BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Bahasa ialah komunikasi yang paling lengkap dan efektif untuk menyampaikan ide, pesan, maksud, perasaan dan pendapat kepada orang lain [1]. Suatu tata bahasa harus bisa dibuktikan secara ilmiah oleh setiap orang, di setiap tempat dan pada setiap waktu [2].

Oleh karena itu, seorang penulis harus memperhatikan penulisan dan struktur kalimat dalam suatu bahasa karena penulisan dan struktur kalimat akan berpengaruh dalam proses penyampaian dan penerimaan pesan. Hal inilah yang mendasari perlunya diadakan penelitian tentang perbaikan kesalahan penulisan kata dan pemeriksaan struktur kalimat.

Berdasarkan penelitian Alinda dan Kania evita dewi pada "Perbandingan Algoritma *Reverse Colusi* dengan Algoritma *Zhu-Takaoka* dalam Pencocokan *String*". Algoritma *Zhu-Takaoka* dan *Reverse Colusi* memiliki tingkat akurasi sebanyak 100% dalam pencocokan *string*, kemudian Algoritma *Reverse Colussi* membutuhkan waktu rata-rata *preprocessing* selama 15.08 ms dan waktu rata-rata pencocokan *pattern* selama 0.13 ms sedangkan Algoritma *Zhu-Takaoka* membutuhkan waktu rata-rata *preprocessing* selama 5.43 ms dan waktu rata-rata pencocokan *pattern* selama 0.05 ms. Dapat disimpulkan bahwa dari kedua algoritma pengembangan *Boyer Moore*, algoritma *Zhu Takaoka* lebih cepat pada proses pencocokan *string* [3], algoritma *Zhu-takaoka* hanya bisa mencocokkan kata tidak bisa memperbaiki kata sehingga dibutuhkan algoritma yang bisa memperbaiki kata.

Berdasarkan penelitian Ni Made Muni Adiyani, I Wayan Santiyasa, Agus Muliantara pada "Implementasi Algoritma *Levenshtein Distance* dan Metode Empiris untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Dokumen Berbahasa Indonesia", terdapat cara untuk membantu mengatasi permasalahan

pada kesalahan pengetikan kata [4]. Karena pada penelitian ini masih sebatas pengecekan kesalahan pengetikan bukan pengecekan struktur kalimat, maka dibutuhkan algoritma untuk pengecekan struktur kalimat.

Berdasarkan penelitian Sri Susanti pada "Analisis Perbandingan Algoritma LCP (*Left-Corner-Parsing*) dan Algoritma CYK (*Cocke-Younger-Kasami*) untuk Memeriksa Pola Kalimat Baku Bahasa Indonesia", terdapat cara untuk mengenali struktur kalimat dengan menggunakan algoritma LCP dan CYK. Dari hasil pengujian yang dilakukan Algoritma CYK memiliki akurasi sebesar 65% dengan laju *error* 0, 35 dan Algoritma LCP memiliki akurasi sebesar 60% dengan laju *error* 0, 40 [5]. Sehingga dapat disimpulkan bahwa Algoritma CYK lebih baik dalam mengenali struktur kalimat.

Berdasarkan permasalahan di atas, penelitian ini akan dilakukan penerapan algoritma *Zhu-takaoka, Levenshtein* dan CYK untuk memperbaiki kesalahan penulisan kata dan perbaikan struktur kalimat bahasa Indonesia pada abstrak skripsi. Algoritma *Zhu-Takoaka* digunakan untuk pencarian kata, algoritma *Levenshtein* digunakan untuk perbaikan kata dan memperbaiki struktur kalimat dan algoritma CYK digunakan untuk mengenali *tag* kata.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dari penelitian ini adalah bagaimana menerapkan *algoritma Zhu-Takaoka, Levenshtein* dan CYK untuk kesalahan penulisan kata dan perbaikan struktur kalimat dalam bahasa Indonesia.

