

DESAIN SISTEM INFORMATION RETRIEVAL PADA EBOOK MENGGUNAKAN LATENT SEMANTIC INDEXING

Gagak Susanto¹, Hari Lugis Purwanto²

Jurusan Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Kanjuruhan Malang^{1,2}
gagak.susanto@unikama.ac.id, hari_lugis@unikama.ac.id

Abstrak *Ebook* adalah sebuah buku dalam versi digital, di mana buku tersebut dapat dibuka melalui perangkat elektronik. Contoh alat elektronik: komputer, *gadget*, *hanphone* dll. Bagi guru/pengajar maupun siswa/mahasiswa tentunya memiliki koleksi *ebook* yang banyak dalam komputer dan lebih cenderung hanya tersimpan dalam *hardisk* dan jarang sekali dibuka padahal isi dalam *ebook* tersebut mengandung bermacam-macam ilmu pengetahuan. Didalam *ebook* banyak informasi bisa kita dapatkan dan menjadi sumber ilmu pengetahuan yang bermanfaat. Supaya dari *ebook* tersebut tentunya dapat dimanfaatkan untuk merubah komputer menjadi asisten cerdas yang sangat bermanfaat bagi pemiliknya. Asisten cerdas akan memberikan referensi dengan cepat mengenai informasi yang dibutuhkan oleh pemilik komputer. Oleh karena itu pemanfaatan *ebook* untuk dasar pembentukan sumber ilmu pengetahuan memerlukan suatu media yang menjembatani antara pemilik dan perangkat. Untuk mempermudah pengelolaan koleksi *ebook* maka diperlukan sebuah sistem *information retrieval*, sehingga sehingga proses pencarian informasi dapat dilakukan dengan cepat. Proses *information retrieval* dilakukan dengan menggunakan metode Latent Semantic Indexing. LSI merupakan 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. Langkah awal dalam pembuatan sistem ini yaitu lebih dahulu peneliti mempersiapkan desain *Information retrieval* pada *Ebook* dan metode pencarian yang dipakai menggunakan Latent Semantic Indexing.

Kata Kunci: *Information Retrieval*, *LSI (Latent Semantic Indexing)*, *Ebook*.

PENDAHULUAN

Para pengguna baik para pengajar, mahasiswa, dan siswa saat memanfaatkan mesin pencarian untuk mencari sebuah informasi terkadang tidak sadar bahwa informasi yang dibutuhkan terdapat pada koleksi *ebook* yang sudah tersimpan di komputernya. Jika mau memanfaatkan koleksi *ebook* juga masih belum lengkapnya aplikasi yang mempunyai fasilitas pencarian sampai ke isi *ebook*.

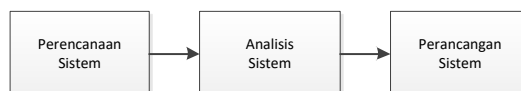
Berangkat dari permasalahan tersebut sebenarnya media komputer baik dalam bentuk personal komputer ataupun laptop yang telah dimiliki para pengajar dapat dijadikan asisten pribadi karena telah dibekali dengan berbagai *file-file* ilmu pengetahuan yang selama ini disimpan dalam bentuk *ebook*. Layaknya seorang asisten kerja, media tersebut seharusnya dapat memberikan informasi atau *feedback* yang cepat dan mudah atas permasalahan yang ada. Oleh karena itu *information retrieval* yang ditanam dalam komputer akan menjadi asisten yang baik bagi para pengajar.

Salah satu metode yang cukup terkenal dan diterapkan oleh raksasa mesin pencarian google untuk penggunaan karakter string dalam dokumen untuk menetapkan relevansi semantik untuk istilah pencarian (*keyword*) yang digunakan atau dengan kata lain, untuk membantu membangun makna sebenarnya dari teks pada posting blog atau halaman web adalah metode Latent Semantic Indexing (LSI). LSI merupakan 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. Langkah awal dalam pembuatan sistem ini yaitu lebih dahulu peneliti mempersiapkan desain *Information Retrieval* pada *Ebook* dan metode yang dipakai pencarian menggunakan Latent Semantic Indexing .

