

ANALISIS KOMBINASI ALGORITMA *WEIGHTED TREE SIMILARITY* DENGAN *TANIMOTO COSINE (TC)* UNTUK PENCARIAN SEMANTIK PADA PORTAL JURNAL

Prima Adi P^{*}, Y Sarngadi Palgunadi

Jurusan Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Sebelas Maret

Jl. Ir. Sutami 36A Kentingan Surakarta

^{*}Email: prima101112@gmail.com

Abstrak

Dewasa ini perkembangan teknologi informasi semakin lama semakin pesat. Dalam teknologi informasi, sistem pengelolaan dokumen yang memuat informasi tak akan lepas dari fungsi pencarian. Ada tiga jenis pencarian yaitu pencarian full-text, pencarian menggunakan metadata biasa, dan pencarian semantik. Teknik full-text dan metadata tersebut memiliki kekurangan pada precision atau recall yang kecil. Penelitian ini mengajukan algoritma Weighted Tree Similarity yang digabung dengan algoritma Tanimoto Cosine untuk perhitungan kemiripan pada pencarian semantik. Metode tersebut bertujuan untuk meningkatkan precision maupun recall pada fungsi search. Pada metode ini metadata disusun berdasarkan tree yang memiliki node berlabel, cabang berlabel dan cabang berbobot. Perlakuan khusus diterapkan pada subtree tertentu yang tidak memiliki label cabang. Perhitungan similarity pada subtree tanpa label cabang menggunakan Tanimoto Cosine, sedangkan perhitungan similarity total menggunakan weighted tree similarity. Struktur metadata disusun berdasarkan informasi semantik semacam taksonomi yang didapat dari stemming menggunakan porter stemmer. Dari hasil uji coba didapatkan bahwa kombinasi Weighted Tree Similarity dengan Tanimoto Cosine memiliki precision 100% dan recall 84.44%, hasil ini lebih baik dibanding bila menggunakan salah satunya saja.

Kata Kunci : *Weighted Tree Similarity, Tanimoto Cosine, TC, Pencarian Semantik.*

1. PENDAHULUAN

Teknologi informasi semakin berkembang, jurnal – jurnal tidak hanya berbentuk cetak saja tetapi juga berbentuk digital dan dikelola dengan sistem pengelola dokumen tertentu. Banyak *developer* yang membuat portal untuk mengelola jurnal di dalam suatu instansi atau universitas. Jumlah jurnal akademik terus meningkat seiring waktu, karena pembuatan jurnal menjadi prasyarat kelulusan mahasiswa di semua jenjang. Maka dari itu pengelolaan dan manajemen jurnal yang baik sangat dibutuhkan. Pada sebuah portal jurnal, fungsi *search* merupakan fungsi yang penting. Fungsi *search* digunakan pengguna untuk mencari jurnal akademik yang dibutuhkan untuk mendukung penelitian – penelitian baru yang akan dibuat. Mengingat *search* merupakan fungsi yang cukup penting untuk pengguna portal jurnal, banyak peneliti yang mengembangkan metode *searching*. Ada beberapa tipe *searching* atau pencarian dokumen, diantaranya *full-text searching*, *metadata search* dan *semantic search* (Sarno & Rahutomo, 2008).

Semantic search adalah pencarian berdasarkan makna kata atau kalimat (Sarno, 2012). pencarian semantik memberikan saran bagi pengguna berdasarkan perhitungan dan penarikan kesimpulan yang dilakukan sistem. Sebuah pemodelan data dibutuhkan untuk mendukung pencarian semantik. Bentuk – bentuk pemodelan data untuk *semantic search* antara lain *weighted tree similarity*, *ontology* dan *weighted directed acyclic graph*. Salah satu yang sering digunakan adalah *weighted tree similarity*. *Weighted tree similarity* adalah algoritma yang digunakan untuk mengukur kemiripan dua buah *tree* (Sarno & Rahutomo, 2008). *Tree* sendiri adalah salah satu struktur data dengan membentuk hierarki struktur pohon dengan sejumlah *node* (titik) yang saling berhubungan. *Weighted tree similarity* awalnya digunakan dalam *e-business* untuk mencocokkan keinginan pembeli dengan persediaan penjual yang paling mirip. *Weighted tree* memiliki konsep *node* atau titik yang berlabel, *arc* atau cabang berlabel, dan *arc* berbobot (Sarno, 2012). Pada cabang *subtree keyword*, judul, dan penulis tidak mempunyai label sehingga dibutuhkan algoritma untuk menghitung bobot khusus pada *subtree* tersebut. Perhitungan *similarity* pada *subtree* ini tidak dapat menggunakan *weighted tree similarity* karena tidak mempunyai label cabang. Oleh karena itu digunakan algoritma lain untuk menghitung *similarity subtree* tersebut. Pada penelitian sarno

