

1



JURNAL INFORMATIKA

Penerbit : Jurusan Teknik Informatika
Institut Teknologi Nasional

Penanggung Jawab : Ketua Jurusan
Teknik Informatika
Institut Teknologi Nasional

Pemimpin Redaksi : Dewi Rosmala

Wakil Pemimpin : Yusup Miftahuddin

Mitra Bestari : Arief Syaichu Rohman

Redaksi Pelaksana : 1. Rio Korio Utoro
2. Irma Amelia Dewi

ISSN : 2087-5266

DAFTAR ISI

No. 3 Vol 6, September - Desember 2015

1-11

Jasman Pardede, Frisky Helgandamar

Aplikasi Peringkasan Dokumen Menggunakan
Algoritma *Interactive Graph-based* Dan *Similarity*

12-21

**Mira Musrini^[1], Andriana^[2],
Muh. Tantra Gazali^[1]**

Identifikasi Tekstur Bahan
Dasar Batik dengan Menggunakan
Fitur Gray Level Difference Method

22-30

Dewi Rosmala, Deden Dodik Ginanjar

Sistem Simulasi Peramalan Waktu Siklus
Lampu Lalu Lintas dengan Metode Holt-winters
Multiplicative dan Webster

31 - 39

Jasman Pardede, Zidni Nurrobi Agam

Implementasi Metode Latent
Semantic Indexing
Pada Aplikasi Information Retrieval

40 - 50

M. Ichwan, Milda Gustiana, Arief Syafiudin

Implementasi Metoda Unit Selection
Synthesizer dalam Pembuatan Speech Synthesizer
Suara Suling Recorder

51 - 69

Uung Ungkawa, Adinda Ria Rumondang Veranita

Implementasi Algoritma Nearest Neighbor
Pada Sistem Rekomendasi Pembelian Rumah

JURNAL INFORMATIKA diterbitkan 3 kali dalam satu tahun.
Berisi tulisan yang diangkat dari hasil penelitian
dan kajian analisis di bidang ilmu pengetahuan dan Teknologi.

Alamat redaksi dan tata usaha :

Jurusan Teknik Informatika Institut Teknologi Nasional
Gedung 2 lantai 2

Jl. PH. Hasan Mustofa No. 23 Bandung 40124

Telp. 022-7272215 || Fax : 022-7202892 || e-mail : d_rosmala@itenas.ac.id

APLIKASI PERINGKASAN DOKUMEN MENGGUNAKAN ALGORITMA *INTERACTIVE GRAPH-BASED* DAN *SIMILARITY*

Jasman Pardede, Frisky Helgandamar

Jurusan Teknik Informatika, Fakultas Teknologi Industri
Institut Teknologi Nasional Bandung

Jasmanpardede78@gmail.com, Friskydamar@gmail.com

ABSTRAK

Peringkasan dokumen adalah sebuah cara yang dilakukan untuk mendapatkan kumpulan informasi penting dari sebuah dokumen. Metoda ekstraksi dokumen yang digunakan adalah algoritma Interactive Graph-Based dan Similarity. Interactive Graph-Based dan Similarity meringkas dokumen dengan menghitung bobot kalimat selanjutnya menghitung matriks kemiripan (Similarity) antara bagian kalimat dalam teks. Dalam peringkasan ini dilakukan proses segmentasi isi dokumen dalam graph-based summarization algorithm, sebuah dokumen direpresentasikan menjadi sebuah graph berarah. Vertex/node pada graph berarah tekstual berupa kalimat-kalimat dalam teks. Edge/link dalam graph tersebut menunjukkan keterhubungan antar vertex/node. Keterhubungan dapat berupa similarity antar kalimat, setelah mendapatkan hasil perhitungan dari pengolahan bobot matriks kemiripan (Similarity) dan didapatkan pelintasan terpendek dimana arah lintasan terpendek memiliki nilai edge sebagai jalur, maka hasil simpul pelintasan terpendek pada graph berarah dapat diimplementasikan pada kalimat untuk menghasilkan ringkasan ekstraksi.

Kata Kunci : *Peringkasan dokumen, Interactive Graph-based dan Similarity.*

ABSTRACT

Automatic Text Summarization is a method to get a collection of important information from a document. Document extraction method used is the algorithm Interactive Graph-Based and Similarity. Interactive Graph-Based and Similarity summarize the document by calculating the weight of the next sentence calculate the similarity matrix similarity between the sentences in the text. Interactive Graph-Based and Similarity summarize the document by calculating the similarity between the sentences in the text. In summarizing the contents of the document segmentation process is carried out in graph-based summarization algorithm, a document is represented into a graph. Vertex / nodes in a graph textual form of the sentences in the text. Edge / link in the graph shows the connectivity between vertices / nodes. Connectedness can be a similarity between sentences. after getting the results of the calculation from the similarity matrix weighting processing and obtain the shortest path in which the direction of the shortest path as the path edge value, then results of node shortest path the in directed graph can be implemented to produce a summary sentence extraction.

