

QUERY EXPANSION PADA SISTEM TEMU KEMBALI INFORMASI DOKUMEN BERBAHASA INDONESIA MENGGUNAKAN PSEUDO RELEVANCE FEEDBACK

Studi kasus: Perpustakaan Universitas Brawijaya

Zanwar Yoga Pamungkas¹, Indriati², Achmad Ridok³

Informatika/Illmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya^{1,2,3}

Jalan Veteran Malang 65145, Indonesia

Email : zanwaryoga.p[at]gmail.com¹, indriati.tif[at]ub.ac.id², acridokb[at]ub.ac.id³

ABSTRAK

Perpustakaan Pusat Universitas Brawijaya memiliki **37.765** koleksi *local content* yang terdiri dari skripsi, disertasi, thesis, penelitian dan jurnal (<http://www.digilib.ub.ac.id/bkg/index.aspx>, diakses 17 Februari 2015). Sistem pencarian diperlukan untuk mempermudah *user* dalam menemukan referensi yang ada pada perpustakaan Universitas Brawijaya. Sistem pencarian jurnal di perpustakaan Universitas Brawijaya yang ada saat ini, hanya berfokus pada judul jurnal. Padahal terdapat objek yang tidak kalah penting untuk digunakan sebagai acuan dalam pencarian jurnal yaitu abstrak. Abstrak dari sebuah jurnal dapat digunakan untuk acuan pencarian karena abstrak merupakan bagian singkat dan ringkas dari suatu karya ilmiah yang mengandung unsur-unsur penting. Proses pencarian menggunakan *information retrieval*. Metode *pseudo relevance feedback* digunakan untuk mengambil beberapa dokumen teratas yang telah terambil dan selanjutnya dokumen teratas tersebut digunakan untuk menentukan kata-kata yang akan disarankan untuk *user*. Penentuan kata-kata menggunakan metode *rocchio relevance feedback*. Penelitian ini membuktikan bahwa dengan *query expansion* dapat meningkatkan nilai rata-rata *precision* sebesar 0,17, nilai rata-rata *recall* sebesar 0.03, dan nilai rata-rata *f-measure* sebesar 0.18 dari nilai sebelum menggunakan *query expansion*.

Kata Kunci: *query expansion, pseudo relevance feedback, information retrieval, rocchio relevance feedback*

ABSTRACT

Central Library of Brawijaya University has **37.765** local content collection that contains of minor thesis, thesis, dissertation, research and journal (<http://www.digilib.ub.ac.id/bkg/index.aspx>, accessed on February 17th, 2015). Searching system needed to ease user in finding reference which exist on the Library of Brawijaya University. Journal searching system that currently exist on Library of Brawijaya University, only focused on the title of the journal. Where there are objects that no less important to be used as reference in journal searching, that is the abstract. Abstract of a journal can be used as searching reference because abstract is a brief and concise part of a scientific work that contains important informations. Searching process using information retrieval with cosine similarity method. Pseudo relevance feedback method used to retrieve a few top documents that has been retrieved before and thus those top documents used to determine the words that will be suggested to the user. Words determination using rocchio relevance feedback method. This research prove that query expansion can increase average precision value into 0.17, average recall value into 0.03 and average f-measure value into 0.18 from the value before using query expansion.

Keywords: *query expansion, pseudo relevance feedback, information retrieval, rocchio relevance feedback*

1 PENDAHULUAN

1.1 Latar Belakang

Perpustakaan Pusat Universitas Brawijaya memiliki **37.765** koleksi *local content* yang terdiri dari skripsi, disertasi, thesis, penelitian dan jurnal (<http://www.digilib.ub.ac.id/bkg/index.aspx>, diakses 17 Februari 2015). Sistem pencarian diperlukan untuk mempermudah user dalam menemukan referensi yang ada pada perpustakaan Universitas Brawijaya. Sistem

pencarian jurnal di perpustakaan Universitas Brawijaya yang ada saat ini, hanya berfokus pada judul jurnal. Padahal terdapat objek yang tidak kalah penting untuk digunakan sebagai acuan dalam pencarian jurnal yaitu abstrak. Abstrak dari sebuah jurnal dapat digunakan untuk acuan pencarian karena abstrak merupakan bagian singkat dan ringkas dari suatu karya ilmiah yang mengandung unsur-unsur penting.

Permasalahan dalam menerapkan mesin pencarian, yaitu tidak semua hasil yang dikeluarkan oleh sistem, sesuai dengan yang diinginkan oleh *user*. Hal ini disebabkan oleh banyak faktor salah satunya yaitu *user* memasukkan *query* yang kurang spesifik. Berdasarkan permasalahan tersebut diperlukan suatu metode untuk memperbaiki *query* yang dimasukkan oleh *user* sehingga dapat meningkatkan kinerja dari sistem pencarian dan meningkatkan kepuasan *user*.

Sebuah mesin pencarian menerapkan Sistem Temu Kembali Informasi atau *Information Retrieval*. *Information Retrieval* (IR) merupakan suatu sistem yang digunakan untuk menemukan kembali (*retrieve*) informasi-informasi yang relevan terhadap kebutuhan *user* dari suatu kumpulan informasi berdasarkan *query* atau masukan dari *user* (Maruhum, 2009). Salah satu teknik untuk perbaikan *query* dalam *Information Retrieval* yaitu *query expansion*.

