

IMPLEMENTASI DAN PERBANDINGAN METODE OKAPI BM25 DAN PLSA PADA APLIKASI *INFORMATION RETRIEVAL*

Jasman Pardede^[1], Milda Gustiana Husada^[1], Rizky Riansyah^[1]

^[1]Jurusan Teknik Informatika, Fakultas Teknik Industri, Itenas Bandung
Jln. PHH. Mustopha No.23 Bandung 40124 Telp. 022.772215
jasman@itenas.ac.id, mghusada@itenas.ac.id, rzkyrs@gmail.com

ABSTRAK

Information retrieval adalah sebuah ilmu yang mempelajari metode untuk mengambil kembali informasi yang tersimpan dari berbagai sumber. Metode yang digunakan dalam penelitian ini adalah metode *Okapi BM25* (BM25) dan *Probabilistic Latent Semantic Analysis* (PLSA). Metode BM25 berfungsi untuk meranking dokumen dengan menyesuaikan *query* yang dimasukan, sedangkan PLSA merupakan metode pengklasteran berbasis topik. Metode BM25 lebih mudah diterapkan, karena BM25 mengacu pada jumlah kalimat dan jumlah kata yang sesuai dengan *query*, sedangkan PLSA membutuhkan topik dalam penerapannya. Hasil pengujian dari penelitian ini menunjukkan jumlah *recall* tetap pada nilai 100% untuk kedua metode, sedangkan untuk nilai *precision* dan *F-Measure* metode BM25 memiliki nilai yang lebih besar dibanding metode PLSA. Rata-rata nilai *F-measure* pada metode BM25 yaitu 61,649 dan rata-rata nilai *F-measure* metode PLSA adalah 56,8877. Apabila dokumen yang tidak sesuai dengan *query* yang didapatkan oleh sistem pencarian semakin banyak, maka nilai *precision*-nya pun akan semakin kecil. Hasil perankingan tersebut akan diurutkan berdasarkan bobot dokumen tertinggi terhadap *query*.

Kata kunci : *Information retrieval, BM25, PLSA, Query.*

ABSTRACT

Information retrieval is a knowledge concerning several methods to recapture information from various relevant resources or collection of information resources. Method used to run this application are *Okapi BM25* (BM25) and *Probabilistic Latent Semantic Analysis* (PLSA). BM25 is a method used to rank documents which are appropriated with its query that is inserted by its user. Meanwhile, PLSA is a clustering method based on topic. BM25 is easier to be used because BM25 method only relies on numbers of sentences and words which are suitable to query, while PLSA needs topic in its implementation. The result of examination in this research revealed the total numbers of recall are keep at 100%. for both methods. Meanwhile for precision value and F measure of BM25 methodology has higher value. Meant the average of f- measure value in BM25 methodology is 61,649 and the average of f-measure in PLSA methodology is 56,887. The more document that are not appropriate with query that the system find, the smaller precision value will be get. Ranking result will be ranked based on the highest score of document toward query.

Key word : *Information retrieval, BM25, PLSA, Query.*

Latar Belakang

Information retrieval (IR) adalah ilmu yang mempelajari tentang metode-metode yang bertujuan untuk mengambil kembali informasi yang tersimpan dari berbagai sumber (*resources*) yang relevan atau koleksi sumber informasi yang dicari dan dibutuhkan. Dengan diterapkannya IR diharapkan bahwa pencarian informasi akan lebih akurat dan lebih mudah.

Dalam melakukan pencarian informasi *user* dapat melakukannya dengan membaca dokumen-dokumen yang dibutuhkan, tetapi proses tersebut tidaklah efisien untuk mendapatkan informasi yang dibutuhkan. Hal ini dikarenakan waktu yang dibutuhkan akan lebih lama dalam mencari informasi, dan informasi yang didapatkan pun tidak akan maksimal. Contoh penerapan IR dalam penelitian ini adalah pencarian dokumen.

Di dalam IR terdapat metode-metode yang berfungsi untuk mencari dokumen berdasarkan *query* yang dibutuhkan oleh *user*. Diantaranya ialah BM25, *Probabilistic Latent Semantic Analysis* (PLSA), *Association Rules* dan lain sebagainya. Dengan pencarian dokumen *user* akan dengan mudah menentukan dokumen mana yang diinginkan.

