Weight Adjusted K-Nearest Neighbor dan *Minimum Spanning Tree* untuk *Information Retrieval System* di Perpustakaan STMIK PPKIA Tarakanita Rahmawati Tarakan

Aida Indriani
Program Studi Teknik Informatika,
STMIK PPKIA Tarakanita Rahmawati
Jl. Yos Sudarso 8 Tarakan, Indonesia
m31d1t4_03@yahoo.com

Gunawan
Program Studi Teknologi Informasi,Institut Saint Terapan dan Teknologi Surabaya
Jl. Ngagel Jaya Tengah 73-77 Surabaya, Indonesia
gunawan@stts.edu

Endyk Novianto
Program Studi Teknik Informatika,
STMIK PPKIA Tarakanita Rahmawati
Jl. Yos Sudarso 8 Tarakan, Indonesia
endyknov@yahoo.co.id

Abstract*—*Information Retrieval System is a process of searching for a set of documents relevant with needs of the user. In the teaching and learning activities, Information Retrieval System much needed. For example at library, teachers and students can search for books that can be used as reference material. With the Information Retrieval System can improve search results more relevant book to the needs of the user.

Library of STMIK PPKIA Tarakanita Rahmawati domiciled in Tarakan, until now have 4215 ± titles consisting of printed books and research. In this research, using 500 data books and software used to create an IR system that is Sphinx Search. The role of Sphinx Search in a IR system is the process of indexing for data set books and search for data relevant books matching the query entered by user.

Methods classification text used in this research are Weight Adjusted K-Nearest Neighbor (WAK-NN). Percentage accuracy of match type classes generated by WAK-NN is 81% on 150 data test with processing time by ± 11 minutes / book. By making use of classification text, it can improve the performance of the effectiveness of the results data books are displayed by Sphinx Search. With produce value of precision 63.6% for 5 different queries. Other than displaying the data books that match the query, in this research can also display a list of books that may be used as reference material / other teachings. The establishment of documents cluster made with method Clustering Minimum Spanning Tree (MST) using the formula Cosine Distance Measure.

Keywords – WAK-N; Information Retrieval System; Sphinx Search; Minimum Spanning Tree

I. PENDAHULUAN

Berdasarkan kenyataan belakangan ini, perkembangan Internet di Indonesia menjadi sebuah penetrasi yang cukup tinggi. Pengguna Internet baru terus bermunculan. Banyaknya Informasi yang tersebar di Internet, tidak bisa dilepaskan dari sebuah media pencarian (*search engine*) tentang apa yang tersaji di dalamnya. Dimulai dari debut Yahoo yang kemudian perkembangannya dikejar dengan kepopuleran Google dan berikutnya ingin diikuti oleh Bing, yang menjadi mesin pencarian buatan Microsoft.

Bentuk pencarian buku saat ini yang diterapkan pada STMIK PPKIA Tarakanita Rahmawati adalah pencarian standar dengan memanfaatkan perintah "like" sebagai sintaks standar pada *Structured Query Language* (SQL) untuk menyajikan hasil informasi yang sesuai dengan *keyword* yang dimasukkan oleh pengguna. Kelemahan fungsi "like" ini nantinya hanya mengacu pada persamaan kata pada judul buku/tugas akhir saja sebagai bentuk sumber penunjukan hasil yang ada, namun tidak disajikan informasi lain yang berkemungkinan dapat disajikan sebagai bentuk solusi hasil.

Disiplin ilmu *Library Information Retrieval* (LIR) dan *Information Retrieval (IR*) menjadi titik tolak metode penting yang dimanfaatkan pada penelitian ini. Kajian terhadap metode yang digunakan dalam pencarian dokumen berdasarkan representasi kebutuhan informasi berupa kata kunci, yaitu *keyword* sebagai bentuk *query* yang diterapkan dengan pemanfaatan Sphinx Search sebagai sistem indexing, *Weight Adjusted K-Nearest Neighbor* (WAK-NN) untuk melakukan proses klasifikasi dari seluruh dokumen buku yang menjadi inputan dan metode *Minimum Spanning Tree* (MST) yang mampu menyajikan informasi yang berrelasi dengan apa yang akan dicari.

