RECORD AND LIBRARY
JOURNAL

Klasifikafi Dokumen Temu Kembali Informasi dengan *K-Nearest Neghbour*

Information Retrieval Document Classified with K-Nearest Neighbor

Alfian Sukma[1], Badrus Zaman, Endah Purwanti
Fakultas Sains dan Teknologi Universitas Airlangga

Abstrak

Seiring dengan majunya perkembangan teknologi menyebabkan jumlah informasi yang tersedia juga semakin berlimpah. Tujuan dari penelitian ini adalah untuk mengetahui bagaimana penerapan sistem pencarian informasi dalam klasifikasi jurnal dengan menggunakan kesamaan kosinus dan K-Nearest Neighbor (KNN). Data yang digunakan sebanyak 160 dokumen dengan kategori seperti Ilmu Fisika dan Teknik, Life Science, Ilmu Kesehatan, dan Ilmu Sosial dan Humaniora. Tahap konstruksi dimulai dengan menggunakan pengolahan pertambangan teks, pembobotan setiap token dengan menggunakan istilah frekuensi-inverse dokumen frekuensi (TF-IDF), menghitung tingkat kemiripan masing-masing dokumen dengan menggunakan kesamaan kosinus dan klasifikasi menggunakan k-Nearest Neighbor. Evaluasi dilakukan dengan menggunakan dokumen pengujian sebanyak 20 dokumen, dengan nilai k = {37, 41, 43}. Sistem evaluasi menunjukkan tingkat keberhasilan dalam mengklasifikasikan dokumen pada nilai k = 43 dengan presisi nilai 0501. Hasil tes menunjukkan bahwa Sistem pengujian 20 dokumen yang digunakan dapat diklasifikasikan sesuai dengan kategori yang sebenarnya.

Kata kunci: sistem temu kembali, informasi kesamaan, cosinus, k-nearest neighbor, klasifikasi dokumen.

Abstract

Along with the rapid advancement of technology development led to the amount of information available is also increasingly abundant. The aim of this study was to determine how the implementation of information retrieval system in the classification of the journal by using the cosine similarity and K-Nearest Neighbor (KNN). The data used as many as 160 documents with categories such as Physical Sciences and Engineering, Life Science, Health Science, and Social Sciences and Humanities. Construction stage begins with the use of text mining processing, the weighting of each token by using the term frequency-inverse document frequency (TF-IDF), calculate the degree of similarity of each document by using the cosine similarity and classification using k-Nearest Neighbor.Evaluation is done by using the testing documents as much as 20 documents, with a value of k = {37, 41, 43}. Evaluation system shows the level of success in classifying documents on the value of k = 43 with a value precision of 0501. System test results showed that 20 document testing used can be classified according to the actual category.

[1] Alifian Sukma. Fakultas Sains dan Teknologi Universitas Airlangga. Jalan Mulyorejo Kampus C Universitas Airlangga Surabaya.T elepon: 0315914042. Email: alifiansukma7@gmail.com

Keywords: information retrieval system, similarity, cosinus, k-nearest neighbor, document classification.

Seiring dengan semakin majunya perkembangan teknologi dan semakin berkembangnya internet, jumlah informasi yang tersedia juga semakin berlimpah.Namun dengan seiring berkembangnya teknologi, jumlah dokumen juga semakin bertambah, mulai dari dokumen akademik hingga non-akademik. Hal ini menyebabkan mahasiswa merasa kesulitan dalam menentukan dokumen yang tepat untuk dijadikan dasar studi literatur. Oleh karena itu dibutuhkan suatu sistem temu kembali informasi yang mampu melakukan klasifikasi terhadap dokumen agar mahasiswa dapat dengan mudah menemukan dokumen atau jurnal sesuai dengan bidang studi yang diinginkan.

