Text Document Retrieval In English Using Keywords of Indonesian Dictionary Based

Jati Sasongko Wibowo dan Sri Hartati

Abstract— Cross language information regtrieval at the present moment is crucial to the amount of available information use the language variety. Mean while the user who have mastering another language is very less. So with the appointment system of cross language information retrieval is expected.

Information retrieval applications on this system to search for text documents in English using keyword of Indonesian dictionary-based. This system was built by several stages starting from collection of documents data by using crawler, entering documents data into a corpus table, punctuation removal, tokenization of documents, stopword removal and term weighting.

The ability of this system can add documents data, dictionaries data and stopword data. It is enable to retrieval text documents in English that contains keyword of Indonesian with dictionary-based. Results of documents retrieval was ranked based on the calculation of term weights. Able to see the contents of the document and open the original document. The development of this system can be used as a cross language information retrieval system on dictionary based.

Keywords— *retrieval information, dictionary based, cross language.*

I. LATAR BELAKANG

Penyimpanan dokumen secara digital berkembang dengan pesat seiring meningkatnya penggunaan komputer. Kondisi tersebut memunculkan masalah untuk mengakses informasi yang diinginkan. Oleh karena itu, walaupun sebagian besar dokumen digital tersimpan dalam bentuk teks dan berbagai algoritma yang efisien untuk pencarian teks telah dikembangkan, teknik pencarian terhadap seluruh isi dokumen yang tersimpan bukanlah solusi yang tepat mengingat pertumbuhan ukuran data yang tersimpan umumnya temu kembali informasi bertujuan untuk membantu pengguna dalam menemukan informasi yang relevan dengan kebutuhan dalam waktu singkat. Akan tetapi banyak teknik-teknik tersebut yang tergantung pada bahasa yang digunakan dalam dokumen [10].

Menemukan atau memilih dokumen teks yang relevan diantara semua yang tersedia adalah sulit, dengan membiarkan dokumen teks tetap tertulis dalam bahasa asing. Padahal tidak banyak para pencari informasi lancar dalam bahasa asing untuk mampu menemukan dokumen teks asing yang relevan terhadap informasi yang diperlukan. Suatu sistem temu kembali informasi yang ditulis dalam bahasa *query* dengan menggunakan bahasa pribumi akan sangat membantu untuk menemukan informasi yang ditulis dalam bahasa asing. Sistim yang demikian disebut suatu *cross-language information retrieval system* (CLIR). Teknik pencarian informasi lintas bahasa berbasis kamus adalah teknik temu kembali informasi yang menerjemahkan dokumen teks dari satu bahasa ke bahasa lain dengan menggunakan suatu kamus. Ada dua macam cara di dalam pencarian informasi lintas bahasa berbasis kamus. Pertama, menerjemahkan dokumen teks ke dalam bahasa *query*, yang kedua dengan menerjemahkan bahasa *query* ke dalam dokumen teks [2].

Di dalam penelitian ini menggunakan cara yang kedua, yaitu menerjemahkan bahasa *query* ke dalam dokumen teks. Cara ini akan lebih efisien yang tidak menerjemahkan semua dokumen teks, ditambah lagi apabila ada penambahan dokumen baru. Hasil dari pencarian informasi yang didapatkan akan diurutkan berdasarkan dari banyaknya kata yang ditemukan berdasarkan kata kunci yang dimasukkan oleh pencari informasi.

J. Sasongko Wibowo, Fakultas Teknologi Informasi, Unisbank, Semarang.

S. Hartati, Computer Science and Electronics Department, Faculty of Mathematics and Natural Sciences, Gadjah Mada University, Sekip Utara PO BOX BLS.21 Yogyakarta 55281, *e-mail:shartati@ugm.ac.id*

