PERINGKASAN MULTI DOKUMEN BERBAHASA INDONESIA MENGGUNAKAN METODE VIKOR

Freski Anshari .,[1], Nelly Indriani W.[2]

[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jalan Dipatiukur 112-116 Bandung
E-mail : freskianshari@gmail.com[1], nelly.indriani@email.unikom.ac.id[2]

ABSTRAK

Metode VIKOR merupakan metode pendukung keputusan yang menyeleksi lebih dari satu kriteria, berfokus pada perangkingan berdasarkan hasil alternatif dan kriteria yang bertentangan. Peringkasan teks otomatis pada multi dokumen adalah peringkasan dengan menggunakan banyak dokumen dengan tema yang sama menggunakan aplikasi yang dijalankan di perangkat komputer. Tahapan dari peringkasan teks otomatis yang akan dilakukan diantaranya adalah penerimaan input teks, preprocessing, mengekstrak beberapa fitur seperti : posisi kalimat, kemiripan dengan judul, kata kunci, bobot koneksi antar kalimat, kalimat yang mengandung kata numerik , frekuensi kata, dan sentralitas. Selanjutnya hasil ekstraksi fitur digunakan sebagai masukkan metode VIKOR. Setelah didapatkan rangking dari seluruh kalimat, selanjutnya memilah 25% kalimat berdasarkan nilai indeks VIKOR yang didapatkan. Kalimat yang telah dipilih akan diproses menggunakan metode TF-IDF dan akan disusun berdasarkan hasil dari cosine similarity. Pengujian intrinsik dilakukan dengan membandingkan ringkasan sistem dengan ringkasan manual hasil dari tiga orang pakar. Berdasarkan hasil pengujian intrinsik yaitu *ROUGE* dari 30 data uji yang dibagi kedalam 10 topik berita. Dapat disimpulkan bahwa peringkasan teks otomatis pada multi dokumen menggunakan metode VIKOR oleh pakar pertama menghasilkan tertinggi dengan rata – rata yang cukup baik yaitu recall sebesar 74,58%, precission 77,5% dan f-measure 75,25%.

Kata kunci : Metode VIKOR, Algoritma, Optimasi, MCDM, Peringkasan teks otomatis *multi* dokumen

1. PENDAHULUAN

Peringkasan teks otomatis (automatic text summarization) merupakan sajian singkat dari sebuah teks dokumen yang mempertahankan urutan isi dari bacaan dengan mengambil inti serta mempertahankan urutan wacana dari bacaan tersebut dengan memanfaatkan aplikasi yang dijalankan di perangkat computer [1]. Multi dokumen merupakan kumpulan dari beberapa dokumen teks yang memiliki kesamaan tema yang biasanya dikumpulkan dalam satu klaster dan kemudian akan menghasilkan keluaran berupa sebuah ringkasan teks dari informasi-informasi utama yang dikemas lebih singkat [2]. Ada berbagai cara yang dapat dilakukan untuk melakukan peringkasan otomatis pada teks salah satunya dengan menggunakan metode *VlseKriterijuska Optimizacija I Komoromisno Resenje* (VIKOR) yang merupakan salah satu metode *Multi Criteria Decision Making* (MCDM). Metode ini merupakan suatu metode pendukung keputusan dengan menyeleksi lebih dari satu kriteria yang berfokus pada perangkingan dan pemilihan berdasarkan dari hasil alternatif dan kriteria yang bertentangan [3].

Penelitian yang dilakukan oleh N.I Widiastuti tentang peringkasan teks otomatis menggunakan metode VIKOR, penelitian tersebut dikembangkan kembali oleh Novia Hadiana P. H. Dengan nilai rata rata yang didapatkan cukup baik, yaitu recall sebesar 74,10%, precission 67,35% dan f-measure 69,97% . Berdasarkan hasil tersebut membuktikan bahwa metode VIKOR cukup baik dalam melakukan peringkasan teks otomatis [4]. Untuk penelitian multi dokumen sebelumnya telah digunakan metode *Textrank*, metode ini menghasilkan hasil peringkasan yang cukup baik. Hal ini ditunjukkan dari nilai akurasi yang didapatkan sebesar 74%. Namun, metode Texrank memiliki kelemahan dalam penentuan tingkat kepentingan dari sebuah kalimat [5]. Pada penelitian multi dokumen lainnya yang menggunakan salah satu metode MCDM yaitu TOPSIS yang juga merupakan metode perangkingan sama seperti VIKOR menghasilkan nilai rata rata yang kurang baik yaitu recall sebesar 0.51, precission 0.52, dan f-measure 0,52 [6]. Namun, dalam cara menentukan perangkingan metode TOPSIS mengambil solusi terdekat dari solusi ideal dan solusi terjauh dari solusi negatif ideal tetapi tidak mempertimbangkan kepentingan jarak relatif dari keduanya. Karena nilai tertinggi dari alternatif dari TOPSIS tidak selalu menjadi solusi ideal terdekat sehingga mempengaruhi hasil dari perangkingannya [7]. Oleh karena itu, penggunaan metode VIKOR dapat mengatasi masalah tersebut. Karena nilai tertinggi dari metode VIKOR merupakan solusi ideal terdekat dalam penentuan hasil untuk pengambilan keputusan.

