JEJARING (Jurnal Teknologi dan Manajemen Informatika) p-ISSN : 2527-3930, e-ISSN : 2614-5448
Volume 4 Nomor 1, Mei 2019 https://journal.uniku.ac.id/index.php/jejaring

PERINGKASAN TEKS OTOMATIS BERITA MENGGUNAKAN METODE MAXIMUM MARGINAL RELEVANCE

Robi Robiyanto[1], Nunu Nugraha[2], Ipnu Apriatna[3]
[1]Fakultas Teknik Universitas Islam Al-Ihya
Jl. Mayasih Cigugur Kuningan
[2,3]Fakultas Ilmu Komputer Universitas Kuningan
Jl.Cut Nyak Dhien no.36A kuningan
robi_robiyanto@gmail.com, nunu.nugraha@uniku.ac.id, ipnu.apriatna@gmail.com

ABSTRAK

Perkembangan teknologi internet berdampak pada semakin banyaknya situs berita berbahasa Indonesia dan menimbulkan ledakan informasi. Ini membutuhkan semua informasi yang dapat diakses dengan cepat dan tidak membutuhkan banyak waktu dalam membaca tajuk berita. Teknologi ringkasan teks otomatis menawarkan solusi untuk membantu kami mencari konten berita dalam bentuk deskripsi singkat (ringkasan). Studi ini dimulai dengan teks preprocessing lima tahap: menyelesaikan kalimat, melipat kasus, tokenizing, filtering, dan stemming. Proses selanjutnya adalah pembobotan tf-idf komputer, pembobotan query relevance dan bobot kesamaan. Ringkasan hasil ekstraksi menggunakan kalimat maksimum relevansi marjinal. Metode ekstraksi maksimum relevansi marjinal adalah metode yang digunakan untuk mengurangi redundansi dalam kalimat rangking ganda pada dokumen.

Kata kunci: Ringkasan, preprocessing teks, tf-idf, query relevance, similarity, maximum marginal relevance

ABSTRACT

Development of Internet technology affects the increasing number of Indonesian language news website and creates an explosion of information. It requires all the information that can be accessed quickly and does'n require a lot of time in reading a news headline. Automatic text summary technology offers a solution to help search us news content in the form of a brief description (summary). The study begins with a five-stage preprocessing text: solving sentence, case folding, tokenizing, filtering, and stemming. The next process is computer tf-idf weighting, weighting query relevance and similarity weights. Summary results from the extraction using the maximum sentence of marginal relevance. Marginal relevance maximum extraction method is the method used to reduce redundancy in multi ranking sentence on the document.

Keywords *: Summary, text preprocessing, tf-idf, query relevance, similarity, maximum marginal relevance*

PENDAHULUAN

Kebutuhan akan informasi yang cepat, tepat dan akurat merupakan suatu hal yang sangat penting di dalam setiap aspek kehidupan. Penggunaan teknologi informasi diharapkan dapat memenuhi semua kebutuhan informasi baik untuk pihak luar maupun dalam, sehingga pada akhirnya penggunaan teknologi informasi akan mempermudah dalam mengolah, menganalisis dan mengomunikasikan informasi yang *relevan* guna mengefisiensikan pengelolaan informasi menjadi sebuah berita yang lebih akurat.

Dengan banyaknya berita *online*, maka semakin banyak pula informasi berita yang diberikan. Oleh karena itu penulis untuk membantu pembaca berita mengurangi waktu membaca dan memberikan panduan cepat untuk memberikan akan informasi menarik. Konsep sederhana ringkasan adalah mengambil bagian penting dari keseluruhan isi dari artikel. Menurut Mani dan Maybury, ringkasan adalah mengambil isi yang paling penting dari sumber informasi yang kemudian menyajikannya kembali dalam bentuk yang lebih ringkas bagi penggunanya (Mani dan Maybury, 1999). Sedangkan menurut Hovy, ringkasan adalah teks yang dihasilkan dari sebuah teks atau banyak teks, yang mengandung isi informasi dari teks asli dan panjangnya tidak lebih dari setengah panjang teks aslinya (Hovy, 2001).

Berkaitan dengan berita merupakan media yang mudah digunakan untuk mendapatkan informasi.

Dalam Kamus Besar Bahasa Indonesia Departemen Pendidikan Nasional Balai Pustaka (2001: 04), Mendefinisikan berita yaitu cerita atau keterangan mengenai kejadian atau peristiwa yang hangat, kabar, laporan, pemeberitahuan, pengumuman. Menurut M. Assegaf (Assegaf, 2010: 47) mengatakan, berita adalah laporan tentang fakta atau ide yang termasa, yang dipilih staf redaksi suatu media. Karena berita mengandung informasi penting dan menarik perhatian maka penyajian berita harus mempertimbangkan aspek isi dan waktu, karenanya kecepatan berita patut menjadi perhatian. Ada istilah *" tiada hari tanpa berita "*, hal ini mengindikasikan bahwa adanya kebutuhan masyarakat akan sebuah berita dalam aktivitas kesehariannya.

