

PERANCANGAN DAN PEMBUATAN APLIKASI PENCARIAN INFORMASI BEASISWA DENGAN MENGUNAKAN COSINE SIMILARITY

Andry Kurniawan, Firdaus Solihin, Fika Hastarita

Prodi Teknik Informatika, Fakultas Teknik, Universitas Trunojoyo

Jl. Raya Telang, PO BOX 2 Kamal, Bangkalan

E-mail : andry.dota@gmail.com

ABSTRAK

Banyak informasi beasiswa yang ada di internet, di satu sisi memunculkan kesulitan bagi para pencari beasiswa untuk mendapatkan informasi beasiswa. Konsep sistem pencarian berbasis information retrieval (IR), dapat digunakan untuk proses mencari informasi beasiswa melalui media internet. Pembangunan IR pada penelitian ini menggunakan konsep vector space model (VSM). Untuk pengumpulan data informasi beasiswa menggunakan Web Crawler, hasil dari proses crawling akan disimpan ke dalam basis data. Aplikasi Web Crawler yang digunakan adalah Web Crawler Vietspider. Kesamaan data beasiswa yang akan diintegrasikan oleh cosine similarity. Dengan menggunakan cosine similarity untuk membentuk data beasiswa yang relevan satu sama lain yang dibutuhkan oleh user berdasarkan query yang dimasukkan. Berdasarkan hasil dari pengujian sistem, didapatkan nilai minimal cosine similarity paling baik adalah threshold 0,1. Dengan rata-rata presentase nilai f-measure atau tingkat efisiensi kinerja sistem ini mencapai 19,9 %.

Kata kunci : Beasiswa, Information Retrieval, Vector Space Model, Web Crawler, Cosine Similarity.

ABSTRACT

Many a scholarship information available in the internet, on the other hand that makes difficult for seeker to find scholarship information. The concept of search system based information retrieval (IR) can be used for the process of searching for scholarship information via internet. The development of IR in this study uses the concept of vector space model (VSM). For the collecting data of scholarship information is using Web Crawler, the results of crawling process are stored in the basis data. The applications that use a web crawler is a web crawler vietspider. The similarity of scholarship data is to be integrated by the cosine similarity to form a scholarship of data relevant to each other is needed by user based on the query entered. Based on the results of the testing system, the score of at least the most well cosine similarity threshold is 0.1. The average percentage value of the f-measure system performance or the level of efficiency reached 19.9%.

Keyword: Scholarship, Information Retrieval, Vector Space model, Web Crawler, Cosine Similarity.

PENDAHULUAN

Akhir-akhir ini banyak sekali berita informasi beasiswa. Mulai dari beasiswa prestasi, beasiswa tidak mampu, pertukaran pelajar ke luar negeri, beasiswa olimpiade, sampai beasiswa di tingkat internasional. Beasiswa adalah suatu pemberian berupa bantuan keuangan yang diberikan kepada perorangan yang bertujuan untuk digunakan demi keberlangsungan pendidikan yang ditempuh. Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan. Pemberian beasiswa dapat dikategorikan pada pemberian cuma-cuma ataupun pemberian dengan ikatan kerja (biasa disebut ikatan dinas) setelah selesainya pendidikan. Lama ikatan dinas ini berbeda-beda, tergantung pada lembaga yang memberikan beasiswa tersebut.

Sesuai dengan adanya peraturan oleh sekolah untuk memperoleh beasiswa, maka diperlukan kriteria-kriteria untuk mendapatkan beasiswa. Untuk membantu pelajar mencari beasiswa diinginkan yang sesuai kriterianya, dibutuhkan sebuah sistem yang mempermudah dalam mendapatkan informasi beasiswa maka dibangunlah aplikasi pencarian informasi beasiswa.

Untuk menerapkan proses dalam membangun aplikasi pencarian informasi beasiswa ini yaitu dengan adanya Information retrieval. Information retrieval (IR) adalah suatu bidang ilmu yang mempelajari cara-cara penelusuran kembali atas dokumen-dokumen yang ada dalam basis data, sehingga didapatkan suatu informasi yang tepat dari apa yang dicari dalam dokumen tersebut. [1] information retrieval melakukan suatu pendugaan atas dokumen-dokumen yang diinginkan pengguna. Sistem basis data retrieval dapat menerima query yang kompleks dan memberikan semua jawaban sesuai kondisi logis dari query bersangkutan [1].

