**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# **IDENTITAS PENGUSUL**

Nama : **Novario Jaya Perdana**

NRP : **5107 100 069**

Dosen Wali : Yudhi Purwananto, S.Kom., M.Kom

1. **JUDUL TUGAS AKHIR**

***Implementasi Perekomendasian Kata Kunci Dokumen Menggunakan Algoritma Google Latent Semantic Distance***

1. **LATAR BELAKANG**

Keakuratan hasil pencarian menggunakan mesin pencari bergantung pada kata kunci yang dimasukkan oleh pengguna. Kurangnya informasi yang dimasukkan sebagai kata kunci dapat mempengaruhi hasil pencarian. Contohnya saat ingin menemukan informasi mengenai “Matahari Department Store”, tetapi yang dimasukan sebagai kata kunci adalah kata “Matahari”, sehingga hasil yang dikeluarkan adalah informasi yang berkaitan dengan matahari sebagai pusat tata surya.

Oleh *Google*, permasalahan ini dipecahkan dengan memberikan beberapa rekomendasi rangkaian kata kunci yang terkait. Sayangnya, kombinasi kata kunci yang diberikan sangat banyak dan merupakan rangkaian kata kunci yang sering dimasukkan oleh para pengguna lainnya. Hal ini mengakibatkan rekomendasi kata kunci tersebut tidak selalu sesuai dengan keinginan pengguna itu sendiri.

Beberapa peneliti telah mengembangkan teknik-teknik dalam pemberian rekomendasi kata kunci, antara lain ekpansi kata kunci dan ­*re-ranking*. Kedua teknik ini menggunakan perangkat-perangkat seperti *semantic nets*, *ontology* [6], dan *Markov chains* [1] untuk memodelkan ciri dan perilaku pengguna. Beberapa informasi berkaitan dengan profil pengguna akan dikumpulkan untuk mengetahui kecenderungan perilaku pengguna. Padahal pengguna sering kali mencari berbagai macam informasi dari berbagai bidang sehingga mempersulit penggambaran ciri perilaku pengguna.

Peneliti lainnya menggunakan *Google Similarity Distance* (NGD) untuk memprediksikan kata kunci yang akan dimasukan oleh pengguna [2]. Cara kerja algoritma ini adalah dengan menghitung hubungan antar kata kunci, lalu menggunakan hasil perhitungan tersebut untuk menyajikan peringkat *k* teratas rangkaian kata kunci terpenting kepada pengguna. Sayangnya, algoritma ini hanya dapat menyediakan informasi dengan satu kata kunci. Padahal agar dapat menemukan informasi yang lebih relevan, rangkaian kata kunci harus diperpanjang [3].

Menurut penelitian yang telah dilakukan oleh Chen et al, keakuratan hasil pencarian menggunakan rangkaian kata kunci yang terdiri dari tiga kata mencapai 70% [3]. Nilai ini lebih besar jika dibandingkan dengan keakuratan sebesar 55% yang didapat dari pencarian dengan menggunakan rangkaian kata kunci yang hanya terdiri dari satu kata [2]. Selain itu, Chen juga menjelaskan bahwa kelas kata yang paling baik digunakan untuk membentuk rangkaian kata kunci adalah kata benda atau kata sifat [2]. Rangkaian kata kunci yang baik adalah gabungan dari kedua kelas kata tersebut.

Perpanjangan rangkaian kata kunci dapat dilakukan dengan mengambil intisari dari informasi yang terdapat pada suatu dokumen. Pengambilan intisari tersebut dapat menggunakan algoritma *Google Latent Semantic Distance* (GLSD). Dengan menggunakan intisari yang diambil dari suatu artikel, pencarian informasi dapat dilakukan dengan lebih akurat dan cepat. Implementasi dari algoritma GLSD inilah yang menjadi perhatian utama dalam tugas akhir ini.

1. **RUMUSAN MASALAH**

Permasalahan dari tugas akhir yang akan dibahas adalah:

* + - 1. Bagaimana mendapatkan kata yang termasuk dalam kelas kata benda atau kata sifat dalam kalimat.
      2. Bagaimana mengimplementasikan algoritma *Google Latent Semantic Distance* (GLSD).
      3. Bagaimana akurasi hasil pencarian menggunakan rekomendasi rangkaian kata kunci yang dibentuk menggunakan algoritma GLSD.

