Sistem Temu Kembali Dokumen Teks dengan Pembobotan Tf-Idf Dan LCS

Munjiah Nur Saadah[1], Rigga Widar Atmagi[2], Dyah S. Rahayu[3], Agus Zainal Arifin[4]
Jurusan Teknik Informatika, Fakultas Teknologi Informasi , Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya, 60111
Email: munjiah.nur11[1], rigga.widar11[2], dyah.rahayu11[3]@mhs.its.ac.id, agusza@cs.its.ac.id[4]

ABSTRAK

Sistem temu kembali dokumen teks membutuhkan metode yang mampu mengembalikan sejumlah dokumen yang memiliki relevansi tinggi sesuai dengan permintaan pengguna. Salah satu tahapan penting dalam proses representasi teks adalah proses pembobotan. Penggunaan LCS dalam penyesuaian bobot Tf-Idf mempertimbangkan kemunculan urutan kata yang sama antara query dan teks di dalam dokumen. Adanya dokumen yang sangat panjang namun tidak relevan menyebabkan bobot yang dihasilkan tidak mampu merepresentasikan nilai relevansi dokumen. Penelitian ini mengusulkan penggunaan metode LCS yang memberikan bobot urutan kata dengan mempertimbangkan panjang dokumen terkait dengan rata-rata panjang dokumen dalam korpus. Metode ini mampu melakukan pengembalian dokumen teks secara efektif. Penambahan fitur urutan kata dengan normalisasi rasio panjang dokumen terhadap keseluruhan dokumen dalam korpus menghasilkan nilai presisi dan recall yang sama baiknya dengan metode sebelumnya.

Kata Kunci: *LCS, sistem temu kembali informasi, pembobotan, Tf-Idf, presisi, recall.*

1. PENDAHULUAN

Saat ini sebagian besar informasi disimpan dalam bentuk digital pada media elektronik sehingga sistem informasi harus mampu mengembalikan sejumlah besar data teks yang dibutuhkan pengguna. Teknologi sistem temu kembali dokumen teks menyediakan cara untuk mendapatkan kembali informasi yang dibutuhkan tersebut. Teknologi ini mampu menemukan dokumen teks yang tersimpan dalam sistem sesuai dengan *query* tertentu yang dimasukkan oleh pengguna, baik *query* dengan kata atau kalimat. Dokumen teks ini ditampilkan dalam urutan menurun mulai dari dokumen yang memiliki nilai relevansi tertinggi dengan *query* yang dimaksud [8]. Dalam perkembangannya digunakan pula klasifikasi dokumen sehingga akan turut memudahkan proses penemuan kembali dokumen teks.

Dua tahapan awal dalam teknologi sistem temu kembali dokumen teks adalah pra-pemrosesan teks dan representasi teks. Pra-pemrosesan teks terdiri dari banyak tahap, misalnya *tokenizing*, *stemming*, dan *stoplisting*. Sedangkan tahap representasi teks biasa dikenal dengan tahap pembobotan teks. Telah banyak penelitian sebelumnya yang mengusulkan metode-metode baru untuk pembobotan teks ini. Metode pembobotan yang sampai saat ini masih sering digunakan yaitu *Term frequency-Inverse document frequency* (Tf-Idf) yang mempertimbangkan seringnya kemunculan term dalam dokumen dan rasio panjang dokumen tersebut di dalam korpus [5]. Selain itu, terdapat pula pembobotan BM25 yang juga mempertimbangkan panjang dokumen dibanding rata-rata panjang dokumen di dalam korpus disertai dengan beberapa parameter penyesuaian [4]. Erenel dan Altincay [1] menggunakan pembobotan yang memanfaatkan transformasi linier pada frekuensi term sebagai fitur untuk mengategorikan dokumen teks. Terdapat pula pembobotan yang didasarkan pada nilai diskriminasi dari hasil pengembalian sebelumnya [6]. Lain halnya dengan Luo dan Xiong [3] yang menggunakan pembobotan berdasarkan konsep semantik di dalam penelitiannya.