1.3 Maksud dan Tujuan

Maksud penelitian ini adalah menerapkan algoritma *Zhu-Takoaka, Levenshtein* dan CYK pada simulator untuk perbaikan kesalahan penulisan kata dan struktur kalimat.

Tujuan penelitian ini adalah untuk mengetahui algoritma *Zhu-Takaoka, Levenshtain* dan CYK dapat memberikan rekomendasi perbaikan kesalahan penulisan kata dan struktur kalimat dalam bahasa Indonesia pada abstrak skripsi.

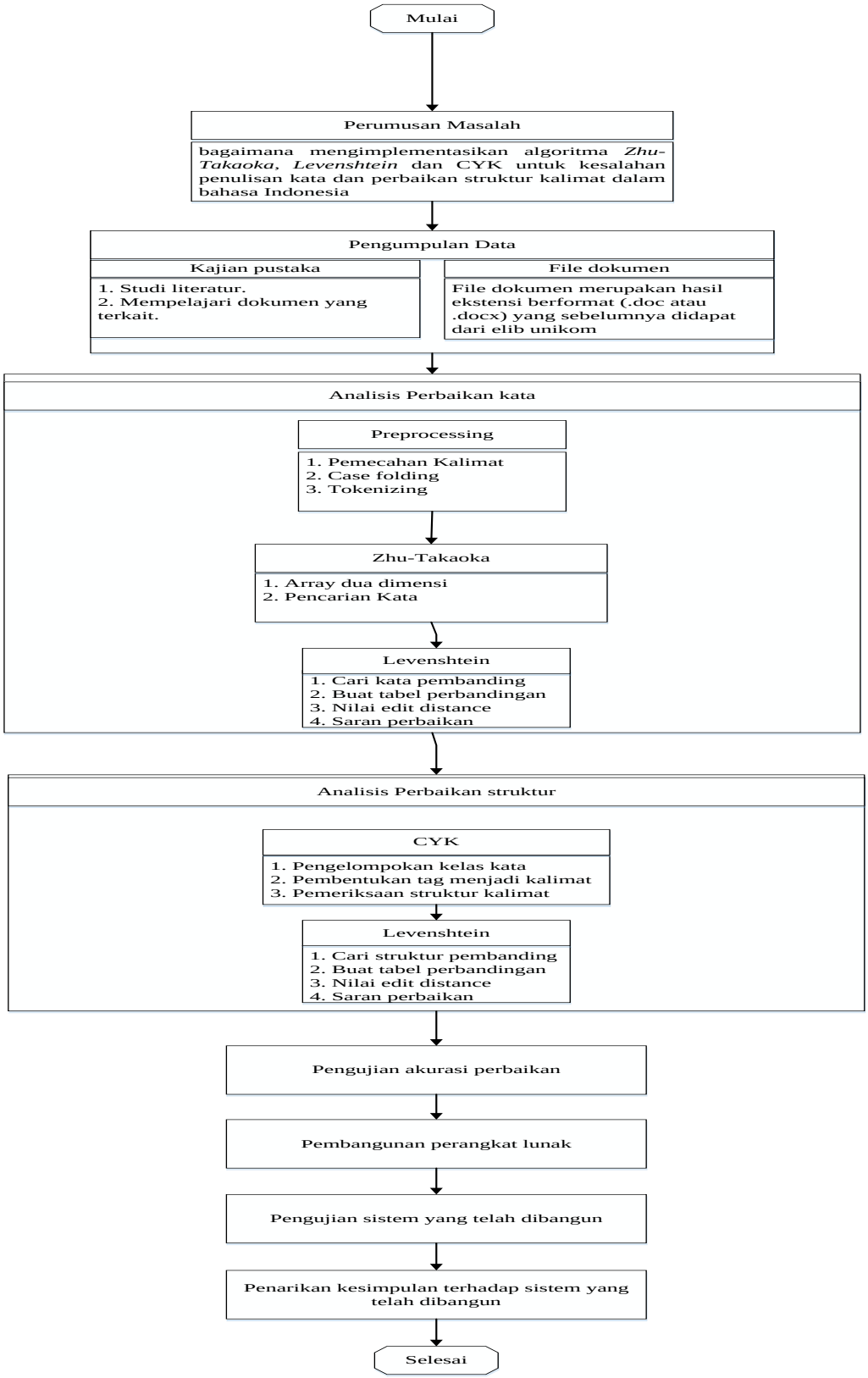
1.4 Batasan Masalah

Batasan masalah pada algoritma *Zhu-Takaoka*, *Levenshtein* dan CYK dalam kasus perbaikan kesalahan penulisan kata adalah sebagai berikut.

1 Pengecekan terbatas hanya pada bahasa Indonesia.
2 Kamus dibuat berdasarkan POS *Tag* Indonesia.
3 Dokumen yang diperiksa berupa inputan dari *file* dengan ekstensi *.doc* atau *.docx.*
4 Pengecekan kalimat hanya kalimat sederhana dan kalimat luas.
5 Pengecekan kata tidak termasuk mengenali kata yang ambigu.
6 Data uji yang digunakan adalah dokumen abstrak skripsi dari elib Unikom.
7 Bahasa pemrograman yang digunakan adalah *C#*.

1.5 Metodelogi Penelitian

kuantitatif adalah suatu proses menemukan pengetahuan yang menggunakan data berupa angka sebagai alat menganalisis keterangan mengenai apa yang ingin diketahui [6]. Dapat dilihat pada Gambar 1.1.



Gambar 1.1 Metodelogi penelitian

berdasarkan alur penelitian dapat dijelaskan sebagai berikut.

1. **Perumusan Masalah**

   Tahapan kedua adalah perumusan masalah. Peneliti merumuskan masalah bagaimana mengimplementasikan algoritma *Zhu-Takaoka*, *Levenshtein* dan CYK untuk kesalahan penulisan kata dan perbaikan struktur kalimat dalam bahasa Indonesia.

2. **Pengumpulan Data**

