PEMBUATAN RINGKASAN OTOMATIS DOKUMEN BERITA BERBAHASA INDONESIA MENGGUNAKAN ALGORITMA *DIJKSTRA*

[1]**M. Rifqi Fu'adi**, [2]Zainal Abidin, [3]Hani Nurhayati

[1,2,3]Jurusan Teknik Informatika FSaintek UIN Maulana Malik Ibrahim Malang
Email : [1]ukielX@gmail.com, [2]zainal@ti.uin-malang.ac.id
[3]hani.hayati@gmail.com,

ABSTRAK

Berita telah menjadi suatu kebutuhan dalam keseharian manusia. Perkem-bang-an teknologi informasi menyebabkan internet menjadi salah satu media untuk menyampaikan berita yang memungkinkan berita dapat diperbarui secara *real-time*. Internet mendorong proses penyampaian berita menjadi sangat cepat, maka dibutuhkan sebuah cara untuk lebih cepat dalam memahami suatu berita. Peringkasan teks secara otomatis dari halaman web dapat digunakan untuk memahami suatu berita dengan lebih cepat. Proses dari aplikasi ini diawali dengan proses ekstraksi halaman web untuk mendapatkan teks berita. Kemudian sistem melakukan *text preprocessing* untuk mengubah teks ke bentuk vektor. Bobot diperoleh dari 5 parameter dan hasilnya akan direpresentasikan dalam bentuk graf. Jalur terpendek diperoleh dengan pencarian menggunakan algoritma *Dijkstra*. Jalur yang terpendek merepresentasikan nomor kalimat yang diambil sebagai ringkasan. Berdasarkan pengujian teknik ini mampu menghasilkan rata-rata *recall* 57%, *precision* 64%, dan *f-measure* 58%. Hal tersebut menunjukkan bahwa sistem dengan penggabungan 2 metode ini efektif dalam mengambil kalimat yang sama dengan yang diambil ahli bahasa sebagai ringkasan.

Kata Kunci: peringkasan, berita, *text preprocessing*, dijkstra, dokumen graf, bobot kalimat, *sentence based summerization*, ekstraksi, halaman web

PENDAHULUAN

Dalam kehidupan sehari-hari manusia tidak akan bisa terlepas dari yang disebut sebagai berita. Suatu berita harus berisi informasi penting dan menarik perhatian, oleh sebab itu maka penyajian berita harus mempertimbangkan aspek isi dan juga aspek waktu. Selain unsur isi, kecepatan penyajian berita juga patut menjadi perhatian. Terdapat istilah yang umum di masyarakat yang berbunyi "tiada hari tanpa berita", hal ini menggambarkan kebutuhan masyarakat akan berita dalam aktifitas kesehariannya.

Seiring dengan kemajuan teknologi informasi, teknologi internet menjadi sarana penting dalam memperoleh berita yang sekarang disebut sebagai media *online*. Dengan adanya sarana tersebut didapat berbagai kemudahan yang diantaranya adalah akses berita yang *up-to-date* karena bisa diperoleh informasi dari waktu ke waktu dengan proses penyajian yang mudah dan sederhana. Selain itu dengan media *online* juga memungkinkan berita dapat disampaikan secara *real-time* karena media *online* dapat menyajikan informasi bahkan saat peristiwa masih berlangsung. Selain itu media *online* juga dapat diakses dari mana saja selama didukung fasilitas internet.

Akan tetapi permasalahan timbul ketika banyak sekali berita yang terbit setiap harinya karena akan menimbulkan adanya banjir berita dan memperpendek umur dari berita sehingga jika tidak cepat dimengerti maka pembaca tetap akan ketinggalan informasi. Oleh sebab itu muncul ide peniliti untuk membantu pembaca berita untuk mengurangi waktu membaca dengan tanpa mengurangi pemahamannya terhadap suatu informasi yaitu dengan sistem peringkasan teks secara otomatis (*Automatic Text Summarization*). Sistem peringkas teks otomatis merupakan teknologi yang diharapkan dapat membantu bagi konsumen berita.

Teks sumber yang digunakan dalam aplikasi ini merupakan berita yang berasal dari halaman website yang diekstrak judul dan kandungan beritanya menggunakan fitur yang telah ditanamkan dalam aplikasi. Fitur pengekstrakan halaman *website* ini menggunakan metode ekstraksi yang telah digunakan dalam penelitian Ekstraksi Teks Otomatis Dari Halaman Web Dengan SQL guna Membantu Mempercepat Penyusunan Korpus (Abidin dan Fatcurrohman, 2011).