Keywords: *Automatic Text Summarization, Interactive Graph-Based and Similarity.*

PENDAHULUAN

Latar Belakang

Peringkasan dokumen adalah sebuah cara yang dilakukan untuk mendapatkan kumpulan informasi penting dari sebuah dokumen. Pengguna dapat memanfaatkan peringkasan untuk mendapatkan intisari suatu dokumen dalam waktu singkat. Tentunya jika jumlah dan ukuran dokumen yang akan diringkas banyak maka peringkasan secara manual sangat tidak efisien. Jika peringkasan dokumen ini dilakukan oleh manusia secara manual tentunya waktu dan tenaga tidak akan sebanding dengan dokumen yang jumlahnya tidak terbatas, untuk itu diperlukan sebuah metode peringkasan dokumen secara otomatis.

Peringkasan dokumen otomatis adalah aplikasi peringkasan dengan menerapkan algoritma tertentu untuk mendapat poin-poin penting dari dokumen asli. Peringkasan dokumen menghasilkan suatu bentuk penyajian lain dari dokumen yaitu ringkasan, yang berisi intisari dari topik yang dibicarakan pada isi dokumen. Oleh karena itu dikembangkan suatu metode peringkasan dokumen secara otomatis menggunakan sistem.

Interactive Graph-based dan *Similarity* merupakan sebuah metode dalam peringkasan dokumen digambarkan menjadi sebuah graph. *Vertex* pada graph tekstual dapat berupa kata-kata, kalimat-kalimat, atau paragraph-paragraph dalam teks. *Edge* dalam graph menunjukkan keterhubungan antar *vertex*. Keterhubungan dapat berupa *similarity* antar kalimat. Pada penelitian ini akan mengimplementasikan *Interactive Graph-Based* dan *Similarity*. *Interactive Graph-Based* dan *Similarity* sesuai dengan namanya, metode ini memakai representasi graph dalam pengelompokan dokumen. Graph yang dibangun merupakan graph berarah dimana arah tersebut menunjukkan struktur kalimat.

Berdasarkan permasalahan yang telah dikemukakan, untuk membangun sebuah aplikasi ringkasan dokumen, maka dilakukan penelitian dengan judul **APLIKASI PERINGKASAN DOKUMEN MENGGUNAKAN INTERACTIVE GRAPH-BASED DAN SIMILARITY.**

Rumusan Masalah

Berdasarkan latar belakang, maka dirumuskan beberapa masalah sebagai berikut :

1. Bagaimana proses *preprocessing* dapat bekerja pada sistem.
2. Bagaimana mengimplementasikan *Interactive Graph-based* dan *Similarity* dalam ekstraksi sebuah dokumen.
3. Bagaimana sebuah dokumen direpresentasikan menjadi sebuah graph.
4. Bagaimana cara menentukan pelintasan terpendek pada graph berarah.

Tujuan

Adapun tujuan dari penelitian ini adalah untuk melakukan analisis dan implementasi metode *Interactive Graph-based* dan *Similarity* pada aplikasi peringkasan ekstraksi dokumen.

Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

1. Dokumen yang digunakan pada penelitian ini adalah dokumen berita olahraga yang berbahasa Indonesia.
2. Batas kalimat yang akan dihitung berjumlah 40.
3. Peringkasan dokumen yang termasuk dalam pendekatan pada peringkasan teks adalah ekstraksi.

4. Aplikasi yang dikembangkan dalam menerapkan metode berbasis desktop.
5. Algoritma Stemming yang digunakan pada penelitian ini adalah algoritma Nazief dan Adriani^[3].

LANDASAN TEORI

Peringkasan^[4]

Ringkasan adalah penyajian karangan atau peristiwa yang panjang dalam bentuk yang singkat dan efektif. Ringkasan adalah intisari karangan tanpa hiasan. Fungsi sebuah ringkasan adalah memahami atau mengetahui isi sebuah buku atau karangan. Ciri-Ciri ringkasan yang baik adalah.

1. Mempersingkat suatu bacaan.
2. Terdapat intisari bacaan.
3. Bentuknya lebih pendek atau lebih ringkas.
4. Struktural wacananya tetap.