(2008), pada *subtree* tanpa label cabang digunakan *cosine similarity* untuk menghitung *similarity*-nya. Penelitian ini akan mencoba algoritma *TanimotoCosine (TC)*. Algoritma *TC* adalah algoritma penggabungan dari *tanimoto similarity* dengan *cosine similarity*, diharapkan dapat menghasilkan *precision* dan *recall* yang lebih baik daripada dengan satu algoritma saja.

2. METODOLOGI

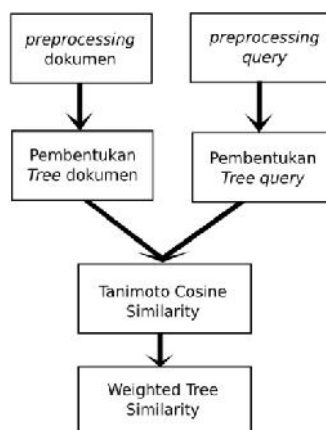
Pelaksanaan penelitian ini melalui beberapa tahapan dan sistematika penelitian mulai dari pengumpulan data, implementasi, sampai dengan analisa hasil dan evaluasi. Berikut ini adalah uraian dari tahapan penelitian ini.

2.1 Pengumpulan Data

Data diperlukan sebagai objek yang akan diolah dan diproses. Data yang dikumpulkan adalah data berupa artikel ilmiah pada jurnal bertipe pdf. Data yang dikumpulkan berjumlah 80 data dengan 40 diantaranya diketahui relevan atau tidaknya terhadap keyword pengujian. Data diambil dari *internet* dan data jurnal dimasukan manual kedalam sistem.

2.2 Implementasi Penelitian

Secara umum langkah kerja dari implementasi penelitian ini diuraikan pada gambar 3.1 di bawah ini



Gambar 1. Langkah kerja implementasi penelitian

proses implementasi meliputi beberapa hal antara lain

2.2.1 Preprocessing Dokumen

Data jurnal yang akan dimasukan dalam database sebelumnya akan dilakukan *preprocessing*. Beberapa tahapan *preprocessing* dokumen text, yaitu tokenisasi, stopword removal, *stemming*, dan perhitungan *term* frekuensi. Hasil dari *preprocessing* merupakan data mentah yang berbentuk *tree* yang sudah berbobot. Struktur *tree* disesuaikan dengan kasus pencarian jurnal ilmiah. Pembobotan akan difokuskan pada *subtree keyword* dan judul. *Tree* tidak diimplementasikan kedalam *WOORuleML*, *tree* direpresentasikan dalam bentuk *array* untuk setiap *subtree* dari *tree* jurnal dan dimasukan kedalam *database*. Penggunaan *array* dimaksudkan agar proses perhitungan lebih cepat. *Stemming* dalam penelitian ini menggunakan *porter stemmer*.

2.2.2 Pembentukan Tree

Dalam penelitian ini pemodelan data *tree* dan implementasi algoritma *weighted tree similarity* dengan *Tanimoto Cosine* dilakukan dengan operasi *array*. Ini dimaksudkan supaya memudahkan perhitungan dan mempercepat proses *retrive*. Bentuk *array* diimplementasikan langsung pada setiap *subtree* dan dimasukan kedalam *database*.

