

# IMPLEMENTASI METODE LATENT SEMANTIC INDEXING PADA FITUR PENCARIAN WEB PARIWISATA

111110388 Yogi Maulana

Program Studi: Teknik Informatika, Sekolah Tinggi Informatika dan Komputer Indonesia  
Malang, Jawa Timur, Indonesia  
E-mail : yogi.ogel@gmail.com

## Abstrak

Pencarian atau *search* pada *website* pada umumnya menggunakan query kesamaan kata kunci atau *keyword* dengan isi dokumen. Hal ini sering menimbulkan hasil pencarian yang kurang efisien dan tidak relevan. Oleh sebab itu perlu adanya suatu metode pencarian yang dapat membantu pengguna untuk menemukan pencarian yang lebih relevan. Metode yang sangat cocok digunakan dalam sebuah pencarian adalah metode *latent semantic indexing* (LSI), LSI adalah pengembangan dari sistem temu kembali atau *information retrieval* (IR). Cara kerja dari metode LSI adalah dengan mencari keterkaitan antara kata kunci dengan isi setiap dokumen yang ada. Sehingga akan mendapatkan hasil pencarian yang lebih relevan. Dari permasalahan mengenai pencarian tersebut. Maka dilakukan penerapan metode *latent semantic indexing* pada pencarian *website*, untuk mendapatkan hasil pencarian yang lebih relevan. Algoritma yang akan digunakan yaitu *Singular Value Decomposition* (SVD). Pada tugas akhir yang akan dibahas adalah tentang penerapan metode *latent semantic indexing* pencarian pada *website* pariwisata. Untuk menghasilkan pencarian yang lebih relevan.

**Keywords:** *Web Pariwisata* , *Search*, *Metode Latent Semantic Indexing*.

## 1. Pendahuluan

Teknologi *internet* menjadi salah satu pilihan untuk mencari informasi yang terdapat pada *website*. Dengan media *website* yang berisikan informasi, diharapkan pengguna mendapatkan informasi yang relevan terkait dengan objek yang dicari. Pada umumnya model pencarian *website* menggunakan pencarian konvensional, yaitu kesamaan *query* dan *keyword*. Cara kerja pencarian konvensional dengan melakukan kecocokan antara sebuah *keyword* dengan isi dari *database* suatu *website*. Dengan model ini hasil pencarian yang tidak relevan sering ditemukan oleh pengguna, sehingga perlu suatu metode pencarian yang menghasilkan dokumen yang relevan, untuk memaksimalkan hasil pencarian. Contoh kasus yang diangkat dalam

penelitian ini adalah pencarian yang terdapat pada *website* pariwisata.

*Information retrieval* (IR) adalah sistem yang otomatis melakukan pencarian atau temu kembali informasi yang relevan. Pencarian oleh pengguna dideskripsikan dalam *query*, menjadi *input* bagi IR, dan kemudian sistem mencari dan menampilkan dokumen yang relevan dengan *query* tersebut. Salah satu metode yang digunakan untuk temu kembali informasi dengan melihat kecocokan *semantic query* dengan koleksi dokumen adalah metode *Latent Semantic Indexing* (LSI). Metode LSI dapat digunakan untuk mengidentifikasi pola dalam hubungan antara istilah sebuah kata yang memiliki sinonim (persamaan kata), serta keterkaitan sebuah kata dengan kata yang lainnya, sehingga menghasilkan pencarian yang lebih efektif dan relevan.

*Latent Semantic Indexing* (LSI) menggunakan algoritma SVD (*Singular-Value Decomposition*) untuk memodelkan relasi asosiatif antara term. Ide dasar SVD adalah menerima kumpulan data dengan dimensi dan variabel tinggi, serta mengurungnya ke dalam ruang dimensi yang berukuran lebih kecil. Tujuannya untuk mendapatkan lebih jelas sub struktur dari data asli dan mengurutkannya mulai dari yang bervariasi sampai dengan tidak bervariasi.