Query expansion yaitu menambahkan *query* tambahan pada *query* awal agar performansi ketepatan meningkat (Fachruddin, 2011). *Query* tambahan tersebut diharapkan dapat memberikan informasi yang lebih akurat untuk *user*. Perluasan *query* dapat dilakukan dengan metode *pseudo relevance feedback*. *Pseudo relevance feedback* (PRF) adalah teknik ekspansi *query* yang mengekstraksi istilah yang sangat berbobot dari dokumen peringkat teratas (diasumsikan bahwa dokumen peringkat teratas dianggap relevan) yang dikembalikan oleh pencarian awal (Zhou & dkk, 2013). *Pseudo relevance feedback* tidak melibatkan pengguna dalam perolehan keputusannya. Cara kerja *pseudo relevance feedback* adalah mengambil dokumen teratas dan secara sederhana mengasumsikan dokumen tersebut relevan (Yugianus, 2013). Dokumen peringkat teratas selanjutnya diproses untuk menemukan kata kunci dengan bobot tertinggi. Kata kunci tersebut digunakan untuk perluasan *query* pada *query* kedua.

Fachruddin (2011) dalam penelitiannya membahas mengenai Analisis dan Implementasi *Pseudo Relevance Feedback* dengan Kueri *Expansion* Menggunakan *Term Selection Value*. Hasilnya metode *Pseudo Relevance Feedback* mampu menaikkan hasil pencarian hingga 15% sesuai dengan parameter yang digunakan. Parameter yang dipakai yaitu batas *similarity*, batas *top-n*, dan panjang *query* (Fachruddin, 2011). Jajali dan Reza (2010) dalam penelitiannya yang berjudul *Information retrieval with concept-based pseudo-relevance feedback in MEDLINE* membahas tentang perluasan *query* menggunakan metode *Pseudo Relevance Feedback*. Hasilnya metode ini dapat meningkatkan hasil pencarian yang relevan sebesar 21% daripada pencarian tanpa menggunakan metode *Pseudo Relevance Feedback* (Jajali & Reza, 2010).

Berdasarkan dari kajian penelitian sebelumnya dan permasalahan yang dihadapi dalam proses pencarian di perpustakaan Universitas Brawijaya, maka diperlukan penelitian untuk mengembangkan mesin pencarian *local content* perpustakaan Universitas Brawijaya. Mesin pencarian tersebut diharapkan dapat meningkatkan hasil pencarian yang relevan serta meningkatkan kepuasan pengguna sistem di perpustakaan Universitas Brawijaya. Penelitian ini mencoba untuk menyelesaikan permasalahan tersebut dengan menerapkan *Pseudo Relevance Feedback* untuk perluasan *query* pada mesin pencarian *local content* perpustakaan Universitas Brawijaya.

1.2 Rumusan Masalah

Berdasarkan paparan latar belakang tersebut, maka masalah yang dapat dirumuskan adalah :

1. Bagaimana menerapkan metode *pseudo relevance feedback* untuk *query expansion* pada pencarian *local content* di Perpustakaan Universitas Brawijaya?
2. Bagaimana perbandingan sistem yang menggunakan metode *pseudo relevance feedback* dengan yang tidak menggunakan metode tersebut?

1.3 Batasan Masalah

Untuk menghindari semakin melebarnya masalah, maka dari rumusan masalah yang telah dipaparkan, batasan masalah yang diterapkan adalah sebagai berikut:

1. Sistem yang dibuat adalah sistem pencarian *local content* pada perpustakaan Universitas Brawijaya.
2. Dokumen yang digunakan adalah abstrak dari *local content* perpustakaan Universitas Brawijaya yang berasal dari database perpustakaan Universitas Brawijaya.
3. Sistem tidak memperhatikan persamaan kata atau sinonim dalam proses *stemming*.
4. Sistem tidak memperhatikan imbuhan sisipan dalam proses *stemming*.
5. Sistem tidak memperhatikan kesalahan ejaan kata.
6. Sistem dibangun menggunakan bahasa pemrograman PHP dan database *Mysql Server*.

1.4 Tujuan

Menerapkan metode *pseudo relevance feedback* pada *query expansion* untuk pencarian jurnal di perpustakaan Universitas Brawijaya.

1.5 Manfaat

Manfaat dari penelitian ini diharapkan:

1. Memberikan hasil pencarian yang lebih akurat sesuai dengan yang diinginkan oleh *user*.
2. Memberikan lebih banyak dokumen yang relevan.

2. TINJAUAN PUSTAKA

2.1 Sistem Temu Kembali Informasi (Information Retrieval)

Information retrieval adalah proses yang terkomputerisasi untuk menghasilkan dokumen yang relevan berdasarkan *input* dari *user* melalui perbandingan antara *query* dari *user* dengan *index* yang dihasilkan dari dokumen yang terdapat dalam *information retrieval* (Yugianus, 2013). Setiap dokumen direpresentasikan oleh kata kunci, kata kunci disebut juga *index term*. *Index term* adalah kata yang secara semantik membantu *user* untuk mengetahui tema utama dari dokumen tersebut (Yugianus, 2013).

Dokumen yang diolah menggunakan *Information retrieval* dapat berupa teks, suara, gambar, atau data dan lain-lain. Koleksi data teks yang dapat dijadikan sumber pencarian juga dapat berupa pesan teks, seperti *email*, *fax*, dan dokumen berita, bahkan dokumen yang beredar di *internet* (Fachruddin, 2011). Dalam Tugas Akhir ini yang menjadi koleksi dokumen adalah dokumen teks dari abstrak jurnal yang diperoleh dari *local content* Perpustakaan Universitas Brawijaya.

2.2 Query Expansion

Query expansion adalah salah satu teknik dasar dalam *relevance feedback*. Di mana sistem akan menambahkan *query* tambahan pada pencarian kedua dari hasil pada pencarian pertama (Fachruddin, 2011). *Query expansion* menggambarkan seperangkat teknik untuk memodifikasi *query* untuk memenuhi suatu kebutuhan informasi. Teknik *Query expansion* dapat dibagi menjadi tiga (Selberg, 1997):

1. Manual Query Expansion (MQE)

Manual Query Expansion (MQE) adalah teknik yang pengguna sendiri yang melakukan modifikasi *query*. Sistem ini tidak membantu pengguna sama sekali.