Text Preprocessing^[8]

User menginputkan sebuah dokumen yang akan diringkas kemudian proses dimulai dari teks *preprocessing*. Dalam teks *preprocessing* ini melalui beberapa tahap yaitu pemecahan kalimat dan *case folding*, *tokenizing*, *filtering*, *stemming*. Pemecahan kalimat adalah tahapan pertama yang akan dijelaskan yaitu memecahkan string teks dokumen menjadi kumpulan kalimat-kalimat berikut ini contoh tabel 1 dan tabel 2 proses pemecahan kalimat:

Tokenizing

Kalimat hasil dari *case folding* dan pemecahan kalimat kemudian dilakukan proses *tokenizing* yaitu menghilangkan karakter pemisah atau *delimiter* yang menyusunnya berupa karakter spasi.

Filtration

Filtration atau stopword removal merupakan proses lanjutan dari tokenizing di dalam *preprocessing* kalimat. Proses *filtration* merupakan proses untuk menghilangkan kata yang 'tidak relevan' pada hasil *parsing* sebuah dokumen teks dengan cara membandingkannya dengan stoplist yang ada. *Stoplist* disebut juga dengan *stopword*. *Stoplist* berisi sekumpulan kata yang 'tidak relevan'. Contohnya didalam bahasa Indonesia stop word dapat disebut sebagai kata tidak penting, misalnya : Kata Ganti (kami, kita, mereka, itu), dan kata Bilangan (beberapa, banyak, sedikit).

Stemming^[3]

Pada bahasa Indonesia, Stemming merupakan suatu proses yang terdapat makna dasar ini melekat pada kata dasar sebuah kata atau makna leksikal. Makna leksikal juga dapat disebut juga makna asli sebuah kata yang belum mengalami afiksasi atau proses penambahan imbuhan ataupun penggabungan dengan kata yang lain.

Mencari Nilai TF-IDF^[1]

Tf-idf digunakan sebagai faktor bobot (W) dalam pencarian informasi dan text mining. Proses yang dilakukan pertama kali adalah menghitung kemunculan jumlah term yang muncul dalam kalimat. sedangkan K1, K2, K3, K4, K5, K6 adalah kalimat yang telah dipecah dalam proses pemecahan kalimat yang berpatokan pada tanda baca titik. Dimana : Term : kata dari hasil stemming, K1,K2,K3,K4,K5,K6: kalimat yang ada dalam 1 dokumen

-Df : total term dalam setiap kalimat

-IDF: inverse dari df yaitu $\text{Log } \frac{N}{df}$.

Kemudian proses dilanjutkan kedalam pencarian bobot kalimat (W) terhadap dokumen (D) dan term (t). Berikut tabel 6 hasil pembobotan

Interactive Graph-Based dan Similarity^[6,7,5]

Algoritma *Interactive Graph-Based* dan *Similarity* adalah sebuah algoritma yang digunakan untuk metode ekstraksi ringkasan (*extractive Summary*) yang dapat meringkas satu atau lebih dari satu dokumen. Secara singkat, sebuah algoritma peringkat berbasis graph adalah sekumpulan benda-benda yang disebut simpul (*node/vertex*) yang dihubungkan oleh sisi (*edge*). *Edge* yang menghubungkan vertex juga disesuaikan dengan kebutuhan dan unit teks yang dipilih. Matriks kemiripan (*Similarity*) biasanya digunakan untuk menyatakan hubungan suatu vertex dengan vertex lain antara kalimat satu dengan kalimat lain. *Similarity* ini juga dapat didefinisikan sendiri, vertex pada graph berarah dapat berupa kalimat-kalimat *edge* dalam graph berarah menunjukkan keterhubungan antar vertex. *Edge* yang menghubungkan vertex juga disesuaikan dengan kebutuhan dan unit teks yang dipilih.

Sebuah $edge\{R_i, R_j\}$ yang berhubungan dengan vertex R_i dan vertex R_j dan dinyatakan dengan $R_i R_j$. R_i dan R_j merupakan ujung (pangkal) dari *edge* tersebut. Maka R_i dan R_j berdekatan atau bertetangga dan menyertai *edge* $R_i R_j$. Kedua *edge* dikatakan berdekatan jika memiliki 1 vertex akhir yang sama, sehingga setiap lintasan (*edge*) yang terpilih dapat melakukan proses pemilihan lintasan terpendek. Nilai *edge* dapat dihitung menggunakan persamaan (1) lebih tepatnya berikut tabel 9 hasil perhitungan *edge*.

$$(R_i, R_j) = \frac{R_i + R_j}{SUM W} = edge \dots \dots (1)$$

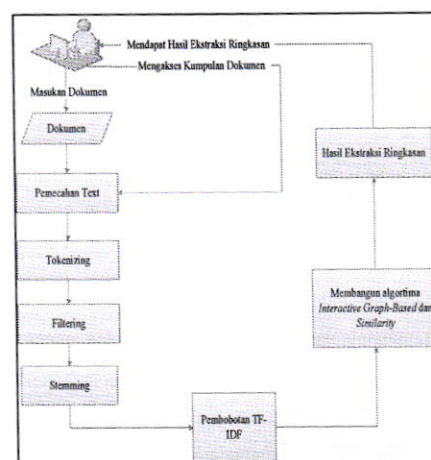
Dimana:

R_i : index dari simpul awal, R_j : index dari simpul tujuan, SUM W : Jumlah bobot keseluruhan, *Edge* : Nilai *edge* pada graph berarah

ANALISIS SISTEM

Peringkasan dokumen otomatis yang dibuat merupakan sistem yang dapat membaca teks single dokumen dan secara otomatis menghasilkan sebuah ringkasan. Peringkasan menggunakan pendekatan ekstraksi menggunakan algoritma *Interactive Graph-Based* dan *Similarity*.

Secara umum proses yang dilakukan dalam aplikasi peringkasan ini adalah proses text preprocessing, pembobotan tf-idf, kemudian perhitungan matriks kemiripan (*similarity*) kalimat dan ekstraksi *Interactive Graph-Based*. Dokumen yang telah dipilih oleh user kemudian dibaca secara keseluruhan oleh sistem untuk mempersiapkan dokumen kepada proses pertama yang dijalankan. Berikut gambar 3 rancangan sistem peringkasan dokumen.



Gambar 3 Rancangan Sistem Peringkasan Dokumen

Tahapan analisa terhadap kebutuhan perancangan aplikasi peringkasan dokumen dengan tahapan proses yang dilakukan adalah.

1. User memilih dokumen yang akan diringkaskan dengan format *.doc, *.docx, dan *.pdf. berikut ini tabel 1 contoh dokumen.

Tabel 1: Dokumen

Chelsea,Denda,Drogba
Isi: <i>Tempo interaktif</i> london: hubungan mesra antara chelsea dengan didier drogba tampaknya akan segera berakhir. Striker pantai gading itu kemungkinan akan mendapat denda 100,000 poundsterling (rp 1,5 miliar) akibat mengkritik klub yang menurutnya tidak mendukung performanya di stamford bridge musim ini. Nilai denda itu setara gaji sepekan yang diterima drogba. Dia juga mengaku tidak berminat untuk kembali bermain setelah dibekap cedera lutut dan mengkritik gaya pemilihan pemain yang ditunjukkan luiz felipe scolari. Komentarnya jelas membuat big phil dan ceo <i>the blues</i> peter Kenyon marah. Atas aksinya ini klub telah memanggil drogba yang mungkin memilih mengakhiri kariernya bersama Chelsea yang dimulainya sejak 2004 setelah hengkang dari marseille.

2. Kemudian sistem melakukan rangkaian pemrosesan tahap text preprocessing yaitu :

- a) Pemecahan kalimat merupakan proses pemecahan string teks dalam dokumen menjadi kumpulan kalimat. Berikut contoh tabel 2 proses pemecahan kalimat.

Tabel 2: Pemecahan Kalimat

no	kalimat
1	<i>interaktif</i> , london: hubungan mesra antara chelsea dengan didier drogba tampaknya akan segera berakhir
2	striker pantai gading itu kemungkinan akan mendapat denda 100,000 poundsterling (rp 1,5 miliar) akibat mengkritik klub yang menurutnya tidak mendukung performanya di stamford bridge musim ini
3	nilai denda itu setara gaji sepekan yang diterima drogba
4	dia juga mengaku tidak berminat untuk kembali bermain setelah dibekap cedera lutut dan mengkritik gaya pemilihan pemain yang ditunjukkan luiz felipe scolari

5	komentarnya jelas membuat big phil dan ceo <i>the blues</i> peter kenyon marah
6	atas aksinya ini klub telah memanggil drogba yang mungkin memilih mengakhiri kariernya bersama chelsea yang dimulainya sejak 2004 setelah hengkang dari marseille.

- b) *Tokenizing* proses yaitu menghilangkan karakter pemisah atau delimiter yang menyusunnya berupa karakter pemisah atau *delimiter* yang menyusunnya berupa karakter spasi. Dari hasil proses *tokenizing* kalimat dipisah menjadi susunan perkata seperti yang ada di tabel 3.
- c) *Filtering* adalah proses untuk menghilangkan kata yang 'tidak relevan' pada hasil *parsing* sebuah dokumen teks dengan cara membandingkannya dengan *stopword* yang ada. *Stoplist* disebut juga dengan *stopword*. *Stoplist* berisi sekumpulan kata yang 'tidak relevan', namun sering sekali muncul dalam sebuah dokumen. Dari tabel 3 dirubah mengalami proses perubahan seperti terdapat pada tabel 4.