Dalam penelitian ini akan dibangun suatu perancangan dan pembuatan aplikasi pencarian informasi

beasiswa dengan menggunakan Cosine Similarity. Untuk pengumpulan data beasiswa menggunakan web crawler, hasil dari proses crawling akan disimpan ke dalam basis data. *Crawler* diawali dengan adanya daftar *URL* yang akan dikunjungi, disebut dengan *seeds*. Setelah *crawler* mengunjungi *URL* tersebut, kemudian mengidentifikasi semua *hyperlink* dari halaman itu dan menambahkan kembali ke dalam *seeds*. Hal ini dinamakan *crawl frontier*. Setelah *web crawler* mengunjungi halaman-halaman *web* yang ditentukan di dalam *seeds*, maka *web crawler* membawa data-data yang dicari oleh *user* kemudian menyimpannya ke sebuah *storage*. Cosine Similarity merupakan sebuah metode penghitungan kemiripan *query* yang dimasukkan dengan isi dokumen. Kemudian dilakukan perangkingan. Dengan menggunakan *cosine similarity* untuk membentuk data beasiswa yang relevan satu sama lain yang dibutuhkan oleh *user* berdasarkan *query* yang dimasukkan. Dan pada akhirnya dapat memudahkan pelajar untuk memperoleh dan mengetahui informasi tentang beasiswa.

Permasalahan

Pada penelitian untuk membangun sistem mesin pencari ini akan dibahas permasalahan yang penting yaitu:

1. Bagaimana membuat aplikasi yang mampu untuk mencari informasi beasiswa yang tersebar di beberapa website?
2. Bagaimana pengaruh aplikasi pencarian informasi beasiswa yang dibuat terhadap efisiensi kinerja sistem?

Batasan Masalah

Dalam penelitian ini permasalahan difokuskan pada masalah-masalah berikut :

1. Pengumpulan data beasiswa menggunakan Web Crawler.
2. Website beasiswa yang akan melakukan proses crawling yaitu

www.scholarship-positions.com/
 www.edu-active.com/
 www.studentscholarships.org/
 www.scholarshubafrika.com/

3. Sistem pencarian menggunakan konsep vector space model.
4. Menggunakan metode perhitungan TF*IDF untuk pembobotan dan cosine similarity untuk mengukur kemiripan query dengan beasiswa lalu dilakukan perankingan.
5. Evaluasi sistem menggunakan recall, precision, dan f-measure.

Tujuan

Penelitian yang berjudul “Perancangan dan Pembuatan Aplikasi Pencarian Informasi Beasiswa dengan menggunakan Cosine Similarity” ini dirancang bertujuan untuk Membangun sistem pencarian informasi beasiswa untuk pengguna dalam memperoleh informasi beasiswa yang diinginkan dengan menggunakan cosine similarity.

METODE

Information retrieval

Information retrieval (IR) adalah suatu bidang ilmu yang mempelajari cara-cara penelusuran kembali atas dokumen-dokumen yang ada dalam basis data. *Information retrieval* berbeda dengan statis retrieval. Sistem statis retrieval umumnya memberikan tepat semua dokumen atau objek yang memenuhi kriteria tertentu. Sedangkan sistem *information retrieval* melakukan suatu pencarian atas dokumen-dokumen yang diinginkan pengguna, dengan melihat tingkat kemiripannya [1].

Preprocessing

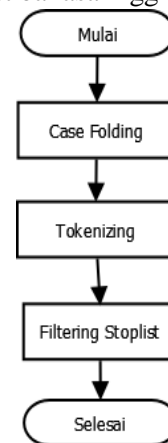
Preprocessing merupakan proses awal dalam pembentukan keyword-keyword dari data beasiswa yang nantinya akan disimpan dalam basis data, dengan beberapa tahapan proses :

a. Case folding, Merupakan proses mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ‘a’ sampai dengan ‘z’ yang diterima.

Karakter-karakter dan tanda baca lainnya selain huruf juga dihilangkan.

b. Tokenizing, Merupakan proses pemenggalan tiap pada isi beasiswa berdasarkan pada spasi dan tanda – (penghubung).

c. Filtering, Merupakan proses penghilangan kata-kata (yang dianggap) sebagai kata yang jarang dicari atau jarang digunakan sebagai keywords pada proses pencarian. Proses ini dilakukan untuk mengurangi waktu yang dibutuhkan pada saat penghitungan frekuensi tiap kata pada dokumen. Pada penelitian ini, stoplist yang digunakan adalah stoplist bahasa inggris.



Gambar 1. Flowchart Proses Preprocessing Sistem.