Di kutip dari jurnal dengan judul "*Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance".* Kesimpulan yang didapat dari penelitian adalah : Merupakan metode yang digunakan untuk mengurangi redudansi dalam perangkingan kalimat pada multi dokumen. Data uji coba diambil dari surat kabar berbahasa Indonesia online.

Seiring dengan kemajuan teknologi dan informasi, tekologi internet menjadi basis penting dalam pemanfaatan media *online.* Keunggulan media *online* adalah informasi bersifat *up to date*, *real time,* dan praktis.

Berdasarkan latar belakang di atas maka penulis mengambil tema **"** PERINGKASAN TEKS OTOMATIS BERITA MENGGUNAKAN METODE MAXIMUM MARGINAL RELEVANCE". Dimana ini bisa berfungsi mempermudah dan mempersingkat waktu ketika user mengakses berita terupdate.

Tinjauan Pustaka

2.1 Konsep Dasar Peringkasan Text Otomatis

Peringkasan Teks Otomatis *(Atomatic Text Summarization)* adalah bentuk ringkasan dari dokumen yang bertujuan untuk menghilangkan term yang dianggap tidak relevan atau redudan dengan menjaga inti makna dari dokumen, sehingga meskipun dokumen tadi memiliki volume yang besar akan tetapi para pengguna dokumen dapat memahami inti makna dengan cepat dan benar.

Proses peringkas dokumen adalah sebuah proses untuk melakukan pengurangan volume dokumen menjadi lebih ringkas, dengan cara mengambil inti dokumen dan membuang term yang dianggap tidak penting tanpa mengurangi makna sebuah dokumen, terdapat dua tipe pembuatan suatu ringkasan yang mengambil bagian terpenting dari teks aslinya yaitu abstrak dan ekstrak. (jurnal *nasional teknologi informasi & komunikasi, 2013)*

1. Abstrak menghasilkan sebuah interprestasi terhadap teks aslinya, dimana sebuah kali mat akan ditransformasikan menjadi kalim at yang lebih singkat.
2. Ekstraksi merupakan ringkasan teks yang di peroleh dengan menyajikan kembali bagian t ulisan yang dianggap topik utama tulisan d engan bentuk yang lebih disederhanakan.

2.2 Konsep dasar Maximum Marginal Relevance

Algoritma maximum marginal relevance (MMR) adalah algoritma yang menggunakan metode ekstrasi ringkasan *(extractive summary)* yang digunakan untuk dokumen tunggal atau multi dokumen dengan menghitung kesamaan antar bagian teks.Cara kerja algoritma MMR meringkas kalimat dengan menghitung kesamaan (*similarity*) antar bagian kalimat. Cara kerja algoritma ini juga mengkombinasikan matrik *cosine similarity* untuk merangking kalimat – kalimat sebagai tanggapan pada query yang diberikan oleh user. Metode *Maximal Marginal Relevance* untuk pemilihan kalimat atau unit teks lain yang memepertimbangkan aspek kerelevanan kalimat dengan *query* dan keterbaruan informasi (*Carbonell & Goldstein* 1998)

Ide dasar dari MMR ini yaitu memberikan penambahan nilai bagi kalimat yang relevan dan memberikan pengurangan nilai redundansi informasi antara kalimat tersebut dengan kalimat lain yang telah terpilih. Sebuah kalimat dikatakan memiliki *marginal relevance* yang tinggi jika kalimat tersebut relevan terhadap isi dari kalimat dan mempunyai kesamaan bobot *term* maksimum dibandingkan dengan *query.*Peringkasan kalimat dengan tipe ekstraktif, nilai akhir diberikan pada kalimat Si dalam MMR dihitung dengan persamaan.Dimana *MMR* adalah

himpunan kalimat relevan yang dipilih, *Sim1* dan *Sim*2 adalah matriks dari kesamaan kalimat.Untuk mendapatkan kumpulan kalimat yang relevan ini, kita memberikan peringkat pertama pada kalimat yang relevan dengan memberikan presentase tertentu dari kalimat tersebut.

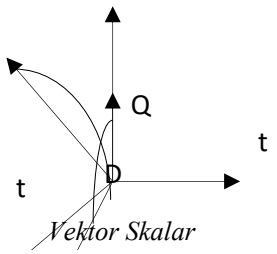
Si adalah kalimat di dokumen, sedangkan *S'* adalah kalimat yang telah dipilih atau telah diekstrak (*Shasha Xie*, 2010).Koefisien λ digunakan untuk mengatur kombinasi nilai untuk memberi penekanan bahwa kalimat tersebut relevan dan untuk mengurangi redudansi. Pada penelitian ini, *Sim1* dan *Sim2* merupakan dua fungsi *similarity* yang merepresentasikan kesamaan kalimat pada seluruh dokumen dan memilih masing-masing kalimat untuk dijadikan ringkasan.*Sim1* adalah matrik *similarity* kalimat *Si* terhadap *query* yang diberikan oleh user sedangkan *Sim2* adalah matrik *similarity* kalimat *Si* terhadap kalimat yang telah diekstrak sebelumnya (*Shasa Xie*, 2010).