BM25 ini merupakan formula terbaik dalam kelas *Best Match*, dikarenakan formula ini efektif dan memiliki ketepatan dalam mengurutkan dokumen berdasarkan *query* yang dicari. Metode lainnya adalah PLSA, PLSA adalah sebuah metode pendekatan probabilitas untuk dua model seperti kata dan dokumen. Metode ini biasanya digunakan dalam aplikasi IR, Pengolahan Bahasa Alami, Sistem Pembelajaran dengan Data Teks, dan lain-lain. Dalam menjalankan kedua metode

tersebut terdapat beberapa proses yang harus dilakukan terlebih dahulu, yaitu *reading file* (*.doc, *.docx dan *.pdf) dan proses *preprocessing* yang meliputi *tokenizing*, *stopword removal* dan *stemming*. Pada tahap *preprocessing* tersebut melibatkan dokumen dan *query* yang dimasukan *user*. Banyaknya kemunculan kata yang sesuai dengan *query* yang dimasukan akan dihitung dan disesuaikan dengan persamaan yang terdapat pada metode BM25 dan PLSA. Dalam menjalankan metode BM25 dokumen maupun *query* akan berbentuk angka yang akan disesuaikan berdasarkan persamaan metode tersebut, sedangkan pada metode PLSA dokumen dan *query* yang dimasukan berbentuk matrix-matrix yang disesuaikan dengan persamaan metode PLSA.

Berdasarkan permasalahan tersebut metode BM25 dan PLSA akan digunakan dalam membangun aplikasi IR pencarian dokumen berdasarkan *query* yang dimasukan.

Rumusan Masalah

Berdasarkan latar belakang yang sudah dikemukakan, terdapat beberapa rumusan masalah sebagai berikut.

1. Bagaimana memahami perhitungan pada metode BM25 dan PLSA.
2. Bagaimana menerapkan *preprocessing* pada sistem.
3. Bagaimana menerapkan dan membandingkan hasil metode BM25 dan PLSA pada aplikasi *information retrieval* pada koleksi dokumen.

Tujuan

Tujuan dari penelitian ini adalah untuk melakukan analisis dan membangun

aplikasi pencarian dokumen dengan menggunakan metode BM25 dan PLSA, serta membandingkan kedua hasil dari metode tersebut.

Batasan Masalah

Di dalam penelitian ini terdapat masalah-masalah yang dibatasi antara lain sebagai berikut.

1. Dokumen yang digunakan merupakan dokumen yang berbahasa Indonesia.
2. *Query* yang digunakan dalam proses pencarian informasi adalah bahasa Indonesia yang baik dan benar.
3. Algoritma yang digunakan pada proses *stemming* adalah algoritma Nazief dan Adriani.
4. Jenis dokumen yang dapat melakukan proses pencarian ini adalah dokumen yang terdapat pada komputer yang memiliki format *.doc, *.txt, *.docx dan *.pdf.
5. Lamanya waktu pencarian dokumen ditentukan oleh banyaknya dokumen dan kalimat yang terdapat pada suatu koleksi dokumen.

Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Metodologi Pengumpulan Data

Pengumpulan data dilakukan dengan cara mempelajari teori-teori dari buku-buku dan juga data-data tertulis yang sudah ada.

2. Metodologi Pengembangan Sistem

Dalam penelitian ini metodologi pengembangan sistem yang digunakan ialah Metodologi *Prototype*. Metodologi ini merupakan pengembangan yang cepat dan didasarkan pada konsep *working model*.

Protoyping membantu dalam menemukan kebutuhan di tahap awal pengembangan, terutama jika *user* tidak yakin di mana masalah berasal. Selain itu *protoyping* juga berguna sebagai alat untuk mendesain dan memperbaiki *user interface* – bagaimana sistem akan terlihat oleh orang-orang yang menggunakannya. Berikut ini merupakan tahapan-tahapan dalam *prototyping*:

1. Pengumpulan kebutuhan sistem.
2. Membangun *prototyping*.
3. Evaluasi *prototyping*.
4. Pengkodean sistem.
5. Pengujian sistem.
6. Evaluasi sistem.
7. Penggunaan sistem.

Landasan Teori

Pada penelitian ini digunakan beberapa landasan teori sebagai acuan pembuatan aplikasi pencarian dokumen menggunakan metode BM25 dan PLSA.

***Information Retrieval* ^[5]**

Information Retrieval (IR) adalah seni dan ilmu dalam mencari informasi pada dokumen, mencari untuk dokumen mereka sendiri, mencari untuk *metadata* dengan gambaran berbentuk dokumen, atau mencari dalam *database*, apakah itu hubungan *database* yang berdiri sendiri atau hiperteks jaringan *database* seperti internet atau intranet, untuk teks, suara, gambar atau data. *Information Retrival System (IRS)* berfungsi untuk mengatur ledakan informasi dalam *literatur* ilmiah dalam beberapa dekade terakhir. Banyak universitas dan perpustakaan umum menggunakan *IR* sistem untuk menyediakan akses ke buku, jurnal, dan dokumen lainnya. *IR* sistem sering kali dihubungkan kepada objek dan *query*.

