PERINGKASAN TEKS OTOMATIS PADA PORTAL BERITA OLAHRAGA MENGGUNAKAN METODE MAXIMUM MARGINAL RELEVANCE

Dimas Firman AL-Hafiidh, Imam Fahrur Rozi, Ika Kusumaning Putri.

[1,2] Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Politeknik Negeri Malang,
[1]alamat e-mail : dimazfhirman@gmail.com, [2] alamat e-mail : imam.rozi@polinema.ac.id,
[3] alamat e-mail : ikakputri@polinema.ac.id.

Abstrak

Berita mengandung fakta atau opini yang mungkin memiliki kecenderungan tertentu. Berita dapat mengakses berbagai media seperti koran, televisi, internet, dan lain-lain. Internet adalah yang terkenal digunakan untuk mengakses berita. Untuk mencari informasi utama tentang berita mungkin membutuhkan waktu. Ini membutuhkan akses yang rapi dan meminimalkan waktu untuk membaca. Oleh karena itu perlu dilakukan rangkuman berita agar perolehan dari berita tersebut lebih efisien dan efektif. Penelitian dimulai dengan lima tahap preprocessing teks: pemecahan kalimat, pelipatan kasus, tokenizing, filtering, dan stemming. Proses selanjutnya adalah menghitung bobot tf-idf-df, bobot relevansi kueri dan bobot kesamaan. Ringkasan dihasilkan dari ekstraksi kalimat menggunakan metode relevansi marginal maksimum. Metode ini digunakan untuk mengurangi redundansi dalam memeringkat kalimat pada banyak dokumen. Pengaruh Maximum Marginal Relevance terhadap hasil akurasi ringkasan sistem yaitu pengujian diambil dari 5 sampel berita berita online menggunakan lamda 0.7 yang kemudian hasil ringkasan tersebut digunakan untuk dibandingkan dengan ringkasan sistem dan ringkasan oleh ahli termasuk. Maya Rahma, setelah itu dicari benar kemudian dicari salah dan luput, bila hasil tersebut diperoleh maka dicari ketelitian, recall, f-measure masing-masing responden dan dicari hasil tes dari mencari rata-rata presisi, recall, f-measure, sehingga efek lamda 0,7 menghasilkan akurasi rata-rata 57,7% Precision. , Ingat 48,5% dan F - Ukur 50,3%.

Kata kunci : Sistem Informasi, Peringkasan teks otomatis pada portal berita olahraga, *maximum marginal relevance tf-idf.*

1. Pendahuluan

Berita merupakan fakta ataupun opini yang membuat banyak orang merasa tertarik untuk mengetahuinya. Berita dapat diperoleh dengan berbagai media seperti koran, surat kabar, televisi, internet dan lain-lain. Pada saat ini, media yang paling sering digunakan untuk memperoleh berita adalah internet. Berita yang ada di internet memiliki berbagai macam topik, salah satunya yaitu berita olahraga.

Berita olahraga merupakan salah satu berita yang memiliki rating yang tinggi. Berita olahraga memiliki berbagai cabang seperti bola, raket, balap dan lain-lain. Situs berita olahraga seperti sport.detik.com, sport.okezone, dan sport.sindonews masuk ke dalam *25 top site* di Indonesia (Alexa.com). Sehingga berita olahraga termasuk berita favorit bagi warga Indonesia. Dalam penyajian berita, hampir semua situs memiliki cara yang sama, dengan penambahan sedikit informasi. Hal ini menyebabkan waktu yang diperlukan untuk mendapatkan berita yang sama lebih banyak. Banyak waktu diperlukan untuk menemukan informasi utama dari berita-berita tersebut. Oleh karena itu diperlukan peringkasan kumpulan berita ini agar perolehan informasi dari berita lebih efisien dan efektif (Rofiqi , 2017).

Peringkasan teks adalah suatu proses penyuntingan sebagian besar informasi penting dari sumber teks. Peringkasan teks khususnya berita olahraga dapat digunakan untuk mengambil isi yang paling penting dari sumber informasi yang kemudian menyajikannya kembali dalam bentuk yang lebih ringkas dari sebuah berita tanpa harus membaca keseluruan dari berita. Namun, proses kegiatan

peringkasan ini masih dilakukan secara manual oleh manusia. Sehingga akan membutuhkan waktu yang lama. Oleh karena itu diperlukan peringkasan teks secara otomatis.

Peringkasan teks otomatis adalah mengambil isi yang paling penting dari sumber informasi yang kemudian menyajikannya kembali dalam bentuk yang lebih ringkas dengan menjaga konten informasinya (Setiadi, Djamal, & Ilsyas, 2018). Melihat hal tersebut maka dibangun aplikasi Berita yang menerapkan peringkasan teks otomatis dengan metode *Maximum Marginal Relevance (MMR)*.

Algoritma Maximum Marginal Relevance (MMR) merupakan metode ekstraksi ringkasan yang digunakan untuk meringkas dokumen tunggal maupun multi dokumen (Setiawan & A, 2016). MMR meringkas dokumen dengan menghitung kesamaan (similarity) antara kalimat teks. Pada peringkasan dokumen dengan metode MMR dilakukan proses segmentasi dokumen menjadi kalimat dan dilakukan pengelompokan sesuai dengan jenis kalimat tersebut. untuk merangking kalimat-kalimat sebagai tanggapan pada query yang diberikan oleh user.

