Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 1, No. 4, April 2017, hlm. 307-319 http://j-ptiik.ub.ac.id

Peringkasan Literatur Ilmu Komputer Bahasa Indonesia Berbasis Fitur Statistik dan Linguistik menggunakan Metode *Gaussian Naïve Bayes*

Muhammad Fhadli[1], Mochammad Ali Fauzi[2], Tri Afirianto[3]

Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]muhammadfhadli20@gmail.com, [2]moch.ali.fauzi@ub.ac.id, [3]tri.afirianto@ub.ac.id

Abstrak

Di tengah era dengan kebutuhan data yang besar ini, peringkasan teks merupakan suatu kebutuhan. Dengan peringkasan teks, setiap orang bisa mendapatkan informasi yang mendeskripsikan keseluruhan data teks yang besar hanya dengan beberapa kalimat. Permasalahan dalam peringkasan teks adalah kualitas hasil ringkasan. Salah satu metode untuk meringkas teks yang dikenal adalah metode TF-IDF, metode ini merupakan metode peringkasan dengan pendekatan statistik. Pendekatan lain untuk meringkas teks adalah pendekatan linguistik. Pada umunya, ringkasan dari sebuah teks terdiri atas kalimat-kalimat yang memiliki fitur-fitur linguistik seperti jumlah kata, jumlah kata kunci, dan posisi kalimat pada teks asli. Fitur-fitur tersebut dapat digunakan untuk mengklasifikasikan suatu kalimat baru kedalam kelas ringkasan atau kelas bukan ringkasan. Hasil peringkasan diperoleh dari kumpulan kalimat pada kelas ringkasan. Pada penelitian ini, penulis melakukan penggabungan fitur statistik dan fitur linguistik untuk melakukan peringkasan teks. Hasil pengujian penelitian ini menunjukkan peringkasan dengan fitur statistik dan linguistik menggunakan metode Naïve Bayes memiliki nilai rata-rata *f- score* 0,206538 dan nilai rata-rata *relative utility* 0,116657.

Kata Kunci: *peringkasan statistik, peringkasan linguistik, Naïve Bayes*

Abstract

In this era which require big amount of data, text summarization becomes a needs. With text summarization, everyone can get information that describe all of big text in just few of sentences. The problem in text summarization is quality of the summarization result. One of the known method for text summarization is TF- IDF, this method is a method for summarizing text using statistical approach. The other approach for summarizing text is statistical approach. In a general way, summarization result is consist of sentences with statistical features such as total of words, total of keywords, and sentence position in the original text. Those features can be used to classify a text into class of summary or class of non summary. The summarization result come from the composite of every sentence in summary class. In this research, writer combines the use of statistical feature and linguistical features to summarize text. The testing result of this research show that summarization with statistical and linguistical features using Naïve Bayes method came with f-score average 0.206538 and realive utility average 0.116657.

Keywords: *statistical summarization, linguistical summarization, Naïve Bayes*

1. PENDAHULUAN

Perkembangan informasi tentang dunia sains dan dunia teknologi yang berkembang jauh sangat pesat dalam dua dekade ini membuat ketergantugan manusia terhadap data dari berbagai macam bidang sangat tinggi, sehingga di masa depan manusia diprediksi akan menyimpan data dalam jumlah yang sangat besar (Emre, 2016). Hadirnya internet di tengah kehidupan manusia juga membuat volume data dalam berbagai format bertambah drastis, termasuk juga volume data yang berupa teks (Oliveira et al., 2016).

Keberadaan data yang sangat banyak di internet membuat pencarian data atau informasi secaca cepat dan efisien di internet menjadi lebih sulit (Babar, 2015). Keberadaan data tersebut sulit didapatkan secara cepat dan efisien karena data yangtersedia tidak terbatas sehingga akan sulit untuk mendeteksi keberadaan

dokumen atau informasi tertentu (Tayal, 2017). Keberadaan dokumen yang banyak tersebut membuat dilema karena manusia harus mencari sedikit informasi penting dari dokumen yang sangat banyak, padahal setiap pengguna internet hanya membutuhkan informasi utama dari dokumen tersebut (Emre, 2016).

Agar pengguna internet bisa mendapatkan informasi yang ringkas namun tanpa menghilangkan informasi yang penting, maka dibutuhkan peringkasan informasi atau peringkasan dokumen (Abbasi-ghalehtaki, 2016). Yang menjadi permasalah utama dalam peringkasan teks adalah kualitas hasil peringkasan. Kesesuaian teks yang akan diringkas dengan hasil peringkasan tentunya harus menghasilkan nilai yang baik. Beberapa penelitian tentang peringkasan teks sudah pernah dilakukan sebelumnya dengan menggunakan berbagai macam metode. Salah satu penelitian sebelumnya menerapkan *Conditional Random Field* pada peringkasan otomatis untuk mendeteksi fitur-fitur dari sebuah teks dengan *Non-negative Matrix Factorization* (Batcha, 2013). Penelitian yang dilakukan oleh Batcha ini berfokus pada pemilihan fitur yang tepat agar menghasilkan ringkasan yang optimal. Namun, metode ini tidak menambahkan fitur statistik pada penghitungannya padahal fitur statistik sangat diperlukan untuk menentukan kalimat yang mengandung informasi utama terbanyak.

Penelitian yang lain juga pernah dilakukan untuk memilih kalimat dengan relevansi yang tinggi agar menghasilkan peringkasan yang baik menggunakan *fuzzy inference system* untuk mengekstrasi fitur (Babar, 2015). Penelitian yang dilakukan Babar dan Patil ini menggabungkan konsep logika *fuzzy* dan algoritma genetika dalam masalah peringkasan. Namun metode ini melakukan peringkasan hanya berdasarkan 4 fitur yaitu kata, posisi, panjang, dan kesamaan. Tentunya ada fitur penting yang terlewatkan dalam penelitian ini.

Dari berbagai macam penelitian yang telah dilakukan dengan berbagai macam objek penelitian yang digunakan, penulis mengembangkan metode dengan pendekatan *machine learning* menggunakan klasifikasi Gaussian Naïve Bayes untuk mengelompokkan suatu kalimat tergolong hasil peringkasan atau tidak. Objek penelitian yang digunakan untuk percobaan peringkasan ini adalah literatur berbahasa Indonesia di bidang ilmu komputer dengan ekstensi *Portable Document Format* (PDF). Penulis menggunakan literatur berbahasa Indonesia dan merupakan literatur di bidang ilmu komputer dalam penelitian ini dengan pertimbangan mudah dalam mencari pakar yang akan menilai hasil penelitian. Hasil dari metode yang diterapkan dalam penelitian ini adalah ringkasan yang diperoleh dari menelurusi keseluruhan isi literatur.