2. Automatic Query Expansion (AQE)

Automatic Query Expansion (AQE) adalah teknik yang memodifikasi *query* tanpa kontrol pengguna. Sebagai contoh, sebuah sistem yang selalu menambahkan istilah sinonim dari *query* awal untuk *query* baru akan dianggap sistem AQE.

3. Interaktif Query Expansion (IQE)

Interaktif Query Expansion (IQE) adalah teknik di mana pengguna memiliki beberapa

interaksi dengan sistem dalam proses *Query Expansion*.

2.3 Pseudo Relevance Feedback

Teknik *Relevance Feedback* ditemukan pertama kali oleh Rocchio. *Relevance feedback* adalah teknik modifikasi *query* yang paling terkenal karena sering diimplementasikan dalam *information retrieval systems*. Dalam siklus *relevance feedback*, *user* disajikan hasil penelusuran, setelah itu *user* memeriksa dan menandai dokumen yang relevan. Cara kerja metode *relevance feedback* yaitu memilih *term* penting, dalam dokumen yang telah dianggap sebagai dokumen relevan oleh *user* dan menambah *term* penting tersebut dalam proses modifikasi *query* (Yugianus, 2013).

Terdapat 3 metode *relevance feedback* berikut ini (Yugianus, 2013):

1. User judgement (explicit relevance feedback)

Explicit relevance feedback diperoleh dari penilai relevan dari relevansi suatu dokumen yang diambil dengan *query* tertentu. Penilai relevan yang dimaksud adalah penilaian yang ditafsirkan sendiri oleh pengguna.

2. User behavior (implicit relevance feedback)

Implicit relevance feedback disimpulkan melalui perilaku pengguna, seperti mencatat dokumen-dokumen yang mereka lakukan dan dokumen yang tidak dipilih untuk dilihat, durasi waktu yang digunakan untuk melihat dokumen, atau *browsing* atau *scrolling* halaman. Perbedaan *implicit* dengan *explicit* mencakup:

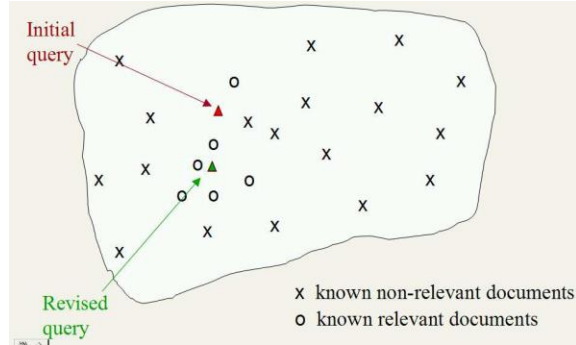
- a. Pengguna pada *implicit system* tidak menilai relevansi untuk kepentingan *information retrieval system*, tetapi hanya memuaskan kebutuhan mereka sendiri.
- b. Pengguna pada *implicit system* tidak perlu diberitahu bahwa perilaku mereka (memilih dokumen) akan digunakan sebagai *relevance feedback*.

3. Top K Retrieved (blind/pseudo relevance feedback)

Blind/pseudo relevance feedback tidak melibatkan pengguna dalam perolehan keputusannya. Cara kerja *blind/pseudo relevance feedback* adalah mengambil dokumen teratas dan secara sederhana mengasumsikan dokumen tersebut relevan. Jika pengambilan dokumen teratas dengan alasan yang cukup maka sistem akan mengambil *query* atau *terms* yang baik pula. Namun bila inisialisasi pengambilan dokumen tidak bagus akan terjadi pengambilan *query* atau *terms* yang buruk.

2.4 Rocchio Relevance Feedback

Algoritma Rocchio (Rocchio 1971) adalah algoritma klasik untuk mengimplementasikan *relevance feedback*. Mekanisme *relevance feedback* menggunakan algoritma Rocchio diperkenalkan dan di populerkan oleh Salton's SMART system sejak 1970 (UP, 2007). Pada konteks *query*, sistem temu kembali informasi mempunyai *query* dari *user* dan menghasilkan dokumen yang relevan dan dokumen tidak relevan yang digambarkan dalam gambar 2.1 berikut ini (UP, 2007):



Gambar 1.1 Aplikasi Algoritma Rocchio

Dalam proses pencarian akan menghasilkan dokumen yang relevan dan dokumen yang tidak relevan. *Query* yang diinputkan oleh *user* berada pada *centroid* dari seluruh dokumen, baik dokumen yang relevan maupun tidak. Kemudian *query* baru seharusnya berada pada *centroid* dokumen yang relevan. *Query* yang telah dimodifikasi dapat dihasilkan dengan persamaan berikut:

$$\vec{q}_m = \alpha \cdot \vec{q}_0 + \beta \frac{1}{|D_r|} \cdot \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \cdot \sum_{\vec{d}_k \in D_{nr}} \vec{d}_k \quad (2.5)$$

Dimana:

- q_m = *vector query* baru
- q_0 = *vector query* awal
- \vec{d}_j = Dokumen *vector* yang relevan
- \vec{d}_k = Dokumen *vector* yang tidak relevan
- D_r = *vector* dokumen yang relevan
- D_{nr} = *vector* dokumen yang tidak relevan
- α = Bobot *query* yang asli
- β = Bobot dokumen relevan
- γ = Bobot dokumen non relevan

2.5 Precision, Recall dan F-Measure

Sistem temu kembali informasi mengembalikan sekumpulan dokumen sebagai jawaban dari *query* pengguna. Terdapat dua kategori dokumen yang dihasilkan oleh sistem temu kembali informasi terkait pemrosesan *query*, yaitu *relevant documents* (dokumen yang relevan dengan *query*) dan *retrieved*

documents (dokumen yang diterima pengguna) (Maulana, 2014). Ukuran umum yang digunakan untuk mengukur kualitas dari data *retrieval* adalah kombinasi *precision* dan *recall*.

