PENINGKATAN KUALITAS METODE *PSEUDO RELEVANCE FEEDBACK* DENGAN PENERAPAN SEGMENTASI DOKUMEN

LAPORAN TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

Oleh :

DIDI KURNIAWAN
10751000157

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2013

PENINGKATAN KUALITAS METODE *PSEUDO RELEVANCE FEEDBACK* DENGAN PENERAPAN SEGMENTASI DOKUMEN

DIDI KURNIAWAN

10751000157

Tanggal Sidang: 13 Juni 2013

Periode Wisuda: November 2013

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Information retrieval system merupakan sistem temu kembali yang berfungsi mencari informasi yang dibutuhkan oleh pengguna dari sekumpulan dokumen yang besar. Pada sistem temu kembali dikenal berbagai model yang dipakai untuk menilai suatu ketepatan pencarian yaitu model Boolean, model ruang vector dan model probabilistik. Model Okapi BM25 adalah salah satu model yang dikembangkan dari model probabilistik. *Relevance feedback* adalah salah satu cara untuk meningkatkan kinerja *information retrieval system* dengan menambahkan *term* atau memperluas inisial *query* diharapkan akan menambah dokumen relevan yang dikembalikan oleh sistem. Pada penelitian ini membahas bagaimana meningkatkan kualitas metode *pseudo relevance feedback* dalam perluasan *query* dengan penerapan segmentasi pada dokumen. Pengujian dilakukan untuk mengetahui apakah ada perubahan dokumen yang ditampilkan sebelum *feedback*(umpan balik) dengan setelah *feedback*. Tahapan pengujian yang dilakukan yaitu dengan mengambil 2,6, dan 8 *term* teratas sebagai *expansion term*. Rata-rata nilai *precision* dan *recall* untuk pencarian sebelum *feedback* adalah 12.6%-96.6%. Rata-rata nilai *precision* dan *recall* untuk pencarian setelah *feedback* tanpa segmentasi adalah 11%-97.6%. Rata-rata nilai *precision* dan *recall* untuk pencarian setelah *feedback* dengan segmentasi adalah 12.6%-96.6%. Hasil dari pengujian tanpa segmentasi memiliki waktu proses yang lama dibandingkan dengan pengujian menggunakan segmentasi dokumen.

Kata Kunci: *Dokumen, Expansion Term, Information Retrieval System, Okapi Bm25, Precision, Pseudo Relevance Feedback, Query, Recall.*

IMPROVEMENT QUALITY PSEUDO RELEVANCE FEEDBACK METHOD WITH APPLICATION SEGMENTATION DOCUMENT

DIDI KURNIAWAN

10751000157

Final Exam Date: Juny, 13 2013

Graduation Ceremony Period: November, 2013

Information Engineering Department

Faculty of Sciences and Technology

State Islamic University of Sultan Syarif Kasim Riau

ABSTRACT

Information retrieval system is system retrieval have function for search information required by user in a large collection documents. In the retrieval system have a some model for evaluate search accuracy is Boolean model, vector space model and probabilistic model. Okapi BM25 model is one of model that developed from probabilistic model. Relevance feedback is one way for improve the performance of information retrieval system by adding term or expanding initial query perhaps increase relevant documents returned by system. This research discussed how to improve quality of pseudo relevance feedback method in query expansion with application segmentation document. The test was do for know whether there are change in the returned document before feedback with after feedback. Step of the test have been done is take top 2, 6, *and* 8 *as expansion term. The average value precision and recall for searching purpose before feedback is* 12.6% - 96.6%. *The average value precision and recall for searching purpose after feedback without segmentation is* 11% - 97.6%. *The average value precision and recall for searching purpose after feedback use segmentation is* 12.6% - 96.6%. *These results of the test without segmentation need long time for process than the test use segmentation document.*

Key words: : *Document, Expansion Term, Information Retrieval System, Okapi Bm25, Precision, Pseudo Relevance Feedback, Query, Recall.*

KATA PENGANTAR

Alhamdulillahi rabbil'alamin, puji syukur kepada Allah S.W.T atas semua nikmat yang telah diberikan. Dengan semua nikmat itu penulis dapat menyelesaikan laporan tugas akhir ini. Salawat dan salam kepada Nabi Muhammad S.A.W karena dengan risau, fikir dan pengorbanan beliau penulis dapat menikmati indahnya Islam.

Laporan tugas akhir ini diajukan sebagai salah satu syarat kelulusan di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim, RIAU. Selama menyelesaikan laporan tugas akhir ini penulis banyak mendapatkan masukan, arahan, dan bimbingan dari semua pihak baik secara langsung maupun tidak langsung. Untuk itu pada kesempatan ini penulis ingin mengucapkan terimakasi kepada :

1. Bapak Prof. DR. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Ibu Dra. Hj. Yenita Morena, M.Si, selaku Dekan Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Ibu DR. Okfalisa, ST, M.Sc, selaku Ketua Jurusan Teknik Informatika, Fakultas Sains dan Teknologi sekaligus sebagai dosen pembimbing akademik penulis.
4. Ibu Fitri Wulandari, S.Si, M.Kom, selaku pembimbing tugas akhir penulis. Terima kasih atas masukan, arahan, motivasi dan kesabarannya sehingga penulis dapat menyelesaikan laporan ini.
5. Bapak Surya Agustian, M.Kom, selaku penguji I yang telah banyak memberikan masukan dan kritikan untuk perbaikkan laporan tugas akhir ini.
6. Ibu Lestari Handayani, S.T, M.Sc, selaku penguji II yang telah banyak memberikan masukan untuk kesempurnaan laporan tugas akhir ini.
7. Kedua orang tua yang telah memberikan dukungan baik moril maupun materil dan yang paling penting do'a beliau.

8. Buat teman-teman yang sering menghabiskan waktu di kosan terima kasih atas bantuan, motifasi, dukungan dan kebersamaanya.
9. Semua teman-teman angkatan 2007 terutama di kelas B, terima kasih atas motivasi, kebersamaan dan kekeluargaannya.
10. Terakhir terima kasih penulis ucapkan kepada Almamater Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau serta semua pihak yang telah membantu dan yang tidak dapat disebutkan satu persatu. Terima kasih atas dukungan dan bantuannya.

Semoga laporan ini dapat bermanfaat bagi penulis khususnya dan pembaca umumnya. Penulis sangat menyadari dalam penyusunan laporan ini masih banyak kekurangannya. Untuk itu penulis harapkan kritik dan sarannya untuk penyusunan yang lebih baik lagi. Akhir kata penulis ucapkan terima kasih.

Pekanbaru, 1 Juli 2013

Penulis

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Latar Belakang

Information retrieval system adalah sistem temu kembali berfungsi mencari informasi yang dibutuhkan oleh pengguna dari sekumpulan dokumen yang besar. Pada proses *information retrieval* (IR) pengguna memasukkan sebuah kata kunci (*keyword*), kemudian kata kunci tersebut akan diterjemahkan oleh IR sebagai suatu *query*, dengan *query* ini sistem akan mencari ke dalam kumpulan dokumen.

Pada sistem temu kembali dikenal berbagai model yang dipakai untuk menilai suatu ketepatan pencarian yaitu model Boolean, model ruang vector dan model probabilistic. Model Boolean adalah model yang pertama kali dikenal dengan menggunakan operator AND, OR dan NOT. Model ini tidak melakukan perangkingan terhadap dokumen yang diambil. Model ruang vector dan model probabilistik melibatkan *term frequency* atau jumlah kemunculan kata dalam perhitungan dan melakukan perangkingan terhadap dokumen yang akan ditampilkan.

Model Okapi BM25 adalah salah satu model yang dikembangkan dari model probabilistik. Model ini sering digunakan dalam penelitian TREC (*Text Retrieval Conference*) yaitu penelitian di bidang IR (*Information Retrieval*) yang dilakukan oleh NIST (*National Institute of Standards and Technology*) dan *Intelligence Advanced Research Projects Activity.* Beberapa penelitian juga dilakukan S. E. Robertson dan S Walker yaitu Okapi/Keenbow at TREC-8 (2000). Dalam penelitiannya mengatakan bahwa Okapi memberikan hasil bagus untuk TREC.

Hasil dari pencarian yang ditampilkan tidak selalu relevan menurut pengguna. Dokumen relevan yang diinginkan pengguna tidak ditampilkan dalam proses pencarian pertama. Salah satu cara untuk memperbaiki hasil dari pencarian pertama adalah dengan *relevance feedback*. *Relevance feedback* adalah salah satu

cara untuk meningkatkan kinerja sistem IR dengan menambahkan *term* atau memperluas inisial *query* diharapkan akan menambah dokumen relevan yang dikembalikan oleh sistem (Cios, dkk. 2007).

Penelitian yang dilakukan Yu, dkk (2003) tentang metode *pseudo relevance feedback* berdasarkan segmentasi dokumen mendapatkan hasil yang baik dalam pengambilan *term* untuk perluasan *query*. Penelitian lain yang dilakukan dalam penelitian Anbiana(2009) memberikan kesimpulan bahwa pada dasarnya suatu dokumen terdiri dari beberapa bagian, sehingga dokumen dapat digantikan oleh segmen yang mewakili bagian tersebut. Untuk mempercepat kinerja sistem dalam pengambilan *term* sebagai *expansion term* bisa dilakukan dengan segmentasi dokumen. Kemudian segmen-segmen tersebut akan mewakili setiap topik pada suatu dokumen.

Berdasarkan penjelasan diatas penulis akan meneliti peningkatan kualitas kinerja metode *pseudo relevance feedback* dalam perluasan *query* dengan menggunakan segmentasi pada dokumen.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, yang menjadi pokok permasalahnya adalah "Bagaimana kualitas kinerja metode *pseudo relevance feedback* dalam *expansion query* dengan penerapan segmentasi pada dokumen?".

1.3 Batasan Masalah

Untuk tidak melebarnya pembahasan perlu adanya batasan. Adapun batasan dalam Tugas Akhir ini adalah sebagai berikut:

1. Data yang dipakai sebagai corpus dalam penelitian ini adalah kumpulan paper SNTIKI 2012 yang berjumlah 75 paper.
2. Algoritma *stemming* yang digunakan adalah algoritma Nazief dan Andriani.
3. Format *file* yang digunakan adalah doc dan docx.
4. *Term* yang dipakai untuk perluasan *query* adalah *term* peringkat 2, 6 dan 8 teratas.

5. Daftar *stopword* yang digunakan dalam penelitian ini berasal dari http://www.scribd.com/doc/61824071/DAFTAR-PUSTAKA.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai oleh penulis dari tugas akhir ini adalah meningkatkan kualitas kinerja dari metode *pseudo relevance feedback* dalam mengekstrak *term* yang akan digunakan untuk perluasan *query* dengan penerapan segmentasi pada dokumen.

1.5 Sistematika Penulisan

Berikut merupakan rencana susunan sistematika penulisan laporan Tugas Akhir yang akan dibuat:

Bab I Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan dari Tugas Akhir yang dibuat.

Bab II Landasan Teori

Bab ini berisikan tentang penelitian-penelitian terdahulu serta dijelaskan juga tentang teori yang meliputi: *information retrieval*, *relevance feedback*, *pseudo relevance feedback*, *stemming*, segmentasi dokumen, *precision* dan *recall*.

Bab III Metodologi Penelitian

Bab ini membahas langkah-langkah yang dilaksanakan dalam proses penelitian, yaitu identifikasi masalah, rumusan masalah, tahap pengumpulan data, analisa, perancangan sistem, implementasi sistem, pengujian, kesimpulan dan saran.

Bab IV Analisa dan Perancangan

Bab ini berisi pembahasan mengenai kebutuhan sistem, yang terdiri dari : analisa sistem IR, perancangan tampilan menu, perancangan antarmuka sistem dan perancangan output.

Bab V Implementasi dan Pengujian

Bab ini berisi tentang langkah-langkah pembuatan *Information Retrieval System* dan tahap-tahap pengujian dari hasil perancangan.

Bab VI Kesimpulan dan Saran

Bagian ini berisi kesimpulan yang dihasilkan dari pembahasan tentang pembuatan *Information Retrieval System* dengan metode *pseudo relevance feedback* berdasarkan segmentasi dokumen, hasil pengujian dan saran.

BAB II
LANDASAN TEORI

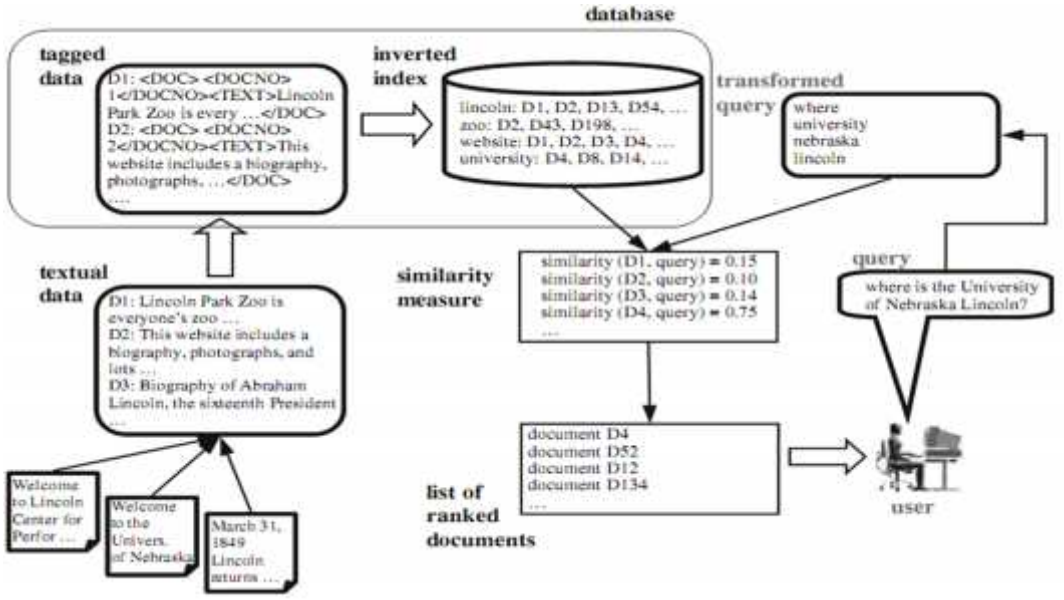
2.1 *Information Retrieval System*

Information retrieval system atau sistem temu kembali adalah proses pencarian data yang tersimpan di dalam kumpulan koleksi untuk mendapatkan informasi yang dibutuhkan (Manning dkk, 2009). Informasi yang tersebut meliputi teks, gambar, *audio*, video dan objek multimedia lainnya. Tujuan dari *information retrieval* (IR) adalah untuk memenuhi kebutuhan informasi pengguna dengan mengambil semua dokumen yang relevan dan dalam waktu yang sama mengambil sedikit mungkin dokumen yang tidak relevan. Sistem IR yang baik memungkinkan pengguna untuk mendapatkan informasi sesuai dengan kebutuhan pengguna. Agar representasi dokumen lebih baik, maka dokumen yang topik dan isinya yang serupa dikelompokan.

2.1.1 Arsitektur *Information Retrieval System*

Arsitektur sistem temu kembali terlihat pada gambar 2.1. secara umum pertama melakukan *preprocessing* terhadap database kemudian implementasi metode untuk mencari kedekatan (*similarity* atau *relevance*) antara dokumen di *database* dengan kebutuhan pengguna yang diinputkan pada sistem temu kembali. Pada tahap *preprocessing* sistem IR mampu menangani dokumen *semi-structured* yang menggunakan *tag* pada bagian dokumennya selain itu sistem akan mengabaikan proses ini untuk dokumen tidak struktur dan meninggalkan *term* tanpa penjelasan.