Metode Gaussian Naïve Bayes digunakan pada penelitian ini sehingga memerlukan data *training* berupa bobot untuk fitur statistik dan fitur linguistik dari setiap kalimat pada 5 literatur ilmu komputer. Fitur yang digunakan pada penelitian ini adalah bobot hasil penghitungan *Term Frequency- Inverse Document Frecuency* (TF-IDF) sebagai fitur statistik dan ditambah dengan 10 fitur linguistik.

Setiap kalimat dalam suatu literatur memiliki 2 kemungkinan yaitu kalimat itu merupakan kalimat yang bisa digolongkan sebagai ringkasan atau kalimat yang tidak bisa digolongkan sebagai hasil ringkasan. Kalimat yang bisa digolongkan sebagai hasil ringkasan tentunya memilik nilai TF-IDF yang tinggi, umumnya terletak di awal paragraf pertama, berisi kata kata yang sesuai judul, dan lain sebagainya (Gupta, 2010). Oleh karena itu, penggunaan fitur statistik dan fitur linguistik perlu dikombinasikan dalam melakukan peringkasan agar tercapai hasil peringkasan dengan akurasi yang optimal. Berdasarkan uraian tersebut dapat disimpulkan bahwa peringkasan literatur seperti di atas membutuhkan suatu metode klasifikasi.

Pada penelitian ini, Gaussian Naïve Bayes dipilih sebagai metode untuk mengklasifikasikan kalimat karena metode ini sederhana dalam penerapannya namun kualitas hasil klasifikasinya tidak kalah dengan metode-metode lain yang lebih rumit selain itu metode ini bisa menghasilkan solusi yang optimal bahkan ketika solusi tidak bisa ditemukan menggunakan asumsi biasa (Griffis, 2016).

Belum ada penelitian yang menggabungkan fitur statistik dan fitur linguistik untuk melakukan peringkasan teks. Oleh karena itu pada penelitan ini, penulis memberikan judul Peringkasan Literatur Ilmu Komputer Bahasa Indonesia Berbasis Fitur Statistik dan Linguistik Menggunakan Metode Naïve Bayes.

2. DASAR TEORI

2.1 Peringkasan Teks

Peringkasan teks didefinisikan sebagai

proses pemadatan teks menjadi versi yang lebih pendek namun tetap mengandung informasi yang bisa mencerminkan keseluruhan teks (Gupta, 2010). Sebelum masuk ke proses peringkasan, terhadap data yang akan diringkas perlu dilakukan *preprocessing* terlebih dahulu. *Preprocessing* data pada umumnya memakan waktu lebih lama dibadingkan dengan proses utama dari pencarian informasi terhadap suatu data (Munková, 2013). Hasil dari *preprocessing* adalah kalimat yang telah dibagi menjadi kata-kata. Pada umumnya terdapat empat tahap untuk melakukan *preprocessing* terhadap suatu teks seperti yang ditunjukkan pada poin 1 sampai 4 (Uysal, 2014).

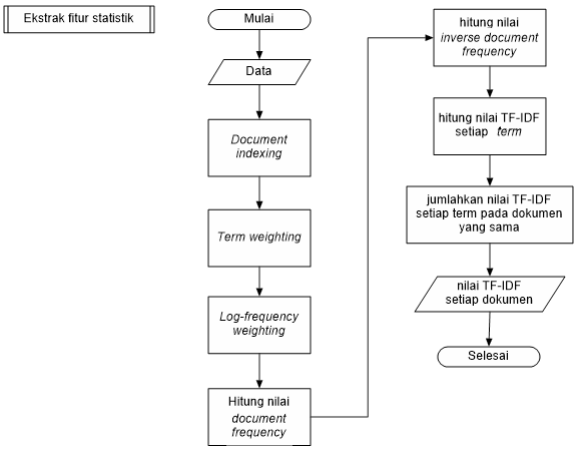
1. Parsing, yaitu pengambilan data yang akan diproses kemudian data tersebut dipisah menjadi kalimat-kalimat di mana setiap kalimat merepresentasikan satu dokumen.
2. *Lexing* atau *tokenization*, yaitu pemisahan setiap kalimat menjadi kata-kata. Pada tahap ini dilakukan juga penghapusan angka, simbol ilmiah, tanda baca, karakter selain huruf alfabet, penghapusan duplikasi kata dan mengubah huruf kapital menjadi huruf kecil.
3. *Filtering*, yaitu pemilihan kata yang akan digunakan sebagai *term. Filtering* bisa dilakukan dengan dua pendekatan yaitu pendekatan *wordlist* dan pendekatan *stopword*. Pendekatan *wordlist* berupa pembiaran kata yang penting pada dokumen dan menghapus kata-kata selainnya untuk dijadikan *term.* Pendekatan *stopword* berupa penghapusan kata-kata yang tidak penting dan membiarkan kata-kata selainnya untuk dijadikan *term* di mana daftar kata yang tidak penting tersebut dikenal dengan istilah *stoplist.*
4. *Stemming*, yaitu mengubah kata berimbuhan menjadi kata dasar dengan cara menghilangkan imbuhan.

2.2 Ekstraksi Fitur

Pada peringkasan ekstraktif terdapat dua pendekatan untuk menentukan suatu kalimat tergolong sebagai kalimat yang penting atau tidak. Pendekatan tersebut adalah pendekatan menggunakan fitur statistik dan pendekatan menggunakan fitur linguistik (Gupta, 2010). Contoh fitur statistik yang digunakan dalam peringkasan adalah fitur TF-IDF yang hanya mengukur tingkat kepentingan kalimat berdasarkan statistic jumlah kata yang penting. Sedangkan contoh fitur linguistik yang digunakan dalam peringkasan adalah fitur lokasi kalimat dan contoh kalimat.

2.2.1 Ekstraksi Fitur Statistik

Fitur statistik yang diekstrak dari sebuah teks adalah skor TF-IDF. TF-IDF didefinisikan sebagai statistik numerik yang bertujuan untuk menunjukkan seberapa penting suatu kata di dalam dokumen, di mana bobot dari kata akan meningkat sebanding dengan jumlah kata tersebut di dalam dokumen (Shouzhong, 2016). Gambar 2.1 menunjukkan langkah-langkah yang perlu dilakukan untuk mendapatkan skor TF-IDF.



Gambar 2.1 Ekstraksi Fitur Statistik

2.2.2 Ekstraksi Fitur Linguistik

Terdapat 7 fitur linguistik yang akan digunakan pada penelitian ini. 7 fitur lingusitik tersebut dijelaskan pada poin nomor 1 sampai 7.

1. Fitur *title word*

Setelah *stopword* dihilangkan dari judul dokumen, maka semakin banyak *term* pada dokumen yang juga muncul di judul menunjukkan bahwa kalimat tersebut semakin penting. Nilai dari fitur ini bisa didapat dengan menghitung jumlah *term* pada judul dokumen yang muncul pada kalimat.

