

# Penerapan Model OKAPI BM25 Pada Sistem Temu Kembali Informasi

Rizqa Raaiqa Bintana<sup>1</sup>, Surya Agustian<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, FST UIN Suska Riau

Jl. HR Soeberantas km 11,5 Panam, Pekanbaru, Riau

e-mail: <sup>1</sup>brizqaraaiqa@yahoo.co.id, <sup>2</sup>sagustian@yahoo.com

## Abstrak

Pesatnya pertumbuhan jumlah dokumen dan informasi digital, menyebabkan proses menemukan dokumen yang diinginkan akan semakin sulit bila dilakukan secara manual. Oleh sebab itu, suatu sistem temu kembali informasi diperlukan sebagai mesin pencari dokumen yang relevan dengan apa yang diinginkan user. Penelitian ini membangun sebuah sistem temu kembali informasi dengan menerapkan model Okapi BM25. Dua tahapan utama yang dilakukan dalam membangun sistem temu kembali informasi dengan model Okapi BM25 ini, yaitu preprocessing dan penerapan model Okapi BM25. Tahapan preprocessing, yaitu koleksi dokumen (*corpus*), tokenisasi, linguistik preprocessing, dan indexing dengan membuat *inverted index*. Setelah tahapan preprocessing dilakukan, diterapkan model Okapi BM25 ke dalam sistem untuk menghitung nilai korelevansi masing-masing dokumen terhadap *query*. Sistem mengembalikan suatu daftar dokumen terurut *descending* berdasarkan nilai korelevansi yang diperoleh. Unjuk kerja sistem diukur dengan parameter *precision* dan *recall*. Dari hasil pengujian, untuk beberapa kata kunci yang diujikan performa sistem dinilai baik dengan *precision* 74% - 92% dan *recall* 100%. Sistem ini dikembangkan secara *portable* sehingga dapat diterapkan pada server lokal untuk memenuhi kebutuhan organisasi dalam menemukan dokumen sesuai dengan koleksi yang dimilikinya.

**Kata kunci :** Okapi BM25, Precision, Recall, Temu Kembali Informasi.

## Abstract

The rapid growth of information and digital documents has caused the process to retrieve desire documents becomes more difficult when performed manually. Therefore, an information retrieval system is needed as the search engine to find relevant documents as user desire. This research is to develop a information retrieval system based on OKAPI BM25 model. Two main steps would do in build an information retrieval system with Okapi BM25 model are preprocessing and applied the model to it. The preprocessing steps are collection of document (*corpus*), tokenization, linguistic preprocessing, and indexing by creating an *inverted index*. After preprocessing, applied Okapi BM25 model to the system to compute the relevant value between each document to user's query. The system retrieve a list of documents which in *descending* sort based on relevant value. System performance is measured by calculating *precision* and *recall*. Experiment on several query key, shows a good performance where *precision* value about 74%-92% and 100% *recall*. This system is *portable* so it can be applied to the local server to meet the needs of the organization in finding a document in accordance with its documents collection.

**Keywords :** Information Retrieval, Okapi BM25, Precision, Recall.

## 1. Pendahuluan

Seiring dengan pesatnya perkembangan dan kemajuan teknologi, telekomunikasi, dan informasi, menjadikan kebutuhan akan informasi meningkat dengan pesat. Contoh sederhananya, beraneka macam informasi saat ini tersedia di *internet*. Hal ini mengakibatkan bagaimana pengguna (*user*) dapat menemukan informasi yang relevan dengan kebutuhannya yang diekspresikan melalui *query*. Cara untuk memperoleh informasi yang relevan adalah melalui sistem komputasi yaitu dengan menggunakan sistem temu kembali informasi.

Sistem temu kembali informasi (*information retrieval system*) digunakan untuk menemukan kembali informasi (dokumen) yang relevan dengan kebutuhan pengguna (*query*) dari suatu kumpulan informasi secara otomatis melalui perhitungan matematis. Ada tiga model klasik untuk perhitungan matematis yang digunakan dalam temu kembali informasi, yaitu model boolean (teori aljabar boolean), model ruang vektor [1], dan model probabilistik [2].