Dengan adanya hal tersebut Aplikasi peringkasan teks otomatis merupakan teknologi yang menawarkan solusi untuk mencari informasi dengan menghasilkan ringkasan (summary) berita olaharaga tersebut. Maka dibangun aplikasi Portal Berita Olahraga yang menerapkan peringkasan teks otomatis dengan metode *Maximum Marginal Relevance*. Dengan adanya aplikasi tersebut diharapkan dapat membantu pembaca dalam mendapatkan informasi dengan cepat.

2. TINJAUAN PUSTAKA

2.1 Peringksan Teks Otomatis

Peringkasan teks otomatis adalah mengambil isi yang paling penting dari sumber informasi yang kemudian menyajikannya kembali dalam bentuk yang lebih ringkas dengan menjaga konten informasinya (Setiadi, Djamal, & Ilsyas, 2018).

2.2 Maximum Marginal Relevance (MMR

Maximum Marginal Relevance (MMR) adalah salah satu dari sekian metode ekstraksi teks yang dapat diterapkan untuk meringkas dokumen tunggal maupun multi dokumen dengan cara melakukan rangking ulang dan membandingkan similarity antar dokumen. Menurut (B & Yaman, 2010), Metode MMR sering digunakan untuk peringkasan teks karena metode MMR sederhana dan efisien (Y & Liu, 2008). Jika kesamaan (similarity) antara satu kalimat dengan kalimat yang lain tinggi, maka kemungkinan terjadi redundansi. Rumus untuk menghitung nilai MMR yang dapat mengurangi redundansi adalah :

$$\mathrm{MMR}(s_i) = \lambda . sim_1(s_i) - (1-\lambda) . max\ sim_2(s_i, s_j$$

Sim adalah kosinus kesamaan antara dua vektor fitur. λ adalah kofisien untuk mengatur relevansi kalimat dan mengurangi redundansi. Nilai parameter λ adalah 1 atau 0 atau diantaranya ($0 < \lambda < 1$). Pada saat parameter $\lambda = 1$ maka nilai MMR yang diperoleh cenderung relevan terhadap dokumen asli. Ketika $\lambda = 0$ maka nilai MMR yang diperoleh akan cenderung releven terhadap kalimat yang diekstrak sebelumnya yang akan dibandingkan. Oleh sebab itu, sebuah kombinasi linear dari kedua kriteria dioptimalkan ketika nilai λ terdapat pada interval $0 < \lambda < 1$. Untuk peringkasan dengan dokumen yang kecil, seperti artikel berita akan menghasilkan hasil ringkasan yang baik, jika nilai parameter $\lambda = 0{,}7$ atau $\lambda = 0{,}8$ (Goldstein, 2008).

Penggunaan rumus MMR dalam perangkingan ulang adalah untuk mendapatkan ringkasan dengan similarity query kalimat tinggi, sedangkan similarity antara kalimat rendah. Pada rumus dibawah ini merupakan rumus yang memperhitungkan relevansi kalimat dengan query. Jadi, rumus tersebut merupakan benih untuk menentukan kalimat yang akan dipilih selanjutnya untuk menjadi ringkasan.

$$\mathrm{MMR}(s_i) = \mathrm{Sim}(s_i, Query)$$

2.3 Cosine Similarity

Metode Cosine Similarity merupakan metode yang digunakan untuk menghitung similarity (tingkat kesamaan) antar dua buah objek. Metode *cosine similarity* ini menghitung kesamaan antara dua buah objek (misalkan D1 dan D2) yang dinyatakan dalam dua buah *vector* dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran. (Pradnyana & ER, 2012). Perhitungan *cosine similarity* yang memperhitungkan perhitungan pembobotan kata pada suatu dokumen dapat dinyatakan dengan perumusan (6.3).

$$CosSim(D_i, Q_i) = \frac{q_i\ x\ d_i}{|q_i|\ x\ |d_i|} = \frac{\sum_{j=1}^{t}(q_{ij}\ x\ d_{ij})}{\sqrt{\sum_{j=1}^{t}(q_{ij})^2\ x\ \sum_{j=1}^{t}(d_{ij})^2}}$$

Dimana:

- $CosSim(D_i, Q_i)$ adalah nilai kesamaan untuk dokumen ke i, dan query ke i

- q_{ij} adalah nilai dari query ke i j
- d_{ij} adalah nilai dari dokumen ke i j

2.4 Pembobotan TF-IDF-DF

Metode Term Frequency-Inverse Document Frequency-Document Frequency (TF-IDF-DF) merupakan modifikasi dari metode TF-IDF, karena metode TF-IDF memiliki kekurangan dalam pembobotan kata. Kekurangannya yaitu adanya anggapan bahwa kata yang tersebar dalam dokumen lain tidak penting, sehingga dianggap tidak ada. Padahal kata yang sering muncul dalam kalimat lain bisa jadi merupakan kata yang penting. Akibatnya, nilai bobot yang tinggi diperoleh pada kata yang memiliki frekuensi tinggi dalam dokumen, sedangkan kata yang tersebar di dokumen lain memiliki perhitungan bobot yang kecil. Oleh karena itu, metode TF-IDF ini dikembangkan lebih lanjut untuk mendapatkan bobot perwakilan dari kata-kata yang diekstrak dengan mempertimbangan penyebaran kata di dokumen lain. Document Frequency (DF) yang mengandung kata ke-i berpengaruh pada topik keseluruhan dokumen, sehingga nilai dalam pembobotan kata akan dikalikan dengan DF kata ke-i (Pramono, L.H., A.S. Rohman & H. Hindersah, 2013).
Rumus pembobotan kata dari modifikasi TF-IDF adalah:

$$W_{ij} = tf_{ij}\ x\ \log\frac{D}{dfj}\ x\ dfj$$

2.5 Teks Preprocessing

Text Preprocessing merupakan tahapan pengolahan teks untuk mempersiapkan teks menjadi data yang dapat diolah lebih lanjut. Suatu teks tidak dapat diproses langsung dengan perhitungan matematis, oleh karena itu dibutuhkan *preprocessing* teks

2.5.1 Pemecahan Kalimat

Segmentasi kalimat merupkan langkah awal dari proses text preprocessing. Dalam proses ini, text berita yang terdiri dari paragraph yang dipecah menjadi beberapa kalimat. Pemisah setiap kalimat berdasarkan tanda baca, seperti tanda titik (.), tanda seru (!) dan tanda Tanya (?).

