



**PENGEMBANGAN APLIKASI *TEXT MINING* UNTUK MENGUBAH *FILE*
TEKS YANG MENGGUNAKAN BAHASA *ALAY* KE DALAM BAHASA
INDONESIA YANG SESUAI EYD**

Oleh

Kadek Anggaradana

Jurusan Pendidikan Teknik Informatika, Fakultas Teknik dan Kejuruan, Universitas

Pendidikan Ganesha (Undiksha)

Email : anggara_dana@ymail.com

ABSTRAK

Bahasa *alay* merupakan bahasa pergaulan anak remaja yang berasal dari kelompok-kelompok tertentu yang sering disebut dengan anak layangan (anak kampung). Bahasa ini tidak mempunyai aturan penulisan yang pasti karena cenderung menggunakan gaya bahasa santai dan tidak memperhatikan norma kesopanan. Uniknya, bahasa pergaulan yang sebenarnya diciptakan untuk kalangan terbatas justru berkembang menjadi bahasa pergaulan yang digunakan bahasa sehari-hari. Keberadaan bahasa *alay* sudah mengubah gaya hidup generasi muda khususnya anak remaja pelajar.

Dampak dari kebiasaan menggunakan bahasa *alay* di kalangan anak remaja pelajar juga akan mempengaruhi pemilihan kata pada saat penulisan *file* laporan atau *file* dokumen lainnya yang notabene harus bersifat formal. Secara tidak sadar terkadang mereka menggunakan kata-kata *alay* atau cenderung menyingkat kata yang tidak biasa dalam laporan yang dibuatnya, yang tentunya tidak sesuai dengan kaidah penulisan bahasa Indonesia yang sesuai dengan EYD. Berdasarkan uraian tersebut, penulis bermaksud untuk mengembangkan aplikasi yang mampu mengubah bahasa *alay* ke dalam bahasa Indonesia yang sesuai dengan EYD (Ejaan Yang Disempurnakan) dengan menggunakan algoritma *text mining* dalam sebuah *file* teks. Sistem akan mengembalikan hasil pencarian kepada *user* yang berupa *list* dari kata-kata bahasa *alay* yang ditemukan dari dokumen tersebut dan kemudian sistem mengganti kata-kata tersebut secara otomatis.

Sistem ini dikembangkan dengan menggunakan *script PHP*, *AJAX* sebagai bahasa pemrograman, dan *MySQL* sebagai pengolah *database*. Dengan adanya sistem ini diharapkan dapat memberikan kemudahan kepada manusia khususnya kalangan intelektual untuk mengecek dan mengubah keberadaan bahasa *alay* dalam dokumen yang dibuatnya.

Kata kunci: Penelitian pengembangan, bahasa *alay*, EYD, *text mining*, *file* teks, dokumen.



**DEVELOPMENT TEXT MINING APPLICATION FOR CHANGE TEXT FILE
USING THE LANGUAGE *ALAY* INTO INDONESIAN
APPROPRIATE TO EYD**

By

Kadek Anggaradana

Education Informatics Engineering Department, Faculty of Engineering and Vocational,
University of Pendidikan Ganesha (Undiksha)

Email: anggara_dana@ymail.com

ABSTRACT

Language *alay* is a term for language used by group of young. This group is called *anak layangan* (village children). This language does not have a definite rule of writing because it tends to use a relaxed style and not paying attention to norms of decency. Interestingly, the language that was actually created for a limited used instead developed into lingua franca that used as everyday language. The existence of language *alay* has changed the lifestyle of the younger generation especially adolescent students.

The impact of the habit of using the language *alay* among adolescent students will also affect the choice of words at the time of writing the report file or other document file which incidentally should be formal. Unconsciously they sometimes use the words *alay* or tend to abbreviate words that are not common in the statement they had made, which is not in accordance with the rules of Indonesian writing in accordance with EYD. Based on the description, author intends to develop applications that can change the language of *alay* into the Indonesian language in accordance to EYD (*Ejaan Yang Disempurnakan*) using text mining algorithms in a text file. The system will return to user lists of words language *alay* that has been found in the document, and then the system will replace the words automatically.

System was developed using the PHP script, AJAX as a programming language and MySQL as database processing. This system is expected to provide tool for the people especially intellectuals person (for the formal uses of language) to check and change the existence of language *alay* in the document that has been made.

Keywords: research development, language *alay*, EYD, text mining, text files, documents.



I. PENDAHULUAN

Bahasa dan masyarakat akan selalu menjadi pasangan yang mengisi satu sama lain, karena adanya interaksi sosial yang menggunakan bahasa sebagai alat komunikasi, sebenarnya masih ada alat lain untuk berkomunikasi akan tetapi bahasa mungkin yang terbaik dalam berkomunikasi. Didalamnya ada penutur dan juga tindak tutur, bahasa yang bersifat universal sangat memungkinkan untuk melahirkan kata-kata atau padanan baru dalam bahasa tersebut.

Bahasa hanya bisa muncul akibat adanya interaksi sosial. Dalam interaksi sosial terjadi saling pengaruh mempengaruhi. Dalam proses interaksi, orang yang lebih aktif melakukan komunikasi akan mendominasi interaksi tersebut. Maka tidak heran apabila suatu bahasa lebih banyak dipakai, maka bahasa itu akan berkembang.

Saat ini kita sudah sangat sering dan sangat familiar sekali dengan yang namanya komunitas anak layangan atau yang lebih dikenal dengan nama *alay*. Alay itu sendiri adalah singkatan dari Anak layangan, Alah lebay, Anak Layu, atau Anak kelayapan yang menghubungkannya dengan anak jarpul (Jarang Pulang).