Precision merupakan proporsi dari suatu set yang diperoleh yang relevan (Maulana, 2014). *Recall* merupakan proporsi dari semua hasil yang relevan di koleksi termasuk hasil yang diperoleh atau dikembalikan (Maulana, 2014). *F-measure* biasa digunakan pada bidang sistem temu kembali informasi untuk mengukur klasifikasi pencarian dokumen dan performa *query classification*. (Maulana, 2014).

Proses evaluasi performa efektivitas dari sistem klasifikasi teks menggunakan suatu standar yang disebut matriks *confusion*. Matriks *confusion* berisi informasi mengenai klasifikasi yang sebenarnya dan prediksi klasifikasi yang dilakukan oleh sistem (Kiftiyani, 2014). Berikut ini adalah bentuk tabel matriks *confusion*:

Tabel 2.1 Confusion Matrix

		Actual Class (expectation)	
		+	-
Predicted Class (Observation)	+	TP	FP
	-	FN	TN

Keterangan:

TP: *True Positive*, menunjukkan bahwa hasil perangkingan oleh sistem memang merupakan dokumen yang sesuai dengan *query*.

FP: *False Positive*, menunjukkan bahwa dokumen yang termasuk dalam hasil perangkingan oleh sistem ternyata tidak sesuai dengan *query*.

FN: *False Negative*, menunjukkan bahwa dokumen yang tidak termasuk dalam hasil perangkingan oleh sistem ternyata seharusnya sesuai dengan *query*.

TN: *True Negative*, menunjukkan bahwa dokumen yang tidak termasuk hasil perangkingan oleh sistem memang sebenarnya tidak sesuai dengan *query*.

Hasil temu kembali setiap *query* dari Tabel 6.1 dihitung nilai *precision*, *recall*, dan *F-Measure* nya (Kiftiyani, 2014).

$$precision = \frac{TP}{TP + FP} \quad (2.6)$$

Nilai *precision* menunjukkan tingkat ketepatan/ketelitian dalam pengukuran hasil pengujian atau tingkat ketepatan antara informasi yang diminta oleh *user* dengan hasil jawaban yang diberikan oleh sistem.

$$recall = \frac{TP}{TP + FN} \quad (2.7)$$

Nilai *recall* menunjukkan tingkat jumlah sedikit dan banyaknya kesesuaian informasi yang didapatkan dari hasil percobaan berdasarkan sudut pandang kelas atau label yang digunakan.

$$akurasi = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.8)$$

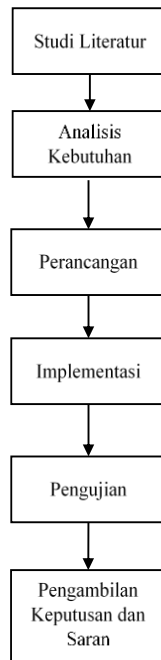
Semakin tinggi nilai akurasi menunjukkan kesesuaian nilai hasil prediksi pengujian dengan nilai aktual (*ground truth*) yang dibandingkan. *F1 measure* merupakan bobot *harmonic mean* pada *recall* dan *precision* dengan persamaan sebagai berikut:

$$F1 = \frac{2 \times recall \times precision}{recall + precision} \quad (2.9)$$

3 METODE PENELITIAN DAN PERANCANGAN

3.1 Metode Penelitian

Metodologi penelitian yang akan dilakukan pada penelitian ini secara umum ditunjukkan pada diagram alir di bawah ini.



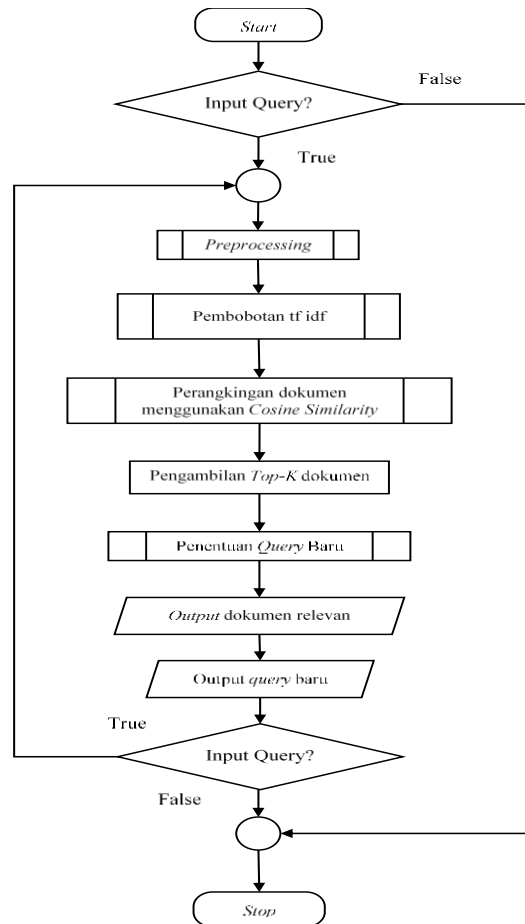
Gambar 3.1 Diagram Alir Metodologi Penelitian

3.2 Perancangan

3.2.1 Flowchart

Flowchart merupakan gambaran secara grafik yang terdiri dari simbol-simbol yang dibuatkan untuk memudahkan pemahaman proses yang terjadi dan

memahami hasil akhir yang terjadi. Sistem ini dimulai dengan masukan berupa *query* dari *user*, kemudian akan di lakukan *text preprocessing* dan pembobotan terhadap *query* tersebut. Selanjutnya dilakukan perangkingan dokumen yang relevan dengan *query user*. Dokumen dengan rangking teratas selanjutnya diambil untuk dilakukan *query reformulation*. Gambar 3.2 menunjukkan diagram alir sistem.



