Computatio: Journal of Computer Science and Information Systems, 2/2 (2018), 186-195

IMPLEMENTASI ALGORITMA *GOOGLE LATENT SEMANTIC DISTANCE* UNTUK EKSTRAKSI RANGKAIAN KATA KUNCI ARTIKEL JURNAL ILMIAH

Novario Jaya Perdana
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
E-mail : novariojp@fti.untar.ac.id

Abstrak

Keakuratan hasil pencarian pada mesin pencari bergantung pada kata kunci yang digunakan. Kurangnya informasi yang diwakili oleh kata kunci dapat menyebabkan berkurangnya akurasi hasil pencarian. Dalam penelitian ini, algoritma Google Latent Semantic Distance digunakan dalam sebuah aplikasi perekomendasian kata kunci dokumen untuk mengekstrak rangkaian kata yang dapat mencerminkan isi dari dokumen tersebut. Rangkaian tersebut dapat digunakan sebagai rekomendasi kata kunci dalam pencarian dokumen menggunakan mesin pencari. Hasil dari implementasi perekomendasian kata kunci dokumen ini memperlihatkan akurasi yang tinggi dalam menemukan kembali dokumen yang relevan pada hasil pencarian teratas.

Kata kunci*-Ekstraksi Informasi, Rangkaian Kata Kunci, Google Latent Semantic Distance*

Abstract

The accuracy of search result using search engine depends on the keywords that are used. Lack of the information provided on the keywords can lead to reduced accuracy of the search result. This means searching information on the internet is a hard work. In this research, a software has been built to create document keywords sequences. The software uses Google Latent Semantic Distance which can extract relevant information from the document. The information is expressed in the form of specific words sequences which could be used as keyword recommendations in search engines. The result shows that the implementation of the method for creating document keyword recommendation achieved high accuracy and could finds the most relevant information in the top search results.

Keywords*-Information Extraction, Keywords Sequence, Google Latent Semantic Distance*

1. PENDAHULUAN

Kata kunci dapat diartikan sebagai ikhtisar singkat dari sebuah teks dokumen, karena kata kunci merupakan rangkaian yang terdiri dari satu atau lebih kata yang dipilih secara khusus untuk menggambarkan keseluruhan isi teks. Oleh karena itu, kata kunci biasanya diambil dari bagian penting dari sebuah teks dokumen seperti judul, abstrak ataupun ringkasan dari isi sebuah dokumen [1]. Kegunaan kata kunci tidak terbatas hanya pada proses pembuatan indeks dokumen dalam mesin pencari ataupun kategorisasi teks, namun juga untuk proses temu

kembali halaman *website*, peringkasan dokumen, dan lain-lain. Jika dapat memperoleh kata kunci yang baik, proses pencarian sebuah dokumen dalam mesin pencari menjadi lebih mudah dan tepat.

Penelitian mengenai ekstraksi kata kunci dokumen telah dilakukan oleh beberapa peneliti sebelum ini. Contohnya dengan pendekatan ekspansi kata kunci dan *re-ranking*. Kedua teknik ini menggunakan perangkat-perangkat seperti *semantic nets*, *ontology*, dan *Markov chains* untuk memodelkan ciri dan perilaku pengguna [2]. Beberapa informasi berkaitan dengan profil pengguna akan dikumpulkan untuk mengetahui kecenderungan perilaku pengguna. Sehingga hasil pencarian akan dicocokan dengan profil pengguna yang telah disimpan sebelumnya. Padahal pengguna sering kali mencari berbagai macam informasi dari berbagai bidang sehingga mempersulit penggambaran ciri perilaku pengguna.

Beberapa pendekatan lainnya pun sudah dilakukan, seperti menggunakan *lexical chains* untuk ekstraksi kata kunci secara otomatis yang sering digunakan untuk pembuatan ringkasan teks secara otomatis [3]. Kata kunci diekstrak dengan menghitung hubungan semantik yang terdapat pada antar kata dalam kalimat. Dari pendekatan ini, didapatkan beberapa kata yang memiliki hubugan semantik terkuat dengan kata lainnya di dalam teks, yang kemudian dijadikan kata kunci. Kekurangan dari pendekatan ini adalah bahwa kata kunci yang didapatkan hanya beberapa kata tunggal, padahal kata kunci yang terbaik terdiri dari beberapa kata yang tergabung sehingga menjadi rangkaian kata kunci [4].

