PERINGKASAN TEKS OTOMATIS PADA MULTI DOKUMEN MENGGUNAKAN TEXTRANK

Rivandi Restu Putra[1], Kania Evita Dewi, S.Pd., M.Si.[2]
[1,2] Program Studi Teknik Informatika
Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia
Jl. Dipatiukur No. 112-114 Bandung
E-mail : rivandirestu@gmail.com

ABSTRAK

Seiring berkembangnya teknologi, semakin banyak teknologi yang mempermudah pekerjaan manusia. Salah satunya adalah peringkasan teks otomatis yang mempermudah manusia untuk memahami sebuah teks tanpa membaca secara keseluruhan. Peringkasan teks otomatis terdapat kategori *single-document* dan *multi-document*. Peringkasan *multi-document* menjadi fokus dari penelitian ini.

Peringkasan teks otomatis pada multi dokumen merupakan peringkasan teks pada beberapa dokumen yang memiliki tema yang sama. Pada penelitian sebelumnya tentang peringkasan teks otomatis pada multi dokumen menggunakan RVM didapatkan bahwa hasil *f-measure* masih rendah sehingga perlu menggunakan metode lain untuk meningkatkan akurasinya.

Pada penelitian sebelumnya tentang peringkasan *single-document* dengan menggunakan *TextRank* diperoleh akurasi sebesar 95% dan baik dalam meringkas sebuah teks berita, sehingga pada penelitian ini akan diukur akurasi *TextRank* pada peringkasan multi dokumen. Dari hasil pengujian yang dilakukan, didapatkan hasil akurasi sebesar 74%, *recall* sebesar 73%, *precission* sebesar 74%, dan *f-measure* sebesar 73%. Sehingga dapat disimpulkan algoritma *TextRank* memiliki *f-measure* yang baik untuk digunakan dalam peringkasan teks otomatis pada multi dokumen.

Kata kunci : Peringkasan Teks Otomatis, Peringkasan Teks Otomatis Multi Dokumen, *TextRank*, Akurasi

1. PENDAHULUAN

Peringkasan teks otomatis (*automatic text summarization*) adalah pembuatan versi yang lebih singkat dari sebuah teks dengan memanfaatkan aplikasi yang dijalankan pada komputer. Hasil peringkasan ini mengandung poin-poin penting dari teks asli. Ringkasan didefinisikan sebagai sebuah teks yang dihasilkan dari satu atau lebih teks, mengandung informasi dari teks asli dan panjangnya tidak lebih dari setengah teks asli [1].

Pada penelitian peringkasan multi dokumen yang dilakukan oleh Tendi Arifin didapatkan bahwa hasil dari akurasi sebesar 67.35%, recall sebesar 47.83%, precission sebesar 35.48, dan f-measure sebesar 40.74% [2]. Setelah diperoleh f-measure, ternyata metode RVM masih kurang baik performansinya sehingga diperlukan metode yang lain untuk meningkatkan akurasi pada peringkasan multi dokumen.

Penelitian yang dilakukan oleh Yuzar Marsyah menyatakan bahwa hasil ringkasan algoritma *textrank* mengandung kalimat yang lebih relevan dengan hasil 95.56% [3]. Algoritma *TextRank* juga lebih cepat dalam mengeluarkan hasil ringkasan dengan 0.0294 detik dibandingkan algoritma LexRank dengan 0.4321 detik [3].

Berdasarkan penjelasan diatas, tugas akhir ini mengimplementasikan Algoritma *TextRank* dalam peringkasan teks otomatis pada multi dokumen menggunakan Bahasa Indonesia

1.1 Maksud dan Tujuan

Berdasarkan permasalahan yang telah diteliti, maka maksud dari penelitian kali ini adalah untuk mengimplementasi algoritma *textrank* dalam pembuatan peringkasan teks otomatis pada multi dokumen.

Sedangkan tujuan penelitian ini adalah untuk mengukur akurasi peringkasan teks otomatis pada multi dokumen dengan menggunakan algoritma *textrank*.

1.2 Algoritma *TextRank*

TextRank merupakan salah satu model perankingan berbasis graf untuk diterapkan dalam graf tekstual. Algoritma perankingan berbasis graf adalah cara untuk menentukan seberapa penting sebuah vertex dalam graf, berdasarkan informasi yang diambil dari struktur graf keseluruhan secara rekursif. Dalam penerapannya pada peringkasan teks otomatis, *TextRank* mengekstraksi kalimat-kalimat beranking tinggi untuk dimasukkan ke dalam ringkasan[3].