Setelah pengguna menginputkan *query*, sistem IR akan mengambil istilah-istilah yang penting kemudian sistem akan membandingkan istilah-istilah tersebut dengan istilah yang ada di database. Hasil dari perbandingan tersebut diberi peringkat dengan terurut menurun (*decreasing*) kemudian dikembalikan ke pengguna (Cios, dkk, 2007).



Gambar 2.1 Arsitektur sistem IR (Cios, dkk, 2007)

2.1.2 *Linguistic preprocessing*

Pembuatan *invertex index* dibutuhkan *linguistic preprocessing* (Cios, dkk, 2007), gunanya untuk mengekstrak kata-kata penting pada suatu dokumen kemudian kata-kata tersebut direpresentasikan sebagai *bag of words*. Proses esktraksi kata ini meliputi dua proses utama:

1. Penghapusan *stop words*. Stop words adalah kata-kata atau term yang tidak berhubungan dengan subjek utama dari database walaupun kata tersebut sering muncul dalam suatu dokumen. Kata-kata tersebut termasuk dalam kata penghubung, kata depan dan sejenisnya. Contoh *stop words* adalah dia, kami, dengan, yang, karena, meskipun, walaupun, dan sebagianya.
2. *Stemming*. Kata-kata yang ada dalam suatu dokumen yang memiliki varian morfologik. Oleh karena itu, setiap kata yang bukan *stop words* akan dirubah kebentuk akar dengan menghilangkan awalan dan akhiran. Hasil dari pengakaran kata tersebut diperoleh bentuk yang berbeda tetapi memiliki makna yang sama.

Manning, dkk (2007) menyatakan ada 4 proses dalam pembuatan *invertex index* yaitu:

1. Mengumpulkan koleksi dokumen yang akan di *index*.

2. Melakukan proses *tokenization* terhadap dokumen tersebut. *Tokenization* adalah proses pemisahan paragraph atau kalimat menjadi potongan-potongan kata tunggal. Dalam proses ini juga dilakukan proses penghapusan tanda baca dan mengubah huruf besar menjadi huruf kecil (*lowercase*).
3. *Linguistic preprocessing* untuk mendapatkan *term* atau kata yang telah dinormalisasi. Dalam proses ini ada 2 hal yang dilakukan:
    a. Penyaringan(*filtration*). Tahap ini melakukan penyaringan terhadap kata yang dapat menggambarkan isi dokumen sehingga memberikan perbedaan terhadap dokumen yang lain. Dalam tahap ini *term* yang sering muncul dianggap sebagai *stop words*
    b. *stemming*. Proses merubah *term* kebentuk akar, merubah bentuk tetapi memiliki makna yang sama.
4. Membuat *invertex index* dari *term*/kata-kata tersebut.

2.1.3 Stemming

Stemming adalah proses yang terdapat dalam sistem IR yang melakukan tranformasi kata-kata yang terdapat dalam dokumen menjadi bentuk akar dari kata-kata tersebut dengan aturan tertentu. Sebagai contoh, kata bersama, kebersamaan, menyamai akan distem ke dalam bentuk akarnya yaitu “sama”. Proses stemming pada teks berBahasa Indonesia berbeda dengan proses stemming pada berBahasa Inggris. Pada teks bahasa Inggris hanya penghilang sufiks. Sedangkan pada Bahasa Indonesia selain sufiks(akhiran), prefix, dan konfiks juga dilakukan (Agusta, 2009). Beberapa algoritma stemming yang sering digunakan yaitu:

1. Algoritma *Porter*, merupakan algortima yang sering digunakan dalam teks berbahasa Inggris tetapi ada juga algortima porter untuk teks berbahasa Indonesia. Algoritma ini mempunyai waktu yang cepat dibandingkan dengan algortima Nazief dan Adriani, tetapi memiliki akurasi yang kecil dibandingkan algoritma Nazief dan Adriani.
2. Algortima Nazief dan Adriani merupakan algoritma untuk teks berbahasa Indonesia dengan akurasi yang baik. Algortima ini sangat dibutuhkan dan

menentukan dalam proses sistem temu kembali. Aturan algortima Nazief dan Adriani mengacu pada morfologi bahasa Indonesia yang mengelompokan imbuhan. Imbuhan dalam bahasa Indonesia terdiri dari imbuhan di depan(awalan), imbuhan di belakang(akhiran), imbuhan di tengah(sisipan) dan kombinasi awalan dan akhiran (konfiks).

Langkah-langkah algoritma Nazief dan Adriani sebagai berikut:

1. Mencari kata yang belum di-stemming di dalam kamus, jika kata ditemukan dalam kamus maka kata tersebut adalah kata dasar.
2. *Inflection suffixes* (-lah, -kah, -ku, -mu, dan –nya) dibuang. Jika berupa partikel (-lah, -kah, -tah, atau –pun) maka langkah ini diulang untuk menghapus *possessive pronouns* (-ku, -mu, dan –nya) jika ada.
3. *Derivation suffixes* (-i, -an atau –kan) jika kata ditemukan dikamus maka algoritma berhenti. Jika tidak lanjut kelangkah 3a
    a. Jika –an sudah dihapus, dan kata terakhir adalah –k maka –k juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti, jika tidak lanjut ke langkah 3b.
    b. Akhiran yang dihapus (-i, -an dan –kan) dikembalikan, lanjut ke langkah 4.
4. Menghapus *Derivation Prefix*, jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
    a. Memeriksa ke dalam tabel awalan dan akhiran yang tidak diizinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
    b. For i=1 to 3 tentukan tipe awalan kemudian hapus awalan. Jika kata dasar belum ditemukan, lakukan langkah 5, jika sudah maka berhenti. Catatan jika awalan kedua sama dengan awalan pertama makan berhenti.
5. Melakukan recoding.
6. Jika semua langkah telah dilakukan tetapi tidak berhasil maka kata tersebut diasumsikan sebagai kata dasar. Proses selesai.

2.2 Model Probabilistik

Model probabilistik adalah model yang menggunakan perhitungan probabilistik untuk mendapatkan informasi yang relevan dengan *query* pengguna. Beberapa model yang menggunakan dasar penghitungan probabilistic yaitu *Binary Independence Model*, model Okapi BM25 dan *Bayesian Netwotk* (Manning, dkk, 2009).

Ukuran kemiripan (*similarity*) sebuah dokumen terhadap *query* dalam model dasar probabilistik dihitung dengan menggunakan rumus seperti pada Persamaan 2.1.

$$RSV\ d = \sum_{t \in q} \log \frac{s + 0{,}5\ /\ (S - s + 0{,}5)}{df\ t\ - s + 0{,}5\ /\ (N - df\ t\ - S + s + 0{,}5)} \quad \text{.....(2.1)}$$

Keterangan:

RSV = *retrieval status value* (nilai untuk perangkingan dokumen).

s = jumlah dokumen yang relevan yang mengandung *termt* pada *query q*.

S = jumlah dokumen yang relevan untuk *queryq*.

df_t = jumlah dokumen dalam *corpus* yang mengandung *term t* pada *query q*.

N = jumlah dokumen dalam *corpus*.

2.2.1 Model Okapi BM25

Nilai dari dokumen *d* didapat dari perhitungan bobot *idf* suatu *term* pada *query* yang dimasukkan oleh user menggunakan rumus pada persamaan 2.2.

$$RSV\ d = \sum_{t \in q} \log \frac{N}{df(t)} \quad \text{.....(2.2)}$$

Dengan memfaktorkan masing-masing term dan panjang dokumen maka persamaan 2.2 dapat dikembangkan seperti pada persamaan 2.3.

$$RSV\ d = \sum_{t \in q} \log \frac{N}{df(t)} \cdot \frac{k1+1\ .\ tf(td)}{k1\ \ 1-b\ +b\times\ Ld/Lave\ \ + tf(td)} \quad \text{.....(2.3)}$$

RSV = *retrieval status value* (nilai untuk perangkingan dokumen).

N = jumlah dokumen dalam *corpus*.

df_t = jumlah dokumen dalam *corpus*yang mengandung *termt* pada *query*.

tf_{td} = frekuensi *termt* dalam dokumen *d*.

L_d = panjang dokumen *d*.

L_{ave} = rata-rata panjang dokumen secara keseluruhan.

k_1 = konstanta frekuensi *term*, nilainya berkisar antara $1{,}2 \leq k_1 \leq 2$ (1,2).

b = konstanta panjang dokumen, nilainya berkisar antara $0 \leq b \leq 1$ (0,75).

2.3 *Relevance feedback*

Relevance feedback adalah salah satu cara untuk meningkatkan kinerja sistem IR dengan memodifikasi query pengguna. Dengan penambahan *term* baru pada inisial *query* diharapkan akan menambah dokumen relevan yang dikembalikan oleh sistem. Dalam penentuan *feedback* ada dua cara yaitu dengan manual yaitu *feedback* yang dilakukan oleh pengguna dan automatis yaitu sistem akan mengasumsikan Top *N* dokumen sebagai *feedback* untuk dokumen yang relevan. Ada dua prosedur dalam penentuan *term* baru untuk mendapatkan *query* baru yaitu dengan secara manual dan automatis.

Penentuan *term* untuk digabungkan dalam *query* lama dengan cara manual adalah dari hasil pencarian pertama pengguna akan memberikan *feedback* kepada sistem kemudian dari beberapa dokumen yang dikembalikan ke sistem pengguna akan menentukan *term* baru yang dipilihnya dari dokumen relevan yang di kembalikan kepada sistem. Sedangkan penentuan *term* baru secara automatis adalah sistem akan menentukan Top *N* dokumen dari hasil pencarian awal kemudian sistem akan mengidentifikasi semua *term* yang ada dalam Top *N* dokumen. Untuk *term* baru yang akan digunakan diambil dari kata yang memiliki nilai maksimal dari *tf.idf*. Kemudian *term* tersebut akan dimasukan kedalam query pengguna yang lama (Cios, dkk, 2007).

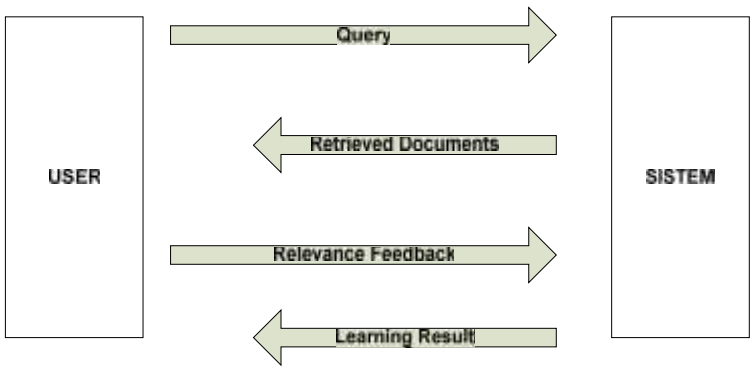
Proses *relevance feedback* ini bisa melalui satu atau lebih iterasi. Proses penelusuran akan mengalami hambatan dalam memformulasi sebuah query yang baik ketika dokumen koleksi kurang baik. Pengubahan *query* awal menjadi *query* baru dalam proses *relevance feedback* akan menggambarkan lebih jelas mengenai kebutuhan informasi yang dibutuhkan oleh pengguna. Dalam penentuan *relevance feedback* oleh pengguna dimaksud untuk mencari dokumen lain selain dokumen yang telah ditemukan.

Ada beberapa metode untuk *relevance feedback*, yaitu metode lokal dan metode global (Adisantoso, 2004).

1. Metode lokal

Ide dari metode lokal adalah memperluas *query* awal berdasarkan informasi yang didapat dari beberapa dokumen urutan teratas yang diambil pertama kali oleh sistem. Metode ini sebenarnya ada dua jenis, *manual-relevance feedback* (umpan balik yang dilakukan oleh pengguna) dan *pseudo-relevance feedback/Automatic relevance feedback* (umpan balik yang dilakukan oleh sistem).

Pada *manual-relevance feedback* sistem menyodorkan beberapa dokumen hasil pencarian pertama. Pengguna memberikan tanda dokumen mana saja yang relevan, kemudian mengembalikannya kembali kepada sistem dari hasil informasi dokumen relevan tersebut, sistem akan memperluas *query* asal dan melakukan pencarian ulang.



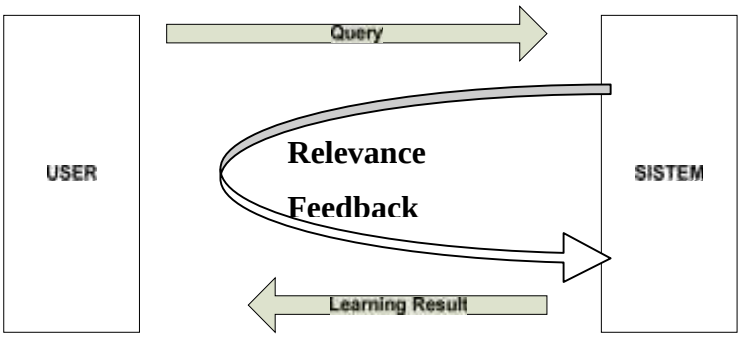
Gambar 2.2 Proses *Manual-Relevance Feedback*

Manual-relevance feedback melakukan 5 buah proses utama, yaitu (Mandala, 2006):

a. Inisialisasi pencarian dokumen.
b. Memberikan hasilnya kepada pengguna.
c. Menerima umpan balik dari pengguna.
d. Membuat *query* baru berdasarkan umpan balik dan melakukan pencarian ulang.
e. Memberikan hasil pencarian ulang kepada pengguna.

Sedangkan pada *pseudo-relevance feedback/Automatic relevance feedback* merupakan cara untuk menggurangi gangguan terhadap pengguna. Dalam cara ini sistem tidak langsung menyodorkan dokumen-dokumen hasil pencarian pertama,

tetapi sistem mengambil beberapa dokumen dengan urutan teratas dari hasil pencarian pertama dan menggunakannya untuk memperluas *query* dengan asumsi bahwa dokumen tadi adalah relevan.



Gambar 2.3 Proses *Pseude-Relevance Feedback*

Proses-proses dalam *pseudo-relevance feedback* adalah sebagai berikut (Mandala, 2006):

a. Inisialisasi pencarian dokumen.
b. N-dokumen pertama yang ditemukan digunakan sebagai umpan balik.
c. Membuat *query* baru dari umpan balik dan melakukan pencarian ulang.
d. Memberikan hasil pencarian kepada pengguna.

2. Metode global

Berbeda dengan metode lokal, sistem dengan metode global melakukan ekspansi *query* terlebih dahulu sebelum dilakukan *retrieval*. Dua alur pemrosesan pada analisis lokal tetap ada, namun diawal sekali terdapat sebuah proses yang akan menghasilkan basis data kata benda yang nantinya akan digunakan untuk ekspansi *query*. Pembuatan basis data kata benda didasarkan pada seringnya kemunculan sebuah kata benda dengan benda lain untuk mendefinisikan sebuah konsep. Semakin sering muncul kata benda dengan sebuah kata benda tertentu maka akan semakin tinggi nilainya.

Disinilah letak perbedaan metode ekspansi analisis lokal dan global. Sedangkan kedua jalur pemrosesannya sama saja. Pada sistem ini ada tambahan masukan, yaitu *file* yang berisi *lexicon* yang akan digunakan untuk membentuk basis data kata benda. *Lexicon* adalah semacam kamus bahasa yang memberikan informasi jenis kata (kata benda, kata sifat, kata keterangan). Proses inilah yang

disebut *noun phrase parsing*. Salah satu metode global yang terkenal adalah *thesaurus*.