1. **BATASAN MASALAH**

Dalam tugas akhir ini, terdapat beberapa batasan dalam proses pengerjaannya. Pengimplementasian algoritma GLSD ini dikemas dalam sebuah aplikasi *desktop* dan dikembangkan dalam bahasa *Java*. Data yang digunakan adalah dokumen digital berformat pdf. Kalimat dari dokumen tersebut akan diuraikan menggunakan sebuah perangkat *pdf parser*, kemudian diklasifikasikan sesuai kelasnya menggunakan perangkat *Part-of-Speech* (POS) *tagger*.

Percobaan kata kunci yang didapat dari hasil implementasi algoritma GLSD akan dilakukan dengan menggunakan mesin pencari *Google*.

1. **TUJUAN dan MANFAAT TUGAS AKHIR**

Tujuan dari tugas akhir ini adalah untuk mendapatkan informasi dari suatu artikel dengan cara membuat rangkaian kata kunci yang menggambarkan artikel tersebut. Rangkaian kata kunci ini dapat digunakan sebagai rekomendasi kepada pengguna sehingga hasil yang didapatkan dari proses pencarian menjadi lebih akurat.

Dengan adanya implementasi dari algoritma GLSD untuk pencarian dokumen ini diharapkan dapat membantu pengguna untuk menemukan informasi yang mereka inginkan dalam internet secara lebih akurat.

1. **RINGKASAN TUGAS AKHIR**

Arsitektur dari sistem yang akan dibangun terlihat pada Gambar 1. Pengimplemetasian algoritma GLSD ini terbagi atas tiga bagian utama, antara lain:



Gambar . Arsitektur sistem.

* + - 1. Penguraian kalimat pada dokumen berformat pdf.

Pada saat program dijalankan, program akan membaca tiap dokumen yang terdapat pada direktori penyimpanan. Tiap dokumen tersebut akan dipindai untuk dicari bagian abstrak. Kemudian, kalimat pada bagian tersebut akan diuraikan kata per kata menggunakan suatu perangkat *pdf parser*.

* + - 1. Penyaringan

Hasil uraian kalimat-kalimat pada abstrak suatu dokumen akan terdiri dari banyak sekali kata. Sistem akan memerlukan waktu komputasi yang sangat lama jika banyak kata yang terdapat dalam *database* untuk dihitung hubungan antar kata tersebut. Oleh karena itu dibutuhkan proses pernyaringan kata untuk menghilangkan kata-kata yang tidak terlalu penting. Ada tiga hal yang harus diperhatikan agar dapat memperoleh kata kunci yang potensial [2], antara lain:

* + - * 1. *Part-of-Speech* (POS) dan kombinasi kata

Sistem akan menggunakan kata-kata yang bertanda sebagai kata benda atau kata sifat untuk membuat rangkaian kata kunci. Hal ini telah dibuktikan oleh beberapa peneliti sebelumnya, [2] dan [3]. Mereka mengatakan bahwa kata-kata yang bertanda sebagai kata benda atau kata kunci adalah kata-kata yang berpotensi sebagai kata kunci pada pencarian informasi menggunakan mesin pencari. Selain itu, hal ini juga dapat meningkatkan akurasi sistem [2].

* + - * 1. Panjang kata

Kata-kata yang baik untuk dijadikan kata kunci adalah kata-kata yang memiliki panjang setidaknya 3 karakter, karena sebagian besar kata kunci memiliki panjang lebih dari 3 karakter, termasuk penyingkatan kata. Salah satu contohnya adalah “GPS” yang merupakan kependekan dari *“Global Positioning System”* dan memiliki panjang sebanyak 3 karakter.

* + - * 1. Banyaknya hasil pencarian menggunakan mesin pencari

Kata-kata yang telah didapatkan dari penguraian kalimat pada tahap pertama akan digunakan dalam mesin pencari, *Google*. Hal ini dimaksudkan untuk mengetahui seberapa baik kata tersebut untuk dijadikan sebagai kata kunci. Kata-kata yang memiliki hasil pencarian yang melebihi batas bisa digunakan sebagai kata kunci.