Nilai parameter λ adalah mulai dari 0 sampai dengan 1 (range [0,1]). Pada saat parameter maka nilai *MMR* yang diperoleh akan cenderung relevan terhadap dokumen asal. Ketika λ = 0 maka nilai *MMR* yang diperoleh cenderung relevan terhadap kalimat yang diekstrak sebelumnya. Oleh sebab itu sebuah kombinasi linier dari kedua kriteria dioptimalkan λ ketika nilai t terdapat pada interval [0,1]. Untuk peringkasan *small* dokumen, seperti pada berita (*news*), menggunakan nilai parameter λ = 0.7 atau λ = 0.8 , karena akan menghasilkan ringkasan yang baik (*Jade Goldstein*, 2008).

Untuk mendapatkan hasil ringkasan yang relevan maka harus menetapkan nilai λ ke nilai yang lebih dekat dengan λ . Kalimat dengan nilai *MMR* yang tertinggi akan dipilih berulang kali ke dalam ringkasan sampai tercapai ukuran ringkasan yang diinginkan

2.3 *Cosine Similarity*

Cosine similarity digunakan untuk menghitung pendekatan relevansi *query* terhadap dokumen. Penentuan relevansi sebuah *query* terhadap suatu dokumen dipandang sebagai pengukuran kesamaan antara vektor *query* dengan vektor dokumen. Semakin besar nilai kesamaan vektor *query* dengan vektor dokumen maka *query* tersebut dipandang semakin relevan dengan dokumen. Saat mesin menerima *query*, mesin akan membangun sebuah vektor Q berdasarkanistilah-istilah pada *query* dan sebuah vektor D berukuran t untuk setiap dokumen. Pada umumnya *cosine similarity(CS)* dihitung dengan rumus *cosinemeasure* (Grossman, 1998).



$$CS(b_1, b_2) = \frac{\sum_{t=1}^{n} W_{t,b_1} W_{t,b_2}}{\sqrt{\sum_{t=1}^{n} W^2_{t,b_1} \sum_{j=1}^{n} W^2_{t,b_2}}}$$

Keterangan :

T : *term* dalam kalimat

$W_{t,b1}$: bobot term t dalam blok b_1

$W_{t,b2}$: bobot term t dalam blok b_2

Dibawah ini terdapat empat jenis model *Cos ine Similarity* yaitu :

1. Model *Dot Product :*
   CS(Q,Di) = (Di)(Q)
2. Model *Dice :*
   S< (Q,Di)= 2(Di)(Q)/(Di2)(Q2)
3. Model *Jaccard :*
   CS(Q,Di) = (DiQ)/((Di)2 + Q2 - |Di|Q|)

Model Cosine Similarity Larson dan Herast :

$$CosSim(dj, q) = \frac{\vec{dj}.\vec{q}}{|\vec{dj}|.|q|} = \frac{\sum_{i=1}^{t}(W_{ij} \cdot W_{iq})}{\sqrt{\sum_{i=1}^{t} Wij^2 \cdot Wiq^2}}$$

2.4 Pembobotan TF-IDF

Pembobotan dapat diperoleh berdasarkan jumlah kemunculan suatu *term* dalam sebuah dokumen *term frequency* (*tf*) dan jumlah kemunculan *term* dalam koleksi dokumen *inverse document frequency* (*idf*). Bobot suatu istilah semakin besar jika istilah tersebut sering muncul dalam suatu dokumen dan semakin kecil jika istilah tersebut muncul dalam banyak dokumen (Grossman, 1998). Nilai *idf* sebuah *term* (kata) dapat dihitung menggunakan persamaan sebagai berikut:

D adalah jumlah dokumen yang berisi *term* (*t*) dan *dfi* adalah jumlah kemunculan (frekuensi) *term* terhadap D. Adapun algoritma yang digunakan untuk

menghitung bobot (W) masing masing dokumen terhadap kata kunci (*query*), yaitu:

$$w_{d,t} = tf_{d,t} * IDF_t$$

Keterangan :
d = dokumen ke–d
t = *term* ke–t dari kata kunci
tf = *term* frekuensi/frekuensi kata
W = bobot dokumen ke–d terhadap ter m ke–t

Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses pengurutan (*sorting*) dimana semakin besar nilai W, semakin besar tingkat kesamaan (*similarity)* dokumen tersebut terhadap kata yang dicari, demikian pula sebaliknya.

2.5 *Text Preprocessing*

Text preprocessing adalah tahapan untuk mempersiapkan teks menjadi data yang akan diolah di tahapan berikutnya.Inputan awal pada proses ini adalah berupa dokumen. *Text preprocessing* pada penelitian ini terdiri dari beberapa tahapan, yaitu: proses pemecahan kalimat, proses *case folding,* proses *tokenizing* kata*,* proses *filtering,* dan proses *stemming.*

2.5.1 Pemecahan Kalimat

Memecah dokumen menjadi kalimat – kalimat merupakan langkah awal tahapan *text preprocessing*. Pemecahan kalimatyaitu proses memecah string teks dokumenyang panjang menjadi kumpulan kalimat-kalimat. Dalam memecah dokumen menjadi kalimat-kalimat menggunakan fungsi **split()**, dengan tanda titik ".", tanda tanya "?" dan tanda tanya "!" sebagai delimiter untuk memotong string dokumen.