TINJAUAN PUSTAKA

Ide dasar dari sebuah ringkasan adalah penyingkatan jadi ringkasan merupakan versi singkat dari sebuah dokumen aslinya. Ringkasan merupakan proses dari pembuatan intisari informasi terpenting dari suatu sumber untuk menghasilkan versi yang lebih ringkas bagi penggunanya (Mani dan Maybury, 1999). Proses utama yang terjadi dalam peringkasan

text adalah: *Topic Identification, Interpretation, Generating*.

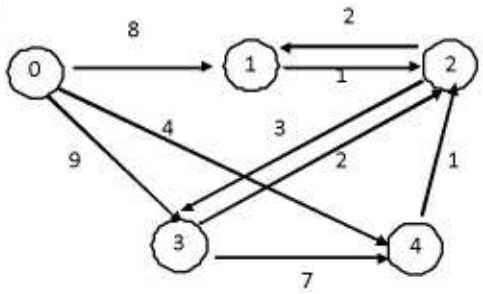
Graf adalah suatu diagram yang memuat informasi tertentu jika diinterpretasikan secara tepat (Jong, 2002). Suatu graf G terdiri dari 2 himpunan yang berhingga, yaitu himpunan titik-titik tidak kosong(V(G)) dan himpunan garis-garis (E(G)). Setiap garis berhubungan dengan satu atau dua titik. Dua titik dikatakan berhubungan (*adjacent*) jika ada garis yang menghubungkan antar keduanya.

Hubungan antar simpul dalam graf kadang-kadang perlu diperjelas. Hubungan tidak cukup menunjukkan simpul-simpul mana saja yang berhubungan langsung, tapi juga seberapa kuatkah hubungan itu. Dalam aplikasinya, bobot suatu garis lebih tepat dibaca sebagai "jarak", "biaya", "panjang", "kapasitas", dan lainnya (Jong, 2002). Gambaran bentuk graf yang berarah dan memiliki bobot hubungan antar simpulnya dapat dilihat pada **gambar 1.**

Algoritma *Dijkstra* adalah algoritma yang ditemukan oleh matematikawan Belanda bernama Edsger Wybe Dijkstra pada tahun 1959. Dijkstra tergolong algoritma g*reedy* yang mencari keputusan terbaik dari yang paling baik.

Operasi dasar dari algoritma Dijkstra adalah garis *relaxation,* yang artinya jika terdapat sebuah garis dari u ke v, kemudian jalur terpendek yang diketahui dari s ke u (d[u]) dapat ditambahkan ke sebuah jalur dari s ke v dengan menambahkan garis (u,v) pada akhir. Jalur ini akan memiliki panjang sebesar d[u]+w(u,v). Jika ini lebih kecil dari d[v], maka nilai d[v] dapat diganti dengan nilai baru. *Edge relaxation* dilakukan hingga semua nilai d[v] menggambarkan nilai jalur terpendek dari s ke v.

Dalam mencari jalur terpendek, algoritma dijkstra dirancang agar masing-masing garis(u,v) diperiksa hanya sekali, ketika d[u] telah mencapai nilai finalnya, himpunan S berisi semua titik yang telah dikunjungi, sedangkan Q berisi simpul-simpul yang lain. Awalnya himpunan S kosong, dan dalam setiap langkah, satu simpul dipindahkan dari Q ke S. Simpul ini dipilih dari simpul dengan nilai terkecil d[u]. Ketika sebuah titik u dipindah ke S, algoritma me-*relax* setiap *outgoing edge* (u,v). *Pseudo code* dari algoritma dijkstra dapat dilihat pada **gambar 2.**



Gambar 1. **Graf Berarah dan Berlabel**

```
function Dijkstra(G, w, s)
 for each vertex v in V[G]
  d[v] := infinity
  previous[v] := undefined
//Jarak dari s ke s
 d[s] := 0
 S := empty set
//Set of all vertices
 Q := V[G]
 while Q is not an empty set
  u := Extract_Min(Q)
  S := S union {u}
 for each edge(u,v) outgoing from u
//Relax(u,v)
  if d[u]+w(u,v)<d[v]
   d[v] := d[u] + w(u,v)
   previous[v] := u
```

Gambar 2. ***Pseudo Code* Algoritma Dijkstra**

RANCANGAN SISTEM

Peringkas dokumen yang akan dibuat merupakan sistem yang membaca sebuah teks dokumen dan secara otomatis menghasilkan (*generate*) sebuah ringkasan. Ringkasan yang dihasilkan akan bersifat indikatif dan generik. Metode yang digunakan pada sistem ini adalah meringkas dengan teknik ekstraksi dengan menggunakan Algoritma *Shortest Path* seperti yang telah digunakan dalam penelitian Jonas dan Kenji (Jonas dan Kenji, 2003). Dalam penelitian Jonas Kenji, jarak antar masing-masing simpul yang dicari jalur terpendek dihitung dengan menggunakan persamaan 1. Untuk menentukan bobot kepentingan (*weight*) kalimat digunakan pembobotan berdasarkan lima parameter (persamaan 2) seperti pada penelitian yang dilakukan oleh Visser dan Wieling(Visser dan Wieling, 2005).

