

Aplikasi Automated Text Summarization (ATS) Pembuat Lead (Teras) Berita dengan Text to Speech (TTS) Menggunakan Algoritma TF-IDF dan Vector Space Model

Ni Komang Mika Karmila¹, Made Windu Antara Kesiman²,
I Gede Mahendra Darmawiguna³
Jurusan Pendidikan Teknik Informatika
Universitas Pendidikan Ganesha
Singaraja, Bali

E-mail: yibi.cantique@yahoo.com¹, dekdnu@yahoo.com², igd.mahendra.d@gmail.com³

Abstrak—Saat membaca surat kabar, masyarakat cenderung hanya akan memilih beberapa berita dari sekian banyak berita yang termuat dalam surat kabar untuk kemudian dibaca. Hal ini dilakukan karena faktor keterbatasan waktu yang dimiliki masyarakat setiap harinya. Hanya dengan membaca *lead* (teras) berita, masyarakat cukup mampu menilai bagaimana isi berita secara keseluruhan. Fenomena ini menjadi tantangan tersendiri bagi para jurnalis (penulis berita) untuk dapat menghasilkan *lead* yang mampu menarik pembaca sebanyak-banyaknya. Akan tetapi, membuat sebuah *lead* bukanlah perkara mudah. Penelitian ini bertujuan (1) merancang aplikasi pembuat *lead*; (2) mengimplementasikan rancangan aplikasi pembuat *lead*. Metode penelitian yang digunakan adalah pengembangan (*development*). Aplikasi ini memiliki dua proses utama yaitu proses peringkasan dengan menerapkan algoritma TF-IDF dan *Vector Space Model* serta proses pembacaan ringkasan dengan memanfaatkan MBROLA dan *diphone database id1* yang dibangkitkan oleh eSpeak. Hasil penelitian ini adalah aplikasi pembuat *lead* yang dapat membuat dan membacakan ringkasan dari suatu artikel berita berbahasa Indonesia dalam format .docx dan .txt. Hasil uji coba menunjukkan bahwa aplikasi pembuat *lead* ini telah dapat melakukan fungsinya dengan baik dari segi fungsional dan konseptual/ struktural. Aplikasi yang diimplementasikan dengan menggunakan bahasa pemrograman Java ini diharapkan dapat membantu meringankan tugas para jurnalis dalam membuat *lead* berita terutama dari segi efisiensi waktu pengerjaan.

Kata kunci— *berita, lead, peringkasan, pembacaan.*

Abstract— When reading the newspaper, people only tend to choose some news to be read. This is done because of the limited time which is had by the society. By reading the leads (the terrace) of the news, people can judge the whole content of the news. This phenomenon becomes a challenge for the journalists (news writers) to write an interesting leads that can attract the readers as much as possible. However, making a lead is not an easy thing. This study aimed to (1) design a lead maker application and (2) apply the lead maker application design. The method which was used in this research was development. The lead maker application has two main processes, namely the process of summarization by applying TF-IDF and Vector Space Model algorithm as well as the process of reading the summarize by utilizing MBROLA and the *diphone database id1* which is generated by eSpeak. The product of this research was the lead maker application that can create and read a summary of a news article in Indonesian language, in .docx and .txt format. The testing result of the lead maker application showed that it run its function well in terms of functional and conceptual/ structural aspects. This application is implemented by using the Java programming language which is expected to ease the task of the journalists to make the news leads, especially in terms of processing time's efficiency.

Keywords— *news, lead, summarization, reading.*



I. PENDAHULUAN

Berita adalah cerita atau keterangan mengenai kejadian atau peristiwa yang hangat [1]. Dengan berita masyarakat bisa mengetahui kondisi politik, sosial, dan ekonomi dalam lingkup lokal, nasional bahkan global. Informasi yang didapat selanjutnya akan dipergunakan untuk menyusun rencana-rencana baru di bidang tertentu dengan mempertimbangkan keadaan terkini yang termuat dalam isi suatu berita. Untuk memenuhi kebutuhan harian akan berita, sebagian besar masyarakat memanfaatkan media berupa surat kabar baik itu surat kabar konvensional dan atau surat kabar elektronik sebagai media untuk menggali informasi terhangat.

Kenyataan berdasarkan hasil survey tentang rasio antara jumlah surat kabar dan jumlah penduduk Indonesia adalah 1:45 [2] merupakan tantangan berat bagi para jurnalis untuk dapat menghasilkan berita yang berkualitas, yang layak untuk dijadikan sumber informasi tambahan bagi masyarakat. Untuk membuat berita yang berkualitas, seorang jurnalis harus benar-benar memperhatikan komposisi dari naskah berita yang akan dibuat. Adapun komposisi naskah berita terdiri atas *head* (judul), *date line* (baris tanggal), *lead* (teras) dan *body* (isi) [3]. Dari komposisi naskah tersebut, ketertarikan calon pembaca bisa dipancing dengan memanfaatkan teras berita yang menarik di samping judul yang memang harus dibuat semenarik mungkin. Hal tersebut disebabkan karena dengan sedikitnya waktu yang dimiliki, pembaca akan menyaring berita-berita yang dianggapnya menarik dari sekian banyak berita yang termuat berdasarkan pertimbangan analisa teras berita.

Permasalahan yang sering dialami seorang jurnalis, terutama jurnalis pemula saat hendak membuat teras berita adalah banyaknya waktu yang diperlukan untuk itu. Karena untuk membuat beberapa kalimat agar mampu menggambarkan isi berita secara utuh bukanlah hal yang mudah. Sebagaimana diketahui, teras berita adalah paragraf yang memuat fakta atau informasi terpenting dari keseluruhan berita [4].

Dengan kemajuan teknologi seperti sekarang ini, pekerjaan seorang jurnalis dalam membuat teras berita bisa lebih dimudahkan dengan adanya perangkat lunak yang dapat membuat ringkasan dari sebuah artikel digital tanpa harus membaca dalam waktu yang cukup lama. Aplikasi tersebut dikenal dengan *Automated Text Summarization* (ATS). Dalam penelitian ini, peneliti hendak

mengembangkan aplikasi ATS dengan menambahkan *Text to Speech* (TTS) Bahasa Indonesia sehingga aplikasi tersebut bisa menghasilkan output berupa suara atau ucapan di samping ringkasan berupa teks dari artikel berita digital.