Pada beberapa penelitian dibidang lain menjelaskan bahwa metode VIKOR merupakan suatu metode yang tepat digunakan dalam sebuah

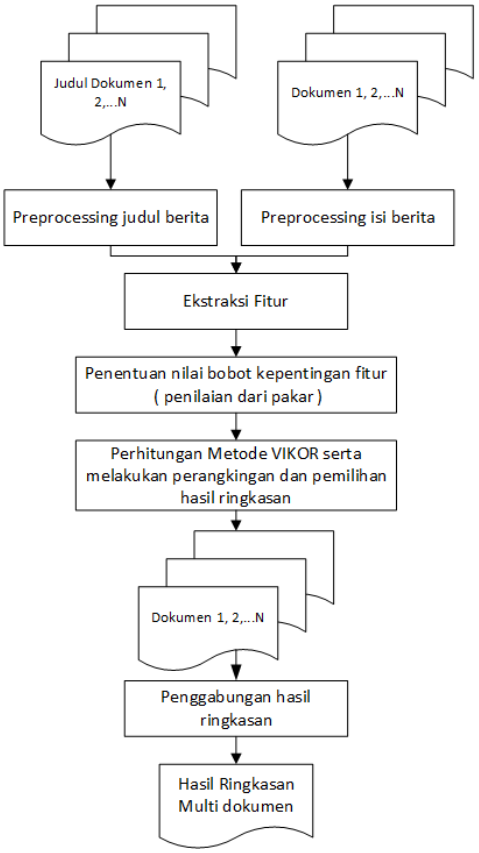
perangkingan karena hasilnya yang baik. Seperti pada penelitian [8]. Menyatakan bahwa metode VIKOR memiliki waktu perhitungan sistem kurang dari 1 detik dan penentuan nilai metode vikor dapat di atur sesuai nilai v nya. Pada penelitian lain menyatakan bahwa bahwa penerapan metode VIKOR dapat membantu menyelesaikan pengambilan keputusan secara praktis dan cepat [9].

Berdasarkan uraian tersebut,maka dalam penelitian ini bertujuan untuk melihat apakah metode VIKOR dapat digunakan untuk peringkasan teks otomatis pada multi dokumen berita berbahasa indonesia. Dalam penelitian ini diharapkan dapat menghasilkan peringkasan teks multi dokumen menggunakan metode VIKOR dengan baik.

2. ISI PENELITIAN

2.1 Analisis Sistem

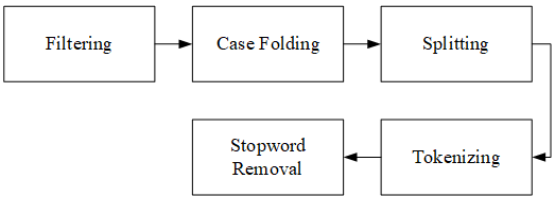
Sistem peringkasan teks otomatis yang akan dibangun pada penelitian ini dapat dilihat pada gambar 1. Dimulai dari penginputan beberapa dokumen masukan, *preprocessing* judul yang akan dipakai pada fitur kemiripan dengan judul. *Preprocessing* isi digunakan untuk pembobotan fitur, perhitungan fitur-fitur ekstraksi lainnya. Hasil dari proses tersebut akan dilakukan proses klasifikasi dan perangkingan menggunakan metode VIKOR sehingga menghasilkan beberapa hasil ringkasan untuk kemudian digabungkan dengan TF-IDF dan *cosine similarity* sehingga menghasilkan ringkasan multi dokumen.



Gambar 1. Analisis Sistem

2.2 Analisis *preprocessing*

Tahap preprocessing merupakan tahapan untuk merapikan dan mempersiapkan dokumen yang akan digunakan dalam proses perhitungan pada ekstraksi fitur. Preprocessing pada penelitian ini terdiri dari berbagai tahapan yang digambarkan pada Gambar 2.



Gambar 2. Analisis *preprocessing*

Tahapan *preprocessing* dilakukan pada data masukan. Dokumen data masukan melalui seluruh tahap dari tahapan *preprocessing*. Hasil dokumen yang telah melewati tahapan *preprocessing* dapat dilihat pada tabel 1.

Tabel 1 Dokumen Hasil *preprocessing*

No	*Splitting* kalimat
1	pyeongchang pemerintah korsel mengerahkan militer mengisi kekosongan pekerja akan bertugas olimpiade musim dingin pyeongchang
2	pasalnya pekerja terkena norovirus
3	virus menyebabkan seseorang mual muntah diare
4	sejauh 32 pekerja dirawat karena norovirus
5	mencegah menyebarnya virus sebanyak 1200 pekerja dikarantina
6	mengatasi kekurangan tenaga kerja keamanan diisolasi sekitar 900 personel militer ditugaskan mengambil alih pekerjaan keamanan sipil bunyi pernyataan panpel olimpiade pyeongchang rabu 7 februari 2018 dilansir insidethegames
7	mereka nantinya bekerja 20 tempat pertandigan pekerja dinyatakan pulih
8	mencegah infeksi norovirus lanjut kementerian keamanan pangan obat obatan menginspeksi kebersihan restoran fasilitas makanan terkait olimpiade
9	pusat pengendalian penyakit korsel memamtau pasien diare meminta orang mengikuti langkah pencegahan norovirus
10	pernyataan itu panpel juga memberikan saran terhadap masyarakat untuk mencuci tangan selama 30 detik dengan air yang mengalir dan sabun setelah pergi ke toilet.

