

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi, maka kegunaan komputer dirasa makin besar. Komputer berperan penting dalam mempermudah pekerjaan sehari – hari. Salah satu manfaat utama komputer adalah sebagai alat bantu untuk membuat karya tulis. Berbagai aplikasi seperti Ms. Word, Notepad, maupun *Open Office Word* sudah ditanamkan pada komputer untuk mempermudah pengguna[3].

Meskipun berbagai aplikasi sudah tersedia, bukan berarti penggunaanya tanpa masalah. Masalah yang dimaksud adalah adanya kesalahan penulisan ejaan dalam prakteknya, yang seringkali disebabkan karena keteledoran pengguna[2]. Berdasarkan hasil survei yang dilakukan terhadap 25 mahasiswa tingkat akhir 88% mahasiswa mendapatkan revisi terkait kesalahan penulisan ejaan bahasa Indonesia[18].

Kesalahan tersebut bisa disebabkan oleh ketidaktahuan penulisan, kesalahan yang berhubungan erat dengan posisi tombol papan ketik dan pergerakan jari, dan kesalahan transmisi dan penyimpanan yang berhubungan dengan pengkodean pada jalur mekanisme transmisi data. Kesalahan-kesalahan yang umumnya terjadi antara lain: penggantian satu huruf, penyisipan satu huruf, penghilangan satu huruf, maupun penukaran dua huruf berdekatan[1].

Proses pengecekan kesalahan pengetikan dengan cara manual akan menghabiskan banyak waktu dan membutuhkan suatu sumber pasti sebagai acuan bahwa kata tersebut memang salah dalam penulisannya[1]. Pengecekann ejaan

secara manual tentunya akan membutuhkan waktu yang lama dan cukup membosankan sehingga kemungkinan adanya *human error* dapat menyebabkan proses pengecekan kata menjadi tidak optimal.

Perangkat lunak pengolah kata seperti *Microsoft Word* telah memberikan fasilitas pendukung untuk pemeriksaan ejaan (*spelling and grammar checker*). Akan tetapi, fasilitas tersebut hanya dapat diaplikasikan untuk Bahasa Inggris. Meskipun bisa menambahkan kata sendiri dalam *dictionary list* pada *custom dictionary*, namun hal tersebut tidak menjamin jalannya penggunaan *spelling and grammar checker* untuk kata Bahasa Indonesia[9].

Untuk membantu proses pengecekan kesalahan ejaan kata bahasa Indonesia diperlukan sebuah program aplikasi yang dapat membantu pekerjaan seorang penulis untuk mengedit naskah tulisan maupun skripsi. Dalam pengecekan ejaan kata, apabila ditemukan kata yang salah maka harus dilakukan pencarian kemungkinan kata yang sesuai. Pada proses pencarian kemungkinan kata tersebut, diperlukan suatu pendekatan pencarian *string* khusus yaitu dengan pendekatan perkiraan (*Approximate String Matching*)[9].

. *Approximate string matching* dapat digunakan untuk pencarian *string* berdasarkan *string* yang sama dan *string* yang memiliki kemiripan penulisan dengan *string* yang terdapat pada kamus. Metode ini dapat digunakan untuk pencarian kata tidak baku karena dapat mengidentifikasi *string* yang sama dan yang memiliki kemiripan penulisan[19]. Pada *Approximate string matching* terdapat tiga macam operasi yang digunakan untuk mentransformasikan suatu *string* menjadi *string* yang lain. Operasi tersebut antara lain operasi penghapusan, penyisipan, dan penggantian. Operasi-operasi ini digunakan untuk menghitung jumlah perbedaan

yang diperlukan untuk pertimbangan kecocokan suatu *string* dengan *string* sumber. Jumlah perbedaan tersebut diperoleh dari penjumlahan semua perubahan yang terjadi dari masing-masing operasi. Penggunaan perbedaan tersebut diaplikasikan dalam berbagai macam algoritma[9], salah satunya algoritma *Levenshtein Distance*.

Algoritma *Levenshtein*, atau sering disebut dengan *Levenshtein Distance* atau *Edit distance* merupakan algoritma pencarian jumlah perbedaan *string* yang ditemukan oleh Vladimir Levenshtein, seorang ilmuwan Rusia, pada tahun 1965[9]. Algoritma *Levenshtein Distance* bekerja dengan menghitung jumlah minimum pentransformasian suatu *string* menjadi *string* lain yang meliputi penghapusan, penyisipan, dan penggantian[19].