Peneliti lainnya menggunakan *Google Similarity Distance* untuk memprediksikan kata kunci yang akan dimasukan oleh pengguna [5]. Cara kerja algoritma ini adalah dengan menghitung hubungan antar kata, lalu menggunakan hasil perhitungan tersebut untuk menyajikan peringkat *k* teratas dari rangkaian kata kunci yang didapatkan kepada pengguna. Seperti halnya dengan pendekatan *lexical chains*, algoritma ini hanya dapat menyediakan informasi dengan satu kata kunci.

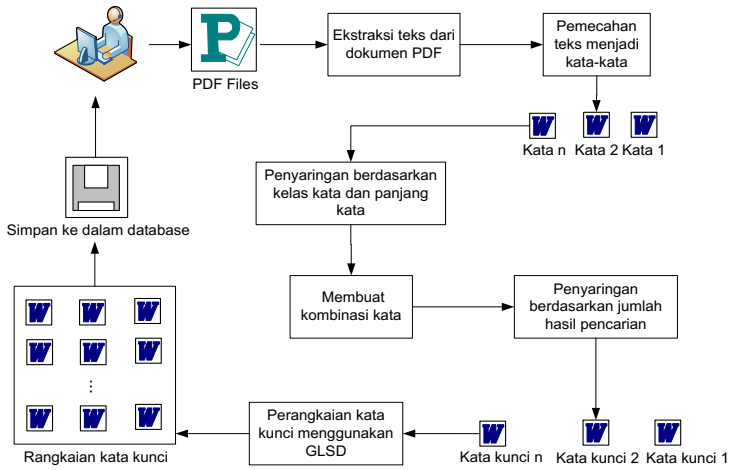
Keakuratan hasil pencarian menggunakan rangkaian kata kunci yang terdiri dari tiga kata lebih besar dibandingkan dengan penggunaan rangkaian kata kunci yang hanya terdiri dari satu kata [4]. Selain itu, kelas kata yang paling baik digunakan untuk membentuk rangkaian kata kunci adalah kata benda atau kata sifat. Rangkaian kata kunci yang baik adalah gabungan dari kedua kelas kata tersebut [5].

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 [4]. Implementasi dari algoritma GLSD inilah yang menjadi perhatian utama dalam penelitian ini. Pada algoritma ini, tiap kata dicocokan kelasnya, kemudian digabung menjadi frasa yang terdiri dari kata benda dan kata sifat. Setiap frasa kemudian dihitung hubungannya menggunakan GLSD. Pada penelitian ini, akan dilihat juga bagian mana dari sebuah artikel ilmiah yang paling baik untuk dijadikan sumber penarikan kata kunci. Oleh karena itu, uji coba dilakukan dengan membandingkan antara teks dari abstrak dan teks dari bagian lain selain abstrak dan daftar referensi serta kata kunci yang sudah disediakan oleh penulis artikel yang bersangkutan. Keberhasilan metode ini dalam menemukan kembali dokumen dihitung dengan menggunakan presisi dan *recall* serta *f-measure*.

2. METODE PENELITIAN

Terdapat beberapa tahap dalam proses mendapatkan rangkaian kata kunci pada algoritma GLSD. Ilustrasi mengenai tahapan yang harus dilalui tersebut dapat dilihat pada Gambar 1. Dimulai dari praproses seperti ekstraksi teks, pemecahan teks menjadi satuan kata, penyaringan

kelas kata, pembuatan kombinasi kata dan penyaringan berdasarkan jumlah hasil pencarian. Setelah itu adalah proses perangkaian kata kunci menggunakan GLSD.



Gambar 1. Tahapan dalam GLSD

2.1. Ekstraksi Teks dari Dokumen

Elemen yang dibutuhkan seluruh proses ini adalah teks dari suatu dokumen. Oleh karena itu, teks dalam dokumen harus diambil terlebih dahulu agar dapat diolah lebih lanjut. Kegiatan utama yang terjadi dalam proses ini adalah pengambilan isi dokumen PDF sebagai data masukan. Selain isi dokumen, diambil juga metadata dari dokumen tersebut. Metadata tersebut antara lain judul makalah ilmiah, dan penulis makalah tersebut. Proses pengambilan isi dokumen PDF ini memanfaatkan bantuan pustaka *pdf parser* yang sudah tersedia yaitu PDFBox [6].

Proses ini dimulai dari membaca metadata dokumen untuk mendapatkan judul dan penulis makalah ilmiah. Setelah itu, memeriksa data masukan pengguna berupa bagian dokumen yang akan dijadikan sebagai sumber ekstraksi. Hasil akhir dari proses ini adalah kumpulan kalimat yang merupakan teks isi dokumen.