II. KAJIAN TEORI

A. Peringkasan Teks Otomatis

Peringkasan teks otomatis (*automatic text summarization*) adalah pembuatan versi yang lebih singkat dari sebuah teks dengan memanfaatkan aplikasi yang dijalankan pada komputer [5].

Untuk memperoleh unit teks yang paling informatif yang akan dimasukkan ke dalam ringkasan, biasanya dilakukan pemberian skor terhadap unit teks tertentu yang dianggap penting dalam sebuah teks dengan berbagai cara, diantaranya yaitu [5]: *positional criteria*, *cue phrase indicator criteria*, *word and phrase frequency criteria*, *query and title overlap criteria*, *cohesive or lexical connectedness criteria*, *discourse structure criteria*, *combination of various module scores*.

B. Sistem Text to Speech

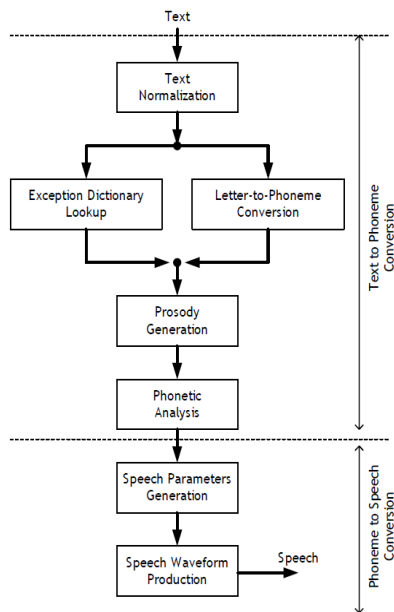
Pada dasarnya, TTS adalah suatu sistem yang dapat mengubah teks menjadi ucapan. Suatu sistem pensintesa ucapan atau *Text to Speech* pada prinsipnya terdiri dari dua sub sistem, yaitu [6]: bagian konverter teks ke fonem (*text to phoneme*) dan bagian konverter fonem ke ucapan (*phoneme to speech*).

Bagian konverter teks ke fonem berfungsi untuk mengolah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi urutan kode-kode bunyi yang direpresentasikan dengan kode fonem, durasi serta *pitch*-nya. Kode-kode fonem adalah kode yang merepresentasikan unit bunyi yang ingin diucapkan. Setiap fonem harus dilengkapi dengan informasi durasi dan *pitch*. Informasi durasi diperlukan untuk menentukan berapa lama suatu fonem diucapkan, sedangkan informasi *pitch* diperlukan untuk menentukan tinggi rendahnya nada pengucapan suatu fonem. Durasi dan *pitch* bersama-sama akan membentuk intonasi suatu ucapan. Kedua informasi ini dalam suatu sistem TTS biasanya dibangkitkan oleh modul pembangkit/model intonasi.

Bagian konverter fonem ke ucapan akan menerima masukan kode-kode fonem serta *pitch*

dan durasi yang telah dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian ini akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan.

Tahapan-tahapan utama konversi dari teks menjadi ucapan dapat dinyatakan dengan diagram seperti terlihat pada Gambar 1.



Gambar 1. Urutan Proses Konversi dari Teks ke Ucapan (dimodifikasi dari Pelton, 1992) [7]

C. Algoritma TF-IDF

Menurut Zhao, *et al.* (2000), TF-IDF (*Term Frequency Inverse Document Frequency*) dikenal sebagai algoritma yang didasarkan pada nilai statistik kemunculan suatu *template* dalam dokumen [8]. TF-IDF adalah nilai bobot dari suatu *template* yang diambil dari nilai TF dan nilai inversi DF, yang didefinisikan dengan [8]:

$$TF - IDF (xyz) = TF (xyz) \times \log \left(1 + \frac{nDoc}{DF (xyz)} \right) \quad (1)$$

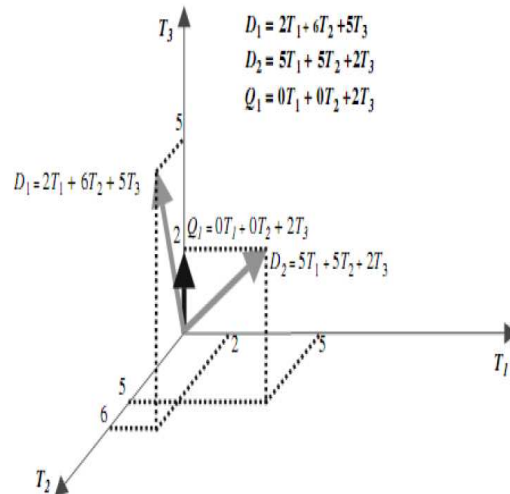
Dimana

- TF: jumlah munculnya binary xyz dalam satu dokumen
- DF: jumlah dokumen yang mengandung *template xyz*
- nDoc: jumlah dokumen

D. Vector Space Model

Vector space model adalah suatu model yang digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu *query* [9]. Pada model ini, setiap dokumen dalam *database* dan *query* pengguna direpresentasikan oleh suatu vektor multi-dimensi [10].

Gambar 2 merupakan contoh dari *vector space model* tiga dimensi untuk dua dokumen D_1 dan D_2 , satu *query* pengguna Q_1 , dan tiga term T_1, T_2, T_3 .