Gambar 3.2 Diagram Alir Sistem

4 IMPLEMENTASI

Aplikasi *Query expansion* sistem temu kembali dokumen bahasa Indonesia ini terdiri dari beberapa proses utama yaitu *preprocessing*, perangkingan dokumen dan penentuan *query* baru. Data berupa abstrak jurnal akan diolah dengan *preprocessing* hingga didapatkan kata dasar unik yang disebut *term*, kemudian ditentukan *rangking* dokumen yang relevan dengan menggunakan *cosine similarity*, selanjutnya

penentuan *query* baru menggunakan metode *Pseudo Relevance feedback* dan *Roccio Relevance Feedback*.

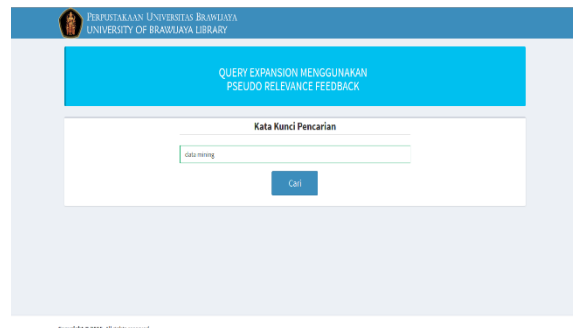
Daftar fungsi pada aplikasi *Query expansion* sistem temu kembali dokumen bahasa Indonesia ini dapat dilihat pada Tabel 4.1.

Tabel 4.1 Daftar Fungsi Pada Sistem

Proses	Fungsi	Keterangan
Preprocessing	function <i>tokenize()</i>	Fungsi untuk memecah kalimat menjadi kata yang berdiri sendiri.
	function <i>stopword ()</i>	Fungsi untuk menghilangkan kata yang tidak penting seperti 'yang', 'adalah', 'itu', dan lain-lain.
	function <i>stemming()</i>	Fungsi untuk memecah kata berimbuhan menjadi kata dasar.
Pembobotan	function <i>tf()</i>	Fungsi untuk menentukan <i>term frequency</i> dari masing-masing dokumen.
	function <i>wtf()</i>	Fungsi untuk menghitung bobot frekuensi tiap kata.
	function <i>idf()</i>	Fungsi untuk menghitung dokumen frekuensi.
Perangkingan Dokumen	function <i>cosineSimilarity()</i>	Fungsi untuk merangking dokumen yang relevan dengan <i>query</i> .
Penentuan Query Baru	function <i>queryBaru()</i>	fungsi untuk menentukan kata-kata yang akan ditampilkan untuk <i>user</i> .

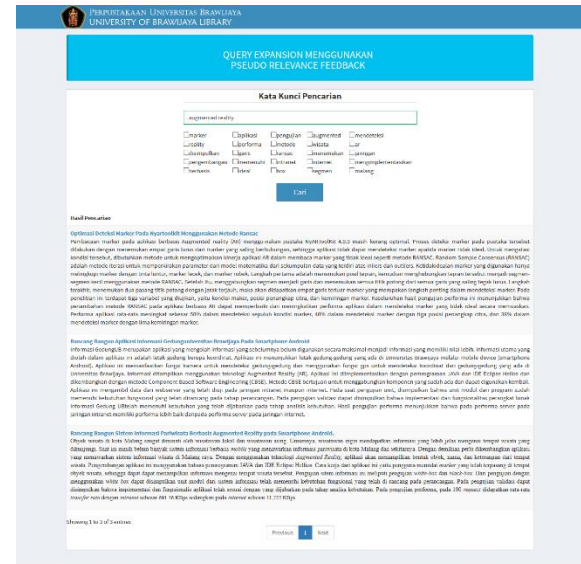
4.1 Implementasi Antar Muka

Tampilan halaman pencarian akan menampilkan *form input* untuk menginput *query* dan tombol *submit*. Gambar halaman *input* pencarian ditunjukkan pada Gambar 4.1.



Gambar 4.1 Halaman Pencarian

Tampilan halaman hasil pencarian akan menampilkan *form input* untuk menginput *query* dan tombol *submit* seperti pada halaman pencarian. Selanjutnya terdapat 25 kata untuk dipilih oleh *user* sebagai *query expansion*. Dan selanjutnya terdapat dokumen hasil pencarian yang telah dirangking dari yang paling relevan. Gambar halaman hasil pencarian ditunjukkan pada Gambar 4.2.



Gambar 4.2 Halaman Hasil

Tampilan halaman detail akan menampilkan detail dari jurnal yang telah dipilih sebelumnya. Gambar halaman detail ditunjukkan pada Gambar 4.3.



Gambar 4.3 Halaman Detail

5 PENGUJIAN

Pengujian akurasi sistem *query expansion* temu kembali informasi dokumen berbahasa Indonesia ini menggunakan metode *Precision*, *Recall*, dan *F-measure*. Nilai *Precision* dihitung menggunakan persamaan (2.6), nilai *Recall* dihitung menggunakan

persamaan (2.7) dan nilai *F-measure* dihitung menggunakan persamaan (2.9).

Pada pengujian ini akan membandingkan akurasi sistem sebelum menggunakan *query expansion* dan setelah menggunakan *query expansion*. Pengujian ini menggunakan dua skenario percobaan.