Tasi dkk. [7] mengusulkan metode pembobotan dengan tambahan fitur urutan kata menggunakan metode *Longest Common Subsequence* (LCS) dalam sistem temu kembali dokumen teks yang terintegrasi. Fitur urutan kata ini ditambahkan ke dalam pembobotan Tf-Idf yang sudah didapatkan sebelumnya. Namun penggunaan fitur urutan kata dalam pembobotan ini masih memiliki kekurangan. Hal tersebut dikarenakan terdapat beberapa dokumen yang sangat panjang namun tidak relevan. Dokumen-dokumen ini biasanya melakukan banyak perulangan kata-kata. Sebaliknya, terdapat dokumen yang pendek namun di dalamnya memuat informasi yang penting sehingga urutan kata *query* yang ada dalam dokumen tersebut lebih sedikit. Oleh karena itu, diperlukan proses penyesuaian terhadap fitur urutan kata tersebut dengan melibatkan fitur panjang dokumen sehingga dokumen yang dihasilkan benar-benar memiliki informasi yang relevan.

Pada penelitian ini diusulkan sebuah metode representasi teks dengan fitur urutan kata LCS yang

melibatkan normalisasi dengan rasio panjang dokumen terhadap keseluruhan dokumen dalam korpus. Dengan adanya representasi teks yang lebih baik diharapkan dokumen yang ditemukan oleh sistem ini memiliki relevansi yang tinggi dengan keinginan pengguna.

2. MODEL BERBASIS VEKTOR

Pengukuran tingkat kesamaan merupakan elemen penting dalam mekanisme sistem temu kembali dokumen teks. Pada penelitian ini, fokus paparan terdapat pada metode yang berbasis vektor dan urutan. Dalam metode berbasis vektor, setiap kata dari *query* maupun dokumen direpresentasikan sebagai elemen dalam lingkungan vektor. Representasi tersebut yang akan digunakan untuk menghitung tingkat kesamaan dokumen. Prosedur metode berbasis vektor dibagi menjadi dua tahapan, representasi teks dan perhitungan kesamaan.

Tf-Idf merupakan salah satu metode popular yang digunakan dalam menentukan bobot setiap kata. Bobot tersebut dapat mencerminkan tingkat kepentingannya dalam sebuah dokumen. Bobot setiap kata akan dipetakan dalam lingkungan vektor, sehingga akan terbentuk lingkungan vektor berdimensi *n*. Pembobotan dengan LCS juga merupakan salah satu metode berbasis vektor yang telah diusulkan pada penelitian sebelumnya. Urutan kata antara dokumen dan *query* dijadikan nilai vektor.

Tahapan selanjutnya setelah representasi teks adalah perhitungan kesamaan. Komputasi kesamaan ditujukan untuk menghitung tingkat kesamaan antara *query* dan dokumen. Tiga metode paling umum yang sering digunakan adalah *Cosine*, *Dice*, dan *Jaccard*.

2.1. Pembobotan Tf-Idf

Tf-Idf adalah perhitungan yang menggambarkan seberapa pentingnya kata (term) dalam sebuah dokumen dan korpus. Proses ini digunakan untuk menilai bobot relevansi term dari sebuah dokumen terhadap seluruh dokumen dalam korpus. *Term-frequency* adalah ukuran seringnya kemunculan sebuah term dalam sebuah dokumen dan juga dalam seluruh dokumen di dalam korpus. *Term frequency* ini dihitung menggunakan persamaan (1) dengan adalah *term frequency* ke-*i* dan adalah frekuensi kemunculan term ke-*i* dalam dokumen ke-*j*. Sedangkan *inverse document frequency* adalah logaritma dari rasio jumlah seluruh dokumen dalam korpus dengan jumlah dokumen yang memiliki term yang dimaksud seperti yang dituliskan secara matematis pada persamaan (2) [2]. Nilai didapatkan dengan mengalikan keduanya yang diformulasikan pada persamaan (3).