2.5.2 *Proses case folding*

Case folding adalah tahapan proses mengubah semua huruf dalam teks dokumen menjadi huruf kecil, serta menghilangkan karakter selain a-z.

2.5.3 *Tokenizing*

Tokenizing adalah proses pemotongan string input berdasarkan tiap kata yang menyusunnya. Pemecahan kalimat menjadi kata-kata tunggal dilakukan dengan men-*scan* kalimat dengan pemisah (*delimiter*) *white space* (spasi, tab, dan *newline*).

2.5.4 *Filtering*

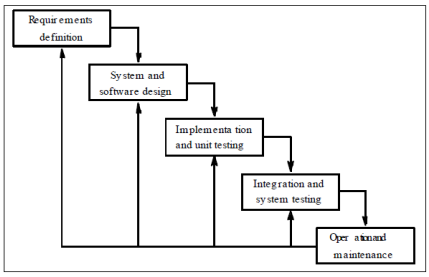
Filtering merupakan proses penghilangan *stopword*. *Stopword* adalah kata – kata yang sering kali muncul dalam dokumen namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Didalam bahasa Indonesia *stopword* dapat disebut sebagai kata tidak penting misalnya "di", "oleh", "pada", "sebuah", "karena" dan lain sebagianya.

2.5.5 *Stemming*

Stemming merupakan proses mencari akar (*root)* kata dari tiap *token* kata yaitu dengan pengembalian suatu kata berimbuhan ke bentuk dasarnya (*stem*) (Tala,2003).

2.6 Metode Pengembangan Sistem

Model pertama yang diterbitkan untukproses pengembangan perngkat lunak diambil dari proses rekayasa perangkat lain. Model ini di ilustrasikan berkat penurunan dari fase ke pase yang lain, model ini dikenal sebagai Model Air Terjun (*Waterfall Method)* atau siklus hidup perangkat lunak. Tahapan – tahapan utama dari model ini memetakan kegiatan – kegiatan pengembangan dasar yaitu :



- *Requirements analysis and definition*: Mengumpulkan kebutuhan secara lengkap kemudian kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.
- *System and software design*: Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.
- *Implementation and unit testing* : desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.
- *Integration and system testing*: Penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing).
- *Operation and maintenance*: mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau

p-ISSN : 2527-3930, e-ISSN : 2614-5448
https://journal.uniku.ac.id/index.php/jejaring

perubahan karena adaptasi dengan situasi sebenarnya. Kekurangan yang utama dari model ini adalah kesulitan dalam mengakomodasi perubahan setelah proses dijalani. Fase sebelumnya harus lengkap dan selesai sebelum mengerjakan fase berikutnya.

Model Watterfall ((Iyan Sammerville, Software Enginering (Rekayasa Perangkat Lunak)

2.7 *Unified Model Langue (UML)*

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. (Sri, Romi: 2003).

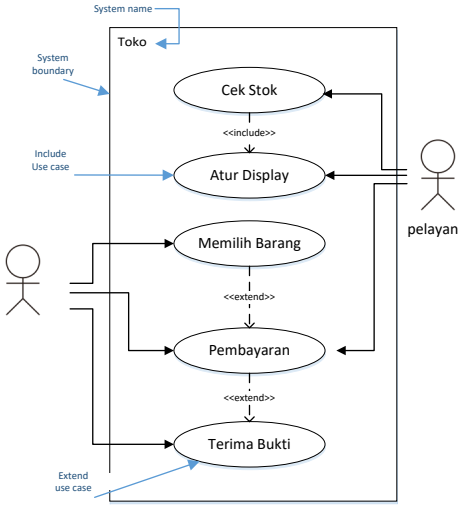
UML digunakan untuk membuat sebuah rancangan dalam pembuatan sebuah apikasi piranti lunak yang dapat berjalan dalam piranti keras, sistem operasi dan jaringan apapun. UML juga merupakan sebuah rancangan dalam pembuatan aplikasi yang nantinya dapat diimplementasikan dengan bahasa pemrograman apapun.

UML merupakan rancangan yang berorientasi objek, dengan penggunaan *class, objek* dan *operation* dalam konsep dasarnya. Dengan demikian rancangan UML lebih cocok diimplementasikan dengan menggunakan bahasa pemrograman berorientasi objek, seperti Java, C++, C# atau VB.NET. Akan tetapi UML juga tetap dapat digunakan untuk merancang aplikasi prosedural.

2.7.1 Diagram Use Case

Use case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang dibuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seseorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. (Sri, Romi: 2003).