Thesaurus menyediakan informasi berdasarkan sinonim dan kata-kata yang saling berhubungan serta frase-frase. *Thesaurus* dapat menambah *recall* tetapi secara signifikan dapat mengurangi *precision*, terutama dengan kata-kata yang ambigu. Keuntungan dari metode ini adalah *robust*, basis data yang tercipta dapat digunakan berulang kali untuk *query* yang berbeda. Sedangkan kerugiannya adalah metode ini memakan tempat (*disk space*) dan perlu waktu cukup lama untuk membangun basis data konsepnya. Secara keseluruhan, metode global tidak sebaik relevance feedback tetapi sama baiknya dengan *pseudo relevance feedback.*

2.3.1 *Pseudo Relevance Feedback*

Pseudo relevance feedback adalah metode untuk memperbaiki hasil dari *information Retrieval system*. Metode ini menggunakan *Top-n* dokumen yang dianggap relevan untuk mengekstrak *term* yang akan digunakan dalam *expansion query*. Hasil dari *expansion query* digunakan kembali dalam *information Retrieval system* untuk mendapatkan dokumen relevan yang tidak muncul pada hasil pencarian pentama. Dalam *pseudo relevance feedback* kualitas dari *expansion query* sangat dipengaruhi oleh dokumen-dokumen peringkat teratas (Anbiana,2009).

Menurut Baeza-Yates dan Ribeiro Neto (Anbiana, 2009), metode ini menganggap sistem yang mengambil *Top-n* dokumen sebagai dokumen relevan lebih baik dari pada pengguna memilih dokumen relevan. Metode ini akan efektif dalam beberapa pengaturan, kemungkinan besar saat *query* asli bersifat panjang dan tepat.

2.3.2 Segmentasi dokumen

Dalam proses ini, dokumen dibagi menjadi beberapa segmen yang masing-masing segmen akan mewakili beberapa topik dari dokumen tersebut. Hasil dari segmentasi dokumen ini akan digunakan dalam pemilihan *term* untuk *expansion query*. Segmen yang akan digunakan untuk *expansion query* akan dilakukan

penyeleksian segmen dengan menggunakan metode okapi BM25. Segmen peringkat teratas *x* yang akan digunakan dalam *expansion query* (Anbiana, 2009).

2.3.3 Penyeleksian *expansion terms*

Dalam penelitian ini dilakukan hal yang mirip dengan metode *pseudo relevance feedback* untuk menyeleksi *expansion terms*. Perbedaannya adalah penyeleksian *term* dilakukan pada segmen peringkat *x* teratas bukan dokumen peringkat *n* teratas. Dalam penelitian Yu, dkk (2003), semua *terms* yang ada pada segmen peringkat *x* kecuali *terms* yang sama dengan *query* awal diberi bobot berdasarkan nilai TSV (*Term Selection Value*) dengan formula sebagai berikut:

$$TSV = w^{(1)} * r/R \quad \text{..... (2.4)}$$

$W^{(1)}$ = *idf* (*invers document frequency*)

r = jumlah segmen yang terseleksi yang mengandung *expansion terms*

R = jumlah segmen terseleksi

Hasil dari pembobotan menggunakan formula diatas akan diambil *terms* yang memiliki TSV *n* tertinggi yang akan dijadikan *expansion query*. Hasil dari *expansion query* ini digunakan untuk mencari dokumen di dalam corpus dengan proses yang sama pada pencarian awal. Hasil dari proses pencarian ini lah yang menjadi hasil akhir *information retrieval system*.

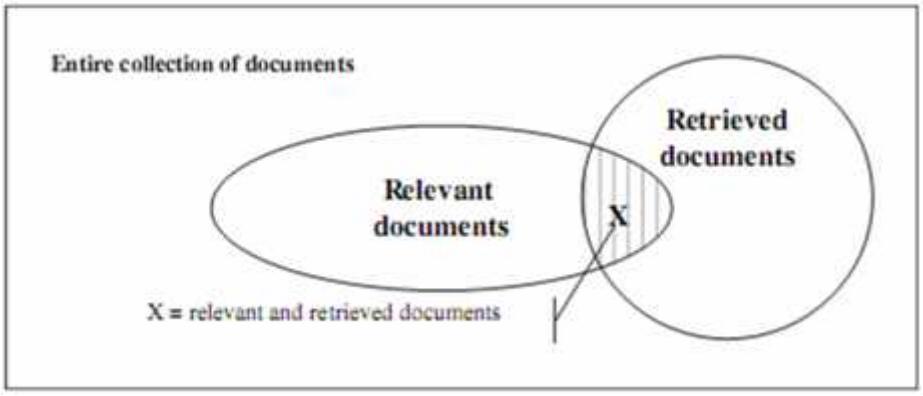
2.4 Kualitas sistem IR

Sistem IR memberikan hasil dari pencarian dari *query* yang dimasuk oleh pengguna. Ada dua kategori dokumen yang dikembalikan oleh sistem IR yaitu *relevant document* (dokumen yang relevan dengan *query*) dan *retrieved document* (dokumen yang diterima pengguna). Ukuran untuk mengunkur kualitas *text retrieval* adalah dengan kombinasi *precision* dan *recall*. *Precision* mengevaluasi kemampuan sistem IR untuk menemukan kembali dokumen *top-ranked* yang paling relevan, dan didefinisikan sebagai persentase dokumen yang di-*retrieve* yang benar-benar relevan terhadap query pengguna (Cios dkk, 2007).

$$precision = \frac{relevant\,docs \cap retrieved\,docs}{retrieved\,docs} \quad \text{..... (2.5)}$$

Recall mengevaluasi kemampuan sistem IR untuk menemukan semua item yang relevan dari dalam koleksi dokumen dan didefinisikan sebagai persentase dokumen yang relevan terhadap *query* pengguna dan yang diterima.

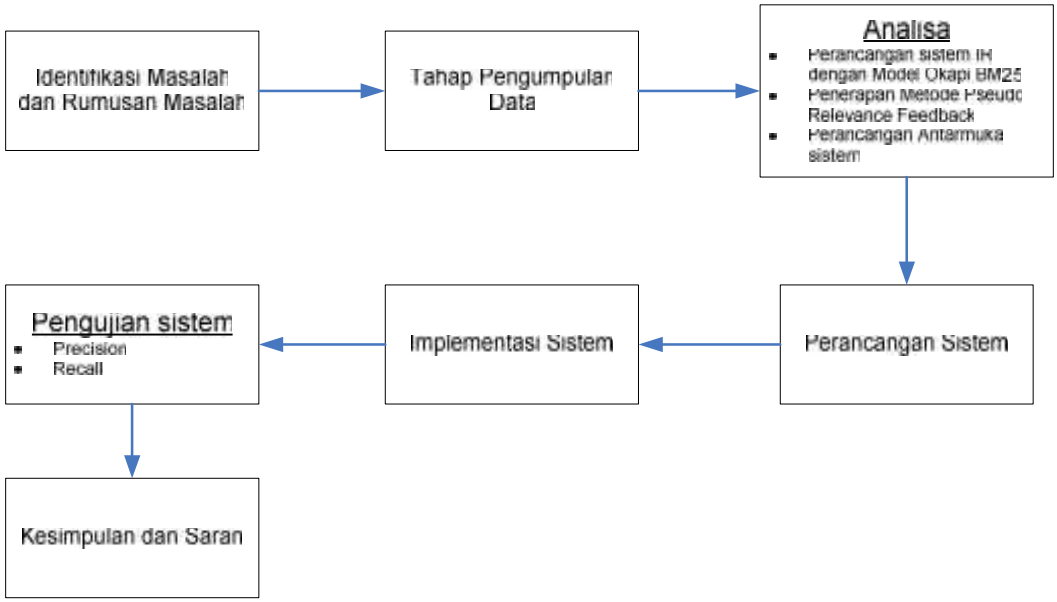
$$recall = \frac{\{relevant\ docs\} \cap \{retrieved\ docs\}}{\{relevant\ docs\}} \quad \text{..... (2.6)}$$



Gambar 2.4 Hubungan antara *relevant documents* dan *retrieved documents*(Sumber: Cios dkk, 2007)

BAB III
METODOLOGI PENELITIAN

Metodologi penelitian menjelaskan langkah-langkah yang akan dilakukan dalam penelitian untuk menjawab perumusan masalah. Dalam penelitian ini adapun langkah-langkah yang akan dilakukan dapat dilihat pada flowchart dibawah ini:



Gambar 3.1 Tahapan Penelitian

3.1 Identifikasi masalah dan rumusan masalah

Identifikasi permasalahan pada penelitian ini adalah adanya dokumen relevan yang kemungkinan tidak ditampilkan pada hasil pencarian berdasarkan query yang diinputkan oleh pengguna. Hal ini yang menjadi permasalah pada penelitian ini bagaimana mengambil dokumen relevan yang tidak ditampilkan pada hasil pencarian pertama.

3.2 Tahap pengumpulan data

Pada tahapan ini akan dijelaskan tetang tahap-tahap pengumpulan data dalam penelitian yang akan dilakukan. Terdapat dua tahap dalam pengumpulan data yang di lakukan :

1. Study Literature. Melakukan pengumpulan informasi melalui jurnal-jurnal ilmiah dan buku-buku yang berhubungan dengan permasalahan pada penelitian tugas akhir ini. Sehingga memperoleh referensi untuk dapat menerapkannya pada tugas akhir ini dan dapat menyelesaikan masalah-masalah saat melakukan penelitian.
2. Pengumpulan dokumen. Kumpulan dokumen yg digunakan adalah kumpulan paper SNTKI 2012.

3.3 Analisa

Pada tahapan ini dijelaskan tentang analisa dalam membangun sistem. Hal-hal yang akan dianalisa adalah :

1. Perancangan sistem IR dengan model Okapi BM25. Dalam perancangan ini tahapan yang dilakukan adalah tahapan pre-processing yang meliputi penghapusan format dan markup pada dokumen, *tokenization*, penghapusan *stop-words*, *indexing* dan pembobotan terhadap *term*. Setelah dilakukan tahapan preprocessing maka dilakukan perangkingan dokumen dengan model Okapi BM25.
2. Penerapan metode Pseudo Relevance Feedback. Hasil dari pencarian menggunakan Okapi BM25 dilakukan *feedback* menggunakan metode *pseudo relevance feedback* dengan penerapan segmentasi pada dokumen untuk mendapatkan hasil yang lebih relevan. Diharapkan dari hasil penerapan ini akan menambah koleksi dokumen yang ditampilkan.

3.4 Perancangan sistem

Tahapan perancangan *information retrieval system* yang akan dilakukan yaitu:

1. Analisa sistem IR untuk pencarian dokumen berdasarkan informasi yang dibutuhkan pengguuna.
2. Perancangan tampilan menu sebagai panduan untuk implementasi.
3. Perancangan antarmuka *information retrieval system* yang baik sehingga mudah digunakan (*user friendly*).
4. Perancangan mesin IR (*IR enginee*) untuk proces pencarian dokumen.

3.5 Implementasi sistem

Pada tahapan implementasi ini akan dilakukan pembuatan modul-modul yang telah dirancang dalam tahap perancangan kedalam bahasa pemrograman. Implementasi sistem akan dilakukan dengan spesifikasi sebagai berikut :

Perangkat Keras

Processor	: Intel Atom N550 1.5 GHz
Memori (RAM)	: 1 GB

Perangkat Lunak

Sistem Operasi	: *Windows 7 Ultimated*
Bahasa Pemrograman	: PHP
Tools Perancang	: XAMPP 1.7.3
Web Browser	: *Chrome*

3.6 Pengujian sistem

Pengujian merupakan tahapan dimana sistem akan diuji untuk mengetahui apakah sistem yang dibangun telah sesuai dengan yang diinginkan dan telah beroperasi dengan baik. Pengujian sistem pencarian tugas akhir/skripsi dilakukan dengan cara mengukur kualitas *text retrieval*. Ukuran yang digunakan untuk mengukur kualitas dari *text retrieval* adalah *precision* dan *recall*. Serta dilakukan penarikan kesimpulan terhadap hasil pengujian sistem.

3.7 Kesimpulan dan saran

Dalam tahapan ini dilakukan penarikan kesimpulan terhadap hasil penelitian yang telah dilakukan untuk mengetahui apakah implementasi sistem yang telah dilakukan dapat beroperasi dengan baik dan sesuai dengan tujuan yang diinginkan, serta memberikan saran-saran untuk pengembangan penelitian selanjutnya agar tercipta suatu rancangan sistem yang sempurna.

BAB IV
ANALISA DAN PERANCANGAN

Bab ini menerangkan bagaimana proses *system information retrieval* dengan model Okapi BM25 dan penerapan metode *Pseudo Relevance feedback* untuk meningkatkan kualitas *system information retrieval*, dan perancangan antarmuka (*interface*).

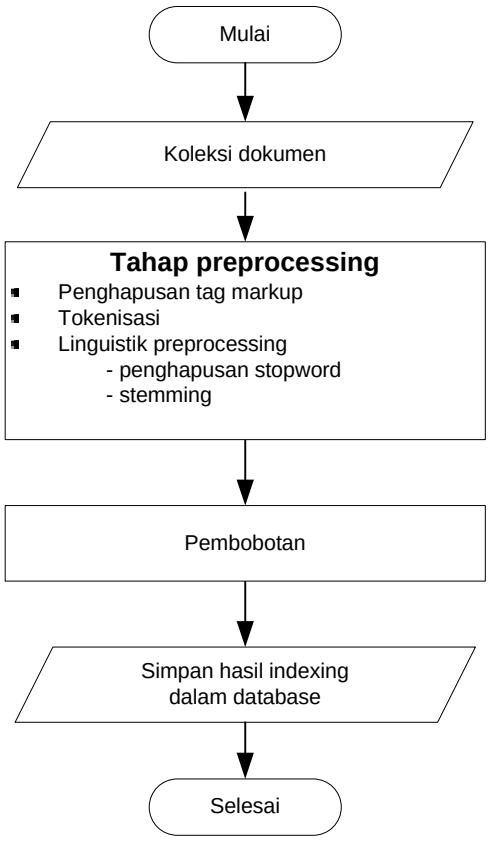
4.1. Analisa perancangan sistem IR dengan model Okapi BM25

Terdapat 2 tahapan utama yang harus dilakukan sebelum melakukan pencarian dokumen dengan model Okapi BM25 yaitu:

1. Tahapan pertama
    a. Koleksi dokumen yang akan di *indexing*.
    b. Penghapusan *tag markup* atau format khusus pada dokumen, tokenisasi, *linguistic preprocessing* (penghapusan *stopwords*).
    c. *Indexing*,
    d. Pembobotan terhadap kata hasil *indexing*.
2. Tahapan kedua
    a. Penginputan *query*.
    b. Tokenisasi, penghapusan *stop-words* pada *query*.

Setelah melewati 2 tahapan utama tersebut maka akan dilakukan pencarian dokumen menggunakan model Okapi BM25 dimana hasil dari pencarian tersebut didapat sejumlah dokumen relevan yang terurut menurun berdasarkan *query* yang diinputkan.

Gambar 4.1 berikut ini menjelaskan tahap-tahap yang dilakukan pada sistem IR dari awal sampai didapatkannya hasil pencarian. Hasil pencarian akan ditampilkan terurut menurun berdasarkan hasil perangkingan menggunakan model Okapi BM25.