Tipe-tipe dari *user* yang menggunakan *Information Retrival System* dibedakan menjadi 5 yaitu *Actual User*, *Potential User*, *Expected User*, *Beneficially Users*, dan *Real User (End User, Intermediary, Novice Users, Expert User)*. Dengan tipe-tipe yang ada, kita dapat mengetahui sampai sejauh mana tingkatan *user* tentang pemahamannya terhadap proses penelusuran informasi.

Berikut ini merupakan bagian-bagian yang terdapat pada *information retrieval system*

1. **Text Operations** (operasi terhadap teks) yang meliputi pemilihan kata-kata dalam *query* maupun dokumen dalam pentransformasian dokumen atau *query* menjadi *terms index* (indeks dari kata-kata).
2. **Query Formulation** (formulasi terhadap *query*) yang memberi bobot pada indeks kata-kata *query*.
3. **Ranking**, mencari dokumen-dokumen yang relevan terhadap *query* dan mengurutkan dokumen tersebut berdasarkan kesesuaiannya dengan *query*.
4. **Indexing**, membangun data indeks dari koleksi dokumen dilakukan terlebih dahulu sebelum pencarian dokumen, *information retrieval* menerima *query* dari *user*, kemudian melakukan perankingan terhadap koleksi berdasarkan kesesuaiannya dengan *query*. Hasil perankingan yang diberikan kepada *user* merupakan dokumen yang relevan dengan *query*, namun relevansi dokumen terhadap suatu *query* merupakan penilaian *user* yang subjektif dan dipengaruhi banyak faktor.

Preprocessing

Preprocessing text dilakukan untuk tujuan penyeragaman dan kemudahan pembacaan (Aji P., Baizal SSi. and Firdaus S.T., 2011). *Preprocessing* terdiri dari beberapa tahapan, Tahapan-tahapan *preprocessing* adalah *tokenizing*, *stopword removal* dan *stemming*. Pada tahapan *stemming* algoritma yang digunakan adalah algoritma Nazief dan Adriani [6].

Metode BM25 [1,2]

Okapi BM25 atau yang biasa disebut dengan BM25 adalah suatu sistem perankingan yang digunakan untuk mengurutkan hasil kecocokan terhadap dokumen-dokumen, berdasarkan kata kunci yang dicarinya. BM25 ini merupakan formula terbaik dalam kelas *best match*, dikarenakan formula ini efektif dan memiliki ketepatan dalam mengurutkan dokumen berdasarkan *query* yang dicari. Berikut ini merupakan persamaan metode BM25. Pada persamaan metode BM25 nilai k_1 , k_3 dan nilai b merupakan parameter atau nilai konstanta [1,2,3].

$$BM25 = \sum_{t \in Q} \log_{10} \frac{(N - n_t)}{n_t} \cdot \frac{(k_1 + 1) f_{d,t}}{K + f_{d,t}} \cdot \frac{(k_3 + 1) f_{q,t}}{k_3 + f_{q,t}} \quad (1)$$

Dimana :

- Q : Input user (*Query*).
- N : Jumlah kalimat pada dokumen.
- n_t : Jumlah *term* yang mengandung *query*.
- $f_{d,t}$: Jumlah *term frequency*.
- $f_{q,t}$: Jumlah *query frequency*.

$$K = k_1 \cdot \left((1 - b) + \frac{b \cdot dl_d}{avl} \right) \quad (2)$$

Dimana :

- dl_d : Jumlah kalimat pada dokumen.

avl : Rata-rata panjang dokumen.
 k_1 : 1,2.
 b : 0.75.
 k_3 : 1000.

Probabilistic Latent Semantic Analysis (PLSA) [4]

PLSA adalah sebuah metode pendekatan probabilitas untuk dua model seperti kata dan dokumen. PLSA merupakan penyempurnaan dari metode *Latent Semantic Analysis* (LSA). Metode ini merupakan teknik *information retrieval* yang berfungsi untuk menganalisis dua keterhubungan kejadian data yang berdasarkan *model statistic* yang disebut *aspect model*. *Aspect model* didefinisikan sebagai sebuah variabel yang tidak terlihat (*latent variable*) dari sebuah dokumen. Berikut ini merupakan persamaan pada metode PLSA.

$$P(d_i, w_j) = P(d_i)P(w_j | d_i), P(w_j | d_i)$$

Dimana

$$\sum_{z \in Z} P(w_j | z_k)P(z_k | d_i) \quad (3)$$

Dimana :

$P(d)$: Probabilitas terhadap dokumen d .