   Tahap ketiga adalah pengumpulan data. Metode pengumpulan data yang dilakukan pada penelitian ini yaitu :

   a. **Kajian Pustaka**

      Kajian pustaka dilakukan dengan mempelajari jurnal, *paper,* dan buku yang terkait dengan topik perbaikan kata, pemeriksaan struktur, tahapan *preprocessing, zhu-takaoka, levenshtein, CYK* dan pengujian akurasi perbaikan.

   b. **Abstrak Skripsi**

      Abstrak skripsi diambil dari elib UNIKOM, abstrak skripsi tersebut kemudian di estensi ke format *file* (.doc atau .docx).

3. **Analisis Perbaikan Kata**

   Tahapan ini berupa analisis perbaikan kata. Adapun analisis yang dilakukan dalam penelitian ini yaitu :

   a. ***Preprocessing***

      Preprocessing merupakan tahapan untuk mempersiapkan teks menjadi data yang akan diolah pada tahapan berikutnya. *Preprocessing* pada penelitian ini terdiri dari beberapa tahapan yaitu proses pemecahan kalimat, *case folding,* dan *tokenizing.*

   b. ***Zhu-Takaoka***

      Merupakan proses pencocokan *string* untuk mencari *tag*. Meliputi pencarian nilai *bad character*, *good suffix* dan pencocokan.

c. ***Levenshtein***

Merupakan proses perbaikan kata pada kata yang tidak memiliki *tag*. Meliputi membentuk tabel perbandingan kata, nilai *edit distance,* dan menghasilkan saran perbaikan kata.

4. **Analisis Perbaikan Struktur**

a. **CYK**

Merupakan proses pemeriksaan struktur, meliputi pembentukan *tag* menjadi kalimat, pemeriksaan struktur kalimat.

b. ***Levenshtein***

Merupakan proses perbaikan pola pada kata yang tidak memiliki *tag*. Meliputi membentuk tabel perbandingan kata, nilai *edit distance,* dan menghasilkan saran perbaikan kata.