5.1 Skenario 1

Pengujian pada skenario ini membandingkan akurasi sistem saat sebelum menggunakan *query expansion* dan setelah menggunakan *query expansion*. Dalam menambahkan kata, pemilihan kata dilakukan secara acak dengan memperhatikan kesesuaian kata dengan *query* berdasarkan kesesuaian menurut penulis.

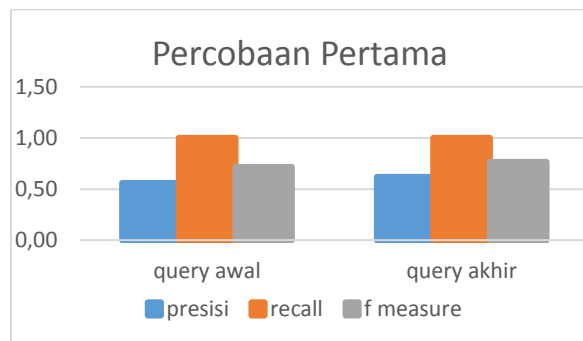
1. Percobaan 1

Percobaan pertama menggunakan *query* awal “permainan game”, dan diekspansi atau ditambah dengan kata “budaya”. Sehingga *query* akhir menjadi “permainan game budaya”. Hasil pengujian percobaan 1 ditunjukkan pada tabel 5.1.

Tabel 5.1 Hasil Pengujian Percobaan 1

Indikator	Query Awal	Query Akhir	Kenaikan
Precision	0,56	0,62	0,06
Recall	1	1	0
F Measure	0,72	0,77	0,05

Sumber: Pengujian



Gambar 5.1 Grafik Percobaan 1

Sumber: Pengujian

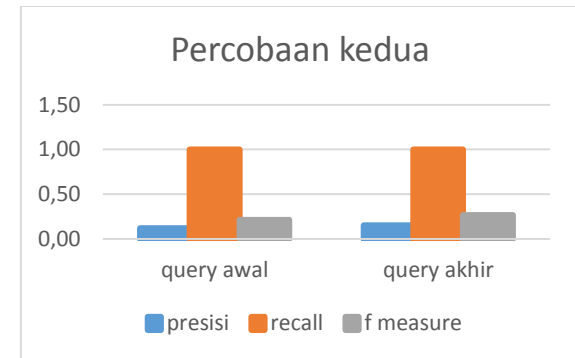
2. Percobaan 2

Percobaan kedua menggunakan *query* awal “klasifikasi text” dan ditambah dengan kata “jurnal”. Sehingga *query* akhir menjadi “klasifikasi text jurnal”. Hasil pengujian percobaan 2 ditunjukkan pada tabel 5.2.

Tabel 5.2 Hasil Pengujian Percobaan 2

Indikator	Query Awal	Query Akhir	Kenaikan
Precision	0,12	0,15	0,03
Recall	1	1	0
F Measure	0,21	0,27	0,05

Sumber: Pengujian



Gambar 5.2 Grafik Percobaan 2

Sumber: Pengujian

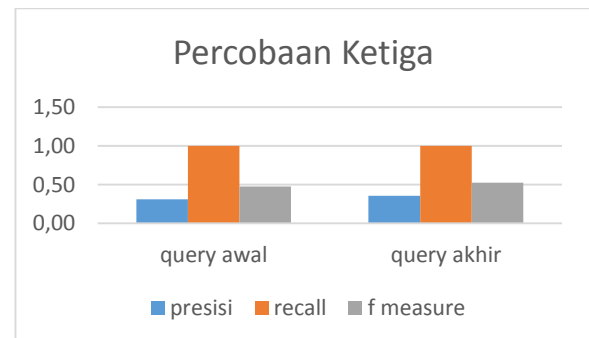
3. Percobaan 3

Percobaan ketiga menggunakan *query* awal “jaringan tiruan”, dan diekspansi atau ditambah dengan kata “syaraf”. Sehingga *query* akhir menjadi “jaringan tiruan syaraf”. Hasil pengujian percobaan tiga ditunjukkan pada tabel 5.3.

Tabel 5.3 Hasil Pengujian Percobaan 3

Indikator	Query Awal	Query Akhir	Kenaikan
Precision	0,31	0,36	0,04
Recall	1	1	0,00
F Measure	0,48	0,53	0,05

Sumber: Pengujian



Gambar 5.3 Grafik Percobaan 3

Sumber: Pengujian

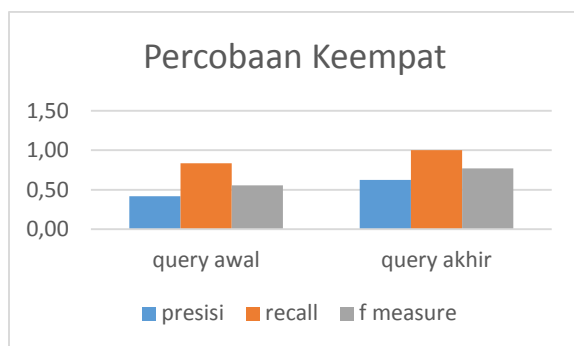
4. Percobaan 4

Percobaan keempat menggunakan *query* awal “data mining”, dan diekspansi atau ditambah dengan kata “text”. Sehingga *query* akhir menjadi “data mining text”. Hasil pengujian percobaan empat ditunjukkan pada tabel 5.4.

