

**PERINGKASAN DOKUMEN MENGGUNAKAN METODE EKSTRAKSI
KALIMAT DENGAN PENDEKATAN
MAXIMUM MARGINAL RELEVANCE DOCUMENT**

SKRIPSI

OLEH :

Nunuk Mahanani

0710962001-96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA**

2009

**PERINGKASAN DOKUMEN MENGGUNAKAN METODE EKSTRAKSI
KALIMAT DENGAN PENDEKATAN
MAXIMUM MARGINAL RELEVANCE DOCUMENT**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar

Sarjana dalam bidang Ilmu Komputer

OLEH :

Nunuk Mahanani

0710962001-96



PROGRAM STUDI ILMU KOMPUTER

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS BRAWIJAYA

2009

LEMBAR PENGESAHAN TUGAS AKHIR

**PERINGKASAN DOKUMEN MENGGUNAKAN METODE EKSTRAKSI
KALIMAT DENGAN PENDEKATAN
MAXIMUM MARGINAL RELEVANCE DOCUMENT**

oleh:

Nunuk Mahanani

0710962001-96

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 25 November 2009
dan dinyatakan memenuhi syarat untuk memperoleh gelar Sarjana Komputer dalam
bidang Ilmu Komputer

Pembimbing I, Pembimbing II,

Drs. A. Ridok, M.Kom

Drs. Marji, MT

NIP. 196808251994031002

NIP. 196708011992031001

Mengetahui,

Ketua Jurusan Matematika

Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, MSc

NIP. 196908071994121001

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Nunuk Mahanani
NIM : 0710962001-96
Jurusan : Matematika
Program Studi : Ilmu Komputer
Penulis tugas akhir berjudul : Peringkasan Dokumen Menggunakan Metode Ekstraksi kalimat dengan Pendekatan *Maximum Marginal Relevance Document*

Dengan ini menyatakan bahwa :

1. Isi dari tugas akhir yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Tugas Akhir ini.
2. Apabila dikemudian hari ternyata Tugas Akhir yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 25 November 2009

Yang menyatakan,

Nunuk Mahanani

NIM. 0710962001

**PERINGKASAN DOKUMEN MENGGUNAKAN METODE EKSTRAKSI
KALIMAT DENGAN PENDEKATAN
MAXIMUM MARGINAL RELEVANCE DOCUMENT**

ABSTRAK

Automatic Text Summarization (ATS) merupakan sebuah aplikasi berbasis komputer yang melakukan proses untuk menghasilkan ringkasan/summary dari sebuah dokumen tetapi tetap memiliki gambaran yang akurat dari isi dokumen tersebut. Dalam penelitian ini akan dibangun sebuah aplikasi peringkasan dokumen dengan metode *extractive summary* yang merupakan cara peringkasan otomatis yang mengambil kalimat atau kata yang diambil secara utuh dari dokumen teks aslinya (Mani and Maybury, 1999). Peringkasan dokumen dengan metode *MMR-D* adalah salah satu metode *extractive summary* yang didasarkan pada pengambilan kalimat yang mempunyai relevansi tertinggi sebagai ringkasan dokumen baru yang ditentukan dengan perhitungan bobot *similarity* (kemiripan) kalimat tersebut pada keseluruhan dokumen (Ganapathiraju. Madhavi K, 2002). Untuk mengevaluasi metode yang digunakan dan kualitas hasil ringkasan yang dihasilkan, dilakukan perhitungan *precision*, *recall* dan *f-measure*(β) (perbandingan kalimat yang relevan dengan ringkasan ideal) ringkasan hasil metode ini dari teks berita berbahasa Indonesia dengan berbagai macam ukuran yang berbeda. Ringkasan ideal yang dipakai adalah ringkasan manual yang dibuat oleh 3 orang abstractor, yang mana satu orang merupakan seorang ahli bahasa Indonesia. Para pengguna ini diberi 15 teks berita yang sudah dipisah-pisah perkalimat untuk dibuat ringkasannya. Selain itu, ringkasan hasil sistem juga dibandingkan dengan ringkasan yang dihasilkan oleh sistem peringkasan otomatis lainnya yaitu sistem peringkasan otomatis dengan algoritma *shortest path*. Hasil ujicoba dan evaluasi menunjukkan bahwa sistem peringkasan ini menghasilkan ketepatan ringkasan dengan rata-rata *recall* sebesar 55%, rata-rata *precision* sebesar 58%, dan rata-rata *f-measure*(β) sebesar 63%. Setelah dibandingkan dengan sistem peringkasan otomatis dengan algoritma *shortest path*, dihasilkan sistem peringkasan dokumen ini lebih baik.

repository.ub.ac.id

AUTOMATIC DOCUMENT SUMMARIZATION USING SENTENCE EXTRACTION WITH A MAXIMUM MARGINAL RELEVANCE APPROACH

ABSTRACT

Automatic Text summarization (ATS) is a computer-based application process to produce a summary of a document but still have an accurate picture of the contents of the document. In this study will build an application for document summarization using Extractive summarizer method, an automated way to take a sentence or word which is taken as a whole from the original text documents (Mani and Maybury, 1999). Summarization of document with MMR-D method is one of the Extractive methods based on the summary of making sentences that have the highest relevance as a summary of the new document that is determined by calculating the weight similiarity of the sentence in the entire document (Ganapathiraju. K Madhavi, 2002). To evaluate the methods and the quality of the resulting summary, do the calculation precision, recall and f-measure (β) (a comparison that is relevant sentence to an ideal summary) summarizing the results of this method of text news in Indonesian language with a variety of different sizes. Ideal summary is manual summary created by 3 people abstractor, which one person is an expert in Indonesian. The user is given the news that 15 texts are separated by sentence to make a summary. In addition, a summary results are also compared with other automatic system summarzation that using shortest path algorithm. Test and evaluation results show that the system generates precision with the average recall of 55%, the average precision of 58%, and the average f-measure (β) of 63%. When compared with the automated system using shortest path algorithm, this system has a better results than the other system.

KATA PENGANTAR

Alhamdulillah rabbil 'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayahnya, Tugas Akhir yang berjudul “Peringkasan Dokumen Menggunakan Metode Ekstraksi Kalimat dengan Pendekatan Maximum Marginal Relevance Dokument” ini dapat diselesaikan. Tugas Akhir ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, universitas Brawijaya.

Semoga Allah melimpahkan rahmat atas Nabi Muhammad SAW yang senantiasa memberikan cahaya petunjuk, dan atas keluarganya yang baik dan suci dengan rahmat yang berkah-Nya menyelamatkan kita pada hari akhirat.

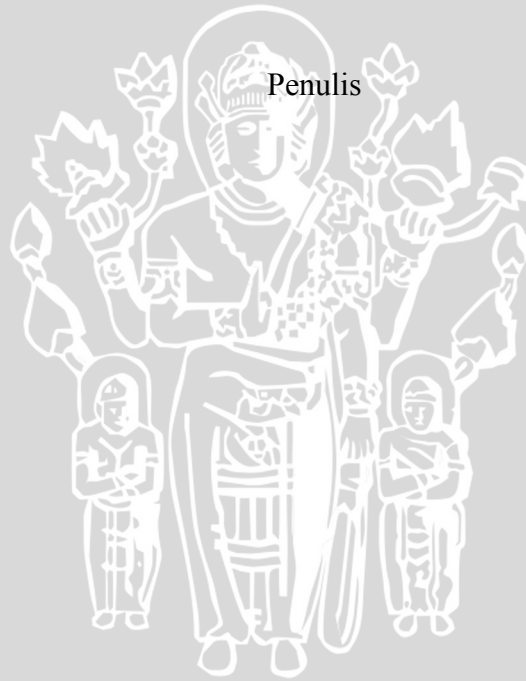
Dalam penyelesaian tugas akhir ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Drs. A. Ridok, M.Kom, selaku pembimbing utama penulisan skripsi ini.
2. Drs. Marji, MT, selaku pembimbing pendamping dalam penulisan skripsi ini dan selaku Ketua Program Studi Ilmu Komputer.
3. Drs. Muh. Arif Rahman, M.Kom selaku Penasihat Akademik.
4. Dr. Agus Suryanto, Msc selaku Ketua Jurusan Matematika.
5. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
6. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan proposal skripsi ini.
7. Orang tua Penulis atas dukungan materi dan doa restu yang tak henti – hentinya kepada Penulis.
8. Kakak serta adik Penulis yang juga memberikan dukungan setiap harinya.
9. Rekan – rekan di Program Studi Ilmu komputer FMIPA Universitas Brawijaya yang telah banyak memberikan bantuannya demi kelancaran pelaksanaan penyusunan tugas akhir ini.
10. Dan semua pihak yang telah membantu dalam penyusunan skripsi ini yang tidak dapat Penulis sebutkan satu persatu.

Semoga penulisan laporan tugas akhir ini bermanfaat bagi pembaca sekalian. Dengan tidak lupa kodratnya sebagai manusia, penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan, dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Malang, November 2009

Penulis



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
LEMBAR PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR PSEUDOCODE.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Metodologi Pemecahan Masalah	4
1.7 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Konsep Automatic Text Summarization	7
2.1.1 Tipe Ringkasan.....	7
2.1.2 Peringkasan Teks Otomatis dengan <i>Extractive Method</i>	8
2.2 Tipe Evaluasi	10
2.3 Proses Text Sumarization	12
2.4 Text Pre-processing	13
2.4.1 Case Folding	13
2.4.2 Parsing	13
2.4.3 Penghilangan Stopword	13
2.5 Cosine	13
2.6 Metode <i>MMR-D</i> (Maximum Marginal Relevansi - Dokumen)	18



2.7 Pengertian dan Penulisan Kata	20
2.7.1 Kata Dasar	20
2.7.2 Kata Turunan	20
2.7.3 Bentuk Ulang	21
2.7.4 Gabungan Kata	21
2.8 Pengertian Kalimat	21
BAB III METODOLOGI PERANCANGAN	23
3.1 Analisis Sistem.....	23
3.1.1 Deskripsi Umum Sistem	23
3.1.2 Batasan Sistem	24
3.1.3 Analisis Kebutuhan Perangkat Lunak	25
3.1.3.1 Analisis Use Case	25
3.1.3.2 Analisis Kelas	25
3.2 Perancangan Sistem	28
3.2.1 Perancangan Proses	28
3.2.1.1 Get Preprocessing	30
3.2.1.2 Proses Inisialisasi	33
3.2.1.3 Proses Get MMR	34
3.2.2 Perancangan Antar Muka	36
3.3 Evaluasi dan Perbandingan Sistem	37
3.4 Contoh Perhitungan Manual	41
BAB IV IMPLEMENTASI DAN UJICOBA	49
4.1 Lingkungan Implementasi	49
4.1.1 Lingkungan Perangkat Keras	49
4.1.2 Lingkungan Perangkat Lunak	49
4.2 Implementasi Proses Pembentuk Sistem	49
4.2.1 Source Code Proses Get Preprocessing	50
4.2.2 Source Code Proses Inisialisasi Sum(0)	51
4.2.3 Source Code Proses Get MMT	52
4.3 Implementasi Antar Muka	54
4.4 Implementasi Uji Coba	59
4.4.1 Hasil Evaluasi	59
4.4.2 Perbandingan Sistem	62
4.4.3 Analisa Hasil	67

BAB V KESIMPULAN DAN SARAN 71

5.1 Kesimpulan 71

5.2 Saran 71

DAFTAR PUSTAKA..... xxiii

LAMPIRAN-LAMPIRAN



DAFTAR GAMBAR

Gambar 2.1 Skalar Vektor (Scalar Product)	15
Gambar 3.1 Diagram <i>Use Case</i>	27
Gambar 3.2 Modul Proses Umum Sistem	28
Gambar 3.3 Alur Pembentukan Ringkasan	32
Gambar 3.4 Alur Get Processing	34
Gambar 3.5 Alur Proses Inisialisasi	35
Gambar 3.6 Alur Proses Get MMR	36
Gambar 3.7 Alur Proses Get Hitung MMR(i)	37
Gambar 3.8 Rancangan antar Muka	39
Gambar 4.1 Antarmuka Proses Set Variabel	57
Gambar 4.2 Antarmuka Dokumen Asal	58
Gambar 4.3 Antarmuka Pemecahan Dokumen Asal Menjadi Kalimat	58
Gambar 4.4 Antarmuka Daftar TF	59
Gambar 4.5 Antarmuka Daftar Nilai Cosine Similiarity	60
Gambar 4.6 Antarmuka Daftar Perhitungan MMR	60
Gambar 4.7 Antarmuka Summary Sistem	61
Gambar 4.8 Grafik <i>precision</i> sistem peringkasan otomatis MMR dan <i>shortest path</i>	67
Gambar 4.9 Grafik <i>recall</i> sistem peringkasan otomatis MMR dan <i>shortest path</i>	68
Gambar 4.10 Grafik <i>f-measure</i> sistem peringkasan otomatis MMR dan <i>shortest path</i>	68

DAFTAR TABEL

Tabel 2.1	Tabel sebaran vektor kata	15
Tabel 2.2	Tabel perhitungan vektor kata	17
Tabel 3.1	Tabel Hasil Ringkasan Ideal	41
Tabel 3.2	Tabel Pembanding Sistem (MMR)	41
Tabel 3.3	Tabel Hasil akhir <i>recall</i> , <i>precision</i> dan <i>f-measure</i> sistem (MMR)	41
Tabel 3.4	Tabel Pembanding Sistem (<i>shortest path</i>)	42
Tabel 3.5	Tabel Hasil akhir <i>recall</i> , <i>precision</i> dan <i>f-measure</i> sistem (<i>shortest path</i>)	42
Tabel 3.6	Tabel Frekuensi/Kemunculan Kata DA dan RL	44
Tabel 3.7	Tabel Perhitungan Frekuensi dan Pembobotan Kata ...	45
Tabel 3.8	Tabel Iterasi	50
Tabel 4.1	Tabel ringkasan ideal	62
Tabel 4.2	Tabel Hasil Sistem MMR	62
Tabel 4.3	Tabel Hasil Pembanding Sistem (MMR)	63
Tabel 4.4	Tabel Hasil akhir <i>recall</i> , <i>precision</i> dan <i>f-measure</i> sistem (MMR)	64
Tabel 4.5	Tabel Hasil akhir <i>precision</i> , <i>recall</i> dan <i>f-measure</i> nilai koefisien lamda (λ) antara [0 – 1]	64
Tabel 4.6	Tabel Hasil Sistem Peringkat (<i>shortest path</i>)	65
Tabel 4.7	Tabel Hasil Pembanding Sistem (<i>shortest path</i>)	65
Tabel 4.8	Tabel Hasil akhir <i>recall</i> , <i>precision</i> dan <i>f-measure</i> sistem (<i>shortest path</i>)	66
Tabel 4.9	Tabel Hasil Pengukuran <i>recall</i> , <i>precision</i> dan <i>f-measure</i> Antara dua Sistem Peringkat Otomatis	66
Tabel 4.10	Tabel Pengukuran Peluang Pemilihan Kalimat Sama.	79



DAFTAR PSEUDOCODE

Pseudocode 4.1 Prosedur proses pemecahan dokumen asal menjadi kalimat 52

Pseudocode 4.2 Prosedur proses pemecahan kalimat menjadi kata 53

Pseudocode 4.3 Prosedur proses penghapusan stopwords..... 53

Pseudocode 4.4 Prosedur proses pembentukan summary awal..... 54

Pseudocode 4.5 Prosedur proses perhitungan persamaan cosine similiarity 35

Pseudocode 4.6 Prosedur proses perhitungan persamaan MMR..... 55

Pseudocode 4.7 Prosedur proses Insert To List 55

Pseudocode 4.8 Prosedur proses Get Konvergen 56



BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat berpengaruh juga pada meningkatnya ketersediaan informasi. Informasi ini dapat diperoleh dari berbagai media cetak maupun elektronik. Ketersediaan informasi yang semakin banyak ini, menuntut adanya sebuah alat yang mampu membuat ringkasan secara otomatis, sehingga informasi bisa didapatkan dengan cepat dan tidak meninggalkan makna aslinya.

Dengan adanya ringkasan, diharapkan pembaca dapat dengan cepat dan mudah memahami makna sebuah teks tanpa harus membaca keseluruhan teks. Hal ini dapat menghemat waktu pembaca karena dapat menghindari pembacaan teks yang tidak relevan. Pengaruh komputer sebagai alat untuk membantu mempermudah pekerjaan manusia pada saat ini sangat berpengaruh terhadap kinerja manusia dalam menyelesaikan sebuah pekerjaan.

Automatic Text Summarization (ATS), yang biasanya disebut dengan *Text Summarization/Peringkasan Text* merupakan sebuah aplikasi berbasis komputer yang melakukan proses untuk menghasilkan ringkasan/summary dari sebuah dokumen tapi tetap memiliki gambaran yang akurat dari isi suatu dokumen. Tujuan utamanya adalah mengambil sumber informasi dengan mengutip sebagian besar isi dokumen yang penting dan menampilkan kepada pembaca dalam bentuk yang ringkas dan sesuai dengan kebutuhan pembaca. Dengan demikian aplikasi ini diharapkan dapat membantu pembaca untuk menyerap informasi yang ada pada dokumen lewat ringkasan/summary. *Text summarization* akan menghasilkan suatu produk teks yang tetap memiliki/mengandung bagian-bagian yang penting dari dokumen aslinya.

Menurut Spark-Jones (1999), terdapat beberapa faktor penting yang dibuat sebagai pertimbangan untuk membuat suatu ringkasan. Faktor – faktor tersebut adalah faktor input dokumen teks apa yang akan diproses, faktor tujuan dan peruntukan kepada siapa ringkasan dibuat dan faktor output bentuk ringkasan yang diinginkan. Paling banyak peringkasan dokumen pada saat ini didasarkan pada ekstraksi kalimat dari dokumen asli. Dalam metode ekstraksi kalimat perhitungan nilai dari setiap kalimat didasarkan posisi kalimat tersebut pada sebuah dokumen (Baxendale dan Edmundson, 1958). Biasanya kalimat inti sebuah dokumen terletak pada awal atau akhir sebuah paragraf. Dengan dilakukannya perhitungan nilai (bobot) sebuah kata dalam kalimat, maka dapat dilihat tingkat relevansi kata tersebut dengan

dokumen. Sebuah kata dikatakan relevan dengan dokumen yaitu apabila kata tersebut mempunyai bobot yang tinggi (Jagadeesh J, dkk. 2004).

Ringkasan (summary) dapat diklasifikasikan menjadi 2 yaitu *extractive summary* dan *abstractive summary* (Mani and Maybury, 1999). *Extractive summary* merupakan cara peringkasan yang mengambil kalimat atau kata yang diambil secara utuh dari dokumen teks aslinya. Cara peringkasan ini tidak akan mengubah kata yang baru yang tidak ada dalam dokumen aslinya. Pada teknik *extractive*, sistem akan menyalin informasi yang dianggap penting dari teks asli menjadi ringkasan (sebagai contoh, klausa utama, kalimat utama, atau paragraf utama). *Abstractive summary* merupakan cara peringkasan yang menambahkan kata baru dan dapat merubah susunan kalimat. Pada *Abstractive summary* umumnya membutuhkan pembangkit kalimat (*language generator*) yang lebih kompleks dan membutuhkan waktu lebih lama.

Pada penelitian sebelumnya telah dibuat sebuah otomatisasi peringkasan dokumen dengan metode ekstraksi kalimat dengan menggunakan gabungan algoritma *shortest path* dan algoritma *dijkstra* (Wahyuni, Indah. 2007). Pada penelitian tersebut ditunjukkan bahwa peringkasan dokumen teks yang dilakukan secara otomatis dengan memperhatikan 5 fitur atau parameter, yaitu lokasi kalimat (*location feature*), kemiripan dengan judul (*title feature*), frekuensi kemunculan kata (*word frequency feature*), adanya kata kunci (*cue word feature*), dan *query feature*.

Pada penelitian kali ini, penulis ingin mencoba suatu metode lain dalam menyelesaikan permasalahan peringkasan dokumen secara otomatis yaitu dengan menggunakan pendekatan *Maximal Marginal Relevance Document (MMR-D)*. Peringkasan dokumen dengan metode *MMR-D* adalah salah satu metode *extractive summary* didasarkan pada pengambilan kalimat yang mempunyai bobot tertinggi sebagai ringkasan dokumen baru yang ditentukan dengan perhitungan bobot *similarity* (kemiripan) kalimat tersebut pada keseluruhan dokumen (Ganapathiraju. Madhavi K, 2002). Metode *MMR-D* mempunyai keunggulan dalam mengurangi *redundancy Sentence* (informasi atau kalimat yang berulang dalam suatu dokumen) yang nantinya akan dibuat ringkasannya (Shasha Xie dan Yang Liu, 2007). Untuk merepresentasikan kalimat individu dan keseluruhan dokumen digunakan *Vector Space Model (VSM)*. Kesamaan setiap kalimat pada sebuah dokumen atau pada query yang diberikan dihitung dengan metode *cosine similarity*. Nilai *cosine similarity* didapatkan dari perhitungan jarak antar vektor kalimat satu dengan kalimat yang lain pada sebuah dokumen. Semakin besar nilai *cosine similarity* maka semakin besar pula relevansi sebuah kalimat

tersebut. Kalimat yang dipilih sebagai kesimpulan adalah kalimat yang memiliki nilai *cosine similarity* yang besar.

Berdasarkan pada latar belakang yang telah dipaparkan, maka penulis mengambil judul pada skripsi ini dengan **“PERINGKASAN DOKUMEN MENGGUNAKAN METODE EKSTRAKSI KALIMAT DENGAN PENDEKATAN MAXIMUM MARGINAL RELEVANCE DOCUMENT”**.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang masalah, maka dalam tugas akhir ini dapat dibuat rumusan masalah yaitu

1. Bagaimana membuat ringkasan/*summary* dari sebuah dokumen teks berbahasa Indonesia dengan menggunakan metode ekstraksi kalimat dengan pendekatan *Maximal Marginal Relevance Document (MMR-D)*?
2. Apakah hasil ringkasan yang dihasilkan oleh peringkasan dokumen ini lebih baik jika dibandingkan dengan peringkasan teks otomatis menggunakan metode *shortest path* yang pernah dilakukan pada penelitian sebelumnya?