5. **Pengujian Akurasi Ringkasan**

Tahapan ini merupakan pengujian akurasi hasil perbaikan kata dan struktur.

6. **Pembangunan Perangkat Lunak**

Metode pembangunan perangkat lunak yang digunakan adalah model *prototype.* Tahapan proses model *prototype* adalah sebagai berikut :

a. Pengumpulan Kebutuhan

Tahapan ini adalah mengumpulkan data dengan cara mengumpulkan jurnal, *paper,* artikel, buku dan bacaan-bacaan yang berkaitan dengan penelitian ini.

b. Perancangan

Yaitu menganalisis masalah yang terbagi menjadi analisis sistem, analisis data masukan, analisis metode, analisis kebutuhan fungsional, analisis kebutuhan non-fungsional, perancangan struktur menu, perancangan antarmuka, dan jaringan semantik.

c. Evaluasi *Prototype*

Yaitu melakukan pengujian dengan melihat akurasi dari hasil penelitian terhadap *software* yang dibangun.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan laporan penelitian ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi penjelasan tentang latar belakan masalah yang menjadi bahan penelitian, merumuskan masalah, menentukan batasan masalah, memaparkan maksud dan tujuan dari penelitian yang dilakukan, menjelaskan metodelogi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi tentang pembahasan teori *text Mining*. Penyederhanaan bentuk CNF. sintaksis bahasa Indonesia meliputi frase, klausa dan kalimat. definisi Algoritma *Zhu-Takaoka*, defensi Algoritma *Levenshtein*, definisi Algoritma CYK, Pengenalan pemrograman berbasis objek, dan konsep dasar dari UML (*unified modelling language*) dan definisi pemrograman C#.

BAB III Analisis Pengenalan Kesalahan Penulisan Kata dan Pemeriksaan Struktur

Pada bab ini membahas mengenai proses *preprocessing*, proses pencarian kata algoritma *Zhu-Takaoka*, proses perbaikan kata menggunakan algoritma *Levenshtein*, proses pemilihan kata, proses pengelompokan kata menggunakan algoritma CYK, proses algoritma CYK dalam pemeriksaan struktur kalimat, proses algoritma *Zhu-Takaoka* dalam mencocokkan struktur kalimat untuk saran perbaikan. Selain itu, perancangan UML hingga perancangan antarmuka dari simulator yang akan dibuat dibahas secara detail pada bab ini.

BAB IV

Pada bab ini akan dijelaskan tahapan-tahapan implementasi dari kebutuhan algoritma dan pengujian berdasarkan analisis yang sudah dilakukan sebelumnya. Mulai dari *input* dokumen, pencarian kata dengan cara membandingkan dengan kata yang ada pada kamus kata menggunakan algoritma *Zhu-Takaoka*. Pada saat setelah kata di cari kata tersebut akan diperbaiki dengan cara membandingkannya dengan kata yang ada pada kamus menggunakan algoritma *Levenshtein*. Pada saat setelah kata diperbaiki algoritma CYK kemudian sistem (simulator) akan mengecek setiap kata dan mengelompokkan sesuai dengan diksi dan akan memeriksa struktur kalimat dan di saat bersamaan algoritma *Levenshtein* akan memberikan saran untuk perbaikan struktur kalimat sesuai dengan bentuk CNF yang telah dianalisis dari sintaksis Bahasa Indonesia. Hasil akhir dari pengecekan sistem akan memberikan *output* berapa banyak saran perbaikan kata, pemeriksaan dan perbaikan struktur kalimat.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi mengenai kesimpulan dari hasil pengujian algoritma yang telah dilakukan. Kekurangan dari implementasi ini dibahas pada bab ini agar dapat dilakukan pengembangan yang lebih baik. Bab ini juga berisi saran-saran yang diberikan untuk perbaikan.