Tabel 3 : *Tokenizing*

Kata	Kata	Kata	Kata
aksinya	Drogba	marseille	hubungan
bermain	Felipe	memanggil	Antara
berminat	Gading	memilih	dengan
big	Gaji	mendukung	akan
blues	Gaya	mengakhiri	Segera
bridge	hengkang	mengkritik	itu
cedera	interaktif	mesra	kemungkina n
coe	kariernya	milliar	Akibat
chelsea	kenyon	musim	yang

denda	Klub	nilai	menurutnya
dibekap	komentarnya	pantai	Tidak
didier	London	pemain	di
dimulainya	Luiz	pemilihan	ini
diterima	Lutut	performanya	Dia
ditunjukkan	Marah	peter	juga
phil	Scolari	satmford	setelah
poundstering	sepekan	striker	Dan
rp	setara	tempo	jelas
the	Atas	telah	sejak
mungkin	bersama		

Tabel 4 : Hasil *Filtration*

kata	kata	kata	Kata
aksinya	drogba	marseille	dibekap
bermain	felipe	memanggil	didier
berminat	gading	memilih	dimulainya
big	gaji	mendukung	diterima
kata	kata	kata	Kata
blues	gaya	mengakhiri	ditunjukkan
bridge	hengkang	mengkritik	phil
cedera	interaktif	mesra	poundstering
coe	kariernya	milliar	rp
chelsea	kenyon	musim	dibekap
denda	klub	nilai	didier
dimulainya			

d) Stemming adalah proses pencarian bentuk dasar suatu kata dengan cara menghilangkan imbuhan. Selain itu stemming juga dapat digunakan untuk mengurangi ukuran dari suatu ukuran index file. Misalnya dalam suatu deskripsi terdapat variant kata “memberikan”, “diberikan”, “memberi” dan “diberi” hanya memiliki akar kata (stem) yaitu “beri”. Hasil proses stemming terdapat pada tabel 5.

3. Dari hasil rangkaian proses *text preprocessing* mempersiapkan teks menjadi data kemudian dilakukan pembobotan kalimat menggunakan TF-IDF. Proses yang dilakukan pertama kali adalah menghitung kemunculan jumlah term yang muncul dalam kalimat. sedangkan K1, K2, K3, K4, K5, K6 adalah kalimat yang telah dipecah dalam proses pemecahan kalimat yang berpatokan pada tanda baca titik.

Dimana : -Term : kata dari hasil stemming -K1,K2,K3,K4,K5,K6: kalimat yang ada dalam 1 dokumen- Df : total term dalam setiap kalimat -IDF: inverse dari df yaitu $\text{Log } \frac{N}{df}$.

Kemudian proses dilanjutkan kedalam pencarian bobot kalimat (W) terhadap dokumen (D) dan term (t). Berikut tabel 6 hasil pembobotan.

Tabel 5 : Hasil *Stemming*

kata	kata	kata	Kata
aksi	drogba	marseille	tunjuk
main	felipe	panggil	marah
minat	gading	pilih	peter

big	gaji	dukung	phil
blues	gaya	akhiri	scolari
bridge	hengkang	kritik	satmford
cedera	interaktif	mesra	poundsterning
coe	kariernya	milliar	sepekan
chelsea	kenyon	musim	striker
denda	klub	nilai	rp
dibekap	komentar	pantai	setara
didier	london	pemain	tempo
mulai	Luiz	pilih	performa
terima	lutut		

4. Dari hasil rangkaian proses *text preprocessing* mempersiapkan teks menjadi data kemudian dilakukan pembobotan kalimat menggunakan TF-IDF. Proses yang dilakukan pertama kali adalah menghitung kemunculan jumlah term yang muncul dalam kalimat. sedangkan K1, K2, K3, K4, K5, K6 adalah kalimat yang telah dipecah dalam proses pemecahan kalimat yang berpatokan pada tanda baca titik.

Dimana : -Term : kata dari hasil stemming -K1,K2,K3,K4,K5,K6: kalimat yang ada dalam 1 dokumen- Df : total term dalam setiap kalimat -IDF: inverse dari df yaitu $\log \frac{N}{df}$.

Kemudian proses dilanjutkan kedalam pencarian bobot kalimat (W) terhadap dokumen (D) dan term (t). Berikut tabel 6 hasil pembobotan.

Tabel 6 : Hasil Bobot TF-IDF

K1	K2	K3	K4	K5	K6
5,44 7	10,76 9	4,66 9	8,73 6	7,00 3	5,62 3

5. Setelah melakukan pembobotan **TF-IDF** selanjutnya diteruskan dengan mensortir (R)bobot kalimat dimana (R) adalah Range menjadi simpul pada graph berarah. Sehingga (R) ditentukan bobot terbesar untuk dijadikan urutan pertama yang akan diterapkan pada simpul awal graph berarah. Kemudian seluruh bobot kalimat dijumlahkan guna mencari lintasan terpendek pada graph berarah. Berikut tabel 7 hasil bobot keseluruhan.