Berdasarkan permasalahan tersebut terdapat permasalahan yang menarik yaitu bagaimana koreksi kesalahan ejaan pada dokumen teks secara otomatis oleh sistem sehingga dapat menghasilkan karya ilmiah yang sesuai dengan kaidah bahasa Indonesia yang baik dan benar. Sehingga disusunlah penelitian yang berjudul **“Implementasi Algoritma *Levenshtein Distance* Untuk Koreksi Kesalahan Ejaan Bahasa Indonesia Pada Dokumen Teks *Microsoft Word*”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, pokok permasalahan dalam penelitian ini adalah Bagaimana menerapkan algoritma *Levenshtein Distance* untuk koreksi kesalahan ejaan bahasa Indonesia pada dokumen teks *Microsoft Word* secara akurat?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dibuat, maka tujuan dari penelitian adalah menerapkan algoritma *Levenshtein Distance* untuk koreksi kesalahan ejaan bahasa Indonesia pada dokumen teks *Microsoft Word* secara akurat.

1.4 Manfaat Penelitian

Manfaat yang dapat diambil dalam penelitian ini adalah:

1. Membantu dalam koreksi ejaan bahasa Indonesia pada dokumen teks yang dapat digunakan kapanpun dan dimanapun.
2. Menambah pengetahuan tentang analisa metode *Levenshtein Distance* yang diterapkan dalam penelitian.

1.5 Batasan Masalah

Agar permasalahan dari penelitian ini tidak terlalu melebar sehingga penelitian ini menjadi fokus, maka penelitian ini dibatasi pada:

1. Algoritma yang digunakan untuk mencari jarak perbedaan antara dua *string* yaitu algoritma *Levenshtein Distance*.
2. Penelitian ini membaca dokumen teks berformat *.docx* dan *.pdf*.
3. Hanya teks dan kata bahasa Indonesia yang akan dikoreksi.
4. Sistem yang akan dibangun berbasis *web* menggunakan bahasa pemrograman PHP dan *Framework CodeIgniter*.
5. Ukuran maksimal dokumen yang diinputkan 20 MB.
6. Sistem mengoreksi kesalahan ejaan berupa penggantian satu huruf, penyisipan satu huruf, penghilangan satu huruf dan penukaran dua huruf yang berdekatan.

7. Data target pada sistem menggunakan 6000 kata bahasa Indonesia yang difilter dari 48000 kata yang berasal dari <http://indodic.com/IndoWordList.zip>.
8. Data target pada sistem menggunakan 6000 kata bahasa Indonesia tidak termasuk nama negara, nama hewan, nama buah dan singkatan.
9. Data target pada sistem berupa kata-kata bahasa Indonesia meliputi kata dasar, kata imbuhan me, ber, ter dan me-kan.
10. Hasil *Output* dokumen hasil koreksi berupa dokumen *.pdf* atau *.docx*.

1.6 Kerangka Pemikiran

Dalam kerangka pemikiran terdapat beberapa langkah atau proses:

1. *Problems*

Permasalahan yang ditemukan merupakan hasil penelitian serta analisis gejala yang terjadi sebagai latar belakang dan kemudian akan dipecahkan dengan sebuah metode dan implementasi.

2. *Opportunity*

Dari permasalahan yang telah diteliti diperoleh peluang – peluang yang bermanfaat dari penelitian ini. Peluang yang didapat akan diterapkan sebagai tujuan dari penelitian.

3. *Approach*

Permasalahan yang telah didapat dan peluang yang dihasilkan dari analisis permasalahan kemudian menggunakan sebuah pendekatan yang bertujuan terciptanya program yang diinginkan.

4. *Software Development*

Menganalisis dan merancang apa saja yang diperlukan dalam pembuatan program.

