Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 3, No. 1, Januari 2019, hlm. 931-939 http://j-ptiik.ub.ac.id

Klasifikasi *Multilabel* Menggunakan Metode *Fuzzy Similarity K-Nearest Neighbor* Untuk Rekomendasi Pencarian Artikel *Online*

Wahyuni Lubis[1], Yuita Arum Sari[2], Mochammad Ali Fauzi[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]wahyunilubis22@gmail.com, [2]yuita@ub.ac.id, [3]moch.ali.fauzi@ub.ac.id

Abstrak

Artikel adalah karya tulis dari pendapat seseorang yang membahas suatu masalah tertentu yang bersifat aktual dan terkadang bersifat kontroversial untuk memberitahu, mempengaruhi, meyakinkan, dan menghibur pembaca. Perkembangan teknologi yang pesat menyebabkan banyaknya artikel yang ditulis secara *online*. Setiap artikel *online* memiliki *label* yang berbeda-beda, dan memungkinkan setiap artikel memiliki lebih dari satu *label*. Jumlah artikel *online* yang ada di internet setiap harinya semakin bertambah yang membuat pembaca kesulitan dalam menemukan informasi yang diinginkan. Klasifikasi yang tepat bisa meningkatkan kualitas *information retrieval*. Metode *Fuzzy Similarity K-Nearest Neighbor* adalah metode klasifikasi *multilabel* yang menggabungkan metode *Fuzzy Similarity Measure* dan MLKNN. Pada penelitian sebelumnya metode FSKNN memiliki kecepatan lebih baik dalam melakukan komputasi *k* tetangga terdekat dan performa yang lebih baik dari metode MLKNN. Langkah-langkah yang dilakukan pada penelitian ini adalah melakukan *text preprocessing*, pembobotan, *clustering* dokumen, klasifikasi dan proses pencarian. Pada penelitian ini didapatkan nilai optimal *F1* dan *BEP* sebesar 0,933 dan 0,937 pada *k*=1 dan *alpha*=0,5. Pada proses rekomendasi pencarian artikel *online* menggunakan metode FSKNN didapatkan nilai *precision* tertinggi sebesar 0,5 dan *recall* 0,8. Dari hasil *F1* dab *BEP* yang didapat, menunjukkan bahwa metode FSKNN cukup baik untuk melakukan klasifikasi *multilabel* artikel *online*.

Kata kunci: *artikel online, klasifikasi multilabel, FSKNN, information retrieval*

Abstract

The article is someone's opinion of the paper that addresses a specific problem that is actual and sometimes controversial to inform, influence, persuade, and entertain the reader. Rapid technological developments led to the large number of articles written online. Each article has a different label online, and allows each article has more than one label. The number of online articles that exist on the internet every day growing which makes the reader's difficulty in finding the desired information. The proper classification can improve the quality of information retrieval. A method of Fuzzy K-Nearest Neighbor Similarity is a method that combines the multilabel classification method of Fuzzy Similarity Measure and MLKNN. Previous research on method FSKNN has better speed in doing computing k nearest neighbors and better performance of the method MLKNN. The steps undertaken in this research is conducting a text preprocessing, document clustering, weighting, classification and search process. On the research of the optimal values obtained this F1 and BEP amounted to 0.933 and 0.937 at k = 1 and alpha = 0.5. On the recommendation of the search articles online using the method FSKNN obtained the highest precision value of 0.5 and 0.8 recall. From the results of F1 and the BEP obtained, indicating that the method FSKNN was kind enough to do a multilabel classification articles online.

Keywords: *articles online, multilabel classification, FSKNN, information retrieval*

1. PENDAHULUAN

Artikel adalah karya tulis dari pendapat seseorang yang membahas suatu masalah tertentu yang bersifat aktual dan terkadang bersifat kontroversial untuk memberitahu, mempengaruhi, meyakinkan, dan menghibur pembaca (Sumadiria, 2004). Penulis artikel bisa dilakukan oleh perorangan atau beberapa orang. Isi dari suatu artikel bersumber pada fakta yang

bersifat faktual dengan mengungkapkan data-data yang diketahui oleh penulis.

Pada zaman dengan perkembangan teknologi yang sangat pesat khususnya internet yang memberi kemudahan untuk masyarakat dalam penggunaanya, mengubah pola interaksi masyarakat dengan memanfaatkan internet sebagai sarana komunikasi, publikasi, serta sarana untuk memperoleh bermacam-macam informasi (Yusup dan Subekti, 2010 : 57). Banyak para penulis artikel yang memanfaatkan keadaan ini. Penulis artikel tidak lagi hanya memanfaatkan media buku atau majalah untuk mempublikasikan karyanya. Media *online* menjadi salah satu media yang memudahkan bagi individu untuk menuliskan sebuah artikel.