2. Fitur s*entence location*

Kalimat pada awal dan akhir keseluruhan dokumen pada umumnya tergolong kalimat yang penting dan berpeluang besar untuk dijadikan sebagai ringkasan. Nilai dari fitur ini merupakan posisi kalimat pada dokumen. Jika kalimat x berada di awal dokumen, maka nilai fitur *sentence location* dari kalimat x adalah 1. Sedangkan jika kalimat x berada di posisi terakhir dari keseluruhan dokumen, maka nilai fitur *sentence location*

dari kalimat x adalah nilai jumlah dokumen pada data tersebut.

3. Fitur *sentence length*

Kalimat yang sangat pendek dan kalimat yang sangat panjang pada umumnya tidak dijadikan sebagai ringkasan. Nilai fitur *sentence length* didapatkan dari jumlah kata pada kalimat sebelum *preprocessing*.

4. Fitur *upper-case word*

Kalimat yang mengandung akronim dan/atau huruf kapital yang banyak, berpeluang besar dimasukkan sebagai ringkasan.

5. Fitur *cue-phrase*

Cue-phrase adalah frasa yang menunjukkan suatu kalimat tergolong penting. Contoh *cue-phrase* adalah : "kesimpulannya", "oleh karena itu", dan "jadi". Kalimat yang mengandung *cue-phrase* pada umumnya dimasukkan sebagai ringkasan. Tidak ada standar khusus untuk penggunaan *cue-phrase* sehingga pada suatu penelitian, *cue-phrase* yang digunakan perlu didiskuiskan terlebih dahulu dengan pakar.

6. Fitur *biased word*

Biased word merupakan daftar kata yang mengandung domain spesifik dari dokumen, seperti kata kunci. Semakin banyak *biased word* yang terdapat pada suatu kalimat, maka kalimat tersebut semakin penting.

7. Fitur *occurrence of non-essential information*

Beberapa kata tertentu menunjukkan bahwa suatu kalimat tergolong kalimat yang tidak penting. Jika suatu kalimat mengandung frasa-frasa berikut ini, maka nilai dari fitur ini pada kalimat tersebut adalah *false*. Contoh frasa-frasa yang menunjukkan bahwa suatu kalimat tidak tergolong kalimat yang penting adalah : "sedangkan" dan "terlebih lagi".

2.3 Naïve Bayes

Naïve Bayes bekerja berdasarkan teorema Bayes dan digunakan dalam permasalahan pengklasifikasian (Zhang and Gao, 2011). Penggunaan metode Naïve Bayes dalam peneletian tentang pemrosesan teks sudah banyak kita jumpai. Salah satu manfaat dari penelitian yang berkaitan dengan metode *Naïve Bayes* dalam bidang pemrosesan teks adalah kita bisa mengetahui penulis dari suatu tulisan hanya dengan mengukur kemiripan pola penggunaan kata pada tulisan tersebut dengan tulisan yang sudah menjadi data latih (Saleh, 2014).

Pada permasalahan Naïve Bayes, data latih direpresentasikan dengan set atribut n-dimensional $X = \{X_1, X_2, \ldots, X_n\}$, dimana setiap nilai X memiliki m atribut yang direpresentasikan dengan $x_1, x_2, \ldots, x_m$. Misalkan dalam suatu kasus terdapat sejumlah o kelas yang direpresentasikan dengan $C_1, C_2, \ldots, C_O$ dan diberikan suatu data uji Y yang belum diketahui nilai C-nya, maka peluang atribut x tergolong sebagai kelas C_i ditunjukkan pada Persamaan 1.

$$P(C_i|Y) = \frac{P(Y|C_i)\ x\ P(C_i)}{P(Y)} \quad (1)$$

Keterangan Persamaan 1 :

$P(C_i|Y)$: *posterior*, yaitu peluang kelas C_i bersyarat data Y

$P(Y|C_i)$: *likelihood*, yaitu peluang ditemukannya data Y dikelas C_i

$P(C_i)$: *prior*, yaitu peluang kelas C_i

$P(Y)$: *evidence*, yaitu peluang data Y

Jika pada kasus tersebut memiliki lebih dari satu atribut yang perlu dilatih, maka nilai *likelihood* bisa diperoleh dengan menggunakan Persamaan 2.

$$P(Y|Ci) = \prod_{k=1}^{n} P(y_k|\ C_i) \quad (2)$$

Keterangan Persamaan 2 :

k : indeks untuk menunjukkan nilai atribut y.

n : jumlah atribut pada kasus.

y_k : nilai dari atribut ke k.

Dalam kondisi lain, jika data digunakan pada suatu atribut merupakan data kontinyu berupa data numerik, maka kita bisa menggunakan persamaan pada Gaussian Naïve Bayes untuk memperoleh *likelihood*. Persamaan 3 menunjukkan cara memperoleh *likelihood* pada data kontinyu.

$$P(Y_i|C_k) = \frac{1}{} e x p^{-\frac{(Y_i - \mu_{ik})^2}{2\sigma^2_{ik}}} \quad ((3)$$

Keterangan Persamaan 3 :

$P(Y_i|C_k)$: peluang ditemukannya atribut ke i dari data Y dikelas C_k

i : indeks untuk menunjukkan atribut

k : indeks untuk menunjukkan kelas

μ : nilai rata-rata dari populasi

π : nilai *phi*, yang setara dengan 3,14

σ^2 : nilai varians dari populasi, rumus untuk menghitung nilai varians ditunjukkan pada Persamaan 5
exp : nilai eksponensial, yang ekuivalen dengan 2,7183

Persamaan 4 menunjukkan rumus untuk menghitung nilai rata-rata dari suatu data numerik.

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \quad (4)$$

Keterangan Persamaan 4 :
μ : nilai rata-rata populasi data
x_i : nilai dari data x pada indeks ke i
n : jumlah dari data x

Persamaan 5 menunjukkan rumus untuk menghitung nilai varians dari suatu data numerik.

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n} \quad (5)$$

σ^2 : nilai varians dari populasi
μ : nilai rata-rata dari populasi
x_i : nilai dari data x pada indeks ke i
n : jumlah dari data x

2.4 Evaluasi Hasil Ringkasan

Terdapat dua pendekatan evaluasi untuk mengukur kualitas suatu peringkasan yaitu pendekatan intrinsik dan pendekatan ekstrinsik (Inderjeet, 2009). Pendekatan instrinsik mengukur kualitas hasil ringkasan dilihat dari *term* dan aturan- aturan yang digunakan pada metode peringkasan, sementara pendekatan ekstrinsik mengukur kualitas hasil ringkasan dilihat dari pengaruh hasil ringkasan terhadap tugas tertentu (Klein et al., 1998). Karena evaluasi dengan pendekatan instrinsik berbeda pada setiap kasus, berikut ini penulis paparkan penjelasan dari dua metode evaluasi dengan pendekatan intrinsik.