1.3 Batasan Masalah

Dari permasalahan yang telah dipaparkan dalam subbab 1.2, berikut ini diberikan batasan masalah untuk menghindari melebarnya masalah yang akan diselesaikan adalah pada penelitian kali ini dokumen yang digunakan hanya dokumen berita berbahasa Indonesia.

1.4 Tujuan

Tujuan yang ingin dicapai dalam pelaksanaan skripsi ini adalah :

1. Membuat aplikasi peringkasan dokumen yang mampu mengimplementasikan metode ekstraksi kalimat dengan pendekatan *Maximal Marginal Relevance Document (MMR-D)* untuk membuat ringkasan sebuah dokumen berbahasa Indonesia.
2. Mengevaluasi ringkasan yang dihasilkan oleh sistem peringkasan dokumen ini.
3. Membandingkan ringkasan hasil perangkat lunak ini dengan ringkasan hasil perangkat lunak yang lain.

1.5 Manfaat

Manfaat yang dapat diambil dari penulisan ini skripsi ini adalah :

1. Menyediakan suatu sistem yang dapat membantu pembaca dalam membuat ringkasan/summary dari suatu dokumen secara otomatis.
2. Membantu pembaca dalam mengetahui isi suatu dokumen secara cepat.
3. Mempersingkat waktu dalam membaca informasi atau berita yang terkandung dalam suatu dokumen.

1.6 Metodologi Pemecahan Masalah

Untuk mencapai tujuan yang dirumuskan sebelumnya, maka metodologi yang digunakan dalam penulisan tugas akhir ini adalah:

1. Studi literatur

Mempelajari teori-teori yang berhubungan dengan konsep peringkasan dokumen menggunakan metode ekstraksi kalimat dengan pendekatan *Maximal Marginal Relevance Document (MMR-D)* dari berbagai referensi.

2. Pendefinisian dan analisis masalah

Mendefinisikan dan menganalisis masalah untuk mencari solusi yang tepat.

3. Perancangan dan implementasi sistem

Membuat perancangan perangkat lunak dengan analisis terstruktur dan mengimplementasikan konsep peringkasan dokumen menggunakan metode ekstraksi kalimat dengan pendekatan *Maximal Marginal Relevance Document (MMR-D)*.

4. Uji coba dan analisa hasil implementasi

Menguji perangkat lunak dan menganalisa hasil dari implementasi tersebut apakah sudah sesuai dengan tujuan yang dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Tugas akhir ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN

Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi pemecahan masalah, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Menguraikan teori-teori yang berhubungan dengan peringkasan dokumen tunggal dengan metode ekstraksi kalimat dengan pendekatan *Maximal Marginal Relevance Document (MMR-D)* dan pemrosesan teks.

3. BAB III METODOLOGI DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai langkah-langkah yang digunakan dalam pemrosesan peringkasan dokumen berbahasa indonesia dengan menggunakan metode ekstraksi kalimat dengan pendekatan *Maximal Marginal Relevance Document (MMR-D)*.

4. BAB IV IMPLEMENTASI DAN UJI COBA SISTEM

Pada bab ini akan dilakukan implementasi sistem, pengujian dan analisa sistem perangkat lunak yang dibangun.

5. BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari seluruh rangkaian penelitian serta saran kemungkinan pengembangannya.



BAB II

DAFTAR PUSTAKA

2.1 Konsep Automatic Text Summarization

Summarization / ringkasan adalah penyajian karangan atau peristiwa yang panjang dalam bentuk yang singkat dan efektif. Menurut Mani dan Maybury, 1999, ringkasan merupakan proses dari pembuatan intisari informasi terpenting dari suatu sumber untuk menghasilkan versi yang lebih ringkas bagi penggunanya.

Peringkasan teks otomatis (*Automatic Text Summarization*) adalah pembuatan versi yang lebih singkat dari sebuah teks dengan memanfaatkan aplikasi yang dijalankan pada komputer. Dalam Hovy (2001), *summary* atau ringkasan didefinisikan sebagai sebuah teks yang dihasilkan dari satu atau lebih teks, mengandung informasi dari teks asli dan panjangnya tidak lebih dari setengah teks asli. Hasil peringkasan ini mengandung poin-poin penting dari teks asli. Tujuan utamanya adalah untuk menunjukkan beberapa kalimat utama dalam artikel yang diambil sebagai ringkasan dengan harapan pembaca akan memahami sebanyak mungkin isi ringkasan sebagai bagian dari artikel dalam waktu singkat (S Budhi. Gregorius, 2004).

2.1.1 Tipe Ringkasan

Berdasarkan teknik pembuatannya, ringkasan dibagi menjadi dua tipe (Mani, 2001), yaitu :

- Teknik ekstraksi (*shallower approaches*)

Pada teknik ekstraksi, sistem yang akan menyalin sebagian kalimat/informasi yang dianggap paling penting dari teks asli menjadi ringkasan (sebagai contoh, klausa utama, kalimat utama, atau paragraf utama). Teknik ini menggunakan metode *statistical*, *linguistical*, dan *heuristic* atau kombinasi dari semuanya dalam menetapkan ringkasan suatu teks. Kalimat yang diambil sebagai ringkasan dihitung dengan menggunakan beberapa parameter.

- Teknik abstraksi (*deeper approaches*).

Pada teknik abstraksi melibatkan parafrase dari teks asli. Abstraktif menghasilkan sebuah interpretasi terhadap teks aslinya. Dimana sebuah kalimat akan ditransformasikan menjadi kalimat yang lebih singkat dan kalimat baru yang tidak terdapat dalam dokumen yang asli.

Pada umumnya, teknik abstraksi dapat meringkas teks lebih kuat daripada teknik ekstraksi, tetapi sistemnya lebih sulit dikembangkan karena mengaplikasikan teknologi representasi

semantik, *inferens* dan *natural language generation* yang merupakan bahasan yang dikembangkan tersendiri (Erkan & Radev, 2004).

Berdasarkan tujuan dari pembuatannya, peringkasan teks dapat dikategorikan berdasarkan maksud, fokus, dan cakupannya (Firmin dan Chrzanowski, 1999), sebagai berikut :

□ Informatif vs Indikatif vs Evaluatif

- Informatif, ringkasan ini menyatakan informasi-informasi penting yang terdapat pada dokumen asal.
- Indikatif, tujuan dari ringkasan ini adalah untuk dijadikan sebuah referensi, yang membantu pembaca untuk mengetahui isi dari teks daripada membaca keseluruhan teks yang ada. Ringkasan ini meliputi topik kunci dari teks asal.
- Evaluatif, atau ringkasan yang melibatkan pembuatan sebuah pertimbangan pada teks asal, seperti suatu tinjauan ulang atau opini.

□ *User-focused* vs *Generic*

- *User-focused (query-relevant)*, ringkasan yang dibuat berdasarkan topik yang dipilih oleh user, sering merupakan jawaban dari *query* yang dimiliki oleh user.
- *Generic*, disebut juga *author-focused*, sifatnya lebih umum dan berdasarkan pada teks aslinya.

□ Dokumen tunggal (*single document*) vs Banyak Dokumen (*multi document*)

- Dokumen tunggal, ringkasan merupakan ringkasan dari satu dokumen
- Banyak Dokumen, ringkasan merupakan hasil ringkasan dari banyak dokumen.

2.1.2 Peringkasan Teks Otomatis dengan *Extractive Method*

Beberapa cara telah dikembangkan agar diperoleh unit teks yang paling informatif yang akan dimasukkan ke dalam ringkasan. Beberapa hal yang biasanya dilakukan adalah pemberian skor terhadap unit teks tertentu yang dianggap penting dalam sebuah teks (Many & Maybury, 1999), diantaranya:

- *Positional criteria*

Lokasi tertentu pada teks (*heading*, judul, paragraf pertama, dll) cenderung mengandung informasi penting. Metode sederhana dengan mengambil paragraf pertama (*lead*) sebagai ringkasan biasanya cukup bagus terutama pada artikel berita.

- *Cue phrase indicator criteria*

Pada beberapa *genre* teks, kata dan frasa tertentu dalam kalimat secara eksplisit menunjukkan seberapa penting kalimat tersebut. Daftar *cue phrase* beserta (positif dan negatif) '*goodness score*' biasanya dibangun manual.

- *Word and phrase frequency criteria*

Luhn memakai distribusi kata *Zipf's law* untuk mengembangkan kriteria ekstraksi : jika sebuah teks mengandung beberapa kata yang biasanya jarang muncul, maka kalimat-kalimat yang mengandung kata-kata tersebut mungkin penting.

- *Query and title overlap criteria*

Metode sederhana tapi berguna adalah dengan memberi skor pada kalimat-kalimat sesuai jumlah kata-kata yang juga muncul pada judul, *heading*, atau query.

- *Cohesive or lexical connectedness criteria*

Kata-kata dapat dihubungkan dengan berbagai cara, meliputi repetisi, *coreference*, sinonim, dan asosiasi semantik pada thesauri. Kalimat dan paragraf dapat diberi skor berdasarkan derajat keterhubungan kata-katanya; semakin terkoneksi diasumsikan semakin penting.

- *Discourse structure criteria*

Pembuatan struktur *discourse* teks dan memberi skor kalimat berdasarkan *discourse centrality*.

- *Combination of various module scores*

Peneliti menemukan bahwa tidak ada metoda tunggal sebaik ekstraksi oleh manusia; menggabungkannya cukup dapat memperbaiki skor.

Beberapa faktor yang mempengaruhi tingginya pemberian skor pada kandidat kalimat untuk dimasukkan dalam ringkasan (Ganapathiraju. Madhavi K, 2002) adalah :

- *Keyword occurrence*

Kalimat dengan kata kunci yang paling sering digunakan di dokumen biasanya mewakili tema dokumen.

- *Title keyword*

Kata – kata yang muncul pada judul kebanyakan juga merepresentasikan tema dokumen.

- *Indicative Phrases*

Kalimat yang berisi kata kunci kesimpulan, misalnya “laporan ini ..., kesimpulannya adalah ..., dll”, mendapatkan nilai yang tinggi.

- *Location heuristic*

Dalam artikel berita, kalimat pertama cenderung merupakan kalimat inti dari artikel.

- *Short-length cutoff*

Kalimat pendek tidak merepresentasikan isi dokumen dan tidak disertakan dalam ringkasan.

- *Uppercase word feature*

Kalimat yang mengandung nama atau singkatan disertakan dalam ringkasan.

Sedangkan faktor yang mengurangi pemberian skor pada kandidat kalimat yang akan dipilih adalah :

- *Pronouns*

Pronouns atau kata ganti seperti dia, mereka, ia, dll, tidak dapat dimasukkan dalam ringkasan.

- *Redudancy in summary*

Dalam artikel berita yang panjang biasanya terdapat perulangan informasi (redundansi). Hal inilah yang mengurangi pemberian skor dalam *summary*.

2.2 Tipe Evaluasi

Terdapat dua klasifikasi metode evaluasi (Sparck dan Galliers , 1996), yaitu :

- ❑ Ekstrinsik : Kualitas ringkasan diukur berdasarkan bagaimana ini membantu penyelesaian tugas *user*.
- ❑ Intrinsik : Hanya diukur dari kualitas hasil (*output*) ringkasan yang dihasilkan.

Evaluasi sistem peringkas yang ada saat ini adalah intrinsik. Pengevaluasi menciptakan sekumpulan ringkasan yang ideal, masing-masing satu untuk menguji teks. Kemudian membandingkan hasil ringkasan sistem dengan itu. Yang diukur adalah *overlap* dari isi ringkasan,seringkali disebut dengan *recall* dan *precision* kalimat atau frase, tapi kadang-kadang dengan *overlap* kata tunggal.

$$Precision = \frac{\# \text{kalimat ringkasan sistem} \cap \text{ringkasan ideal}}{\# \text{kalimat ringkasan sistem}} \quad (2.1)$$

$$Recall = \frac{\# \text{kalimat ringkasan sistem} \cap \text{ringkasan ideal}}{\# \text{kalimat ringkasan ideal}} \quad (2.2)$$

Kombinasi antara nilai precision dan recall menghasilkan *f-measure*.

$$f\text{-measure}(\beta) = \frac{(1+\beta^2) * \text{precision} * \text{recall}}{\beta^2 * (\text{precision} + \text{recall})} \quad (2.3)$$

Precision dan *recall* yang dinyatakan dalam persamaan 2.1 dan persamaan 2.2 digunakan untuk mengukur kualitas sebuah ringkasan. Nilai *precision* menunjukkan bahwa sebuah ringkasan Pengukuran *precision* dan *recall* ini sangat dipengaruhi oleh panjang ringkasan ideal dan juga panjang ringkasan yang akan dievaluasi. Nilai *f-measure* yang dinyatakan dalam persamaan 2.3 merupakan kombinasi antara nilai *precision* dan *recall*. Koefisien β pada persamaan *f-measure* digunakan untuk menghitung nilai rasio dari rata-rata *precision* dan *recall*. Nilai koefisien β terdapat pada range ($0 \leq \beta \leq \infty$), yaitu apabila nilai $\beta = 0$ maka *f-measure* (β) = *precision*, nilai $\beta = \infty$ maka *f-measure* (β) = *recall*, dan jika nilai $\beta = 1$ maka akan diperoleh nilai *f-measure* (β) yang mengindikasikan bahwa nilai *recall* dan *precision* mempunyai nilai yang sama penting dalam dokumen. Apabila nilai $\beta < 1$ maka nilai *f-measure* (β) akan ditekankan pada nilai *precision* yang lebih penting, apabila nilai $\beta > 1$ maka nilai *f-measure* (β) akan ditekankan pada nilai *recall* yang lebih penting. Pada penelitian kali ini digunakan nilai $\beta < 1$ karena pada penelitian ini nilai evaluasi lebih ditekankan pada perhitungan *precision* dengan asumsi semakin baik nilai *precision* maka kualitas ringkasan juga semakin baik (wikipedia, 2008).

2.3 Proses Text Sumarization

Secara umum terdapat tiga tahapan dalam *Text Summarization*, yaitu :

1. *Topic Identification*

Langkah ini meliputi identifikasi faktor yang sangat penting tentang apa yang dibicarakan dalam teks tersebut. Terdapat beberapa teknik untuk melakukannya, diantaranya:

- Dalam beberapa tipe teks, informasi penting terdapat dalam bagian-bagian tertentu dalam teks tersebut, seperti dalam judul, kalimat pertama, kalimat terakhir dan lain sebagainya.
- Beberapa kata atau frase mengidentifikasi intisari dari suatu teks, contohnya dalam bahasa Inggris : *in summary*, *in conclusion*, *this paper*, dan lain sebagainya.
- Beberapa kata, tergantung isi dari teks tersebut cenderung muncul lebih sering. Ini dijadikan faktor penentu topik dari suatu teks (*word frequency*)
- Ada juga topik yang diidentifikasi dari jumlah pengertiannya/makna dibandingkan kata.

2. Interpretation

Interpretasi peringkas yang ekstraktif berdasarkan pada metode yang digunakan, sedangkan pada peringkas yang abstraktif interpretasi ditunjukkan dengan cara penggabungan pengertian yang serupa menjadi satu, penghilangan redundansi, dan lain sebagainya.

3. Generating

Langkah terakhir adalah pembangkitan atau pembentukan hasil akhir. Terdiri dari penggabungan frase, pencetakan kata atau frase dan pembangkitan kalimat. Terdapat beberapa metode yang dapat digunakan, diantaranya :

- *Extraction*, hasil akhir berisi kalimat atau frase yang dihasilkan setelah semua tahap proses pada *Text Summarization* selesai dilakukan.
- *Topic List*, hasil akhir berisi kata-kata yang sering muncul atau penggabungan pengertian yang telah diinterpretasi.
- *Phrase Concatenation*, dua atau beberapa frase yang memiliki pengertian yang serupa digabungkan.

Sentence Generation, hasilnya adalah kalimat baru, dimana inputnya adalah pengertian-pengertian yang telah digabungkan yang berhubungan dengan topik tersebut.

2.4 Text Pre-processing

Untuk dapat memproses suatu dokumen teks diperlukan beberapa tahapan, yaitu *case folding*, *parsing*, dan penghilangan *stopword*.

2.4.1 Case Folding

Dalam tahapan ini dilakukan semua huruf dalam dokumen diubah menjadi huruf kecil. Hanya huruf 'a' sampai 'z' yang diterima, sedangkan karakter selain huruf dihilangkan dan dianggap sebagai token, yaitu karakter dasar yang sudah tidak dapat diturunkan lagi.

2.4.2 Parsing

Parsing merupakan proses penguraian kalimat menjadi kata-kata dengan menggunakan analisis sintaks atau tata bahasa. Bahasa Indonesia memiliki aturan dalam pembentukan frase maupun kalimat. Aturan inilah yang menjadi acuan dalam proses *parsing*.

2.4.3 Penghilangan Stopword

Stopword adalah kata-kata yang seringkali muncul dalam dokumen namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Kata-kata yang termasuk dalam *stopword* diantaranya artikel, preposisi, dan kata hubung. Misalnya 'dan',

‘sebenarnya’, ‘sebuah’, dan seterusnya. Jumlah *stopword* sangat banyak hingga mencapai ratusan, dapat dilihat pada lampiran.

2.5 Cosine

Salah satu perhitungan kesamaan antar kalimat yang sering digunakan adalah cosine similarity. Pada pendekatan metode ini, setiap dokumen (atau pada kasus ini setiap kalimat) direpresentasikan menggunakan sebuah model vektor jarak (*vector space model*). Cosine similarity telah banyak digunakan untuk menghitung similarity dokumen dengan query yang diberikan, yang mana setiap kalimat dalam dokumen direpresentasikan menggunakan bobot term.

Model ruang vektor dilakukan untuk melihat nilai kemiripan suatu dokumen terhadap suatu query. Pembobotan secara otomatis biasanya berdasarkan jumlah kemunculan suatu istilah dalam sebuah dokumen (*term frequency/tf*) dan jumlah kemunculannya dalam koleksi dokumen (*inverse document frequency/idf*). Bobot suatu istilah semakin besar jika istilah tersebut sering muncul dalam suatu dokumen dan semakin kecil jika istilah tersebut muncul dalam banyak dokumen (Grossman, 1998). Nilai **idf** sebuah kata/term dapat dihitung menggunakan persamaan 2.4.

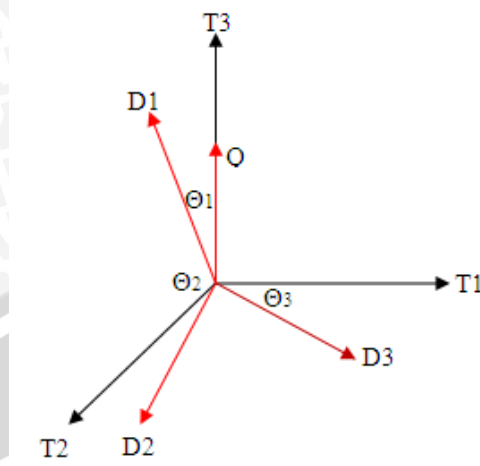
$$\mathbf{IDF} = \mathbf{Log (D/dfi)} \dots\dots (2.4)$$

Dimana **D** adalah jumlah dokumen yang berisi term (t) dan **dfi** adalah jumlah kemunculan/frekuensi kata terhadap **D**.

Pada penemuan sebelumnya, telah dibandingkan penghitungan nilai term yang berbeda sebagai pendekatan perangkingan dari kalimat yang penting dalam sebuah dokumen dan hasilnya menunjukkan bahwa perhitungan nilai **tf-idf** hasilnya sangat kuat (Murray & Renal, 2007).

Dalam penelitian kali ini perhitungan Cosine Similarity digunakan untuk menghitung pendekatan relevansi kalimat query dengan dokumen asal maupun dokumen ringkasan pembanding. Penentuan relevansi sebuah query kalimat pada dokumen dipandang sebagai pengukuran kesamaan (similarity measure) antara vektor query dengan vektor dokumen. Semakin sama nilai vektor query dengan vektor dokumen maka query kalimat tersebut dipandang semakin relevan dengan dokumen. Salah satu pengukuran kesesuaian yang baik adalah dengan memperhatikan perbedaan arah (*direction different*) dari kedua vektor tersebut. Perbedaan arah kedua vektor dalam geometri dianggap sebagai sudut yang terbentuk oleh

kedua vektor tersebut. Hubungan antara vektor query dengan vektor dokumen direpresentasikan pada gambar 2.1.



Gambar 2.1 Skalar Vektor (Scalar Product)

Pada gambar 2.1 dapat diketahui bahwa terdapat 3 buah vektor dokumen dan sebuah vektor query dengan masing – masing koordinat :

$$D1 : 2T1 + 3T2 + 5T3$$

$$D2 : 3T1 + 7T2 + 0T3$$

$$D3 : 3T1 + 3T2 + 1T3$$

$$Q : 0T1 + 0T2 + 2T3$$

Dari data pada Gambar 2.1 dapat dihitung jarak antar term, dimana setiap term merepresentasikan query sebagai sebuah titik dengan koordinat (0,0,2). Tabel 2.1 merupakan tabel representasi titik koordinat setiap term dari dokumen diatas :

Tabel 2.1 Tabel sebaran vektor kata

Index Term	Collection			Query = T3
	Term Counts, tf			Query Counts
Term Space	D1	D2	D3	Q
T1	2	3	3	0
T2	3	7	3	0
T3	5	0	1	2
Coordinate	(2,3,5)	(3,7,0)	(3,3,1)	(0,0,2)
Magnitude	D1	D2	D3	Q

- Kolom 1 mendeskripsikan jarak term (term space). Pada data diatas terdiri dari 3 dimensi : auto, car, insurance.
- Kolom 2, 3 dan 4 mendeskripsikan perhitungan term (term count). Perhitungan ini berisi koordinat – koordinat setiap point pada term space yang berhubungan pada setiap dokumen. Pada kasus diatas, koordinate setiap point adalah (2,3,5), (3,7,0) and (3,3,1). Jika koordinat asal adalah (0,0,0) maka pergantian setiap point dari koordinat asal dapat direpresentasikan sebagai sebuah vektor. Jarak (magnitude) dari vektor ini dapat dihitung menggunakan rumus Pithagoras.
- Kolom 4 mendeskripsikan koordinat dari query, yang pada kasus ini adalah (0,0,2).

