Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 3, No. 3, Maret 2019, hlm. 2302-2311 http://j-ptiik.ub.ac.id

Rekomendasi *Multilabel* Otomatis Pada Artikel Dengan Algoritme *Fuzzy C-Means* dan *K Nearest Neighbor*

Muhammad Bima Zehansyah[1], Yuita Arum Sari[2], Sigit Adi Nugroho[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]mbimazehansyah@gmail.com, [2]yuita@ub.ac.id, [3]sigit.adinu@ub.ac.id

Abstrak

Perkembangan teknologi informasi saat ini sangat cepat khususnya pada media elektronik. Hal tersebut didukung dengan adanya sebuah wadah untuk menyalurkan suatu peristiwa, pendapat, serta gagasan yang berasal dari masyarakat yang disebut *citizen journalism* yang dikemas dalam bentuk artikel *online*. Besarnya antusiasme *citizen journalism* tersebut sayangnya kurang didukung pada pelabelan secara otomatis pada artikel yang akan dibuat, salah satunya terdapat pada situs kompasiana.com dengan adanya pelabelan otomatis diharapkan mempermudah pengguna tanpa perlu melakukan pelabelan secara manual. Salah satu cara melakukan pelabelan otomatis yaitu dengan cara melakukan klasifikasi *multilabel* yaitu memprediksi label pada suatu artikel yang memungkin artikel tersebut dapat memiliki lebih dari satu label, dengan adanya klasifikasi *multilabel* juga bertujuan dapat meningkatkan kualitas *information retrieval*. Metode klasifikasi *multilabel* salah satunya dengan menggunakan algoritme *Fuzzy C- Means dan K Nearest Neighboar* (FCM-KNN) dengan adanya proses pengelompokkan pada data diharapkan menghemat waktu komputasi dalam pencarian *k* tetangga terdekat pada proses klasifikasi *Multilabel K Nearest Neighboar* (ML-KNN). Pada penelitian ini didapatkan pengujian terbaik saat melakukan proses klasifikasi yaitu saat $k = 1$, yang mana didapatkan evaluasi F1 = 93,33 % dan evaluasi BEP sebesar 93,75%. Dari hasil didapat menunjukkan bahwa penerapan metode klasifikasi FCM-KNN dapat digunakan untuk melakukan *multilabel* secara otomatis pada artikel *online*.

Kata kunci: *artikel online, klasifikasi multilabel, FCM-KNN, information retrieval*

Abstract

The development of information technology is currently very fast especially in electronic media. It is supported by the existence of a container to channel events, opinions, and ideas that came from the community of so-called citizen journalism that is packaged in the form of articles online. The magnitude of such citizen journalism enthusiasm unfortunately less supported on labeling automatically on articles to be created, one of which is present on the site kompasiana.com with the automatic labeling is expected to facilitate the users without need to do manually labeling. One way of doing that is by way of automatic labeling do multilabel classification i.e. predict the label on an article which facilitates the article can have more than one label, with the multilabel classification is also aims can improve the quality of information retrieval. Multilabel classification method by using the Fuzzy C-Means algorithm and K Nearest Neighboar (FCM-KNN) with the process of grouping data expected to save time in the search for computing k nearest neighbors on the process Multilabel classification K Nearest Neighboar (ML-KNN). In this research the best test is obtained when performing classification process i.e. when k = 1, which obtained F1 = 93.33% of evaluation and evaluation of BEP of 93.75%. From the results obtained shows that the application of the method of classification of FCM-KNN can be used to perform the multilabel automatically on articles online.

Keywords: *articles online, multilabel classification, FCM-KNN, information retrieval*

1. PENDAHULUAN

Perkembangan teknologi informasi saat ini sangat cepat khususnya pada media elektronik. Hal tersebut didukung dengan adanya sebuah wadah untuk menyalurkan suatu peristiwa, pendapat, serta gagasan yang berasal dari masyarakat yang disebut jurnalisme warga (*citizen journalism*) yang dikemas dalam bentuk

artikel *online*. Di Indonesia sendiri artikel yang berasal dari *citizen journalism* memiliki dampak positif bagi masyarakat salah satunya berupa informasi dan pengetahuan baru bagi khalayak, dengan adanya artikel yang berasal dari *citizen journalism* juga memberikan dampak bagi masyarakat yang dahulunya menjadi entitas pasif berubah menjadi entitas aktif (Nasrullah, 2014).

Kompasiana.com merupakan salah satu *website* media sosial yang ber-*platform blog* yang sering digunakan *citizen journalism* untuk menyampaikan suatu aspirasi berupa peristiwa, pendapat, serta gagasan secara *online*. Artikel yang dibuat oleh *citizen journalism* terbukti sebagai sarana informatif dan edukatif khususnya bagi pengguna media sosial. Substansi pada artikel dapat bersumber pada fakta yang bersifat kenyataan dengan mengungkapkan data-data yang diketahui oleh perorangan ataupun kelompok tertentu. Kompasiana sendiri telah terbentuk pada tahun 2008 yang awalnya merupakan blog khusus untuk jurnalis, pada tahun 2009 berubah menjadi blog untuk *citizen journalism* (Kompasiana.com, 2018).

Antusiasme *citizen journalism* sebagai penulis artikel cukuplah besar untuk menyampaikan suatu aspirasi dalam pembuatan artikel *online*. Pada situs kompasiana.com juga memungkinkan pelabelan dapat dilakukan lebih dari satu *label*. Pelabelan otomatis salah satunya dapat dilakukan dengan klasifikasi *multilabel* yang mana cara penerapannya berbeda dengan pelabelan menggunakan klasifikasi *single label*, jika dalam klasifikasi *single label* dokumen dapat dilabelkan hanya dalam satu *label*, namun pada klasifikasi *multilabel* memungkinkan suatu artikel dapat dilabelkan lebih dari satu *label* (Zhang & Zhou, 2006).

