PENERAPAN METODE *LATENT SEMANTIC INDEXING* PADA *SEARCH ENGINE*

Edward Ferdian[1], Rian Hadisaputra[2], Nurkholis Madjid[3]

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if13006@students.if.itb.ac.id[1], if13026@students.if.itb.ac.id[2], if13047@students.if.itb.ac.id[3]

Abstrak

Search Engine memegang peranan penting dalam perkembangan pencarian informasi – informasi melalui jaringan internet. Perkembangan teknologi *Search engine* pun semakin pesat. Jika dulu kita hanya mengenal yahoo! sebagai *search engine* yang paling populer dan banyak penggunanya, maka sekarang kepopuleran yahoo! mulai pudar dengan munculnya lawan tangguh yaitu Google. Salah satu keunggulan Google yaitu penggunaan algoritma yang lebih kompleks dan lebih mangkus sehingga dalam setiap pencarian, Google mampu menghasilkan hasil pencarian yang lebih banyak dengan penggunaan waktu yang lebih singkat. Salah satu metode yang digunakan adalah metode *indexing*. Saat ini Google telah mengembangkan metode baru yang mampu mempermudah penyaringan informasi yang disebut *Latent Semantic Indexing*.

***Kata kunci**: search engine, Latent Semantic Indexing*

1. Pendahuluan

Pada awal perkembangannya, sebelum ditemukannya internet, orang-orang bertukar informasi dengan menggunakan *File Transfer Protocol* (FTP). Di dalam suatu grup yang kecil, untuk mendapatkan kembali informasi yang dibutuhkan di FTP tentunya sangat mudah. Namun, untuk suatu grup yang besar hal ini menjadi masalah.

Untuk menyelesaikan masalah tersebut seorang bernama Alan Emtage, mahasiswa Universitas McGill di Montreal membuat suatu software yang dapat mencari informasi yang dibutuhkan sesuai dengan permintaan pengguna. Software ini memiliki database yang menyimpan nama – nama alamat web, bukan lagi arsip – arsip seperti di FTP.[1]

Namun, seiring bertambahnya waktu internet muncul dan berkembang dengan pesat. Jumlah alamat internet (URL) pun jumlahnya mencapai jutaan. Tentunya, tidak akan efektif lagi jika pengguna harus mencari informasi yang dibutuhkan berdasarkan nama arsip. Untuk itulah dibutuhkan *search engine* yang tidak hanya mencari informasi dari internet melalui nama arsipnya tetapi juga dapat mencari berdasarkan kata kunci.

Dalam makalah ini akan djelaskan metode *Latent Semantic Indexing* yang merupakan salah satu metode yang digunakan oleh *search engine* paling populer saat ini, Google.

2. *Search engine*

2.1 Pengertian *Search engine*

Search engine adalah sebuah sistem yang diperuntukkan untuk pencarian dan pengambilan informasi untuk menampilkan hasilnya. Biasanya sistem ini berbasis indeks beberapa dokumen HTML, sehingga pencarian dapat dengan mudah dilakukan.[2]

2.2 Algoritma *Search engine*

Algoritma *search engine* merupakan suatu instruksi yang menyusun sekumpulan halaman web berdasarkan kata kunci. Instruksi tersebut menganalisis komponen-komponen dari sebuah halaman web, seperti judul, struktur *link* ke halaman lain, dan sebagainya.

Ketika pengguna mencari dokumen maka *search engine* akan mengakses data yang telah dikumpulkan sebelumnya. Pencarian tersebut dilakukan berdasarkan kata kunci yang dimasukkan oleh pengguna.

Pada umumnya algoritma *search engine* mencari kata dalam dokumen dan menghitung banyaknya kemunculan kata tersebut. Kemudian dokumen yang memiliki lebih banyak jumlah kata kunci tersebut berada di urutan paling atas. Tetapi cara ini kurang efektif sebab banyaknya kemunculan kata kunci tidak selalu menentukan isi dokumen. Dan bahkan

seringkali tidak berhubungan sama sekali dengan apa yang dicari oleh pengguna.

3. *Latent Semantic Indexing*

Latent Semantic Indexing adalah sebuah metode baru dalam algoritma *search engine* yang sedang dikembangkan Google Corporation. Dengan metode ini, Google menganalisis kata kunci dengan cara baru, bukan lagi berdasarkan pencocokkan kata secara leksikal. Kata yang dicari tidak hanya kata kuncinya saja seperti pada algoritma pada umumnya, tetapi kata-kata yang berhubungan dengan kata kunci juga dicari.