Banyaknya informasi yang ada di internet membuat pembaca merasa kebingungan untuk mencari informasi yang sesuai (Sukma , Zaman dan Purwanti, 2015). Setiap situs yang menyediakan artikel *online* sudah memberikan kemudahan terhadap pembaca dengan memberikan fitur *label* terhadap setiap artikel. Adanya fitur tersebut para pembaca sedikit dimudahkan ketika melakukan pencarian terhadap artikel berdasarkan *label* yang diinginkanya. Suatu artikel yang ada pada media *online* memungkinkan memiliki lebih dari satu *label* yang sesuai. Banyak situs yang memberikan fitur pencarian dengan *query* untuk lebih mempermudah para pembaca mencari artikel yang sesuai. Fitur ini memberikan kemudahan terhadap pembaca karena hasil yang ditampilkan berdasarkan dari *query* yang diberikan *user*. Fitur pencarian berdasarkan *query* masih memiliki kelemahan, karena pengecekan *query* dan isi artikel berdasarkan frasa jika *query* lebih dari satu kata. Metode pencarian yang seperti itu membuat *user* harus benar-benar memasukkan *query* yang sangat sesuai dan tidak semua *user* paham tentang masalah *query*.

Klasifikasi merupakan suatu metode yang mengelompokkan dokumen berdasarkan kelas yang dimilikinya. Klasifikasi sering digunakan untuk memudahkan dalam proses information retrival, sebab dokumen yang berada pada satu *label* memiliki nilai kemiripan yang dekat. Pada penelitian yang dilakukan oleh Agus Zainal Arifin dan Ari Novan Setiono (2002) menyebutkan bahwa proses klasifikasi yang tepat akan meningkatkan kualitas *information retrival*.

Teknik klasifikasi yang umum dilakukan adalah mengelompokkan suatu data hanya pada satu *label*, hal tersebut bisa diterapkan jika data yang dikelompokkan hanya memiliki satu *label*. Teknik tersebut dirasa kurang cocok untuk data artikel *online* yang memungkinkan satu artikel untuk memiliki lebih dari satu *label*. Klasifikai *multilabel* adalah salah satu teknik yang bisa menyelesaikan permasalahan tersebut. Pada penelitian yang dilakukan oleh Zhang dan Zhou (2005) mengatakan bahwa permasalahan klasifikasi *multilabel* banyak dijumpai pada kehidupan nyata, contohnya adalah untuk pengelompokan teks dokumen yang memiliki lebih dari satu label.

Teknik klasifikasi *multilabel* memiliki banyak metode, diantaranya adalah FSKNN yang merupakan gabungan dari metode fuzzy similarity measure (FSM) dan K-Nearest Neighbor. Metode FSKNN melakukan *clustering* terlebih dahulu menggunakan metode FSM dan dilanjutkan dengan proses klasifikasi menggunakan metode K-NN. Pada penelitian yang dilakukan oleh Jiang,Tsai dan Lee (2012) menggunakan metode FSKNN memiliki keunggulan dari metode ML-KNN. Evaluasi dari penelitian tersebut dilihat dari segi F1 dan BEP. Metode ML-KNN memiliki nilai F1 dan BEP sebesar 96,83% sedangkan metode FSKNN memiliki nilai F1 dan BEP sebesar 97,27%.

Berdasarkan penjelasan pada latar belakang diatas penelitian ini mengambil judul "Klasifikasi *Multilabe*l Menggunakan Metode *Fuzzy Similarity K-Nearest Neighbor* Untuk Rekomendasi Pencarian Artikel *Online*".

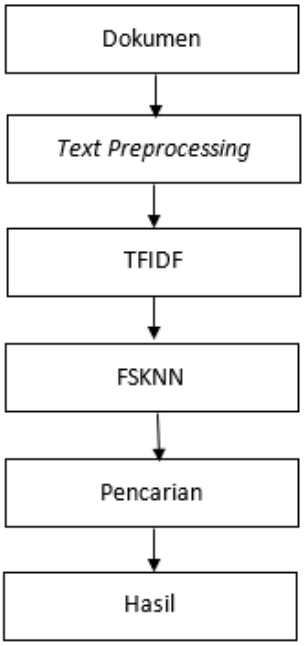
2. METODOLOGI PENELITIAN

2.1. *Dataset*

Dataset yang digunakan berupa dokumen artikel *online* yang diambil dari situs www.kompasiana.com sebanayak 100 artikel. Data yang digunakan hanya terdiri dari 3 *label* yaitu politik, ekonomi dan regional. Dari keseluruhan data yang diambil, sebanyak 80 dokumen digunakan sebagai data *training* dan 20 data *testing*.

2.2. Metode

Pada proses klasifikasi *multilabel* artikel *online* menggunakan metode FSKNN untuk rekomendasi pencarian memiliki beberapa tahapan. Berikut merupakan rancangan diagram alur dari sistem yang digambarkan pada Gambar 1.



