PENGARUH *TEXT PREPROCESSING* DAN KOMBINASINYA PADA PERINGKAS DOKUMEN OTOMATIS TEKS BERBAHASA INDONESIA

Hadiyatun Najjichah[1], Abdul Syukur [2], Hendro Subagyo [3]
[12]Pasca Sarjana Teknik Informatika Universitas Dian Nuswantoro
[3]Lembaga Ilmu Pengetahuan Indonesia

ABSTRACT

Numbers of information increased in concordance to the growth of them digitally. And many of informations available on the internet are in textual form. It is necessary for the information seekers to get the what they require. Automatic Text Summarization is a process of summarizing done by machine through certain methods to get a shorter form of document while still preserving the gist. This research is to examine the influence of preprocessing text and its combination to the Automatic Text Summarization of Bahasa Indonesia. Method used are segmentation, Tokenization, Stopword removal, Stemming and N-Gram. There are 14 steps of combination. The results shows that indeed there is influence of those combinations to the Automatic Text Summarization.The highest F-Measure is resulted on combinations step of Tokenization >> 2-gram >> Summarization, with 66% accuracy. While the lowest is resulted from the combination step of Tokenization >> 3-gram >> Summarization and process of Tokenization >> 4-gram >> Summarization with 63% accuracy.

Keywords : Automatic text Summarization, extractive feature based summary, Tokenization, Stopword removal, stemmin

1. PENDAHULUAN

Jumlah informasi yang tersedia saat ini berkembang pesat sejalan dengan pertumbuhan informasi secara *digital*. Saat ini, banyak informasi yang berasal dari internet dalam bentuk tekstual. Hal ini menjadi sebuah tantangan untuk menemukan dokumen yang relevan di web yang ditangani dengan teknik pencarian informasi digunakan di mesin pencari seperti *Google, Bing, Yahoo,* dan lain-lain[1]. Seorang pengguna umumnya mencari data melalui peringkat atas halaman dan mencoba untuk menemukan potongan-potongan informasi yang dia inginkan secara manual. Ringkasan singkat yang diambil dari tiap halaman akan sangat membantu dalam situasi seperti itu.

Peringkas dokumen teks otomatis mempunyai dua tipe yaitu metode berbasis ekstraksi dan metode berbasis abstraksi. Metode berbasis abstraksi dibuat untuk mendapatkan informasi yang terkandung dalam sumber asli dan menghasilkan teks yang mengungkapkan informasi yang sama dalam cara yang lebih ringkas. Sedangkan Peringkas teks berbasis ekstraksi dibuat dengan memilih potongan teks seperti (kata, frasa, kalimat, paragraf) dari teks aslinya dan mengorganisir mereka dalam cara untuk menghasilkan ringkasan yang koheren2]. Dan dalam metode berbasis ekstraksi terdapat dua metode yaitu metode fitur dan metode LSA (Latent Semantic Analysis).

Sebagai metode peringkas otomatis, metode peringkasan berbasis fitur maupun LSA mendasarkan proses komputasi untuk menemukan bagian terpenting dari dokumen asal dengan perhitungan bobot atas *term-term* dalam model vektor ruang (Vector Space Model, VSM) sebagai representasi dokumen teks untuk input komputasi. Dalam hal ini, berbagai teknik *preprocessing* dokumen teks memainkan peran penting sebagai langkah awal yang akan berpengaruh terhadap tingkat akurasi dan waktu komputasi proses peringkasan otomatis. Beberapa teknik *preprocessing* yang umum digunakan adalah :

segmentation, Tokenization, Stopword removal, Stemming dan *N-Gram.*

Berbagai macam penelitian mengenai teknik peringkasan dokumen otomatis dan teknik *preprocessing* nya telah dilakukan dan memberikan sumbangan yang signifikan. Diantaranya adalah :[4], menawarkan metode berbasis algoritma *fuzzy logic* dalam ekstraksi kalimat untuk keperluan peringkasan. Hasilnya menunjukkan peringkasan dengan metode *fuzzy logic* lebih tinggi dibanding dengan metode peringkas dalam *Microsoft Word* 2007. [8], menggunakan metode *Latent Semantic Analysis* (LSA *)*, untuk keperluan peringkasan dokumen teks berbahasa Turki. [5], menawarkan teknik N-Gram untuk meningkatkan LSI. Diperoleh hasil yang positif dalam hal *querying* dan *clustering* dokumen dalam database. [9], mencoba untuk mengkombinasikan beberapa metode teks *preprocessing* untuk mendeteksi *plagiarism*. [6], meneliti tentang berbagai metode peringkasan dokumen otomatis dengan menggunakan teknik ekstraksi fitur. [7], menawarkan teknik fitur untuk meringkas dokumen dengan dua tahap teks *preprocessing*, *Stopword* dan *Stemming* tanpa melibatkan teknik *segmentation*, *Tokenization* dan *N-Gram.*

