BAB 1

PENDAHULUAN

1.1. Latar Belakang Masalah

Bahasa merupakan alat komunikasi lingual manusia baik secara lisan maupun tulisan. Dalam membuat suatu karya ilmiah, penggunaan Bahasa Indonesia harus sesuai dengan EYD (Ejaan yang Disempurnakan). Struktur atau pola kalimat yang disusun dari kata-kata yang dibentuk harus dapat membuat pembaca mudah mengerti maksud dari tulisan pada karya ilmiah tersebut. Kesalahan dalam pengetikkan dapat membuat kesalahpahaman para pembaca. Kesalahan struktur kalimat seperti tidak diketahui subjeknya juga dapat membuat pembaca tidak mengetahui siapa pelaku yang disebutkan dalam kalimat tersebut.

Berdasarkan sebuah penelitian [1], pemeriksaan kesalahan pengetikkan biasa dilakukan ketika tulisan telah selesai dibuat. Tidak sulit jika pemeriksaan dilakukan terhadap ukuran teks yang pendek. Apabila ukuran teks sudah mencapai lebih dari ratusan atau bahkan ribuan akan menyulitkan pemeriksaan. Dengan penerapan sebuah program, masalah tersebut dapat diatasi. Tidak hanya kesalahan kata, pemeriksaan pola kalimat dalam sebuah karya ilmiah pun perlu dilakukan.

Terdapat penelitian [2] yang telah melakukan perbaikan kata pada kesalahan pengetikkan menggunakan algoritma Levenshtein dengan mekanisme penambahan, penyisipan, dan penghapusan karakter. Algoritma Levenshtein merupakan algoritma yang digunakan untuk mencari jumlah operasi *string* yang paling sedikit untuk mentransformasikan suatu *string* menjadi *string* yang lain [3]. Algoritma ini menggunakan tabel dua dimensi dalam melakukan pengecekkan *string*. Jumlah operasi *string* disebut sebagai *edit distance*. *Edit distance* diambil dari hasil akhir (baris dan kolom terakhir) pada tabel. Hasil inilah yang akan digunakan untuk melihat apakah suatu *string* dapat diubah menjadi *string* lain. Apabila hasil akhir bernilai 0, maka tidak ada yang perlu diperbaiki. Jika nilai hasil akhir lebih dari 0, maka terdapat kesalahan dalam *string*. Semakin kecil nilai *edit distance*, semakin dekat dengan *string* yang dimaksud. Pada penelitian ini

pengecekkan masih sebatas pengecekkan kesalahan pengetikkan, bukan pengecekkan pola kalimat Berbahasa Indonesia. Sehingga belum diketahui cara perbaikan pola kalimat menggunakan algoritma Levenshtein.

Pada penelitian lain [4] terdapat cara untuk mengenali pola kalimat dengan menggunakan algoritma CYK. Algoritma CYK adalah algoritma untuk menentukan apakah suatu untai dapat diterima oleh suatu tata bahasa bebas-konteks (CFG) yang sudah dinormalkan dalam bentuk *Chomsky Normal Form* (CNF). Pengecekkan dilakukan dengan mengelompokkan setiap kata sesuai dengan KBBI (Kamus Besar Bahasa Indonesia). Aturan-aturan pengelompokkan kata dapat dibuat dalam bentuk CNF.

Oleh karena itu dalam penelitian ini akan dilakukan implementasi dari kedua algoritma yaitu algoritma CYK dan Levenshtein untuk memperbaiki struktur kalimat dan kesalahan pengetikkan Bahasa Indonesia pada suatu karya ilmiah. Algoritma CYK digunakan untuk mengenali pola kalimat dan algoritma Levenshtein digunakan sebagai bantuan untuk memperbaiki pola kalimat dan kesalahan pengetikkan Bahasa Indonesia.