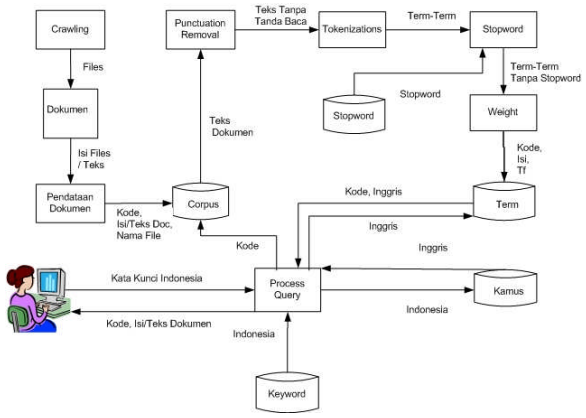
II. CARA PENELITIAN

Perolehan data dokumen dari kumpulan artikel publikasi yang diambil dari satu alamat di internet dengan menggunakan *crawler*. Data artikel ini kemudian disimpan di dalam folder atau direktori sebagai dokumen asli dalam bentuk file. File dokumen berikutnya diproses untuk dimasukkan ke dalam data base berupa nama file, folder atau direktori file dan isi dari dokumen yang disimpan sebagai *corpus*. Dari data base *corpus* berikutnya diproses kembali atau yang disebut dengan tokenisasi. Proses tokenisasi biasanya ada tiga tahapan yang dilakukan, pertama melakukan proses penghapusan seluruh tanda baca yang terdapat dalam dokumen tersebut termasuk karakter-karakter yang tidak terlihat misalnya tab, enter dsb kecuali spasi. Kedua, melakukan proses pemisahan kata-kata dari seluruh kalimat dari data dokumen, pemisahan ini menggunakan spasi sebagai tanda pemisahnya. Ketiga melakukan proses *stopword removal* yaitu penghapusan kata-kata yang biasanya tidak digunakan sebagai kata kunci dalam pencarian dokumen. Pada saat proses tokenisasi dilakukan secara bersamaan juga dilakukan proses *term weighting* atau pembobotan kata yaitu setiap kata dalam dokumen diberikan bobot berdasarkan banyaknya kata pada tiap dokumen. Setelah proses tokenisasi dan pembobotan kata selesai hasilnya berupa kumpulan kata-kata yang berasal dari data dokumen, kode yang menyatakan sumber dokumennya dan bobot tiap-tiap kata yang kemudian disimpan dalam data base *term*.

Perolehan data kamus diperoleh dari aplikasi penerjemah komersial yang nantinya digunakan sebagai basis dari penerjemahan kata kunci pencarian dokumen. Kamus yang digunakan dengan bentuk indonesia-inggris dengan model one to many yaitu setiap kata minimal mempunyai satu arti kata. Dalam hal ini setiap kata bahasa indonesia minimal mempunyai satu arti kata dalam bahasa inggris. Setelah didapatkan data *corpus*, data *term* dan data kamus maka dibuatkan suatu *query* berdasarkan kata kunci pencarian dokumen. Dari kata kunci yang dimasukkan menggunakan bahasa indonesia akan diterjemahkan melalui data kamus. Kata hasil terjemahan yang menggunakan bahasa inggris digunakan untuk melakukan pencarian kata yang terdapat dalam data *term*. Hasil pencarian dari data *term* digunakan untuk melakukan pencarian pada data *corpus* berdasarkan kode dokumen. Apabila sampai data *corpus* pencarian yang dilakukan dapat menghasilkan data yang dicari berarti dokumen yang dicari juga dapat diketemukan. Hasil tersebut akan ditampilkan dan diurutkan berdasarkan pembobotan kata. Dari hasil yang dicapai ini maka dokumen yang dicari dapat ditemukan.

II.I *LOGICAL PROCESS* SISTEM TEMU KEMBALI INFORMASI

Sistem temu kembali informasi dengan berbasis kamus dibangun dari beberapa proses atau modul sehingga membentuk sistem temu kembali informasi. Proses-proses yang terdapat pada sistem temu kembali informasi terdiri dari : pengumpulan data dokumen (*crawling*), proses *input* data dokumen (*corpus*), proses *input* data kamus, proses *input* data *stopword*, proses penerjemahan (translation), proses penghapusan tanda baca (*punctuation removal*), proses penghapusan *stopword* (*stopword removal*), proses tokenisasi (tokenizations), proses pembobotan (*weighting*), proses pengindeksan kata (*term indexing*), proses *query*, dan proses *ranking*. *Logical process* sistem temu kembali informasi secara diagram dapat dilihat dalam gambar 1.

