Sistem Peringkas Berita Otomatis berbasis *Text Mining* menggunakan *Generalized Vector Space Model*: Studi Kasus Berita diambil dari Media Massa *Online*

Budhi Kurniawan Wangsa[1], Darmawan Utomo[2] , Saptadi Nugroho[3]

Program Studi Sistem Komputer,
Fakultas Teknik Elektronika dan Komputer
Universitas Kristen Satya Wacana, Salatiga
[1]budhi.wangsa.ftje@gmail.co.id, [2]darmawan@staff.uksw.edu, [3]saptadi_nugroho@yahoo.com,

Ringkasan

Makalah ini akan membahas mengenai sistem yang memiliki fungsi utama membentuk ringkasan dari dokumen secara otomatis dengan menggunakan metode yang bersifat *text mining*. Sistem akan menggunakan berita sebagai dokumen sumber yang akan dibentuk ringkasannya. Sistem ini bersifat *desktop based* dan menggunakan internet sebagai sumber pencarian dokumen berita. Pencarian akan menggunakan *focused crawler* dan bersifat *text mining* yakni hanya diambil teks beritanya saja. Metode *generalized vector space model* (GVSM) adalah metode untuk menilai tingkat kemiripan tiap kalimat terhadap suatu topik dokumen. Dengan metode GVSM ini dapat diketahui kalimat mana yang lebih berbobot terhadap suatu dokumen sehingga dapat dilakukan peringkasan dengan memperhatikan tingkat kemiripan kalimat. Dari hasil perancangan dan pengujian didapat tingkat kesuksesan *focused crawler* sebesar 53% sementara dari kuesioner hasil ringkasan menggunakan metode GVSM dinilai secara rata-rata 2,71 dari skala 1-4 oleh empat puluh orang responden. Sistem mampu meringkas sebanyak 754 berita dari 797 berita yang didapat atau sekitar 94% dari berita yang didapat. Sehingga didapat kesimpulan bahwa sistem yang dirancang mampu mencari berita secara terarah sekaligus meringkas berita dengan hasil yang dapat diterima.

Kata Kunci: *focused crawler*, *generalized vector space model*, peringkasan otomatis.

1. Latar Belakang

Dunia informasi yang semakin tidak mengenal batas ruang dan waktu membuat semua orang bisa mengakses informasi kapan saja dan di mana saja. Salah satu informasi yang sangat banyak dicari adalah berita. Dalam prosesnya pengguna yang mengakses berita yang ingin dicari melalui internet biasa menggunakan bantuan mesin pencari seperti Google, Bing, dan berbagai mesin pencari lainnya. Akan tetapi hasil yang didapat hanya berupa halaman *web* yang mengandung informasi berita secara umum seperti judul dan *link* menuju berita tersebut.

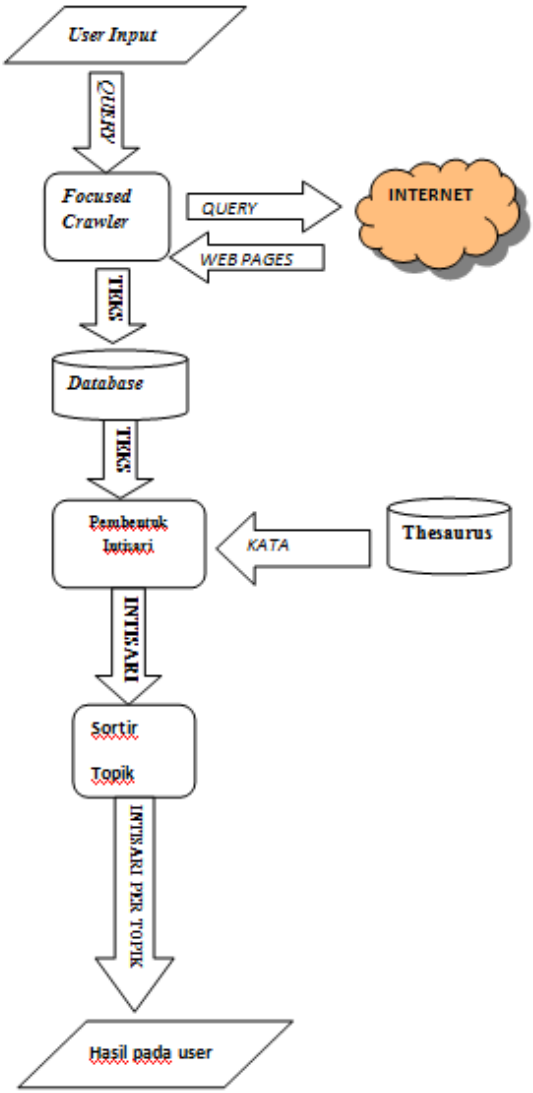
Ringkasan adalah suatu pokok permasalahan dari suatu paragraf ataupun suatu dokumen [1]. Dengan melihat sebuah ringkasan saja seorang pembaca dapat memahami garis besar dari suatu berita tanpa perlu membaca secara detil berita tersebut. Secara

umum pembaca dari suatu berita hanya fokus melihat pada garis besar suatu berita yang dicari sebelum melihat lebih detil lagi berita hasil pencarian [2].