Gambar 4.1 Skema alur proses system IR

4.1.1. Tahapan pertama pada *system information retrieval*

Tahapan pertama dari *system information retrieval* terdiri dari koleksi dokumen yang akan di *indexing*, tahap *preprocessing*, *indexing*, pembobotan.

4.1.1.1. Koleksi dokumen

Koleksi dokumen merupakan hal yang penting dari *system information retrieval*. Koleksi dokumen adalah kumpulan teks yang akan dilakukan tahap *preprocessing* untuk mendapatkan *terms* yang untuk digunakan dalam pencarian dokumen menggunakan model Okapi BM25. Dalam penelitian ini koleksi dokumen yang digunakan adalah kumpulan paper SNTIKI(seminar nasional teknologi informasi, komunikasi dan industri) yang berjumlah 75 paper dalam bentuk format *word* dokumen(doc/docx).

4.1.1.2. Tahap *preprocessing*

Sebelum dilakukan pencarian dokumen, koleksi dokumen yang digunakan dalam penelitian ini harus melalui tahap *preprocessing*. Tahap preprocessing adalah proses mengesktrak *terms* yang ada pada dokumen untuk dibuat *inverted index* dan pembobotan. Tahap *preprocessing* ini meliputi penghapusan *tag markup*, tokenisasi, dan *linguistic preprocessing*. Sebagai contoh kasus dalam tahap *preprocessing* ini dokumen yang digunakan adalah dok1.txt, dok2.txt, dok3.txt, dok4.txt, dok5.txt.

1. Penghapusan tag markup.

   Sebelum dilakukan tokenisasi, dokumen tersebut harus dihapus tag markup atau format khusus yang ada pada dokumen. Biasanya dokumen yang memiliki format khusus adalah dokumen dengan ekstensi .htm atau .html, contohnya tag <p>,<body>,<title>. sedangkan dokumen dengan ektensi .doc tidak memiliki format khusus sehingga proses ini bisa tidak dilakukan.

2. Tokenisasi.

   Pada proses ini semua kata akan dipisah berdasarkan spasi dan pada tahap ini juga dilakukan penghapusan tanda baca dan merubah huruf besar menjadi huruf kecil(*lowercase*). Berikut contoh dok1.text untuk proses tokenisasi dan penghapusan tanda baca.

Dok1: content based image retrieval cbir bekerja dengan cara mengukur kemiripan citra dengan semua citra yang ada dalam database sehingga ketepatan pencarian berbanding lurus dengan jumlah citra dalam database pencarian citra yang paling mirip mempunyai tingkat lamanya pencarian dengan melakukan klasifikasi citra yang bertujuan untuk mengurangi waktu pencarianlebih singkat dan akurat pada cbir implementasi shape base thresholding untuk klasifikasi citra serta mengukur tingkat akurasi dan waktu klasifikasinya penelitian ini dirancang aplikasi perangkat lunak dengan pemograman java jdk aplikasi klasifikasi citra cbir yang akan mampu mengekstrak fitur warna dan tekstur dari sebuah citra dengan menggunakan shape base thereshold color histogram dan entropi base histogram. hasil dari proses ekstraksi fitur kemudian digunakan oleh perangkat

lunak dalam proses learning dan klasifikasi dengan metode shape base threshold perangkat lunak dibangun dengan metode analisis dan perancangan terstruktur kemudian diimplementasikan dengan java jdk adapun citra learning yang terdapat pada 8 kelas citra fitur yang di simpan quary database yaitu 596 citra bmp dan jpg dengan ukuran 400x400 sebagai sample pengujian dan masing masing citra yang terdapat pada quary data base yaitu color histogram dan shape base thereshold histogram yang berbeda aplikasi klasifikasi citra cbir yang dihasilkan kemudian diuji dengan parameter tingkat akurasi dan waktu klasifikasi hasil pengujian menunjukkan bahwa kombinasi fitur warna dan tekstur memberikan tingkat akurasi yang lebih tinggi dibandingkan dengan klasifikasi berdasarkan fitur warna saja atau tekstur saja namun membutuhkan waktu klasifikasi yang lebih lama.

3. *Linguistic preprocessing*

   Setelah dilakukan tokenisasi maka tahap selanjutnya yaitu *linguistic preprocessing*. Dalam proses ini ada dua tahap yang dilakukan yaitu penghapusan *stopwords* dan *stemming*.

   a. Penghapusan *stopwords*

      Semua kata tunggal yang telah dipisah pada proses tokenisasi maka dilakukan penghapusan terhadap kata tunggal yang tergolong pada kata sambung, kata tanya, kata sapaan dan kata yang tidak mendeskripsikan isi dari dokumen. Dalam penelitian ini *stopwords* yang digunakan berjumlah 558 kata termasuk stopwords untuk bahasa inggris. Berikut contoh penghapusan *stopwords* untuk dok1.txt

Dok1: content based image retrieval cbir bekerja cara mengukur kemiripan citra semua citra ada database a ketepatan pencarian berbanding lurus jumlah citra database pencarian citra paling mirip mempunyai tingkat lamanya pencarian melakukan klasifikasi citra bertujuan mengurangi waktu pencarian lebih singkat akurat cbir implementasi shape base thresholding klasifikasi citra serta mengukur tingkat akurasi waktu klasifikasinya penelitian dirancang aplikasi perangkat lunak pemograman java jdk aplikasi klasifikasi citra cbir mampu mengekstrak fitur warna tekstur citra menggunakan shape base

thereshold color histogram entropi base histogram hasil proses ekstraksi fitur digunakan perangkat lunak proses learning klasifikasi metode shape base threshold perangkat lunak dibangun metode analisis perancangan terstruktur diimplementasikan java jdk citra learning terdapat 8 kelas citra fitur di simpan quary database 596 citra bmp jpg ukuran 400x400 sebagai sample pengujian masing masing citra terdapat query data base color histogram shape base thereshold histogram berbeda aplikasi klasifikasi citra cbir dihasilkan diuji parameter tingkat akurasi waktu klasifikasi hasil pengujian menunjukkan kombinasi fitur warna tekstur memberikan tingkat akurasi lebih tinggi dibandingkan klasifikasi berdasarkan fitur warna tekstur membutuhkan waktu klasifikasi lebih lama

b. *Stemming*

Algoritma *stemming* yang digunakan dalam penelitian ini adalah algoritma nazief dan andriani. Contoh penghapusan himbuhan “meny” untuk kata”menyapu” menjadi “sapu”. Berikut hasil *stemming* untuk contoh dokumen dok1.txt.

Dok1: content based image retrieval cbir kerja cara ukur mirip citra semua citra ada database tepat cari banding lurus jumlah citra database cari citra paling mirip punya tingkat lama cari laku klasifikasi citra tuju kurang waktu cari lebih singkat akurat cbir implementasi shape base thresholding klasifikasi citra serta ukur tingkat akurasi waktu klasifikasi teliti rancang aplikasi angkat lunak pemograman java jdk aplikasi klasifikasi citra cbir mampu ekstrak fitur warna tekstur citra guna shape base thereshold color histogram entropi base histogram hasil proses ekstraksi fitur guna angkat lunak proses learning klasifikasi metode shape base threshold angkat lunak bangun metode analisis rancang struktur implementasi java jdk citra learning dapat 8 kelas citra fitur simpan query database 596 citra bmp jpg ukuran 400x400 sample uji citra dapat quary data base color histogram shape base thereshold histogram beda aplikasi klasifikasi citra cbir hasil uji parameter tingkat akurasi waktu klasifikasi hasil uji tunjuk kombinasi

fitur warna tekstur beri tingkat akurasi lebih tinggi banding klasifikasi dasar fitur warna tekstur butuh waktu klasifikasi lebih lama

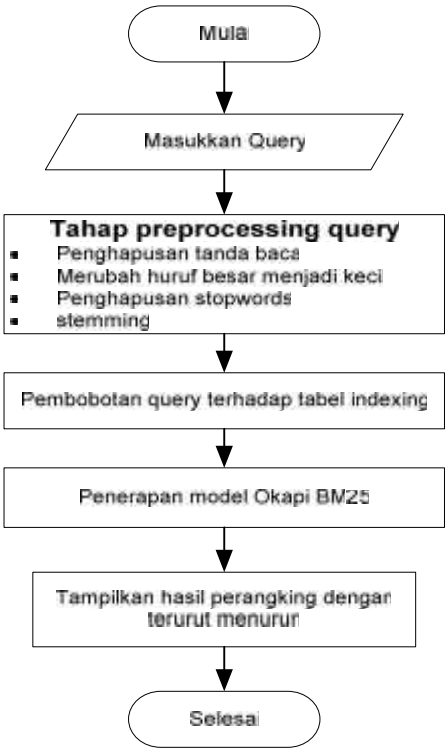
Hasil proses *preprocessing* untuk dok2.txt, dok3.txt, dok4.txt, dok5.txt dapat dilihat pada lampiran A-4.

4.1.1.3. Pembobotan

Dokumen yang telah melalui tahap *preprocessing* akan didapat sejumlah kata tunggal. Kata-kata tunggal tersebut akan dilakukan pembobotan dengan pembobotan Tf/Idf berdasarkan rumus 2.2. hasil pembobotan untuk contoh kasus dapat dilihat pada lampiran A-11.

4.1.2. Penginputan *query* dan pencarian dokumen

Setelah proses pembuatan *indexing* untuk dokumen corpus selesai maka dilakukan pencarian untuk mendapatkan dokumen yang relevan berdasarkan *query* yang diinputkan. Contoh *query* yang diinputkan adalah “Apa hasil dari penelitian sistem pengelolaan zakat? ”. untuk lebih jelas proses pencarian dengan memasukkan sejumlah *query* dapat dilhat pada flowchart berikut ini:



Gambar 4.2 Flowchart untuk tahap pencarian

4.1.2.1. Tahap *preprocessing query*

1. Penghapusan tanda baca

   Apa hasil penelitian sistem pengelolaan zakat

2. Merubah huruf besar menjadi huruf kecil

   apa hasil dari penelitian sistem pengelolaan zakat

3. Penghapusan *stopwords*

   hasil penelitian sistem pengelolaan zakat

4. Penerapan *stemming*

   hasil teliti sistem kelola zakat

5. Setelah tahap *preprocessing* selesai dilaksana pada *query*, selanjutnya dilakukan pembobotan *query* seperti yang dilakukan pada corpus yaitu menggunakan Tf/Idf. Hasil pembobotan *query* berdasarkan tabel A.1 dapat dilihat pada tabel dibawah ini:

Tabel 4.1 hasil pembobotan *query*

No	Kata	tf					df	idf
		dok1	dok2	dok3	dok4	dok5		
1	hasil	3	1	2	3	0	4	0.097
2	teliti	1	6	1	0	2	4	0.097
3	sistem	0	2	0	0	1	2	0.398
4	kelola	0	0	1	0	0	1	0.699
5	zakat	0	0	1	0	0	1	0.699

4.1.3. Penerapaan model Okapi BM25

Selanjutnya penerapan model Okapi BM25 untuk mencari tingkat kerelevanan *query* dengan sejumlah dokumen corpus. Tingkat kerelevanan ini dihitungan berdasarkan rumus 2.3 pada bab 2 dan hasil dari perhitungan ini akan ditampilkan kepada pengguna dengan terurut menurun.

$$\text{RSV d} = \sum_{t \in q} \log \frac{N}{df(t)} \cdot \frac{k1 + 1 \ . \ tf(td)}{k1 \quad 1 - b \ + b \times \ Ld/Lave \quad + \ tf(td)}$$

Panjang dokumen (L_d) untuk contoh kasus:

$|L_{d1}| = 151$

$|L_{d2}| = 90$

$|L_{d3}| = 83$

$|L_{d4}| = 109$

$|L_{d5}| = 96$

K1 = 1.2 dan b = 0.75

Panjang rata-rata (L_{ave}) dokumen corpus adalah

$$L_{ave} = \frac{|Ld1| + |Ld2| + |Ld3| + |Ld4| + |Ld5|}{5}$$

$$L_{ave} = \frac{151+90+83+109+96}{5}$$

$L_{ave} = 105.8$

Perhitungan *Retrieval status value* (RSV) untuk masing-masing dokumen:

$$\text{RSV(dok1)} = 0.097x \frac{1.2+1\ x\,3}{1.2\quad 1-0.75+0.75x\ \frac{151}{105.8}\quad +3} + \quad 0.097\,x \frac{1.2+1\ x\,1}{1.2\quad 1-0.75+0.75x\ \frac{151}{105.8}\quad +1} +$$

$$0.398x \frac{1.2+1\ x\,0}{1.2\quad 1-0.75+0.75x\ \frac{151}{105.8}\quad +0} + 0.699x \frac{1.2+1\ x\,0}{1.2\quad 1-0.75+0.75x\ \frac{151}{105.8}\quad +0} +$$

$$0.699x \frac{1.2+1\ x\,0}{1.2\quad 1-0.75+0.75x\ \frac{151}{105.8}\quad +0}$$

$$= 0.213$$

$$\text{RSV(dok2)} = 0.097\,x \frac{1.2+1\ x\,1}{1.2\quad 1-0.75+0.75x\ \frac{90}{105.8}\quad +1} + \ 0.097\,x \frac{1.2+1\ x\,6}{1.2\quad 1-0.75+0.75x\ \frac{90}{105.8}\quad +6} +$$

$$0.398x \frac{1.2+1\ x\,2}{1.2\quad 1-0.75+0.75x\ \frac{90}{105.8}\quad +2} + 0.699x \frac{1.2+1\ x\,0}{1.2\quad 1-0.75+0.75x\ \frac{90}{105.8}\quad +0} +$$

$$0.699x \frac{1.2+1\ x\,0}{1.2\quad 1-0.75+0.75x\ \frac{90}{105.8}\quad +0}$$

$$= 0.866$$

$$\text{RSV(dok3)} = 0.097\,x \frac{1.2+1\ x\,2}{1.2\quad 1-0.75+0.75x\ \frac{83}{105.8}\quad +2} + \ 0.097\,x \frac{1.2+1\ x\,1}{1.2\quad 1-0.75+0.75x\ \frac{83}{105.8}\quad +1} +$$

$$0.398x \frac{1.2+1\ x\,0}{1.2\quad 1-0.75+0.75x\ \frac{83}{105.8}\quad +0} + 0.699x \frac{1.2+1\ x\,1}{1.2\quad 1-0.75+0.75x\ \frac{83}{105.8}\quad +1} +$$

$$0.699x \frac{1.2+1\ x\,1}{1.2\quad 1-0.75+0.75x\ \frac{83}{105.8}\quad +1}$$

$$= 1.838$$

$$\text{RSV(dok4)} = 0.097\,x\,\frac{1.2+1\;\;x\,3}{1.2\quad 1-0.75+0.75x\;\frac{109}{105.8}\quad +3} + 0.097\,x\,\frac{1.2+1\;\;x\,0}{1.2\quad 1-0.75+0.75x\;\frac{109}{105.8}\quad +0} +$$

$$0.398x\,\frac{1.2+1\;\;x\,0}{1.2\quad 1-0.75+0.75x\;\frac{109}{105.8}\quad +0} + 0.699x\,\frac{1.2+1\;\;x\,0}{1.2\quad 1-0.75+0.75x\;\frac{109}{105.8}\quad +0} +$$

$$0.699x\,\frac{1.2+1\;\;x\,0}{1.2\quad 1-0.75+0.75x\;\frac{109}{105.8}\quad +0}$$

$$= 0.151$$

$$\text{RSV(dok5)} = 0.602\,x\,\frac{1.2+1\;\;x\,0}{1.2\quad 1-0.75+0.75x\;\frac{96}{105.8}\quad +0} + 0.097\,x\,\frac{1.2+1\;\;x\,2}{1.2\quad 1-0.75+0.75x\;\frac{96}{105.8}\quad +2} +$$

$$0.398x\,\frac{1.2+1\;\;x\,1}{1.2\quad 1-0.75+0.75x\;\frac{96}{105.8}\quad +1} + 0.699x\,\frac{1.2+1\;\;x\,0}{1.2\quad 1-0.75+0.75x\;\frac{96}{105.8}\quad +0} +$$

$$0.699x\,\frac{1.2+1\;\;x\,0}{1.2\quad 1-0.75+0.75x\;\frac{96}{105.8}\quad +0}$$

$$= 0.557$$

Berdasarkan perhitungan diatas maka dokumen yang akan ditampilkan kepada pengguna dengan terurut menurun adalah dok3, dok2, dok5, dok1, dok4.