Secara matematis algoritma SVD memecah sebuah dokumen. Array dari dokumen berisi kumpulan kata kemudian dirubah menjadi angka yang mewakili setiap dokumen. Kemudian akan menghasilkan array yang setiap barisnya mewakili sebuah dokumen dan tiap kolomnya mewakili jumlah kata pada dokumen. Kemudian array akan dirubah menjadi matrix. Tujuan dirubah menjadi matrix agar mudah dalam perhitungan SVD, diantaranya transpose, inverse, perkalian, akar. Begitu juga dengan *query* yang berisi *keyword*, *query* diperlakukan sama dengan sebuah dokumen. Kemudian dilakukan ranking dengan sudut cosinus. Hasilnya berupa angka relevansi antara *query* dan dokumen. ini yang menjadi acuan sebuah dokumen tersebut relevan dengan *keyword* pencarian.

Metode LSI sangat cocok digunakan dalam pencarian dokumen yang lebih relevan, contoh kasus pada pencarian *website* pariwisata. Hasil pencarian akan ditampilkan

berdasarkan urutan dokumen yang paling relevan dengan *keyword* pencarian. Dokumen berisi sekumpulan kata yang berisi informasi yang berkaitan dengan objek yang dicari.

### 1.1 Rumusan Masalah

Bagaimana membangun *search* pada website pariwisata, guna mempermudah pencarian dengan hasil yang efektif dan *keyword* atau kata kunci yang saling berhubungan, dengan metode *Latent Semantic Indexing* untuk mendapatkan hasil pencarian yang relevan.

### 1.2 Tujuan

Membuat *search* dengan menerapkan metode *Latent Semantic Indexing* sehingga menghasilkan pencarian yang lebih relevan.

### 1.3 Batasan Masalah

Ruang lingkup permasalahannya hanya akan dibatasi pada:

1. Pencarian pada website pariwisata, menggunakan metode *Latent Semantic Indexing*.
2. Data yang digunakan berupa text dokumen pada website
3. Hasil pencarian berupa urutan dokumen.
4. Algoritma yang digunakan adalah SVD (*Singular-Value Decomposition*).

## 2. Information Retrieval System

### 2.1 Pengertian Information Retrieval System

*Information Retrieval* (IR) *system* adalah sistem yang digunakan dengan tujuan untuk menemukan kembali (*retrieve*) informasi yang relevan dari sekumpulan informasi. Salah satu implementasi dari IR *system* adalah *search engine* atau pencarian yang terdapat pada website. Dimana pengguna dapat mencari dokumen yang dibutuhkan melalui *search engine*.

IR *system* bekerja dalam pencarian informasi yang tidak memiliki struktur, hal ini sama dengan bentuk *query* atau ekspresi pencarian pengguna. Dokumen adalah salah satu contoh informasi yang tidak terstruktur.

### 2.2 Metode Latent Semantic Indexing

Metode *Latent Semantic Indexing* (LSI) adalah pengembangan dari *Information Retrieval* (IR), LSI adalah sebuah algoritma yang mirip dengan proses pemikiran yang sebenarnya. LSI dibuat untuk mendukung information retrieval dan memecahkan masalah ketidaksesuaian antara kamus pemakai dengan penulis dokumen. Asumsi yang mendasari LSI adalah terdapat sebuah struktur pokok atau “*latent*” yang mempresentasikan hubungan antar kata. LSI menerima sebuah vektor atau matrik dari sekumpulan

dokumen, dimana setiap baris mewakili satu term (bisa kata atau frase), tiap kolom mewakili satu dokumen, dan tiap selnya akan berisi nilai bobot kata terhadap dokumen.

### 2.3 Konsep Latent Semantic Indexing

Konsep Latent Semantic Indexing (LSI) merupakan metode IR yang membangun struktur koleksi dokumen dalam bentuk ruang vektor dengan menggunakan teknik aljabar linier, yaitu singular value decomposition.