Dalam model boolean [3, 8], kata (*term*) dalam sebuah *query* dihubungkan dengan menggunakan operator logika (ekspresi boolean) sehingga model ini tidak mendukung *partial matching* pada *query*. Dokumen yang dikembalikan adalah dokumen yang benar-benar sesuai dengan pernyataan boolean atau *query* yang diinputkan. Model boolean tidak melakukan perangkingan terhadap dokumen yang di-*retrieve* karena model ini tidak melakukan penghitungan nilai korelevansi antara dokumen terhadap *query*.

Model ruang vektor dan model probabilistik mendukung *partial matching* dan pengguna dapat mengekspresikan *query*-nya secara bebas. Kedua model ini melakukan penghitungan nilai kemiripan

(*similarity*) atau kerelevanan antara dokumen terhadap *query*. Sehingga perangkingan dokumen dapat dilakukan berdasarkan nilai kemiripan atau kerelevanan yang diperoleh. Pada model ruang vektor jumlah kemunculan suatu *term* dalam dokumen sangat berpengaruh dalam menentukan nilai *term frequency* (*tf*). Nilai *tf* akan berpengaruh dalam pembobotan dan penghitungan nilai *similarity* antara dokumen terhadap *query*. Namun, pada model probabilistik *term frequency* (*tf*) tidak berpengaruh dalam penghitungan nilai kerelevanan antara dokumen terhadap *query*.

Model Okapi BM25 (Stephen Robertson, Stephen Walker) merupakan perluasan dari model probabilistik yang melibatkan *tf* dalam penghitungan nilai kerelevanannya[3-5]. Model ini telah digunakan secara luas dan berhasil melewati suatu kumpulan informasi dan tugas pencarian, khususnya dalam aktifitas *TREC evaluations* [3]. Menurut Michael Speriosu dan Tetsuya Tashiro, dalam hasil penelitian mereka menyatakan bahwa kinerja model Okapi BM25 mengungguli *Language Modeling Algorithms*[6]. Untuk dapat menerapkan dan mengetahui nilai kesesuaian (*relevant*) model Okapi BM25 terhadap *query* dan dokumen yang dikembalikan, maka penellitian ini akan membahas tentang penerapan model Okapi BM25 pada sistem temu kembali informasi.

## 2. Information Retrieval

Standar ISO 2382/1, mendefinisikan *Information Retrieval* (IR) sebagai tindakan, metode dan prosedur untuk menemukan kembali data yang tersimpan, kemudian menyediakan informasi mengenai subyek yang dibutuhkan. Tindakan tersebut mencakup *text indexing*, *inquiry analysis*, dan *relevance analysis*. Data mencakup teks, tabel, gambar, ucapan, video, dan lainnya serta informasi mencakup pengetahuan terkait yang dibutuhkan untuk mendukung penyelesaian masalah, akuisisi pengetahuan, dan lainnya [7]. *Information retrieval system* bertujuan memenuhi kebutuhan informasi pengguna (*user*) dengan cara menemukan dan me-*retrieve* dokumen-dokumen yang relevan dengan *query* pengguna.

### 2.1. Pembuatan Index

Langkah-langkah pembuatan *inverted index*[3][4], yaitu:

1. Mengumpulkan dokumen yang akan di-*index* (dikenal dengan nama *corpus* atau koleksi dokumen).
2. Penghapusan format dan *markup* dari dalam dokumen.  
Pada tahap ini semua *tag markup* dan format khusus dihapus dari dokumen, terutama pada dokumen yang mempunyai banyak *tag markup* dan format seperti dokumen (X)HTML.
3. Pemisahan rangkaian kata (*tokenization*).  
Pada tahapan ini, seluruh kata di dalam kalimat, paragraf atau halaman dipisahkan menjadi *token* atau potongan kata tunggal atau *termmed word*. Tahapan ini juga akan menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua *token* ke bentuk huruf kecil (*lowercase*).
4. Melakukan *linguistic preprocessing* untuk menghasilkan daftar kata (*token* atau *term*) yang ternormalisasi. Dua hal yang dilakukan dalam tahap ini adalah:
  - a. Penyaringan (*filtration*)  
Pada tahapan ini ditentukan *term* mana yang akan digunakan untuk merepresentasikan dokumen sehingga dapat mendeskripsikan isi dokumen dan membedakan dokumen tersebut dari dokumen lain di dalam koleksi. Tahap ini biasanya melakukan pembuangan stop word (*term* yang sering muncul) dan *term* yang jarang sekali muncul, sesuai dengan hukum Zipf.
  - b. Konversi *term* ke bentuk akar (*stemming*)  
*Stemming* adalah proses konversi *term* ke bentuk akarnya. Seperti yang telah dijelaskan sebelumnya.
5. Mengindeks dokumen (*indexing*). Sebuah indeks selalu memetakan kembali dari setiap *term* ke dokumen dimana *term* tersebut muncul. Pengindeksan dilakukan dengan membuat *inverted index* (atau disebut juga *inverted file*) yang terdiri dari *dictionary* dan *postings*. *Inverted index* merupakan konversi dari dokumen asli yang mengandung sekumpulan kata ke dalam daftar kata (*dictionary*) yang berasosiasi dengan dokumen terkait dimana kata-kata tersebut muncul (*postings*). *Dictionary* adalah daftar kata yang diperoleh dari hasil pengindeksan dokumen.