4.2. Analisa penerapan metode Pseudo Relevance Feedback

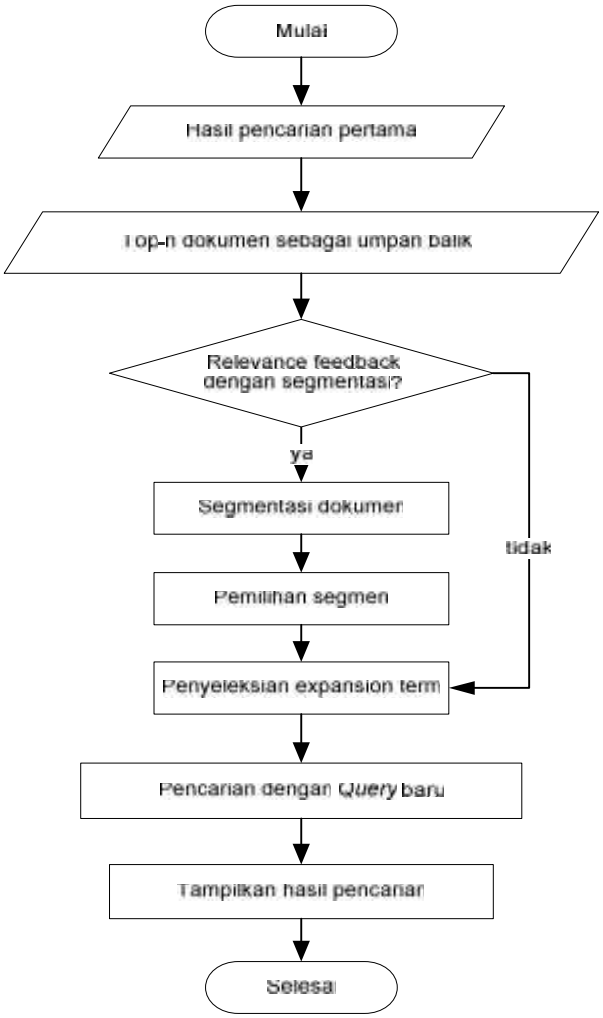
Hasil dari pencarian dokumen pertama yang dilakukan menggunakan model Okapi BM25 didapat beberapa dokumen yang terambil dari corpus. Langkah-langkah untuk memperluas *query* awal menggunakan metode *pseudo relevance feedback* adalah sebagai berikut:

1. Inisialisasi pencarian awal dokumen
2. Top-*n* dokumen dari hasil pencarian awal digunakan sebagai umpan balik(*feedback*).
3. Membuat *query* baru menggunakan dokumen dari umpan balik dan melakukan pencarian ulang menggunakan *query* baru yang sudah diperluas.
4. Menampilkan hasil dari pencarian tersebut.

4.2.1. Inisialisasi pencarian awal dokumen

Pencarian awal dokumen telah dilakukan menggunakan model Okapi BM25 dan didapat dokumen-dokumen relevan berdasarkan *query* yang diinputkan pengguna. Pada contoh kasus diatas yang telah dilakukan pencarian dokumen menggunakan model Okapi BM25 didapat dok3, dok2, dok5, dok1, dok4 yang

dikembalikan oleh sistem ke pengguna. Dokumen inilah yang akan digunakan sebagai umpan balik untuk mendapatkan *query* baru. Untuk lebih jelas tahapan *relevance feedback* dapat dlihat pada *flowchart* dibawah ini:



Gambar 4.3 Flowchart untuk proses *relevance feedback*

4.2.2. Top-*n* dokumen sebagai *feedback*(umpan balik)

Top-*n* yang digunakan untuk umpan balik dalam penelitian ini adalah 5 dokumen teratas. Pada contoh kasus diatas top-*n* yang digunakan adalah 2 dokumen teratas yaitu **dok3 dan dok2**. Dokumen tersebut akan diproses untuk mengekstrak *term* yang akan digunakan untuk *expansion query*.

4.2.3. *Relevance feedback* tanpa segmentasi

Pada tahap ini dokumen *feedback* yang diambil oleh sistem tidak dilakukan segmentasi. Dokumen yang diambil oleh sistem akan langsung

dilakukan penyeleksian term sebagai *expansion term* dengan rumus 2.4. Hasil dari rumus tersebut akan diurutkan berdasarkan nilai *Term Selection Value*(TSV) terbesar.

4.2.4. *Relevance feedback* dengan segmentasi

Adapun langkah-langkah yang harus dilalui untuk mendapatkan *expansion terms* adalah sebagai berikut:

1. Segmentasi dokumen
2. Pemilihan segmen
3. Penyeleksian *expansion terms*

1. Segmentasi dokumen

Segmentasi dokumen dilakukan untuk membagi dokumen menjadi beberapa segmen dan juga untuk mempercepat kinerja sistem dalam pemilihan *expansion terms*. Dari contoh kasus diatas dok2 adalah dokumen yang digunakan sebagai umpan balik. Dokumen tersebut akan dibagi menjadi beberapa segmen dengan aturan setiap segmen terdiri dari 200 kata, tetapi khusus untuk contoh kasus setiap segmen terdiri dari 20 kata. Sebelum dilakukan segmentasi terhadap dokumen yang diumpan balik diatas, terlebih dahulu dilakukan tahap preprocessing, dan stemming.

Contoh kasus

Dok3 : Salah satu program lembaga pengelolaan zakat adalah menyalurkan bantuan pendidikan gratis (Sekolah Juara) untuk calon siswa Sekolah Dasar dan Sekolah Menengah Pertama. Permasalahan dalam penyaluran bantuan pendidikan gratis ini, banyak kriteria yang harus diperhitungkan serta penentuan calon penerima dari golongan tidak mampu (sangat dhuafa dan dhuafa). Penelitian ini menggunakan Principal *Component Analysis* untuk mengelompokan calon penerima dalam tiga kelompok yaitu sangat dhuafa, dhuafa dan bukan dhuafa. Selanjutnya kelompok sangat dhuafa dan dhuafa diurutkan untuk mendapatkan urutan penerima menggunakan *Fuzzy Analitycal Hierarchical* Process. Terdapat 7 kriteria pengelompokkan dhuafa dan 4 kriteria digunakan untuk perangkingan

sebagai hasil keputusan. Hasil perhitungan dapat disimpulkan bahwa penerapan metode *Principal Component Analysis* dan *Fuzzy Analitycal Hierarchical Process* dapat menyelesaikan permasalahan multicriteria, seperti pada kasus pengelompokkan dan penentuan bantuan pendidikan gratis Sekolah Juara.

Dok2 : HCI merupakan studi perencanaan dan perancangan sistem komputer yang bertujuan membantu aktivitas manusia secara produktif dan aman ketika berinteraksi (Preece, 1994). Fokus studi HCI pada penelitian ini adalah pada aspek interaksi manusia dan komputer yang identik dengan istilah pendayagunaan (*usability*). *Usability* ditujukan agar sistem komputer yang dibuat tersebut mudah digunakan dan dipelajari baik secara individu ataupun kelompok. Penelitian ini dilakukan untuk menganalisis dan mengevaluasi *usability* dari *video game*. Terdapat sepuluh prinsip *usability* yang dibuat oleh Nielsen (1993), tetapi dalam penelitian ini hanya delapan prinsip yang relevan untuk dijadikan variabel dalam mengevaluasi *usability* dari *video game*. Penelitian ini menggunakan kuisioner sebagai instrument penelitian, dengan skala *Likert* sebagai skala pengukuran. Hasil penelitian menunjukan *video game* telah memenuhi 7 *variabel* sehingga dapat dinyatakan *video game* sudah bersifat *usable*. Guna meningkatkan *usability* dari *video game*, direkomendasikan untuk melakukan perbaikan terhadap aspek "*recognition rather than recall*" demi terwujudnya kemudahan, kepuasan dan kegunaan dari *video game*.

Hasil dari preprocessing dan stemming

Dok3 : salah program lembaga kelola zakat salur bantu didik gratis sekolah juara calon siswa sekolah dasar sekolah tengah masalah salur bantu didik gratis kriteria hitung serta tentu calon terima golong dhuafa dhuafa teliti principal component analysis kelompok calon terima dhuafa dhuafa dhuafa dhuafa dhuafa urut dapat urut terima fuzzy analitycal hierarchical process 7 kriteria kelompok dhuafa 4 kriteria rangking hasil putus hasil hitung simpul terap metode principal component analysis fuzzy analitycal

hierarchical process selesai masalah multicriteria kasus kelompok tentu bantu didik gratis sekolah juara

Dok2 : hci studi rencana rancang sistem komputer tuju aktivitas manusia produktif aman interaksi preece 1994 fokus studi hci teliti aspek interaksi manusia komputer identik istilah dayaguna usability usability tuju sistem komputer mudah ajar individu teliti analisis evaluasi usability video game prinsip usability nielsen 1993 teliti prinsip relevan variabel evaluasi usability video game teliti kuisioner instrument teliti skala likert skala ukur hasil teliti video game 7 variabel video game sifat usable guna tingkat usability video game rekomendasi baik aspek recognition rather than recall demi wujud mudah puas guna video game

Hasil dari tahap *preprocessing* dan *stemming* yang telah dilakukan diatas akan dibentuk segmen-segmen yang terdiri dari 20 kata. Setiap 20 kata tersebut mewakili setiap segmen. Berikut hasil dari segmentasi dokumen :

Tabel 4.2 hasil segmentasi dokumen

dok	**segmen**	**kata**
3	1	salah program lembaga kelola zakat salur bantu didik gratis sekolah juara calon siswa sekolah dasar sekolah tengah masalah salur bantu
	2	didik gratis kriteria hitung serta tentu calon terima golong dhuafa dhuafa teliti principal component analysis kelompok calon terima dhuafa dhuafa
	3	dhuafa dhuafa dhuafa urut dapat urut terima fuzzy analitycal hierarchical process 7 kriteria kelompok dhuafa 4 kriteria rangking hasil putus
	4	hasil hitung simpul terap metode principal component analysis fuzzy analitycal hierarchical process selesai masalah multicriteria kasus kelompok tentu bantu didik
	5	gratis sekolah juara
2	1	hci studi rencana rancang sistem komputer tuju aktivitas manusia produktif aman interaksi preece 1994 fokus studi hci teliti aspek interaksi
	2	manusia komputer identik istilah dayaguna usability usability tuju sistem komputer mudah ajar individu teliti analisis evaluasi usability video game prinsip

	3	usability nielsen 1993 teliti prinsip relevan variabel evaluasi usability video game teliti kuisioner instrument teliti skala likert skala ukur hasil
	4	teliti video game 7 variabel video game sifat usable guna tingkat usability video game rekomendasi baik aspek recognition rather than
	5	recall demi wujud mudah puas guna video game

2. Pemilihan segmen

Pemilihan segmen dilakukan dengan cara pembobotan Tf/Idf kemudian dilakukan perangkingan dengan model Okapi BM25. Segmen yang memiliki nilai yang tinggi terhadap *query* "Apa hasil dari penelitian sistem pengelolaan zakat?", segmen tersebut adalah segmen yang terpilih. Sebelum dilakukan pembobotan terhadap segmen-segmen tersebut harus dilakukan tahap *preprocessing* dan *stemming*. Hasil dari proses *indexing* segmentasi dokumen dapat dilihat pada lampiran tabel A.2.

Setelah proses *indexing* selesai, maka dilakukan pemilihan segmen menggunakan model Okapi BM25 untuk mendapatkan segmen yang akan digunakan pada tahap selanjutnya.

Table 4.3 pembobotan *query* terhadap *indexing* segmen

No	Kata	dok										df	idf
		3					2						
		Segmen											
		1	2	3	4	5	1	2	3	4	5		
1	hasil	0	0	1	1	0	0	0	1	0	0	3	0.523
2	teliti	0	1	0	0	0	1	1	3	1	0	5	0.301
3	sistem	0	0	0	0	0	1	1	0	0	0	2	0.699
4	kelola	1	0	0	0	0	0	0	0	0	0	1	1.000
5	zakat	1	0	0	0	0	0	0	0	0	0	1	1.000

Hasil dari perhitungan untuk mendapatkan segmen terpilih dapat dilihat pada lampiran hal A-17. Hasil perangkingan segmen menggunakan model Okapi BM25 adalah dok3segmen1, dok2segmen1, dok2segmen2, dok2segmen3, dok3segmen3, dok3segmen4, dok3segmen2, dan dok2segmen4. Jadi segmen yang terpilih adalah **dok3segmen1** karena segmen tersebut berada pada tingkat pertama hasil perangkingan dengan model Okapi BM25.

3. Penyeleksian *expansion terms*

Penyeleksian *expansion terms* dilakukan berdasarkan rumus 2.4. Semua *terms* kecuali *term* dari *query* awal diberikan bobot sebagai berikut:

Table 4.4 Hasil penyeleksian *expansion terms*

no	kata	tf	idf	TSV
1	salur	1	1	0.1
2	program	1	1	0.1
3	lembaga	1	1	0.1
4	salah	2	1	0.1
5	siswa	1	1	0.1
6	dasar	1	1	0.1
7	tengah	1	1	0.1
8	bantu	2	0.698	0.0698
9	sekolah	3	0.698	0.0698
10	juara	1	0.698	0.0698
11	calon	1	0.698	0.0698
12	masalah	1	0.698	0.0698
13	didik	1	0.522	0.0522
14	gratis	1	0.522	0.0522

Hasil dari pembobotan pada tabel 4.4 diatas telah diurutkan berdasarkan nilai TSV dari yang terbesar sampai terkecil, maka *term* yang diambil sebagai *expansion query* adalah *term* yang berada diperingkat 5 teratas yaitu " salur program lembaga salah siswa".