2.3 Ekstraksi Fitur

Ekstraksi fitur adalah proses pengolahan kalimat menggunakan fitur-fitur sehingga menghasilkan nilai-nilai fitur yang akan digunakan untuk peringkasan dokumen. Proses ini dilakukan setelah melalui proses preprocessing. Fitur-fitur yang digunakan adalah posisi kalimat, kata kunci,

kemiripan dengan judul, bobot kemiripan antar kalimat, kalimat numerik, frekuensi kata, sentralitas.

2.3.1 Posisi Kalimat (F1)

Secara umum kalimat penting dalam sebuah berita terletak pada kalimat awal atau akhir [10]. Posisi kalimat dapat dihitung dengan rumus (1) sebagai berikut:

$$Posisi\ kalimat\ = Max\left(\frac{1}{i}, \frac{1}{n-i+1}\right) \quad (1)$$

Keterangan :
i = urutan kalimat pada dokumen
n = jumlah keseluruhan kalimat pada dokumen

2.3.2 Kata Kunci (F2)

Kata kunci diperiksa jumlah frekuensi kemunculannya pada kalimat. Kata kunci yang didapatkan akan diambil ≤ 10 kata. Bobot 0,1 akan diberikan jika kata kunci muncul pada kalimat [10]. Bobot 0 akan diberikan jika kata kunci tidak muncul. Kata kunci yang diambil dapat dilihat pada tabel 2.

Tabel 2 Daftar Kata Kunci

No	Kata	Jumlah
1	Pekerja	5
2	Norovirus	4
3	Pyeongchang	3
4	Olimpiade	3
5	Keamanan	3
6	Virus	2
7	Diare	2
8	Pernyataan	2
9	Militer	2
10	Mencegah	2

Dari daftar kunci yang didapatkan akan diperiksa kemunculannya pada setiap kalimat. Setiap kemunculan akan diberi nilai 0,1 dan jika tidak ada kemunculan akan diberi nilai 0. Hasil dari perhitungan kata kunci dapat dilihat pada tabel 3.

Tabel 3 Perhitungan kata kunci

Kalimat	$\sum total\ bobot$
S1	0,5
S2	0,2
S3	0,2
S4	0,2
S5	0,3
S6	0,6
S7	0,1
S8	0,4
S9	0,2
S10	0,1

2.3.3 Kemiripan Kalimat Dengan Judul (F3)

Judul dalam sebuah berita merupakan pemadatan dari sebuah berita dan menggambarkan isi dari berita tersebut [11]. Jika kalimat banyak memiliki kesamaan dengan judul maka kalimat tersebut memiliki tingkat kepentingan yang tinggi. Fitur ini dihitung dengan menggunakan persamaan 2 sebagai berikut:

$$Kemiripan\ = \frac{|Kata\ di\ judul \cap kata\ di\ Kalimat(i)|}{|Kata\ di\ judul \cup kata\ di\ Kalimat(i)|} \quad (2)$$

Judul yang telah melewati tahap *preprocessing* kemudian dilakukan pengecekan kecocokan dengan setiap kalimat. Perhitungan kemiripan dengan judul pada semua kalimat dapat dilihat pada tabel 4.

Tabel 4 Hasil Perhitungan fitur 3

Kalimat	Bobot kemiripan
S1	$score(S1) = \frac{5}{22} = 0{,}2273$
S2	$score(S2) = \frac{1}{13} = 0{,}0796$
S3	$score(S3) = \frac{0}{15} = 0$
S4	$score(S4) = \frac{1}{13} = 0{,}0455$
S5	$score(S5) = \frac{0}{17} = 0$
S6	$score(S6) = \frac{2}{36} = 0{,}1143$
S7	$score(S7) = \frac{0}{19} = 0$
S8	$score(S8) = \frac{2}{23} = 0{,}08696$
S9	$score(S9) = \frac{2}{21} = 0{,}09524$
S10	$score(S10) = \frac{0}{22} = 0$

2.3.4 Bobot Koneksi antar Kalimat (F4)

Fitur ini menghitung banyak kata yang sama antar kalimat satu dengan kalimat lain [11]. Perhitungan bobot kalimat dapat dilihat pada tabel 5.