Sistem temu kembali informasi adalah suatu sistem yang mampu dalam menyimpan, mendapatkan, dan melakukan *maintenance* informasi (Kowalski, 2000). Dalam proses sistem temu kembali informasi terdapat pengolahan data yang berupa *text mining*. Kegiatan *text mining* untuk pengolahan dokumen dilakukan dalam 4 tahapan utama. Pertama, yaitu *text pre-processing* yang merupakan tahap awal dalam pengolahan dokumen untuk merubah dokumen menjadi bentuk paling sederhana dengan menggunakan tokenisasi. Kedua, dengan menggunakan *text transformasi* mengolah hasil dari tokenisasi yang berupa kumpulan kata menjadi bentuk dasar dari kata yang terkumpul dengan menggunakan metode *stemming*. Ketiga adalah *feature selection*, dalam tahap ini dibuat suatu *stoplist* yang berisi kumpulan kata-kata yang tidak relevan dengan isi dari tiap dokumen yang diolah, kemudian dilakukan *stopword removal* untuk menghilangkan kata-kata yang terkandung didalam *stoplist* tersebut.Dan yang terakhir adalah *pattern discovery*. *Pattern discovery* merupakan tahap penting dalam pengolahan sistem temu kembali, karena dalam tahap ini ditentukan proses pengolahan yang akan dilakukan pada dokumen-dokumen yang telah terkumpul. Dalam *pattern discovery* terdapat 2 jenis learning yang akan digunakan, yaitu *supervised* dan *ansupervised learning*.

Metode yang digunakan dalam penelitian ini adalah metode *cosine similarity* dan *K-Nearest Neighbor* (KNN) sebagai klasifikasi dokumen. Dengan menggunakan metode *cosine similarity*, klasifikasi dokumen akan lebih mudah dilakukan, *cosine similarity* akan mengolah dengan merubah dokumen tersebut menjadi bentuk vektor dan melakukan perbandingan antara vektor dokumen tersebut sehingga menghasilkan suatu nilai similaritas antar dokumen.

Kemudian metode *k-nearest neighbor* dapat melakukan klasifikasi terhadap dokumen-dokumen yang telah menghasilkan nilai similaritas.Sebelum dilakukan klasifikasi dengan *k-nearest neighbor* dilakukan sorting dari nilai similaritas berdasarkan nilai similaritas terbesar hingga terkecil. Berdasarkan nilai "k" yang telah diinisialisasi pengguna, dapat dilihat jumlah mayoritas kategori dokumen training yang muncul dalam lingkup nilai "k" sebagai dasar untuk melakukan klasifikasi terhadap dokumen.

Metode Penelitian

Data yang digunakan dalam penelitian ini adalah kumpulan dari dokumen jurnal berbahasa inggris. Datayang telah terkumpul selanjutnya diproses melalui tahap pengolahan. Adapun yang termasuk dalam lingkup pengolahan data, yaitu: (1) Mengolah dokumen dengan melalui tahap-tahap dari *text mining*, mulai dari tokenisasi, *filtering*, dan *stemming* dengan menggunakan algoritma porter. (2) Pemberian bobot terhadap setiap token

hasil pengolahan *text mining* dengan menggunakan *term frequency-inverse document frequency* (TF-IDF). (3) Menganalisis data dari perhitungan dengan menggunakan metode *cosine similarity* untuk mengetahui tingkat similaritas antar dokumen. (3) Klasifikasi dengan menggunakan *K-Nearest Neighbor*. Dengan melakukan klasifikasi berdasarkan jumlah mayoritas terbanyak dari dokumen training yang mucul dalam lingkup nilai *k* yang telah ditentukan pengguna atau dengan menggunakan nilai *cosine similarity* terbesar. (4) Evaluasi berdasarkan nilai *recall, precision* dan *f-Measure* untuk mengetahui tingkat keberhasilan sistem temu kembali klasifikasi dokumen.