Gambar 2. Contoh *Vector Space Model* Menurut Cios (2007) [10]

Semakin dekat dua vektor dalam *vector space model*, maka semakin mirip pula dua dokumen yang diwakili oleh dua vektor tersebut. Kemiripan antar dokumen dapat dihitung dengan menggunakan suatu fungsi ukuran kemiripan (*similarity measure*). Menurut Cios (2007), jika terdapat dua vektor dokumen d_j dan q , serta t term diekstrak dari koleksi dokumen maka nilai *cosinus* antara d_j dan q didefinisikan sebagai berikut [10]:

$$\cos \theta \rightarrow \text{sim}(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^t (W_{i,j} \cdot W_{i,q})}{\sqrt{\sum_{i=1}^t W_{i,j}^2} \cdot \sqrt{\sum_{i=1}^t W_{i,q}^2}} \quad (2)$$

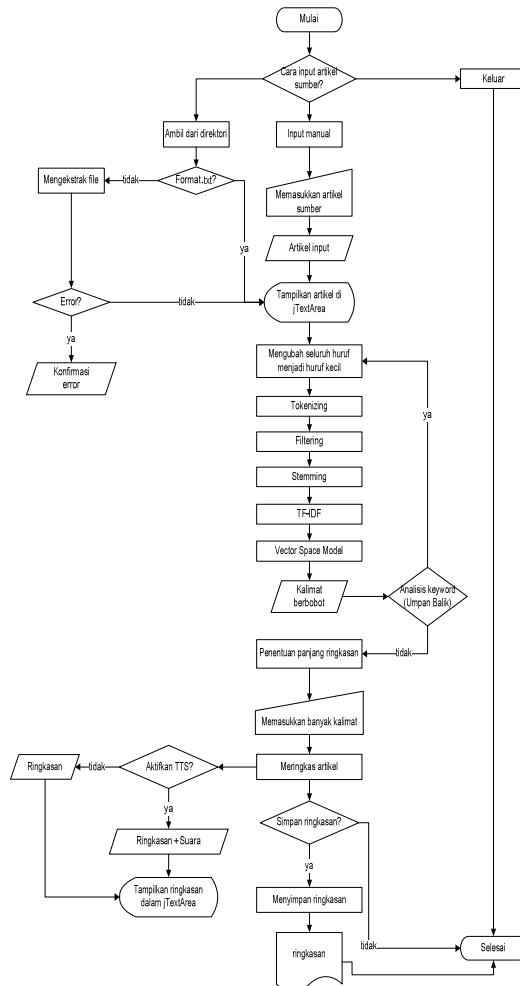
III. METODOLOGI

A. Analisis Masalah dan Solusi

Berita telah menjadi kebutuhan harian yang penting bagi masyarakat saat ini, karena berita dipandang sebagai sumber informasi yang selanjutnya dapat dijadikan bahan pertimbangan dalam pengambilan keputusan di bidang politik, sosial dan ekonomi. Salah satu sumber berita harian masyarakat adalah surat kabar. Akibat terbatasnya waktu, masyarakat cenderung akan memilih beberapa berita dari sekian banyak berita yang termuat dalam surat kabar untuk kemudian dibaca. Teknik yang biasa digunakan masyarakat untuk memilih berita dalam surat kabar adalah dengan menganalisa hasil bacaan pada paragraf pertama (*lead/teras* berita) disamping juga mempertimbangkan aspek redaksi judul. Dengan membaca *lead* (*teras* berita), masyarakat cukup mampu menilai bagaimana isi berita secara keseluruhan karena *lead* merupakan paragraf pertama pada sebuah artikel berita yang biasanya memuat kalimat-kalimat paling informatif yang mampu menggambarkan isi berita secara keseluruhan.

Fenomena ini menjadi tantangan tersendiri bagi para jurnalis (penulis berita) untuk dapat menghasilkan *lead* yang mampu menarik pembaca sebanyak-banyaknya. Dalam dunia jurnalistik, membuat sebuah *lead* bukanlah perkara mudah karena untuk membuat beberapa kalimat agar mampu menggambarkan isi berita secara utuh memerlukan waktu yang tidak sedikit.

Berdasarkan analisis masalah di atas, peneliti mencoba membuat suatu usulan solusi untuk mengatasi kondisi tersebut. Solusi yang dimaksud yaitu berupa pembuatan sebuah aplikasi pembuat *lead*. Aplikasi pembuat *lead* akan dirancang sedemikian rupa sehingga pada akhirnya mampu menghasilkan ringkasan berupa kalimat-kalimat paling informatif dari sebuah artikel berita digital. Selain menghasilkan ringkasan yang berupa teks, aplikasi yang akan dibuat juga dilengkapi dengan fitur pengucapan yang dalam hal ini adalah pengucapan (pembacaan) ringkasan yang telah dihasilkan dari proses sebelumnya. Dengan demikian, pekerjaan seorang jurnalis dalam membuat *lead* dapat diringankan terutama dari segi penggunaan waktu, selain itu adanya fitur yang bisa mengucapkan (membacakan) hasil ringkasan akan semakin memanjakan pengguna aplikasi nantinya. Gambar 3 merupakan diagram alir dari proses-proses dalam aplikasi pembuat *lead* (*teras* berita).



Gambar 3. Diagram Alir Aplikasi Pembuat *Lead* (*Teras*) Berita

B. Analisis Perangkat Lunak

Pada sub analisis perangkat lunak ini, peneliti akan memaparkan mengenai tahapan awal dari pengembangan perangkat lunak. Adapun komponen yang terkait dalam analisis perangkat lunak ini adalah sebagai berikut.

1) Kebutuhan Perangkat Lunak

Berdasarkan analisis terhadap pengembangan aplikasi terdapat beberapa proses yang dapat diimplementasikan, yaitu: membaca file teks berformat .docx dan .txt, melakukan proses penyeragaman huruf-huruf dalam artikel input menjadi huruf kecil secara keseluruhan, melakukan proses pemotongan artikel input berdasarkan kata-kata penyusunnya yang dinamakan dengan proses *tokenizing*, melakukan proses

pengambilan kata-kata penting hasil *token* yang dinamakan dengan proses *filtering*, melakukan proses pencarian akar kata dari setiap kata hasil *filtering* yang dinamakan dengan proses *stemming*, melakukan proses perhitungan bobot kalimat dengan menggunakan algoritma TF-IDF dan *Vector Space Model* yang kemudian dilanjutkan dengan proses perangkaian berdasarkan nilai cosinus sudut untuk menentukan tingkat keinformatifan suatu kalimat dalam artikel input, melakukan proses pemberian umpan balik berupa penambahan kata ke dalam daftar kata tidak penting yang merupakan hasil analisis *keyword* oleh pengguna sistem, melakukan proses peringkasan dimana panjang ringkasan disesuaikan dengan inputan pengguna sistem, melakukan proses pembacaan ringkasan dengan memanfaatkan modul TTS berbahasa Indonesia, melakukan proses penyimpanan ringkasan ke dalam file teks berformat .docx dan .txt, dan melakukan proses manajemen data baik itu data kata dasar maupun data kata tidak penting.

