Information Retrieval Menggunakan *Latent Semantic Indexing* Pada *Ebook*

Gaguk Susanto[1], Hari Lugis Purwanto[2]
[1,2] Universitas Kanjuruhan Malang
[1]gaguk.susanto@unikama.ac.id, [2] hari_lugis@unikama.ac.id

ABSTRAK

Ebook merupakan buku elektronik sebagai pengganti buku kertas yang dapat dibuka dalam perangkat elektronik (smartphone, laptop atau PC). Bagi pengajar maupun siswa 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. Ilmu pengetahuan dari ebook tersebut tentunya dapat dimanfaat 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. Sistem information retrieval dibutuhkan untuk memanfaatkan sumber pengetahuan yang ada dalam koleksi ebook tersebut sehingga proses pencarian informasi dapat dilakukan dengan cepat. Proses information retrieval dilakukan dengan menggunakan metode Latent Semantic Indexing. LSI merupakan sebuah metode yang diterapkan pada information retrieval untuk mencari/menemukan informasi berdasarkan makna keseluruhan (conceptual topic atau meaning) dari suatu dokumen bukan hanya makna kata per kata. Oleh karena itu sistem information retrieval menggunakan latent semantic indexing pada ebook sangat diperlukan bagi semua masyarakat yang ingin menjadikan perangkat komputernya menjadi asisten pintar.
Kata Kunci: *Ebook, Information Retrieval, Latent Semantic Indexing*

ABSTRACT

Ebook is an electronic book as a replacement for paper books that can be opened in an electronic device (smartphone, laptop or PC). For the teacher and student must have a collection of ebooks that many in the computer and is more likely to simply stored in the hard drive and rarely opened when the contents of the ebook contains an assortment of science. The science of the ebook can certainly be utilized to transform a computer into an intelligent assistant that is very beneficial for the owner. Intelligent assistant will provide a quick reference to the information required by the owner of the computer. Therefore, the use of the ebook for establishing the source of knowledge requires a media bridge between the owners and the device. Information retrieval systems are needed to take advantage of existing knowledge sources in the ebook collection so that the information search process can be done quickly. The process of information retrieval is done by using the Latent Semantic Indexing. LSI is a method that is applied to information retrieval to search / find information based on the overall meaning (conceptual topic or meaning) of a document not only the meaning of words per word. Therefore the information retrieval system using latent semantic indexing on the ebook is necessary for all people who want to make computer devices into intelligent assistants.
Keywords: *Ebook, Information Retrieval, Latent Semantic Indexing*

1. PENDAHULUAN

Tak jarang para pengajar, guru, ataupun dosen memiliki koleksi *ebook* yang cukup banyak sebagai sumber bahan ajarnya. Namun dengan banyaknya *file ebook* tersebut tak jarang justru hanya menjadi *file* koleksi saja di dalam laptop atau komputer setelah dirasa telah memahami isinya. Suatu saat ketika membutuhkan suatu informasi yang sebenarnya terkandung dalam koleksi *ebook* tersebut sering mengalami kebingungan dimana harus menemukannya karena keterbatasan ingatan manusia. Langkah utama untuk menemukan suatu informasi tersebut adalah dengan membaca kembali *ebook* yang dimiliki. Namun tak jarang *ebook* itu sendiri memiliki jumlah halaman yang sangat banyak sehingga pastinya untuk menemukan suatu permasalahan yang dicari membutuhkan waktu yang cukup lama dan tak sedikit mereka lebih memilih untuk mencari melalui mesin pencarian *online* yang sebenarnya hal tersebut sudah ada dalam koleksi *ebook* mereka tanpa menghabiskan kuota internet dan hal tersebut membuat *ebook* mereka hanya menjadi *file* koleksi di komputer yang tidak pernah lagi dibuka dan dibaca.