II. LANDASAN TEORI

2.1 Klasifikasi Weight Adjusted K-Nearest Neighbor

[1]Algoritma *K-Nearest Neighbor* (K-NN) mengklasifikasikan dokumen uji berdasarkan k tetangga terdekat. Contoh-contoh pelatihan dapat dianggap sebagai vektor dalam ruang fitur multidimensi. Ruang ini dipartisi menjadi daerah dengan lokasi dan label pelatihan sampel. Sebuah titik dalam ruang ditetapkan ke kelas dimana sebagai besar poin pelatihan milik kelas yang dalam k terdekat pelatihan sampel. Biasanya *Euclidean distance* atau *Cosine similarity* digunakan. Selama tahap klasifikasi, data pelatihan (kelas yang perlu diidentifikasi) juga direpresentasikan

sebagai vektor dalam ruang fitur. Jarak atau kesamaan dari vektor uji untuk semua vektor pelatihan dihitung dan k sampel pelatihan terdekat yang dipilih.

[2] Salah satu varian dari K-NN adalah algoritma *Weight Adjusted K-Nearest Neighbor* (WAK-NN). Algoritma WAK-NN lebih mempertimbangkan pembobotan pada atribut untuk meningkatkan akurasi klasifikasi. Algoritma ini lebih banyak digunakan untuk dataset berupa teks. Dalam algoritma ini kata pembeda yang diberi bobot lebih dan penting. Algoritma ini untuk membedakan kata-kata pembeda dengan memeriksa hubungan informasi antara kata dan label kelas.

[3] Setiap pola yang mengatur bobot dari berbagai fitur (yakni, terms) harus melakukan dua tugas. Pertama, harus menggolongkan berbagai terms sesuai dengan klasifikasi. Kedua, harus menyesuaikan bobot dari berbagai terms dengan klasifikasi yang telah tersedia. [4] Kunci utama dari WAK-NN adalah meng-inisialisasi bobot vektor, bagaimana menemukan bobot vektor terbaik dan bagaimana menggunakan bobot vektor untuk menentukan kemiripan yang lebih baik.

Untuk pengklasifikasian teks dengan WAK-NN digunakan perhitungan *Weight Initialization Using Mutual Information* (WIUMI) yang berfungsi untuk mendapatkan nilai bobot vector. Formula untuk *Weight Initialization Using Mutual Information* adalah sebagai berikut.

$$\text{MI(w)} = \sum_{c \in C}(P(c,w)\log\frac{P(c,w)}{P(c)P(w)} + P(c,\overline{w})\log\frac{P(c,\overline{w})}{P(c)p(\overline{w})})$$

dimana P(c) adalah probabilitas dari class c, P(w) adalah probabilitas dari adanya kata, P($\overline{w}$) adalah probabilitas dari tidak adanya kata, P(c,w) dan P(c, $\overline{w}$) adalah probabilitas gabungan.

Setelah bobot vector diketahui, maka langkah berikutnya adalah perhtiungan *Weighted Cosine Similarity Measure* (WCSM) yang berfungsi untuk mendapatkan nilai *cosine similarity* suatu dokumen. Formula untuk *Weighted Cosine Similarity Measure* adalah sebagai berikut.

$$\cos(X,Y,W) = \frac{\sum_{t \in T}(X_t \; x \; W_t) x \; (Y_t \; x \; W_t)}{\sqrt{\sum_{t \in T}(X_t \; x \; W_t)^2} \; x \; \sqrt{\sum_{t \in T}(Y_t \; x \; W_t)^2}}$$

dimana Xt dan Yt adalah TF normalisasi t kata untuk X dan Y, masing-masing, dan Wt adalah berat t kata.

Setelah nilai *cosine similarity* suatu dokumen diketahui, langkah berikutnya adalah fungsi objektif yaitu *Weight Adjustment Based on Objective Function* (WABoOF) yang berfungsi untuk menghitung inisialisasi bobot vektor terbaik. Formula untuk *Weight Adjustment Based on Objective Function* adalah sebagai berikut.

$$\text{Obj (D, W, p)} = |\{d|d \in D \text{ and Correct(d, D, W, p)}\}|$$

di mana D adalah matriks dokumen pelatihan, W adalah vektor bobot, dan predicat Correct(d,D,W,p) adalah benar jika dari k tetangga terdekat dari d dari D dihitung menggunakan ukuran kosinus tertimbang, tetangga mayoritas berasal dari kelas yang sama seperti d dan jumlah dari kesamaan dengan tetangga ini mayoritas paling tidak p persen dari jumlah tetangga k kesamaan jumlah.