$P(z/d)$: Probabilitas terhadap topik z yang disesuaikan dengan dokumen d .

$P(w/z)$: Probabilitas terhadap kata w yang disesuaikan dengan topik z .

Nilai $P(d)$, $P(z/d)$ dan $P(w/z)$ dapat ditentukan dengan cara memaksimalkan fungsi *likelihood* L seperti yang terdapat pada persamaan berikut.

$$L = \sum_{i=1}^I \sum_{j=1}^J n(d_i, w_j) \log P(d_i, w_j) \quad (4)$$

Dimana :

$n(d, w)$: merupakan bobot term pada dokumen.

Di dalam metode perhitungan PLSA, terdapat algoritma yang disebut Algoritma *Expectation Maximization* (EM), algoritma ini digunakan untuk memperkirakan nilai maksimum *likelihood* dalam model variabel laten. Terdapat dua langkah dalam algoritma ini yaitu : langkah *Expectation* (E-step) dan langkah *Maximization* (M-step). Proses E-Step berfungsi untuk menghitung probabilitas posterior untuk variabel z berdasarkan pada perkiraan parameter saat itu, dan proses M-Step berfungsi untuk meng-*update* parameter yang digunakan untuk menghitung nilai probabilitas posterior variabel z , yang akan digunakan dalam perhitungan nilai *likelihood*. Berikut ini merupakan persamaan pada proses E-Step.

$$P(z | d, w) = \frac{p(w_j | z_k) p(z_k | d_i)}{\sum_{k=1}^K p(w_j | z_k) p(z_k | d_i)} \quad (5)$$

Sedangkan persamaan untuk proses M-Step adalah sebagai berikut.

$$P(w | z) = \frac{\sum_{i=1}^I n(d_i, w_j) P(z_k | d_i, w_j)}{\sum_{i=1}^I \sum_{j=1}^J n(d_i, w_j) P(z_k | d_i, w_j)}$$

dan

$$P(z_k | d_i) = \frac{\sum_{j=1}^J n(d_i, w_j) p(z_k | d_i, w_j)}{\sum_{j=1}^J \sum_{k=1}^K n(d_i, w_j) p(z_k | d_i, w_j)} \quad (6)$$

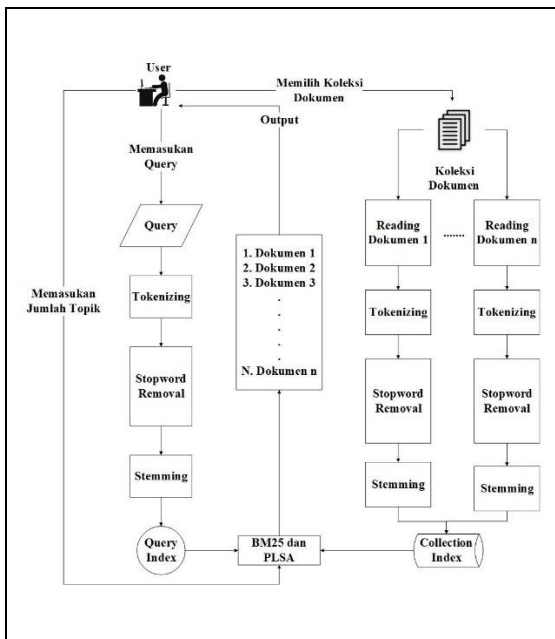
Setelah mendapatkan matrix perhitungan dokumen dan matrix perhitungan *query* maka selanjutnya matrix tersebut dihitung menggunakan *cosine similarity* dengan persamaan sebagai berikut.

$$\frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (7)$$

Rancangan Sistem

Pada penelitian ini terdapat dua tahapan dalam pembuatan pencarian dokumen dengan menggunakan metode BM25 dan PLSA yaitu tahap analisis dan desain aplikasi.

Pada aplikasi pencarian dokumen ini terdapat dua proses operasi, yaitu operasi terhadap koleksi dokumen dan proses terhadap *query* yang dimasukan oleh *user*. Berikut ini merupakan proses kerja aplikasi pencarian dokumen yang terdapat pada Gambar 1.