2.4.1 Precision dan *Recall*

Pada pengujian dengan menggunakan pendekatan ini, diperlukan indeks-indeks dokumen yang dikeluarkan sistem sebagai ringkasan dan indeks-indeks dokumen yang dipilih oleh pakar sebagai kalimat ringkasan (Nenkova dan McKeown, 2011). Pada pengujian ini, jika nilai *precission* dan *recall* tinggi maka kualitas hasil ringkasan baik, sedangkan jika nilai *precission* dan *recall* rendah atau salah satu dari nilai tersebut rendah maka kualitas hasil ringkasan bisa disebut kurang baik (Inderjeet, 2009).

Dalam melakukan pengujian dengan pendekatan *precision* dan *recall*, kesimpulan hasi pengujian ini didapatkan dari perhitungan *f-score*. *F- score* merupakan nilai yang mengkombinasikan nilai *precision* dan *recall* (Steinberger and Ježek, 2009).

Persamaan 6 dan 7 menunjukkan cara penghitungan nilai *precision* dan *recall* pada suatu pengujian, dimana satuan dari nilai *precision* dan *recall* adalah persen. Jumlah kalimat yang dimaksud pada Persamaan 6 dan 7 adalah jumlah kalimat yang dipilih menjadi ringkasan oleh sistem ataupun oleh pakar. Persamaan 8 menunjukkan cara penghitungan *f-score* dimana jika *precision* dan *recall* memiliki nilai maksimal maka *f-score* akan bernilai 1 dan jika *precision* dan *recall* memiliki nilai maksimal maka *f- score* akan bernilai 0, sehingga hasil ringkasan dengan kualitas sangat baik memiliki *f-score* dengan nilai 1.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+NP} \quad (7)$$

Keterangan Persamaan 6 dan 7
TP : jumlah indeks dokumen sistem dan pakar yang beririsan
$TP + FP$: jumlah dokumen yang dipilih sistem sebagai ringkasan
$TP + NP$: jumlah dokumen yang dipilih pakar sebagai ringkasan

$$f - score = \frac{2\ Precision\ Recall}{Precision + Recall} \quad (8)$$

2.4.2 *Relative Utility*

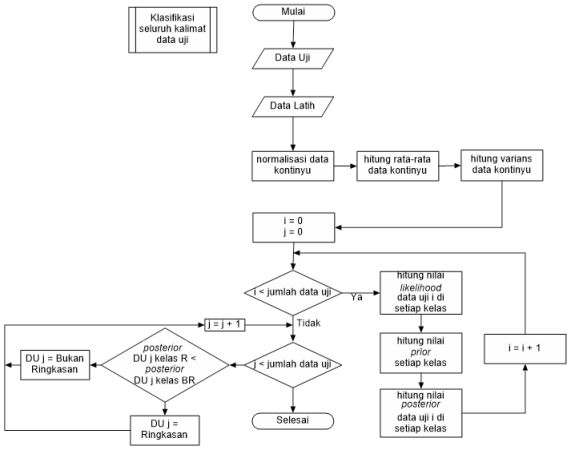
Sebelum melakukan pengujian, teks yang akan diuji diberikan kepada pakar agar pakar memberikan skor 0-10 kepada setiap kalimat, dimana kalimat dengan nilai yang besar merupakan kalimat yang perlu dijadikan sebagai ringkasan (Nenkova dan McKeown, 2011). Hasil dari pengujian ini diukur dengan satuan persen dan penghitungan nilai *relative utility (RU)* ini ditunjukkan pada Persamaan 9 berikut.

$$RU = \frac{total\ skor\ kalimat\ terpilih}{total\ skor\ kalimat\ pada\ dokumen} \quad (9)$$

3. PERANCANGAN DAN IMPLEMENTASI

3.1 Perancangan

Setelah semua fitur diekstrak, langkah selanjutnya adalah melakukan klasifikasi terhadap setiap data uji. Proses klasifikasi data uji secara garis besar ditunjukkan pada Gambar 4.6. Klasifikasi ini bertujuan untuk menggolongkan data uji tersebut kedalam kelas ringkasan atau bukan ringkasan. Pada poin 1 sampai 7 dibawah ini penulis memberikan contoh pengklasifikasian untuk data uji 1.



1. Representasi data latih dan data uji 1 dapat ditunjukkan pada Tabel 3.1.
2. Sesuai tahap pada dasar teori, tahap pertama sebelum klasifikasi adalah normalisasi data numerik. Nilai *content word* dari D1 adalah 1, nilai maksimal dan minimal dari kolom *content word* adalah 3,583 dan -0,2. Nilai hasil normalisasi Tabel ditunjukkan pada Tabel 3.2. Jika batas minimal dan batas maksimal hasil normalisasi yang penulis tentukan adalah 0 dan 1, maka hasil normalisasi *content word* dari D1 bisa diperoleh dari proses berikut.

$$D1_{contentword}{}' = \frac{(X - min)\ x\ (max^* - min^*)}{(\max - min) + min^*}$$

$$D1_{contentword}{}' = \frac{(1 - (-0{,}2))\ x\ (1 - 0)}{(3{,}583 - (-0{,}2)) + 0}$$

$$D1_{contentword}{}' = \frac{1{,}2\ x\ 1}{3{,}783 + 0}$$

$$D1_{contentword}{}' = \frac{1{,}2}{3{,}783}$$

$$D1_{contentword}{}' = 0{,}317$$

3. Langkah selanjutnya adalah menghitung nilai rata-rata setiap fitur pada setiap kelas dari Tabel menggunakan Persamaan 4. Hasil penghitungan nilai rata-rata setiap fitur pada setiap kelas ditunjukkan pada Tabel 3.3. Berikut ini adalah contoh cara menghitung nilai rata-rata dari fitur *content word (CW)* pada kelas R. Terdapat 3 data pada kelas R, sehingga penyebut pada contoh di bawah ini bernilai 3.

$$\mu = \frac{CW\ D1 + CW\ D5 + CW\ D6}{jumlah\ data\ kelas\ R}$$

$$\mu = \frac{0{,}317 + 0{,}660 + 0{,}845}{3}$$

$$\mu = \frac{0{,}823}{3}$$

$$\mu = 0{,}679$$

4. Langkah keempat adalah menghitung nilai varians setiap fitur pada setiap kelas dari Tabel 3.2 menggunakan Persamaan 5. Tabel 4.4 menunjukkan nilai varians dari setiap fitur pada setiap kelas. Nilai rata-rata dari fitur *upper-case word* pada kelas R adalah 0,75, penghitungan nilai varians dari fitur *upper-case word* pada kelas R ditunjukkan di bawah ini. Penulis mempersingkat penulisan *uppercase* kelas R menjadi *UPR*, agar mudah dibaca.