2.2. Pemecahan Teks Dokumen Menjadi Kata-Kata

Tahap ini dimulai dari pemecahan teks menjadi satuan-satuan kata. Satuan-satuan kata tersebut kemudian diubah ke dalam bentuk huruf kecil (*lower case*). Setelah itu, dijalankan proses tokenisasi, yaitu proses penghilangan karakter-karakter selain huruf dan tanda sambung seperti angka, simbol dan tanda baca lainnya. Hasil akhir dari proses ini adalah kumpulan kata-kata terpisah yang bersih dari karakter aneh selain huruf dan tanda sambung. Kumpulan kata ini akan menjadi data masukan bagi proses selanjutnya, yaitu proses penyaringan.

2.3. Identifikasi Jenis Kata Sesuai Kelas Kata Menggunakan POS Tagger

Tahap berikutnya adalah identifikasi jenis kata berdasarkan kelas kata. Hal ini dimaksudkan agar didapatkan kata-kata yang dapat menggambarkan isi dokumen. Kata-kata yang dipilih untuk dimasukkan dalam daftar kata adalah kata-kata yang termasuk dalam kelas kata benda dan kata sifat. Proses identifikasi jenis kata ini menggunakan bantuan pustaka penanda *part-of-speech* (*POS tagger*). Pustaka tersebut adalah *Stanford Log-Linear Part-Of-*

Speech Tagger, yaitu sebuah pustaka untuk mengenali kelas kata dari suatu kata dalam kalimat [7].

Proses identifikasi jenis kata ini dimulai dengan menandakan kata-kata dalam kumpulan kata dengan kelas kata masing-masing. Setelah itu, setiap kata diperiksa kelas katanya. Jika kata tersebut termasuk ke dalam kata benda dan kata sifat serta jumlah karakter dalam kata tersebut lebih dari tiga, maka kata tersebut ditambahkan ke dalam daftar kata. Hasil akhir dari proses ini adalah daftar kata-kata yang termasuk ke dalam kata benda dan kata sifat. Daftar kata ini akan menjadi data masukan untuk tahapan pembentukan frasa.

2.4. Pembentukan Frasa

Agar rangkaian kata kunci yang dibangun lebih bermakna, elemen dari rangkaian kata kunci tersebut harus dapat mewakili isi dokumen secara keseluruhan. Salah satu caranya adalah dengan membentuk frasa dari kata-kata yang telah diekstrak. Frasa dibentuk dari kombinasi antara kata benda dengan kata benda atau kata sifat dengan kata benda.

Terdapat beberapa kondisi yang harus dipenuhi oleh suatu kata jika kata tersebut ingin ditambahkan ke dalam rangkaian kata. Kondisi pertama adalah kata tersebut termasuk ke dalam kelas kata benda, kata sifat atau *determiner*. Jika kata tersebut tidak memenuhi kondisi tersebut, maka kata tersebut tidak dapat ditambahkan ke dalam daftar rangkaian kata. Jika kata tersebut memenuhi kondisi tersebut, maka kondisi kedua harus diperiksa. Kondisi kedua adalah kata yang muncul setelah kata tersebut harus termasuk ke dalam kelas kata benda saja. Jika kondisi ini dapat dipenuhi, maka frasa dapat dibentuk dari kombinasi antara kata pertama dan kata kedua. Namun jika kondisi ini tidak dapat dipenuhi, maka frasa tidak dapat dibentuk, tetapi kata pertama dapat dimasukan ke dalam daftar kata kunci.

Hasil akhir dari proses pembentukan frasa ini adalah daftar kata dan frasa yang berpotensial menjadi salah satu anggota dalam rangkaian kata. Daftar kata dan frasa ini menjadi data masukan bagi proses identifikasi kata khusus menggunakan jumlah hasil pencarian pada mesin pencari.

2.5. Identifikasi Kata Khusus Menggunakan Jumlah Hasil Pencarian Pada Mesin Pencari

Daftar kata dan frasa dari hasil proses pembentukan frasa masih mengandung kata-kata dan frasa-frasa yang bersifat umum dan belum menggambarkan isi dokumen secara lebih baik. Oleh karena itu, diperlukan adanya proses penyaringan tambahan untuk mendapatkan kata-kata yang lebih khusus. Proses penyaringan tersebut adalah identifikasi kata khusus dengan memanfaatkan jumlah hasil pencarian pada mesin pencari.