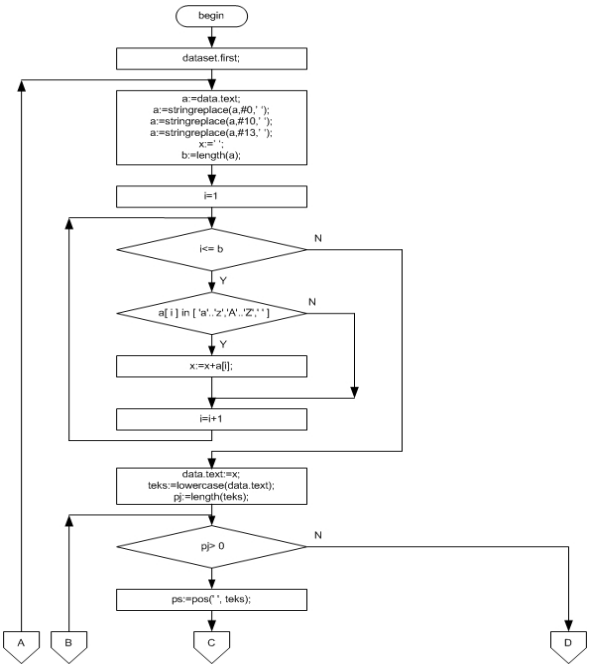


Gambar 1. Diagram *Logical Process* Sistem Temu Kembali Informasi

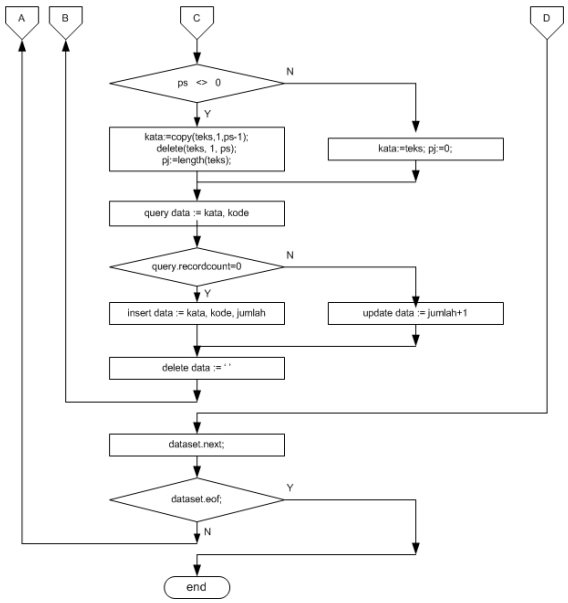
II.II TOKENISASI DOKUMEN

Poses tokenisasi yaitu proses pemisahan kata dari dokumen dengan menggunakan karakter spasi sebagai tanda pemisahnya. Proses ini diawali dari mengambil isi dokumen dari tabel *corpus*, selanjutnya dokumen teks yang ditampilkan dilakukan proses pembacaan seluruh karakter yang terdapat pada dokumen, baik

karakter huruf, angka, tanda baca dan karakter yang tidak terlihat. Pada tiap karakter yang dibaca akan dicocokan dengan kondisi yang telah ditentukan yaitu apabila karakter yang dibaca merupakan huruf kecil atau huruf besar maka akan disimpan dalam sebuah variabel teks, termasuk karakter spasi. Sedangkan karakter selain huruf kecil, huruf besar dan spasi tidak disimpan dalam variabel terks.