$$tf(i) = \frac{freq_i(d_j)}{\sum_{i=1}^{k} freq_i(d_j)}, \qquad (1)$$

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|} \qquad (2)$$

$$(tf - idf)_{ij} = tf_i(d_j).idf_i \qquad (3)$$

2.2. Pembobotan LCS

LCS digunakan untuk menghitung relasi berurutan yang paling panjang antara *query* dengan dokumen. Dalam penelitian Tasi dkk. [7], LCS diadopsi di dalam sistem temu kembali dokumen teks sebagai fitur bobot. Nilai LCS antara *query q* dengan dokumen ke-*j* yang telah didapatkan tersebut kemudian dinormalisasi dengan persamaan (4) dengan *m* adalah jumlah term dalam query dan *n* adalah jumlah term di dalam dokumen.

$$LCS(q, j)_{normalisasi} = \frac{LCS(q, j)}{m + n} \qquad (4)$$

Nilai normalisasi LCS ini kemudian digunakan untuk penyesuaian disesuaikan dengan pembobotan sebelumnya yang sudah ada, yaitu dengan bobot yang didapatkan dari Tf-Idf. Nilai bobot akhir untuk dokumen yang memiliki urutan kata sesuai *query* lebih tinggi dibandingkan dokumen yang tidak memiliki urutan kata yang sesuai dengan query. Hal tersebut berdampak pula pada nilai kesamaan antara *query* dengan dokumen. Dokumen yang memiliki bobot urutan kata memiliki nilai kesamaan yang lebih tinggi.

2.3. Perhitungan kesamaan

Perhitungan kesamaan yang umum digunakan dalam metode berbasis vektor diantaranya adalah *Dice*, *Jaccard*, dan *Cosine*. Berdasarkan pembuktian secara matematis oleh Tasi dkk. diketahui bahwa metode *Cosine* tidak tepat digunakan dalam menghitung kesamaan bobot yang melibatkan fitur urutan kata. Di dalam penelitian tersebut, Tasi dkk. menghasilkan kesimpulan bahwa pengukuran kesamaan yang cocok untuk sistem temu kembali dokumen teks dengan pembobotan Tf-Idf dan LCS ini adalah *Dice* dan *Jaccard* [7].

3. MODIFIKASI LCS

Penggunaan fitur urutan kata sebagai pertimbangan penentuan bobot di dalam sistem temu kembali dokumen teks merupakan sebuah kontribusi. Namun hal tersebut masih memiliki kekurangan yang dapat menyebabkan ketidaktepatan hasil pengembalian dokumen. Ketidaktepatan tersebut disebabkan adanya dokumen yang tidak standar

ukuran kandungan informasinya. Terdapat dokumen yang sangat panjang namun tidak memiliki kandungan informasi yang cukup penting. Di dalam dokumen ini terdapat kemungkinan banyak perulangan kata yang memiliki uratan sama dengan *query*. Hal ini menyebabkan peningkatan nilai bobot LCS yang akan berdampak pula pada nilai kesamaan dokumen terhadap *query*. Di lain pihak, terdapat dokumen yang memiliki informasi penting namun panjang dokumennya kecil sehingga nilai bobot LCS dokumen tersebut juga akan bernilai rendah. Hal ini menyebabkan dokumen ini memiliki nilai kesamaan yang kecil. Pada akhirnya akan terjadi ketidaktepatan dokumen yang dikembalikan oleh sistem.

Oleh karena itu, bobot LCS yang digunakan untuk sistem temu kembali dokumen teks ini harus disesuaikan dengan panjangnya dokumen. Dalam penelitian ini diusulkan sebuah modifikasi pembobotan urutan kata yang mempertimbangkan panjang dokumen. Modifikasi LCS ini dihitung menggunakan persamaan (5) dengan *LCS(q, j)* adalah nilai LCS query dengan dokumen ke-*j* dan ndl_j adalah rasio panjang dokumen ke-*j* dengan rata-rata panjang seluruh dokumen di dalam korpus.

