Volume 5, No. 2, Oktober 2012

ISSN: 0216-9495

PENGEMBANGAN MESIN PENCARIAN DAN ANTI PLAGIARISME MENGGUNAKAN LATENCY SEMANTIC ANALYSIS PADA MEDIA PUBLIKASI PENELITIAN (E-JOURNAL)

Hermawan [1] Firdaus Sholihin [1]

[1] Jurusan Teknik Informatika, Fakultas Teknik, Universitas Trunojoyo Madura

Abstrak: *Information Retrieval (IR) memegang peranan kunci untuk mencari informasi yang relevan dari beragam jenis repository media baik terstruktur maupun tidak. Salah satu metode yang luas penggunaannya pada IR dan terus berkembang adalah Latency Semantic Analysis (LSA). LSA menggunakan pendekatan secara kontekstual dengan melakukan ekstraksi susunan kata pada korpus, untuk kemudian membangun ontologi domain pengetahuan yang saling terkait tanpa memerlukan referensi. Dengan Penggunaan metode LSA maka pencarian diharapkan lebih sesuai dengan kebutuhan pengguna dan memiliki tingkat fleksibilitas yang lebih baik karena tidak melakukan pencocokan secara teks penuh. Dalam penerapannya metode ini sesuai untuk pengembangan mesin pencarian dan anti plagiasi atau Computer Aided Plagiarism Detection (CaPD) yang diterapkan kedalam media pengelolaan e-Jurnal. Dikarenakan e-Jurnal memerlukan tingkat kebutuhan yang tinggi disertai tuntutan kualitas untuk terhindar dari plagiasi.*

Kata Kunci: Computer-assisted plagiarism detection (CaPD), e-Jurnal, Information Retrieval (IR), Latency Semantic Analysis (LSA), Mesin Pencarian (Search Engine)

PENDAHULUAN

Untuk memanfaatkan dokumentasi publikasi jurnal menjadi sumber pengetahuan yang berdaya guna dapat tercapai dengan tersedianya manajemen konten pada e-Journal yang didukung dengan kemampuan pencarian informasi yang baik. Karena itu penggunaan sistem pencarian informasi *Information Retrieval* (IR) yang bisa berjalan secara efektif dan efisien sangat dibutuhkan [1].

Secara mendasar IR dapat digunakan dengan mudah melalui penggunaan pencarian teks penuh melalui *Structure Query Language* (SQL). Namun SQL hanya dapat mengenali teks yang benar-benar sama sehingga adanya perbedaan susunan teks tidak dapat dikenali. Karena itu penerapan metode pencarian ini memiliki keterbatasan pada cakupan kedalamannya sehingga dibutuhkan IR spasial [2].

Salah satu metode IR spasial yang secara luas penggunaannya adalah menggunakan *Vector Space Model* (VSM). VSM mampu melakukan pengujian kemiripan dokumen berdasarkan kesamaan statistic frekuensi, namun tidak dapat digunakan untuk pengujian berdasarkan semantic susunan kata atau *term*. Sehingga jika terdapat dokumen dengan topik yang sama namun dengan penggunaan kata yang berbeda maka metode ini tidak efektik. Karena itu dibutuhkan metode yang dapat mendukung pengujian kesamaan berdasarkan topik atau hubungan semantic [3].

Diantara metode yang terbaru dan terus berkembang untuk membangun hubungan semantik secara bebas tanpa memerlukan referensi kamus adalah LSA. Metode ini banyak dimanfaatkan oleh mesin-mesin pencarian modern saat ini seperti Google, bing, Yahoo dan lainnya.

LSA mampu membangun relasi semantik antar kata antar konten antar dokumen sehingga dapat dibangun pola kumpulan obyek yang saling berhubungan sesuai dengan kedekatan similaritas frekuensi dan susunan semantik yang terhitung secara statistic [4]. Metode ini diujicobakan untuk diterapkan pada konten berbahasa Indonesia yang sampai saat ini belum memiliki sumber ontologi semantik yang memadahi sebagai sumber referensi.

Volume 5, No. 2, Oktober 2012

ISSN: 0216-9495

Latent Semantic Analysis

Latent Semantic Analysis (LSA) adalah sebuah teori dan metode untuk menggali dan merepresentasikan konteks yang digunakan sebagai sebuah arti kata dengan memanfaatkan komputasi statistik untuk sejumlah *corpus* yang besar. *Corpus* adalah kumpulan teks yang memiliki kesamaan subjek atau tema. [7]
Tahapan LSA [8] meliputi 3 tahap utama yaitu:

1. Parsing Text dan pembobotan TF IDF
   Parsing adalah sebuah proses yang dilakukan seseorang untuk menjadikan sebuah kalimat menjadi lebih bermakna atau berarti dengan cara memecah kalimat tersebut menjadi kata-kata atau frase-fras. Parsing di dalam pembuatan aplikasi dokumen yang semula berupa kalimat-kalimat berisi kata-kata dan tanda pemisah antar kata seperti titik (.), koma (,), spasi atau tanda pemisah lainnya menjadi kata-kata saja baik itu berupa kata-kata penting maupun tidak penting. Parsing text dibagi menjadi 3 bagian, yaitu:
   a. *Tokenizing*
      Tokenizing merupakan proses mengidentifikasi unit terkecil (token) dari suatu struktur kalimat. Tujuan dilakukannya *tokenizing* ini adalah untuk mendapatkan *term-term* yang nantinya akan diindeks. Pengklasifikasian *token* dilakukan untuk teks yang dipisahkan dengan spasi atau enter dalam suatu dokumen.
      Adapun beberapa kasus yang ditangani oleh *tokenizing* yaitu: 1) *handling special character*, untuk pengambilan polanya menggunakan *regular expression*. 2) *phrase*, selain special character, *Tokenizing* juga dapat menangani beberapa pola frase seperti nama, tempat dan kata sifat. 3) *white space*, karakter ini diabaikan oleh *tokenizing* dan dianggap sebagai pemisah.
   b. *Filtering*
      Filtering merupakan proses dimana token-token yang didapat dari proses t*okenizing* akan diseleksi dari token-token yang dianggap tidak penting (*stoplist*). Stoplist merupakan kata yang sering muncul dan diabaikan pada proses *filtering*.
   c. *Stemming*
      Stemming adalah suatu proses yang bertujuan untuk mengambil kata dasar dari kata yang berimbuhan atau kata tunggal dari kata bentukan. Hal itu mengurangi jumlah *term* yang berbeda dalam koleksi.
2. Singular Vector Decomposition
   Perhitungan dekomposisi matriks menjadi tiga bagian matriks baru dan pengerucutan matriks hasil dekomposisi tersebut sehingga menjadi matriks baru [9]. Adapun rumus yang digunakan, yaitu:

   $$A_{mn} = U_{mm} \, x \, S_{mn} \, x \, V_{nn}{}^{T}$$

   Dimana :
   A : matriks yang didekomposisi
   U : matriks ortogonal U (matriks vector singular kiri)
   S : matriks diagonal S (matriks nilai singular)
   V : transpose matrik ortogonal V
   m : jumlah baris matriks
   n : jumlah kolom matriks
3. Pembobotan TF-IDF
   Term Frequency-Invers Document Frequency(TF-IDF) adalah suatu perhitungan bobot yang sering digunakan dalam text mining. TF-IDF merupakan suatu cara untuk memberikan bobot hubungan suatu term terhadap dokumen. Dengan rumus sebagai berikut:

   $$Wij = tf \, x \, idf$$
   $$Wij = tfij \, x \, (\, log \, N/n + 1)$$

   Dimana :
   W_{ij}: bobot kata ke-j dan dokumen ke-i
   Tf_{ij}: jumlah kemunculan kata ke-j dalam dokumen ke-i
   N: jumlah semua dokumen
   n: jumlah dokumen yang mengandung term ke-j

ISSN: 0216-9495

4. Perhitungan Kesamaan Dokumen
   Perhitungan dilakukan untuk menentukan nilai similarity dari setiap dokumen dengan dokumen yang lainnya. Adapun rumus yang digunakan yaitu rumus cosine

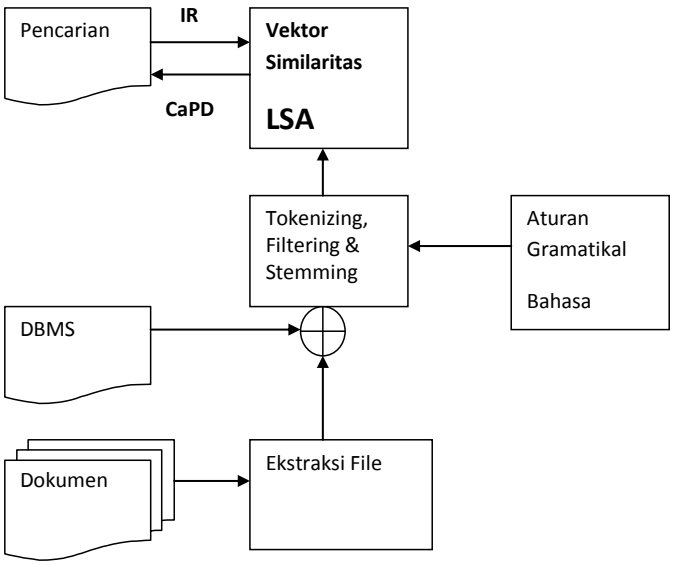
$$Similarity\ Value\ (SV) = \cos\theta = \frac{AB}{\|A\|\|B\|}$$

similarity:
Dimana :
A : vector A
B : vector B
||A|| : panjang vector A
||B|| : panjang vector B

METODE

Rancangan Sistem

Blok diagram dari sistem yang akan dikembangkan dalam penelitian ini ditunjukkan pada Gambar 2.



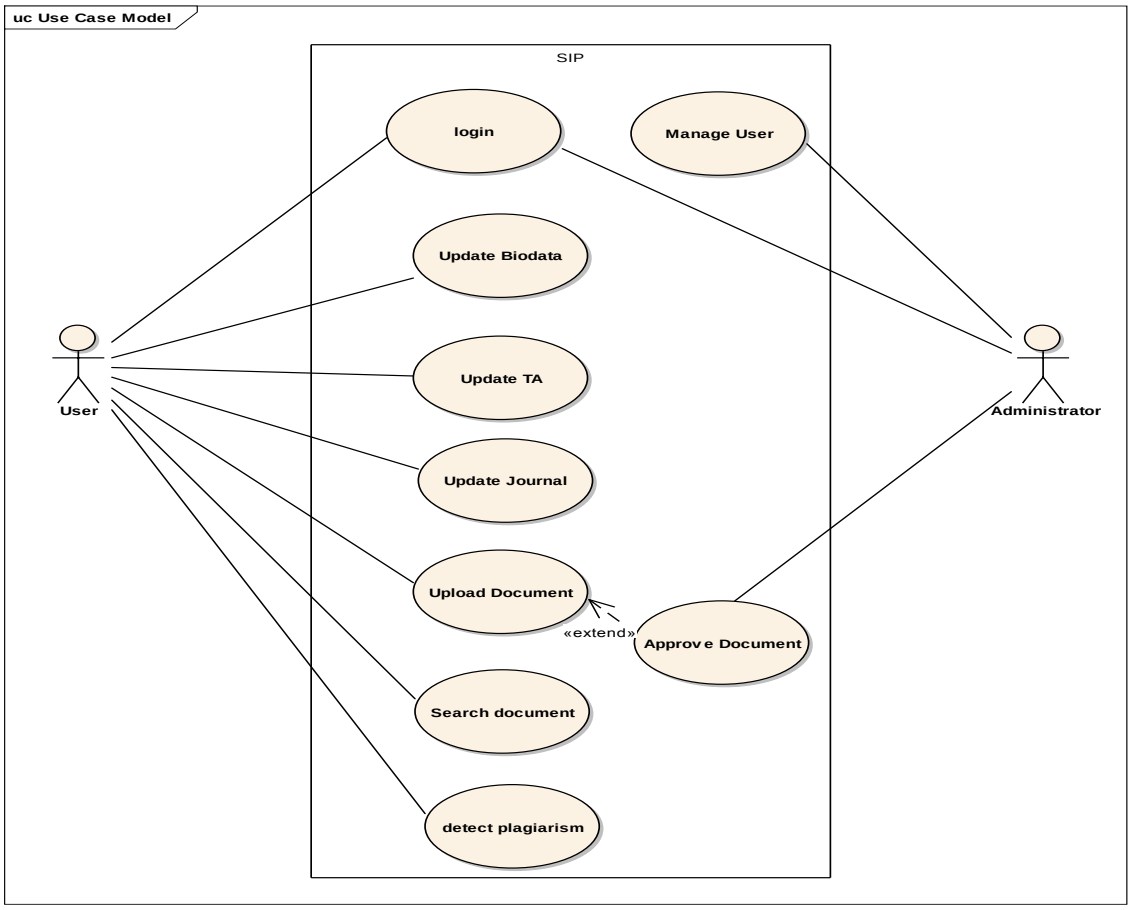
Gambar 2. Blog Diagram Sistem Pencarian Dan Anti Plagiarism E-Jurnal

Pemenuhan kebutuhan fungsional untuk e-Jurnal ditunjukkan pada diagram usecase Gambar 3.

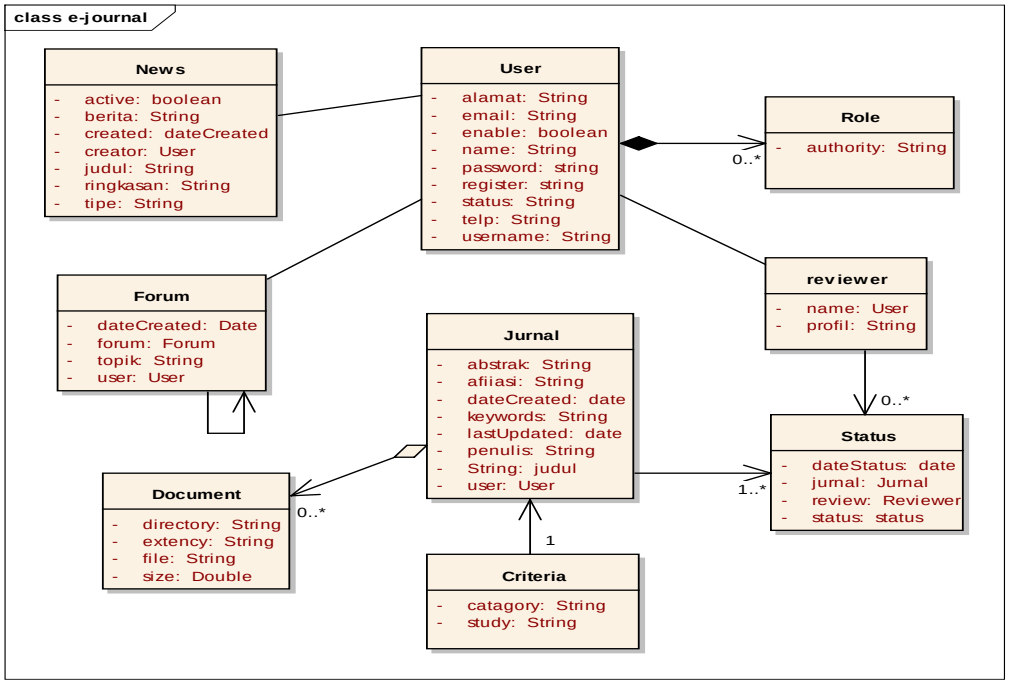
Untuk memenuhi kebutuhan data dan aplikasi dari kebutuhan fungsional dibutuhkan desain *class diagram* yang dapat menghasilkan aplikasi dasar dan struktur table penyimpanan. Desain class diagram ditunjukkan pada Gambar 4.

Volume 5, No. 2, Oktober 2012

ISSN: 0216-9495



Gambar3. Usecase Diagram Kebutuhan Fungsional e-Jurnal



Gambar 4. Class Diagram Desain Database dan Aplikasi

ISSN: 0216-9495

Implementasi VSM Konvernsional

Untuk mengimplementasikan VSM diperlukan proses ekstraksi korpus melalui tahapan tokenizing, filtering serta Stemming. Untuk kemudian dari kata yang dihasilkan disusun dalam matriks berpasangan antara kata tersebut dengan dokumen yang terkoleksi atau biasa disebut *Terms By Document Matrix* (TDM).

Tabel 2: Tabel Hasil Ekstrasi Text TDM

Term	Dokumen					DF
	D1	D2	D3	D4	D5	
Term1	1	0	1	1	1	4
Term2	1	1	0	0	1	3
Term3	0	0	1	0	0	1
Term4	0	2	1	1	0	3
Term5	1	1	3	0	0	3
Term6	0	0	0	0	1	1
Term7	0	0	1	1	2	3
Term8	0	1	0	1	1	3
Term9	1	0	0	0	0	1
Term10	0	2	0	0	2	2

Dari ruang matriks yang ada kemudian diberikan pembobotan Tf-Idf dan selanjutnya ditempatkan sesuai dengan indeks TDM. Dari Matriks yang ada dapat diujikan similaritas antar dokumen sesuai dengan bobot nilai tiap indeks yang terdapat pada vektor kata didalam TDM. Selanjutnya adalah mencari nilai TF-IDF dari masing-masing term. Dan dipresentasikan dalam sebuah matriks.

Tabel 3: TF-IDF didalam TDM

Term	Dokumen				
	D1	D2	D3	D4	D5
Term1	1.09	0	1.09	1.09	1.09
Term2	1.22	1.22	0	0	1.22
Term3	0	0	1.69	0	0
Term4	0	2.44	1.22	1.22	0
Term5	1.22	1.22	3.66	0	0
Term6	0	0	0	0	1.69
Term7	0	0	1.22	1.22	2.44
Term8	0	1.22	0	1.22	1.22
Term9	1.69	0	0	0	0
Term10	0	2.78	0	0	2.78

Implementasi LSA

Setelah matriks VSM terbentuk, pengembangan selanjutnya dapat diimplementasikan perhitungan statistik LSA dengan melakukan perhitungan SVD pada matriks VSM sehingga dihasilkan matriks *Latency*

ISSN: 0216-9495

Semantic Index (LSI). Dari LSI ini kemudian diuji cobakan untuk melakukan perhitungan similaritas dengan scenario sebagai berikut:

1. Perhitungan similaritas pada matriks penuh A dari LSI, perhitungan dilakukan pada matriks penuh yaitu $A = USV^T$ yang ditunjukkan pada Tabel

   Tabel 4: Matriks LSI

Term	Dokumen			
	D1	D2	D3	D4
Term1	0.9	0.0	1.3	0.4
Term2	1.2	1.2	-0.0	0.0
Term3	0.1	-0.00	1.6	0.4
Term4	-0.2	2.5	1.4	0.5
Term5	1.4	1.2	3.5	0.5
Term6	0.1	-0.0	-0.1	0.4
Term7	-0.0	-0.0	1.3	1.2
Term8	-0.2	1.2	0.2	0.5
Term9	1.6	0.0	0.1	-0.4
Term10	0.2	2.8	-0.2	2.6

2. Perhitungan similaritas pada matriks dekomposisi yaitu matrix ortogonal V sebagai ortogonal-kanan LSI dengan masukan eksternal.

   $A_2 = U.S.V^T$

   $A_2^T US^{-1} = VSU^T US^{-1}$

   $V = A^T US^{-1}$

   Tabel 5: Matriks Orthogonal V

0.2	-0.2	0.2	-0.9
0.6	0.2	-0.8	-0.1
0.5	-0.8	0.2	0.3
0.2	0.0	0.1	0.3

   Pada matrix V dapat diujicobakan untuk perhitungan similaritasnya dengan masukan vektor Q, dengan persamaan

   $Q = Q^T US^{-1}$

   Dimana nilai similaritas antara masukan dengankoleksi dokumen yang diwakili matriks V adalah Similarity $\text{Cos}\alpha = \frac{Q.V}{|Q||V|}$

Analisa performansi VSM Konvensional dan LSA

Untuk selanjutnya setelah dilakukan implementasi VSM Konvernsional dan LSA dapat ditarik kesimpulan bagaimana karakter performansi pada kedua metode tersebut.

ISSN: 0216-9495

Implementasi e-Jurnal dengan penerapan LSA

Setelah penerapan LSA sudah didapatkan pola yang optimal maka metode tersebut dapat diimplementasikan pada e-jurnal yang dikembangkan baik untuk sistem pencarian maupun deteksi plagiarisme.

HASIL DAN PEMBAHASAN

Untuk pengujian pada pengembangan sistem, spesifikasi sistem minimum yang digunakan ditunjukkan pada Tabel.

Tabel 6: Kebutuhan Software

Kebutuhan Software	Spesifikasi
Operating System	Windows 7
Framework Web	Grails versi 2.1.1
Database	H2 SQL
Web Server	Tomcat Grails
Library Matrix	Ejml Java Versi 0.2
IDE	NetBeans 6.9

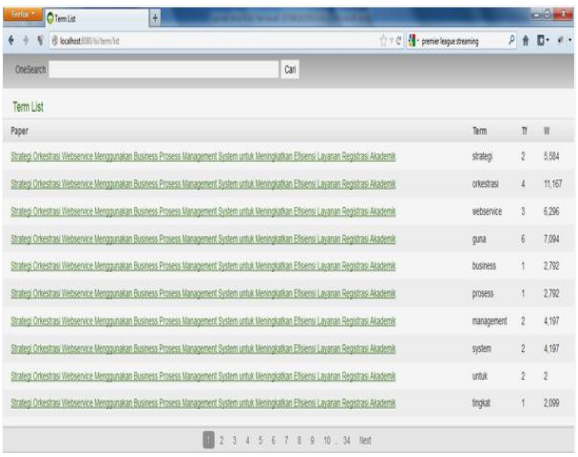
Tabel 7: Kebutuhan Hardware

Kebutuhan Hardware	Spesifikasi
Komputer PC	IBM Lenovo X100e, Memori 2Ghz, Prosesor 1.4 GHz single core

Pengujian VSM Konvensional

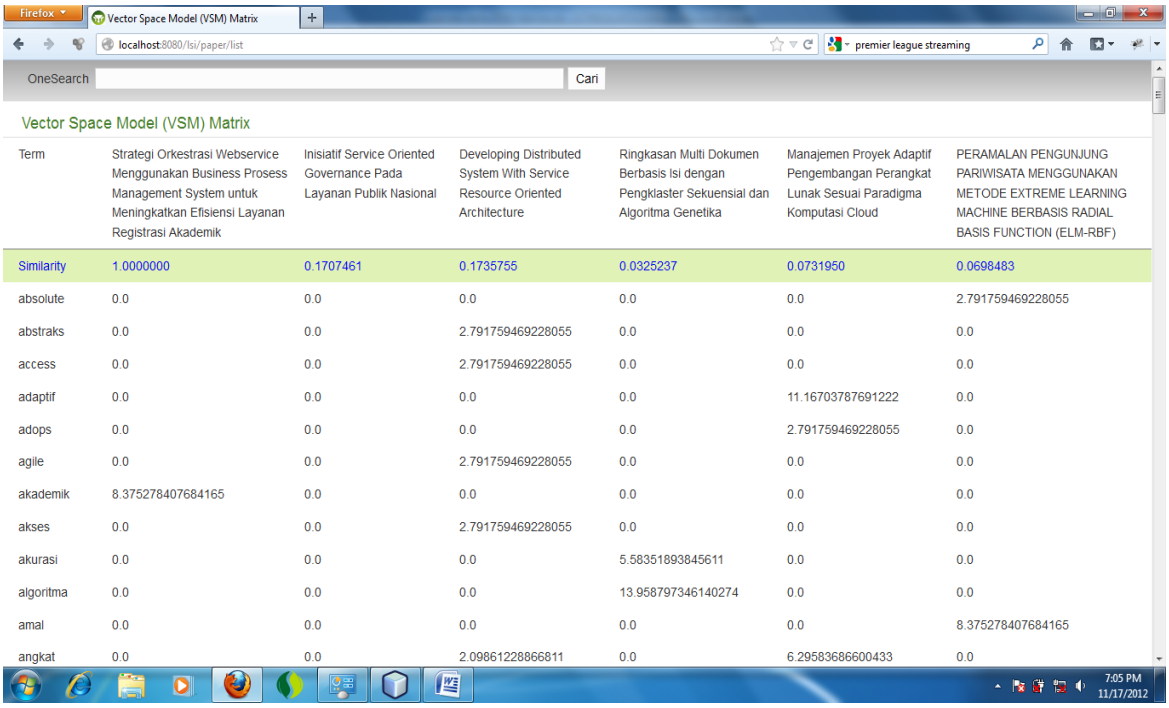
Untuk membangun VSM matrik sesuai tahapan pada metodologi diperlukan analisa kata melalui proses (tokenizing, filtering dan stemming) untuk mengekstraksi kata dari korpus yang terdapat didalam dokumen e-jurnal. Korpus yang diambil dari dokumen hanya meliputi atribut judul, abstrak dan keyword. Data yang digunakan untuk pengujian ditunjukkan pada Gambar 5 sedangkan proses ekstraksi sebagaimana ditunjukkan pada Gambar 6.

Pada Gambar 4.1 proses analisa kata pada VSM dapat dilakukan secara langsung setiap masukan data dokumen baru ataupun melalui tahap demi tahap untuk melihat hasil ekstraksi sebagaimana ditunjukkan pada Gambar 4.2. Setelah itu dapat pula dilakukan proses menghitung frekuensi kata Tf dan bobot Idf sehingga didapatkan bobot lokal dan global dari setiap kata yang terkandung dalam dokumen, sebagaimana ditunjukkan pada Gambar, sedangkan hasil perhitungan similaritasnya ditunjukkan pada Gambar.



Gambar 5. Hasil Implementasi Akstraksi dan Pembobotan VSM

ISSN: 0216-9495



Vector Space Model (VSM) Matrix

localhost:8080/lsi/paper/list

OneSearch Cari

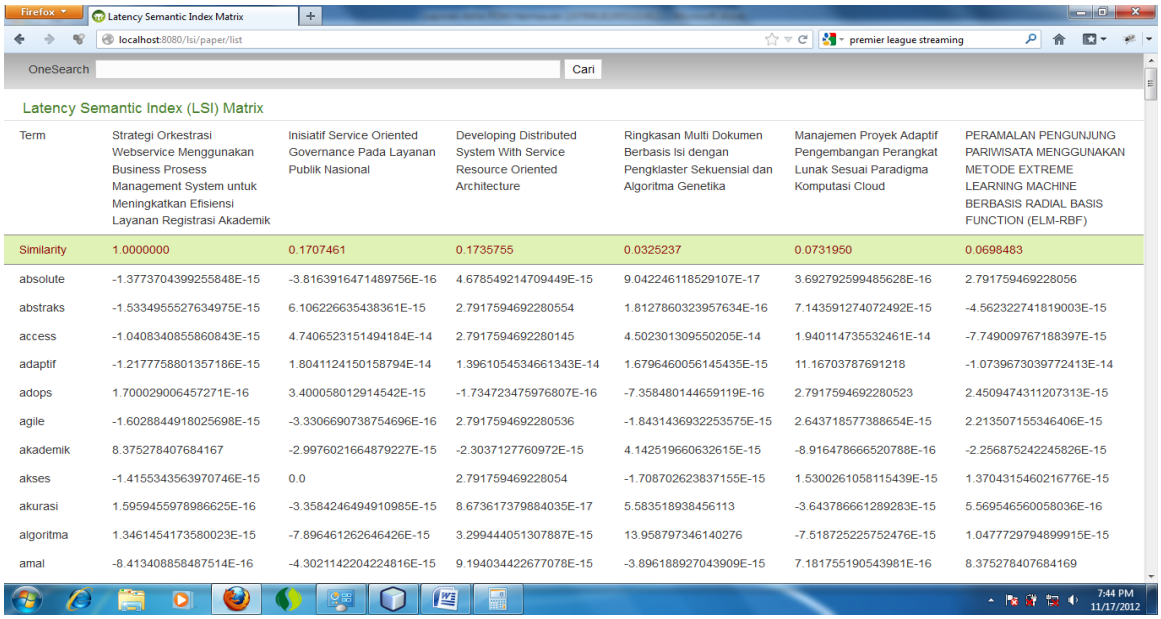
Vector Space Model (VSM) Matrix

Term	Strategi Orkestrasi Webservice Menggunakan Business Prosess Management System untuk Meningkatkan Efisiensi Layanan Registrasi Akademik	Inisiatif Service Oriented Governance Pada Layanan Publik Nasional	Developing Distributed System With Service Resource Oriented Architecture	Ringkasan Multi Dokumen Berbasis Isi dengan Pengklaster Sekuensial dan Algoritma Genetika	Manajemen Proyek Adaptif Pengembangan Perangkat Lunak Sesuai Paradigma Komputasi Cloud	PERAMALAN PENGUNJUNG PARIWISATA MENGGUNAKAN METODE EXTREME LEARNING MACHINE BERBASIS RADIAL BASIS FUNCTION (ELM-RBF)
Similarity	1.0000000	0.1707461	0.1735755	0.0325237	0.0731950	0.0698483
absolute	0.0	0.0	0.0	0.0	0.0	2.791759469228055
abstraks	0.0	0.0	2.791759469228055	0.0	0.0	0.0
access	0.0	0.0	2.791759469228055	0.0	0.0	0.0
adaptif	0.0	0.0	0.0	0.0	11.16703787691222	0.0
adops	0.0	0.0	0.0	0.0	2.791759469228055	0.0
agile	0.0	0.0	2.791759469228055	0.0	0.0	0.0
akademik	8.375278407684165	0.0	0.0	0.0	0.0	0.0
akses	0.0	0.0	2.791759469228055	0.0	0.0	0.0
akurasi	0.0	0.0	0.0	5.58351893845611	0.0	0.0
algoritma	0.0	0.0	0.0	13.958797346140274	0.0	0.0
amal	0.0	0.0	0.0	0.0	0.0	8.375278407684165
angkat	0.0	0.0	2.09861228866811	0.0	6.29583686600433	0.0

Gambar 6. Implementasi Perhitungan Similaritas pada VSM

Pengujian Latency Semantic Analysis (LSA)

Setelah melalui proses perhitungan SVD diperoleh matriks LSI yang kemudian diujicobakan perhitungan similaritasnya sebagaimana ditunjukkan hasilnya pada Gambar 7. Pada Gambar 7 menunjukkan bahwasanya nilai indeks LSI memiliki nilai indeks yang berbeda dengan VSM, tetapi pada perhitungan similaritas dihasilkan *dot product* yang sama, sehingga matriks VSM dan LSI adalah terbukti bersesuaian dan identik mewakili representasi dokumen dan kata.



Latency Semantic Index Matrix

localhost:8080/lsi/paper/list

OneSearch Cari

Latency Semantic Index (LSI) Matrix

Term	Strategi Orkestrasi Webservice Menggunakan Business Prosess Management System untuk Meningkatkan Efisiensi Layanan Registrasi Akademik	Inisiatif Service Oriented Governance Pada Layanan Publik Nasional	Developing Distributed System With Service Resource Oriented Architecture	Ringkasan Multi Dokumen Berbasis Isi dengan Pengklaster Sekuensial dan Algoritma Genetika	Manajemen Proyek Adaptif Pengembangan Perangkat Lunak Sesuai Paradigma Komputasi Cloud	PERAMALAN PENGUNJUNG PARIWISATA MENGGUNAKAN METODE EXTREME LEARNING MACHINE BERBASIS RADIAL BASIS FUNCTION (ELM-RBF)
Similarity	1.0000000	0.1707461	0.1735755	0.0325237	0.0731950	0.0698483
absolute	-1.37737043992558 48E-15	-3.8163916471489756E-16	4.678549214709449E-15	9.042246118529107E-17	3.692792599485628E-16	2.791759469228056
abstraks	-1.5334955527634975E-15	6.106226635438361E-15	2.7917594692280554	1.8127860323957634E-16	7.143591274072492E-15	-4.562322741819003E-15
access	-1.0408340855860843E-15	4.7406523151494184E-14	2.7917594692280145	4.502301309550205E-14	1.940114735532461E-14	-7.749009767188397E-15
adaptif	-1.2177758801357186E-15	1.8041124150158794E-14	1.3961054534661343E-14	1.6796460056145435E-15	11.16703787691218	-1.0739673039772413E-14
adops	1.700029006457271E-16	3.400058012914542E-15	-1.734723475976807E-16	-7.358480144659119E-16	2.7917594692280523	2.4509474311207313E-15
agile	-1.6028844918025698E-15	-3.3306690738754696E-16	2.7917594692280536	-1.8431436932253575E-15	2.643718577388654E-15	2.213507155346406E-15
akademik	8.375278407684167	-2.997602166487922 7E-15	-2.3037127760972E-15	4.142519660632615E-15	-8.916478666520788E-16	-2.256875242245826E-15
akses	-1.4155343563970746E-15	0.0	2.791759469228054	-1.708702623837155E-15	1.5300261058115439E-15	1.3704315460216776E-15
akurasi	1.5959455978986625E-16	-3.3584246494910985E-15	8.673617379884035E-17	5.583518938456113	-3.643786661289283E-15	5.569546560058036E-16
algoritma	1.3461454173580023E-15	-7.896461262646426E-15	3.299444051307887E-15	13.958797346140276	-7.518725225752476E-15	1.0477729794899915E-15
amal	-8.413408858487514E-16	-4.302114220422481 6E-15	9.194034422677078E-15	-3.896188927043909E-15	7.181755190543981E-16	8.375278407684169

Gambar 7. Hasil Perhitungan SImilaritas LSI Penuh

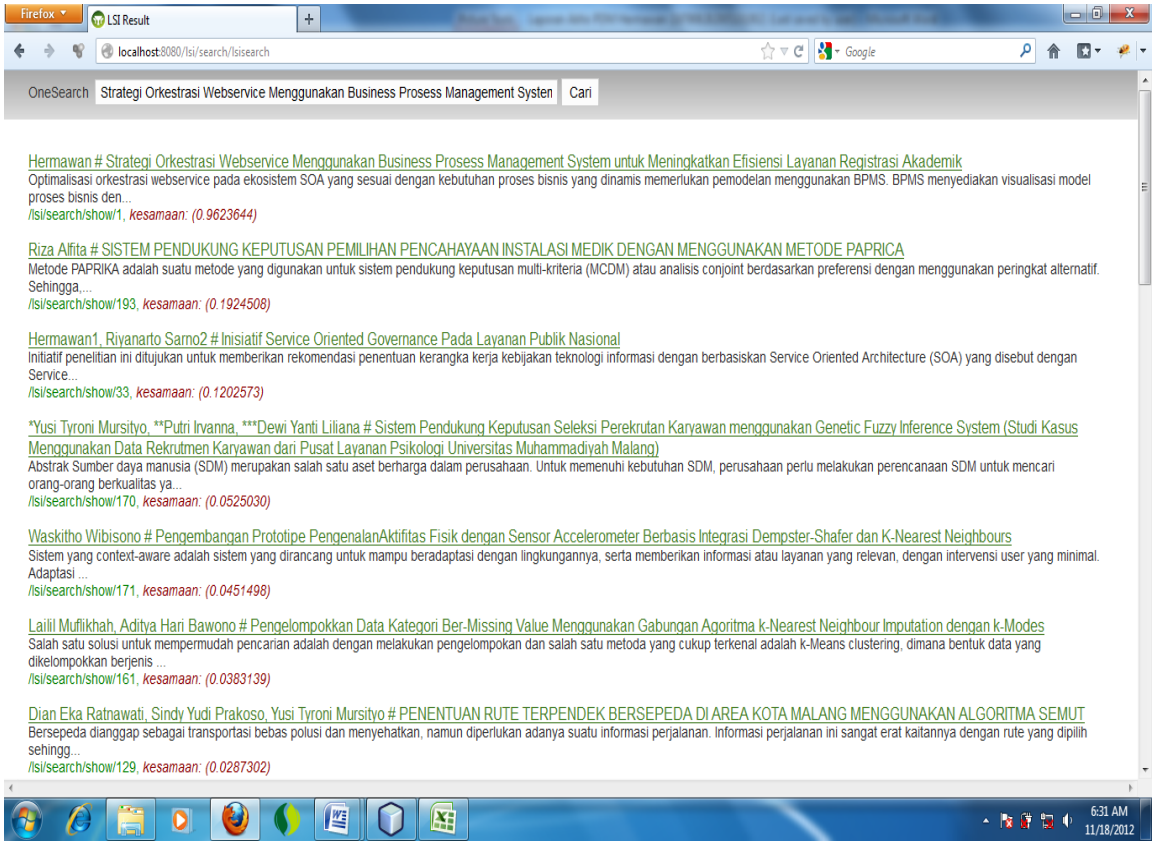
ISSN: 0216-9495

Pengujian Pencarian Menggunakan VSM dan LSI

Dari pembentukan matriks VSM dan LSI didapatkan bahwasanya untuk perbandingan nilai similaritas antar dokumen dari Matriks A_1 VSM dan A_2 LSI menghasilkan nilai *dot product* yang sama pada pengujian. Sehingga penggunaan matriks A_2 LSI tidak memberikan pengaruh apapun.

Untuk kemudian pengujian dilakukakan melalui pengujian similaritas dengan masukan data eksternal. Pada VSM sistem pencarian dari data masukan eksternal didapatkan nilai similaritas tertinggi rata-rata pada rentang 0.58 sedangkan terendah pada nilai 0.0… dengan korpus input satu kalimat seperti judul, yaitu contoh: ”*Strategi Orkestrasi Webservice Menggunakan Business Prosess Management System untuk Meningkatkan Efisiensi Layanan Registrasi Akademik”.*

Dengan pengujian masukan yang sama sebagaimana pada masukan pada VSM yaitu korpus masukan judul “*Strategi Orkestrasi Webservice Menggunakan Business Prosess Management System untuk Meningkatkan Efisiensi Layanan Registrasi Akademik*”, hasil LSI pada pengujian ini memiliki hasil keluaran pengujian similaris tertinggi jauh lebih besar daripada VSM mendekati nilai 1 dan rata-rata lebih besar dari 0.9, sedangkan nilai terendah nilai -0.0… Hasil pengujian system pencarian menggunakan LSI ditunjukkan pada Gambar 8.



Gambar 8. Sistem Pencarian Menggunakan LSI Pada Matriks Ortogonal V

Dari pengujian ini perbedaan nilai similaritas antara VSM dan LSI dapat ditunjukkan pada Tabel dan Gambar 8 dengan menunjukkan hasil similaritas 6 dokumen dengan indeks tertinggi dari 24 dokumen terkoleksi.

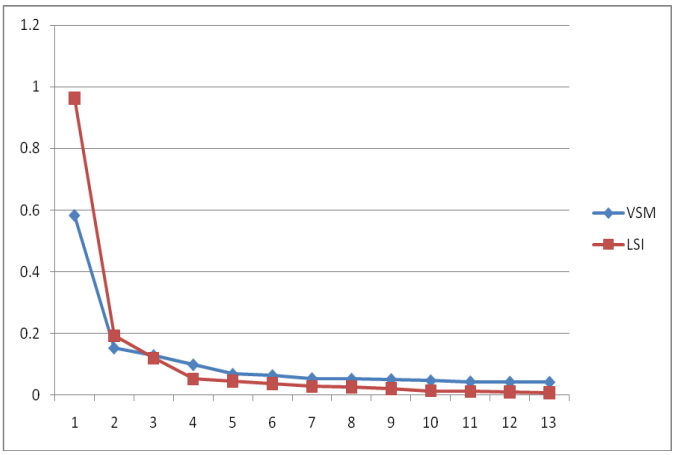
analisa data pada Tabel 4.1 dan Gambar 4.7 grafik perbandingan similaritas antara VSM dan LSI didapatkan bahwasanya LSI mampu meningkatkan nilai similaritas sehingga memiliki nilai presisi yang lebih baik dibandingkan VSM, sedangkan karena kecenderungan nilai similaritas dari VSM adalah landai maka cenderung akan memiliki nilai *recall* yang lebih besar.

ISSN: 0216-9495

Pada pengujian dengan masukan lainnya didapatkan pola yang sama pula, dimana rata-rata keluaran similaritas pencarian VSM mencapai nilai tertinggi 0.5-0.7, sedangkan pada LSI rata-rata mencapai diatas 0.9. sehingga kesimpulan analisa presisi LSI diatas VSM adalah terbukti.

Tabel 8: Indek Pencarian Menggunakan LSI Pada Matriks V

	Hasil pengujian similaritas dokumen					
VSM	0.583582	0.152771	0.128707	0.098537	0.067819	0.06382
LSI	0.962364	0.192451	0.120257	0.052503	0.04515	0.038314



Gambar 9. Grafik perbandingan Similaritas antara VSM dan LSI

Hasil Implementasi E-Jurnal Yang Menerapkan LSA

Dari perancangan sistem informasi e-Jurnal diterapkan implementasinya sebagaimana ditunjukkan hasilnya pada pengembangan konten e-jurnal untuk Sistem Informasi SENASTIK 2012, sebagaimana ditunjukkan pada Gambar.

Gambar 10. Implementasi E-Jurnal Yang Menerapkan Pencarian Spacing Retrieval LSI

Volume 5, No. 2, Oktober 2012

ISSN: 0216-9495

KESIMPULAN

1. Matrik keseluruhan pada VSM konvensional A_1 dan matriks LSI dari LSA A_2 memiliki dimensi yang tinggi dengan nilai ekstrinsik berbeda namun identik sehingga menghasilkan nilai *dot product* similaritas yang sama.
2. Matrik dekomposisi dari LSI yaitu matrik ortogonal kanan-atas atau *matriks V* dapat secara signifikan meningkatkan performansi presisi penilaian similaritas yang lebih baik dibandingkan VSM, juga dengan jumlah besar dimensi matrik yang lebih kecil sejumlah dokumen dengan indeks n^2 sehingga dapat meningkatkan kecepatan komputasi, dimana untuk selanjutnya dapat dilakukan kompresi dengan penentuan rank.
3. Dari hasil pengujian dengan jumlah 24 dokumen terkoleksi sudah didapatkan bahwasanya nilai presisi dapat ditentukan dengan mudah untuk membatasi hasil pencarian yang relevan dengan menentukan nilai ambang batas *threshold* similaritas, dimana untuk LSI dengan menentukan threshold diatas 0.7 didapatkan presisi hasil pencarian mendekati 100%.
4. Untuk pengujian plagiarisme, LSI memiliki kehandalan yang lebih baik dibandingkan VSM konvensional, karena LSI memiliki nilai presisi yang lebih tinggi dimana VSM konvensional memiliki presisi rata-rata antara 0.5 ~ 0.7 untuk nilai similaritas tertinggi, sedangkan LSI rata-rata mencapai 0.9 ~ 1. Dari analisa tersebut didapatkan dengan LSI nilai similaritas diatas 0.9 dokumen dianggap sama, sehingga untuk penilaian kesamaan dokumen penelitian dapat dikatakan terjadi plagiarisme.

SARAN

Untuk perbaikan selanjutnya pada hasil penelitian ini dapat dikembangkan penelitian berikutnya yaitu diantaranya:

1. Menentukan nilai kesamaan pada bagian sub-korpus sehingga pencarian lebih terpercaya dengan diketahui posisi similaritas antar dokumennya.
2. Untuk meningkatkan kecepatan pencarian dapat digunakan klasterisasi dokumen [10][11]
3. Dapat dikembangkan sistem untuk pencarian pada sistem terdistribusi [12]

Daftar Pustaka

R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*. Addison-Wesley Harlow, England,1999.

Beall, J. (2008). The Weakness of Full-Text Searching. *The Journal of Academic* , 438-444.

Cambridge, U. (2009). *An Introduction to Information Retrieval*. Cambridge: Cambridge UP.

Aarti Gupta, T. O. (2007). Using Ontologies and the Web to Learn Lexical Semantics. *IJCAI'07 Proceedings of the 20th international joint conference on Artifical intelligence* , 1009-1019.

KEMDIKNAS. (2010). *Peraturan dan Penanggulangan Plagiat di Perguruan Tinggi.* Jakarta: Kementrian Pendidikan Nasional RI.

Salha Alzahrani, N. S. (2011). Understanding Plagiarism Linguistic Patterns, Textual Features and Detection Methods. *IEEE* , 133 - 149 .

Landauer, T. K. (1998). *An Introduction to Latent Semantic Analysis.* Colorado USA: University of Colorado at Boulder.

Geib, J. (2011). *Latent semantic sentence clustering for multi-document summarization.* Cambridge UK: Cambridge University.

Garcia, D. E. (2006, 10 21). SVD and LSI Tutorial 4: Latent Semantic Indexing (LSI) How-to Calculations. *4* , pp. 1-12.

Chow T.W.S., R. M. (2009). Multilayer SOM With Tree-Structured Data for Efficient Document Retrieval and Plagiarism Detection. *IEEE* , 1385 - 1402.

SEBASTIANI, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys, Vol. 34, No. 1* , pp. 1–47.

Hermawan, Riyanarto Sarno. Developing Distributed System With Service Resource Oriented Architecture. TELKOMNIKA, Vol.10, No.2, Juni 2012, pp. 1~12, ISSN: 1693-6930.

Corresponding authors email address: hermawan.unijoyo@yahoo.co.id