Bahasa *alay* dapat diartikan sebagai variasi bahasa yang bersifat sementara yang biasanya berupa singkatan menggabungkan huruf dengan angka, memperpanjang atau memperpendek dan mencampurkan huruf besar dan kecil membentuk sebuah kata maupun kalimat. Uniknya, bahasa pergaulan yang sebenarnya diciptakan untuk kalangan terbatas justru berkembang menjadi bahasa pergaulan yang digunakan bahasa sehari-hari. Keberadaan bahasa *alay* sudah mengubah gaya hidup generasi muda khususnya anak remaja pelajar.

Dampak dari kebiasaan menggunakan bahasa *alay* dikalangan anak remaja pelajar juga akan mempengaruhi pemilihan kata pada saat penulisan *file* laporan atau *file* dokumen lainnya yang notabene harus bersifat formal. Secara tidak sadar terkadang mereka menggunakan kata-kata *alay* atau cenderung menyingkat kata yang tidak biasa dalam laporan yang dibuatnya, yang tentunya tidak sesuai dengan kaidah penulisan bahasa Indonesia yang sesuai dengan EYD.

Berdasarkan uraian tersebut, penulis bermaksud untuk mengembangkan aplikasi yang mampu mengubah bahasa *alay* ke dalam bahasa Indonesia yang sesuai dengan



EYD (Ejaan Yang Disempurnakan) dengan menggunakan algoritma *text mining* dalam sebuah *file* teks. Sistem akan mengembalikan hasil pencarian kepada *user* yang berupa *list* dari kata-kata bahasa *alay* yang ditemukan dari dokumen tersebut dan kemudian sistem mengganti kata-kata tersebut secara otomatis. Sistem ini diharapkan dapat memberikan kemudahan kepada manusia khususnya kalangan intelektual untuk mengecek dan mengubah keberadaan bahasa *alay* dalam dokumen yang dibuatnya.

II. METODE PENELITIAN

2.1 Text Mining

Text mining merupakan kegiatan menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Andre Kurniawan dan Hapnes Toba (2009) menyatakan bahwa, yang membedakan data *mining* dengan *text mining* adalah proses analisis terhadap suatu datanya. *Data Mining* atau KDD adalah proses untuk menemukan pengetahuan dari sejumlah besar data yang disimpan baik di dalam *databases*, *data warehouses* atau tempat penyimpanan informasi lainnya. Sedangkan untuk *text mining* sering disebut dengan *Keyword-Based Association Analysis*. *Keyword-Based Association Analysis* merupakan sebuah analisa yang mengumpulkan *keywords* atau *terms* (istilah) yang sering muncul secara bersamaan dan kemudian menemukan hubungan asosiasi dan korelasi diantara *keywords* atau *terms* itu.

Secara garis besar dalam melakukan implementasi *text mining* terdiri dari dua tahap besar yaitu *pre-processing* dan *processing*.

a. Pre-processing

Tahap *preprocessing* adalah tahap dimana aplikasi melakukan seleksi data yang akan diproses pada setiap dokumen. Setiap kata akan dipecah-pecah menjadi struktur bagian kecil yang nantinya akan mempunyai makna sempit. Ada beberapa hal yang perlu dilakukan pada tahap *pre-processing* ini, yaitu:



- 1) *Tokenizing*
- 2) *Filtering*
- 3) *Stemming*
- 4) *Tagging*

Tujuan dilakukan *pre-processing* adalah memilih setiap kata dari dokumen dan merubahnya menjadi kata dasar yang memiliki arti sempit.

1) *Tokenizing*

Tokenizing atau *parsing* adalah sebuah proses yang dilakukan seseorang untuk menjadikan sebuah kalimat menjadi lebih bermakna atau berarti dengan cara memecah kalimat tersebut menjadi kata-kata atau *frase-frase*.

Tokenizing dalam pembuatan aplikasi *text mining* ini merupakan proses penguraian *file* teks yang semula berupa kalimat-kalimat berisi kata-kata dan tanda pemisah antar kata seperti titik (.), koma (,), spasi dan tanda pemisah lain menjadi kata-kata saja baik itu berupa kata penting maupun kata tidak penting.

2) *Filtering*

Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil *token*. Dimana dalam tahap ini bisa digunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). Dalam tahap ini penulis menggunakan algoritma *stoplist*. *Stoplist / stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang. Contoh *stopwords* adalah “yang”, “dan”, “di”, “dari” dan seterusnya.

3) *Stemming*

Tahap *stemming* adalah tahap mencari *root* kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Atau dengan kata lain pada tahap ini dilakukan proses penghilangan *prefiks* dan *sufiks* dari kueri dan istilah-istilah yang dihasilkan dari proses *filtering*. *Stemming* dilakukan atas dasar asumsi bahwa kata-kata yang memiliki *stem* yang sama dianggap memiliki makna yang serupa sehingga pengguna tidak keberatan untuk memperoleh dokumen-dokumen yang di dalamnya terdapat kata-kata dengan *stem* yang sama dengan kuerinya.



4) *Tagging*

Tahap *tagging* adalah tahap mencari bentuk awal/root dari tiap kata lampau atau kata hasil *stemming*. Tahap ini biasanya dilakukan untuk proses *text mining* berbahasa Inggris. Jadi dalam penelitian ini, tahap *tagging* akan di abaikan, karena *text mining* penelitian ini hanya untuk teks yang menggunakan bahasa Indonesia.