2) Tujuan Pengembangan Perangkat Lunak

Aplikasi ini diharapkan mampu memenuhi proses-proses sebagai berikut: mampu membaca file teks berformat .docx dan .txt, mampu melakukan proses penyeragaman huruf-huruf dalam artikel input menjadi huruf kecil secara keseluruhan, mampu melakukan proses pemotongan artikel input berdasarkan kata-kata penyusunnya yang dinamakan dengan proses *tokenizing*, mampu melakukan proses pengambilan kata-kata penting hasil *token* yang dinamakan dengan proses *filtering*, mampu melakukan proses pencarian akar kata dari setiap kata hasil *filtering* yang dinamakan dengan proses *stemming*, mampu melakukan proses perhitungan bobot kalimat dengan menggunakan algoritma TF-IDF dan *Vector Space Model* yang kemudian dilanjutkan dengan proses perangkaian berdasarkan nilai cosinus sudut untuk menentukan tingkat keinformatifan suatu kalimat dalam artikel input, mampu melakukan proses pemberian umpan balik berupa penambahan kata ke dalam daftar kata tidak penting yang merupakan hasil analisis *keyword* oleh pengguna sistem, mampu melakukan proses peringkasan dimana panjang ringkasan disesuaikan dengan inputan pengguna sistem, mampu melakukan proses

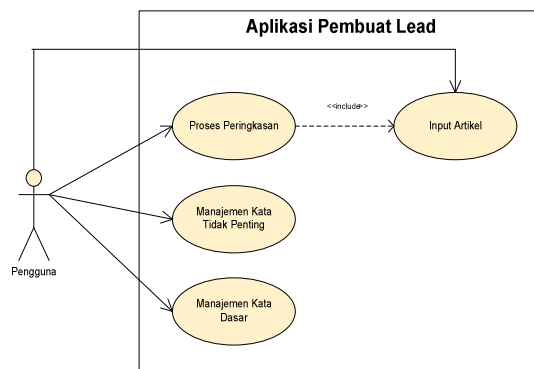
pembacaan ringkasan dengan memanfaatkan modul TTS berbahasa Indonesia, mampu melakukan proses penyimpanan ringkasan ke dalam file teks berformat .docx dan .txt, dan mampu melakukan proses manajemen data baik itu data kata dasar maupun data kata tidak penting.

3) Masukan dan Keluaran Perangkat Lunak

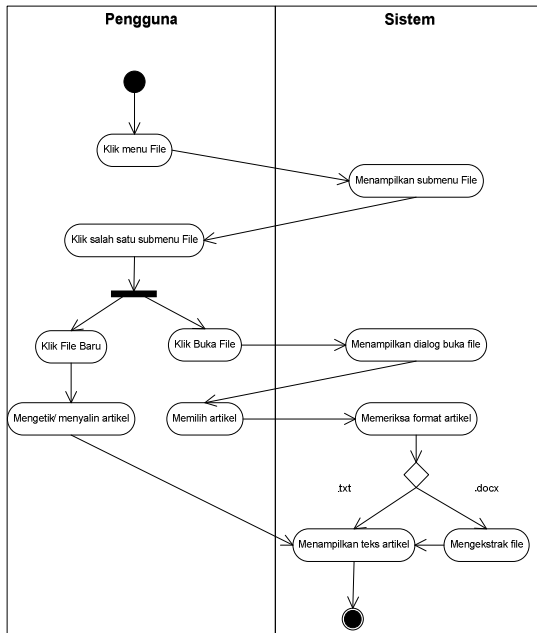
Masukan untuk aplikasi pembuat *lead* ini antara lain: artikel berita digital berformat .docx dan .txt, panjang ringkasan yang diinputkan oleh pengguna sistem, kata dari daftar *keyword* yang ditambahkan ke dalam daftar kata tidak penting saat terjadinya proses pemberian umpan balik oleh pengguna sistem, instruksi pengaktifan TTS, instruksi penyimpanan output dan data kata. Sedangkan keluaran dari aplikasi pembuat *lead* ini antara lain: daftar kata hasil *tokenizing*, daftar kata hasil *filtering*, daftar kata hasil *stemming*, daftar *keyword* lengkap dengan nilai *df* dan *idf* masing-masing, daftar kalimat yang telah terurut berdasarkan nilai cosinus sudut, teks hasil peringkasan, suara yang merupakan hasil pengaktifan TTS untuk membaca teks hasil peringkasan, file teks ringkasan berformat .docx atau .txt, daftar kata dasar maupun kata tidak penting hasil manajemen data kata, informasi mengenai aplikasi pembuat *lead* dan tata cara penggunaannya serta informasi mengenai pembuat aplikasi.

4) Model Fungsional Perangkat Lunak

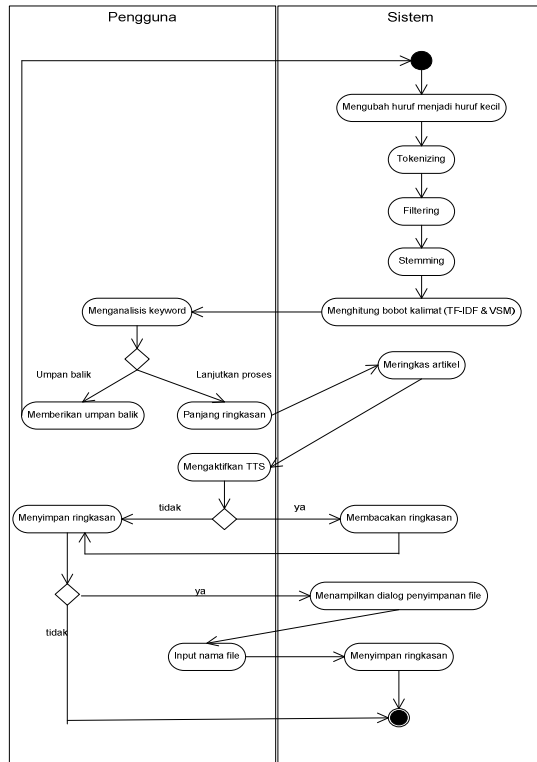
Adapun rancangan dari aplikasi pembuat *lead* ini akan digambarkan dengan menggunakan *Use Case Diagram*.