Gambar 2. Flowchart Stopword Removal, Tokenisasi dan Pembobotan Dokumen



Gambar 3. Flowchart Stopword Removal, Tokenisasi dan Pembobotan Dokumen

Dari variabel teks yang terbentuk akan didapatkan dokumen teks yang sudah tidak terdapat tanda baca, tetapi masih dimungkinkan untuk terdapat karakter yang tidak terlihat. Sehingga perlu dilakukan proses selanjutnya yaitu menghapus karakter yang tidak terlihat seperti tab, enter, blank dan sebagainya yang diwakili oleh tanda #0, #9, #10 dan sebagainya. Dari proses ini dihasilkan dokumen yang bersih dari tanda baca dan karakter yang tidak terlihat kecuali spasi.

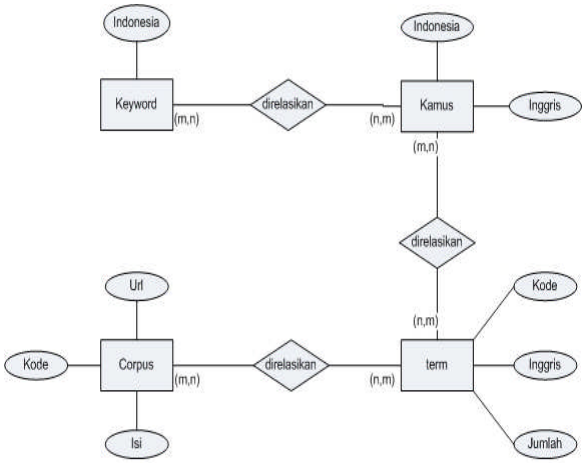
Berikutnya dilakukan proses tokenisasi yaitu pemisahan kata berdasarkan karakter spasi. Pembacaan dokumen teks dibaca per karakter, pembacaan karakter akan berhenti apabila ketemu karakter spasi dan hasilnya berupa sebuah *term*. *Term* hasil tokenisasi tersebut dicek ke daftar *stopword*, apabila hasilnya nihil maka disimpan ke tabel *term*, tetapi apabila hasilnya sebaliknya maka *term* hasil tokenisasi tersebut tidak disimpan dalam tabel. Setelah tersimpan dilanjutkan lagi untuk membaca karakter selajutnya sehingga karakter yang terdapat dalam dokumen tersebut selesai terbaca semua. Hasilnya maka sebuah tabel *term* yang terdiri dari kata-kata yang terdapat dalam dokumen teks dari tabel *corpus*.

Untuk menghindari adanya karakter yang tidak terlihat ikut tersimpan dalam tabel *term* maka dilakukan penghapus karakter yang tidak terlihat dari tabel *term* menggunakan *query* delete spasi, karena karakter yang tidak terlihat apabila tersimpan dalam tabel dianggap sebagai spasi. Pada proses tokenisasi ini juga terdapat proses pembobotan *term* yaitu pemberian bobot nilai pada tiap *term* berdasarkan banyaknya *term* yang terdapat di dalam satu dokumen. Proses pembobotan ini dilakukan pada saat karakter dari proses tokenisasi akan disimpan dalam tabel *term* akan dicocokkan terlebih dahulu di dalam tabel *term*. Apabila tidak ada dalam tabel maka setiap *term* akan disimpan dengan bobot nilai satu, dan apabila ada dalam tabel maka nilai bobot yang sudah ada dalam tabel akan ditambah nilai satu. Proses tokenisasi dalam bentuk flowchart dapat dilihat dalam gambar 2. dan gambar 3.

II.III *ENTITY RELATIONSHIP DIAGRAM*

Dalam pemrosesan data, metode pemodelan data menggunakan ERD (*Entity Relationshyp Diagram*) atau Diagram Hubungan Entitas yang memungkinkan perekayasaan perangkat lunak untuk mengidentifikasi objek data dan hubungannya dengan menggunakan