2.2 Sphinx Search

[5] *Sphinx search* adalah *full text search engine* yang merupakan salah satu teknik untuk melakukan pencarian dokumen atau database yang disimpan dalam komputer. Selama pencarian mesin pencari melewati dan memeriksa seluruh kata yang ada pada dokumen dan mencoba untuk mencocokkan kata-kata tersebut dengan query yang diberikan.

Untuk mempermudah pengguna dalam menggunakan *library sphinx search* maka *sphinx search* menyediakan fasilitas yang dapat membantu pengguna dalam pemakaiannya. Adapun fasilitas utama dari *sphinx search* antara lain:

1. **indexer**, untuk membuat indeks dalam format *full-text*.
2. **search**, *command line* untuk melakukan (mencoba) *query* terhadap hasil indeks.
3. **searchd**, *daemon* untuk memproses pencarian dari perangkat lunak lain, misalnya skrip web.
4. **sphinxapi**, pustaka API untuk bahasa pemrograman berbasis web, baru tersedia untuk PHP

[6] Selain fasilitas, Sphinx juga mempunyai fitur-fitur yang dapat digunakan dalam *proses indexing* dan *searching*. Adapun fitur-fitur utama dari *sphinx search* adalah sebagai berikut:

1. Pengindeksan dan *searching* yang tinggi
2. Pengindeksan yang terkini dan *query tools* (fleksibel dan banyak fitur *text tokenizer*, bahasa *query-query language*, beberapa mode rangking yang berbeda,dll);
3. Hasil set *post-processing* yang maju (pernyataan dgn SELECT, WHERE, ORDER BY, GROUP BY, dll lebih dari hasil pencarian *text*);
4. Membuktikan *scalability* hingga milyaran dokumen, terabytes data, dan ribuan *query* per detik;
5. Mudah diintegrasikan dengan sql dan xml *data source*;
6. Kemudahan *scaling* dengan pencarian yang dibagi-bagi (*distributed search*).

[7] Untuk perangkingan dokumen pada *Sphinx Search*, digunakan pembobotan BM25 atau Okapi. Pembobotan BM25 adalah pembobotan yang mengurutkan set dokumen berdasarkan term kueri yang muncul pada setiap dokumen koleksi. Hubungan antara *term* kueri dengan dokumen dipengaruhi oleh parameter k1 (parameter untuk kalibrasi skala frekuensi *term*) dan parameter b (parameter untuk kalibrasi skala panjang dokumen). Nilai parameter yang optimal untuk pembobotan BM25 adalah k1=1.2 dan b=0.75. Penghitungan bobot suatu dokumen berdasarkan *term* t dinyatakan dalam persamaan.

$$score(Q, D_i) = \sum_{t \in q} \log\left[\frac{N}{df_t}\right] \cdot \frac{(k_1+1)tf_{ti}}{K + tf_{ti}}$$

dengan $K = k_1 ((1 - b) + (b \times (L_{di} / L_{ave})))$;

Keterangan:

$\log\left[\frac{N}{df_t}\right]$ adalah *inverse document frequency*

tf_{ti} adalah frekuensi *term* t pada dokumen i

Ldi dan Lave adalah panjang dokumen Di dan rata-rata panjang dokumen dalam koleksi

K1 dan b adalah parameter pengskalaan terhadap tf dan panjang dokumen

2.3 Clustering Minimum Spanning Tree

[8] *Cluster* didefinisikan sebagai upaya pengelompokkan data ke dalam *cluster* sehingga data-data didalam *cluster* yang sama memiliki lebih kesamaan data dibandingkan dengan data-data pada *cluster* yang berbeda. [9] Kategori algoritma *cluster* yang banyak dikenal adalah *Hierarchical Clustering*. *Hierarchical clustering* adalah salah satu algoritma *clustering* yang dapat digunakan untuk meng-*cluster* dokumen (*document clustering*).