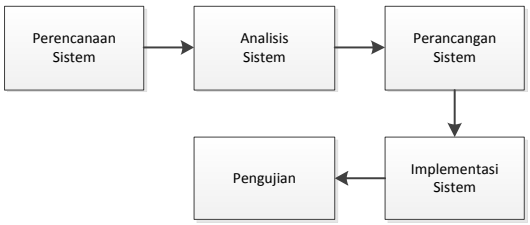
Berangkat dari permasalahan tersebut sebenarnya media komputer baik dalam bentuk personal komputer ataupun laptop yang telah dimiliki para pengajar dapat dijadikan asisten pribadi karena telah dibekali dengan berbagai *file-*

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)[1]. LSI merupakan sebuah metode *automatic indexing* dan *retrieval* dengan memanfaatkan *semantic structure* (struktur asosiasi *terms* dengan dokumen) yang secara implisit terdapat dalam suatu dokumen untuk digunakan dalam pencarian dokumen yang relevan dengan *terms* dalam *query*[2].

2. METODOLOGI PENELITIAN

Tahapan metodelogi penelitian yang dilakukan dalam penelitian ini bisa dilihat pada Gambar 1.



Gambar 1. Tahapan Metodelogi Penelitian

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan tahapan-tahapan sebagai berikut[3]:

Perencanaan Sistem (*Systems Planning*)

Tim peneliti sebanyak 2 orang melakukan rapat konsolidasi untuk merencanakan langkah-langkah penelitian. Sistem yang akan dibuat ditujukan untuk proses pencarian informasi dengan memanfaatkan sumber pengetahuan yang ada dalam koleksi *ebook* dengan menggunakan metode LSI. Bahasa pemograman yang akan digunakan adalah JAVA. Harapan dari sistem yang akan dikembangkan adalah sistem mampu melakukan pencarian pada sekumpulan *ebook* berdasarkan kata kunci yang dimasukkan, dan sistem akan menampilakan daftar *ebook* yang mengandung kata kunci pencarian. Untuk mendapatkan informasi tentang aspek kebutuhan dan kesulitan pengguna, peneliti melakukan penyebaran kuisioner terhadap pengguna *ebook* baik dosen, mahasiswa, guru dan pegawai. Jumlah kuisioner yang disebarkan sebanyak 50 lembar, namun kembali sejumlah 40 lembar. Jumlah pertanyaan pada kuisioner ada 7 pertanyaan. Hasil penilaian para respoder pada masing-masing pertanyaan yaitu: Nilai 76% pada informasi 1 memiliki maksud bahwa para responden memliki jumlah koleksi *ebook* pada kisaran 11-20 *ebook*. Hal ini menandakan bahwa pengguna terbiasa dengan membaca referensi menggunakan *ebook*. Kebutuhan pengguna akan *ebook* memang lebih sedikit dari buku kertas, namun sudah mulai meningkat di periode berikutnya. Adapun Nilai 75% pada informasi 2 menunjukkan bahwa pengalaman pengguna dalam mengelola koleksi *ebook* termasuk pada kategori mudah. Pengguna terbiasa menggunakan aplikasi pembaca *ebook* sekaligus juga mengelola koleksi *ebook* pada aplikasi yang sama. Nilai 70% pada informasi 3 menyatakan bahwa pengguna mudah dalam melakukan pencarian koleksi *ebook* baik melalui fitur pada aplikasi, *explorer* dan cara lain karena sudah pernah membaca *ebook* tersebut. Nilai 65% pada informasi 4 menunjukkan bahwa pencarian *ebook* berdasarkan kata kunci masih dalam kategori tepat, namun sangat dekat dengan ketidaktepatan. Hal ini memiliki maksud bahwa ketepatan tersebut berdasarkan pada meta data yang ada pada *ebook* tersebut. Untuk nilai 71% pada informasi 5 adalah ketepatan pencarian berfokus pada judul, penulis atau isi namun ketika beberapa responden diwawancara lebih lanjut menyatakan bahwa meta data seerpti judul dan nama penulis cukup tepat untuk mendapatkan hasil yang tepat. Pada informasi 7 dengan nilai 78% menunjukkan maksud bahwa fitur pada aplikasi pengelola koleksi *ebook* sangat diharapkan memiliki tambahan baru seperti mampu melakukan pencarian berdasarkan isi. Berdasarkan deskripsi kuisioner pada tahap kelayakan disimpulkan bahwa ada kelayakan pada kebutuhan pengguna terhadap peningkatan fitur pencarian yang didasarkan pada kata kunci yang dibandingkan dengan isi *ebook*. Bahkan, aplikasi *ebook* yang dapat melakuan pencarian berdasarkan isi masih belum ditemukan, sehingga penelitian ini perlu untuk dikembangkan.

