

**PERINGKAS DOKUMEN OTOMATIS MENGGUNAKAN
METODE EKSTRAKSI KALIMAT DAN *FUZZY INFERENCE*
SYSTEM MODEL MAMDANI**

SKRIPSI

Oleh :

**TRI CAHYO ROMADHONA
0610960061**



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

UNIVERSITAS BRAWIJAYA



**PERINGKAS DOKUMEN OTOMATIS MENGGUNAKAN
METODE EKSTRAKSI KALIMAT DAN *FUZZY INFERENCE*
SYSTEM MODEL MAMDANI**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Ilmu Komputer

Oleh :

**TRI CAHYO ROMADHONA
0610960061**



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI

PERINGKAS DOKUMEN OTOMATIS MENGGUNAKAN METODE EKSTRAKSI KALIMAT DAN *FUZZY INFERENCE* SYSTEM MODEL MAMDANI

Oleh:
TRI CAHYO ROMADHONA
0610960061

Telah dipertahankan di depan Majelis Penguji
pada tanggal 19 Mei 2011
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Ilmu Komputer

Pembimbing I,

Pembimbing II,

Drs. Achmad Ridok, M.Kom
NIP. 196808251994031002

Djoko Pramono, ST
NIP. 197801082005011002

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas
Brawijaya

Dr. Abdul Rouf Alghofari
NIP. 196709071992031001

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Tri Cahyo Romadhona
NIM : 0610960061
Jurusan : Matematika
Program Studi : Ilmu Komputer
Penulis Skripsi berjudul : Peringkat Dokumen Otomatis
Menggunakan Metode Ekstraksi
Kalimat dan *Fuzzy Inference System*
Model Mamdani

Dengan ini menyatakan bahwa :

1. Isi dari skripsi 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 skripsi ini.
2. Apabila dikemudian hari ternyata skripsi 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, 4 Mei 2011
Yang menyatakan,

Tri Cahyo Romadhona
NIM. 0610960061

UNIVERSITAS BRAWIJAYA



PERINGKAS DOKUMEN OTOMATIS MENGGUNAKAN METODE EKSTRAKSI KALIMAT DAN *FUZZY INFERENCE* SYSTEM MODEL MAMDANI

ABSTRAK

Peringkas dokumen otomatis (*automatic text summarization*) adalah pembuatan versi ringkas dari sebuah teks yang dijalankan pada komputer tanpa mengurangi makna asli yang terkandung didalamnya dan membantu pengguna agar lebih cepat memahami informasi yang terkandung didalamnya. Secara umum terdapat 2 macam pembuatan peringkas dokumen teks otomatis yaitu menggunakan metode ekstraksi dan abstraksi. Pada metode ekstraksi sebuah ringkasan dokumen teks diperoleh dari rangkaian kata, kalimat maupun frasa yang diambil secara utuh dari dokumen asal, sehingga hasil yang didapat merupakan bagian dari dokumen tersebut.

Pada penelitian ini dibuat implementasi metode ekstraksi kalimat dimana akan dilakukan pembobotan kalimat yang diperoleh dari 6 fitur pada setiap kalimat, hasil nilai tiap fitur tersebut akan digunakan sebagai *input* nilai fuzzy menggunakan inferensi model mamdani. Hasil akhir yang didapat berupa nilai akhir bobot tiap kalimat, sehingga kalimat yang menjadi kandidat ringkasan diambil dari kalimat dengan nilai tertinggi hingga terendah sesuai dengan besar ringkasan yang diinginkan. Pengujian sistem dilakukan untuk menghitung nilai *precision* dan *recall* dengan besar ringkasan 25% dan 30% dan akan dilakukan dibandingkan dengan *AutoSummarize (Microsoft Word)* dan sebagai acuan digunakan ringkasan buatan manusia seorang Guru Bahasa Indonesia.

Setelah dilakukan ujicoba terhadap 15 dokumen berita yang diambil secara acak, didapatkan hasil nilai rata – rata *precision* sebesar 0.453001 dan *recall* sebesar 0.468134 untuk besar ringkasan sebesar 25%. Dan nilai rata – rata *precision* sebesar 0.589149 dan *recall* sebesar 0.601721 untuk besar ringkasan sebesar 50%. Sedangkan *AutoSummarizer* yang menghasilkan nilai rata – rata dari *precision* sebesar 0.349274 dan *recall* 0.354928 untuk besar ringkasan sebesar 25% dan untuk hasil ringkasan sebesar 50% menghasilkan nilai rata – rata *precision* sebesar 0.512515 dan *recall* 0.521096. dari hasil yang didapat, dapat disimpulkan bahwa sistem peringkas dokumen ini lebih baik.

UNIVERSITAS BRAWIJAYA



AUTOMATIC DOCUMENT SUMMARIZATION USING SENTENCE EXTRACTION AND FUZZY INFERENCE SYSTEM MAMDANI MODEL

ABSTRACT

Automatic text summarization (ATS) is making a short version of document text which run on the computer without reducing the pure meaning and to help user more quickly understand the information. In general there are two types of making automatic text summarization using extraction method and abstraction method. In extraction method a summary of text obtained from series of word, sentence or phrase which taken from the origin document, so the result taken from the part of that document.

In this reserch implementation sentence extraction method which will be obtained by weighting the sentence of 6 feature in every sentence, the value of each feature will be used as an input using fuzzy inference system mamdani model. The result is final weight of each sentence, so that the sentence candidate of summary taken from the sentence with the highest score to the lowest score according how much summary desired. The test system is performed to calculate value of precision and recall with summary of 25% and 50% and compare to AutoSummarize (Microsoft Word) and as a reference using man-made summary who teach Indonesian Language.

After a trial of 15 news documents that were taken randomly, the average result value precision is 0.468134 and recall is 0.453001 for 25% of the summary. And the average value precision is 0.589149 and recall is 0.601721 for a large summary of 50%. While AutoSummarizer produces average value of the precision is 0.349274 and recall is 0.354928 with large of summary of 25% and for the summary of 50% average value of precision is 0.521096 and recall is 0.512515. from the result obtained, it can be concluded that this automatic text summary more better.

UNIVERSITAS BRAWIJAYA



Kata Pengantar

Alhamdulillah, segala puji bagi Allah SWT Tuhan Semesta Alam yang telah mencurahkan, rahmat dan hidayah-Nya kepada seluruh makhluk-Nya. Sholawat serta salam terhatur kepada nabi kita Rosulullah Muhammad SAW dan semoga kita selalu istiqomah dalam meneladaninya. Karena hanya nikmat dan anugrah dari Allah semata sehingga dapat menyelesaikan skripsi dengan judul “PERINGKAS DOKUMEN OTOMATIS MENGGUNAKAN METODE EKSTRAKSI KALIMAT DAN *FUZZY INFERENCE SISTEM MODEL MAMDANI*”.

Tidak lupa pula penulis mengucapkan terima kasih kepada pihak – pihak yang memberikan dukungan dan bantuan sehingga skripsi ini dapat terselesaikan dengan baik. Dengan ketulusan dan kerendahan hati penulis mengucapkan terima kasih kepada :

1. Drs. Achmad Ridok, M.Kom, selaku pembimbing I yang telah meluangkan waktu tenaga dan pikiran serta perhatiannya sejak pengusulan penelitian hingga selesainya laporan skripsi ini.
2. Djoko Pramono, ST, selaku pembimbing II yang bersedia meluangkan waktu dan kesabarannya untuk meberikan bimbingan sampai dengan selesainya laporan skripsi ini.
3. Marji, MT., selaku ketua program studi Ilmu Komputer, jurusan Matematika, FMIPA Universitas Brawijaya.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Universitas Brawijaya.
5. Kedua Orang Tuaku Ibuku dan Bapakku yang telah membesarkan, merawat dan mendidik dengan kasih sayang. dan juga segenap keluarga di Lamongan, Malang, Cirebon dan Jogja, terima kasih atas support selama ini.
6. Eko Yuli K dan Sri Umi Rahma, faza dan hilwa terima kasih atas dukungan moril dan meteril
7. D. M. Widodo, S.SA selaku Guru Bahasa Indonesia SMKN I Sarirejo atas bantuan sehingga skripsi ini terselesaikan dengan baik dan juga kepada Agustin Rahmawati.
8. Teman2ku ilkomers 2006 serta seluruh mahasiswa dan alumni ilkom.

9. Semua pihak yang telah memberikan bantuan dan dukungan sehingga skripsi ini terselesaikan dengan baik.

Penulis sadari masih banyak kekurangan dalam penulisan laporan ini yang disebabkan oleh keterbatasan kemampuan dan pengalaman. Oleh karena itu penulis menghargai saran dan kritik yang sifatnya membangun demi perbaikan dan mutu isi penulisan. Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca dan bisa diambil manfaatnya.

Malang, Mei 2011

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN.....	iv
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
DAFTAR <i>SOURCECODE</i>	xxi
DAFTAR LAMPIRAN.....	xxiii
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat.....	3
1.6 Metodologi Pemecahan Masalah.....	4
1.7 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	
2.1 Konsep Peringkat Teks.....	7
2.2 Penggolongan teks <i>Summary</i>	7
2.3 <i>Text Mining</i>	9
2.3.1 <i>Preprocessing</i>	9
2.3.1.1 <i>Segmentation</i>	10
2.3.1.2 <i>Tokenization</i>	10
2.3.1.3 <i>StopWord</i>	10
2.3.1.4 <i>Stemming</i>	10
2.3.2 <i>Processing</i>	10
2.3.2.1 Kemiripan Judul (<i>Title Feature</i>).....	11
2.3.2.2 Panjang Kalimat (<i>Sentence Length</i>).....	11
2.3.2.3 Posisi Kalimat (<i>Sentence Position</i>).....	11
2.3.2.4 Bobot Kata (<i>Term Weight</i>).....	12
2.3.2.5 Huruf Besar (<i>Upper-case Word</i>).....	12

2.3.2.6	Data Numerik (<i>numerical data</i>).....	12
2.4	<i>Stemming</i> pada Bahasa Indonesia.....	13
2.4.1	Struktur Morfologi Kata Bahasa Indonesia.....	13
2.5	Logika <i>Fuzzy</i>	18
2.5.1	Pengertian Logika <i>Fuzzy</i>	18
2.5.2	Himpunan <i>Fuzzy</i>	19
2.5.3	Fungsi Keanggotaan <i>Fuzzy</i>	20
2.6	Sistem Inferensi <i>Fuzzy</i> Metode Mamdani.....	22
2.7	Penelitian Sebelumnya.....	24
2.7.1	Suanmali, Ladda. Salim, Naomie dan Binwahlan, Mohammed Salem. 2009.....	24
2.7.2	Kyoomarsi, Khosravi, Esfandiar dan Dehkordy. 2009.....	25
2.8	Tipe Evaluasi.....	26

BAB III METODOLOGI DAN PERANCANGAN

3.1	Analisa Sistem.....	31
3.1.1	Deskripsi Umum Sistem.....	31
3.1.2	Batasan Sistem.....	32
3.2	Perancangan Sistem.....	32
3.2.1	Perancangan Proses.....	33
3.2.1.1	Preproses.....	34
3.2.1.2	Proses.....	38
3.2.2	Logika <i>Fuzzy</i>	47
3.2.2.1	Desai Sistem <i>Fuzzy</i>	47
3.2.2.2	Proses Fuzzyfikasi.....	47
3.2.2.3	<i>Rule Fuzzy</i>	52
3.2.3	Perancangan Tabel.....	56
3.3	Perancangan Antar Muka.....	57
3.4	Perancangan Pengujian.....	58
3.4.1	Bahan Pengujian.....	58
3.4.2	Skenario dan Kriteria Pengujian.....	58
3.4.2.1	Evaluasi Hasil Ringkasan.....	58
3.4.2.2	Perbandingan Sistem.....	58
3.4	Perhitungan Manual.....	61

BAB IV HASIL DAN PEMBAHASAN

4.1	Lingkungan Implementasi.....	69
-----	------------------------------	----

4.1.1	Lingkungan Implementasi Perangkat Keras.....	69
4.1.2	Lingkungan Implementasi Perangkat Lunak.....	69
4.2	Implementasi Program.....	69
4.2.1	Implementasi <i>Text Preprocessing</i>	69
4.2.1.1	Segmentasi.....	69
4.2.1.2	Menghapus Tanda Baca.....	70
4.2.1.3	Menghapus <i>StopWord</i>	71
4.2.1.4	<i>Stemming</i>	72
4.2.2	Implementasi <i>Text Processing</i>	76
4.2.2.1	Perhitungan Fitur Judul.....	76
4.2.2.2	Perhitungan Fitur Panjang Kalimat.....	76
4.2.2.3	Perhitungan Fitur Posisi Kalimat.....	77
4.2.2.4	Perhitungan Fitur Bobot Kalimat	79
4.2.2.5	Perhitungan Fitur Huruf Besar.....	80
4.2.2.6	Perhitungan Fitur Numerik	81
4.2.3	Implementasi Logika <i>Fuzzy</i>	82
4.2.3.1	Perhitungan Derajat Keanggotaan.....	82
4.2.3.2	Penentuan Aturan <i>Fuzzy</i>	83
4.2.3.3	Defuzzifikasi.....	84
4.3	Implementasi Antar Muka.....	85
4.4	Implementasi Uji Coba.....	86
4.4.1	Skenario Uji Coba.....	86
4.4.2	Hasil Uji Coba.....	87
4.4.2.1	Pengujian Hasil Ringkasan Sebesar 25%.....	87
4.4.2.2	Pengujian Hasil Ringkasan Sebesar 50%.....	89
4.4.3	Analisa Hasil.....	91

BAB V PENUTUP

5.1	Kesimpulan.....	93
5.2	Saran.....	93

UNIVERSITAS BRAWIJAYA



DAFTAR GAMBAR

Gambar 2.1	Kurva Fungsi Keanggotaan.....	21
Gambar 2.2	Persamaan garis.....	23
Gambar 2.3	Diagram Blok Proses <i>Text Summarization</i> (Kyoomarsi Khosravi, Esfandiar dan Dehkordy, 2009).....	24
Gambar 3.1	Alur Penelitian.....	30
Gambar 3.2	Blok Diagram Proses Umum Sistem.....	32
Gambar 3.3	Diagram Alir Proses Umum Pembentukan <i>Text</i> <i>Summary</i>	33
Gambar 3.4	Diagram Alir Proses Penghilangan <i>StopWord</i>	35
Gambar 3.5	Diagram Alir Proses <i>Stemming</i>	37
Gambar 3.6	Diagram Alir Proses Perhitungan Fitur Judul.....	39
Gambar 3.7	Diagram Alir Proses Perhitungan Fitur Panjang Kalimat	40
Gambar 3.8	Diagram Alir Proses Perhitungan Fitur Posisi Kalimat	42
Gambar 3.9	Diagram Alir Proses Perhitungan Fitur Bobot Kata.....	44
Gambar 3.10	Diagram Alir Proses Perhitungan Fitur Huruf Besar	45
Gambar 3.11	Diagram Alir Proses Perhitungan Fitur Data Numerik	46
Gambar 3.12	Sistem <i>Fuzzy</i> Penentuan Nilai Bobot Kalimat.....	47
Gambar 3.13	Diagram Alir Proses Fuzzyfikasi	48
Gambar 3.14	Himpunan Fuzzy Variabel Kesamaan Judul (<i>Title</i> <i>Feature</i>).....	49
Gambar 3.15	Himpunan Fuzzy Variabel Panjang Kalimat (<i>Sentence Length</i>).....	49
Gambar 3.16	Himpunan Fuzzy Variabel Posisi Kalimat (<i>Sentence Position</i>).....	50
Gambar 3.17	Himpunan <i>Fuzzy</i> Variabel Bobot Kata (<i>Term Weight</i>).....	50
Gambar 3.18	Himpunan <i>Fuzzy</i> Variabel Huruf Besar (<i>Upper-Case Word</i>).....	51
Gambar 3.19	Himpunan <i>Fuzzy</i> Variabel Data Numerik (<i>Numerical Data</i>).....	51

Gambar 3.20	Himpunan <i>Fuzzy</i> Variabel <i>Output</i>	52
Gambar 3.21	Diagram Alir Proses Inferensi Model Mamdani....	55
Gambar 3.22	Diagram Alir Proses Defuzzifikasi	56
Gambar 3.23	Form Perancangan Aplikasi Peringkat Dokumen...	58
Gambar 3.23	Proses Defuzzifikasi dengan Metode Centroid.....	66
Gambar 4.1	Implementasi Antar Muka	86
Gambar 4.2	Grafik nilai <i>precision</i> sebesar 25%.....	88
Gambar 4.3	Grafik Nilai <i>Recall</i> Sebesar 25%... ..	89
Gambar 4.4	Grafik Nilai <i>Precision</i> Sebesar 50%.....	90
Gambar 4.5	Grafik Nilai <i>Recall</i> Sebesar 50%.....	91



DAFTAR TABEL

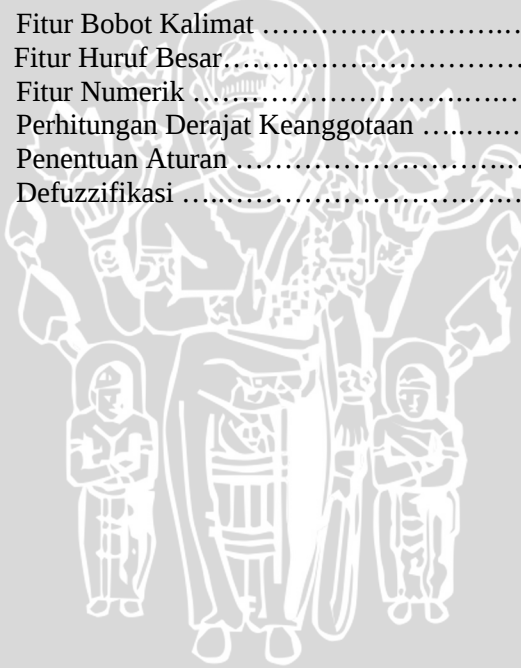
Tabel 2.1	Pasangan <i>Konfiks</i> yang diperbolehkan.....	15
Tabel 2.2	<i>Prefiks</i> Ganda.....	16
Tabel 2.3	Kelompok Aturan Pertama Pada Partikel Infleksional.....	17
Tabel 2.4	Kelompok Aturan Kedua Pada Partikel Infleksional Kata Ganti Milik.....	16
Tabel 2.5	Kelompok Aturan Ketiga Pada <i>Prefiks</i> <i>Derivasional</i>	17
Tabel 2.6	Kelompok Aturan Keempat Terhadap <i>Prefiks</i> <i>Derivasional</i> Kedua.....	18
Tabel 2.7	Kelompok Aturan Kelima Terhadap <i>Sufiks</i> <i>Derivasional</i>	18
Tabel 3.1	Tabel <i>StopWord</i>	56
Tabel 3.2	Tabel Aturan <i>Fuzzy</i>	57
Tabel 3.3	Informasi Berita.....	59
Tabel 3.4	Hasil Ringkasan Abstraktor.....	59
Tabel 3.5	Hasil Ringkasan Sistem.....	59
Tabel 3.6	Hasil Ringkasan <i>Auto Summarize</i>	59
Tabel 3.7	Perbandingan Hasil Ringkasan Sistem dengan <i>Auto Summarize</i> Sebesar 25%.....	60
Tabel 3.8	Perbandingan Hasil Ringkasan Sistem dengan <i>Auto Summarize</i> Sebesar 50%.....	60
Tabel 3.9	Frekuensi Kata.....	62
Tabel 3.10	Nilai Fitur per Kalimat.....	62
Tabel 4.1	Perbandingan Hasil Ringkasan Sistem dengan <i>Auto Summarizer</i> Sebesar 25%.....	87
Tabel 4.2	Nilai Rata – rata <i>Precisoin</i> dan <i>Recall</i> Ringkasan Sistem dengan <i>Auto Summarize</i> Sebesar 25%.....	88
Tabel 4.3	Perbandingan Hasil Ringkasan Sistem dengan <i>Auto Summarizer</i> Sebesar 50%.....	89
Tabel 4.4	Nilai Rata – rata <i>Precisoin</i> dan <i>Recall</i> Ringkasan Sistem dengan <i>Auto Summarize</i> Sebesar 50%.....	90

UNIVERSITAS BRAWIJAYA



DAFTAR SOURCECODE

Sourcecode 4.1	Segmentasi Paragraf dan Kalimat	70
Sourcecode 4.2	Menghapus Tanda Baca	71
Sourcecode 4.3	Menghapus <i>StopWord</i>	72
Sourcecode 4.4	<i>Stemming</i> Partikel Infleksional	72
Sourcecode 4.5	<i>Stemming</i> Kata Ganti Infleksional	73
Sourcecode 4.6	<i>Stemming</i> Prefiks Derivasional Pertama	74
Sourcecode 4.7	<i>Stemming</i> Prefiks Derivasional Kedua	75
Sourcecode 4.8	<i>Stemming</i> Sufiks Derivasional	75
Sourcecode 4.9	Fitur Judul	76
Sourcecode 4.10	Fitur Panjang Kalimat	77
Sourcecode 4.11	Fitur Posisi Kalimat	78
Sourcecode 4.12	Fitur Bobot Kalimat	80
Sourcecode 4.13	Fitur Huruf Besar	81
Sourcecode 4.14	Fitur Numerik	81
Sourcecode 4.15	Perhitungan Derajat Keanggotaan	82
Sourcecode 4.16	Penentuan Aturan	83
Sourcecode 4.17	Defuzzifikasi	85



UNIVERSITAS BRAWIJAYA



DAFTAR LAMPIRAN

Lampiran 1	Informasi Data Pengujian	97
Lampiran 2	Daftar <i>Stopword</i>	103

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



BAB I PENDAHULUAN

1.1 Latar Belakang

Seiring berkembangnya teknologi, perkembangan informasi juga berkembang dari hari ke hari. Ribuan informasi muncul dan beredar tiap waktu dari berbagai media cetak maupun elektronik. Banyaknya informasi yang terdapat pada dokumen melebihi kapasitas untuk membaca keseluruhannya. Kebutuhan akan informasi yang tepat dan cepat menuntut adanya alat untuk membuat ringkasan secara otomatis, sehingga informasi yang didapat lebih cepat dan tepat tanpa kehilangan makna asli dari dokumen asal (Azrifah dan Martin, 2009).

Peringkasan dokumen otomatis yaitu meringkas sebuah dokumen teks yang dilakukan oleh mesin untuk menampilkan versi yang lebih ringkas dari versi aslinya tanpa mengurangi makna asli yang terkandung didalamnya dan dapat membantu pengguna agar lebih cepat memahami informasi yang terdapat pada dokumen aslinya (Suanmali, Salim, dan Binwahlan, 2009).

Secara umum terdapat dua tipe dari pembuatan peringkasan dokumen teks secara otomatis, yaitu dengan menggunakan metode ekstraksi dan abstraksi. Pada metode ekstraksi sebuah ringkasan dokumen teks diperoleh dari rangkaian kata yang diambil secara utuh dari dokumen asal. Sehingga hasil ringkasan yang diperoleh tidak berupa kata, kalimat maupun rangkaian kalimat yang baru, tetapi merupakan salinan beberapa rangkaian kalimat yang terdapat dari dokumen asal. Pengertian dari rangkaian kalimat sebagai contoh frasa, klausa kalimat, kalimat utama maupun paragraph utama. Sedangkan yang dimaksud dengan abstraksi yaitu membuat rangkaian kata baru yang tidak terdapat pada dokumen asal. Sehingga ringkasan yang dihasilkan merupakan rangkaian kata – kata baru yang memang tidak terdapat pada dokumen asal. Metode ini menyerupakan cara dan tingkah laku manusia dalam membuat ringkasan dokumen. Sampai saat ini dalam mengimplementasikan metode ini masih dirasa sangat sulit dikarenakan metode ini menerapkan *natural language processing* (Jazek dan Steinberger, 2008).

Penelitian mengenai peringkasan teks otomatis pertama kali dipelopori oleh Luhn 1958 menggunakan rumus yang masih sering kita jumpai yaitu penggunaan *term frequencies* (Jazek dan Steinberger, 2008).

Sedangkan penelitian lain *text summary* pernah dilakukan oleh Ladda Suanmali, Naomie Salim dan Mohamed Salem Binwahlan dengan menggunakan ekstraksi kalimat dengan 8 fitur dan menerapkannya pada *fuzzy if-then rule*, fitur kalimat yang diekstrak akan dihitung nilai bobot tiap kalimatnya dan dijadikan sebagai input variabel *fuzzy* yaitu VL (*Very Low*), L (*Low*), M (*Medium*) H (*High*), VH (*Very High*). Lima variabel input fuzzy yang digunakan tersebut akan diterapkan pada *fuzzy IF-THEN RULE* dan akan didapat variabel *output* berupa tingkat kepentingan kalimat yaitu: *Important, Medium, Unimportant*. Kalimat yang menjadi kalimat ringkasan adalah kalimat yang mempunyai variabel output *Important*. Fungsi keanggotaan yang digunakan menggunakan fungsi keanggotaan segitiga (Suanmali, Salim, dan Binwahlan, 2009). pada penelitian selanjutnya yang dilakukan oleh Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami, dan Pooya Khosravayan Dehkordy hampir sama dengan penelitian sebelumnya yang menggunakan ekstraksi kalimat namun fitur kalimat yang mereka gunakan sebanyak 5, kemudian hasil ekstraksi kalimat yang didapat dijadikan input bagi logika *fuzzy*. Perbedaan pada penelitian sebelumnya yaitu kalimat ringkasan yang terpilih diperoleh dari perhitungan defuzzifikasi dengan menerapkan model Mamdani dengan metode *centroid* (Kyoomarsi, Khosravi, Esfandiar dan Dehkordy, 2009).

Metode mamdani sering juga dikenal dengan nama metode *min max*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan *output* diperlukan 4 tahapan, antara lain pembentukan himpunan fuzzy yaitu variabel input maupun output dibagi menjadi satu atau lebih himpunan fuzzy. Aplikasi fungsi implikasi yaitu fungsi implikasi yang digunakan adalah *min*. komposisi aturan yaitu pembentukan aturan yang akan dipakai. Dan yang terakhir *defuzzifikasi* yaitu pembentukan output menggunakan metode *centroid*, dimana pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat daerah *fuzzy* (Kusumadewi dan Hari, 2004).

Skripsi ini akan mengimplementasikan metode ekstraksi kalimat dan setiap kalimat akan dihitung bobotnya. Dalam memberikan nilai bobot tiap kalimat akan digunakan 6 fitur. Penerapan logika fuzzy disini menggabungkan antara 2 penelitian sebelumnya yaitu *inference system* yang dilakukan oleh Ladda Suanmali, Naomie Salim dan Mohamed Salem Binwahlan, yaitu variabel input maupun output fuzzy yang digunakan serta *rule* dan fungsi keanggotaan. dalam memperoleh

kandidat kalimat ringkasan akan mengadopsi dari penelitian Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami, dan Pooya Khosravayan Dehkordy yang menggunakan metode mamdani dalam memperoleh nilai akhir melalui proses defuzzifikasi. Berdasarkan latar belakang yang telah dikemukakan, judul yang diambil dalam skripsi ini adalah **“peringkasan dokumen otomatis menggunakan metode ekstraksi kalimat dan fuzzy inference system model mamdani”**.

1.2. Rumusan Masalah

Permasalahan yang ditangani dalam skripsi ini adalah :

1. Bagaimana merancang dan mengimplementasikan perangkat lunak peringkasan dokumen teks otomatis dengan metode ekstraksi kalimat yang diterapkan pada *fuzzy inference system* model mamdani.
2. Bagaimana hasil perbandingan antara sistem peringkasan dokumen teks otomatis yang menggunakan ekstraksi kalimat dan *fuzzy* model mamdani dengan *AutoSummarize*.

1.3. Tujuan

Tujuan pembuatan skripsi ini adalah :

1. Membuat dan mengimplementasikan perangkat lunak peringkasan dokumen teks otomatis dengan metode ekstraksi kalimat yang diterapkan pada *fuzzy inference system* model mamdani.
2. Mengetahui hasil perbandingan antara sistem peringkasan dokumen teks otomatis yang menggunakan ekstraksi kalimat dan *fuzzy* model mamdani dengan *AutoSummarize*.