1.3 Text Preprocessing

Text Pre-Processing adalah suatu tahapan yang dilakukan untuk mempersiapkan teks sumber menjadi data yang akan diolah pada tahapan

selanjutnya. Jenis masukkan yang dibutuhkan pada proses ini adalah berupa teks dokumen, yang nantinya teks tersebut akan melalui beberapa tahapan utama yaitu:. Adapun tahapan pada *text processing*, diantaranya.

1. *Splitting*, teks berita yang utuh akan dipisah menjadi beberapa kalimat-kalimat.
2. Tokenizing kata, pemotongan kalimat menjadi kata-kata tunggal dengan (" ") spasi sebagai delimeter.
3. *Filtering*, menghilangkan tanda baca dan simbol.
4. *Case folding*, proses mengubah seluruh teks berita menjadi huruf kecil.
5. *Stopword Removal*, menghilangkan kata-kata umum yang sering muncul. Kata-kata ini dihilangkan karena tidak memiliki arti.

2 ISI PENELITIAN

2.1 Analisis Masalah

Analisis masalah adalah suatu gambaran masalah yang diangkat dalam penulisan penelitian tentang implementasi metode *TextRank* pada peringkasan teks otomatis berbahasa Indonesia. Pada penelitian peringkasan multi dokumen yang dilakukan oleh Tendi Arifin didapatkan bahwa hasil dari akurasi sebesar 67.35%, recall sebesar 47.83%, precission sebesar 35.48, dan f-measure sebesar 40.74% [2]. Setelah diperoleh f-measure, ternyata metode RVM masih kurang baik performansinya sehingga diperlukan metode yang lain untuk mengukur akurasi pada peringkasan multi dokumen.

2.2 Analisis Sistem Yang Akan Dibangun

Analisis sistem peringkasan teks otomatis ini dibangun dengan memiliki 3 tahapan utama yaitu:

1. Tahap pertama proses praproses yang merupakan tahap dimana dokumen yang akan digunakan dibersihkan dari noise atau dari hal yang tidak mempunyai pengaruh dalam ringkasan. Praproses yang dilakukan dalam tahap ini antara lain: *Splitting*, *tokenizing*, *filtering*, *Case Folding* dan *stopword removal.*

2. Tahap kedua adalah Algoritma *Textrank*. Pada tahap ini melakukan inisialisasi bobot awal kalimat dan melakukan perhitungan *Content Overlap Similarity* untuk menghitung bobot hubungan antar kalimat. Setelah itu masuk perhitungan *TextRank* untuk mendapatkan setiap bobot akhir dari setiap kalimat yang nantinya setiap bobot ini akan di ranking untuk mendapatkan hasil akhir dari ringkasan.
3. Tahap ketiga yaitu melakukan perhitungan TF-IDF dan Cosine Similarity untuk mengurutkan hasil ringkasan dari beberapa dokumen dan menggabungkannya menjadi 1 hasil ringkasan

2.3 Analisis Text Preprocessing

Diberikan sebuah dokumen berita yang akan di analisis oleh algoritma *textrank*. Contoh dokumen adalah sebagai berikut:

Timnas Islandia melanjutkan cerita ajaib mereka tahun ini. Setelah mencapai perempat final di Piala Eropa 2016 timnas Islandia untuk pertama kalinya masuk putaran final Piala Dunia 2018 di Russia. Lolosnya Islandia semakin luar biasa karena mereka lolos langsung tanpa perlu melalui fase play-off Piala Dunia 2018. Itu setelah Gylfi Sigurdsson dan kawan-kawan memuncaki Grup I dengan 22 poin dari 10 partai. Islandia sukses menyingkirkan Turki dan Ukraina yang notabene lebih berpengalaman. Mereka memaksa Kroasia harus melalui babak play-off sebelum memastikan diri lolos ke Piala Dunia 2018

Contoh dokumen tersebut akan di proses ketahapan selanjutnya.