Vector Space Model

Pada sistem temu balik informasi, kemiripan antar dokumen didefinisikan berdasarkan representasi model ruang vektor (*vector space model, VSM*). Pada VSM, setiap dokumen di dalam *basis data* dan *query* direpresentasikan oleh suatu vektor multi-dimensi. Pada model VSM ini [2]:

- a. *Vocabulary* merupakan kumpulan semua *term* berbeda yang tersisa dari dokumen setelah *preprocessing* dan mengandung *t term index*. Term-term ini membentuk suatu ruang vektor.
- b. Setiap *term i* di dalam dokumen atau *query j*, diberikan suatu bobot (*weight*) bernilai real w_{ij} .
- c. Dokumen dan *query*

diekspresikan sebagai vektor t dimensi $d_j = (w_1, w_2, \dots, w_{ij})$ dan terdapat n dokumen di dalam koleksi, yaitu $j = 1, 2, \dots, n$.

Dalam model ruang vektor, koleksi dokumen direpresentasikan oleh matriks *term document* (atau matriks *term-frequency*). Setiap sel dalam matriks bersesuaian dengan bobot yang diberikan dari suatu *term* dalam dokumen yang ditentukan. Nilai nol berarti *term* tidak terdapat di dalam dokumen.

Metode pembobotan kata merupakan suatu cara untuk menghitung bobot kata (w_i) dalam dokumen (t_k). *TF-IDF* (*Term Frequency - Inversed Document Frequency*) digunakan untuk menghitung bobot (W) dari setiap *term* dalam dokumen, didapat dengan mengalikan *tf-idf*. Rumusnya adalah sebagai berikut [1]:

$$W_{dt} = tf_{dt} * IDF_t \dots\dots\dots(1)$$

Dimana:

- d = dokumen ke- d
- t = term ke- t dari dokumen
- W = bobot dokumen ke- d terhadap term ke- t
- tf = banyaknya term i pada sebuah dokumen
- idf = Inversed Document Frequency, $\log_2(n/df)$
- df = banyak dokumen yang mengandung term i

Cosine Similarity

Untuk menghitung seberapa besar kemiripan antar dokumen, dapat menggunakan suatu fungsi ukuran kemiripan (*similarity measure*). Dengan menggunakan fungsi Ukuran ini memungkinkan perbandingan dokumen sesuai dengan kemiripan atau relevan terhadap query.

Salah satu perhitungan kemiripan antar dokumen yang digunakan dalam penelitian ini adalah *Cosine similarity*. *Cosine similarity* merupakan metode fungsi pengukuran kesamaan antara dua vektor yang berbeda dengan mengukur cosinus dari sudut antara mereka.

Semakin besar nilai cosinus (maksimal 1) maka semakin mirip dokumen yang dibandingkan. Nilai cosinus 1 menyatakan kemiripan 100%, sedangkan nol menyatakan ketidakmiripan 100% [3]. *Cosine similarity* didefinisikan dalam rumus fungsi perhitungan berikut:

Dimana:

- d = dokumen, q = query, t = term
- w_{ij} = TF-IDF kata ke i dari dokumen ke j
- w_{iq} = TF-IDF kata ke i dari query

$$similarity(\vec{d}_j, \vec{q}) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| |\vec{q}|} = \frac{\sum_{i=1}^t (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^t w_{ij}^2 \cdot \sum_{i=1}^t w_{iq}^2}} \dots\dots\dots(2)$$

Evaluasi

Evaluasi digunakan untuk mengukur kinerja suatu sistem demi menghasilkan perbaikan pada proses pengambilan informasi. Ukuran umum yang digunakan untuk mengukur kualitas dari *text retrieval* adalah kombinasi dari *precision* dan *recall*. Metode yang umum digunakan adalah *recall*, *precision*, dan *f-measure* [4].

1. *Recall*

Recall adalah proporsi jumlah dokumen teks yang relevan terkenali

$$Recall = \frac{Relevant \cap Retrieved\ Basiswa}{Relevant\ Basiswa} \dots\dots\dots(3)$$

di antara semua dokumen teks relevan yang ada pada koleksi.