$$LCS_{\text{modifikasi}}(j) = \log \frac{(1+LCS(q,j))}{(1+ndl_j)} \qquad (5)$$

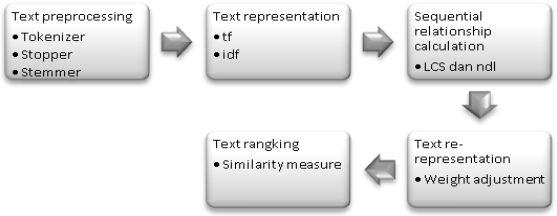
4. SISTEM TEMU KEMBALI DOKUMEN TEKS

Usulan pembobotan baru tersebut diaplikasikan ke dalam sistem temu kembali dokumen teks yang utuh. Secara singkat, alur metodologi dan proses yang ada pada sistem temu kembali dokumen teks tersebut diilustrasikan pada Gambar 1.

Pertama, pada tahap pra-pemrosesan teks dilakukan ekstraksi term dari *query* dan dokumen teks untuk disimpan di repositori. Proses pertama adalah memisahkan *query* dan dokumen teks ke dalam token-token dan mengabaikan tanda baca dan angka. Proses tersebut dinamakan tokenisasi. Kemudian *stoppper* akan menghapus kata-kata di teks yang termasuk dalam *stoplist*. Kata-kata yang termasuk dalam *stoplist* adalah kata-kata tidak penting yang tidak mengandung informasi. Proses selanjutnya adalah *stemming* yaitu mendapatkan kata dasar dari term yang tersisa dari dua proses sebelumnya. *Stemmer* yang digunakan adalah *stemmer* bahasa Indonesia (Jelita A. dkk., 2005).

Kedua, memberikan bobot Tf-Idf term dalam *query* dan dokumen teks pada tahap representatsi teks. Perhitungan bobot Tf-Idf dilakukan pada kata dasar hasil dari tahapan pra-pemrosesan teks. Ketiga, modifikasi LCS dengan penilaian panjang dokumen digunakan untuk menghitung hubungan sekuensial pada *query* dan teks.

Keempat, hubungan sekensial digunakan untuk menyesuaikan bobot Tf-Idf pada tahap *re-representation* teks [7]. Tahap ini digunakan untuk menambahkan bobot Tf-Idf dari term berdasarkan hubungan sekuensial antara *query* dan teks yang nilainya dihasilkan dari modifikasi LCS. Terdapat dua konsep untuk menyesuaikan bobot Tf-Idf. Konsep pertama adalah proses penyesuaian dengan hanya menyesuaikan term sekuensial karena berhubungan dengan informasi sekuensial. Term sekuensial adalah term yang terdapat dalam hubungan sekuensial antara *query* dan teks. Konsep kedua adalah untuk mengurangi perbedaan dalam term sekuensial. Dengan cara ini, bobot term sekuensial bertambah sehingga perbedaan dalam term sekuensial akan berkurang untuk meningkatkan nilai kesamaan antara *query* dan teks.



Gambar 1. Metodologi Sistem Temu Kembali Dokumen Teks

Langkah terakhir, perhitungan kesamaan digunakan untuk menghitung nilai kesamaan antara *query* dan teks dalam tahap perangkingan teks. Pengukuran nailai kesamaan yang digunakan adalah *Dice* sesuai dengan hasil pada penelitian Tasi dkk [7].

5. HASIL UJI COBA DAN ANALISA

Pengujian dilakukan terhadap 140 buah dokumen berita berbahasa Indonesia yang tergolong dalam 11 buah kategori. Untuk mengetahui kemampuan sistem yang diusulkan dilakukan perhitungan nilai presisi dan *recall* terhadap 57 kali uji coba dengan *query* yang berbeda. Nilai presisi dan *recall* sistem yang diusulkan adalah 30,36% dan 96,84%. Nilai presisi merupakan presentase antara jumlah dokumen yang relevan dan mampu dikembalikan oleh sistem dengan jumlah keseluruhan dokumen yang dikembalikan. Sedangkan *recall* adalah presentase antara jumlah dokumen yang relevan dan mampu dikembalikan oleh sistem dengan jumlah keseluruhan dokumen relevan. Oleh karena itu, untuk mengukur sistem ini tidak hanya dilihat dari nilai presisinya, tetapi juga nilai *recall*-nya. Hal ini dikarenakan nilai presisi hanya melibatkan seluruh dokumen yang mampu dikembalikan oleh sistem tanpa melihat dokumen tersebut relevan atau tidak dengan *query*. Walaupun sistem ini hanya

menghasilkan presisi sebesar 30,36%, namun sistem ini mampu bekerja secara efektif dalam mengembalikan sejumlah dokumen teks yang relevan dengan *query* karena menghasilkan *recall* sebesar 96,84%.