Pembentukan ringkasan secara otomatis bisa dilakukan dengan berbagai macam metode. Metode yang paling umum dilakukan adalah dengan mencari pokok pikiran utama atau kalimat utama dari setiap paragraf berita. Salah satu cara pencarian dilakukan dengan melakukan perhitungan menggunakan metode *natural language processing* (NLP) yaitu GVSM. Inti utama pada metode GVSM adalah melihat nilai kemiripan dokumen terhadap suatu *query* pencarian. Pembobotan nilai dilakukan dengan melakukan pengecekan istilah satu persatu dimana istilah tersebut akan dicari jumlahnya dalam dokumen tersebut. Semakin banyak ditemukan maka istilah tersebut akan mendapat bobot yang lebih besar [3].

2. Perancangan Sistem

Perancangan terdiri dari perancangan *focused crawler* dan perancangan peringkas berita otomatis.



Gambar 1. Blok diagram sistem yang dirancang.

Dari diagram pada Gambar 1 dapat diperjelas blok beserta masukan dan keluaran dari sistem. Secara berurutan blok-blok yang ada pada sistem ini adalah *focused crawler*, kemudian pembentuk ringkasan, dan terakhir penyortir topik. Blok-blok tersebut akan berjalan secara berurutan dari awal hingga akhir proses.

Masukan dari blok pertama adalah *query* pencarian dari pengguna yang akan dijadikan patokan pencarian oleh sistem. Keluaran adalah halaman-halaman yang berasal dari situs-situs media massa Indonesia yang tersebar di *internet*. *Crawler* bertugas mencari dan menyalin halaman halaman berita dari situs media massa yang sesuai dan kemudian disimpan pada *database*. *Crawler* yang digunakan bersifat *focused* yang berarti bahwa *crawler* akan mencari hasil yang paling relevan dengan *query* pencarian. Halaman-halaman akan diambil teksnya saja karena teks adalah bagian yang akan diproses oleh sistem di blok kedua. *Link* dari halaman-halaman juga akan disimpan sebagai acuan yang akan disertakan juga di akhir proses.

Blok selanjutnya yang akan dijalankan oleh sistem adalah peringkas berita. Masukan yang akan digunakan dari blok ini ada dua yaitu halaman-halaman hasil pencarian yang dilakukan oleh blok pertama dan kata-kata yang bersumber dari *database* thesaurus. Masukan pertama berfungsi sebagai dokumen masukan utama yang akan diproses menjadi suatu ringkasan. Dokumen yang berupa teks berita hasil pencarian akan dibentuk ringkasannya menggunakan algoritma GVSM.

2.1. *Focused Crawler*

Pada bagian ini akan dijelaskan mengenai perancangan hingga perealisasian *focused crawler* menggunakan algoritma genetik. *Focused crawler* adalah modifikasi dari *web crawler* biasa yaitu *web crawler* yang memiliki proses seleksi dalam penelusuran halaman-halaman *web*. *Focused crawler* bisa membedakan hasil yang relevan dan tidak relevan. *Focused crawler* pada umumnya membutuhkan suatu masukan untuk dijadikan patokan pencarian. Masukan yang sangat mempengaruhi *focused crawler* adalah *link* yang menjadi acuan pencarian. Dengan berpatokan pada *link* tersebut maka *crawler* akan mencari halaman yang memiliki kemiripan tinggi dengan halaman pada *link* tersebut. Kemiripan bisa dilihat dari *domain link* yang ditelusuri atau juga dari isi halaman yang ditelusuri.

Penerapan algoritma genetik pada *focused crawler* adalah pada pembentukan populasi baru dari populasi awal hasil pencarian. Populasi awal yang dimaksud adalah kumpulan halaman-halaman *web* hasil pencarian yang paling pertama. Setelah itu dilakukan dua proses yaitu mutasi dan rekombinasi (*cross over*) untuk mendapatkan populasi baru yang juga memiliki relevansi dengan masukan. Proses rekombinasi dari halaman halaman *web* menggunakan sebuah variabel *cross over* sebagai salah satu patokan. Masing masing URL akan memiliki *cross-over rate* sendiri. Variabel *cross-over* adalah total nilai Jscore dari semua *web* yang memuat URL yang sedang dihitung nilai *cross-over rate*-nya. URL dengan *cross-over rate* terbesar kembali masuk ke dalam antrian *crawler*.