2. *Precision*

Precision adalah proporsi jumlah dokumen teks yang relevan terkenali di

$$Precision = \frac{Relevant \cap Retrieved\ Basiswa}{Retrieved\ Basiswa} \dots\dots\dots(4)$$

antara semua dokumen teks yang terpilih oleh sistem. Rumus *precision* adalah sebagai berikut:

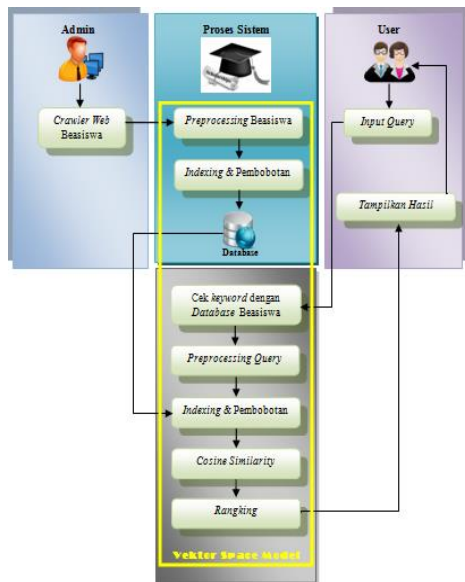
3. *F-Measure*

F-Measure adalah nilai yang mewakili seluruh kinerja sistem yang merupakan rata-rata dari nilai *precision* dan *recall*. Rumus *F-Measure* dapat dilihat pada Persamaan berikut:

Rancangan Sistem

$$F\text{-measure} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \dots \dots \dots (5)$$

Sebelum membuat program aplikasi, terlebih dahulu dilakukan proses perancangan sistem. Hal ini dilakukan supaya aplikasi yang dibuat dapat berfungsi sesuai dengan yang diharapkan sehingga mampu berfungsi maksimal. Gambaran umum dari sistem ini dapat dilihat pada gambar 2.



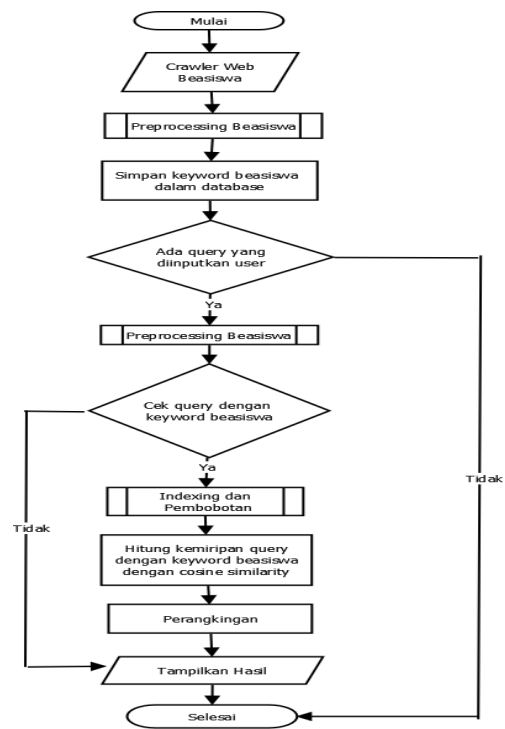
Gambar 2. Gambaran Umum Sistem

Deskripsi keterangan dari gambar diatas dapat dijelaskan sebagai berikut yang pertama adalah Admin, merupakan pengguna yang berhak melakukan kontrol penuh pada sistem dan basis data. Admin mencrawler web beasiswa lalu disimpan ke basis data. Kemudian sistem melakukan proses preprocessing pada data beasiswa. Sehingga dihasilkan keyword beasiswa yang akan tersimpan dalam basis data beasiswa. Kemudian keyword yang tersimpan dalam basis data tersebut dilakukan indexing dan pembobotan.

User, adalah pengguna yang akan menginput query dan melihat informasi hasil pencarian. User menginputkan query kedalam aplikasi. Kemudian sistem melakukan proses preprocessing query sehingga diperoleh keyword query. Selanjutnya sistem akan melakukan pencocokan Keyword query dengan

keyword beasiswa yang telah tersimpan dalam basis data. Jika ada keyword yang cocok/sama, maka keyword query akan di index dan dilakukan proses pembobotan. Tapi jika tidak ada yang cocok, maka tidak ada hasil yang ditampilkan (no result). Kemudian dilakukan proses perhitungan kemiripan cosine similarity antara bobot keyword beasiswa dan bobot keyword query. Lalu dilakukan proses perangkaian. Yang terakhir adalah mengukur kinerja system, dilakukan evaluasi menggunakan recall, precision dan f-measure.