Secara umum, konsep LSI meliputi beberapa point seperti diilustrasikan pada gambar 2.3 yaitu:

- 1) *Text Operations* pada *Query* dan *Document Collection*. Query dari pengguna dan koleksi dokumen dikenakan proses text operations. Proses text operations meliputi,
  - a. mem-parsing setiap kata dari koleksi dokumen,
  - b. membuang kata-kata yang merupakan stop words,
  - c. mem-stemming kata-kata yang ada untuk proses selanjutnya.
- 2) *Matrix Creation*. Hasil *text operations* yang dikenakan pada koleksi dokumen dikenakan proses matrix creation. Proses *matrix creation* meliputi,
  - a. menghitung frekuensi kemunculan dari kata,
  - b. membangun matriks kata-dokumen seperti diilustrasikan pada gambar 2.3 Baris matriks menunjukkan kata dan kolom matriks menunjukkan dokumen. Sebagai contoh, elemen matriks pada baris ke-1 dan kolom ke-2 menunjukkan frekuensi kemunculan kata ke-1 pada dokumen ke-2.
- 3) *SVD Decomposition*.
  - a. Matriks kata-dokumen yang terbentuk,  $A$  berukuran  $m \times n$ , selanjutnya dikenakan dekomposisi SVD (singular value decomposition). Hasil SVD berupa 3 (tiga) buah matriks, dapat ditulis menjadi  $A=USV$ , sehingga dapat ditulis.

$$A = \begin{matrix} & U & & S & & V^T \\ \begin{matrix} A = [u_1 & u_2 & \dots & u_k] \end{matrix} & \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_k \end{bmatrix} & \begin{matrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_k^T \end{bmatrix} \end{matrix} \end{matrix}$$

- a. Rank dari matriks  $A$ ,  $k$  adalah banyaknya entry tak nol yang terletak pada diagonal utama matriks  $S$ ,  $k$  juga merupakan banyaknya nilai singular dari  $A$ .
- b. Diperoleh hasil perkalian baru yaitu.

$U_r, S_r, V_r^T$  dengan  $U_r = [u_1 \ u_2 \ \dots \ u_r]$ ,

$$S_r = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r \end{bmatrix}, \text{ dan } V_r^T = \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_r^T \end{bmatrix}.$$

4) *Query Vector Creation.*

Vektor query,  $q$  dibentuk seperti membangun sebuah kolom dari matriks kata-dokumen. Contoh vektor query, adalah

$$q = \begin{matrix} T_1 \\ T_2 \\ \vdots \\ T_m \end{matrix} \begin{matrix} \text{Query} \\ \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_m \end{bmatrix} \end{matrix},$$

5) *Query Vector Mapping.*

Vektor query, dipetakan ke dalam ruang vektor berdimensi menjadi

$$Q = q^T U_r S_r^{-1}$$

6) *Ranking.*

Proses ranking menggunakan vektor dokumen dan vektor query. Ranking dari dokumen relevan ditentukan oleh besar sudut yang dibentuk oleh vektor query dan vektor dokumen. Semakin kecil sudut yang dibentuk, semakin relevan query dengan dokumen, perhitungan ini menggunakan *cosinus sim* untuk mencari sudut dari kedua vektor.

Vektor dokumen :

$$V_r = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_n \end{bmatrix}, \text{ dengan } D_j^T = \begin{bmatrix} d_{j1} \\ \vdots \\ d_{jr} \end{bmatrix}, j = 1, 2, \dots, n$$

Vector query :

$$Q = \begin{bmatrix} q_1 \\ \vdots \\ q_r \end{bmatrix}$$

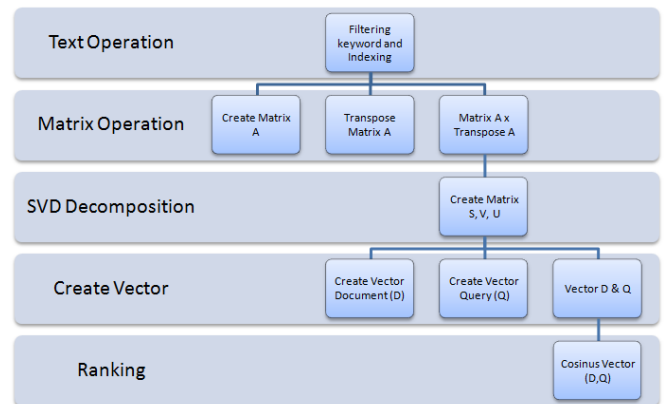
Rumus *cosinus sim* :

$$\text{Sim}(Q, D_j) = \cos(Q, D_j) = \frac{Q \cdot D_j}{|Q||D_j|} = \frac{1}{|Q||D_j|} \sum_{i=1}^r q_j \times d_{ji}$$