Hasil

Sistem temu kembali informasi bisa menjadi solusi dari permasalahan pesatnya perkembangan teknologi informasi agar bisa memudahkan pengguna dalam menemukan informasi.Sudah banyak penelitian yang membahas berbagai macam jenis dari sistem temu kembali informasi antara lain adalah: (1) Jurnal yang berjudul "Mengukur Tingkat Kesamaan Paragraf Menggunakan *Cosine similarity* untuk Mendetekai Plagiarisme". Dalam penelitian ini, sistem menggunakan metode *cosine similarity* untuk mengukur tingkat kesamaan antar paragraf. Paragraf akan dikatakan mirip jika besar sudut yang dihasilkan antara vektor paragraf yang diuji dengan paragraf pembanding memiliki besar cosinus bernilai 1(satu). Penelitian ini menghasilkan aplikasi yang dapat mendeteksi plagiarisme dari suatu paragraf dengan berdasarkan dari hasil *cosine similarity* (Isa, 2013). (2) Jurnal yang kedua berjudul "Sistem Temu Kembali Informasi dengan Metode *Vector Space Model*". Dalam penelitian ini,menghasilkan sistem yang dapat melakukan temu kembali informasidengan menggunakan metode *vector space model* (Amin, 2012). (3) Berdasarkan tinjauan pustaka mengenai penelitian sebelumnya, klasifikasi dokumen dalam penelitian ini menggunakan *cosine similarity* sebagai perhitungan jarak sebagai dasar melakukan klasifikasi dengan menggunakan *K- Nearest Neighbour.*

Text mining

Text mining lebih berfokus pada teknik dan metodelogi dalam lingkup temu kembali sistem informasi. T*ext mining* adalah suatu proses *knowledge-based* dimana pengguna berinteraksi dan bekerja dengan sekumpulan dokumen dengan menggunakan beberapa alat analisis. Dalam tahapan proses *text mining* dibagi dalam 4 tahap utama, yaitu *text processing, text transformation, feature selection,* dan *pattern discovery* (Jones, 2004). *Text preprocessing* merupakan tahap tokenisasi yang merupakan proses pemecahan teks menjadi bentuk kata atau biasa disebut sebagai token (Jones, 2004).*Text transformation* merupakan tahap *stemming* yang berfungsi untuk merubah kata-kata yang berimbuhan menjadi bentuk dasar. Pada proses ini akan menggunakan algoritma porter. Karena algoritma porter adalah algoritma yang sesuai dengan dokumen yang berbahasa inggris. *Feature selection* merupakan tahap yang bertujuan untuk mengurangi dimensi dari kumpulan teks yang dihasilkan dari tahap transformasi, dengan kata lain, menghapus kata-kata yang tidak berkaitan dengan isi dokumen atau dengan menggunakan *stopword removal*. *Pattern discovery* merupakan tahap penentuan dari pola *test* yang akan diolah. *Supervised learning* merupakan suatu teknik pembelajaran yang menggunakan suatu label atau kategori kelas yang diberikan pada data latih (*training*) yang kemudian digunakan sebagai dasar untuk melakukan klasifikasi pada data baru.

Term frequency dan inverse document frequency

Term Frequency dan *Inverse Document Frequency* (TF-IDF) merupakan pembobotan yang sering digunakan dalam penelusuran informasi dan *text mining* (Turney dkk, 2010).*Term frequency* adalah pembobotan yang sederhana dimana penting tidaknya sebuah kata dianggap sama atau sebanding dengan jumlah kemunculan kata tersebut dalam dokumen, sementara itu *inverse document frequency* (IDF) adalah pembobotan yang mengukur penting sebuah kata dalam dokumen dilihat pada seluruh dokumen secara global. Fungsi untuk menghitung nilai TF dapat dilihat dalam persamaan 2.1 berikut ini:

$$TF(d,t) = f(d,t) \qquad (2.1)$$

Dimana :
f(d,t) : kemunculan kata *t* dalam dokumen *d*.