Tujuan dari LSI adalah mendapatkan suatu pemodelan yang efektif untuk merepresentasikan hubungan antara kata kunci dan dokumen yang dicari. Dari sekumpulan kata kunci, yang tadinya tidak lengkap dan tidak sesuai, menjadi sekunpulan objek yang berhubungan.

3.1 Metode LSI

Membandingkan dua *term*:
Hasil kali titik dari dua baris vector X mencerminkan bahwa dua *term* memiliki pola yang sama dalam sebuah dokumen. [3]

Membandingkan dua dokumen :
Hasil kali titik dari dua kolom vektor X. [3]

Membandingkan sebuah *term* dengan dokumen.

Contoh :

Query: human-computer interaction
Dataset:

c1 Human machine interface for Lab ABC computer application
c2 A survey of user opinion of computer system response time
c3 The EPS user interface management system
c4 System and human system engineering testing of EPS
c5 Relations of user-perceived response time to error measurement
m1 The generation of random, binary, unordered trees
m2 The intersection graph of paths in trees
m3 Graph minors IV: Widths of trees and well-quasi-ordering
m4 Graph minors: A survey

{X}=

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

r (human.user) = -.38
r (human.minors) = -.29

gambar 1 : matriks konteks X di atas, dibentuk dari judul lima artikel tentang *human-computer interaction* dan empat artikel tentang *graph theory*. Isi dari sel menandakan jumlah kemunculan sebuah kata(baris) dalam judul (kolom) untuk kata yang muncul, paling sedikit di dua buah judul. [4]

{X}={W}{S}{P}'

{W}=

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

S { }=

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

P { }=

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

Gambar 2 : SVD yang lengkap dari matriks konteks X [4]

$\{\hat{X}\}$ =

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

r (human.user) = .94
r (human.minors) = -.83

Gambar 3: rekonstruksi dua dimensi dari matriks X yang terdapat pada gambar 1, berdasarkan pada kolom dan baris yang diarsir dari SVD yang ditunjukkan pada gambar 2. Membandingkan baris dan sel yang diarsir dan dikotakkan dari gambar 1 dan gambar 3 menggambarkan bagaimana LSA mengenali kesamaan hubungan dengan mengubah nilai perkiraan data. [4]

Langkah pertama adalah merepresentasikan teks sebagai matriks yang setiap barisnya mewakili kata yang unik dan setiap kolom mewakili kalimat. Setiap sel berisikan frekuensi kemunculan kata di setiap kolom. Selanjutnya, isi dari sel merupakan transformasi preliminary yag detilnya akan dideskripsikan kemudian, yang masing – masing frekuensi sel diberi bobot oleh sebuah fungsi yang menghasilkan keutamaan kata dalam sebuah kalimat.

Selanjutnya, metode ini mengaplikasikan SVD (*Singular Value Decomposition*) ke dalam matriks. Dalam SVD, matriks persegi didekomposisi menjadi tiga matriks lainnya. Matriks pertama mendeskripsikan baris asli sebagai vector turunan nilai factor orthogonal. Satu matriks lagi mendeskripsikan kolom seperti sebelumnya. Matriks ketiga adalah matriks diagonal yang memuat nilai skala jika ketiga komponen matriks dikalikan.

Dari matriks-matriks yang telah dibuat, akan ditemukan kata yang berhubungan dengan kata kunci yang dicari. Kemudian, dari kata-kata tersebut, dicarilah halaman-halaman web yang sesuai dari daftar alamat web yang terdapat pada basis data.

4. Kesimpulan

Metode *Latent Semantic Indexing* (LSI), sangat bermanfaat untuk digunakan pada *search engine*, karena dengan metode ini *search engine* dapat mencari dokumen yang diinginkan oleh pengguna dengan lebih akurat. Hal ini terbukti pada *search engine* Google yang mampu menghasilkan pencarian yang akurat dengan pemanfaatan waktu yang lebih sedikit.

Daftar Pustaka

[1] http://www.search-marketing.info/search-engine-history/ diakses tanggal 19 Mei 2005 pukul 12.00 WIB.
[2]http://bovis.gyuvet.ch/3dict/323webde.html diakses tanggal 19 Mei 2005 pukul 12.00 WIB.
[3]http://www.cs.arizona.edu/classes/cs630/spring03/slides/jan-29.ppt diakses tanggal 19 Mei 2005 pukul 12.00 WIB.
[4]http://lsa.colorado.edu/papers/dp1.LSAintro.pdf diakses tanggal 19 Mei 2005 pukul 12.00 WIB