Gambar 1 Rancangan Sistem Aplikasi Pencarian Dokumen

Dokumen-dokumen yang dapat terbaca oleh sistem adalah dokumen-dokumen yang memiliki format *.docx, *.doc dan *.pdf yang terdapat pada perangkat komputer *user*.

Berikut ini merupakan tahapan-tahapan yang dilakukan dalam menjalankan aplikasi pencarian dokumen.

1. *User* memasukkan alamat di mana koleksi dokumen berada.
2. *User* memasukan *query* yang diinginkan.
3. *User* memasukan jumlah topik dan tekan tombol *search*.
4. Sistem akan melakukan proses *preprocessing* pada dokumen-dokumen yang berada pada koleksi dokumen dan *query* yang sudah *user* inputkan. *Preprocessing* tersebut meliputi *tokenizing*, *stopword removal* dan *stemming*.
5. Proses *tokenizing* merupakan proses pemisahan kalimat menjadi kata-kata.
6. Proses *stopword removal* merupakan proses yang berfungsi untuk menghilangkan kata-kata yang sudah didapatkan pada proses *tokenizing* yang tidak memiliki arti penting yang disesuaikan dengan kamus kata yang sudah disesuaikan.
7. Proses *stemming* merupakan proses penghapusan imbuhan pada kata-kata yang sudah melalui proses *stopword removal*. setelah melalui proses *stemming* maka kata-kata tersebut sudah menjadi *root word* atau kata dasar, kata dasar ini yang disebut dengan kumpulan *term*^[6]. *Term* yang sudah dihasilkan tersebut meliputi *term* dokumen dan *term query*.
8. Sistem akan memeriksa *term query* yang terdapat pada masing-masing dokumen, kemudian sistem akan menghitung bobot dokumen dengan menggunakan persamaan BM25 dan PLSA.
9. Sistem akan merangking dokumen mana yang memiliki nilai lebih besar berdasarkan *query* yang dimasukan.

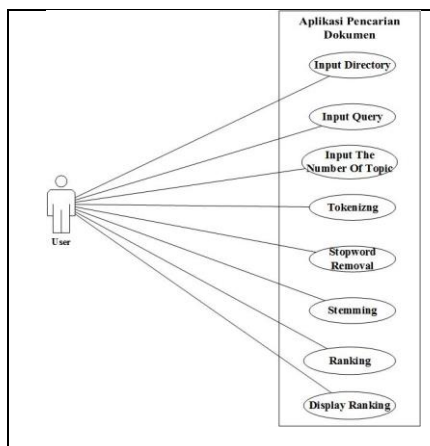
Use Case Diagram

Berdasarkan analisis sistem yang dilakukan, berikut ini merupakan fungsi-fungsi yang dilakukan oleh aplikasi pencarian dokumen yaitu *input directory*, *input query*, *tokenizing*, *stopword removal*, *stemming*, *ranking* dan *display ranking*. Seperti yang terdapat pada Gambar 2.

Fungsi *input directory* memiliki fungsi untuk menentukan koleksi dokumen yang ingin dilakukan proses pencarian. Fungsi *input query* berfungsi untuk memasukan *query* yang diinginkan. Fungsi *input the number of the topic* memiliki fungsi untuk menentukan jumlah topik yang diinginkan.

Fungsi *tokenizing* berfungsi untuk memisahkan kalimat menjadi kumpulan kata. Fungsi *stopword removal* berfungsi untuk menghilangkan kata-kata yang tidak memiliki arti. Fungsi *stemming* berfungsi untuk mengubah kata-kata menjadi kata dasa atau *root word*.

Fungsi *ranking* berfungsi untuk melakukan proses pembobotan pada dokumen dan mengurutkan sesuai bobot dokumen. Fungsi *display ranking* berfungsi untuk memperlihatkan hasil dari perankingan dokumen.



Gambar 2 Use Case Diagram

Implementasi Sistem

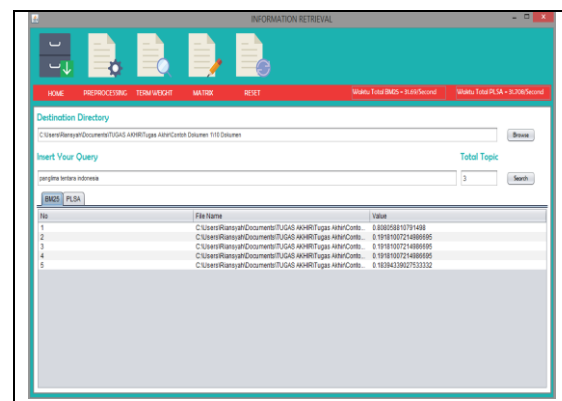
Untuk mengimplementasikan rancangan sistem ini membutuhkan bahasa pemrograman Java JDK 1.8 atau versi sebelumnya. Selain itu juga menggunakan *software* pendukung, yaitu NetBeans IDE 8.1.