1. Splitting, teks berita yang utuh akan dipisah menjadi beberapa kalimat-kalimat.

Tabel 1 Splitting

NO	Kalimat
S1	Timnas Islandia melanjutkan cerita ajaib mereka tahun ini.
S2	Setelah mencapai perempat final di Piala Eropa 2016, timnas Islandia untuk pertama kalinya masuk putaran final Piala Dunia 2018 di Russia.
S3	Lolosnya Islandia semakin luar biasa karena mereka lolos langsung tanpa perlu melalui fase play-off Piala Dunia 2018.
S4	Itu setelah Gylfi Sigurdsson dan kawan-kawan memuncaki Grup I dengan 22 poin dari 10 partai.
S5	Islandia sukses menyingkirkan Turki dan Ukraina yang notabene lebih berpengalaman.
S6	Mereka memaksa Kroasia harus melalui babak play-off sebelum memastikan diri lolos ke Piala Dunia 2018

2. Tokenizing kata, pemotongan kalimat menjadi kata-kata tunggal dengan (" ") spasi sebagai delimeter

Tabel 2 *Tokenizing*

No	Hasil Tokenizing		
S1	Timnas islandia melanjutkan	cerita ajaib mereka	tahun ini.
S2	Setelah mencapai perempat final di Piala Eropa	2016, timnas islandia untuk pertama kalinya masuk	putaran final piala dunia 2018 di Russia.
S3	Lolosnya Islandia semakin luar biasa karena mereka	lolos langsung tanpa perlu melalui fase play-off	Piala Dunia 2018.
S4	Itu setelah Gylfi Sigurdsson dan kawan-kawan	memuncaki Grup I dengan 22 Poin	dari 10 Partai.
S5	Islandia sukses menyingkirkan Turki dan	Ukraina yang notabene lebih berpengalaman.	
S6	Mereka memaksa Kroasia harus melalui babak	play-off sebelum memastikan diri lolos ke	Piala Dunia 2018.

3. Filtering, menghilangkan tanda baca dan simbol.

Tabel 3 *Filtering*

No	Hasil Filtering		
S1	Timnas islandia melanjutkan	cerita ajaib mereka	tahun ini
S2	Setelah mencapai perempat final di Piala Eropa	2016 timnas islandia untuk pertama kalinya masuk	putaran final piala dunia 2018 di Russia
S3	Lolosnya Islandia semakin luar biasa karena mereka	lolos langsung tanpa perlu melalui fase playoff	Piala Dunia 2018
S4	Itu setelah Gylfi Sigurdsson dan kawan-kawan	memuncaki Grup I dengan 22 poin	dari 10 Partai
S5	Islandia sukses menyingkirkan Turki dan	Ukraina yang notabene lebih berpengalaman.	
S6	Mereka memaksa Kroasia harus melalui babak	playoff sebelum memastikan diri lolos ke	Piala Dunia 2018

4. Case folding, proses mengubah seluruh teks berita menjadi huruf kecil.

Tabel 4 *Case Folding*

No	Hasil Case Folding		
S1	timnas islandia melanjutkan	cerita ajaib mereka	tahun ini
S2	setelah mencapai perempat final di piala eropa	2016 timnas islandia untuk pertama kalinya masuk	putaran final piala dunia 2018 di russia
S3	lolosnya islandia semakin luar biasa karena mereka	kolos kangsung tanpa perlu melalui fase playoff	piala dunia 2018
S4	itu setelah gylfi sigurdsson dan kawan-kawan	memuncaki grup i dengan 22 poin	dari 10 partai
S5	islandia sukses menyingkirkan turki dan	ukraina yang notabene lebih berpengalaman	
S6	mereka memaksa kroasia harus melalui babak	playoff sebelum memastikan diri lolos ke	piala dunia 2018

5. Stopword Removal, menghilangkan kata-kata umum yang sering muncul. Kata-kata ini dihilangkan karena tidak memiliki arti.

Tabel 5 *Stopword Removal*

No	*Hasil Stopword Removal*		Panjang Kalimat
S1	timnas islandia cerita	ajaib tahun	5
S2	mencapai perempat final piala eropa 2016 timnas islandia	kalinya masuk putaran final piala dunia 2018 russia	16
S3	lolosnya islandia lolos langsung fase playoff piala	dunia 2018	9
S4	gylfi sigurdsson kawan-kawan memuncaki grup i	22 poin 10 partai	10
S5	islandia sukses menyingkirkan turki ukraina	notabene berpengalaman	7
S6	memaksa kroasia babak playoff memastikan diri	lolos piala dunia 2018	10

2.4 Algoritma *Textrank*

TextRank merupakan salah satu model perankingan berbasis graf untuk diterapkan dalam graf tekstual. Algoritma perankingan berbasis graf adalah cara untuk menentukan seberapa penting sebuah vertex dalam graf, berdasarkan informasi yang diambil dari struktur graf keseluruhan secara rekursif. Dalam penerapannya pada peringkasan teks otomatis, *TextRank* mengekstraksi kalimat-kalimat beranking tinggi untuk dimasukkan ke dalam ringkasan.