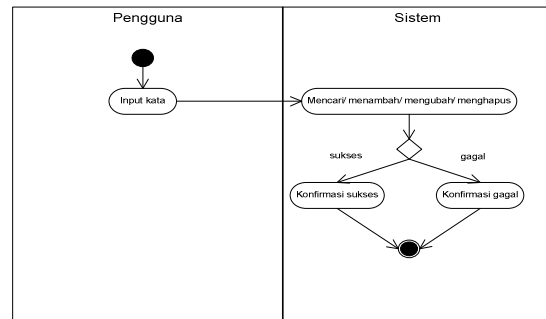
Gambar 4. Use Case Diagram Aplikasi Pembuat Lead



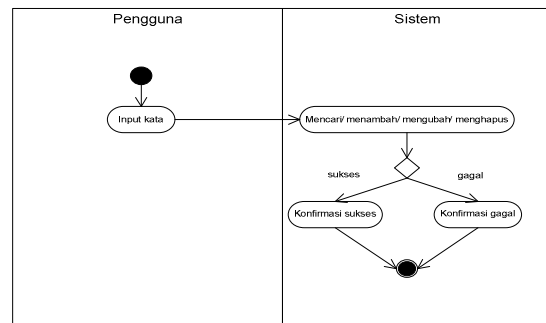
Gambar 5. Activity Diagram dari Proses Input Artikel pada Aplikasi Pembuat Lead



Gambar 6. Activity Diagram dari Proses Peringkasan pada Aplikasi Pembuat Lead



Gambar 7. Activity Diagram dari Proses Manajemen Kata Tidak Penting pada Aplikasi Pembuat Lead



Gambar 8. Activity Diagram dari Proses Manajemen Kata Dasar pada Aplikasi Pembuat Lead

C. Perancangan Perangkat Lunak

Tahap perancangan perangkat lunak merupakan tahap selanjutnya setelah melakukan analisis perangkat lunak. Adapun bagian-bagian dari tahapan ini dapat dijabarkan sebagai berikut.

1) Batasan Perancangan Perangkat Lunak

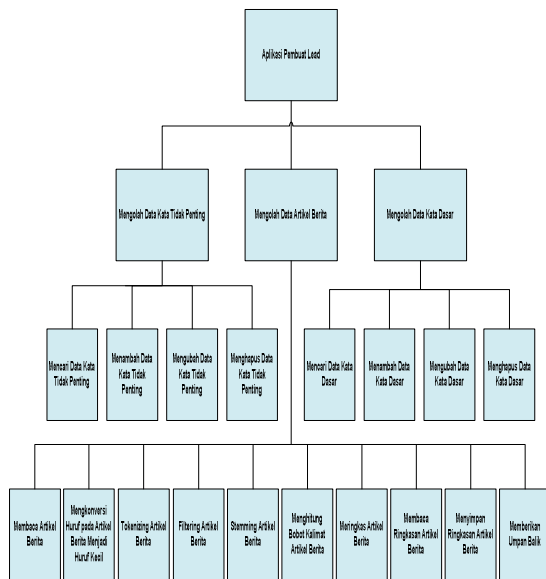
Adapun batasan perancangan dari aplikasi pembuat lead (teras) berita adalah sebagai berikut:

- Input dan output sistem adalah artikel berita berformat .docx dan .txt.
- Pada artikel input, panjang karakter maksimal dalam sebuah kata adalah 50 sedangkan untuk sebuah kalimat adalah 500.
- Panjang ringkasan diukur berdasarkan banyak kalimat yang ditentukan sendiri oleh pengguna sistem.
- Ringkasan yang dihasilkan sistem terdiri atas salinan kalimat-kalimat paling informatif yang ada dalam artikel input.
- Fitur TTS bersifat *optional*.

- Dalam proses manajemen kata, jumlah awal kata yang disiapkan masing-masing adalah 757 kata untuk kata tidak penting dan 28.526 kata untuk kata dasar.
- Sistem akan dikhususkan untuk artikel berita digital berbahasa Indonesia.
- *Lead* (teras) berita yang dihasilkan sistem berjenis *lead* ringkasan.

2) Perancangan Arsitektur Perangkat Lunak

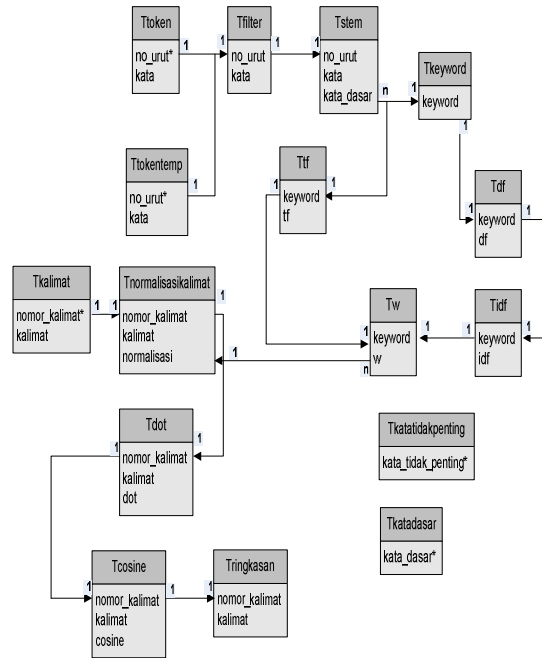
Perancangan arsitektur perangkat lunak menggambarkan bagian-bagian modul, struktur ketergantungan antar modul, dan hubungan antar modul dari perangkat lunak yang dibangun. Pada bagian ini terdapat *structure chart* sebagai kendali fungsional yang digambarkan seperti Gambar 9 untuk aplikasi pembuat *lead*.