Sudut θ_1 , θ_2 dan θ_3 menggambarkan kesamaan dokumen D1, D2 dan D3 dengan Q. Apabila Q adalah vektor query dan D adalah vektor dokumen yang merupakan 3 buah vektor dalam ruang berdimensi –n dan θ adalah sudut yang dibentuk oleh kedua vektor tersebut, maka :

$$Q \cdot D = |Q| \cdot |D| \text{Cos}\theta \dots\dots (2.5)$$

Dimana $Q \cdot D$ adalah hasil perkalian dalam (*inner product/scalar product/dot product*) kedua vektor sedangkan $|Q|$ dan $|D|$ merupakan panjang vektor (jarak euclidean) suatu vektor dengan titik nol. *Dot product* (scalar product / inner product) adalah perhitungan nilai koordinat sebuah term pada sebuah dokumen atau pada kasus ini adalah kalimat (DR.E Gracia, 2004). Nilai $|Q|$ dan $|D|$ dapat dihitung menggunakan persamaan 2.6 dan persamaan 2.7.

$$|D| = \sum_{i=1}^n D_i^2 \dots\dots (2.6)$$

$$|Q| = \sum_{i=1}^n Q_i^2 \dots\dots (2.7)$$

Berdasarkan persamaan 2.5, persamaan 2.6 dan persamaan 2.7 maka rumus kesamaan (*similarity*) antara dua vektor adalah

$$\text{Sim}(Q,D) = \text{Cos}(Q,D) = \frac{Q \cdot D}{|Q| \cdot |D|} \dots\dots (2.8)$$

Contoh perhitungan Cosine Similiarity :

Berdasarkan contoh tabel data dari Dr. Grossman, diberikan sebuah dokumen dengan kalimat pada masing – masing paragraf adalah :



- Dokumen 1 (D1) : Shipment of gold damaged in a fire
- Dokumen 2 (D2) : Delivery of silver arrived in a silver truck
- Dokumen 3 (D3) : Shipment of gold arrived in a truck

Sedangkan query yang diberikan adalah :

Query (Q) : Gold silver truck

Dari data yang diberikan tersebut, maka data dapat direpresentasikan dalam tabel 2.2.

Tabel 2.2 Tabel perhitungan vektor kata

TERM VECTOR MODEL BASED ON $w_i = tf_i * IDF_i$												
Query, Q: "gold silver truck"												
D ₁ : "Shipment of gold damaged in a fire"												
D ₂ : "Delivery of silver arrived in a silver truck"												
D ₃ : "Shipment of gold arrived in a truck"												
D = 3; IDF = log(D/df _i)												
	Counts, tf_i							Weights, $w_i = tf_i * IDF_i$				
Terms	Q	D ₁	D ₂	D ₃	df _i	D/df _i	IDF _i	Q	D ₁	D ₂	D ₃	
a	0	1	1	1	3	3/3 = 1	0	0	0	0	0	
arrived	0	0	1	1	2	3/2 = 1.5	0.1761	0	0	0.1761	0.1761	
damaged	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0	
delivery	0	0	1	0	1	3/1 = 3	0.4771	0	0	0.4771	0	
fire	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0	
gold	1	1	0	1	2	3/2 = 1.5	0.1761	0.1761	0.1761	0	0.1761	
in	0	1	1	1	3	3/3 = 1	0	0	0	0	0	
of	0	1	1	1	3	3/3 = 1	0	0	0	0	0	
silver	1	0	2	0	1	3/1 = 3	0.4771	0.4771	0	0.9542	0	
shipment	0	1	0	1	2	3/2 = 1.5	0.1761	0	0.1761	0	0.1761	
truck	1	0	1	1	2	3/2 = 1.5	0.1761	0.1761	0	0.1761	0.1761	

$$|D1| = \sqrt{0.4771^2 + 0.4771^2 + 0.1761^2 + 0.1761^2}$$

$$= 0.7192$$

$$|D2| = \sqrt{0.1761^2 + 0.4771^2 + 0.9542^2 + 0.1761^2}$$

$$= 1.0955$$

$$|D3| = \sqrt{0.1761^2 + 0.1761^2 + 0.1761^2 + 0.1761^2}$$

$$= 0.3522$$

$$|Q| = \sqrt{0.1761^2 + 0.4771^2 + 0.1761^2} = 0.5382$$

Kemudian dihitung semua nilai dot product yang terjadi :

$$Q \cdot D1 = 0.1761 * 0.1761 = 0.0310$$

$$Q \cdot D2 = 0.4771 * 0.9542 + 0.1761 * 0.1761 = 0.4862$$

$$Q \cdot D3 = 0.1761 * 0.1761 + 0.1761 * 0.1761 = 0.0620$$

Langkah terakhir dihitung nilai kesamaannya (similarity) :

$$\cosine\theta_{D1} = \frac{Q \cdot D1}{|Q| * |D1|} = \frac{0.0310}{0.5382 * 0.7192} = 0.0801$$

$$\cosine\theta_{D2} = \frac{Q \cdot D2}{|Q| * |D2|} = \frac{0.4862}{0.5382 * 1.0955} = 0.8246$$

$$\cosine\theta_{D3} = \frac{Q \cdot D3}{|Q| * |D3|} = \frac{0.0620}{0.5382 * 0.3522} = 0.3271$$

Dari perhitungan nilai cosine similarity tersebut kemudian diurutkan kebawah (descending), sehingga didapatkan :

Rank 1 : Dokumen 3 = 0.3271

Rank 2 : Dokumen 2 = 0.0801

Rank 3 : Dokumen 1 = 0.0801

Maka dapat diketahui bahwa paragraf yang mempunyai relevansi terbesar terhadap query (Q) adalah dokumen 2.

2.6 Metode *MMR-D* (Maximum Marginal Relevansi - Dokumen)

Metode *MMR-D* (Maximum Marginal Relevansi - Dokumen) merupakan salah satu metode extractive sumary, digunakan dalam peringkasan dokumen tunggal maupun banyak dokumen, mempunyai karakter yang efektif dan sederhana. Pada metode ini, fungsi *MMR-D* digunakan untuk menghitung kesamaan (*similarity*) antara bagian teks.

Dalam kasus peringkasan dokumen dengan metode *MMR-D* ini, dilakukan segmentasi dokumen menjadi kalimat dan dilakukan pengelompokan sesuai dengan gender kalimat tersebut. *MMR-D* digunakan dengan mengkombinasikannya dengan matrik *cosine similarity* untuk merangking kalimat – kalimat sebagai tanggapan terhadap query yang diberikan oleh user maupun sistem. Peringkat paling atas adalah petikan yang dipaparkan oleh dokumen asli.

Kebanyakan mesin pencarian IR modern menghasilkan daftar perangkingan dari dokumen yang diukur dari penurunan relevansi terhadap permintaan user (user's query). Penaksiran pertama untuk mengukur hasil peringkasan yang relevant adalah mengukur hubungan antar informasi yang ada dalam dokumen dengan query yang diberikan oleh user dan menambah kombinasi linear sebagai sebuah matrik. Dalam kasus ini kombinasi linear disebut sebagai "*marginal relevance*" (Carbonell, Jaime & Jade Goldstein, 1998).

Sebuah dokumen dikatakan mempunyai *marginal relevance* yang tinggi, apabila dokumen tersebut relevan terhadap isi dari dokumen dan mempunyai kesamaan bobot term yang maximum dibandingkan dengan dokumen yang dipilih sebelumnya. Pada teknik *MMR-D* digunakan dokumen latih sebagai pembanding similarity kata yang relevan dengan dokumen asal.

Pada peringkasan dengan cara ekstraktif, nilai akhir diberikan pada kalimat S_i dalam MMR dihitung dengan persamaan berikut :

$$\text{MMR} = \max [\lambda \times \max \text{Sim1} (S_i, D) - ((1 - \lambda) \times \max \text{Sim2} (S_i, \text{Summ}))] \quad (2.9)$$

Dimana D adalah dokumen asal keseluruhan sedangkan Summ direpresentasikan sebagai kumpulan kalimat yang telah diekstrak ke dalam ringkasan. λ digunakan untuk mengatur kombinasi nilai untuk memberi penekanan bahwa kalimat tersebut relevan dan untuk mengurangi redundansi. Dalam kasus ini, Sim1 dan Sim2 merupakan dua fungsi similarity untuk merepresentasikan kesamaan kalimat pada seluruh dokumen dan memilih masing-masing kalimat untuk dijadikan ringkasan. Sim1 adalah similarity kalimat S_i terhadap dokumen asal dan Sim2 adalah similarity kalimat S_i terhadap ringkasan.

Pada metode ini, nilai parameter λ adalah 1 atau 0 atau diantaranya (range $[0,1]$). Ketika parameter $\lambda=1$ maka nilai MMR yang diperoleh akan cenderung relevan terhadap dokumen asal. Ketika $\lambda=0$ maka nilai MMR akan cenderung relevan terhadap dokumen hasil ringkasan yang masih akan dibandingkan. Oleh karena itulah, sebuah kombinasi linear dari kedua kriteria dioptimalkan ketika nilai λ terdapat pada interval $[0,1]$.

Pengguna yang menginginkan informasi ruang sampel disekitar query, harus menetapkan λ pada nilai yang lebih rendah. Sedangkan bagi pengguna yang ingin fokus untuk memperkuat dokumen-dokumen yang agar lebih relevan, harus menetapkan λ ke nilai lebih dekat dengan 1. Kalimat dengan nilai MMR tertinggi akan berulang kali dipilih kedalam ringkasan sampai tercapai ukuran ringkasan yang diinginkan.

2.7 Pengertian dan Penulisan Kata

Kata adalah kesatuan terkecil yang diperoleh sesudah sebuah kalimat dipecah, dan mengandung suatu ide. Suatu bentuk yang memiliki arti atau sering disebut morfem sudah merupakan suatu kata, tetapi konsep tentang kata tidak hanya meliputi morfem bebas

melainkan juga semua bentuk gabungan antara morfem bebas dan morfem terikat, atau morfem dasar dengan morfem dasar.

2.7.1 Kata Dasar

Kata yang berupa kata dasar ditulis sebagai suatu kesatuan. Misalnya :

- Ibu percaya bahwa engkau tahu
- Kantor pajak penuh sesak
- Buku itu sangat tebal

2.7.2 Kata Turunan

- Imbuhan (awalan, sisipan, akhiran) ditulis serangkaian dengan kata dasarnya. Misalnya : bergetar, penetapan, menengok, mempermainkan, dll.
- Awalan (*perfixs*) adalah suatu unsur struktural yang dikaitkan didepan sebuah kata dasar atau bentuk dasar. Yang dimaksud disini adalah hubungan antar awalan dan kata dasar adalah hubungan struktural, yaitu bahwa semua unsur (awalan dan kata dasar) merupakan bagian dari kata turunan (kata yang berawalan). Yang dimaksud didalam awalan adalah ber-, me-, pe-, di-, ke-, ter-, dan se-
- Sisipan (infiks) adalah semacam morfem terikat yang disisipkan pada sebuah kata antara konsonan pertama dengan vokal pertama. Jenis morfem ini pemakaiannya terbatas pada beberapa kata saja. Sisipan yang terdapat dalam bahasa Indonesia adalah -el-, -er-, dan -em-.
- Akhiran (sufiks) adalah semacam morfem terikat yang diletakkan dibelakang suatu morfem dasar. Macam – macam akhiran yang terdapat dalam bahasa Indonesia adalah akhiran -an, -kan, dan -i.
- Jika bentuk dasar berupa gabungan kata, awalan atau akhiran ditulis serangkai dengan kata yang langsung mengikuti atau mendahuluinya. Misalnya : bertepuk tangan, garis bawahi, menganak sungai, sebar luaskan.
- Jika bentuk dasar yang berupa gabungan kata mendapat awalan dan akhiran sekaligus, unsur gabungan kata ditulis serangkai. Misalnya : menggarisbawahi, menyebarluaskan, dilipatgandakan, penghancurleburan.
- Jika salah satu unsur gabungan kata hanya dipakai dalam kombinasi, gabungan kata itu ditulis serangkai. Misalnya : adipati, biokimia, dasawarsa, mahasiswa.

2.7.3 Bentuk Ulang

Bentuk ulang ditulis secara lengkap dengan menggunakan tanda hubung. Misalnya : anak-anak, gerak-gerik, dll.

2.7.4 Gabungan Kata

- Gabungan kata, termasuk istilah khusus, yang mungkin menimbulkan kesalahan pengertian dapat ditulis dengan tanda hubung untuk menegaskan pertalian unsur yang bersangkutan. Misalnya : alat pandang-dengar, anak-istri saya, dll.
- Gabungan kata berikut ditulis serangkai. Misalnya : acapkali, kacamata, bilamana, sediakala, segitiga, sukarela, dll.

2.8 Pengertian Kalimat

Kalimat adalah satuan bahasa terkecil, dalam wujud lisan atau tulisan, yang mengungkapkan pikiran yang utuh. Bentuk kalimat yang paling sederhana baik harus memiliki subyek (S) dan predikat (P). Apabila tidak memiliki unsur subyek dan predikat, maka deretan kata tersebut disebut dengan frasa.

Dalam wujud lisan kalimat diucapkan dengan suara naik turun, dan keras lembut, disela jeda, dan diakhiri dengan intonasi akhir. Dalam wujud tulisan kalimat dimulai dengan huruf kapital dan diakhiri dengan tanda titik (.), tanda tanya (?), dan tanda seru (!).

Menurut fungsinya, jenis kalimat dapat dirinci menjadi kalimat pernyataan, kalimat pertanyaan, kalimat perintah, dan kalimat seruan. Semua jenis kalimat itu disajikan dalam bentuk positif dan negatif. Dalam bahasa lisan, intonasi yang khas menjeaskankapan berhadapan dengan salah satu jenis itu. Dalam bahasa tulisan, perbedaannya dijelaskan oleh bermacam-macam tanda baca.

BAB III

METODOLOGI DAN PERANCANGAN

Pada bab metodologi dan perancangan ini akan dibahas metode, rancangan yang digunakan dan langkah – langkah yang dilakukan dalam penelitian peringkasan dokumen menggunakan metode ekstraksi kalimat dengan pendekatan *maximum marginal relevance document*.

Penelitian dilakukan dengan tahapan – tahapan berikut ini :

1. Mempelajari metode yang akan digunakan pada sistem peringkasan ini, yaitu metode ekstraksi kalimat dengan pendekatan *maximum marginal relevance document* (MMR-D).
2. Menganalisa dan merancang perangkat lunak peringkasan dokumen dengan menggunakan pendekatan *maximum marginal relevance document* (MMR-D).
3. Membuat perangkat lunak berdasarkan analisis dan perancangan yang telah dilakukan.
4. Ujicoba perangkat lunak peringkasan dokumen, dengan menggunakan data teks berita.
5. Evaluasi hasil ringkasan yang dibuat oleh sistem berdasarkan kemunculan kalimat dalam ringkasan sistem dengan kalimat yang muncul pada sistem peringkasan lainnya. Dalam hal ini, sistem pembandingan yang digunakan adalah sistem peringkasan yang menggunakan gabungan metode *graph* algoritma *shortest path*. Yang akan dibandingkan adalah besar *f-measure* yang merupakan gabungan dari *recall* dan *precision*, untuk ukuran ringkasan tertentu.

3.1 Analisis Sistem

Pada subbab analisis sistem ini akan dibahas berbagai hasil analisis terhadap sistem dan elemen – elemen yang terkait dan semua hal yang dibutuhkan dalam proses peringkasan dokumen.

3.1.1 Deskripsi Umum Sistem

Peringkasan dokumen yang akan dibuat merupakan sistem yang membaca sebuah inputan berupa teks dokumen yang secara otomatis menghasilkan sebuah ringkasan. Metode yang digunakan pada sistem ini adalah ekstraksi dengan pendekatan *Maximum Marginal Relevance (MMR)* document.

Peringkasan dibuat berdasarkan ekstraksi dokumen, dimana dokumen dipecah sampai dengan unit terkecil (kata). Dalam pendekatan MMR-D ini dibutuhkan 2 kelompok data, yaitu

data *test* dan data *training*. Data test merupakan teks dokumen awal (original document) yang akan dicari ringkasannya, sedangkan data training merupakan teks ringkasan latihan yang akan dijadikan pembandingan data test sehingga dapat dibuat ringkasannya. Ketika sebuah teks dokumen diringkas, proses yang dilakukan adalah :

- User memasukkan teks dokumen yang akan diringkas.
- Sistem akan men-generate teks ringkasan latihan yang akan digunakan sebagai data training. Teks ringkasan latihan ini diperoleh secara random. Dalam penelitian ini, sistem akan men-generate satu ringkasan latihan baru.
- Kemudian oleh sistem, dokumen tersebut diparsing menjadi kalimat – kalimat dan kata – kata.
- Untuk setiap kalimat pada dokumen test, dihitung similarity-nya terhadap keseluruhan dokumen dan terhadap dokumen ringkasan latihan.
- Hasil perhitungan inilah yang selanjutnya digunakan untuk menghitung relevansi perkalimat dalam dokumen terhadap dokumen itu sendiri dan terhadap dokumen training (dalam hal ini dokumen ringkasan latihan).
- Kalimat – kalimat dengan nilai relevansi yang tertinggi lah yang akan diambil sebagai ringkasan.

3.1.2 Batasan Sistem

Batasan dari sistem yang akan dikembangkan adalah :

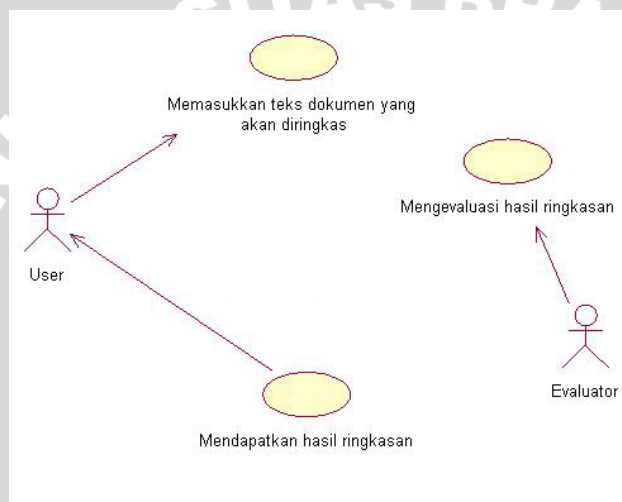
1. Dalam sistem ini hanya menggunakan satu data training (ringkasan latihan) sebagai data pembandingnya.
2. Proses penghilangan *stopword* (kata umum) tergantung pada bahasa yang digunakan, dalam sistem ini adalah menggunakan bahasa Indonesia.
3. Dalam sistem ini tidak terdapat proses *stemming*.
4. Faktor judul dan *heading* (bab dan subbab) dalam sistem ini tidak dipertimbangkan.
5. Ukuran pemotongan kalimat – kalimat berdasarkan nilai MMR ditentukan oleh user.
6. Jumlah iterasi hasil ringkasan ditentukan oleh user.
7. Nilai koefisien λ dalam sistem ini sudah ditentukan yaitu 0,6.
8. Nilai koefisien β untuk perhitungan evaluasi f-measure (β) pada penelitian ini sudah ditentukan yaitu 0,9.
9. Nilai threshold pengambilan kalimat ditentukan oleh user. Nilai threshold ini mempengaruhi jumlah kalimat dalam ringkasan.

3.1.3 Analisis Kebutuhan Perangkat Lunak

3.1.3.1 Analisis Use Case

Dalam sistem peringkas dokumen ini pengguna yang akan mendapatkan ringkasan dari suatu teks, dapat memasukkan teks dokumen pada area yang telah disediakan. dalam sistem ini dokumen teks sudah disimpan dalam komputer, dengan format *.txt. Dengan menekan tombol ringkas, maka akan muncul ringkasan pada area yang ditentukan. Sedangkan evaluasi ringkasan yang dihasilkan sistem dibutuhkan seorang evaluator.

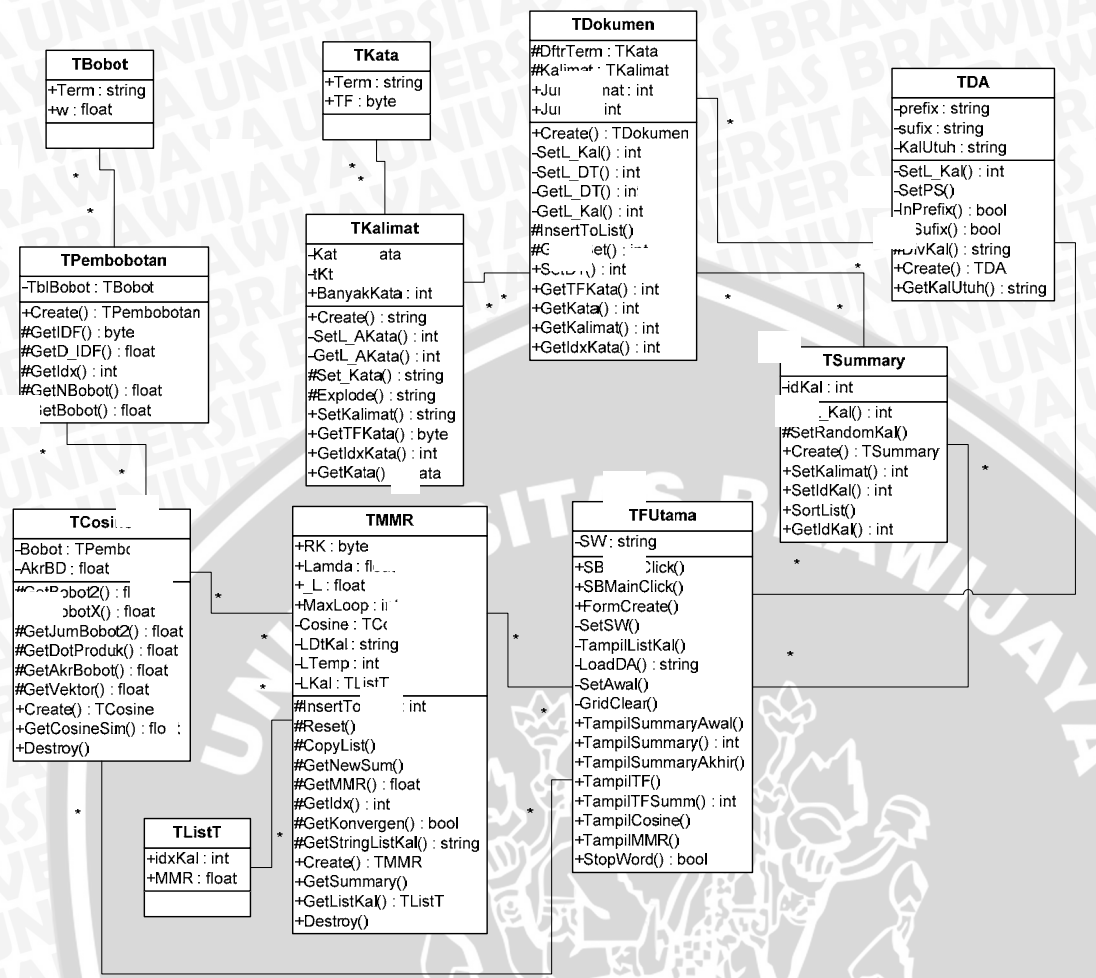
Gambaran sistem dari sisi pengguna dinyatakan pada gambar 3.1 :



Gambar 3.1 Diagram *Use Case*

3.1.3.2 Analisis Kelas

Kelas – kelas yang diperlukan dalam pembuatan perangkat lunak peringkas dokumen otomatis ini dinyatakan pada gambar 3.2 :



Gambar 3.2 Diagram Kelas

Kelas TFUtama

Kelas ini merupakan kelas interface yang berfungsi untuk memproses tampilan semua object yang berhubungan dengan sistem peringkasan otomatis ini, yaitu menampilkan hasil akhir kalimat ringkasan, untuk memasukkan data dokumen asal, untuk memasukkan variabel-variabel yang diperlukan (variabel rasio kesimpulan, maksimal looping dan variabel lamda), untuk menampilkan hasil proses perhitungan yang dilakukan (perhitungan frekuensi kata, cosine dan MMR). Di dalam kelas TFUtama dilakukan proses penghilangan stopword (kata-kata yang tidak penting) dalam kalimat setelah dilakukan *parsing*. Pada kelas TFUtama digunakan pemanggilan terhadap kelas TMMR, kelas TDokumen, kelas TCosine, kelas TDA dan kelas TSumm.