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang diatas, maka rumusan masalah dari penelitian ini adalah bagaimana mengimplementasikan algoritma CYK dan Levenshtein untuk memperbaiki struktur kalimat dan kesalahan pengetikkan Bahasa Indonesia pada suatu karya ilmiah.

1.3. Maksud dan Tujuan

Adapun maksud dari penelitian ini adalah mengimplementasikan algoritma CYK dan Levenshtein ke dalam sebuah simulator perbaikan struktur kalimat dan kesalahan pengetikkan.

Tujuan dari penelitian ini untuk mengetahui sejauh mana algoritma CYK dan Levenshtein dapat memberikan rekomendasi perbaikan kata dan pola kalimat Bahasa Indonesia pada suatu karya ilmiah.

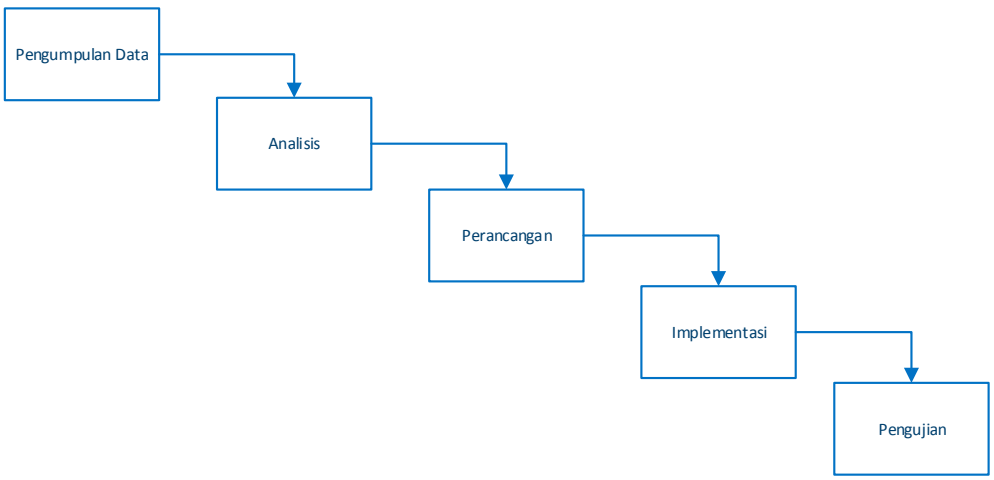
1.4. Batasan Masalah

Agar pembahasan menjadi lebih jelas dan terarah, maka permasalahan dibatasi terhadap masalah-masalah berikut:

1. Pemeriksaan kalimat hanya terbatas pada Bahasa Indonesia saja.
2. Aturan produksi dibuat dalam bentuk CNF agar dapat dilakukan *parsing* oleh algoritma CYK.
3. Kamus dan pengelompokkan kata diambil berdasarkan POS Tag Indonesia.
4. Seluruh kata dalam Bahasa Inggris akan dianggap sebagai istilah, artinya seluruh istilah termasuk ke dalam kelompok kata nomina.
5. Kalimat yang diperiksa berupa inputan dari *file* dengan ekstensi .doc atau .docx
6. Pengecekkan sintaksis hanya berupa frase, klausa, dan kalimat.
7. Pengecekkan kalimat hanya berupa kalimat sederhana dan kalimat luas.
8. Pengecekkan kalimat tidak termasuk mengenali kalimat yang ambigu.
9. Data uji yang digunakan adalah dokumen abstrak skripsi dari Teknik Informatika Unikom.
10. Bahasa pemrograman yang digunakan adalah C#.

1.5. Metodologi Penelitian

Tahapan penelitian dari tugas akhir yang akan dilalui yaitu:



Gambar 1.1 Tahap metodologi penelitian

1. Pengumpulan Data

   Tahap ini merupakan tahap awal dari penelitian. Pada tahap ini dilakukan pengumpulan data dengan menggunakan metode studi literatur dan wawancara. Pengumpulan data meliputi teori tentang Bahasa Indonesia dan sintaksisnya, algoritma CYK, algoritma Levenshtein, dan bahasa pemrograman C#.