Flowchart Sistem
Flowchart Vector Space Model



Gambar 3. Flowchart Vector Space Model

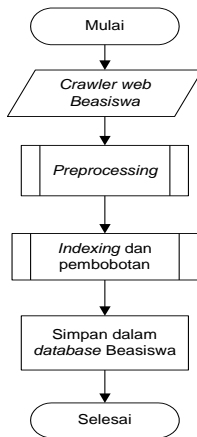
Deskripsi keterangan dari gambar diatas dapat dijelaskan sebagai berikut :

1. Evaluasi dilakukan dengan menggunakan 15 query sampel. Masing-masing 5 query yang terdiri dari 1 kata, 5 query yang terdiri dari 2 kata dan 5 query yang terdiri dari 3 kata.
2. Untuk mendapatkan nilai relevant beasiswa, masing-

masing query akan di check pada keseluruhan beasiswa yang ada dalam basis data yang mengandung kata query, berdasarkan pada user oriented measure.

3. Jika ada yang cocok maka, dilakukan pembatasan nilai minimal cosine similarity dengan menggunakan threshold.
4. Kemudian relevan beasiswa di check dengan beasiswa yang ter-retrieve untuk mendapatkan irisan relevan retrieve beasiswa.
5. Jika ada yang sama maka dilakukan evaluasi dengan mencari nilai recall, precision dan f-measure.
6. Untuk mencari recall, precision dan f-measure.
7. Hasil dari relevan beasiswa dan irisan relevan retrieve beasiswa kemudian akan dimasukkan dalam rumus recall dan precision.

Flowchart Utama Admin



Gambar 4. Flowchart Utama Admin Pada Preprocessing Dan Pembobotan Beasiswa

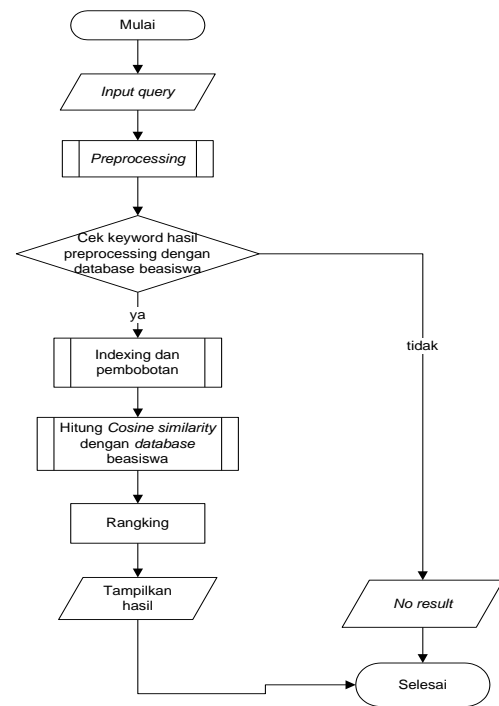
Deskripsi keterangan dari gambar diatas dapat dijelaskan sebagai berikut :

1. Admin mecrawler website beasiswa.
2. Preprocessing beasiswa. Pada preprocessing dilakukan proses case folding, tokenizing, filtering

stopword menggunakan stoplist bahasa inggris.

3. Jika sudah dilakukan preprocessing maka akan terbentuk lah keyword beasiswa.
4. Lalu dilakukan indexing dan pembobotan pada keyword beasiswa tersebut dengan menggunakan TF*IDF.
5. Kemudian disimpan dalam basis data.

Flowchart User Dan Proses Penanganan Query



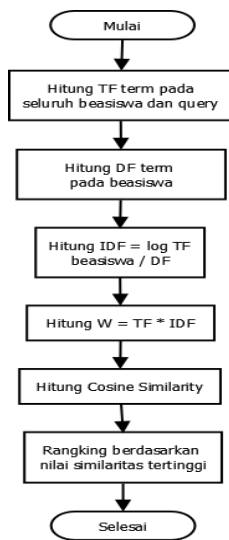
Gambar 5. Flowchat User Pada Proses Penanganan Query

Deskripsi keterangan dari gambar diatas dapat dijelaskan sebagai berikut :

1. User menginputkan query.
2. Dilakukan preprocessing query (langkah preprocessing query sama dengan preprocessing beasiswa) sehingga menghasilkan keyword query.
3. Keyword query dicocokkan dengan basis data keyword data beasiswa. jika ada yang cocok, maka akan dilakukan proses indexing dan

- pembobotan, namun jika tidak ada maka hasilnya akan *no result* (selesai).
4. Kemudian dilakukan penghitungan kemiripan *keyword query* dan *keyword* beasiswa dengan fungsi *cosine similarity*.
 5. Lalu dilakukan proses perangkingan.