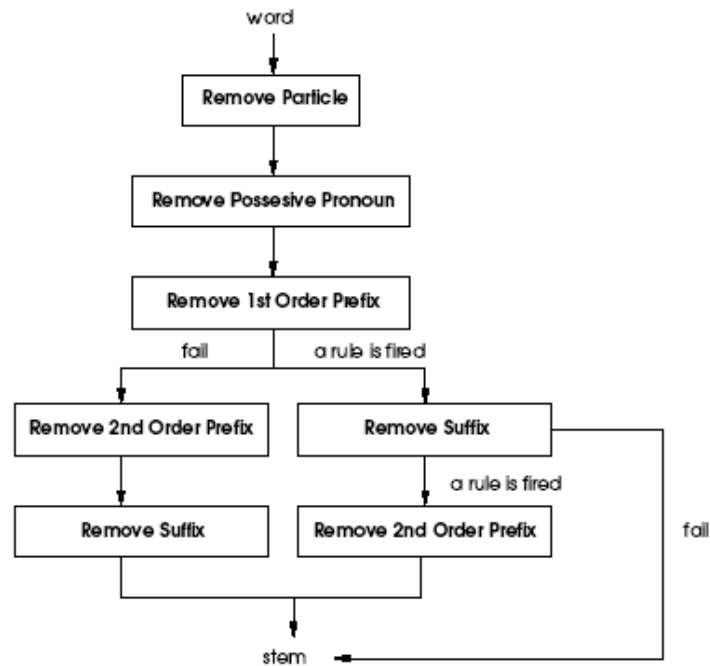
b. *Processing*

Tahap ini merupakan tahap penentuan seberapa jauh keterhubungan antar kata-kata antar dokumen yang ada. Peneliti biasanya menggunakan metode-metode tertentu untuk menentukan keterhubungan antar dokumen.

Secara umum terdapat dua jenis metode pada tahap ini yaitu metode yang tidak melakukan perhitungan bobot kalimat dan yang melakukan perhitungan bobot kalimat. Metode yang tidak menghitung bobot kalimat hanya mengambil beberapa kata penting untuk kemudian diproses sesuai permasalahan yang diteliti. Metode-metode yang menghitung bobot kalimat menggunakan bobot *term* (kata maupun pasangan kata) dari setiap *term* yang terdapat dalam kalimat tersebut. Ada banyak algoritma yang bisa digunakan untuk menghitung bobot kalimat.

2.2 Metode Porter Stemmer

Porter Stemmer for Bahasa Indonesia dikembangkan oleh Fadillah Z. Tala pada tahun 2003. Implementasi *Porter Stemmer for Bahasa Indonesia* berdasarkan English *Porter Stemmer* yang dikembangkan oleh W.B. Frakes pada tahun 1992. Karena bahasa Inggris datang dari kelas yang berbeda, beberapa modifikasi telah dilakukan untuk membuat Algoritma *Porter* dapat digunakan sesuai dengan bahasa Indonesia. Metode ini akan digunakan pada tahap *stemming*, yaitu mengubah kata hasil *filtering* yang semula masih berimbuhan (*word*) menjadi kata dasar (*stem*). Desain dari *Porter Stemmer untuk Bahasa Indonesia* dapat dilihat pada Gambar 1.



Gambar 1. Ilustrasi Proses *Stemming* Menggunakan Metode Porter

Berikut ini contoh ilustrasi dari metode tersebut.

a. *Remove Partikel*

Pada tahap ini sistem akan menghapus partikel yang terdapat pada sebuah kata.

Contoh ilustrasinya dapat dilihat pada Tabel 1.

Tabel 1 Daftar Partikel

| Suffix | Replacement | Additional Condition | Examples |
|--------|-------------|----------------------|----------------|
| kah | NULL | NULL | bukukan → buku |
| lah | NULL | NULL | adalah → ada |
| pun | NULL | NULL | bukupun → buku |

b. *Remove Possesive Pronoun*

Pada tahap ini sistem akan menghapus kata ganti milik yang terdapat pada sebuah kata. Contoh ilustrasinya dapat dilihat pada Tabel 2.

Tabel 2 Daftar *Possesive Pronoun*

| Suffix | Replacement | Additional Condition | Examples |
|--------|-------------|----------------------|----------------|
| ku | NULL | NULL | bukuku → buku |
| mu | NULL | NULL | bukumu → buku |
| nya | NULL | NULL | bukunya → buku |

c. *Remove first order of derivational prefixes*

Pada tahap ini sistem akan menghapus imbuhan pertama yang terdapat pada sebuah kata. Contoh ilustrasinya dapat dilihat pada Tabel 3.

Tabel 3 Daftar *first order of derivational prefixes*

| Prefix | Replacement | Additional Condition | Examples |
|--------|-------------|----------------------|-----------------|
| meng | NULL | NULL | mengukur → ukur |
| meny | s | V...* | menyapu → sapu |
| men | NULL | NULL | menduga → duga |
| mem | p | V...* | memilah → pilah |
| mem | NULL | NULL | membaca → baca |

| Prefix | Replacement | Additional Condition | Examples |
|--------|-------------|----------------------|------------------|
| me | NULL | NULL | merusak → rusak |
| peng | NULL | NULL | pengukur → ukur |
| peny | s | V...* | penyapu → sapu |
| pen | NULL | NULL | penduga → duga |
| pem | p | V...* | pemilah → pilah |
| pem | NULL | NULL | pembaca → baca |
| di | NULL | NULL | diukur → ukur |
| ter | NULL | NULL | terdalam → dalam |
| ke | NULL | NULL | kekasih → kasih |

d. *Remove second order of derivational prefixes*

Pada tahap ini sistem akan menghapus imbuhan kedua yang terdapat pada sebuah kata. Contoh ilustrasinya dapat dilihat pada Tabel 4.

Tabel 4 Daftar *second order of derivational prefixes*

| Prefix | Replacement | Additional Condition | Examples |
|--------|-------------|----------------------|------------------|
| ber | NULL | NULL | berlari → lari |
| bel | NULL | NULL | belajar → ajar |
| be | NULL | NULL | bekerja → kerja |
| per | NULL | NULL | perjelas → jelas |

| Prefix | Replacement | Additional Condition | Examples |
|--------|-------------|----------------------|-----------------|
| pel | NULL | NULL | pelajar → ajar |
| pe | NULL | NULL | pekerja → kerja |

e. *Remove derivational suffixes*

Pada tahap ini sistem akan menghapus akhiran yang terdapat pada sebuah kata.