* + - 1. Algoritma *Google Latent Semantic Distance* (GLSD)
         1. *Google Similarity Distance* (NGD)

NGD adalah algoritma yang dapat digunakan untuk menghitung hubungan antara 2 kata dalam kata kunci [4]. Algoritma ini dirumuskan sebagai berikut:

(1)

Keterangan:

= banyaknya hasil pencarian menggunakan kata *x*.

= banyaknya hasil pencarian menggunakan kata *y*.

= banyaknya hasil pencarian menggunakan kata *x* dan *y*.

= banyaknya dokumen keseluruhan.

* + - * 1. *2-Gram Sequential Google Similirity Distance* (2-gram NGD)

Algoritma GLSD merupakan modifikasi dari 2-gram NGD. Untuk itu, sebelum mengetahui cara kerja GLSD, perlu untuk mengetahui 2-gram NGD. Penggunaan 2-gram NGD disini adalah untuk mengetahui hubungan antara 2 kata kunci yang saling berhubungan. Hubungan antara 2 kata kunci tersebut dapat diketahui dengan mencari probabilitas kemunculan keduanya.

Jika terdapat kata *x* pada suatu artikel, maka kata *y* juga harus ada dalam artikel tersebut dan berada dibelakang kata *x*. Hubungan antara keduanya didefinisikan dengan notasi “”. Probabilitas kemunculan keduanya dinyatakan dengan rumus:

(2)

Keterangan:

= banyaknya kemunculan *x* dan *y* dalam hasil pencarian.

= banyaknya dokumen keseluruhan.

Sementara itu, probabilitas kemunculan *x* itu sendiri pada hasil pencarian dirumuskan dengan:

(3)

(4)

sehingga,

(5)

dimana:

(6)

Dengan demikian, didapatkan rumus NGD yang baru, yaitu:

(7)

Kemudian, hasil NGD dapat dihitung dengan mengunakan rumus (7). Berdasarkan rumus tersebut, jumlah dokumen hasil pencarian dengan menggunakan kata kunci *x* dan kata kunci *y* yang mengikutinya akan mempengaruhi hasil NGD. Perhitungan secara keseluruhan untuk rangkaian “” dibagi menjadi dua kondisi.

1. Jika , maka

|  |
| --- |
| ;    ; |

1. Jika , maka

|  |
| --- |
| ;  ; |

Dari rumus tersebut, dapat diketahui bahwa jika jumlah hasil pencarian kata *x* lebih besar dari hasil pencarian kata yang mengikutinya, kata *y*, maka hasilnya adalah 1. Kebalikannya, jika jumlah hasil pencarian kata *x* lebih kecil dari hasil pencarian kata yang mengikutinya, *y*, maka hubungan keduanya dicari dengan menggunakan rumus:

(7)

* + - * 1. *Google Latent Semantic Distance*

Algoritma dasar NGD tidak dapat menghitung hubungan antar kata kunci yang berjumlah lebih dari dua. Oleh karena itu, dibutuhkan adanya modifikasi pada rumus dasar NGD. Hal ini dapat dicapai dengan menggunakan algoritma *Probabilistic Latent Semantic* (PLSA) [3].

PLSA adalah algoritma yang digunakan untk menganalisa kemunculan data secara bersamaan. Algoritma ini menggunakan *probability density function* untuk menemukan kemungkinan hubungan antara dokumen dan kata kunci [5].

Contohnya, terdapat kumpulan data *D* yang terdiri dari pasangan kata kunci dan dokumennya . Jika di dalam *D* tersebut terdapat *N* dokumen, maka kumpulan dokumen dapat diekspresikan sebagai . Dalam D juga terdapat kumpulan kata kunci sebanyak M yang dapat diekspresikan sebagai . Karena dokumen dan kata kunci saling berhubungan, maka terdapat variabel tersembunyi (*latent variable*) yang dinotasikan sebagai . Oleh karena itu, probabilitas kemunculan bersama dari pasangan dapat dihitung dengan menggunakan rumus:

(8)

(9)

Hubungan antar kata dalam rangkaian kata kunci dapat dilihat pada Gambar 2. Gambar tersebut melukiskan hubungan yang terjadi antara 3 buah kata dimana kata “r” dipengaruhi oleh kata “q” dan kata “s” dipengaruhi oleh kata “r”. Dengan adanya kata “r” maka hubungan antara kata “q” dan “s” akan lebih kuat.



Gambar . Hubungan antar kata pada kata kunci.