Teknik Pengujian

Pada perancangan aplikasi pencarian dokumen ini, dilakukan pengujian yang berfungsi untuk menguji fungsionalitas terhadap aplikasi yang telah dibuat. Tujuan pengujian ini dilakukan yaitu untuk mengetahui apakah aplikasi berjalan sesuai dengan perancangan pembuatan. Berikut ini merupakan proses pengujian aplikasi pencarian dokumen.

Halaman Menu Utama

Gambar 3 memperlihatkan tampilan menu utama pada aplikasi pencarian dokumen. Pada halaman utama ini menampilkan *form* pengisian *directory*, pengisian *query*, pengisian jumlah topik dan hasil perankingan aplikasi pencarian dokumen.

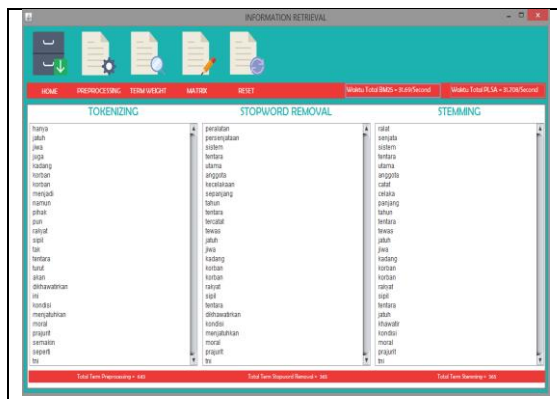


Gambar 3 Menu Utama

Halaman Preprocessing

Pada menu *preprocessing* akan menampilkan hasil *preprocessing* pada

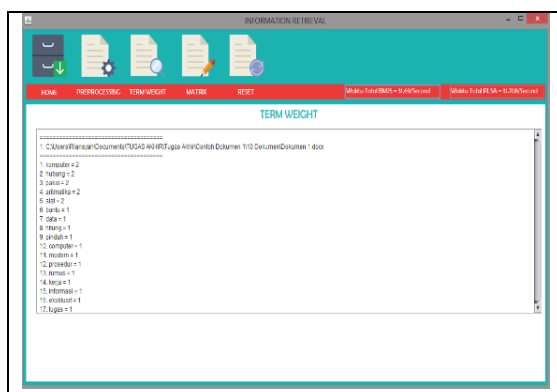
dokumen-dokumen yang terdapat pada koleksi dokumen seperti yang ditunjukkan pada Gambar 4.



Gambar 4 Halaman *Preprocessing*

Halaman *Term Weight*

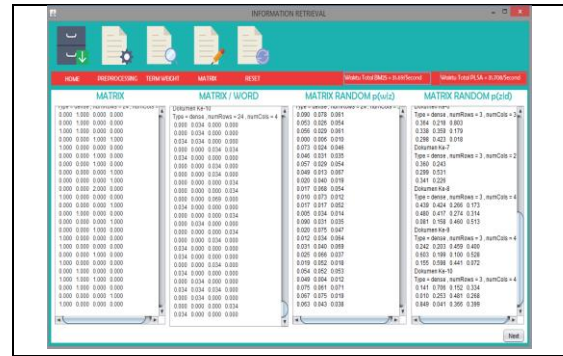
Pada halaman *term weight* merupakan halaman yang menampilkan daftar kata dan jumlah kata yang terdapat pada dokumen-dokumen yang berada pada koleksi dokumen seperti yang terlihat pada Gambar 5.



Gambar 5 Halaman *Term Weight*

Halaman *Matrix*

Pada halaman *Matrix* diperlihatkan proses perhitungan pada metode *PLSA* seperti yang terdapat pada Gambar 6.