$$\sigma^2 = \frac{(UPR_{D1} - \mu_{UPR})^2 + (UPR_{D5} - \mu_{UPR})^2 + (UPR_{D6} - \mu_{UPR})^2}{jumlah\ data\ kelas\ R}$$

$$\sigma^2 = \frac{(1 - 0{,}75)^2 + (0{,}75 - 0{,}75)^2 + (0{,}5 - 0{,}75)^2}{3}$$

$$\sigma^2 = \frac{(0{,}25)^2 + (0)^2 + (-0{,}25)^2}{3}$$

$$\sigma^2 = \frac{0{,}062 + 0 + 0{,}062}{3}$$

$$\sigma^2 = \frac{0{,}125}{3}$$

$$\sigma^2 = 0{,}416$$

5. Setelah mendapatkan nilai varians, maka langkah selanjutnya dalah menghitung nilai *likelihood* pada semua fitur numerik dan non- numerik. Tabel 4.5 menunjukkan nilai *likelihood* dari setiap fitur pada setiap kelas. NIlai *likelihood* pada fitur selain *occurrence of non- essential* dihitung menggunakan Persamaan 3. Berikut ini penuliskan paparkan contoh menghitung nilai

likelihood dari fitur *content word* pada kelas R dengan data uji 1 sebagai nilai *Y*.

$$P(CW|R) = \frac{1}{\sqrt{2\ x\frac{22}{7}\ x\ 0{,}047}}(2{,}17)^{-\frac{(3{,}583-0{,}607)^2}{2\ x\ 0{,}047}}$$
$$P(contentword|R) = 8{,}315\ x\ (2{,}17)^{-1{,}061}$$
$$P(contentword|R) = 1{,}675$$

Untuk nilai *likelihood* pada fitur *occurrence of non-essential(OCN)* diperoleh dengan menggunakan bagian dari Persamaan 2.6 seperti cara di bawah ini.

$$P(OCN|R) = \frac{jumlah\ data\ false\ pada\ kelas\ R}{jumlah\ data\ kelas\ R}$$
$$P(OCN|R) = \frac{1}{3}$$
$$P(OCN|R) = 0{,}33$$

6. Langkah terakhir dari klasifikasi data uji 1 adalah menghitung peluang data data uji 1 berada dikelas R dan BR menggunakan Persamaan 2.6. Persamaan 2.6 digunakan untuk membandingkan nilai $P(R|DU1)$ dan nilai $P(BR|DU1)$, sehingga pada penghitungan, nilai $P(Y)$ bisa kita abaikan karena bernilai sama pada setiap kelas. Dari hasil penghitungan kedua nilai tersebut, maka DU1 akan digolongkan ke kelas dengan nilai *P* terbesar. Nilai $P(DU1|R)$ diperoleh dari persamaan 2.7, yaitu dengan mengalikan nilai *likelihood* setiap fitur dari kelas R sehingga diperoleh nilai 1,219. 3 dari 6 data latih berada pada kelas R sehingga nilai $P(R)$ adalah $\frac{3}{6}$ atau $\frac{1}{2}$.

$$P(R|DU1) = P(DU1|R)\ x\ P(R)$$
$$P(R|DU1) = 1{,}219\ x\ \frac{1}{2}$$
$$P(R|DU1) = 0{,}609$$

7. Nilai *posterior* data uji 1 di kelas R lebih besar dari nilai *posterior* dara uji 1 di kelas BR, sehingga data uji 1 digolongkan ke dalam kelas *R* dan dan dijadikan sebagai ringkasan. Pengklasifikasian dilanjutkan terhadap DU2, DU3, dan DU4. Hasil lengkap nilai *posterior* dari setiap data uji pada setiap kelas ditunjukkan pada Tabel 4.6, dimana DU1 dan DU3 tergolong ke dalam kelas ringkasan karena nilai peluang ringkasan pada DU1 dan DU3 lebih besar daripada peluang bukan ringkasan-nya.
8. Tabel 4.6 menunjukkan data uji yang termasuk ringkasan adalah DU1 dan DU3. Oleh karea itu, ringkasan dari data pada Tabel 3.3 adalah gabungan dari DU1 dan DU3, seperti yang ditunjukkan pada Tabel 3.20.

3.2 Implementasi

Tampilan awal ketika sistem pertama kali dijalankan ditunjukkan pada Gambar 3.2.1. Representasi data awal yang merupakan hasil *parsing* langsung ditampilkan perdokumen pada tab 1 bersamaan dengan judul dokumen PDF dan kata kunci seperti pada Gambar 3.2.2. Gambar 3.2.3 menunjukkan tokenisasi, *filtering*, dan *stemming* dari data awal pada Gambar 3.2.2.

Gambar 3.2.4 merupakan proses penghitungan skor TF-IDF atau ekstraksi fitur statistik pada setiap dokumen. Gambar 3.2.4 berisi tahapan penghitungan skor TF-IDF secara detail seperti yang ditunjukkan pada subbab 3.3.2.1. Gambar 3.2.5 menunjukkan hasil ekstraksi fitur linguistik pada setiap dokumen. Gambar 3.2.6 menunjukkan proses klasifikasi data uji yang berupa normalisasi, nilai rata-rata, nilai varians, nilai *likelihood*, nilai *posterior*, dan kelas dari data uji. Sementara Gambar 3.2.7 menampilkan ringkasan dari dokumen PDF yang telah dimasukkan, hasil ringkasan ditampilkan dalam bentuk paragraf.

Tabel 3.1.1 Representasi Data Latih dan Data Uji

Fitur Linguistik	**Content word**	**Title word**	**Sentence location**	**Sentence length**	**Upper-case word**	**Cue-phrase**	**Biased word**	**Occurrence of non-essential**	**Kelas**
D1	1	15	1	16	5	2	3	FALSE	R
D2	0.4	3	8	4	1	0	0	TRUE	BR
D3	0.8	1	4	17	1	0	0	FALS	BR

								E	
D4	-0.2	4	6	2	2	1	1	FALSE	BR
D5	2.3	1	15	13	4	1	4	TRUE	R
D6	3	3	2	11	3	2	1	TRUE	R
DU1	3.5831387	2	1	17	3	0	2	FALSE	?

Tabel 3.1.2 Representasi Data Latih dan Data Uji

Fitur Linguistik	Content word	Title word	Sentence location	Sentence length	Upper-case word	Cue-phrase	Biased word	Kelas
D1	0,317	1	0	0,933	1	1	0,75	R
D2	0,158	0,142	0,5	0,133	0	0	0	BR
D3	0,264	0	0,214	1	0	0	0	BR
D4	0	0,214	0,357	0	0,25	0,5	0,25	BR
D5	0,660	0	1	0,733	0,75	0,5	1	R
D6	0,845	0,142	0,071	0,6	0,5	1	0,25	R
D7	1	0,0711	0	1	0,5	0	0,5	?