Hasil uji coba sistem yang diusulkan juga dibandingkan dengan hasil uji sistem Tasi dkk. Tabel 1 menampilkan rata-rata nilai presisi dan *recall* dari sistem yang diusulkan dibandingkan dengan sistem Tasi dkk. Nilai presisi maupun recall antara kedua sistem sama menunjukkan bahwa metode normalisasi bobot urutan kata dan Tf-Idf tidak berbeda secara signifikan meskipun bobot yang diberikan oleh metode yang diusulkan lebih besar dibanding metode Tasi dkk. Nilai presisi dan *recall* metode yang diusulkan bernilai sama dengan metode Tasi dkk, diakibatkan kondisi pencarian dokumen teks menggunakan *query* pencarian yang sudah berurutan.

Bobot Tf-Idf dijumlahkan dengan bobot LCS hanya menghasilkan perbandingan linier meskipun nilainya lebih besar. Penambahan nilai bobot tersebut terjadi pada hampir keseluruhan dokumen yang memiliki nilai bobot Tf-Idf tinggi secara sebanding.

Tabel 1. Rata-rata nilai presisi dan recall sistem yang diusulkan dengan sistem Tasi dkk.

Nilai rata-rata	Sistem yang diusulkan	Sistem Tasi dkk.
Presisi (%)	**30,32**	**30,32**
Recall (%)	**96,84**	**96,84**

6. KESIMPULAN

Sistem temu kembali dokumen teks pada penelitian ini menggunakan penambahan fitur urutan kata. Percobaan yang dilakukan membuktikan penggunaan fitur urutan kata sama efektifnya dengan sistem temu kembali yang terintegrasi yang digunakan Tasi dkk. Metode yang diusulkan mampu mengembalikan sejumlah dokumen teks yang relevan sesuai dengan permintaan pengguna. Nilai presisi dan recall dari metode yang diusulkan dan metode Tasi dkk bernilai sama sehingga dapat direpresentasikan sebagai kesamaan dari kehandalaan dan efektivitas sistem temu kembali dokumen teks.

7. DAFTAR PUSTAKA

[1] Erenel, Z. & Altincay, H. 2012. Nonlinear transformation of term frequencies for term weighting in text categorization (2012) Engineering Application of Artificial Intelligence 25. Pp. 1505-1514.

[2] Jones, K.S., 1973. Indexing term weighting, Inf. Storage Retr, 9, Pp. 619-633.

[3] Luo, Q., Chen, E. & Xiong, H. 2011.A semantic term weighting scheme for text categorization. 38. Pp. 12708-12716.

[4] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. 1995. Okapi at TREC-3, In D. Harman (Ed.) Proceedings of the third Text Retrieval Conference (TREC-3). Pp. 109-126.

[5] Salton, G., Mc.Gill, M. J. Introduction to Modern Information Retrieval. New York: Mc Graw Hill Book. Co; 1983.

[6] Song, S. & Myaeng S. H. 2012. A novel term weighting scheme based on discrimination power obtained from past retrieval results. Information Processing and Management. 48. Pp. 919-920.

[7] Tasi, Cheng-Shiun, Huang, Yong-Ming, Liu, Chien-Hung, Huang, Yueh-Min. 2012.Applying VSM and LCS to develop an integrated text retrieval mechanism. Expert System with Applications,Pp. .3974-3982.

[8] van Rijsbergen, C.J, Information Retrieval, 2nd ed. London: Butterworths; 1979.