Contoh ilustrasinya dapat dilihat pada Tabel 5.

Tabel 5 Daftar *derivational suffixes*

| Suffix | Replacement | Additional Condition | Examples |
|--------|-------------|----------------------|------------------|
| kan | NULL | NULL | ambilkan → ambil |
| an | NULL | NULL | makanan → makan |
| i | NULL | NULL | tandai → tanda |

Jadi jika diberikan contoh kata “memperbaikinya”, maka yang pertama kali dikerjakan oleh sistem adalah menghapus kata ganti milik “nya”, sehingga katanya menjadi “memperbaiki”, kemudian menghapus imbuhan “mem”, kemudian menghapus *suffix* (akhiran) “i” dan yang terakhir adalah menghapus imbuhan kedua “per” sehingga akhir katanya menjadi “baik” yang merupakan *rule* dari kata “memperbaikinya”. *Rule-rule* inilah yang nantinya akan dijadikan dasar untuk mengecek apakah suatu kata, termasuk kata alay atau bukan.



2.3 Teknik Konversi *File* Teks

Untuk melakukan konversi teks, user hanya diminta untuk meng-upload *file* teks yang akan dikonversi. Dimana *file* yang di-upload harus bertipe text *file* (*.txt), kemudian sistem akan melakukan proses mining terhadap teks yang sudah di-upload tadi. Jika selama proses mining berjalan, sistem menemukan kata/bahasa *alay* didalam teks tersebut maka sistem akan mengubah kata tersebut ke dalam bahasa Indonesia yang sesuai dengan EYD.

III. ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah dan Usulan Solusi

Dampak yang diakibatkan dari maraknya penggunaan bahasa *alay* sebagai bahasa pergaulan dikalangan anak remaja pelajar yaitu ketidaktepatanya untuk menentukan pemilihan kata pada saat penulisan *file* laporan atau *file* dokumen lainnya yang notabene harus bersifat formal. Secara tidak sadar terkadang mereka menggunakan kata-kata *alay* atau cenderung menyingkat kata yang tidak biasa dalam laporan yang dibuatnya, yang tentunya tidak sesuai dengan kaidah penulisan bahasa Indonesia yang baik dan benar.

Langkah awal yang bisa dilakukan untuk mengetahui apakah dalam suatu *file* dokumen mengandung bahasa *alay* atau tidak adalah dengan cara mengecek isi *file* dokumen dari awal sampai akhir dan kemudian mengganti kata *alay* tersebut dengan kata baku yang sesuai dengan bahasa Indonesia yang baik dan benar. Jika hal tersebut dilakukan secara manual bukanlah merupakan suatu pekerjaan yang rumit, asalkan kita sudah mengetahui karakteristik atau bentuk-bentuk dari bahasa *alay*. Namun yang menjadi permasalahan adalah apabila isi dan jumlah dari *file* dokumen dalam satuan besar, maka kita akan menghabiskan banyak waktu hanya untuk mengecek apakah di dalam dokumen masih terdapat bahasa *alay* atau tidak.

Berdasarkan uraian tersebut, penulis bermaksud untuk mengembangkan aplikasi yang mampu mengubah bahasa *alay* ke dalam bahasa Indonesia yang sesuai dengan EYD (Ejaan Yang Disempurnakan) dengan menggunakan algoritma *text mining* dalam sebuah *file* teks. Sistem ini nantinya diharapkan mampu membantu seorang penulis untuk mengecek apakah dalam *file* dokumen yang dibuatnya masih mengandung bahasa



alay atau tidak. Sistem akan mengembalikan hasil pencarian kepada *user* yang berupa *list* dari kata-kata bahasa *alay* yang ditemukan dari dokumen tersebut dan kemudian sistem mengganti kata-kata tersebut secara otomatis.

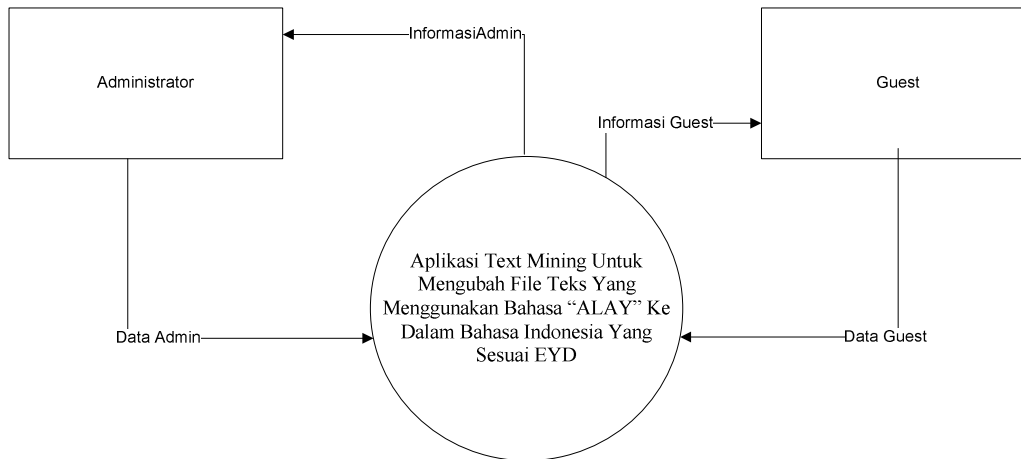
3.2 Analisis Perangkat Lunak

Berdasarkan analisis terhadap pengembangan aplikasi *text mining* untuk mengubah *file* teks yang menggunakan bahasa *alay* ke dalam bahasa Indonesia yang sesuai EYD, terdapat beberapa proses yang dapat diimplementasikan, adalah sebagai berikut.