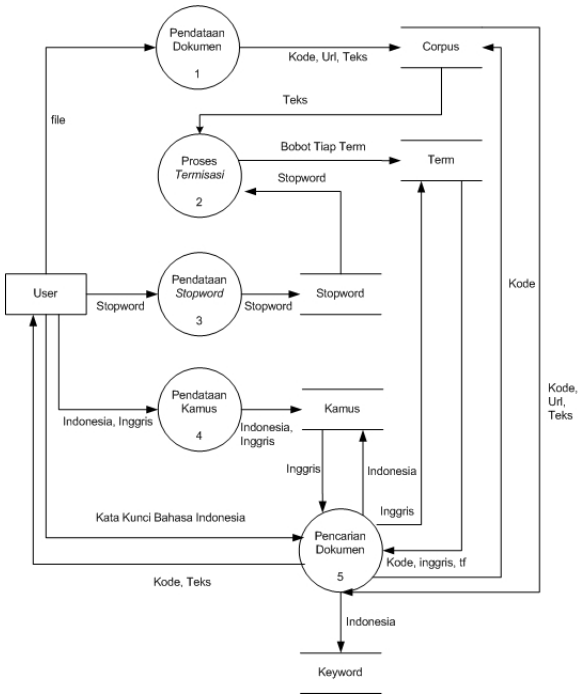
notasi grafis. *Entity Relationshyp Diagram* digunakan untuk memudahkan struktur data dan hubungan antar data, karena hal ini relatif kompleks. Dengan *Entity Relationship Diagram* dapat dilakukan pengujian model dengan mengabaikan proses yang harus dilakukan. Dalam rancangan sistem basis data untuk Sistem Temu Kembali Dokumen Bahasa Inggris, digunakan *Entity Relationshyp Diagram* atau Diagram Hubungan Entitas dan desain tabel untuk menggambarkan atribut-atributnya yang ditunjukkan pada Gambar 4.



Gambar 4. Entity Relationship Diagram Sistem Temu Kembali Dokumen

II.IV DIAGRAM ALIR DATA LEVEL 1

Diagram Alir Data Level 1 pada gambar 5 menggambarkan proses-proses yang terdapat dalam aplikasi pencarian dokumen teks berbasis kamus. Semua proses yang terjadi berhubungan langsung dengan user atau dengan kata lain yang melakukan proses tersebut atas perintah dari user kecuali proses *term weighting* dan *term indexing*. Proses *input* dokumen merupakan proses untuk memasukkan file dokumen ke dalam tabel *corpus*, dimana data yang dimasukkan berupa isi dokumen, path dari dokumen dan kode dokumen.



Gambar 5. Diagram Alir Data Level 1 Sistem Temu Kembali Dokumen

Proses *punctuation removal* merupakan proses untuk menghilangkan tanda baca pada dokumen. Dokumen tersebut sebelumnya diambil dari tabel *corpus*. Hasil dari proses *punctuation removal* berikutnya dilakukan proses tokenisasi, dimana kata-kata yang terdapat dalam dokumen dipisahkan berdasarkan karakter spasi dan hasilnya berupa *term-term*. Tiap *term* tersebut kemudian dilakukan pengecekan terhadap daftar *stopword*, apabila *term* yang dihasilkan terdapat dalam daftar *stopword* maka *term* tersebut tidak dimasukkan dalam tabel, sebaliknya apabila *term* yang dihasilkan tidak terdapat dalam daftar *stopword* maka *term* tersebut disimpan dalam tabel. Pada saat penyimpanan tersebut sekaligus dilakukan proses pembobotan pada tiap-tiap *term* yang tersimpan. Dengan kata lain setiap *term* yang akan disimpan akan dicek dahulu apakah sudah ada dalam tabel apa belum, apabila belum maka *term* tersebut akan diberikan bobot dengan nilai satu, tetapi apabila *term* tersebut sudah ada dalam tabel maka nilai *term* yang telah ada didalam tabel ditambah dengan nilai satu. Setelah proses tokenisasi dan pembobotan selesai berarti telah dilakukan juga proses index terhadap *term-term* tersebut.