Kata/frasa yang secara khusus dipergunakan dalam suatu bidang tertentu biasanya hanya terkandung dalam sedikit dokumen. Hal ini dapat dilihat dari jumlah pencarian menggunakan kata tersebut pada mesin pencari. Proses identifikasi ini menggunakan bantuan API (*Application Programming Interface*) dari mesin pencari *Yahoo! Search* [8]. Hasil akhir dari proses ini adalah daftar kata/frasa khusus yang dapat digunakan sebagai salah satu anggota dalam rangkaian kata kunci. Daftar kata khusus ini akan menjadi data masukan bagi proses perhitungan bobot rangkaian kata kunci.

2.6. Pembentukan Kata Kunci Menggunakan Google Latent Semantic Distance

Tahapan terakhir adalah perangkaian kata kunci menggunakan algoritma GLSD. Proses yang terdapat dalam tahap perangkaian ini adalah perangkaian kata kunci yang kemudian dilanjutkan dengan perhitungan bobot rangkaian untuk mendapatkan rangkaian yang paling baik. Rangkaian kata kunci dibentuk dari kombinasi kata-kata dan frasa yang terdapat dalam daftar kata khusus hasil proses penyaringan sebelumnya. Rangkaian tersebut terdiri dari tiga buah kata/frasa khusus. Proses perangkaiannya adalah dengan memasangkan satu kata/frasa dengan 2 kata/frasa lainnya.

Perangkaian kata kunci boleh membentuk rangkaian yang telah ada sebelumnya. Hal ini tergantung pada letak kata kunci pada rangkaian yang akan dibentuk. Setelah rangkaian kata kunci terbuat, rangkaian tersebut perlu dihitung bobotnya. Bobot tersebut menunjukan hubungan antar anggota dalam rangkaian tersebut. Setelah semua rangkaian kata kunci telah dibuat dan dihitung bobotnya, rangkaian-rangkaian tersebut perlu diurutkan secara menurun sesuai hasil perhitungan GLSD karena rangkaian kata kunci yang diperlukan adalah rangkaian dengan nilai GLSD tertinggi. Terdapat enam kondisi yang harus diperiksa saat akan menggunakan GLSD. Kondisi-kondisi ini terkait jumlah dokumen yang berhasil ditemukan untuk setiap kata kunci, yaitu $f(x)$ untuk jumlah hasil pencarian menggunakan kata x, $f(y)$ untuk jumlah hasil pencarian menggunakan kata y, dan $f(z)$ untuk jumlah hasil pencarian menggunakan kata z. Kondisi-kondisi tersebut antara lain:

1. $f(x) > f(y) > f(z)$

Kondisi pertama adalah saat jumlah hasil pencarian dokumen yang mengandung $\boldsymbol{x}$ lebih besar dibandingkan yang mengandung $\boldsymbol{y}$ dan keduanya lebih besar dibandingkan dengan hasil pencarian menggunakan kata $\boldsymbol{z}$. Hubungan antara ketiga kata tersebut dalam rangkaian dapat menggunakan rumus (1).

$$\boldsymbol{GLSD(x,y,z) = \frac{-\log f(y) - \log f(x) + 2\log N}{-\log f(x) + \log N}} \quad (1)$$

2. $f(x) > f(z) > f(y)$

Kondisi kedua adalah saat jumlah hasil pencarian dokumen yang mengandung $\boldsymbol{x}$ lebih besar dibandingkan yang mengandung $\boldsymbol{y}$, namun hasil pencarian menggunakan kata $\boldsymbol{z}$ lebih besar dibandingkan dengan hasil pencarian yang menggunakan kata $\boldsymbol{y}$. Hubungan antara ketiga kata tersebut dalam rangkaian dapat menggunakan (2).

$$\boldsymbol{GLSD(x,y,z) = \frac{-2\log f(y) + 2\log N}{-\log f(z) + \log N}} \quad (2)$$

3. $f(y) > f(x) > f(z)$

Kondisi ketiga adalah saat jumlah hasil pencarian dokumen yang mengandung $\boldsymbol{y}$ lebih besar dibandingkan yang mengandung $\boldsymbol{x}$, dan hasil pencarian menggunakan kata $\boldsymbol{x}$ lebih besar dibandingkan dengan hasil pencarian yang menggunakan kata $\boldsymbol{z}$. Hubungan antara ketiga kata tersebut dalam rangkaian dapat menggunakan (3).

$$\boldsymbol{GLSD(x,y,z) = \frac{-\log f(x) - \log f(z) + 2\log N}{-\log f(z) + \log N}} \quad (3)$$