Proses mutasi adalah proses pencarian secara *meta-search*. Potongan potongan kata kunci dari halaman-halaman *web* pada populasi awal akan digunakan sebagai patokan. Potongan potongan tersebut kemudian dijadikan *query* dalam pencarian pada tiga mesin pencari yaitu Bing, Yahoo, dan Google. Sepuluh hasil terbaik dari setiap mesin pencari akan dimasukkan ke dalam antrian *crawler*. Proses mutasi menggunakan variabel *mutation rate* sebagai variabel kemungkinan terjadinya mutasi.

Berdasarkan hasil penelitian dan pengujian, nilai *mutation rate* yang besar akan menambah halaman *web* hasil pencarian di mesin pencari menjadi lolos seleksi.

Sementara nilai *cross-over rate* yang besar akan menambah kemungkinan halaman-halaman *web* baru yang saling berkaitan [4].

Pencarian menggunakan *focused crawler* bersifat *text mining* yakni hanya diambil teks beritanya saja. Dalam melakukan *text mining* dibutuhkan langkah langkah secara urut yaitu: *tokenizing, filtering, stemming, analyzing* [5]. *Tokenizing* adalah tahap pemecahan kalimat menjadi kata-kata penyusun. Sementara *filtering* adalah proses seleksi kata-kata yang dianggap penting dan kemudian akan dilakukan *stemming* yaitu diambil kata dasar dari kata- kata hasil *filtering*. *Analyzing* adalah proses analisa keterhubungan kata-kata dengan dokumen atau paragraf. Proses *analyzing* inilah yang membutuhkan metode-metode *natural language processing* (NLP) untuk mendapatkan hasil yang tepat dan sesuai dengan tata bahasa yang berlaku [6].

2.2. Perancangan Peringkas Berita Otomatis

Pada bagian ini akan dibahas mengenai perhitungan dengan metode GVSM dan perancangan peringkas berita otomatis menggunakan metode GVSM.

2.2.1. Perhitungan Metode GVSM

Generalized vector space model (GVSM) adalah salah satu bentuk pemodelan aljabar yang biasa digunakan untuk menggambarkan teks dan dokumen ke dalam bentuk vektor. Metode ini biasa digunakan untuk menentukan nilai kemiripan dokumen terhadap suatu kata *query*. Pada metode GVSM hasil akhir yang merupakan nilai kemiripan atau relevansi suatu dokumen dinyatakan dalam suatu nilai yaitu *similarity* coefficient (SC). Rumus perhitungan SC pada GVSM adalah sebagai berikut :

$$SC(\mathbf{q},\boldsymbol{d}_i)= \frac{((\sum_{j=1}^{n}\sum_{i=1}^{n}(\boldsymbol{w}_{qj}*\boldsymbol{d}_{ij}))*(\boldsymbol{t}_s\boldsymbol{t}_j))}{\sqrt{\sum_{j=1}^{n}(\boldsymbol{w}_{qj})^2\sum_{j=1}^{n}(\boldsymbol{d}_{ij})^2}} \quad (1)$$

SC = *similarity coefficient*
$\mathbf{w}_{qj}$ = nilai kata *j* terhadap *query q* = tf_{qj} * idf_j
$\boldsymbol{d}_{ij}$ = nilai kata *j* pada kalimat *i* = tf_{ij} * idf_j
tf_{ij} = *term frequency* = kemunculan kata *j* pada kalimat $\boldsymbol{d}_i$
idf_j = inverse document frequency = $\log\left[\frac{d}{dj}\right]$
d = jumlah total kalimat
d_j = jumlah kalimat yang mengandung kata j
$\sum_{j=1}^{t}(\boldsymbol{w}_{qj})^2$ = Lw_{qj} = panjang vektor *query*
$\sum_{j=1}^{t}(\boldsymbol{d}_{ij})^2$ = Ld_{ij} = panjang vektor kalimat
i = indeks kalimat
j = indeks kata
$\mathbf{q}$ = indeks *query*

Dimana $\mathbf{t}_s$ dan $\mathbf{t}_j$ bisa ditentukan secara berbeda beda dalam setiap penggunaannya. Beberapa penelitian menggunakan korelasi antar istilah pada indeks istilah sebagai inputan. Sementara pada penelitian Tsatsaronis menggunakan pembobotan tambahan yaitu keterkaitan secara semantik menggunakan aplikasi Thesaurus yaitu *wordnet* [7]. Sehingga metode GVSM menjadi metode VSM yang memiliki nilai tambah dari segi keterkaitan secara bahasa dan tidak hanya menghitung *similarity coefficient* secara matematis berdasarkan pada jumlah kemunculan tapi juga berdasarkan kepada kaitannya dengan tata bahasa.