Tabel 3.1.3 Rata-rata Fitur Setiap Kelas

Kelas	Content word	Title word	Sentence location	Sentence length	Upper-case word	Cue-phrase	Biased word
R	0,607	0,380	0,357	0,755	0,75	0,833	0,666
BR	0,140	0,119	0,357	0,377	0,083	0,166	0,083

Tabel 3.1.4 Varians Fitur Setiap Kelas

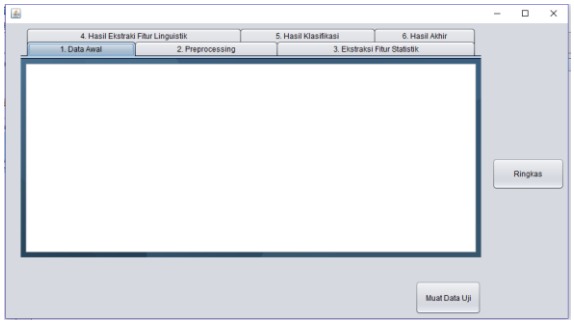
Kelas	Content word	Title word	Sentence location	Sentence length	Upper-case word	Cue-phrase	Biased word
R	0,047	0,195	0,207	0,018	0,041	0,055	0,097
BR	0,011	0,007	0,013	0,196	0,013	0,055	0,013

Tabel 3.1.5 Peluang Fitur pada Setiap Kelas

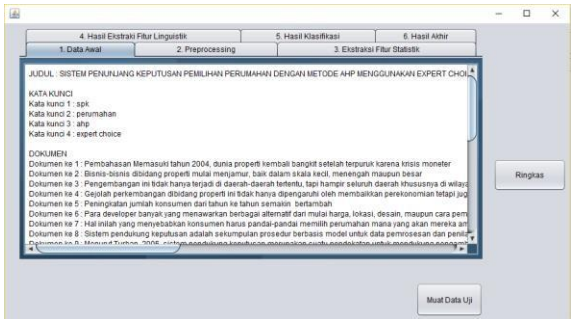
Kelas	Content word	Title word	Sentence location	Sentence length	Upper-case word	Cue-phrase	Biased word	Occurrence of non essential
R	1,675	1,60	1,413	4,326	4,522	0,013	3,557	0,333
BR	8,90464E-13	43,575	0,270	0,758	0,055	5,592	0,0555	0,666

Tabel 3.1.6 Kelas dari Setiap Data Uji

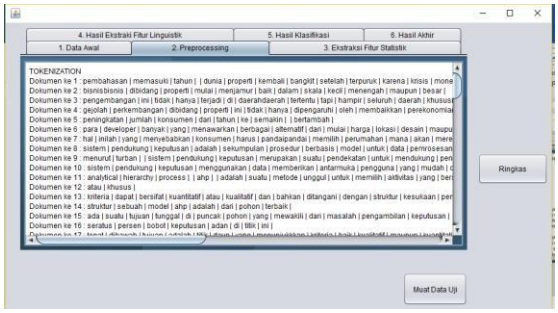
Data Uji	Peluang Ringkasan / Peluang Bukan Ringkasan	Kelas
DU1	0,609709665 / 4,55295E-14	Ringkasan
DU2	0,000407765 / 0,502969282	Bukan ringkasan
DU3	0,80438516 / 0,00124053	Ringkasan
DU4	2,27648E-14 / 0,00020388	Bukan ringkasan



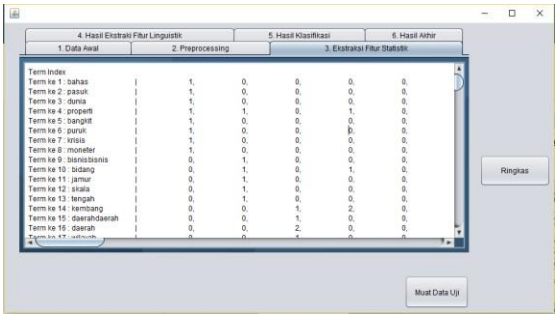
Gambar 3.2.1 Antar Muka Awal Hasil Implementasi



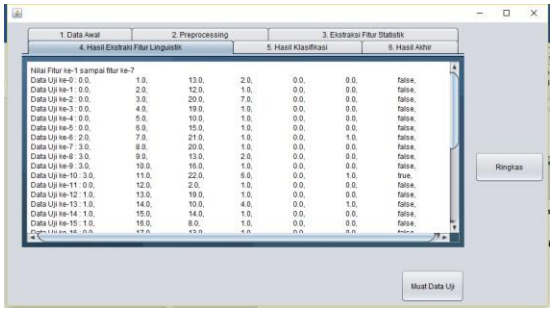
Gambar 3.2.2 Antar Muka Hasil *Parsing* dari Data Uji



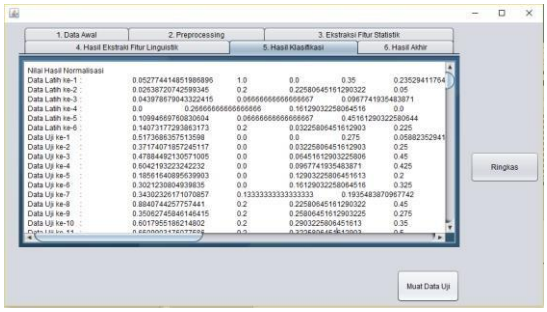
Gambar 3.2.3 Antar Muka Hasil *Preprocessing* Data Uji



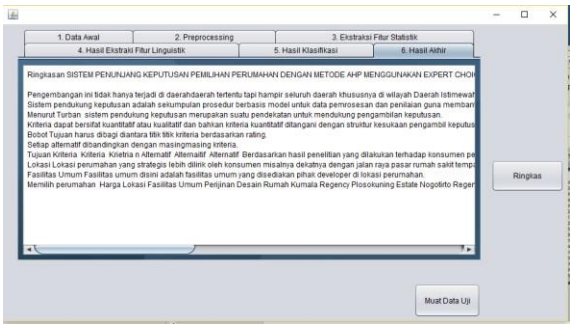
Gambar 3.2.4 Antar Muka Hasil Ekstraksi Fitur Statistik



Gambar 3.1.5 Antar Muka Hasil Esktraksi Fitur Linguistik



Gambar 3.1.6 Antra Muka Proses Klasifikasi Data Uji



Gambar 3.1.7 Antar Muka Hasil Peringkasan

4. PENGUJIAN DAN ANALISIS

4.1 Hasil Pengujian Terhadap Fitur Statistik

Pada bagian ini penulis melakukan penghitungan nilai *precision*, *recall*, dan *relative utility* dari hasil ringkasan dengan basis fitur statistik. Peringkasan pada subbab ini

merupakan peringkasan dengan menggunakan skor TF-IDF, dimana pada akhir penghitungan setiap kalimat akan memiliki skor TF-IDF. Kemudian, ringkasan dipilih dari sejumlah *n*% kalimat yang memiliki skor TF-IDF terbesar.