Berdasarkan beberapa penelitian diatas, muncul berbagai pertanyaan yang bisa diajukan untuk melakukan penelitian lebih lanjut seperti; bagaimanakah hasil teks *preprocessing N-Gram* bila digunakan pada peringkas otomatis berbasis fitur ? Bagaimanakah pengaruh kombinasi berbagai teknik teks *preprocessing* bila diterapkan pada peringkas otomatis berbasis fitur ? Penelitian ini ingin meneliti pengaruh berbagai kombinasi atas ke lima teknik teks *preprocessing* yang umum digunakan diatas terhadap tingkat akurasi dan waktu komputasi peringkas otomatis berbasis fitur.

Pilihan atas metode peringkas otomatis berbasis fitur dalam penelitian ini didasarkan atas berbagai pertimbangan. Metode fitur memiliki kelebihan, meskipun pada saat bersamaan juga memiliki kekurangan dibanding metode berbasis LSA. Metode fitur mempertimbangkan setidaknya 5 fitur dalam perhitungan skor pembototan *term*, dimana frekuensi munculnya *term* juga menjadi salah satunya. Metode LSA hanya mendasarkan diri kepada pembobotan berbasis frekuensi *term*. Metode fitur memiliki kelemahan yang dalam hal ini menjadi kelebihan metode LSA yaitu kemampuan untuk melakukan LSI (Latent Semantic Indexing) sebagai bagian dari SVD (Singular Value Decomposition) dalam reduksi fitur. SVD akan memperkecil dimensi VSM sekaligus menemukan keterkaitan makna semantik atas *term-term* didalam dokumen.

Karena kelemahan metode peringkasan otomatis berbasis fitur tersebut diatas, diyakini bahwa peluang untuk memperkecil terjadinya *noise* akibat munculnya banyak nilai nol pada *VSM* sebagai representasi dokumen akan bergantung kepada secara signifikan kepada tahapan *preprocessing*-nya. Berbagai kombinasi teknik *segmentation, tokenizatio, Stopword removal, Stemming,* dan *N-Gram* bisa memperbaiki tingkat akurasi peringkasan daripada yang sudah dicapai oleh penelitian sebelumnya yang tidak mengkombinasikan secara lebih lengkap berbagai teknik *preprocessing* tersebut.

Selain itu, karena sifat sebagian teknik *preprocessing* diatas; *segmentation, Stopword removal,* dan *Stemming,* adalah *language dependent,* berbagai kombinasi teknik tersebut belum diaplikasikan untuk dokumen teks berbahasa Indonesia pada penelitian-penelitian sebelumnya.

1.1. Rumusan Masalah

Berdasarkan analisa masalah sebelumnya, masalah dalam penelitian ini dirumuskan sebagai berikut : signifikansi kombinasi lima teknik *teks preprocessing, segmentation, Tokenization, Stopword removal, Stemming* dan *N-Gram* pada tingkat akurasi dan waktu komputasi Peringkas Dokumen Otomatis Berbasis Fitur Atas Dokumen Teks berbahasa Indonesia. Diyakini bahwa kombinasi yang tepat atas lima teknik *preprocessing* tersebut akan mengurangi *noise* yang pada metode peringkasan berbasis fitur dipengaruhi secara signifikan oleh tahapan *preprocessing*-nya. Dengan demikian tingkat akurasi peringkasan maupun waktu komputasi bisa menjadi lebih optimal.

1.2. Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah di atas, maka dapat dirumuskan tujuan penelitian sebagai berikut :

Untuk membuktikan teknik *text preprocessing; segmentation, Tokenization, Stopword removal, Stemming* dan *N-Gram* dan kombinasinya pada tingkat akurasi dan waktu komputasi Peringkas Teks Otomatis Dokumen berbahasa Indonesia dan untuk mengetahui kombinasi terbaik dari kelima teknik teks *preprocessing* tersebut pada peringkas teks otomatis dokumen berbahasa Indonesia.

1.3. Manfaat penelitian

Manfaat dari penelitian ini adalah :

a. Manfaat ilmiah dari penelitian ini diharapkan dapat digunakan untuk mengetahui tingkat akurasi dan waktu komputasi dari hasil percobaan berbagai kombinasi lima teknik teks *preprocessing; segmentation, Tokenization, Stopword removal, Stemming* dan *N-Gram* pada Peringkas Teks Otomatis Dokumen berbahasa Indonesia

b. Manfaat ilmiah hasil penelitian tersebut diatas diharapkan dapat memberikan sumbangan bagi pengembangan teori yang berkaitan dengan Peringkas Otomatis Dokumen Teks berbahasa Indonesia.