4. $f(y) > f(z) > f(x)$

Kondisi keempat adalah saat jumlah hasil pencarian dokumen yang mengandung $\boldsymbol{y}$ lebih besar dibandingkan yang mengandung $\boldsymbol{x}$, dan hasil pencarian menggunakan kata $\boldsymbol{z}$ lebih besar dibandingkan dengan hasil pencarian yang menggunakan kata $\boldsymbol{x}$. Hubungan antara ketiga kata tersebut dalam rangkaian dapat menggunakan (4).

$$\boldsymbol{GLSD(x,y,z) = \frac{-2\log f(x) + 2\log N}{-\log f(x) + \log N} = 2} \quad (4)$$

5. $f(z) > f(x) > f(y)$

Kondisi kelima adalah saat jumlah hasil pencarian dokumen yang mengandung $\boldsymbol{z}$ lebih besar dibandingkan yang mengandung $\boldsymbol{x}$, dan hasil pencarian menggunakan kata $\boldsymbol{x}$ lebih besar dibandingkan dengan hasil pencarian yang menggunakan kata $\boldsymbol{y}$. Hubungan antara ketiga kata tersebut dalam rangkaian dapat menggunakan (5).

$$\boldsymbol{GLSD(x,y,z) = \frac{\log f(z) - \log f(x) - 2\log f(y) + 2\log N}{-\log f(x) + \log N}} \quad (5)$$

6. $f(z) > f(y) > f(x)$

Kondisi keenam adalah saat jumlah hasil pencarian dokumen yang mengandung $\boldsymbol{z}$ lebih besar dibandingkan yang mengandung $\boldsymbol{x}$, namun hasil pencarian menggunakan kata $\boldsymbol{x}$ lebih kecil dibandingkan dengan hasil pencarian yang menggunakan kata $\boldsymbol{y}$. Hubungan antara ketiga kata tersebut dalam rangkaian dapat menggunakan (6).

$$\boldsymbol{GLSD(x,y,z) = \frac{\log f(z) - 2\log f(x) - \log f(y) + 2\log N}{-\log f(x) + \log N}} \quad (6)$$

Setiap pasangan kata akan dibuatkan semua kemungkinan urutan dan dihitung bobot GLSD dari setiap kemungkinan tersebut. Sehingga untuk setiap 3 kata, maka akan didapatkan 6 macam rangkaian. Untuk memilih rangkaian mana yang akan dijadikan sebagai rekomendasi akhir, maka dipilih rangkaian dengan nilai GLSD tertinggi [4].

3. UJI COBA

Data yang digunakan dalam uji coba ini merupakan dokumen makalah ilmiah yang disimpan dalam bentuk *Portable Document Format* (PDF) pada direktori lokal. Dokumen yang digunakan untuk uji coba berjumlah 130. Semua dokumen tersebut termasuk dalam bidang *Arts and Humanities* pada situs web *www.sciencedirect.com* dan terbagi dalam beberapa jurnal, antara lain *Assesing Writing, Computer and Composition, English for Specific Purposes, Journal of Historical Geography*, dan *Political Geography*.

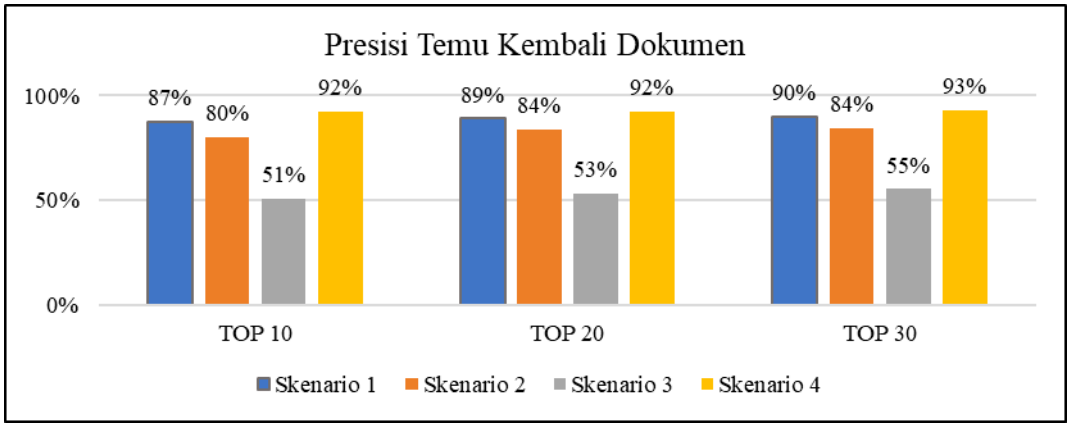
Untuk menguji kemampuan algoritma ini dalam merangkai kata kunci, dibuat empat skenario dalam uji coba ini. Skenario pertama adalah dengan menggunakan bagian abstrak dokumen sebagai teks sumber dalam pembuatan rangkaian kata kunci. Skenario kedua adalah dengan menggunakan bagian selain abstrak dan referensi. Skenario ketiga adalah dengan menggunakan teks dari abstrak namun tanpa melewati proses pembentukan frasa. Skenario

keempat adalah dengan menggunakan kata kunci yang telah disediakan pada bagian abstrak dokumen.