TCosine

Kelas ini digunakan untuk menghitung nilai kesamaan (*similarity*) perkaliat terhadap dokumen asal dan dokumen ringkasan latih. Pada kelas TCosine dilakukan pemanggilan terhadap kelas TPembobotan. Hasil perhitungan dalam kelas Tcosine selanjutnya disimpan dalam sebuah *array*. Kelas ini digunakan pada kelas TMMR dan kelas TFUtama.

TPembobotan

Pada kelas TPembobotan dilakukan proses perhitungan bobot kata perkaliat dan perdokumen (DA dan RL) dan dilakukan perhitungan IDF yang selanjutnya digunakan untuk proses perhitungan cosine (*similarity*) kalimat. Kelas ini digunakan pada kelas Tcosine. Hasil perhitungan dalam kelas TPembobotan selanjutnya disimpan dalam sebuah *array*.

TMMR

Kelas TMMR digunakan untuk perhitungan hasil MMR (relevansi) perkaliat terhadap dokumen asal (DA) dan ringkasan latih (RL). Pada kelas ini, terdapat fungsi *GetKonvergen* yang digunakan untuk mendapatkan kalimat ringkasan akhir. Pada kelas TMMR dilakukan pemanggilan terhadap kelas TCosine dan TSummary. Pada kelas ini juga dilakukan pemanggilan terhadap kelas TListT yang merupakan *record* untuk menyimpan hasil perhitungan MMR.

TDA

Kelas TDA merupakan turunan dari kelas TDokumen. Kelas ini berfungsi untuk memecah (*parsing*) dokumen asal menjadi kalimat dan melakukan proses pengecekan suffix dan prefix yang ada dalam dokumen. Hasil akhir dari kalimat pecahan akan disimpan dalam sebuah *array*. Pada kelas TDA dilakukan pemanggilan terhadap kelas TKalimat.

TDokumen

Kelas TDokumen berfungsi untuk memberikan indeks kalimat dan indeks kata pada dokumen. Pada kelas ini dilakukan pemanggilan terhadap kelas TKalimat. Kelas TDokumen merupakan kelas induk dari kelas TDA dan Tsummary.

TKalimat

Pada kelas TKalimat dilakukan proses pemecahan (*parsing*) kalimat menjadi kata (*term*). Setelah dipecah dilakukan perhitungan frekuensi kemunculan tiap kata tersebut pada sebuah kalimat. Pada kelas ini juga dilakukan pemanggilan terhadap kelas TKata yang merupakan *record* untuk menyimpan hasil pemecahan kata dan frekuensi kemunculannya.

TSummary

Kelas TSummary merupakan kelas turunan dari kelas TDokumen. Pada kelas ini dilakukan pembentukan kalimat *summary* (ringkasan latih) yang selanjutnya kalimat *summary* ini akan digunakan pada perhitungan cosine dan MMR. Pada kelas TSummary terdapat procedure *SetRandomKal* yang berfungsi untuk memilih kalimat yang digunakan sebagai kalimat *summary* latih awal.

3.2 Perancangan Sistem

Berdasarkan analisis yang telah dilakukan, berikut ini akan dibahas proses dan arsitektur sistem peringkas dokumen yang akan dibangun ini.

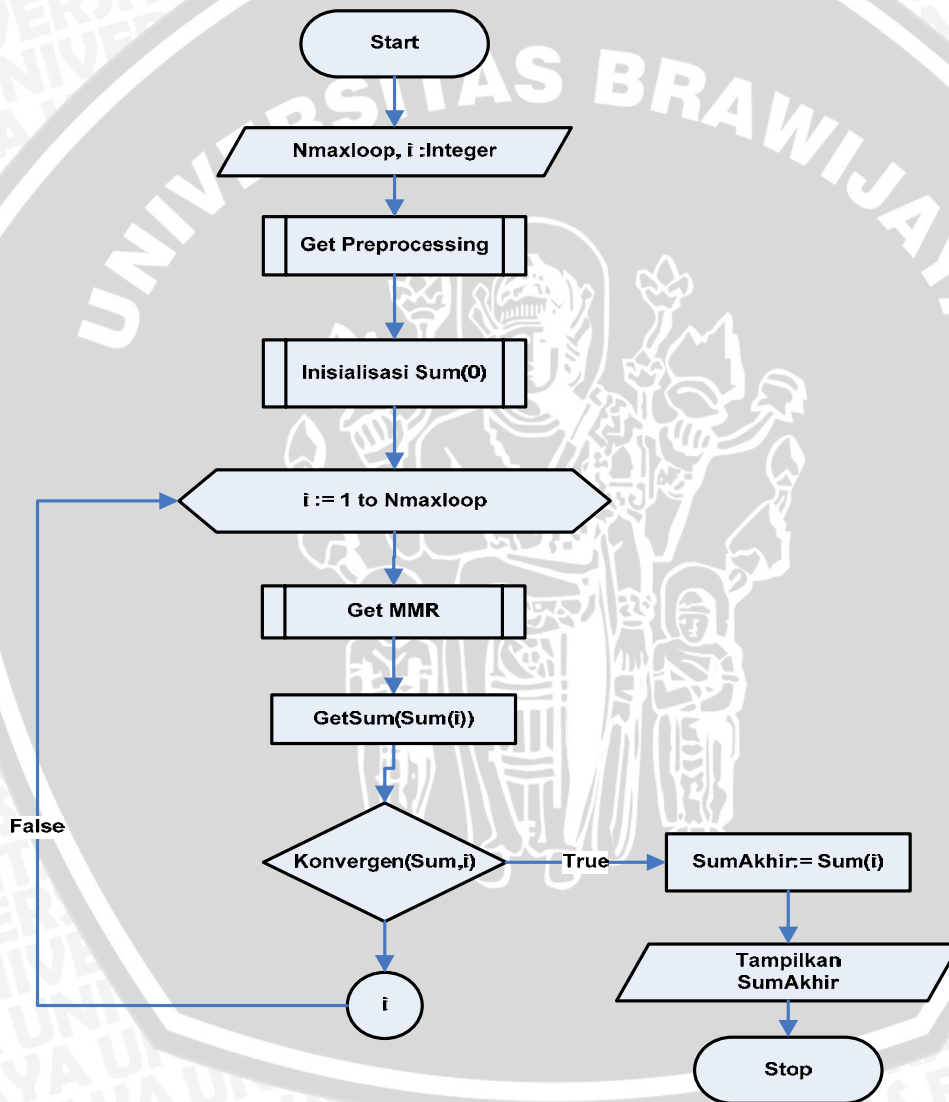
3.2.1 Perancangan Proses

Teks yang dimasukkan pada sistem ini merupakan teks artikel berita berbahasa indonesia. Struktur dari teks berita terdiri dari judul dan paragraf – paragraf. Meskipun demikian, pada penelitian kali ini hanya menggunakan kalimat dan kata - kata pada dokumen teks yang digunakan sebagai bahan penelitian, fitur judul dan paragraf tidak dipertimbangkan.

Dalam penelitian ini, pada awalnya teks dokumen akan dipecah menjadi kalimat dan kata pada proses *Get Preprocessing*. Selanjutnya akan dilakukan proses *Inisialisasi summary* yang mana pada proses ini dilakukan pembentukan *summary* latih yang akan digunakan untuk perhitungan proses selanjutnya. Pemilihan kalimat *summary* latih dilakukan secara random sesuai dengan threshold yang diberikan oleh user. Setelah kalimat-kalimat *summary* latih terpilih selanjutnya dilakukan proses perhitungan similiarity per kalimat (*Si*) terhadap dokumen asal dan terhadap dokumen ringkasan latih. Perhitungan ini bertujuan untuk mencari relevansi kalimat yang paling tinggi terhadap dokumen asal dan ringkasan latih, sehingga dapat diraih sebuah ringkasan, dalam penelitian ini ditunjukkan pada proses *GetMMR* dan *GetSum*. Setelah proses proses *GetMMR* dan *GetSum* selesai dilakukan selanjutnya dilakukan pengecekan apakah kalimat ringkasan (*Summ(i)*) yang diperoleh sudah konvergen. Jika benar

maka kalimat ringkasan akhir (SummAkhir(i)) adalah ringkasan itu sendiri (Summ(i)). Jika salah maka proses perhitungan relevansi kalimat akan terus dilakukan sampai batas iterasi maksimal (NMaxLoop) yang telah ditentukan Batas konvergen dalam penelitian ini yaitu apabila kalimat-kalimat pembentuk summary pada iterasi ke-i keluar lagi pada iterasi-iterasi sebelumnya.

Alur proses pembentukan ringkasan ditunjukkan pada Gambar 3.3 :



Gambar 3.3 Alur proses pembentukan ringkasan

3.2.1.1 Get Preprocessing

Pada proses ini dilakukan pemecahan teks dokumen menjadi kalimat - kalimat dan kata - kata. Setiap kalimat pada teks dokumen diberikan nomor kalimat. Dalam hal ini dokumen teks dimasukkan oleh user.

Pemecahan kalimat

Untuk memecah (parsing) teks yang masuk menjadi kalimat digunakan beberapa batas penentu sebuah kalimat (*sentence delimiter*). Beberapa hal berikut yang bisa digunakan untuk menentukan akhir dari sebuah kalimat :

- Sebuah kalimat diakhiri dengan 1 atau lebih titik (.), tanda seru (!), tanda tanya (?).
- *Sentence delimiter* biasanya diikuti oleh tanda petik dua, seperti pada contoh berikut :
Ibu berkata : "Jangan pulang sore-sore!"
- Delimiter terakhir harus diikuti oleh satu atau lebih *white space*(spasi, tab, *newline*).
- Kata pertama kalimat berikutnya seharusnya dimulai dengan huruf kapital.
- *Sentence delimiter* tidak seharusnya menjadi bagian dari singkatan (*abbreviation*).

Pemecahan kata

Mengikuti pemecahan kalimat, pemecahan kalimat menjadi kata-kata tunggal juga dilakukan, yaitu dengan men-*scan* kalimat dengan pemisah *white space* (spasi, tab, *newline*). Sebuah kata didefinisikan sebagai sebuah runtutan karakter - karakter ([A-Z], [a-z] dan / atau [0,9]).

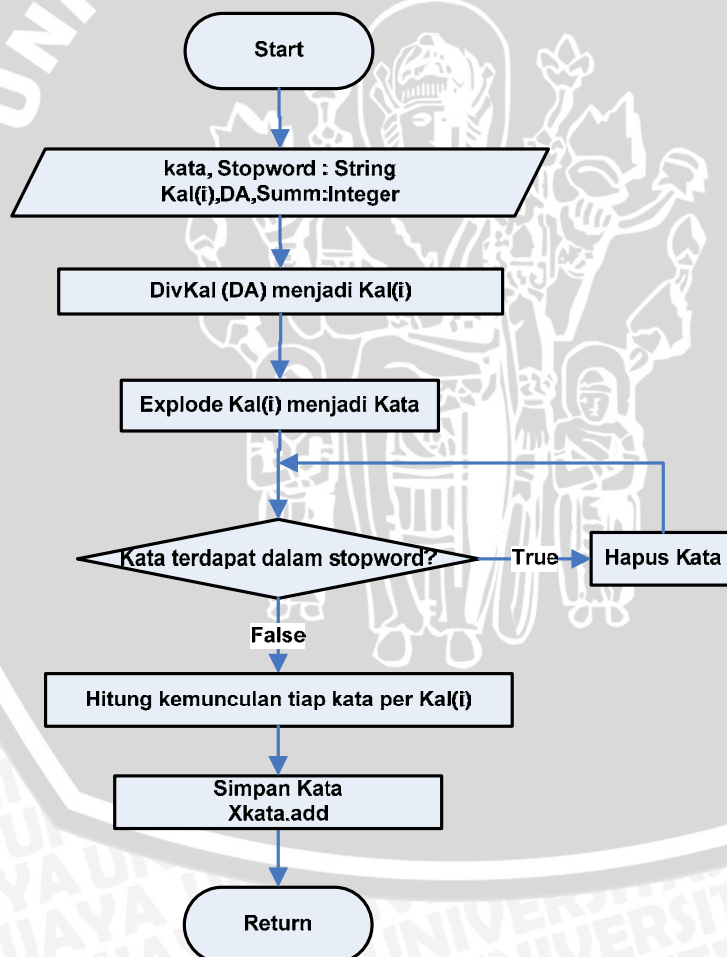
Hapus stopwords

Stop word merupakan kata- kata yang tidak berpengaruh atau tidak memiliki arti penting untuk menentukan topik dari suatu teks. Yang termasuk ke dalam *stop word* adalah kata ganti orang (saya, kamu, dia, anda, engkau,...), kata hubung (dan, tidak, atau,...), kata depan (di, ke, pada,...).

Pada Gambar 3.4 merupakan Alur Proses preprocessing yang memiliki langkah-langkah sebagai berikut :

1. User memasukkan data dokumen yang akan dibuat ringkasan. Dalam hal ini dokumen awal (DA) bertipe *.txt.

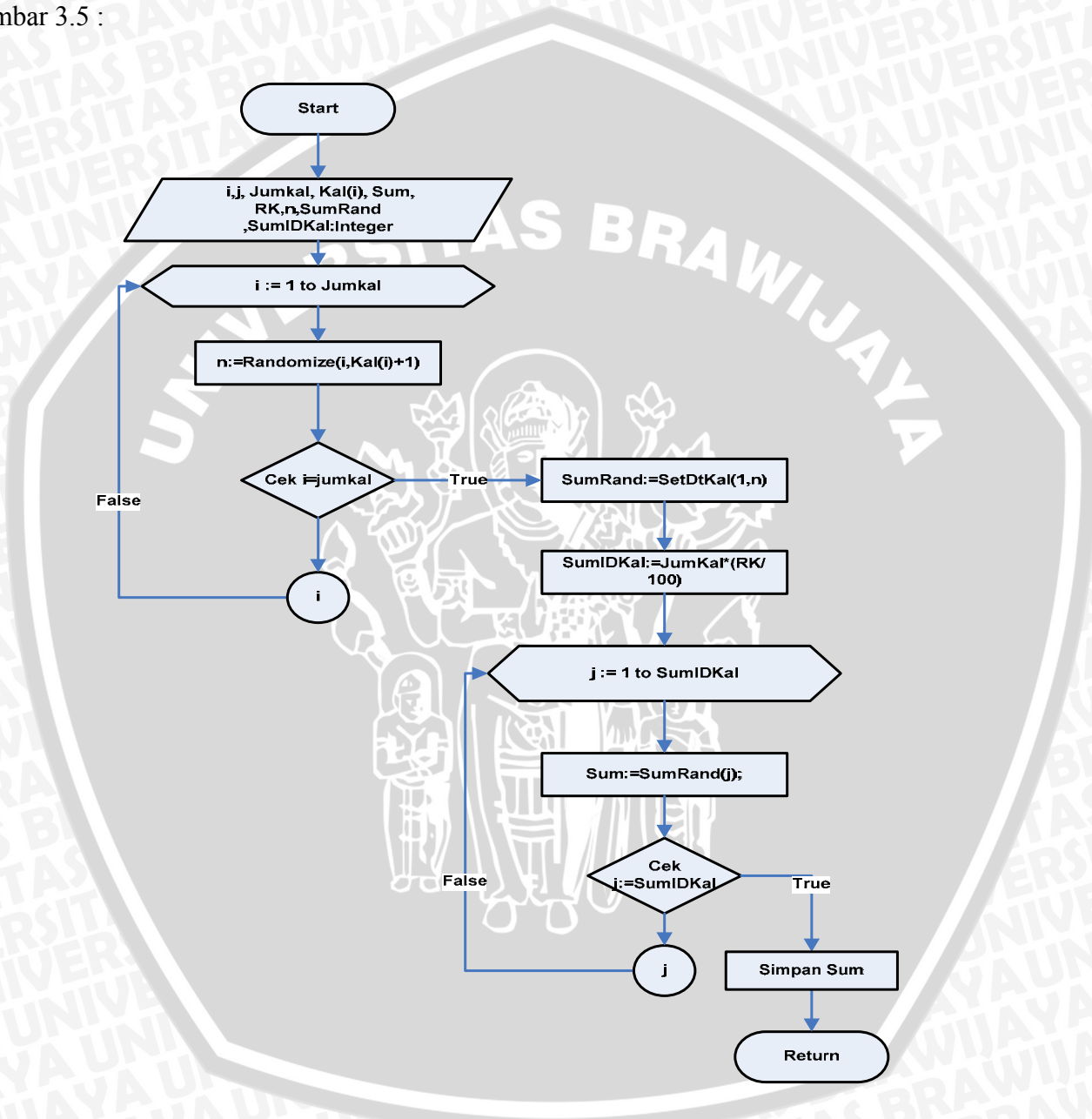
2. User menentukan batas jumlah prosentase kalimat kesimpulan terhadap dokumen asli (batas *threshold* kalimat kesimpulan yang diambil) .
3. User memasukkan batas jumlah iterasi perhitungan MMR sampai diperoleh nilai iterasi MMR yang konvergen.
4. Sistem akan mengambil data daftar kata dari *stopword*.
5. Dokumen asal (DA) akan dipecah (*parsing*) menjadi kalimat – kalimat dan diberi nomer kalimat (Si).
6. Kalimat – kalimat tersebut (DA/Summ) diparsing menjadi kata – kata kemudian dicari dan dihilangkan *stopword* nya.
7. Hitung banyaknya kemunculan tiap kata perkalimat (Si).
8. Hasil akhir dari kata – kata ini disimpan untuk proses berikutnya.