METODOLOGI PENELITIAN

Tahapan metodologi penelitian yang dilakukan dalam penelitian ini bisa dilihat pada gambar 1.



Gambar 1. Tahapan Metodologi Penelitian

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan tahapan- tahapan sebagai berikut :

A. Perancangan Sistem

Pada tahap ini, peneliti melakukan analisis kelayakan (*feasibility*) dengan memfokuskan pada aspek kebutuhan dan kesulitan pengguna. Untuk mendapatkan informasi tentang aspek kebutuhan dan kesulitan pengguna, peneliti melakukan penyebaran kuisisioner terhadap pengguna ebook yang berada di lingkungan perpustakaan, seperti dosen, mahasiswa, guru dan pegawai.

B. Analisis Sistem

Melakukan studi literatur yang berhubungan dengan permasalahan, mengklasifikasikan permasalahan dan solusinya dan membandingkan permasalahan yang akan diselesaikan ini, dan membuat analisa dan identifikasi kebutuhan sistem.

C. Perancangan Sistem

Feature dan operasi-operasi apa saja di deskripsikan dengan jelas, diantaranya menganalisa alur data diantara *entity*, membuat arsitektur sistem dan algoritma perhitungan metode *latent sematic indexing*, membuat database dan hubungan antar tabel yang ada di database, membuat user interface. Perangkat lunak yang dipakai untuk mendeskripsikan dari kasus ke sistem diantaranya sebagai berikut :

1. *FlowChart*
2. *Context Diagram* (CD)
3. *Data Flow Diagram* (DFD)
4. *Entitiy Relationship* (ERD)..

HASIL DAN PEMBAHASAN

A. Perancangan Sistem

Jumlah kuisisioner yang disebarakan sebanyak 50 lembar, namun kembali sejumlah 40 lembar. Adapun hasil dari kuisisioner secara lengkap terlampir, sedangkan kesimpulan hasil kuisisioner dijelaskan pada tabel 1 berikut.

Tabel 1. Hasil Kuisisioner Kebutuhan Pengguna

No	Informasi	Persentase
1	Jumlah koleksi <i>ebook</i> yang dimiliki.	76%
2	Pengalaman dalam mengelola koleksi <i>ebook</i>	75%
3	Pencarian <i>ebook</i> menggunakan aplikasi, secara manual di <i>explorer</i> atau cara lainnya.	70%
4	Ketepatan pencarian <i>ebook</i> terhadap kebutuhan berdasarkan kata kunci.	65%

5	Ketepatan Kata kunci berdasar judul, penulis atau isi.	71%
6	Pengalaman menggunakan aplikasi yang menemukan <i>ebook</i> berdasarkan isi.	76%
7	Fitur yang diperlukan dalam mengelola koleksi <i>ebook</i> perlu ditambah.	78%

B. Analisis Sistem

Hasil analisa dari jawaban kuisener yang telah diisi responden, maka di temukan adanya permasalahan dalam pencarian informasi yang relefan di sebuah koleksi *ebook* yaitu masih belum adanya fitur pencarian berdasarkan isi *ebook*, sehingga diperlukan fitur yang mampu mencari informasi yang relefan di sekumpulan *ebook*. Maka dari itu aplikasi yang akan dibangun ini untuk membantu permasalahan tersebut, dan dalam pencarian sebuah informasi ini menggunakan metode yang dipilih adalah *Latent Semantic Indexing* dibandingkan dengan isi ebook. Bahkan, aplikasi ebook yang dapat melakukan pencarian berdasarkan isi masih belum ditemukan, sehingga penelitian ini perlu untuk dikembangkan.

C. Perancangan Sistem

Deskripsi Sistem

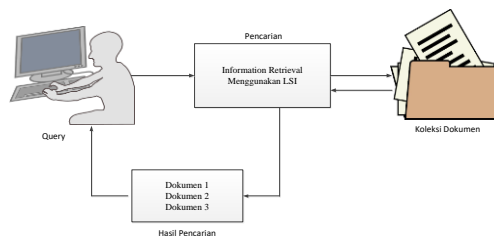
Salah satu aplikasi umum dari sistem *information retrieval* adalah search engine atau mesin pencarian yang terdapat pada jaringan internet. Pengguna dapat mencari halaman web yang dibutuhkannya melalui search engine. Contoh lain dari *Information Retrieval* system adalah sistem informasi perpustakaan.