1. Membaca dan menyimpan *file* teks
2. Menguraikan *file* teks yang semula berupa kalimat-kalimat berisi kata-kata dan tanda pemisah antar kata seperti titik (.), koma (,), spasi dan tanda pemisah lain menjadi kata-kata saja baik itu berupa kata penting maupun kata tidak penting, tahap ini disebut dengan *tokenizing*.
3. Membandingkan kata-kata hasil *tokenizing* dengan daftar kata baku, baik untuk kata penting maupun kata tidak penting, tahap ini disebut dengan *filtering*
4. Mengembalikan berbagai bentukan kata ke dalam suatu representasi yang sama. Atau dengan kata lain pada tahap ini dilakukan proses penghilangan *prefiks* dan *sufiks* dari kueri dan istilah-istilah yang dihasilkan dari proses *filtering*
5. Membandingkan kata-kata hasil *stemming* dengan daftar kata baku dan dengan daftar kata *alay*.
6. Mengubah kata-kata yang dicurigai mengandung makna *alay*.
7. Mengembalikan daftar kata-kata dari dokumen yang belum terdefinisi.

3.3 Perancangan Perangkat Lunak

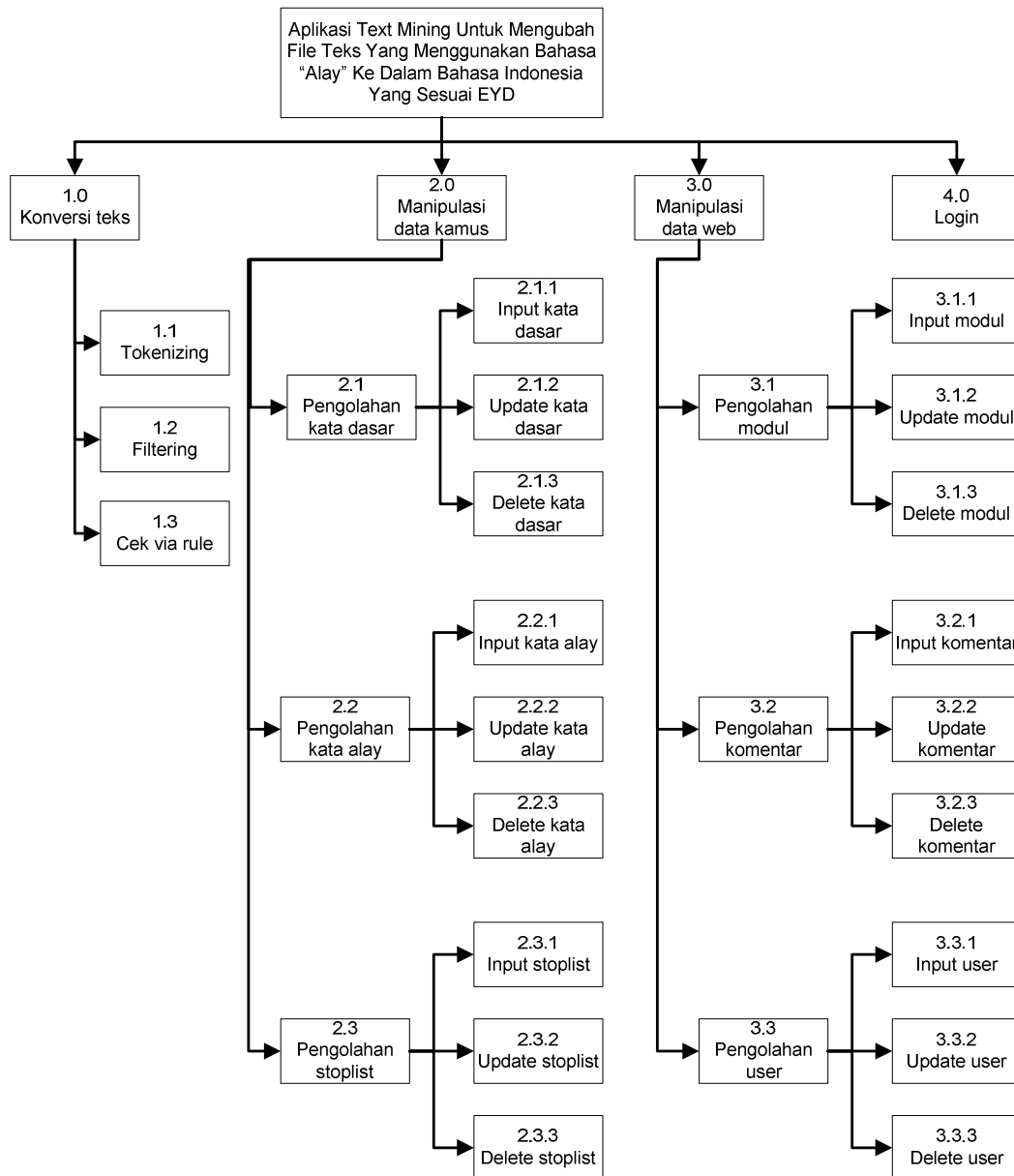
Perancangan perangkat lunak pengembangan aplikasi *text mining* untuk mengubah *file* teks yang menggunakan bahasa *alay* ke dalam bahasa Indonesia yang sesuai EYD adalah masukan data disimpan ke *database* dan selanjutnya di proses dengan memanfaatkan algoritma *text mining*. Berikut adalah gambaran mengenai diagram *konteks* dan arsitektur perangkat lunak yang akan dibangun.