Rangkaian kata kunci yang telah berhasil dibuat kemudian diambil satu perwakilan untuk diukur keberhasilannya dalam proses temu kembali dokumen. Rangkaian kata kunci yang digunakan adalah rangkaian pada setiap dokumen yang memiliki nilai GLSD tertinggi. Keberhasilan mendapatkan dokumen dilihat dari terdapatnya dokumen dimaksud pada hasil pencarian menggunakan mesin pencari secara *online*. Letak kemunculan dokumen dilihat dalam tiga kemungkinan, 10 peringkat tertinggi, 20 peringkat tertinggi, dan 30 peringkat tertinggi. Tingkat keberhasilan sistem dalam membuat rangkaian kata kunci dari dokumen diukur dengan menggunakan presisi dan *recall*.

3.1. Hasil Temu Kembali Dokumen Menggunakan Rangkaian Kata Kunci Hasil GLSD

Kualitas penemuan kembali dokumen menggunakan rangkaian kata kunci yang dibuat oleh perhitungan GLSD terbukti sangat baik. Hal ini dapat dilihat dari hasil perhitungan presisi untuk semua skenario uji coba, yang mendapatkan nilai lebih dari 50%, seperti yang terlihat pada gambar 2.

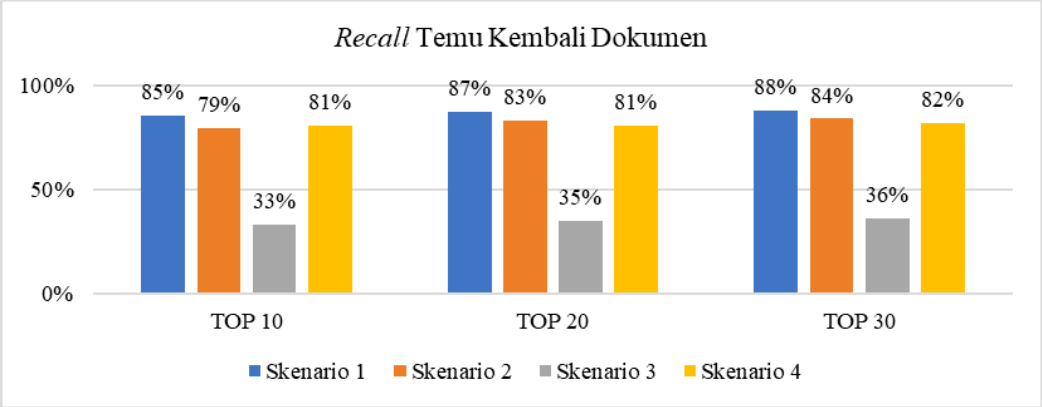


Gambar 2. Perhitungan presisi untuk setiap skenario pada tiap kelompok peringkat.

Dari keempat skenario, hanya skenario 3 yang mendapatkan nilai presisi sekitar 50%. Skenario lainnya mendapatkan nilai lebih dari 80%. Hal ini berarti dari setiap uji coba yang dilakukan, sebagian besar rangkaian kata kunci yang dibuat berhasil menemukan kembali dokumen yang diinginkan dengan tepat. Nilai presisi skenario 4 memang yang paling besar, hal ini dikarenakan kata kunci yang digunakan merupakan kata-kata hasil pilihan penulis artikel itu sendiri. Dimana kata-kata tersebut dipilih yang sedekat mungkin menggambarkan isi dari artikel.

Pembentukan frasa pada skenario 1 dan 2 sangat memengaruhi hasil penemuan kembali dokumen, dibuktikan dengan perbedaan yang jauh antara presisi kedua skenario tersebut dengan skenario 3. Kecilnya nilai presisi pada skenario 3 dikarenakan kata kunci yang terdapat dalam rangkaian hanya terdiri dari satu kata saja. Kata-kata tersebut tidak dapat mewakili isi dokumen secara keseluruhan karena bersifat umum. Dengan adanya pembentukan frasa, keberagaman hasil temu kembali dokumen bisa ditekan sehingga dokumen yang diinginkan dapat dengan mudah ditemukan pada 30 peringkat teratas menggunakan rangkaian kata kunci yang telah dibuat.