Information Retrieval sistem terutama berhubungan dengan pencarian informasi yang isinya tidak memiliki struktur. Demikian pula ekspresi kebutuhan pengguna yang disebut query, juga tidak memiliki struktur. Hal ini yang membedakan IR system dengan sistem basis data. Dokumen adalah contoh informasi yang tidak terstruktur. Isi dari suatu dokumen sangat tergantung pada pembuat dokumen tersebut. Sebagai suatu sistem, *Information Retrieval* system memiliki beberapa bagian yang membangun sistem secara keseluruhan.

Untuk memperoleh sebuah informasi pada sebuah sekumpulan data ebook terkadang sulit karena perlu membuka ebook dahulu baru mencari kata yang relevan. Ternyata hal tersebut membutuhkan waktu dan perhatian yang sangat besar. Hasil query di dokumen dikatakan relevan pada umumnya mempunyai arti:

- Memuat kata atau kalimat yang sama dengan kata yang dicari.
- Memuat kata atau kalimat yang maknanya sama dengan yang dicari.

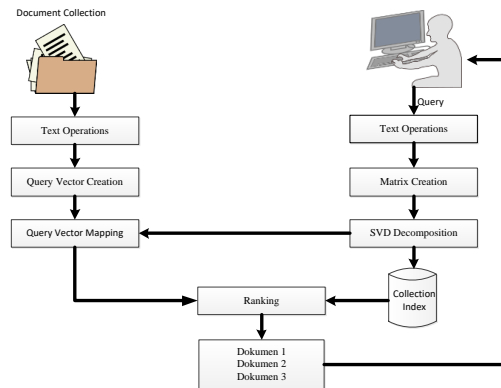
Secara garis besar desain LSI Search Engine yang akan dibangun pada penelitian ini bisa dilihat pada gambar 2.



Gambar 2. Arsitektur LSI Search Engine

B. Metode Latent Semantic Indexing Secara Keseluruhan

Secara global, alur proses metode Latent Semantic Indexing (LSI) dapat diilustrasikan dalam gambar 3.

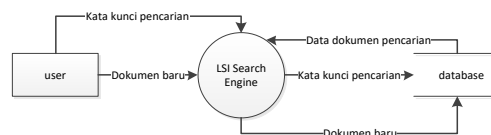


Gambar 3. Alur proses dari metode Latent Semantic Indexing

Alur proses dari metode Latent Semantic Indexing dibagi 2 (dua) kolom, yaitu kolom sebelah kanan yaitu query dan kolom sebelah kiri yaitu, koleksi dokumen. Pada proses sebelah kanan, query diproses melalui operasi teks, kemudian vektor query dibentuk. Vektor query yang dibentuk dipetakan menjadi vektor query terpeta (*mapped query vector*). Dalam membentuk query terpeta, diperlukan hasil dekomposisi nilai singular dari koleksi dokumen. Pada koleksi dokumen, dilakukan operasi teks pada koleksi dokumen, kemudian matriks kata dokumen (*terms-documents matrix*) dibentuk, selanjutnya dilakukan dekomposisi nilai singular (*Singular Value Decomposition*) pada matriks kata-dokumen. Hasil dekomposisi disimpan dalam *collection index*. Untuk melakukan pengurutan atau ranking dapat dilakukan dengan cara menghitung relevansi antara vektor *query* terpeta dan *collection index*. Selanjutnya, hasil perhitungan relevansi ditampilkan ke pengguna.

C. Context Diagram (CD)

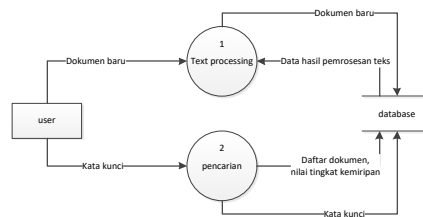
Desain Context Diagram pada gambar 4. menerangkan bahwa gambaran secara umum yang melibatkan satu entitas yaitu *user*. Pertama *user* memasukkan sekumpulan file *ebook* ke dalam sistem, dan sistem akan menyimpan data index *ebook* ke database. Setelah sistem itu *user* memasukkan kata kunci pencarian ke sistem LSI *Search Engine*. Sistem akan mencari kata kunci yang sesuai dengan isi *ebook* yang tersimpan di *database*, dan jika ada maka akan ditampilkan daftar *ebook* yang isinya relevan dengan kata kunci pencarian ke *user*.