2. KERANGKA PEMIKIRAN PENGARUH *TEXT PREPROCESSING* DAN KOMBINASINYA PADA PERINGKAS DOKUMEN OTOMATIS TEKS BERBAHASA INDONESIA

Berbagai kombinasi teknik *segmentation, tokenizatio, Stopword removal, Stemming,* dan *N-Gram* bisa memperbaiki tingkat akurasi peringkasan. Metode fitur mempertimbangkan setidaknya 5 fitur dalam perhitungan skor pembototan *term*, frekuensi munculnya *term* juga menjadi salah satunya. Diyakini bahwa kombinasi yang tepat atas lima teknik *preprocessing* tersebut akan mengurangi *noise* yang pada metode peringkasan berbasis fitur dipengaruhi secara signifikan oleh tahapan *preprocessing*-nya. Dengan demikian tingkat akurasi peringkasan maupun waktu komputasi bisa menjadi lebih optimal.

Metode berbasis fitur memiliki karakteristik sebagai berikut. [2].

a. *Title feature*

Merupakan penghitungan jumlah kata yang ada dalam kalimat judul, rasio dari jumlah kata dalam kalimat dokumen dengan kata yang terdapat pada judul, kata yang terdapat pada kalimat dokumen yang merupakan bagian dari kata dalam judul mempunyai skor yang tinggi [2].

$$score(S_i) = \frac{No.TitleWordinSi}{No.WordinTitle} \quad (4)$$

b. *Sentence length*

Fitur ini berguna untuk menyaring kalimat pendek seperti *datelines* dan nama penulis yang biasa ditemukan pada artikel-artikel berita, dimana kalimat pendek tersebut tidak diharapkan muncul pada ringkasan dokumen. Adalah rasio dari jumlah kata dalam kalimat dengan jumlah kata yang terdapat pada kalimat terpanjang pada suatu dokumen.

$$Skor\ (S_i) = \frac{Jumlahkatayangterdapatpadakalimat}{jumlahkatayangterdapatpadakalimatterpanjang} \quad (5)$$

c. *Term weight*

Menghitung frekuensi munculnya sebuah *term* pada dokumen yang biasa digunakan untuk menentukan penting tidaknya posisi kalimat pada sebuah dokumen.Perhitungan rata-rata TF-ISF *(Term Frequency, Inverse Sentence Frequency)* adalah sebagai berikut:

$$Skor\ (S_i) = \frac{JumlahTF-ISFdalamkalimat}{MaksimaljumlahTF-ISF} \quad (6)$$

$$TSF - ISF = term\ frequency\ \times idf$$

$$= term\ frequency\ \times \log\left(\frac{df}{N}\right)$$

Keterangan :
df = jumlah kalimat yang mengandung kata x
term frequency = jumlah kata pada dokumen (dalam bentuk matrik)
N = jumlah kalimat dalam pada dokumen

d. *Sentence to sentence similiarity*

Kesamaan antar kalimat, dimisalkan kalimat *s*, pengukuran kesamaan antara kalimat *s* dengan kalimat lainnya dengan menghitung rasio dari ringkasan kesamaan kalimat pada kalimat *s* tersebut dengan maksimum ringkasan jumlah dari keseluruhan kesamaan kalimat pada dokumen.

$$sim(s_i, s_j) = \frac{\Sigma_{k=1}^{n} w_{ik} \times w_{jk}}{\sqrt{\Sigma_{k=1}^{n} w_{ik}^2} \times \sqrt{\Sigma_{k=1}^{n} w_{jk}^2}} \quad (8)$$

Keterangan :
w_{ik} = Bobot kata pada kalimat i
w_{jk} = Bobot kata pada kalimat j