Dalam penelitian ini vektor $\mathbf{t}_s$ dan $\mathbf{t}_j$ ditentukan sebagai vektor frekuensi kata pada kalimat ($\mathbf{t}_j$) pada suatu kalimat yang sedang dalam proses perhitungan nilai SC dan vektor jumlah sinonim ($\mathbf{t}_s$) dari masing-masing kata pada vektor $\mathbf{t}_j$, Kata yang dimasukkan hanya jika kata tersebut juga berkaitan dengan *query*. Jika kata tidak berkaitan maka kata tidak dimasukkan ke dalam vektor. Nilai perkalian titik (skalar) kedua vektor tersebut akan dijadikan pengali pada pembilang dari perhitungan SC. Dimensi vektor sinonim disamakan dengan dimensi vektor kata. Dengan kata lain semakin banyak sinonim dari kata yang frekuensinya besar dan juga berkaitan dengan *query* pada suatu kalimat akan menyebabkan kalimat tersebut memiliki nilai SC yang lebih besar.

2.2.2. Perancangan *User Interface* pada Sistem

UI pada sistem ini terbagi ke dalam beberapa *form* yang terpisah. Tujuan *form* yang dibuat terpisah agar memudahkan pengerjaan sistem dan juga memudahkan pengguna memilih *form* yang sesuai dengan kebutuhan. *Form* dirancang dengan jenis *Windows Form* menggunakan bahasa pemrograman Visual C# .NET. Dengan menggunakan *compiler* Microsoft Visual Studio 2010 dan berbasis pada .NET Framework 4.

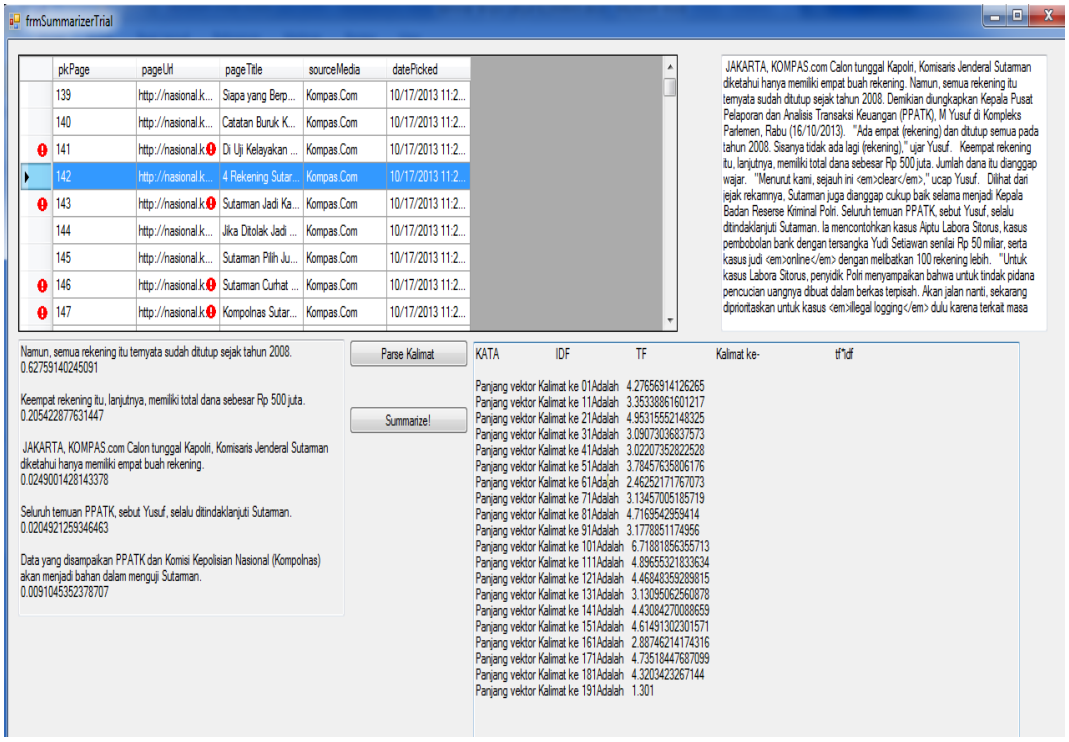
Berikut modul-modul UI yang ada pada sistem

a. Form Awal (frmAdmin)
b. Form Crawling (frmCrawling)
c. Form Summarizer (frmSummarizer)
d. Form Debugging Mode Summarizer (frmSummarizerTrial)