Gambar 1. Diagram Alur Sistem

2.2.1 *Prepocessing*

Data yang dimasukkan pada awalnya berupa data yang tidak terstruktur sehingga dibutuhkan proses *text preprocessing* untuk mengolah data yang berupa teks menjadi terstruktur. *Text preprocessing* memiliki beberapa tahapan yang berurutan yang dimulai dari proses *case folding*, *tokenizing*, *filtering* dan *stemming*.

Tahap awal yang dilakukan adalah *case folding* yaitu mengubah seluruh kata menjadi huruf kecil (Manning, Raghavan dan Schütze, 2009). Proses ini digunakan unutuk menyeragamkan seluruh kata karena data yang tidak terstruktur seperti teks dokumen memiliki kata-kata yang mengandung huruf kapital. Selanjutnya adalah proses tokenizing yaitu tahapan dimana data teks dilakukan pemotongan menjadi satu kata atau per token.

Proses *filtering* adalah suatu proses yang membuang atau menghapus kata pada dokumen yang dianggap tidak penting berdasarkan *stopword list* dari tala. Penghapusan kata berdasarkan kamus yang disusun dan berisi kata-kata yang perlu dihapus. Selanjutnya adalah proses *stemming* yaitu suatu proses yang digunakan untuk membentuk kata dasar. Proses *stemming* disini bertugas untuk membuang imbuhan-imbuhan tersebut sehingga diperoleh suatu kata dasar. Algoritma *stemming* yang digunakan adalah algoritma Nazier.

2.2.2 *Term Frequency Inverse Document Frequency* (TFIDF)

Pembobotan TFIDF merupakan metode untuk pembobotan yang paling umum digunakan untuk menggambarkan dokumen ke dalam model ruang vektor. TFIDF dapat digunakan pada permasalahan information retrieval. TFIDF juga di kenal efisien, mudah dan memiliki hasil yang akurat. TFIDF bisa diimplementasikan untuk pemodelan vektor klasifikasi teks ataupun pengelompokan teks. Metode pembobotan ini akan menghitung nilai dari *term frequency* (TF) dan juga menghitung nilai dari *Inverse Document Frequency* (IDF) pada setiap kata di dalam dokumen. *Term frequency* merupakan pembobotan dengan menghitung banyaknya kemunculan kata pada setiap dokumen.

$$W_{TF\ (t,d)} = 1 + log(tf_{(t,d)}) \quad (1)$$

Keterangan:
$W_{TF_{(t,d)}}$: Pembobotan *term frequency*
$tf_{(t,d)}$: frekuensi kemunculan kata *t* pada suatu dokumen *d*

Sementara pada *inverse document frequency* (IDF) merupakan pembobotan yang mengukur seberapa pentingnya sebuah kata dalam dokumen dilihat dari keseluruhan dokumen secara global (Purwanti, 2015).

$$W_{IDF\ (t,d)} = log\frac{N}{df_{(t,d)}} \quad (2)$$

Keterangan:
$W_{IDF\ (t,d)}$: pembobotan IDF (Invers Document)
N : banyaknya jumlah dokumen
$df_{(t,d)}$: banyaknya jumlah dokumen yang memiliki kata *t* pada dokumen

Setelah diketahui nilai dari $W_{TF\ (t,d)}$ dan juga $W_{IDF\ (t,d)}$ maka selanjutnya akan menghitung nilai dari TFIDF.

$$TFIDF_{(t,d)} = W_{TF\ (t,d)} \times W_{IDF\ (t,d)} \quad (3)$$

2.2.3. *Fuzzy Similarity Measure K-Nearest Neighbor* (FSKNN)

Metode FSKNN adalah salah satu metode klasifikasi yang bisa digunakan untu melakukan pengelompokan dokumen *multilabel*. FSKNN merupakan gabungan dari metode *fuzzy similarity measure* dan *k-nearest neighbors*. Tahapan pertama yang dilakukan adalah meakukan *clustering* terlebih dahulu pada proses *training*. Langkah-langah yang dilakukan dalam

tahap *training* antara lain:

1. Menghitung distribusi kata dan dokumen pada setiap *label*.

$$dt(t_i, c_j) = \frac{\sum_{v=1}^{l} w_{iv} y_{jv}}{\sum_{v=1}^{l} w_{iv}} \quad (4)$$

$$dd(t_i, c_j) = \frac{\sum_{v=1}^{l} sgn(w_{iv}) y_{jv}}{\sum_{v=1}^{l} y_{iv}} \quad (5)$$