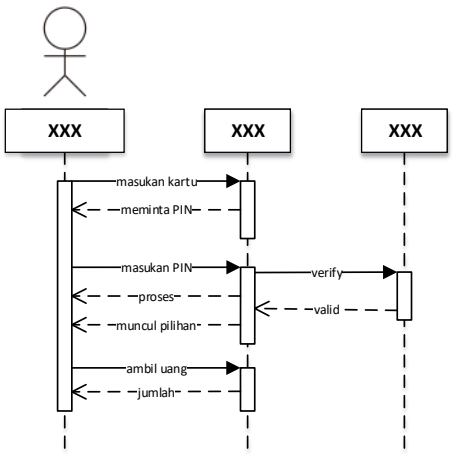
Contoh Use Case :



Contoh *Use Case Diagram*

2.7.2 *Diagram Sequence*

Sequence diagram menggambarkan interaksi antara objek di dalam dan di sekitar sistem (termasuk pengguna, *display,* dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). (Sri, Romi: 2003).



Contoh *Sequence Diagram*

p-ISSN : 2527-3930, e-ISSN : 2614-5448
https://journal.uniku.ac.id/index.php/jejaring

ANALISIS DAN DESAIN

3.1 Analisis Sistem

3.1.1 Sistem yang berjalan

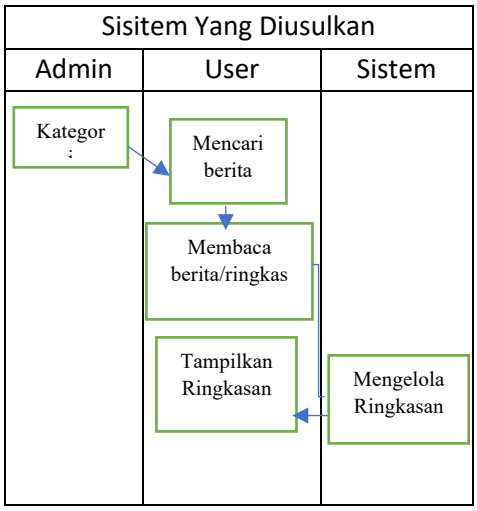
Gambar 3.1 Flowmap yang Sedang Berjalan

Selama ini untuk mendapatkan informaasi dari sebuah berita diharuskan untuk membaca keseluruhan dari isi berita tersebut, hal ini tentu akan memakan waktu mengingat penyajian berita harus mempertimbangkan aspek isi dan waktu. Membaca berita secara menyeluruh dalam sebuah artikel belum tentu membuat pembaca mengetahui ringkasan inti dari berita tersebut. Untuk itu di perlukan sbuah sistem yang dapat meringkas sebuah berita sehingga dapat mudah di mengerti oleh pembaca.

3.1.2 Sitem Yang Diusulkan

Dengan sistem yang diusulkan, diharapkan dapat membantu pembaca untuk mengetahui ringkasan inti dari suatu berita, selain itu diharapkan sistem ini dapat membantu pembaca dalam pencarian isi berita secara *relevance* dengan cepat dan tidak membutuhkan banyak waktu dalam mencari isi berita.

Adapun usulan sistem yang akan dibangun.

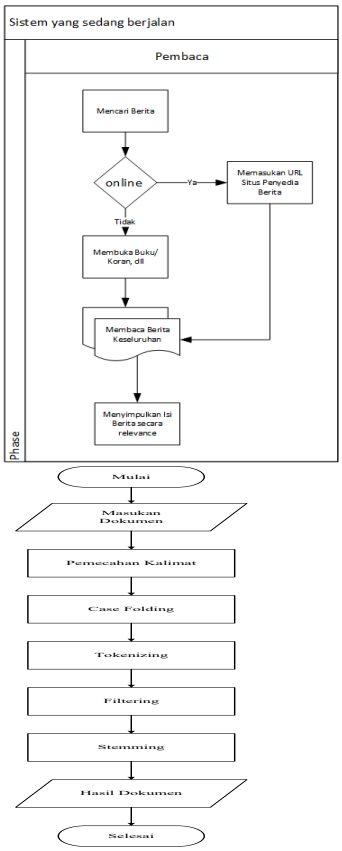


Flowmap Usulan

3.1.3 Algoritma Metode *Maximum Marginal Relevance*

3.1.3.1 Proses *Text Preprosessing*

Text preprocessing adalah tahapan untuk mempersiapkan teks menjadi data yang akan diolah ditahapan berikutnya. Inputan awal pada proses ini adalah berupa dokumen. Text preprocessing pada penelitian ini adalah sebagai berikut:



Flowchart Preprocessing

1. Proses Pemecahan Kalimat

   Pemecahan kalimat yaitu proses memecah string teks dokumen yang panjang menjadi kumpulan kalimat- kalimat.

   Contoh Soal

Tabel 3.3 Contoh Kalimat

Contoh Soal	
Query	soal biologi sma
Cepat menyelesaikan soal kimia SMA, terampil mandiri menyelesaikan soal biologi untuk SMA, dan menjadi juara olimpiade SMA.	