Tabel 5 Perhitungan bobot koneksi antar kalimat

Kalimat	$\sum total$
S1	9
S2	7
S3	2
S4	7
S5	6
S6	7
S7	4
S8	7
S9	5
S10	2

2.3.5 Kalimat Numerik (F5)

Pada peringkasan teks khususnya teks berita haruslah mempertimbangkan kata yang mengandung unsur numerik. Karena kalimat yang mengandung numerik terdapat kalimat yang penting [11]. Kalimat numerik dihitung dengan rumus 3.

$$Scoref5\ (Si)\ \frac{banyak\ angka\ dalam\ kalimat(s)}{panjang(s)} \quad (3)$$

2.3.6 Frekuensi Kata (F6)

Kemunculan kata dicek keberadaannya pada setiap dokumen, karena kebermaknaan suatu kata dapat ditentukan dengan seberapa banyak kata tersebut muncul dalam setiap dokumen. Untuk itu digunakan sebuah metode yang dapat mengekstrak data frekuensi kata dalam dokumen yaitu tf-isf. Tf-isf merupakan sebuah metode statistical yang digunakan untuk mengevaluasi kebermaknaan kata dalam sebuah kalimat pada dokumen [10]. Perhitungan frekuensi kata dapat dilihat pada rumus 4 dan 5.

$$isf = log\left(\frac{N}{n_i}\right) (4)$$

$$frekuensi\ kata = Tf_i * isf\ (5)$$

Keterangan :
N = jumlah keseluruhan kalimat pada dokumen
ni =jumlah kata yang mengandung kata ke-i

Hasil perhitungan fitur frekuensi kata dapat dilihat pada tabel 6.

Tabel 6 Hasil Perhitungan Frekuensi Kata

Kalimat	TF-ISF
S1	10.6198
S2	2.6989
S3	4.3979
S4	2.6989
S5	4.6989
S6	23.7167
S7	4.3010
S8	12.3187
S9	10.7959
S10	12.3979

2.3.7 Sentralitas (F7)

Sentralitas dalam sebuah kalimat di dokumen merupakan kemiripan kata antara kalimat yang satu dengan yang lainnya. Bila sebuah kalimat memiliki sentralitas yang tinggi,maka kalimat tersebut memiliki kemungkinan tertinggi untuk dimasukan kedalam hasil ringkasan [10]. Sentralitas digunakan untuk mencari hubungan antar kalimat-kalimat yang ada di dokumen sampai ditemukan nilai sentralitas tertinggi yang dimiliki oleh sebuah kalimat. Nilai sentralitas dihitung dengan menggunakan rumus 6.

$$Sentralitas = \frac{|Kata\ di\ Kalimat(i) \cap kata\ di\ Kalimat(j)|}{|Kata\ di\ Kalimat(i) \cup kata\ di\ Kalimat(j)|} \quad (6)$$

Keterangan:
i = indeks kalimat ke-i
j = indeks kalimat ke-j

Hasil perhitungan sentralitas dapat dilihat pada tabel 7.

Tabel 7 Hasil Perhitungan Sentralitas

Kalimat	Nilai sentralitas maks
S1	0.07692
S2	0.25
S3	0.0909
S4	0.25
S5	0.1
S6	0.07692
S7	0.1111
S8	0.0556
S9	0.0625
S10	0.05128

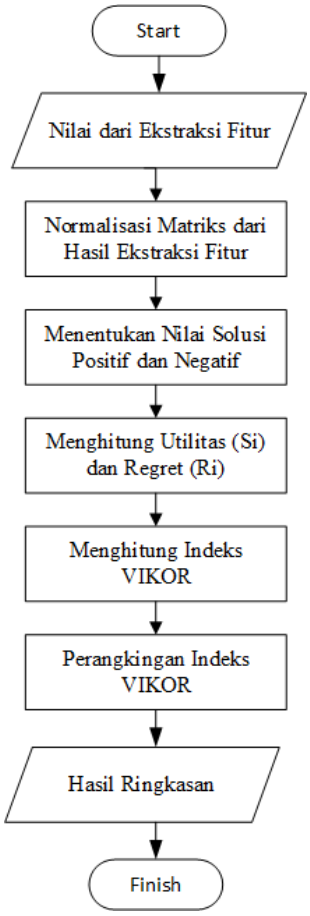
Setelah bobot semua fitur didapatkan maka nilai tersebut akan dipakai dalam perhitungan metode VIKOR. Hasil dari semua fitur dapat dilihat pada tabel 8.

Tabel 8 Hasil perhitungan semua Fitur

	F1	F2	F3	F4	F5	F6	F7
S1	1	0,5	0,273	9	0	10.619	0.077
S2	0,5	0,2	0,077	7	0	2.698	0.25
S3	0,333	0,2	0	2	0	4.397	0.091
S4	0,25	0,2	0,077	7	0.25	2.698	0.25
S5	0,2	0,3	0	6	0,167	4.699	0.1
S6	0,2	0,6	0,114	7	0,115	23.716	0.077
S7	0,25	0,1	0	4	0,2	4.301	0.111
S8	0,333	0,4	0,087	7	0	12.319	0.056
S9	0,5	0,2	0,095	5	0	10.796	0.062
S10	1	0,1	0	2	0,076	12.398	0.051

2.4 Metode VIKOR

VIKOR merupakan salah satu metode *multi criteria decision making* (MCDM) yang berfokus pada perangkingan dan pemilihan berdasarkan dari hasil alternatif dan kriteria yang bertentangan yang menghasilkan solusi yang mungkin paling dekat dengan solusi ideal [3].