III. HASIL DAN PEMBAHASAN

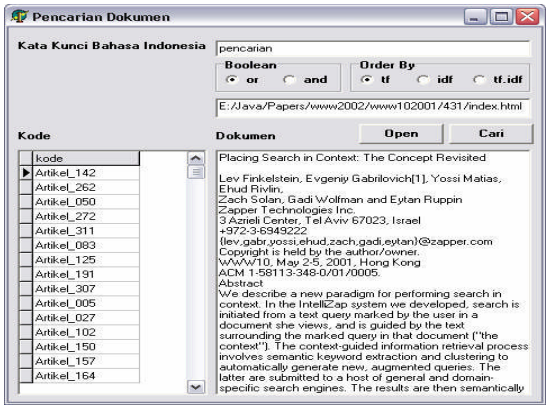
III.I PENCARIAN DOKUMEN

Dalam ujicoba ini menggunakan data 325

dokumen files yang diambilkan dari kumpulan artikel publikasi dari www2000.org yang disimpan dalam tabel *corpus*. Menggunakan data kamus indonesia-inggris dari aplikasi komersial transtool sebanyak 64090 record data yang disimpan dalam tabel kamus. Dari data *corpus* sebanyak 325 record data diproses menggunakan tokenisasi menghasilkan tabel *term* dengan data sebanyak 19391 record data. Tabel keyword dihasilkan dari kata kunci yang dimasukkan sejumlah tiga kata. Kata kunci yang digunakan dalam pengujian ini yaitu kata "sistem", "pencarian" dan "informasi". Relasi tabel yang digunakan yaitu tabel keyword direlasikan dengan tabel kamus, tabel kamus direlasikan dengan tabel *term*, tabel *term* direlasikan dengan tabel *corpus*. gambar 6 menggambarkan *running process* aplikasi pencarian dokumen. Tabel 1 menunjukkan data yang digunakan dalam ujicoba penelitian ini. Dan salah satu hasil uji coba dapat dilihat pada tabel 2.

Tabel 1. Tabel dan Jumlah Data untuk Pengujian Aplikasi

No	Nama Tabel	Jumlah Data
1	*Corpus*	325
2	Kamus	64090
3	*Term*	19391
4	Keyword	3



Gambar 6. Tampilan Proses Pencarian Dokumen

Tabel 2. Hasil Pengujian dari Kombinasi Kata Kunci

No	Kata Kunci	Hasil *Query* Logika Or Tabel : Keyword, Kamus, *Term*, *Corpus*							
1.	Sistem	indonesia	inggris	kode	df	tf	d	idf	tfidf
		sistem	system	Artikel_002	27	6	325	1.081	6.483
		sistem	system	Artikel_228	27	4	325	1.081	4.322
		sistem	system	Artikel_116	27	3	325	1.081	3.242
2.	Pencarian	indonesia	inggris	kode	df	tf	d	idf	tfidf
		pencarian	retrieval	Artikel_142	14	4	325	1.366	5.463
		pencarian	retrieval	Artikel_262	14	4	325	1.366	5.463
		pencarian	retrieval	Artikel_050	14	2	325	1.366	2.732
3.	Informasi	indonesia	inggris	kode	df	tf	d	idf	tfidf
		informasi	information	Artikel_139	27	8	325	1.081	8.644
		informasi	information	Artikel_236	27	8	325	1.081	8.644
		informasi	information	Artikel_142	27	4	325	1.081	4.322
4.	Sistem Pencarian	indonesia	inggris	kode	df	tf	d	idf	tfidf
		sistem	system	Artikel_002	27	6	325	1.081	6.483
		pencarian	retrieval	Artikel_142	14	4	325	1.366	5.463
		pencarian	retrieval	Artikel_262	14	4	325	1.366	5.463
5.	Sistem Informasi	indonesia	inggris	kode	df	tf	d	idf	tfidf
		informasi	information	Artikel_139	27	8	325	1.081	8.644
		informasi	information	Artikel_236	27	8	325	1.081	8.644
		sistem	system	Artikel_002	27	6	325	1.081	6.483
6.	Pencarian Informasi	indonesia	inggris	kode	df	tf	d	idf	tfidf
		informasi	information	Artikel_139	27	8	325	1.081	8.644
		informasi	information	Artikel_236	27	8	325	1.081	8.644
		informasi	information	Artikel_142	27	4	325	1.081	4.322
7.	Sistem Pencarian Informasi	indonesia	inggris	kode	df	tf	d	idf	tfidf
		informasi	information	Artikel_139	27	8	325	1.081	8.644
		informasi	information	Artikel_236	27	8	325	1.081	8.644
		sistem	system	Artikel_002	27	6	325	1.081	6.483