### 2.2. Pembobotan Kata (Term Weighting)

Setiap *term* yang telah di-*index* diberikan bobot sesuai dengan skema pembobotan yang dipilih, apakah pembobotan lokal, global atau kombinasi keduanya. Jika menggunakan pembobotan lokal maka, pembobotan *term* diekspresikan sebagai *tf* (*term frequency*). Namun, jika pembobotan global yang digunakan maka, pembobotan *term* didapatkan melalui nilai *idf* (*inverse document frequency*). Beberapa aplikasi juga ada yang menerapkan pembobotan kombinasi keduanya yaitu, dengan mengalikan bobot lokal dan global (*tf . idf*) [3, 5].

$$idf(t) = \log\left(\frac{N}{df(t)}\right) \quad (1)$$

Keterangan:

N : jumlah dokumen dalam *corpus*.

$df_t$  : *document frequency* atau jumlah dokumen dalam *corpus* yang mengandung *term*.

### 2.3. Model OKAPI BM25

Model Okapi BM25 merupakan kombinasi model probabilistik dan pembobotan lokal (*term frequency*) [4]. Nilai paling sederhana untuk dokumen  $d$  adalah hanya dengan menghitung bobot  $idf$  dari *term* yang terdapat pada *query* yang diinputkan oleh *user*.

$$RSV(d) = \sum_{t \in q} \log \frac{N}{df(t)} \quad (2)$$

Persamaan 2 dapat dikembangkan dengan memfaktorkan frekuensi masing-masing *term* dan panjang dokumen seperti yang terjadi pada model Okapi BM25. Persamaan model Okapi BM25 ditunjukkan pada Persamaan 3 [3].

$$RSV(d) = \sum_{t \in q} \log \left[ \frac{N}{df(t)} \right] \cdot \frac{(k_1+1) \cdot tf(td)}{k_1((1-b)+b \times (L_d/L_{ave})) + tf(td)} \cdot \frac{(k_3+1) \cdot tf(tq)}{k_3 + tf(tq)} \quad (3)$$

Persamaan 3 digunakan jika *user* memasukkan atau menggunakan *query* yang panjang atau dalam bentuk paragraf sehingga memungkinkan terdapat *term* yang ganda pada *query*. Namun jika *query* yang dimasukkan dalam bentuk informasi yang pendek (tidak terdapat *term* yang ganda pada *query*), maka dapat digunakan Persamaan 4 [3].

$$RSV(d) = \sum_{t \in q} \log \left[ \frac{N}{df(t)} \right] \cdot \frac{(k_1+1) \cdot tf(td)}{k_1((1-b)+b \times (L_d/L_{ave})) + tf(td)} \quad (4)$$

Keterangan:

RSV(d) : *retrieval status value* untuk suatu dokumen.

N : jumlah dokumen dalam *corpus*.

df<sub>t</sub> : jumlah dokumen dalam *corpus* yang mengandung *term* pada *query* q.

tf<sub>td</sub> : frekuensi *term* dalam dokumen d.

tf<sub>tq</sub> : frekuensi *term* dalam *query* q.

L<sub>d</sub> : panjang dokumen d.

L<sub>ave</sub> : rata-rata panjang dokumen secara keseluruhan.

k<sub>1</sub> : konstanta frekuensi *term*, nilainya  $1,2 \leq k_1 \leq 2$ .

k<sub>3</sub> : konstanta frekuensi *term*, nilainya  $1,2 \leq k_3 \leq 2$ .

b : konstanta panjang dokumen, nilainya  $0 \leq b \leq 1$ . (0,75).