e. *Thematic word*

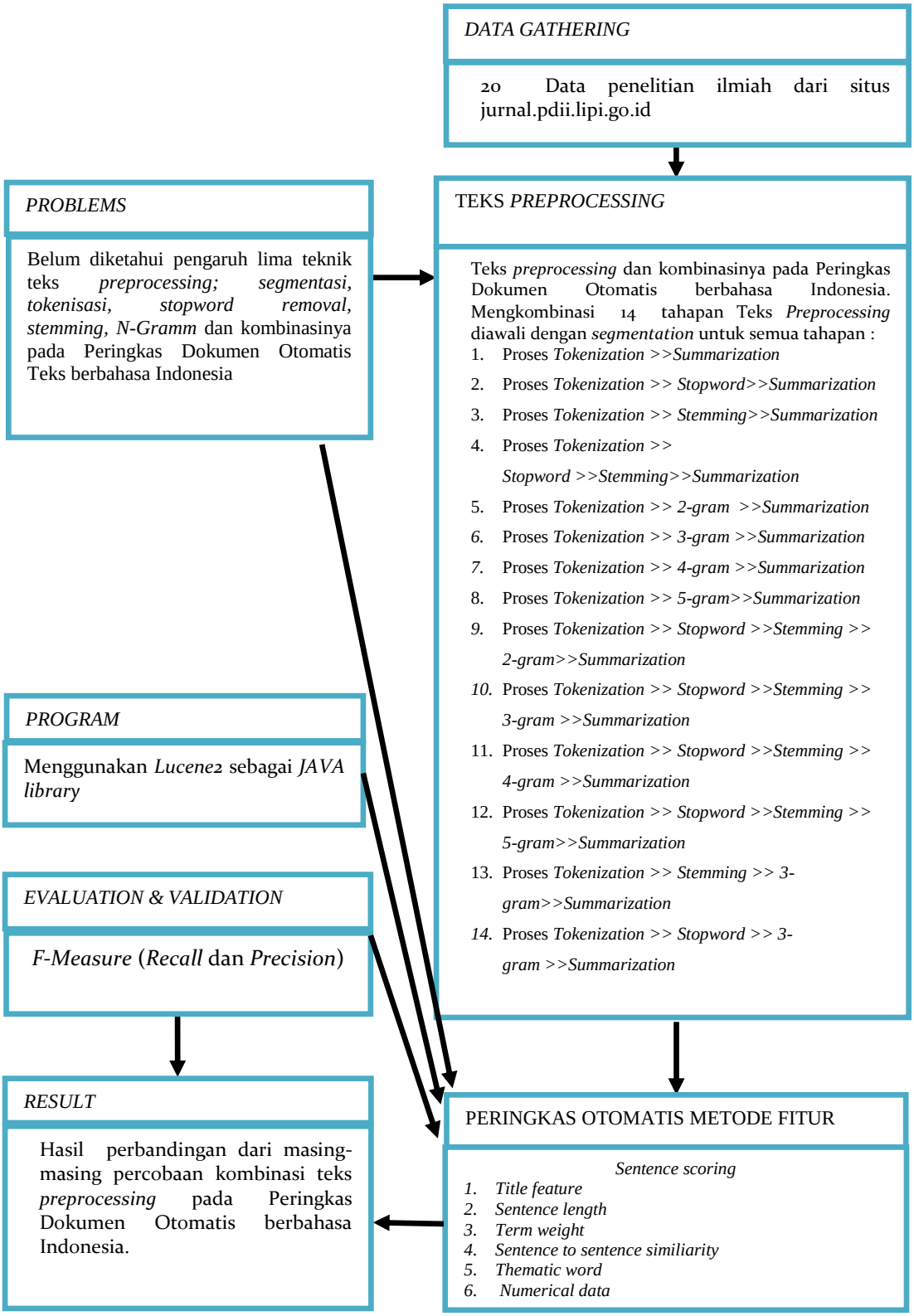
Merupakan jumlah kata tematik yang ada dalam kalimat. Fitur ini penting karena *term* yang ada dalam dokumen sering terkait dengan topik suatu dokumen. Jumlah kata tematik menunjukan kata-kata dengan relativitas maksimal. Skor dalam fitur ini dihitung dari rasio jumlah kata tematik dalam kalimat dengan panjang kalimat dalam dokumen.

$$Skor\ (S_i) = \frac{jumlahkatatematikdalamkalimat}{panjangkalimat(jumlahkatapadakalimat)} \quad (9)$$

f. *Numerical data*

Adalah jumlah data numerik yang ada dalam kalimat, kedudukan kalimat yang mengandung data numerik adalah penting karena dimungkinkan akan masuk kedalam isi ringkasan dokumen[11]. Skor untuk fitur ini merupakan rasio dari jumlah kata numerik pada kalimat dengan panjang kalimat dalam dokumen.

$$Skor\ (S_i) = \frac{jumlahdatanumerik}{panjangkalimat\ (jumlahkatapadakalimat)}$$



Gambar 1. Kerangka Pemikiran

3. METODE PENELITIAN

3.1. Pengumpulan Data

Pengumpulan data awal dimulai dengan pengambilan dataset yang berasal dari data penelitian ilmiah. Dataset dalam penelitian ini berjumlah 20 dokumen Data penelitian ilmiah yang berasal dari situs jurnal.pdii.lipi.go.id.