Gambar 9. Struktur Chart Aplikasi Pembuat Lead

3) Perancangan Struktur Data Perangkat Lunak

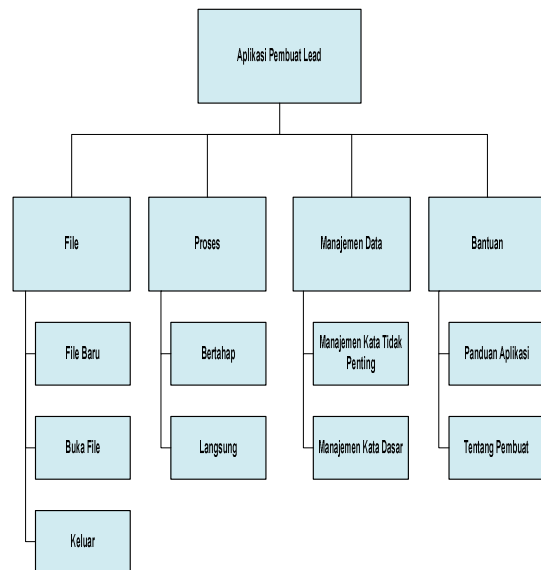
Dalam pengembangan aplikasi pembuat *lead* ini peneliti menggunakan sebuah database yaitu *lead* yang memiliki 16 tabel yaitu Token, Tfilter, Tstem, Tkeyword, Tkalimat, Ttf, Tdf, Tidf, Tw, Tnormalisasikalimat, Tdot, Tcosine, Tringkasan, Tkatadasar, Tkatatidakpenting, dan Ttokenemp. Gambar 10 adalah gambaran relasi antar tabel pada aplikasi pembuat *lead*.



Gambar 10. Relasi Antartabel pada Aplikasi Pembuat Lead

4) Perancangan Antarmuka Perangkat Lunak

Perancangan antarmuka perangkat lunak dibedakan menjadi perancangan struktur menu dan perancangan layar antarmuka.



Gambar 11. Struktur Menu Aplikasi Pembuat Lead

IV. PEMBAHASAN

A. Implementasi Perangkat Lunak

Implementasi dari aplikasi pembuat *lead* meliputi lingkungan implementasi, batasan implementasi, implementasi arsitektur, implementasi struktur data dan implementasi layar antarmuka perangkat lunak.

1) Lingkungan Implementasi Perangkat Lunak

Lingkungan implementasi aplikasi pembuat *lead* meliputi lingkungan perangkat keras dan perangkat lunak sebagai berikut.

- Spesifikasi Perangkat Keras
 - a. Seperangkat komputer dengan spesifikasi:
 - a) Intel Core 2 4 1,6 GHz
 - b) Harddisk 90 GB
 - c) Memory 2560 MB
 - d) Monitor 14"
 - e) Keyboard
- Spesifikasi Perangkat Lunak
 - a. Sistem Operasi Linux (Ubuntu 12.04 LTS)
 - b. NetBeans IDE 7.2.1
 - c. MySQL
 - d. eSpeak

2) Batasan Implementasi Perangkat Lunak

Adapun batasan implementasi dari aplikasi pembuat *lead* (teras) berita adalah sebagai berikut:

- Input dan output sistem adalah artikel berita berformat .docx dan .txt.
- Pada artikel input, panjang karakter maksimal dalam sebuah kata adalah 50 sedangkan untuk sebuah kalimat adalah 500.
- Panjang ringkasan diukur berdasarkan banyak kalimat yang ditentukan sendiri oleh pengguna sistem.
- Ringkasan yang dihasilkan sistem terdiri atas salinan kalimat-kalimat paling informatif yang ada dalam artikel input.
- Fitur TTS bersifat *optional*.

- Dalam proses manajemen kata, jumlah awal kata yang disiapkan masing-masing adalah 757 kata untuk kata tidak penting dan 28.526 kata untuk kata dasar.

- Sistem akan dikhususkan untuk artikel berita digital berbahasa Indonesia.

- *Lead* (teras) berita yang dihasilkan sistem berjenis *lead* ringkasan.

3) Implementasi Arsitektur Perangkat Lunak

Rancangan arsitektur perangkat lunak diimplementasikan menjadi dua belas *class* utama yang meliputi MenuForm.java, TokenizingForm.java, FilteringForm.java, StemmingForm.java, TFIDFForm.java, UmpanBalikForm.java, PeringkasanForm.java, KataDasarForm.java, KataTidakPentingForm.java, ProsesLangsungForm.java, PanduanForm.java, dan PembuatForm.java.

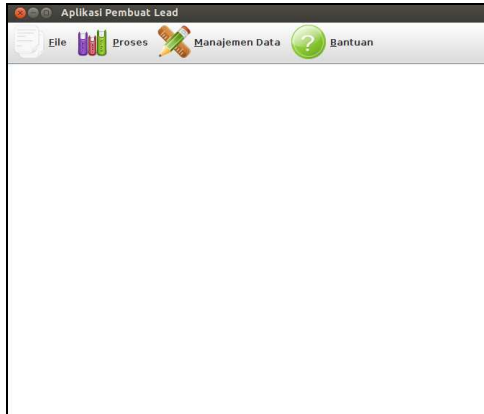
4) Implementasi Struktur Data Perangkat Lunak

Rancangan struktur data perangkat lunak diimplementasikan menggunakan perangkat lunak sistem manajemen basis data SQL (*Structured Query Language*) berupa MySQL. Dalam pengembangan aplikasi pembuat *lead* ini peneliti menggunakan sebuah *database* yaitu *lead* yang memiliki 16 tabel sebagai berikut: Ttoken, Tfilter, Tstem, Tkeyword, Tkalimat, Ttf, Tdf, Tidf, Tw, Tnormalisasikalimat, Tdot, Tcosine, Tringkasan, Tkatadasar, Tkatatidakpenting, dan Ttokenemp.