2.5.2 Case Folding

Paragraf berita yang telah dipotong menjadi kalimat akan menjalankan proses case folding. Case folding adalah proses mengubah semua teks menjadi karakter dengan huruf kecil dan membuang semua karakter selain a-z. Jika terdapat tanda baca, bilangan numerik dan simbol akan dihilangkan.

2.5.3 Tokenezing

Suatu proses untuk mengubah kalimat menjadi kata-kata tunggal. Pemotongan kalimat berdasarkan delimiter yang menyusunnya, yaitu spasi (“ “). Proses ini bertujuan agar dapat melakukan proses stemming.

2.5.4 Filtering

Pada proses filtering dilakukan pembuangan stopwatd. Stopward adalah kata-kata yang tidak memiliki makna atau kata yang kurang berarti dan sering muncul dalam kumpulan kata-kata. Stopword dapat berupa kata penghubung, kata ganti, preposisi, dll, seperti: dia, antara, akan, demi, karena, atau, bahwa, bila, juga, kalau, hingga, bagi kecuali, oleh, dan lain-lain.

2.5.5 Stemming

Proses selanjutnya steming yaitu mengembalikan suatu kata ke bentuk akarnya (root word) dengan aturan-aturan tertentu, sehingga setiap kata memiliki representasi yang sama.

2.6 Tipe Evaluasi

Mengukur tingkat akurasi hasil ringkasan oleh sistem terhadap hasil ringkasan manual dapat menggunakan tiga parameter yaitu precision, recall, dan f-measure. Mencari nilai precision, recall, dan F-measure, maka harus mencari terlebih dahulu precision, recall, dan F-measure setiap responden dengan system, kemudian mencari rata-rata precision, recall, dan F-measure pada artikel berita. Berikut perhitungan precision, recall, dan F-measure:

1. Responden 1 (R1) dengan sistem pada artikel 1:

   Perhitungan nilai precision (P):

$$P1 = \frac{correct}{(\text{correct} + \text{wrong})}$$

$$= \frac{2}{(2+0)} = 1$$

   Sedangkan perhitungan nilai recall (R):

$$R1 = \frac{\text{correct}}{(\text{correct} + \text{missed})}$$

$$= \frac{2}{(2+1)} = 0{,}666667$$

2. Responden 2 (R2) dengan sistem pada artikel 1:

   Perhitungan nilai precision (P):

$$P2 = \frac{correct}{(\text{correct} + \text{wrong})}$$

$$= \frac{1}{(1+1)} = 0{,}5$$

Sedangkan perhitungan nilai recall (R):

$$R2 = \frac{\text{correct}}{(\text{correct} + \text{missed})}$$

$$= \frac{1}{(1+2)} = 0{,}333333$$

3. Responden 3 (R3) dengan sistem pada artikel 1:

Perhitungan nilai precision (P):

$$P3 = \frac{correct}{(\text{correct} + \text{wrong})}$$

$$= \frac{1}{(1+1)} = 0{,}5$$

Sedangkan perhitungan nilai recall (R):

$$R3 = \frac{\text{correct}}{(\text{correct} + \text{missed})}$$

$$= \frac{1}{(1+2)} = 0{,}333333$$

Langkah berikutnya adalah mencari rata-rata precision, recall, dan F-measure
pada artikel 1

$$P = \frac{P1 + P2 + P3}{3} = \frac{1 + 0{,}5 + 0{,}5}{3}$$
$$= 0{,}666667$$

$$R = \frac{R1 + R2 + R3}{3}$$
$$= \frac{0{,}666667 + 0{,}333333 + 0{,}333333}{3}$$
$$= 0{,}444444$$

Selanjutnya mencari nilai F-measure (F)

$$F = \frac{2 * R * P}{(\text{R} + \text{P})}$$

$$= \frac{2 * 0{,}444444 * 0{,}666667}{0{,}444444 + 0{,}666667} = 0{,}533333$$

Keterangan:

Correct : jumlah kalimat yang diekstrak oleh sistem dan manusia.

Wrong : jumlah kalimat yang diekstrak oleh sistem tetapi tidak diekstrak oleh manusia.

Missed : jumlah kalimat yang diekstrak oleh masnusia tetapi tidak diekstrak oleh sistem.