$$cost_{i,j} = \frac{(i - j)^2}{overlap_{i,j} \cdot weight_j} \quad (1)$$

$$W = \alpha \frac{F}{\max(F)} + \beta \frac{C}{\max(C)} + \gamma \frac{P}{\max(P)} + \chi \frac{Q}{\max(Q)} + \omega \frac{J}{\max(J)} \quad (2)$$

Urutan proses yang tergambar pada blok diagram gambar 3 merupakan tahap yang dilakukan sistem dalam melakukan proses peringkasan. Proses diawali dengan *text preprocessing*, yaitu proses memecah naskah sumber menjadi paragraf, kalimat dan kata untuk digunakan dalam proses pembentukan graf, perhitungan bobot dan

pembentukan ringkasan. Untuk menentukan bobot kepentingan kalimat, maka setiap kalimat dihitung bobot kepentingannya menggu-nakan 5 fitur (parameter), yaitu lokasi kalimat (*F*), kemiripan dengan judul (*T*), frekuensi kemunculan kata (*F*), adanya kata kunci (*C*), dan kemiripan dengan *query* (*Q*).

Algoritma *shortest path* hanya mungkin diterapkan pada graf, jadi untuk dapat dicari jalur terpendeknya maka naskah sumber harus dikonversi ke dalam bentuk dokumen graf. Konversi dilakukan dengan merepresentasikan setiap kalimat menjadi simpul dari graf dan adanya kesamaan kata (*overlap*) antar masing-masing kalimat menjadi *vertex* antar simpul dalam graf. Selanjutnya graf diberi label berupa *cost* yang dalam pencarian jalur terpendek digambarkan sebagai jarak antar simpul-simpul graf. Bobot garis dari dua kalimat yang paling memiliki kesamaan dan nilai kepentingan yang besar akan memiliki *cost* yang kecil.

Setelah graf dokumen yang memiliki label terbentuk, maka jalur terpendek dapat dicari menggunakan Algoritma *Shortest Path*. Dalam penelitian ini pencarian jalur terpendek dilakukan dengan menerapkan algoritma Dijkstra. Titik awal pencarian dimulai dari simpul yang menggambarkan kalimat pertama dari naskah sumber dengan tujuan simpul terakhir yang menggambarkan kalimat terakhir dari naskah sumber.

Jalur terpendek dari graf representasi naskah sumber digunakan sebagai acuan pengambilan kalimat. Indeks kalimat yang memiliki nomor sesuai jalur terpendek yang ditemukan diambil sebagai kalimat yang menjadi ringkasan dari naskah. Kalimat-kalimat yang diambil lalu dirangkai menjadi satu naskah baru yang lebih pendek tanpa menghilangkan unsur informasi dari naskah aslinya karena sistem ini mengambil kalimat berdasar pada bobot kepentingan suatu kalimat.

Gambar 3. **Blok Diagram Proses Peringkasan**

UJI COBA DAN HASIL

Uji coba sistem peringkasan dilakukan dengan membandingkan 31 hasil ringkasan sistem dengan *hand made summarization* yang dihasilkan ahli bahasa Indonesia terhadap teks sumber yang sama. Teks sumber merupakan berita yang diunduh secara acak dari situs berita www.tempo.co yang diterbitkan bulan Mei 2012. Hasil pengujian disajikan pada **tabel 1**.

Dari hasil uji coba yang dilakukan diperoleh nilai rata-rata *recall* sebesar 57%, *precision* 64%, dan *f-measure* sebesar 58%. Hal ini menunjukkan sistem mampu memilih beberapa kalimat yang sama dengan yang dipilih oleh ahli dalam pembuatan ringkasan manual.

Dalam menentukan tingkat keberhasilan sistem peringkasan ini peneliti masih belum mendapatkan informasi apakah dengan nilai *recall*, *precision* dan *f-measure* yang menunjukkan angka rata-rata di atas 50% dapat dikatakan bahwa hasil ringkasan itu baik atau tidak. Dalam penelitian lain

yang telah dilakukan oleh edmudson (Edmudson, 1969) dikatakan bahwa sulit mengambil kesimpulan terhadap *performance* sistem peringkasan dari nilai *precision* dan *recall*. Kesulitan itu disebabkan karena nilai-nilai *precission* dan *recall* memiliki besar yang relatif terhadap ringkasan manual. Selain karena nilai yang relatif, kesulitan dalam menentukan perfoma sistem menggunakan perhitungan *precission* dan *recall* juga disebabkan karena ringkasan yang dibuat manusia bersifat subjektif yang berarti berbeda antara orang yang satu dengan orang yang lain dan tidak ada satupun ringkasan yang benar.