2. Analisis

   Pada tahap ini dilakukan analisis terhadap kedua algoritma yaitu CYK dan Levenshtein. Analisis dimulai dari memahami sintaksis Bahasa Indonesia, membuat bentuk CNF dari sintaksis yang sudah dianalisis, memahami cara kerja algoritma CYK dan Levenshtein, hingga menganalisis cara penggabungan kedua algoritma.

3. Perancangan

   Tahap ini merupakan tahap untuk perancangan *simulator* yang akan dibuat berdasarkan analisis yang telah dilakukan. Pada tahap ini dilakukan perancangan diagram UML (*Unified Language Model*) dan perancangan antarmuka (*user interface*).

4. Implementasi

   Tahap ini menerjemahkan hasil perancangan yang telah dibuat ke dalam bahasa pemrograman. *Tools* yang digunakan yaitu Visual Studio 2012 dengan bahasa pemrograman C#.

5. Pengujian

   Setelah *simulator* dibuat maka selanjutnya melakukan pengujian algoritma melalui *simulator* yang telah dibuat. Pengujian dilakukan dengan memasukkan dokumen abstrak skripsi ke *simulator* dengan menghasilkan keluaran yaitu saran perbaikan kata dan pola kalimat.

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 belakang masalah yang menjadi bahan penelitian, merumuskan masalah, menentukan batasan masalah, memaparkan maksud dan tujuan dari penelitian yang dilakukan, menjelaskan metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi tentang pembahasan teori sintaksis Bahasa Indonesia, mempelajari teori bahasa dan automata dengan melihat pada hirarki *Chomsky*, mengenal tata bahasa bebas konteks dan penyederhanaannya, penyederhanaan bentuk CNF, definisi Algoritma CYK berikut dengan cara kerjanya, definisi Algoritma Levenshtein berikut dengan cara kerjanya, pengenalan pemrograman berbasis objek, dan konsep dasar dari UML (*Unified Modelling Language*).

BAB III ANALISIS DAN PERANCANGAN

Pada bab ini dibahas mengenai kebutuhan algoritma CYK dalam mem-*parsing string* agar dapat dikenali struktur kalimatnya. Pada bab sebelumnya telah dibahas mengenai sintaksis Bahasa Indonesia. Bab ini juga menjelaskan bagaimana sintaksis tersebut akan diolah menjadi bentuk CNF agar algoritma CYK dapat melakukan *parsing string*. Bab ini juga membahas kebutuhan algoritma Levenshtein dalam memperbaiki struktur kalimat serta mengoreksi kesalahan ejaan pada dokumen Berbahasa Indonesia yang pengujiannya akan dilakukan pada abstrak skripsi di Teknik Informatika Unikom. Selain itu, perancangan UML hingga perancangan antarmuka dari *simulator* yang akan dibuat dibahas secara detil pada bab ini.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dijelaskan tahapan-tahapan implementasi dari kebutuhan algoritma dan pengujian berdasarkan analisis yang sudah dilakukan sebelumnya. Mulai dari *input* dokumen, kemudian sistem (*simulator*) akan mengecek setiap kata dan mengelompokkannya sesuai dengan diksi atau kelompok katanya dengan bantuan POS Tag Bahasa Indonesia. Apabila sistem menemukan kata yang termasuk ke dalam kelompok kata X (*unknown*), kata tersebut akan dicek dengan

cara membandingkannya dengan kata yang ada pada kamus menggunakan algoritma Levenshtein. Pada saat yang bersamaan algoritma Levenshtein akan memberikan saran untuk kalimat yang termasuk ke dalam kelompok kata X digantikan sesuai dengan bentuk CNF yang telah dianalisis dari sintaksis Bahasa Indonesia. Hasil akhir dari pengecekkan sistem akan memberikan *output* berapa banyak saran perbaikan kata dan pola 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.