Flowchart Indexing, Cosine Similarity Dan Perangkingan



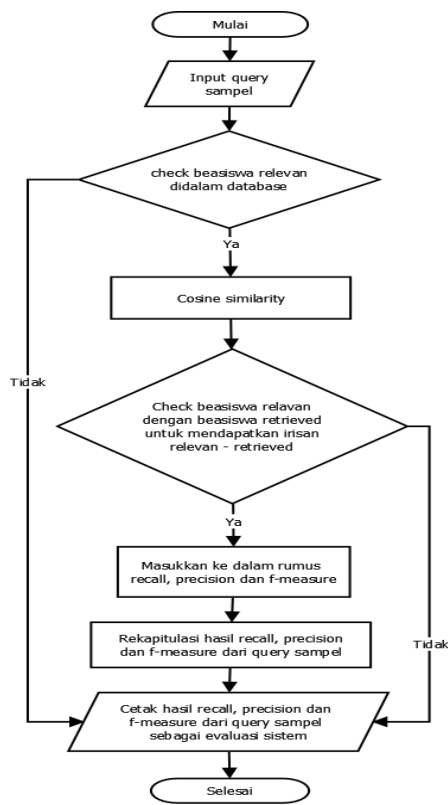
Gambar 6. *Flowchart Indexing, Cosine Similarity Dan Perangkingan*

Deskripsi keterangan dari gambar diatas dapat dijelaskan sebagai berikut :

1. Tahap pertama adalah mencari frekuensi kemunculan term dalam setiap beasiswa (TF beasiswa) dan dalam query (TF query).
2. Kemudian akan ditemukan jumlah frekuensi kemunculan term dalam seluruh beasiswa (DF).
3. Lalu hitung IDF, dengan rumus : $\text{Log Jumlah seluruh beasiswa TF} / \text{DF}$.
4. Lakukan pembobotan dengan rumus : $W = \text{TF} * \text{IDF}$.
5. Langkah selanjutnya adalah melakukan perhitungan kemiripan antara term dalam query dengan term dalam beasiswa dengan menggunakan cosine similarity.

6. Yang selanjutnya akan dirangking.
7. Yang terakhir adalah perangkingan. Perangkingan dilakukan berdasarkan nilai terbesar dari hasil cosine similarity akan menempati rangking tertinggi. Dan diurutkan seterusnya.
8. List result, hasil ditampilkan pada user.

Flowchart Evaluasi



Gambar 7. *Flowchart Evaluasi*

Deskripsi keterangan dari gambar diatas dapat dijelaskan sebagai berikut :

1. Evaluasi dilakukan dengan menggunakan 15 query sampel. Masing-masing 5 query yang terdiri dari 1 kata, 5 query yang terdiri dari 2 kata dan 5 query yang terdiri dari 3 kata.
2. Untuk mendapatkan nilai relevant beasiswa, masing-masing query akan di check pada keseluruhan

beasiswa yang ada dalam basis data yang mengandung kata query, berdasarkan pada user oriented measure.

3. Jika ada yang cocok maka, dilakukan pembatasan nilai minimal cosine similarity dengan menggunakan threshold.
4. Kemudian relevan beasiswa di check dengan beasiswa yang ter-retrieve untuk mendapatkan irisan relevan retrieve beasiswa.
5. Jika ada yang sama maka dilakukan evaluasi dengan mencari nilai recall, precision dan f-measure.
6. Untuk mencari recall, precision dan f-measure.
7. Hasil dari relevan beasiswa dan irisan relevan retrieve beasiswa kemudian akan dimasukkan dalam rumus recall dan precision.
8. Setelah diketahui nilai recall & precision nya, akan dihitung nilai f-measure nya dengan menggunakan rumus diatas.

HASIL DAN PEMBAHASAN

Skenario Uji Coba Sistem

Uji coba dilakukan dengan melakukan perbandingan Threshold nilai minimal cosine similarity yang dapat di retrieve oleh sistem. Tujuannya adalah mencari Threshold dengan hasil paling baik, untuk membentuk sistem pencarian yang tepat, akurat dan efisien berdasarkan dari rata-rata analisa f-measure. Pembatasan nilai cosine similarity sebelumnya telah ditentukan oleh penulis berdasarkan trial error, yaitu dengan Threshold minimal 0,1, 0,2, 0,3 dan 0,4.