Keterangan :
$dt(t_i, c_j)$: distribusi kata ke- *i* terhadap *label j*
w_{iv} : bobot kata ke- *i* pada dokumen ke- *v*
y_{jv} : vector label ke- *j* pada dokumen ke- *v*
$dd(t_i, c_j)$: distribusi dokumen terhadap *label j*

2. Menghitung nilai keanggotaan setiap kata pada setiap *label*.

$$\mu_R(t_i, c_j) = \frac{dt(t_i, c_j)}{\max dt(t_u, c_v)} \times \frac{dd(t_i, c_j)}{\max dd(t_i, c_v)} \quad (6)$$

Keterangan :
$\mu_R(t_i, c_j)$: derajat keanggotan kata ke- *i* terhadap *label j*
$dt(t_i, c_j)$: distribusi kata ke- *i* terhadap *label j*
$dd(t_i, c_j)$: distribusi dokumen terhadap *label j*

3. Menghitung nilai keanggotaan setiap kata terhadap setiap dokumen.

$$\mu_d(t_i) = \frac{w_i}{max_{1 \le v \le m} w_v} \quad (7)$$

Keterangan :
$\mu_d(t_i)$: derajat keanggotaan stiap kata ke- *i* terhadap dokumen
m : jumlah kata

4. Menghitung nilai *similarity* setiap dokumen terhadap setiap *label*.

$$sim(d, c_j) = \frac{\sum_{i=1}^{m} \mu_R(t_i, c_j) \otimes \mu_d(t_i)}{\sum_{i=1}^{m} \mu_R(t_i, c_j) \oplus \mu_d(t_i)} \quad (8)$$

Keterangan:
$sim(d, c_j)$: Nilai kemiripan dokumen ke- *d* terhadap *label j*
$\mu_R(t_i, c_j)$: derajat keanggotan kata ke- *i* terhadap *label j*
$\mu_d(t_i)$: derajat keanggotaan setiap kata ke- *i* terhadap dokumen

5. Menghitung nilai keanggotaan dokumen pada setiap *cluster*.

$$\mu_{c_j}(d) = \frac{sim(d, c_j)}{max_{1 \le v \le p} \text{sim(d, } c_v)} \quad (9)$$

Keterangan :
$\mu_{c_j}(d)$: derajat keanggotan dokumen ke- *d* terhadap *cluster j*
p : jumlah *label*

6. Melakukan *clustering* dokumen ke dalam *cluster* yang sesuai. Jika nilai derajat keanggotaan setiap dokumen terhadap suatu *cluster* lebih besar sama dengan nilai *threshold*, maka dokumen tersebut merupakan anggota dari *cluster* yang dimaksud.

$$S_v = \{d_u | \mu_{c_v}(d_u) \ge \propto, 1 \le u \le l\} \quad (10)$$

Keterangan :
S_v : himpunan *cluster* ke- *v*
l : jumlah dokumen

7. Menghitung nilai *prior*.

$$P(H_j = 1) = \frac{S + \sum_{i=1}^{l} y_{ji}}{2S + l} \quad (11)$$

$$P(H_j = 0) = 1 - P(H_j = 1) \quad (12)$$

Keterangan :
$P(H_j = 1)$: peluang *label* ke- *j* berniali 1
S : konstanta antara 0 sampai 1
y_{ji} : nilai l*abel* ke-*j* data ke-*i*
$P(H_j = 0)$: peluang *label* ke-*j* bernilai 0

8. Menemukan *k* tetangga terdekat pada himpunan pencarian yang telah terbentuk. Perhitungan ini memanfaatkan pembobotan tf-idf.

$$n_j^t = \sum_{r=v1}^{v_k} y_{jr} \quad (13)$$

Keterangan :
n_j^t : label *vector t* tetangga terdekat
y_{jr}: label *vector* ke- *j* pada dokumen ke- *r*

9. Menghitung nilai *likelihood.*

$$Z(e, j) = \sum_{i=1}^{l} y_{ji} \delta_{ei}(j) \quad (14)$$

$$\tilde{z}(e, j) = \sum_{i=1}^{l} \tilde{y} j i_{\delta_{ei}}(j) \quad (15)$$

$$\delta_{ei}(j) = \begin{Bmatrix} 1 \; if \; e = n_j^i \\ 0 \; if \; e \ne n_j^i \end{Bmatrix} \quad (16)$$

$$P(E = e | H_j = 1) = \frac{S + Z(e, j)}{(k+1)S + \sum_{v=0}^{k} Z(v, j)} \quad (17)$$

$$P(E = e | H_j = 0) = \frac{S + \tilde{Z}(e, j)}{(k+1)S + \sum_{v=0}^{k} \tilde{Z}(v, j)} \quad (18)$$

Keterangan :
$P(E = e | H_j = 1)$: peluang label ke- *j* bernilai 1 jika nilai *E* = *e*
$P(E = e | H_j = 0)$: peluang label ke- *j* bernilai 0 jika nilai *E* = *e*
e : 0, 1, ……, *k* tetangga terdekat

