau *pattern* biasanya berjumlah antara 2 sampai 20 karakter.
4. Sinopsis buku yang digunakan hanya buku berbahasa Sunda, berformat txt.
5. *Output* dari perangkat lunak adalah waktu *preprocessing* dan waktu pencocokan dari algoritma yang digunakan, jumlah karakter tak berulang pada sumber *string* yang dicari, dan jumlah pola yang ditemukan.
6. Parameter yang digunakan dalam analisis perbandingan adalah akurasi yang dihasilkan dan kecepatan waktu proses yang dibutuhkan oleh kedua algoritma.
7. Analisis pembangunan perangkat lunak menggunakan pendekatan analisis terstruktur.

1.5 Metodologi Penelitian

Metode penelitian eksperimen bertujuan untuk menyelidiki kemungkinan sebab akibat dengan cara melakukan percobaan berulang kepada satu atau lebih kondisi perlakuan dan membandingkan hasilnya [7].

Pada penelitian ini akan dilakukan perbandingan antara dua algoritma untuk menguji akurasi yang dihasilkan dan berapa waktu yang diperlukan dalam satu kali proses pencocokkan dengan beberapa kali percobaan.

Dari kedua pernyataan tersebut metode eksperimen adalah metode yang paling sesuai untuk digunakan pada penelitian ini. Tahapan yang akan dilalui pada penelitian ini adalah sebagai berikut.

1. Tahap Pengumpulan Data

Metode pengumpulan data yang digunakan pada penelitian ini antara lain :

a. Studi literatur.

   Studi Literatur dilakukan dengan cara mengumpulkan dan mempelajari jurnal, *paper*, makalah dan artikel yang berkaitan dengan masalah yang akan dibahas dalam pembuatan tugas akhir ini.

b. Studi lapangan.

   Studi lapangan dilakukan untuk mengamati kegiatan secara langsung dilapangan dan mengumpulkan semua informasi yang dibutuhkan.

c. Wawancara.

   Wawancara dilakukan kepada Bapak Mamat Sasmita selaku pengelola sekaligus pemilik dari Rumah Baca Buku Sunda untuk mengetahui segala informasi yang dibutuhkan pada penelitian ini.

2. Tahap Analisis Algoritma

Teknik analisis algoritma menggunakan teknik eksperimen dengan cara melakukan perbandingan antara algoritma *Zhu-Takaoka* dan algoritma *Reverse Colussi* terhadap akurasi yang dihasilkan dan kompleksitas waktu yang digunakan dalam pencocokkan *string* pada aplikasi Perbandingan Algoritma *Zhu-Takaoka* dan Algoritma *Reverse Colussi*.

3. Tahap Pembangunan Perangkat Lunak

Waterfall model adalah model yang mengusulkan sebuah pendekatan kepada pembangunan *software* yang sistematik dan sekuensial, metode ini sangat baik digunakan untuk produk *software* yang sudah jelas kebutuhannya di awal, untuk meminimalisir kesalahan[8]. Dari pernyataan tersebut *waterfall model* dirasa model pembangunan perangkat lunak yang sesuai pada penelitan ini. Berikut beberapa tahapan *waterfall model*.

a. *Requirements Analysis and Definition*

   Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

b. *Sistem and Software Design*

   Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.