3.2. Tahap *Preprocessing*

Preprocessing adalah suatu tahapan mengubah teks asli sebagai masukan dan menerapkan beberapa rutinitas dasar untuk mengubah atau menghilangkan unsure tekstual yang tidak berguna dalam pengolahan lebih lanjut [1].

a. Segmentasi Kalimat
   Segmentasi Kalimat adalah sebuah tahapan memisahkan sumber teks atau teks asli menjadi kalimat.
b. *Tokenization*
   Tokenization merupakan tahapan penguraian string teks menjadi *term* atau kata. Tujuan dari *Tokenization* yaitu memisahkan kata-kata dalam sebuah paragraf, kalimat atau halaman ke dalam kata tunggal.
c. *Stopword* Removal
   Stopword removal, merupakan tahapan penghapusan kata-kata yang tidak relevan dalam penentuan topik sebuah dokumen dan yang sering muncul pada dokumen, misalnya "dan", "atau", "sebuah","adalah", pada dokumen berBahasa Indonesia.
d. *Stemming*
   Stemming merupakan tahapan pengubahan suatu kata menjadi akar katanya dengan menghilangkan imbuhan awal atau akhir pada kata tersebut.
e. N-Gram
   N-gram adalah irisan N-karakter dari sebuah string12. Secara umum panjang dari sebuah string adalah k. Karakter kosong ("_") digunakan pada awal atau akhir dari sebuah n-gram. Sebagai contoh kata DATA mempunyai panjang k+1. Perbedaan untuk masing-masing n-gram terletak pada pembagian perkarakternya. Jika 2-gram , maka kata akan dibagi per 2 karakter, untuk 3-gram, maka kata akan dibagi per 3 karakter, untuk 4-gram, kata akan dibagi per 4 karakter. Contoh kata DATA memiliki *N-gram* sebagai berikut :
   2-gram : _D, DA, AT, TA, A_
   3-gram : _DA, DAT, ATA, TA_, A_ _
   4-gram : _DAT, DATA, ATA_, TA_ _, A_ _ _
f. *Term Weighting*
   Term Weighting adalah istilah frekuensi kejadian dalam suatu dokumen yang sering digunakan untuk melakukan perhitungan terhadap penting tidaknya suatu kalimat, skor kalimat dapat dihitung sebagai jumlah dari skor / nilai kata dalam kalimat tersebut [5].

3.3. Eksperimen dan Pengujian Model

Untuk pengujian prototipe dalam penelitian ini menggunakan *Lucene* sebagai *Java library*. *Lucene* menyediakan fungsi untuk *Stopword removal* dan *Stemming* untuk tahapan *preprocessing.Lucene* juga menyediakan perhitungan *F-Measure* untuk akurasi. Beberapa percobaan yang akan diujikan dalam penelitian ini yaitu dengan cara mengkombinasi tahapan *preprocessing* yaitu :

a. Proses *Tokenization >>Summarization*
b. Proses *Tokenization >> Stopword>>Summarization*
c. Proses *Tokenization >> Stemming>>Summarization*

d. Proses *Tokenization* >> *Stopword* >>*Stemming*>>*Summarization*
e. Proses *Tokenization* >> 2-gram >>*Summarization*
f. Proses *Tokenization* >> 3-gram >>*Summarization*
g. Proses *Tokenization* >> 4-gram >>*Summarization*
h. Proses *Tokenization* >> 5-gram>>*Summarization*
i. Proses *Tokenization* >> *Stopword* >>*Stemming* >> 2-gram>>*Summarization*
j. Proses *Tokenization* >> *Stopword* >>*Stemming* >> 3-gram >>*Summarization*
k. Proses *Tokenization* >> *Stopword* >>*Stemming* >> 4-gram >>*Summarization*
l. Proses *Tokenization* >> *Stopword* >>*Stemming* >> 5-gram>>*Summarization*
m. Proses *Tokenization* >> *Stemming* >> 3-gram>>*Summarization*
n. Proses *Tokenization* >> *Stopword* >> 3-gram >>*Summarization*

Pengujian Model dilakukan dengan mengamati tingkat hasil dari pengujian metode yang diusulkan yaitu kombinasi tahapan teks *preprocessing* pada peringkas dokumen teks be-Bahasa Indonesia.

3.4. Evaluasi dan Validasi Hasil

Dalam penelitian ini digunakan *F-Measure* Untuk mengukur kinerja dari *Summarization. F-Measure* diperoleh dengan pengukuran *recall* dan *precision. Recall* adalah rasio dokumen yang relevan yang terambil dengan jumlah seluruh dokumen dalam koleksi dokumen, sedangkan *precision* adalah rasio jumlah dokumen relevan terambil dengan seluruh jumlah dokumen terambil. Nilai interval *recall* dan *precision* berada antara 0 dan 121].