3. Metodologi penelitian

Pada penelitian ini, peringkasan teks otomatis dengan menggunakan metode TF-IDF-DF untuk pembobotan kata dan menggunakan metode MMR untuk peringkasannya. Inputan teks berupa artikel berita sebagai single dokument yang merupakan bahan mentah untuk menghasilkan ringkasan (summary). Peringkasan teks otomatis dengan metode TF-IDF-DF dan MMR terdiri dari tahap-tahap berikut:

(1) Artikel berita diinput dengan mamasukkan query berupa judul dan teks beritanya.
(2) Segmentasi kalimat
Memecah paragraf menjadi kalimat-kalimat. Pemecahan dilakukan berdasarkan tanda baca berupa tanda titik (.), tanda tanya (?) dan tanda seru (!). Pemisahannya menggunakan fungsi split().
(3) Case folding
Proses mengubah huruf kapital menjadi huruf kecil dan membuang semua tanda baca, angka dan simbol.
(4) Tokenizing
Memecah kalimat menjadi kata berdasarkan spasi antara kata.
(5) Filtering
Proses pembuangan kata yang tidak berpengaruh terhadap proses peringkasan. Kumpulan kata tersebut berupa stopword, sehingga kata tersebut tidak mengganggu proses pembobotan kata nantinya.
(6) Stemming
Merupakan proses pencarian kata dasar dengan cara membuang imbuhan yang terdapat pada kata (kembali dalam bentuk akarnya).
(7) Perhitungan pembobotan kata dengan metode TF-IDF-DF
Pada tahap ini, penghitungan bobot kata dimulai dengan mencari nilai TF (Term Frequency), yaitu mencari nilai banyaknya kata yang muncul dalam suatu kalimat. Berikutnya mencari nilai IDF(Invers Document Frequency) merupakan perhitungan jumlah kata (term) dalam seluruh kalimat pada dokumen. Terakhir menghitung nilai DF(Document Frequency) yaitu nilai jumlah kalimat yang mengandung suatu kata.
(8) Perhitungan cosine similarity
Menghitung kesamaan antara satu kalimat dengan seluruh kalimat lain dan antara query (judul) dengan seluruh kalimat.
(9) Perhitungan MMR
Tahap ini menghitung nilai relevansi antara nilai cosine similarity query dengan seluruh kalimat dan kalimat dengan seluruh kalimat (Mustaqhfiri, M., Z. Abidin & R. Kusumawati, 2011).
(10) Perankingan kalimat
Ringkasan artikel berita diperoleh dari memilih tiga kalimat (ukuran ringkasan yang diinginkan dengan skor MMR yang tertinggi (Mustaqhfiri, M., Z. Abidin & R. Kusumawati, 2011).

(11) Menentukan nilai precision, recall, dan F-measure.

Sebuah sistem informasi dikatakan baik jika tingkat precision, recall, dan F measure-nya tinggi.

3.1 Data

Data yang diolah berupa berita yang didapatkan dari portal berita olahraga. Data pertama yang akan diambil adalah data pokok berita yaitu tanggal, kategori, link berita, judul dan deskripsi singkat. Setelah mendapat link melakukan proses scraping dan mendapatkan konten dari berita.

3.2 Metode Pengambilan Data

Portal berita yang di pakai dalam pengambilan data merupakan portal berita sports.okezone.com, sport.detik.com, sport.sindonews.com. Dimana terdapat tiga berita yang dipakai. Data diambil menggunakan teknik scraping pada portal berita olahraga tersebut.

3.3 Metode Pengolahan Data

Untuk memperoleh hasil ringkasan artikel berita olahraga, sistem harus melalui beberapa tahap seperti :

1. Text preprocessing
2. Pembobotan kata dengan TF-IDF-DF
3. Menghitung Cosine similarity
4. Menghitung Metode MMR
5. Ekstraksi Ringkasan

3.3.1 Text Preprocessing

Tujuan dari tahap text preprocessing yaitu merubah artikel berita menjadi kata-kata yang siap diproses untuk perhitungan bobot kata. Beberapa proses dari text preprocessing, yaitu segmentasi kalimat, case folding, tokenizing, filtering, dan stemming. Berikut ini adalah salah satu contoh dokumen yang diinputkan dalam proses text preprocessing disertai tahapan proses text preprocessing, Berikut merupakan sampel artikel berita sebagai contoh terdapat pada tabel 3.1:

Tabel 3.1 Sampel artikel berita dan query.

Judul : Liverpool dan Marc Marquez Masuk Nominasi Laureus World Sports Awards 2020.

Isi :

BERLIN - Liverpool masuk dalam dua nominasi untuk merebut penghargaan tahunan (Laureus World Sports Awards 2020) di Berlin, 17 Februari mendatang.

Dua nominasi tersebut yakni tim terbaik dan comeback terbaik 2019.

Dikutip dari laman resmi Laureus, Jumat (17/1), Liverpool masuk dalam daftar tim terbaik tahun ini setelah memenangkan Liga Champions, Piala Super UEFA, dan Piala Dunia Klub FIFA.

3.3.2 Pemecahan Kalimat

Merupakan pemecahan paragraf menjadi kalimat. Pemecahan dilakukan dengan memisahkan berdasarkan tanda baca titik (.), tanda tanya (?) dan tanda seru (!). Berikut Merupakan Hasil dari proses segmentasi kalimat terlihat pada tabel 3.2.

Tabel 3. 1 Hasil Segmentasi Kalimat

No.	Kalimat
Q	Liverpool dan Marc Marquez Masuk Nominasi Laureus World Sports Awards 2020.
D1	BERLIN - Liverpool masuk dalam dua nominasi untuk merebut penghargaan tahunan (Laureus World Sports Awards 2020) di Berlin, 17 Februari mendatang.
D2	Dua nominasi tersebut yakni tim terbaik dan comeback terbaik 2019.
D3	Dikutip dari laman resmi Laureus, Jumat (17/1), Liverpool masuk dalam daftar tim terbaik tahun ini setelah memenangkan Liga Champions, Piala Super UEFA, dan Piala Dunia Klub FIFA.

3.3.3 Case Folding

Paragraf berita yang telah dipotong menjadi kalimat akan menjalankan proses case folding. Case folding adalah mengubah semua teks menjadi karakter dengan huruf kecil dan membuang semua karakter selain a-z. Selain itu, tanda baca, bilangan numerik dan simbol juga dihilangkan. Berikut merupakan hasil dari case folding terdapat pada tabel 3.3.