### 2.4. Evaluasi Unjuk Kerja Retrieval

Sistem IR mengembalikan sekumpulan dokumen sebagai jawaban dari *query* pengguna. Terdapat dua kategori dokumen yang dihasilkan oleh sistem IR terkait pemrosesan *query*, yaitu *relevant documents* (dokumen yang relevan dengan *query*) dan *retrieved documents* (dokumen yang diterima pengguna). Secara garis besar, dua ukuran dasar dan umum yang digunakan untuk mengukur kualitas *information retrieval* adalah *precision* dan *recall*. *Precision* (P) mengevaluasi kemampuan sistem IR untuk menemukan kembali dokumen *top-ranked* yang paling relevan [3, 7], dan didefinisikan sebagai rasio antara jumlah dokumen yang di-*retrieve* yang relevan dengan *query* pengguna terhadap jumlah dokumen yang di-*retrieve*.

$$Precision = \frac{\{relevant docs\} \cap \{retrieved docs\}}{\{retrieved docs\}} \quad (5)$$

*Recall* (R) mengevaluasi kemampuan sistem IR untuk menemukan semua dokumen yang relevan dari dalam koleksi dokumen, dan didefinisikan sebagai rasio antara jumlah dokumen yang relevan dengan *query* pengguna yang di-*retrieve* terhadap jumlah dokumen yang relevan di dalam *database* (koleksi dokumen), yang di-*retrieve* dan tidak di-*retrieve*.

$$Recall = \frac{\{relevant docs\} \cap \{retrieved docs\}}{\{relevant docs\}} \quad (6)$$

Persamaan 5 dan Persamaan 6 dapat diperjelas melalui Tabel 1 di bawah ini.

Tabel 1. *Precision* dan *Recall* [3]

	<b>Relevant</b>	<b>Nonrelevant</b>
<b>Retrieved</b>	<i>true positives (tp)</i>	<i>false positives (fp)</i>
<b>Not retrieved</b>	<i>false negatives (fn)</i>	<i>true negatives (tn)</i>

Maka:  $P = tp / (tp + fp)$

$R = tp / (tp + fn)$

## 3. Pengujian dan Analisa

### 3.1. Rencana Pengujian

Pengujian dilakukan dengan menginputkan 3 *query* yang berbeda untuk 100 koleksi dokumen (*corpus*) yang sudah dibaca isinya, sesuai dengan [9], untuk melihat bagaimana performa model OKAPI BM25 terhadap model ruang vektor.

Tabel 2. Daftar *query* yang diinputkan

No	Query	Jumlah dokumen relevan dalam <i>corpus</i>
1	Penyelundupan kayu	21
2	Candi borobudur	12
3	Limbah buyat	14

### 3.2. Hasil Pengujian

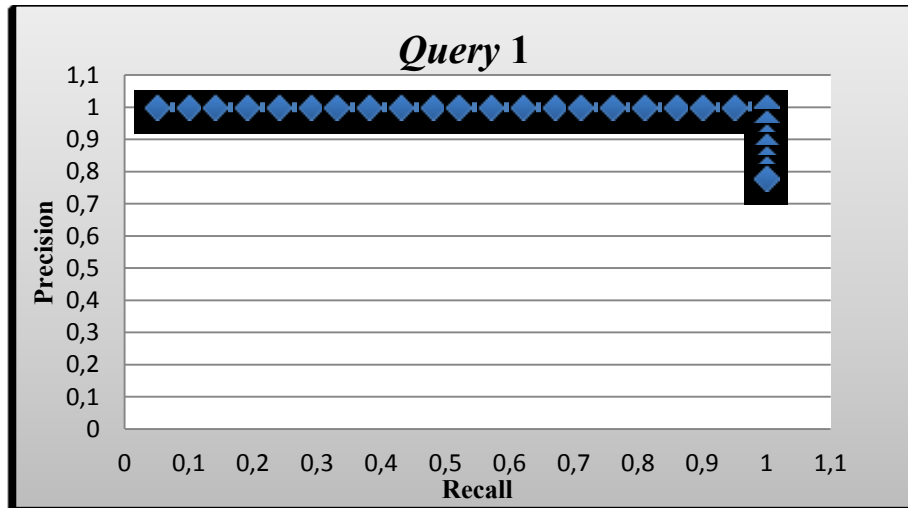
Tabel 3-5 menunjukkan evaluasi relevansi dokumen yang dihasilkan oleh sistem yang dilakukan secara subjektif, dengan membaca isi dokumen per dokumen, kemudian dinilai apakah relevan atau tidak dengan *query* yang diberikan. Teknik pengukuran ini dibutuhkan untuk mengukur seberapa jauh panjang pelacakan dokumen relevan yang bisa dihasilkan oleh mesin pencari, terutama bila hasil pencarian jumlahnya sangat besar (misalnya ribuan).