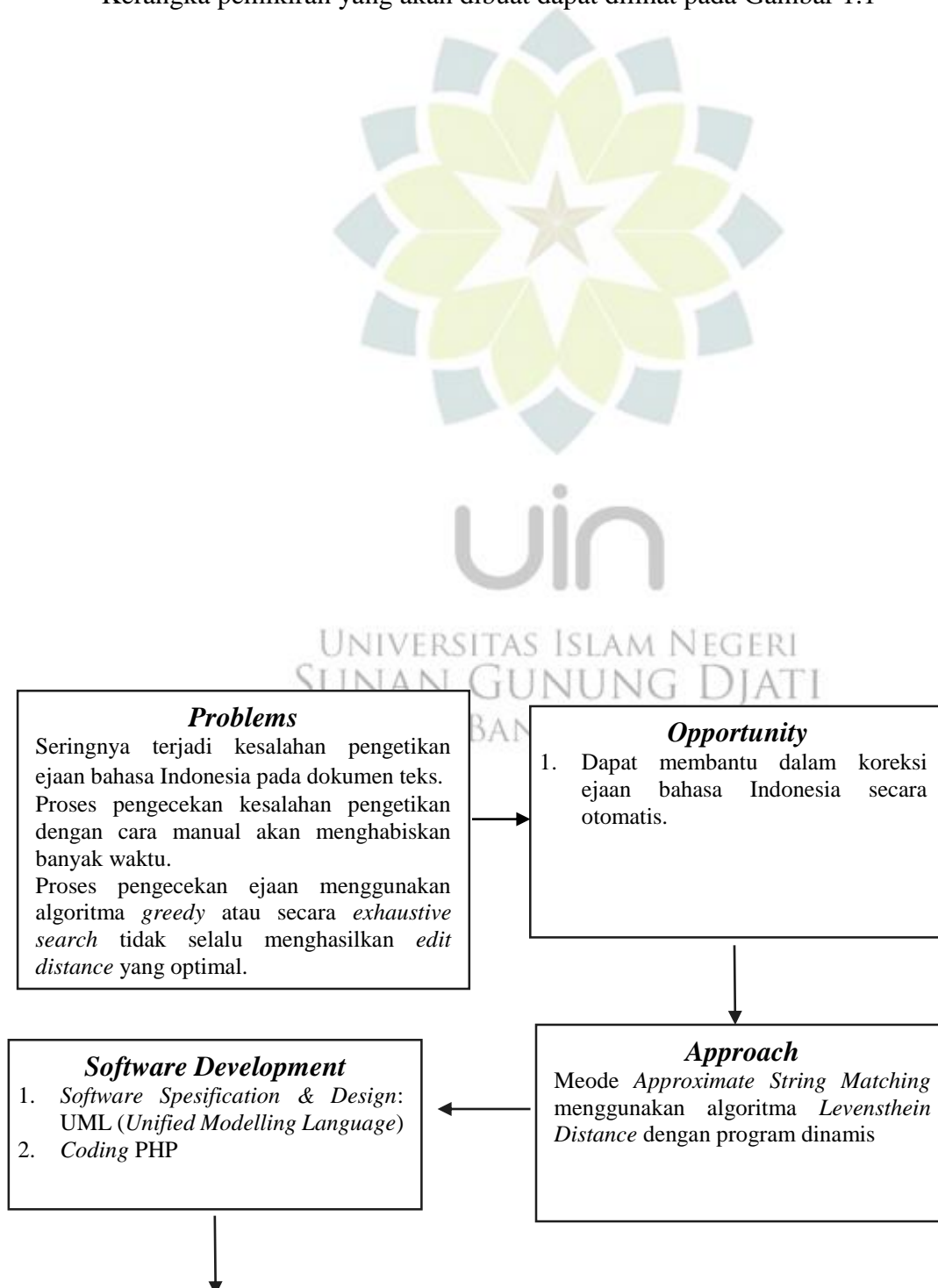
5. *Software Implementation*

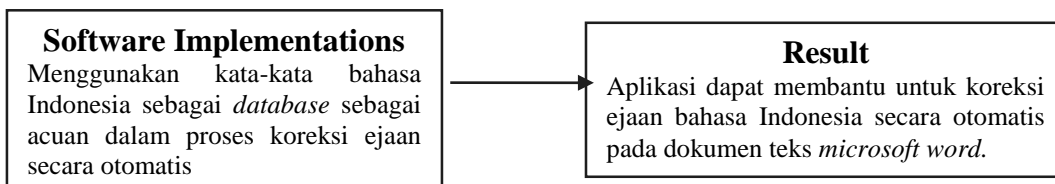
Bagaimana kerja dari program yang telah dibuat, dengan pengujian hasil dengan permasalahan yang sebelumnya diteliti.

6. *Result*

Hasil dari program yang berupa *output* dari program dan hasil dari penelitian yang menghasilkan tujuan dari penelitian.

Kerangka pemikiran yang akan dibuat dapat dilihat pada Gambar 1.1





Gambar 1. 1 Kerangka Pemikiran

1.7 Metodologi

Metodologi penelitian ini terdiri dari dua tahap yaitu tahap pengumpulan data dan tahap pengembangan perangkat lunak.

1.7.1 Teknik Pengumpulan Data

Dalam proses perancangan sistem yang akan dibangun, data yang digunakan menggunakan *database* KBBI. Sedangkan dalam proses cara kerja dan menganalisa permasalahan metode yang digunakan adalah studi literatur.