Dari hasil uji coba ini dapat dijelaskan bahwa pencarian lintas bahasa khususnya bahasa Indonesia ke bahasa inggris dapat dilakukan dengan menggunakan kamus sebagai penerjemahnya. Dibuktikan dengan kata kunci yang dimasukkan pada tiap-tiap uji coba dapat diterjemahkan dan hasil terjemahan digunakan untuk melakukan pencarian dan dapat menghasilkan dokumen yang dicari sesuai dengan kata kunci. Keberhasilan pencarian tidak lepas dari kata kunci yang dimasukkan disesuaikan dengan domain dokumen yang terdapat dalam database. Juga kelengkapan kamus terhadap domain dokumen yang ada. Sehingga dapat diketahui apabila ada tiga kata kunci yang digunakan dalam pencarian dokumen dan dapat menghasilkan informasi maka kombinasi dari ketiga kunci tersebut juga dapat dipastikan menghasilkan informasi sesuai dengan kata kunci yang dimasukkan.

Secara bertahap uji coba yang dilakukan dapat dijelaskan sebagai berikut kata kunci yang digunakan yaitu kata “pencarian”, “sistem” dan “informasi” dan kombinasi dari ketiganya. Sebagai salah satu contoh yang dapat dilihat dari hasil uji coba ini kata “sistem” dalam aplikasi ini dapat menerjemahkan kata “sistem” menjadi “*system*”, dan dari kata “*system*” baru dilakukan pencarian dan menghasilkan 27 dokumen, dalam arti terdapat 27 dokumen yang mengandung kata “*system*” dari total jumlah dokumen sebanyak 325 dokumen. Dapat dilihat pula bahwa kata “*system*” dalam dokumen “artikel_002” terdapat 6 buah kata, kata “*system*” dalam “artikel_228” terdapat 4 buah kata dan seterusnya. Apabila hasil pencarian ini diurutkan berdasarkan dokumen yang mempunyai *term* yang berkaitan dengan kata kunci paling banyak maka “artikel_002” akan terletak paling atas.
Contoh yang lain dalam uji coba ini yaitu menggunakan kata kunci “pencarian” yang diterjemahkan menjadi dua kata “*retrieval*” dan “seeking”. Dokumen yang terdapat kata “*retrieval*” berjumlah 14 dokumen, dan dokumen yang terdapat kata “seeking” berjumlah 4 dokumen. Dan terdapat 1 dokumen yang mempunyai kata “*retrieval*” dan “seeking” dalam satu dokumen. Sehingga jumlah dokumen yang berhubungan dengan kata kunci “pencarian” terdapat 17 dokumen.

IV. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan pada sistem maka dapat disimpulkan bahwa :