Pembobotan IDF menganggap bahwa bobot sebuah kata akan semakin besar jika kata tersebut semakin sering muncul dalam satu dokumen, tidak dalam banyak dokumen. Fungsi untuk menghitung nilai IDF dapat dilihat dalam persamaan 2.2 berikut ini

$$IDF(t) = \log({}^{N}/_{df(t)}) \qquad (2.2)$$

Dimana :
df(t) : Jumlah dokumen yang memiliki kata *t.*

Sedangkan fungsi untuk menghitung TF-IDF sesuai dengan rumus 2.3.

$$TFIDF = TF(d,t).IDF(t) \qquad (2.3)$$

Cosine similarity merupakan perhitungan yang paling umum digunakan dalam perhitungan similarity antar dokumen. Jika x dan y merupakan vektor dokumen, maka.

$$\cos(x,y) = \frac{x.y}{||x||\,||y||} \qquad (2.4)$$

Dimana, x.y merupakan dot product dengan persamaan

$$x.y = \sum_{k=1}^{n} x_k y_k \qquad (2.5)$$

Sedangkan ||x|| merupakan panjang dari vektor x, dimana persamaan dari ||x|| adalah

$$||x|| = \sqrt{\sum_{k=1}^{n} x_k^2} = \sqrt{x.x} \qquad (2.6)$$

K-*Nearest Neighbor* (KNN) merupakan salah satu metode dari vector space model yang berfungsi untuk mengklasifikasikan suatu objek[8]. KNN melakukan klasifikasi dengan berdasarkan ada penentuan *boundary* atau ruang lingkup secara lokal. Pada dasarnya klasifikasi KNN akan berdasarkan pada suatu hipotesis dimana suatu dokumen tes *d* akan memiliki kategori atau label yang sama dengan kategori dari dokumen *training* yang berposisi dalam ruang lingkup sebesar *k* yang mengelilingi dokumen *d.* Algoritma dari klasifikasi *k-Nearest Neighbor* dapat dilihat dalam gambar 2.2

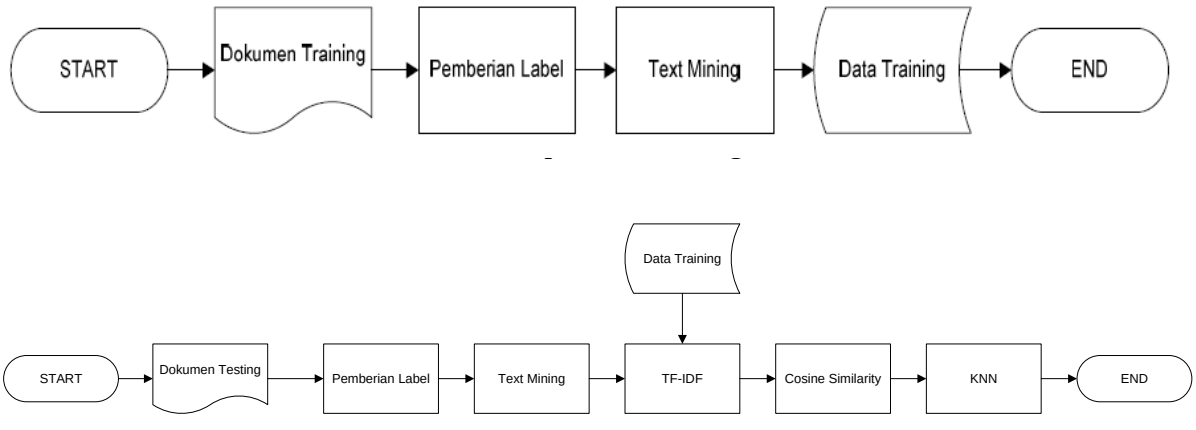
```
TRAIN-KNN(C, D)
1  D' ← PREPROCESS(D)
2  k ← SELECT-K(C, D')
3  return D', k

APPLY-KNN(C, D', k, d)
1  S_k ← COMPUTENEARESTNEIGHBORS(D', k, d)
2  for each c_j ∈ C
3  do p_j ← |S_k ∩ c_j|/k
4  return arg max_j p_j
```