1.4. Batasan Masalah

Batasan Masalah pada skripsi adalah :

1. Hanya digunakan pada dokumen teks yang menggunakan Bahasa Indonesia.
2. pada penelitian ini hanya digunakan pada dokumen tunggal.

1.5. Manfaat

Manfaat yang diperoleh dari penulisan tugas akhir ini yaitu menyediakan aplikasi perangkat lunak (*software*) yang dapat membuat peringkasan dokumen teks otomatis dari dokumen yang panjang menjadi lebih ringkas agar lebih informatif mendekati hasil ringkasan yang

dibuat manusia.

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 *sentence based summary* metode ekstraksi kalimat serta konsep mengenai logika *fuzzy* model mamdani.
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 hasil rancangan tersebut dengan metode ekstraksi kalimat dan diterapkan pada logika *fuzzy* model mamdani
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 menarik kesimpulannya.

1.7. Sistematika Penulisan

Skripsi 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 metode ekstraksi kalimat dan logika *fuzzy* model mamdani.
3. BAB III METODOLOGI DAN PERANCANGAN SISTEM
Pada bab ini akan dijelaskan mengenai metode ekstraksi kalimat dan perhitungan bobot kalimat melalui 6 fitur serta pembuatan *inferensi fuzzy mamdani*.
4. BAB IV PEMBAHASAN

Pada bab ini akan dilakukan implementasi *ekstraksi* kalimat dengan pembobotan kalimat serta menerapkannya pada *fuzzy inference system* model mamdani dan pengujian keakuratan sistem serta analisis sistem perangkat lunak yang dibangun yaitu apakah hasil sesuai yang diinginkan.

5. BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari seluruh rangkaian penelitian serta saran kemungkinan pengembangannya.



UNIVERSITAS BRAWIJAYA



BAB II TINJAUAN PUSTAKA

2.1 Konsep Peringkas Teks

Peringkas teks otomatis dapat diartikan sebagai sebuah metode atau cara bagaimana mengubah sebuah dokumen teks yang panjang dan besar menjadi dokumen dokumen yang lebih singkat, lebih ringkas dengan mengambil poin – poin penting yang terkandung didalamnya (Al-Hashemi, 2010). Tujuan dari peringkas teks yaitu mengambil intisari informasi dari dokumen sumber dan menjadikanya lebih singkat tanpa menghilangkan topik utama sehingga dapat membantu pengguna lebih cepat memahami makna yang terkandung didalamnya. Terdapat 2 metode mendasar dalam meringkas text yaitu ekstraksi dan abstraksi. Ekstraksi yaitu mengambil sebagian atau potongan kata, kalimat, frasa maupun klausa untuk disusun menjadi kandidat ringkasan sedangkan abstraksi yaitu dalam penyusunan ringkasan akan menghasilkan kalimat baru yang tidak terdapat dalam dokumen sumber, metode ini menerapkan *natural language processing* yang cara kerjanya hampir menyamai cara kerja manusia dalam membuat sebuah ringkasan (Jazek dan Steinberger, 2008).

2.2 Penggolongan Teks *Summary*

Ada beberapa penggolongan dalam *text summary*, menurut (Eduard dan Chin-yew) dalam jurnalnya yang berjudul “*Automated Text Summarization and the Summarist System*” penggolongan *text summary* dapat dikategorikan berdasarkan:

- Input sumber dokumen:
 - o *Single-document* : sumber utama dalam penyusunan ringkasan berupa dokumen tunggal.
 - o *Multiple document* : sumber dokumen yang digunakan dalam penyusunan ringkasan menggunakan lebih dari satu dokumen.
- Domain :
 - o Spesifik : ruang lingkup (domain) yang digunakan dalam ringkasan hanya berfokus pada ruang lingkup yang sempit (berfokus pada satu masalah). Input diperoleh dari satu domain masalah yang dibatasi.
 - o Umum : ruang lingkup yang digunakan berfokus pada

masalah – masalah umum. Input yang digunakan berupa berbagai teks yang berasal dari berbagai macam domain

- Asal ringkasan:

- Ekstraksi : mengambil potongan maupun cuplikan kata, kalimat, frasa, paragraph maupun klausa yang kemudian dipilih sesuai dengan rangking bobot potongan tersebut untuk menjadi sebuah ringkasan.
- Abstraksi : hasil ringkasan berupa untaian kata – kata baru yang tidak terdapat dalam dokumen asal. dalam abstraksi ini hampir mirip dengan cara manusia dalam membuat ringkasan karena menerapkan *natural language processing*.

- Hubungan antar kata atau kalimat:

- Lancar : sebuah ringkasan dapat dikatakan terhubung secara lancar apabila ringkasan tersebut disusun sesuai dengan grammar yang benar, dan antar kalimat satu dengan lainnya terdapat hubungan berdasarkan aturan keterkaitan kalimat.
- Tak lancar : hubungan antar kalimat dikatakan tak lancar apabila ringkasan tersebut terbentuk atas bagian teks yang tersusun tidak sesuai dengan grammar yang benar, dan antar kalimatnya tidak terhubung dengan baik.

- Level pemrosesan :

- *Surface level* : pada level ini pemrosesan sebuah informasi hanya pada fitur level yang dangkal (*shallow*) beserta kombinasinya. Contoh yang termasuk fitur ini penggunaan GSM (*General Statistical Method*) seperti lokasi kata, kata kunci, istilah kata, atau penggunaan domain yang spesifik. Hasil yang didapat nantinya berupa ekstraksi.
- *Deeper level* : dalam level ini biasanya dapat digunakan dalam method ekstraksi maupun abstraksi. Level ini menggunakan *natural language generation*. Penekatan pada level ini menggunakan *semantic analysis* untuk menentukan bagian mana suatu Entitas saling terhubung.

- Tujuan :

- *Indicative* : ringkasan yang memberikan informasi singkat serta hanya menampilkan bagian yang dianggap penting. Jenis ringkasan ini biasanya digunakan sebagai pembantu pengguna dalam menentukan apakah isi dokumen asal cocok sesuai yang dicari.

- o *Informative* : ringkasan yang mewakili dokumen secara keseluruhan termasuk bagian – bagian yang dianggap mendetail
- Penggunanya :
 - o *Generic* : tujuan utama pembuatan ringkasan jenis ini untuk komunitas pengguna yang luas. Semua domain topik dalam hal ini dianggap sama penting
 - o *Query-Based* : hasil ringkasan didasarkan fokus topik yang sesuai dengan pengguna butuhkan

2.3 Text Mining

Salah satu bagian dari *data mining* adalah *text mining*. Yang dimaksud dengan data mining yaitu suatu cara untuk memperoleh sedikit data yang berharga dari sejumlah banyak data yang tak berharga. Konsep data mining berasal dari filosofi menambang batuan permata (*mining*) yaitu mencari secuil batuan mulia dari sekumpulan ribuan batu yang tak berharga. Dalam konteks *text mining* sendiri artinya menggali informasi yang berupa data data dalam bentuk teks bisa berupa Koran, majalah, artikel, paper, buku, makalah dengan tujuan mencari informasi yang berharga didalamnya.

Dalam implementasinya, *text mining* secara garis besar dipisahkan menjadi dua tahap yaitu *preprocessing* dan *processing*. Tahap *preprocessing* merupakan tahap penyeleksiian terhadap data pada tiap – tiap dokumen, pada tahap ini dilakukan beberapa proses lain yaitu: *segmentation*, *tokenization* *stopword* dan *stemming*. Tujuan dari tahap ini adalah mempersiapkan kata untuk dilakukan proses pada tahap selanjutnya, misalkan peyeragaman dalam bentuk huruf kecil (*lower case*) maupun perubahan menjadi bentuk kata dasar. Pada tahap kedua yaitu *processing* yaitu tahap inti dimana setiap kata akan diolah sesuai dengan algoritma tertentu.

2.3.1 Preprocessing

Preprosing merupakan tahap yang dilakukan sebelum tahap inti dijalankan, pada tahap ini terdapat 4 hal yang dilakukan antara lain *segmentation*, *tokenization* *stopword* dan *stemming*, tujuan yang diharapkan dalam proses ini adalah mempersiapkan baik kata maupun kalimat agar bisa digunakan pada tahap inti (*prosesing*).

2.3.1.1 *Segmentation*

Segmentasi merupakan proses dimana sebuah dokumen akan dipecah pecah berdasarkan kalimat.

2.3.1.2 *Tokenization*

Proses *tokenization* yaitu pemisahan kalimat menjadi potongan – potongan kata, yang kemudian akan digunakan pada proses selanjutnya

2.3.1.3 *Stopword*

Stopword dapat diartikan sebagai kata – kata yang sering muncul maupun yang mempunyai frekuensi tinggi kemunculannya pada dokumen namun tidak mempunyai manfaat informasi. Biasanya *stopword* merupakan kata yang tergolong kata sambung, preposisi maupun konjugasi. Salah satu keuntungan dengan menghilangkan *stopword* yaitu mengurangi kapisitas perhitungan tiap kata dan dapat mengurangi indeks.

2.3.1.4 *Stemming*

Stemming adalah teknik yang digunakan untuk mengubah suatu kata menjadi bentuk dasarnya. Pada kaidah bahasa Indonesia proses steaming biasanya adalah untuk merubah kata berimbuhan pada awalan, akhiran maupun sisipan menjadi bentuk kata dasar. Algoritma yang digunakan dalam proses steaming menggunakan algoritma Porter Steamer. Dalam konteks steaming bahasa Indonesia kata menyapu, disapu dan tersapu mempunya arti yang sama karena mempunyai arti kata dasar yang sama yaitu sapu.

2.3.2 *Processing*

Tahap selanjutnya setelah *preprocessing* yaitu melakukan tahap *processing*. Tahap *processing* ini merupakan tahap dilakukan pengolahan kata menggunakan algoritma tertentu. Dalam konteks penelitian ini setiap kata maupun kalimat yang telah diuraikan pada tahap sebelumnya akan dihitung nilainya. Pada perhitungan kalimat akan dilakukan perhitungan berdasarkan fitur – fitur kalimat yang digunakan. Sedangkan untuk pengolahan kata akan dilakukan baik perhitungan bobot, kesamaan dengan topik maupun jumlah kemunculan kata perkalimat.

2.3.2.1 Kemiripan Judul (*title feature*)

secara umum judul dari sebuah bacaan menggambarkan isi dari bacaan tersebut, dengan dasar seperti ini maka sebuah kalimat yang mempunyai kata mirip dengan judulnya maka mempunyai peluang besar sebagai kalimat ringkasan. Nilai bobot didapat dari jumlah kata dalam kalimat yang sama dengan kata dalam judul dibagi dengan jumlah keseluruhan kata dalam kalimat tersebut (Suanmali, Salim, dan Binwahlan, 2009).

$$F1(K) = \frac{\text{jumlah kata yang sama dengan judul}}{\text{jumlah kata dalam judul}} \quad (2.1)$$

2.3.2.2 Panjang Kalimat (*Sentence Length*)

fitur ini berguna menyaring kalimat yang terlalu singkat. Kalimat yang terlalu singkat tidak diharapkan masuk dalam ringkasan, kalimat seperti ini biasanya berisi cuma *dateline* dan nama pengarang dalam artikel berita. Digunakan *ratio* perbandingan antar kalimat. Perhitungan nilai bobot yaitu perbandingan jumlah kata dalam kalimat dibagi dengan jumlah kata dalam kalimat yang terpanjang (Suanmali, Salim, dan Binwahlan, 2009).

$$F2(K) = \frac{\text{jumlah kata dalam kalimat}}{\text{jumlah kata dalam kalimat terpanjang}} \quad (2.2)$$

2.3.2.3 Posisi Kalimat (*Sentence Position*)

Posisi kalimat juga turut berpengaruh dalam letak kalimat topik. Dalam paragraf berbahasa indonesia kalimat topik terletak pada awal kalimat, akhir kalimat maupun di tengah paragraf. Karena sulitnya mengidentifikasi secara pasti dan jaranganya penggunaan kalimat topik yang terletak ditengah, maka pada penelitian ini hanya akan menggunakan sistem ranking berdasarkan pada posisi kalimat dimana nilai tertinggi terletak pada kalimat pertama. (Fatah dan Ren, 2008). Aturan yang dipakai :

$$F3(K) = \begin{matrix} 5/5 & \text{untuk } 1^{st}, \\ 4/5 & \text{untuk } 2^{nd}, \end{matrix}$$

3/5 untuk 3rd,
 2/5 untuk 4th,
 1/5 untuk 5th,

2.3.2.4 Bobot Kata (*term weight*)

Bobot kalimat dapat dihitung dari jumlah tiap kata penyusunnya, dengan menerapkan metode *tf.isf* (*Term Frequency Inverse Sentence Frequency*) dapat dihitung nilai tiap kata (Suanmali, Salim, dan Binwahlan, 2009).

$$W_t = tf_i \times isf_i = tf_i \times \log \frac{N}{n_i} \quad (2.3)$$

$$F4(K) = \frac{\sum_{i=1}^j W_i(S_i)}{\text{Max}(\sum_{i=1}^j W_i(S_i^N))} \quad (2.4)$$

Dimana :

tf_i = frekuensi kata pada dokumen

N = jumlah kalimat

n_i = jumlah kalimat dimana kata ke t muncul

j = jumlah kata dalam kalimat

2.3.2.5 Huruf Besar (*Upper-case Word*)

Kata yang mengandung huruf besar mempunyai peluang termasuk dalam kata yang dipentingkan. Perhitungan rumusnya merupakan rasio antara jumlah kata yang mengandung huruf besar dibagi dengan jumlah kata dalam kalimat (Gupta dan Singh. 2010).

$$F5(K) = \frac{\text{banyaknya kata yang terdapat huruf besar}}{\text{banyak kata dalam kalimat}} \quad (2.5)$$

2.3.2.6 Data Numerik (*numerical data*)

Kalimat yang terdalamnya terdapat data berupa bilangan mempunyai kemungkinan termasuk dalam kalimat ringkasan. Bobot nilai fitur ini dihitung dari banyaknya jumlah kata yang terdapat bilangan dalam kalimat dibagi dengan panjang kalimat (Fatah dan Ren, 2008).

$$PG(K) = \frac{\text{banyaknya kata yang terdapat angka}}{\text{banyak kata dalam kalimat}}$$

(2.6)

2.4 Stemming pada Bahasa Indonesia

Stemming yang digunakan pada bahasa Indonesia ini menggunakan algoritma yang telah dibangun oleh Porter. Algoritma porter sendiri telah disesuaikan ke dalam berbagai macam bahasa. Dalam konteks Bahasa Indonesia algoritma porter telah dimodifikasi oleh Fadillah Z. Tala pada tahun 2003.

2.4.1 Struktur Morfologi Kata Bahasa Indonesia

Secara umum dalam Bahasa Indonesia afiks (imbuhan) terdiri dari *sufiks* (akhiran), *infiks* (sisipan), dan prefiks (awalan) maupun kombinasi dari ketiganya. Walaupun *infiks* ada dalam tata bahasa Indonesia namun karena jarang sekali dipakai dan penanganan tergolong sulit maka pada proses steaming yang dibangun akan mengabaikan kata yang mengandung *infiks* dan hanya diterapkan pada *sufiks* dan prefiks (Fadillah Z Tala, 2003).

Dalam struktur morfologi pada kata Bahasa Indonesia dapat dibagi menjadi dua yaitu infleksional dan derivasional. Infleksional merupakan struktur sederhana yang dinyatakan dengan penambahan *sufiks* yang tidak akan berakibat perubahan arti pada kata dasar. Infleksional sendiri dibagi menjadi dua golongan yaitu:

1. *Sufiks* -lah, -kah, -pun, -tah. *Sufiks* ini sebenarnya partikel yang tidak mempunyai arti kata. Keberadaanya pada sebuah kata hanya digunakan untuk mempertegas. Contohnya:

Kamu	+	kah	=> kamukah
Saya	+	lah	=> sayalah

2. *Sufiks* -ku, -mu, -nya. *Sufiks* ini diletakkan pada kata ganti kepunyaan. contohnya:

Pensil	+	mu	=> pensilmu
Mobil	+	ku	=> mobilku

(Fadillah Z Tala, 2003)

Masing – masing *sufiks* pada golongan 1 maupun 2 mungkin terdapat pada kata yang sama. Ketika kedua *sufiks* muncul secara

bersama, mereka akan mengikuti aturan secara tegas. *Sufiks* yang kedua selalu mendahului *sufiks* yang kedua (Fadillah Z Tala, 2003). Aturannya dapat ditulis sebagai berikut:

Infleksional = (kata dasar + kata ganti) | (kata dasar + partikel) | (kata dasar + kata ganti + partikel)

Penambahan *sufiks* infleksional pada kata tidak akan merubah pengejaan kata asalnya. Dengan kata lain tidak ada peleburan karakter pada kata berimbuhan. Kata dasar dapat ditentukan dengan mudah pada struktur *infleksional* (Fadillah Z Tala, 2003).

Struktur *derivational* pada Bahasa Indonesia terdiri dari prefik, *sufiks*, dan gabungan kombinasi keduanya. Prefix yang paling umum digunakan adalah *ber-*, *di-*, *ke-*, *meng-*, *peng-*, *per-*, *ter-*. (Fadillah Z Tala, 2003) Contohnya:

Ber	+	lari	=> berlari
Di	+	makan	=> dimakan
Ke	+	kasih	=> kekasih
Meng	+	ambil	=> mengambil
Peng	+	atur	=> pengatur
Per	+	panjang	=> perpanjang
Ter	+	baca	=> terbaca

Beberapa prefks seperti *ber-*, *meng-*, *peng-*, *per-*, *ter-* mungkin akan muncul pada benuk yang berbeda. Bentuk pada setiap prefiks tergantung pada karakter pertama pada kata yang ditambahkan. Tidak seperti struktur infleksional, misalkan kata menyapu yang dibangun dari prefiks *meng-* dan kata dasar sapu. Prefiks *meng-* dirubah menjadi *meny-* dan karakter pertama pada kata dasar akan dihilangkan aturan pada beberapa penambahan prefiks dapat dilihat pada lampiran. (Fadillah Z Tala, 2003)

Sufiks derivasional yaitu *-i*, *-kan*, *-an*. Contohnya:

Gula	+	i	=> gulai
Makan	+	an	=> makanan

Beri + kan => berikan

Berbeda dengan prefiks, penambahan suffiks tidak akan merubah ejaan pada kata dasar pada kata berimbuhan (Fadillah Z Tala, 2003).

Seperti yang disebutkan di awal, struktur derivasional juga mengenali *konfiks* yaitu merupakan kombinasi penambahan bersama sama antara prefiks dan *sufiks* pada sebuah kata untuk menjadi kata berimbuhan (Fadillah Z Tala, 2003). Contohnya:

Per + main + an => permainan

Ke + menang + an => kemenangan

Ber + jauh + an => berjatuhan

Meng + ambil + kan => mengambil

Tidak semua kombinasi prefiks dan *sufiks* dapat digunakan secara bersama – sama membentuk *konfiks*. Ada beberapa kombinasi dari prefiks maupun *sufiks* yang tidak diperbolehkan. *Kofiks* yang diperbolehkan dapat dilihat pada tabel 2.1

Tabel 2.1 Pasangan *Konfiks* yang diperbolehkan (Fadillah Z Tala, 2003)

Prefik	<i>Sufiks</i>
Ber	I
Di	An
Ke	i kan
Meng	An
Peng	i kan
Ter	An

Prefiks maupun *konfiks* dapat ditambahkan pada kata yang telah diberi *konfiks* maupun *prefiks*, dimana hasilnya berupa struktur prefiks ganda. Seperti struktur pada *konfiks*, tidak semua prefiks maupun *konfiks* dapat ditambahkan pada *konfiks* maupun prefiks untuk membentuk prefiks ganda. Ada aturan sendiri untuk membentuk prefiks ganda, aturan tersebut dapat ditunjukkan pada tabel 2.2.

Tabel 2.2 *Prefiks Ganda* (Fadillah Z Tala, 2003)

prefiks 1	prefiks 2
Meng	Per
Di	Ber
Ter	
Ke	

Morfologi struktur kata derivational

Prefiks = prefiks + kata dasar

Sufiks = kata dasar + *sufiks*

Konfiks = prefiks + kata dasar + *sufiks*

Prefiks ganda = (prefiks 1 + prefiks 2) | (prefiks + *konfiks*) | (prefiks 1 + prefiks 2 + *sufiks*)

Kemungkinan terakhir yang terjadi adalah penambahan *sufiks* infleksional pada suatu kata yang telah mendapat prefiks, *sufiks*, *konfiks*, maupun prefiks ganda. Hal ini dinamakan multiple *sufiks*. Sehingga dapat disimpulkan bahwa struktur morfologi kata bahasa Indonesia secara umum adalah sebagai berikut (Fadillah Z Tala, 2003)

Struktur morfologi = [prefiks 1] + [prefiks 2] + kata dasar + [*sufiks*] + [kata ganti] + [partikel].

dimana [...] menunjukkan pilihan.

Berdasarkan analisa morfologi yang telah dijelaskan pada subbab 2.4.1, didapatkan lima kelompok aturan proses stemming pada bahasa Indonesia. Kelima aturan tersebut adalah menangani partikel infleksional, urutan prefiks derivasional pertama dan kedua, serta *sufiks* derivasional, sebagaimana akan ditunjukkan Tabel 2.3, 2.4, 2.5, 2.6, dan 2.7 (Fadillah Z Tala, 2003).

Kondisi ukuran adalah jumlah minimum suku kata dalam sebuah kata. Karena dalam bahasa Indonesia, kata dasar setidaknya mempunyai 2 suku kata.

Tabel 2.3 Kelompok Aturan Pertama Pada Partikel Infleksional (Fadillah Z Tala, 2003)

<i>sufiks</i>	pengganti	Kondisi ukuran	Kondisi tambahan	Contoh
Kah	NULL	2	NULL	diakah → dia
Lah	NULL	2	NULL	adalah → ada
Pun	NULL	2	NULL	apapun → apa

Tabel 2.4 Kelompok Aturan Kedua Pada Partikel Infleksional Kata Ganti Milik (Fadillah Z Tala, 2003)

<i>sufiks</i>	pengganti	Kondisi ukuran	Kondisi tambahan	Contoh
Ku	NULL	2	NULL	bukuku → buku
Mu	NULL	2	NULL	bukumu → buku
Nya	NULL	2	NULL	bukunya → buku

Tabel 2.5 Kelompok Aturan Ketiga Pada Prefiks Derivasional (Fadillah Z Tala, 2003)

<i>sufiks</i>	pengganti	Kondisi ukuran	Kondisi tambahan	Contoh
Meng	NULL	2	NULL	mengukur → ukur
Meny	S	2	V...*	menyapu → sapu
Men	NULL	2	NULL	menduga → duga
Mem	P	2	V...	memilah → pilah
Mem	NULL	2	NULL	membaca → baca
Me	NULL	2	NULL	merusak → rusak
Peng	NULL	2	NULL	pengukur → ukur
Peny	S	2	V...	penyapu → sapu
Pen	NULL	2	NULL	penuduh → tuduh
Pem	P	2	V...	pemilah → pilah
Pem	NULL	2	NULL	pembaca → baca
Di	NULL	2	NULL	diukur → ukur
Ter	NULL	2	NULL	terbaca → baca
Ke	NULL	2	NULL	kekasih → kasih

Tabel 2.6 Kelompok Aturan Keempat Terhadap Prefiks Derivasional Kedua (Fadillah Z Tala, 2003)

<i>sufiks</i>	Pengganti	Kondisi ukuran	Kondisi tambahan	Contoh
Ber	NULL	2	NULL	berlar → lari
Bel	NULL	2	Ajar	belajar → ajar
Be	NULL	2	K*er...	bekerja → kerja
Per	NULL	2	NULL	perjelas → jelas
Pel	NULL	2	Ajar	pelajar → pelajar
Pe	NULL	2	NULL	pekerja → pe

Tabel 2.7 Kelompok Aturan Kelima Terhadap *Sufiks* Derivasional (Fadillah Z Tala, 2003)

<i>sufiks</i>	Pengganti	Kondisi ukuran	Kondisi tambahan	Contoh
-kan	NULL	2	Prefik bukan anggota {ke, peng}	tarikan → tarik, mengambil → ambil
-an	NULL	2	Prefik bukan anggota {di, meng, ter}	makanan → makan, perjanjian → janji
-i	NULL	2	Prefik bukan anggota {ber, ke, peng}	tandai → tanda, mendapati → dapat

2.5. Logika Fuzzy

2.5.1 Pengertian Logika Fuzzy

Dalam bahasa inggris logika *fuzzy* diartikan sebagai logika kabur atau tidak jelas, yaitu logika yang mengandung unsur ketidakjelasan. Logika fuzzy merupakan suatu cara untuk memetakan suatu ruang input ke dalam ruang output (Kusumadewi dan Hari, 2004). Logika fuzzy digunakan untuk menterjemahkan suatu besaran yang dapat diekspresikan dengan bahasa (*linguistic*), misalkan besaran data tentang suhu yang dapat diekspresikan dengan dingin, sedang, panas maupn sangat panas.

Dalam logika biasa yaitu logika tegas, kita hanya mengenal 2 nilai

yaitu 0 dan 1, 0 mewakili salah sedangkan 1 mewakili benar. Namun pada logika fuzzy kita juga mengenal nilai antara benar dan salah. Kebenaran dalam logika fuzzy biasanya dinyatakan dengan derajat keanggotaan yang mempunyai range antara 0 dan 1. Sebagai contoh pada kasus sehari-hari orang dikatakan tua jika berusia 60 tahun keatas. Dalam logika tegas biasa orang yang berusia 60 tahun namun kurang 1 hari dia dikatakan belum tua, sedangkan pada logika fuzzy orang tersebut dikatakan hampir tua.

Menurut (Kusumadewi dan Hari, 2004) dalam bukunya aplikasi logika fuzzy sebagai pendukung keputusan mengemukakan alasan mengapa menggunakan logika fuzzy antara lain:

1. Konsep logika mudah dimengerti, konsep matematis yang mendasari penalaran logika fuzzy sangat sederhana
2. Logika fuzzy sangat fleksibel
3. Logika fuzzy memiliki toleransi terhadap data – data yang tidak tepat
4. Logika fuzzy mampu memodelkan fungsi – fungsi non-linear yang sangat kompleks
5. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman – pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan
6. Logika fuzzy dapat bekerjasama dengan teknik – teknik kendali secara konvensional
7. Logika fuzzy didasarkan pada bahasa alami

2.5.2 Himpunan Fuzzy

Himpunan *fuzzy* merupakan pengelompokan nilai berdasarkan variabel bahasa (*linguistic variable*) yang dinyatakan dalam fungsi keanggotaan dalam semesta U . sedangkan keanggotaannya sendiri dinyatakan nilainya berkisar antara 0.0 sampai 1.0 (Saelan, 2009).

Beberapa hal yang perlu diketahui dalam memahami sistem fuzzy yaitu:

- a. Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy misalkan suhu, temperatur, umur.
- b. Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy

- c. Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan dalam pengoprasian variabel fuzzy. Biasanya semesta pembicaraan ini merupakan bilangan real yang selalu monoton naik dari kiri ke kanan. Nilai ini bisa berupa bilangan positif maupun bilangan negatif, dan untuk kasus tertentu semesta pembicaraan ini tidak dibatasi nilai akhirnya.

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , sering ditulis dengan $\mu_A [x]$, memiliki 2 kemungkinan yaitu

- Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan atau
- Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

2.5.3 Fungsi Keanggotaan Fuzzy

Fungsi-fungsi keanggotaan *fuzzy* terparameterisasi satu dimensi yang umum digunakan diantaranya adalah:

1. Fungsi keanggotaan segitiga, disifati oleh parameter $\{a,b,c\}$ yang didefinisikan sebagai berikut :

$$\text{segitiga}(x,a,b,c) = \begin{cases} 0; & x < a \\ (x-a)/(b-a); & a \leq x \leq b \\ (c-x)/(c-b); & b \leq x \leq c \\ 0; & c \leq x \end{cases} \quad (2.7)$$

bentuk yang lain dari persamaan di atas adalah

$$\text{segitiga}(x,a,b,c) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (2.8)$$

parameter $\{a,b,c\}$ (dengan $a < b < c$) yang menentukan koordinat x dari ketiga sudut segitiga tersebut, seperti terlihat pada Gambar 2.1(a).