Pada tahap awal dari algoritma *textrank* ini adalah menghitung kesamaan/similarity antar segmen atau antar kalimat dengan content overlap, dari pengambilan kalimat yang sama terhadap setiap segmen kalimat. Jika terdapat 2 kata yang sama dalam 1 kalimat, maka tetap dihitung 1. Berikut adalah rumus dari content overlap similarity:

$$Similarity\ (\mathrm{Si}, \mathrm{Sj}) = \frac{|\{\mathrm{Wk} \mid \mathrm{Wk} \in \mathrm{Si}\ \&\ \mathrm{Wk} \in \mathrm{Sj}\}|}{\log(|Si|) + \log(|\mathrm{Sj}|)}$$

Setelah itu, terapkan rumus similarity content overlap pada semua kalimat.Dibawah ini adalah contoh perhitungan *content overlap similarity* untuk kalimat 2 dan 3

Sim(S2,S3) = 4 / (log 16 + log 9) = 1,8532

Tabel 6 Similarity Conten Overlap

	S1	S2	S3	S4	S5	S6
S1	-	1.0509	0.6049	0	0.6476	0
S2	1.0509	-	1.3899	0	0.488	0.474
S3	0.6049	1.8532	-	0	0.5558	2.1536
S4	0	0	0	-	0	0
S5	0.6476	0.488	0.5558	0	-	0
S6	0	0.4746	2.1536	0	0	-

Setelah dilakukan pre-processing dan similarity, pada tahap ini dilakukan perhitungan bobot antar kalimat yang nantinya akan diranking untuk mendapatkan hasil ringkasan. Untuk menghitung bobot kalimat dengan *textrank* diperukan rumus:

$$\mathrm{WS}(\mathrm{S_i}) = (1 - \mathrm{d}) + \mathrm{d} * \sum_{\mathrm{Vj} \in \mathrm{Adj(Vi)}} \frac{\mathrm{W_{ji}}}{\sum_{\mathrm{Vk} \in \mathrm{Adj(Vj)}} \mathrm{W_{jk}}} \mathrm{WS}(\mathrm{S_j})$$

Dimana:

d merupakan dumping factor yang diisi dengan 0.85, W_{ji} merupakan similarity dari kalimat yang ingin dihitung, W_{jk}merupakan kalimat yang terhubung dengan W_j, S_j diisi dengan 1.

Rumus ini diterapkan pada semua kalimat untuk mendapatkan bobot setiap kalimat yang hasilnya akan di ranking.

Tabel 7 Ranking Kalimat

No	Bobot
S3	1.6962
S2	1.1877
S1	0.8472
S6	0.6577
S5	0.6113
S4	0.15

Jika ekstraksi kalimat dilakukan 25% dari teks asli, maka terdapat 2 kalimat yang masuk kedalam hasil ringkasan, yaitu kalimat 2 dan kalimat 3. Setelah mendapatkan hasil ringkasan pada dokumen tunggal, dilakukan proses ini sebanyak dokumen yang ada, lalu semua hasil ringkasan digabungkan.

2.5 TF-IDF

Setelah didapatkan hasil ringkasannya, kalimat yang akan masuk kedalam TF-IDF adalah kalimat yang

telah dilakukan proses preprocesing. Setelah itu dilakukan TF-IDF untuk mendapatkan hasil bobot kata dari hasil ringkasan. Semua hasil ringkasan digabung menjadi satu, pada tema ini terdapat 7 kalimat hasil ringkasan. Setelah itu menghitung kemunculan satu kata pada setiap kalimat

Sebagai contoh, kata “timnas”