Pada tahap yang nomor 8 untuk menemukan *k* tetangga terdekat perlu adanya perhitungan kemiripan antar dokumen menggunakan *vector space model*. Pengukuran kemiripan dokumen menggunakan nilai *cosine*.

$$cos\,\theta = \frac{\sum_{i=1}^{n} W_{i,j} W_{i,q}}{\sqrt{\sum_{i=1}^{n} W_{i,j}^2} \sqrt{\sum_{i=1}^{n} W_{i,q}^2}} \quad (19)$$

Keterangan:

$cos\ \theta$: nilai kemiripan antar dokumen
$W_{i,j}$: nilai bobot *term* ke– i pada dokumen ke– j
$W_{i,q}$: nilai bobot *term* ke- i pada dokumen ke – q

2.2.4 Pencarian

Pada proses pencarian, *input* yang diberikan adalah sebuah *query* yang selanjutnya dilakukan proses *text preprocessing*. *Query* yang diberikan dianggap sebagai sebuah dokumen baru yang selanjutnya akan dihitung bobot setiap kata dari hasi *text preprocessing*. Setelah dilakukan pembobotan maka selanjutnya *query* akan dikelompokkan atau *clustering* terhadap data *training*. Proses selanjutnya adalah mencari K tetangga terdekat dari *query* terhadap data *training* dan dilakukan proses pemberian label atau klasifikasi pada *query*. Setelah *query* memiliki nilai label, maka setiap dokumen *corpus* yang memiliki label yang sama dengan *query* akan dihitung kemiripannya menggunakan persamaan (19). Dokumen yang memiliki nilai kemiripan 10 tertinggi yang akan ditampilkan oleh sistem.

2.3. Evaluasi

Evaluasi bertujuan untuk mengetahui seberapa baik performa dari suatu metode. Berikut merupakan beberapa evaluasi yang digunakan untuk proses proses klasifikasi dan hasil pencarian.

2.3.1. *F Measure* (F1) dan *Break Even Point* (BEP)

F Measure (F1) adalah pengukuran yang menilai timbal balik antara *precision* dan *recall* atau biasa disebut *mean harmonic* (Sasaki, 2007). *Break Even Point* (BEP) merupakan suatu keadaan dimana nilai *precision* sama dengan nilai *recall* (Sebastiani,2002). Perhitungan nilai F1 dan BEP yang digunakan adalah *micro average.*

$$MicroP = \frac{\sum_{i=1}^{p} TP_i}{\sum_{i=1}^{p}(TP_i+FP_i)} \quad (20)$$

$$MicroR = \frac{\sum_{i=1}^{p} TP_i}{\sum_{i=1}^{p}(TP_i+FN_i)} \quad (21)$$

$$F1\ Measure = \frac{2\times MicroP\times MicroR}{MicroP+MicroR} \quad (22)$$

$$BEP = \frac{MicroP + MicroR}{2} \quad (23)$$

Keterangan:
P = Jumlah kategori.
$MicroP$ = *micro average precision*
$MicroP$ = *micro average recall*
TP = Jumlah *record* positif dari kategori dokumen yang diklasifikasikan sebagai positif.
FP = Jumlah *record* negatif dari kategori dokumen yang tidak diklasifikasikan sebagai positif.
FN = Jumlah *record* positif dari kategori dokumen yang diklasifikan sebagai negatif.

2.3.2. *Precision Recall* Relevansi

Precision relevansi adalah pecahan dari suatu item yang diambil merupakan item yang relevan untuk item yang diambil atau bisa disebut peluang item yng diambil adalah item yang relevan. *Recall* relevansi adalah peluang suatu item yang relevan akan diambil (Manning, et.al, 2008).

$$Precision = \frac{\sum data\ yang\ diambil\ adalah\ relevan}{\sum data\ yang\ diambil} \quad (24)$$

$$Recall = \frac{\sum data\ yang\ diambil\ adalah\ relevan}{\sum data\ yang\ relevan} \quad (25)$$