Recall adalah rasio dokumen yang relevan yang terambil dengan jumlah seluruh dokumen dalam koleksi dokumen, sedangkan *precision* adalah rasio jumlah dokumen relevan terambil dengan seluruh jumlah dokumen terambil. Untuk perhitungan *recall* (R), *precision* (P) dan *F-measure* (F) menggunakan persamaan berikut:

$$P = \frac{TP}{TP + FP} \qquad (11)$$

$$R = \frac{TP}{TP + FN} \qquad (12)$$

$$F = 2.\frac{Precision.Recall}{Precision+Recall} \qquad (13)$$

4. HASIL DAN PEMBAHASAN

4.1. Percobaan

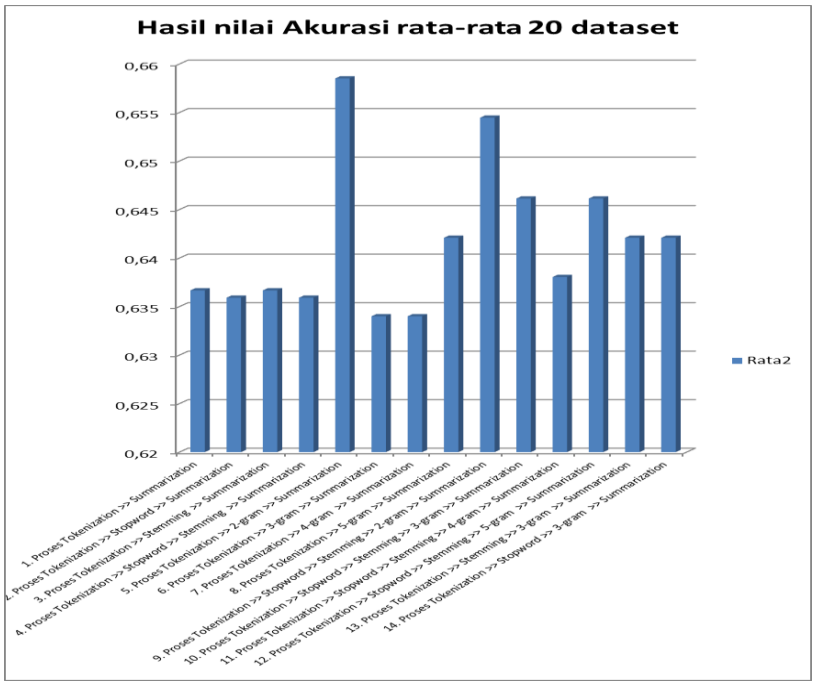
Dalam penelitian ini percobaan dilakukan dengan mengkombinasi tahapan teks *preprocessing* pada akurasi peringkas dokumen otomatis teks berbahasa Indonesia yaitu dimulai dari dataset yang sudah dikumpulkan sebanyak 20 dokumen berupa dataset penelitian ilmiah. Tahapan percobaan yang dilakukan yaitu *preprocessing, feature Summarization*, dan evaluasi performa dihitung dengan *F-measure (recall dan precision)*. *Preprocessing* yang akan dicobakan yaitu *segmentation, Tokenization, Stopword, Stemming* dan n-gram. N-gram yang akan diujicobakan yaitu 2-gram, 3-gram, 4-gram, 5-gram. Analisis Hasil Kombinasi Teks *Preprocessing* pada Peringkas Dokumen Otomatis Teks Berbahasa Indonesia.

4.2. Akurasi

Dari percobaan yang dilakukan dapat dibuktikan bahwa mengkombinasi tahapan teks *preprocessing* memberikan pengaruh pada nilai akurasi hasil.Dari 20 dataset yang digunakan, dilakukan 14 eksperimen hasil kombinasi tahapan teks *preprocessing.*

Tabel 1. Hasil Akurasi Rata-Rata dari Kombinasi Tahapan *Preprocessing*

Kombinasi Proses Tahapan *Preprocessing*	**RataAkurasi dari Dt1-Dt20 (F-Measure)**
1. Proses *Tokenization* >> *Summarization*	0,636667506
2. Proses *Tokenization* >> *Stopword* >> *Summarization*	0,63591312
3. Proses *Tokenization* >> *Stemming* >> *Summarization*	0,636667506
4. Proses *Tokenization* >> *Stopword* >> *Stemming* >> *Summarization*	0,63591312
5. Proses *Tokenization* >> 2-gram >> *Summarization*	0,658510643
6. Proses *Tokenization* >> 3-gram >> *Summarization*	0,634000839
7. Proses *Tokenization* >> 4-gram >> *Summarization*	0,634000839
8. Proses *Tokenization* >> 5-gram >> *Summarization*	0,642089074
9. Proses *Tokenization* >> *Stopword* >> *Stemming* >> 2-gram >> *Summarization*	0,654466525
10. Proses *Tokenization* >> *Stopword* >> *Stemming* >> 3-gram >> *Summarization*	0,646133192
11. Proses *Tokenization* >> *Stopword* >> *Stemming* >> 4-gram >> *Summarization*	0,638044957
12. Proses *Tokenization* >> *Stopword* >> *Stemming* >> 5-gram >> *Summarization*	0,646133192
13. Proses *Tokenization* >> *Stemming* >> 3-gram >> *Summarization*	0,642089074
14. Proses *Tokenization* >> *Stopword* >> 3-gram >> *Summarization*	0,642089074