Gambar 3.4 Alur Get Preprocessing

3.2.1.2 Proses Inisialisasi

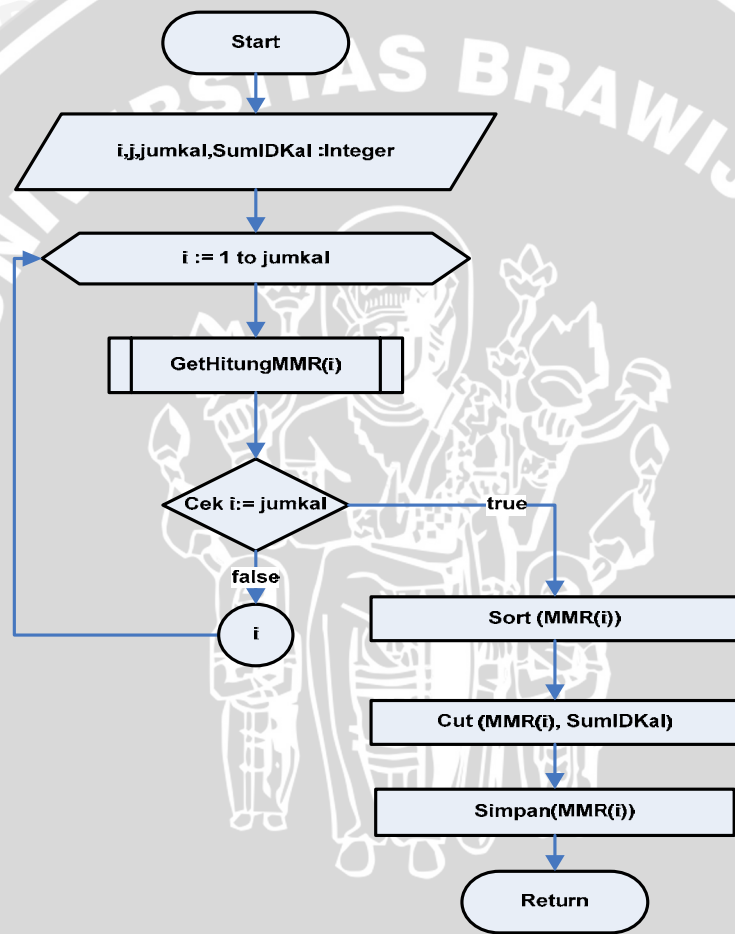
Pada proses Inisialisasi dilakukan proses pembentukan ringkasan latihan untuk pertama kalinya. Kalimat-kalimat pada ringkasan latihan ini diambil secara random sesuai dengan batas threshold yang diberikan oleh user. Alur proses pembentukan ringkasan ditunjukkan pada Gambar 3.5 :



Gambar 3.5 Alur Proses Inisialisasi

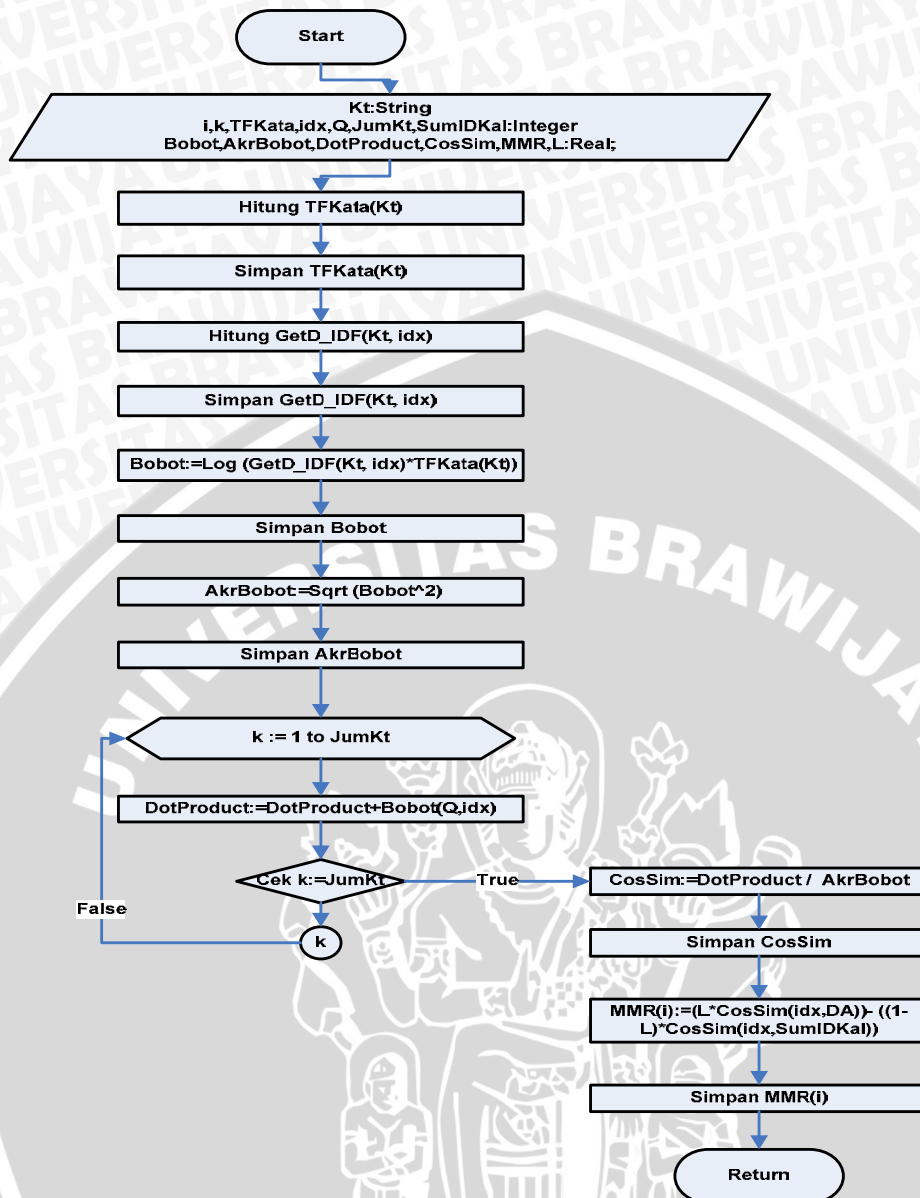
3.2.1.3 Proses Get MMR

Pada proses GetMMR dilakukan perhitungan relevansi perkalamat(i) (MMR(i)) terhadap DA dan Summ. Setelah semua kalimat diperoleh nilai MMR(i) nya selanjutnya dilakukan pengurutan kebawah kalimat-kalimat yang mempunyai MMR(i). Langkah selanjutnya yaitu dilakukan pemotongan kalimat yang mempunyai nilai MMR(i) terbesar berdasarkan batas *threshold* yang diberikan oleh user. Kalimat-kalimat yang sudah diperoleh inilah yang selanjutnya akan digunakan sebagai kandidat kalimat ringkasan. Alur proses *Get MMR* ditunjukkan pada Gambar 3.6 :



Gambar 3.6 Alur proses *Get MMR(i)*

Didalam proses *GetMMR* terdapat proses *GetHitungMMR(i)* yang digunakan untuk menghitung secara detail perhitungan nilai relevansi perkalamat(i) terhadap dokumen(DA/Sum). Alur proses *Get MMR* ditunjukkan pada Gambar 3.7 :



Gambar 3.7 Alur proses *GetHitungMMR(i)*

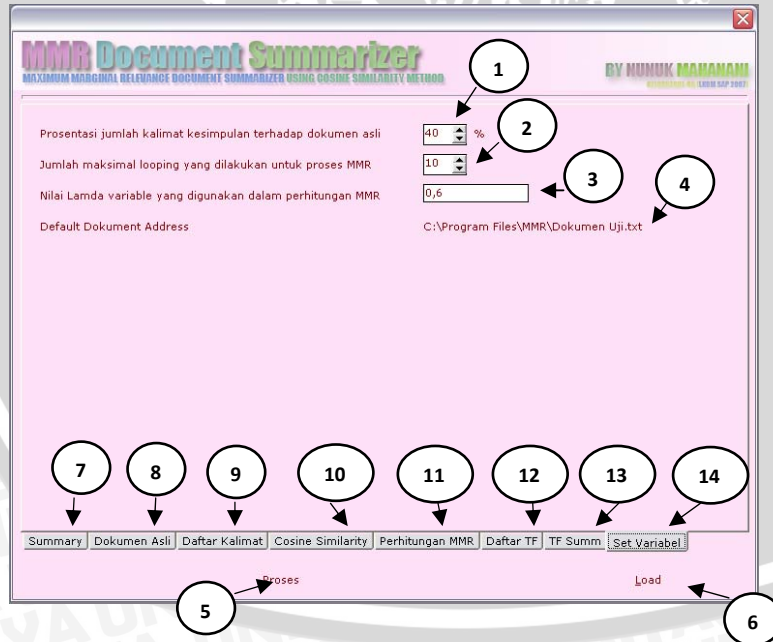
3.2.2 Perancangan Antar Muka

Beberapa hasil analisis secara keseluruhan terdapat beberapa bagian yang dibutuhkan dalam antar muka sistem peringkas dokumen ini, yaitu :

1. Bagian untuk memasukkan prosentase jumlah kalimat kesimpulan terhadap dokumen asli.
2. Bagian untuk memasukkan jumlah maksimal looping untuk perhitungan MMR.
3. Nilai variabel lamda yang digunakan untuk perhitungan MMR.
4. Alamat dokumen asal (DA)
5. Tombol untuk memilih dokumen asal yang diinginkan oleh user.

6. Tombol untuk menjalankan proses perhitungan dan memunculkan hasil ringkasan secara bersamaan.
7. Bagian untuk menampilkan hasil ringkasan akhir.
8. Bagian untuk menampilkan dokumen asli (DA) yang dimasukan oleh user.
9. Bagian untuk menampilkan daftar kalimat hasil pecahan/parsing dari dokumen asal (DA).
10. Bagian untuk menampilkan hasil perhitungan cosine similiarity kalimat (Si) per-iterasi.
11. Bagian untuk menampilkan hasil perhitungan MMR (maximum marginal relevance) kalimat (Si) per-iterasi.
12. Bagian untuk menampilkan frekuensi kata per-kalimat (Si) pada dokumen asal (DA).
13. Bagian untuk menampilkan frekuensi kata pada dokumen asal (DA) dan ringkasan latih (RL).
14. Bagian untuk memasukkan variabel – variabel yang dibutuhkan dalam perhitungan.

Semua bagian dan komponen – komponen tersebut akan disajikan dalam satu halaman antar muka. Rancangan dari antarmuka ini dapat dilihat pada gambar 3.8:



Gambar 3.8 Rancangan Antar Muka

3.3 Evaluasi dan Perbandingan Sistem

Pada subbab evaluasi dan perbandingan sistem ini akan dilakukan perancangan uji coba dari sistem peringkasan dokumen otomatis menggunakan Metode MMR, pengujian terhadap sistem apakah telah sesuai dengan analisis dan perancangan, maupun evaluasi ringkasan yang dihasilkan. Evaluasi hasil ringkasan dilakukan dengan membandingkan hasil output sistem peringkasan dokumen ini dengan hasil output sistem peringkasan yang lain, yaitu sistem peringkasan dokumen otomatis menggunakan algoritma *shortest path*.

Dalam penelitian kali ini sistem peringkasan yang lain yang digunakan adalah sistem peringkasan yang menggunakan algoritma *shortest path*. Yang akan dibandingkan adalah besarnya *f-measure* (kombinasi nilai dari *recall* dan *precision*) antara hasil ringkasan sistem dan hasil ringkasan pembanding dengan ukuran ringkasan dan dokumen asal yang sama yang masing-masing dibandingkan dengan ringkasan ideal. Definisi ringkasan ideal adalah ringkasan yang dibuat oleh seorang *abstractor*, dalam penelitian ini *abstractor* yang dipilih adalah seorang ahli bahasa Indonesia. Sebuah ringkasan dikatakan baik, apabila nilai *f-measure*nya lebih besar.

Dalam sistem peringkasan yang menggunakan algoritma *shortest path*, pembentukan ringkasan dilakukan dengan merepresentasikan sebuah teks atau dokumen sebagai graf dan ringkasan yang dibentuk merupakan *shortest path* (jalur terpendek) dari dokumen graf tersebut.

Sedangkan pada sistem peringkasan ini, dilakukan perhitungan relevansi / hubungan kepentingan antara kalimat (S_i) terhadap dokumen asal (DA) dan terhadap ringkasan latih (RL). Relevansi kalimat ini dihitung melalui proses similarity / kesamaan kalimat. Kalimat – kalimat yang diambil menjadi ringkasan adalah kalimat yang mempunyai tingkat relevansi tertinggi.

Di dalam analisis penelitian ini akan dicari apakah peringkasan dokumen menggunakan metode MMR (*Maximum Marginal Relevance*) lebih baik dibandingkan dengan peringkasan yang menggunakan algoritma *shortest path*.

Pada tahap ini, uji coba dilakukan sebanyak tiga kali yaitu pengujian terhadap sistem peringkasan dokumen (MMR), pengujian terhadap sistem peringkasan dokumen (*shortest path*) dan pengujian terhadap ringkasan ideal. Seperti yang telah diketahui bersama, bahwa tidak ada satupun ringkasan yang ideal, baik yang dibuat oleh sistem peringkasan yang telah ada, bahkan yang dibuat oleh manusia sekalipun. Tetapi, dikarenakan tujuan dari pembuatan

ringkasan ini adalah untuk menghasilkan ringkasan yang mendekati ringkasan buatan manusia, maka yang akan dipakai pada penelitian ini adalah ringkasan hasil buatan manusia. Oleh karena itu teks berita Indonesia yang akan diringkas oleh sistem ini, terlebih dahulu diringkas oleh beberapa *abstractor* (pengguna).

Untuk menentukan ringkasan ideal, maka dilakukan pengujian dengan melibatkan satu orang *abstractor* merupakan seorang ahli bahasa Indonesia. *Abstractor* (pengguna) ini diberi 15 teks berita yang sudah dipisah-pisah perkalimat dan diberi nomer kalimat. Selanjutnya *abstractor* akan memilih nomer kalimat mana saja dalam berita per teks dokumen tersebut yang yang masing – masing akan diperoleh ringkasan. Ukuran (jumlah kalimat) dari ringkasan diusahakan mendekati ukuran yang telah ditentukan. Tabel hasil ringkasan ideal ditunjukkan pada Tabel 3.1.

Tabel 3.1 Tabel Hasil Ringkasan Ideal

No Dokumen	Jumlah kalimat	Jumlah Kalimat Ringkasan	No Kalimat yang dipilih

Setelah diperoleh hasil ringkasan ideal maka dapat dihitung hasil perbandingan sistem peringkas (MMR) terhadap ringkasan ideal. Tabel pembanding sistem untuk sistem peringkas dokumen dengan metode MMR ditunjukkan pada Tabel 3.2.

Tabel 3.2 Tabel Pembanding Sistem (MMR)

No Doc	Overlap MMR	Precision	Recall	F-measure

Dari 15 nilai *recall*, *precision* dan *f-measure* akan diperoleh nilai yang berbeda-beda tersebut. Untuk mendapatkan hasil akhir *recall*, *precision* dan *f-measure*, masing-masing dihitung nilai rata-rata *recall*, *precision* dan *f-measure*. Hasil akhir *recall*, *precision* dan *f-measure* sistem (MMR) ditunjukkan pada Tabel 3.2.

Tabel 3.3 Tabel Hasil akhir *recall*, *precision* dan *f-measure* sistem (MMR)

	<i>Recall</i>	<i>Precision</i>	<i>F-measure</i>
Rata-rata			

Dengan data yang sama dilakukan pengujian sistem terhadap sistem peringkasan dokumen (*shortest path*). Tabel 3.3 menunjukkan hasil *precision*, *recall* dan *F-measure* sistem peringkasan dokumen (*shortest path*) perdokumen. Sedangkan hasil akhir nilai perhitungan *precision*, *recall* dan *F-measure* sistem (*shortest path*) ditunjukkan pada Tabel 3.4.

Tabel 3.4 Tabel Pembanding Sistem (*shortest path*)

No Doc	Overlap <i>shortest path</i>	Precision	Recall	F-measure

Tabel 3.5 Tabel Hasil akhir *recall*, *precision* dan *f-measure* sistem (*shortest path*)

	<i>Recall</i>	<i>Precision</i>	<i>F-measure</i>
Rata-rata			

3.4 Contoh Perhitungan Manual

Berikut ini dicontohkan bagaimana cara kerja sistem untuk membentuk ringkasan dari sebuah teks dokumen.

Puluhan Jenazah Ditemukan di Krematorium di Georgia

Puluhan jenazah yang telah membusuk ditemukan di gudang penyimpanan dan hutan di belakang sebuah krematorium di Noble, Georgia, Amerika Serikat (AS). Sampai Minggu (17/2) malam, pihak berwajib menemukan 97 jenazah dan merencanakan untuk melanjutkan pencarian hari Senin. Dikhawatirkan, jumlah jenazah yang ditemukan di luar krematorium itu melebihi angka 200.

"Mungkin kami akan menemukan 200 jenazah, bahkan lebih," kata Kris Sperry, dokter forensik di Walker County, Georgia, di mana krematorium itu terletak. Penemuan mengerikan itu terjadi dalam sebuah pencarian polisi yang intensif pada tanah luas milik Krematorium Tri-State di Noble, sebuah kota kecil sekitar 140 kilometer sebelah barat laut Atlanta.

Para penyidik mengatakan, mereka telah menemukan sekitar 100 jenazah dalam berbagai tingkat pembusukan, namun memperkirakan bahwa sebuah pemeriksaan pada lima ruang besi pemakaman di krematorium itu akan meningkat sampai sejumlah 200.

Manajer krematorium Ray Brent Marsh, yang dikenai lima tuduhan penipuan, dibebaskan dari tahanan hari Minggu setelah menyerahkan uang tanggungan sebesar 25.000 dollar AS. Pihak berwajib menuduh Marsh (28) menerima jenazah-jenazah untuk dikremasi dari perusahaan-perusahaan pemakaman, namun malah mengubur jenazah-jenazah itu di hutan atau menumpuk mereka di ruang-ruang besi di tanah milik Krematorium Tri-State yang luasnya sekitar lima hektar itu. Mereka mengatakan, tuduhan-tuduhan lain kemungkinan besar akan dikenakan pada Marsh.

- Parsing kalimat dokumen asal (DA), pembentukan ringkasan latih (RL) dan penghapusan stopword. Dalam penelitian ini pembentukan RL dilakukan secara random (memilih kalimat – kalimat dari DA sebanyak jumlah ringkasan yang telah ditentukan oleh user). Dalam perhitungan manual ini, *threshold* penentuan panjang ringkasan adalah 40% dari jumlah kalimat DA.
- Pada iterasi pertama contoh perhitungan manual pada sistem ini, RL yang diperoleh adalah kalimat ke 7,3,5,4.
- Langkah pertama yang dilakukan yaitu dilakukan pemecahan kata pada dokumen asal, kemudian dihitung frekuensi kemunculan tiap kata per dokumen asal (DA) dan per ringkasan latih (RL). Tabel 3.6 menunjukkan jumlah Frekuensi/Kemunculan Kata per DA dan RL

Tabel 3.6 Tabel Frekuensi/Kemunculan Kata DA dan RL

kata	fDA	fRL	Kata	fDA	fRL
jenazah	9	2	jumlah	1	1
ditemukan	2	1	Luar	1	1
gudang	1	0	melebihi	1	1
hutan	2	0	200	3	2
belakang	1	0	Kris	1	1
krematorium	7	3	sperry	1	1
noble	1	0	dokter	1	1
georgia	2	0	forensik	1	1
amerika	1	0	walker	1	1
serikat	1	0	country	1	1
(AS)	2	1	penemuan	1	1
minggu	2	1	polisi	1	1
(17/2)	1	0	intensif	1	1
berwajib	2	0	tanah	2	1
menemukan	3	1	Milik	2	1
97	1	0	Tri-State	2	1
melanjutkan	1	0	Kota	1	1
pencarian	2	0	140	1	1
senin	1	0	kilometer	1	1
dikhawatirkan	1	1	barat	1	1

laut	1	1	dikenai	1	1
atlanta	1	1	tuduhan	3	1
penyidik	1	0	penipuan	1	1
mengatakan	2	0	dibebaskan	1	1
100	1	0	tahanan	1	1
tingkat	1	0	menyerahkan	1	1
pembusukan	1	0	uang	1	1
memperkirakan	1	0	sebesar	1	1
pemeriksaan	1	0	25.000	1	1
lima	3	1	Dollar	1	1
ruang	3	0	menuduh	1	0
besi	2	0	(28)	1	0
pemakaman	2	0	menerima	1	0
meningkat	1	0	dikremasi	1	0
sejumlah	1	0	perusahaan	2	0
manajer	1	1	mengubur	1	0
ray	1	1	menumpuk	1	0
brebt	1	1	hektar	1	0
marsh	3	1	dikenakan	1	0

- Dilakukan perhitungan cosine similiarity tiap kalimat (S_i) terhadap dokumen (DA) dan Ringkasan Latih (RL).

$IDF_i = \log (D/df_i)$ dimana $D=2$;

Perhitungan pada Tabel 3.7 ini menunjukkan langkah – langkah perhitungan pencarian similarity untuk kalimat pertama (S1).

Tabel 3.7 Tabel Perhitungan Frekuensi dan Pembobotan Kata

	tfi	count,tfi			dfi	D/dfi	Weight, $w_i = t_{fi} * IDF_i$		
		fS1	fDA	fRL1			S1	DA	RL1
jenazah	1	9	2	2	0,0000	0,0000	0,0000	0,0000	
ditemukan	1	2	1	2	0,0000	0,0000	0,0000	0,0000	
gudang	1	1	0	1	1,0000	1,0000	1,0000	0,0000	
hutan	1	2	0	1	1,0000	1,0000	2,0000	0,0000	
belakang	1	1	0	1	1,0000	1,0000	1,0000	0,0000	
krematorium	1	7	3	2	0,0000	0,0000	0,0000	0,0000	
noble	1	1	0	1	1,0000	1,0000	1,0000	0,0000	
georgia	1	2	0	1	1,0000	1,0000	2,0000	0,0000	
amerika	1	1	0	1	1,0000	1,0000	1,0000	0,0000	
serikat	1	1	0	1	1,0000	1,0000	1,0000	0,0000	
(AS)	1	2	1	2	0,0000	0,0000	0,0000	0,0000	
minggu	0	2	1	2	0,0000	0,0000	0,0000	0,0000	
(17/2)	0	1	0	1	1,0000	0,0000	1,0000	0,0000	
berwajib	0	2	0	1	1,0000	0,0000	2,0000	0,0000	
menemukan	0	3	1	2	0,0000	0,0000	0,0000	0,0000	
97	0	1	0	1	1,0000	0,0000	1,0000	0,0000	

melanjutkan	0	1	0	1	1,0000	0,0000	1,0000	0,0000
pencarian	0	2	0	1	1,0000	0,0000	2,0000	0,0000
senin	0	1	0	1	1,0000	0,0000	1,0000	0,0000
dikhawatirkan	0	1	1	2	0,0000	0,0000	0,0000	0,0000
jumlah	0	1	1	2	0,0000	0,0000	0,0000	0,0000
luar	0	1	1	2	0,0000	0,0000	0,0000	0,0000
melebihi	0	1	1	2	0,0000	0,0000	0,0000	0,0000
200	0	3	2	2	0,0000	0,0000	0,0000	0,0000
kris	0	1	1	2	0,0000	0,0000	0,0000	0,0000
sperry	0	1	1	2	0,0000	0,0000	0,0000	0,0000
dokter	0	1	1	2	0,0000	0,0000	0,0000	0,0000
forensik	0	1	1	2	0,0000	0,0000	0,0000	0,0000
walker	0	1	1	2	0,0000	0,0000	0,0000	0,0000
country	0	1	1	2	0,0000	0,0000	0,0000	0,0000
penemuan	0	1	1	2	0,0000	0,0000	0,0000	0,0000
polisi	0	1	1	2	0,0000	0,0000	0,0000	0,0000
intensif	0	1	1	2	0,0000	0,0000	0,0000	0,0000
tanah	0	2	1	2	0,0000	0,0000	0,0000	0,0000
milik	0	2	1	2	0,0000	0,0000	0,0000	0,0000
Tri-State	0	2	1	2	0,0000	0,0000	0,0000	0,0000
kota	0	1	1	2	0,0000	0,0000	0,0000	0,0000
140	0	1	1	2	0,0000	0,0000	0,0000	0,0000

kilometer	0	1	1	2	0,0000	0,0000	0,0000	0,0000
barat	0	1	1	2	0,0000	0,0000	0,0000	0,0000
laut	0	1	1	2	0,0000	0,0000	0,0000	0,0000
atlanta	0	1	1	2	0,0000	0,0000	0,0000	0,0000
penyidik	0	1	0	1	1,0000	0,0000	1,0000	0,0000
mengatakan	0	2	0	1	1,0000	0,0000	2,0000	0,0000
100	0	1	0	1	1,0000	0,0000	1,0000	0,0000
tingkat	0	1	0	1	1,0000	0,0000	1,0000	0,0000
pembusukan	0	1	0	1	1,0000	0,0000	1,0000	0,0000
memperkirakan	0	1	0	1	1,0000	0,0000	1,0000	0,0000
pemeriksaan	0	1	0	1	1,0000	0,0000	1,0000	0,0000
lima	0	3	1	2	0,0000	0,0000	0,0000	0,0000
ruang	0	3	0	1	1,0000	0,0000	3,0000	0,0000
besi	0	2	0	1	1,0000	0,0000	2,0000	0,0000
pemakaman	0	2	0	1	1,0000	0,0000	2,0000	0,0000
meningkat	0	1	0	1	1,0000	0,0000	1,0000	0,0000
sejumlah	0	1	0	1	1,0000	0,0000	1,0000	0,0000
manajer	0	1	1	2	0,0000	0,0000	0,0000	0,0000
ray	0	1	1	2	0,0000	0,0000	0,0000	0,0000
brebt	0	1	1	2	0,0000	0,0000	0,0000	0,0000
marsh	0	3	1	2	0,0000	0,0000	0,0000	0,0000
dikenai	0	1	1	2	0,0000	0,0000	0,0000	0,0000

tuduhan	0	3	1	2	0,0000	0,0000	0,0000	0,0000
penipuan	0	1	1	2	0,0000	0,0000	0,0000	0,0000
dibebaskan	0	1	1	2	0,0000	0,0000	0,0000	0,0000
tahanan	0	1	1	2	0,0000	0,0000	0,0000	0,0000
menyerahkan	0	1	1	2	0,0000	0,0000	0,0000	0,0000
uang	0	1	1	2	0,0000	0,0000	0,0000	0,0000
sebesar	0	1	1	2	0,0000	0,0000	0,0000	0,0000
25.000	0	1	1	2	0,0000	0,0000	0,0000	0,0000
dollar	0	1	1	2	0,0000	0,0000	0,0000	0,0000
menuduh	0	1	0	1	1,0000	0,0000	1,0000	0,0000
(28)	0	1	0	1	1,0000	0,0000	1,0000	0,0000
menerima	0	1	0	1	1,0000	0,0000	1,0000	0,0000
dikremasi	0	1	0	1	1,0000	0,0000	1,0000	0,0000
perusahaan	0	2	0	1	1,0000	0,0000	2,0000	0,0000
mengubur	0	1	0	1	1,0000	0,0000	1,0000	0,0000
menumpuk	0	1	0	1	1,0000	0,0000	1,0000	0,0000
hektar	0	1	0	1	1,0000	0,0000	1,0000	0,0000
dikenakan	0	1	0	1	1,0000	0,0000	1,0000	0,0000