Dokumen yang dapat ditemukan oleh rangkaian kata kunci hasil GLSD pun hampir sesuai dengan jumlah dokumen yang digunakan dalam uji coba ini. Hal ini dapat dilihat dari hasil perhitungan *recall* pada gambar 3. Hampir semua skenario berhasil mendapatkan nilai *recall* lebih dari 80% untuk setiap kelompok pemeringkatan. Hanya skenario 3 yang mendapatkan nilai *recall* sekitar 30%.



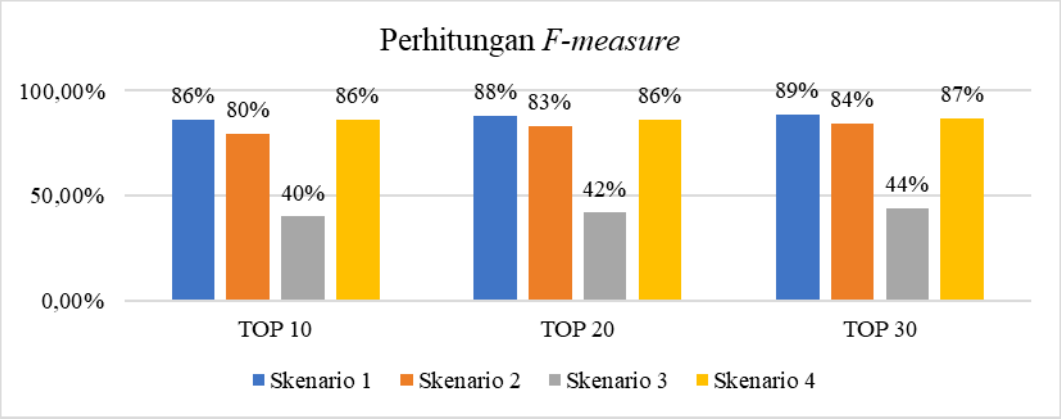
Gambar 3. Perhitungan *Recall* untuk setiap skenario pada tiap kelompok peringkat.

Tingginya nilai *recall* skenario 1 dan 2 menggambarkan bahwa sistem berhasil membuat rangkaian kata kunci yang dapat mewakili isi dokumen sehingga rangkaian tersebut dapat menemukan kembali dokumen tersebut. Pembentukan frasa juga sangat memengaruhi nilai *recall*. Hal ini dapat dilihat dari hasil perhitungan *recall* pada skenario 3 yang hanya sekitar 30%. Kecilnya nilai recall ini dikarenakan sedikitnya satuan kata khusus yang terkandung dalam dokumen yang digunakan dalam uji coba. Sebagian dari dokumen bahkan tidak dapat dibuatkan kata kuncinya karena tidak memiliki kata-kata khusus didalamnya.

Dari hasil perhitungan presisi dan *recall* tersebut, kemudian dihitung pula nilai *f-measure* untuk mengevaluasi kedua nilai tersebut. Nilai hasil perhitungan *f-measure* dapat dilihat pada gambar 4. Skenario 1 dan 2 mendapatkan nilai *f-measure* yang tinggi, bahkan hampir menyamai nilai yang didapatkan pada skenario 4. Tingginya nilai ini menggambarkan bahwa sistem yang dibangun menggunakan metode GLSD ini berhasil merangkaikan kata kunci untuk dokumen ilmiah.

Selain itu, berdasarkan nilai *f-measure*, dimana skenario 1 memiliki nilai tertinggi, dapat dikatakan bahwa teks yang paling baik untuk digunakan sebagai sumber perangkaian kata kunci adalah teks yang berada pada bagian abstrak dokumen. Bagian abstrak dokumen dapat dikatakan sebagai uraian singkat dari isi artikel, sehingga susunan katanya pun dipilih secara khusus sehingga dapat menggambarkan isi artikel secara keseluruhan. Teks pada bagian lainnya dalam artikel tidak terlalu optimal dalam perangkaian kata kunci karena teks pada bagian tersebut mengandung banyak sekali kata dan sebagian besar bukan merupakan kata-kata khusus.

Penggunaan frasa pun menjadi penting karena hal tersebut terbukti memengaruhi hasil penemuan kembali dokumen. Dengan adanya pembentukan frasa, maka kata kunci yang dirangkai menjadi lebih spesifik sehingga dapat mewakili isi dokumen dengan lebih baik. Hal ini terbukti dari nilai *f-measure* skenario 3 yang merupakan paling kecil dibandingkan nilai skenario lainnya.