Tabel 7 : Hasil Bobot Keseluruhan

R1	K2 : 10,769
R2	K1 : 5,447
R3	K3 : 4,669
R4	K4 : 8,736
R5	K5 : 7,003
R6	K6 : 5,623
SUM W	42,247

6. Tahap akhir adalah membangun algoritma Interactive Graph-based dan Similarity. Tahapan – tahapan membangun algoritma tersebut dengan menghitung matriks kemiripan (Similarity) dan membangun graph berarah, Kemudian sistem melakukan rangkaian pemrosesan tahap akhir menghitung matriks kemiripan (Similarity) dan membangun graph berarah. Menghitung Matriks Kemiripan dan membangun graph berarah adalah Matriks kemiripan (Similarity) biasanya digunakan untuk menyatakan hubungan suatu vertex dengan vertex lain antara kalimat satu dengan kalimat lain. Similarity ini juga dapat didefinisikan sendiri, vertex pada

graph berarah dapat berupa kalimat-kalimat edge dalam graph berarah menunjukkan keterhubungan antar vertex. Edge yang menghubungkan vertex juga disesuaikan dengan kebutuhan dan unit teks yang dipilih. Selanjutnya dari hasil bobot TF-IDF, ditentukan Matriks kemipiran (Similarity) sesuai tujuan untuk mencari lintasan graph terpendek dengan menggunakan graph berarah maka perhitungan untuk matriks kemipiran (Similarity) yaitu : (R1,R2), (R1,R3), (R1,R4), (R1,R5), (R1,R6), (R2,R3), (R2,R4), (R2,R5), (R2,R6), (R3,R4), (R3,R5), (R3,R6), (R4,R5), (R4,R6), dan (R5,R6). Berikut tabel 8 matriks kemipiran (Similarity):

Tabel 8 : Matriks Kemipiran

	R1	R2	R3	R4	R5	R6
R1	(R1,R1)	(R1,R2)	(R1,R3)	(R1,R4)	(R1,R5)	(R1,R6)
R2	(R2,R1)	(R2,R2)	(R2,R3)	(R2,R4)	(R2,R5)	(R2,R6)
R3	(R3,R1)	(R3,R2)	(R3,R3)	(R3,R4)	(R3,R5)	(R3,R6)
R4	(R4,R1)	(R4,R2)	(R4,R3)	(R4,R4)	(R4,R5)	(R4,R6)
R5	(R5,R1)	(R5,R2)	(R5,R3)	(R5,R4)	(R5,R5)	(R5,R6)
R6	(R6,R1)	(R6,R2)	(R6,R3)	(R6,R4)	(R6,R5)	(R6,R6)

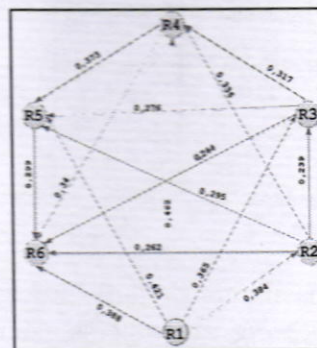
Sebuah edge{R1, R2} yang berhubungan dengan vertex R1 dan vertex R2 dan dinyatakan dengan R1 R2. R1 dan R2 merupakan ujung (pangkal) dari edge tersebut. Maka R1 dan R2 berdekatan atau bertetangga dan menyertai edge R1 R2. Kedua edge dikatakan berdekatan jika memiliki 1 vertex akhir yang sama, sehingga setiap lintasan (edge) yang terpilih dapat melakukan proses pemilihan lintasan terpendek. Nilai edge dapat dihitung menggunakan persamaan (1) lebih tepatnya berikut tabel 9 hasil perhitungan edge.

$$(R_i, R_j) = \frac{R_i + R_j}{SUM W} = edge \dots \dots (1)$$

Tabel 9 : Hasil perhitungan edge

	R1	R2	R3	R4	R5	R6
R1	0	0,384	0,365	0,462	0,421	0,388
R2	0	0	0,239	0,336	0,295	0,262
R3	0	0	0	0,317	0,276	0,244
R4	0	0	0	0	0,373	0,34
R5	0	0	0	0	0	0,299
R6	0	0	0	0	0	0

Dari hasil perhitungan tabel 8 kemudian masukan nilai edge pada graph berarah sesuai dengan sisi yang telah dijelaskan. Nilai edge yang telah dimasukan pada graph tersebut dapat di lihat pada gambar 1.