2. Fungsi keanggotaan trapesium, disifati oleh parameter $\{a,b,c,d\}$ yang didefinisikan sebagai berikut:

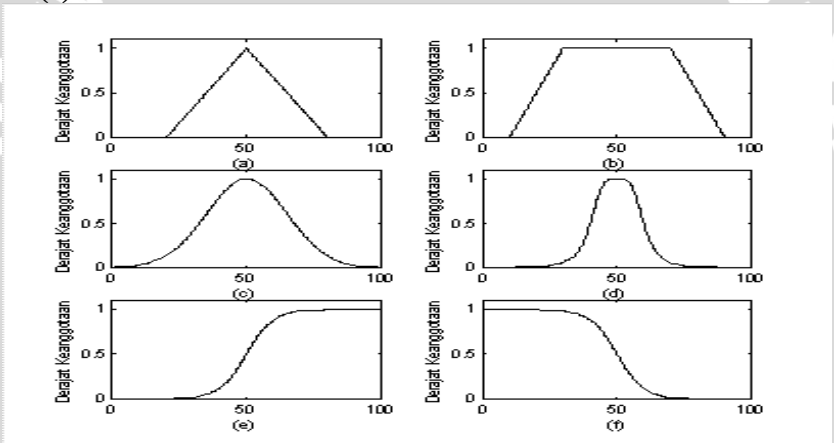
$$\text{trapesium}(x,a,b,c,d) = \begin{cases} 0; & x < a \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x)/(d-c); & c \leq x \leq d \\ 0; & d \leq x \end{cases} \quad (2.9)$$

parameter $\{a,b,c,d\}$ (dengan $a < b < c < d$) yang menentukan koordinat x dari keempat sudut trapesium tersebut, seperti terlihat pada Gambar 2.1(b).

3. Fungsi keanggotaan Gaussian, disifati oleh parameter $\{c,s\}$ yang didefinisikan sebagai berikut :

$$\text{gaussian}(x,c,\sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (2.10)$$

Fungsi keanggotaan Gauss ditentukan oleh parameter c dan s yang menunjukkan titik tengah dan lebar fungsi, seperti terlihat pada Gambar 2.1(c).



Gambar 2.1 Kurva fungsi keanggotaan, (a).segitiga ($x;20,50,80$), (b).trapesium ($x;10,30,70,90$), (c).gaussian($x;50,15$), (d).bell($x;10,2,50$), (e).sigmoid ($x;0,2,50$) dan (f).sigmoid($x;-0.2,50$).

4. Fungsi keanggotaan *generalized bell*, disifati oleh parameter $\{a,b,c\}$ yang didefinisikan sebagai berikut:

$$\text{bell}(x,a,b,c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^b} \quad (2.11)$$

parameter b selalu positif, supaya kurva menghadap kebawah, seperti terlihat pada Gambar 2(d).

5. Fungsi keanggotaan sigmoid, disifati oleh parameter $\{a,c\}$ yang didefinisikan sebagai berikut:

$$\text{sig}(x,a,c) = \frac{1}{1+\exp[-a(x-c)]} \quad (2.12)$$

parameter a digunakan untuk menentukan kemiringan kurva pada saat $x = c$. Polaritas dari a akan menentukan kurva itu kanan atau kiri terbuka, seperti terlihat pada Gambar 2.1(d) dan 2.1(e).

2.6 Sistem Inferensi Fuzzy Metode Mamdani

Metode *mamdani* diperkenalkan oleh Ibrahim Mamdani pada tahun 1975. Metode *Mamdani* sering juga disebut metode Max-Min. Untuk mendapatkan output diperlukan 4 tahapan. Tahapan tersebut yaitu :

1. Pembentukan Himpunan Fuzzy
Pada Metode *Mamdani*, baik variable input maupun output dibagi satu atau lebih himpunan fuzzy.
2. Aplikasi fungsi Implikasi (Aturan)
Pada metode *Mamdani*, fungsi implikasi yang digunakan adalah minimum.
3. Komposisi Aturan

Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dari beberapa korelasi antar aturan. Aturan yang digunakan dalam melakukan inferensi sistem fuzzy yaitu menggunakan metode *MAX (Maximum)*.

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah fuzzy dan mengaplikasikannya ke output dengan menggunakan operator *OR*. Jika semua proposisi telah diaplikasikan maka output akan berisi suatu himpunan fuzzy yang merefleksikan kontribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan :

$$\mu_{sf}[xi] \leftarrow \text{Max} (\mu_{sf}[xi], \mu_{kf}[xi]) \quad (2.13)$$

Dengan :

$\mu_{sf}[xi]$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke- i

$\mu_{kf}[xi]$ = nilai keanggotaan konsekuen *fuzzy* aturan ke- i .

4. Penegasan (*Defuzzy*)

Proses defuzzifikasi adalah proses pengubahan *crisp input* menjadi *fuzzy input*. Untuk mentransformasikan *crisp input* menjadi *fuzzy input*, diperlukan *membership function* untuk tiap input. Proses *fuzzy input* mengambil nilai *crisp input* dan membandingkannya dengan *membership function* yang telah ada untuk menghasilkan harga *fuzzy input*. Dalam proses defuzzifikasi dapat diolah lebih dari satu macam *crisp input*. Metode defuzzifikasi yang dipakai pada komposisi aturan Mamdani, yaitu metode *centroid (center of gravity)*

Pada metode ini setiap *membership function output* yang memiliki nilai di atas nilai fuzzy output akan dipotong. Pemotongan ini disebut *lamda cut*. Hasilnya (*membership function* yang terpotong) digabung dulu baru dihitung *center of gravity* keseluruhannya.

Defuzzifikasi pada komposisi aturan mamdani dengan menggunakan metode centroid. Dimana pada metode ini solusi crisp diperoleh dengan cara mengambil titik pusat daerah fuzzy. Secara Umum dirumuskan pada persamaan 2.14

$$\mu(x) = \frac{\sum_{i=1}^n x_i \mu(x_i)}{\sum_{i=1}^n \mu(x_i)} \quad (2.14)$$

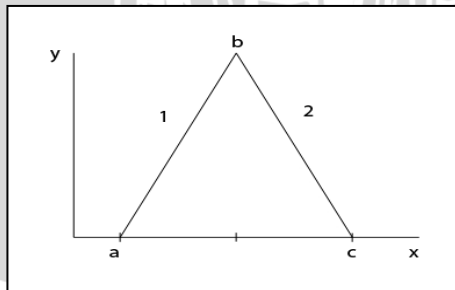
Pada proses defuzzifikasi perhitungan perpotongan garis dengan variabel *linguistic* menggunakan persamaan :

Persamaan Garis 1 :

$$x = (b-a) * y + a \quad (2.15)$$

Persamaan garis 2 :

$$x = b - (b-a) * y \quad (2.16)$$



Gambar 2.2 Persamaan Garis

Ada dua keuntungan menggunakan *metode centroid*, yaitu (Kusumadewi dan Hari, 2004):

1. Nilai defuzzyfikasi akan bergerak secara halus sehingga perubahan dari suatu himpunan fuzzy juga akan berjalan dengan halus.
2. Lebih mudah dalam perhitungan

2.7 Penelitian Sebelumnya

Penelitian mengenai peringkasan teks otomatis pertama kali dilakukan oleh luhn tahun 1958 dengan menggunakan *term frequencies*, luhn mengusulkan bahwa sebuah kata yang sering mncl pada sebuah dokumen mempunyai peluang yang besar sebagai kandidat ringkasan. Pada tahun yang sama Baxendele 1958 mengusulkan bahwa posisi sebuah kalimat turut mempengaruhi kandidat dalam ringkasan. Tahun 1969 Edmudson meneliti mengenai ekstraksi kalimat yang didasarkan pada dua penelitian sebelumnya dan ditambah fitur *cue phrase*(kata kunci) pada kalimat, Edmudson juga mengusulkan pemberian pembobotan pada kalimat (Das dan Martins. 2007).

2.7.1 Suanmali, Ladda. Salim, Naomie dan Binwahlian, Mohammed Salem. 2009

Pada penelitian kali ini, sebuah ringkasan dihasilkan dari ekstraksi kalimat, kemdian setiap kalimat dihitung nilai bobot melalui 8 fitur antara lain : (*title feature*), panjang kalimat (*sentence length*), bobot istilah (*term weight*), posisi kalimat (*sentence position*), kesamaan antar kalimat (*sentence to sentence similarity*), kata benda yang tepat (*proper noun*), kata tematik (*thematic word*) dan data numerik (*numerical data*). Hasil output didapat berupa nilai bobot kalimat perfitur yang nantinya akan digunakan sebagai input pada *fuzzy if-then rule*. Fungsi keanggotaan yang digunakan menggunakan grafik segitiga (*triangular*) dimana setiap fungsi keanggotaan per fitur dibagi menjadi 5 variabel : (*low (L) and very low (VL), Median (M) and important values (high (H) and very high (VH)*).

Untuk memperoleh nilai output digunakan rumus fungsi segitiga

$$triangular(x,a,b,c) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (2.17)$$

dimana a dan c menyatakan sisi sebelah kiri dan kanan pada gambar fungsi segitiga (kaki segitiga), sedangkan b menunjukkan titik puncak segitiga

Sebagai contoh salah satu aturan yang digunakan

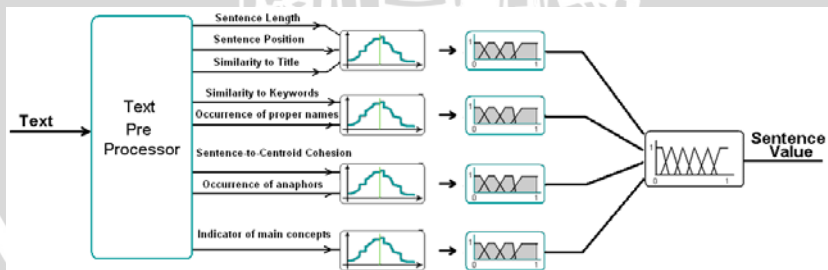
IF (NoWordInTitle is VH) and (SentenceLength is H) and (TermFreq is VH) and (SentencePosition is H) and (SentenceSimilarity is VH) and (NoProperNoun is H) and (NoThematicWord is VH) and (NumericalData is H) THEN (Sentence is important)

Sedangkan variable *output* yang digunakan yaitu *unimportant*, *average* dan *important*. Dari perhitungan diatas, maka akan diketahui kalimat mana saja yang dianggap penting untuk menjadi kandidat ringkasan. (Suanmali, Salim, dan Binwahlan, 2009)

2.7.2 Kyoomarsi, Khosravi, Esfandiar dan Dehkordy. 2009

Pada penelitian ini, sebuah ringkasan (*summary*) didapat dari metode ekstraksi kalimat dimana dilakukan pembobotan tiap kalimat. Dari berbagai fitur pada pebobotan kalimat, dalam penelitian ini Cuma digunakan 5 fitur antara lain : *sentence length*, *sentence position*, *similarity to title*, *similarity to keyword*, *occurance of proper noun*, *sentence to centroid cohesion*, *occurance of anarphord*, *indicator of main concept* Sama seperti penelitian sebelumnya, hasil ekstraksi kalimat akan dinilai bobot tia fitur dan akan menjadi inputan bagi fuzzy.

Logika fuzzy yang digunakan menggnakan model mamdani dimana fungsi keanggotaanya memakai fungsi segitiga.



Gambar 2.3 Diagram Blok Proses *Text Summarization* (Kyoomarsi, Khosravi, Esfandiar dan Dehkordy. 2009)

Pada fuzzy model mamdani langkah yang di ikuti

I. Fuzzification of the factors value,

II. Inference using rule base,

III. Defuzzification and sentence ranking.

Sedangkan pada fungsi keanggotaan segitga menerapkan rumus.

$$\text{triangular}(x,a,b,c) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (2.18)$$

pada model mamdani proses defuzzifikasi digunakan metode *centroid*

$$\mu(x) = \frac{\sum_{i=1}^n x_i \mu(x_i)}{\sum_{i=1}^n \mu(x_i)} \quad (2.19)$$

(Kyoomarsi, Khosravi, Esfandiar dan Dehkordy. 2009)

2.8 Tipe Evaluasi

Metode untuk mengevaluasi hasil ringkasan biasanya digunakan *precision dan recall*. *Precision* sendiri diartikan sebagai nilai yang menandakan seberapa besar sistem berhasil mengekstrak kalimat topik yang sesuai. Sedangkan *recall* sendiri diartikan nilai yang menandakan seberapa besar sistem gagal mengekstrak kalimat topik yang sesuai. Evaluasi sebuah ringkasan sendiri termasuk dalam topik yang sulit, baik evaluasi yang dilakukan oleh sistem maupun oleh abstraktor yang profesional, dikarenakan tidak ada definisi yang ideal sebuah ringkasan. Terdapat dua klasifikasi metode evaluasi (Sparck dan Galliers, 1996), yaitu:

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

Evaluasi yang digunakan saat ini bersifat intrinsik, yaitu pengevaluasi membuat ringkasan yang ideal kemudian hasilnya dibandingkan dengan ringkasan sistem. Jika dimisalkan *correct* adalah jumlah kalimat yang berhasil di ekstrak sistem sesuai dengan kalimat yang diekstrak manusia, *wrong* adalah jumlah kalimat yang diekstrak sistem tetapi tidak terdapat dalam kalimat yang diekstrak manusia dan *missed* adalah

jumlah kalimat yang diekstrak manusia tetapi tidak diekstrak oleh sistem, maka rumus *precision* dan *recall* ditunjukkan pada persamaan 2.20 dan 2.21. (Hovy 2003).

$$\text{precision} = \frac{\text{correct}}{\text{correct} + \text{wrong}} \quad (2.20)$$

$$\text{recall} = \frac{\text{correct}}{\text{correct} + \text{missed}} \quad (2.21)$$



UNIVERSITAS BRAWIJAYA



BAB III

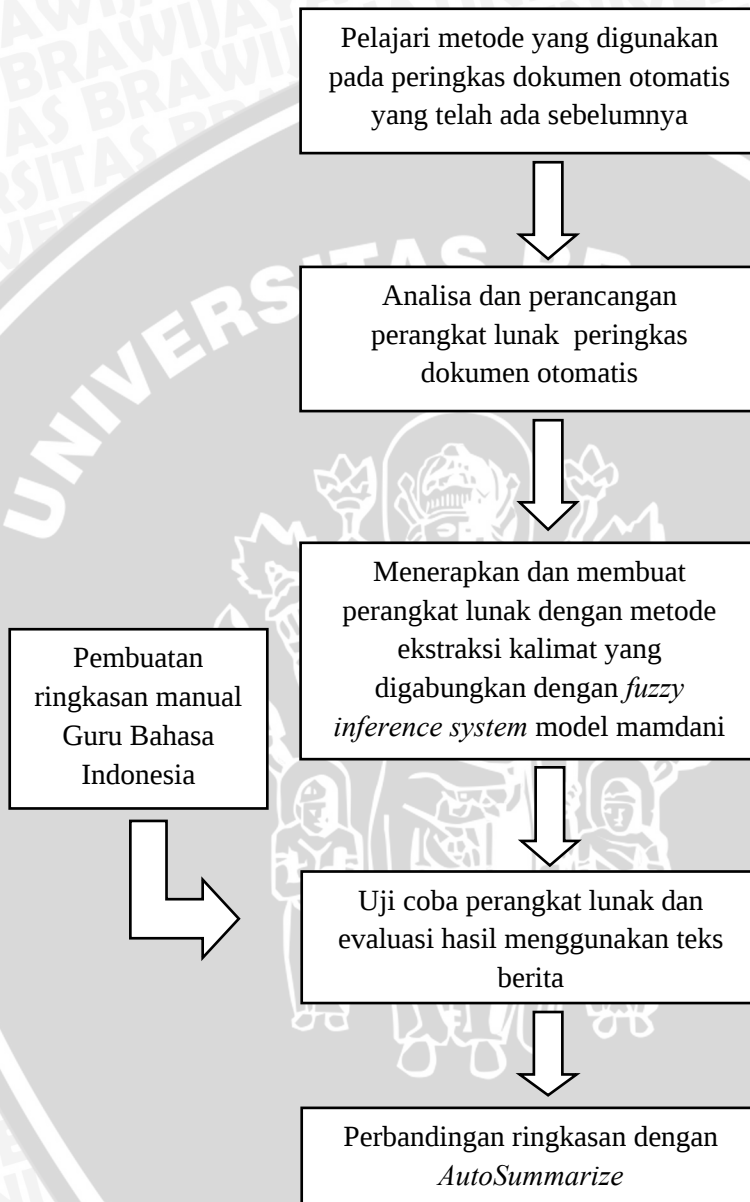
METODOLOGI DAN PERANCANGAN

Pada bab ini berisikan metodologi dari penelitian yang dilakukan dan rancangan sistem yang dibuat sebagai perangkat uji coba dalam penelitian pembuatan ringkasan dokumen otomatis dengan metode ekstraksi kalimat dan menggunakan *fuzzy inference system* model mamdani.

Penelitian dilakukan dengan tahapan – tahapan berikut ini:

1. Mempelajari berbagai metode yang digunakan pada peringkasan dokumen yang telah ada sebelumnya, yaitu metode ekstraksi kalimat dan *fuzzy inference system* model mamdani, yang telah dijelaskan pada bab 2.
2. Menganalisa dan merancang perangkat lunak peringkasan dokumen dengan menggabungkan metode yang terdapat pada penelitian sebelumnya, yaitu ekstraksi kalimat dan *fuzzy inference system* model mamdani.
3. Menerapkan dan membuat perangkat lunak berdasarkan analisa dan perancangan yang telah dilakukan dengan metode ekstraksi kalimat dan digabung dengan menerapkan *fuzzy inference system* model mamdani.
4. Uji coba perangkat lunak peringkasan dokumen, dengan menggunakan teks berita.
5. Untuk mendapatkan ringkasan ideal, teks berita diberikan pada Salah Satu Guru SMK Bidang Studi Bahasa Indonesia.
6. Mengevaluasi dan membandingkan hasil ringkasan yang dibuat oleh sistem dengan hasil sistem lain. Dalam hal ini sistem lain yang digunakan sebagai pembanding adalah *AutoSummarize (Microsoft Word)* yang akan dibandingkan nilai *precision* dan *recall*.

Langkah – langkah yang dilakukan dalam penelitian ini dapat ditunjukkan pada Gambar 3.1



Gambar 3.1 Alur Penelitian

3.1 Analisa Sistem

Pada subbab analisa sistem akan dibahas mengenai deskripsi secara umum sistem dan batasan – batasan yang ada pada sistem. Penjelasan yang lebih lengkap akan dibahas pada subbab 3.2.1.

3.1.1 Deskripsi Umum Sistem

Peringkasan dokumen merupakan sistem yang dikembangkan untuk melakukan pembuatan ringkasan secara otomatis dengan membaca dokumen secara keseluruhan dan secara otomatis akan menghasilkan (*generate*) sebuah ringkasan. Metode yang digunakan menggunakan ekstraksi kalimat dan *fuzzy inference system* model mamdani. Dalam pelaksanaannya akan dibagi menjadi dua tahap. Tahap pertama yaitu preproses yang terdiri dari *segmentation*, *tokenization*, penghilangan *stopword* dan *stemming*. *Stemming* yang digunakan pada penelitian ini penulis hanya menggunakan *stemming* yang menangani awalan maupun akhiran seperti dijelaskan pada subbab 2.4.1. Tahap kedua merupakan tahap proses dimana merupakan tahap inti, setiap kalimat akan dihitung nilai bobotnya menggunakan 6 fitur antara lain kemiripan dengan judul (*title feature*), panjang kalimat (*sentence length*), bobot istilah (*term weight*), posisi kalimat (*sentence position*), adanya huruf besar (*Upper-Case Word*) dan data numerik (*numerical data*). Hasil dari bobot tiap kalimat tersebut akan menjadi input bagi perhitungan fuzzy mamdani dan diperoleh kandidat kalimat yang menjadi ringkasan.

Pada tahap pertama proses yang dilakukan adalah:

1. *User* memasukkan teks dokumen yang akan diringkaskan.
2. Sistem akan melakukan *segmentation* yaitu menguraikan dokumen menjadi kalimat dan dilanjutkan dengan *tokenization* yaitu pemecahan kalimat menjadi kata.
3. Setelah pemecahan kata, sistem akan memeriksa pada daftar *stopword*, jika sama dengan daftar *stopword* maka kata tersebut dibuang, jika tidak maka akan dilanjutkan dengan proses *stemming*.

Pada tahap kedua proses yang dilakukan adalah:

1. Setiap kalimat akan dihitung nilai bobot berdasarkan pada 6 fitur antara lain : kemiripan dengan judul (*title feature*), panjang kalimat (*sentence length*), bobot istilah (*term weight*), posisi kalimat (*sentence position*), adanya huruf besar (*Upper-Case Word*) dan

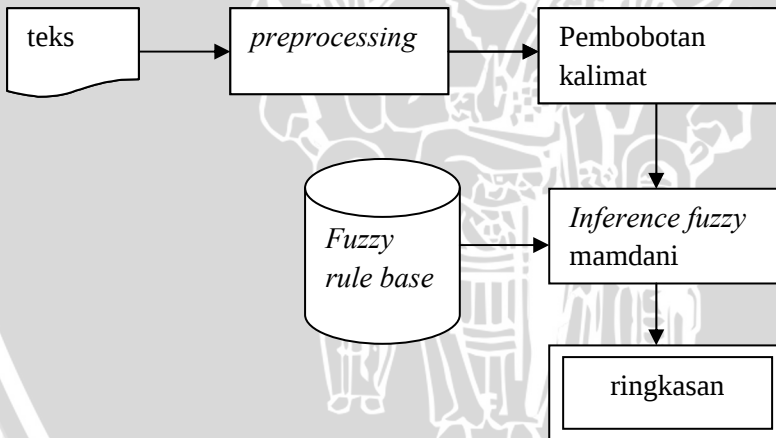
data numerik (*numerical data*).

2. Setelah didapatkan bobot tiap kalimat maka dilakukan perhitungan logika *fuzzy inference system* model mamdani. Hasil yang didapat berupa variabel *fuzzy* yaitu *important*, *average* dan *unimportant* beserta nilai *crisp* bobot akhir kalimat.

3.1.2 Batasan Sistem

Batasan sistem yang akan dikembangkan antara lain :

1. Proses penghilangan *stopword* tergantung pada bahasa yang digunakan, sehingga batasan yang dipakai sistem menggunakan batasan Bahasa Indonesia.
2. Proses *stemming* menggunakan algoritma *Porter Steamer* yang telah dimodifikasi oleh Fadillah Z. Tala pada tahun 2003, namun dengan mengabaikan adanya kata sisipan (*sufiks*).
3. Format dokumen *rich text* (.rtf).
4. Stuktur dokumen berupa judul dan isi tanpa memperhatikan *heading* (bab maupun subbab), sehingga yang digunakan berupa dokumen berita.



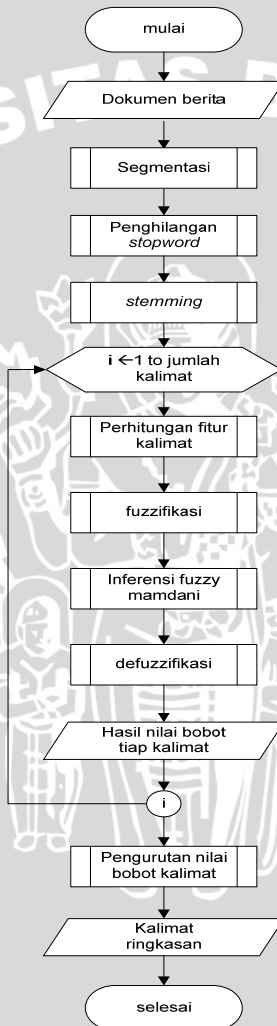
Gambar 3.2 Blok Diagram Proses Umum Sistem

3.2 Perancangan Sistem

Berdasarkan analisa dilakukan maka berikut ini akan dibahas mengenai perancangan proses yang terdapat dalam sistem peringkasan dokumen otomatis yang akan dikembangkan.

3.2.1 Perancangan Proses

Secara garis besar proses peringkasan dibagi menjadi 2 tahapan yaitu preproses dan proses. Teks yang dimasukkan dalam sistem ini terdiri dari judul dan isi. Perhitungan fitur hingga perhitungan yang menggunakan logika fuzzy akan dibahas disini.



Gambar 3.3 Diagram Alir Proses Umum Pembentukan *Text Summary*

3.2.1.1 Preproses

Preproses merupakan proses awal yang dilakukan sebelum proses inti dijalankan. Dikarenakan pembentukan ringkasan ini menuntut perhitungan fitur kalimat maka sebuah dokumen akan dipecah menjadi paragraf kemudian menjadi kalimat dan terakhir menjadi kata yang terpisah. Beberapa langkah – langkah yang dilakukan pada preproses antara lain :

a. *Segmentation*

Segmentation atau pemecahan kalimat yaitu memecah sebuah kalimat menjadi kalimat – kalimat. Aturan yang digunakan sebagai berikut:

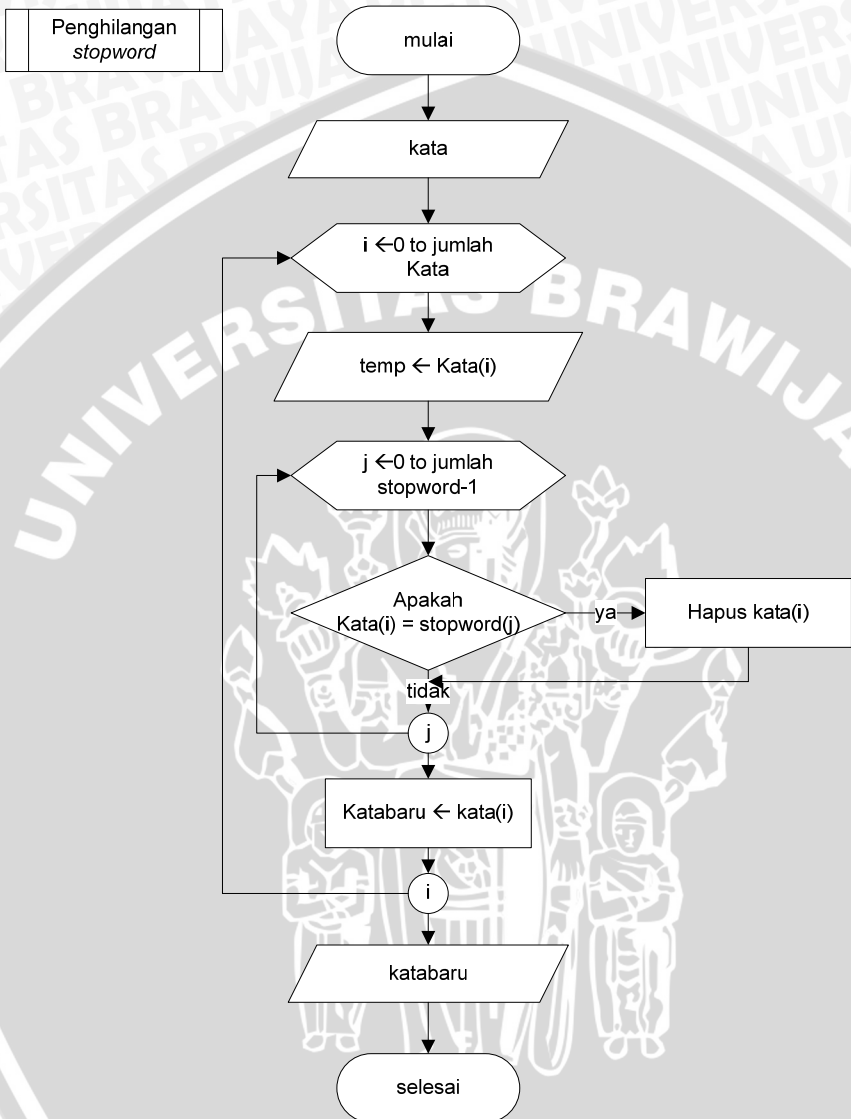
- Sebuah kalimat diakhiri 1 atau lebih tanda titik (.) tanda tanya (?).
- *Sentence delimiter* biasanya diikuti oleh tanda petik dua, misalnya : pak guru berkata : “*Kerjakan tugas ini!*”
- Delimiter terakhir harus diikuti oleh satu atau lebih *white space* (spasi, tab, *newline*).
- Kata pertama kalimat berikutnya seharusnya dimulai dengan huruf kapital.
- *Sentence delimiter* tidak seharusnya menjadi bagian dari singkatan (*abbreviation*).

b. *Tokenization*

Tahap *Tokenization* adalah tahap pemecahan dokumen menjadi kata kata. Pemecahanya dilakukan dengan men-*scan* kalimat pemisah *white space* (spasi, tab, *newline*). Sebuah kata didefinisikan sebagai suatu runtutan karakter – karakter ([A-Z], [a-z] dan / atau [0,9]).

c. Penghilangan *Stopword*

Proses yang dilakukan adalah menghilangkan kata-kata yang tidak penting seperti kata penghubung dan kata tanya yang dikenal dengan *stopword*. Daftar *stopword* yang dihilangkan terdapat pada Lampiran 1 dengan sumber Tala(2003).