Analisis Sistem (*Systems Analysis*)

Analisis Permasalahan

Hasil analisa dari jawaban kuisioner yang telah diisi responden, maka ditemukan adanya permasalahan dalam pencarian informasi yang relevan di sebuah koleksi *ebook* yaitu masih belum adanya fitur pencarian berdasarkan isi *ebook*, sehingga diperlukan fitur yang mampu mencari informasi yang relevan di sekumpulan *ebook*.

Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional ini meliputi analisis kebutuhan sistem dan analisis kebutuhan data. Adapun kebutuhan sistem yang diperlukan antara lain adalah sistem memerlukan koleksi *ebook* yang nantinya akan dipakai sebagai bank data. Kebutuhan data yang diperlukan yaitu *ebook* yang berbahasa indonesia saja, karena sistem masih belum mampu untuk *ebook* berbahasa yang lain.

Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional antara lain letak *file ebook* perlu diketahui supaya mengetahui lokasi direktorinya.

Perancangan Sistem (*Systems Design*)

Adapun perangkat lunak yang digunakan untuk menggambarkan suatu sistem diantaranya sebagai berikut :

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

Implementasi Sistem (*Systems Implementation*)

Tahap berikutnya adalah implementasi yaitu mengimplementasikan rancangan dari tahap-tahap sebelumnya dan melakukan uji coba. Dalam implementasi, dilakukan aktivitas-aktivitas sebagai berikut:

- Pembuatan *database* sesuai skema rancangan. *Database* yang digunakan adalah MySQL.
- Pembuatan aplikasi berdasarkan desain sistem. Bahasa pemrograman yang digunakan adalah JAVA
- Pengujian dan perbaikan aplikasi (*debugging*). Saat pembuatan sistem juga dilakukan pengujian oleh pemrogram untuk mengetahui apakah sistem berjalan sesuai dengan konsep yang diinginkan.

Pengujian

Sistem yang telah dibangun diujicobakan dengan permasalahan nyata guna memperoleh akurasi sistem.

3. HASIL DAN PEMBAHASAN

Deskripsi Sistem

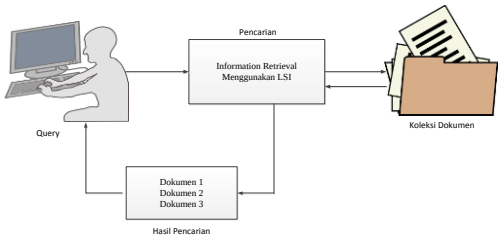
Salah satu aplikasi umum dari IR system adalah *search engine* atau mesin pencarian yang terdapat pada jaringan internet[4]. Pengguna dapat mencari halaman web yang dibutuhkannya melalui *search engine*. Contoh lain dari sistem *information retrieval* adalah sistem informasi perpustakaan.

Sistem *information retrieval* 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 sistem *information retrieval* dengan sistem basis data. Dokumen adalah contoh informasi yang tidak terstruktur. Isi dari suatu dokumen sangat tergantung pada pembuat dokumen tersebut. Sebagai suatu sistem, Sistem *information retrieval* 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.

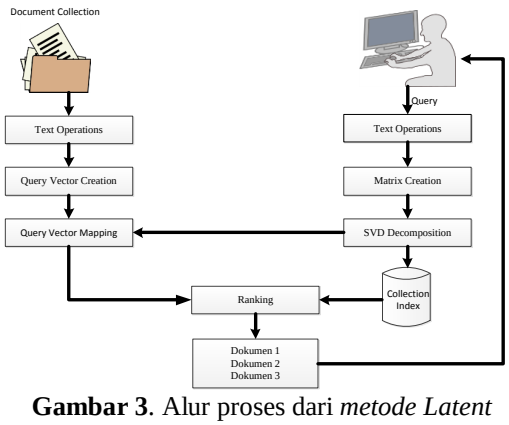
Secara garis besar Sistem *information retrieval* menggunakan metode LSI yang akan dibangun pada penelitian ini bisa dilihat pada Gambar 2.