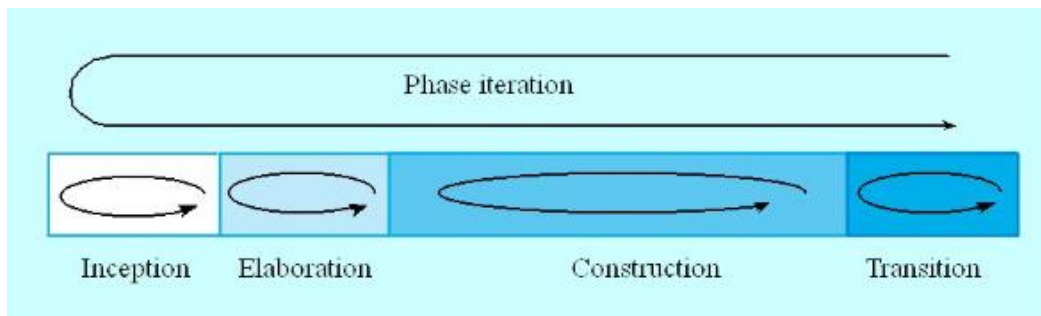
Studi literatur yaitu metode pengumpulan data dengan mempelajari literatur, paket modul, buku-buku pedoman dan segala kepustakaan lainnya yang dianggap perlu untuk lebih mempertajam konsep dan teori yang mendukung permasalahan yang dibahas.

1.7.2 Pengembangan Sistem

Model proses pengembangan perangkat lunak yang digunakan dalam pembuatan sistem ini yaitu menggunakan model RUP (*Rational Unified Process*). RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP adalah sebuah produk proses perangkat lunak yang dikembangkan oleh Rational Software yang di akuisisi oleh IBM di bulan Februari 2003[11].

RUP memiliki empat buah tahap atau fase yang dapat dilakukan pula secara iteratif[11].

Alur hidup RUP dapat dilihat pada Gambar 1.2[8].



Gambar 1. 2 Alur hidup RUP

Berikut ini penjelasan untuk setiap fase pada RUP

1. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Berikut adalah tahap yang dibutuhkan pada tahap ini:

- a. Memahami ruang lingkup dari proyek (termasuk pada biaya, waktu, kebutuhan, resiko dan lain sebagainya)
- b. Membangun kasus bisnis yang dibutuhkan

2. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

3. *Construction* (kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *initial operational capability milestone* atau batas/tonggak kemampuan operasional awal.

4. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *initial operational capability milestone* atau batas/tonggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap. Keempat fase pada RUP dijalankan secara berurutan dan iteratif dimana setiap iterasi yang digunakan untuk memperbaiki iterasi berikutnya.

1.8 Sistematika Penulisan

Sistematika penulisan tugas akhir ini disusun dengan tujuan memberikan gambaran umum tentang penelitian yang telah dilakukan. Adapun penulisan tugas akhir ini dibagi kedalam enam bab. Yang disusun berdasarkan sistematika berikut:

BAB I PENDAHULUAN

Bab ini merupakan pengantar yang memberikan gambaran mengenai permasalahan-permasalahan yang kemudian akan dibahas pada bab-bab selanjutnya. Terdapat tujuh pokok bahasan dalam bab ini, yaitu Latar Belakang Masalah, Perumusan Masalah, Tujuan Penelitian, Batasan Masalah, Metodologi Penelitian, Kerangka Pemikiran dan Sistematika Penulisan.

BAB II STUDI PUSTAKA

Bab ini membahas mengenai teori-teori yang berkaitan dengan topik yang dibuat berdasarkan hasil penelitian dan hal-hal yang berguna dalam penulisan laporan.

BAB III PERANCANGAN SISTEM

Bab ini membahas kajian aplikasi yang dibuat dari mulai melakukan analisis sistem samapi perancangan sistem itu sendiri. Perancangan sistem terdiri dari *use case diagram*, *activity diagram*, *class diagram* dan *sequence diagram*.

BAB IV IMPLEMENTASI SISTEM

Bab ini menjelaskan implementasi dari sistem yang telah dibangun baik itu *software* yang diperlukan, *hardware* yang mendukung, termasuk pengujian sistem yang telah dibangun secara umum maupun terperinci.

BAB V PENUTUP

Bab ini berisi pernyataan singkat berupa sebuah kesimpulan dari perancangan sistem yang telah dibangun dan saran untuk pembuatan sistem yang lebih baik dari sistem yang telah dibangun.