7) *Hasil akhir.*

Perhitungan *cosinus* sudut antara query, dan dokumen, diperoleh dan diurutkan berdasarkan dari yang paling besar sampai yang terkecil. Nilai *cosinus* sudut yang terbesar menunjukkan dokumen yang paling relevan dengan query.

### 3. Analisis Penerapan Metode LSI



Gambar 1. Bagan Analisa Penerapan Metode LSI

Dari gambar bagan di atas, dapat di gambarkan langkah-langkah penerapan metode LSI, berikut ini penjelasan setiap bagan :

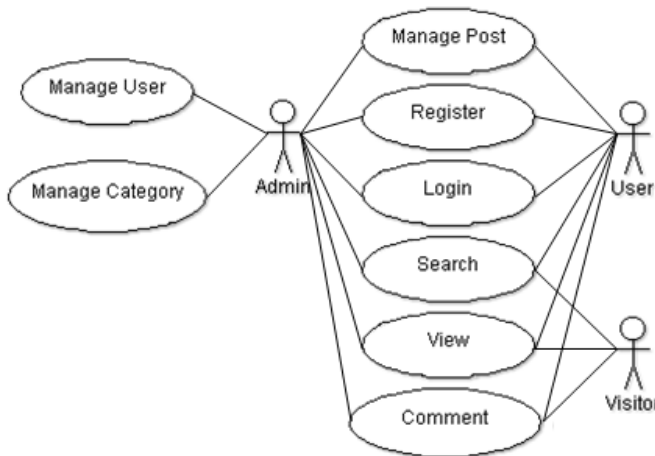
1. Tahap *Text Operation* adalah tahapan ketika pengguna melakukan *input keyword*, kemudian sistem akan melakukan *indexing* dan *filtering*.
  - *Input keyword* adalah memasukan kata kunci pada pencarian.
  - *Indexing* adalah tahapan mendata semua kata yang berhubungan dengan *keyword* pencarian.
  - *Filtering* adalah tahapan menghilangkan kata sambung, seperti (di, ke, dan).
2. *Matrix Operation* adalah tahapan membuat kata menjadi sebuah *matrix*, yang nantinya akan di gunakan untuk menghitung *svd decomposition*.

- *Create matrix A* yaitu membuat *matrix* bernama A dari index kata yang sudah dibuat.
- *Transpose matrix A* yaitu melakukan operasi transpose terhadap *matrix A*
- Perkalian *Matrix A* dan *Transpose Matrix A*, adalah tahapan mencari nilai dari hasil perkalian antara 2 *matrix*, dan akan menghasilkan *matrix* baru, hasilnya akan digunakan untuk menghitung SVD Decomposition.

3. SVD Decomposition adalah algoritma yang akan digunakan dalam metode LSI, dekomposisi dari SVD akan menghasilkan 3 *matrix* baru yaitu *matrix S*, *V* dan *U*, dalam penerapannya, akan menggunakan *library Package JAMA*.
4. *Create Vector* adalah tahapan mencari vector dokumen dan vector *query*, menggunakan rumus  $\mathbf{d} = \mathbf{d}^T \mathbf{U}_k \mathbf{S}_k^{-1}$ , Sedangkan vector *query*  $\mathbf{q} = \mathbf{q}^T \mathbf{U}_k \mathbf{S}_k^{-1}$
5. *Ranking* adalah tahapan untuk mencari *similarity*, atau kesamaan antara sudut dari *vector document* dan sudut *vector query*, dalam tahapan ini dapat menggunakan rumus similarity sim, yaitu :