Contohnya, hubungan antara kata *Mona Lisa* dan kata *Leonardo Da Vinci* sangat erat karena sudah banyak yang mengenali bahwa lukisan berjudul *Mona Lisa* dilukis oleh *Leonardo Da Vinci*. Apabila di antara keduanya diberikan kata *Musee du Lourve*, maka akan memberikan pengertian bahwa lukisan berjudul *Mona Lisa* yang sekarang berada di *Musee du Lourve* dilukis oleh *Leonardo Da Vinci*. Hal ini akan meningkatkan akurasi karena meningkatkan hubungan antara si pelukis dan lukisannya, sehingga akan meningkatkan fokus pada topik pencarian.

Dengan menggunakan dasar pemikiran dari PLSA, NGD dapat dimodifikasi sehingga dapat digunakan untuk memprediksi hubungan antar lebih dari dua kata kunci. Rangkaian kata kunci yang digunakan mengikuti aturan “” yang dapat dicari dengan menggunakan rumus:

(10)

Dengan menggunakan rumus ini, tingkat kepentingan kata *y* dapat dievaluasi. Dengan menggunakan skenario pengaruh *x* lebih besar dari pengaruh *y* dan pengaruh *y* lebih besar dari pengaruh *z* atau dapat dikatakan(), maka

(11)

(12)

Sehingga,

(13)

Persamaan (13) dapat digunakan untuk mengganti persamaan (6) agar dapat mengakomodasi tambahan kata pada rangkaian yang akan dibuat. Proses perhitungannya menjadi seperti ini:

|  |
| --- |
|  |

Dengan demikian, algoritma 2-gram NGD dapat digunakan tanpa menggunakan mediator *y*. Algoritma tersebut digunakan untuk mengetahui hubungan antara “”. Perhitungan mencari hubungan kata kunci dengan rangkaian “” dapat menggunakan algoritma 3-gram NGD, atau disebut juga *Google Latent Semantic Distance* (GLSD). Jika nilai GLSD lebih kecil dibandingkan dengan nilai 2-gram NGD, maka rangkaian tersebut dapat dipakai sebagai kata kunci. Terdapat enam kondisi yang harus diperhatikan dalam perhitungan menggunakan GLSD yaitu:

|  |
| --- |
|  |

Perhitungan tersebut adalah saat hasil pencarian dengan menggunakan kata *x* lebih besar dibandingkan hasil pencarian dengan menggunakan kata *y* dan lebih besar dari hasil pencarian dengan menggunakan kata *z*. Jika kondisinya seperti itu, maka hubungan antara tiga kata, *x, y* dan *z*, dapat dihitung dengan menggunakan rumus:

(14)

|  |
| --- |
|  |

Kondisi yang kedua adalah dimana hasil pencarian dengan menggunakan kata *x* lebih besar dibandingkan hasil pencarian dengan menggunakan kata *z* dan lebih besar dari hasil pencarian dengan menggunakan kata *y*. Hubungan antara kata *x, y* dan *z* pada kondisi ini dapat dihitung dengan menggunakan rumus:

(15)

|  |
| --- |
|  |

Kondisi yang ketiga adalah dimana hasil pencarian dengan menggunakan kata *y* lebih besar dibandingkan hasil pencarian dengan menggunakan kata *x* dan lebih besar dari hasil pencarian dengan menggunakan kata *z*. Hubungan antara kata *x, y* dan *z* pada kondisi ini dapat dihitung dengan menggunakan rumus:

(16)

|  |
| --- |
|  |

Kondisi yang keempat adalah dimana hasil pencarian dengan menggunakan kata *z* lebih besar dibandingkan hasil pencarian dengan menggunakan kata *x* dan lebih besar dari hasil pencarian dengan menggunakan kata *y*. Hubungan antara kata *x, y* dan *z* pada kondisi ini dapat dihitung dengan menggunakan rumus:

(17)

|  |
| --- |
|  |

Kondisi yang kelima adalah dimana hasil pencarian dengan menggunakan kata *y* lebih besar dibandingkan hasil pencarian dengan menggunakan kata *z* dan lebih besar dari hasil pencarian dengan menggunakan kata *x*. Hubungan antara kata *x, y* dan *z* pada kondisi ini bernilai 2.