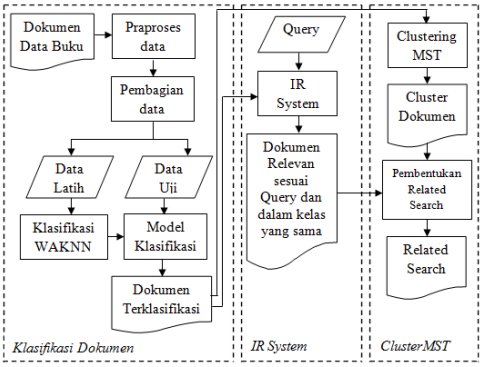
[10] Salah satu *Hierarchical Clustering* adalah *Minimum Spanning Tree*. Metode hirarkhis dimulai dengan mempertimbangkan setiap komponen dari populasi untuk menjadi sebuah *cluster*. Selanjutnya, dua kelompok dengan jarak minimum antara mereka yang menyatu untuk membentuk satu cluster. Proses ini diulang sampai semua komponen dikelompokkan ke dalam jumlah yang diperlukan akhir dari *cluster*. Untuk menghitung jarak antara setiap dua dari centroid dengan menggunakan *Cosine Distance Measure*, dengan formula sebagai berikut.

$$Cos\ (W_1, W_2) = \frac{\sum_{i=1}^{n} p_i\ q_i}{\sqrt{\sum_i^n p^2\ \sum_i^n q^2}}$$

dimana p dan q merupakan nilai normalisasi untuk masing-masing *term* untuk dokumen p dan q.

III. PERANCANGAN SISTEM

Bahan yang digunakan pada penelitian ini adalah 500 data buku dari Perpustakaan STMIK PPKIA Tarakanita Rahmawati. Data yang akan diindex berupa sinopsis buku. Arsitektur sistem digunakan untuk menggambarkan sistem kerja yang digunakan pada proses analisa dan implementasi, adapun arsitektur sistem dari keseluruhan sistem yang digunakan ditunjukkan pada Gambar 1.



Gambar 1. *Architecture System*

Dari dokumen data buku dilakukan praproses data (tokenisasi, *stop words*, *stemming*), setelah melalui praproses data dilanjutkan dengan membagi data menjadi 2 bagian yaitu data latih dan data uji. Data latih (70% dari jumlah dokumen) digunakan untuk mendapatkan dokumen yang terklasifikasi dengan menggunakan metode *Weight Adjusted* k-*Nearest Neighbor* untuk mendapatkan pola model klasifikasi yang nantinya akan digunakan untuk data uji. Data uji (30% dari jumlah dokumen) digunakan sebagai bahan untuk menguji dalam penentuan klasifikasi dokumen dengan menggunakan model klasifikasi yang telah didapatkan dari data latih sehingga diperoleh dokumen terklasifikasi.

Dokumen terklasifikasi (dari klasifikasi dokumen) digunakan sebagai sumberdaya, bersama dengan *query* untuk proses *indexing* yang dilakukan oleh *library Information Retrieval System* (dalam penelitian ini menggunakan *library Sphinx Search*) yang merupakan tugas pokok pada tahapan *pre-processing* di dalam *Information Retrieval*. Dari hasil pengindeksan koleksi dokumen sesuai dengan *query*, dapat diperoleh dokumen-dokumen yang relevan sesuai *query* dan dalam kelas yang sama.

Pembentukan *cluster related search* dapat diperoleh dengan pembentukan *cluster-cluster* yang dibuat berdasarkan hasil dokumen relevan sesuai *query* dan dalam kelas yang sama dengan menggunakan metode *Clustering Minimum Spanning Tree* (MST).

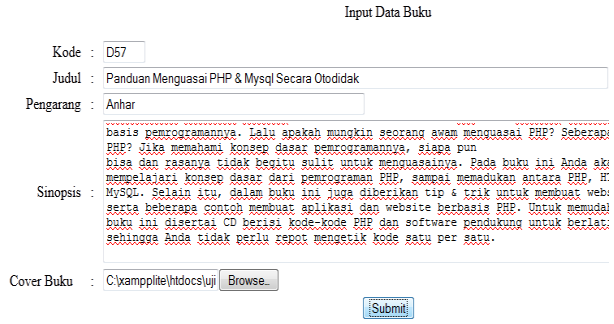
IV. HASIL DAN PEMBAHASAN

1. Klasifikasi Dokumen

Proses klasifikasi terhadap 500 data buku dilakukan dengan membagi data menjadi 2 bagian, yaitu data latih dan data uji. Model klasifikasi yang digunakan dalam data uji diperoleh dari klasifikasi WAK-NN untuk data latih sebanyak 350 data. Dengan menggunakan model klasifikasi dari data latih, data uji sebanyak 150 data menghasilkan prosentase akurasi kecocokan jenis kelas yang dihasilkan oleh WAK-NN sebesar 81% dengan waktu proses sebesar ± 11 menit/buku.