Gambar 2. *Algoritma K-Nearest Neighbor*

Untuk klasifikasi dengan menggunakan 1NN, klasifikasi akan berjalan tidak begitu kuat. Klasifikasi dari setiap dokumen uji bergantung pada kelas dari setiap dokumen training, yang bisa terjadi kesalahan ketegori atau *atypical*. Klasifikasi dengan menggunakan KNN dengan $k > 1$ akan lebih kuat. Untuk parameter k dalam KNN sering dipilih berdasarkan pada pengalaman atau pengetahuan tentang masalah klasifikasi yang dihadapi. Lebih diutamakan untuk menggunakan nilai k yang memberikan hasil ganjil untuk mengurangi kemungkinan munculnya kategori berbeda dalam jumlah yang sama atau jumlah yang seri (Manning dkk, 2008).

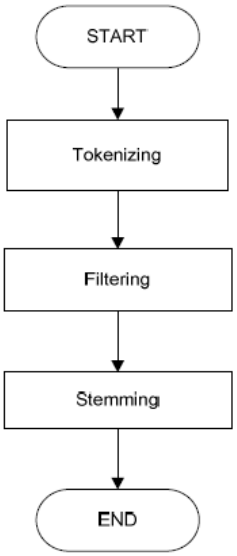
Pengumpulan data dan informasi dilakukan dengan mengambil jurnal berbahasa Inggris yang telah disediakan oleh situs penyedia jurnal Internasional *sciencedirect.com*. Data yang terkumpul sebanyak 180 dokumen yang terdiri atas 40 dokumen training untuk setiap kategori, dan 20 dokumen testing yang terdiri atas 4 Jurnal *Physical Sciences and Engineering*, 5 jurnal *Life Science*, 5 jurnal *Health Science*, dan 6 jurnal *Social Sciences and Humanities.*

Sistem dalam klasifikasi jurnal ini memiliki alur dengan tahap pertama melalui pembelajaran sistem atau training system dengan tahapan yang dapat diliat dalam Gambar 2Setelah melalui tahap pembelajaran, kemudian dilanjutkan dengan tahap pengujian atau testing system, dimana alur dari proses pengujian sistem dapat dilihat dalam Gambar 2:



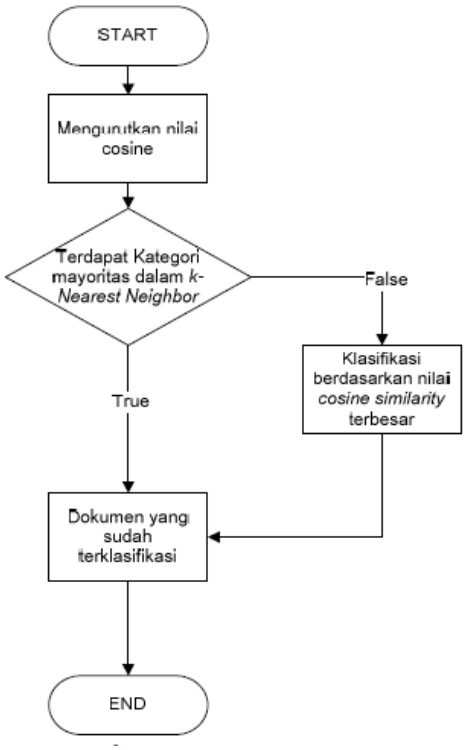
Gambar 3. Alur proses Training Sistem

Pada tahap pengolahan *text mining* sesuai dengan alur pada Gambar 3 dokumen yang diolah akan dirubah kedalam bentuk token yang paling sederhana.