Gambar 6 Halaman *Matrix*

Pengujian *Precision*, *Recall* dan *F-Measure* [1]

Precision atau ketepatan adalah perbandingan jumlah total dokumen relevan yang didapatkan oleh sistem dengan jumlah seluruh dokumen yang terambil oleh sistem baik dokumen tersebut relevan ataupun tidak.

$$\text{Precision} = \frac{\text{relevant item retrieved}}{\text{retrieved items}} \quad (8)$$

Recall atau kelengkapan adalah perbandingan jumlah total dokumen relevan yang didapatkan oleh sistem dengan jumlah seluruh dokumen yang ada pada koleksi dokumen baik dokumen tersebut terambil oleh sistem maupun tidak.

$$\text{Recall} = \frac{\text{relevant items retrieved}}{\text{relevant items}} \quad (9)$$

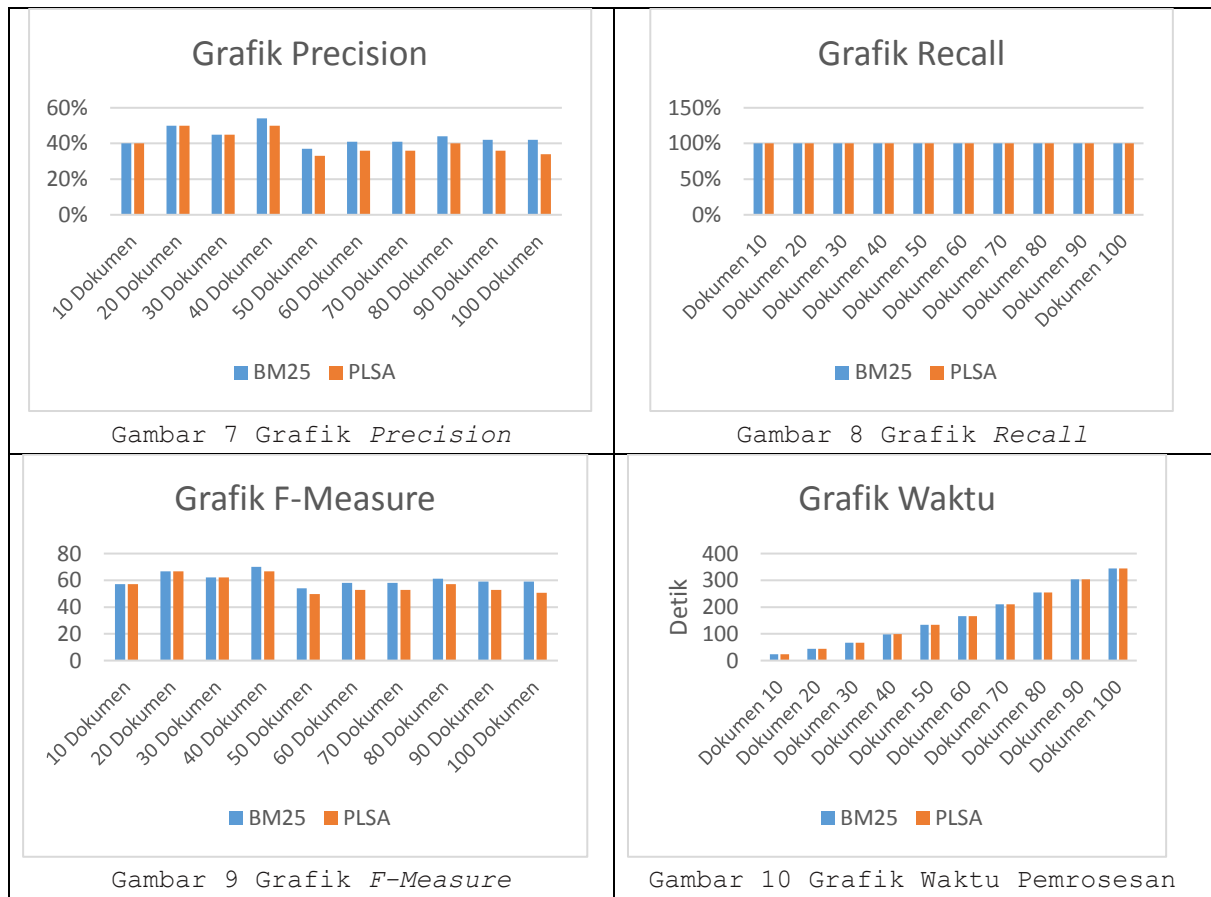
F-Measure adalah suatu perhitungan evaluasi yang mengkombinasikan antara *precision* dan *recall*. Dimana nilai *recall* dan *precision* pada suatu keadaan memiliki nilai yang berbeda. Ukuran yang menampilkan timbal balik antara *recall* dan *precision* adalah *F-Measure*.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (10)$$