Tabel 5.4 Hasil Pengujian Percobaan 4

Indikator	Query Awal	Query Akhir	Kenaikan
Precision	0,42	0,63	0,21
Recall	0,83	1	0,17
F Measure	0,56	0,77	0,21

Sumber: Pengujian



Gambar 5.4 Grafik Percobaan 4
Sumber: Pengujian

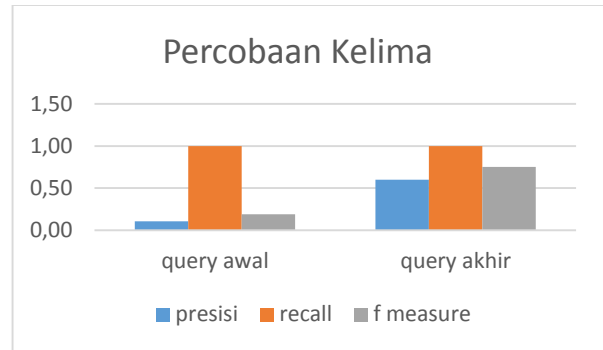
5. Percobaan 5

Percobaan kelima menggunakan *query* awal “algoritma”, dan diekspansi atau ditambah dengan kata “kompresi”. Sehingga *query* akhir menjadi “algoritma kompresi”. Hasil pengujian percobaan 5 ditunjukkan pada tabel 5.5.

Tabel 5.5 Hasil Pengujian Percobaan 5

Indikator	Query Awal	Query Akhir	Kenaikan
Precision	0,10	0,60	0,50
Recall	1	1	0,00
F Measure	0,19	0,75	0,56

Sumber: Pengujian



Gambar 5.5 Grafik Percobaan 5
Sumber: Pengujian

Berdasarkan kelima percobaan diperoleh hasil rata-rata kenaikan *precision*, *recall* dan *f-measure*. Berikut rekap hasil percobaan satu sampai lima ditunjukkan pada tabel 5.6

Tabel 5.6 Hasil Pengujian Percobaan 1 sampai 5

Percobaan	Kenaikan		
	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>
1	0,06	0	0,05
2	0,03	0	0,05
3	0,04	0	0,05
4	0,21	0,17	0,21
5	0,50	0	0,56

Sumber: Pengujian

Dari table 5.6 maka didapat rata-rata kenaikan *precision* sebesar 0,17, rata-rata kenaikan *recall* sebesar 0,03 dan rata-rata kenaikan *f-measure* sebesar 0,18.

5.2 Skenario 2

Pengujian pada skenario ini untuk mengetahui pengaruh penambahan jumlah kata saat menggunakan *query expansion*. Dalam menambahkan kata, pemilihan kata dilakukan secara acak dengan memperhatikan kesesuaian kata dengan *query* berdasarkan kesesuaian menurut penulis.

1. Percobaan 6

Pada percobaan ini, penulis menggunakan *query* “permainan game”, selanjutnya untuk penambahan satu kata ditambahkan *query* “budaya”, untuk penambahan dua kata ditambahkan dengan kata “budaya android”, dan untuk penambahan tiga kata menggunakan kata “budaya gobak sodor”. Hasil pengujian percobaan 5 ditunjukkan pada tabel 5.7.

Tabel 5.7 Hasil Pengujian Percobaan 6

Jumlah Kata	Precision	Recall	F-Measure
1 kata	0,62	1	0,77
2 kata	0,18	1	0,31
3 kata	0,13	1	0,22

Sumber: Pengujian

2. Percobaan 7

Pada percobaan ini, penulis menggunakan *query* “algoritma”, selanjutnya untuk penambahan satu kata ditambahkan *query* “kompresi”, untuk penambahan dua kata ditambahkan dengan kata “kompresi file”, dan untuk penambahan tiga kata menggunakan kata “kompresi file audio”. Hasil pengujian percobaan 7 ditunjukkan pada tabel 5.8.

Tabel Error! No text of specified style in document..8 Hasil Pengujian Percobaan 7

Jumlah Kata	Precision	Recall	F-Measure
1 kata	0,60	1	0,75
2 kata	0,63	0,71	0,67
3 kata	0,44	1	0,62

Sumber: Pengujian

3. Percobaan 8

Pada percobaan ini, penulis menggunakan *query* “klasifikasi text”, selanjutnya untuk penambahan satu kata ditambahkan *query* “jurnal”, untuk penambahan dua kata ditambahkan dengan kata “jurnal ilmiah”, dan untuk penambahan tiga kata menggunakan kata “jurnal ilmiah penelitian”. Hasil pengujian percobaan 8 ditunjukkan pada tabel 5.9.

Tabel 5.9 Hasil Pengujian Percobaan 8

Jumlah Kata	Precision	Recall	F-Measure
1 kata	0,15	1	0,27
2 kata	0,4	1	0,57
3 kata	0,36	1	0,53

Sumber: Pengujian

Dari tiga percobaan pada skenario 2 diperoleh data pada tabel 5.10.

Tabel 5.10 Hasil Pengujian Percobaan 6, 7 dan 8

Jumlah Penambahan	Precision	Recall	F-Measure
1 kata	0,62	1	0,77
	0,60	1	0,75
	0,13	1	0,22

2 kata	0,18	1	0,31
	0,63	0,71	0,67
	0,44	1	0,62
3 kata	0,13	1	0,22
	0,4	1	0,57
	0,36	1	0,53

Sumber: Pengujian

Dari tabel 5.10 diperoleh nilai rata-rata untuk setiap penambahan pada jumlah kata ditunjukkan pada tabel 5.11

Tabel 5.11 Rata-Rata Precision, Recall Dan F-Measure Terhadap Jumlah Kata

Jumlah Kata	Precision	Recall	F-Measure
1 kata	0,45	1,00	0,58
2 kata	0,42	0,90	0,53
3 kata	0,30	1,00	0,44

Sumber: Pengujian

Dari percobaan 6 sampai 8 diperoleh bahwa penambahan jumlah kata berpengaruh dalam nilai *precision*, *recall* dan *f-measure*. Selain itu, penambahan jumlah kata juga berpengaruh pada kenaikan *ranking* dokumen yang paling relevan.