Query yang digunakan dalam uji coba ini sebanyak 15 kata kunci atau query sampel, masing-masing 5 query yang terdiri dari 2 kata, 5 query yang terdiri dari 3 kata dan 5 query yang terdiri dari 4 kata. Tujuannya adalah mengukur perbandingan keakuratan sistem pencarian dengan menggunakan parameter query 2 kata, query 3 kata dan query 4 kata. Masing-masing query

sampel dimasukkan dalam Threshold pembatasan cosine similarity 0,1, 0,2, 0,3 dan 0,4. Selanjutnya Query sampel yang dimasukkan dalam Threshold tersebut akan dievaluasi dengan mencari nilai maksimal recall, rata-rata precision, kemudian ditemukan nilai f-measure nya, untuk mengetahui tingkat kinerja system. Berdasarkan dari keseluruhan hasil uji coba diatas, dapat kita rekap dan di prosentasekan. Hasil dari prosentase dapat kita lihat dari tabel dibawah :

Tabel 1. Rekapitulasi hasil uji coba tanpa pembatasan nilai cosine similarity

No	Query	Tanpa Threshold		
		Recall	Precision	F-measure
1.	2 Kata	1	0,5637	0,7060
2.	3 Kata	1	0,4877	0,6437
3.	4 Kata	1	0,3993	0,5678
	Rata-rata	1	0,4836	0,6392

Tabel 2. Rekapitulasi hasil uji coba pembatasan nilai cosine similarity dengan threshold minimal 0,1

No	Query	Cosine Similarity 0,1		
		Recall	Precision	F-measure
1.	2 Kata	0,1387	0,5886	0,2191
2.	3 Kata	0,1160	0,4959	0,1806
3.	4 Kata	0,1367	0,3891	0,2001
	Rata-rata	0,1305	0,4912	0,1999

Tabel 3. Rekapitulasi hasil uji coba pembatasan nilai cosine similarity dengan threshold minimal 0,2

No	Query	Cosine Similarity 0,2		
		Recall	Precision	F-measure
1.	2 Kata	0,0219	0,15	0,0381
2.	3 Kata	0,0265	0,5333	0,0503
3.	4 Kata	0,0160	0,25	0,0301
	Rata-rata	0,0215	0,3111	0,0395

Tabel 4. Rekapitulasi hasil uji coba pembatasan nilai cosine similarity dengan threshold minimal 0,3

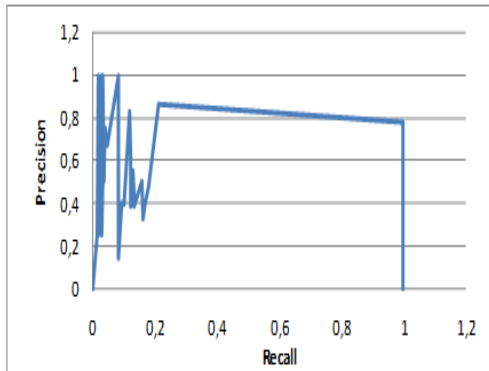
No	Query	Cosine Similarity 0,3		
		Recall	Precision	F-measure
1.	2 Kata	0,0219	0,4	0,0410
2.	3 Kata	0,0031	0,2	0,0062
3.	4 Kata	0	0	0
	Rata-rata	0,0083	0,2	0,0157

Tabel 5. Rekapitulasi hasil uji coba pembatasan nilai cosine similarity dengan threshold minimal 0,4

No	Query	Cosine Similarity 0,4		
		Recall	Precision	F-measure

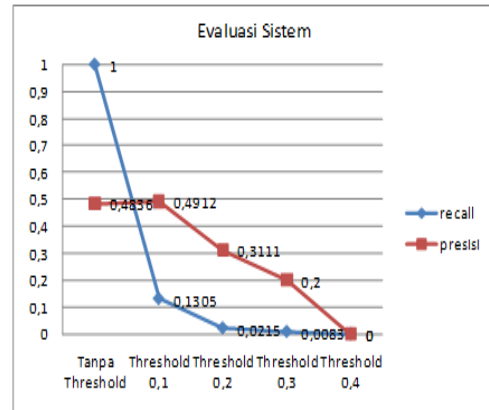
		on		
1.	2 Kata	0	0	0
2.	3 Kata	0	0	0
3.	4 Kata	0	0	0
Rata-rata		0	0	0

Untuk memperjelas nilai *recall*, *precision* dan *f-measure* dari hasil uji coba evaluasi sistem, dapat dipresentasikan dalam grafik interval berikut :

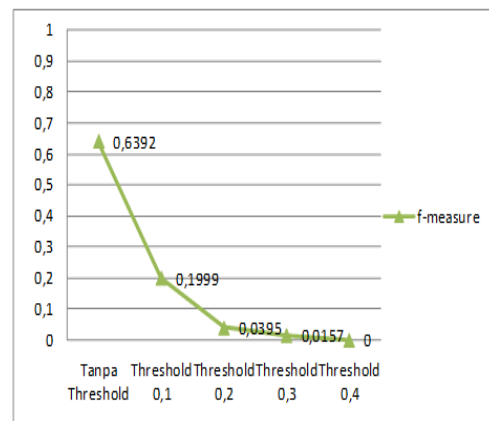


Gambar 8. Grafik uji coba nilai *cosine* similarity dengan tanpa threshold, 0,1 threshold, 0,2 threshold, 0,3 threshold, dan 0,4 threshold, dari query 2 kata, 3 kata dan 4 kata.