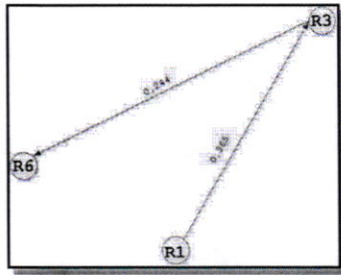


Gambar 1 Konstruksi Graph Berarah

Sumber (Noviarni, "Teori Grpah", 2 Januari 2015, [http://matematika-](http://matematika-diskrit-)

[universitas.blogspot.com/2011/10/graph.html](http://matematika-diskrit-universitas.blogspot.com/2011/10/graph.html))

Bila graph di atas merupakan graph berarah, untuk menentukan jalur terpendek antara simpul R1 dan R6. Maka menentukannya hendak menghitung jalur terpendek dari simpul R1 ke R6, mencapai simpul R6 dari simpul R1 dipilih nilai edge yang tekecil untuk dijadikan sebagai jalur lintasan terpendek. Sehingga lintasan terpendek yang dibentuk adalah R1-R3-R6. Berikut gambar 2 lintasan terpendek.



Gambar 2 Konstruksi Grpah Berarah Lintasan Terpendek

Maka hasil peringkasan yang disajikan kepada user dapat dilihat pada tabel 10. Hasil ringkasan ini merupakan contoh cara kerja dari aplikasi peringkasan dokumen.

Tabel 10 : Tabel Hasil Ekstraksi

No	Kalimat
R1	Striker pantai gading itu kemungkinan akan mendapat denda 100,000 poundsterling (rp 1,5 miliar) akibat mengkritik klub yang menurutnya tidak mendukung performanya di stamford bridge musim ini
R3	Nilai denda itu setara gaji sepekan yang diterima drogba.
R6	Atas aksinya ini klub telah memanggil drogba yang mungkin memilih mengakhiri kariernya bersama Chelsea yang dimulainya sejak 2004 setelah hengkang dari marseille.

IMPLEMENTASI DAN PENGUJIAN

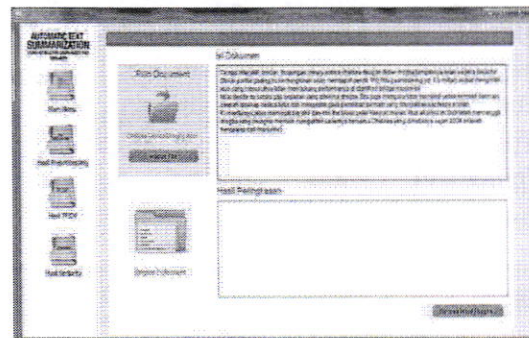
Pada aplikasi *automatic summary*, dilakukan pengujian *testing* yang berfungsi untuk menguji fungsionalitas *Get Summary* yang telah dibuat. Tujuan

Tabel 11: Hasil Pengujian

dari proses pengujian ini dilakukan untuk memeriksa apakah *Get Summary* ini telah melakukan fungsionalitas sesuai dengan perancangan pembuatan.

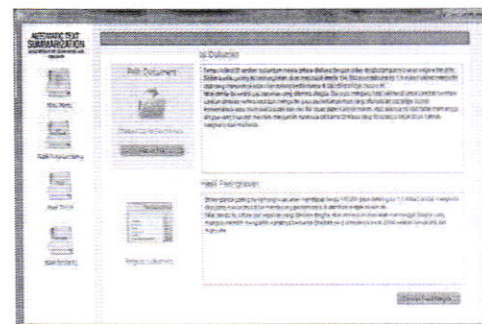
Pengujian Fungsi Peringkasan Dokumen

Untuk dapat menguji fungsionalitas *Summary*, dilakukan pengujian fungsionalitas ini apakah berfungsi dengan baik. diperoleh hasil pengujian seperti yang ditunjukkan pada Gambar 4 dan Gambar 5.



Gambar 4. Pengujian Fungsionalitas Get Summary

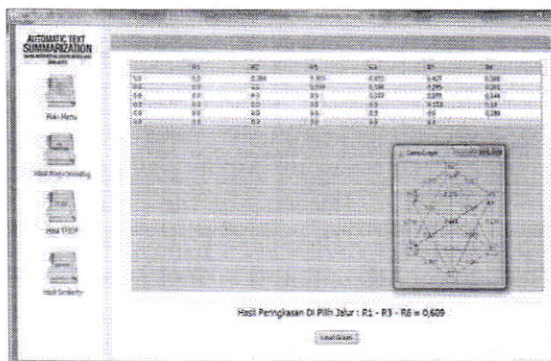
Dokumen yang diringkas pada proses ini adalah dokumen bertita olahraga dengan judul Chelsea Denda Drogba. Dengan menghasilkan isi ringkasan yang telah melalui proses tahap pengujian seperti yang ditunjukkan pada Gambar 5 dan tabel 11.