2.2.3 Implementasi Tanimoto Cosine

Dalam penelitian ini implementasi *Tanimoto Cosine* diterapkan saat proses *searching*. Dengan memecah term *query* pengguna menjadi tree dengan struktur tree yang sama dengan tree dalam datanase. *Term query* pengguna akan dibagi kedalam *subtree* judul, keyword, dan penulis. Setiap *subtree* merupakan vektor *term* dimana perhitungannya akan menggunakan *Tanimoto Cosine*. *Tanimoto Cosine* merupakan gabungan antara *cosine similarity* dengan *tanimoto similarity*, persamaan cosine similarity sendiri dapat dilihat pada persamaan berikut.

$$\sum ((S_i)(w_i + w_i')/2) \quad (1)$$

$CosSim(I, J)$ merupakan *cosine similarity* dari *subtree i* dan *j* dengan jumlah vektor *k*. a_i dan a_j adalah bobot dari vektor term yang sama dalam *subtree I* dan *J*. Perhitungan *tanimoto similarity* dapat dilihat pada persamaan berikut

$$T(I, J) = \frac{\sum_k a_i a_j}{\sqrt{\sum_k a_i^2} + \sqrt{\sum_k a_j^2} - \sum_k a_i a_j} \quad (2)$$

$T(I, J)$ adalah *tanimoto similarity* dari *subtree I* dan *J* dengan jumlah vektor *k*. a_i dan a_j adalah bobot dari vektor term yang sama dalam *subtree I* dan *J*. Perhitungan *Tanimoto Cosine* dapat dirumuskan sebagai berikut

$$TC(Z_i, Z_j) = T(Z_i, Z_j) \times CosSim(Z_i, Z_j) \quad (3)$$

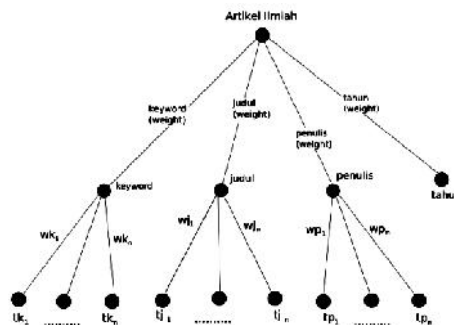
Dalam persamaan tersebut $T(Z_i, Z_j)$ adalah perhitungan similarity menggunakan *tanimoto similarity* dan $CosSim(Z_i, Z_j)$ adalah perhitungan similarity menggunakan *cosine similarity*. Penggabungan keduanya terbukti lebih akurat (Sharma, 2011).

2.2.4 Implementasi Tanimoto Cosine

Weighted tree similarity digunakan untuk menghitung similarity total dari 2 buah tree. Setelah di dapatkan similarity dari 3 *subtree* utama maka akan dihitung similarity total dengan persamaan berikut.

$$\sum ((S_i)(w_i + w_i')/2) \quad (4)$$

Bobot masing – masing cabang dalam tree dokumen statis dan sama untuk setiap dokumen. Yaitu, judul 0.3, keyword 0.4, penulis 0.2, tahun 0.1. keyword mempunyai bobot paling besar karena tujuan pencarian adalah keyword atau kata kunci jurnal. Gambar 2 merupakan gambaran tree dokumen.



Gambar 2. Penyajian tree dokumen

2.3 Analisis Hasil dan Evaluasi

Setelah dimasukan data kedalam database dan data dimodelkan dalam bentuk tree, dilakukan perhitungan *precision* dan *recall*. Perhitungan ini dimaksudkan untuk mengetahui baik atau tidaknya metode *search* yang digunakan. *Precision* dan *recall*.

Precision adalah perbandingan antara hasil *search* yang relevan terhadap semua data yang berhasil di *retrive*. Sedangkan *recall* adalah perbandingan hasil relevan yang berhasil di *retrive* terhadap semua data relevan yang ada di database. keduanya dapat dirumuskan sebagai berikut.

$$recall = \frac{S_r}{D_r} \quad precision = \frac{S_r}{S} \quad (5)$$

Dengan S_r adalah jumlah hasil *search* yang relevan, D_r adalah jumlah semua data dalam database yang relevan, dan S adalah seluruh data yang berhasil di *retrive*.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Penentuan Batas *Similarity*

Setelah dilakukan implementasi, sistem diuji dengan keyword tertentu untuk menentukan batas relevan dari hasil *search*. *Keyword* dimasukan dan dianalisa hasil *search* yang muncul. Dengan demikian dapat ditentukan batas *similarity* untuk data relevan. Tabel 1 merupakan tabel batas *similarity*.