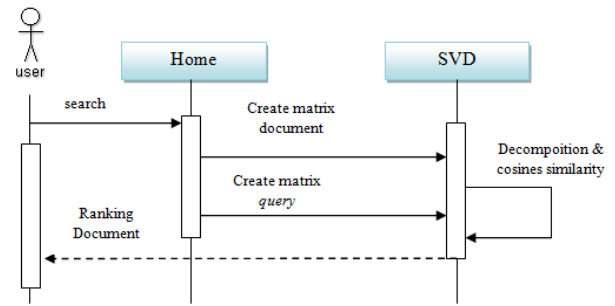
$$\text{Sim}(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} + \mathbf{d}}{|\mathbf{q}| + |\mathbf{d}|}$$

### 3.1 Use Case Diagram Sistem



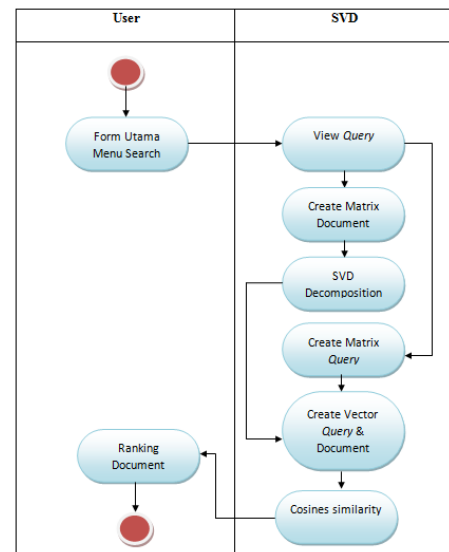
Gambar 2. Use Case Diagram Sistem

### 3.2 Sequence Diagram Search



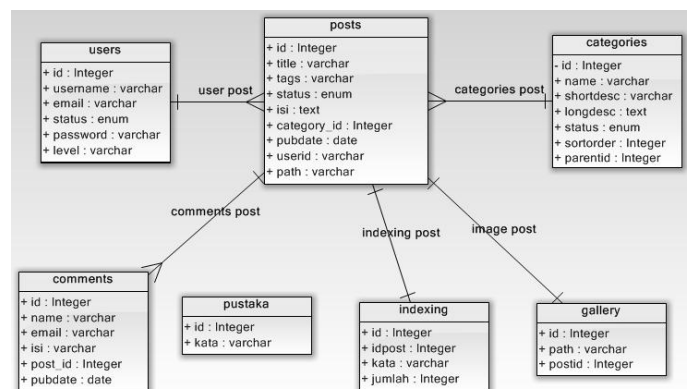
Gambar 3. Sequence Diagram Sistem

### 3.3 Activity Diagram Search



Gambar 4. Activity Diagram Search

### 3.4 ERD (Entity Relationship Diagram)



Gambar 5. ERD

### 3.5 Uji coba Pencarian dengan metode LSI

Tahapan uji coba atau *testing* terhadap dua pencarian menggunakan *keyword* atau kata kunci yang sama. Tujuannya yaitu menguji hasil pencarian yang lebih relevan antara pencarian berdasarkan *query* dengan pencarian menggunakan metode LSI. Sehingga akan didapatkan kelemahan dan kelebihan antara kedua pencarian. Dalam uji coba ini akan dilakukan pengujian terhadap 14 dokumen. Berikut ini adalah list dokumen dapat dilihat dalam Tabel 1.

Tabel 1. Tabel Dokumen

No	Dokumen
1	Gunung Semeru
2	Gunung Bromo
3	Pantai Goa Cina
4	Pantai Watu Ulo
5	Gunung Arjuna
6	Gunung Kelud
7	Pantai Balaikambang
8	Gunung Sinabung
9	Pantai Sendang biru
10	Pantai Manggar
11	Gunung Banyak
12	Pantai Kuta
13	Gunung Kawi
14	Gunung Merapi