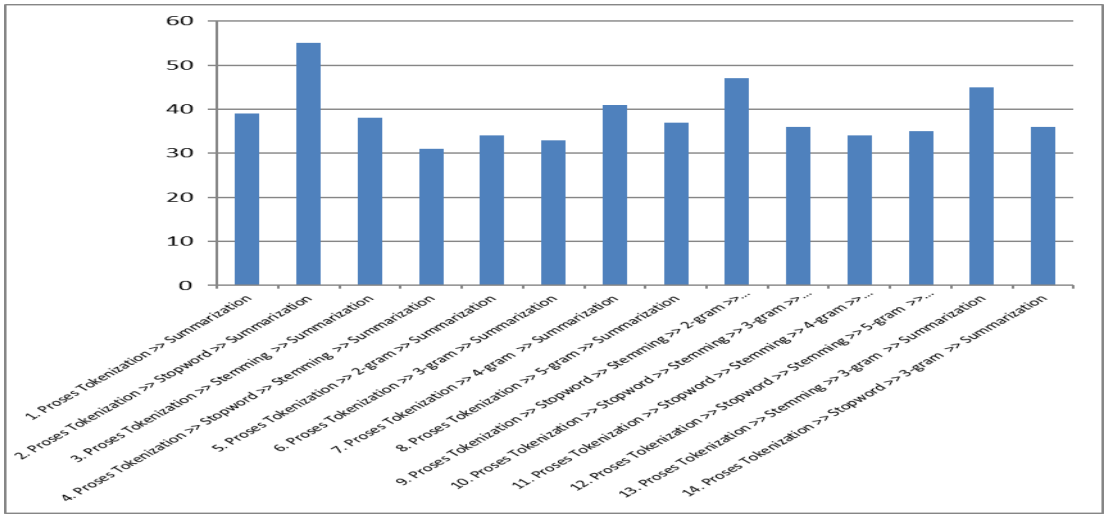
Gambar 2. Nilai Akurasi Rata-Rata untuk Masing-Masing Proses Kombinasi Teks *Preprocessing*

Dari gambar 3 menunjukkan bahwa nilai *F-Measure* dari 20 data set tersebut sebesar 66%, yang diperoleh dari proses kombinasi kelima yaitu *Tokenization*>> 2-gram >>*Summarization.*

Tabel 2. Hasil Waktu Rata-Rata dari Kombinasi Tahapan *Preprocessing*

Proses Summary	Waktu (seconds)
1. Proses *Tokenization* >> *Summarization*	39
2. Proses *Tokenization* >> *Stopword* >> *Summarization*	55
3. Proses *Tokenization* >> *Stemming* >> *Summarization*	38
4. Proses *Tokenization* >> *Stopword* >> *Stemming* >> *Summarization*	31
5. Proses *Tokenization* >> 2-gram >> *Summarization*	34
6. Proses *Tokenization* >> 3-gram >> *Summarization*	33
7. Proses *Tokenization* >> 4-gram >> *Summarization*	41
8. Proses *Tokenization* >> 5-gram >> *Summarization*	37
9. Proses *Tokenization* >> *Stopword* >> *Stemming* >> 2-gram >> *Summarization*	47
10. Proses *Tokenization* >> *Stopword* >> *Stemming* >> 3-gram >> *Summarization*	36
11. Proses *Tokenization* >> *Stopword* >> *Stemming* >> 4-gram >> *Summarization*	34
12. Proses *Tokenization* >> *Stopword* >> *Stemming* >> 5-gram >> *Summarization*	35
13. Proses *Tokenization* >> *Stemming* >> 3-gram >> *Summarization*	45
14. Proses *Tokenization* >> *Stopword* >> 3-gram >> *Summarization*	36

Grafik Waktu Proses *Summarization* dengan Kombinasi *Text Preprocessing* tercantum pada gambat berikut ini.