Gambar 4. Alur proses Text Mining

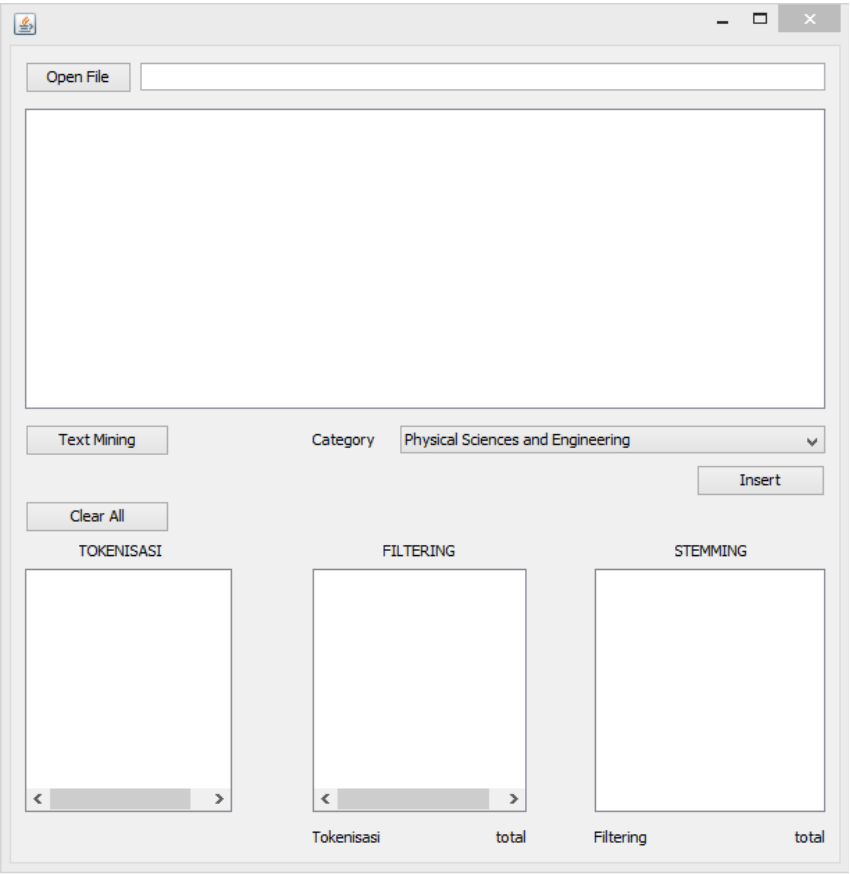
Kemudian token dari dokumen tersebut akan diberikan nilai atau bobot dengan menggunakan TF-IDF sesuai dengan persamaan 2.3 yang akhirnya akan didapatkan suatu nilai *cosine similarity* antara dokumen training dengan dokumen *query* sesuai dengan persamaan 2.4. Setelah mendapatkan nilai cosine similarity antara dokumen query dengan setiap dokumen training, klasifikasi dengan menggunakan *K-Nearest Neighbor* sesuai dengan alur yang tertera dalam Gambar 5.



Gambar 5. Alur proses K Nearest Neighbor

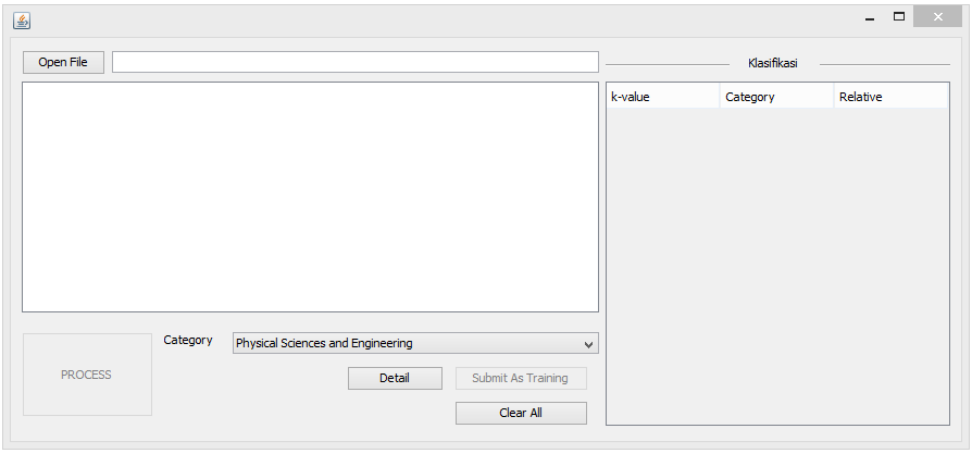
Implementasi Sistem

Pada awal sistem berjalan, *user* diminta untuk melakukan pelatihan atau training pada sistem selanjutnya system melakukan pengolahan terhadap dokumen training. Form untuk melakukan pengolahan dokumen training dapat dilihat sesuai dengan gambar 6:



Gambar 6. Form Pengolahan Dokumen Training

Setelah user memasukan dokumen training dengan jumlah yang samadalam setiap kategori, kemudian user dapat melakukan pengolahan dokumen baru atau dokumen *query* untuk diklasifikasikan. Selanjutnya dokumen *query* dikategorikan kedalam salah satu dari empat kategori dengan melakukan perbandingan antara dokumen *query* dengan dokumen training yang telah diolah sebelumnya. *form* dari proses pengklasifikasian dokumen baru dapat dilihat dalam gambar 7 dibawah ini.



Gambar 7. Antarmuka Pengklasifikasian Dokumen baru/query

Uji coba sistem

Dalam melakukan uji coba terhadap sistem, dilakukan dengan menggunakan 20 dokumen *query*. Uji coba sistem kemudian dilakukan dengan cara menghitung nilai dari *recall, precision* dan *F-measures*. Sebagai contoh, *input* dari jurnal yang berjudul "*PTSD symptom severity is associated with increased recruitment of top-down attentional control in a trauma-exposed sample*?" yang merupakan jurnal dengan kategori *Physical Sciences and Engineering.*Dengan menggunakan nilai *k* = 37 seperti yang dapat dilihat dalam tabel 4.9 menghasilkan nilai *recall* sebesar 0.35, *precision*sebesar 0.378 , dan *F-measures* sebesar 0.364 dengan perhitungan sebagai berikut.

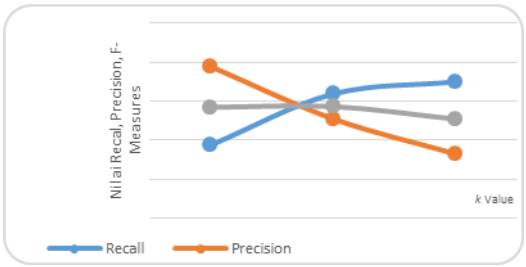
$$Recall = \frac{Jumlah\ dokumen\ relevan\ yang\ didapatkan\ sistem}{Jumlah\ dokumen\ dalam\ database} = \frac{14}{40} = 0.35$$

$$Precision = \frac{Jumlah\ dokumen\ relevan\ yang\ didapatkan\ sistem}{Jumlah\ dokumen\ yang\ didapatkan\ sistem} = \frac{14}{39} = 0.378$$

$$F - Measures = \frac{2 \times (Recall \times Precision)}{Recall + Precision} = \frac{2 \times (0.35 \times 0.378)}{0.35 + 0.378} = 0.364$$

Evaluasi sistem

Evaluasi sistem dilakukan dengan tujuan untuk melihat tingkat keberhasilan dari sistem temu kembali informasi klasifikasi jurnal. Evaluasi sistem dilakukan dengan melihat dari hasil perhitungan *recall, precision*, dan *F-Measures* terhadap hasil uji coba yang telah dilakukan dengan menggunakan 20 dokumen testing dan dengan menggunakan nilai *k* = {37, 41, 43}. Dari hasil uji coba dengan menggunakan nilai *k* tersebut didapatkan hasil nilai rata-rata dari *recall, precision*, dan *F-measures* yang tercantum dalam Gambar 8:



Gambar 8. Grafik Nilai Rata-Rata *Recall, Precision,dan F-Measures* hasil uji coba sistem

Untuk lebih jelasnya dapat dilihat dalam tabel 1 yang berisi nilai rata-rata *recall, precision,* dan *F-Measures* dari hasil uji coba.