Angka-angka dalam tabel 3, 5 dan 7 diinterpretasikan sebagai grafik interpolasi Precision-Recall seperti ditunjukkan pada Gambar 1-3. Performa sistem dapat dilihat dari seberapa panjang garis mendatar atau landai, sampai ia menurun dengan drastis. Semakin panjang garis tersebut, maka semakin banyak dokumen relevan yang dihasilkan pada peringkat-peringkat teratas, sehingga panjang pelacakan (penelusuran) dokumen relevan oleh user menjadi lebih singkat.

Secara keseluruhan, pengukuran precision-recall dari sistem dapat dihitung dengan menggunakan data pada Tabel 4, 6, dan 8, dengan membandingkan antara dokumen relevan yang ter-*retrieve*, sesuai dengan persamaan (5) dan (6). Hasil akhirnya dapat dinyatakan dalam persen.

Tabel 3. *Precision (P)* dan *recall (R)* untuk *query* 1

Rank	Id doc	Relevant?	Precision (P) ; Recall (R)
1	32	yes	$P = 1/1 = 1$ ; $R = 1/21 = 0.05$
2	24	yes	$P = 2/2 = 1$ ; $R = 2/21 = 0.1$
3	35	yes	$P = 3/3 = 1$ ; $R = 3/21 = 0.14$
4	26	yes	$P = 4/4 = 1$ ; $R = 4/21 = 0.19$
5	21	yes	$P = 5/5 = 1$ ; $R = 5/21 = 0.24$
6	33	yes	$P = 6/6 = 1$ ; $R = 6/21 = 0.29$
7	37	yes	$P = 7/7 = 1$ ; $R = 7/21 = 0.33$
8	38	yes	$P = 8/8 = 1$ ; $R = 8/21 = 0.38$
9	22	yes	$P = 9/9 = 1$ ; $R = 9/21 = 0.43$
10	29	yes	$P = 10/10 = 1$ ; $R = 10/21 = 0.48$
11	40	yes	$P = 11/11 = 1$ ; $R = 11/21 = 0.52$
12	28	yes	$P = 12/12 = 1$ ; $R = 12/21 = 0.57$
13	20	yes	$P = 13/13 = 1$ ; $R = 13/21 = 0.62$
14	23	yes	$P = 14/14 = 1$ ; $R = 14/21 = 0.67$
15	36	yes	$P = 15/15 = 1$ ; $R = 15/21 = 0.71$
16	31	yes	$P = 16/16 = 1$ ; $R = 16/21 = 0.76$
17	30	yes	$P = 17/17 = 1$ ; $R = 17/21 = 0.81$
18	25	yes	$P = 18/18 = 1$ ; $R = 18/21 = 0.86$
19	34	yes	$P = 19/19 = 1$ ; $R = 19/21 = 0.9$
20	39	yes	$P = 20/20 = 1$ ; $R = 20/21 = 0.95$
21	27	yes	$P = 21/21 = 1$ ; $R = 21/21 = 1$
22	68	no	$P = 21/22 = 0.95$ ; $R = 21/21 = 1$
23	49	no	$P = 21/23 = 0.91$ ; $R = 21/21 = 1$
24	72	no	$P = 21/24 = 0.88$ ; $R = 21/21 = 1$
25	95	no	$P = 21/25 = 0.84$ ; $R = 21/21 = 1$
26	80	no	$P = 21/26 = 0.81$ ; $R = 21/21 = 1$
27	84	no	$P = 21/27 = 0.78$ ; $R = 21/21 = 1$



Gambar 1. Grafik tradeoff pada query 1

Tabel 4. Precision-Recall keseluruhan untuk query 1

	<i>Relevant</i>	<i>Nonrelevant</i>
<i>Retrieved</i>	21	6
<i>Not retrieved</i>	0	73