Tabel 3. 2 Hasil Case Folding

No.	Kalimat
Q	liverpool dan marc marquez masuk nominasi laureus world sports awards
D1	berlin liverpool masuk dalam dua nominasi untuk merebut penghargaan tahunan laureus world sports awards di berlin februari mendatang
D2	dua nominasi tersebut yakni tim terbaik dan comeback terbaik
D3	dikutip dari laman resmi laureus jumat liverpool masuk dalam daftar tim terbaik tahun ini setelah memenangkan liga champions piala super uefa dan piala dunia klub fifa

3.3.4 Tokenizing

Merupakan proses pemotongan kalimat menjadi kata-kata. Pemotongan kalimat berdasarkan delimiter yang menyusunnya, yaitu spasi (" "). Berikut merupakan hasil dari tokenizing pada artikel berita sebagai contoh terdapat pada tabel 3.4. :

Tabel 3. 4 Hasil Tokenizing

NO	KATA	NO	KATA	NO	KATA
	Q	21	world	41	jumat
1	liverpool	22	sports	42	liverpool
2	dan	23	awards	43	masuk
3	marc	24	di	44	daftar
4	marquez	25	berlin	45	tim
5	masuk	26	februari	46	terbaik
6	nominasi	27	mendatang	47	tahun
7	laureus		**D2**	48	ini
8	world	28	dua	49	setelah
9	sports	29	nominasi	50	menang
10	awards	30	tersebut	51	liga
	D1	31	yakni	52	champions
11	berlin	32	tim	53	piala
12	liverpool	33	terbaik	54	super
13	masuk	34	comeback	55	uefa
14	dalam	35	terbaik	56	dan

3.3.5 Filtering

Dalam tahap filtering ini melakukan pembuangan stopword. Stopword adalah kata-kata

yang tidak memiliki makna atau kata yang kurang berarti dan sering muncul dalam kumpulan kata. Berikut merupakan hasil dari filter pada artikel berita sebagai contoh terdapat pada tabel 3.5:

Tabel 3. 5 Hasil Filtering

NO	KATA	NO	KATA	NO	KATA
	Q	19	sports	37	jumat
1	liverpool	20	awards	38	liverpool
2	marc	21	berlin	39	termasuk
3	marquez	22	februari	40	daftar
4	masuk	23	mendatang	41	tim
5	nominasi		**D2**	42	terbaik
6	laureus	24	dua	43	tahun
7	world	25	nominasi	44	menang
8	sports	26	tersebut	45	liga
9	awards	27	yakni	46	champions
	D1	28	tim	47	piala
10	berlin	29	terbaik	48	super
11	liverpool	30	comeback	49	uefa
12	masuk	31	terbaik	50	piala
13	nominasi		**D3**	51	dunia
14	merebut	32	dikutip	52	Klub
15	harga	33	dari	53	Fifa
16	tahunan	34	laman		
17	laureus	35	resmi		
18	world	36	laureus		

3.3.6 Stemming

Stemming, yaitu mengembalikan suatu kata ke bentuk akarnya (rootword), sehingga setiap kata memiliki representasi yang sama. Dalam metode ini hanya menangani afiks (imbuhan) prefiks (awalan) dan sufiks (akhiran) saja. Hal ini disebabkan oleh jarangnya terjadi kasus penambahan imbuhan infiks (sisipan) dalam bahasa Indonesia. Berikut merupakan hasil dari steming pada artikel berita sebagai contoh terdapat pada tabe 3.6. :

Tabel 3. 6 Hasil Stemming

NO	KATA	NO	KATA	NO	KATA
	Q	19	sports	37	jumat
1	liverpool	20	awards	38	liverpool
2	marc	21	berlin	39	masuk
3	marquez	22	februari	40	daftar
4	masuk	23	dating	41	tim
5	nominasi		**D2**	42	baik
6	laureus	24	dua	43	tahun
7	world	25	nominasi	44	menang
8	sports	26	tersebut	45	liga
9	awards	27	yakni	46	champions
	D1	28	tim	47	piala
10	berlin	29	baik	48	super
11	liverpool	30	comeback	49	uefa
12	masuk	31	baik	50	piala
13	nominasi		**D3**	51	dunia
14	rebut	32	kutip	52	klub
15	harga	33	dari	53	fifa
16	tahun	34	laman		
17	laureus	35	resmi		
18	world	36	laureus		

3.3.7 Algoritma TF-IDF-DF

Setelah proses text preprocessing, tahap selanjutnya yaitu penghitungan bobot kata dengan algoritma TF-IDF-DF. Matriks kata untuk penghitungan bobot kata disajikan pada tabel 4.9. Berikut adalah proses perhitungan bobot kata pada term ”a” :

$$W_{ij} = tf_{ij}\ x\ \log\frac{N}{dfj}\ x\ df_i$$

Untuk hasil perhitungan bobot kata pada semua term dengan metode TF-IDF-DF

Berikut merupakan hasil dari proses algoritma pada artikel berita sebagai contoh terdapat pada tabel 3.7:

Tabel 3. 7 Hasil

	TF							W			
	Q	D1	D2	D3	DF	D	IDF(D/DF)	Q	D1	D2	D3
Liverpool	1	1	0	1	3	4	0.124939	0	0.374816	0	0.374816
Marc	1	0	0	0	1	4	0.60206	1	0	0	0
Marquez	1	0	0	0	1	4	0.60206	1	0	0	0
Masuk	1	1	0	1	3	4	0.124939	0	0.374816	0	0.374816
Nominasi	1	1	1	0	3	4	0.124939	0	0.374816	0.4	0
Laureus	1	1	0	1	3	4	0.124939	0	0.374816	0	0.374816
World	1	1	0	0	2	4	0.30103	1	0.60206	0	0
Sports	1	1	0	0	2	4	0.30103	1	0.60206	0	0
Fifa	0	0	0	1	1	4	0.60206	0	0	0	0.60206

Keterangan:
Q : query(judul berita)
tf : term frequency
df : document frequency
idf : inverse document frequency (Log10(N/DF)
W : Bobot Kata (TF * IDF * DF)

Pada kolom tf terdapat angka nol (0) artinya dalam suatu kalimat tidak terdapat kata tersebut, sedangkan angka selain nol (0) menandakan banyaknya kata tersebut didalam suatu kalimat.