Gambar 2. Arsitektur Sistem *information retrieval*

Metode *Latent Semantic Indexing* Secara Keseluruhan

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



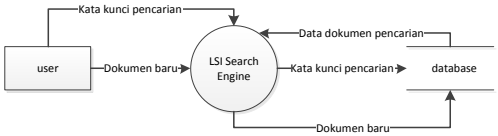
Gambar 3. Alur proses dari *metode Latent Semantic Indexing*

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*. Proses *ranking* dilakukan dengan menghitung relevansi antara vektor *query* terpeta dan *collection index*. Selanjutnya, hasil perhitungan relevansi ditampilkan ke pengguna.

Context Diagram (CD)

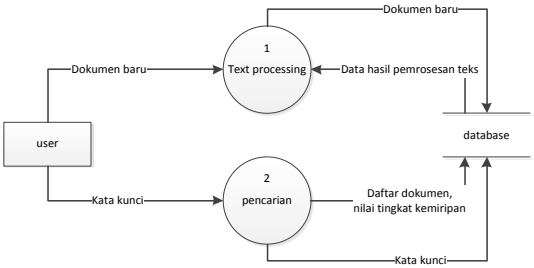
Context Diagram pada Gambar 4 menerangkan bahwa gambaran secara umum yang melibatkan satu entitas yaitu: *user*. Pertama *user* memasukkan sekumpulan *file ebook* ke dalam sistem, dan sistem akan menyimpan data *index ebook* ke *database*. Setelah sistem itu *user* memasukkan kata kunci pencarian ke sistem *information retrieval*. 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* Sistem *information retrieval*

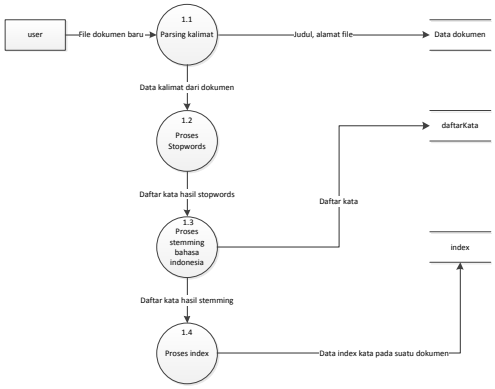
Data Flow Diagram Level 1 (DFD Level-1)

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*. 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 5 memperlihatkan DFD Level 1 Sistem *information retrieval*.



Gambar 5. DFD Level 1 Sistem *information retrieval*

Pada DFD level 1.1 ini *user* memasukkan dokumen baru dan selanjutnya dokumen baru tersebut akan diproses 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 daftar kata, dan dilanjutkan proses indeks, yaitu membangun basis data indeks dari koleksi dokumen, karena dalam pencarian sangat dibutuhkan indeksing. Hasil indexing akan disimpan ke data indeks. Lebih jelasnya DFD level 1.1 bisa dilihat pada Gambar 6.

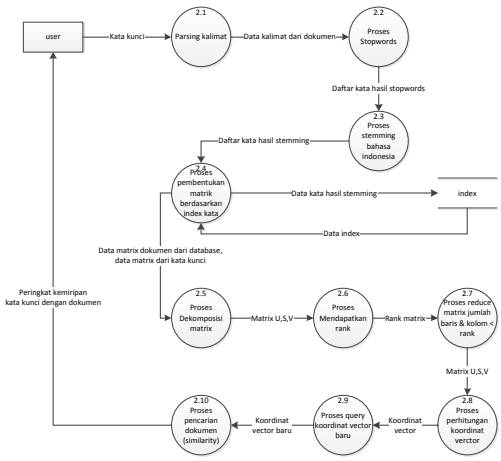


Gambar 6. DFD Level 1.1 Parsing kalimat

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 diproses *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 di-*reduce* jumlah baris kolom, dan mendapatkan *matrix* baru yaitu *matrix* U,S,V. Selanjutnya dilakukan proses perhitungan koordinat *vector*, sehingga didapatkan koordinat *vector*. Koordinat *vector* akan diproses *query* koordinat *vector* baru, dan hasilnya koordinat *vector* 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.