Tabel 1. Hasil Evaluasi

No	*Overlap*	*Precision*	*Recall*	*f-Measure*
1	7	0.78	0.54	0.64
2	14	0.78	0.82	0.80
3	5	0.45	0.33	0.38
4	5	0.63	0.45	0.53
5	5	0.83	0.50	0.63
6	9	0.69	0.90	0.78
7	3	0.60	0.38	0.46
8	5	0.50	0.83	0.63
9	8	0.57	0.67	0.62
10	3	0.50	0.21	0.30
11	4	0.80	0.67	0.73
12	4	0.80	0.27	0.40
13	7	0.54	0.70	0.61
14	9	0.75	0.38	0.50
15	9	0.60	0.69	0.64
16	6	0.40	0.33	0.36
17	5	0.45	0.56	0.50
18	3	0.75	0.18	0.29
19	5	0.38	0.56	0.45
20	7	0.58	0.28	0.38
21	6	0.60	0.75	0.67
22	6	0.46	0.67	0.55
23	6	0.43	0.60	0.50
24	8	0.67	0.50	0.57
25	4	0.80	0.50	0.62
26	4	0.57	0.67	0.62
27	8	0.89	0.89	0.89
28	10	0.91	0.71	0.80
29	5	0.50	0.56	0.53
30	8	0.73	0.73	0.73
31	9	0.82	0.82	0.82

KESIMPULAN

Membangun peringkas dokumen otomatis dengan menggabungkan dua metode yaitu *Sentence Based Summarization* dan *Extraction Based Summarization Using a Shortest Path Algorithm* dapat dilakukan. Hal ini dapat diketahui dari pengujian hasil ringkasan sistem dibandingkan dengan ringkasan ahli(guru bidang studi bahasa Indonesia) yaitu yang menghasilkan rata-rata *recall* sebesar 57%, *precision* sebesar 64%, dan *f-measure* sebesar 58%. Hasil tersebut menunjukkan bahwa sistem dapat mengambil kalimat yang sama dengan yang diambil oleh ahli bahasa.

Algoritma Dijkstra dapat digunakan untuk peringkasan teks otomatis, asalkan teks yang akan dicari direpresentasikan terlebih dulu menjadi graf dokumen yang lengkap dengan masing-masing kalimat sebagai simpul dan *cost* tingkat kepentingan dari masing-masing kalimat sebagai jarak antar simpul.

DAFTAR PUSTAKA

Abidin, Zainal dan Fatchurrohman. 2011. *Ekstraksi Teks Otomatis dari Halaman Web dengan SQL Guna Membantu Mempercepat Penyusunan Korpus*. Seminar Nasional Green Technology 2. Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.

Edmudson, H.P. 1969. *New Methods in Automatic Extracting*. Journal of Assosiation for Computing Machinery, 16(2):264-285.

Harianto, Bambang. 2007. *Esensi-esensi Bahasa Pemrograman Java*. Bandung: Informatika Bandung.

Jones, K Sparck and Galliers, Julia R. 1996. *Evaluating natural language processing system : An analysis and review*. New York : Springer.

Kupiec, Chen, Pedersen, 1995. *A Trainable Document Summarizer*. Proceedings of the 18th ACM SIGIR.

Mani, Bloedorn, 1998. *Machine learning of generic and user focussed summarization*. Proceedings of the 15h National Conference on AI.

Mani, Inderjeet. 2001. *Summarization evaluation: an overview*. The MITRE Corporation, W640 11493 Sunset Hills Road Reston, VA 20190-5214 USA.

Nomoto, Matsumoto, 2001. *A new approach to unsupervised text summarization*. Proceedings of the 24th ACM SIGIR.

Siang, Jong Jek. 2002. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta : Andi

Sjobergh, Jonas and Araki, Kenji. 2003. *Extraction based summarization using a shortest path algorithm, dr-hato.se/research/shortpath.pdf*, Diakses tanggal : 22 Oktober 2011

Visser,W.T dan Wieling M.B. 2005. *Sentence-based Summarization of Scientific Documents*. Department of Computing Science, University of Groningen

Stubblebine, Tony. 2007. *Regular Expression Pocket Reference*. Canada: O'Reilly Media, Inc.

Fajar, Indra dan Gustian Siregar, Dede Tarwidi. *Algoritma Mencari Lintasan Terpendek*. Laboratorium Ilmu dan Rekayasa Komputasi Departemen Teknik