3.3.8 Algoritma Cosine Similarity

Jika bobot kata telah diperoleh, selanjutnya mencari nilai cosine similarity. Perhitungan cosine similarity dibagi menjadi 2 tahap, yaitu:

a) Perhitungan relevansi antaradokumen dan query (judul) Menghitung cosinus sudut dari dua vektor, yaitu W (bobot) dari tiap dokumen atau kalimat dengan W (bobot) dari query (judul).

b) Perhitungan similarity antara dokumen Menghitung cosine sudut vektor W (bobot) suatu kalimat dengan vektor W (bobot) kalimat yang lain.

Tabel 3. 8 Relevansi antara Query(judul) dengan dokumen(dokumen1).

	D1	D2	D3
Q	0.603128	0.074677	0.057353

Tabel 3. 9 Similarity antara kalimat dengan membandingkan antara dokumen.

	D1	D2	D3
D1	1	0.053497	0.193932
D2	0.053497	1	0.320626
D3	0.193932	0.320626	1

Keterangan :
Q : query (judul dalam artikel berita)
Di : Dokumen/kalimat (i=1,2,3...)

3.3.9 Metode Maximum Marginal Relevance

Setelah perhitungan cosine similarity diperoleh, maka tahap berikutnya menghitung nilai MMR. Algoritma maximum marginal relevance digunakan untuk merangking kalimat-kalimat sebagai tanggapan terhadap query yang diberikan user. Perhitungan MMR dilakukan dengan iterasi mengkombinasikan 2 matrik cosine similarity, yaitu relevansi antara query terhadap keseluruhan kalimat dan similarity antara kalimat. Prinsip perhitungan metode MMR adalah mengambil kalimat dengan nilai tertinggi dari setiap perhitungan iterasi. Iterasi akan berhenti, jika nilai hasil MMR maksimum sama dengan nol (0). Adapun nilai parameter λ yang digunakan pada perhitungan MMR adalah λ = 0,7 (Carbonell dan Goldstein, 1998 :335). Proses perhitungan MMR sebagai berikut dengan catatan $Sim1(Si,Q)$ adalah relevance query. Sedangkan $1(Si,S')$ adalah similarity kalimat terhadap kalimat yang diekstrak :

$$\text{MMR}(s_i) = \lambda.sim_1(s_i) - (1-\lambda).max\ sim_2(s_i, s_j$$

1. Perhitungan iterasi ke- 1

	MMR1	MMR2	MMR3
Sim	0	0	0
i1	0.42219	0.052274	0.040147
i2		0.036225	-0.01803
i3			-0.25985

Dari hasil perhitungan pada iterasi ke-1, diperoleh nilai maximum MMR = 0,42219 pada D1 atau pada kalimat 1. Oleh karena itu, kalimat 1 akan dipilih sebagai ringkasan. Tabel 4.12 merupakan hasil perhitungan MMR pada iterasi ke-1.

Tabel 4.12 Perhitungan MMR iterasi ke-1

	D1	D2	D3
i1	0.42219	0.052274	0.040147
i2			
i3			

2. Perhitungan iterasi ke-2

Pada iterasi ke- 2, nilai maximum MMR pada iterasi ke- 1 akan digunakan untuk menghitung similarity pada max$Sim2(Si,Sj)$ yaitu max$Sim2(Si,S1)$. Tabel 4.13 ditunjukkan nilai Si yang digunakan.

Tabel 4.13 Nilai Si untuk perhitungan MMR iterasi ke-2

	D1	D2	D3
i1			
i2	-	0.036225	-0.01803
i3	-	-	-0.25985

Dari hasil perhitungan pada iterasi ke-2, diperoleh nilai maximum MMR = 0,036225 pada D2 atau pada kalimat 2. Oleh karena itu, kalimat 2 akan dipilih sebagai ringkasan.
Tabel 4.14 menunjukkan MMR iterasi ke-2.

Tabel 4.14 Nilai MMR iterasi ke-2

	D1	D2	D3
i1	0.42219	0.052274	0.040147
i2	-	0.036225	-0.01803
i3	-	-	-0.25985

3. Perhitungan iterasi ke- 3

Pada iterasi ke- 3, untuk menghitung similarity pada max$Sim2(Si,Sj)$, dicari dengan membandingkan nilai maksimum similarity antara D1(i1) dengan D2 (i2), lihat tabel 4.15. Dengan mencari terlebih dahulu similarity maksimum D1 (kalimat ke 1) terhadap semua dokumen kecuali D1 dan D2 dan similarity maksimum D2 (kalimat ke 2) terhadap semua dokumen kecuali D1 dan D2. Setelah itu dicari nilai maksimum keduanya untuk menghitung max$Sim2(Si,Sj)$.