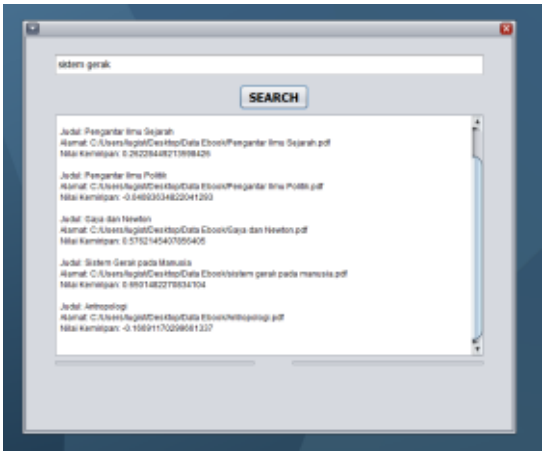


Gambar 7. DFD Level 2.1 Pencarian

Pengujian Sistem

Pengujian sistem sebelum digunakan user langsung

Pengujian ini dilakukan oleh *developer* sistem untuk mengetahui apakah sistem sudah berjalan sesuai dengan *rule* yang diinginkan. Pengujian dengan kata kunci "sistem gerak" dan dicari pada kumpulan *ebook* sebanyak 6 *ebook*. *File ebook* dengan nama sistem gerak pada manusia memiliki nilai kemiripan tertinggi. Hasil pengujian bisa dilihat pada Gambar 8.



Gambar 8. Hasil Pengujian dengan kata kunci "sistem gerak"

Pengujian Sistem ke User

Pada tahap pengujian ini, peneliti melakukan pengujian sistem dengan memberikan kebebasan pada responden untuk menggunakan 6-10 *ebook* yang dimiliki. Kemudian setiap responden akan melakukan pencarian dengan menuliskan kata kunci sebanyak 5 kali proses pencarian. Setelah dilakukan uji coba oleh 40 responden yang terdiri dari 5 dosen, 3 guru dan 32 mahasiswa diperoleh nilai hasil uji coba dengan skala 10. Pengujian sistem ini ditargetkan memiliki nilai lebih dari 7,5. Sehingga untuk mengetahui signifikansi pengujian dari nilai para responden diuji dengan menggunakan *one sample t-test* serta uji normalitas data. Hasil uji menunjukkan bahwa t hitung = 1.300. T tabel diperoleh dengan df = 39, sig 5% (1 *tailed*) = 1.685. Karena – t tabel < dari t hitung (-1.685 < 1.300), artinya nilai pengujian sistem oleh responden paling tinggi 7,5 ditolak, bahkan lebih dari yang diduga yaitu sebesar 7,7044. Hasil uji normalitas data menunjukkan nilai *Kol-Smirnov* sebesar 1.038 dan *Asymp*. Sig tidak signifikan yaitu sebesar 0.232 (> 0.05), sehingga dapat disimpulkan data responden pengujian sistem berdistribusi normal

4. KESIMPULAN DAN SARAN

Beberapa *point* yang dapat disimpulkan dalam penelitian ini adalah:

1. Dari kuisioner pada tahap kelayakan disimpulkan bahwa ada kelayakan pada kebutuhan pengguna terhadap peningkatan fitur pencarian yang didasarkan pada kata kunci yang dibandingkan dengan isi *ebook*.

2. Sistem *information retrieval* menggunakan LSI dapat berjalan sesuai dengan yang diharapkan peneliti, dan bisa melakukan pencarian kata sesuai dengan kata kunci dan menampilkan tingkat kemiripan dokumen *ebook* yang tersimpan di *database*.

Saran untuk perbaikan metode LSI dalam penelitian ini adalah

1. Pengelolaan memori yang lebih baik sehingga banyak dokumen yang diproses dapat lebih banyak.
2. Metode yang dipakai bisa dikombinasikan dengan metode yang lain untuk memperbaiki kecepatan.

5. REFERENSI

[1] A. Alhenshiri, "Web Information Retrieval and Search Engines Techniques," *Al-Satil J.*, pp. 55–92, 2010.

[2] Muhammad and dkk, "Sistem Temu Kembali Informasi Dalam Dokumen Menggunakan Metode Latent Semantic Indexing," *J. Masy. Inform.*, vol. 3, no. 5, 2011.

[3] Y. Bassil, "A Simulation Model for the Spiral Software Development Life Cycle," *Int. J. Eng. Technol. (iJET*, vol. 02, no. 05, 2012.

[4] H. Bunyamin, "Algoritma Umum Pencarian Informasi Dalam Sistem Temu Kembali Informasi Berbasis Metode Vektorisasi Kata dan Dokumen," *J. Inform. UKM*, vol. 2, no. 1, pp. 85–91, 2005.