Gambar 3.4 Diagram Alir Proses Penghilangan *StopWord*

d. Stemming

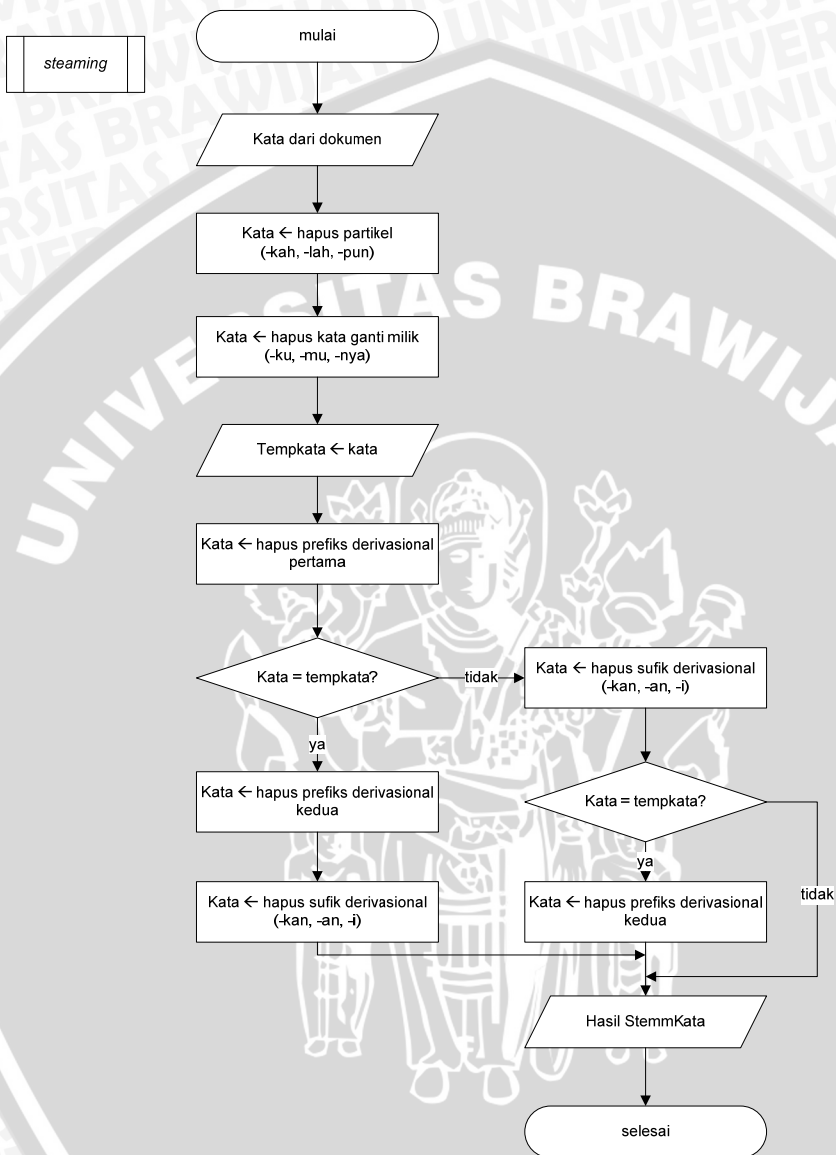
Stemming yaitu proses untuk mengubah bentuk kata – kata di dalam dokumen menjadi kata dasarnya (*root word*) dilandasi dengan aturan –

aturan tertentu. Proses *stemming* ditujukan untuk meningkatkan tingkat performansi sistem, jadi *stemming* bukanlah suatu pekerjaan dalam etimologi atau *grammar*. Maka dari itu, dalam algoritma *stemming* sendiri masih bisa ditolerir kesalahannya. *Stemming* yang digunakan adalah *porter stemmer* yang telah diubah dalam Bahasa Indonesia Fadillah Z Tala, tahun 2003.

Stemming dalam Bahasa Indonesia berbeda dengan *stemming* pada Bahasa Inggris. Pada dokumen yang menggunakan Bahasa Inggris proses yang dilakukan pada *stemming* hanya penghilangan *sufiks* (akhiran). Sedangkan pada *stemming* yang terdapat pada Bahasa Indonesia proses *stemming* dilakukan dengan penghilangan *sufiks* (akhiran), *infiks* (sisipan) dan *prefiks* (awalan). Karena dalam Bahasa Indonesia *infiks* (sisipan) jarang dipakai dan mempunyai tingkat kesulitan yang cukup tinggi, maka proses *stemming* yang dibangun kali ini hanya pada *prefiks* (awalan) dan *sufiks* (akhiran).

Algoritma *porter stemmer* yang akan dibangun telah dijelaskan pada subbab 2.4.1, tahapannya sebagai berikut:

1. Hapus *sufiks* infleksional yang berupa partikel (“-lah”, “-kah”, dan “-tah”).
2. Hapus *sufiks* infleksional yang berupa kata ganti kepunyaan (“-ku”, “-mu”, dan “-nya”).
3. Hapus awalan *derivasional* pertama (“me-” dan semua variasinya, “peng-”, “peny-”, “pen-”, “pem-”, “di-”, “ter-”, dan “ke-”). Jika tidak ada lanjutkan ke langkah 4a, jika ada cari maka lanjutkan ke langkah 4b.
4. a. Hapus awalan *derivasional* kedua (“be-” dan semua variasinya, “pe-”, “per-”, dan “pel-”). Lanjutkan ke langkah 5a.
b. Hapus akhiran *derivasional* (“-kan”, “-an”, dan “-i”). Jika tidak ditemukan maka kata tersebut diasumsikan sebagai *root word*. Jika ditemukan maka lanjutkan ke 5b.
5. a. Hapus akhiran *derivasional*. Kata akhir diasumsikan sebagai *root word*
b. Hapus awalan *derivasional* kedua. Kata akhir diasumsikan sebagai *root word*.



Gambar 3.5 Diagram Alir Proses *Stemming*

3.2.1.2 Proses

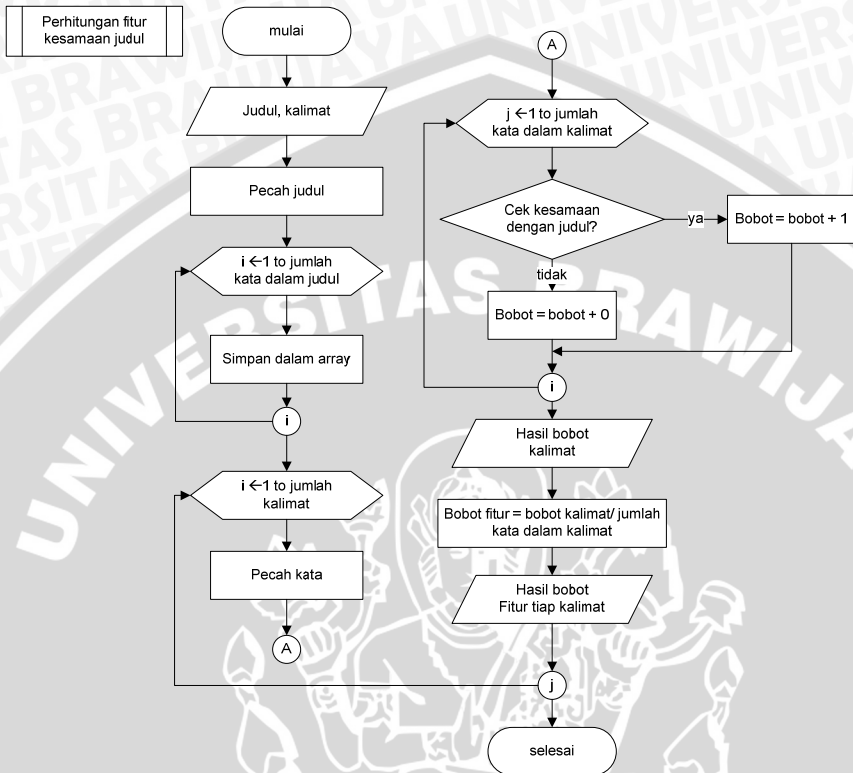
Tahap proses merupakan tahap inti dimana setiap kata akan diproses dengan algoritma tertentu. Dalam penelitian kali ini, proses yang dilakukan adalah perhitungan nilai bobot setiap kalimat dengan kriteria fitur berikut: kemiripan dengan judul (*title feature*), panjang kalimat (*sentence length*), bobot kata (*term weight*), posisi kalimat (*sentence position*), kesamaan antar kalimat (*sentence to sentence similarity*)) dan data numerik (*numerical data*).

a. Kemiripan Judul (*title feature*)

secara umum judul dari sebuah bacaan menggambarkan isi dari bacaan tersebut, dengan dasar seperti ini maka sebuah kalimat yang mempunyai kata mirip dengan judulnya maka mempunyai peluang besar sebagai kalimat ringkasan. Nilai bobot didapat dari jumlah kata dalam kalimat yang sama dengan kata dalam judul dibagi dengan jumlah keseluruhan kata dalam kalimat tersebut. Pada fitur ini dibatasi tidak pada kata yang mempunyai sinonim melainkan kata dianggap sama hanya jika struktur morfologinya sama, misalkan kata baju dan pakaian tidak dianggap sebagai kata yang sama melainkan sebagai 2 kata yang berbeda.

Langkah perhitungan fitur judul sebagai berikut:

1. Pecah judul menjadi kata
2. menyimpan dalam array
3. untuk tiap kalimat pecah menjadi kata
4. untuk tiap kata dalam kalimat cek dengan array judul,
5. jika sama dengan kata dalam judul maka bobot ditambah 1, jika tidak bobot ditambah 0
6. hitung bobot akhir tiap kalimat
7. didapat bobot kalimat
8. perhitungan nilai fitur = bobot kalimat/ jumlah kata dalam kalimat
9. didapat nilai fitur tiap kalimat



Gambar 3.6 Diagram Alir Proses Perhitungan Fitur Judul

b. Panjang Kalimat (*Sentence Length*)

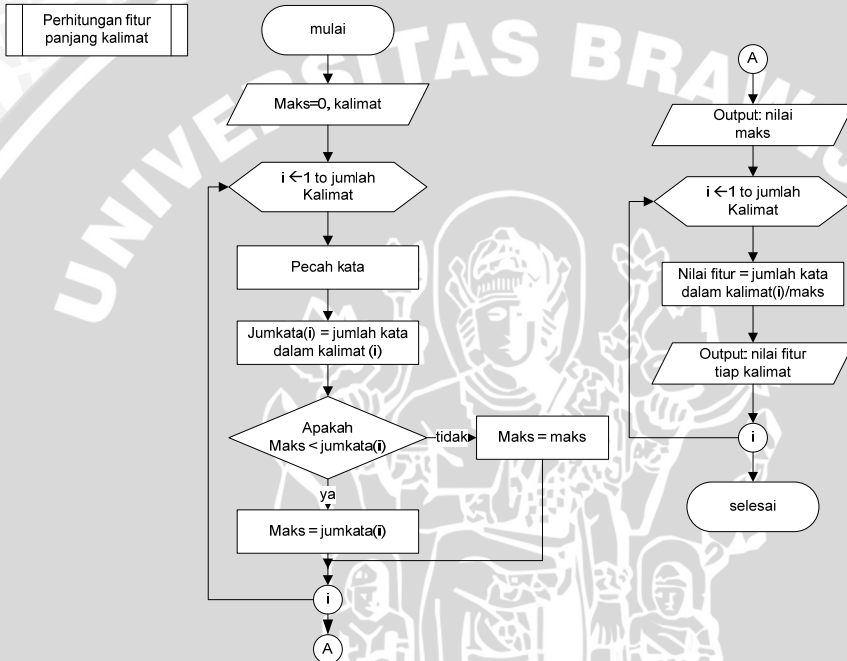
fitur ini berguna menyaring kalimat yang terlalu singkat. Kalimat yang terlalu singkat tidak diharapkan masuk dalam ringkasan, kalimat seperti ini biasanya berisi cuma *dateline* dan nama pengarang dalam artikel berita. Digunakan *ratio* perbandingan antar kalimat. Perhitungan nilai bobot yaitu perbandingan jumlah kata dalam kalimat dibagi dengan jumlah kata dalam kalimat yang terpanjang.

Langkah perhitungan fitur panjang kalimat sebagai berikut:

1. inisialisasi variabel maks = 0
2. untuk tiap kalimat hitung jumlah kata dalam kalimat
3. jika nilai maks < jumlah kata dalam kalimat ke-i maka nilai maks

diganti dengan jumlah kata dalam kalimat ke- i , jika tidak maka nilai maks tetap

4. didapat nilai maks
5. untuk tiap kalimat hitung nilai fitur *sentence length* dengan rumus jumlah kata dalam kalimat ke- i dibagi maks
6. didapat nilai fitur *sentence length* tiap kalimat



Gambar 3.7 Diagram Alir Proses Perhitungan Fitur Panjang Kalimat

c. Posisi Kalimat (*Sentence Position*)

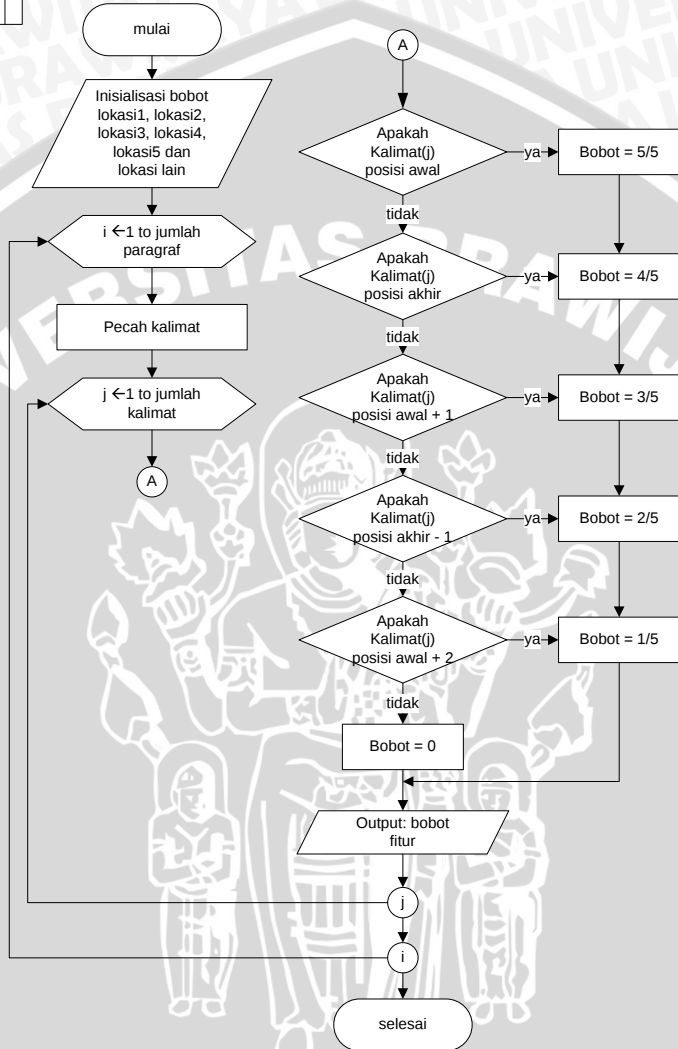
Posisi kalimat juga turut berpengaruh dalam letak kalimat topik. Dalam paragraf berbahasa indonesia kalimat topik terletak pada awal kalimat, akhir kalimat maupun di tengah paragraf. Karena sulitnya mengidentifikasi secara pasti dan jaranganya penggunaan kalimat topik yang terletak ditengah, maka pada penelitian ini hanya akan menggunakan sistem rangking berdasarkan pada posisi kalimat yang diadopsi dari adopsi aturan letak kalimat topik pada paragraf bahasa indonesia dimana nilai tertinggi terletak pada kalimat pertama, nilai tertinggi kedua terletak di akhir kalimat, nilai tertinggi ke tiga terletak

pada kalimat ke dua, nilai tertinggi ke empat terletak pada kalimat akhir dikurangi 1. Begitu juga seterusnya akan berselang - selang dimulai dari awal kalimat kemudian ke akhir kalimat dilanjut ke awal kalimat ditambah 1 dan dilanjut ke akhir kalimat dikurangi 1, sehingga didapat nilai terendah berada pada kalimat yang posisinya ditengah paragraf. Aturan yang dipakai :

$$F3(K) = \begin{array}{l} 5/5 \text{ untuk posisi awal,} \\ 4/5 \text{ untuk posisi akhir,} \\ 3/5 \text{ untuk posisi awal} + 1, \\ 2/5 \text{ untuk posisi akhir} - 1, \\ 1/5 \text{ untuk posisi awal} + 2, \\ 0/5 \text{ untuk posisi yang lain.} \end{array}$$

Langkah perhitungan fitur panjang kalimat sebagai berikut:

1. menginisialisasi bobot tiap posisi kalimat
2. melakukan perulangan sejumlah banyaknya paragraf
3. memecah kalimat dalam tiap paragraf
4. melakukan pengecekan pada posisi kalimat
5. jika kalimat pada posisi akhir kasih nilai 5/5
6. jika kalimat pada posisi akhir kasih nilai 4/5
7. jika kalimat pada posisi awal + 1 kasih nilai 3/5
8. jika kalimat pada posisi akhir - 1 kasih nilai 2/5
9. jika kalimat pada posisi awal + 2 kasih nilai 1/5
10. jika kalimat pada posisi lain kasih nilai 0



Gambar 3.8 Diagram Alir Proses Perhitungan Fitur Posisi Kalimat

d. Bobot Kata (*term weight*)

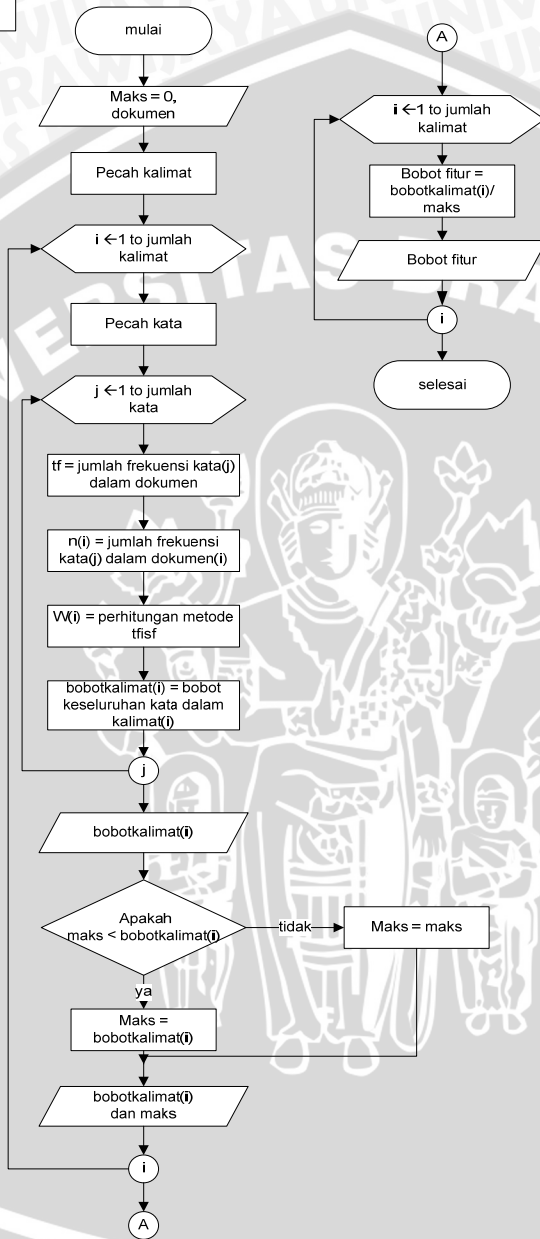
Bobot kalimat dapat dihitung dari jumlah tiap kata penyusunya, dengan menerapkan metode *tf.isf* (*Term Frequency Inverse Sentence Frequency*) dapat dihitung nilai tiap kata.

Langkah perhitungan fitur panjang kalimat sebagai berikut:

1. menginisialisasi nilai maks = 0
2. memecah dokumen menjadi kalimat
3. melakukan perulangan sejumlah banyaknya kalimat
4. memecah kalimat menjadi kata
5. menghitung frekuensi kemunculan tiap kata dalam dokumen
6. menghitung frekuensi kemunculan kata ke-j dalam kalimat ke-i
7. melakukan perhitungan rumus *tfidf*
8. bobot kalimat ke-i didapat dari jumlah bobot kata ke-j
9. mencari nilai tertinggi dari kalimat ke-i dan dijadikan nilai maks
10. bobot fitur diperoleh dari bobot kalimat ke-i dibagi maks



Perhitungan fitur
bobot kata



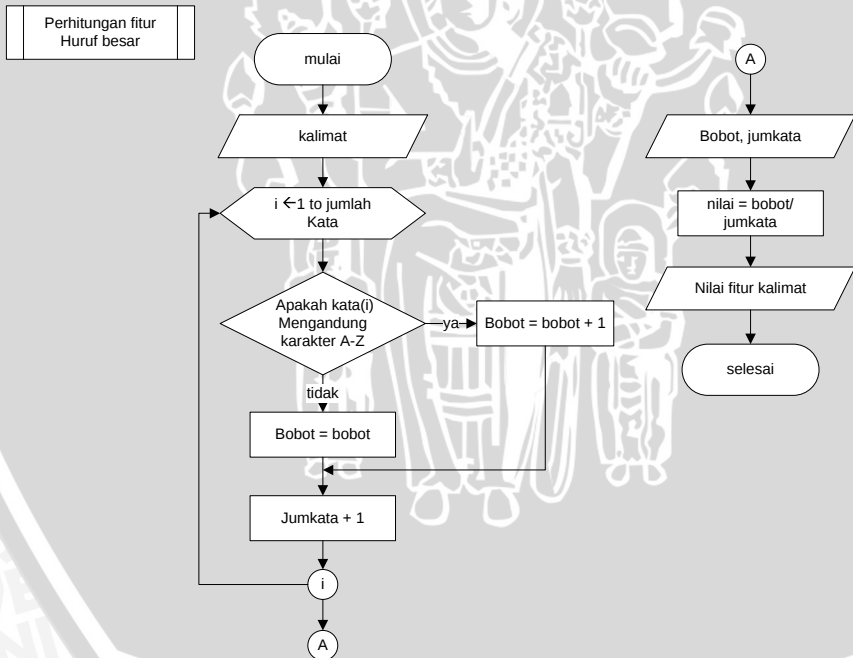
Gambar 3.9 Diagram Alir Proses Perhitungan Fitur Bobot Kata

e. Huruf Besar (*Upper-case Word*)

Kata yang mengandung huruf besar mempunyai peluang termasuk dalam kata yang dipentingkan. Perhitungan rumusnya merupakan rasio antara jumlah kata yang mengandung huruf besar dibagi dengan jumlah kata dalam kalimat.

Langkah perhitungan fitur panjang kalimat sebagai berikut:

1. melakukan perulangan sejumlah banyaknya kata
2. memelakukan pengecekan apakah kata ke-i mengandung huruf besar, jika ya maka nilai bobot akan ditambah 1 sedangkan jika tidak ditambah
3. menghitung jumlah kata dalam kalimat
4. melakukan pengecekan apakah bobot kalimat sama dengan nol, jika ya maka nilai fitur sama dengan 1/jumlah kata, jika tidak maka nilai fitur sama dengan bobot kalimat dibagi jumlah kata dalam kalimat



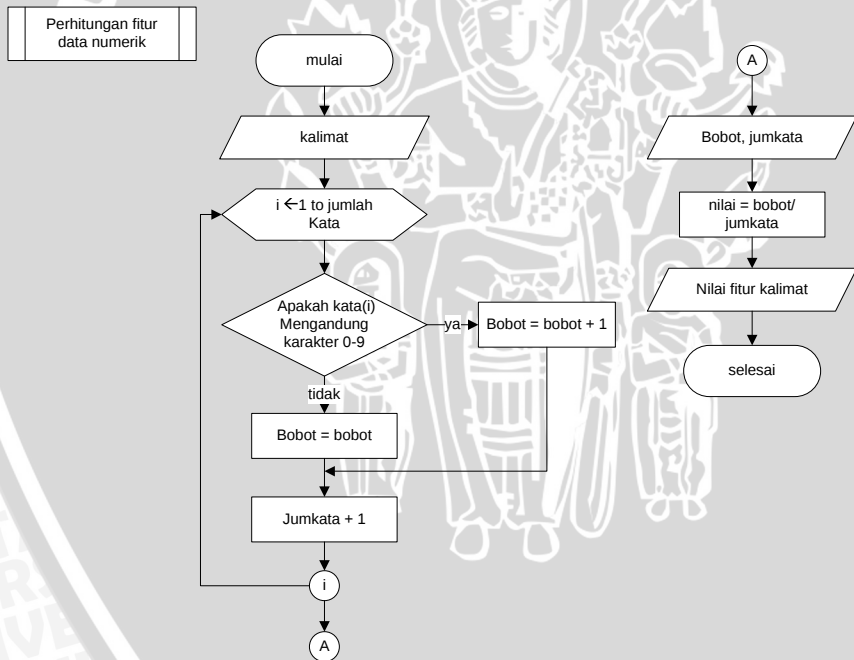
Gambar 3.10 Diagram Alir Proses Perhitungan Fitur Huruf Besar

f. Data Numerik (*numerical data*)

Kalimat yang terdapatnya terdapat data berupa bilangan mempunyai kemungkinan termasuk dalam kalimat ringkasan. Bobot nilai fitur ini dihitung dari banyaknya jumlah kata yang terdapat bilangan dalam kalimat dibagi dengan panjang kalimat.