Gambar 1 Diagram Konteks

Tabel 1 Kamus Data *Diagram Konteks*

| No | Data | Keterangan |
|----|------------------------|--|
| 1 | <i>Data Admin</i> | <ul style="list-style-type: none"> - Data <i>users</i> - Data kata dasar - Data kata <i>alay</i> - Data <i>stoplist</i> - Data modul - Data komentar - Data teks |
| 2 | <i>Informasi Admin</i> | <ul style="list-style-type: none"> - Informasi <i>users</i> - Informasi kata dasar - Informasi kata <i>alay</i> - Informasi musik - Informasi <i>stoplist</i> - Informasi modul - Informasi komentar - Informasi hasil perbandingan kata - Informasi hasil perubahan teks |
| 3 | <i>Data Guest</i> | <ul style="list-style-type: none"> - Data teks - Data komentar |
| 4 | <i>Informasi Guest</i> | <ul style="list-style-type: none"> - Informasi hasil perbandingan kata - Informasi hasil perubahan teks - Informasi komentar |

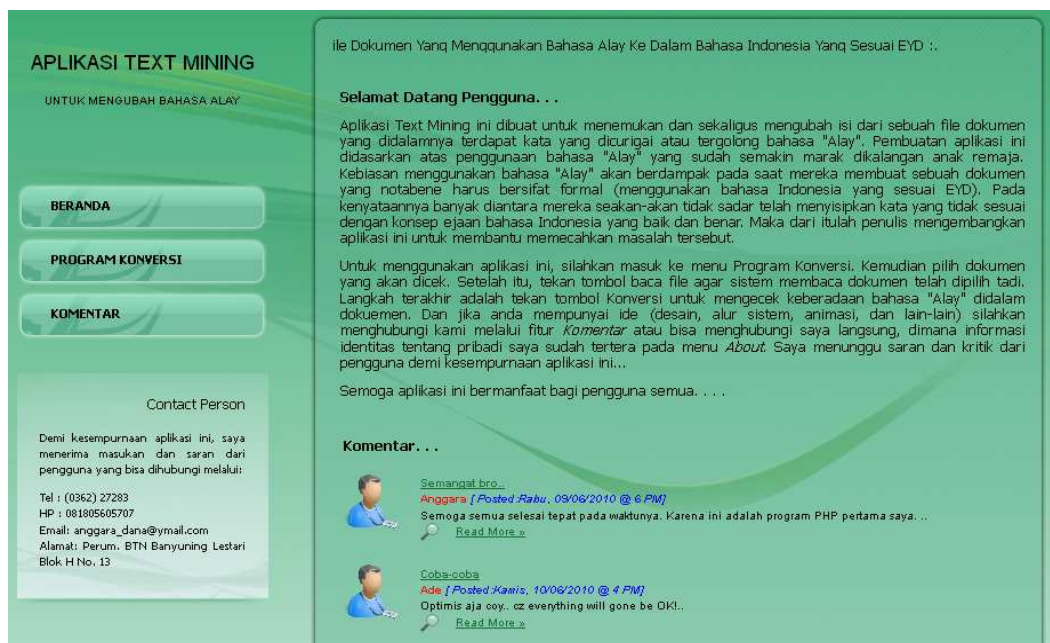


Gambar 2 Rancangan Modul Sistem

IV. IMPLEMENTASI DAN PENGUJIAN

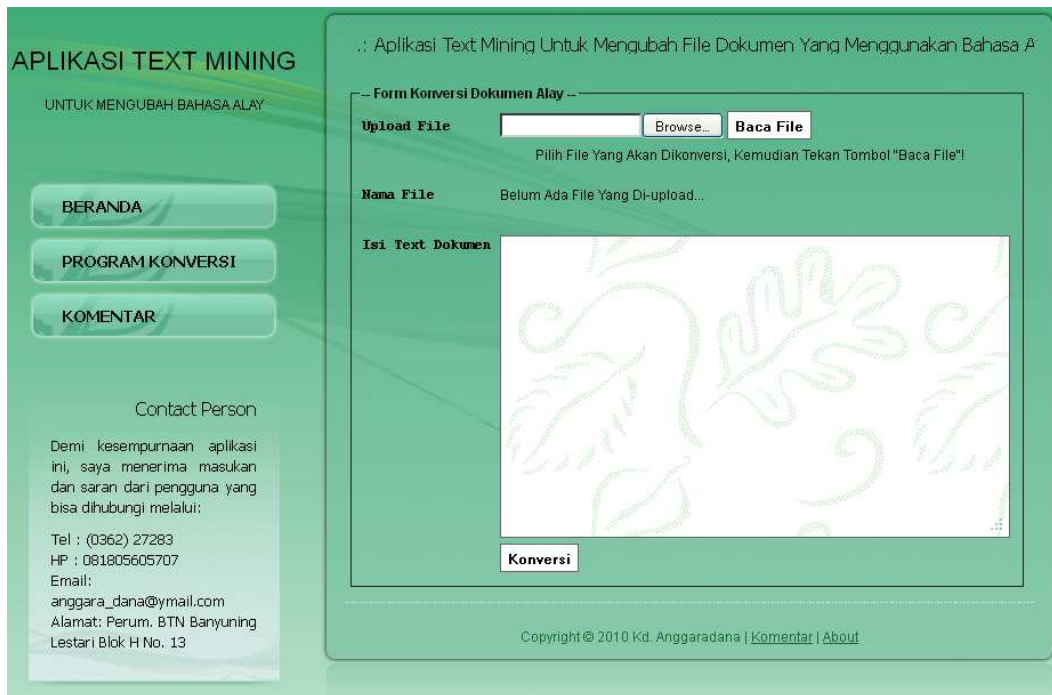
4.1 Implementasi Perangkat Lunak

Data Flow Diagram (DFD) dan Rancangan Arsitektur Perangkat Lunak diimplementasikan dengan menggunakan *PHP* dan *AJAX* sebagai bahasa pemrograman. Berikut ini tampilan Menu Utama Aplikasi *Desktop*.