1. Untuk mengisi kolom tf, hitung kemunculan kata “timnas” pada setiap kalimat. Kata “timnas” hanya muncul pada 1 kalimat saja, sehingga pada TF-R1 bernilai 1 dan TF-R lainnya diberi nilai 0
2. Selanjutnya pada kolom df, jumlahkan kemunculan kata “timnas” dalam semua kalimat hasil ringkasan. Karena timnas hanya muncul 1 kali pada kalimat 1, maka df bernilai 1
3. Mengisi kolom idf yaitu dengan menggunakan persamaan 2.3 dengan memasukkan nilai n dan df. Nilai n=7 dan df=1, sehingga idf kata “timnas” =log(7/1) = 0.8451
4. Mengisi kolom 2, yaitu menggunakan persamaan 2.4 dengan memasukkan nilai tf(proses no.1) dan idf(proses no.3). Sehingga mengisi W-R1= tf-S1.idf = 1x 0.8451= 0.8451 dan w selanjutnya.

2.6 Cosine Similarity

Setelah Memperoleh nilai tf-idf dan nilai yang dibutuhkan oleh cosine similarity selanjutnya menghitung cosine similarity dengan menggunakan rumus:

$$Sim(R1, Ri) = \frac{\Sigma(R1, Ri)}{\sqrt{R1^2}\sqrt{Ri^2}}$$

Contoh perhitungan untuk Similarity untuk R1 dan R2 adalah sebagai berikut:

$$\frac{0.60864}{\sqrt{10.011}\sqrt{3.702388}} = 0.099970915$$

Untuk perhitungan pada dokumen selanjutnya dapat menggunakan rumus yang sama. Setelah itu didapatkan hasil dari similarity setiap dokumen:

Tabel 8 Cosine Similarity

	R1	R2	R3	R4	R5	R6	R7
R1	0	0.099	0	0	0.012	0.046	0
R2	0.099	0	0.040	0	0.013	0.064	0
R3	0	0.040	0	0.049	0.121	0.036	0.016
R4	0	0	0.049	0	0	0	0
R5	0.012	0.013	0.121	0	0	0.050	0.048
R6	0.046	0.064	0.036	0	0.050	0	0.045
R7	0	0	0.01	0	0.048	0.045	0
TOTAL	0.158	0.217	0.263	0.049	0.244	0.243	0.110

Pada hasil dari tabel diatas dapat dilihat nilai dari setiap dokumen yang ada, setelah itu diurutkan dari besar hingga kecil dan diapatkan hasil ringkasan multi dokumen.
Berikut adalah pengurutan dari hasil ringkasan multi dokumen:

Tabel 9 Pengurutan Hasil Ringkasan

No	Kalimat	Nilai	Dokumen
1	R3	0.263992	2
2	R5	0.244686	3
3	R6	0.243739	4
4	R2	0.217598	1
5	R1	0.158503	1
6	R7	0.110609	4
7	R4	0.049438	2

2.7 Pengujian Fungsionalitas

Pengujian Fungsionalitas bertujuan untuk menampilkan fungsionalitas dari sistem agar dapat terlihat hasil yang sebenarnya pada sistem peringkasan teks dalam poin–poin yang akan diujikan. Setiap poin pengujian tersebut akan dijelaskan kembali pada hasil pengujian. Berikut adalah table dari pengujian fungsionalitas:

Tabel 10 Pengujian Fungsionalitas

No	Nama Proses	Poin Pengujian	Jenis Pengujian
1	Proses Pilih Tema Berita	Memilih tema berita	*Blackbox*
		Menampilkan teks yang akan diringkas	*Blackbox*
2	Preprocessing	Menampilkan kalimat	*Blackbox*
		Menampilkan token	*Blackbox*
3	Proses TextRank	Menampilkan proses TextRank	*Blackbox*
		Menampilkan hasil proses TextRank	*Blackbox*
4	Proses TF-IDF dan Cosine Similarity	Menampilkan proses TF-IDF dan Cosine Similarity	*Blackbox*

No	Nama Proses	Poin Pengujian	Jenis Pengujian
		Menampilkan hasil proses TF-IDF dan Cosine Similarity	*Blackbox*
		Menampilkan hasil peringkasan multi dokumen	*Blackbox*