Gambar 3 Metode VIKOR

2.4.1 Bobot Kepentingan Fitur

Perhitungan bobot fitur dilakukan sebagai data masukkan untuk fungsi subjektif metode VIKOR. Untuk bobot fitur sendiri ditentukan oleh pakar [3], dimana pakar dalam bidang peringkasan teks otomatis berbahasa Indonesia ini berasal dari latar belakang pendidikan sastra bahasa Indonesia yaitu Mahmud Fasya M.A, pakar 2 Dina Novalia S.S, dan pakar ke 3 Tri Ratna Nigrum S,Pd. Bobot fitur yang digunakan dalam penelitian ini diberikan bobot sebagai berikut :

Tabel 9 Bobot Fitur

No	Fitur	Pakar 1	Pakar2	Pakar 3
1	Posisi Kalimat	14,29%	15%	20%
2	Kata Penting/Kata kunci	21,43%	20%	20%
3	Kemiripan dengan Judul	25%,	15%	30%
4	Koneksi antar Kalimat	7,14%.	10%	5%
5	Kalimat yang mengandung kata numerik	3,57%	5%	5%
6	Frekuensi Kata/ TF-ISF	17,86%,	25%	10%
7	Sentralitas	10,71%	10%	10%

2.4.2 Normalisasi Matriks

Matriks keputusan yang telah didapatkan dari beberapa kriteria yang diambil dari perhitungan bobot ekstraksi fitur akan di normalisasi untuk menghasilkan matriks yang baru dari bobot fitur-fitur sebelumnya. Perhitungan normalisasi matriks menggunakan rumus 7.

$$f_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m}\left(x_{ij}^{2}\right)}}, i = 1, \dots, n; j = 1, \dots, m \quad (7)$$

Keterangan :

x_{ij} = elemen matriks dari kolom *i* baris ke *j*

n = jumlah kalimat

m = jumlah fitur

Hasil normalisasi matrik dapat dilihat pada tabel 10:

Tabel 10 Hasil Normalisasi Fitur

	F1	F2	F3	F4	F5	F6	F7
S1	0.0716	0.0358	0.0195	0.6443	0.0000	0.7603	0.0055
S2	0.0664	0.0266	0.0102	0.9301	0.0000	0.3586	0.0332
S3	0.0688	0.0413	0.0000	0.4126	0.0000	0.9072	0.0188
S4	0.0333	0.0266	0.0102	0.9311	0.0333	0.3590	0.0333
S5	0.0262	0.0393	0.0000	0.7862	0.0218	0.6157	0.0131
S6	0.0081	0.0243	0.0046	0.2830	0.0047	0.9588	0.0031
S7	0.0425	0.0170	0.0000	0.6798	0.0340	0.7309	0.0189
S8	0.0235	0.0282	0.0061	0.4937	0.0000	0.8688	0.0039
S9	0.0420	0.0168	0.0080	0.4198	0.0000	0.9064	0.0052
S10	0.0794	0.0079	0.0000	0.1587	0.0061	0.9841	0.0041
	F1	F2	F3	F4	F5	F6	F7

2.4.3 Menentukan Solusi Positif & Negatif Ideal

Solusi positif ideal adalah nilai terbesar dari masing-masing fitur ekstraksi. sedangkan solusi negatif ideal adalah nilai terkecil dari masing-masing fitur ekstraksi. Nilai maksimal dan minimal dari hasil normalisasi fitur dapat dilihat pada tabel 11.

Tabel 11 Nilai Solusi Ideal

Solusi ideal	F1	F2	F3	F4	F5	F6	F7
A^*	0.0794	0.0413	0.0195	0.9311	0.0340	0.9841	0.0333
A^-	0.0081	0.0079	0.0000	0.1587	0.0000	0.3586	0.0031

2.4.4 Menghitung Utilitas (S) dan Regret (R)

Dalam perhitungan ukuran utilitas dan regret diperlukan nilai bobot fitur (W) yang diberikan oleh semua pakar. Pada penelitian ini contoh perhitungan menggunakan bobot dari pakar 1. Perhitungan Utilitas (S) dan Regret (R) menggunakan rumus 8 dan 9.

$$S_i = \sum_{j=1}^{n} W_j \frac{(f_j^* - f_{ij})}{(f_j^* - f_j^-)} \quad (8)$$

$$R_i = max_j \left[W_j \frac{(f_j^* - f_{ij})}{(f_j^* - f_j^-)} \right] \quad (9)$$

Keterangan :

W = bobot dari fitur

f_j^* = nilai maksimal fitur dari normalisasi matriks

f_j^- = nilai minimal fitur dari normalisasi matriks

f_{ij} = nilai fitur dari hasil normalisasi matriks

Si = Nilai Utilitas

Ri = Nilai maksimal dari perhitungan Utilitas(Si)

Hasil perhitungan Utilitas dan Regret dapat dilihat pada tabel 12.