Tabel 1. Tabel batas similarity

No	Keyword	Cosine	Tanimoto
1	Wireless security	0.276	0.045
2	Semantic search	0.483	0.069
3	aca	0.111	0.004
4	topsis	0.117	0.003
	Rata-rata	0.247	0.030

Dari batasan *similarity* yang didapat, dapat ditentukan batasan *similarity* untuk *tanimoto cosine* dengan cara menerapkan rumusan *Tanimoto Cosine* pada persamaan (3) adalah 0.007. Dengan pembulatan kebawah maka batas untuk masing masing metode adalah 0.25 untuk *cosine similarity* 0.03 untuk *tanimoto similarity* dan 0.01 untuk *Tanimoto Cosine*.

3.2 Hasil Pengujian

Setelah ditentukan batas *similarity* minimal untuk masing masing kombinasi algoritma maka dilakukan pengujian dengan 10 *keyword* tertentu. *Keyword* untuk pengujian ini ditentukan oleh penulis. Guna menghitung *recall* maka perlu diketahui jumlah data yang relevan untuk sebuah *keyword* dalam database. Analisis dilakukan terhadap hasil pengujian pencarian dengan menghitung *precision* dan *recall*. Berdasarkan persamaan (5). *precision* dan *recall* merupakan variabel untuk menguji metode pencarian. Besarnya *precision* dan *recall* menunjukan baik tidaknya metode pencarian. Perbandingan *precision* ditampilkan pada tabel 2 berikut

Tabel 2. Perbandingan Precision

No	Keyword	Weighted Tree Similarity +		
		Cosine	Tanimoto	TC
1	ant colony	100.00%	100.00%	100.00%
2	citizenship education	100.00%	100.00%	100.00%
3	data mining	100.00%	100.00%	100.00%
4	Network Security	100.00%	100.00%	100.00%
5	tanimoto coefficient	100.00%	100.00%	100.00%
6	weighted tree similarity	100.00%	100.00%	100.00%
7	fuzzy logic	100.00%	100.00%	100.00%
8	data hiding with histogram	100.00%	100.00%	100.00%
9	ontology	100.00%	0.00%	100.00%
10	shape matching	100.00%	100.00%	100.00%
	Rata – rata	100.00%	90.00%	100.00%

Dari tabel 2 menunjukkan bahwa ketiga kombinasi metode baik untuk pencarian semantik pada portal jurnal dalam segi *precision*. Ini terlihat dari rata-rata *precision* dari setiap metode cukup besar, bahkan pada *cosine similarity* dan *tanimoto cosine* rata-rata *precision* 100% sempurna untuk metode pencarian. Sedangkan pada *tanimoto similarity* rata-rata hanya 90% dikarenakan pada salah satu keyword pengujian *tanimoto* tidak memberikan result satupun. *Precision* bisa bernilai besar dikarenakan keyword tidak diambil secara random melainkan diambil langsung dari koleksi database. Dalam metode *searching* selain *precision* terdapat variabel pengujian lain yaitu *recall*. Perhitungan *recall* dari hasil nya dipaparkan pada Tabel 3.

Tabel 3. Perbandingan Recall

No	Keyword	Weighted Tree Similarity +		
		Cosine	Tanimoto	TC
1	ant colony	33.33%	33.33%	66.67%
2	citizenship education	100.00%	100.00%	100.00%
3	data mining	100.00%	75.00%	100.00%
4	Network Security	90.00%	90.00%	100.00%
5	tanimoto coefficient	50.00%	50.00%	50.00%
6	weighted tree similarity	80.00%	80.00%	100.00%
7	fuzzy logic	55.56%	11.11%	77.78%
8	data hiding with histogram	100.00%	100.00%	100.00%
9	ontology	100.00%	0.00%	50.00%
10	shape matching	100.00%	100.00%	100.00%
	Rata – rata	80.89%	63.94%	84.44%

Pada tabel 3 menunjukkan bahwa kombinasi antara *weighted tree similarity* dengan *tanimoto cosine* merupakan kombinasi paling baik diantara ketiganya dalam segi recall. Ditunjukkan dengan rata-rata *recall* dari 10 keyword pengujian 84.44% sedangkan *cosine* 80.89% dan *tanimoto* hanya memiliki rata-rata *recall* 63.94%.