|  |
| --- |
|  |

1. Jika , maka

Kondisi yang keenam adalah dimana hasil pencarian dengan menggunakan kata *z* lebih besar dibandingkan hasil pencarian dengan menggunakan kata *y* dan lebih besar dari hasil pencarian dengan menggunakan kata *x*. Hubungan antara kata *x, y* dan *z* pada kondisi ini dapat dihitung dengan menggunakan rumus:

(18)

1. **METODOLOGI**

Pengerjaan tugas akhir ini terdiri dari beberapa tahap, yaitu :

1. Studi Literatur

Pada tahapan ini akan dilakukan studi literatur mengenai metode yang digunakan, diantaranya :

* + - * 1. Berbagai pustaka mengenai rumusan masalah (*paper* mengenai sistem temu kembali informasi yang menggunakan algoritma *Google Latent Semantic Distance* dan algoritma lainnya yang mendukung).
        2. Cara menguraikan kalimat pada halaman bertipe html menjadi rangkaian kata-kata dan menandainya dengan fungsi kata tersebut pada kalimat.
        3. Metode *Google Latent Semantic Distance*.

1. Implementasi Perangkat Lunak

Implementasi merupakan tahap untuk membangun sistem. Implementasi terdiri atas perancangan dan pembuatan mesin pencari dokumen dengan menggunakan algoritma *Google Latent Semantic Distance*.

1. Pengujian dan Evaluasi

Pada tahap ini dilakukan uji coba terhadap sistem yang telah dibuat, mengamati kinerja sistem yang baru dibuat, serta mengidentifikasi kendala yang mungkin timbul. Pengujian dilakukan dengan mencoba memasukkan beberapa kata kunci. Evaluasi dianggap berhasil jika keakuratan hasil temu kembali informasi mencapai 70%.

1. Penyusunan Buku Tugas Akhir

Tahap terakhir merupakan penyusunan laporan. Laporan ini memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perancangan yang telah dibuat. Secara garis besar, buku laporan tugas akhir ini terdiri atas beberapa bagian yaitu:

1. Pendahuluan
   1. Latar Belakang
   2. Permasalahan
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Uji Coba dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka
7. **JADWAL KEGIATAN TUGAS AKHIR**

Tugas akhir ini diharapkan bisa dikerjakan menurut jadwal sebagai berikut:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Kegiatan | Bulan (2011) | | | | | | | |
| Februari | | Maret | | April | | Mei | |
| 1 | Penyusunan Proposal Tugas Akhir |  |  |  |  |  |  |  |  |
| 2 | Studi Literatur |  |  |  |  |  |  |  |  |
| 3 | Implementasi Perangkat Lunak |  |  |  |  |  |  |  |  |
| 4 | Pengujian dan Evaluasi |  |  |  |  |  |  |  |  |
| 5 | Penyusunan Buku Tugas Akhir |  |  |  |  |  |  |  |  |

**DAFTAR PUSTAKA**

1. Borges, J., dan Lavene, M. 2007. “Evaluating Variable-length Markov Chain Models for Analysis of User Web Navigation Sessions”. *IEEE Transaction on Knowledge and Data Engineering*. Volume 19 (4), 441-452.
2. Chen, P.I., dan Lin, S.J. 2010. “Automatic Keyword Prediction using Google Similarity Distance”.Jurnal *Expert System with Aplication*, Volume 37 (3), 1928-1938.
3. Chen, P.I., Lin, S.J., dan Chu, Y.C. 2011. “Using Google Latent Semantic Distance to Extract the Most Relevant Information”. Jurnal *Expert System with Application*, Volume 38 (6), 7349-7358.
4. Cilibrasi, R.L., dan Vitanyi, P.M.B. 2007. “The Google Similarity Distance”. *IEEE Transcations on Knowledge and Data Engineering*. Volume 19 (3), 370-383.
5. Hoffman, T. 1999. “Probabilistic Latent Semantic Indexing”. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 50-57.
6. Zhong, N. 2002. “Representation and Construction of Ontologies for Web Intelligence”. *International Journal of the Foundation of Computer Science*. Volume 13 (4), 555-570.

**LEMBAR PENGESAHAN**

###### **Surabaya, 29 Oktober 2010**

Menyetujui,

Dosen Pembimbing

Diana Purwitasari, S.Kom, M.Sc

NIP : 19780410 200312 2 001