Tabel 12 Nilai Utilitas dan Regret

Kalimat	Utilitas (Si)	Regret (Ri)
S1	0.2754	0.0986
S2	0.4540	0.1786
S3	0.4284	0.2500
S4	0.4849	0.1785
S5	0.5721	0.2500
S6	0.6481	0.1908
S7	0.6266	0.2500
S8	0.5807	0.1715
S9	0.5846	0.1573

S10	0.6687	0.2500
Nilai Max	0.6687	0.2500
Nilai Min	0.2754	0.0986

2.4.5 Menghitung Indeks VIKOR

Indeks VIKOR dihitung dengan menggunakan bobot nilai (v) yang telah ditentukan. Adapun dalam sebuah jurnal menyatakan bahwa nilai (v) yang digunakan adalah yang $\leq$ 0,5 , jika kedua aspek dianggap relevan maka nilai (v) yang digunakan adalah 0,5 [12]. *Utilitas (Si)* dan *Regret (Ri)* akan disubtitusi kedalam persamaan. Perhitungan indeks VIKOR menggunakan persamaan 10.

$$Q_i = \left[v \frac{(S_i - S^-)}{(S^* - S^-)}\right] + \left[(1-v) \frac{(R_i - R^-)}{(R^* - R^-)}\right] (10)$$

Keterangan :
v = nilai bobot indeks vikor
S^*= nilai maksimal *utilitas*
S^-= nilai minimal *utilitas*
R^*= nilai maksimal *regret*
R^-= nilai minimal *regret*

Hasil perhitungan indeks VIKOR akan dirangking berdasarkan nilai terkecil ke yang tterbesar. Hasil perhitungan dan perangkingan indeks VIKOR dapat dilihat pada tabel 13.

Tabel 13 Nilai Indeks VIKOR

Kalimat	Qi	Rangking
S1	0	1
S2	0.4913	2
S4	0.5302	3
S9	0.5871	4
S8	0.6289	5
S3	0.6945	6
S6	0.7785	7
S5	0.8771	8
S7	0.9465	9
S10	1	10

2.4.6 Hasil Peringkasan

Dalam pengambilan kalimat yang akan digunakan untuk hasil ringkasan, maka digunakan kompresi 25% dari banyak kalimat yang ada didalam dokumen [5]. Hasil peringkasan dapat dilihat pada tabel 14.

Tabel 14 Hasil Peringkasan

Kalimat	Isi kalimat	Rangking
S1	pyeongchang pemerintah korsel mengerahkan militer mengisi kekosongan pekerja akan bertugas olimpiade musim dingin pyeongchang.	1
S2	pasalnya pekerja terkena norovirus.	2

Karena ini merupakan penelitian tentang *multi* dokumen maka keluaran dari hasil ringkasan merupakan hasil dari beberapa dokumen dengan topik yang sama. Semua dokumen melewati proses yang sama.

2.5 Penggabungan Hasil Ringkasan

Setelah hasil ringkasan dari beberapa dokumen telah didapatkan dengan menggunakan metode VIKOR maka keseluruh hasil ringkasan tersebut digabungkan dengan menghitung term kemunculan kata dengan TF-IDF dan menghitung kesamaan antar dokumen dengan *cosine similarity* untuk kemudian disusun dan menghaslilkan ringkasan multi dokumen.

2.5.1 Metode TF-IDF

Metode Tf-Idf adalah metode yang memeriksa dan memberikan bobot hubungan kemunculan kata (*term*) antara kalimat satu dengan kalimat yang lainnya. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu *term frequency* (tf) merupakan frekuensi kemunculan term j pada kalimat i. *Document frequency* (df) adalah banyaknya kalimat dimana suatu term j muncul [13]. Secara proses, metode ini sama dengan metode IF-ISF. TF-IDF dapat dihitung dengan rumus 11 dan 12.

$$idf = log\left(\frac{N}{idf}\right) (11)$$

$$Wij = Tf_{ij} * idf (12)$$

Keterangan :
N : jumlah dari seluruh kalimat
Idf : nilai kemunculan kata
W_{ij} : bobot kata dari kalimat
Tf_{ij} : jumlah kemunculan kata dalam kalimat

2.5.2 Cosine Similarity

Cosine similarity merupakan metode untuk mengukur jarak antara dokumen atau kalimat dengan pendekatan relevansi *query*. Semakin besar nilai kesamaan vektor query dengan vektor kalimat maka query tersebut dipandang semakin relevan dengan kalimat [10]. Nilai yang dipakai dalam perhitungan *cosine similarity* adalah nilai perhitungan TF-IDF. Perhitungan *cosine similarity* menggunakan rumus 13.

$$Sim\ (Si, Sj) = \frac{\sum_{j=1}^{m} W_{ij} * W_{ij}}{\sqrt{\sum_{j=1}^{m}(W_{ij}{}^2)} * \sqrt{\sum_{j=1}^{m}(W_{ij}{}^2)}} (13)$$

Keterangan
Si, Sj : similarity antara kalimat Si dengan kalimat Sj
W_{ij} : bobot kata dari kalimat pada daftar kalimat Si dan Sj

Hasil perhitungan nilai *cosine similarity* dapat dilihat pada tabel 15.