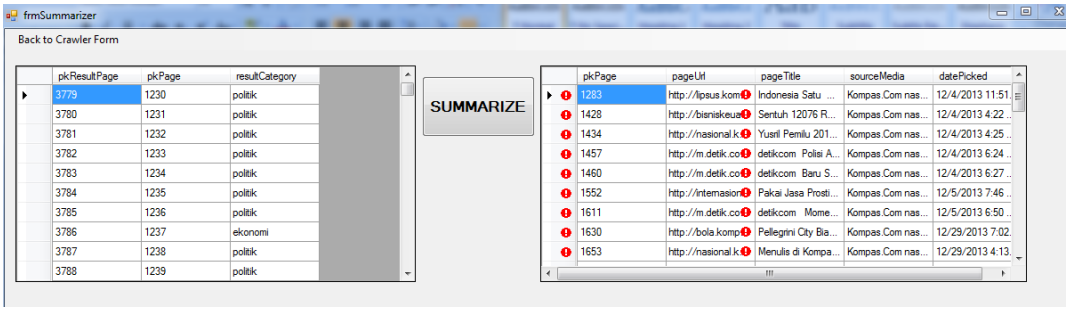
Form frmSummarizer trial adalah bagian yang berfungsi melihat rincian proses peringkasan seperti tertampil pada Gambar 2.

Sementara *form* frmSummarizer adalah *form* yang berfungsi meringkas berita dalam jumlah banyak sekaligus dan juga melihat jumlah berita yang gagal diringkas. Sejumlah 6% berita yang tidak teringkas dikarenakan berbagai faktor antara lain berita tidak ada isinya, atau berita terlalu kotor karena banyak *tag* HTML yang tidak dapat dibersihkan. *Form* frmSummarizer tertampil pada Gambar 3.

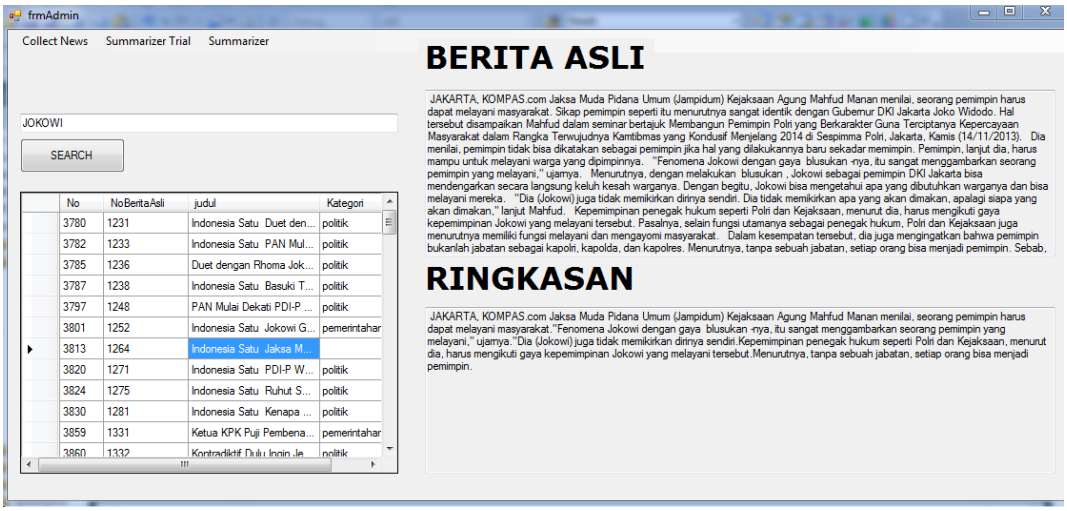
Hasil akhir dari proses peringkasan yang dapat dilihat dengan mudah oleh pengguna ada pada frmAdmin dimana pengguna dapat mencari berita dengan judul yang sesuai dan sudah terkategori secara otomatis. Dengan hasil akhir tertampil pada Gambar 4.



Gambar 2. Tampilan frmSummarizerTrial



Gambar 3. Tampilan *frmSummarizer*



Gambar 4. Tampilan *frmAdmin*

3. Pengujian dan Analisis

Pada bagian ini akan dijelaskan dua pengujian yang telah dilakukan yaitu pengujian *focused crawler* dan pengujian hasil ringkasan berita.

3.1. Pengujian *Focused Crawler*

Focused crawler diuji dengan memperhatikan beberapa bagian penting yang berkaitan antara lain parameter dari pengguna dan juga melihat jenis-jenis keluaran untuk tiap-tiap masukan yang berbeda-beda. Beberapa parameter dibuat sama dengan penelitian *focused crawler* sebelumnya sedangkan beberapa parameter disesuaikan dengan penelitian ini. Keluaran yang didapat juga merupakan hasil modifikasi dari *focused crawler* sebelumnya sehingga setiap keluaran perlu diperhatikan secara detail.