Gambar 3. Implemnatasi Halaman Utama Pengguna

Halaman ini merupakan halaman awal pengguna sebelum melakukan aktifitas yang dia inginkan.



Gambar 4 Implementasi Halaman Konversi Teks

Halaman konversi teks merupakan aplikasi yang disediakan untuk melakukan konversi teks. Untuk melakukan konversi, *user* diminta untuk meng-*upload file* yang akan di konversi, dan kemudian sistem akan membaca *file* tersebut dan siap untuk dikonversi.

4.2 Pengujian Perangkat Lunak

Secara umum hasil pengujian menunjukkan sistem sudah bisa menangani data masukan yang tidak valid dan menampilkan output sesuai dengan apa yang direncanakan. Hasil pengujian konseptual menunjukkan bahwa sistem telah melaksanakan mekanisme perhitungan dan mekanisme logika sesuai dengan apa yang direncanakan. Berikut ini merupakan hasil dari pengujian fungsional sistem.

Tabel 2 Pelaksanaan Uji Kasus

| Kasus | Uraian | Penanganan | Pelaksanaan |
|-------|----------------------------|--|-------------|
| 1.1 | Teks yang tidak mengandung | Seluruh isi dokumen dapat di-mining dengan baik oleh sistem. Tidak ada pesan kesalahan | √ |

| Kasus | Uraian | Penanganan | Pelaksanaan |
|-------|---|---|-------------|
| | kata-kata <i>alay</i> | yang disampaikan oleh sistem. | |
| 1.2 | Teks yang mengandung kata-kata asing | Sistem akan memberi warna jika terdapat kata asing dalam dokumen teks. Hal ini terjadi karena sistem tidak punya pengetahuan terhadap bahasa asing (contohnya: bahasa Inggris). | √ |
| 1.3 | Teks yang mengandung kata <i>alay</i> | Sistem berjalan dengan baik, karena teks <i>alay</i> yang diuji sudah ada dalam kamus bahasa <i>alay</i> | √ |
| 1.4 | Teks yang mengandung kata <i>alay</i> yang cenderung menyingkat atau mengurangi jumlah karakter katanya | Khusus untuk kata <i>alay</i> yang cenderung menyingkat atau mengurangi jumlah karakter katanya, sistem tidak mampu mengkonversinya dengan baik, terkecuali kata tersebut sudah tersimpan dalam kamus bahasa <i>alay</i> . Kelemahan ini disebabkan karena format singkatan katanya benar-benar tidak beraturan. <i>Feedback</i> -nya adalah kata tersebut akan dikembalikan lagi kepada <i>user</i> untuk <i>edit</i> secara manual. | √ |
| 1.5 | Teks <i>alay</i> yang menggunakan penambahan huruf yang berlebihan, dikombinasikan dengan kata-katanya yang disingkat | Sistem mampu mengurangi jumlah karakter yang berlebihan tersebut, tetapi jika kata-katanya mengalami penyingkatan, maka kemungkinan sistem tidak mampu mengkonversinya dengan baik. Sehingga sistem akan memberikan <i>feedback</i> kepada <i>user</i> untuk diubah secara manual. | √ |
| 1.6 | Teks <i>alay</i> mengganti karakter huruf dengan karakter angka, | Sistem mampu mengubah karakter angka tersebut menjadi karakter huruf. Dengan <i>formula</i> ('0'='o', '1'='i', '3'='e', '4'='a', '5'='s', '6'='g', '7'='t', '8'='b', dan '9'='g'). <i>Formula</i> ini dirumuskan berdasarkan analisis | √ |

| Kasus | Uraian | Penanganan | Pelaksanaan |
|-------|--|--|-------------|
| | dikombinasikan dengan kata-katanya yang disingkat | terhadap teks <i>alay</i> yang berkembang di lapangan. Tetapi jika kata-katanya mengalami penyingkatan, maka kemungkinan sistem tidak mampu mengkonversinya dengan baik. Sehingga sistem akan memberikan <i>feedback</i> kepada <i>user</i> untuk diubah secara manual. | |
| 1.7 | <p>Teks <i>alay</i> yang menggunakan rule-rule tertentu yang penggunaannya konsisten, dikombinasikan dengan kata-katanya yang disingkat.</p> <p>Rule yang dimaksud contohnya:</p> <ul style="list-style-type: none"> - kaw<u>and</u> = kawan - cy<u>ank</u> = cyang - cl<u>id</u> = cl<u>i</u>t - cl<u>i</u>t = sl<u>i</u>t - c<u>y</u>ang = s<u>y</u>ang - x<u>m</u>s = s<u>m</u>s - dan lain-lain | Sistem mampu mengubah rule-rule tersebut dengan baik, asalkan rule tersebut sudah tersimpan dalam database. <i>Formula</i> ini dirumuskan berdasarkan analisis terhadap teks <i>alay</i> yang berkembang di lapangan. Tetapi jika kata-katanya mengalami penyingkatan, maka kemungkinan sistem tidak mampu mengkonversinya dengan baik. Sehingga sistem akan memberikan <i>feedback</i> kepada <i>user</i> untuk diubah secara manual. | √ |