Tabel 1 Nilai Rata-Rata *Recall, Precision,* dan *F-Measures*

k-Value	*F-Measures*	*Recall*	*Precision*
37	0.498	0.538	0.5169
41	0.524	0.511	0.5173
43	0.539	0.501	0.5193

Berdasarkan pada data yang tercantum dalam Tabel 4.1,maka implementasi dari sistem temu kembali informasi klasifikasi jurnal dapat bekerja dengan baik ketika menggunakan metode *k-nearest neighbor* (KNN) dilakukan dengan menggunakan nilai *k* = 43. Nilai *F-Measures* yang dihasilkan oleh *k* = 43 sebesar 0.5193 lebih besar jika dibandingkan dengan *k*=37 dengan *F-Measures* sebesar 0.5169 dan *k* = 41 dengan *F-Measures* sebesar 0.5173.

Simpulan

Evaluasi sistem dari penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa merancang dan membangun sistem klasifikasi dokumen dengan menggunakan metode *K-nearest neighbour* dan *cosine similarity* sebagai perhitungan jarak menghasilkan tingkat keberhasilan f-measures sebesar 0.5193 dengan nilai k=43. Data yang digunakan dalam sistem klasifikasi dokumen berjumlah 160 dokumen training dengan pembagian 40 dokumen untuk setiap kategori yaitu *physical sciences and engineering*, *life science*, *health science*, dan *social sciences and humanities.* Untuk dokumen yang digunakan sebagai testing sebanyak 20 dokumen yang terdiri dari 4 dokumen berkategori *Physical Sciences and Engineering*, 5 dokumen berkategori *life science*, 5 dokumen berkategori *Health Science*, dan 6 dokumen berkategori *social sciences and humanities.*

Dari hasil uji coba sistem didapatkan hasil bahwa 20 dokumen testing yang digunakan,dapat terklasifikasi sesuai dengan kategori sebenarnya. Namun pada beberapa kasus pada dokumen 4, 6, 9, 10, 12, 13, 14, dan 16 terjadi kesalahan dalam melakukan klasifikasi. Hal ini dikarenakan selain nilai k yang digunakan,juga karena jumlah dari token dalam dokumen training yang mempengaruhi nilai *cosine similarity* dokumen uji dan pada akhirnya mempengaruhi hasil klasifikasi KNN. Evaluasi sistem menunjukan tingkat keberhasilan dalam mengklasifikasikan dokumen dengan dengan nilai k

= 43 yang menghasilkan nilai recall sebesar 0.539 , precision sebesar 0.501, dan F-Measures sebesar 0.5193.

Saran dari penelitian ini yaitu (1) Untuk meningkatkan tingkat keberhasilan sistem klasifikasi dapat menambahkan jumlah dokumen training dalam setiap kategori agar dapat membedakan dengan kategori lain. Selain itu juga berpengaruh pada jumlah dokumen yang relevan agar padat lebih banyak mucul dalam lingkup nilai *k*. (2) Dalam menentukan nilai *k* pada klasifikasi *k-nearest neighbor* dapat menggunakan metode tambahan yaitu *k-fold cross validation.*

Referensi

Kowalski, G. J. (2000). *Information storage and retrieval systems: theory and implementation*. United States of America.

Isa, T. M. (2013). Mengukur tingkat kesamaan paragraf menggunakan vector space model untuk mendeteksi plagiarisme, *Seminar Nasional dan ExpoTeknik Elektro 2013***.** *Banda Aceh: FMIPA, Universitas Syiah Kuala.*

Amin, F. (2012). Sistem temu kembali informasi dengan metode vector space model. *Jurnal Sistem Informasi Bisnis, 2*.

Jones, K. S. (2004). A statistical interpretation of term specify and its application in retrieval. *Journal of Documentation, 60*(2), 493-502.

Turney, P. D., Pantel, & Patrick. (2010). From frequency to meaning: Vector Space Models of Semantics. *Journal of Artificial Intelegence Reseach, 37*, 141-188.

Manning, C. D, Raghavan, P., & Schutze, H. (2008). *An introduction to information retrieval*. Cambridge: Cambridge University Press.