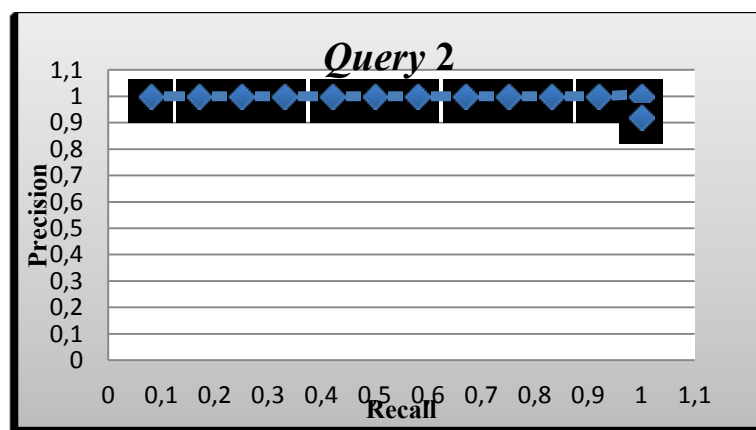
$$\text{Precision} \rightarrow P = tp / (tp + fp) = 21 / (21+6) = 21/27 = 0,78$$

$$\text{Recall} \rightarrow R = tp / (tp + fn) = 21 / (21+0) = 21/21 = 1$$

Pengujian untuk query 1 “penyelundupan kayu” dapat mengembalikan seluruh dokumen relevan pada korpus (21 dokumen), namun mengembalikan juga dokumen tidak relevan sebanyak 6 dokumen. Dengan demikian nilai recall-nya adalah 1 (100%). Tetapi precision tidak mencapai 100% karena ada dokumen tidak relevan yang dikembalikan.

Tabel 5. Precision (P) dan recall (R) untuk query 2

Rank	Id doc	Relevant?	Precision (P) ; Recall (R)
1	66	yes	P = 1/1 = 1 ; R = 1/12 = 0.08
2	65	yes	P = 2/2 = 1 ; R = 2/12 = 0.17
3	56	yes	P = 3/3 = 1 ; R = 3/12 = 0.25
4	64	yes	P = 4/4 = 1 ; R = 4/12 = 0.33
5	60	yes	P = 5/5 = 1 ; R = 5/12 = 0.42
6	59	yes	P = 6/6 = 1 ; R = 6/12 = 0.5
7	57	yes	P = 7/7 = 1 ; R = 7/12 = 0.58
8	61	yes	P = 8/8 = 1 ; R = 8/12 = 0.67
9	62	yes	P = 9/9 = 1 ; R = 9/12 = 0.75
10	58	yes	P = 10/10 = 1 ; R = 10/12 = 0.83
11	63	yes	P = 11/11 = 1 ; R = 11/12 = 0.92
12	67	yes	P = 12/12 = 1 ; R = 12/12 = 1
13	68	no	P = 12/13 = 0.92 ; R = 12/12 = 1



Gambar 2. Grafik tradeoff pada query 2

Tabel 6. Precision-Recall keseluruhan untuk *query 2*

	<b>Relevant</b>	<b>Nonrelevant</b>
<b>Retrieved</b>	12	1
<b>Not retrieved</b>	0	87

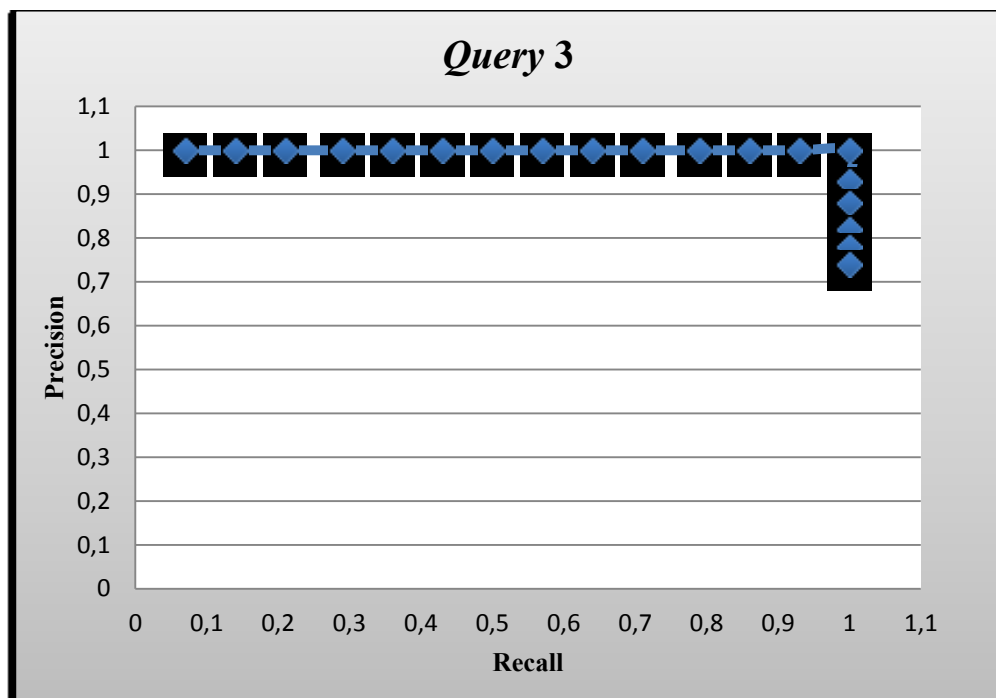
$Precision \rightarrow P = tp / (tp + fp) = 12 / (12+1) = 12/13 = 0,92$