Tabel 15 Nilai Cosine similarity

	S1	S2	S3	S4	S5	S6	S7	S8
S1	-	0.092	0.029	0.059	0.014	0	0.233	0
S2	0.092	-	0	0	0	0	0	0

	S1	S2	S3	S4	S5	S6	S7	S8
S3	0.029	0	-	0.148	0.068	0	0.053	0
S4	0.059	0	0.148	-	0.138	0	0.143	0
S5	0.014	0	0.068	0.138	-	0	0.049	0.107
S6	0	0	0	0	0	-	0	0
S7	0.233	0	0.053	0.143	0.049	0	-	0
S8	0	0	0	0	0.107	0	0	-
Total	0.426	0.092	0.298	0.488	0.376	0	0.478	0.107

Hasil dari nilai *cosine similarity* kemudian diurutkan berdasarkan nilai terbesar ke yang terkecil. Karena semakin besar nilai *similarity* maka kalimat tersebut merupakan kalimat penting. Hasil pengurutan dapat dilihat pada tabel 16.

Tabel 16 Hasil Pengurutan

Dokumen	Kalimat	Nilai	Urutan
1	S1	0.2492	1
4	S7	0.2476	2
2	S3	0.2381	3
3	S6	0.1958	4
1	S2	0.1005	5
2	S4	0.0914	6
4	S8	0	7
3	S5	0	8

3. PEMBAHASAN

3.1 Pengujian Akurasi

Pengujian akurasi yang akan dilakukan adalah dengan cara melakukan evaluasi hasil peringkasan dengan menggunakan metode evaluasi intrinsik. Salah satunya dengan menggunakan ROUGE [14]. Perhitungan akurasi dengan metode ini dengan cara melihat hasil peringkasan dari sistem dan hasil peringkasan manual dari manusia (pakar). Dokumen yang digunakan berjumlah 30 dokumen berita yang terbagi dalam 10 topik. Evaluasi ini mengukur nilai *recall* dan *precission* kalimat dan kombinasi antara keduanya yang menghasilkan f-measure. Nilai – nilai tersebut didapatkan melalui rumus 14, 15, dan 16.

$$recall = \frac{hits}{hits+misses} \quad (14)$$

$$precission = \frac{hits}{hits+noice} \quad (15)$$

$$f-measure = \frac{2*precission*recall}{recall+precission} \quad (16)$$

Keterangan

$hits$= Jumlah kata pada kalimat yang berhasil diekstrak sistem yang sesuai dengan hasil ekstrak manusia (pakar)

$misses$= Jumlah kata pada kalimat yang diekstrak manusia (pakar) tetapi tidak terdapat dalam hasil ekstrak sistem

$noice$ = Jumlah kata pada kalimat yang diekstrak sistem tetapi tidak terdapat dalam kalimat yang diekstrak manusia

Rumus ini diterapkan pada 30 dokumen berita pada masing-masing pakar. Hasil perhitungan akurasi dapat dilihat pada tabel 17.

Tabel 17 Hasil rata-rata Akurasi ROUGE 30 data

Pakar	Rata-rata Recall	Rata-rata Precission	Rata-rata F-measure
1	0.7458	0.7750	0.7525
2	0.6317	0.6702	0.6439
3	0.6934	0.7822	0.7272

3.2 Pengujian Kohesi

Evaluasi kohesi bertujuan untuk menghitung kepaduan antar kalimat. Evaluasi kohesi dihitung berdasarkan nilai *cosine similarity* yang didapatkan dari kalimat-kalimat yang dihasilkan oleh sistem berdasarkan bobot dari masing-masing pakar. Jika nilai mendekati 0 maka kalimat ringkasan tersebut tidak padu, dan jika nilai mendekati 1 maka kalimat ringkasan itu sudah padu [15]. Pengujian kohesi dapat diitung menggunakan rumus 17.

$$CoH = \frac{\log(C_s * 9 + 1)}{\log(M * 9 + 1)} \quad (17)$$

$$N_s = \frac{(o) * (o-1)}{2}$$

$$C_s = \frac{\sum \forall Si, Sj \in summary\ Sim_{cos}(Si, Sj)}{N_s}$$

$$M = maxSim_{cos}(i, j), i, j \leq N$$

Keterangan :

CoH : nilai kohesi dari hasil ringkasan

C_s : rata-rata nilai similarity dari semua kalimat hasil ringkasan

$Sim_{cos}(Si, Sj)$: cosine similarity antara kalimat Si dan Sj

N_s : nilai similarity yang tidak bernilai 0

o : jumlah kalimat hasil ringkasan

N : jumlah kalimat di dokumen

M : nilai maksimum similarity kalimat dalam dokumen

Hasil pengujian kohesi dapat dilihat pada tabel 18.