Langkah perhitungan fitur panjang kalimat sebagai berikut:

1. melakukan perulangan sejumlah banyaknya kata
2. melakukan pengecekan apakah kata ke-i mengandung angka, jika ya maka nilai bobot akan ditambah 1 sedangkan jika tidak ditambah
3. menghitung jumlah kata dalam kalimat
4. melakukan pengecekan apakah bobot kalimat sama dengan nol, jika ya maka nilai fitur sama dengan 1/jumlah kata, jika tidak maka nilai fitur sama dengan bobot kalimat dibagi jumlah kata dalam kalimat

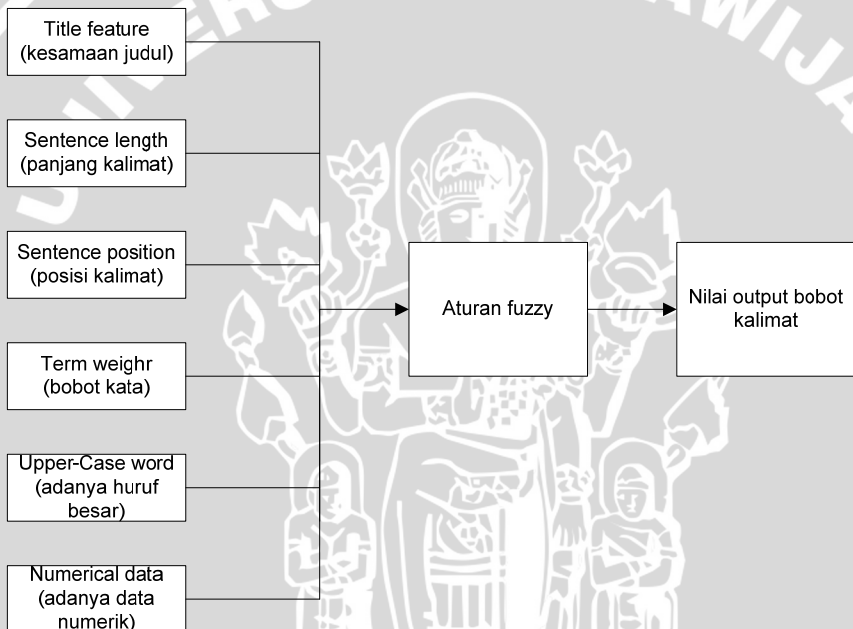


Gambar 3.11 Diagram Alir Proses Perhitungan Fitur Data Numerik

3.2.2 Logika Fuzzy

3.2.2.1 Desain Sistem Fuzzy

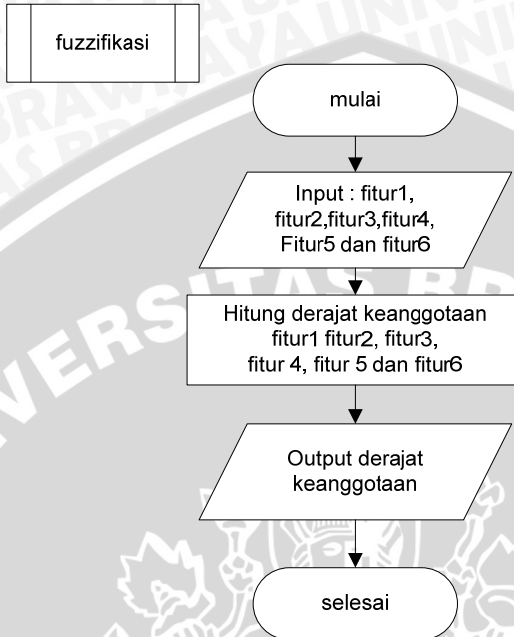
Penentuan kalimat yang akan menjadi kalimat ringkasan didasarkan pada nilai fitur tiap kalimat. Fitur yang digunakan antara lain, kemiripan dengan judul (*title feature*), panjang kalimat (*sentence length*), bobot istilah (*term weight*), posisi kalimat (*sentence position*), adanya huruf besar (*Upper-Case Word*) dan data numerik (*numerical data*). Desain sistem Fuzzy dalam penentuan kandidat kalimat ringkasan dapat dilihat pada gambar 3.xx.



Gambar 3.12 Sistem *Fuzzy* Penentuan Nilai Bobot Kalimat

3.2.2.2 Proses Fuzzyfikasi

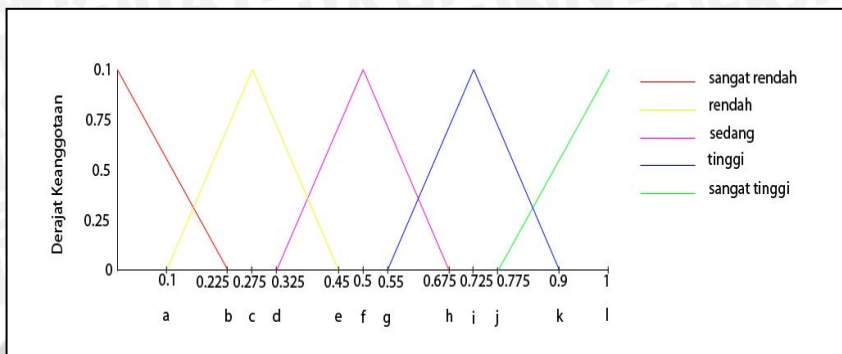
Pada proses ini nilai *crisp input* dari fitur kalimat akan diubah menjadi *fuzzy input* berdasarkan fungsi keanggotaan masing – masing fitur.



Gambar 3.13 Diagram Alir Proses Fuzzyfikasi

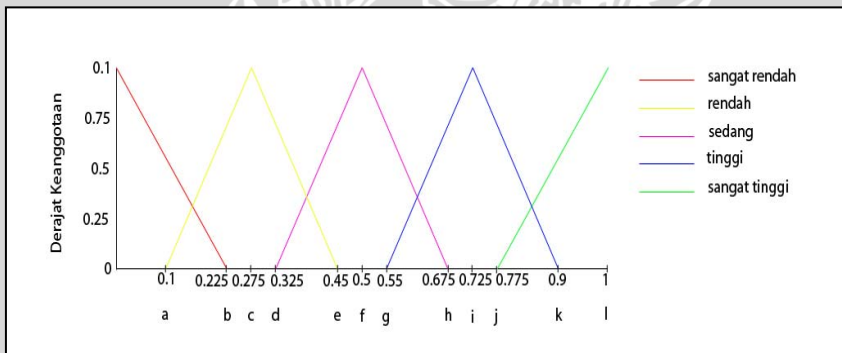
Terdapat 6 *crisp input* yang akan diubah menjadi *fuzzy input* antara lain: kemiripan dengan judul (*title feature*), panjang kalimat (*sentence length*), bobot istilah (*term weight*), posisi kalimat (*sentence position*), adanya huruf besar (*Upper-Case Word*) dan data numerik (*numerical data*). Fitur ini sebagai variabel fuzzy diperoleh dari beberapa jurnal mengenai *sentence based extraction*.

Variabel kesamaan judul (*title feature*) terdiri dari 5 kelompok yaitu sangat rendah, rendah, sedang, tinggi dan sangat tinggi. Fungsi keanggotaan yang dipakai adalah representasi kurva segitiga. Derajat keanggotaan untuk variabel kesamaan judul (*title feature*) yaitu :



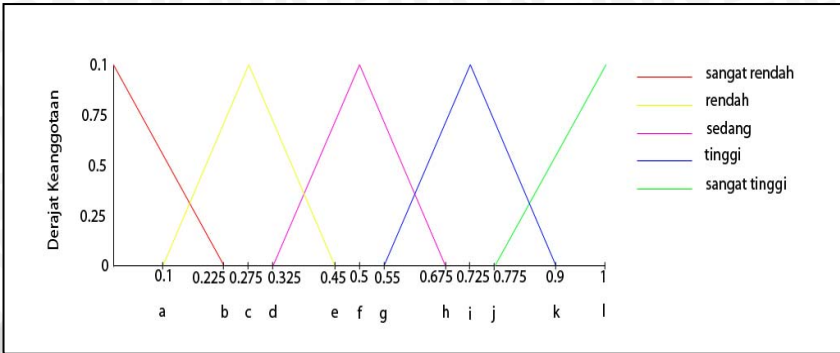
Gambar 3.14 Himpunan *Fuzzy Variabel Kesamaan Judul (Title Feature)*

Variabel panjang kalimat (*sentence length*) terdiri dari 5 kelompok yaitu sangat rendah, rendah, sedang, tinggi dan sangat tinggi. Fungsi keanggotaan yang dipakai adalah representasi kurva segitiga. Derajat keanggotaan untuk variabel panjang kalimat (*sentence length*) yaitu :



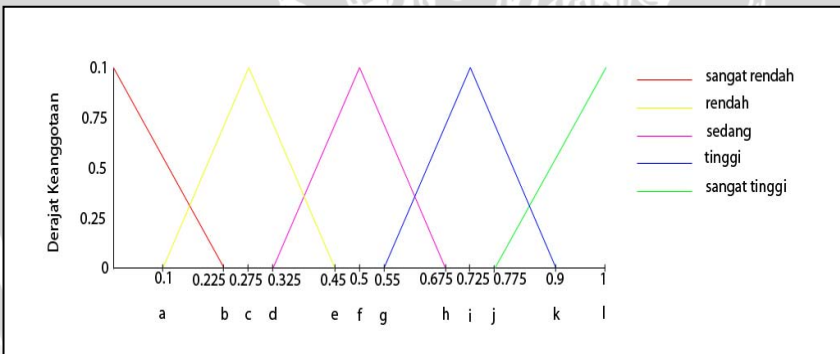
Gambar 3.15 Himpunan *Fuzzy Variabel Panjang Kalimat (Sentence Length)*

Variabel posisi kalimat (*sentence position*) terdiri dari 5 kelompok yaitu sangat rendah, rendah, sedang, tinggi dan sangat tinggi. Fungsi keanggotaan yang dipakai adalah representasi kurva segitiga. Derajat keanggotaan untuk variabel posisi kalimat (*sentence position*) yaitu :



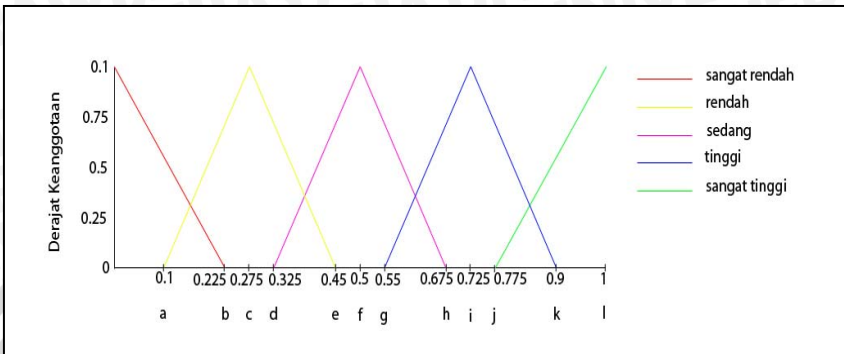
Gambar 3.16 Himpunan *Fuzzy Variabel Posisi Kalimat (Sentence Position)*

Variabel bobot kata (*term weight*) terdiri dari 5 kelompok yaitu sangat rendah, rendah, sedang, tinggi dan sangat tinggi. Fungsi keanggotaan yang dipakai adalah representasi kurva segitiga. Derajat keanggotaan untuk variabel bobot kata (*term weight*) yaitu :



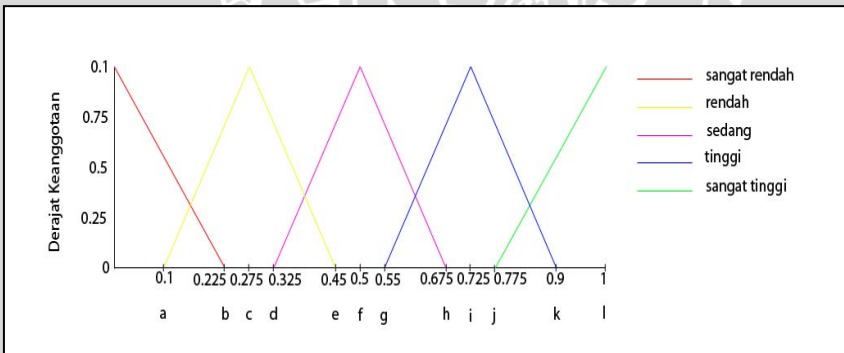
Gambar 3.17 Himpunan *Fuzzy Variabel Bobot Kata (Term Weight)*

Variabel huruf besar (*upper-case word*) terdiri dari 5 kelompok yaitu sangat rendah, rendah, sedang, tinggi dan sangat tinggi. Fungsi keanggotaan yang dipakai adalah representasi kurva segitiga. Derajat keanggotaan untuk variabel huruf besar (*upper-case word*) yaitu :



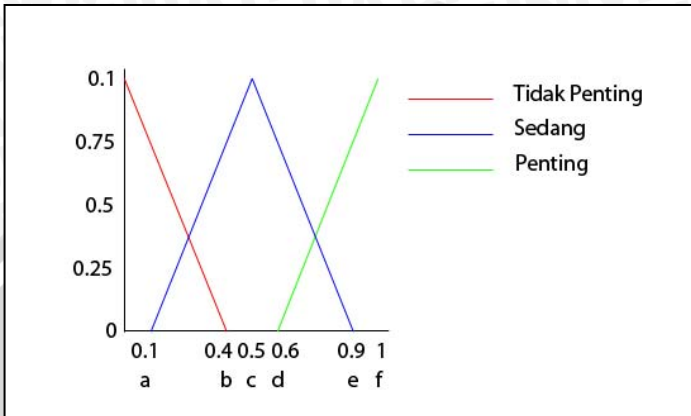
Gambar 3.18 Himpunan *Fuzzy* Variabel Huruf Besar (*Upper-Case Word*)

Variabel data numerik (*numerical data*) terdiri dari 5 kelompok yaitu sangat rendah, rendah, sedang, tinggi dan sangat tinggi. Fungsi keanggotaan yang dipakai adalah representasi kurva segitiga. Derajat keanggotaan untuk variabel data numerik (*numerical data*) yaitu :



Gambar 3.19 Himpunan *Fuzzy* Variabel Data Numerik (*Numerical Data*)

Sedangkan untuk himpunan variabel *output* berupa nilai tingkat kepentingan kalimat sebagai kalimat ringkasan. Fungsi keanggotaan yang digunakan berupa representasi kurva segitiga dengan nama variabel tidak penting, sedang dan penting.



Gambar 3.20 Himpunan Fuzzy Variabel Output

3.2.2.3 Rule Fuzzy

Aturan (*rule*) yang dipakai dalam *inferensi fuzzy* diperoleh melalui konsultasi pakar (Guru Bahasa Indonesia) serta acuan pada jurnal penelitian yang dilakukan oleh Ladda Suanmali, Naomie Salim dan Mohamed Salem Binwahlen tahun 2009.

Aturan yang dipakai antara lain :

[R1] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Sangat Rendah AND Data Numerik = Sangat Rendah THEN Kalimat = Tidak Penting

[R2] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Sangat Rendah AND Data Numerik = Rendah THEN Kalimat = Tidak Penting

[R3] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Sangat Rendah AND Data Numerik = Sedang THEN Kalimat = Tidak Penting

[R4] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Sangat Rendah AND Data Numerik = Tinggi THEN Kalimat = Tidak Penting

[R5] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Sangat Rendah AND Data Numerik = Sangat Tinggi THEN Kalimat = Tidak Penting

[R6] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Rendah AND Data Numerik = Sangat Rendah THEN Kalimat = Tidak Penting

[R7] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Rendah AND Data Numerik = Rendah THEN Kalimat = Tidak Penting

[R8] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Rendah AND Data Numerik = Sedang THEN Kalimat = Tidak Penting

[R9] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Rendah AND Data Numerik = Tinggi THEN Kalimat = Tidak Penting

[R10] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Rendah AND Data Numerik = Sangat Tinggi THEN Kalimat = Sedang

[R11] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Sedang AND Data Numerik =

Sangat Rendah THEN Kalimat = Tidak Penting

[R12] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Sedang AND Data Numerik = Rendah THEN Kalimat = Tidak Penting

[R13] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Sedang AND Data Numerik = Sedang THEN Kalimat = Tidak Penting

[R14] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Sedang AND Data Numerik = Tinggi THEN Kalimat = Sedang

[R15] IF Kesamaan Judul = Sangat Rendah AND Panjang Kalimat = Sangat Rendah AND Posisi Kalimat = Sangat Rendah AND Bobot Kata = Sangat Rendah AND Huruf Besar = Sedang AND Data Numerik = Sangat Tinggi THEN Kalimat = Sedang

.....
.....

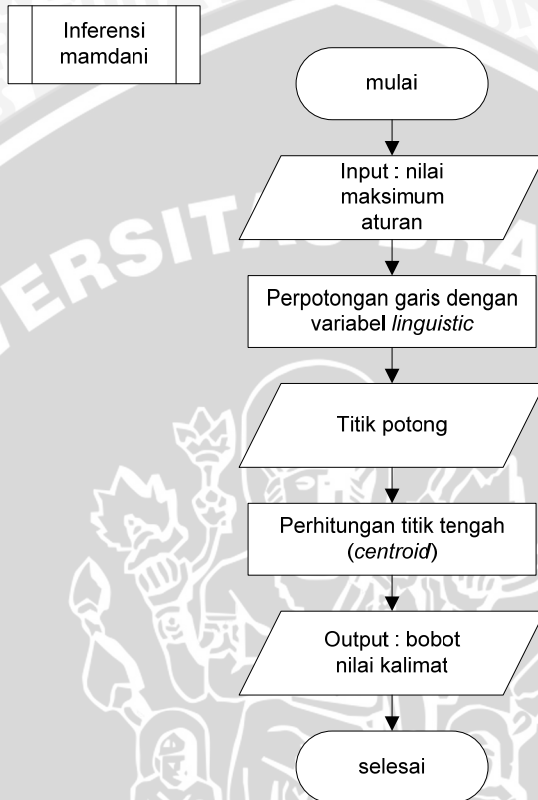
[R16624] IF Kesamaan Judul = Sangat Tinggi AND Panjang Kalimat = Sangat Tinggi AND Posisi Kalimat = Sangat Tinggi AND Bobot Kata = Sangat Tinggi AND Huruf Besar = Sangat Tinggi AND Data Numerik = Tinggi THEN Kalimat = Penting

[R16625] IF Kesamaan Judul = Sangat Tinggi AND Panjang Kalimat = Sangat Tinggi AND Posisi Kalimat = Sangat Tinggi AND Bobot Kata = Sangat Tinggi AND Huruf Besar = Sangat Tinggi AND Data Numerik = Sangat Tinggi THEN Kalimat = Penting

3.2.2.4 Inferensi Mamdani

Aturan yang dibuat akan disimpan dalam *database*. Pada proses inferensi mamdani, aturan tersebut digunakan dan pengambilan keputusan pada model mamdani menggunakan metode *Min-Max*. Kemudian fungsi komposisi yang digunakan adalah fungsi *Max*, proses

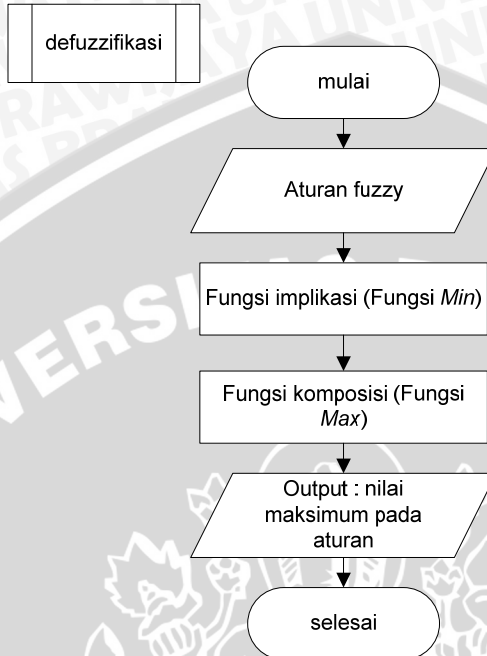
defuzzifikasi menggunakan metode centroid.



Gambar 3.21 Diagram Alir Proses Inferensi Model Mamdani

3.2.2.5 Defuzzifikasi

Proses defuzzifikasi adalah proses pengubahan *crisp input* menjadi *fuzzy input*. Metode defuzzifikasi yang dipakai pada komposisi aturan Mamdani, yaitu metode *centroid (center of gravity)*



Gambar 3.22 Diagram Alir Proses Defuzzifikasi

3.2.3 Perancangan Tabel

Dari perancangan proses yang telah dipaparkan di atas, maka diperlukan beberapa file eksternal untuk menyimpan sejumlah data yang diperlukan. Dalam hal ini dibutuhkan *database* dengan 2 tabel. Tabel pertama berupa tabel *stoplist*, digunakan untuk menyimpan kata – kata yang dianggap tidak bermanfaat. Kolom yang digunakan *id_stoplist* dan *stop_list*, detail atribut dapat dilihat pada tabel 3.1.

Tabel 3.1 Tabel *StopWord*

Kolom	Tipe	Panjang	Allow Null
id_stop_word	int(<i>primary key</i>)	4	No
stop_word	Varchar	50	No

Sedangkan untuk tabel kedua yaitu tabel *aturan fuzzy* digunakan untuk menyimpan sejumlah aturan – aturan dalam *inference system*, kolom yang digunakan meliputi *id_aturan*, *aturan_1*, *aturan_2*, *aturan_3*,

aturan_4, aturan_5 dan hasil. Mengenai detail tabel dapat dilihat pada tabel 3.2.

Tabel 3.2 Tabel Aturan *Fuzzy*

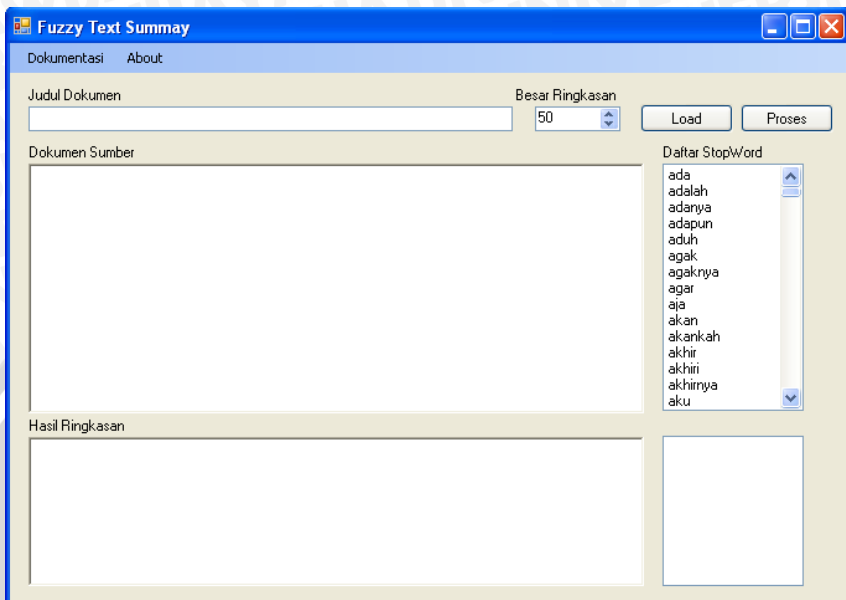
Kolom	Tipe	panjang	Allow Null
id_aturan	int(<i>Primar Key</i>)	4	No
Aturan_1	Varchar	20	No
Aturan_2	Varchar	20	No
Aturan_3	Varchar	20	No
Aturan_4	Varchar	20	No
Aturan_5	Varchar	20	No
Hasil	Varchar	20	No

3.3 Perancangan Antarmuka

Sistem ini akan diterapkan pada komputer personal (PC). Secara garis besar sistem akan membaca file sumber yang diberikan user kemudian akan mengekstrak informasi yang ada di dalamnya dan dilakukan proses pembobotan serta perhitungan menggunakan logika fuzzy untuk mendapatkan ringkasan.

Adapun bagian yang dibutuhkan dalam proses *summary* sebagai berikut:

1. **Field Judul Dokumen** : untuk mengisikan Judul dokumen yang akan diproses.
2. **Tombol Load** : untuk membuka dokumen dengan *open dialog* yaitu bantuan bagi *user* untuk memilih dokumen.
3. **Tombol Proses** : untuk memulai proses peringkasan dokumen
4. **Field Dokumen Sumber** : Tempat menampilkan dokumen asli
5. **Field Hasil Ringkasan** : Tempat menampilkan hasil ringkasan
6. **ListBox Daftar Stopword** : untuk menampilkan daftar *stopword*.
7. **listBox Besar Ringkasan** : untuk parameter besarnya hasil ringkasan sebesar 25% atau 50% dari dokumen sumber



Gambar 3.23 Form Perancangan Aplikasi Peringkas Dokumen

3.4 Perancangan Pengujian

Pada subbab ini akan dibahas perancangan uji coba baik uji coba pada perangkat lunak maupun evaluasi pada hasil ringkasan sistem. Evaluasi ringkasan meliputi evaluasi yang berdasarkan ringkasan ideal (dalam hal ini ringkasan yang dibuat oleh manusia yang ahli dibidangnya) dan akan dibandingkan dengan hasil sistem lain.

3.4.1 Bahan pengujian

Bahan yang digunakan dalam dalam pengujian kali ini berasal dari teks berita yang diambil secara acak dari sekumpulan berita Kompas yang diterbitkan pada bulan Januari 2001 hingga bulan Desember 2001. Versi *offline* dari bahan pengujian dapat diperoleh dari alamat <http://ilps.science.uva.nl/datafiles/bahasaindonesia/kompas.zip>.

3.4.2 Skenario dan Kriteria Pengujian

3.4.2.1 Evaluasi Hasil Ringkasan

Untuk mengetahui kualitas hasil ringkasan sistem, maka digunakan ringkasan ideal (ringkasan buatan manusia) sebagai acuan. Seperti yang dijelaskan pada tinjauan pustaka bahwa ringkasan dinilai ideal

merupakan hal sulit mengingat tidak adanya definisi dan parameter yang jelas mengenai ringkasan ideal. Namun dikarenakan tujuan pembuatan ringkasan ini dimaksudkan agar hasil ringkasan yang diperoleh mendekati hasil ringkasan buatan manusia, maka ringkasan ideal diperoleh dari ringkasan yang dibuat manusia. Berikut komponen dalam pembuatan ringkasan ideal (buatan manusia).

- *Abstraktor* : Guru SMK Mata Pelajaran Bahasa Indonesia
- Metode : Sejumlah teks berita berbahasa indonesia akan diberikan oleh *abstraktor* yang ahli dibidangnya, kemudian tugas *abstraktor* tersebut akan memilih kalimat mana saja yang akan dijadikan kalimat ringkasan.

Tabel 3.3 Informasi Berita

No.	Id_Berita	Judul	Jumlah paragraf	Jumlah kalimat

Tabel 3.4 Hasil Ringkasan Abstraktor

No.	Id_berita	Ringkasan (No. Kalimat)	Jumlah kalimat	Besar ringkasan

Tabel 3.5 Hasil Ringkasan Sistem

No.	Id_berita	Ringkasan (No. Kalimat)	Jumlah kalimat	Besar ringkasan

Tabel 3.6 Hasil ringkasan *Auto Summarize*

No.	Id_berita	Ringkasan (No. Kalimat)	Jumlah kalimat	Besar ringkasan

3.4.2.2 Perbandingan Sistem

Hasil ringkasan yang dibandingkan adalah hasil ringkasan sistem dengan *Autosummarizer* pada *Microsoft Word 2007*. Perhitungan perbandingannya menggunakan nilai *precision* dan *recall* yang dijelaskan pada tinjauan pustaka. Parameter yang menjadi acuan perbandingan adalah besarnya hasil ringkasan yaitu 25% dan 50% dari dokumen sumber

Tabel 3.7 Perbandingan Hasil Ringkasan Sistem dengan *Auto Summarize* Sebesar 25%

No.	Id_Berita	<i>Precision</i> sistem	<i>Recall</i> sistem	<i>Precision</i> <i>Autosummarizer</i>	<i>Recall</i> <i>Autosummarizer</i>

Tabel 3.8 Perbandingan Hasil Ringkasan Sistem dengan *Auto Summarize* Sebesar 50%

No.	Id_Berita	<i>Precision</i> sistem	<i>Recall</i> sistem	<i>Precision</i> <i>Autosummarizer</i>	<i>Recall</i> <i>Autosummarizer</i>

3.5 Perhitungan Manual

Berikut ini contoh perhitungan yang dilakukan sistem dalam membentuk ringkasan dari sebuah teks:

Baju Bekas Impor Masih Banjir ke Indonesia

Walaupun ada pelarangan baju bekas impor masuk ke Indonesia, namun penyelundupan pakaian bekas masih tetap marak. Awal tahun ini, Bea dan Cukai menggagalkan penyelundupan 2.200 bal baju bekas impor.

Penyelundupan terjadi di perairan Tanjung Balai Karimun, Provinsi Kepulauan Riau. Baju bekas itu diselundupkan dengan menggunakan kapal. "Tiap bal berisi antara 200-300 lembar baju bekas. Total nilai baju bekas itu mencapai Rp 2,8 miliar," kata Dirjen Bea dan Cukai, Thomas Sugijata, Kamis (7/1/2010).

Selain kegagalan awal tahun, Bea dan Cukai menggagalkan 11 kasus penyelundupan pakaian bekas sepanjang tahun 2009. Total, ada 17.775 bal atau setara dengan 4,5 juta helai pakaian bekas yang disita. Pakaian bekas itu bernilai Rp 18 miliar.

Kendati demikian, perdagangan pakaian bekas masih marak di sejumlah pasar di Indonesia.