$Recall \rightarrow R = tp / (tp + fn) = 12 / (12+0) = 12/12 = 1$

Sedangkan pengujian untuk *query 2* "candi borobudur" dapat mengembalikan seluruh dokumen relevan pada korpus (12 dokumen), dan hanya mengembalikan satu dokumen tidak relevan. Dengan demikian nilai *recall*-nya adalah 1 (100%). *Precision* mencapai 92% lebih tinggi dari *query* sebelumnya.

Tabel 7. *Precision (P)* dan *recall (R)* untuk *query 3*

<i>Rank</i>	<i>Id doc</i>	<i>Relevant?</i>	<i>Precision (P) ; Recall (R)</i>
1	42	yes	P = 1/1 = 1 ; R = 1/14 = 0.07
2	54	yes	P = 2/2 = 1 ; R = 2/14 = 0.14
3	48	yes	P = 3/3 = 1 ; R = 3/14 = 0.21
4	53	yes	P = 4/4 = 1 ; R = 4/14 = 0.29
5	52	yes	P = 5/5 = 1 ; R = 5/14 = 0.36
6	43	yes	P = 6/6 = 1 ; R = 6/14 = 0.43
7	55	yes	P = 7/7 = 1 ; R = 7/14 = 0.5
8	51	yes	P = 8/8 = 1 ; R = 8/14 = 0.57
9	49	yes	P = 9/9 = 1 ; R = 9/14 = 0.64
10	44	yes	P = 10/10 = 1 ; R = 10/14 = 0.71
11	41	yes	P = 11/11 = 1 ; R = 11/14 = 0.79
12	46	yes	P = 12/12 = 1 ; R = 12/14 = 0.86
13	47	yes	P = 13/13 = 1 ; R = 13/14 = 0.93
14	45	yes	P = 14/14 = 1 ; R = 14/14 = 1
15	69	no	P = 14/15 = 0.93 ; R = 14/14 = 1
16	72	no	P = 14/16 = 0.88 ; R = 14/14 = 1
17	50	no	P = 14/17 = 0.82 ; R = 14/14 = 1
18	74	no	P = 14/18 = 0.78 ; R = 14/14 = 1
19	99	no	P = 14/19 = 0.74 ; R = 14/14 = 1



Gambar 3. Grafik *tradeoff* pada *query 3*

Tabel 8. Precision-Recall keseluruhan untuk *query* 3

	<b>Relevant</b>	<b>Nonrelevant</b>
<b>Retrieved</b>	14	5
<b>Not retrieved</b>	0	81

$$\text{Precision} \rightarrow P = tp / (tp + fp) = 14 / (14+5) = 14/19 = 0,74$$

$$\text{Recall} \rightarrow R = tp / (tp + fn) = 14 / (14+0) = 14/14 = 1$$

Pengujian untuk *query* 3 “limbah buyat” juga dapat mengembalikan semua dokumen relevan pada korpus (14 dokumen), sehingga nilai *recall*-nya adalah 1 (100%). Sedangkan dokumen tidak relevan dikembalikan sebanyak 5 dokumen. Dengan demikian nilai *precision*-nya juga tidak dapat mencapai 100% karena ada dokumen tidak relevan yang dikembalikan pada hasil pencarian.