5) Implementasi Layar Antarmuka Perangkat Lunak

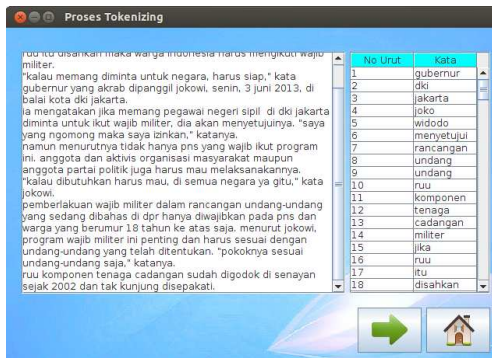
Bagian implementasi layar antarmuka perangkat lunak memaparkan tampilan-tampilan layar antarmuka aplikasi pembuat *lead* yang merupakan hasil implementasi dari rancangan antarmuka perangkat lunak menggunakan bahasa pemrograman Java dengan editor Netbeans IDE 7.2.1.

- Implementasi *Form Utama*



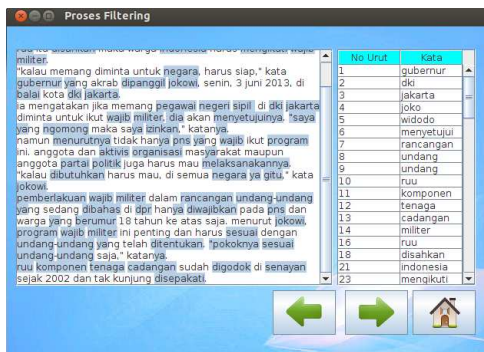
Gambar 12. Implementasi *Form Utama*

- Implementasi *Form Tokenizing*



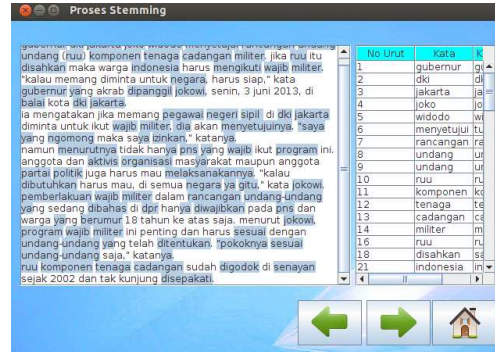
Gambar 13. Implementasi *Form Tokenizing*

- Implementasi *Form Filtering*



Gambar 14. Implementasi *Form Filtering*

- Implementasi *Form Stemming*



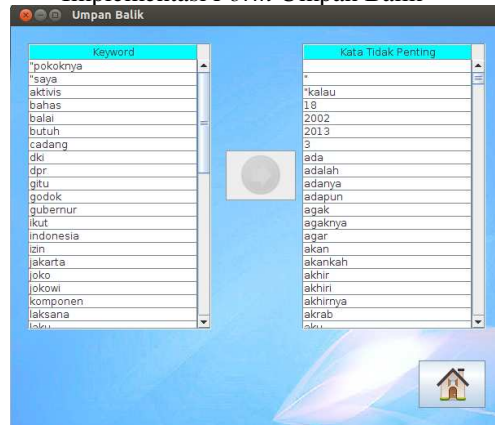
Gambar 15. Implementasi *Form Stemming*

- Implementasi *Form TF-IDF dan Vector Space Model*



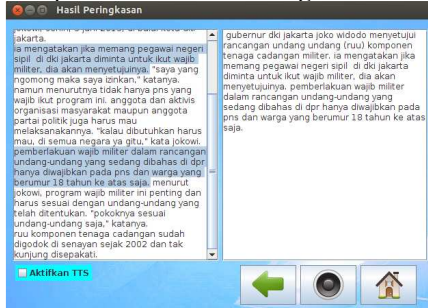
Gambar 16. Implementasi *Form TF-IDF dan Vector Space Model*

- Implementasi *Form Umpan Balik*



Gambar 17. Implementasi *Form Umpan Balik*

• Implementasi Form Peringkasan



Gambar 18. Implementasi Form Peringkasan

• Implementasi Form Peringkasan Langsung



Gambar 19. Implementasi Form Peringkasan Langsung

• Implementasi Form Manajemen Kata Dasar



Gambar 20. Implementasi Form Manajemen Kata Dasar

• Implementasi Form Manajemen Kata Tidak Penting



Gambar 21. Implementasi Form Manajemen Kata Tidak Penting

B. Pengujian Perangkat Lunak

Pengujian merupakan tahapan setelah implementasi dalam pengembangan perangkat lunak.

1) Tujuan Pengujian Perangkat Lunak

Pengujian aplikasi pembuat *lead* ditujukan untuk menguji sistem secara fungsional (*black box testing*) dan secara konseptual/struktural (*white box testing*).

2) Pelaksanaan Pengujian Perangkat Lunak

Pelaksanaan pengujian perangkat lunak dilakukan oleh: 1) Peneliti, untuk pengujian fungsionalitas sistem; 2) 28 orang responden yang terdiri dari 27 orang mahasiswa dan 1 orang dosen (berasal dari civitas akademika di lingkungan Universitas Pendidikan Ganesha), untuk pengujian kualitas teks hasil peringkasan; 3) Ahli bidang informatika (dosen di lingkungan jurusan Pendidikan Teknik Informatika, Universitas Pendidikan Ganesha), untuk pengujian kebenaran penerapan algoritma. Pengujian ini dilakukan secara bertahap dari tanggal 31 Mei sampai dengan tanggal 2 Juli 2013 dengan menggunakan tiga instrumen uji yaitu:

- Instrumen pengujian fungsionalitas aplikasi pembuat *lead*.
- Instrumen pengujian kualitas teks hasil peringkasan.
- Instrumen pengujian *white box* (kebenaran penerapan algoritma).