3. HASIL DAN PEMBAHASAN

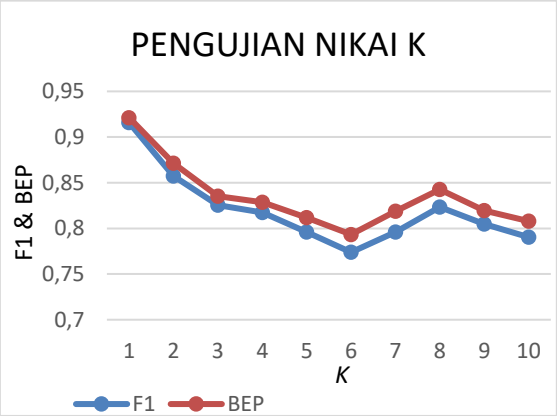
3.1 Pengujian Nilai *K* dan *Alpha*

Pada pengujian nilai *k* dan nilai *alpha*, nilai *k* dan *alpha* pada masing-masing skenario pengujian kualitas metode FSKNN dalam melakukan klasifikasi *multilabel* artikel *online* dipilih pada rata-rata *F1* dan *BEP* tertinggi. Nilai *k* dengan *F1* dan *BEP* tertinggi masing-masing adalah 0,915 dan 0,921 ketika nilai *K* = 1. Nilai *alpha* dengan *F1* dan *BEP* tertinggi ketika nilai *alpha* = 0,5 yaitu masing-masing 0,845 dan 0,855.

Tabel 1. Pengujian Nilai *K*=1

Alpha	*F1*	*BEP*
0,1	0,933	0,937
0,2	0,933	0,937
0,3	0,933	0,937
0,4	0,933	0,937
0,5	0,933	0,937
0,6	0,933	0,937
0,7	0,933	0,937

0,8	0,933	0,937
0,9	0,908	0,912
1	0,783	0,8

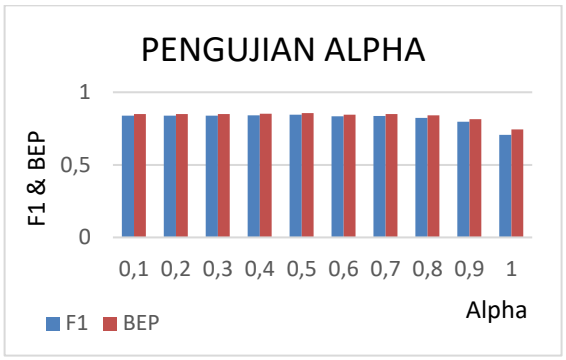


Gambar 2. Grafik Pengujian Nilai *K*

Berdasarkan pengujian nilai *K* didapatkan bahwa nilai tertinggi yaitu masing-masing *F1* dan *BEP* 0,933 dan 0,937 ketika nilai *K =1*. Nilai *K* memengaruhi ketika suatu data mencari tetangga terdekatnya menggunakan nilai kemiripan (*cosine simalrity*). Ketika nilai *K = 1* artinya suatu data hanya akan memiliki *1* tetangga yang paling dekat dengannya. Hal itu berarti bahwa saat melakukan prediksi terhadap data uji hanya dipengaruhi oleh data yang benar-benar paling mirip dengannya. Berbeda ketika nilai *K* semakin besar, maka suatu data akan memiliki jumlah tetangga terdekat yang juga banyak. Hal ini memungkinkan suatu data memiliki tetangga yang sebenarnya tidak mirip denganya. Tetanga yang sebenarnya tidak mirip tersebut nantinya akan berperngaruh dalam proses prediksi. Sehingga semakin banyak tetangga yang tidak mirip dengannya semakin besar kemungkinan suatu data akan diprediksi dengan benar.

Tabel 2. Pengujian Nilai *Alpha*=0,5

k	***F1***	***BEP***
1	0,933	0,937
2	0,875	0,875
3	0,848	0,854
4	0,838	0,848
5	0,805	0,820
6	0,783	0,8
7	0,808	0,825
8	0,861	0,875
9	0,861	0,875
10	0,836	0,85

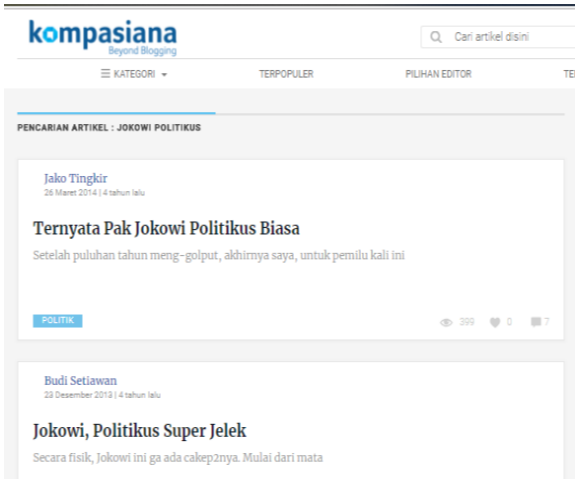


Gambar 3. Grafik Pengujian Nilai *Alpha*

Berdasarkan pengujian nilai *alpha* didapatkan rata-rata tertinggi ketika nilai *alpha = 0,5*. Nilai *alpha* merupakan ambang batas suatu data dimasukkan pada *cluster*. Semakin kecil nilai *alpha* maka suatu data akan cenderung masuk pada suatu *cluster* lebih dari satu. Pada saat suatu data masuk pada lebih satu *cluster* maka data tersebut cenderung memiliki tetangga lebih banyak yang mana nantinya akan diambil sebanyak *K* tetangga. Ketika nilai *alpha* semakin besar maka suatu data cenderung hanya masuk pada satu *cluster*. Suatu data yang hanya masuk satu *cluster* memiliki jumlah tetangga yang lebih sedikit, karena saat pengambilan *K* tetangga hanya dipilih dari tetangga yang memiliki *cluster* yang sama.