Gambar 3. Grafik Waktu Proses *Summarization* dengan Kombinasi *Text Preprocessing*

Gambar tersebut menunjukkan bahwa waktu tercepat dalam proses *Summarization* diperoleh pada proses kombinasi *Tokenization >> Stopword >> Stemming >> Summarization.*

Gambar 2 menunjukkan bahwa proses kombinasi tahapan *preprocessing* mempunyai pengaruh pada hasil summary, dibuktikan dengan F-Measure yang tidak sama pada masing-masing proses kombinasi *preprocessing*.Walaupun perbedaan nilainya tidak begitu jauh. Akan tetapi jika beberapa dokumen yang F-Measure nya hampir sama, sebenarnya kalau di cek dibandingkan isi dokumennya sebenarnya isinya berbeda. Gambar 2 juga menunjukkan bahwa hasil terbaik di dapat dari proses kombinasi yang ke 5 yaitu Proses *Tokenization*>> 2-gram >>*Summarization*, dimana hasil akurasi nya mencapai 66%. Sedangkan hasil kombinasi *preprocessing* yang menunjukkan nilai terendah pada kombinasi proses Proses *Tokenization*>> 3-gram >> *Summarization* dan *Tokenization*>> 4-gram >>*Summarization* yaitu 63%.

5. PENUTUP

Berdasarkan percobaan-percobaan yang telah dilakukan dapat disimpulkan bahwa kombinasi tahapan teks *preprocessing* memberikan pengaruh pada peringkas dokumen otomatis teks ber-Bahasa Indonesia. Pengaruh kombinasi tahapan teks *preprocessing* ini dapat dilihat pada tingkat *F-Measure* yang berbeda-beda pada masing-masing kombinasi tahapan teks *preprocessing*. Tingkat akurasi tertinggi dihasilkan pada percobaan kombinasi tahapan teks *preprocessing* untuk tahapan *Tokenization*>> 2-gram >>*Summarization*, hasil akurasi nya mencapai 66% atau 0,658510643. Sedangkan tingkat akurasi terendah dihasilkan pada kombinasi tahapan teks *preprocessing* untuk tahapan *Tokenization*>> 3-gram >>*Summarization* dan Proses *Tokenization*>> 4-gram >>*Summarization* yaitu 63% atau sebesar 0,634000839.

Dari sisi waktu komputasi juga terjadi optimasi waktu pada kombinasi dengan tingkat akurasi tertinggi, yaitu 34 *seconds*, yang meskipun bukan waktu terendah yang dicapai dibandingkan dengan waktu komputasi oleh kombinasi yang lain, tetapi tetap termasuk kelompok waktu paling cepat dengan waktu tercepat yang lainnya adalah 31 *seconds.* Hal ini membuktikan terjadinya reduksi *noise* seperti yang diprediksikan sebelumnya.

PERNYATAAN ORISINALITAS

“ Saya menyatakan dan bertanggung jawab dengan sebenarnya bahwa artikel ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya”

[**Hadiyatun Najjichah**]

DAFTAR PUSTAKA

[1] Gleb Sizov, "Extraction-Based Automatic *Summarization*," *Department of Computer and Information Science*, pp. 1-81, Juni 2010.

[2] Ladda Suanmali, "Automatic Text *Summarization* Using Feature Based fuzzy Extraction," *Jurnal Teknologi Maklumat*, pp. 105-115, Desember 2008.

[3] Makbule Gulcin Ozsoy, Ilyas Cicekli, and Ferda Nur Alpaslan, "Text *Summarization* of Turkish Texts using Latent Semantic Analysis," *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 869–876, Agustus 2010.

[4] Ladda Suanmali, Naomie Salim, and Salem Mohammed Binwahlan, "Automatic Text *Summarization* Using Feature-Based Fuzzy Extraction," *Jurnal Teknologi Maklumat*, Desember 2008.

[5] Ahmed Guven, O.Ozgur Bozkurt, and Oya Kalipsiz, "Advanced Information Extraction with N-Gram based LSI," *World Academy of Science, Engineering and Technology*, pp. 13-18, 2006.

[6] Vishal Gupta and Gurpreet S. Lehal, "A Survey of Text Mining Techniques and Applications," *JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE*, 2009.

[7] L. H Chong and Y. Y Chen , "Texs *Summarization* for Oil and Gas News Article," *World Academy of Science and Technology*, pp. 291-294, 2009.

[8] Makbule Gulcin Ozsoy, Cicekli Ilyas, and Ferda Nur Alpaslan, "Text *Summarization* of Turkish Texts using Latent Semantic Analysis," *Proceedings of the 23rd International Conference on Computational Linguistics* , pp. 869–876, Agustus 2010.

[9] Zdenek ceska and Chris Fox, "The Influence of Text Pre Processing on Plagiarism Detection," *International Conference RANLP*, pp. 55-59, 2009.

[10] Lady Agusta, "Perbandingan Algoritma *Stemming* Porter Dengan Algoritma Nazief & Adriani Untuk *Stemming* Dokumen Teks," *Konferensi Nasional Sistem dan Informatika*, pp. 196-201, November 2009.

[11] Ladda Suanmali, Naomie Salim, and Mohammed Salem, "Fuzzy Logic Based Method for Improving Text *Summarization*," *International Journal of Computer Science and Information Security*, vol. 2, 2009.