1. Berdasarkan hasil uji coba aplikasi yang dilakukan dari proses pencarian terhadap 325 dokumen berdasarkan keyword ”*system*”, “pencarian” dan “informasi” yang di*input*kan memberikan hasil yang cukup positif, dalam arti dapat merepresentasikan pencarian dari keseluruhan dokumen yang ada.
2. Aplikasi mampu membantu pengguna dengan tidak melakukan menerjemahkan keyword yang dimasukkankan pada saat melakukan pencarian dokumen yang berbeda bahasa.
3. Aplikasi mampu menampilkan hasil pencarian dokumen dan mengurutkannya berdasarkan perhitungan bobot. Dokumen yang mempunyai bobot yang besar akan diletakkan pada urutan yang teratas diikuti bobot yang lebih kecil demikian seterusnya sehingga dokumen yang mempunyai bobot terkecil akan terletak diurutan paling bawah.
4. Kamus digital yang telah tersedia dapat ditambahkan oleh pengguna dengan memasukkan kata-kata baru sekaligus artinya dengan jumlah sebanyak-banyaknya dengan harapan bahwa hasil dari terjemahan keyword yang dimasukkan menghasilkan arti yang diinginkan dalam pencarian dokumen.
5. Selain kamus digital yang dapat diolah oleh pengguna, *stopword* dalam aplikasi ini dapat juga diolah oleh pengguna dengan menambahkan kata-kata baru yang termasuk dalam kategori *stopword*.
6. Pengguna dapat menambahkan dokumen dengan menggunakan robot/spider/*crawler* dalam aplikasi ini.

V. SARAN

1. Aplikasi ini dalam pencarian dokumen masih berdasarkan pada kata saja, sehingga perlu dikembangkan lagi dengan model pencarian dalam bentuk frasa, klausa ataupun kalimat.
2. Aplikasi ini perlu dikembangkan dan di ujicoba dengan lebih komprehensif dalam berbagai macam bahasa, baik dari penggunaan data kamus, data dokumen dan data *stopword*. Sehingga aplikasi ini dapat digunakan secara universal oleh pengguna dengan berbagai macam bahasa.

VI. DAFTAR PUSTAKA

[1] Adriani, M., and Wahyu, I., 2005, "The Performanceof a Machine Translation-Based English-Indonesian CLIR System", *Cross-Language Evaluation Forum*

[2] Adriani, M. and Croft, W.B., 1997, "The Effectiveness of a Dictionary-Based Technique for Indonesian-English Cross-Language Text Retrieval", *Center for Intelligent Information Retrieval*, Computer Science Department, University of Massachusetts, USA

[3] Asian, J., Williams, H.E. and Tahaghoghi, S.M.M., 2005, "Stemming Indonesian", *Australasian Computer Science Conference*

[4] Attar, R. and A.S. Fraenkel., 1997, "Local Feedback in Full-Text Retrieval Systems", *Journal of the Association for Computing Machinery*, 24: 397- 417

[5] Baeza-Yates, R. and Ribeiro-Neto, B. 1999, *Modern Information Retrieval.* Addison-Wesley

[6] Ballesteros, Lisa and Croft, W.B., 1996, "Dictionary Methods for Cross - Lingual Information Retrieval", *Proceedings of the 7th International DEXA Conference on Database and Expert Systems Applications*, 791-801

[7] Davis, Mark, W. and Ogden, W.C.,1997, "Implementing Cross-Language Text Retrieval Systems for Large-scale Text Collections and the World Wide Web", *AAAI Symposium on Cross-Language Text and Speech Retrieval*

[8] Hull, D.A. and Grefenstette, G., 1996, "Experiments in Multilingual Information Retrieval",*In Proceedings of the 19th Annual International ACM SIGIR Conferenceon Research and Development in Information Retrieval*, 1996.

[9] Manning, C.D., Raghavan P. and Schutze, H., 2008, *Introduction to Information Retrieval*, Cambridge University Press

[10] Ridha, A. 2000, "Pengindeksan Otomatis dengan Istilah Tunggal untuk Dokumen Berbahasa Indonesia", *Skripsi. Jurusan Ilmu Komputer IPB*, Bogor

[11] Salton, G., 1968, "A Comparison Between Manual and Automatic Indexing Methods", *Technical Report No. 68- 11 Department of Computer Science*, Cornell University, Ithaca, N.Y

[12] Salton, G. 1970, "Automatic processing of foreign language documents", *Journal of the American Society for Information Science*

[13] Salton, G. 1989, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley

[14] Sheridan, P. and Ballerini J.P., 1996, "Experiments in Multilingual Information Retrieval using the SPIDE System", *In Proceedings of the 19th ACM SIGIR Conference on Researchand Development in Information Retrieval*