3.2 Pengujian Kualitaif Pencarian

Pada pengujian kualitatif pencarian dilakukan dengan memasukkan *query* yang terdiri dari 2 kata yang susunan frasanya diubah. Proses pertama pengujian kualitatif pencarian memasukkan *query* “Jokowi politikus” pada situs *kompasiana.com* dan selanjutnya *query* dirubah susunan frasanya menjadi “politikus Jokowi”.



Gambar 3. Hasil Pencarian "Jokowi Poltikus"

Gambar 4. Hasil Pencarian "Politikus Jokowi"

Pada proses selanjutnya adalah melakukan pencarian pada sistem dengan memasukkan *query* "politik Jokowi" dan selanjutnya *query* dirubah penyusunan frasanya menjadi "Jokowi politik".

Tabel 3. Hasil pencarian "Politik Jokowi"

No	Dokumen yang ditampilkan
1	Dokumen 44
2	Dokumen 48
3	Dokumen 64
4	Dokumen 43
5	Dokumen 56
6	Dokumen 63
7	Dokumen 18
8	Dokumen 65
9	Dokumen 08
10	Dokumen 10

Tabel 4. Hasil Pencarian "Jokowi Politik"

No	Dokumen yang ditampilkan
1	Dokumen 44
2	Dokumen 48
3	Dokumen 64
4	Dokumen 43
5	Dokumen 56
6	Dokumen 63
7	Dokumen 18
8	Dokumen 65
9	Dokumen 08
10	Dokumen 10

Berdasarkan gambar 3 dan 4 didapatkan bahwa terdapat perbedaan dari artikel yang didapatkan dari proses pencarian *kompasiana.com*. Pada proses pertama pencarian situs *kompasiana.com* menampilkan artikel-artikel yang memiliki isi yang sesuai dengan *query*, pada proses kedua tidak artikel yang ditampilkan. Hal ini disebabkan karena proses pencarian pada situs *kompasiana.com* mempedulikan susuan frasa dari *query*. Ketika *query* yang dimasukkan diubah susuan frasanya, pada pencarian situs *kompasiana.com* menganggap bahwa *query* tersebut memiliki arti yang berbeda.

Hasil proses pencarian yang dilakukan oleh situs *kompasiana.com* dapat dikatakan bahwa akan menampilkan dokumen yang benar-benar sesuai dengan frasa *query*. Ketika suatu frasa *query* yang diberikan tidak sesuai dengan dokumen yang ada, sistem tidak akan menampilkan dokumen sama sekali. Kelebihan dari hasil pencarian ini menampilkan dokumen yang sangat sesuai dengan *query* yang diberikan dan kelemahannya adalah setiap *user* memberikan *query* yang benar untuk mendapatkan hasil yang sesuai.

Berdasarkan tabel 3 dan 4 didapatkan bahwa dokumen yang ditampilkan tidak ada perbedaan meskipun susunan frasa dari *query* telah diubah yaitu sistem menampilakan dokumen 44, 48, 64, 43, 56, 63, 18, 65, 8, 10. Hal ini terjadi karena sistem tidak mempedulikan susunan frasa dari suatu *query*. Pada proses pencarian sistem hanya mempedulikan nilai bobot dari setiap katanya.

Hasil proses pencarian yang dilakukan oleh sistem akan menampilkan dokumen yang memiliki kemiripan dengan *query*. Ketika suatu *query* yang diberikan tidak memiliki frasa yang sesuai dengan dokumen yang ada, sistem akan

menampilkan dokumen yang dianggap mirip dengan *query*. Kelebihan dari pencarian ini adalah *user* tidak harus memikirkan *query* yang benar, kelemahan dari pencarian ini adalah dokumen yang ditampilkan bisa jadi tidak sesuai dengan *query* tetapi yang memiliki kemiripan dengan *query*.