Tabel 1 Hasil Pengujian

NO	Jumlah Dokumen	Dokumen Yang Ditemukan (BM25)	Dokumen Yang Ditemukan (PLSA)	Precision (%)		Recall (%)		F-Measure		Waktu (Detik)	
				BM25	PLSA	BM25	PLSA	BM25	PLSA	BM25	PLSA
1	10	5	5	40	40	100	100	57,142	57,142	23.548	23.585
2	20	6	6	50	50	100	100	66,666	66,666	43.387	43.44
3	30	9	11	55	45	100	100	70,967	62,068	67.186	67.201
4	40	11	12	54	50	100	100	70,129	66,666	97.286	99.327
5	50	16	18	37	33	100	100	54,014	49,624	133.155	133.171
6	60	17	19	41	36	100	100	58,156	52,941	166.131	166.146
7	70	17	19	41	36	100	100	58,156	52,941	210.832	210.848
8	80	18	20	44	40	100	100	61,111	57,142	254.01	254.042
9	90	19	22	42	36	100	100	59,154	52,941	303.574	303.606
10	100	19	23	42	34	100	100	59,154	50,746	343.867	343.883

Berikut ini merupakan grafik-grafik pengujian berdasarkan nilai *precision*, *recall*, *f-measure* dan waktu pemrosesan. Pada Gambar 7 memperlihatkan grafik *precision*, Gambar 8 memperlihatkan grafik *recall*, Gambar 9 memperlihatkan grafik *F-Measure* dan Gambar 10 memperlihatkan grafik waktu yang pemrosesan.



Tabel 1 memperlihatkan hasil pengujian aplikasi pencarian dokumen menggunakan *precision*, *recall* dan *F-Measure*. Pada pengujian di atas terlihat bahwa nilai *precision* metode BM25 lebih besar dibandingkan dengan metode PLSA, hal ini dikarenakan pada pencarian dokumen dengan menggunakan metode BM25 terdapat beberapa dokumen tidak sesuai dengan *query* yang tidak terdeteksi bobot dokumennya, sedangkan pada pencarian dengan metode PLSA baik dokumen yang sesuai dengan *query* maupun dokumen tidak sesuai *query* terdeteksi bobot dokumennya. Pada pengujian di atas memperlihatkan nilai *recall* yang tetap pada nilai 100%, ini menandakan bahwa semua dokumen yang sesuai dengan *query* pada koleksi dokumen dapat ditemukan oleh sistem.

Kesimpulan

Berdasarkan hasil penelitian dan pengujian sistem yang telah dilakukan dapat disimpulkan sebagai berikut :

1. Penerapan metode BM25 dan PLSA pada aplikasi pencarian dokumen dapat dilakukan dengan dokumen yang memiliki format *.doc, *.docx dan *.pdf.
2. Metode BM25 lebih efektif dalam melakukan perangkingan dokumen, karena metode PLSA lebih mengutamakan nilai dokumen dibandingkan dengan kata yang memiliki relevansi terhadap *query*.
3. Berdasarkan pengujian yang dilakukan seperti yang terlihat pada Tabel 1, metode BM25 memiliki waktu pemrosesan yang lebih cepat dibandingkan dengan metode PLSA.
4. Nilai rata-rata *F-Measure* pada metode BM25 adalah 61,649 dan nilai rata-rata

F-Measure pada metode PLSA adalah 56,8877.

Daftar Pustaka

1. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze., 2009. *An Introduction to Information Retrieval*. Cambridge University Press Cambridge, England.
2. Graham Bennett, Falk Scholer, Alexandra Uitdenbogerd., 2008. *A Comparative Study of Probabilistic and Language Models for Information Retrieval*. Australian Computer Society, Inc.
3. James Rukka Embongbulan, Yanuar Firdaus A.W., Angelina Prima Kurniati, *Analisis dan Implementasi Metoda BM25 pada Information Retrieval*. Bandung, Program Studi S1 Teknik Informatika (Universitas Telkom) 2010.
4. Ratri Anggardani Prayitno, Warih Maharani, Ade Romadhony. *Opinion Retrieval dengan Menggunakan Probabilistic Latent Semantic Analysis*, Bandung, Program Studi S1 Teknik Informatika (Universitas Telkom) 2012.
5. Muchlisin Riadi, *Information Retrieval System (IRS)*, (<http://www.kajianpustaka.com/2012/10/information-retrieval-system-irs.html>, diakses 31 Oktober 2015.).
6. Tanto, Liyan, 2011, *Stemming Bahasa Indonesia dengan Algoritma Nazief dan Andriani* (Online), (<https://liyantanto.wordpress.com/2011/06/28/stemming-bahasa-indonesia-dengan-algoritma-nazief-dan-andriani/>), Diakses pada tanggal 5 November 2015.