Tabel 3 *Sample Uji Kasus White Box Testing*

| Kasus | <i>File Teks</i> | Keterangan |
|-------|--|--|
| 1.1 | Sekarang penerapan bahasa Alay sudah diterapkan di situs jejaring sosial tersebut, yang lebih parahnya lagi sudah bukan menyingkat kata lagi, namun sudah merubah kosakatanya bahkan cara penulisannya pun bisa membuat sakit mata orang yang membaca karena menggunakan huruf besar kecil yang diacak ditambah dengan angka dan karakter tanda baca. Bahkan arti kosakatanya pun menceng jauh dari yang dimaksud. | Teks yang tidak mengandung kata-kata <i>alay</i> |
| 1.2 | Dewasa ini, bidang pendidikan mengalami kemajuan yang cukup pesat. Institusi pendidikan berskala besar seperti Univesitas Pendidikan Ganesha dengan jumlah jurusan yang lebih dari satu dan jumlah mahasiswa yang besar, telah banyak menghasilkan lulusan dari berbagai jurusan. Tentunya laporan tugas akhir sebagai syarat kelulusan mahasiswa disimpan untuk digunakan sebagai referensi bagi pihak-pihak yang membutuhkan seperti misalnya mahasiswa-mahasiwa yang sedang mengerjakan tugas akhir. | Teks yang mengandung kata-kata asing |
| 1.3 | <u>km cantiek dech</u> hari <u>nie...</u> <u>ciyus miapa??</u> | Teks yang mengandung kata alay |
| 1.4 | km cantiek dech hari nie... ciyus miapa?? <u>diam hdupq</u> cuma ada km... sumpah <u>dmi than</u> , q gk <u>bhong...</u> | Teks yang mengandung kata alay yang cenderung menyingkat atau mengurangi jumlah karakter katanya |

| Kasus | File Teks | Keterangan |
|-------|---|---|
| 1.5 | <p>km cantiek dech hari nie... ciyus <u>miapaaaaaaaa??</u> dlam hdupq cuma ada km... sumpah dmi than, q gkk <u>bhoooooooooooooooooong...</u> q <u>syankkkkkkk</u> <u>bangettttttttt</u> ama km....</p> | <p>Teks <i>alay</i> yang menggunakan penambahan huruf yang berlebihan, dikombinasikan dengan kata-katanya yang disingkat</p> |
| 1.6 | <p>km cantiek <u>d3ch h4r1 n1e</u>... ciyus <u>m14p4??</u> <u>d4l4m hidupq</u> cuma <u>4d4</u> km... <u>sump4h dm1 tuh4n</u>, q gk <u>b0h0ng...</u></p> | <p>Teks <i>alay</i> mengganti karakter huruf dengan karakter angka, dikombinasikan dengan kata-katanya yang disingkat</p> |
| 1.7 | <p>km cantiek <u>dech</u> hari nie... ciyus miapa?? dlam hdupq cuma ada km... sumpah dmi than, q gk bhong... q <u>cayank</u> <u>bangets</u> ama km....</p> | <p>Teks <i>alay</i> yang menggunakan rule-rule tertentu yang penggunaanya konsisten, dikombinasikan dengan kata-katanya yang disingkat.</p> |



V. PENUTUP

5.1 Simpulan

Simpulan yang dapat diambil dari hasil pengujian dan analisis terhadap pengembangan aplikasi *text mining* untuk mengubah *file* teks yang menggunakan bahasa *alay* ke dalam bahasa Indonesia yang sesuai EYD yaitu sebagai berikut.

1. Sistem mampu melakukan pengelolaan data yang diperlukan seperti data kata dasar, data kata *alay*, data *stoplist*, data modul, data komentar dan data *user*.
2. Sistem mampu membuat kamus kata dasar bahasa Indonesia, kamus kata *alay* dan kamus *stoplist*.
3. Sistem mampu mengecek keberadaan bahasa lain dalam *file* teks selain bahasa Indonesia yang baik dan benar. Maka dari itu sistem ini juga bisa dikategorikan ke dalam sistem temu kembali teks/string.

Jika di dalam *file* teks yang sedang diujikan mengandung kata *alay*, maka secara otomatis sistem akan mengubah kata *alay* tersebut menjadi kata yang sesuai dengan kaedah bahasa Indonesia yang sesuai dengan EYD.

5.2 Saran

Aplikasi ini belum bisa menangani proses *mining* terhadap dokumen yang menggunakan bahasa selain bahasa Indonesia. Sehingga khusus untuk dokumen yang menggunakan bahasa selain bahasa Indonesia, tidak bisa diproses dan ditangani oleh sistem ini.

Dari kelemahan di atas, diharapkan kepada pembaca yang tertarik untuk melakukan perbaikan dan penyempurnaan dari pengembangan aplikasi *text mining* untuk mengubah *file* teks yang menggunakan bahasa *alay* ke dalam bahasa Indonesia yang sesuai EYD dapat melengkapi kekurangan yang ada pada aplikasi ini, sehingga dapat menghasilkan aplikasi yang lebih sempurna dari sebelumnya.



ISSN 2252-9063

*Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika
(KARMAPATI)*

Volume 2, Nomor 2, Pebruari 2013

VI. DAFTAR PUSTAKA

Aliassyah, Midun. 2011. Paradigma Pergeseran Kaidah Bahasa Indonesia Ke Dalam Karakter Bahasa Alay: Tinjauan Analisis Kontrastif Dan Ilmu Semantik. http://maliassyah.blogspot.com/2011_04_08_archive.html. (diakses tanggal 5 Juni 2012).

Kurniawan, Andre dan Hapnes Toba. "Pembuatan Aplikasi Bergerak temu Ulang *File Elektronik* Berbahasa Indonesia Dengan Memanfaatkan Java CLDC". Makalah disajikan dalam *Seminar Aplikasi Teknologi Informasi 2009 (SNATI 2009)*. Universitas Kristen Maranatha. Yogyakarta 20 Juni 2009.

Proboyekti, Umi. 2008. *Software Process Model I*. <http://lecturer.ukdw.ac.id/othie/softwareprocess.pdf>. (diakses tanggal 14 Februari 2012).