3.3 Pengujian *Precision Recall* Pencarian

Pada pengujian ini akan menghitung nilai *precision* dan *recall* dari hasil pencarian yang diperoleh. Pengujian ini dilakukan dengan memberi tiga *query* yang berbeda yaitu "Politik Jokowi", "Keadaan Ekonomi Indonesia" dan "Pemilu 2018". Setiap *query* yang diberikan akan dihitung nilai *precision* dan *recall* berdasarkan hasil dokumen yang ditampilkan. Sistem ini secara umum menampilkan dokumen sebanyak sepuluh artikel *online.*

Tabel 5. Hasil Pengujian *Precision Recall* Pencarian

Query	*Precision*	*Recall*
Politik Jokowi	0,4	0,5
Keadaan Ekonomi Indonesia	0,5	0,56
Pemilu 2018	0,4	0,8

Berdasarkan tabel 5 didapatkan bahwa nilai *precision* yang diperoleh cukup kecil yaitu bernilai 0,4 dan 0,5. Hal ini berarti jika sistem kurang begitu baik dalam menolak dokumen yang tidak relevan di dalam dokumen yang diperoleh. Kemampuan sistem untuk menolak dokumen yang tidak relevan menjadi kurang baik karena proses pencarian pada sistem menggunakan metode FSKNN berdasarkan kemiripan dokumen. Proses pencarian menggunakan metode ini menyebabkan dokumen yang tidak relevan tetapi jika masih memiliki kemiripan dengan *query* yang diberikan akan ditampilkan oleh sistem.

Pada tabel 3 didapatkan bahwa nilai *recall* 0,5 pada *query* pertama, 0,56 untuk *query* kedua dan 0,8 untuk *query* ketiga. Nilai keseluruhan *recall* lebih bagus dari nilai *precision*. Hal ini berarti bahwa sistem lebih bisa menemukan dokumen yang relevan untuk ditampilkan. Kemungkinan sistem menemukan dokumen yang relevan lebih banyak menggunakan metode FSKNN karena menggunakan nilai kemiripan dokumen dan *query*, ketika secara frasa antara dokumen dan *query* tidak sesuai maka masih dipertimbangkan untuk ditampilkan jika nilai kemiripannya memenuhi.

4. KESIMPULAN

Pada proses pengujian berdasarkan nilai *F1* dan *BEP* didapatkan nilai tertinggi masing-masing 0,933 dan 0,937 dan nilai rata-rata tertinggi masing-masing 0,845 dan 0,855. Dengan nilai rata-rata diatas kualitas metode FSKNN dalam melakukan klasifikasi *multilabel* artikel *online* berhasil melakukan klasifikasi *multilabel* dengan benar sebesar *84%,* nilai ini dirasa sudah cukup baik.

Metode FSKNN mempengaruhi hasil *information retrival* karena *query* yang dimasukkan diklasifikasikan terlebih dahulu dan kemudian dokumen yang memiliki label yang sama dengan *query* dihitung kemiripan terhadap *query* dan ditampilkan 10 dokumen yang memiliki nilai kemiripan tertingggi. Metode FSKNN bisa meningkatkan nilai *recall* yang didapatkan tetapi memungkinan untuk mendapatkan nilai *precision* yang kecil.

Berdasarkan penelitian yang sudah dilakukan, masih terdapat beberapa kekurangan yang dapat diperbaiki pada penelitian selanjutnya. Adapun saran untuk penelitian selanjutnya yaitu pada tahap text preprocessing sebaiknya dilakukan perbaikan pada proses stemming karena pembentukan kata dasar yang dihasilkan masih kurang baik sehingga dihasilkan nilai evaluasi yang lebih baik, menambahkan seleksi fitur karena banyaknya fitur yang diperoleh berjumlah ribuan menyebabkan adanya fitur-fitur yang tidak relevan dan membutuhkan waktu yang cukup lama dalam proses perhitungannya.

5. DAFTAR PUSTAKA

Arifin, A., Setiono A. 2002. Klasifikasi Dokumen Berita Kejadian Berbahasa dengan Algoritma *Single Pass Clustering*. *Proceeding of SITIA ITS*

Jiang, J., Tsai, S., Lee, S. 2012. FSKNN: *Multi-label text categorization based on fuzzy similarity and k nearest neighbors*. Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 804. Taiwan

Manning, C. D., Raghavan, P. & Schutze, H. 2008. *Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press.

Sebastiani, F. 2002. *Machine learning in automated text categorization. ACM Computing Surveys*, 34(1), 1–47.

Sukma, A., Zaman, B., dan Purwanti, E., 2015. Klasifikafi Dokumen Temu Kembali Informasi *dengan K-Nearest Neghbour Information Retrieval Document Classified with K-Nearest Neighbor*. *Record and Library Journal*, Vol.1,Nomor. 2 Juli – Desember.

Sumadiria, Haris AS. 2004. Jurnalistik Indonesia, Menulis Berita dan Feature. Bandung: Simbiosa Rekatama Media.

Yusup, Pawit M., Subekti, P. 2010. Teori dan Praktik Penelusuran Informasi (*Information Retrieval).* Jakarta: Kencana.

Zhang, M., Zhou, Z. 2005. A *k-Nearest Neighbor Based Algorithm for Multi-label Classification*. *IEEE Xplore Digital Library*, 718-721.