Gambar 4. Context Diagram Search Engine

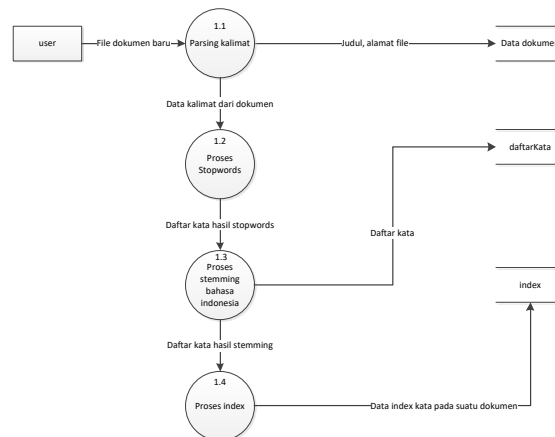
D. Data Flow Diagram Level 1 (DFD Level-1)

User saat memasukkan data baru, maka akan dilakukan *text processing* yang tujuannya menghilangkan imbuhan kata depan, imbuhan kata belakang, kata hubung, sehingga hanya kata dasar yang akan disimpan ke database. *User* juga bisa melihat hasil *text processing* yang setelah selesai. Berikutnya user memasukkan kata kunci pencarian ke dalam proses pencarian, dan akan diproses kata kunci yang telah dimasukkan dengan mencari ke database. Hasil dari pencarian akan menghasilkan tingkat kemiripan yang ditemukan di dokumen dan ditampilkan ke *user*. Gambar 4.4 memperlihatkan DFD Level 1 LSI *Search Engine*.



Gambar 5. DFD Level 1 LSI Search Engine

Pada DFD level 1.1 ini *user* memasukkan dokumen baru dan selanjutnya dokumen baru tersebut akan di proses *parsing* kalimat dengan tujuan menentukan struktur sebuah kalimat berdasarkan *grammar* dan *lexicon* tertentu. Lokasi file dokumen dan judul akan disimpan ke data dokumen. Hasil dari proses parsing kalimat adalah data kalimat yang selanjutnya dilakukan proses *stopwords* supaya kalimat yang mengandung kata umum (*common words*) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna dihilangkan, contohnya “yang”, “di”, “ke”. Daftar kata hasil *stopwords* akan di proses *stemming* supaya mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root word*) dengan menggunakan aturan-aturan tertentu. Daftar kata dari proses *stemming* akan disimpan ke data daftarkata, dan di lanjutkan proses *index*, yaitu membangun basis data indeks dari koleksi dokumen, karena dalam pencarian sangat dibutuhkan *indexing*. Hasil *Indexing* akan disimpan ke data *index*. Lebih jelasnya DFD level 1.1 bisa dilihat pada gambar 6.