Subbab 4.1.1 membahas hasil pengujian dari peringkasan statistik dengan jumlah kalimat yang dikeluarkan sistem sebanyak 10% dari jumlah kalimat pada data uji. Subbab 4.1.2 membahas hasil pengujian dari peringkasan statistik dengan jumlah kalimat yang dikeluarkan sistem sebanyak 20% dari jumlah kalimat pada data uji.

4.1.1 Hasil Pengujian Fitur Statisik Skenario 1

F-score dan nilai *relative utility* yang maksimal berada pada pengujian Data Uji 4. Nilai *f- score* minimal berada pada pengujian Data Uji 3 dan Data Uji 5, yaitu 0. Nilai *relative utility* minimal berada pada pengujian Data Uji 3. Hasil lengkap dari pengujian fitur statistik dengan skenario 1 ditunjukkan pada Tabel 3.1.

4.1.2 Hasil Pengujian Fitur Statisik Skenario 2

F-score dan nilai *relative utility* yang maksimal berada pada pengujian Data Uji 4, tetapi nilai tersebut lebih rendah dari pengujian Data Uji 4 pada Tabel 3.1. Nilai *f-score* minimal berada pada pengujian Data Uji 5, yaitu 0.2307. Nilai *relative utility* minimal berada pada pengujian Data Uji 2. Hasil lengkap dari pengujian fitur statistik dengan skenario 2 ditunjukkan pada Tabel 6.2.

Nilai rata-rata *f-score* dan *relative utility* pada skenario 2 lebih besar daripada skenario 2, sehingga pengulis membuat kesimpulan bahwa hasil pengujian peringkasan berbasis statistik dengan skenario 2 lebih unggul dari pengujian dengan skenario 1. Peringkasan statistik dengan skenario 2 lebih unggul daripada peringkasan statitik dengan skenario 1 karena peringkasan statistik dengan skenario 2 memiliki jumlah data yang dikeluarkan lebih banyak daripada peringkasan statistik dengan skenario 1. Karena peringkasan statistik dengan skenario 2 memiliki jumlah kalimat ringkasan yang lebih banyak, maka peluang menemukan kalimat ringkasan sistem dan pakar yang beririsan lebih besar, sehingga nilai *precision*, *recall*, dan *relative utility* pada peringkasan statistik dengan skenario 2 lebih tinggi daripada skenario 1.

Tabel 4.1.1 Hasil Pengujian Peringkasan Statistik dengan Keluaran 10%

Data	*Precision*	*Recall*	*F-score*	*Relative Utility*
Data Uji 1	0.1666667	0.1	0.125	0.104972
Data Uji 2	0.3333333	0.125	0.1818182	0.066667
Data Uji 3	0	0	0	0.012821
Data Uji 4	0.7142857	0.555556	0.625	0.317241
Data Uji 5	0	0	0	0.064103
Rata-rata			0.1863336	0.113161

Tabel 4.1.1 Hasil Pengujian Peringkasan Statistik dengan Keluaran 20%

Data	Precisi on	Recall	F-score	Relative Utility
Data Uji 1	0.2307692	0.3	0.26086956	0.237569
Data Uji 2	0.3333333	0.25	0.28571428	0.133333
Data Uji 3	0.2222222	0.4	0.28571426	0.217949
Data Uji 4	0.4	0.666667	0.5	0.510345
Data Uji 5	0.1875	0.3	0.230769231	0.230769
Rata- rata			0.312613473	1.329965

4.2 Hasil Pengujian Terhadap Fitur Linguistik

Pada bagian ini penulis melakukan penghitungan nilai *precision*, *recall*, dan *relative utility* dari hasil ringkasan dengan basis fitur linguistik. Peringkasan pada subbab ini merupakan peringkasan dengan menggunakan fitur yang hanya terdapat pada subbab 3.3.2.2. Hasil ringkasan diperoleh setelah melakukan klasifikasi seperti yang dijelaskan pada subbab 3.3.3 dimana hasil ringkasan adalah kumpulan kalimat yang berada pada kelas R.

F-score dan nilai *relative utility* yang maksimal dari penelitian ini berada pada pengujian Data Uji 5. Nilai *f-score* minimal berada pada pengujian Data Uji 1. Hasil lengkap dari pengujian fitur linguistik ditunjukkan pada Tabel 6.3. Tabel 6.3 menunjukkan bahwa rata-rata *f-score* dan *relative utility* pada peringkasan

linguistik lebih rendah daripada rata-rata *f-score* dan *relative utility* pada peringkasan statistik dengan skenario 2 tetapi lebih tinggi daripada rata-rata *f-score* dan *relative utility* pada peringkasan statistik dengan skenario 1. Ini menunjukkan bahwa kualitas peringkasan linguistik berada di posisi kedua terbaik dari ketiga basis peringkasan yang disebutkan pada kalimat sebelum ini.

Peringkasan statistik dengan skenario 1 memiliki persentase kalimat ringkasan yang sedikit, sehingga peluang untuk menemukan menemukan kalimat ringkasan sistem dan pakar yang beririsan kecil. Hal sebaliknya berlaku untuk peringkasan statistik dengan skenario 2. Pada peringkasan dengan pendekatan linguistik, jumlah kalimat ringkasan tidak dapat ditentukan oleh penulis karena kalimat yang dapat dijadikan ringkasan hanyalah kalimat yang berada pada kelas R. Keadaan seperti yang dijelaskan pada kalimat sebelumnya membuat peringkasan dengan pendekatan linguistik memiliki jumlah kalimat ringkasan yang tidak sebanyak jumlah kalimat ringkasan pada peringkasan statistik dengan skenario 2, namun tidak lebih sedikit daripada jumlah kalimat ringkasan pada peringkasan statistik dengan skenario. Sehingga, kualitan peringkasan dengan pendekatan linguistik berada pada posisi kedua dari ketiga pendekatan yang telah disebutkan pada kalimat sebelumnya.

Tabel 4.2.1 Hasil Pengujian Peringkasan Statistik dengan Keluaran 20%

Data	Precision	Recall	F-score	Relative Utility
Data Uji 1	0	0	0	0.016575
Data Uji 2	1	0.25	0.4	0.133333
Data Uji 3	0.1666667	0.1	0.125	0.051282
Data Uji 4	1	0.111111	0.2	0.055172
Data Uji 5	0.3125	0.5	0.384615385	0.378205
Rata-rata			0.221923077	0.126914

4.3 Hasil Pengujian Terhadap Fitur Statistik dan Linguistik

Pada bagian ini penulis melakukan penghitungan nilai *precision*, *recall*, dan *relative utility* dari hasil ringkasan dengan basis fitur linguistik dan statistic yang telah dijelaskan sejak awal pada penelitian ini. Peringkasan dengan fitur statistik dan linguistik telah dijelaskan secara lengkap pada subbab 3.3. Hasil ringkasan diperoleh setelah melakukan klasifikasi seperti yang dijelaskan pada subbab 3.3.3 dimana hasil ringkasan adalah kumpulan kalimat yang berada pada kelas R.