Tahap pertama penghilangan stopword (dijelaskan pada bab 3.2.1.1.c) dan steaming (dijelaskan pada bab 3.2.1.1.d) dan dilakukan pemisahan per kalimat, didapatkan

Kalimat Judul : Baju Bekas Impor Masih Banjir ke Indonesia

Kalimat1 : larang baju bekas impor Indonesia selundup pakai bekas tetap marak

Kalimat 2 : Awal tahun Bea Cukai gagal selundup 2.200 bal baju bekas impor

Kalimat 3 : selundup air Tanjung Balai Karimun Provinsi Kepulauan Riau

Kalimat 4 : Baju bekas selundup guna kapal

Kalimat 5 : bal isi 200-300 lembar baju bekas

Kalimat 6 : nilai baju bekas capai Rp 2,8 miliar kata Dirjen Bea Cukai
Thomas Sugijata Kamis 7/1/2010

Kalimat 7 : gagal awal tahun Bea Cukai gagal 11 kasus selundup pakai
bekas tahun 2009

Kalimat 8 : Total 17.775 bal setara 4,5 juta helai pakai bekas sita

Kalimat 9 : Pakai bekas nilai Rp 18 miliar

Kalimat 10 : dagang pakai bekas masih marak pasar Indonesia

Tabel 3.9 Frekuensi Kata

Kata	Jumlah	Kata	jumlah	Kata	Jumlah
11	1	Dagang	1	masih	1
18	1	Dirjen	1	miliar	2
2009	1	Gagal	3	Nilai	2
2200	1	Guna	1	pakai	5
17775	1	Helai	1	pasar	1
7/1/2010	1	Impor	2	provinsi	1
28	1	Indonesia	2	Riau	1
200-300	1	Isi	1	Rp	2
45	1	Juta	1	selundup	5
Air	1	Kamis	1	setara	1
Awal	2	Kapal	1	Sita	1
Baju	5	Karimun	1	sugijati	1
Bal	3	Kasus	1	tahun	3
Balai	1	Kata	1	tanjung	1
Bea	3	Kepulauan	1	tetap	1
Bekas	10	Larang	1	thomas	2
Capai	1	Lembar	1	Total	2
Cukai	3	Marak	2		

Tabel 3.10 Nilai Fitur per Kalimat

	F1	F2	F3	F4	F5	F6
K1	0.8	0.625	1	0.57896125	0.090909091	0
K2	0.6	0.6875	0.8	0.833567678	0.272727273	0.090909091
K3	0	0.5625	1	0.438619439	0.666666667	0

K4	0.4	0.3125	0.8	0.281983999	0.2	0
K5	0.4	0.375	0.6	0.349424668	0	0.166666667
K6	0.4	1	0.4	0.963953389	0.5	0.125
K7	0.2	0.8125	1	1	0.142857143	0.142857143
K8	0.2	0.625	0.8	0.617659975	0.1	0.2
K9	0.2	0.375	0.6	0.423095009	0.333333333	0.166666667
K10	0.4	0.4375	1	0.433621689	0	0

Keterangan:

F1 = fitur 1 (kesamaan judul)

F2 = fitur 2 (panjang kalimat)

F3 = fitur 3 (posisi kalimat)

F4 = fitur 4 (bobot kata)

F5 = fitur 5 (adanya huruf besar)

F6 = fitur 6 (adanya angka)

K(i) = kalimat(i)

Perhitungan Kalimat Pertama

dari masing – masing nilai crips fitur kalimat pertama akan diubah menjadi input fuzzy

- fitur 1 (kesamaan judul) = 0.8 termasuk dalam variabel tinggi dan sangat tinggi.

Derajat keanggotaan untuk variabel tinggi

$$\mu_{\text{tinggi}}[0.8] = (0.9-0.8)/(0.9-0-0.725) = 0.57142857$$

Derajat keanggotaan untuk variabel sangat tinggi

$$\mu_{\text{sangattinggi}}[0.8] = (0.8-0.775)/(1-0.775) = 0.111111111$$

- fitur 2 (panjang kalimat) = 0.625 termasuk dalam variabel sedang dan tinggi.

Derajat keanggotaan untuk variabel sedang

$$\mu_{\text{sedang}}[0.625] = (0.675-0.625)/(0.675-0.5) = 0.285714286$$

Derajat keanggotaan untuk variabel tinggi

$$\mu_{\text{tinggi}}[0.625] = (0.625-0.55)/(0.725-0.55) = 0.428571429$$

- fitur 3 (posisi kalimat) = 1 termasuk dalam variabel sangat tinggi

Derajat keanggotaan untuk variabel sangat tinggi

$$\mu_{\text{sangattinggi}}[1] = 1$$

- fitur 4 (bobot kata) = 0.57896125 termasuk dalam variabel sedang dan tinggi

Derajat keanggotaan untuk variabel sedang

$$\mu_{\text{sedang}}[0.57896125] = (0.675 - 0.57896125) / (0.675 - 0.5) = 0.548792857$$

Derajat keanggotaan untuk variabel tinggi

$$\mu_{\text{tinggi}}[0.57896125] = (0.57896125 - 0.55) / (0.725 - 0.55) = 0.165492857$$

- fitur 5 (adanya huruf besar) = 0.090909091 termasuk dalam variabel sangat rendah

Derajat keanggotaan untuk variabel sangat rendah

$$\mu_{\text{sangatrendah}}[0.090909091] = (0.225 - 0.090909091) / (0.225 - 0) = 0.595959596$$

- fitur 6 (adanya angka) = 0.090909091 termasuk dalam variabel sangat rendah

Derajat keanggotaan untuk variabel sangat rendah

$$\mu_{\text{sangatrendah}}[0] = 1$$

Rule (Aturan) Fuzzy yang dipakai sebanyak 8 yaitu:

IF Kesamaan Judul = Sangat Tinggi (0.111111111) AND Panjang Kalimat = Tinggi (0.428571429) AND Posisi Kalimat = Sangat Tinggi (1) AND Bobot Kata = Tinggi (0.165492857) AND Huruf Besar = Sangat Rendah (0.595959596) AND Data Numerik = Sangat Rendah (1) THEN Kalimat = Penting (0.111111111)

IF Kesamaan Judul = Sangat Tinggi (0.111111111) AND Panjang Kalimat = Tinggi (0.428571429) AND Posisi Kalimat = Sangat Tinggi (1) AND Bobot Kata = Sedang (0.548792857) AND Huruf Besar = Sangat Rendah (0.595959596) AND Data Numerik = Sangat Rendah (1) THEN Kalimat = Penting (0.111111111)

IF Kesamaan Judul = Sangat Tinggi (0.111111111) AND Panjang Kalimat = Sedang (0.285714286) AND Posisi Kalimat = Sangat Tinggi (1) AND Bobot Kata = Tinggi (0.165492857) AND Huruf Besar =

Sangat Rendah (0.595959596) AND Data Numerik = Sangat Rendah (1)
THEN Kalimat = Penting (0.111111111)

IF Kesamaan Judul = Sangat Tinggi (0.111111111) AND Panjang Kalimat = Sedang (0.285714286) AND Posisi Kalimat = Sangat Tinggi (1) AND Bobot Kata = Sedang (0.548792857) AND Huruf Besar = Sangat Rendah (0.595959596) AND Data Numerik = Sangat Rendah (1) THEN Kalimat = Penting (0.111111111)

IF Kesamaan Judul = Tinggi (0.57142857) AND Panjang Kalimat = Tinggi (0.428571429) AND Posisi Kalimat = Sangat Tinggi (1) AND Bobot Kata = Tinggi (0.165492857) AND Huruf Besar = Sangat Rendah (0.595959596) AND Data Numerik = Sangat Rendah (1) THEN Kalimat = Penting (0.165492857)

IF Kesamaan Judul = Tinggi (0.57142857) AND Panjang Kalimat = Tinggi (0.428571429) AND Posisi Kalimat = Sangat Tinggi (1) AND Bobot Kata = Sedang (0.548792857) AND Huruf Besar = Sangat Rendah (0.595959596) AND Data Numerik = Sangat Rendah (1) THEN Kalimat = Penting (0.428571429)

IF Kesamaan Judul = Tinggi (0.57142857) AND Panjang Kalimat = Sedang (0.285714286) AND Posisi Kalimat = Sangat Tinggi (1) AND Bobot Kata = Tinggi (0.165492857) AND Huruf Besar = Sangat Rendah (0.595959596) AND Data Numerik = Sangat Rendah (1) THEN Kalimat = Penting (0.165492857)

IF Kesamaan Judul = Tinggi (0.57142857) AND Panjang Kalimat = Sedang (0.285714286) AND Posisi Kalimat = Sangat Tinggi (1) AND Bobot Kata = Sedang (0.548792857) AND Huruf Besar = Sangat Rendah (0.595959596) AND Data Numerik = Sangat Rendah (1) THEN Kalimat = Sedang (0.285714286)

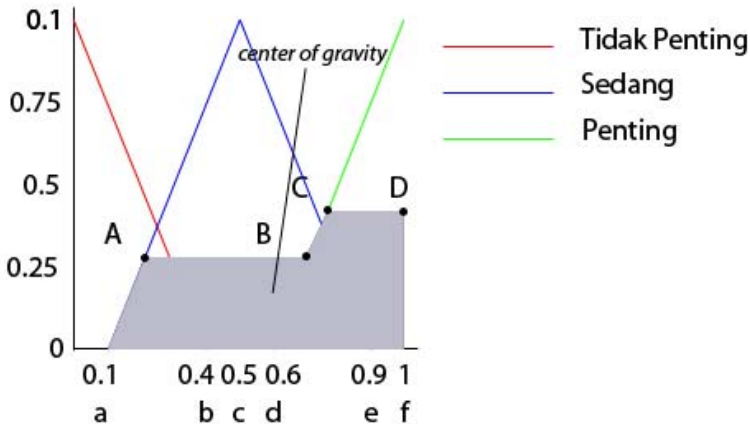
Menghasilkan

Penting (0.111111111), Penting (0.111111111), Penting (0.111111111),
Penting (0.111111111), Penting (0.165492857), Penting (0.428571429),
Penting (0.165492857) dan Sedang (0.285714286)

Keanggotaan diperoleh dengan mengambil nilai terbesar setiap elemen sehingga didapat

Penting (0.428571429) dan Sedang (0.285714286)

Proses komposisinya sebagai berikut



Gambar 3.24 Proses Defuzzifikasi dengan Metode Centroid

Mengacu pada persamaan (3.7) dan (3.8)

$$\begin{aligned} A &= (c - a) * y + a \\ &= (0.5 - 0.1) * 0.285714286 + 0.1 \\ &= 0.214285714 \end{aligned}$$

$$\begin{aligned} B &= (f - d) * y + d \\ &= (1 - 0.6) * 0.285714286 + 0.6 \\ &= 0.714285714 \end{aligned}$$

$$\begin{aligned} C &= (f - d) * y + d \\ &= (1 - 0.6) * 0.428571429 + 0.6 \\ &= 0.771428572 \end{aligned}$$

$$\begin{aligned} D &= f - (f - f) * y \\ &= 1 - (1 - 1) * 0.428571429 \\ &= 1 \end{aligned}$$

Setelah diketahui titik potongnya maka dihitung nilai defuzzifikasinya

$$\begin{aligned} \mu &= \frac{(A + B) * 0.285714286 + (C + D) * 0.428571429}{2 * 0.285714286 + 2 * 0.428571429} \\ &= \frac{(0.214285714 + 0.714285714) * 0.285714286}{2 * 0.285714286 + 2 * 0.428571429} \\ &\quad + \frac{(0.771428572 + 1) * 0.428571429}{2 * 0.285714286 + 2 * 0.428571429} \\ &= \frac{0.265306123 + 0.759183674}{1.42857143} = 0.717142857 \end{aligned}$$

Maka didapat nilai bobot dari kalimat pertama adalah K1 = 0.717142857

Dengan cara yang sama setiap kalimat akan dihitung nilai bobotnya, kemudian akan diurutkan dimulai dari yang paling besar, pengambilan kalimat ringkasan diperoleh dari kalimat yang mempunyai bobot nilai tertinggi tergantung berapa persen besar ringkasan yang diinginkan.

Didapat nilai akhir kalimat:

K1 = 0.717142857

K2 = 0.734856453

K3 = 0.7

K4 = 0.671428571

K5 = 0.429439415

K6 = 0.885714285

K7 = 0.7

K8 = 0.671428571

K9 = 0.724603174

K10 = 0.646031746

sehingga didapat kalimat ringkasan:

Besar ringkasan sebesar 50% = K1, K2, K3, K6, K9

Besar ringkasan sebesar 25% = K2, K6, K9

Keterangan:

K = kalimat

UNIVERSITAS BRAWIJAYA



BAB IV

HASIL DAN PEMBAHASAN

4.1 Lingkungan Implementasi

Lingkungan implementasi yang dijelaskan pada subbab ini meliputi lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem peringkasan dokumen otomatis menggunakan ekstraksi kalimat dan Fuzzy Inference System model Mamdani adalah sebuah Notebook dengan spesifikasi:

1. Processor 1.6GHz Celeron
2. Memori 1024 MB DDR2
3. Kapasitas hard disk 40GB

4.1.2 Lingkungan Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem peringkasan dokumen otomatis menggunakan ekstraksi kalimat dan Fuzzy Inference System model Mamdani adalah :

1. Sistem Operasi Windows XP SP2.
2. Microsoft Visual Studio Express edition 2008 dengan bahasa pemrograman C#.
3. Microsoft Office Word 2007.

4.2 Implementasi Program

Pada subbab implementasi Program akan dijelaskan mengenai implementasi dari rancangan perangkat lunak yang dijelaskan sebelumnya.

4.2.1 Implementasi Text Preprocssing

4.2.1.1 Segmentasi

Tahap awal dari sistem peringkasan dokumen otomatis adalah melakukan segmentasi. Pada tahap ini sebuah dokumen text akan dipecah menjadi paragraf dan kalimat. Tahap segmentasi dapat ditunjukkan pada *sourcecode* 4.1

```

public void Segmentasi()
{
    int IP = 0, IK = 0;
    ParagraphAsli = new string[JumParagraph];
    KalimatAsli = new string[JumKalimat + 1];
    for (int i = 0; i < DokumenAsli.Length; i++)
    {
        //segmentasi paragraph
        if ((DokumenAsli[i] == '\n') && (DokumenAsli[i + 1] ==
'\n'))
        {
            IP = IP + 1;
        }
        else if (DokumenAsli[i] == '\n') { }
        else
        {
            ParagraphAsli[IP] = ParagraphAsli[IP] + DokumenAsli[i];
        }

        //segmentasi kalimat
        if ((DokumenAsli[i] == '.') &&
            (i < DokumenAsli.Length - 1) &&
            ((DokumenAsli[i + 1] == ' ') || (DokumenAsli[i + 1] ==
'\n'))))
        {
            IK = IK + 1;
        }
        else if ((DokumenAsli[i] == ' ') && (DokumenAsli[i - 1] ==
'.')) { }
        else if ((DokumenAsli[i] == '\n') && (DokumenAsli[i + 1] ==
'\n')) { }
        else if (DokumenAsli[i] == '\n') { }
        else
        {
            KalimatAsli[i] = KalimatAsli[i] + DokumenAsli[i];
        }
    }
    KalimatAsli[iK + 1] = JudulAsli;
}

```

SourceCode 4.1 Segmentasi Paragraf dan Kalimat

4.2.1.2 Menghapus Tanda Baca

Setelah dilakukan segmentasi paragraf dan kalimat, maka setiap kata akan dihilangkan tanda baca. Tahap penghilangan tanda baca dapat dilihat pada *SourceCode 4.2*

```

public ArrayList HapusTandaBaca(string[] BuffKalimat)
{
    ArrayList AL = new ArrayList();
    for (int i = 0; i < BuffKalimat.Length; i++ )
    {

```

```

string s = string.Empty;
//A-Z = 65-90, a-z = 97-122, 0-9 = 48-57 space = 32
for (int j = 0; j < BuffKalimat[i].Length; j++ )
{
    if (((int)(BuffKalimat[i][j]) >= 65) &&
        ((int)(BuffKalimat[i][j]) <= 90)) ||
        (((int)(BuffKalimat[i][j]) >= 97) &&
        ((int)(BuffKalimat[i][j]) <= 122)) ||
        (((int)(BuffKalimat[i][j]) >= 48) &&
        ((int)(BuffKalimat[i][j]) <= 57)))
    {
        s = s + BuffKalimat[i][j];
    }
    else if ((int)(BuffKalimat[i][j]) == 32)
    {
        s = s + ' ';
    }
    else { }
}
AL.Add(s);
}
return AL;
}

```

SourceCode 4.2 Menghapus Tanda Baca

4.2.1.3 Menghapus *Stopword*

Langkah selanjutnya adalah menghapus *Stopword*, *Stopword* dapat diartikan kata – kata umum yang dianggap tidak penting. Tahap penghilangan *stopword* dapat dilihat pada *SourceCode 4.3*.

```

public ArrayList HapusStopWord(ArrayList BuffKalimat)
{
    int a = 0;
    bool flag = false;
    string gabung = string.Empty;
    ArrayList AL = new ArrayList();
    ArrayList BufKata = new ArrayList();
    ArrayList TempListKata = new ArrayList();
    for (int i = 0; i < BuffKalimat.Count; i++ )
    {
        TempListKata = PisahKata(BuffKalimat[i].ToString());
        for (int j = 0; j < TempListKata.Count; j++ )
        {
            for (int k = 0; k < StopWord.Count; k++ )
            {
                if (string.Compare(TempListKata[j].ToString().ToLower(),
                    StopWord[k].ToString().ToLower()) == 0)
                {
                    flag = true;
                }
            }
        }
    }
}

```

```

        if (flag)
        {
            flag = false;
        }
        else
        {
            BufKata.Add(TempListKata[j]);
            a++;
        }
    }

    for (int j = 0; j < BufKata.Count; j++)
    {
        if (gabung == string.Empty)
        {
            gabung = BufKata[j].ToString();
        }
        else
        {
            gabung = gabung + ' ' + BufKata[j];
        }
    }

    BufKata.Clear();
    a = 0;
    AL.Add(gabung);
    gabung = string.Empty;
}

return AL;
}

```

SourceCode 4.3 Menghapus *StopWord*

4.2.1.4 Stemming

Sebagaimana yang telah dijelaskan pada bab 2, ada 5 tahapan dalam proses *stemming* yaitu:

1. Penanganan Terhadap Partikel Infleksional

```

public string StemSuffiksInfleksional1(string BuffText)
{
    string temp = string.Empty;
    int posisi;
    if (BuffText.ToLower().EndsWith("lah") ||
        BuffText.ToLower().EndsWith("pun") ||
        BuffText.ToLower().EndsWith("kah") ||
        BuffText.ToLower().EndsWith("tah"))
    {
        posisi = BuffText.Length - 3;
        temp = BuffText.Substring(0, posisi);
    }
    Else

```

```

{
    temp = BuffText;
}

return temp;
}

```

SourceCode 4.4 Stemming Partikel Infleksional

Potongan fungsi *stemming* diatas digunakan untuk menangani kata yang mempunyai akhiran “-lah”, “-kah”, “-pun”, “-tah”.

2. Penanganan Terhadap Kata Ganti Infleksional

```

public string stemmSuffiksInfleksional2(string BuffText)
{
    BuffText.ToLower();
    string temp = string.Empty;
    int posisi;
    if (BuffText.ToLower().EndsWith("ku") ||
    BuffText.ToLower().EndsWith("mu"))
    {
        posisi = BuffText.Length - 2;
        temp = BuffText.Substring(0, posisi);
    }
    else if (BuffText.ToLower().EndsWith("nya"))
    {
        posisi = BuffText.Length - 3;
        temp = BuffText.Substring(0, posisi);
    }
    else
    {
        temp = BuffText;
    }

    return temp;
}

```

SourceCode 4.5 Stemming Kata Ganti Infleksional

Potongan fungsi *stemming* diatas digunakan untuk menangani kata ganti kepunyaan yang mempunyai akhiran “-ku”, “-mu”, “-nya”.

3. Penanganan Terhadap Prefiks Derivasional Pertama

```

public string StemmPrefiksDerivasional1(string BuffText)
{
    string temp = string.Empty;
    int posisi;
    if (BuffText.ToLower().StartsWith("meng") ||
    BuffText.ToLower().StartsWith("peng"))
    {

```

```

    posisi = BuffText.Length - 4;
    temp = BuffText.Substring(4, posisi);
}
else if (BuffText.ToLower().StartsWith("meny") ||
BuffText.ToLower().StartsWith("peny"))
{
    posisi = BuffText.Length - 4;
    temp = BuffText.Substring(4, posisi);
    temp = 's' + temp;
}
else if (BuffText.ToLower().StartsWith("men") ||
BuffText.ToLower().StartsWith("mem") ||
BuffText.ToLower().StartsWith("pen") ||
BuffText.ToLower().StartsWith("pem") ||
BuffText.ToLower().StartsWith("ter"))
{
    posisi = BuffText.Length - 3;
    temp = BuffText.Substring(3, posisi);
    if (((BuffText.ToLower().StartsWith("mem")) ||
(BuffText.ToLower().StartsWith("pem")))) && ((temp[0] == 'a') ||
(temp[0] == 'i') || (temp[0] == 'u') || (temp[0] == 'e') ||
(temp[0] == 'o'))))
    {
        temp = 'p' + temp;
    }
}
else if (BuffText.ToLower().StartsWith("me") ||
BuffText.ToLower().StartsWith("di") ||
BuffText.ToLower().StartsWith("ke"))
{
    posisi = BuffText.Length - 2;
    temp = BuffText.Substring(2, posisi);
}
return temp;
}

```

SourceCode 4.6 Stemming Prefiks Derivasional Pertama

Potongan fungsi *stemming* diatas digunakan untuk menangani prefiks derivasional berawalan seperti “-me”, “-mem”, “-men”.

4. Penanganan Terhadap Prefiks Derivasional Kedua

```

public string StemPrefiksDerivasional2(string BuffText)
{
    string temp = string.Empty;
    int posisi;
    if (BuffText.ToLower().StartsWith("ber") ||
BuffText.ToLower().StartsWith("bel") ||
BuffText.ToLower().StartsWith("per") ||
BuffText.ToLower().StartsWith("pel"))
    {
        posisi = BuffText.Length - 3;
    }
}

```

```

    temp = BuffText.Substring(3, posisi);
}
else if (BuffText.ToLower().StartsWith("pe") ||
BuffText.ToLower().StartsWith("be"))
{
    posisi = BuffText.Length - 2;
    temp = BuffText.Substring(2, posisi);
}
return temp;
}

```

SourceCode 4.7 *Stemming* Prefiks Derivasional Kedua

Potongan fungsi *stemming* diatas digunakan untuk menangani prefiks derivasional berawalan “-ber”, “-bel”, “-peng”, “-me”.

5. Penanganan Terhadap *Sufiks* Derivasional

```

public string StemmsufiksDerivasional(string BuffText)
{
    string temp = string.Empty;
    int posisi;
    if(BuffText.ToLower().EndsWith("kan"))
    {
        posisi = BuffText.Length - 3;
        temp = BuffText.Substring(0, posisi);
    }
    else if (BuffText.ToLower().EndsWith("an"))
    {
        posisi = BuffText.Length - 2;
        temp = BuffText.Substring(0, posisi);
    }
    else if (BuffText.ToLower().EndsWith("i"))
    {
        posisi = BuffText.Length - 1;
        temp = BuffText.Substring(0, posisi);
    }
    else
    {
        temp = BuffText;
    }
    return temp;
}

```

SourceCode 4.8 *Stemming Sufiks* Derivasional

Potongan fungsi *stemming* diatas digunakan untuk menangani prefiks derivasional yang mempunyai awalan “kan”, “-an”, “-i”.

4.2.2 Implementasi Text Processing

4.2.2.1. Perhitungan Fitur Judul

Proses selanjutnya adalah perhitungan nilai pada fitur judul, dimana akan dihitung jumlah kata yang sama dengan judul pada tiap kalimat. Kata disini dibatasi bukan kata yang mempunyai arti yang sama (sinonim). Kalimat dapat dikategorikan sebagai kandidat kalimat ringkasan apabila semakin banyak kata yang sama dengan kalimat judul. Tahap perhitungan fitur judul dapat dilihat pada *Source Code* 4.9.

```
public double Fitur1(string BuffText)
{
    int TempNilai = 0;
    ArrayList TempListKata = new ArrayList();
    ArrayList TempListJudul = new ArrayList();
    TempListJudul = PisahKata(JudulProses);
    TempListKata = PisahKata(BuffText);
    Double Pembagi = TempListKata.Count;
    for (int i = 0; i < TempListKata.Count; i++ )
    {
        for (int j = 0; j < TempListJudul.Count; j++ )
        {
            if (string.Compare(TempListKata[i].ToString().ToLower(),
            TempListJudul[j].ToString().ToLower()) == 0)
            {
                TempNilai = TempNilai + 1;
            }
        }
    }
    TempNilai = TempNilai / Pembagi;
    return TempNilai;
}
```

SourceCode 4.9 Fitur Judul

Proses diatas mula – mula dilakukan perulangan sepanjang kalimat dan sepanjang kalimat judul. Apabila terdapat kata yang sama maka variabel TempNilai akan bertambah satu, artinya dalam kalimat tersebut terdapat sejumlah TempNilai kata yang sama dengan kalimat judul. Hasil akhir berupa nilai *return* perbandingan banyaknya kata yang sama dengan judul kalimat dengan panjang kalimat tersebut.

4.2.2.2 Perhitungan Fitur Panjang Kalimat

Perhitungan fitur selanjutnya adalah menghitung panjang suatu kalimat, semakin panjang kalimat maka kalimat tersebut semakin dapat menjadi kandidat dalam kalimat ringkasan.

```
public double Fitur2(string BuffText)
```

```

{
    double Max = 0, JumKataKalimat = 0, TempJumKata = 0;
    double TempNilai = 0;
    for (int i = 0; i < ALKalimat.Count; i++)
    {
        MatchCollection Kata = Regex.Matches(ALKalimat[i].ToString(),
        @"[\S]+");
        TempJumKata = Kata.Count;
        if (TempJumKata > Max)
        {
            Max = TempJumKata;
        }
    }
    MatchCollection KataBuff = Regex.Matches(BuffText, @"[\S]+");
    JumKataKalimat = KataBuff.Count;
    TempNilai = (JumKataKalimat / Max);
    return TempNilai;
}

```

SourceCode 4.10 Fitur Panjang Kalimat

Perhitungan fitur ini dimulai dengan mencari kalimat terpanjang. Setelah didapat maka nilai kembalian dari fungsi ini berupa perbandingan kalimat ke-i dengan kalimat terpanjang.

4.2.2.3 Perhitungan Fitur Posisi Kalimat

Posisi kalimat dalam Bahasa Indonesia menentukan letak dalam ide pokok dalam paragraf. Letak ide sendiri dapat di awal paragraf maupun diakhir. Perhitungan lebih jelas dapat dilihat pada Source Code 4.11.

```

public void Fitur3()
{
    ArrParagraph = new int[JumParagraph];
    for (int i = 0; i < ParagraphAsli.Length; i++)
    {
        for (int j = 0; j < ParagraphAsli[i].Length; j++)
        {
            if ((j != ParagraphAsli[i].Length - 1) &&
            (ParagraphAsli[i][j] == '.') && (ParagraphAsli[i][j + 1] == ' '))
            {
                ArrParagraph[i] = ArrParagraph[i] + 1;
            }
            else if ((j == ParagraphAsli[i].Length - 1) &&
            (ParagraphAsli[i][j] == '.'))
            {
                ArrParagraph[i] = ArrParagraph[i] + 1;
            }
        }
    }
    int awal, akhir;
}

```

```

int pembatas = 0;
double Pembagi;
for (int i = 0; i < JumParagraph; i++ )
{
    Pembagi = 5;
    awal = 0;
    akhir = ArrParagraph[i];
    for (int j = 0; j < ArrParagraph[i]; j++ )
    {
        if(( j + 1 ) % 2 == 1)
        {
            NilaiFitur[(pembatas + awal), 2] = Pembagi / 5;
            awal = awal + 1;
            Pembagi = Pembagi - 1;
            if (Pembagi < 0)
            {
                Pembagi = 0;
            }
        }
        else if ((j + 1) % 2 == 0)
        {
            NilaiFitur[(pembatas + akhir - 1), 2] = Pembagi / 5;
            akhir = akhir - 1;
            Pembagi = Pembagi - 1;
            if (Pembagi < 0)
            {
                Pembagi = 0;
            }
        }
    }
    pembatas = pembatas + ArrParagraph[i];
}
}

```

SourceCode 4.11 Fitur Posisi Kalimat

Perhitungan fitur ini dimulai dengan memisah teks menjadi paragraf dan setiap paragraf akan dihitung ada berapa kalimat didalamnya. Setelah proses ini selesai kemudian dilakukan *looping* sepanjang jumlah paragraf, dan didalam *looping* tersebut terdapat *looping* lagi sejumlah kalimat. Perhitungan nilai dari posisi kalimat dimulai disini, ketika kalimat tersebut berada pada posisi awal dalam paragraf maka nilainya tertinggi pertama, apabila posisinya pada akhir paragraf maka nilainya tertinggi kedua. Nilai terendah terdapat pada kalimat dengan posisi ditengah paragraf.