Tabel 4.15 Nilai similarity yang digunakan

	D1	D2	D3
D1		0.053497	0.193932
D2	0.053497		0.320626
D3	0.193932	0.320626	

Perhitungan perbandingan nilai similarity maksimum $(Si,1)$ dan similarity maksimum $(Si,S4)$ sebagai berikut :

Similarity maksimum $(Si,1)$:
$Sim2(S2,S1)$ = 0,053497
$Sim2(S3,S1)$ = 0,193932

Nilai similarity maksimum $(Si,1)$ adalah 0,193932.

Dari hasil perhitungan pada iterasi ke-3, diperoleh nilai maximum MMR < 0 pada D2 dan D3, sehingga tidak ada kalimat yang dipilih sebagai ringkasan dan iterasi perhitungan akan berhenti. Hasil perhitungan MMR terlihat pada tabel 4.16:

Tabel 4.16 Hasil perhitungan MMR

	D1	D2	D3
Sim	0	0	0
i1	0.42219	0.052274	0.040147
i2	-	0.036225	-0.01803
i3	-	-	-0.25985

Keterangan:
Iterasi: query / judul dalam artikel berita (i=1,2,3...)
Di: Dokument/kalimat (i=1,2,3...)

Pada tabel 4.16 dipaparkan bahwa pada iterasi 1 kalimat (dokumen) yang tertinggi terdapat pada kalimat 1 (D1), sehingga kalimat 1 (D1) menjadi ringkasan. Pada iterasi 2, yang menjadi ringkasannya berikutnya adalah kalimat 2 (D2). Kalimat 1 (D1) pada iterasi 2 tidak terdapat nilai MMR, karena kalimat 1 (D1) telah dipilih sebelumnya menjadi ringkasan. Iterasi 3 tidak ada kalimat yang menjadi ringkasan, karena nilai max MMR < = 0. Perangkingan hasil ringkasan terlihat seperti tabel 4.17:

Tabel 4.17 Hasil perangkingan kalimat

Ranking	Kalimat (D) ke	Max MMR
1	D1	0.42219
2	D2	0.036225

Pada Tabel 4.17, kalimat yang menjadi ringkasan adalah kalimat 1 (D1) dengan nilai maksimal MMR adalah 0,42219 dan kalimat 2 (D2) dengan nilai maksimal MMR adalah 0,036225.

Tabel 5. 1 Mengambil sampel berita sebanyak 5 data untuk di uji.

berita		sumber
1	LIVERPOOL - Liverpool sudah melupakan euforia setelah kembali lagi ke jalur kemenangan dengan mengalahkan AFC Bournemouth	sindonews
2	Sofyan Hadi Wafat, Persija Jakarta Berduka	sindonews
3	Legenda Persija Jakarta, Sofyan Hadi Tutup Usia	sindonews
4	Meski Negatif Tertular Virus Corona, PSG Masih Pantau Kondisi Mbappe	sindonews
5	Hasil Lengkap Pertandingan NBA, Rabu (11/3)	sindonews

Sehingga didapatkan hasil max MMR bahwa ranking 1 akan dijadikan sebagai ringkasan dari berita olahraga tersebut.

4. Hasil Uji Coba

Mengukur tingkat akurasi hasil ringkasan oleh sistem terhadap hasil ringkasan manual dapat menggunakan tiga parameter yaitu precision, recall, dan f-measure. Mencari nilai precision, recall, dan F-measure, maka harus mencari terlebih dahulu precision, recall, dan F-measure setiap responden dengan system, kemudian mencari rata-rata precision, recall, dan F-measure pada artikel berita. Berikut perhitungan precision, recall, dan F-measure:

Tabel 5. 2 Menentukan ringkasan dari Responden/Pakar dan sistem

berita	r1	r2	r3	sistem
1	1,2,3,4	1,2,5,10	1,2,3	8,4,1,7,3,11
2	4,5	1,2,4,8	1,4,5	1,5
3	2,3	1,2,11	1,2,3	1,2,7
4	1,2,3	1,2,3,11	1,2,4	2,4
5	1,2,3,4	1,2,4,8	1,2,3,6	6,3

Hasil ringkasan artikel berita yang dilakukan sistem terhadap artikel berita dari portal berita online memperoleh hasil terlihat pada tabel 5.12.

Tabel 5. 3 Hasil ringkasan artikel berita

berita ke	r1	r2	r3	sistem	(rata2)precission	(rata2)recall	f-measure
1	1,2,3,4	1,2,5,10	1,2,3	8,4,1,7,3,11	0.333333	0.555555556	0.41666667
2	4,5	1,2,4,8	1,4,5	1,5	0.666667	0.472222222	0.55284553
3	2,3	1,3,11	1,2,3	1,2,7	0.555556	0.666666667	0.60606061
4	1,2,3	1,2,3,11	1,2,4	2,4	0.666667	0.416666667	0.51282051
5	1,2,3,4	1,2,3,4,8	1,2,3,6	6,3	0.666667	0.316666667	0.42937853
					0.577778	0.485555556	0.50355437

Keterangan:
Ri: hasil ringkasan yang dilakukan oleh responden (i=1,2,3).
Sistem: hasil ringkasan yang dilakukan oleh sistem.
Precision: kemampuan sistem memanggil dokumen yang relevan.
Recall: kemampuan sistem memanggil dokumen yang tidak relevan.
F-measure: nilai akurasi.