F-score dan nilai *relative utility* yang maksimal dari penelitian ini berada pada pengujian Data Uji 5. Nilai *f-score* minimal berada pada pengujian Data Uji 1. Hasil lengkap dari pengujian fitur linguistik ditunjukkan pada Tabel 6.4. Tabel 6.3 menunjukkan bahwa rata-rata *f-score* dan *relative utility* pada peringkasan dengan fitur statistik dan linguistik lebih rendah daripada rata-rata *f-score* dan *relative utility* pada peringkasan statistik dengan skenario 2 dan peringkasan linguistik, tetapi lebih tinggi daripada rata-rata *f-score* dan *relative utility* pada peringkasan statistik dengan skenario 1. Ini menunjukkan bahwa kualitas peringkasan statistik dan linguistik berada di posisi ketiga terbaik dari keempat pendekatan peringkasan yang disebutkan pada kalimat sebelum ini.

Pada peringkasan dengan pendekatan statistik dan linguistik, nilai *f-score* dan *relative utility* pada Data Uji 1 sampai Data Uji 4 ekuivalen. Perbedaan nilai *f-score* dan *relative utility* hanya terdapat pada Data Uji 5, di mana selisih *f-score* Data Uji 5 pada Tabel 6.3 dan 6.4 adalah 0,07692338 dan selisih *relative utility* Data Uji 5 pada Tabel 6.3 dan 6.4 adalah 0,51282. Selisih tersebut bukanlah nilai yang besar, sehingga penulis menganalisis bahwa kualitas hasil ringkasan dengan pendekatan linguistik memiliki kualitas yang sama dengan peringkasan dengan pendekatan statistik dan linguistik.

Tabel 4.2.1 Hasil Pengujian Peringkasan Statistik dan Linguistik

Data	*Precision*	*Recall*	*F-score*	*Relative Utility*
Data Uji 1	0	0	0	0.016575
Data Uji 2	1	0.25	0.4	0.133333
Data Uji 3	0.166667	0.1	0.125	0.051282
Data Uji 4	1	0.111111	0.2	0.055172
Data Uji 5	0.25	0.4	0.307692	0.326923

Rata-rata	[redacted]	[redacted]	0.206538	0.116657

5. KESIMPULAN DAN SARAN

Kualitas hasil ringkasan metode *Gaussian Naïve Bayes* dengan fitur statistik dan linguistik tidak lebih baik daripada peringkasan menggunakan metode *Gaussian Naïve Bayes* dengan fitur linguistik. Di mana peringkasan metode *Gaussian Naïve Bayes* dengan fitur statistik dan linguistik memiliki nilai rata-rata *f-score* 0,206538 dan rata- rata *relative utility* 0,116657.

Peringkasan ini menggunakan 1 fitur statistik dan 7 fitur linguistik, sehingga penulis berharap penelitian ini bisa dikembangkan dengan jumlah fitur linguistik yang lebih banyak. Penggunaan data uji juga perlu diperbarui dengan menambahkan kalimat ringkasan agar peluang menemukan kalimat ringkasan besar.

6. DAFTAR PUSTAKA

Abbasi-ghalehtaki, R., Khotanlou, H. and Esmaeilpour, M., 2016. Fuzzy evolutionary cellular learning automata model for text summarization. *Swarm and Evolutionary Computation*, [online] pp.1–16.

Babar, S.A. and Patil, P.D., 2015. Improving Performance of Text Summarization. *Procedia Computer Science*, [online] 46(Icict 2014), pp.354–363.

Batcha, N.K., Aziz, N.A. and Shafie, S.I., 2013. CRF Based Feature Extraction Applied for Supervised Automatic Text Summarization. *Procedia Technology*, [online] 11(Iceei), pp.426–436.

Das, D., 2007. A Survey on Automatic Text Summarization Single-Document Summarization. pp.1–31.

Emre, F., Akay, D. and Yager, R.R., 2016. An overview of methods for linguistic summarization with fuzzy sets. *Expert Systems with Applications*, [online] 61, pp.356–377.

Griffis, J.C., Allendorfer, J.B. and Szaflarski, J.P., 2016. Voxel-based Gaussian naïve Bayes classification of ischemic stroke lesions in individual T1-weighted MRI scans. *Journal of Neuroscience Methods*, [online] 257, pp.97–108.

Gupta, V., Science, C. and Lehal, G.S., 2010. A Survey of Text Summarization Extractive Techniques. 2(3), pp.258–268.

Inderjeet, M., 2009. Summarization Evaluation: An Overview. *Pflege Zeitschrift*, 62(6), pp.337–341.

Klein, G., Hirschman, L., Firmin, T., Diego, S., Mani, I. and House, D., 1998. The TIPSTER SUMMAC Text Summarization Evaluation. *Methods*, pp.77–85.

Mehta, P., 2016. From Extractive to Abstractive Summarization : A Journey. pp.100–106.

Munková, D., Munk, M. and Vozár, M., 2013. Data pre-processing evaluation for text mining: Transaction/sequence model. *Procedia Computer Science*, 18, pp.1198–1207.

Nenkova, A. and McKeown, K., 2011. Automatic Summarization. *Foundations and Trends® in Information Retrieval*, [online] 5(3), pp.235–422.

Oliveira, H., Ferreira, R., Lima, R., Lins, R.D., Freitas, F., Riss, M. and Simske, S.J., 2016. Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. *Expert Systems with Applications*, [online] 65, pp.68–86.

Saleh, A., El, M. and Menai, B., 2014. Naı ¨ ve Bayes classifiers for authorship attribution of Arabic texts. *Journal of King Saud University - Computer and Information Sciences*, [online] 26(4), pp.473–484.

Shouzhong, T., 2016. Mining microblog user interests based on TextRank with TF-IDF factor. *The Journal of China Universities of Posts and Telecommunications*, [online] 23(5), pp.40–46.

Steinberger, J. and Ježek, K., 2009. Evaluation measures for text summarization. *Computing and Informatics*, 28(2), pp.251–275.

Tala, F.Z., 2003. A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. *M.Sc. Thesis, Appendix D*, pp, pp.39–46.

Tayal, M.A., Raghuwanshi, M.M. and Malik, L.G., 2017. ATSSC: Development of an approach based on soft computing for text summarization. *Computer Speech & Language*, [online] 41, pp.214–235.

Uysal, A.K. and Gunal, S., 2014. The impact of preprocessing on text classification.

Information Processing and Management, [online] 50(1), pp.104–112.

Zhang, W. and Gao, F., 2011. Procedia Engineering An Improvement to Naive Bayes for Text Classification. [online] 15, pp.2160–2164.