Parameter yang merupakan bagian utama dari *focused crawler* dan tidak mengalami perubahan untuk setiap percobaan adalah sebagai berikut :

1. *populationSize* adalah ukuran populasi, banyaknya halaman *web* yang membentuk sebuah populasi pada implementasi Algoritma Genetik.
2. *generationSize* adalah ukuran generasi, banyaknya perulangan (iterasi) pembentukan populasi yang dilakukan pada tahap Algoritma Genetik.
3. *crossoverRate* adalah probabilitas *cross-over*, peluang terjadinya rekombinasi pada tahap Algoritma Genetik.
4. *mutationRate* adalah probabilitas mutasi, peluang terjadinya mutasi pada tahap Algoritma Genetik [4].

Keempat parameter tersebut diatur didalam sebuah *file* bernama App.config dan kemudian akan diambil oleh program untuk kemudian digunakan dalam proses *crawling*. Nilai-nilai untuk keempat parameter tersebut antara lain :

1. *populationSize* : 100 (dalam *range* 1-100)
2. *generationSize* : 100 (dalam *range* 1-100)
3. *crossoverRate* : 0.7 (dalam *range* 0-1)
4. *mutationRate* : 0.7 (dalam *range* 0-1)

Pengujian *focused crawler* ditekankan pada seberapa baik *crawler* bisa mendapatkan berita-berita yang relevan dengan masukan selama proses *crawling*. Masukan yang diberikan berbeda-beda setiap kali pengujian dan yang membedakan antara setiap masukan adalah *domain lexicon* dan *keyword* dari setiap percobaan pencarian. Sementara beberapa percobaan memiliki juga kesamaan yaitu dalam topik berita yang dicari. Sesuai dengan pembatasan topik pada penelitian ini difokuskan ke dalam enam topik. Sehingga dalam proses pencarian dan juga menentukan jenis berita yang akan dicari tetap mengacu pada enam topik yang sudah ditentukan. Keenam topik yang dijadikan patokan dalam pengujian adalah sebagai berikut :

1. Politik
2. Kriminal
3. Kesehatan
4. Olahraga
5. Pemerintahan
6. Ekonomi

Untuk masing-masing topik tersebut akan dilakukan pencarian sebanyak lima kali dengan *keyword* dan *domain lexicon* yang berbeda-beda untuk menentukan pencarian.

Setelah dilakukan pencarian maka akan dilihat pula keluaran dari sistem dan dilakukan penilaian terhadap sistem.

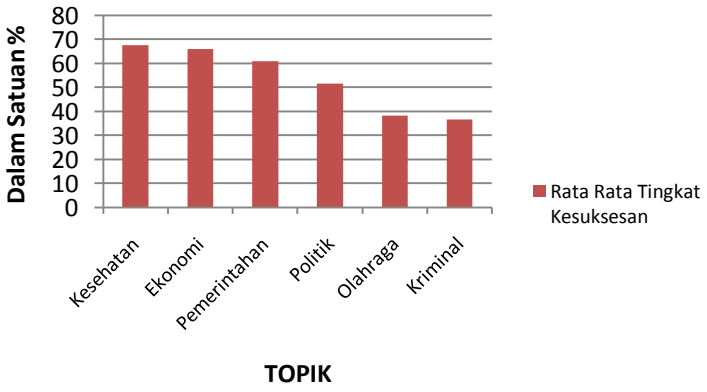
Dalam menentukan *keyword* dan juga *domain lexicon* digunakan cara manual yaitu memilih langsung halaman *web* apa yang akan dijadikan *domain lexicon* atau dengan kata lain menjadi titik awal *crawling* dan juga menentukan *keyword* apa yang sesuai dengan *domain lexicon* yang sudah ditentukan.

Penilaian terhadap hasil *crawling* adalah berupa tingkat kesuksesan. Tingkat kesuksesan adalah perbandingan berapa banyak berita yang relevan dengan jumlah berita. Penilaian mengenai apakah suatu berita relevan atau tidak dilakukan secara *manual* dengan membaca satu persatu berita yang disimpan setiap pencarian. Tingkat kesuksesan dalam satuan persentase keberhasilan dengan nilai tertinggi 100% dan nilai terendah 0%. Ringkasan dari hasil pengujian *focused crawler* ditampilkan pada Tabel 1 berikut.