Berdasarkan rumus perhitungan di bab 2 untuk pencarian cosine similiarity, maka didapatkan nilai bobot dan dot product :

$$\begin{aligned}
 |S1| &= 2.645751 & S1.DA &= 9 \\
 |DA| &= 8.124038 & S1.RL &= 0 \\
 |RL| &= 0 & |S1|.|DA| &= 21,49419 \\
 & & |S1|.|RL| &= 0
 \end{aligned}$$

Setelah nilai bobot dan nilai dotproduct diperoleh, maka dapat dihitung similarity kalimat ke-i (S1) terhadap dokumen asal (DA) dan terhadap ringkasan latih (RLi).

$$\text{Cosine}(S1, DA) = \frac{S1 \cdot DA}{|S1| \cdot |DA|} = \frac{9}{21,49419} = 0,418718$$

$$\text{Cosine}(S1, RL1) = \frac{S1 \cdot RL}{|S1| \cdot |RL|} = \frac{0}{0} = \epsilon$$

Dalam perhitungan nilai cosine antar kalimat ini terdapat nilai yang tak terdefinisi (ϵ), sehingga nilai tersebut tidak dimasukkan dalam perhitungan selanjutnya.

Setelah nilai similarity ditemukan kemudian dilakukan perhitungan nilai MMR (relevansi) kalimat tersebut terhadap dokumen. Dengan rumus yang telah diberikan pada bab 2 yaitu

$$\text{MMR} = \max [\lambda \times \max \text{Sim1}(S_i, D) - ((1 - \lambda) \times \max \text{Sim2}(S_i, \text{Summ}))]$$

Dengan nilai koefisien yang telah ditetapkan sebesar 0,6, maka diperoleh nilai MMR kalimat 1 pada iterasi-1 adalah

$$\text{MMR}(S1) = (0,6 * 0,199048) - ((1 - 0,6) * \epsilon) = 0,251231$$

Pencarian nilai MMR pada perhitungan tersebut dilakukan pada semua kalimat (Si) pada dokumen (DA). Sehingga diperoleh nilai MMR(Si) adalah :

S1	0,251231
S2	0,241209
S3	ϵ
S4	ϵ
S5	ϵ
S6	0,362441

S7	ϵ
S8	0,423587
S9	0,073855

Setelah nilai MMR perkalamat (S_i) diperoleh kemudian kalimat – kalimat tersebut diurutkan berdasarkan nilai MMRnya dan dilakukan pemotongan sesuai dengan nilai koefisien yang diberikan oleh user ($MMR_{Max} = 40\%$ dari panjang dokumen).

S8	0,423587
S6	0,362441
S1	0,251231
S2	0,241209
S9	0,073855
S3	ϵ
S4	ϵ
S5	ϵ
S7	ϵ



Pada pengurutan hasil perhitungan MMR tersebut maka dapat diketahui bahwa kalimat – kalimat hasil summary dari iterasi-1 adalah kalimat 8, 6, 1, 2.

Dilakukan pengecekan terhadap kalimat – kalimat pembentuk summary dengan summary sebelumnya, apabila belum konvergen maka dilanjutkan iterasi selanjutnya.

Setelah dilakukan 3 kali iterasi didapatkan kalimat – kalimat pembentuk summary yang konvergen, yaitu kalimat hasil MMR tertinggi pada iterasi ke-1 muncul kembali pada iterasi ke-3.

Tabel 3.8 Tabel Iterasi

R	I	II	Ringk. Baru
S7	S8	S7	S8
S3	S6	S9	S6
S5	S1	S5	S1
S4	S2	S4	S2

Berdasarkan tabel 3.8 dapat diketahui kalimat – kalimat yang digunakan sebagai ringkasan adalah :

- Kalimat ke-1 : Puluhan jenazah yang telah membusuk ditemukan di gudang penyimpanan dan hutan di belakang sebuah krematorium di Noble, Georgia, Amerika Serikat (AS).
- Kalimat ke-2 : Sampai Minggu (17/2) malam, pihak berwajib menemukan 97 jenazah dan merencanakan untuk melanjutkan pencarian hari Senin.
- Kalimat ke-6 : Para penyidik mengatakan, mereka telah menemukan sekitar 100 jenazah dalam berbagai tingkat pembusukan, namun memperkirakan bahwa sebuah pemeriksaan pada lima ruang besi pemakaman di krematorium itu akan meningkat sampai sejumlah 200.
- Kalimat ke-8 : Pihak berwajib menuduh Marsh (28) menerima jenazah-jenazah untuk dikremasi dari perusahaan-perusahaan pemakaman, namun malah mengubur jenazah-jenazah itu di hutan atau menumpuk mereka di ruang-ruang besi di tanah mi-lik Krematorium Tri-State yang luasnya sekitar lima hektar itu.

BAB IV

IMPLEMENTASI DAN UJI COBA

Implementasi merupakan proses transformasi representasi rancangan ke bahasa pemrograman yang dapat dimengerti oleh komputer. Pada bab ini akan dibahas hal-hal yang berkaitan dengan implementasi sistem peringkasan dokumen.

4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dipaparkan disini meliputi lingkungan perangkat keras dan lingkungan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem peringkasan dokumen ini adalah sebagai berikut :

1. Prosesor Intel Pentium Dual-Core 1,73 GHz
2. RAM 3062 MB
3. *Harddisk* dengan kapasitas 80 GB
4. Monitor 14"
5. Keyboard
6. Mouse

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem peringkasan dokumen ini adalah :

1. Sistem Operasi Windows XP
2. *Borland Delphi 7*
3. *Microsoft Office Visio 2003*

4.2 Implementasi Proses Pembentuk Sistem

Berdasarkan hasil perancangan sistem pada subbab 3.2 , mengenai analisa proses pembentuk sistem dalam sistem peringkasan dokumen ini, maka pada subbab ini akan dijelaskan implementasi perancangan sistem tersebut.

4.2.1 Source Code Proses Get Preprocessing

Langkah awal dari proses pembentukan ringkasan adalah dilakukan proses get preprocessing. Didalam proses ini, dilakukan proses pemecahan dokumen asal menjadi

kalimat. Setelah dipecah menjadi kalimat selanjutnya dilakukan proses pemecahan kalimat menjadi kata dan penghapusan stopword. Prosedur proses pemecahan dokumen asal, prosedur pemecahan kalimat menjadi kata dan prosedur penghapusan stopword dapat dilihat pada *Pseudocode 4.1, Pseudocode 4.2, Pseudocode 4.3.*

```
procedure TDA.DivKal(Value: string);
var
  i,j, TdK: integer;
  temp : string;
begin
  temp := '';
  j := MinLoop;
  TdK := 0;
  for i:= MinLoop to Length(Value) do
  begin
    if (Trim(Value[i]) = '(') then inc(TdK);
    if (Trim(Value[i]) = ')') then dec(TdK);

    if (Trim(Value[i]) <> '.') then
      temp:= temp + Value[i]
    else
      if ((Trim(Value[i+1]) < #47) or
(Trim(Value[i+1]) > #58)) then
        if (Tdk = 0) and
          (not Self.InPrefix(Value,i)) and (not
Self.InSufix(Value,i)) then
          begin
            if temp <> '' then
              begin
                Self.SetL_Kal(j);
                Self.Kalimat[j]:=
TKalimat.Create(Trim(temp));
                Self.KalUtuh[j]:= Trim(LowerCase(temp));
                inc(j); temp:= '';
              end;
            end else
              temp:= temp + '.';
          end;
        end;
      end;
    end;
  end;
end;
```

Pseudocode 4.1 Prosedur proses pemecahan dokumen asal menjadi kalimat

```
procedure TKalimat.Explode(Value: string);
var
  i: integer;
  temp : string[25];
begin
  temp:= "";
  Self.BanyakKata:= 1;
  for i:= 0 to Length(Value) - 1 do
  begin
    if ((Value[i] > #96) and (Value[i] < #123)) or
      ((Value[i] > #63) and (Value[i] < #91)) then
      temp:= temp + Value[i]
    else
      if (Trim(temp) <> "") then
      begin
        if not (FUtama.StopWord(temp)) then
          Self.Set_Kata(LowerCase(Trim(temp)));
        temp:= "";
      end;
    end;
  end;
end;
```

Pseudocode 4.2 Prosedur proses pemecahan kalimat menjadi kata

```
function TFUtama.StopWord(Term: String): Boolean;
var
  i: integer;
begin
  result:= False;
```

```

for i:= MinLoop to High(Self.SW) do
  if LowerCase(Trim(Self.SW[i])) = LowerCase(Trim(Term))
then
  begin
    Result:= True;
    Break
  end;
end;

```

Pseudocode 4.3 Prosedur proses penghapusan stopwords

4.2.2 Source Code Proses Inisialisasi Sum(0)

Didalam proses inisialisasi Sum(0) ini, dilakukan pembentukan ringkasan latih awal yang kalimat – kalimatnya diperoleh secara random. Prosedur proses pembentukan summary awal dapat dilihat pada *Pseudocode 4.4*.

```

procedure TSummary.SetRandomKal;
var
  i,n: Integer;
begin
  for i:= MinLoop to Self.JumKalimat do
  begin
    n:= RandomRange(MinLoop, (DA.JumKalimat + 1));
    while Self.GetIdxKal(n) >= MinLoop do
    begin
      Randomize;
      n:= RandomRange(MinLoop, (DA.JumKalimat + 1));
    end;
    Self.SetIdKal(i, n);
    Self.Kalimat[i]:= DA.GetKalimat(Self.GetIdKal(i));
  end;
  Self.SetDT;
end;

```

Pseudocode 4.4 Prosedur proses pembentukan summary awal**4.2.3 Source Code Proses Get MMR**

Pada proses GetMMR dilakukan perhitungan relevansi perkalamat(i) (MMR(i)) terhadap DA dan Summ. Perhitungan relevansi kalimat dilakukan dengan dua tahap yaitu perhitungan Simmilarity kalimat (Si) terhadap DA dan RL menggunakan persamaan cosine similiarity dan perhitungan relevansi kalimat (Si) menggunakan persamaan MMR. Prosedur proses perhitungan cosine similiarity dan prosedur perhitungan MMR dapat dilihat pada *Pseudocode 4.5* dan *Pseudocode 4.6*.

```
function TCosineSim.GetCosineSim(Q: Integer; idx: idxDok):
real;
var
temp: real;
begin
temp:= Self.GetVektor(Q,idx);
if temp <> 0 then
Result:= Self.GetDotProduk(Q, idx) / temp
else
if idx = 0 then Result:= -100 else Result:= 100;

FUtama.TampilCosine(Q,idx, Self.GetDotProduk(Q,
idx),temp,Result);
end;
```

Pseudocode 4.5 Prosedur proses perhitungan persamaan cosine similiarity

```
function TMMR.GetMMR(idx: integer): real;
var
Sim1,Sim2: real;
begin
```



```

Sim1:= Self.Cosine.GetCosineSim(idx,idxDA);
Sim2:= Self.Cosine.GetCosineSim(idx,1);
Result:= (Self.Lamda * Sim1) - (Self._L * Sim2);
FUtama.TampilMMR(idx, Sim1,Sim2, Result);
end;

```

Pseudocode 4.6 Prosedur proses perhitungan persamaan MMR

Setelah semua kalimat diperoleh nilai MMR(i) nya selanjutnya, kemudian posisi kalimat dimasukkan kedalam list yang telah terurut berdasarkan nilai MMR(Si). Prosedur proses pengurutan posisi kalimat berdasarkan nilai MMR(Si) dapat dilihat pada *Pseudocode 4.7*.

```

procedure TMMR.InsertToList(idx:Integer; Value : Real);
var
  j,i: integer;
begin
  i:= GetIdx(Value);
  if i < DA.JumKalimat then
  begin
    j:= DA.JumKalimat;
    while j > i do
    begin
      Self.LKal[j]:= Self.LKal[j - 1];
      Dec(j);
    end;
    Self.LKal[i].idxKal:= idx;
    Self.LKal[i].MMR:= Value;
  end else
  begin
    Self.LKal[i].idxKal:= idx;
    Self.LKal[i].MMR:= Value;
  end;
end;

```

Pseudocode 4.7 Prosedur proses Insert To List

Untuk pembentukan ringkasan akhir, pada setiap iterasi dilakukan pengecekan apakah kalimat – kalimat hasil ringkasan pada iterasi (i) sudah konvergen. Pada *Pseudocode 4.8* akan ditunjukkan proses pencarian nilai MMR perkalimat (Si) sampai didapatkan hasil kalimat – kalimat yang konvergen.

```
function TMMR.GetKonvergen(idx: Integer): Boolean;
var
  i: Integer;
  stemp : string;
begin
  Result:= false;
  stemp:= Self.GetStringListKal;
  for i:= 0 to Self.MaxLoop do
    if Trim(Self.LDtKal[i]) = " then
      Break else
    if Trim(Self.LDtKal[i]) = Trim(stemp) then
      begin
        Result:= true;
        Break;
      end;
    if Result = false then
      Self.LDtKal[idx]:= stemp;
  end;
```

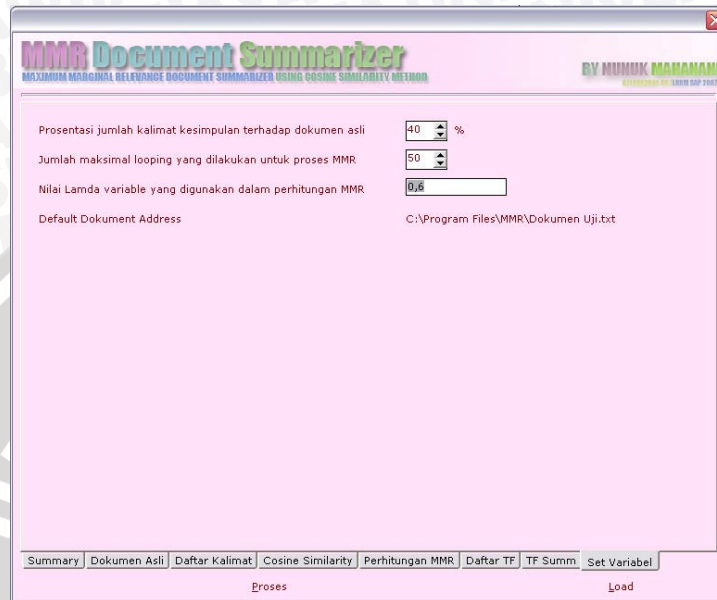
Pseudocode 4.8 Prosedur proses Get Konvergen

4.3 Implementasi Antar Muka

Antarmuka sistem peringkasan dokumen ini terdiri dari delapan bagian seperti yang ditunjukkan pada Gambar 4.1 sampai dengan Gambar 4.8.

Gambar 4.1 adalah antarmuka proses set variabel. Didalam antarmuka ini, pengguna memasukkan teks dokumen yang akan dibuat ringkasannya dengan cara mengakses file teks

(dengan format Text Dokumen) yang telah disimpan dengan menekan tombol Load. Kemudian pengguna mengisikan beberapa variabel, antara lain : variabel prosentase jumlah kalimat kesimpulan yang diinginkan dan jumlah maksimal looping yang akan dilakukan pada proses perhitungan MMR.



Gambar 4.1 Antarmuka Proses Set Variabel

Setelah pengguna memasukkan dokumen asal dan variabel yang dibutuhkan maka secara otomatis dokumen asal akan terpecah berdasarkan posisi kalimatnya dan dapat diketahui frekuensi kata perkaliat. Gambar 4.2, Gambar 4.3 dan Gambar 4.4 adalah antarmuka dokumen asal, antarmuka pemecahan dokumen asal menjadi kalimat, dan antarmuka daftar term frekuensi hasil pemecahan kalimat-kalimat dokumen asal menjadi kata-kata.



Gambar 4.2 Antarmuka Dokumen Asal



Gambar 4.3 Antarmuka Pemecahan Dokumen Asal Menjadi Kalimat

MMR Document Summarizer
MAXIMUM MATHSAL BELAJARAN DOCUMENT SUMMARIZER USING COSINE SIMILARITY METHOD

Term	DA	Kal 1	Kal 2	Kal 3	Kal 4	Kal 5	Kal 6	Kal 7	Kal 8	Kal 9
amerika	1	1	0	0	0	0	0	0	0	0
angka	1	0	0	1	0	0	0	0	0	0
barat	1	0	0	0	0	1	0	0	0	0
belakang	1	1	0	0	0	0	0	0	0	0
berbagai	1	0	0	0	0	0	1	0	0	0
berwajah	2	0	1	0	0	0	0	0	1	0
besi	2	0	0	0	0	0	1	0	1	0
brant	1	0	0	0	0	0	0	1	0	0
country	1	0	0	0	1	0	0	0	0	0
dibebaskan	1	0	0	0	0	0	0	1	0	0
dikensi	1	0	0	0	0	0	0	1	0	0
dikenakan	1	0	0	0	0	0	0	0	0	1
dikawatarkan	1	0	0	1	0	0	0	0	0	0
dikremasi	1	0	0	0	0	0	0	0	1	0
ditemukan	2	1	0	1	0	0	0	0	0	0
dokter	1	0	0	0	1	0	0	0	0	0
dollar	1	0	0	0	0	0	0	1	0	0
forensik	1	0	0	0	1	0	0	0	0	0
georgia	2	1	0	0	1	0	0	0	0	0
gulfang	1	1	0	0	0	0	0	0	0	0
han	2	0	1	0	0	0	0	1	0	0
hektar	1	0	0	0	0	0	0	0	1	0
hutan	2	1	0	0	0	0	0	0	1	0
intensif	1	0	0	0	0	1	0	0	0	0
jenazah	9	1	1	1	1	0	1	0	4	0
jumlah	1	0	0	1	0	0	0	0	0	0
kata	1	0	0	0	1	0	0	0	0	0
kilometer	1	0	0	0	0	1	0	0	0	0
kota	1	0	0	0	0	1	0	0	0	0
krematorium	7	1	0	1	1	1	1	1	1	0
kris	1	0	0	0	1	0	0	0	0	0
laut	1	0	0	0	0	1	0	0	0	0

Buttons: Summary, Dokumen Asli, Daftar kalimat, Cosine Similarity, Perhitungan MMR, Daftar TF, TF Summ, Set Variabel, Proses, Load

Gambar 4.4 Antarmuka Daftar TF

Proses pembentukan summary dilakukan saat pengguna menekan tombol proses. Ketika tombol proses ditekan, maka proses yang berjalan adalah proses perhitungan similiarity perkaliat (Si) terhadap dokumen asal (DA) dan dokumen ringkasan latih (Si), dan proses pembentukan summary. Gambar 4.5, Gambar 4.6 dan Gambar 4.7 adalah antarmuka daftar nilai Cosine Similiarity perkaliat (Si), antarmuka daftar perhitungan MMR per kaliat (Si) dan antarmuka summary sistem yang dihasilkan.