Tabel 18 Pengujian Kohesi

Topik	Pakar 1	Pakar 2	Pakar 3
1	0.3246	0.1970	0.2619
2	0.2875	0.2948	0.2481
3	0.2018	0.1766	0.2339
4	0.3184	0.3305	0.2968
5	0.3563	0.3479	0.3563

Topik	Pakar 1	Pakar 2	Pakar 3
6	0.1704	0.2070	0.1728
7	0.2951	0.2271	0.3487
8	0.2108	0.2355	0.3085
9	0.2482	0.3754	0.2849
10	0.3557	0.2726	0.3557
Nilai Min	0.1704	0.1766	0.1728
Nilai Max	0.3563	0.3479	0.3563

4. PENUTUP

4.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, penerapan metode VIKOR untuk peringkasan teks *multi* dokumen sudah cukup baik. Pernyataan tersebut didukung oleh hasil rata – rata pengujian akurasi untuk 30 data uji yang dibagi menjadi 10 topik berita menggunakan metode ROUGE. Pengujian itu dilakukan dengan membandingkan hasil ringkasan sistem dan ringkasan manual yang didapatkan dari tiga orang pakar. Pakar 1 menghasilkan nilai nilai rata – rata tertinggi yaitu recall sebesar 74.58%, precission 77.50% dan f-measure 75,25%.

4.2 Saran

Saran dari penulis untuk penilitian ini untuk pengembangan penelitian peringkasan teks otomatis *multi* dokumen kedepannya yaitu :

1. Walaupun hasil yang didapatkan cukup baik, tapi pada penelitian ini belum mencapai nilai akurasi yang tinggi. Diperlukan penelitian lebih lanjut dengan menggunakan metode lain maupun pengembangan metode VIKOR seperti *fuzzy* VIKOR dll.
2. Penentuan kalimat yang akan dijadikan ringkasan masih kurang dalam perhitungan keterkaitan antar kalimatnya. Hal ini menyebabkan kalimat ringkasan multi dokumen yang dihasilkan menjadi tidak padu karena kurangnya keterkaitan antar kalimat hasil ringkasan dokumen satu dengan yang lain. Diharapkan untuk pengujian dengan menggunakan pakar lebih diperhitungkan lagi keterkaitan antar kalimatnya.

DAFTAR PUSTAKA

[1] Mustaqhfiri, M. Abidin, Z. Kusumawati, R., "Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance," pp. 134-147, 2012.

[2] Hovy, E. Lin, C.Y., "Automated text summarization and the summarist system," pp. 197-214, 1998.

[3] Widiastuti, N.I., "Rancangan dasar : VIKOR untuk Peringkasan Otomatis," pp. 15-22, 2017.

[4] Pahlawan, N.H., "Metode Vikor untuk Peringkasan teks Berbahasa Indonesia," pp. 1-10, 2018.

[5] Putra, R.R, "Peringkasan Teks Otomatis Pada Multi Dokumen Menggunakan Textrank," pp. 1-8, 2018.

[6] Malallah, S. Ali, Z.H., "TOPSIS with Multiple Linear Regression for Multi-Document Text Summarization," *Iraqi Journal of Science,* vol. 58, pp. 1298-1307, 2017.

[7] Opricovic, S. Tzeng, G.H., "Compromise Solution by MCDM methods : A comparative analys of VIKOR and TOPSIS," *Decision Aiding,* vol. 156, pp. 445-455, 2004.

[8] Pramulanto, E.C. Imrona, Drs.M. Darwiyanto, E.ST.,MT, "Aplikasi Pendukung Keputusan untuk Pemilihan Produk Asuransi dengan Metode Entropy dan Vikor pada AJB Bumiputera 1912 Jepara," dalam *e-proceeding of Engineering*, Bandung, 2015.

[9] Siregar, A. Ginting, P. Mesran. Sianturi, L.T., "Implementasi Metode VIKOR dalam Pemilihan Supplier Bahan Baku," vol. I, pp. 132-138, 2017.

[10] K, Nandhini. S.R, Balasundaran, "Extracting easy to understand summary using differential evolutional algorithm," *Swarm end Evolutionary Computation,* vol. 16, pp. 19-27, 2014.

[11] Aristoteles, "Penerapan Algoritma Genetika Pada Peringkasan Teks Dokumen Bahasa Indonesia," dalam *Prosiding Semirata FMIPA Universitas Lampung*, Lampung, 2013.

[12] Opricovic, S. Tzeng, G.H., "Extended VIKOR method in comparison with outranking methods," *Decision Support,* vol. 178, pp. 514-529, 2006.

[13] Ramos, J, "Using TF-IDF to Determine Word Relevance in Document Queries," dalam *Proceeding of the first Instructional Conference on Machine Learning*, 2003.

[14] David, M.W., "Evaluation : from Precission, Recall and F-measure to ROC, informedness, markedness & correlation," *Journal of Machine Learning Technologies,* vol. 2, no. 1, pp. 37-63, 2011.

[15] Darmawan, R., "Hybrid Keyword Extraction Algorithm and Cosine Similarity for Improving Sentences Cohesion in Text Summarization," *Journal of Intellegent System,* vol. 1, pp. 109-114, 2015.