3) Evaluasi Hasil Pengujian Perangkat Lunak

Berdasarkan hasil pengujian melalui instrumen pengujian fungsionalitas diketahui bahwa aplikasi pembuat *lead* telah dapat berfungsi dengan baik. Semua kebutuhan yang menjadi tujuan pengembangan perangkat lunak telah terimplementasi dengan baik termasuk masukan dan keluaran sistem. Berdasarkan hasil pengujian melalui instrumen pengujian kualitas teks hasil peringkasan dapat diketahui bahwa secara umum, ringkasan dari kelima artikel yang diujicobakan dianggap sudah cukup mewakili isi artikel aslinya dengan persentase kesesuaian tertinggi yaitu 100% untuk artikel berjudul "Gempa 5,4 SR Guncang Bengkulu", dan persentase terendah sebesar 64,29% untuk artikel



berjudul “Eyang Subur Serious Ingin *Nyapres* dari Partai Demokrat”. Keakuratan hasil peringkasan sangat bergantung pada nilai bobot *keyword* yang merupakan dasar dari perhitungan nilai cosinus sudut masing-masing kalimat. Penentuan *keyword* sangat ditentukan oleh beberapa proses sebelumnya terutama proses *filtering* yang merupakan penentu berhak atau tidaknya suatu kata menjadi *keyword*. Mengingat hal tersebut, aktifitas pemberian umpan balik menjadi sangat penting karena jumlah kata tidak penting dalam *database* sangat terbatas. Untuk meningkatkan kualitas hasil peringkasan oleh sistem, maka tambahan pengetahuan dari pengguna mutlak diperlukan. Berdasarkan hasil pengujian melalui instrumen pengujian *white box* dapat diketahui bahwa aplikasi pembuat *lead* telah dapat mengimplementasikan algoritma TF-IDF dan *Vector Space Model* dengan baik. Setiap tahap perhitungan dalam algoritma tersebut sudah terimplementasi dengan baik sehingga menghasilkan hasil perhitungan yang sesuai.

V. SIMPULAN

Berdasarkan pengembangan aplikasi pembuat *lead*, dapat ditarik kesimpulan sebagai berikut.

1. Pengembangan aplikasi pembuat *lead* berita dengan cara menggabungkan ATS dan TTS berhasil menciptakan sebuah aplikasi yang dapat membuat ringkasan sekaligus memiliki kemampuan untuk membacakan ringkasan dari suatu artikel berita berbahasa Indonesia dalam format .docx dan .txt.
2. Hasil pengujian menunjukkan bahwa aplikasi pembuat *lead* berita telah dapat melakukan fungsinya dengan baik dari segi fungsional maupun konseptual/ struktural. Kualitas ringkasan yang dihasilkan cukup memuaskan dengan persentase kesesuaian tertinggi mencapai 100% dan terendah sebesar 64,29%.
3. Untuk meningkatkan kualitas teks hasil peringkasan, proses umpan balik menjadi kunci utama yang perlu diperhatikan.
4. Dalam pengembangan selanjutnya, diharapkan bisa dikembangkan aplikasi serupa berbasis web dengan format file input yang lebih variatif.

REFERENSI

- [1] Depdiknas (Departemen Pendidikan Nasional). (Eds) 2011. *Kamus Besar Bahasa Indonesia*. PT Gramedia Pustaka Utama. Jakarta.
- [2] Hs, Lasa. 2009. Peran Perpustakaan dan Penulis dalam Peningkatan Minat Baca Masyarakat. *VISI PUSTAKA*. 11(2) : 6.
- [3] Budiman, Rommy. 2012. *Penulisan Press Release*. Tersedia pada <http://www.google.co.id/url?sa=t&rct=j&q=komposisi%20naskah%20berita&source=web&cd=9&cad=rja&ved=0CGMQFjAJ&url=http%3A%2F%2Fkk.mercubuana.ac.id%2Ffiles%2F42002-14-117173164814.doc&ei=BIDkUMXSGsbnrAec1IDIAQ&usq=AFQjCNFA0K1-KkNOW0ypNvkIVHU5AWW-RA&bv=bv.1355534169.d.bmk> (diakses tanggal 27 Desember 2012).
- [4] Kompasiana. 2011. *Penulisan Berita*. Tersedia pada <http://edukasi.kompasiana.com/2011/12/19/penulisan-berita-423445.html> (diakses tanggal 3 Januari 2013).
- [5] Purwasih, Nurzaitun. 2009. Peringkasan Teks Otomatis Dokumen Tunggal Berbahasa Indonesia Menggunakan *Graph-based Summarization Algorithm* dan *Similarity* (Studi Kasus Artikel Berita). *Skripsi* Institut Teknologi TELKOM.
- [6] Arman, A.A. 2004. *Teknologi Pemrosesan Bahasa Alami sebagai Teknologi Kunci untuk Meningkatkan Cara Interaksi antara Manusia dengan Mesin*. Tersedia pada http://www.itb.ac.id/focus/focus_file/Pidato%20Ilmiah%20pada%20Sidang%20Terbuka%20PMB%202004.pdf (diakses tanggal 9 Desember 2012).
- [7] ----- . 2002. *Konversi dari Teks ke Ucapan*. Tersedia pada <http://indotts.melsa.net.id/Konversi%20dari%20Teks%20ke%20Ucapan.pdf>(diakses tanggal 17 Juli 2012).
- [8] Saputra, I. P. A. 2011. *Penggunaan algoritma TFIDF dalam proses hierarchical template matching*. Makalah disajikan pada Konferensi Nasional Sistem dan Informatika. Bali. November 12.
- [9] Herwansyah, Adhit. 2009. Aplikasi Pengkategorian Dokumen dan Pengukuran Tingkat Similaritas Dokumen Menggunakan Kata Kunci pada Dokumen Penulisan Ilmiah Universitas Gunadarma. Tersedia pada http://www.gunadarma.ac.id/library/articles/graduate/computer-science/2009/Artikel_10105046.pdf (diakses tanggal 16 Juni 2013).
- [10] Aisah, Siti. 2012. Text Mining dengan Algoritma Vector Space Model untuk Pencarian pada Ensiklopedia Hadits Berbasis Web. *Skripsi* Jurusan Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim.