



ISSN 1693-1629

Edisi 5

Jurnal Ilmiah

ilmu komputer

Publikasi Hasil Penelitian Yang Diterbitkan Departemen Ilmu Komputer
Institut Pertanian Bogor

Hak cipta © IPB (Institut Pertanian Bogor)

Bogor Agricultural University

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mengutip sumbernya.
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan artikel atau tulisan untuk keperluan sosial.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkannya dan memperbahayakannya sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

Analisis Algoritma Triple-DES untuk Penyandian Pesan
Sony Hartono Wijaya, Sugi Guritman dan Wisnu Ananta Kusuma

Evaluasi Penambahan Dokumen dalam
Sistem Temu Kembali Informasi
Julio Adisantoso, Yeni Herdiyeni, dan Ika Kartika

Kontrol Kongesti TCP-Friendly, Survei dan Taksonomi
Heru Sukoco

Perencanaan Berbasis Web dengan Menggunakan
Map Server dan PHP Script
*Stadi Kasus Kampus Institut Pertanian Bogor Darmaga)
Julio Adisantoso, Firman Ardiansyah dan Leny Rijaelita*

Pencarian Pola Data Audio dalam Interval Tertentu
Menggunakan Jaringan Syaraf Tiruan Rekuren
Mushthofa, Prpto Tri Supriyo dan Agus Buono

Probabilistic Neural Network Based on Multinomial Model
and EM Algorithm in Classification, Fusion and Change
Detection Context of Optical and SAR Images
Wawan Setiawan, Aniaty Murni, Benyamin Kusumoputro dan Selly Feranie

Sistem Pakar Penentuan Metode Statistika pada
Peubah Tunggal
*(Expert System for Selecting Statistical Techniques for Univariate)
Yani Nurhadryani, Marimin, Bambang Sumantri dan Hendra Yufit Riskiawan*



Vol.3 no.2 / Oktober 2005



Jurnal Ilmiah **ilmu komputer**

Diterbitkan oleh: Departemen Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam - Institut Pertanian Bogor

Edisi 5 / Vol. 3. No. 2 Oktober 2005

ISSN : 1693-1629. Tanggal 4 April 2003

Susunan Redaksi

Penanggung Jawab :

Ketua Departemen Ilmu Komputer FMIPA IPB
(Dr.Ir. Sri Nurdiyati, M.Sc)

Pemimpin Redaksi :

Irman Hermadi, S.Kom, MS

Dewan Redaksi :

Prof. Dr. Ir. Marimin, M.Sc
Dr. Ir. Kudang Boro Seminar, M.Sc
Dr. Ir. Sugi Guritman

Redaktur Pelaksana :

Irman Hermadi, S.Kom, MS
Drs. WD. Prabowo
Bambang Soetedjo (*Produksi*)

Sekretariat Jurnal Ilmiah **ilmu komputer** :

Departemen Ilmu Komputer FMIPA IPB
Jln. Raya Pajajaran, Kampus Baranangsiang Bogor 16144
Telp/Fax : 0251-356653 , E-mail : jurnal@ikom.fmipa.ipb.ac.id
Rekening : Tabungan Taplus BNI Pajajaran Bogor.
No: 3031184 a.n.: Annisa/Jurnal Iikom

Jurnal Ilmiah **ilmu komputer** diterbitkan dua kali setahun, memuat tulisan ilmiah yang berhubungan dengan **bidang Ilmu Komputer**. Jurnal ini merupakan media publikasi ilmiah dan menerima tulisan dari luar IPB, berupa hasil penelitian atau bahasan tentang metodologi.

Pihak perorangan / alumni yang telah memperoleh Jurnal Ilmu Komputer mohon mengganti biaya cetak Rp.50.000,-/expl, ditransfer melalui Tabungan Taplus BNI Pajajaran Bogor. No.Rek : 3031184 a.n.: Annisa / Jurnal Iikom.

Hak Cipta Dilindungi Undang-Undang

© Hak cipta milik IPB (Institut Pertanian Bogor)

Bogor Agricultural University

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.



Sekapur Sirih

Pembaca yang budiman,

Alhamdulillah, Jurnal Ilmiah **Ilmu komputer** Volume 3 No.2 atau Edisi ke-5 telah terbit dan sampai di hadapan pembaca.

Selamat Dies Natalis IPB ke 42, pada tanggal 1 September 2005.

Selamat datang Kami ucapkan kepada Ibu Dr. Ir. Sri Nurdiati, M.Sc yang telah terpilih menjadi Ketua Departemen Ilmu Komputer FMIPA IPB periode 2005-2009. Kami haturkan pula terima kasih kepada Bapak Ir. Agus Buono, M.Si, M.Kom atas pengabdian beliau selama ini dalam mengelola Departemen Ilmu Komputer FMIPA IPB.

Pada terbitan kali ini, Kami menyajikan tujuh hasil penelitian dengan topik yang cukup beragam, antara lain tentang: analisis algoritma kriptografi, temu kembali informasi, *network-related*, aplikasi *neural network* dalam pencarian pola data audio dan pengenalan citra, sistem pakar, dan *web-based mapping*. Semoga materi yang disajikan dapat menambah khasanah pembaca tentang dunia Ilmu Komputer.

Untuk kedua kalinya Redaksi menerima tulisan dari luar lingkungan IPB, yaitu kolaborasi antara penulis dari Jurusan Ilmu Komputer Universitas Pendidikan Indonesia dan Jurusan Ilmu Komputer Fakultas Ilmu Komputer Universitas Indonesia. Terima kasih atas kepercayaan yang diberikan kepada Kami untuk mempublikasikan karyanya. Selanjutnya, kami mengundang pembaca untuk mengirimkan tulisan hasil penelitian untuk diterbitkan dalam jurnal ini. Pedoman penulisan dapat dibaca di halaman sampul belakang

Kritik dan saran Kami harapkan untuk dapat terus melakukan perbaikan. Silakan tulis dan sampaikan lewat email ke alamat: jurnal@ikom.fmipa.ipb.ac.id. Atas partisipasi Anda, kami haturkan terima kasih.

Kampus IPB Baranangsiang, Bogor
Awal Oktober 2005

Salam,

Redaksi

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.



Daftar Isi

Sekapur Sirih	i
Daftar Isi	iii
Analisis Algoritma <i>Triple-DES</i> untuk Penyandian Pesan <i>Sony Hartono Wijaya, Sugi Guritman dan Wisnu Ananta Kusuma</i>	1
Evaluasi Penambahan Dokumen dalam Sistem Temu Kembali Informasi <i>Julio Adisantoso, Yeni Herdiyeni dan Ika Kartika</i>	11
Kontrol Kongesti <i>TCP-Friendly</i>, Survei dan Taksonomi <i>Heri Sukoco</i>	22
Pemetaan Berbasis <i>Web</i> dengan Menggunakan <i>Map Server</i> dan <i>PHP Script</i> <i>(Studi Kasus Kampus Institut Pertanian Bogor Darmaga)</i> <i>Julio Adisantoso, Firman Ardiansyah dan Leny Riajelita</i>	30
Pencarian Pola Data Audio dalam Interval Tertentu Menggunakan Jaringan Syaraf Tiruan Rekuren <i>Mushthofa, Prapto Tri Supriyo dan Agus Buono</i>	40
Probabilistic Neural Network Based on Multinomial Model and EM Algorithm in Classification, Fusion and Change Detection Context of Optical and SAR Images <i>Wawan Setiawan, Aniati Murni, Benyamin Kusumoputro dan Selly Feranie</i>	51
Sistem Pakar Penentuan Metode Statistika pada Peubah Tunggal <i>(Expert System for Selecting Statistical Techniques for Univariate)</i> <i>Yani Nurhadryani, Marimin, Bambang Sumantri dan Hendra Yufit Riskiawan</i>	64

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

Evaluasi Penambahan Dokumen dalam Sistem Temu Kembali Informasi

Julio Adisantoso¹, Yeni Herdiyeni¹ dan Ika Kartika²

¹ Staf Pengajar Departemen Ilmu Komputer, FMIPA IPB

² Mahasiswa Departemen Ilmu Komputer, FMIPA IPB

Abstrak

Saat ini pengguna cenderung menyukai pencarian berdasarkan makna. Hal ini disebabkan oleh adanya masalah sinonim dan polisemi dalam pemilihan penggunaan kata. Salah satu teknik yang mencoba mengatasi masalah tersebut Latent Semantic Indexing (LSI). Dalam pengaplikasiannya, LSI dapat menggunakan Singular Value Decomposition untuk mengestimasi struktur penggunaan kata dalam dokumen. Cara yang paling tepat untuk menambahkan dokumen atau istilah adalah melalui penghitungan ulang SVD (recomputing SVD). Namun hal ini menjadi kendala karena dibutuhkan memory yang cukup besar dan waktu yang semakin lama untuk menghitung ulang matriks istilah-dokumen yang semakin besar. Cara lain yang dapat digunakan untuk mengatasi kendala tersebut adalah dengan menggunakan teknik folding-in. Tujuan penelitian ini adalah untuk melihat pengaruh penambahan dokumen terhadap hubungan tersembunyi antara semua istilah yang secara kontekstual berdekatan artinya dengan menggunakan teknik folding-in.

Tercakup didalamnya proses pembentukan matriks istilah-dokumen dengan menggunakan parsing, penghilangan stop list, serta stemming. Pembobotan istilah untuk dokumen menggunakan skema pembobotan lxn, sedangkan pembobotan istilah untuk kueri menggunakan skema pembobotan cfx. Pengujian menggunakan 150 dokumen untuk membangkitkan matriks istilah-dokumen asal dan 110 dokumen untuk evaluasi penambahan dokumen. Pengukuran kinerja temu kembali dilakukan dengan menggunakan average precision untuk mengetahui rank optimal terhadap sepuluh kueri. Ternyata pada penelitian ini dengan pemilihan rank yang kecil akan memberikan hasil dengan tingkat akurasi yang cukup tinggi dengan tertanganinya masalah sinonim. Penambahan dokumen dengan folding-in memberikan hasil yang memuaskan. Melalui penambahan 110 dokumen ke dalam koleksi sebagian besar nilai recall bisa dipertahankan apabila menggunakan rank yang kecil, meskipun terdapat distorsi seiring dengan penambahan dokumen.

Kata kunci: Temu Kembali Informasi, Folding-in, Latent Semantic, Indexing, Singular Value Decomposition.

PENDAHULUAN

Latar Belakang

Saat ini banyak sekali informasi yang dikembangkan secara digital, namun apabila tidak diiringi dengan kemampuan untuk bisa diorganisasi, dimanipulasi ataupun dilakukan pencarian informasi yang diinginkan secara akurat dan cepat, maka informasi tersebut bisa menjadi tidak bermanfaat (Witter & Berry 1998). Oleh karena itu dibutuhkanlah suatu sistem yang mampu menangani permasalahan tersebut yaitu sistem temu kembali informasi.

Sistem temu kembali informasi bertujuan untuk memberikan data yang relevan terhadap permintaan atau kueri yang diberikan oleh pengguna secara otomatis dan akurat. Salah satu pendekatan yang biasa

digunakan adalah metode pencocokan secara leksikal yaitu melalui pencocokan antara istilah (*term*) dalam koleksi dokumen dengan istilah pada kueri yang dimasukan oleh pengguna. Namun, metode pencocokan secara leksikal tersebut bisa memberikan hasil yang tidak tepat. Hal ini disebabkan oleh dua hal yaitu :

- adanya ketidakcocokan antara istilah yang digunakan oleh pengguna dengan istilah yang ada di dalam koleksi karena banyaknya kata yang bisa digunakan untuk menyatakan suatu konsep (*sinonim*),
- adanya makna ganda yang terdapat pada suatu kata tertentu (*polisemi*) sehingga dapat menghasilkan dokumen yang tidak relevan dengan istilah dalam kueri yang dimasukkan oleh pengguna.

Salah satu teknik dalam temu kembali informasi yang mencoba menjembatani masalah-masalah tersebut adalah menggunakan pendekatan *Latent*

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 b. Pengutipan tidak merugikan kepentingan yang wajar IPB.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

Semantic Indexing (LSI). LSI mengasumsikan bahwa terdapat hubungan tersembunyi dalam penggunaan kata yang sebagian disamakan dengan pemilihan kata yang beragam (Deerwester et al. 1990). Dalam pengaplikasiannya, LSI dapat menggunakan *Singular Value Decomposition* sebagai salah satu cara untuk mengestimasi struktur penggunaan kata dalam dokumen. Pencarian kemudian dilakukan pada nilai singular yang dihasilkan oleh *SVD* yang tersimpan dalam basis data (Berry et al. 1994).

Apabila terdapat dokumen dan atau istilah yang ingin ditambahkan ke dalam basis data, maka cara yang paling tepat adalah melalui penghitungan ulang *SVD* (*recomputing SVD*), disertai penambahan dokumen atau istilah yang baru (Berry & Fierro 1995). Kendala yang dihadapi dalam proses penghitungan ulang *SVD* tersebut adalah dibutuhkan *memory* yang cukup besar dan waktu yang semakin lama untuk menghitung ulang matriks istilah-dokumen yang semakin besar.

Cara lain yang dapat digunakan untuk mengatasi kendala tersebut adalah dengan menggunakan teknik *folding-in*. Dengan teknik ini maka waktu yang dibutuhkan untuk menambahkan dokumen dan istilah baru tidak terlalu lama. Begitu juga dengan penggunaan memori yang tidak terlalu besar (O'Brien 1994).

Tujuan

Penelitian ini bertujuan untuk melihat pengaruh penambahan dokumen terhadap hubungan tersembunyi antara semua istilah yang secara kontekstual berdekatan artinya, dengan menggunakan teknik *folding-in*.

Ruang Lingkup

Penelitian ini terbatas pada evaluasi jumlah *rank* dan nilai *cut-off*. Proses yang terlibat dalam pembentukan matriks istilah-dokumen secara spesifik, penggunaan memori dan penghitungan kecepatan tidak termasuk dalam penelitian ini.

TINJAUAN PUSTAKA

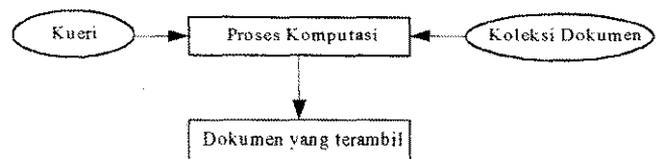
Sistem Temu Kembali Informasi

Sistem temu kembali informasi adalah suatu bentuk sistem yang melakukan proses penemuan kembali

informasi yang diperlukan. Secara konseptual sistem temu kembali informasi dibagi menjadi tiga komponen utama (Gambar 1) yaitu :

- kueri, merepresentasikan permintaan informasi (*information need*),
- proses komputasi, melakukan proses pengujian antara dokumen yang sesuai dengan permintaan informasi,
- koleksi dokumen (*corpus*), yaitu kumpulan dokumen yang berbasis vektor, yang menjadi objek pencarian dari sistem temu kembali informasi.

Sistem temu kembali informasi memerlukan masukan dari pengguna berupa kueri. Melalui kueri inilah diharapkan dapat menghasilkan keluaran yang relevan dengan permintaan pengguna. Salah satu pendekatan dalam merepresentasikan kueri tersebut adalah dengan mengolahnya menjadi bentuk vektor. Dengan menggunakan bentuk vektor diharapkan dapat menentukan kemiripan atau kesesuaian antara kueri yang ada dengan dokumen yang terdapat dalam koleksi.



Gambar 1. Konseptual Sistem Temu Kembali Informasi (Salton 1989).

Metode Ruang Vektor

Menurut Salton (1989), metode ruang vektor adalah suatu metode untuk menerapkan sistem temu kembali informasi. Misalkan sudah tersedia sekumpulan istilah yang dapat mendeskripsikan sejumlah dokumen, maka representasi dokumen dan kueri dalam model ruang vektor ini dinyatakan dalam bentuk:

1. vektor istilah dokumen

$$d_i = (a_{i1}, a_{i2}, \dots, a_{in})$$

2. vektor istilah kueri

$$q_j = (q_{j1}, q_{j2}, \dots, q_{jn})$$

Dengan a_{ik} dan q_{jk} merepresentasikan nilai dari istilah ke- k pada kueri q_j dan dokumen d_i . Biasanya a_{ik} (atau q_{jk}) bernilai 1 jika istilah ke- k muncul dalam dokumen d_i (atau kueri q_j) dan bernilai 0 jika sebaliknya. Nilai a_{ik} (atau q_{jk}) bisa juga lebih besar dari 1 untuk menyatakan seberapa penting istilah tersebut dalam dokumen atau kueri.

Keuntungan penggunaan metode ruang vektor dalam sistem temu kembali informasi :

- Dapat menentukan peringkat dari dokumen yang ditemukembalikan berdasarkan nilai kemiripan yang diperoleh dokumen tersebut.
- Jumlah dokumen yang ditemukembalikan dapat disesuaikan dengan kebutuhan pengguna.

Matriks istilah-dokumen

Dalam model ruang vektor (*vector space model*), matriks istilah-dokumen digunakan untuk merepresentasikan sekumpulan dokumen (*corpus*). Matriks ini menyatakan hubungan antar istilah dengan dokumen dalam sistem temu kembali informasi.

Jika a_{ij} menyatakan elemen matriks istilah-dokumen A pada baris ke- i dan kolom ke- j , maka salah satu representasi yang paling sering digunakan adalah frekuensi kemunculan istilah ke- i dalam dokumen ke- j sebagai nilai a_{ij} . Setiap baris pada matriks A menyatakan vektor istilah dan setiap kolomnya menyatakan vektor dokumen. Jelas sekali bahwa A merupakan matriks yang berukuran besar dan jarang (*banyak terdapat angka 0*).

$$A = \begin{matrix} & \text{dokumen} \\ \text{istilah} & \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \end{matrix}$$

Pembobotan

Fungsi utama dari pembobotan adalah untuk meningkatkan efektifitas penemukembalian informasi. Dokumen yang relevan dengan kebutuhan pengguna harus terambil dan dokumen yang tidak relevan tidak akan terambil.

Pembobotan ini dapat dilakukan dengan berbagai macam cara. Jika a_{ij} menyatakan elemen matriks istilah-dokumen A pada baris ke- i dan kolom ke- j , maka salah satu cara yang dapat digunakan untuk pembobotan adalah dengan menghitung frekuensi kemunculan istilah ke- i (f_{ij}) dalam dokumen ke- j sebagai nilai a_{ij} ($a_{ij} = f_{ij}$).

Cara lainnya adalah seperti yang dipaparkan oleh *Salton & Buckley (1998)*. Setiap kombinasi pembobotan dideskripsikan dengan menggunakan dua buah triplet. Triplet pertama digunakan untuk memboboti istilah dalam dokumen (*document term*) dan triplet kedua digunakan untuk memboboti istilah dalam kueri (*query term*). Masing-masing triplet ini terdiri dari tiga buah komponen yaitu komponen lokal, global dan normal dengan hubungan sebagai berikut :

$$a_{ij} = t_{ij} g_i d_j$$

dengan t_{ij} : komponen istilah lokal (*hanya berdasarkan informasi di dalam dokumen ke- j*), g_i : komponen istilah global (*berdasarkan informasi mengenai istilah ke- i di seluruh dokumen*), dan d_j : komponen normalisasi yang menyatakan apakah kolom-kolom (*dalam hal ini dokumen-dokumen*) dinormalisasi.

Setiap komponen tersebut memiliki berbagai formula. Formula untuk setiap komponen dapat dilihat dalam **Tabel 1,2** dan **3**. Semua formula tersebut menggunakan basis 2 untuk perhitungan logaritma. χ bernilai 1 jika $t > 0$ dan bernilai 0 jika $t = 0$.

Tabel 1. Komponen istilah lokal

Simbol	Formula untuk t_{ij}	Deskripsi
b	$\chi(f_{ij})$	Binary
t	f_{ij}	Term Frequency
c	$0.5 \left(\chi(f_{ij}) + \frac{f_{ij}}{\max_k f_{kj}} \right)$	Augmented Normalized Term Frequency
l	$\log(f_{ij} + 1)$	Log

Tabel 2. Komponen istilah global

Simbol	Formula untuk g_i	Deskripsi
x	1	No Change
f	$\log \left(\frac{n}{\sum_j \chi(f_{ij})} \right)$	Term Frequency
p	$\log \left(\frac{n - \sum_j \chi(f_{ij})}{\sum_j \chi(f_{ij})} \right)$	Augmented Normalized Term Frequency

Tabel 3. Komponen normal istilah

Simbol	Formula untuk d_i	Deskripsi
x	1	No Change
n	$(\sum_{i=1}^m (g_i t_{ij})^2)^{1/2}$	Normal

Kombinasi dari ketiga huruf tersebut nantinya akan digunakan untuk memberi bobot pada istilah. Pembobotan juga dilakukan terhadap istilah dalam kueri yang dimasukkan oleh pengguna yang direpresentasikan dalam bentuk vektor q , yaitu :

$$q = [q_i]$$

dengan q_i merepresentasikan bobot dari istilah i dalam kueri. Pembobotan istilah untuk kueri tidak sama dengan pembobotan untuk dokumen-dokumen. Pada pemrosesan kueri, berlaku :

$$q_i = g_i \hat{i}_j$$

g_i dihitung berdasarkan frekuensi dari istilah-istilah dalam koleksi dokumen sedangkan \hat{i}_j dihitung menggunakan rumus yang sama dengan rumus yang digunakan untuk t_{ij} yang diberikan pada Tabel 1 dengan f_{ij} diganti menjadi f_i yaitu frekuensi istilah ke- i dalam kueri.

Latent Semantic

Latent Semantic adalah suatu hubungan makna tersembunyi antara dua string yang berbeda, meliputi hubungan sinonim dan polisemi yang maknanya menyertakan dua string tersebut, kesamaan konsep, dan konsep yang berhubungan. Sistem temu kembali informasi yang mampu mengatasi latent semantic akan mengembalikan dokumen-dokumen yang beberapa istilahnya memiliki hubungan tersembunyi dengan string yang diberikan pada kueri, tanpa harus memberikan string yang sama dengan string yang terdapat dalam dokumen tersebut sehingga dapat menambah efektifitas sistem temu kembali informasi sebesar 30% dibandingkan penggunaan metode biasa (Deerwester et al. 1990).

Apabila nilai recall untuk kueri yang dipilih cukup tinggi, maka suatu dokumen akan memiliki kemungkinan yang semakin besar untuk ikut terambil dalam pemrosesan kueri, termasuk dokumen yang memiliki kasus latent semantic. Latent semantic

memiliki kemungkinan lebih besar teratasi jika ada dua istilah yang memiliki hubungan makna, sering muncul dalam dokumen secara bersamaan.

Singular Value Decomposition (SVD)

SVD adalah salah satu teknik eksplorasi data yang dapat mereduksi dimensi matriks tanpa menyebabkan kehilangan informasi yang berarti. Jika A adalah matriks berukuran $m \times n$ dengan rank r dan $0 < k < r$, dekomposisi nilai singular dapat digunakan untuk mendapatkan sebuah matriks dalam $R^{m \times n}$ dengan rank k yang paling dekat dengan matriks A , relatif terhadap norma Frobenius. Norma Frobenius merupakan norma yang diturunkan dari hasil kali dalam untuk ruang vektor berdimensi $R^{m \times n}$.

Misalkan A adalah matriks istilah-dokumen, maka SVD dari A adalah $A = U \Sigma V^T$, dengan U adalah matriks ortogonal $m \times m$, V adalah matriks ortogonal $n \times n$ dan Σ adalah matriks diagonal yang mengandung nilai singular pada diagonal utamanya dalam urutan menurun

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_{\min(m,n)})$$

Awalnya dekomposisi menghasilkan r nilai singular. Dari r nilai singular ditentukan k nilai singular signifikan yang akan menentukan rank dari matriks A (Kolda & O'Leary 1998). Perkalian matriks-matriks hasil dekomposisi oleh SVD dapat digunakan untuk membangun sebuah aproksimasi rank- k untuk A dengan hanya menggunakan $k \leq r$, yaitu :

$$A \approx A_k = U_k \Sigma_k V_k^T,$$

dengan U_k dan V_k terdiri dari k kolom pertama dari U dan V berturut-turut, Σ_k adalah submatriks berukuran $k \times k$ yang elemen diagonalnya adalah k elemen diagonal pertama dari elemen diagonal Σ . A_k adalah aproksimasi terbaik untuk rank k dari A .

Penerapan SVD pada Sistem Temu Kembali Informasi

Pada sistem temu kembali informasi, digunakan aproksimasi rank- k dari matriks istilah-dokumen (A). Karena matriks hasil aproksimasi SVD cukup dekat ke matriks asalnya, pengembalian dokumen menggunakan matriks tersebut dapat diharapkan sama baiknya seperti saat menggunakan matriks asalnya.

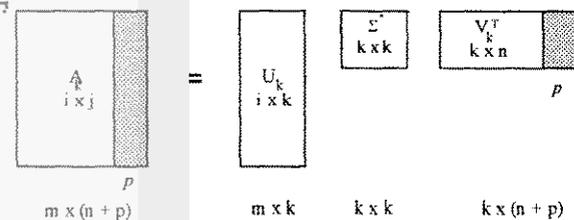
Namun pada kenyataannya, bukan hanya lebih baik SVD juga dapat mengembalikan dokumen lebih banyak (Kolda & O'Leary 1998). Hal ini disebabkan karena struktur tersembunyi (latent) bisa diperkirakan dengan membuang kata-kata yang tidak terlalu penting.

Penambahan Informasi dalam Basis Data

Misalkan basis data yang dibangkitkan oleh LSI telah tersedia yaitu, koleksi dokumen yang telah dilakukan proses *parsing*, matriks istilah-dokumen yang telah dibangkitkan dan SVD dari matriks istilah-dokumen yang telah dihitung. Jika terdapat istilah dan dokumen baru yang ingin ditambahkan maka terdapat dua alternatif yang bisa dilakukan yaitu menghitung ulang SVD (*recomputing SVD*) dengan menggunakan matriks istilah-dokumen yang baru atau melakukannya langsung tanpa menghitung ulang SVD. *Updating* adalah proses penambahan istilah atau dokumen ke dalam basis data yang telah dibangkitkan oleh LSI. *Recomputing SVD* bukan merupakan metode *updating* karena melakukan penghitungan ulang LSI dengan menggunakan istilah atau dokumen baru yang telah ditambahkan (Berry & Fierro 1995).

Updating dapat berupa *folding-in* dan *SVD-updating*. *SVD-updating* merupakan metode penambahan dokumen dengan menggunakan struktur *latent* semantik yang telah ada dan terdiri dari tiga tahap: *updating terms*, *updating documents* dan *updating term weights* (O'Brian, 1994).

Seperti halnya *SVD-updating*, *folding-in* juga menggunakan struktur *latent* semantik yang telah ada sehingga tidak menyebabkan penghitungan kembali SVD pada basis data yang telah diperbaharui. Penambahan dokumen dengan metode *folding-in* dilakukan dengan cara menambahkan vektor dokumen baru yang telah terboboti ke dalam himpunan vektor-vektor dokumen yang telah ada atau kolom dari V_k (Gambar 2).



Gambar 2. Representasi penambahan p dokumen

Seperti vektor kueri, \underline{d} didefinisikan sebagai vektor dokumen baru yang telah terboboti dengan ukuran $i \times 1$. Proyeksi \underline{d}_p terhadap \underline{d} ke dalam model LSI yang sudah ada, dapat dilakukan sebagai berikut (Berry & Fierro 1995) :

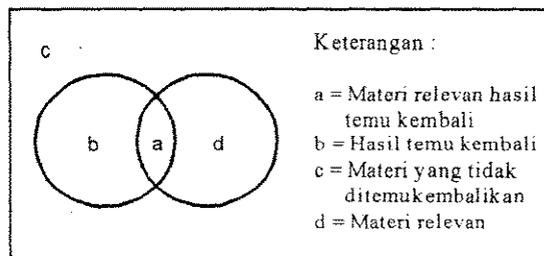
$$\underline{d}_p = \underline{d}^T U_k \Sigma_k^{-1}$$

Recall dan Precision

Suatu sistem temu kembali informasi memiliki kinerja yang diukur melalui efisiensi dan efektifitas. Efisiensi diukur melalui waktu pemrosesan kueri sampai selesai ditemukembalikan, sedangkan efektifitas diukur melalui *recall* dan *precision*.

Recall adalah rasio jumlah dokumen relevan yang ditemukembalikan dan jumlah dokumen yang relevan. *Recall* mengukur seberapa lengkap suatu pencarian itu, apakah semua dokumen yang relevan telah ditemukembalikan. Semakin tinggi nilai *recall* maka semakin sedikit pula dokumen yang *hilang* dalam pengembalian.

Precision adalah rasio jumlah dokumen relevan hasil temu kembali terhadap seluruh jumlah dokumen hasil temu kembali. *Precision* mengukur seberapa tepat suatu sistem dalam melakukan suatu pencarian, apakah semua dokumen hasil temu kembali relevan atau tidak. Semakin tinggi nilai *precision* maka semakin sedikit pula sistem mengembalikan dokumen yang tidak diinginkan.



Gambar 3. Pembagian koleksi oleh sistem temu kembali informasi.

Berdasarkan Gambar 3, *recall* (R) dan *precision* (P) dapat dinyatakan sebagai berikut :

$$R = \frac{a}{a + d}$$

$$P = \frac{a}{a + b}$$

Secara teori, *recall* dan *precision* tidak saling berhubungan. Namun pada prakteknya *recall* yang tinggi dapat dicapai dengan mengorbankan *precision*. Begitu juga sebaliknya, *precision* yang tinggi dapat dicapai dengan mengorbankan *recall*. Oleh sebab itu,

meningkatnya nilai *recall* seringkali berarti penurunan nilai *precision*. Hal ini terjadi karena untuk meningkatkan kemungkinan terambilnya seluruh dokumen yang relevan, maka pengguna harus memeriksa banyak materi, diantaranya materi yang sebenarnya tidak relevan.

Average Precision

Average precision adalah suatu ukuran evaluasi dalam sistem temu kembali informasi yang diperoleh dengan menghitung rata-rata tingkat *precision* pada berbagai tingkat *recall* (Grossman, 2002). *Average Precision* dihitung dengan menggunakan rumus,

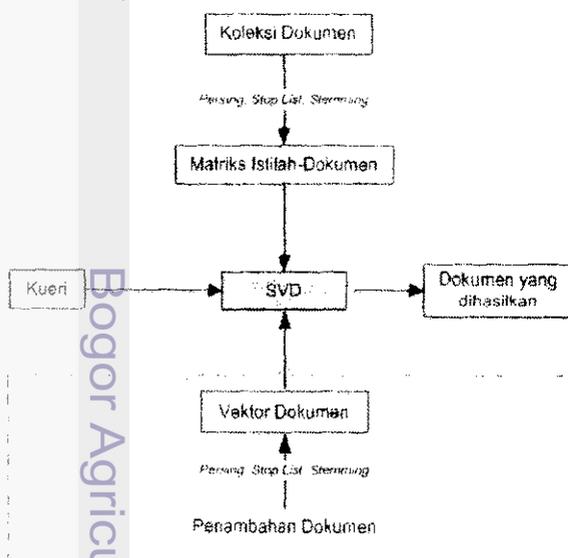
$$\bar{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q}$$

dengan $\bar{P}(r)$ adalah *average precision* pada tingkat *recall* r , N_q adalah jumlah kueri yang digunakan serta $P_i(r)$ adalah *precision* pada tingkat *recall* r (Baeza-Yates & Ribeiro-Neto 1999).

METODOLOGI

Gambaran Umum Sistem

Secara garis besar langkah-langkah pengerjaan sistem tertera pada **Gambar 4** berikut.



Gambar 4. Gambaran Umum Sistem.

Koleksi Dokumen

Koleksi dokumen yang digunakan terdiri dari abstrak skripsi dari fakultas *MIPA* dan fakultas *Teknologi Pertanian IPB* serta abstrak skripsi dan tesis dari *Fakultas Ilmu Komputer, Universitas Indonesia*. Koleksi pengujian ini terdiri dari berbagai bidang antara lain : sistem pakar, sistem informasi, sistem temu-kembali informasi, sistem kecerdasan buatan, sistem pengenalan wajah, pola analisis citra, analisis algoritma, jaringan komputer, pengolahan basis data, serta pembuatan desain alat. Jumlah dokumen yang digunakan adalah 260 buah, dengan perincian 150 abstrak digunakan untuk membangkitkan matriks istilah-dokumen dan 110 abstrak lainnya digunakan dalam proses penambahan dokumen.

Matriks Istilah-Dokumen

Pembentukan matriks istilah-dokumen dilakukan dengan berbagai cara. Salah satunya adalah melalui langkah-langkah sebagai berikut:

1. Melakukan proses *parsing* yaitu pemilahan dokumen menjadi unit-unit yang lebih kecil berupa kata. Proses *stemming* dilakukan untuk istilah dalam bahasa Indonesia sehingga istilah yang digunakan adalah istilah yang bersih dari imbuhan (*kata dasar*). Hal ini dilakukan untuk memperkecil jumlah istilah yang digunakan dalam matriks istilah-dokumen. Untuk istilah dalam bahasa Inggris tidak dilakukan *stemming* mengingat istilah dalam bahasa Inggris tidak terlalu banyak.
2. Melakukan penghilangan kata-kata yang termasuk dalam *stop list* yaitu istilah atau *string* yang sering muncul dalam koleksi dokumen dan tidak memiliki arti yang penting seperti kata tugas: *ini, itu, dan* serta *string* yang berupa angka seperti angka tahun. dilanjutkan dengan menghilangkan kata-kata yang tidak signifikan dalam membedakan dokumen atau kueri.
3. Melakukan pemotongan lebih lanjut melalui proses *stemming* (dengan menggunakan *kata dasar*). Setelah didapatkan matriks istilah dokumen, maka dilakukan pembobotan terhadap elemen matriks dengan menggunakan skema pembobotan *l_{kn}* (Salton & Buckley 1998) yaitu :

$$a_{ij} = \frac{\log(f_{ij} + 1)}{\sqrt{\sum_{k=1}^m (\log(f_{kj} + 1))^2}}$$

yang berarti tiap istilah memiliki komponen lokal \log , tidak ada komponen global, dan terdapat normalisasi pada kolom-kolom dari matriks. Matriks istilah-dokumen yang telah terboboti kemudian diolah dengan menggunakan *SVD* sehingga didapatkan aproksimasi matriks A_k . Dengan menggunakan nilai k yang tepat, maka hubungan tersembunyi antar semua istilah dapat ditemukan. Pemilihan nilai k dilakukan melalui serangkaian percobaan (Berry et al. 1994). Nilai k yang digunakan dalam percobaan ini adalah: 5, 10, 20, 40, 50, 60, 70, 80, 90 dan 100

Penrosesan kueri

Sebelum dilakukan pemrosesan, istilah pada kueri juga harus diboboti terlebih dahulu. Skema pembobotan yang digunakan pada kueri adalah *cfx* (Salton & Buckley 1998) yaitu :

$$q_i = \left(0.5 \chi(\hat{f}_i) + 0.5 \left(\frac{\hat{f}_i}{\max_k \hat{f}_k} \right) \right) \log \left(\frac{n}{\sum_{j=1}^n f_{ij}} \right)$$

dengan nilai $\chi(\hat{f}_i) = 1$ jika $\hat{f}_i > 0$ dan $\chi(\hat{f}_i) = 0$ jika $\hat{f}_i = 0$.

Jumlah kueri yang digunakan untuk evaluasi adalah 10 buah. Dalam sistem temu-kembali informasi terdapat berbagai macam cara untuk melakukan pemrosesan kueri diantaranya dengan mengalikan suatu vektor kueri dengan matriks A_k ,

$$\underline{s} = \underline{q}^T A_k$$

sehingga menghasilkan suatu vektor \underline{s} (*score vector*) yang menyatakan nilai (*score*) dokumen-dokumen dalam sistem terhadap kueri. Jika vektor \underline{s} diurutkan mengecil, maka didapatkan urutan relevansi dokumen-dokumen dalam sistem terhadap kueri tersebut. Untuk menentukan tingkat relevansi dokumen terhadap *query* digunakan nilai *cut-off* yaitu 0.001, 0.005 dan 0.01. Kinerja temu kembali dievaluasi dengan menggunakan *recall-precision* serta *average precision* pada tingkat *recall* 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0.

Penambahan Dokumen

Penambahan dokumen dilakukan dengan menggunakan metode *folding-in*. Sebelum ditambahkan, dokumen-dokumen tersebut juga mendapat perlakuan yang sama seperti pada pembentukan matriks istilah-dokumen awal. Perlakuan tersebut meliputi proses *parsing*, penghilangan *stop list* serta pembobotan dengan menggunakan skema

pembobotan lcn , sehingga dihasilkan vektor dokumen yang siap untuk ditambahkan.

Penambahan vektor dokumen ke dalam matriks A_k ini dilakukan secara bertahap yaitu :

1. Penambahan 20 dokumen
2. Penambahan 60 dokumen
3. Penambahan 90 dokumen
4. Penambahan 110 dokumen.

Penambahan bertahap ini dilakukan untuk melihat sampai sejauh mana pengaruh penambahan dokumen terhadap matriks A_k . Evaluasi dengan menggunakan *average precision* selanjutnya dilakukan untuk setiap tahap penambahan dokumen ke dalam matriks istilah-dokumen awal (A_k). *Average precision* ini dihitung dengan menggunakan tingkat *recall* yang sama dengan sebelum penambahan dokumen. Setelah dilakukan penambahan maka akan dihasilkan aproksimasi matriks istilah dokumen yang baru yaitu matriks A_p .

Ortogonalitas

Proses penambahan dokumen dengan menggunakan metode *folding-in* ini dapat menyebabkan terjadinya distorsi terhadap ortogonalitas V_k . Hal ini terjadi karena ditambahkannya sub-matriks non-ortogonal \underline{d}_p ke dalam V_k . Jumlah distorsi yang ditimbulkan ini dapat dihitung dengan menggunakan rumus :

$$\left\| \hat{V}_k^T \hat{V}_k - I_k \right\|_2$$

Oleh karena itu, setelah dilakukan penambahan dokumen maka langkah selanjutnya adalah menentukan jumlah nilai ortogonalitas yang hilang (Berry et al. 1994).

Lingkungan Pengembangan

Lingkungan pengembangan yang digunakan adalah sebagai berikut:

- Perangkat lunak: *Windows XP Professional, MATLAB 6.05.*
- Perangkat keras: *Athlon XP 1 GHz, 256 MB RAM.*

HASIL DAN PEMBAHASAN

Kueri

String pada kueri diusahakan dalam bentuk yang singkat. Karena sinonim, polisemi dan kesamaan konsep biasanya terjadi pada *string* yang pendek, maka pemilihan *string* yang pendek akan memudahkan terambilnya dokumen yang mengandung *string* yang memiliki hubungan makna dengan *string* yang terdapat pada kueri. Kueri yang digunakan antara lain :

1. *Algorithm Analysis*
2. *Expert System*
3. *Management Information System*
4. *Information Retrieval*
5. *Human Computer Interaction*
6. *Geographic Information System*
7. *Computer Networking*
8. *Pattern Recognition*
9. *Face recognition*
10. *Object Oriented*

Kueri-kueri ini sengaja dipilih dalam bahasa Inggris yang merupakan sinonim dari kata-kata yang ada dalam koleksi dokumen uji. Hal ini dilakukan untuk melihat apakah dokumen-dokumen yang tidak mengandung kata-kata dalam kueri namun sebenarnya relevan, bisa ditemukembali. Pengategorian koleksi dokumen dilakukan secara manual dengan membaca isi dokumen.

Matriks istilah-dokumen

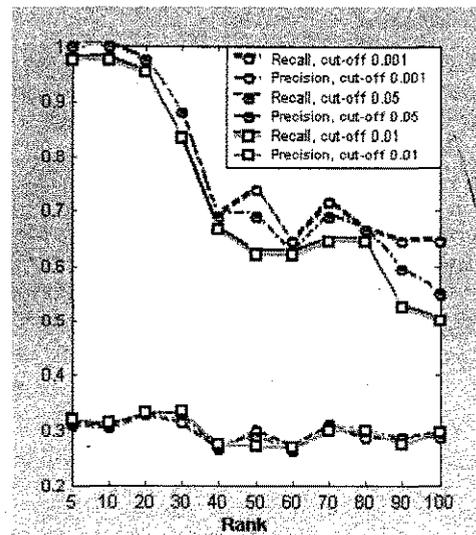
Jumlah istilah yang digunakan dalam matriks adalah 1529 sehingga didapatkan ukuran matriks sebelum dilakukan penambahan dokumen adalah 1529 x 150 dengan jumlah elemen bukan nol (*non-zero elements*) sebanyak 6670.

Pemrosesan Kueri

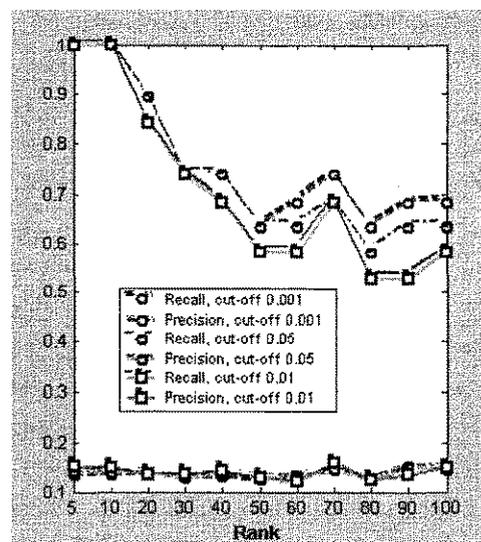
Dari tiga nilai *cut-off* (0.001, 0.005 dan 0.01) yang diuji pada kesepuluh kueri, ternyata tidak memberikan perbedaan yang besar apabila nilai *k* yang digunakan kecil, baik dari segi *recall* maupun *precision*. Namun untuk nilai *k* yang besar dapat terlihat bahwa penggunaan nilai *cut-off* 0.001 dapat meningkatkan nilai *recall-precision* (Gambar 5). Oleh karena itu nilai *cut-off* yang digunakan untuk percobaan selanjutnya adalah 0.001.

Hubungan tersembunyi antar istilah dalam matriks aproksimasi bisa muncul melalui pemilihan nilai *k* yang tepat. Misalkan pada percobaan dengan menggunakan kueri '*expert system*', diharapkan

dokumen yang mengandung kata '*sistem pakar*' akan ikut terambil bahkan dokumen yang sama sekali tidak mengandung kata '*sistem pakar*' namun masuk ke dalam kelompok sistem pakar juga diharapkan ikut terambil. Dari percobaan yang dilakukan, terdapat 18 dokumen dalam koleksi pengujian yang termasuk dalam kelompok sistem pakar. Meskipun hanya 1 dokumen yang didalamnya terdapat kata '*expert system*' namun dengan pemilihan nilai *k* yang tepat maka hubungan tersembunyi ini dapat dikenali seperti yang bisa dilihat pada Gambar 6.



Gambar 5. Evaluasi dengan kueri 'management Information system'

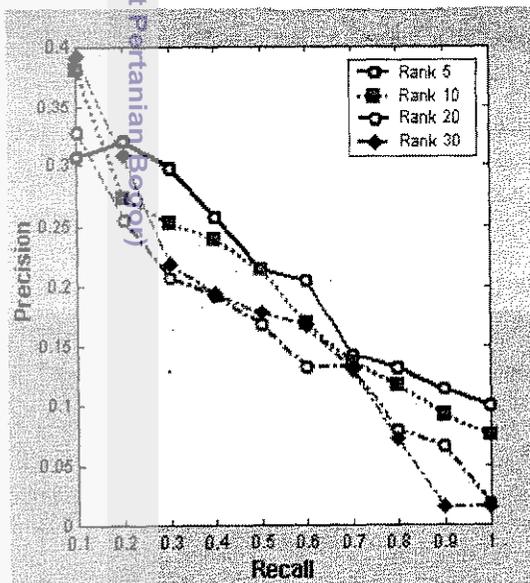


Gambar 6. Evaluasi dengan kueri 'expert system'

Bahkan dengan memilih nilai $k = 5$ atau $k = 10$ sebagai rank yang digunakan, semua dokumen yang relevan berhasil ditemukembali. Data yang digunakan dapat dilihat pada **Gambar 6**.

Dari percobaan dengan menggunakan sepuluh kueri, ternyata diperoleh tingkat precision yang rendah. Salah satu faktor yang menyebabkannya adalah karena adanya masalah polisemi. Kegagalan terjadi pada fakta bahwa setiap istilah direpresentasikan hanya sebagai satu titik dalam ruang (Deerwester et al. 1990).

Secara keseluruhan, grafik hasil penelitian dengan sepuluh kueri terdapat pada **Gambar 7**. Semakin besar nilai k yang digunakan, maka nilai *average precision* yang dihasilkan cenderung semakin mengecil. Hal ini terjadi karena terdapat banyak kata-kata yang tidak perlu dalam data yang digunakan. Oleh karena itu pada penelitian ini, penggunaan nilai k yang kecil cukup memadai untuk menemukan hubungan tersembunyi tanpa menghilangkan informasi yang penting.



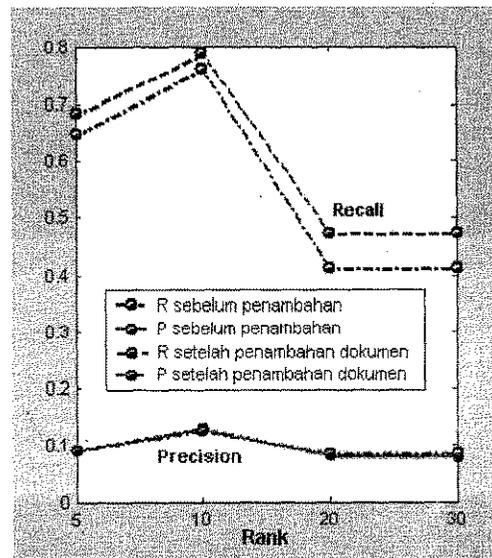
Gambar 7. Average Precision sebelum penambahan dokumen

Pengaruh penambahan dokumen terhadap average precision

Dari percobaan sebelumnya, nilai *average precision* yang tinggi sebagian besar dihasilkan pada selang k antara 5 sampai 50. Oleh karena itu pada percobaan selanjutnya prediksi nilai k yang digunakan adalah 5, 10, 20, 30, 40, 50.

Penambahan 20 dokumen ke dalam matriks aproksimasi awal cenderung meningkatkan nilai *recall-precision* (Gambar 8). Meskipun terdapat penurunan nilai *recall-precision*, namun tidak terlalu besar (Gambar 9). Ini berarti penambahan 20 dokumen tersebut tidak merusak representasi matriks aproksimasi awal.

Pada setiap proses penambahan dokumen, didapatkan nilai *recall* dan *precision* yang tidak jauh berbeda (*memiliki pola yang mirip*) antara sebelum dan sesudah dilakukan penambahan dokumen. Hal ini disebabkan karena ketepatan *folding-in* bergantung pada hubungan istilah-dokumen yang sama antara sebelum dan sesudah penambahan dokumen (O'Brien 1994).

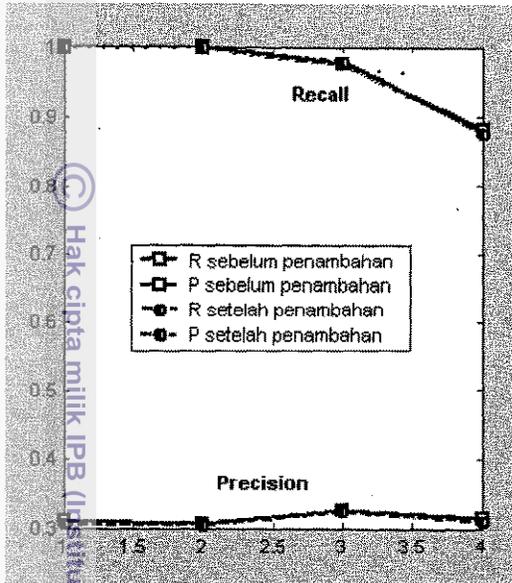


Gambar 8. Penambahan 20 dokumen dengan kueri 'analysis algorithm'

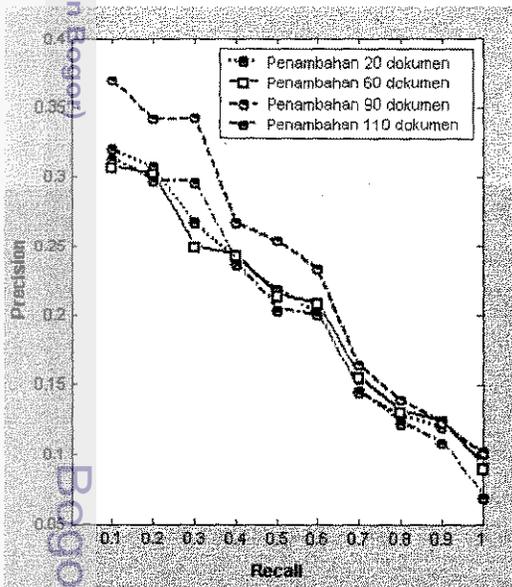
Dari **Gambar 8** dan **Gambar 9** diperoleh nilai *recall-precision* yang tinggi melalui penggunaan nilai k yang kecil. Nilai *recall-precision* yang dihasilkan juga cenderung meningkat untuk semua kueri. Meskipun terdapat beberapa kali penurunan nilai *recall-precision*, namun penurunan tersebut tidak terlalu besar. Hasil evaluasi dengan menggunakan *recall-precision* dan *average precision* untuk semua kueri pada berbagai tingkat penambahan dokumen.

Terlihat bahwa nilai *average precision* yang dihasilkan cenderung mengecil untuk nilai k yang semakin besar. Berdasarkan rata-rata geometri diperoleh nilai k optimal untuk sepuluh kueri adalah 5. Pengaruh

penambahan dokumen terhadap nilai *average precision* dengan nilai $k = 5$ dapat dilihat pada **Gambar 10**.



Gambar 9. Penambahan 20 dokumen dengan kueri 'management information system'

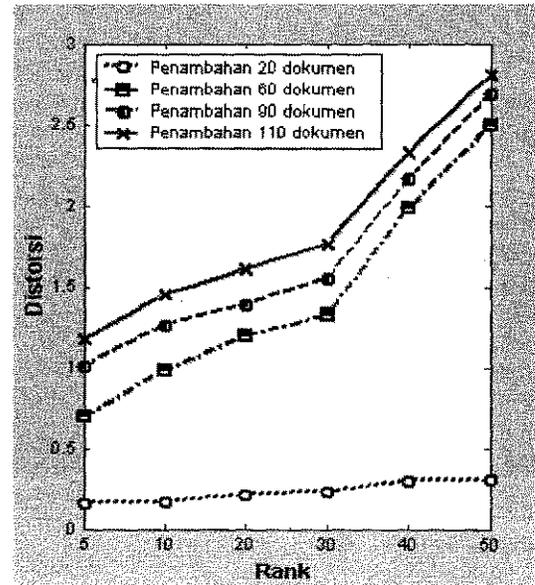


Gambar 10. Average Precision pada nilai $k = 5$

Ortogonalitas

Dari **Gambar 10**, terdapat beberapa kali penurunan nilai *average precision*. Menurunnya nilai *average precision* ini karena metode *folding-in* tidak

mempertahankan ortogonalitas dari \hat{v}_k dengan ditambahkan sembarang vektor dokumen terboboti ke dalam \hat{v}_k . Hal ini menyebabkan terjadinya distorsi yang muncul setelah penambahan dokumen baru (Berry et al. 1994). Jumlah distorsi yang timbul akibat dilakukannya penambahan dokumen bisa dilihat pada **Gambar 11**.



Gambar 11. Jumlah distorsi ditimbulkan akibat penambahan dokumen

Meskipun terdapat distorsi, pengaruh penambahan dokumen sampai dengan 110 buah dapat menghasilkan nilai *recall* yang cukup tinggi apabila menggunakan pemilihan nilai k yang tepat. Dengan demikian hubungan tersembunyi antara semua istilah yang secara konseptual berdekatan artinya tetap dapat ditemukan walaupun telah dilakukan penambahan dokumen. Oleh karena itu secara keseluruhan penambahan dokumen dengan metode *folding-in* dapat memberikan hasil yang memuaskan.

KESIMPULAN DAN SARAN

Kesimpulan

Melalui pemilihan *rank* yang tepat maka akan dihasilkan sebuah aproksimasi matriks istilah-dokumen baru yang dapat memberikan hasil dengan tingkat akurasi yang cukup tinggi. Hal ini bisa terjadi karena penerapan *LSI* berbasis *SVD* terbukti dapat meningkatkan hasil temu kembali sehingga bisa



menangani masalah sinonim dalam pemasukan kueri. Tergantungnya masalah sinonim ini bisa dilihat dari tingginya nilai *recall* yang dihasilkan.

Penambahan dokumen dengan menggunakan teknik *folding-in* bisa dijadikan sebagai alternatif yang baik. Hal ini bisa dilihat dari pengaruh penambahan 110 dokumen ke dalam koleksi yang telah ada, meskipun terdapat distorsi seiring dengan penambahan dokumen, namun melalui pemilihan *rank* yang tepat maka sebagian besar nilai *recall* bisa dipertahankan.

Saran

Penelitian ini bisa dikembangkan lebih lanjut dengan menggunakan teknik *folding-in* pada penerapan *LSI* berbasis lainnya seperti *Semi-Discrete Decomposition* atau *probability PCA*.

DAFTAR PUSTAKA

Baeza-Yates, R. & B. Ribeiro-Neto. 1999. *Modern Information Retrieval*. England: Addison-Wesley.

Berry, M.W., Susan T. Dumais, Gavin W. O'Brien. 1994. *Using Linear Algebra for Intelligent Information retrieval*. Department of Computer Science. University of Tennessee, Knoxville, T.N.

Berry, M. W. & R.D Fierro. 1995. *Low-Rank Orthogonal Decompositions for Information Retrieval Applications*. Department of Computer Science. University of Tennessee, Knoxville, T.N.

Berry, M. W. , Susan T. Dumais, Todd A. Letsche. 1995. *Computational Methods for Intelligent Information Access*. Proceedings of Supercomputing 1995.

Berry, M. W. , Z. Drmac & E. R. Jessup. 1998. *Matrices, Vector Space, and Information Retrieval*. SIAM REVIEW. Vol. 41, No. 2, pp. 335-362.

Deerwester, Scott. S. Dumais, G. Furnas & R. Harshman. 1990. *Indexing by Latent Semantic Analysis*. Journal of the American Society for Information Science, 41(6):391-40.

Grossman, D. *IR Book*. http://www.ir.iit.edu/~dagr/cs529/files/ir_book/ [7 Maret 2002]

Kolda, T. & D. O'Leary. 1998. *A semi-discrete matrix decomposition for latent semantic indexing in information retrieval*. ACM Trans. Inform. Systems, pp. 322-346.

Lancaster, F. & A. Warner. 1993. *Information Retrieval Today*. Arlington: Information Resources Press .

O'Brien, G. W. 1994. *Information Managements Tools for Updating an SVD-Encoded Indexing Scheme*. Department of Computer Science . University of Tennessee, Knoxville, T.N.

Salton, G & C. Buckley. 1988. *Term Weighting approaches in automatic text retrieval*. Inf. Process Manage. 24, pp. 512-523.

Salton, G. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. England: Addison-Wesley.

Witter, D. I & Michael W. Berry. 1998. *Downdating the Latent Semantic Indexing Model for Conceptual Information retrieval*. Department of Computer Science. University of Tennessee, Knoxville, T.N.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.