		Banyak, Gunung Kawi, Gunung Semeru, Gunung Bromo	
4	Pariwisata	Gunung Bromo, Pantai Goacina, Pantai watu ulo, Gunung Arjuna, Gunung Kelud, Pantai Balaikambang, Pantai kuta, Gunung Merapi	8
5	Pariwisata Malang	Gunung Bromo, Gunung Kawi, Gunung Banyak, Gunung Arjuna, Gunung Kelud, Gunung Merapi, Gunung Sinabung, Pantai Watu Ulo, Pantai goa cina, Pantai sendang biru, Pantai Balaikambang, Pantai Kuta.	13
6	Objek Wisata	Gunung Bromo, Gunung Semeru, Gunung Kawi, Gunung Banyak, Gunung Arjuna, Gunung Kelud, Gunung Merapi, gunung Sinabung	8
7	Pantai Goa Cina Malang	Pantai Goa cina, Pantai Sendang biru, Pantai Balaikambang, Pantai Watu ulo, Pantai kuta, Pantai manggar	6

### 3.6 Hasil Pencarian dengan metode LSI

Berikut ini adalah hasil pencarian dengan metode *latent semantic indexing* dapat dilihat dalam Tabel 2.

Tabel 2. Tabel Hasil Pencarian LSI

No	Kata kunci	Dokumen	Jumlah
1	Pantai	Pantai Manggar, Pantai Kuta, Pantai Balaikambang, Pantai Sendang biru, Pantai Goa cina, Pantai Watu ulo.	6
2	Pantai malang	Pantai Goa Cina, Pantai Sendang Biru, Pantai Balaikambang, Pantai Watu ulo, Pantai Kuta, Pantai Manggar	6
3	Gunung	Gunung Sinabung, Gunung Merapi, Gunung Kelud, Gunung Arjuna, Gunung	8

Berdasarkan hasil pencarian dengan metode LSI. Pencarian menghasilkan dokumen yang relevan jika di aplikasikan pada website pariwisata. Namun jumlah dokumen dan isi dari dokumen berpengaruh pada hasil pencarian. Karena kelebihan pencarian LSI yaitu menghasilkan pencarian yang memiliki keterkaitan kata antara dokumen satu dengan dokumen lainnya. Sehingga menghasilkan pencarian yang saling berkaitan.

## 4. Kesimpulan

Hasil uji coba pencarian dengan metode *latent semantic indexing* (LSI) dapat memberikan hasil pencarian dokumen saling berkaitan antara dokumen satu dan lainnya ketika diaplikasikan pada *website* pariwisata. Sehingga hasil pencarian lebih relevan dan optimal.

## Daftar Pustaka

- [1]Al Fatta, Hanif. (2007). *Analisis dan Perancangan Sistem Informasi*. Yogyakarta : Andi.
- [2]Andri, Kristanto. (2003), *Perancangan Sistem Informasi dan Aplikasinya*. Jakarta : Gava Media.
- [3]Bunyamin, Hendra. (2005). *Information Retrieval System dengan Metode Latent Semantic Indexing*. Tesis Departemen Teknik Informatika ITB.
- [4]Fowler, Martin. (2005). *UML Distilled Edisi 3*. Yogyakarta : Andi.
- [5]Henderi. (2008). *Unified Modelling Language : Konsep dan Implementasinya Pada Pemodelan Sistem Berorientasi Objek dan Visual (Buku I)*. Tangerang: Raharja Enrichment Centre.
- [6]Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, (2004) . *Metode Desain & Analisis Sistem Edisi 6, Mc Graw Hill Education*. Yogyakarta : Andi.
- [7]Jogiyanto, Hartono. (2005). *Analisis & Desain Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta : Andi.
- [8]Munawar. (2005). *Pemodelan Visual menggunakan UML*. Yogyakarta : Graha Ilmu.
- [9]Rumbaugh, James, Jacobson, Ivar and Booch, Grady. (2005). *The Unified Modeling Language User Guide*. New York : Addison Wesley Longman.
- [10]Satzinger, Jackson, Burd. (2010). *System Analysis and Design in a Changing World*. New York : Cengage Learning.
- [11]Scarvada, A.J, Tatiana Bouzdine-Chameeva, Susan Meyer Goldstein, Julie M. Hays, Arthur V. Hill. (2004). *A Review of the Causal Mapping Practice and Research Literature*. Mexico : POM Conference.