4.2.5. Pencarian dengan *Query* baru

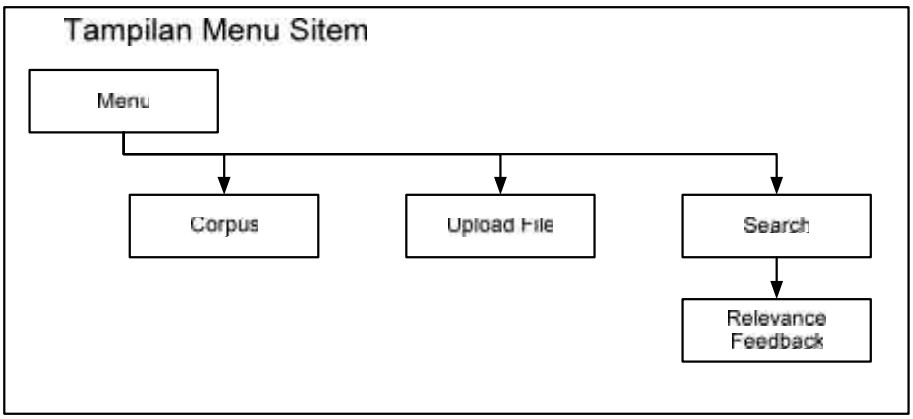
Pencarian dilakukan untuk mencari kembali menggunakan *query* baru hasil *expansion query* yaitu dengan menggabungkan *expansion term* yang telah terpilih dengan *query* asli. Diharapkan hasil dari pencarian dengan *query* baru ini akan didapatkan sejumlah dokumen relevan yang pada pencarian awal tidak ditampilkan. Perhitungan rumus Okapi BM24 untuk *query* baru dapat dilihat pada lampiran A. Dokumen yang ditampilkan pada pencarian menggunakan *query* baru adalah dok3, dok5, dok2, dok1, dok4.

4.2.6. Perancangan tampilan sistem

Tahap perancangan sistem ini bertujuan sebagai acuan untuk tahap implementasi, dan juga memberikan gambaran antarmuka sistem yang akan dibangun. Perancangan tampilan sistem yang akan dibuat harus memenuhi aspek kenyaman dan kemudahan untuk digunakan oleh pengguna. Adapun beberapa dari rancangan tampilan tersebut, yaitu :

1. Tampilan Menu Sistem

   Gambar 4.4 merupakan rancangan tampilan struktur menu sistem, pengguna dapat memilih salah satu menu yang disediakan dalam pengoperasian sistem.



Gambar 4.4 Rancangan Tampilan Menu Sistem

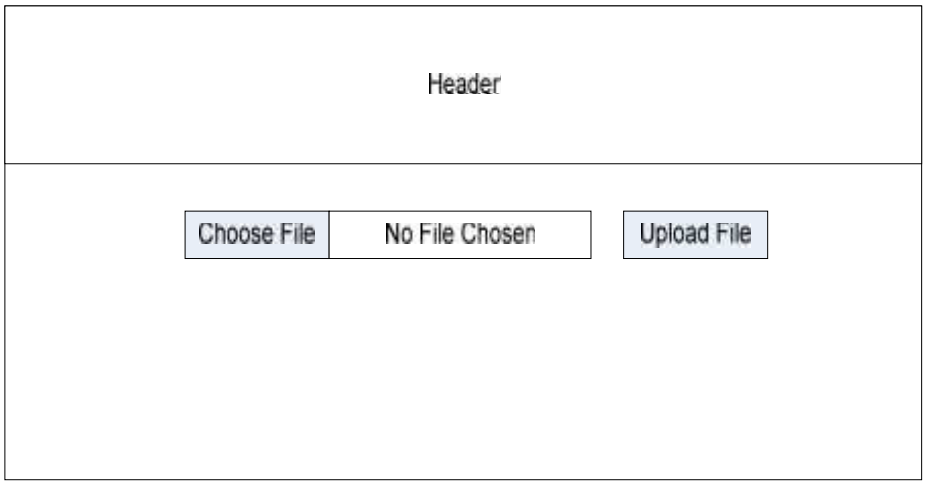
2. *Form* Utama sistem

   Gambar 4.5 adalah tampilan utama ketika pengguna menjalankan sistem. Pada *form* utama ini ada menu Corpus(koleksi dokumen), *Upload* data dan *Search*. Menu corpus adalah untuk melihat daftar koleksi dokumen yang ada di dalam sistem. Menu *Upload* adalah menu untuk *Upload file* dokumen baru kemudian dilakukan *preprocessing* pada *file* dokumen tersebut. Menu *Search* adalah menu untuk mencari dokumen dengan memasukkan *query* sebagai katakuncinya.

Gambar 4.5 Rancangan tampilan utama sistem

3. *Form* Upload Dokumen

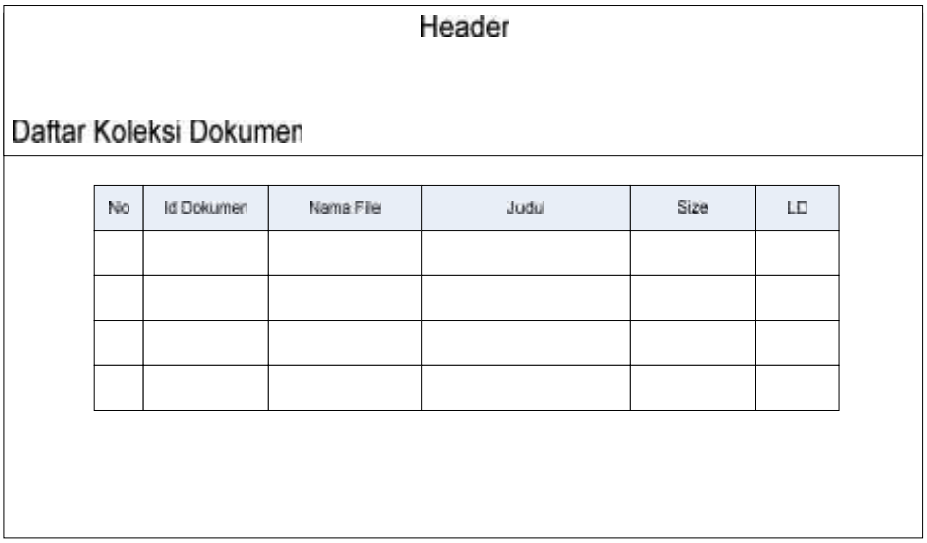
   Gambar 4.6 merupakan rancangan *form* koleksi dokumen, *form* ini berguna sebagai media penginputan dokumen yang akan digunakan dalam pencarian data koleksi pada sistem.



Gambar 4.6 Rancangan Form Upload Dokumen

4. Daftar Corpus

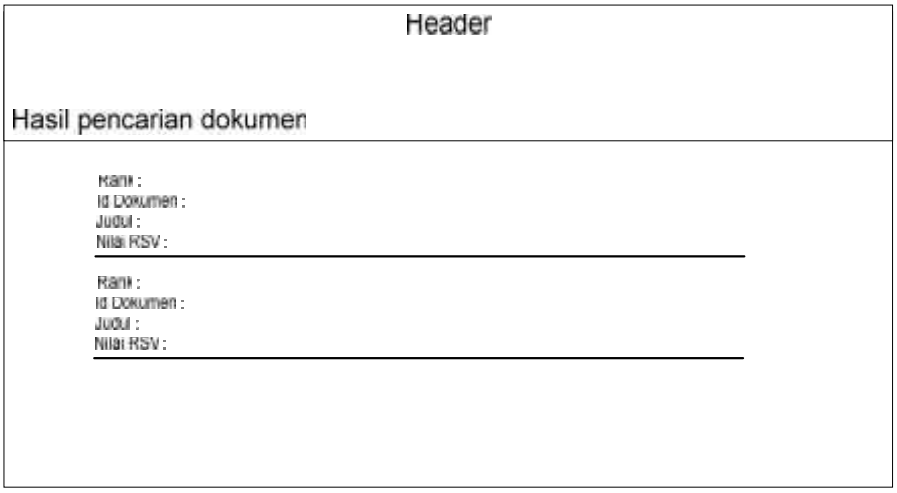
   Gambar 4.7 merupakan rancangan tampilan daftar corpus yang telah di upload dan diproses oleh sistem.



Gambar 4.7 Rancangan tampilan daftar Corpus

5. Tampilan Hasil pencarian

   Gambar 4.8 merupakan rancangan hasil pencarian, pengguna menginputkan *query* pada inputan teks lalu menekan tombol *Search* dan sistem akan menampilkan dokumen yang relevan sesuai dengan *query* pengguna tersebut.



Gambar 4.8 Rancangan tampilan hasil pencarian

BAB V
IMPLEMENTASI DAN PENGUJIAN

5.1. Implementasi

Tahap implementasi ini dilakukan setalah tahap analisa dan perancangan selesai dilakukan. Dalam tahap implementasi ini akan dilakukan pengkodingan terhadap apa yang telah dilakukan pada tahap analisa dan perancangan. Diharapkan hasil dari implementasi ini sesuai dengan apa yang menjadi tujuan awal penelitian ini.

5.1.1. Batasan Implementasi

Batasan dalam implementasi yaitu koleksi dokumen yang digunakan sebagai corpus berekstensi (*extension*) *.docx, dan *.doc untuk ukuran file maksimal 4Mb.

5.1.2. Lingkungan implementasi

Tahap implementasi dan pengujian ini dilakukan pada perangkat keras dan perangkat lunak dengan spesifikasi seperti dibawah:

1. Perangkat keras

   Processor : Intel Atom N550 1.5 GHz

   Ram : 1 GB

   Harddisk : 160 GB

2. Perangkat lunak

   Operating system : *Windows 7 Ultimate*

   Bahasa pemrograman : PHP

   Web Browser : *Google Chrome*

5.1.3. Hasil implementasi

Hasil implementasi yang telah dilakukan pengkodingan ditampilkan pada subbab ini. Adapun hasil analisa dan perancangan yang telah dilakukan

pengkodingan adalah antarmuka (*interface*) sistem, tampilan daftar stopwords, hasil stemming, daftar corpus, dan hasil perangkingan model Okapi BM25.

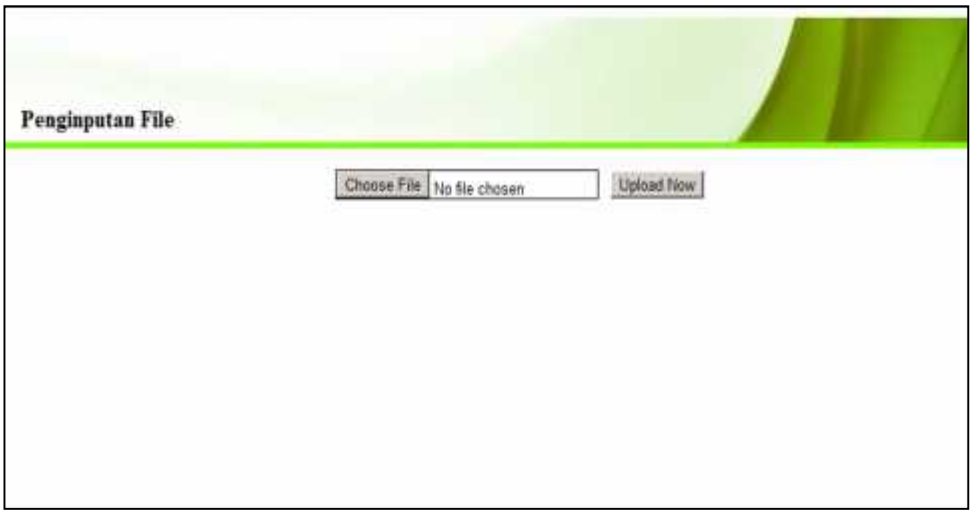
1. Tampilan antarmuka (*interface*)

Pada tampilan utama ini pengguna memasukkan query untuk dilakukan pencarian dokumen.

Gambar 5.1 Tampilan form utama

2. Tampilan *upload* dokumen

Proses penginputan dilakukan dengan memilih dokumen yang akan diproses ke tahap indexing. Setelah dokumen berhasil diupload maka dilakukan proses indexing dengan menekan tombol proses indexing.



Gambar 5.2 Tampilan penginputan *file*

3. Tampilan daftar dokumen corpus

Informasi yang disimpan dalam database untuk daftar dokumen corpus adalah id dokumen, nama *file* dokumen, judul dokumen, *size* dokumen, Ld (panjang dokumen).

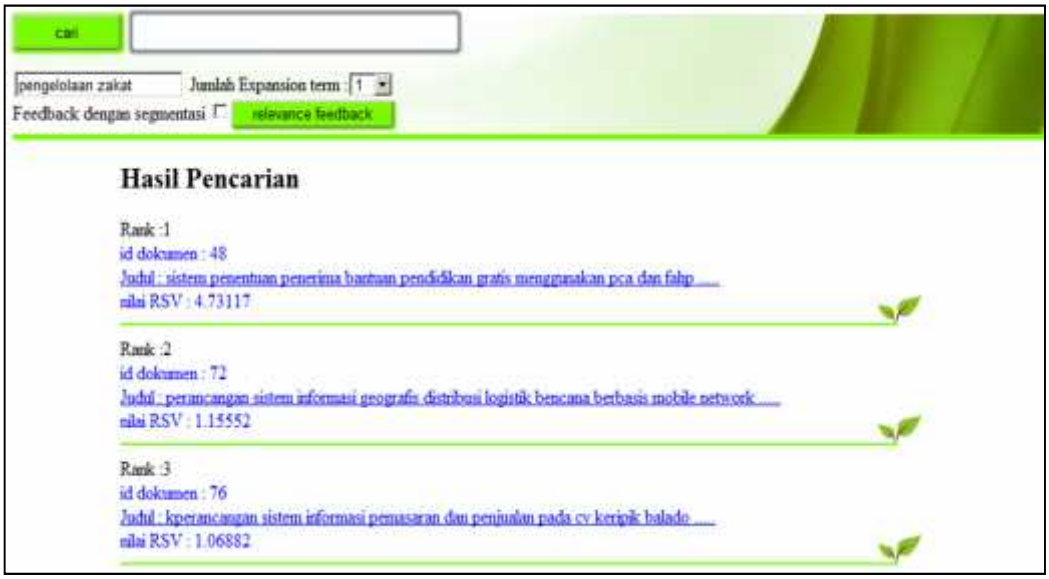
Daftar Koleksi Dokumen

ID DOKUMEN	NAME	JUDUL	SIZE	LD
1	CR2.doc	pengukuran maturity level e learning nurva widia idria matta jurusan	720896	1979
2	CR5.docx	spk untuk memilih metodologi pengembangan si studi kasus stmik atma	295889	2354
3	CR7.doc	perancangan aplikasi penerimaan mahasiswa baru berbasis service oriented architecture soa	2528768	2961
4	CR8.docx	klasifikasi citra content based image retrieval dengan metode shape base	2479865	2610
5	CR9.docx	penerapan teknologi 3d pada web berbasis vrml studi kasus pt	1549828	2105
6	CR10.docx	optimalisasi jumlah produksi minuman menggunakan metode fuzzy linear programming di	245915	2129
7	CR11.doc	robot pemisawa barang berbasis mikrokontroler atmega8535l dengan pengendali remote muhammad	370176	2043
8	CR15.docx	sistem pakar penyakit kulit pada manusia menggunakan metode certainty factor	545296	945
9	CR16.docx	sistem informasi inventory pengeboran minyak menerapkan metode material requirement planning	1015200	1114
10	CR18.docx	sistem informasi tugas akhir menggunakan model ruang veldor studi kasus	129499	1309
11	CR19.docx	evaluasi performansi video games ditinjau dari aspek usability studi kasus	85829	2206
12	CR20.docx	perhitungan rerating motor induksi akibat tegangan tidak seimbang dengan metode	126043	1319
13	CR21.doc	kriptografi dan kompresi pesan singkat pada android pizaini1 fabi yanto2	2022400	1501
14	CR22.docx	perangcangan perangkat lunak kriptografi citra digital dengan fft kunci chaos	1352873	1803
15	CR24.docx	analisis sebaran puskesmas untuk peningkatan pelayanan kesehatan dengan metode fuzzy	1227420	1348
16	CR25.doc	pengembangan model kompetensi peningkatan skill mahasiswa suranto1 1 teknik industri	2933760	1837

Gambar 5.3 Tampilan daftar dokumen corpus

4. Tampilan hasil pencarian dokumen

Query yang dimasukkan oleh pengguna akan diproses dan akan dicari dokumen relevan dengan *query* tersebut. Hasil dari pencarian akan ditampilkan dengan terurut menurun.