Gambar 5. Hasil Pengujian Fungsionalitas Get Summary

No	Dokumen	Jumlah kalimat sebelum diringkas	Jumlah kalimat sesudah diringkas	Waktu Meringkas (detik)	Proses Pengujian Peringkasan Dilakukan Sebanyak	Berhasil
1	Chelsea Denda Drogba.doc	6	3	2,58	8	5

Berdasarkan dari hasil pengujian fungsi *get summary*, diperoleh lintasan terpendek dengan hasil peringkasan yang dipilih yaitu jalur R1, R3, dan R6. Untuk menentukan jalur yang dipilih dari hasil perhitungan matriks kemiripan (*Similarity*) yang paling terkecil ditunjukkan pada Gambar 6.



Gambar 6. Hasil Peringkasan dipilih Jalur Lintasan Terpendek

Hasil perhitungan dari pengolahan bobot matriks kemiripan (*Similarity*) dan didapatkan pelintasan terpendek dimana arah lintasan terpendek memiliki nilai edge sebagai jalur. Nilai edge tersebut dipilih nilai yang terkecil untuk dijadikan sebagai jalur lintasan terpendek. Bahwa kalimat yang memiliki lintasan terpendek ditunjukkan pada tabel 12.

Tabel 12 Hasil Ekstraksi Dokumen

No	Kalimat
R1	Striker pantai gading itu kemungkinan akan mendapat denda 100,000 poundsterling (rp 1,5 miliar) akibat mengkritik klub yang menurutnya tidak mendukung performanya di stamford bridge musim ini
R3	Nilai denda itu setara gaji sepekan yang diterima drogba.
R6	Atas aksinya ini klub telah memanggil drogba yang mungkin memilih mengakhiri kariernya bersama Chelsea yang dimulainya sejak 2004 setelah hengkang dari marseille.

Dari gambaran yang didapat pada tabel tersebut hasil ringkasan mengacu pada banyaknya kemunculan kata dasar dihitung menggunakan TF-IDF untuk menentukan tingkat kemunculan kata dasar pada setiap kalimat, kemudian kalimat yang memiliki kandungan kata dalam judul dokumen berita olahraga "Chelsea Denda Drogba" akan diurutkan sesuai hasil bobot perhitungan matriks kemiripan (*Similarity*). Kalimat yang memiliki kesamaan paling besar antara isi kata pada dokumen yang berjudul "Chelsea Denda Drogba" adalah kalimat yang paling mendekati terhadap hasil ringkasan ekstraksi.

KESIMPULAN

Kesimpulan yang dapat diperoleh dari penelitian ini berdasarkan fakta pengujian adalah :

1. Hasil simpul pelintasan terpendek pada graph berarah dapat diimplementasikan pada kalimat untuk menghasilkan ringkasan ekstraksi.
2. Hasil perhitungan dari pengolahan bobot matriks kemiripan (*Similarity*) menghasilkan pelintasan terpendek dimana arah lintasan terpendek memiliki nilai edge sebagai jalur. Nilai edge tersebut dipilih yang terkecil untuk dijadikan sebagai jalur lintasan terpendek, bahwa kalimat yang memiliki lintasan terpendek ditunjukkan pada tabel 12.
3. Akurasi hasil peringkasan juga dipengaruhi oleh jumlah kalimat ringkasan; hasil pengujian secara umum menunjukkan jumlah kalimat sebelum diringkas berjumlah 6 dan hasil setelah melakukan peringkasan berjumlah 3, bahwa hasil pengujian ringkasan ditunjukkan pada tabel 11.

DAFTAR PUSTAKA

- [1] Iyan Mulyana, Sena Ramadona, Herfina, 2013 "Penerapan Terms Frequency-Inverse Document Frequency Pada Sistem Peringkasan Teks Otomatis Dokumen Tunggal Berbahasa Indonesia".
- [2] Aristoteles, 2013 "Penerapan Algoritma Genetika Pada Peringkasan Teks Dokumen Bahasa Indonesia".
- [3] Dyan, Rian, Agus (2012), "Algoritma Nazief dan Adriani".
- [4] _____, Ringkasan, <https://henipujiarti.wordpress.com/2011/04/27/ringkasan/>, diunduh pada tanggal 21 Desember 2014, pukul 13:00 WIB.

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

[6] Nikhil Garg, Benoit Favre, Korbinian Reidhammer, Dilek Hakkani-Tur "ClusterRank: A Graph Based Method for Meeting Summarization".

[7] Rada Mihalcea, Paul Tarau "A Language Independent Algorithm for Single and Multiple Document Summarization".

[8] Rajaraman, A. Ullman, J. D. (2011), Mining of Massive Datasets.