5.3 Analilis Hasil Pengujian

Dari hasil pengujian diketahui bahwa *query expansion* dapat mempengaruhi nilai *precision*. Dari percobaan 1 sampai 5 diperoleh nilai *precision* yang cenderung naik jika dilakukan *query expansion*. Namun untuk nilai *recall* cenderung tetap. Kenaikan nilai *precision* dan tetapnya nilai *recall* mengakibatkan nilai *f-measure* meningkat. Namun nilai *precision*, *recall* dan *f-measure* ini belum tentu naik jika penambahan kata untuk *query* selanjutnya tidak berhubungan dengan *query* awal. Atau dengan ditambahkan kata, *query* baru menjadi *query* yang tidak mengandung makna. Pada pengujian ini, penambahan kata disesuaikan dengan *query* awal, sehingga *query* baru memiliki arti yang baru. Sebagai contoh *query* awal “algoritma” ditambah dengan kata “kompresi”, sehingga menjadi “algoritma kompresi” dan memiliki arti yang lebih spesifik dari pada *query* “algoritma”.

Penambahan jumlah kata berpengaruh terhadap nilai *precision*, *recall* dan *f-measure*. Semakin banyak kata yang ditambahkan, nilai *precision*, *recall* dan *f-measure* semakin naik. Hal ini dikarenakan *query* semakin spesifik dan jumlah dokumen yang terambil cenderung tetap. Penambahan jumlah kata ini akan menaikkan *ranking* dari dokumen yang relevan. Semakin banyak kata yang ditambahkan, *query*

menjadi semakin spesifik dan dokumen yang diinginkan urutannya akan naik.

6 PENUTUP

6.1 Kesimpulan

Kesimpulan dari hasil skripsi *Query Expansion* Pada Sistem Temu Kembali Informasi Dokumen Berbahasa Indonesia menggunakan *Pseudo Relevance Feedback* dengan Studi Kasus Perpustakaan Universitas Brawijaya sebagai berikut:

1. *Pseudo Relevance feedback* dapat digunakan sebagai metode untuk menerapkan *query expansion*. *Pseudo relevance feedback* hanya metode untuk menentukan dokumen-dokumen yang dianggap relevan dari dokumen yang telah di retrieve oleh sistem. Pada skripsi ini *pseudo relevance feedback* dikombinasikan dengan *Rocchio Relevance Feedback* untuk dapat menentukan *query* baru yang akan digunakan untuk proses *query expansion*.
2. Pengujian sistem *query expansion* menggunakan *pseudo relevance feedback* menghasilkan kenaikan rata-rata *precision* (akurasi) sebesar 0.17, kenaikan rata-rata *recall* sebesar 0.03 dan kenaikan rata-rata *f-measure* sebesar 0.18. Penambahan jumlah kata pada *query expansion* juga berpengaruh terhadap *precision*, *recall* dan *f-measure*.

6.2 Saran

Saran dari hasil penelitian ini untuk pengembangan lebih lanjut yaitu memperhatikan dan mencari metode lain dalam pemilihan kata baru sehingga kata-kata yang ditampilkan dan dipilih oleh *user* semakin baik dan memiliki kesamaan yang tinggi dengan *query* awal dari *user*.

DAFTAR PUSAKA

- Anita. (2012). Direct Term Feedback untuk Temu-Kembali Informasi Bahasa Indonesia Menggunakan Model Bahasa. *Institut Pertanian Bogor*.
- Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. New York: ACM Press Book.
- Elfron, M. (2007). Query Expansion and Dimensionality Reduction: Notions of Optimality in Rocchio Relevance Feedback and Latent Semantic Indexing. *Science Direct*.
- Fachruddin, M. (2011). Analisis dan Implementasi Pseudo Relevance Feedback dengan Kueri Expansion Menggunakan Term Selection Value. *Institut Teknologi Bandung*.
- Jajali, V., & Reza, M. (2010). Information retrieval with concept-based pseudo-relevance feedback in MEDLINE. *Springer-Verlag London Limited*.
- Kiftiyani, U. (2014). Perbandingan Algoritma Naive Bayes dan K Nearest Neighbor untuk Perangkingan Dokumen Berbahasa Arab. *PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER*.
- Maruhum, T. (2009). Analisis dan Implementasi Query Expansion Pada Information Retrieval Berdasarkan Penarikan Kesimpulan Dengan Fuzzy Rules. *Institut Teknologi Bandung*.
- Maulana, D. (2014). Question Answering System Berbasis Clustering pada Buku Pedoman PTIIK Dengan Menggunakan Algoritma Levenshtein Distance. *Program Teknologi Informasi dan Ilmu Komputer*.
- Selberg, E. W. (1997). Information Retrieval Advances using Relevance Feedback. *Departement of Computer Science and Engineering University of Washington*.
- UP, C. (2007). Relevance feedback and query. *Cambridge University Press*.
- Yugianus, P. (2013). PENGEMBANGAN SISTEM PENELUSURAN KATALOG PERPUSTAKAAN DENGAN METODE ROCCHIO RELEVANCE FEEDBACK. *Universitas Brawijaya*.
- Zafikri, A. (2008). Implementasi Metode Term Frequency Inverse Document Frequency (TF-IDF) pada Sistem Temu Kembali Informasi. *UNIVERSITAS SUMATERA UTARA*.
- Zhou, D., & dkk. (2013). Collaborative pseudo-relevance feedback. *Expert Systems with Applications*.
- Zhu, D., & Carterette, B. (2012). Improving Health Records Search Using Multiple Query Expansion Collections. *University of Delaware*.