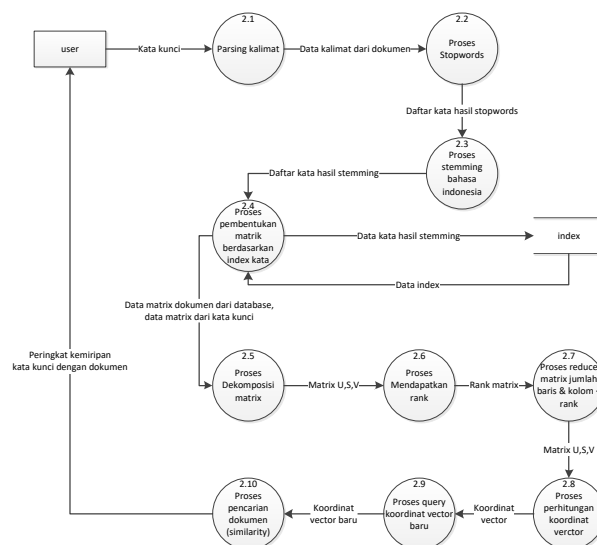


Gambar 6. DFD Level 1.1 Parsing kalimat

E. Data Flow Diagram Level 1.2

Pada DFD level 1.2 ini *user* memasukkan kata pencarian, kata yang dicari akan diparsing dengan tujuan menentukan struktur sebuah kalimat berdasarkan *grammar* dan *lexicon* tertentu. Hasil dari proses parsing adalah data kalimat yang selanjutnya dilakukan proses *stopwords* supaya kalimat yang mengandung kata umum (*common words*) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna dihilangkan, contohnya “yang”, “di”, “ke”. Daftar kata hasil *stopwords* akan di proses *stemming* supaya mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root word*) dengan menggunakan aturan-aturan tertentu. Daftar kata dari proses *stemming* akan diproses menjadi suatu matrix berdasarkan *index* kata, dan data hasil *stemming* juga disimpan ke data *index*. *Index* yang ada akan diambil dari data *index* sebagai pembentuk matrix. Berikutnya di lanjutkan proses *index*, yaitu membangun basis data indeks dari koleksi dokumen. Data matrix dokumen dari database data matrix dari data kunci akan diproses dekomposisi matrix, sehingga menghasilkan matrix U,S,V. Matrix U,S,V di rangking sehingga mendapatkan *rangking* matrix. *Rangking* matrix akan direduce jumlah baris kolom, dan mendapatkan matrix baru yaitu matrix U,S,V.

Selanjutnya dilakukan pembentukan dari koordinat *vector*, sehingga didapatkan koordinat *vector*. Dari Koordinat *vector* tersebut nantinya akan dicari *query* koordinat *vector* yang baru,. Dari koordinat *vector* baru akan diproses pencarian dokumen yaitu mencari tingkat kedekatannya (*similarity*), dan hasilnya adalah dokumen yang mempunyai tingkat kemiripan dengan kata yang dicari user. DFD level 2.1 bisa dilihat pada gambar 7.



Gambar 7. DFD Level 2.1 Pencarian

PNUTUP

A. Kesimpulan

Beberapa point yang dapat disimpulkan dalam penelitian ini adalah desain yang dirancang sudah menggambarkan alur sistem utama dan alur proses pada metode Latent Semantic Indexing .

B. Saran

Saran untuk perbaikan dalam penelitian ini adalah metode yang dipakai bisa di kombinasikan dengan metode yang lain untuk memperbaiki kecepatan, sehingga desain yang dibuat ikut menyesuaikan juga.

DAFTAR RUJUKAN

- Alhenshiri, A. A. (2010). Web Information Retrieval and Search Engines Techniques. Al- Satil, 55–92.
- B,D., & Hetami, A. (2015). PERANCANGAN INFORMATION RETRIEVAL (IR) UNTUK PENCARIAN IDE POKOK TEKS ARTIKEL BERBAHASA INGGRIS DENGAN PEMBOBOTAN VECTOR SPACE MODEL. Jurnal Ilmiah Teknologi Dan Informasi ASIA, 9(1).
- Berry, M. W., Dumais, S. T., & Brien, G. W. O. (2005). Using Linear Algebra for Intelligent Information Retrieval.
- Chen, C.-M. C. C.-M., Stoffel, N., Post, M., Basu, C., Bassu, D., & Behrens, C. (2001).
- Telcordia LSI Engine: implementation and scalability issues. Proceedings Eleventh

International Workshop on Research Issues in Data Engineering Document Management for Data Intensive Business and Scientific Applications RIDE 2001, (Ride),51–58.

Cholifah, Purwananto, Y., & Bramantoro, A. (2012). Aplikasi Information Retrieval Untuk Pembentukan Tesaurus Berbahasa Indonesia Secara Otomatis. SCAN, II.

Deerwester, S., & Dumais, S. (1990). Indexing by latent semantic analysis. Journal of the American ... (Vol. 41).

Hayati, S., Budi, A. S., Handoko, E., Fisika, M. P., & Jakarta, U. N. (2015).