Tabel 1. Ringkasan Pengujian *Focused Crawler*

Topik Berita	Tingkat Kesuksesan Percobaan ke- (Dalam satuan %)					Rata-Rata (Dalam satuan %)
	1	2	3	4	5	
Kesehatan	80,0	87,5	71,4	32,1	66,6	67,5
Ekonomi	64,4	76,4	63,6	85,7	40,0	66,0
Pemerintahan	100,0	50,0	53,8	50,0	50,0	60,7
Politik	90,7	57,8	29,4	50,0	30,0	51,6
Olahraga	27,7	26,3	20,0	54,5	62,5	38,2
Kriminal	31,4	39,2	55,5	7,6	48,5	36,5

Sementara perbandingan rata-rata tingkat kesuksesan pencarian untuk tiap-tiap topik berita digambarkan dalam diagram pada Gambar 5.



Gambar 5. Rata-Rata Tingkat Kesuksesan Pencarian Tiap Topik

Dari Tabel 1 dapat diambil rata-rata kesuksesan percobaan dari keseluruhan topik adalah 53%. Nilai tersebut lebih rendah dari target yang dipasang. Ada empat faktor yang menyebabkan rendahnya nilai tersebut antara lain.

1. Berita yang sudah masuk tidak akan masuk lagi sehingga mengakibatkan kemungkinan berita-berita yang dicari pada pencarian terakhir sudah diambil oleh *crawler* pada pencarian pencarian awal. Misalkan di awal sudah ditemukan berita mengenai melemahnya rupiah maka jika pada pencarian berikutnya

mencari topik melemahnya rupiah jumlah berita menurun karena sudah diambil pada pencarian pertama.

2. *Starting page* yang ideal sangat sedikit.
3. Gaya bahasa dan judul yang beragam
4. Judul topik yang merupakan *headline* (misalkan: Korupsi Banten) banyak tercantum di banyak halaman berita lain sehingga menyebabkan halaman berita lain ikut terambil

3.2. Pengujian Peringkasan Berita

Pada pengujian sistem peringkas berita poin yang diujikan adalah ringkasan dari berita. Pengujian ini menggunakan bantuan kuesioner untuk mempercepat pengumplan data dari responden. Bahan yang digunakan responden untuk menilai meliputi ringkasan berita, berita asli, dan judul berita. Sementara responden memberi identitas berupa nama dan latar belakang pekerjaan. Responden yang diminta untuk mengisi kuesioner berjumlah empat puluh orang responden dengan latar belakang pekerjaan yang berbeda-beda. Jumlah berita yang diberikan pada responden untuk dinilai adalah sejumlah lima berita dengan masing masing berita memiliki ringkasan dan berita asli itu sendiri. Selain itu responden juga diminta memberikan komentar tambahan serta kalimat yang seharusnya ada atau tidak ada dalam suatu ringkasan.

Penilaian dilakukan dengan metode klasifikasi yaitu responden memberikan penilaian terhadap masing masing berita dengan klasifikasi sebagai berikut. Sementara untuk data berita yang disertakan dalam kuesioner dijabarkan dalam Tabel 3.

Tabel 2. Kriteria Penilaian

Kriteria	Nilai
Sangat Baik	4
Baik	3
Buruk	2
Sangat Buruk	1

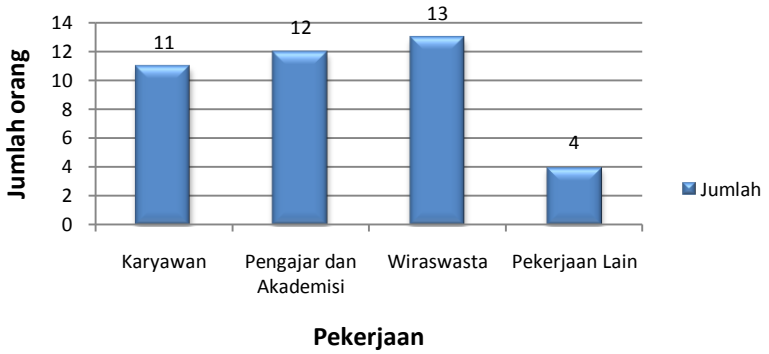
Tabel 3. Data Berita pada Kuesioner

No Berita	Judul Berita	Kategori
1	Wakil Ketua MK Yakin Tak Ada Lagi Hakim Konstitusi Terjerat Kasus	Kriminal
2	Indonesia Juara Umum Yamaha ASEAN Cup Race 2013	Olahraga
3	Gempa 67 SR Guncang Maluku Tak Berpotensi Tsunami	Peristiwa
4	Dr Lo Kalau Mau Kaya Ya Jangan Jadi Dokter tapi Pedagang	Tokoh
5	Belajar Bahasa Inggris Sambil Rayakan Halloween	Pendidikan

Responden pada pengujian hasil ringkasan ini terdiri dari 40 orang responden yang terbagi dalam kategori pekerjaan yang berbeda-beda. Dari semua pekerjaan tersebut dapat dikelompokkan untuk mempermudah melihat gambaran dari ragam sampel. Kategori kategori pekerjaan dari responden antara lain :

a. Karyawan dimana termasuk didalamnya karyawan swasta, pegawai negeri sipil, desainer interior, dan *manager*.
b. Pengajar atau akademisi yaitu antara lain guru, pengajar kursus, dosen, dan asisten laboratorium.
c. Wiraswasta, pengusaha, wiraswasta komputer, kontraktor dan *retailer* obat
d. Pekerjaan lain seperti pendeta, ibu rumah tangga, pensiunan, dan mahasiswa.