Gambar 4. Perhitungan *f-measure* untuk setiap skenario pada setiap kelompok peringkat.

4. KESIMPULAN

Setelah melakukan serangkaian uji coba pada sistem yang telah dibuat, didapatkan hasil bahwa jumlah kata hasil ekstraksi dokumen memengaruhi jumlah rangkaian kata kunci yang dapat dibangun dari kata-kata tersebut. Semakin banyak kata yang dapat diekstrak dari suatu dokumen, menyebabkan jumlah rangkaian kata kunci semakin banyak. Hal ini disebabkan dari pilihan kata dan frasa yang digunakan dalam dokumen. Jika dokumen tersebut mengandung banyak kata-kata khusus ataupun frasa tertentu dalam suatu bidang, maka rangkaian kata kunci yang dihasilkan semakin banyak. Selain itu, sistem berhasil membuat rangkaian kata kunci yang dapat mewakili isi dokumen sehingga rangkaian tersebut dapat menemukan kembali dokumen tersebut. Sebagian besar rangkaian kata kunci yang dibangun pun berhasil mendapatkan kembali dokumen dalam pencarian menggunakan mesin pencari.

Teks dari bagian abstrak lebih baik digunakan sebagai teks sumber dibandingkan dengan teks dari keseluruhan isi dokumen selain abstrak dan referensi. Hal ini dikarenakan kata-kata yang berada pada seluruh isi dokumen tidak selalu fokus terhadap isi dokumen tersebut, berbeda dengan kata-kata pada bagian abstrak. Kata-kata bagian abstrak dipilih secara khusus oleh para penulis agar dapat menggambarkan isi dokumen secara keseluruhan dengan singkat dan padat. Keberhasilan rangkaian kata kunci juga dibantu dengan adanya pembentukan frasa dari kata-kata kunci yang didapatkan. Dua buah kata yang terdiri dari kata kerja dan kata sifat akan memberikan arti yang lebih spesifik dibandingkan dengan hanya menggunakan satu kata saja. Kata-kata yang bersifar umum tidak dapat mewakili isi dokumen secara keseluruhan.

Sistem ini hanya menggunakan artikel dalam jurnal ilmiah yang dijadikan data uji coba. Artikel ilmiah biasanya sudah terbagi dalam struktur yang rapi. Hal ini pun memudahkan penulis untuk merangkaikan kata kunci setiap artikel. Untuk pengembangan selanjutnya, akan lebih baik jika dapat membuat sebuah sistem yang dapat merangkaikan kata kunci dari artikel lainnya yang tidak memiliki struktur baku, seperti artikel majalah atau koran, ataupun *blog* dalam internet.

DAFTAR PUSTAKA

[1] D. P. Sari dan A. Purwarianti, "Ekstraksi kata kunci otomatis untuk dokumen bahasa Indonesia, studi kasus: Artikel jurnal ilmiah koleksi PDII LIPI," *BACA: Jurnal Dokumentasi dan Informasi,* vol. 35, no. 2, pp. 139-147, 2014.

[2] J. Borges dan M. Lavene, "Evaluating Variable-length Markov Chain Models for Analysis of User Web Navigation Sessions," dalam *IEEE Transaction on Knowledge and Data Engineering*, 2007.

[3] G. Ercan dan I. Cicekli, "Using lexical chains for keyword extraction," *Information Processing and Management,* vol. 43, no. 6, pp. 1705-1714, 2007.

[4] P. I. Chen, S. J. Lin dan C. Y. Chu, "Using Google Latent Semantic Distance to Extract the Most Relevant Information," *Expert Systems with Applications,* pp. 7349-7358, 2011.

[5] P. I. Chen dan S. J. Lin, "Automatic Keyword using Google Similarity Distance," *Expert Systems with Applications,* pp. 1928-1938, 2010.

[6] The Apache Software Foundation, "Apache PDFBox - Java PDF Library," 2008. [Online]. Available: http://pdfbox.apache.org/index.html. [Diakses 9 Juli 2011].

[7] The Stanford Natural Language Processing Group, "Stanford Log-linear Part-Of-Speech Tagger," [Online]. Available: http://nlp.stanford.edu/software/tagger.shtml. [Diakses 3 March 2011].

[8] F. McCown, "Yahoo's new Search BOSS API," 10 July 2008. [Online]. Available: http://frankmccown.blogspot.com/2008/07/yahoos-new-search-boss-api.html. [Diakses 29 March 2011].