Gambar 5.4 Hasil pencarian dokumen

5. Tampilan hasil *relevance feedback* tanpa segmentasi

Hasil *relevance feedback* menggunakan metode *pseudo relevance feedback* tanpa segmentasi dapat dilihat pada gambar dibawah ini. Query awal yang telah di stemming adalah “kelola zakat” dan setelah dilakukan *feedback* tanpa segmentasi menjadi “kelola zakat siswa d”.

Gambar 5.5 Hasil *relevance feedback* tanpa segmentasi

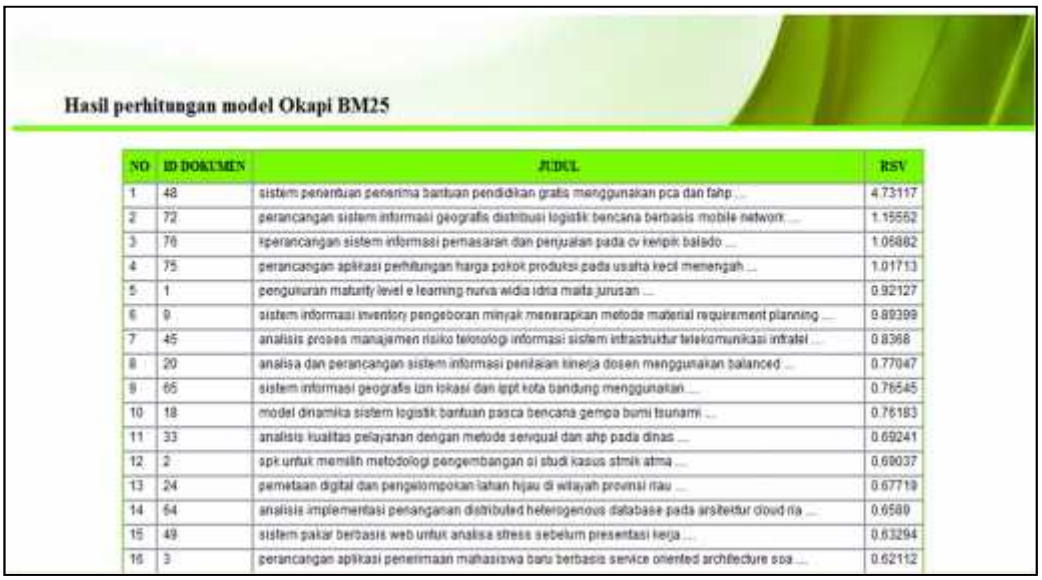
6. Tampilan hasil *relevance feedback* dengan segmentasi

Hasil *relevance feedback* menggunakan metode *pseudo relevance feedback* tanpa segmentasi dapat dilihat pada gambar dibawah ini. Query awal yang telah di stemming adalah “kelola zakat” dan setelah dilakukan *feedback* tanpa segmentasi menjadi “kelola zakat siswa d”.

Gambar 5.6 Hasil *relevance feedback* dengan segmentasi

7. Tampilan nilai perhitungan model Okapi BM25

Hasil perhitungan model Okapi BM25 dapat dilihat pada gambar dibawah ini. Hasil perhitungan tersebut telah diurutkan menurun dari yang terbesar hingga terkecil berdasarkan nilai RSV(*Retrieval Status Value*).



NO	ID DOKUMEN	JUDUL	RSV
1	48	sistem penentuan penerima bantuan pendidikan gratis menggunakan pca dan fahp	4.73117
2	72	perancangan sistem informasi geografis distribusi logistik bencana berbasis mobile network	1.15552
3	76	iperancangan sistem informasi pemasaran dan penjualan pada cv keripik balado	1.05882
4	75	perancangan aplikasi perhitungan harga pokok produksi pada usaha kecil menengah	1.01713
5	1	pengukuran maturity level e learning nurva widia idria maita jurusan	0.92127
6	9	sistem informasi inventory pengeboran minyak menerapkan metode material requirement planning	0.89399
7	45	analisis proses manajemen risiko teknologi informasi sistem infrastruktur telekomunikasi infratel	0.8368
8	20	analisa dan perancangan sistem informasi penilaian kinerja dosen menggunakan balanced	0.77047
9	65	sistem informasi geografis izin lokasi dan ippt kota bandung menggunakan	0.76545
10	18	model dinamika sistem logistik bantuan pasca bencana gempa bumi tsunami	0.76183
11	33	analisis kualitas pelayanan dengan metode servqual dan ahp pada dinas	0.69241
12	2	spk untuk memilih metodologi pengembangan si studi kasus stmik atma	0.69037
13	24	pemetaan digital dan pengelompokan lahan hijau di wilayah provinsi riau	0.67719
14	64	analisis implementasi penanganan distributed heterogenous database pada arsitektur cloud ria	0.6589
15	49	sistem pakar berbasis web untuk analisa stress sebelum presentasi kerja	0.63294
16	3	perancangan aplikasi penerimaan mahasiswa baru berbasis service oriented architecture soa	0.62112

Gambar 5.7 Hasil perhitungan model Okapi BM25

5.2. Pengujian sistem

Pengujian sistem akan dilakukan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan analisa dan tujuan dari dibangunnya sistem ini. Untuk mengetahui hasil dari sistem ini apakah sudah sesuai dengan yang diharapkan, dilakukan penghitungan *precision* dan *recall* berdasarkan dokumen yang dikembalikan oleh sistem.

5.2.1. Tahapan pengujian

Tahapan pengujian yang dilakukan adalah sebagai berikut:

1. Memasukkan *query* yang berbeda sebanyak 3 kali.
2. Menghitung nilai *precision* dan *recall* dari hasil pencarian pertama menggunakan model Okapi BM25.
3. Melakukan *feedback* setelah melakukan pencarian pertama.
4. Pengujian *feedback* tanpa segmentasi dengan jumlah *expansion term*-nya adalah 2, 6 dan 8.

5. Pengujian *feedback* dengan segmentasi jumlah *expansion term*-nya adalah 2, 6, dan 8.
6. Membandingkan nilai *precision* dan *recall* dari hasil pencarian sebelum *feedback*, *feedback* tanpa segmentasi dan *feedback* dengan segmentasi.

5.2.2. Hasil pengujian

Query yang dimasukkan sebagai pengujian sistem adalah sebagai berikut:

Tabel 5.1 Daftar *query*

No	*Query*	Jumlah Dokumen Relevan
1	Sistem pendukung keputusan	10
2	Aplikasi mobile	8
3	Sistem pakar	6

Setelah dilakukan pengujian menggunakan *query* diatas maka nilai *precision* dan *recall* dapat dicari menggunakan rumus 2.5 dan 2.6. Tabel 5.2 dibawah ini adalah hasil pencarian pada 75 dokumen corpus yang ada didalam *database* menggunakan *query* pada tabel 5.1.

Tabel 5.2 Hasil pencarian dengan model Okapi BM25

No	Query	Hasil dokumen	Relevan	Non relevan
1	sistem pendukung keputusan	67	9	58
2	aplikasi mobile	51	8	43
3	sistem pakar	66	6	60

Berdasarkan tabel 5.2 untuk *query* 1 “sistem pendukung keputusan” jumlah dokumen yang ditampilkan adalah 67 dokumen, jumlah dokumen relevan adalah 9 dokumen dan jumlah dokumen yang tidak relevan adalah 58. Nilai *precision* dan *recall* dari *query* 1 adalah:

Tabel 5.3 Pengujian *query* 1

	relevant	***Non relevant***
retrieved	9	58
no retrieved	1	7

Precision = *relevant*/(*relevant* + *non relevant*) = 9/(9+58) = 9/67 = 0.134

Recall = *relevant* / (*relevant* + *no retrieved*) = 9/(9+1) = 9/10 = 0.9

Berdasarkan tabel 5.2 total dokumen yang ditampilkan dari hasil pencarian dengan *query* 2 “aplikasi mobile” adalah 58 dokumen, dokumen relevan adalah 8 dokumen dan dokumen tidak relevan adalah 50 dokumen. Nilai *precision* dan *recall* adalah:

Tabel 5.4 Pengujian *query* 2

	relevant	***Non relevant***
retrieved	8	43
no retrieved	0	24

Precision = *relevant*/(*relevant* + *non relevant*) = 8/(8+43) = 8/51 = 0.156

Recall = *relevant* / (*relevant* + *no retrieved*) = 8/(8+0) = 8/8 = 1

Berdasarkan tabel 5.2 total dokumen yang ditampilkan dari hasil pencarian dengan *query* 3 “sistem pakar” adalah 66 dokumen, dokumen relevan adalah 6 dokumen dan dokumen tidak relevan adalah 60 dokumen. Nilai *precision* dan *recall* adalah:

Tabel 5.5 Pengujian *query* 3

	relevant	***Non relevant***
retrieved	6	60
no retrieved	0	9

Precision = *relevant*/(*relevant* + *non relevant*) = 6/(6+60) = 6/66 = 0.09

Recall = *relevant* / (*relevant* + *no retrieved*) = 6/(6+0) = 6/6 = 1

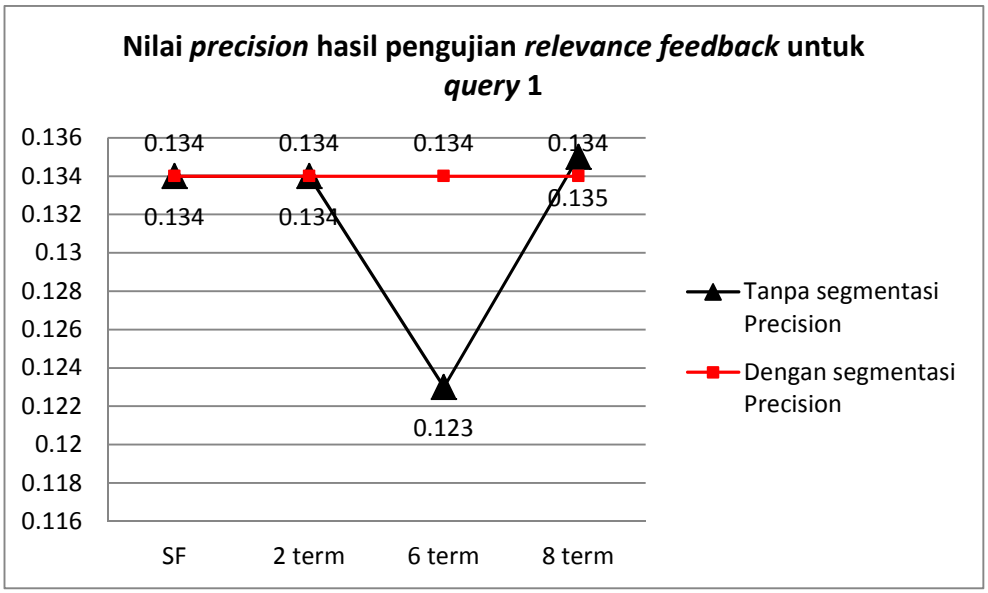
5.2.3. Pengujian *relevance feedback*

Pengujian *relevance feedback* dilakukan untuk mengetahui apakah metode *pseudo relevance feedback* yang diterapkan pada sistem yang dibangun sesuai dengan tujuan dari penelitian ini. Pengujian yang dilakukan adalah pengujian *relevance feedback* tanpa segmentasi dokumen dan *relevance feedback* dengan segmentasi dokumen dengan jumlah *expansion term*-nya adalah 2, 6 dan 8 *term*.

1. Pengujian *relevance feedback* untuk *query* 1

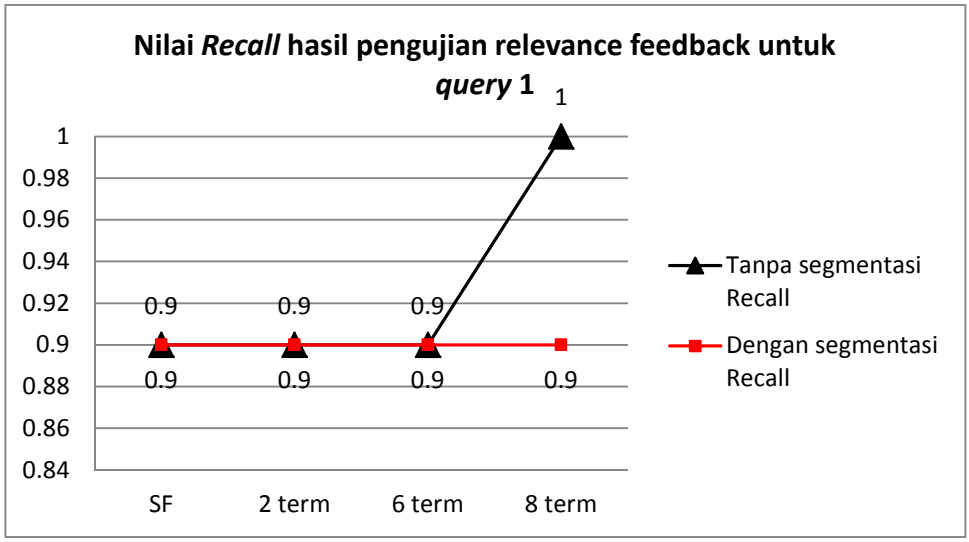
Nilai *precision relevance feedback query* 1 dengan jumlah *expansion term*-nya 2 untuk pengujian tanpa segmentasi dan dengan segmentasi adalah 0.134. Sedangkan nilai *precision* dengan jumlah *expansion term*-nya 6 untuk

pengujian tanpa segmentasi mengalami penurunan yaitu 0.123 dan pengujian dengan segmentasi nilai *precision*-nya tetap yaitu 0.134. kemudian nilai *precision* dengan jumlah *expansion term*-nya 8 untuk pengujian tanpa segmentasi yaitu 0.135 dan pengujian dengan segmentasi yaitu 0.134. Untuk lebih jelas dapat dilihat pada gambar 5.8 grafik nilai *precision* hasil pengujian *relevance feedback* untuk *query* 1 mulai dari pencarian sebelum *feedback*(SF).