4.2.2.4 Perhitungan Fitur Bobot Kalimat

Bobot Kalimat didapat dari penjumlahan dari bobot tiap – tiap kata penyusunnya. Perhitungan lebih jelas dapat dilihat pada Source Code 4.12.

```
public double Fitur4(string BuffText)
{
    double [] tf, n, bobotkata;
    ArrayList TempListKataKalimat = new ArrayList();
    ArrayList TempListKataDokumen = new ArrayList();
    TempListKataKalimat = PisahKata(BuffText);
    TempListKataDokumen = PisahKata(FullDokumen);
    ArrayList TempListKataKalimat2 = new ArrayList();
    tf = new double[TempListKataKalimat.Count];
    n = new double[TempListKataKalimat.Count];
    bobotkata = new double[TempListKataKalimat.Count];
    double TempNilai = 0;
    double No = 0;
    for (int i = 0; i < TempListKataKalimat.Count; i++ )
    {
        for (int j = 0; j < ALKalimat.Count; j++)
        {
            TempListKataKalimat2 = PisahKata(ALKalimat[j].ToString());
            for (int k = 0; k < TempListKataKalimat2.Count; k++ )
            {
                if(string.Compare(TempListKataKalimat[i].ToString().ToLower(),
                TempListKataKalimat2[k].ToString().ToLower()) == 0)
                {
                    No = 1;
                }
            }
            n[i] = n[i] + No;
            No = 0;
        }
        for (int j = 0; j < TempListKataDokumen.Count; j++ )
        {
            if(string.Compare(TempListKataKalimat[i].ToString().ToLower(),
            TempListKataDokumen[j].ToString().ToLower()) == 0)
            {
                tf[i] = tf[i] + 1;
            }
        }
        bobotkata[i] = tf[i] * (Math.Log10(JumKalimat / n[i]));
        TempNilai = TempNilai + bobotkata[i];
    }

    for (int i = 0; i < JumKalimat; i++ )
    {
        if(NilaiFitur[i, 3] > MaxF4)
        {
            MaxF4 = NilaiFitur[i, 3];
        }
    }
}
```

```

    }
    for (int i = 0; i < JumKalimat; i++ )
    {
        NilaiFitur[i, 3] = NilaiFitur[i, 3] / MaxF4;
        listBox2.Items.Add(NilaiFitur[i, 3]);
    }
    Return TempNilai;
}

```

SourceCode 4.12 Fitur Bobot Kalimat

Perhitungan fitur ini dimulai dengan perhitungan berapa jumlah kata ke-i dalam dokumen tersebut. Kemudian dilanjut dengan menghitung nilai kata ke-i tersebut muncul di berapa kalimat dalam dokumen. Setelah didapat nilai tersebut kemudian bobot kata dapat dihitung dengan rumus. Bobot kalimat didapat dari penjumlahan dari masing – masing bobot kata penyusunya. Nilai Fitur yang didapat merupakan perbandingan bobot kalimat ke-i dengan bobot terbesar kalimat pada dokumen tersebut.

4.2.2.5 Perhitungan Fitur Huruf Besar

Kata yang mempunyai huruf besar berpeluang kata tersebut merupakan kata yang dipentingkan, semakin banyak kalimat yang mempunyai jumlah kata dengan huruf besar maka kalimat tersebut berpeluang besar sebagai kandidat ringkasan. Perhitungan lebih jelas dapat dilihat pada Source Code 4.13.

```

public double Fitur5(string BuffText)
{
    double TempNilai = 0, temp=0, PjKalimat;
    ArrayList TemplistKata = new ArrayList();
    TemplistKata = PisahKata(BuffText);
    for (int i = 0; i < TemplistKata.Count; i++ )
    {
        for (int j = 0; j < TemplistKata[i].ToString().Length; j++ )
        {
            if(((int)(TemplistKata[i].ToString()[j]) >= 65) &&
            ((int)(TemplistKata[i].ToString()[j]) <= 90 ))
            {
                temp = 1;
            }
        }
        TempNilai = TempNilai + temp;
        temp = 0;
    }
    MatchCollection KataBuff = Regex.Matches(BuffText, @"[\S]+");
}

```

```

PjKalimat = KataBuff.Count;
TempNilai = TempNilai / PjKalimat;
return TempNilai;
}

```

SourceCode 4.13 Fitur Huruf Besar

Perhitungan fitur ini dimulai dengan pemisahan kalimat menjadi array kata, kemudian setiap kata dicek apakah mempunyai karakter berhuruf besar. Nilai fitur ini didapat dari perbandingan antar jumlah huruf besar dalam kalimat dibandingkan dengan panjang kalimat.

4.2.2.6 Perhitungan Fitur Numerik

Kalimat yang mempunyai kata yang terdalamnya terapat bilangan numerik dapat diperhitungkan sebagai kalimat yang dipentingkan. Perhitungan lebih jelas dapat dilihat pada Source Code 4.14.

```

public double Fitur6(string BuffText)
{
    double TempNilai = 0, temp = 0, PjKalimat;
    ArrayList TempListKata = new ArrayList();
    TempListKata = PisahKata(BuffText);
    for (int i = 0; i < TempListKata.Count; i++)
    {
        for (int j = 0; j < TempListKata[i].ToString().Length; j++)
        {
            if (((int)(TempListKata[i].ToString()[j]) >= 48) &&
                ((int)(TempListKata[i].ToString()[j]) <= 57))
            {
                temp = 1;
            }
        }
        TempNilai = TempNilai + temp;
        temp = 0;
    }
    MatchCollection KataBuff = Regex.Matches(BuffText, @"[\S]+");
    PjKalimat = KataBuff.Count;
    TempNilai = TempNilai / PjKalimat;
    return TempNilai;
}

```

SourceCode 4.14 Fitur Numerik

Sama halnya dengan perhitungan fitur huruf besar, perhitungan fitur ini dimulai dengan pemisahan kalimat menjadi array kata, kemudian setiap kata dicek apakah mempunyai karakter berupa bilangan numerik. Nilai fitur ini didapat dari perbandingan antar jumlah bilangan numerik dalam kalimat dibandingkan dengan panjang kalimat.

4.2.3 Implementasi Logika Fuzzy

4.2.3.1 Perhitungan Derajat Keanggotaan

Fungsi diatas digunakan untuk menghitung derajat keanggotaan pada tiap – tiap fitur dalam setiap kalimat. Perhitungan lebih jelas dapat dilihat pada *Source Code* 4.15.

```
Public void HitungDerajatKeanggotaan(
{
    int k = 0;
    double temp;
    DerajatKeanggotaan = new double[JumKalimat,6,2];
    VarFuzzy = new string[JumKalimat, 6, 2];
    for (int i = 0; i < JumKalimat; i++)
    {
        for (int j = 0; j < 6; j++)
        {
            k = 0;
            if ((NilaiFitur[i, j] >= 0) && (NilaiFitur[i, j] <= 0.225))
            {
                DerajatKeanggotaan[i, j, k] = (0.225 - NilaiFitur[i, j]) /
(0.225 - 0);
                VarFuzzy[i, j, k] = "sangat rendah";
                k++;
            }
            if ((NilaiFitur[i, j] >= 0.1) && (NilaiFitur[i, j] <=
0.275))
            {
                DerajatKeanggotaan[i, j, k] = (NilaiFitur[i, j] - 0.1) /
(0.275 - 0.1);
                VarFuzzy[i, j, k] = "rendah";
                k++;
            }
            .....
            if ((NilaiFitur[i, j] >= 0.775) && (NilaiFitur[i, j] <= 1))
            {
                DerajatKeanggotaan[i, j, k] = (NilaiFitur[i, j] - 0.775) /
(1 - 0.775);
                VarFuzzy[i, j, k] = "sangat tinggi";
                k++;
            }
            if (VarFuzzy[i, j, 1] == null)
            {
                VarFuzzy[i, j, k] = "kosong";
                DerajatKeanggotaan[i, j, k] = -1;
            }
        }
    }
}
```

SourceCode 4.15 Perhitungan Derajat Keanggotaan

Perhitungan pada fungsi ini adalah menghitung nilai derajat keanggotaan setelah didapat nilai pada tiap – tiap fitur per kalimat. Nilai tersebut disimpan dalam array 3 dimensi dimana dimensi pertama menyatakan kalimat ke-i, dimensi kedua menyatakan fitur ke-j, dan dimensi ketiga menyatakan derajat keanggotaanya. Perhitungan diatas juga berlaku untuk menentukan variabel *fuzzy*.

4.2.3.2 Penentuan Aturan *Fuzzy*

Setelah didapat derajat keanggotaan pada tiap – tiap fitur perkalimat dan juga variabel *fuzzy*, langkah berikutnya yaitu menentukan kombinasi aturan dari variabel fuzzy di tiap fiturnya. Perhitungan lebih jelas dapat dilihat pada *Source Code* 4.16.

```
string Query = "select * from Aturan_Fuzzy where Aturan_1 = '" +
VarFuzzy[NoKalimat, 0, i] + "' and Aturan_2 = '" +
VarFuzzy[NoKalimat, 1, j] + "' and Aturan_3 = '" +
VarFuzzy[NoKalimat, 2, k] + "' and Aturan_4 = '" +
VarFuzzy[NoKalimat, 3, l] + "' and Aturan_5 = '" +
VarFuzzy[NoKalimat, 4, m] + "' and Aturan_6 = '" +
VarFuzzy[NoKalimat, 5, n] + '"";
OleDbCommand oleCommand = new OleDbCommand(Query, oleConnection);
try
{
    oleConnection.Open();
    OleDbDataReader oleDataReader = oleCommand.ExecuteReader();
    while (oleDataReader.Read())
    {
        TempALString.Add(oleDataReader.GetValue(7));
    }
    oleConnection.Close();
}
catch (OleDbException ex)
{
    textBox1.Text = ex.Message;
    oleConnection.Close();
}
Temp.Add(DerajatKeanggotaan[NoKalimat, 0, i]);
Temp.Add(DerajatKeanggotaan[NoKalimat, 1, j]);
Temp.Add(DerajatKeanggotaan[NoKalimat, 2, k]);
Temp.Add(DerajatKeanggotaan[NoKalimat, 3, l]);
Temp.Add(DerajatKeanggotaan[NoKalimat, 4, m]);
Temp.Add(DerajatKeanggotaan[NoKalimat, 5, n]);
TempDoubleAngka = FungsiMin(Temp);
TempALAngka.Add(TempDoubleAngka);
Temp.Clear();
FungsiMax(TempALAngka, TempALString);
```

SourceCode 4.16 Penentuan Aturan *Fuzzy*

Aturan *fuzzy* didapat dari kombinasi semua variabel *fuzzy* yang disimpan dalam *database*, jumlah aturan yang diperoleh pada tiap kalimat tidakla sama, tergantung kombinasinya. Tiap aturan tersebut mempunyai variabel *fuzzy output* yang nilainya didapat dari nilai minimum derajat keanggotaan fitur penyusunnya. fungsi max yang dipanggil digunakan untuk mencari nilai max derajat keanggotaan pada variabel output *fuzzy* yang telah dikelompokkan.

4.2.3.3 Defuzzifikasi

Proses defuzzifikasi adalah proses pengubahan *crisp input* menjadi *fuzzy input*. Untuk mentransformasikan *crisp input* menjadi *fuzzy input*, diperlukan *membership function* untuk tiap input. Perhitungan lebih jelas dapat dilihat pada *Source Code 4.17*.

```
public void Defuzzifikasi(ArrayList Angka, ArrayList Huruf)
{
    double A = 0, B = 0, C = 0, D = 0, TempNilaiAkhirFitur = 0;
    if(Angka.Count == 2)
    {
        if ((Convert.ToDouble(Angka[0])) >
        (Convert.ToDouble(Angka[1])))
        {
            if(Huruf[0].ToString() == "tidak penting")
            {
                A = 0;
                B = (0.4 - ((-1 * Convert.ToDouble(Angka[0])) * (0.4 - 0)));
                C = (0.4 - ((-1 * Convert.ToDouble(Angka[1])) * (0.4 - 0)));
                D = (0.9 - ((-1 * Convert.ToDouble(Angka[1])) * (0.9 -
                0.5)));
                TempNilaiAkhirFitur = (((A + B) *
                Convert.ToDouble(Angka[0])) + ((C + D) *
                Convert.ToDouble(Angka[1]))) / ((2 * Convert.ToDouble(Angka[0])) +
                (2 * Convert.ToDouble(Angka[1])));
            }
            else if(Huruf[0].ToString() == "sedang")
            {
                A = ((Convert.ToDouble(Angka[0]) * (0.5 - 0.1)) + 0.1);
                B = (0.9 - ((-1 * Convert.ToDouble(Angka[0])) * (0.9 -
                0.5)));
                C = (0.9 - ((-1 * Convert.ToDouble(Angka[1])) * (0.9 -
                0.5)));
                D = 1;
                TempNilaiAkhirFitur = (((A + B) *
                Convert.ToDouble(Angka[0])) + ((C + D) *
                Convert.ToDouble(Angka[1]))) / ((2 * Convert.ToDouble(Angka[0])) +
                (2 * Convert.ToDouble(Angka[1])));
            }
        }
    }
}
```

```

}
.....
.....

else if (Angka.Count == 1)
{
    if(Huruf[0].ToString() == "tidak penting")
    {
        A = 0;
        B = (0.4 - ((-1 * Convert.ToDouble(Angka[0])) * (0.4 - 0)));
        TempNilaiAkhirFitur = ((A + B) *
Convert.ToDouble(Angka[0])) / (2 * Convert.ToDouble(Angka[0]));
    }
    else if (Huruf[0].ToString() == "sedang")
    {
        A = ((Convert.ToDouble(Angka[0]) * (0.5 - 0.1)) + 0.1);
        B = (0.9 - ((-1 * Convert.ToDouble(Angka[0])) * (0.9 -
0.5)));
        TempNilaiAkhirFitur = ((A + B) * Convert.ToDouble(Angka[0]))
/ (2 * Convert.ToDouble(Angka[0]));
    }
    else if (Huruf[0].ToString() == "penting")
    {
        A = ((Convert.ToDouble(Angka[0]) * (1 - 0.6)) + 0.6);
        B = 1;
        TempNilaiAkhirFitur = ((A + B) * Convert.ToDouble(Angka[0]))
/ (2 * Convert.ToDouble(Angka[0]));
    }
}
TempALBobotKalimat.Add(TempNilaiAkhirFitur);
}

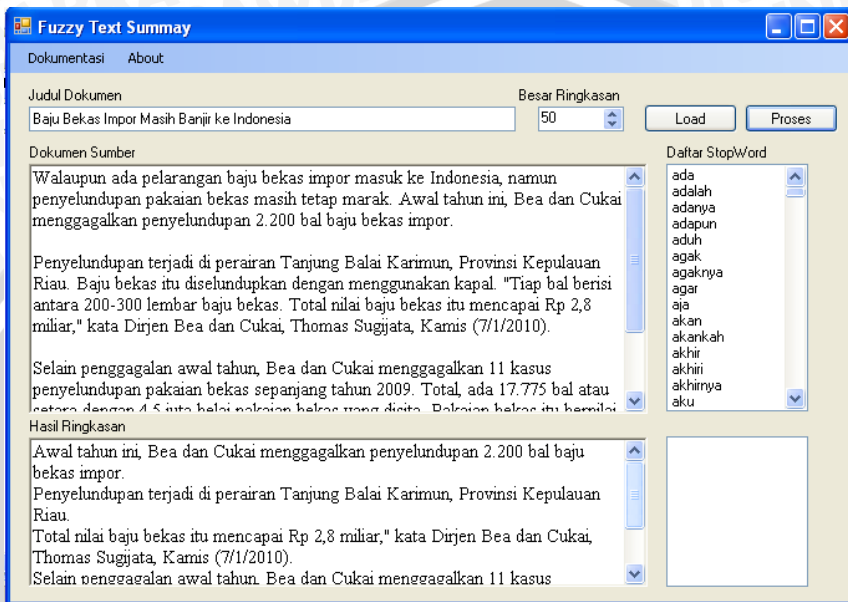
```

SourceCode 4.17 Defuzzifikasi

Defuzzifikasi digunakan untuk menentukan nilai akhir bobot kalimat, perhitungan defuzzifikasi diperoleh dari *crisp input* diubah menjadi *fuzzy input* menggunakan *membership function* untuk grafik *output* untuk grafik *output*. Pada defuzzifikasi ini digunakan metode *centroid* dimana setelah didapatkan titik perpotongan dengan *membership function* maka masing – masing perpotongan titik tersebut akan dihitung nilai tengah sesuai dengan rumus pada subbab 2.6. hasil yang didapat akan disimpan dalam TempBobotKalimat hingga didapat semua nilai bobot kalimat.

4.3 Implementasi Antar Muka

Berdasarkan rancangan antar muka pada subbab 3.3, maka dihasilkan gambar antar muka yang ditunjukkan pada gambar 4.1 berikut.



Gambar 4.1 Implementasi Antar Muka

4.4 Implementasi Uji Coba

Pada subbab ini akan dijelaskan mengenai implementasi uji coba sistem seperti yang telah dijelaskan pada subbab 3.4, pada pengujian ini akan dibandingkan antara ringkasan sistem dengan ringkasan dari *Autosummarizer* (Microsoft Word 2007), dengan pembandingnya adalah ringkasan buatan manusia dalam hal ini adalah orang yang ahli dibidangnya (Guru Bahasa Indonesia).

4.4.1 Skenario Uji Coba

Berdasarkan skenario pengujian pada subbab 3.4.2, maka pengujian sistem akan dilakukan 2 kali. Pertama akan dilakukan pengujian dengan besar ringkasan sebesar 50% dari dokumen asal dan pengujian kedua dilakukan pengujian besar ringkasan sebesar 25% dari dokumen asal. Perhitungan ini dimaksudkan untuk menghitung nilai *precision* dan

recall baik dari hasil ringkasan sistem maupun ringkasan hasil *Autosummarizer* seperti yang telah dijelaskan pada subbab 2.8.

Pengujian ini dilakukan terhadap 15 dokumen berita yang dipilih secara random masing – masing akan dibuat ringkasannya sebesar 50% dan sebesar 25%. Baik dari hasil ringkasan sistem maupun ringkasan dari *Autosummarizer* kemudian akan dibandingkan dengan hasil ringkasan manusia (seorang *abstractor* guru Bahasa Indonesia). Maksud utama pengujian ini adalah membandingkan apakah hasil ringkasan sistem lebih baik daripada hasil ringkasan *Autosummarizer*.

4.4.2 Hasil Uji Coba

4.4.2.1 Pengujian Hasil Ringkasan Sebesar 25%

Pada pengujian ini dilakukan perhitungan nilai *precision* dan *recall* pada masing – masing dokumen yang dengan besar ringkasan sebesar 25% dari dokumen asal dengan nilai pembandingnya berasal dari hasil ringkasan buatan manusia. Adapun hasilnya dapat dilihat pada tabel 4.1.

Tabel 4.1 Perbandingan Hasil Ringkasan Sistem dengan *Auto Summarizer* Sebesar 25%

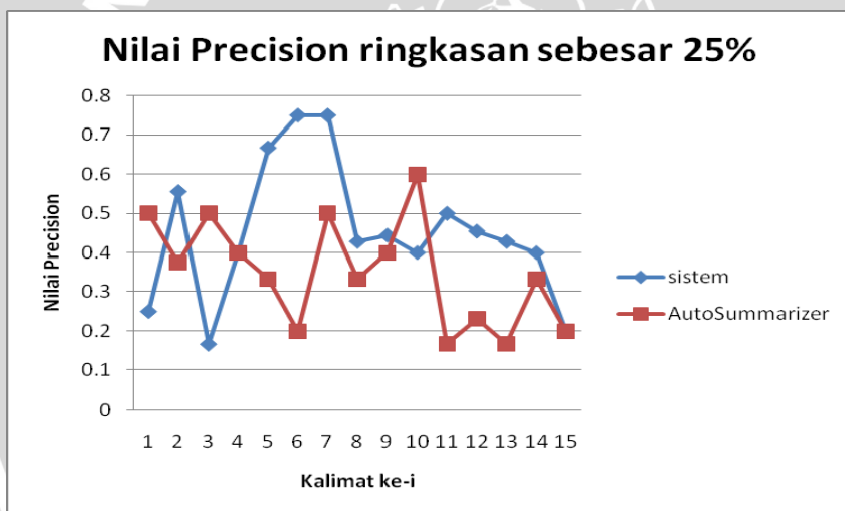
No	Id_berita	<i>Precision</i> <i>sistem</i>	<i>Recall</i> <i>sistem</i>	<i>Precision</i> <i>Auto</i> <i>Summarizer</i>	<i>Recall</i> <i>Auto</i> <i>Summarizer</i>
1	BR001	0.25	0.25	0.5	0.5
2	BR002	0.555556	0.555556	0.375	0.375
3	BR003	0.166667	0.166667	0.5	0.5
4	BR004	0.4	0.4	0.4	0.4
5	BR005	0.666667	0.666667	0.333333	0.333333
6	BR006	0.75	0.75	0.2	0.25
7	BR007	0.75	0.75	0.5	0.5
8	BR008	0.428571	0.6	0.333333	0.4
9	BR009	0.444444	0.4	0.4	0.4
10	BR010	0.4	0.4	0.6	0.6
11	BR011	0.5	0.6	0.166667	0.25
12	BR012	0.454545	0.454545	0.230769	0.272727
13	BR013	0.428571	0.428571	0.166667	0.142857
14	BR014	0.4	0.4	0.333333	0.2
15	BR015	0.2	0.2	0.2	0.2

Dari data tabel diatas, dapat dihitung nilai rata – rata dari *precision* dan *recall* yang disajikan dalam tabel 4.2.

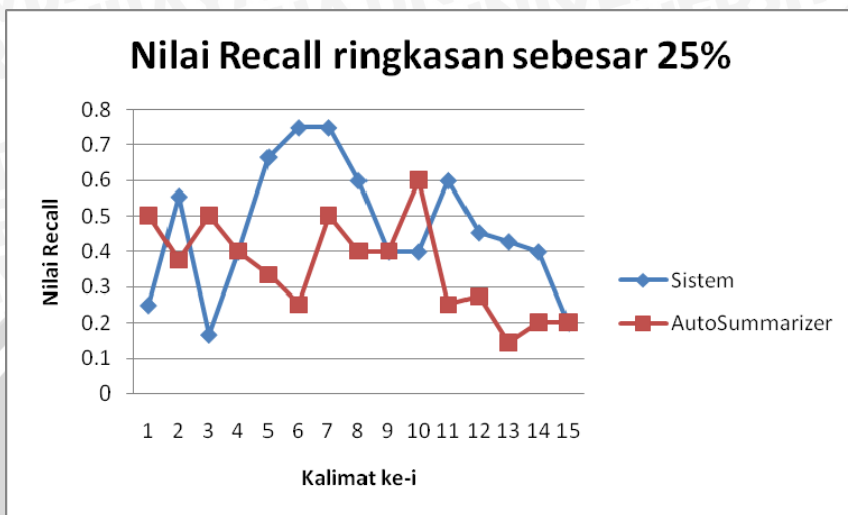
Tabel 4.2 Nilai Rata – rata *Precisoin* dan *Recall* Ringkasan Sistem dengan *Auto Summarize* Sebesar 25%

<i>Precision sistem</i>	<i>Recall sistem</i>	<i>Precision Auto Summarizer</i>	<i>Recall Auto Summarizer</i>
0.453001	0.468134	0.349274	0.354928

Hasil perhitungan pada tabel 4.1 dapat disajikan dalam bentuk grafik seperti pada gambar 4.2 dan 4.3.



Gambar 4.2 Grafik Nilai *Precision* Sebesar 25%



Gambar 4.3 Grafik Nilai *Recall* Sebesar 25%

4.4.2.2 Pengujian Hasil Ringkasan Sebesar 50%

Pada pengujian ini dilakukan perhitungan nilai precision dan recall pada masing – masing dokumen yang dengan besar ringkasan sebesar 50% dari dokumen asal dengan nilai pembandingnya berasal dari hasil ringkasan buatan manusia. Adapun hasilnya dapat dilihat pada tabel 4.3.

Tabel 4.3 Perbandingan Hasil Ringkasan Sistem dengan *Auto Summarize* sebesar 50%

No	Id_berita	<i>Precision sistem</i>	<i>Recall sistem</i>	<i>Precision Auto Summarize</i>	<i>Recall Auto Summarize</i>
1	BR001	0.666667	0.571429	0.833333	0.714286
2	BR002	0.611111	0.647059	0.470588	0.470588
3	BR003	0.636364	0.583333	0.333333	0.333333
4	BR004	0.777778	0.777778	0.444444	0.444444
5	BR005	0.727273	0.727273	0.545455	0.545455
6	BR006	0.5	0.5	0.625	0.625
7	BR007	0.625	0.625	0.5	0.5
8	BR008	0.538462	0.636364	0.428571	0.545455
9	BR009	0.473684	0.5	0.6	0.666667

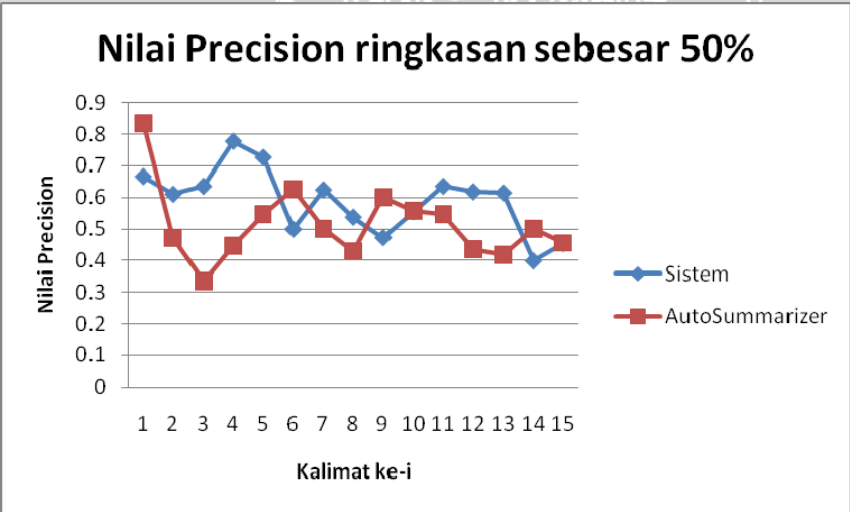
10	BR010	0.555556	0.555556	0.555556	0.555556
11	BR011	0.636364	0.7	0.545455	0.6
12	BR012	0.619048	0.590909	0.434783	0.454545
13	BR013	0.615385	0.666667	0.416667	0.416667
14	BR014	0.4	0.444444	0.5	0.444444
15	BR015	0.454545	0.5	0.454545	0.5

Dari data tabel diatas, dapat dihitung nilai rata – rata dari *precision* dan *recall* yang disajikan dalam tabel 4.4

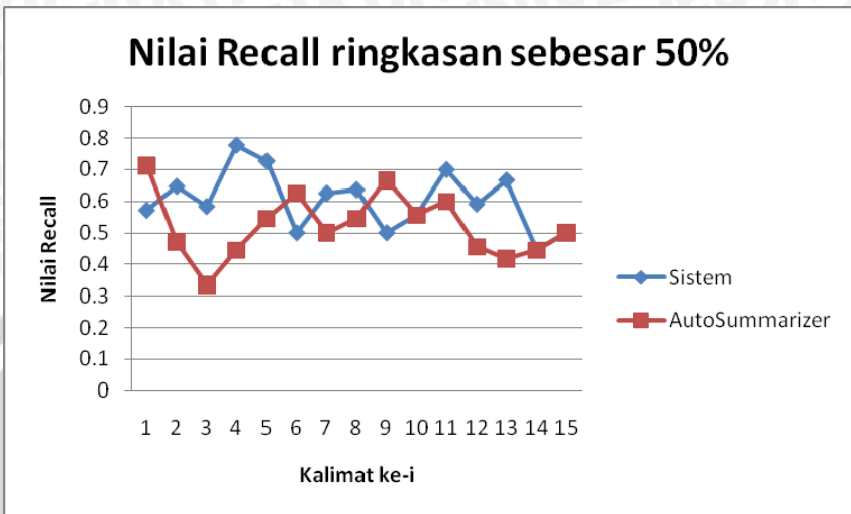
Tabel 4.4 Nilai Rata – rata *Precisoin* dan *Recall* Ringkasan Sistem dengan *Auto Summarizer* Sebesar 50%

<i>Precision sistem</i>	<i>Recall sistem</i>	<i>Precision Auto Summarize</i>	<i>Recall Auto Summarize</i>
0.589149	0.601721	0.512515	0.521096

Hasil perhitungan pada tabel 4.3 dapat disajikan dalam bentuk grafik seperti pada gambar 4.4 dan 4.5.