Tabel 3.4 Pemecahan Kalimat

Pemecahan Kalimat	
Query	soal biologi sma
Dokumen 1	Cepat menyelesaikan soal kimia SMA,

p-ISSN : 2527-3930, e-ISSN : 2614-5448
https://journal.uniku.ac.id/index.php/jejaring

Dokumen 2	terampil mandiri menyelesaikan soal biologi untuk SMA,
Dokumen 3	dan menjadi juara olimpiade SMA.

2. Proses *Case Folding*
   Tahapan proses mengubah semua huruf dalam teks dokumen menjadi huruf kecil, serta menghilangkan karakter selain a-z.

Tabel 3.5 Proses *Case Folding*

Proses *Case Folding*	
Query	soal biologi sma
Dokumen 1	cepat menyelesaikan soal kimia sma,
Dokumen 2	terampil mandiri menyelesaikan soal biologi untuk sma,
Dokumen 3	dan menjadi juara olimpiade sma.

3. Proses *Tokenizing* Kata
   Proses pemotongan string input berdasarkan tiap kata yang menyusunnya.

Tabel 3.6 Proses *Tokenizing*

Proses *Tokenizing*		
Query	soal biologi sma	
Dokumen 1	Dokumen 2	Dokumen 3
cepat	Terampil	Dan
menyelesaikan	Mandiri	Menjadi
soal	Menyelesaikan	Juara
kimia	soal biologi	olimpiade
Sma	untuk	Sma
	Sma	

4. Proses Pembuangan *Filtering*
   Merupakan proses penghilangan *stopword*. *Stopword* adalah kata – kata yang sering kali muncul dalam dokumen namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Didalam bahasa Indonesia *stopword* dapat disebut sebagai kata tidak penting misalnya “di”, ”oleh”, “pada”, ”sebuah”, ”karena” dan lain sebagianya.

Tabel 3.7 Proses Pembuangan *Filtering*

Proses Pembuangan *Filtering*		
Query	soal biologi sma	
Dokumen 1	Dokumen 2	Dokumen 3
cepat	Terampil	menjadi
menyelesaikan	Mandiri	Juara
soal	menyelesaikan	olimpiade
kimia	soal	Sma
Sma	Biologi	
	Sma	

5. Proses *Stemming*
   Merupakan proses mencari akar (*root)* kata dari tiap *token* kata yaitu dengan pengembalian suatu kata berimbuhan ke bentuk dasarnya (*stem*).

Tabel 3.8 Proses *Stemming*

Proses *Stemming*		
Query	soal biologi sma	
Dokumen 1	Dokumen 2	Dokumen 3
cepat	Terampil	Jadi
selesai	Mandiri	Juara
soal	Selesai	olimpiade
kimia	soal	Sma
Sma	Biologi	
	Sma	

3.1.3.2 Pembobotan TF-IDF kata

Pembobotan dapat diperoleh berdasarkan jumlah kemunculan suatu *term* dalam sebuah dokumen *term frequency* (*tf*) dan jumlah kemunculan *term* dalam koleksi dokumen *inverse document frequency* (*idf*).

$$\text{IDF} = \log(\frac{D}{idf})$$

1. Menghitung *Document frequency (df)* adalah merupakan frekuensi kemunculan term (t) pada dokumen (d)

p-ISSN : 2527-3930, e-ISSN : 2614-5448
https://journal.uniku.ac.id/index.php/jejaring

Tabel 3.9 Ilustrasi menghitung TF *(trem frekuensi)*

Ilustrasi menghitung TF *(trem frekuensi)*							
Query	Q	Dokumen 1	D1	Dokumen 2	D2	Dokumen 3	D3
Soal	1	cepat	1	Terampil	1	Jadi	1
Biologi	1	selesai	1	Mandiri	1	Juara	1
Sma	1	soal	1	Selesai	1	Olimpiade	1
		kimia	1	soal	1	Sma	1
		Sma	1	Biologi	1		
				Sma	1		

Tabel 3.10 Menghitung DF *(Document Frekuensi)*

Menghitung DF (Document Frekuensi)					
Trem	Q	D1	D2	D3	DF
cepat	0	1	0	0	1
selesai	0	1	1	0	2
soal	1	1	1	0	3
kimia	0	1	0	0	1
sma	1	1	1	1	4
terampil	0	0	1	0	1
mandiri	0	0	1	0	1
biologi	1	0	1	0	2
jadi	0	0	0	1	1
juara	0	0	0	1	1
olimpiade	0	0	0	1	1

2. Menghitung invers document frequency (idf)

$$\text{IDF} = \log(\frac{D}{df})$$

Tabel 3.11 Menghitung *invers document frequency* (IDF)

Menghitung invers document frequency (IDF)						
Trem	Q	D1	D2	D3	DF	IDF
cepat	0	1	0	0	1	0,60206
selesai	0	1	1	0	2	0,30103
soal	1	1	1	0	3	0,12494
kimia	0	1	0	0	1	0,60206
sma	1	1	1	1	4	0
terampil	0	0	1	0	1	0,60206
mandiri	0	0	1	0	1	0,60206
biologi	1	0	1	0	2	0,30103
jadi	0	0	0	1	1	0,60206
juara	0	0	0	1	1	0,60206
olimpiade	0	0	0	1	1	0,60206