MMR Document Summarizer
MAXIMUM MATHSAL BELAJARAN DOCUMENT SUMMARIZER USING COSINE SIMILARITY METHOD

Index Kalimat	DA / Summary	Dot Produk : Vektor Jarak	Cosine Similarity
Kalimat DA ke (1)	DA	1,178047757762935130 ; 2,907983663147204020	0,405108107274570195
Kalimat DA ke (1)	Summary	0,000000000000000000 ; 0,860843482143654581	0,000000000000000000
Kalimat DA ke (2)	DA	0,000000000000000000 ; 0,000000000000000000	-100,0000000000000000
Kalimat DA ke (2)	Summary	0,000000000000000000 ; 0,000000000000000000	100,0000000000000000
Kalimat DA ke (3)	DA	0,000000000000000000 ; 0,000000000000000000	-100,0000000000000000
Kalimat DA ke (3)	Summary	0,000000000000000000 ; 0,000000000000000000	100,0000000000000000
Kalimat DA ke (4)	DA	0,815871524605108955 ; 2,600979659302399050	0,313563207496920959
Kalimat DA ke (4)	Summary	0,000000000000000000 ; 0,769961817624335132	0,000000000000000000
Kalimat DA ke (5)	DA	1,812381165789131110 ; 3,678340709642005100	0,492717045225949502
Kalimat DA ke (5)	Summary	0,000000000000000000 ; 1,088990444993774410	0,000000000000000000
Kalimat DA ke (6)	DA	0,000000000000000000 ; 0,000000000000000000	-100,0000000000000000
Kalimat DA ke (6)	Summary	0,000000000000000000 ; 0,000000000000000000	100,0000000000000000
Kalimat DA ke (7)	DA	0,000000000000000000 ; 0,000000000000000000	-100,0000000000000000
Kalimat DA ke (7)	Summary	0,000000000000000000 ; 0,000000000000000000	100,0000000000000000
Kalimat DA ke (8)	DA	2,621139270275854470 ; 4,313236808626240480	0,607696583008089585
Kalimat DA ke (8)	Summary	0,287255618478906516 ; 1,276837225979869490	0,224974344915762153
Kalimat DA ke (9)	DA	0,234246867528909802 ; 1,300489826651199520	0,180122029552308355
Kalimat DA ke (9)	Summary	0,000000000000000000 ; 0,384980908812167565	0,000000000000000000
Kalimat DA ke (1)	DA	0,000000000000000000 ; 0,000000000000000000	-100,0000000000000000
Kalimat DA ke (1)	Summary	0,000000000000000000 ; 0,000000000000000000	100,0000000000000000
Kalimat DA ke (2)	DA	0,63433408026195755 ; 1,91320066405059910	0,331556235402391405
Kalimat DA ke (2)	Summary	0,000000000000000000 ; 0,875823497202266599	0,000000000000000000
Kalimat DA ke (3)	DA	0,453095291447282722 ; 1,91320066405059910	0,236825881100170832
Kalimat DA ke (3)	Summary	0,000000000000000000 ; 0,875823497202266599	0,000000000000000000
Kalimat DA ke (4)	DA	0,000000000000000000 ; 0,000000000000000000	-100,0000000000000000
Kalimat DA ke (4)	Summary	0,000000000000000000 ; 0,000000000000000000	100,0000000000000000
Kalimat DA ke (5)	DA	0,000000000000000000 ; 0,000000000000000000	-100,0000000000000000
Kalimat DA ke (5)	Summary	0,000000000000000000 ; 0,000000000000000000	100,0000000000000000
Kalimat DA ke (6)	DA	0,906190582894565555 ; 2,566827241823298810	0,353039179314176499
Kalimat DA ke (6)	Summary	0,000000000000000000 ; 1,175040525621518640	0,000000000000000000
Kalimat DA ke (7)	DA	1,74770858449617310 ; 3,42243632241053670	0,518569431611513920
Kalimat DA ke (7)	Summary	0,000000000000000000 ; 1,566720700828692800	0,000000000000000000

Buttons: Summary, Dokumen Asli, Daftar Kalimat, Cosine Similarity, Perhitungan MMR, Daftar TF, TF Summ, Set Variabel, Proses, Load

Gambar 4.5 Antarmuka Daftar Nilai Cosine Similiarity

coba untuk mencari nilai koefisien lamda (λ) yang paling optimal. Data hasil ringkasan ideal diperoleh dari hasil ringkasan dokumen berita yang diberikan pada 1 orang *abstractor* seorang ahli bahasa Indonesia yang diberikan 15 dokumen untuk dibuat ringkasannya.

Tabel 4.1 menunjukkan hasil ringkasan ideal yang telah dibuat oleh *abstractor*.

Tabel 4.1 Tabel ringkasan ideal

No Dokumen	Jumlah Kalimat Doc	Jumlah Kalimat Ringkasan	No Kalimat Ringkasan
Doc 1	29	11	1,5,7,15,12,16,18,19,23,24,27
Doc 2	46	18	1,3,4,6,8,12,14,18,20,24,27,29,30,31,36,40,43,44
Doc 3	45	15	1,4,5,10,9,11,15,16,18,23,27,30,36,42,45
Doc 4	20	9	1,3,6,7,9,16,17,18,20
Doc 5	35	9	1,5,8,12,14,17,29,33,34
Doc 6	64	17	1,4,5,9,18,25,28,29,33,37,39,46,52,58,60,63,64
Doc 7	36	9	2,3,4,10,13,24,28,33,35
Doc 8	14	4	1,4,8,13
Doc 9	59	10	1,3,5,8,19,22,25,28,32,34
Doc 10	50	12	1,3,5,7,10,16,22,24,31,35,46,50
Doc 11	51	12	5,7,12,49,19,21,30,39,41,44,47,50
Doc 12	37	12	1,5,6,10,13,16,22,23,29,33,34,36
Doc 13	54	12	1,10,15,19,23,27,29,31,34,47,51,55
Doc 14	47	12	1,2,3,5,8,10,12,23,24,35,39,41
Doc 15	30	9	1,5,10,13,17,19,20,26,28

Sesuai dengan subbab batasan sistem yang telah dijelaskan pada bab 3 maka nilai koefisien lamda (λ) yang digunakan adalah 0,6.

Tabel 4.2 menunjukkan hasil ringkasan yang dibuat oleh sistem peringkas (MMR).

Tabel 4.2 Tabel Hasil Sistem MMR

No Dokumen	Jumlah Kalimat ringkasan	No Kalimat
Doc 1	12	15,24,16,19,15,1,5,23,13,26,17,7
Doc 2	14	12,18,1,6,32,29,3,30,27,8,15,36,24,23
Doc 3	14	9,1,4,29,10,39,18,22,8,19,42,11,12,5
Doc 4	8	20,3,16,18,12,15,8,7
Doc 5	10	9,14,33,29,15,8,20,5,32,7
Doc 6	16	26,22,5,3,28,25,29,1,37,48,27,9,50,34,39,18
Doc 7	11	10,22,2,36,35,28,17,27,24,26,4
Doc 8	6	3,1,7,12,8,11
Doc 9	15	26,29,45,3,19,20,10,27,21,9,28,25,5,31,5
Doc 10	12	38,7,50,10,24,39,41,31,28,26,46,40
Doc 11	13	44,5,34,7,12,19,11,49,9,21,3,31,50
Doc 12	11	15,12,31,5,8,16,6,11,1,23,33
Doc 13	13	2,6,10,27,29,31,33,40,47,48,51,54,55
Doc 14	14	15,1,12,35,43,28,11,5,2,39,24,16,34,18
Doc 15	9	11,13,14,29,17,1,20,23,6

Setelah diketahui hasil ringkasan yang diperoleh dari ringkasan ideal dan ringkasan sistem (MMR) maka dapat dihitung *precision*, *recall* dan *f-measure*(β) sistem peringkas (MMR) terhadap ringkasan ideal.

Berdasarkan batasan sistem yang telah dijelaskan pada bab II dan bab III untuk perhitungan evaluasi *f-measure*(β) digunakan nilai koefisien $\beta = 0,9$.

Tabel 4.3 menunjukkan hasil perhitungan *precision*, *recall*, dan *f-measure*(β) sistem peringkas (MMR).

Tabel 4.3 Tabel Hasil Pembandingan Sistem (MMR)

No Doc	Overlap MMR	Precision	Recall	F-measure(β)
--------	-------------	-----------	--------	----------------------

Doc 1	24,16,19,1,5,23,7	0,67	0,73	0,78
Doc 2	18,1,6,32,3,30,27,8,36	0,87	0,67	0,85
Doc 3	9,1,4,18,19,42,11,5	0,64	0,60	0,69
Doc 4	20,3,16,18,7	0,63	0,56	0,66
Doc 5	14,33,29,8,5	0,50	0,56	0,59
Doc 6	5,28,25,29,1,37,9,39,18	0,56	0,53	0,61
Doc 7	2,35,28,24,4	0,54	0,67	0,67
Doc 8	1,8	0,30	0,50	0,42
Doc 9	3,19,28,25	0,33	0,50	0,44
Doc 10	7,50,10,24,31,46	0,50	0,50	0,56
Doc 11	44,5,7,19,49,21,50	0,61	0,67	0,71
Doc 12	5,16,6,1,23,33	0,54	0,50	0,58
Doc 13	10,27,29,31,47,51,55	0,54	0,58	0,62
Doc 14	1,12,35,5,2,39,24	0,50	0,58	0,60
Doc 15	13,12,17,1,20	0,56	0,56	0,63

Dari 15 nilai *precision*, *recall*, dan *f-measure*(β) yang berbeda – beda tersebut, masing-masing dihitung nilai rata-ratanya. Tabel 4.4 menunjukkan hasil akhir nilai *recall*, *precision* dan *f-measure* sistem (MMR).

Tabel 4.4 Tabel Hasil akhir *recall*, *precision* dan *f-measure*(β) sistem (MMR)

	<i>Precision</i>	<i>Recall</i>	<i>F-measure</i> (β)
Rata-rata	0,55	0,58	0,63

Dengan cara yang sama, yaitu dengan membandingkan hasil ringkasan sistem (MMR) dengan hasil ringkasan ideal maka dapat dihitung nilai koefisien lamda (λ) yang paling optimal. Batasannya yaitu nilai lamda (λ) tersebut mempunyai hasil akhir *precision*, *recall* dan *f-measure* yang paling baik terhadap sistem peringkas otomatis. Hasil perhitungan yang lebih lengkap disajikan pada lampiran I. Tabel 4.5 menunjukkan hasil akhir *precision*, *recall* dan *f-measure* nilai koefisien lamda (λ) antara range [0 – 1].

Tabel 4.5 Tabel Hasil akhir *precision*, *recall* dan *f-measure* nilai koefisien lamda (λ) antara [0 – 1]

Lamda	<i>precision</i>	<i>recall</i>	<i>f-measure</i>
0,1	0,308	0,312	0,307
0,2	0,355	0,353	0,351
0,3	0,315	0,309	0,309
0,4	0,324	0,321	0,320
0,5	0,355	0,354	0,352
0,6	0,55	0,58	0,56
0,7	0,331	0,324	0,339
0,8	0,321	0,328	0,321
0,9	0,381	0,381	0,377

4.4.2 Perbandingan Sistem

Untuk pengujian sistem peringkas (*shortestpath*) maka data yang sama yaitu 15 buah dokumen berita diuji pada software sistem peringkas dengan metode *shortestpath* yang sudah ada. Tabel 4.6 menunjukkan hasil kalimat ringkasan pada sistem peringkas (*shortest path*).

Tabel 4.6 Tabel Hasil Sistem Peringkasan (*shortest path*)

No Dokumen	Jumlah Kalimat ringkasan	No Kalimat
Doc 1	9	1,12,15,16,18,19,23,26,27
Doc 2	14	1,5,8,9,10,12,14,18,23,24,29,30,36,45
Doc 3	15	1,2,8,10,11,12,18,27,29,30,34,35,37,41,46
Doc 4	9	1,3,8,10,11,12,15,16,20
Doc 5	7	1,14,15,19,20,32,33
Doc 6	20	1,3,4,5,9,10,12,22,26,27,29,31,33,34,35,37,42,49,50,64
Doc 7	12	1,2,3,9,10,13,16,17,18,24,26,36
Doc 8	9	1,2,3,4,5,7,11,13,14
Doc 9	18	1,5,8,10,13,16,21,25,27,28,29,32,35,45,47,55,57,59
Doc 10	18	1,3,4,5,6,8,10,16,28,29,35,37,39,40,46,48,49,50
Doc 11	12	1,5,9,12,21,24,28,44,49,50,51,29
Doc 12	12	1,3,4,5,16,19,22,23,24,32,34,37
Doc 13	11	2,4,10,15,27,29,34,38,40,54,56
Doc 14	15	1,3,5,12,15,16,17,18,21,22,35,43,46,47
Doc 15	10	1,2,13,16,17,19,23,24,28,29

Dengan cara yang sama dilakukan pengujian hasil ringkasan sistem peringkasan (*shortest path*) dengan hasil ringkasan ideal yang menghasilkan nilai *precision*, *recall*, dan *f-measure*(β) perdokumen. Tabel 4.7 menunjukkan hasil perhitungan *precision*, *recall*, dan *f-measure*(β) sistem peringkasan (*shortest path*).

Tabel 4.7 Tabel Hasil Pembanding Sistem (*shortest path*)

No Doc	Overlap <i>shortest path</i>	Precision	Recall	F-measure(β)
--------	------------------------------	-----------	--------	----------------------

Doc 1	1,12,15,16,18,19,23,27	0,89	0,73	0,90
Doc 2	1,8,14,18,24,29,30,36	0,64	0,50	0,63
Doc 3	1,10,11,18,27,30	0,40	0,40	0,45
Doc 4	1,3,16,20	0,44	0,44	0,49
Doc 5	1,14,33	0,43	0,33	0,42
Doc 6	1,4,5,9,29,33,37,64	0,40	0,47	0,48
Doc 7	2,3,10,13,24	0,42	0,55	0,53
Doc 8	1,4,13	0,30	0,75	0,48
Doc 9	1,5,8,25,28	0,28	0,50	0,40
Doc 10	1,3,5,10,16,35,46,50	0,44	0,67	0,59
Doc 11	5,12,21,44,49,50	0,50	0,50	0,56
Doc 12	1,5,16,22,23,34	0,50	0,50	0,56
Doc 13	10,15,27,29,34	0,45	0,42	0,49
Doc 14	1,3,5,12,35	0,30	0,42	0,39
Doc 15	1,12,13,17,19,28	0,60	0,67	0,71

Setelah diperoleh nilai *precision*, *recall*, dan *f-measure*(β) perdokumen maka hasil akhirnya dapat diketahui pada Tabel 4.8.

Tabel 4.8 Tabel Hasil akhir *recall*, *precision* dan *f-measure*(β) sistem (*shortest path*)

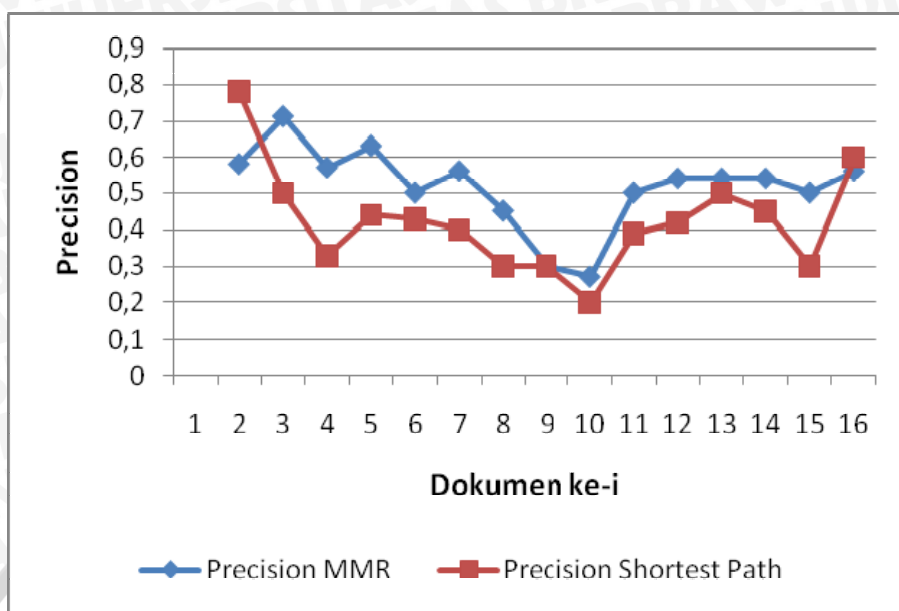
	<i>Recall</i>	<i>Precision</i>	<i>F-measure</i> (β)
Rata-rata	0,47	0,52	0,54

Tabel 4.9 menunjukkan hasil pengukuran *recall*, *precision* dan *f-measure*(β) antara dua sistem peringkasan otomatis (MMR dan *shortest path*).

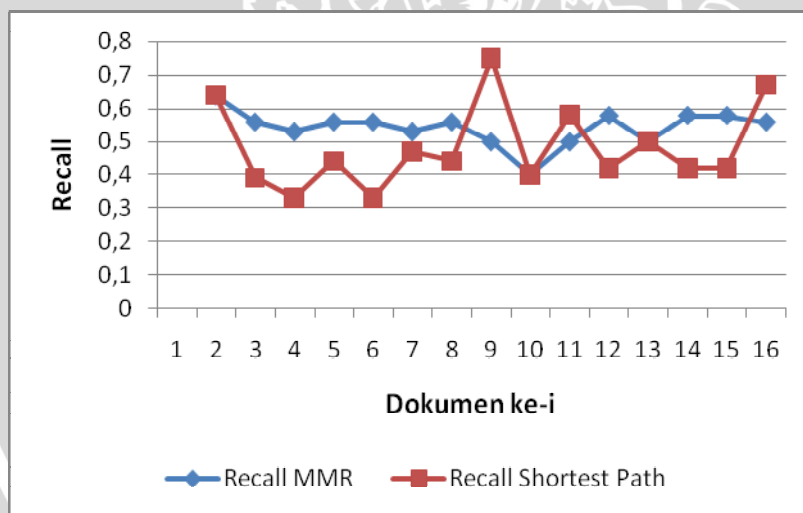
Tabel 4.9 Tabel Hasil Pengukuran *recall*, *precision* dan *f-measure*(β) Antara dua Sistem Peringkasan Otomatis

Precision MMR	Precision Shortest Path	Recall MMR	Recall Shortest Path	F-measure MMR	F-measure Shortest Path
0,67	0,89	0,73	0,73	0,78	0,90
0,87	0,64	0,67	0,50	0,85	0,63
0,64	0,40	0,60	0,40	0,69	0,45
0,63	0,44	0,56	0,44	0,66	0,49
0,50	0,43	0,56	0,33	0,59	0,42
0,56	0,40	0,53	0,47	0,61	0,48
0,54	0,42	0,67	0,55	0,67	0,53
0,30	0,30	0,50	0,75	0,42	0,48
0,33	0,28	0,50	0,50	0,44	0,40
0,50	0,44	0,50	0,67	0,56	0,59
0,61	0,50	0,67	0,50	0,71	0,56
0,54	0,50	0,50	0,50	0,58	0,56
0,54	0,45	0,58	0,42	0,62	0,49
0,50	0,30	0,58	0,42	0,60	0,39
0,56	0,60	0,56	0,67	0,63	0,71

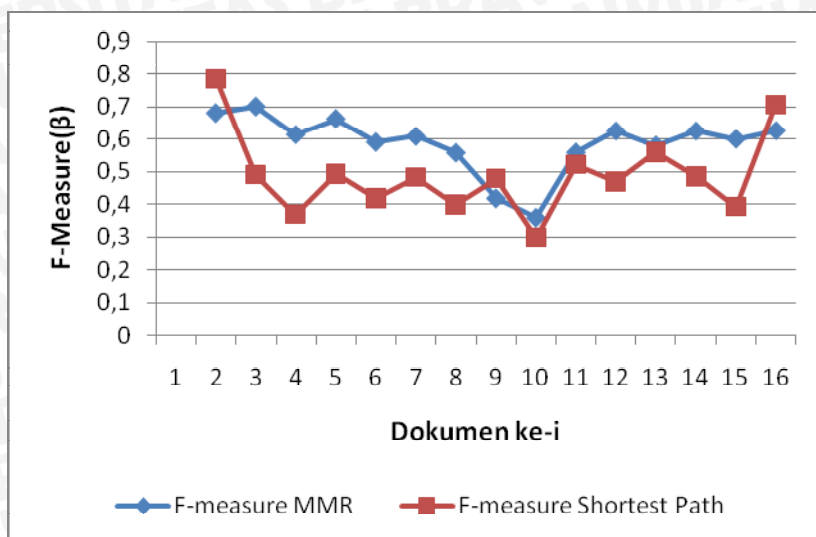
Gambar 4.8, gambar 4.9 dan gambar 4.10 menunjukkan grafik *precision*, *recall* dan *f-measure* sistem peringkasan otomatis MMR dan *shortest path*(β).



Gambar 4.8 Grafik *precision* sistem peringkas otomatis MMR dan *shortest path*



Gambar 4.9 Grafik *recall* sistem peringkas otomatis MMR dan *shortest path*



Gambar 4.10 Grafik f -measure(β) sistem peringkasan otomatis MMR dan *shortest path*

Selain itu juga dilakukan perhitungan peluang pemilihan kalimat-kalimat yang sama antara dua sistem peringkasan ini, yang hasilnya ditunjukkan pada Tabel 4.10.

Tabel 4.10 Hasil Pengukuran peluang pemilihan kalimat yang sama

Doc ke-	Overlap Kalimat	Σ Kalimat Sama	Jumlah Kalimat	Peluang
D1	15, 16, 19, 1, 23, 26	7	12	0,58
D2	12, 18, 1, 29, 30, 8, 36, 24, 23	9	14	0,64
D3	1, 29, 10, 18, 8, 11, 12	7	14	0,50
D4	20, 3, 16, 12, 15, 8	6	8	0,75
D5	14, 33, 15, 20, 32	5	10	0,50
D6	26, 22, 5, 3, 29, 1, 37, 27, 9, 50, 34	11	16	0,69
D7	10, 2, 36, 17, 26, 16	5	11	0,45
D8	3, 1, 11	3	6	0,50
D9	29, 45, 10, 27, 21, 28, 25, 5, 13	9	15	0,60

D10	50, 10, 39, 28, 46, 40	6	12	0,50
D11	44, 5, 12, 49, 9, 21, 50	7	13	0,54
D12	5, 16, 1, 23	4	11	0,36
D13	2, 10, 27, 29, 40, 54	6	13	0,46
D14	15, 1, 12, 35, 43, 28, 5, 16, 18	9	14	0,64
D15	13, 12, 29, 17, 1, 23	6	9	0,67

Dari nilai hasil pengukuran peluang pemilihan kalimat yang sama didapatkan hasil rata-rata peluang sebesar 0,56.