Perbandingan jumlah responden untuk tiap-tiap pekerjaan digambarkan melalui diagram pada Gambar 6.



Gambar 6. Jumlah Responden Berdasarkan Kategori Pekerjaan

Ringkasan dari penilaian responden dapat dilihat pada Tabel 4 yaitu berupa rata-rata penilaian untuk tiap-tiap ringkasan dari berita 1 hingga berita 5.

Tabel 4. Ringkasan Hasil Penilaian Responden

Berita Ke-	1	2	3	4	5
Rata-rata (Sesuai Kriteria Penilaian)	2,60	2,52	2,77	3,00	2,65

Dari Tabel 4 dapat diambil rata-rata keseluruhan penilaian dari berita 1 hingga berita ke 5 adalah 2,71 atau jika mengacu pada kriteria penilaian dapat dikatakan mendekat ke arah baik. Hasil ringkasan yang dibuat oleh sistem oleh karena itu dapat dikatakan secara umum dapat diterima oleh pembaca berita. Akan tetapi beberapa responden memberikan komentar tambahan terhadap hasil ringkasan dimana komentar tersebut antara lain :

a. Ringkasan yang bagus harus berisi detil kejadian dan data lengkap.
b. Isi berita tidak menyeluruh seperti aslinya.
c. Perhatikan detil bahasan

4. Kesimpulan

Dari hasil perancangan dan pengujian sistem peringkas berita otomatis ini dapat diambil sejumlah kesimpulan bahwa metode GVSM selain berfungsi untuk menilai keterkaitan antar topik per dokumen juga dapat digunakan dalam menilai tingkat keterkaitan kalimat dengan topik suatu dokumen. *Focused crawler* yang sudah dimodifikasi dalam hal pengambilan dan pemilihan isi halaman *web* mampu membatasi pencarian meskipun tidak mengubah algoritma pada *focused crawler* terlalu banyak. *Focused crawler* masih memiliki kelemahan dalam mencari berita dengan topik yang sesuai karena memang pada dasarnya *crawler* dirancang untuk menelusuri keseluruhan isi web dan tidak hanya isi berita pada halaman *web* saja. Akibat kelemahan ini tingkat keberhasilan hanya mencapai 53%.

Hasil ringkasan yang dibuat sistem secara umum dapat dikatakan mendekati baik dan dapat diterima oleh responden yaitu dengan penilaian sebesar 2,71 (dari skala 1-4). Berita yang ringkas tidak bisa langsung dinilai baik oleh responden jika peringkasan

berita terlalu banyak menyingkirkan bagian-bagian yang memiliki informasi penting pada berita.

Daftar Pustaka

[1] G. Keraf, *Komposisi*, Nusa Indah, Ende, 1994.

[2] R. Mandala, "Sistem Pengidentifikasi Otomatis Keterkaitan Topik antar Paragraf dalam Dokumen Ekspositori," *Seminar Nasional Aplikasi Teknologi Informasi*, Yogyakarta, 2004.

[3] M. E. A. Haryono, "Pembentukan Intisari Topik secara Otomatis dalam suatu Paragraf dengan Model Vector Space Model," *Seminar Nasional Aplikasi Teknologi Informasi*, Yogyakarta, 2005.

[4] B. W. Yohanes, *Implementasi Algoritma Genetik untuk Membangun Domain-Specifik Collections*, Skripsi, Universitas Kristen Satya Wacana, Salatiga, 2009.

[5] J. Jeconiah, *Sistem Analisa Spatio-Temporal, Informasi Bencana Banjir di Indonesia Menggunakan Web Mining*, Skripsi-FTEK, Universitas Kristen Satya Wacana, Salatiga, 2012.

[6] A. Kao dan S. Poteet, "Text Mining and Natural Language Processing – Introduction to Special Issues," *Boeing Phantom Works*, Vol. 7, Issue 1, Seattle, 2007.

[7] G. Tsatsaronis dan V. Panagiotopoulou, "A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness," *Proceedings of the EACL 2009 Student Research Workshop*, Athens University of Economics and Business, Athens, 2009.