3. untuk menghitung bobot (W) masing masing dokumen terhadap kata kunci (*query*)

$$W_{d,t} = tf_{d,t} * IDF_t$$

Tabel 3.12 Menghitung Bobot W

Menghitung bobot (W)										
Trem	Q	D1	D2	D3	DF	IDF	W			
							Q	D1	D2	D3
cepat	0	1	0	0	1	0,60206	0	0,6	0	0
selesai	0	1	1	0	2	0,30103	0	0,3	0,3	0
soal	1	1	1	0	3	0,12494	0,125	0,12	0,12	0
kimia	0	1	0	0	1	0,60206	0	0,6	0	0
sma	1	1	1	1	4	0	0	0	0	0
terampil	0	0	1	0	1	0,60206	0	0	0,6	0
mandiri	0	0	1	0	1	0,60206	0	0	0,6	0
biologi	1	0	1	0	2	0,30103	0,301	0	0,3	0
jadi	0	0	0	1	1	0,60206	0	0	0	0,6
juara	0	0	0	1	1	0,60206	0	0	0	0,6
olimpiade	0	0	0	1	1	0,60206	0	0	0	0,6

Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses pengurutan (*sorting*) dimana semakin besar nilai W, semakin besar tingkat kesamaan (*similarity)* dokumen tersebut terhadap kata yang dicari, demikian pula sebaliknya.

3.1.3.3 Pembobotan *similarity* kalimat

Semakin besar tingkat kesamaan (*similarity)* dokumen tersebut terhadap kata yang dicari, demikian pula sebaliknya.

- Bobot W untuk D1 = 0.12 + 0 + 0 = 0.12
- Bobot W untuk D2 = 0.12 + 0 + 0.6 = 0.72
- Bobot W untuk D3 = 0 + 0 + 0 = 0

Maka dari itu tingkat kesamaan yang dicari terdapat di D2 (Document 2)

Tabal 3.13 Hasil Bobot W

Dokumen 2	terampil mandiri menyelesaikan soal biologi untuk sma,

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

4.1.1 Implementasi Program

Implementasi yang dilakukan antara lain adalah menerapkan perancangan antar muka ke dalam bentuk aplikasi, perancangan basis data ke dalam bentuk tabel database, pembuatan kode program dan sebagainya.

4.1.2 Perangkat Lunak yang Digunakan

Sistem operasi yang digunakan pada pembuatan aplikasi ini adalah Windows 7 Ultimate. Sedangkan bahasa pemrograman yang digunakan untuk membangun perangkat lunak ini adalah PHP. Database dibangun menggunakan MySQL. Aplikasi pemograman yang penulis gunakan adalah Notepad++ .

p-ISSN : 2527-3930, e-ISSN : 2614-5448
https://journal.uniku.ac.id/index.php/jejaring

4.1.3 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras terpenting untuk membuat aplikasi ini dijelaskan dalam tabel 4.1 sebagai berikut :

Tabel 4.1 Spesifikasi Perangkat Server

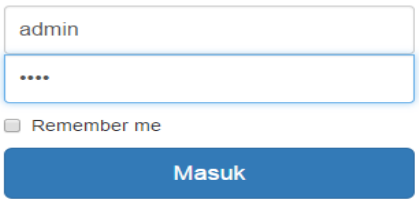
Jenis	Perangkat Keras
Processor	Intel® Core™ i3-4010U CPU @ 1.70GHz
RAM	2048MB RAM
Hardisk	320 GB
VGA Card	Intel® HD Grafik Familly 32 MB
Resolusi Display	1366 x 768

4.1.4 Implementasi Sistem

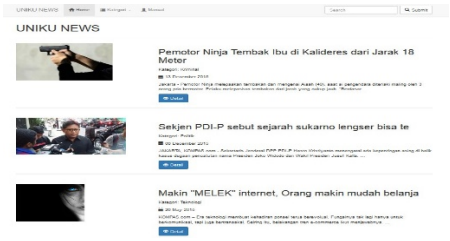
Tahap implementasi sistem merupakan tahap penerapan sistem agar dapat dioperasikan secara optimal sesuai kebutuhan. Proses implementasi dilakukan sebagai hasil akhir dari desain aplikasi peringkasan berita dengan menggunakan metode *Maximum Marginal Relevance.*

4.1.5 Implementasi Antar Muka

Pada tahap implementasi ini antar muka yang dibuat pada tahap perancangan, diimplementasikan menjadi bentuk aplikasi mobile yang dibangun dengan menggunakan perangkat lunak yang dijelaskan pada implementasi program. Adapun bentuk aplikasi hasil implementasi tersebut disajikan sebagai berikut :