4.4.3 Analisa Hasil

Dari implementasi uji coba yang telah dilakukan pada sistem peringkasan dokumen berita otomatis dengan metode MMR diperoleh nilai rata-rata *recall* sebesar 55%, *precision* sebesar 58%, dan *f-measure*(β) sebesar 63%. Sesuai dengan hasil nilai *recall*, *precision* dan *f-measure*(β) yang diperoleh tersebut, penulis belum mendapatkan informasi apakah hasil ringkasan dengan sistem ini sudah baik atau belum. Seperti yang disebutkan pada penelitian yang dilakukan oleh Edmudson (Edmudson, 1969), sulit untuk mengambil kesimpulan terhadap *performance* sistem dari nilai *precision* dan *recall*, dikarenakan nilai-nilai tersebut besarnya relatif terhadap ringkasan ideal, padahal ringkasan yang dibuat manusia pun bersifat subjektif, berbeda antara orang yang satu dengan orang yang lain, dan tidak ada satupun ringkasan yang benar. Meskipun demikian untuk mengantisipasi hasil yang meragukan maka pada penelitian ini untuk mendapatkan ringkasan ideal, ringkasan 1 teks dibuat oleh seorang tenaga ahli dalam bahasa Indonesia.

Sedangkan untuk perbandingan sistem peringkasan dokumen ini dengan sistem peringkasan dokumen dengan metode *shortest path* mempunyai hasil rata-rata *precision*, *recall* dan *f-measure*(β) lebih baik. Sistem peringkasan dokumen (*shortest path*) mempunyai nilai rata-rata *recall* sebesar 47%, *precision* 52%, dan *f-measure*(β) sebesar 54%. Nilai diatas diperoleh dengan membandingkan hasil ringkasan dua sistem peringkasan otomatis dengan ringkasan ideal yang mana mempunyai data isi dokumen yang sama. Berdasarkan grafik *f-*

$measure(\beta)$ yang telah dibuat bisa dilihat bahwa nilai $f-measure(\beta)$ ringkasan MMR memiliki sebaran lebih stabil dibandingkan $f-measure(\beta)$ ringkasan shortest path.

Nilai koefisien lamda (λ) yang digunakan paling optimal pada sistem peringkasan otomatis (MMR) adalah sebesar 0,6 dengan nilai rata-rata *recall* sebesar 55%, *precision* 58%, dan *f-measure* sebesar 56%. Nilai tersebut diperoleh dengan ujicoba nilai lamda dengan range [0-1] terhadap ringkasan sistem.

Dengan tidak melibatkan ringkasan ideal, dihitung peluang kesamaan kedua sistem dalam pemilihan kalimat yang akan masuk dalam ringkasan, didapatkan besar rata-rata peluang pemilihan kalimat yang sama sebesar 56%. Hal ini berarti sebagian besar kedua sistem memilih kalimat yang sama sebanyak setengah dari jumlah kalimat yang terdapat dalam ringkasan.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang didapat pada Tugas Akhir ini adalah :

1. Peringkasan dokumen dengan menggunakan metode Maximal Marginal Relevance Document (*MMR-D*) ini menghasilkan nilai rata - rata *recall* sebesar 55%, nilai rata-rata *precision* 58% dan nilai rata-rata *f-measure* sebesar 63%.
2. Setelah dilakukan perbandingan dengan sistem peringkasan yang lain, yaitu sistem peringkasan dengan menggabungkan dua metode yaitu pembobotan kalimat dan algoritma *shortests path*, dihasilkan kesimpulan bahwa sistem peringkasan dokumen ini lebih baik. Dengan data dokumen uji coba yang sama sistem peringkasan *shortests path* menghasilkan nilai rata - rata *recall* sebesar 47%, nilai rata – rata *precision* 53%, dan nilai rata – rata *f-measure* sebesar 54%.

5.2 Saran

Beberapa saran pengembangan lebih lanjut yang dapat diberikan oleh penulis adalah :

1. Dokumen yang akan diringkaskan tidak sebatas dokumen yang memiliki judul saja, tetapi juga yang terdiri dari bab, subbab-subbab.
2. Dilakukan proses *stemming* pada kata yang ada dalam dokumen, supaya untuk fitur frekuensi dapat lebih akurat.
3. Pembuatan ringkasan ideal, diserahkan kepada orang yang benar-benar ahli.

DAFTAR PUSTAKA

Carbonell, Goldstein, 1998. The use of MMR, diversity based reranking for reordering documents and producing summaries. *Proceedings of the 21st ACM SIGIR*.

H.P. Edmudson. 1969. New Methods in Automatic Extracting. *Journal of Assosiation for Computing Machinery*, 16(2):264-285.

Mani, I. 2001. Summarization evaluation: an overview. *Proceedings of the NTCIR Workshop 2 Meeting on Evaluation of Chinese and Japanese Text Retrieval and Text Summarization*.

Mani, I. and M.T. Maybury. 1999. Advance in Automatic Text Summarization. The MIT, Press :Cambridge.

Gracia, E. 2004. The Term Count Model, <http://www.miislita.com/term-vector/term-vector-2.html> , tanggal akses : 17 Maret 2009.

Gracia, E. 2004. Term Vector Calculation : A Fast Track Tutorial, <http://www.miislita.com/information-retrieval-tutorial/term-vector-fast-track.pdf> , tanggal akses : 17 Maret 2009.

J. Kupiec, J. Pederson, and F. Chen, "A trainable document summarizer," in *ACM SIGIR '95*, 1995, pp. 68–73.

Gabriel Murray, Steve Renals, and Jean Carletta, "Extractive summarization of meeting recordings," in *Proceedings of Interspeech*, September 2005.

Shasha Xie, Yang Liu, "Using Corpus and Knowledge-Based imiliarity Measure In Maximum Marginal Relevance For Meeting Summarization", in *Proceedings of Interspeech*, May 2007.

Zhou Yong, Li Youwen Xia, and Shixiong, "An Improved KNN Text Clasification Algorithm Based on Clustering", in *Journal of Computers*, VOL. 4, NO. 3, March 2009.

http://en.wikipedia.org/wiki/Precision_and_recall , tanggal akses : 14 September 2009.

http://www.itttelkom.ac.id/library/index.php?view=article&catid=20%3Ainformatika&id=540%3Ainformation-retrieval&option=com_content&Itemid=15 , tanggal akses : 14 September 2009.

<http://www.jawaposonline.com> , tanggal akses : 10 November 2009.

<http://www.seputar-indonesia.com/ediscetak/content/view/280916/38/> , tanggal akses : 10 November 2009.



Lampiran I

Jumlah kalimat ringkasan

	doc MMR	doc ideal
doc 1	12	11
doc 2	14	18
doc 3	14	15
doc 4	8	9
doc 5	10	11
doc 6	16	23
doc 7	11	10
doc 8	6	4
doc 9	15	14
doc 10	12	12
doc 11	13	13
doc 12	11	12
doc 13	13	12
doc 14	14	12
doc 15	9	9

Perhitungan Lamda (λ)

Lamda = 0,1	overlap	precision	recall	f-measure
doc 1	3	0,250	0,273	0,261
doc 2	5	0,357	0,278	0,313
doc 3	6	0,429	0,400	0,414
doc 4	4	0,500	0,444	0,471
doc 5	2	0,200	0,182	0,190
doc 6	4	0,250	0,174	0,205
doc 7	1	0,091	0,100	0,095
doc 8	3	0,500	0,750	0,600
doc 9	3	0,200	0,214	0,207
doc 10	3	0,250	0,250	0,250

doc 11	3	0,231	0,231	0,231
doc 12	6	0,545	0,500	0,522
doc 13	3	0,231	0,250	0,240
doc 14	5	0,357	0,417	0,385
doc 15	2	0,222	0,222	0,222
	rata2	0,308	0,312	0,307

Lamda = 0,2	overlap	precision	recall	f-measure
doc 1	6	0,500	0,545	0,522
doc 2	6	0,429	0,333	0,375
doc 3	7	0,500	0,467	0,483
doc 4	2	0,250	0,222	0,235
doc 5	4	0,400	0,364	0,381
doc 6	10	0,625	0,435	0,513
doc 7	3	0,273	0,300	0,286
doc 8	3	0,500	0,750	0,600
doc 9	3	0,200	0,214	0,207
doc 10	4	0,333	0,333	0,333
doc 11	4	0,308	0,308	0,308
doc 12	5	0,455	0,417	0,435
doc 13	3	0,231	0,250	0,240
doc 14	3	0,214	0,250	0,231
doc 15	1	0,111	0,111	0,111
	rata2	0,355	0,353	0,351

Lamda = 0,3	overlap	precision	recall	f-measure
doc 1	3	0,250	0,273	0,261
doc 2	5	0,357	0,278	0,313
doc 3	8	0,571	0,533	0,552
doc 4	2	0,250	0,222	0,235
doc 5	3	0,300	0,273	0,286
doc 6	9	0,563	0,391	0,462
doc 7	1	0,091	0,100	0,095

doc 8	2	0,333	0,500	0,400
doc 9	4	0,267	0,286	0,276
doc 10	3	0,250	0,250	0,250
doc 11	3	0,231	0,231	0,231
doc 12	5	0,455	0,417	0,435
doc 13	2	0,154	0,167	0,160
doc 14	6	0,429	0,500	0,462
doc 15	2	0,222	0,222	0,222
	rata2	0,315	0,309	0,309

Lamda = 0,4	overlap	precision	recall	f-measure
doc 1	3	0,250	0,273	0,261
doc 2	6	0,429	0,333	0,375
doc 3	6	0,429	0,400	0,414
doc 4	3	0,375	0,333	0,353
doc 5	2	0,200	0,182	0,190
doc 6	6	0,375	0,261	0,308
doc 7	4	0,364	0,400	0,381
doc 8	2	0,333	0,500	0,400
doc 9	4	0,267	0,286	0,276
doc 10	5	0,417	0,417	0,417
doc 11	6	0,462	0,462	0,462
doc 12	4	0,364	0,333	0,348
doc 13	2	0,154	0,167	0,160
doc 14	3	0,214	0,250	0,231
doc 15	2	0,222	0,222	0,222
	rata2	0,324	0,321	0,320

Lamda = 0,5	overlap	precision	recall	f-measure
doc 1	4	0,333	0,364	0,348
doc 2	6	0,429	0,333	0,375
doc 3	7	0,500	0,467	0,483
doc 4	3	0,375	0,333	0,353

doc 5	4	0,400	0,364	0,381
doc 6	5	0,313	0,217	0,256
doc 7	3	0,273	0,300	0,286
doc 8	2	0,333	0,500	0,400
doc 9	4	0,267	0,286	0,276
doc 10	6	0,500	0,500	0,500
doc 11	4	0,308	0,308	0,308
doc 12	5	0,455	0,417	0,435
doc 13	2	0,154	0,167	0,160
doc 14	5	0,357	0,417	0,385
doc 15	3	0,333	0,333	0,333
	rata2	0,355	0,354	0,352

Lamda = 0,6	overlap	precision	recall	f-measure
doc 1	7	0,583	0,636	0,609
doc 2	7	0,500	0,389	0,438
doc 3	7	0,500	0,467	0,483
doc 4	4	0,500	0,444	0,471
doc 5	5	0,500	0,455	0,476
doc 6	9	0,563	0,391	0,462
doc 7	3	0,273	0,300	0,286
doc 8	2	0,333	0,500	0,400
doc 9	4	0,267	0,286	0,276
doc 10	4	0,333	0,333	0,333
doc 11	5	0,385	0,385	0,385
doc 12	6	0,545	0,500	0,522
doc 13	7	0,538	0,583	0,560
doc 14	7	0,500	0,583	0,538
doc 15	4	0,444	0,444	0,444
	rata2	0,451	0,446	0,445

Lamda = 0,7	overlap	precision	recall	f-measure
-------------	---------	-----------	--------	-----------

doc 1	5	0,417	0,455	0,435
doc 2	6	0,429	0,333	0,375
doc 3	4	0,286	0,267	0,276
doc 4	4	0,500	0,444	0,471
doc 5	4	0,400	0,364	0,381
doc 6	7	0,438	0,304	0,359
doc 7	3	0,273	0,300	0,286
doc 8	3	0,500	0,750	0,600
doc 9	4	0,267	0,286	0,276
doc 10	3	0,250	0,250	0,250
doc 11	4	0,308	0,308	0,308
doc 12	3	0,273	0,250	0,261
doc 13	3	0,231	0,250	0,240
doc 14	4	0,286	0,333	0,308
doc 15	1	0,111	0,111	0,111
	rata2	0,331	0,334	0,329

Lamda = 0,8	overlap	precision	recall	f-measure
doc 1	4	0,333	0,364	0,348
doc 2	3	0,214	0,167	0,188
doc 3	5	0,357	0,333	0,345
doc 4	4	0,500	0,444	0,471
doc 5	4	0,400	0,364	0,381
doc 6	5	0,313	0,217	0,256
doc 7	3	0,273	0,300	0,286
doc 8	3	0,500	0,750	0,600
doc 9	4	0,267	0,286	0,276
doc 10	3	0,250	0,250	0,250
doc 11	5	0,385	0,385	0,385
doc 12	4	0,364	0,333	0,348
doc 13	2	0,154	0,167	0,160
doc 14	4	0,286	0,333	0,308
doc 15	2	0,222	0,222	0,222

	rata2	0,321	0,328	0,321
--	-------	-------	-------	-------

Lamda = 0,9	overlap	precision	recall	f-measure
doc 1	4	0,333	0,364	0,348
doc 2	5	0,357	0,278	0,313
doc 3	2	0,143	0,133	0,138
doc 4	5	0,625	0,556	0,588
doc 5	2	0,200	0,182	0,190
doc 6	10	0,625	0,435	0,513
doc 7	5	0,455	0,500	0,476
doc 8	3	0,500	0,750	0,600
doc 9	3	0,200	0,214	0,207
doc 10	5	0,417	0,417	0,417
doc 11	5	0,385	0,385	0,385
doc 12	6	0,545	0,500	0,522
doc 13	4	0,308	0,333	0,320
doc 14	4	0,286	0,333	0,308
doc 15	3	0,333	0,333	0,333
	rata2	0,381	0,381	0,377

Lampiran 2

Daftar Kata Stop Word

1	ada	36	Apalah	71	berapalah
2	adalah	37	Apatah	72	berapapun
3	adanya	38	Atau	73	berkali-kali
4	adapun	39	ataukah	74	berlebihan
5	aduh	40	ataupun	75	berlebihhankah
6	agak	41	bagai	76	berlebihanlah
7	agaknya	42	bagaikan	77	bermacam
8	agar	43	bagaimana	78	bermacam- macam
9	aja	44	bagaimanakah	79	bersama-sama
10	akan	45	bagaimanapun	80	beserta
11	akankah	46	bagi	81	betul
12	akhir	47	bagikan	82	betulkah
13	akhirnya	48	bahkan	83	biasa
14	aku	49	bahwa	84	biasanya
15	akukah	50	bahwasanya	85	bila
16	akulah	51	balik	86	bilakah
17	akupun	52	banyak	87	bilamana
18	amat	53	banyak-banyak	88	bisakah
19	amatlah	54	beberapa	89	blak-blakan
20	amboi	55	beda	90	bolak-balik
21	anda	56	begini	91	boleh
22	andai	57	beginikah	92	bolehkah
23	andaian	58	beginilah	93	bolehlah
24	andaikan	59	begitu	94	bukan
25	andaikata	60	begitukah	95	bukankah
26	andalah	61	begitulah	96	bukanlah
27	antar	62	begitupun	97	bukannya
28	antarkan	63	belum	98	cukup

29	antarlaha	64	belumhah	99	cuma
30	apa	65	benar	10 0	dahulu
31	apaan	66	benar-benar	10 1	dalam
32	apa-apaan	67	benarkah	10 2	dan
33	apabila	68	benarlah	10 3	dapat
34	apakah	69	berapa	10 4	dapatkan
35	apalagi	70	berapakah	10 5	dari

106	daripada	147	gitu	188	kalian
107	datang	148	hai	189	kalianlah
108	dekat	149	hal	190	kami
109	demi	150	hal-hal	191	kamikah
110	demikian	151	hampir	192	kamilah
111	demikianlah	152	hanya	193	kamipun
112	dengan	153	hanyalah	194	kamu
113	dengan demikian	154	harus	195	kamukah
114	di	155	haruslah	196	kamulah
115	dia	156	hendak	197	kamupun
116	diakah	157	hendaknya	198	kan
117	dialah	158	hilang	199	kapan
118	diantar	159	hingga	200	kapankah
119	diantara	160	ia	201	kapan- kapan
120	diantaranya	161	ialah	202	kapanlah

121	didahulukan	162	ibarat	203	kapanpun
122	didapati	163	ingin	204	karena
123	dikarenakan	164	inginkah	205	karenanya
124	dimana	165	inginkan	206	ke
125	dini	166	inginlah	207	kecil
126	diri	167	inginnya	208	kedapatan
127	dirinya	168	ini	209	kedekatan
128	disana	169	inikah	210	kemudian
129	disanakah	170	inilah	211	kemungkinan
130	disanalah	171	itu	212	kenapa
131	disanapun	172	itukah	213	kenapa
132	disini	173	itukan	214	kepada
133	disinilah	174	itulah	215	kepadanya
134	disitu	175	jadi	216	ketika
135	ditentukan	176	jangan	217	khususnya
136	dong	177	jangkalan	218	kini
137	duga	178	janganlah	219	kinilah
138	dugaan	179	jika	220	kira
139	dulu	180	jikalau	221	kira-kira
140	enggak	181	juga	222	kiranya
141	enggaknya	182	jugakah	223	kita
142	entah	183	justru	224	kita-kita
143	entahlah	184	kala	225	kitalah
144	entar	185	Kalau	226	kok
145	gara-gara	186	kalaulah	227	ku
146	gimana	187	Kalaupun	228	kuat-kuat
229	lagak	270	memang	311	sajalah
230	lagi	271	memanglah	312	saling
231	lagian	272	mengapa	313	sama
232	lagi-lagi	273	mereka	314	samakah
233	lah	274	merekalah	315	sama-sama
234	lain	275	merekapun	316	sambil

235	lainlah	276	merupakan	317	sampai
236	lain-lain	277	meski	318	sampaikan
237	lainnya	278	meskilah	319	sana
238	lakunya	279	meskipun	320	sang
239	lalai	280	minta	321	sangat
240	lalu	281	mirip	322	sangat
241	lalu-lalang	282	misal	323	sanggah
242	lama-lama	283	misalnya	324	saya
243	lamun	284	mu	325	sayakah
244	lebih	285	mula	326	sayalah
245	lepas	286	mungkin	327	sayapun
246	lihat	287	mungkinkah	328	sebab
247	lika-liku	288	mungkinlah	329	sebabnya
248	macam	289	musti	330	sebagai
249	macam- macam	290	nah	331	sebagaimana
250	maka	291	namun	332	sebagaimanan ya
251	makanya	292	nan	333	sebagainya
252	makin	293	nanti	334	sebaik- baiknya
253	malah	294	nantikan	335	sebaiknya
254	malahan	295	nantinya	336	sebaliknya
255	mampu	296	nun	337	sebanyak
256	mampukah	297	oh	338	sebanyaknya
257	mampukan	298	oi	339	sebegini
258	mana	299	oke	340	sebeginikah
259	manakala	300	oleh	341	sebegitu
260	manalagi	301	olehnya	342	sebegitukah
261	masih	302	pada	343	sebelum
262	masihkah	303	padahal	344	sebelumnya
263	masing- masing	304	padanya	345	sebenarnya

264	mau	305	paling	346	seberapa
265	maukah	306	paling-paling	347	seberapakah
266	maulah	307	para	348	sebetulnya
267	maupun	308	pasti	349	sebisa-bisanya
268	melainkan	309	pastikan	350	sebisanya
269	melulu	310	pastilah	351	sebuah

352	sebut	392	semakin
353	secara	393	semasih
354	sedalam- dalamnya	394	semaunya
355	sedang	395	sementara
356	sedangkan	396	semestinya
357	sedang-sedang	397	semisalnya
358	sedapat- dapatnya	398	sempat
359	sedemikian	399	sempat- sempatnya
360	sedikitnya	400	semua
361	segala	401	semuanya
362	segalanya	402	semula
363	segera	403	sendiri
364	seharusnya	404	sendirinya
365	seharusnya	405	seolah
366	sehingga	406	seolah-olah
367	sejak	407	seorang
368	sejenak	408	sepanjang
369	sekali	409	sepantasnya
370	sekalian	410	sepantasnyalah
371	sekaligus	411	seperti
372	sekali-kali	412	sepertinya
373	sekalipun	413	sepintas
374	sekarang	414	seraya

375	sekaranglah	415	sering
376	sekarangpun	416	seringnya
377	seketika	417	serta
378	sekiranya	418	sertai
379	sekitar	419	sertakan
380	sekitarnya	420	serupa
381	sela	421	sesaat
382	selagi	422	sesama
383	selain	423	sesegeera
384	selaku	424	sese kali
385	selalu	425	seseorang
386	selama- lamanya	426	sesuatu
387	selamanya	427	sesuatunya
388	selebihnya	428	sesuatupun
389	seluruh	429	sesudah
390	seluruhnya	430	sesudahnya
391	semacam	431	sesumbar

432	setelah	464	terdahulu
433	setiap	465	terdapat
434	setidaknya	466	terdiri
435	setidak- tidaknya	467	terhadap
436	seumpamanya	468	terhadapnya
437	sewaktu	469	terkadang
438	si	470	terlalu
439	siapa	471	terlebih
440	siapakah	472	tersebut
441	siapalah	473	tersebutkan
442	siapapun	474	tersebutlah
443	silakan	475	tetapi

444	sini	476	tiap
445	suatu	477	tiap-tiap
446	sudah	478	tidak
447	sudahkah	479	toh
448	sudahlah	480	tutur
449	supaya	481	uh
450	tadi	482	umpama
451	tadinya	483	untuk
452	tah	484	waduh
453	tak	485	wah
454	tampak	486	wahai
455	tanpa	487	walaupun
456	tapi	488	walaupun
457	telah	489	walhasil
458	tentang	490	wong
459	tentu	491	wow
460	tentukan	492	ya
461	tentulah	493	yaitu
462	tentunya	494	yakni
463	terbiasa	495	yang