2.8 Pengujian Akurasi

Setelah meranking setiap kalimat pada setiap dokumen, dilakukan pengujian akurasi dengan membandingkan hasil ringkasan manual dengan hasil ringkasan sistem. Pengujian akurasi merupakan ukuran yang menunjukan kedekatan hasil analisis dengan hasil sebenarnya. Pada pengujian akurasi ini, kualitas ringkasan ditentukan oleh *accuracy*. Nilai *accuracy* didapatkan melalui rumus berikut :

$$Accuracy = \frac{\sum Kalimat\ Sistem\ yang\ relevan\ dengan\ Hasil\ Manual}{\sum Kalimat\ Sistem\ yang\ muncul}$$

Untuk penerapan pada penelitian ini, jumlah kalimat sistem yang relevan dengan hasil manual adalah 57 dan jumlah kalimat sistem yang muncul adalah 77 sehingga:

$$Accuracy = \frac{57}{77} = 0{,}74$$

Persentase 0,74 * 100% = 74%

Sehingga hasil akurasi yang didapatkan adalah 74%

2.9 Pengujian Performansi

Pengujian performansi ini dilakukan dengan tujuan untuk mengetahui hasil dari penerapan metode *TextRank* dalam menghasilkan kalimat hasil ringkasan. Pada pengujian ini akan digunakan 24 dokumen berita dari ringkasan yang dihasilkan sistem kemudian akan dibandingkan dengan hasil ringkasan manual yang dihasilkan oleh ahli bahasa indonesia yang merupakan lulusan Sarjana Sastra Indonesia, mengingat studi kasus yang diambil di dalam penelitian ini yaitu dokumen teks berbahasa Indonesia[5].

Pengujian performansi yang dilakukan adalah dengan cara melakukan evaluasi hasil peringkasan dengan metode evaluasi instrinsik. Evaluasi ini mengukur nilai recall, precission dan f-measure. Nilai tersebut didapatkan dari rumus berikut:

$$\boldsymbol{recall} = \frac{hits}{hits + misses}$$

Dimana:
$hits$ = Jumlah kata pada kalimat yang berhasil diekstrak sistem yang sesuai dengan hasil ekstrak manusia
$misses$ = Jumlah kata pada kalimat yang diekstrak manusia tetapi tidak terdapat dalam hasil ekstrak sistem

$$Precission = \frac{hits}{hits + noice}$$

Dimana:
$hits$ = Jumlah kata pada kalimat yang berhasil diekstrak sistem yang sesuai dengan hasil ekstrak manusia
$noice$ = Jumlah kata pada kalimat yang diekstrak sistem tetapi tidak terdapat dalam kalimat yang diekstrak manusia.

$$\boldsymbol{f\text{-}measure} = \frac{2 * precission * recall}{precission + recall}$$

Dimana:
Precission = hasil dari perhitungan *precission*
Recall = hasil dari perhitungan *recall*

Rumus ini dihitung kepada 24 dokumen yang diuji. Maka didapatkan hasil sebagai berikut:

Tabel 11 Pengujian Performansi

Task Name	Avg_Recall	Avg_Precision	Avg_F-Score
DOKUMEN19	0.52	0.975	0.67826
DOKUMEN18	0.83582	0.83582	0.83582
DOKUMEN17	1	1	1
DOKUMEN16	1	1	1
DOKUMEN15	0.68657	0.66667	0.67647
DOKUMEN14	1	1	1
DOKUMEN13	0.72414	0.91304	0.80769
DOKUMEN24	0.70455	0.70455	0.70455
DOKUMEN12	0.68	0.59649	0.63551
DOKUMEN23	0.66667	0.70968	0.6875
DOKUMEN11	0.47059	0.35294	0.40336
DOKUMEN22	0.97917	0.97917	0.97917
DOKUMEN10	0.46939	0.62162	0.53488
DOKUMEN21	0.98	0.98	0.98
DOKUMEN20	0.97959	0.97959	0.97959
DOKUMEN4	0.01653	0.03333	0.0221
DOKUMEN3	0.67925	0.62069	0.64865
DOKUMEN2	0.575	0.60526	0.58974
DOKUMEN1	1	1	1
DOKUMEN8	0.62069	0.36	0.4557

DOKUMEN7	0.98507	0.98507	0.98507
DOKUMEN6	0.41176	0.38889	0.4
DOKUMEN5	1	1	1
DOKUMEN9	0.58491	0.49206	0.53448
Rata-Rata	0.732071	0.741661	0.730773