5. Kesimpulan

Dengan membangun sistem infomasi berita menggunakan Metode *Maximum Marginal Relevance* dapat digunakan untuk memudahkan dalam memproses peringkasan teks pada peringkasan berita. Peringkasan teks dengan metode MMR dan TF-IDF pada berita yang diambil dari portal berita online berskala internasional/nasional menghasilkan ringkasan. Pengaruh *Maximum Marginal Relevance* terhadap hasil akurasi ringkasan sistem yaitu, pengujian diambil dari data sampel berita online sebanyak 5 berita sindonews menggunakan lamda 0,7 yang nanti nya hasil ringkasan digunakan untuk dibandingkan dengan ringkasan dari sistem dan ringkasan oleh pakar meliputi an. Maya Rahma, dan setelah itu dicari *correct (jumlah kalimat yang diekstrak oleh sistem dan manusia)* setelah itu dicari *wrong (jumlah kalimat yang diekstrak oleh sistem tetapi tidak diekstrak oleh manusia)* dan *missed (jumlah kalimat yang diekstrak oleh masnusia tetapi tidak diekstrak oleh sistem)*, ketika hasil tersebut diperoleh lalu dicari *precision, recall, f-measure* dari setiap responden dan hasil pengujian nantinya didapat dari mencari rata-rata dari *precision, recall, f-measure*, sehingga pengaruh lamda 0,7 menghasilkan rata-rata akurasi Precision 57.7%, Recall 48.5% dan F – Measure 50.3%

6. Saran

1) Sistem dapat melakukan scraping secara otomatis berdasarkan tanggal yang diinputkan oleh pengunjung secara *realtime*.
2) Proses *scraping* dilakukan dengan pengkodean yang dinamis, sehingga jika terjadi perubahan tampilan dari target *scraping* dapat diubah melalui aplikasi.
3) Menambahkan atau merubah fitur sesuai dengan dokumen yang akan diringkas.
4) Proses scraping pada portal berita yang seharusnya mempunyai banyak halaman seperti berita tribun yang seharusnya perlu di ringkas.
5) Pengembangan lebih lanjut disarankan untuk memasukkan kalimat pertama sebagai bahan pertimbangan dalam menentukan ringkasan karena secara umum pada berita, kalimat pertama telah menggambarkan isi berita.

Daftar Pustaka:

Ambar. (2019). *DosenBahasa Teks Berita*. Dosenbahasa.com. https://dosenbahasa.com/teks-berita

Das, A. F. T. M. (2007). A Survey on Automatic Text Summarization. *Language Technologies Institute Carnegie Mellon University*, 1–31.

Dwi Hadya Jayani, H. W. (2019). *Berapa Pengguna Internet di Indonesia?* KataData.co.id. https://databoks.katadata.co.id/datapublish/2019/09/09/berapa-pengguna-internet-di-indonesia

Mamad, J. (2020). *10 SITUS (WEB PORTAL) BERITA INDONESIA TERBAIK DAN TERPERCAYA*. CENTERKLIK. https://www.centerklik.com/situs-web-portal-berita-indonesia-terbaik-terpercaya/

Melita, R., Amrizal, V., Suseno, H. B., Dirjam, T., Studi, P., Informatika, T., & Sains, F. (2018). *(Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus : Syarah Umdatil Ahkam)*. *11*(2).

Murray, G., Renals, S., & Carletta, J. (2005). Extractive summarization of meeting recordings. *9th European Conference on Speech Communication and Technology*, 593–596.

POLINEMA. (2019). *Panduan penulisan laporan tugas akhir*.

Portal web. (2019). Wikipedia. https://id.wikipedia.org/wiki/Portal_web

Prasad, D. R. S., Uplavikar, N. M., Wakhare, S. S., Jain, V., & Yedke, T. A. (2012). Feature Based Text Summarization. *International Journal of Advances in Computing and Information Researches*, *1*(2), 15–18. http://ijacir.com/mojms/index.php/IJACIR/article/view/53

Sasmito, G. W. (2017). Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Informatika:Jurnal Pengembangan IT (JPIT)*, *2*(1), 6–12.

Savanti, N., Gotami, W., & Dewi, R. K. (2018). Peringkasan Teks Otomatis Secara Ekstraktif Pada Artikel Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode Latent Semantic Analysis. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, *2*(9), 2821–2828.

Simarangkir, M. S. H. (2017). Studi Perbandingan Algoritma - Algoritma Stemming Untuk Dokumen Teks Berbahasa Indonesia. *Jurnal Inkofar*, *1*(1), 41–47.

Trisaputra, Y., & Abriantini, G. (2016). *Aplikasi Peringkasan Teks Berita Otomatis Menggunakan Pembobotan Kalimat Aplikasi Peringkasan Teks Berita Otomatis Menggunakan Pembobotan Kalimat Pendahuluan*. *August*.

Wardhana, S. R., Yunianto, D. R., Arifin, A. Z., & Purwitasari, D. (2015). Pembobotan Kata Berbasis Preferensi Dan Hubungan Semantik Pada Dokumen Fiqih Berbahasa Arab. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *2*(2), 132. https://doi.org/10.25126/jtiik.201522146

Wicaksana, D. A., Adikara, P. P., & Adinugroho, S. (2018). Clustering Dokumen Skripsi Dengan Menggunakan Hierarchical Agglomerative Clustering. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, *2*(12).

Winata, F., Rainarli, E., Informatika, T., & Indonesia, U. K. (2016). Implementasi cross method latent semantic analysis untuk meringkas dokumen berita berbahasa indonesia 1,2. *Techno.COM*, *15*(4), 266–277.

Yulyardo, Okta Purnama Rahadian, M. S. (2018). *PERINGKAS TEKS OTOMATIS (AUTOMATIC TEXT SUMMARIZATION)*. Binus.ac.id. https://mti.binus.ac.id/2018/12/26/peringkas-teks-otomatis-automatic-text-summarization/

https://www.alexa.com/topsites/countries/ID