Hasil ini pada tabel 2 dan tabel 3 menunjukkan bahwa kombinasi antara *weighted tree similarity* dengan *Tanimoto Cosine* merupakan kombinasi paling baik untuk pencarian semantik pada portal jurnal. Ditunjukkan dengan *precision* dan *recall* yang paling besar diantara ketiga kombinasi metode.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan penelitian, hasil serta pembahasan yang telah dilakukan dan dipaparkan diatas maka dapat ditarik kesimpulan bahwa kombinasi *weighted tree similarity* dengan *Tanimoto Cosine* menghasilkan hasil pencarian yang lebih baik. Hal ini dibuktikan dengan tingginya *precision* yang mencapai 100% dan *recall* 84.44%. Hasil ini lebih baik dari dua kombinasi lainnya.

4.2 Saran

Adapun saran yang dapat diberikan pada penelitian ini adalah penambahan deteksi sinonim per-kata pada leaf node agar hasil pencarian lebih relevan dan pembobotan dinamis sesuai *search query* dengan metode tertentu untuk meningkatkan akurasi pencarian.

DAFTAR PUSTAKA

- Cha, Sung-Hyuk ; Choi, Seungseok ; Tappert, Charles C. 2009. *Anomaly between Jaccard and Tanimoto Coefficients. Proceedings of Student-Faculty Research Day*, CSIS, Pace University.
- Clarke, S., & Willett, P. 1997. *Estimating the recall performance of search engines. ASLIB Proceedings*, 49 (7), 184-189.
- Halim, Yosephine; Somya, Ramos ; Fibriani, Charitas. 2012. *Algoritma Extended Weighted Tree Similarity untuk Memberikan Solusi Memasak pada J2ME*. JdC, Vol. 1, No 1.

- Jeffrey Beall. 2008. *The Weaknesses of Full-Text Searching*. *The Journal of Academic Librarianship*, Vol.34, No.5:438-444.
- Jiang, Xing ; Tan, Ah-Hwee. *OntoSearch: A Full-Text Search Engine for the Semantic Web*. School of Computer Engineering Nanyang Technological University.
- Jin, Jing et al. 2005. *Towards a Weighted-Tree Similarity Algorithm for RNA Secondary Structure Comparison*. IEEE. Proceedings of the Eighth International Conference on High-Performance Computing in Asia-Pacific Region 2005.
- Jizba ; Richard. 2007. *Measuring Search Effectiveness*. Search Techniques & Strategies. Creighton university.
- Powers, D.M.W. 2011. *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Marketness & Correlation*. *Journal of Machine Learning Technologies*. Vol. 2 No. 1 : 37-63.
- Sa'adah, Umi ; Sarno, Riyanarto ; Yuhana1, Umi L. 2013. *Latent Semantic Analysis and Weighted Tree Similarity For Semantic Search In Digital Library*. The Proceedings of The 7th ICTS, Bali. 159 – 164.
- Sarno, Riyanarto; Rahutomo, Faisal. 2008. *Penerapan Algoritma Weighted Tree Similarity Untuk Pencarian Semantic*. JUTI. Vol. 7 No.1 : 35-42.
- Setyawan, Sholeh Hadi ; Sarno, Riyanarto. 2005. *Fuzzy Logics Incorporated to extended Weighted Tree Similarity Algorithm for Agent Matching in Virtual Market*. Information and Communication Technology Seminar, Vol. 1 No. 1:49-54.
- Sharma, Alok ; Lal, Sunil P. 2011. *Tanimoto Based Similarity Measure for Intrusion Detection System*. *Journal of Information Security*. Vol 2 : 195-201
- Srividhya, V. ; Anitha, R. 2010. *Evaluating Preprocessing Techniques in Text Categorization*. International Journal of Computer Science and Application Issue 2010. 49-51.
- Sudeepthi1, G; Anuradha, G; Babu, MS. Prasad. 2012. *A Survey on Semantic Web Search Engine*. IJCSI Vol.9 No.1 : 241-245.
- Sukarsa. I Made ; Putra, I Made Bayu Dwi ; Sasmita, I Gusti Made Arya. 2013. *Weighted Tree Similarity Semantic Search For E-Commerce Content*. Journal of Theoretical and Applied Information Technology. JATIT & LSS. Vol. 55 No.3: 327-335.
- Willett, Peter. 1998. *Chemical Similarity Searching*. *J. Chem. Inf. Comput. Sci.*, Vol. 38, No. 6: 983-996.