Dari grafik pada Gambar 8, menunjukkan, nilai akhir mengarah pada angka 1 pada recall dan mengarah angka sekitar 0,79 pada precision. Dengan recall mendapatkan nilai 1 sehingga tingkat kemampuan aplikasi untuk menampilkan semua beasiswa yang relevan dengan *query* mencapai presentase **100 %** atau Sempurna. Dan precision yang mencapai sekitar 0,79 yang artinya aplikasi memiliki tingkat keakuratan dengan persentase **79 %** yang hampir mendekati **80 %**. Nilai recall dan precision yang digunakan pada grafik diatas adalah nilai recall serta precision dari Nilai *Cosine* Similarity dengan Tanpa Threshold, 0,1 Threshold, 0,2 Threshold, 0,3 Threshold, dan 0,4 Threshold, dari Query 2 Kata, 3 Kata dan 4 Kata.



Gambar 9. Grafik nilai rata-rata hasil uji coba evaluasi sistem pada *recall* dan *precision*. Dari grafik pada Gambar 9, menunjukkan, semakin tinggi pembatasan nilai *threshold*, maka semakin kecil nilai rata-rata *recall* nya. Dan pada *precision* nya hampir sama, semakin tinggi pembatasan nilai *threshold*, maka semakin kecil nilai rata-rata *precision* nya.



Gambar 10. Grafik Nilai Rata-Rata Hasil Uji Coba Evaluasi Sistem pada *f-measure*.

Dari grafik pada Gambar 10. menunjukkan, Semakin tinggi nilai pembatasan *threshold*, maka nilai *f-measure* nya akan semakin kecil. Yang artinya tingkat efisiensi sistem akan semakin kecil jika menggunakan pembatasan nilai *threshold* yang tinggi. Uji coba dengan pembatasan nilai **Threshold Cosine Similarity minimal 0,1** pada hasil *retrieve* sistem, didapatkan hasil rata-rata *f-measure* paling tinggi. Yang artinya pembatasan nilai *cosine*

similarity minimal 0,1 yang digunakan dalam sistem ini.

Teknik Universitas Trunojoyo Madura; 2013.

SIMPULAN

Dari hasil uji coba yang dilakukan, didapatkan nilai *f*-measure paling baik dengan menggunakan Threshold 0,1 yang memperoleh nilai rata-rata 0,1999, sehingga tingkat efisiensi sistem ini mempunyai presentase mencapai 19,9 %. Sedangkan nilai rata-rata keseluruhan *precision* nya sebesar 0,4912, atau memiliki tingkat keakuratan dengan presentase 49,1 %. Dan nilai rata-rata keseluruhan *recall* nya mendapatkan 0,1305, atau tingkat kemampuan aplikasi untuk menampilkan semua informasi beasiswa yang relevan dengan query mencapai presentase 13 %.

- [4] Mastur,M. Perbandingan efektifitas antara penghapusan stoplist dengan penghapusan stoplist dan kata umum pada dokumen hasil klasifikasi pretopology. Bangkalan: Skripsi Jurusan Teknik Informatika Fakultas Teknik Universitas Trunojoyo Madura; 2012.

SARAN

Saran untuk kedepannya diharapkan ada penelitian dengan metode lain yang lebih baik untuk menyelesaikan sistem pencarian informasi beasiswa yang lebih akurat.

DAFTAR PUSTAKA

- [1] Manning, Christopher, D., Raghavan, P., dan Schütze ,H. An Introduction to Information retrieval. Cambridge: University Press; 2008.
- [2] Darmawan, H.A., Wuriyanto,T., Masturi,A. Rancang Bangun Aplikasi Search engine Tafsir Al-Qur'an Menggunakan Teknik Text Mining Dengan Algoritma VSM (Vector space model). Surabaya: Program Studi Sistem Informasi STIKOM; 2010.
- [3] Fahmi,A. Rancangan bangun sistem pencarian dan hirarki pasal-pasal tentang lalu lintas dan angkutan jalan dengan menggunakan cosine similarity. Bangkalan: Skripsi Jurusan Teknik Informatika Fakultas