### 3.3. Analisa Hasil Pengujian

Berdasarkan pengujian dan data yang diperoleh, dapat disebutkan bahwa:

1. Performa sistem sangat baik untuk koleksi yang digunakan, ditunjukkan dengan grafik interpolasi Precision-Recall (Gambar 1,2,3) yang mendatar pada nilai 1 sampai Recall mencapai 100%. Artinya dokumen relevan terurut pada ranking teratas tanpa diselingi oleh dokumen-dokumen yang tidak relevan.
2. Untuk model Okapi BM25, persentase kualitas *retrieval* (sesuai dengan Tabel 4, 6, 8) untuk *query* 1 terhadap jumlah dokumen yang berhasil di-*retrieve* oleh sistem, yaitu *precision* 78% dan *recall* 100%. Sedangkan persentase kualitas *retrieval* untuk *query* 2 berdasarkan jumlah dokumen yang berhasil di-*retrieve*, yaitu *precision* 92% dan *recall* 100%. Dan persentase kualitas *retrieval* untuk *query* 3 berdasarkan jumlah dokumen yang di-*retrieve*, yaitu *precision* 74% dan *recall* 100%.
3. Persentase kualitas *retrieval* untuk jumlah dokumen relevan yang berhasil di-*retrieve* oleh sistem pada model Okapi BM25, yaitu *precision* 74% - 92% (0,74 – 0,92) dan *recall* 100%. Secara keseluruhan, ketepatan hasil pencarian sudah cukup baik.

### 4. Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini, yaitu:

1. Sistem IR (*information retrieval*) menyediakan sejumlah kemungkinan respon atau jawaban (berupa dokumen) yang dianggap relevan dan membiarkan pengguna memilih mana yang benar atau tepat menurut kebutuhannya (*query*).
2. Kemampuan sistem IR dengan model Okapi BM25 dalam me-*retrieve* dokumen yang relevan dengan *query* pengguna dinilai baik berdasarkan nilai *precision* yang cukup tinggi, dan berada pada peringkat teratas pelacakan.
3. Sistem dapat mengembalikan seluruh dokumen relevan (*recall* mencapai 100%), dengan sedikit false positif (dokumen tidak relevan).
4. Sistem IR dengan model Okapi BM25 ini dapat digunakan untuk situs-situs pencarian lokal organisasi, seperti perpustakaan digital (*digital library*), arsip dokumen hukum, teknik, dan sebagainya

### Referensi

- [1] Salton, Gerard, dan Christopher Buckley. “Term-Weighting Approaches In Automatic Text Retrieval,” *Department of Computer Science, Cornell University*, 19 November 1987. (Rev. 26 Januari 1988).
- [2] Robertson, Stephen, dan Hugo Zaragoza. “The Probabilistic Relevance Framework: BM25 and Beyond,” *Foundation and Trends in Information Retrieval*. Vol. 3, No. 4, hal. 333-389, 2009.
- [3] Manning, Christopher D., Prabhakar Raghavan, dan Hinrich Schütze. “*An Introduction to Information Retrieval*”. Cambridge, England: Cambridge University Press, 2009.
- [4] Garcia, E. “A Tutorial on Okapi BM25,” 30 Juni 2011. (Rev. 2 Agustus 2011).
- [5] Garcia, E “*Document Indexing Tutorial for Information Retrieval Students and Search Engine Marketers*” Desember. 2005. [Online] Available <http://www.miiisita.com/information-retrieval-tutorial/indexing.html>, (20 Oktober 2011)
- [6] Speriosu, Michael, dan Tetsuya Tashiro. “Comparison of Okapi BM25 and Language Modeling Algorithms for NTCIR-6,” *Justsystems Corporation*, 14 September 2006. (Rev. 6 Desember 2007).
- [7] J, Cios Krzysztof, Pedrycz W., Swiniarski R.W., dan Kurgan L.A. “*Data Mining: A Knowledge Discovery Approach*”. Springer, 2007.
- [8] Fox, E, dkk “*Chapter 15: Extended Boolean Models*” [Online] Available <http://orion.lcg.ufrj.br/Dr.Dobbs/books/book5/chap15.htm>, diakses 26 Oktober 2011.
- [9] Suryani, Irma. “Tugas Akhir: Sistem Temu Kembali Informasi (*Information Retrieval*) Dengan Metode *Term Frequency - Inverse Document Frequency* (TF-IDF) Menggunakan Model *Neural Network* (NN)”. Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN SUSKA Riau. Pekanbaru. 2012.