c. *Implementation and Unit Testing*

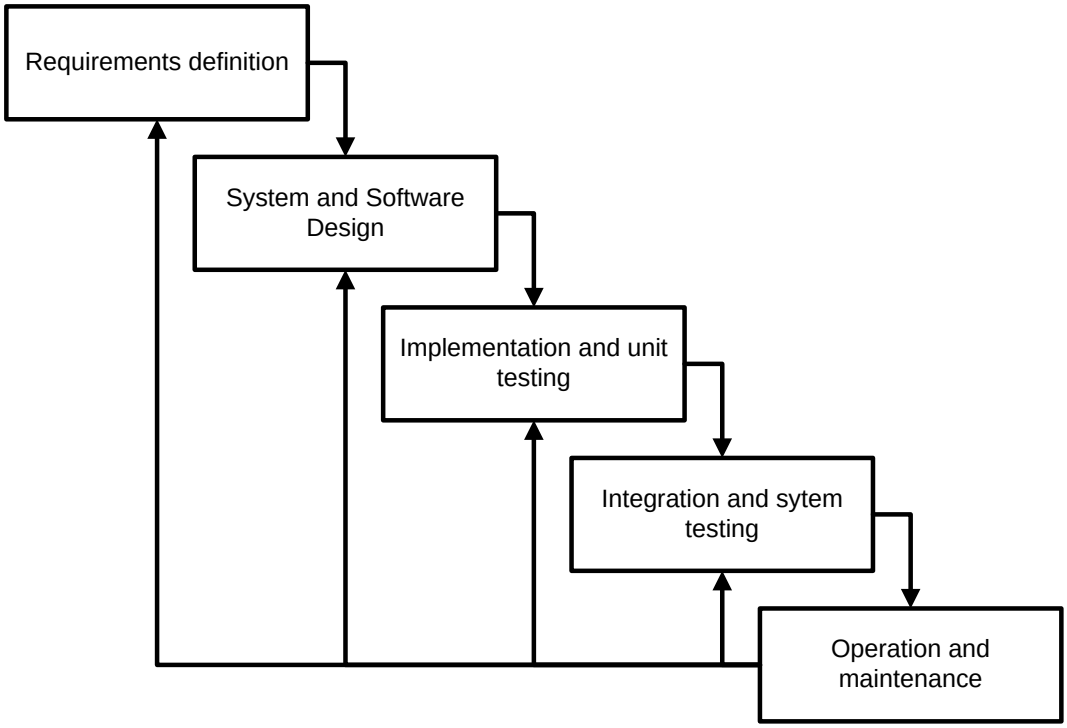
   Desain program diterjemahkan kedalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji secara unit.

d. *Integration and Sistem Testing*

   Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).

e. *Operation and Maintenance*

   Mengoperasikan program di lingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.



Gambar 1.1 *Waterfall*[8]

4. Tahap Pengujian

Pengujian yang dilakukan pada penelitian ini adalah menggunakan pengujian *Unit Testing*. Hasil dari pengujian ini digunakan untuk menarik kesimpulan yang nantinya akan jadi hasil akhir.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi dibuat untuk memberikan gambaran umum mengenai penelitian yang dilaksanakan. Penyusunan skripsi ini dibagi ke dalam lima bab yang disusun secara sistematis sesuai dengan pokok-pokok permasalahan yang dibahas. Sistematika penulisan tugas akhir ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi penjelasan mengenai latar belakang masalah, perumusan masalah, maksud dan tujuan, batasan masalah, metodologi penelitian dan sistematika penulisan yang dilakukan pada penelitian ini.

BAB 2 TINJAUAN PUSTAKA

Bab ini membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian seperti pengertian *String*, *String Matching*, Algoritma, Algoritma *String Matching*, Algoritma *Zhu-Takaoka*, Algoritma *Reverse Colussi*, Kompleksitas Algoritma, Notasi Big-O, Metode Pembangunan Perangkat Lunak yang digunakan, Bahasa Pemrograman yang digunakan, Tools yang digunakan, Pemodelan Data yang dilakukan, pendekatan untuk pembangunan software yang digunakan, dan pengujian algoritma yang digunakan.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi pembahasan mengenai analisis dan perancangan algoritma *Zhu-Takaoka* dan algoritma *Reverse Colussi* yang akan diimplementasikan pada aplikasi Perbandingan Algoritma *Zhu-Takaoka* dan Algoritma *Reverse Colussi*.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini berisi pembahasan mengenai implementasi algoritma *Zhu-Takaoka* dan algoritma *Reverse Colussi* dari hasil tahapan analisis algoritma dan perancangan sistem yang dibangun, serta uji coba dan hasil pengujian sistem menggunakan Unit Testing.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh dari hasil pengujian yang dilakukan untuk mengetahui tingkat akurasi dan waktu yang dibutuhkan oleh algoritma *Zhu-Takaoka* dan *Reverse Colussi* pada saat mencocokkan *string* dalam satu kali proses dan saran untuk pengembangan selanjutnya.