2.10Pengujian Evaluasi Kohesi

Evaluasi kohesi bertujuan untuk menghitung kepaduan antar kalimat. Jika nilai mendekati 0 maka kalimat ringkasan tersebut tidak padu, dan jika nilai mendekati 1 maka kalimat ringkasan tersebut sudah padu [6]. Berikut adalah hasil dari perhitungan evaluasi kohesi:

$$CoH = \frac{Log(C_S * 9+1)}{Log\ (M * 9+1)} \quad Ns = \frac{(o)*(o-1)}{2}$$

$$C_s = \frac{\Sigma_{\forall Si,Sj \epsilon Summary}\ Similarity\ (Si,Sj)}{N_S}$$

$$M = maxSimilarity\ (Si, Sj), Si, Sj\ \leq N$$

Pada penelitian ini rumus dihitung pada 5 tema, lalu didapatkan hasil sebagai berikut:

Tabel 12 Pengujian Evaluasi Kohesi

Tema	Nilai Kohesi
Kualifikasi Piala Dunia 2018	**0.102966662**
Transfer Hazard	**0.322349521**
Pemecatan Arsene Wenger	**0.24898729**
Kontrak Ozil	**0.326778014**
Transfer Carton Cole	**0.339610696**

Kesimpulan dari pengujian yang telah dilakukan menunjukkan bahwa pengujian fungsional sistem menggunakan metode black box untuk peringkasan teks otomatis pada multi dokumen yang dikembangkan saat ini berjalan dengan semestinya.

Setelah di ekstraksi 25% untuk mendapatkan hasil ringkasan, didapatkan beberapa pemilihan kalimat yang berbeda antara sistem dan manual, hal ini dikarenakan bobot yang didapatkan oleh sistem hanya berdasarkan jumlah kata yang sama antar kalimat, sedangkan dalam sebuah ringkasan dibutuhkan pemahaman makna kalimat sebagai aspek pembentukan hasil ringkasan. Hal ini merupakan kelemahan dari Algoritma *TextRank* yang tidak bisa mendapatkan makna dari suatu kalimat. Hasil akurasi untuk metode *TextRank* didapatkan nilai rata-rata recall 73%, precission 74%, dan f-measure 73% berdasarkan pengujian 24 dokumen teks berita berbahasa Indonesia

3 PENUTUP

Berdasarkan penelitian mengenai peringkasan teks otomatis pada multi dokumen menggunakan *TextRank*, didapat bahwa metode *TextRank* ini dapat diimplementasikan dalam kasus peringkasan teks multi dokumen. Hasil dari pengujian membuktikan nilai keakurasian yang dihasilkan oleh algoritma TextRank adalah akurasi sebesar 74%, recall sebesar 73%, precission sebesar 74%, dan f-measure sebesar 73% dengan jumlah berita sebanyak 24 dengan didapatkan 57 jumlah kalimat yang sama antara sistem dengan manual dari total 77 kalimat pada semua dokumen.

Berdasarkan hasil pengujian yang telah dilakukan, maka saran yang diharapkan setelah dilakukannya analisis peringkasan teks otomatis pada multi dokumen menggunakan *TextRank* adalah hasil pengujian antara sistem dengan manual, algoritma *TextRank* memiliki kelemahan untuk memahami makna dari kalimat. Maka dibutuhkan metode lain yang bisa memahami makna dari sebuah kalimat untuk mendapatkan hasil akurasi yang lebih baik

DAFTAR PUSTAKA

[1] N. Purwasih, "Sistem Peringkas Teks Otomatis untuk Dokumen Tunggal Berita Berbahasa Indonesia dengan Menggunakan Graph-based Summarization Algorithm dan Similarity," 2009

[2] T. Arifin, "Peringkasan Teks Otomatis Pada Multi Dokumen Menggunakan Relevan Vector Machine," 2017.

[3] Y. Marsyah, "Perbandingan Kinerja Algoritme TextRank dengan Algoritme LexRank Pada Peringkasan Dokumen Bahasa Indonesia," 2013.

[4] M. Muchammad, A. Zainal and K. Ririen, "Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance," maret 2012.

[5] N. P. Lestari, "Uji Recall And Precission Sistem Temu Kembali Informasi OPAC Perpustakaan ITS Surabaya," 2016

[6] R. S. W. Rizki Darmawan, "Hybrid Keyword Extraction Algorithm and Cosine Similarity for Improving Sentences Cohesion in Text Summarization," 2015.