Gambar 4.4 Grafik Nilai *Precision* Sebesar 50%



Gambar 4.5 Grafik Nilai *Recall* Sebesar 50%

4.4.3 Analisa Hasil

Secara keseluruhan sistem peringkasan dokumen yang menggunakan metode ekstraksi kalimat dan digabungkan dengan logika *Fuzzy* model mamdani menghasilkan nilai rata – rata dari *precision* sebesar 0.453001 dan *recall* sebesar 0.468134 untuk besar ringkasan sebesar 25% dari dokumen asal. Sedangkan untuk besar ringkasan sebesar 50% menghasilkan nilai rata - rata *precision* 0.589149 dan *recall* 0.601721. nilai tersebut masih lebih baik jika dibandingkan dengan hasil ringkasan yang dibuat dari *Autosummarizer* yang menghasilkan nilai rata – rata dari *precision* sebesar 0.349274 dan *recall* 0.354928 untuk besar ringkasan sebesar 25% dan untuk hasil ringkasan sebesar 50% menghasilkan nilai rata – rata *precision* sebesar 0.512515 dan *recall* 0.521096.

Dari hasil uji yang dilakukan, dapat diketahui bahwa nilai rata – rata *precision* dan *recall* untuk hasil ringkasan sistem lebih besar dibandingkan nilai *precision* dan *recall* dari hasil ringkasan *AutoSummarizer*. Sedangkan untuk hasil besar ringkasan sebesar 50% nilai rata – rata *precision* dan *recall* lebih besar dibaningkan dengan hasil ringkasan sebesar 25%.

Ada beberapa hal yang mempengaruhi hasil uji coba tersebut, antara lain metode *stemming* yang digunakan, karena metode *stemming* yang

digunakan hanya memotong awalan dan akhiran tanpa adanya pengecekan berupa kata dasar atau bukan. Misalkan kata “tabrakan” akan menghasilkan kata dasar “tabra” karena dipotong berupa akhiran “kan”, kata tersebut akan mempunyai arti yang berbeda dengan “menabrak” padahal bentuk kata dasar keduanya sama. Kekurangan pada metode ini akan berpengaruh pada fitur kesamaan judul dan fitur bobot kata, karena kedua fitur ini sama – sama menghitung nilai kesamaan antar kata.

Jumlah kalimat dalam satu paragraf juga berpengaruh pada hasil uji coba sistem, pada data uji coba dokumen berita banyak ditemui beberapa paragraf yang cuma terdiri dari 1 kalimat. Sehingga kalimat tersebut mempunyai nilai yang tinggi pada fitur posisi kalimat. Kerja sistem akan optimal pada fitur posisi kalimat apabila dalam satu paragraf terdiri lebih dari 5 kalimat.

Hal lain yang juga berpengaruh pada hasil uji coba sistem yaitu pemilihan kandidat kalimat ringkasan. Sistem ini akan mengenali kalimat yang terdiri dari kata penting dari dokumen, sehingga kerja sistem akan optimal jika kalimat tersebut mengandung deskripsi panjang mengenai sebuah dokumen yang terdiri dari banyak kata penting. Dalam hal ini sistem tidak bias mengekstrak kalimat kandidat ringkasan yang berupa kalimat deskripsi singkat.

BAB V PENUTUP

5.1 Kesimpulan

Kesimpulan yang dihasilkan dalam skripsi ini adalah:

1. Telah dibuat sistem peringkasan dokumen teks otomatis dengan menggunakan penggabungan dua metode yaitu pembobotan kalimat dan *fuzzy inference system* model mamdani pada dokumen teks berita berbahasa indonesia dan telah dilakukan uji coba sistem dengan menghitung nilai *precision* dan *recall* untuk besar ringkasan sebesar 25% dan 50%.
2. Peringkasan dokumen teks otomatis dengan menggabungkan dua metode yaitu pembobotan kalimat dan *fuzzy inference system* model mamdani menghasilkan rata – rata *precision* sebesar 0.453001 dan *recall* sebesar 0.468134 untuk besar ringkasan sebesar 25%. Dan nilai rata – rata *precision* sebesar 0.589149 dan *recall* sebesar 0.601721 untuk besar ringkasan sebesar 50%. Sedangkan *AutoSummarize* yang menghasilkan nilai rata – rata dari *precision* sebesar 0.349274 dan *recall* 0.354928 untuk besar ringkasan sebesar 25% dan untuk hasil ringkasan sebesar 50% menghasilkan nilai rata – rata *precision* sebesar 0.512515 dan *recall* 0.521096.
3. Setelah dilakukan perbandingan dengan sistem peringkasan lain yaitu *AutoSummarize (Microsoft Word 2007)* dihasilkan kesimpulan bahwa sistem peringkasan dokumen ini lebih baik.

5.2 Saran

Beberapa saran mengenai pengembangan lebih lanjut yang dapat diberikan penulis antara lain:

1. Penanganan metode *stemming* yang digunakan mampu mengenali sisipan dan dapat membedakan apakah kalimat tersebut merupakan kata dasar atau bukan.
2. Dokumen yang akan diringkaskan tidak sebatas hanya dokumen yang memiliki judul dan isi saja, namun dapat diperluas pada dokumen yang mempunyai bab maupun subbab.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- Al-Hashemi, Rafeq. 2010. *Text Summarization Extraction System (TSES) using Extracted Keyword*. International Arab Journl of e-Technology, Vol. 1, No. 4, June 2010.
- Azrifah, Masrah dan Martin, Trevor. 2009. *Similarity-Based Estimation for Document Summarization using Fuzzy Sets*. International Journal of Computer Science and Security, Volume (1) : Issue (4)
- Das, Dipanjan dan Martins, Andre F.T. 2007. *A Survey on Automatic Summarization*. Language Technologies Institute. Carnegie Mellon University.
- Fatah, Mohamed Abdel. Dan Ren, Fuji. 2008. *Automatic Text Summarization*. World Academy of Science, Engineering and Technology 37 2008.
- Gupta, Vishal dan Singh Lehal, Gurpreet. 2010. *A Survey of Text Summarization Extractive Techniques* Journal of Emerging Technologies in Web Intelegence, Vol. 2, No. 3, August 2010
- Hovy, Eduard dan Lin, Chin-Yew. *Automated Text Summarizaion and the Summarist System*. Information Science Institute of the University of Southern California.
- Hovy, E. 2003. *Text Summarization*. Dalam R. Mitkov, *The Oxford Handbook of Computational Linguistic*. Oxford: Oxford University Press
- Isfahani, Fariba Rahimi. Kyoomarsi, Farshad, Khosravi, Hamid. Eslami, Esfandiar. Tajodin, Asgar dan Dehkordy, Pooya Khosravyan. 2008. *Aplication of Fuzzy logic in the Improvement of Text Summarization*. IADIS International Conference Informatics

- Jagadeesh, j. Pingali, Prasad. Varma, Vasudeva. 2005. *Sentence Extraction Based Single Document Summarization*. Workshop on Document Summarization, 19th and 20th March, 2005, IIIT Allahabad.
- Jazek, Karel dan Stainberger, Josef. 2008. *Automatic Text Summarization (The state of the art 2007 and new challenges)*. Václav Sná¹el (Ed.): Znalosti 2008, pp. 1{12, ISBN 978-80-227-2827-0.
- Kusumadewi, Sri. 2003. *Artificial Intelegence (Teknik dan Aplikasinya)*. Graha Ilmu. Yogyakarta
- Kusumadewi, Sri dan Purnomo, Hari . 2004. *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan*. Graha Ilmu. Yogyakarta
- Kyoomarsi, Farshad. Khosravi, Hamid. Eslami, Esfandiar dan Dehkordy, Pooya Khosravyan. 2009. *Optimizing Machine Learning Approach Based on Fuzzy Logic in Text Summarization* International Journal of Hybrid Information Technology Vol.2, No.2, April, 2009
- Suanmali, Ladda. Salim, Naomie dan Binwahlan, Mohammed Salem *Fuzzy Logic Based Method for Improving Text Summarization*. 2009. (IJCSIS) International Journal of Computer Science and Information Security, Vol. 2, No. 1, 2009
- Saelan, Athia. 2009. *Logika Fuzzy*. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika. Institut teknologi Bandung.
- Tala, Fadillah Z. 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Master of Logic Project. Institute for Logic, Language and Computation. Universiteit van Amsterdam. The Netherlands.

Lampiran 1 Informasi Data Pengujian

Tabel Info dokumen

No	id_berita	Judul	Jumlah Paragraf	Jumlah kalimat
1	BR001	100 Rumah Lebih Hangus di Medan	5	13
2	BR002	Target Divestasi BCA Mungkin Tak Tercapai	19	34
3	BR003	Warga Perlu Air Minum, Makanan Matang, dan Obat	11	22
4	BR004	Ledakan Mesi di Makasar Lukai Seorang Perempuan	6	18
5	BR005	Ratusan Pengungsi Banjir Nias Perlu Bantuan	8	21
6	BR006	Perawat Layani Aborsi, Ditangkap Polisi	6	16
7	BR007	Ujung Menteng Berangsur Normal	7	16
8	BR008	Xanana Harapkan Megawati Hadir	9	25
9	BR009	Masyarakat Ambon Tunjukkan Sikap Mampu Menahan Diri	17	38
10	BR010	Jangan Lagi Merekap Bank Bermasalah	5	18
11	BR011	Menlu RI Siap Jadi Fasilitator Dialog Korea	8	21
12	BR012	Momentum bagi Pemulihan Ekonomi Sudah Hilang	16	43
13	BR013	Pajak Kendaraan Dipaksakan Naik untuk Kejar APBD	11	26
14	BR014	PKB Tahun 2001 berat bagi Presiden	7	17
15	BR015	Presiden Setuju Pembentukan Provinsi Kepri	9	20

Tabel Ringkasan system

No	id_berita	Ringkasan (No. Kalimat)	Jumlah Kalimat	Besar Ringkasan
1	BR001	1, 2, 6, 9, 10, 11, 12	7	50%
2	BR001	1, 2, 6, 11	4	25%
3	BR002	1, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 15, 16, 20, 21, 24, 26, 28	18	50%
4	BR002	1, 3, 4, 8, 11, 20, 23, 25, 31	9	25%
5	BR003	1, 3, 6, 7, 12, 14, 16, 18, 19, 21, 22	11	50%
6	BR003	1, 6, 7, 12, 19, 21	6	25%
7	BR004	1, 2, 3, 4, 5, 12, 13, 16, 18	9	50%
8	BR004	1, 3, 4, 13, 18	5	25%
9	BR005	1, 3, 4, 6, 7, 9, 10, 12, 14, 15, 16	11	50%
10	BR005	3, 4, 7, 12, 14, 15	6	25%
11	BR006	1, 2, 3, 4, 7, 12, 14, 15	8	50%
12	BR006	1, 3, 7, 12	4	25%
13	BR007	1, 2, 4, 5, 7, 11, 13, 14	8	50%
14	BR007	1, 2, 5, 11	4	25%
15	BR008	1, 2, 4, 5, 6, 7, 8, 15, 16, 18, 20, 21, 23	13	50%
16	BR008	1, 4, 5, 6, 7, 8, 16	7	25%
17	BR009	1, 3, 5, 6, 8, 9, 10, 11, 12, 20, 22, 25, 26, 27, 29, 30, 33, 34, 36	19	50%
18	BR009	1, 3, 8, 9, 10, 11, 22, 27, 30, 34	10	25%
19	BR010	1, 2, 3, 4, 6, 9, 10, 11, 14	9	50%
20	BR010	1, 3, 4, 10, 11	5	25%
21	BR011	1, 4, 8, 9, 10, 11, 16, 17, 18, 19, 21	11	50%
22	BR011	1, 4, 8, 11, 16, 17	6	25%
23	BR012	1, 3, 4, 6, 8, 9, 10, 11, 12, 16, 18, 19, 20, 21, 22, 23, 30, 34, 36, 3, 38, 43	22	50%
24	BR012	3, 4, 8, 9, 10, 18, 19, 20, 22, 30, 37	11	25%

25	BR013	2, 5, 6, 9, 11, 12, 14, 17, 18, 19, 20, 21, 23	13	50%
26	BR013	5, 6, 9, 11, 14, 17, 18	7	25%
27	BR014	1, 2, 4, 5, 7, 9, 10, 13, 14	9	50%
28	BR014	1, 2, 4, 5, 10	5	25%
29	BR015	1, 4, 6, 8, 9, 11, 12, 13, 14, 16	10	50%
30	BR015	4, 9, 11, 13, 16	5	25%

Tabel Ringkasan *AutoSummarizer (Microsoft Word 2007)*

No	id_berita	Ringkasan (No. Kalimat)	Jumlah Kalimat	Besar Ringkasan
1	BR001	1, 4, 6, 7, 8, 10, 12	7	50%
2	BR001	4, 7, 10, 12	4	25%
3	BR002	2, 4, 5, 7, 9, 11, 12, 13, 14, 15, 20, 21, 25, 26, 30, 32, 35	18	50%
4	BR002	2, 5, 7, 9, 11, 12, 21, 32	9	25%
5	BR003	1, 2, 7, 10, 11, 13, 14, 18, 20, 21, 22	11	50%
6	BR003	10, 11, 13, 14, 18, 20	6	25%
7	BR004	3, 4, 6, 7, 9, 10, 12, 15, 17	9	50%
8	BR004	4, 6, 9, 12, 15	5	25%
9	BR005	1, 2, 9, 10, 11, 12, 16, 17, 18, 19, 21	11	50%
10	BR005	1, 2, 9, 10, 17, 19, 21	7	25%
11	BR006	1, 4, 5, 6, 7, 8, 14, 15, 16	9	50%
12	BR006	5, 7, 9, 14, 15, 16	5	25%
13	BR007	3, 4, 5, 6, 7, 10, 12, 14	8	50%
14	BR007	5, 6, 10, 14	4	25%
15	BR008	1, 2, 7, 9, 10, 13, 14, 16, 18, 21, 22, 23, 24, 25	14	50%
16	BR008	1, 2, 10, 18, 23, 24	6	25%
17	BR009	1, 3, 6, 8, 9, 11, 12, 14, 16, 19, 24, 26, 28, 30, 31, 32, 34, 35, 36, 38	20	50%
18	BR009	1, 6, 8, 9, 19, 30, 31, 34, 36, 38	10	25%
19	BR010	1, 2, 7, 8, 9, 10, 13, 15, 18	9	50%

20	BR010	1, 7, 9, 10	4	25%
21	BR011	1, 2, 6, 7, 8, 9, 15, 17, 18, 19, 21	11	50%
22	BR011	1, 2, 17, 18, 19	6	25%
23	BR012	1, 2, 4, 9, 12, 13, 15, 16, 18, 21, 22, 23, 25, 27, 28, 30, 31, 32, 33, 34, 39, 40	22	50%
24	BR012	1, 4, 12, 13, 15, 21, 25, 27, 28, 30, 31, 32, 39	13	25%
25	BR013	1, 8, 10, 12, 14, 16, 17, 19, 22, 23, 25, 26	12	50%
26	BR013	1, 14, 16, 17, 23, 26	6	25%
27	BR014	1, 4, 5, 6, 9, 10, 11, 14	9	50%
28	BR014	1, 10, 11	5	25%
29	BR015	1, 3, 6, 7, 9, 10, 11, 14, 15, 16, 17	10	50%
30	BR015	3, 6, 9, 11, 16	5	25%

Tabel Ringkasan Guru Bahasa Indonesia SMK

No	id_berita	Ringkasan (No. Kalimat)	Jumlah Kalimat	Besar Ringkasan
1	BR001	1, 3, 5, 6, 7, 10, 12	7	50%
2	BR001	1, 5, 7, 12	4	25%
3	BR002	1, 2, 3, 5, 10, 11, 12, 13, 16, 17, 18, 19, 21, 24, 26, 28, 35	18	50%
4	BR002	1, 2, 6, 10, 11, 12, 13, 21, 26, 28	9	25%
5	BR003	1, 3, 4, 8, 9, 10, 12, 18, 19, 20, 21, 22	11	50%
6	BR003	1, 8, 9, 10, 18, 22	6	25%
7	BR004	1, 3, 5, 9, 12, 13, 15, 16, 18	9	50%
8	BR004	1, 5, 12, 13, 15	5	25%
9	BR005	2, 3, 4, 6, 7, 10, 11, 12, 14, 16, 17	11	50%
10	BR005	2, 3, 4, 6, 10, 12, 14	7	25%
11	BR006	1, 2, 4, 7, 8, 11, 15, 16	8	50%
12	BR006	1, 2, 11, 15	4	25%
13	BR007	1, 5, 6, 10, 11, 12, 13, 14	8	50%

14	BR007	1, 5, 10, 11	4	25%
15	BR008	1, 2, 4, 6, 8, 11, 13, 16, 17, 18, 22	11	50%
16	BR008	1, 8, 11, 16, 18	5	25%
17	BR009	3, 4, 6, 8, 9, 12, 14, 16, 20, 23, 27, 28, 30, 32, 33, 34, 35, 37, 38	19	50%
18	BR009	4, 6, 9, 12, 14, 20, 23, 27, 30, 34	10	25%
19	BR010	1, 2, 5, 7, 9, 11, 14, 17, 18	9	50%
20	BR010	1, 7, 9, 11, 17	5	25%
21	BR011	1, 3, 5, 8, 10, 15, 16, 17, 18, 19	10	50%
22	BR011	1, 5, 8, 10, 16	5	25%
23	BR012	1, 2, 3, 7, 8, 9, 11, 13, 14, 15, 16, 18, 20, 21, 23, 24, 26, 27, 20, 35, 38, 43	22	50%
24	BR012	1, 3, 7, 8, 14, 15, 16, 18, 20, 23, 30	11	25%
25	BR013	1, 2, 6, 9, 11, 12, 17, 20, 21, 22, 25, 26	12	50%
26	BR013	1, 2, 6, 9, 12, 17, 21	7	25%
27	BR014	1, 5, 8, 9, 10, 12, 15, 16, 17	9	50%
28	BR014	1, 8, 9, 12, 16	5	25%
29	BR015	1, 4, 5, 7, 8, 10, 11, 13, 17, 19	10	50%
30	BR015	1, 5, 7, 8, 11	5	25%

UNIVERSITAS BRAWIJAYA



Lampiran 2 Daftar *StopWord*

Ada	artinya	belakangan
adalah	asal	belum
adanya	asalkan	belumlah
adapun	atau	benar
aduh	ataukah	benarkah
agak	ataupun	benarlah
agaknya	awal	berada
agar	awalnya	berakhir
Aja	Ayo	berakhirilah
akan	ayolah	berakhirnya
akankah	bagai	berapa
akhir	bagaikan	berapakah
akhiri	bagaimana	berapalah
akhirnya	bagaimanakah	berapapun
Aku	bagaimanapun	berarti
akulah	bagi	berawal
akupun	bagian	berbagai
amat	bahkan	berdatangan
amatlah	bahwa	berikan
anda	bahwasanya	berikut
andalah	baik	berikutnya
andaikan	bakal	berjumlah
andaikata	bakalan	berkali
antar	banyak	berkata
antara	bapak	berkehendak
antaranya	beberapa	berkeinginan
Apa	begini	berkenan
apaan	beginian	berlainan
apabila	beginikah	berlalu
apakah	begitu	berlangsung
apalagi	begitukah	berlebihan
apapun	begitulah	bermacam
Apatah	begitupun	bermaksud

Bermula	dahulu	dijawab
Bersama	dalam	dijelaskan
Bersiap	dan	dijelaskannya
Bertanya	dapat	dikarenakan
Berturut	dari	dikatakan
Bertutur	daripada	dikatakannya
Berujar	dekat	dikerjakan
Berulang	demi	diketahui
Berupa	demikian	diketuahuinya
Berupaya	demikianlah	dikira
Betul	dengan	dilakukan
Betulkah	depan	dilalui
Biasa	di	dilihat
Biasanya	dia	dimaksud
Bila	diakhiri	dimaksudkan
Bilakah	diakhirinya	dimaksudkannya
Bias	diakui	dimaksudnya
Bisakah	diakuinya	diminta
Boleh	diakah	dimintai
Bolehkah	dialah	dimisalkan
Bolehlah	diantara	dimulai
Buat	diantaranya	dimulailah
Buatan	diberi	dimungkinkan
Bukan	diberikan	dini
Bukankah	diberikannya	dipastikan
Bukanlah	dibuat	diperbuat
Bukannya	dibuatnya	diperbuatnya
Bulan	didapat	dipergunakan
Bung	didatangkan	diperkirakan
Cara	digunakan	diperlihatkan
Caranya	diibaratkan	diperlukan
Cukup	diibaratkannya	diperlukannya
Cukupkah	diingat	dipersoalkan
Cukuplah	diingatkan	dipertanyakan
Cuma	diinginkan	dipunyai

diri	hanya	jelas
dirinya	hanyalah	jelaskan
disampaikan	hari	jelasmah
disebut	harus	jelasnya
disebutkan	haruslah	jika
disebutkannya	harusnya	jikalau
disinilah	hendak	juga
ditambahkan	hendaklah	jumlah
ditandaskan	hendaknya	jumlahnya
ditanya	hingga	justru
ditanyai	la	kala
ditanyakan	ialah	kalau
ditegaskan	ibarat	kalaulah
ditujukan	ibaratkan	kalaupun
ditunjuk	ibaratnya	kali
ditunjuki	Ibu	kalian
ditunjukkan	ingin	kami
ditunjukkannya	inginkah	kamilah
ditunjuknya	inginkan	kamu
dituturkan	Ini	kamulah
dituturkannya	inikah	kan
diucapkan	inilah	kapan
diucapkannya	Itu	kapankah
diungkapkan	itukah	kapanpun
dong	itulah	karena
dua	jadi	karenanya
dulu	jadikan	katakan
enggak	jadilah	katakanlah
enggaknya	jadinya	katanya
entah	jangan	ke
entahlah	jangankan	keadaan
guna	janganlah	kebetulan
gunakan	jawab	kecil
Hal	jawaban	keinginan
hampir	jawabnya	kelamaan

kelihatan	macam	memperkirakan
kelihatannya	maka	memperlihatkan
kembali	makanya	mempersiapkan
kemudian	makin	mempersoalkan
kemungkinan	malah	mempertanyakan
kemungkinannya	malahan	mempunyai
kenapa	mampu	memulai
kenyataanya	mampukah	memungkinkan
kepada	mana	menambahkan
kepadanya	manakala	menandaskan
kesampaian	manalagi	menanti
keseluruhan	masa	menantikan
keseluruhannya	masalah	menanya
keterlalu	masalahnya	menanyai
ketika	masih	menanyakan
khususnya	masihkah	mendapat
kini	masing	mendapatkan
kinilah	mau	mendatang
kira	maupun	mendatangi
kiranya	melainkan	mendatangkan
kita	melakukan	menegaskan
kitalah	melalui	mengakhiri
kok	melihatnya	mengapa
lagi	memang	mengatakan
lagian	memastikan	mengatakannya
lah	memberi	mengenai
lain	memberikan	mengerjakan
lainnya	membuat	mengetahui
lalu	memerlukan	menggunakan
lama	memihak	menghendaki
lamanya	meminta	mengibaratkan
lanjut	memintakan	mengibaratkannya
lanjutnya	memisalkan	mengingat
lebih	memperbuat	mengingatkan
lewat	mempergunakan	menginginkan

mengira	mulai	pihaknya
mengucapkan	mulailah	pukul
mengucapkannya	mulanya	pula
mengungkapkan	mungkin	pun
menjadi	mungkinkah	punya
menjawab	nah	rasa
menjelaskan	namun	rasanya
menuju	nanti	rata
menunjuk	nantinya	rupanya
menunjuki	nyaris	saat
menunjukkan	nyatanya	saatnya
menunjuknya	oleh	saja
menurut	olehnya	sajalah
menuturkan	pada	saling
menyampaikan	padahal	sama
menyangkut	padanya	sambil
menyatakan	pak	sampai
menyebutkan	paling	sampaikan
menyeluruh	pantas	sangat
menyiapkan	para	sangatlah
merasa	pasti	satu
mereka	pastilah	saya
merekalah	penting	sayalah
merupakan	pentingnya	se
meski	per	sebab
meskipun	percuma	sebabnya
meyakini	perlu	sabagai
meyakinkan	perlukah	sebagaimana
minta	perlunya	sebagainya
mirip	pernah	sebaik
misal	persoalan	sebaiknya
misalkan	pertama	sebaliknya
misalnya	pertanyaan	sebanyak
mudah	pertanyakan	sebegini
mula	pihak	sebegitu

sebelum	sekarang	seorang
sebelumnya	sekaranglah	sepanjang
sebenarnya	seketika	sepantasnya
seberapa	sekiranya	sepantasnyalah
sebesar	sekitar	seperlunya
sebetulnya	sekitarnya	seperti
sebisanya	sekurangnya	sepertinya
sebuah	sela	sepihak
sebut	selain	sering
sebutlah	selaku	seringnya
sebutnya	selalu	serta
secara	selama	serupa
secukupnya	selamanya	sesaat
sedang	selanjutnya	sesama
sedangkan	seluruh	sesampai
sedemikian	seluruhnya	sesegerakan
sedikit	semacam	sesekali
sedikitnya	semakin	seseorang
seenaknya	semampu	sesuatu
segala	semampunya	sesuatunya
segalanya	semasa	sesudah
segera	semasih	sesudahnya
seharusnya	semata	setelah
sehingga	semaunya	setempat
seingat	sementara	setengah
sejak	semisal	seterusnya
sejauh	semisalnya	setiap
sejenak	sempat	setiba
sejumlah	Semua	setibanya
sekadar	Semuanya	setidaknya
sekadarnya	Semula	setinggi
sekali	Sendiri	seusai
sekalian	Sendirian	sewaktu
sekaligus	Sendirinya	siap
sekalipun	Seolah	siapa

siapakah	tentunya	tinggi
siapapun	tepat	toh
Sini	terakhir	tunjuk
sinilah	terasa	turut
soal	terbanyak	tutur
soalnya	terdahulu	tuturnya
suatu	terdapat	ucap
sudah	terdiri	ucapnya
sudahkah	terhadap	ujar
sudahlah	terhadapnya	ujarnya
supaya	teringat	umum
tadi	terjadi	umumnya
tadinya	terjadilah	ungkap
tahu	terjadinya	ungkapnya
tahun	terkira	untuk
Tak	terlalu	usah
tambah	terlebih	usai
tambahnya	terlihat	waduh
tampak	termasuk	wah
tampaknya	ternyata	wahai
tandas	tersampaikan	waktu
tandasnya	tersebut	waktunya
tanpa	tersebutlah	walaupun
tanya	tertentu	walaupun
tanyakan	tertuju	wong
tanyanya	terus	yaitu
tapi	terutama	yakin
tegas	tetap	yakni
tegasnya	tetapi	yang
telah	tiap	kecuali
tempat	tiba	beginilah
tengah	tidak	bekerja
tentang	tidakkah	kurang
tentu	tidaklah	melihat
tentulah	tiga	kelima

Lima
Luar
A
B
C
D
E
F
G
H
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
dll
dkk