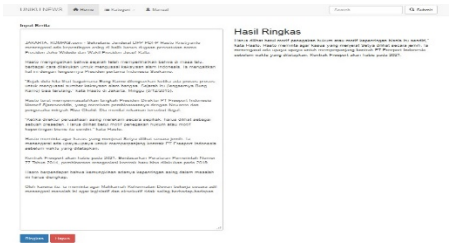


Login

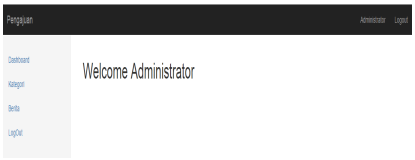


Antarmuka Menu Berita

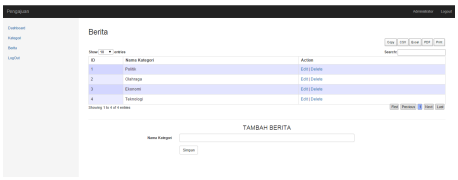
Antarmuka Menu Detail Berita



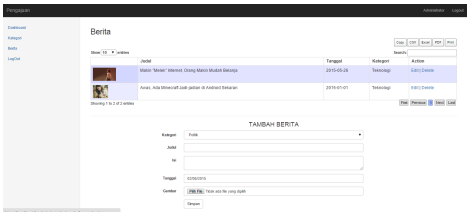
Antarmuka Menu Ringkas *(MMR)*



Antarmuka Menu Dashboard Admin



Antarmuka Menu Kategori



Antarmuka Menu Tambah Berita

KESIMPULAN

Berdasarkan hasil penelitian dan pengujian aplikasi peringkasan berita menggunakan metode Maximum Marginal Relevance yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut :

p-ISSN : 2527-3930, e-ISSN : 2614-5448
https://journal.uniku.ac.id/index.php/jejaring

1. Sistem yang dibangun dapat membantu dalam menarik kesimpulan dari konten berita yang disediakan.
2. Metode maximum marginal relevance dapat digunakan untuk meringkas single dokumen secara otomatis dengan menggunakan judul artikel berita sebagai query, hasil dari uji coba yang dilakukan menghasilkan rata-rata recall 60%, precision 77%, dan f-measure 66% berdasarkan perbandingan sistem dengan ringkasan manual.

DAFTAR PUSTAKA

[1.] (Widhiarta., S.Kom, Pemrograman Internet 1, STIMIK AMIKOM Yogyakarta, Yogyakarta, 2008.)

[2.] (Akhmad Sofwan. Komunitas eLearning IlmuKomputer, STMIK Budi Luhur, Copyright © 2003-2007 IlmuKomputer.Com)

[3.] (Triswansyah Yuliano, Pengenalan PHP, IlmuKomputer.com, Copyright © 2003-2007)

[4.] Donald Bell, UML basics: An introduction to the Unified Modeling Language, IBM Rational Software, 2003)

[5.] (Al Bahra Bin Ladjamudin, Siklus hidup *(Life Cycle)* dengan model-model *waterfall* 2006: 18)

[6.] (Arief Ramdhan, Seri Pelajaran Komputer Internet dan Aplikasinya, Elex Media Komputindo, Jakarta, 2005)

[7.] (Sr. Maria Assumpta Rumanti, Dasar – Dasar Public Relations, Grasindo, Jakarta 2002)

[8.] (Budi Permana, S.Kom, Cepat Mahir Bahasa Pemrogramanepat Mahir Bahasa Pemrograman PPH, IlmuKomputer.com, Copyright © 2003-2013)

[9.] (Roger S.Pressman, Ph.D, Pengertian White box- Black box 2010)

[10.] (Rochayah Machali, Pedoman Bagi Penerjemah, Mizan Pustaka, Bandung 2009)

[11.] (Krismiaji, dalam bukunya yang berjudul Sistem Informasi Akuntansi (2010:71)

[12.] Hovy, E. and Lin, C. Y. (1999). Automated text summarization in summarist. In Mani, I. and Maybury, M. T., editors, Advances in Automatic Text Summarization, pages 81-94. MIT Press

[13.] Xie, Shasha. 2010. Automatic Extractive Summarization Meeting Corpus. Dissertation. Dallas: The University of Texas at Dallas.

[14.] Grossman, D., dan Ophir, F. 1998. Information Retrieval: Algorithm and Heuristics. Kluwer Academic Publisher.

[15.] Golstein, Jade and Carbonell, Jaime. 1998. Summarization: Using MMR for Diversity Based-Reranking and Evaluating Summaries. Langauge Technologies Istitute. Carnegie Mellon University.

[16.] Erwin A.H., Muhammad. 2005. Sistem Pengidentifikasi Otomatis Pokok Kalimat Suatu Paragraf Dalam Dokumen Ekspositori Dengan Model Ruang Vektor. Laboratorium Pemrograman dan Informatika Teori. Yogyakarta: Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

[17.] (*Sparck dan Galliers*, konsep Dasar Metode 1996),

[18.] (Iyan Sammerville, *Software Enginering* (Rekayasa Perangkat Lunak) Edisi 6 Jilid 1, Erlangga, Jakarta, 2003)