Gambar 5.8 Grafik nilai *precision* hasil *relevance feedback* untuk *query* 1

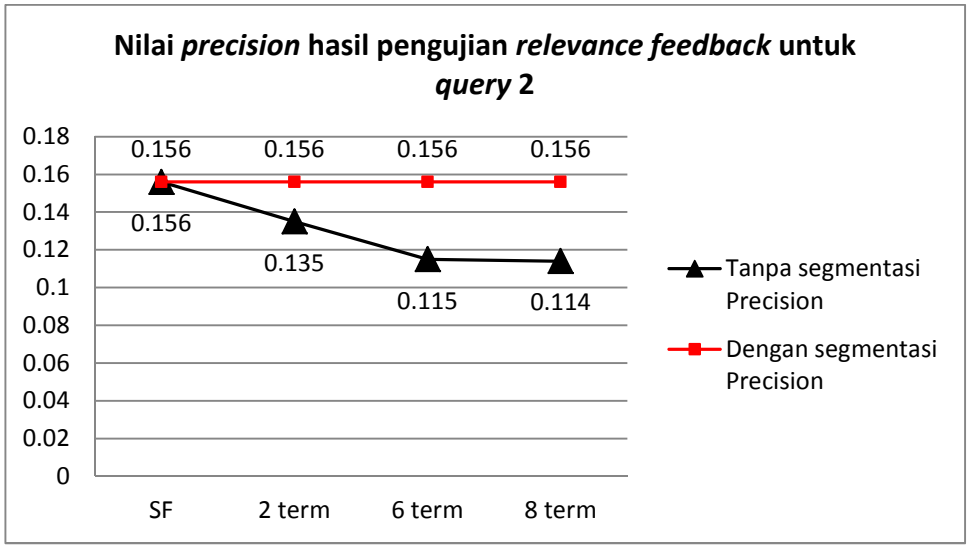
Nilai *recall relevance feedback query* 1 dengan jumlah *expansion term*-nya 2, 6 untuk pengujian tanpa segmentasi dan dengan segmentasi adalah 0.9. Sedangkan nilai *recall* dengan jumlah *expansion term*-nya 8 untuk pengujian tanpa segmentasi mengalami peningkatan yaitu 1 dan pengujian dengan segmentasi nilai *recall*-nya tetap yaitu 0.9. Untuk lebih jelas dapat dilihat pada gambar 5.9 grafik nilai *recall* hasil pengujian *relevance feedback* untuk *query* 1 mulai dari pencarian sebelum *feedback*(SF).



Gambar 5.9 Grafik nilai *recall* hasil *relevance feedback* untuk *query* 1

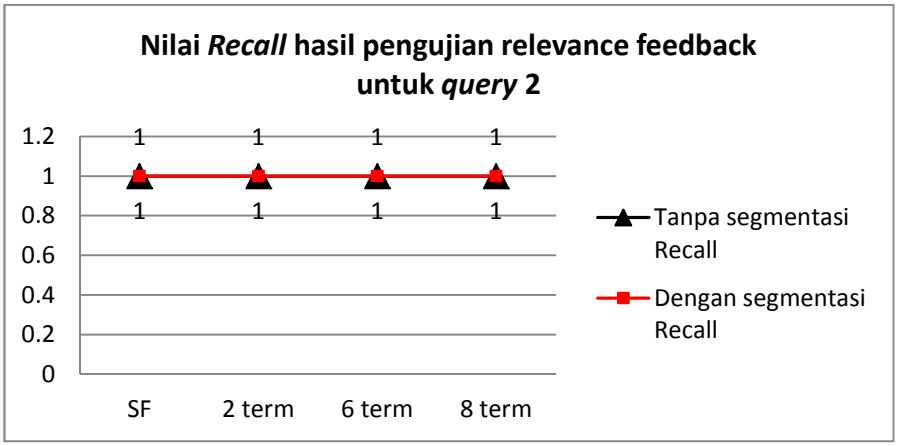
2. Pengujian *relevance feedback* untuk *query* 2

Nilai *precision relevance feedback query* 2 untuk pengujian tanpa segmentasi mengalami penurunan mulai dari jumlah *expansion term*-nya 2 sampai 8 yaitu 0.135, 0.115, 0.114. Sedangkan untuk pengujian dengan segmentasi nilai *precision*-nya stabil yaitu 0.156. Untuk lebih jelas dapat dilihat pada gambar 5.10 grafik nilai *precision* hasil pengujian *relevance feedback* untuk *query* 2 mulai dari pencarian sebelum *feedback*(SF).



Gambar 5.10 Grafik nilai *precision* hasil *relevance feedback* untuk *query* 2

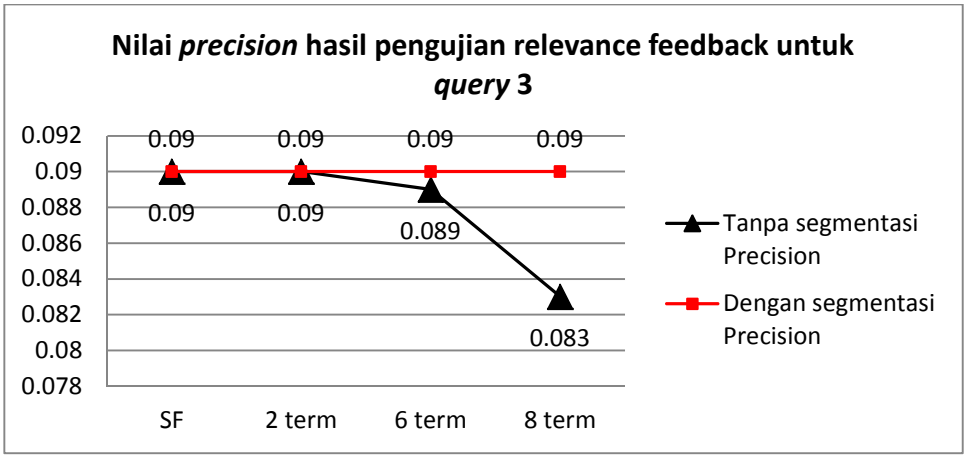
Nilai *recall relevance feedback query* 2 dengan jumlah *expansion term*-nya 2, 6, dan 8 untuk pengujian tanpa segmentasi dan dengan segmentasi adalah sama yaitu 1. Untuk lebih jelas dapat dilihat pada gambar 5.11 grafik nilai *recall* hasil pengujian *relevance feedback* untuk *query* 2 mulai dari pencarian sebelum *feedback*(SF).



Gambar 5.11 Grafik nilai *recall* hasil *relevance feedback* untuk *query* 2

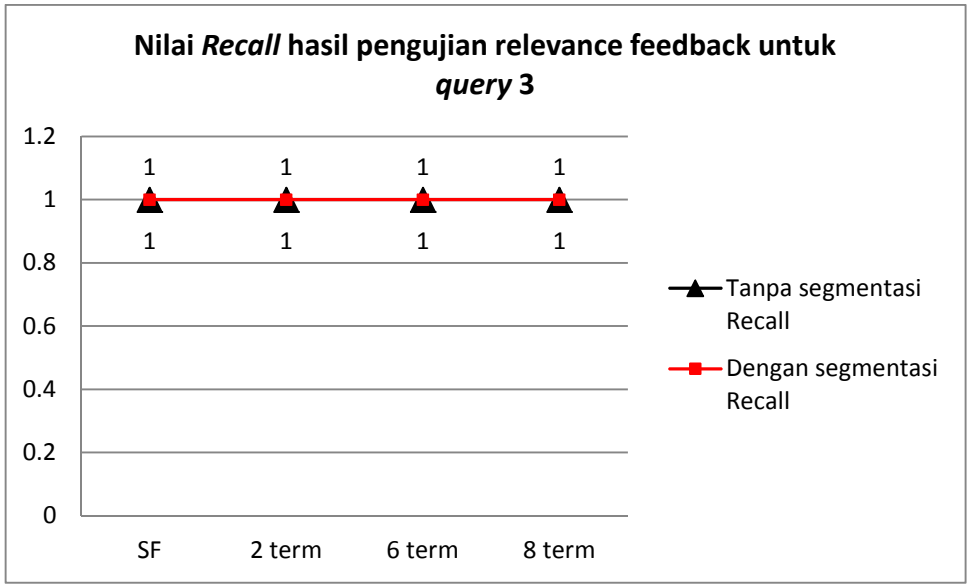
3. Pengujian *relevance feedback* untuk *query* 3

Nilai *precision relevance feedback query* 3 untuk pengujian tanpa segmentasi mengalami penurunan mulai dari jumlah *expansion term*-nya 2 sampai 8 yaitu 0.09, 0.089, 0.083. Sedangkan untuk pengujian dengan segmentasi nilai *precision*-nya stabil yaitu 0.09. Untuk lebih jelas dapat dilihat pada gambar 5.12 grafik nilai *precision* hasil pengujian *relevance feedback* untuk *query* 3 mulai dari pencarian sebelum *feedback*(SF).



Gambar 5.12 Grafik nilai *precision* hasil *relevance feedback* untuk *query* 3

Nilai *recall relevance feedback query* 3 dengan jumlah *expansion term*-nya 2, 6, dan 8 untuk pengujian tanpa segmentasi dan dengan segmentasi adalah sama yaitu 1. Untuk lebih jelas dapat dilihat pada gambar 5.13 grafik nilai *recall* hasil pengujian *relevance feedback* untuk *query* 3 mulai dari pencarian sebelum *feedback*(SF).



Gambar 5.13 Grafik nilai *recall* hasil *relevance feedback* untuk *query* 3

Tabel 5.6 dibawah ini merupakan hasil *query* awal yang telah diperluas dengan metode *pseudo relevance feedback* dengan mengambil *expansion term* 2, 6 dan 8 teratas dari dokumen *feedback*.

Tabel 5.6 Hasil perluasan *query* awal

Query awal	Perluasan *query* tanpa segmentasi			Perluasan *query* dengan segmentasi		
	2 term	6 term	8 term	2 term	6 term	8 term
Sistem pendukung keputusan	sistem dukung putus elemen luhur	sistem dukung putus elemen luhur hierarchy 01systems hirarki	sistem dukung putus elemen luhur hierarchy 01 systems hirarki bobot 2000	sistem dukung putus pintas intuisi	sistem dukung putus pintas intuisi pegembangan multikriteria melatarbelakangi ketidakakuratan	sistem dukung putus pintas intuisi pegembangan multikriteria
aplikasi mobile	aplikasi mobile lokasi manusia	aplikasi mobile lokasi manusia analisa posisi diagram komunikasi	aplikasi mobile lokasi manusia analisa posisi diagram komunikasi aktifitas deskripsi	aplikasi mobile readfile awa	aplikasi mobile readfile awa launch firsttextfieldform tampilancari alquran	aplikasi mobile readfile awa launch firsttextfieldform tampilancari alquran formpililihan jendela
Sistem pakar	sistem pakar hadap server	sistem pakar hadap server url dokumen gunadarma script	sistem pakar hadap server url dokumen gunadarma script browser rasa	sistem pakar gondran diinterrelasikan	sistem pakar gondran diinterrelasikan prolog exsys crystal lisp	sistem pakar gondran diinterrelasikan prolog exsys crystal lisp menghim jemu

5.2.4. Kesimpulan pengujian

Hasil dari pengujian yang telah dilakukan, dapat disimpulkan bahwa:

1. Presentase kualitas *retrieval* untuk *query* 1 dengan model Okapi BM25 sebelum dilakukan *feedback* yaitu *precision* 13% dan *recall* 90%, untuk *query* 2 presentase *retrieval* yaitu *precision* 15% dan *recall* 100%, untuk *query* 3 presentase *retrieval* yaitu *precision* 9% dan *recall* 100%.
2. Presentase kualitas *retrieval* setelah dilakukan *feedback* tanpa segmentasi dokumen untuk *query* 1 nilai rata-rata *precision* dengan pengujian 2, 6 dan 8 adalah 13.6% sedangkan rata-rata nilai *recall*-nya adalah 93% . Untuk *query* 2 dengan pengujian 2, 6 dan 8 *term* rata-rata nilai *precision*-nya adalah 12% dan rata-rata nilai *recall*-nya adalah 100%. Untuk *query* 3 dengan pengujian 2, 6, 8 *term* rata-rata nilai *precision*-nya adalah 8% dan rata-rata nilai *recall* adalah 100%.
3. Presentase kualitas *retrieval* setelah dilakukan *feedback* dengan segmentasi dokumen untuk *query* 1 nilai rata-rata *precision* dengan pengujian 2, 6 dan 8 adalah 13,4% sedangkan rata-rata nilai *recall*-nya adalah 90% . Untuk *query* 2 dengan pengujian 2, 6, dan 8 *term* rata-rata nilai *precision*-nya adalah 15,6% dan rata-rata nilai *recall*-nya adalah 100%. Untuk *query* 3 dengan pengujian 2, 6, 8 *term* rata-rata nilai *precision*-nya adalah 9% dan rata-rata nilai *recall* adalah 100%.

BAB VI
PENUTUP

6.1. Kesimpulan

Kesimpulan dari penelitian tugas akhir ini adalah:

1. Kemampuan sistem IR yang dibuat untuk proses pencarian tanpa *relevance feedback* memiliki rata-rata *precision recall* yaitu 12.6% dan 96.6%. Untuk proses pencarian setelah *relevance feedback* tanpa segmentasi dokumen memiliki rata-rata nilai *precision recall*-nya yaitu presentasi kinerja 11% dan 97.6%. Sedangkan untuk proses pencarian setelah *relevance feedback* dengan segmentaasi dokumen nilai rata-rata *precision recall*-nya adalah 12.6% dan 96.6%.
2. Hasil pengujian yang dilakukan terlihat bahwa kualitas pengambilan dokumen sistem IR untuk model Okapi BM25 dan penerapan metode *pseudo relevance feedback* dengan segmentasi memiliki presentasi yang stabil terhadap hasil *feedback* tanpa segmentasi dokumen.
3. Berdasarkan hasil pengujian pada lampiran B, proses pencarian dokumen menggunakan *relevance feedback* tanpa segmentasi memiliki waktu proses jauh lebih lama dari pada *relevance feedback* dengan segmentasi dokumen.
4. *Expansion term* yang didapat dari hasil *relevance feedback* kebanyakan adalah kata yang tidak berhubungan dengan inisial *query*.

6.2. Saran

Adapun saran penulis dalam penelitian ini untuk pengembangan selanjutnya adalah sebagai berikut:

1. Koleksi dokumen corpus pada penelitian ini sebaiknya ditambah dengan dokumen-dokumen lain untuk mendapatkan hasil yang lebih baik dalam pencarian.

2. Sebelum proses *preprocessing* untuk *indexing* sebaiknya dilakukan perbaikan pada dokumen dari segi penulisan kata-kata, supaya dalam proses tokenisasi *term* tidak ada *term* yang tergabung tanpa spasi.
3. Untuk penelitian selanjutnya pemilihan segmen untuk *expansion term* sebaik nya dilakukan pada bagian awal dokumen atau pada bab pendahuluan.
4. Untuk *expansion term* yang akan dipakai pada *query* baru sebaiknya dilakukan pemilihan oleh pengguna untuk mendapatkan *expansion term* yang berhubungan dengan inisial *query*.

DAFTAR PUSTAKA

Adisantoso J dan Ridha A, *Corpus Dokumen Teks Bahasa Indonesia Untuk Pengujian Efektifitas Temu Kembali Informasi*, Laporan Hibah Penelitian SP4, Departemen Ilmu Komputer FMIPA IPB, Bogor, 2004.

Agusta, Ledy. *Perbandingan Algortima Stemming Porter dengan Algortima Nazief dan Adriani untuk Stemming Dokumen Teks Bahasa Indonesia*. Konferensi Nasional Sistem dan Informatika. Novermber 2009.

Anbiana, Elenur Dwi. *Pseudo Relevance Feedback pada temu kembali menggunakan segementasi dokumen*. Skripsi Departemen Ilmu Komputer, FMIPA, IPB, Bogor, 2009.

Cios, Krzstof J. Etc., *Data Mining A Knowledge Discovery Approach,* Springer, 2007.

Manning, Christopher D, dkk. *An Introduction to Information Retrieval*. Cambridge University Press. Online Edition, April 2009.

Mandala, Rila. *Evaluasi Efektifitas Metode Machine-Learning pada Search-Engine,* Seminar Nasional Aplikasi Teknologi Informasi, ISSN: 1907-5022. Yogyakarta, 2006.

Robertson, S E and Walker, S. *Okapi/Keenbow at TREC-8*. Cambridge University, London, 2000.