Untuk menentukan jenis klasifikasi dari data buku baru, dimulai dengan melakukan penginputan data buku. Proses

input data buku ditunjukkan pada Gambar 2. Dokumen data buku yang digunakan untuk menentukan jenis klasifikasi meliputi kode buku, judul, pengarang, sinopsis dan pemilihan cover buku. Untuk klasifikasi sendiri didapatkan setelah melalui proses pengklasifikasian dengan menggunakan Metode WAK-NN.



Gambar 2. *Form Application of New Document*

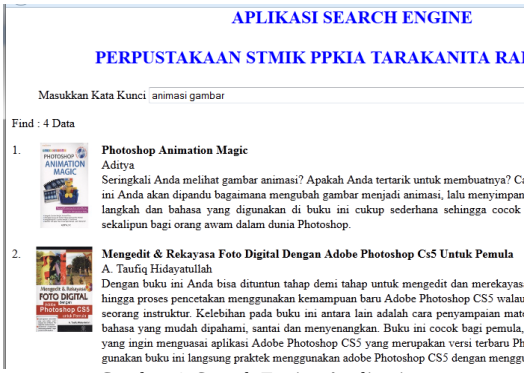
Pada Gambar 2, petugas bagian perpustakaan melakukan input data buku baru dengan memasukkan kode buku, judul, pengarang, sinopsis dan cover buku. Setelah semua dimasukkan, langkah berikutnya yaitu memilih tombol submit. Setelah melewati semua tahap input data buku, akan tampil data buku baru termasuk jenis kelas yang ditampilkan secara otomatis dengan melalui proses perhitungan klasifikasi metode WAK-NN. Tampilan data buku baru termasuk jenis kelas yang ditampilkan ditunjukkan pada gambar 3.

Gambar 3. *New Document and Types Class*

Pada Gambar 3, dapat dilihat bahwa dokumen D60 dengan judul buku "Menyusun Laporan Keuangan & Auditing di Excel" termasuk jenis kelas "Application Software for Business" atau "K1". Apabila jenis kelas sudah sesuai dengan isi buku maka proses penentuan klasifikasi telah berhasil, jika tidak sesuai petugas perpustakaan dapat mengganti jenis kelas yang sesuai dengan isi buku dengan cara memilih list klasifikasi yang ada pada tampilan kemudian diakhir dengan memilih tombol simpan.

2. Searching

Pada *search engine* sederhana hanya memiliki 1 (satu) kolom *input query/keyword*, sedangkan untuk fitur MatchMode dan RangkingMode yang digunakan adalah SPH_MATCH_ANY untuk fitur MatchMode dan SPH_RANK_BM25 untuk fitur RangkingMode. Berikut adalah tampilan aplikasi *Search Engine* Perpustakaan STMIK PPKIA Tarakanita Rahmawati ditunjukkan pada gambar 4.



Gambar 4. *Search Engine Application*

Pada Gambar 4, pengguna memasukkan *query* "animasi gambar", setelah meng-klik tombol *search* dokumen yang sesuai dengan *query* dan dalam kelas yang sama ditemukan 4 data.

Peran klasifikasi dokumen dengan WAK-NN dalam IR System dapat meningkatkan efektifitas kinerja yang lebih baik dibandingkan dengan IR *System* dengan *Vector Space Model* (VSM). Pengukuran efektifitas kinerja dapat dilakukan dengan menghitung nilai *precision* dan *recall* seperti yang ditunjukkan pada Tabel I.

TABEL I. *VALUE OF PRECISION AND RECALL*

No.	Query (kata kunci)	IR System dengan VSM		IR System dengan Sphinx Search dan WAKNN	
		Precision	Recall	Precision	Recall
1	Animasi Gambar	20%	100%	64%	100%
2	Fuzzy Logic	77%	100%	90%	90%
3	Jaringan Linux	18%	100%	40%	75%
4	Mobile Ponsel	38%	100%	57%	80%
5	Citra Digital	14%	100%	67%	100%
Rata-Rata Precision dan Recall		**33,4%**	**100%**	**63,6%**	**89%**

3. Clustering

Pada penelitian ini, *clustering Minimum Spanning Tree* digunakan untuk menghitung kemiripan antara dokumen yang dihasilkan oleh *sphinx search* sesuai dengan *query* pengguna dan dalam kelas yang sama terhadap dokumen lainnya untuk membentuk suatu daftar *related search* yaitu berupa daftar buku yang mungkin dapat dijadikan referensi oleh pengguna. Pembentukan *cluster* daftar buku lainnya ditunjukkan pada Gambar 5.

Gambar 5. *The Result of Cluster*

V. SIMPULAN

Dari pembahasan pada bab–bab sebelumnya pada penelitian ini, dapat diambil kesimpulan sebagai berikut :

1. Dengan menggunakan metode klasifikasi teks yaitu metode *Weight Adjusted K-Nearest Neighbor* (WAK-NN), dari segi kecocokan jenis kelas yang dihasilkan oleh WAK-NN terhadap penentuan jenis kelas yang dilakukan oleh Bagian Perpustakaan tergolong baik. Prosentasi akurasi kecocokan jenis kelas terhadap 150 data uji sebesar 81% dengan waktu proses sebesar ± 11 menit/buku. Untuk data buku baru, tingkat prosentasi kecocokan sebesar 90% terhadap 10 data buku baru dengan waktu proses sebesar ± 72,5 detik/buku terhadap 70 dokumen.
2. Dengan pengklasifikasian data buku termasuk tugas akhir dengan metode WAK-NN dapat membantu meningkatkan efektifitas kinerja dari *Information Retrieval System* (IR *System*) Sphinx Search. Dari uji coba yang dilakukan, IR *System* dengan WAK- NN memperoleh nilai *precision* sebesar 63,6 % lebih tinggi dibandingkan dengan nilai *precision* dari IR *System* dengan *Vector Space Model* (VSM) yaitu sebesar 33,4%.
3. Pembentukan *Related Search* dari dokumen yang dihasilkan oleh *Sphinx Search* dapat dilakukan dengan menggunakan metode *Clsutering Minimum Spanning Tree* dan rumus *Cosine Distance Measure* yang digunakan untuk menghitung jarak antar dokumen. Proses pembentukan dokumen *cluster* membutuhkan waktu ± 14 jam untuk 500 dokumen.

DAFTAR PUSTAKA

[1] Muhammed Miah. Improved k-NN Algorithm for Text Classification. University of Texas at Arlington, TX, USA.

[2] Mehdi Moradian, Aahmad Baraani. 2009. KNNBA: k-Nearest-Neighbor-Based-Association Algorithm. Journal of Theoretical and Applied Information Technology. Vol6 No. 1. 123-129.

[3] Shrikanth Shankar, George Karypis. A Feature Weight Adjustment Algorithm for Document Categorization. Minneapolis, MN 55455, USA.

[4] Eui-Hong (Sam) Han, George Karypis, Vipin Kumar. 1999. Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification. Minneapolis, MN 55455, USA.

[5] Arief Purwanto, Yan. Sphinx, SQL full-text Search Engine. http://www.yanrf.com/blog/65/sphinx-sql-full-text-search-engine. (diakses tanggal 15 Maret 2012).

[6] Sphinx Technologies Inc. 2008. Sphinx2.0.4-release reference manual. http://sphinxsearch.com. (diakses tanggal 15 Maret 2012).

[7] Kristina Paskianti. 2011. Klasifikasi Dokumen Tumbuhan Obat menggunakan Algoritma KNN Fuzzy. Thesis Fakultas Matematika dan Ilmu Pengetahuan Alam IPB. Bogor.

[8] Amir Hamzah. Temu Kembali Informasi berbasis Kluster untuk Sistem Temu Kembali Informasi Teks Bahasa Indonesia. Jurusan Teknik Informatika, Fakultas Teknologi Industri. Institut Sains & Teknologi AKPRIND Yogyakarta.

[9] Gregorius S. Budhi, Arlinah I. Rahardjo, Hendrawan Taufik. 2008. Hierarchical Clustering untuk Aplikasi Automated Text Integration. Seminar Nasional Aplikasi Teknologi Informasi 2008 (SNATI 2008) Hal. C-27-C-32.

[10] Yixing Sun. 2007. Using the Organizational and Narrative Thread Structures in an e-Book to Support Comprehension. http://www.comp.rgu.ac.uk/staff/sy/PhD_Thesis_html/page_67.htm (diakses tanggal 07 Januari 2013).