Klasifikasi dokumen secara *multilabel* juga bertujuan untuk pencarian yang lebih efektif (Saracoglu, Tutuncu, & Allahverdi, 2008). Hal tersebut dikarenakan suatu dokumen yang telah dikelompokan telah sesuai bedasarkan labelnya. Beberapa penelitian sebelumnya yang berkaitan tentang pelabelan secara *multilabel* antara lain oleh Zhang & Zhou (2006) dimana telah berhasil mengklasifikasikan dokumen teks secara *multilabel* menggunakan metode *Multi Label K-Nearest Neighbors* (ML-KNN), yang mana suatu dokumen berada dalam koleksi dokumen akan dicari *k*-tetangga yang terdekat, lalu menggunakan metode MAP (*Maximum a Posteriori*) untuk menentukan label pada suatu dokumen teks.

Klasifikasi dokumen secara *multilabel* salah satunya dapat dilakukan menggunakan s*emi-supervised* oleh Afrianto & Kurniawati (2013) yaitu menggunakan algoritme *Fuzzy C-* Means dan *K- Nearest Neighbors* (FCM-KNN) yang bertujuan mengelompokkan dokumen terlebih dahulu sebelum proses pemberian label pada dokumen yang mana penerapan *soft clustering* yaitu *fuzzy clustering* memiliki kelebihan yaitu data dapat memiliki lebih dari 1 *cluster* yang direpresentasikan dari nilai derajat keanggotaan yang terbentuk akan tetapi penerapannya pada *Fuzzy C- Means* masi terdapat beberapa permasalahan yaitu algoritme tersebut bertipe partisional yang mana data dapat dibentuk lebih dari 2 *cluster* (kusumadewi, 2010). Penelitian tersebut membandingkan salah satu metode klasifikasi *multilabel* yaitu *Multilabel K-Nearest Neighbors* ML-KNN yang mana bertujuan menghemat waktu komputasi dalam pencarian *k* tetangga terdekat serta meningkatkan performa dalam pelabelan, hal tersebut dibuktikan pada hasil akhir performa pada metode FCM-KNN yang mana saat *k* tetangga terdekat 10 dimana hasil evaluasi pada F1 *Measure* dan *Break Even Point* (BEP) memiliki hasil yang lebih baik jika dibandingkan dengan metode ML-KNN. Penelitian lain yang menggunakan metode FCM-KNN sebagai pembelajaran multi label yaitu oleh Priandini, Zaman, & Purwanti (2017) yaitu menggunakan data jurnal internasional pada *science direct* dengan menggunakan metode FCM-KNN, dari penelitian tersebut dengan mengelompokkan dokumen terlebih dahulu dapat menghemat waktu untuk menentukan nilai k pada pemberian label, selain itu hasil evaluasi yang didapat pada F-measures dan *Precision Recall* yaitu sekitar 0,7, hal tersebut membuktikan bahwa metode FCM-KNN cukup baik dalam menghemat waktu komputasi pencarian *k* tetangga terdekat.

Bedasarkan permasalahan pada latar belakang serta penelitian sebelumnya, maka peneliti mengambil judul penelitian "Rekomendasi *multilabel* otomatis pada artikel dengan algoritme *Fuzzy C-Means* dan *K-Nearest Neighboar*".

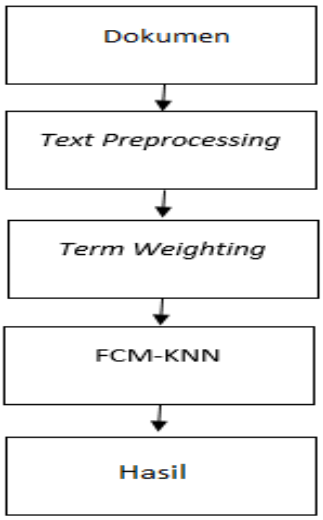
2. METODOLOGI PENELITIAN

2.1. *Dataset*

Dataset berupa dokumen artikel *online* yang memungkinkan masing-masing memiliki

label ekonomi, politik, dan regional. Data yang digunakan sebanyak 100 dataset yang terdapat pada situs *www.kompasiana.com* .

2.2. Metode



Gambar 1. Diagram Alur Sistem

Pada proses klasifikasi *multilabel* artikel *online* menggunakan metode FCM-KNN untuk pelabelan secara otomatis pada artikel yang ditunjukkan pada Gambar 1.

2.2.1 *Text Prepocessing*

Data artikel *online* yang memiliki 3 pelabelan yaitu politik, ekonomi, dan regional awalnya tidak terstruktur, proses *text prrocessing* membantu menstrukturkan data artikel *online* yang memiliki 3 label.

Langkah awal dari proses *text preprocessing* merupakan proses *case folding* yang dapat didefinisikan sebagai proses untuk peniadaan huruf selain abjad dan mengubah aksara menjadi aksara kecil. Langkah berikutnya yaitu terdapat proses *tokenizing* yaitu memotong kalimat menjadi satu *term*. Langkah selanjutnya merupakan proses *filtering* yaitu menghapus *term* yang tidak memiliki makna, *stoplist* yang digunakan merupakan *stoplist* dari (Talla, 2003). Langkah terakhir yaitu proses *stemming*, mengubah *term* menjadi kata dasar. Algoritme *stemming* yang digunakan berasal dari algoritme Nazief Andriani.

2.2.2 *Term Weighting*

Pembobotan kata dapat direpresentasikan pada metode TFIDF bertujuan mencari nilai harmoni dari pembobotan frekuensi setiap kata yang dibentuk dalam sekumpulan matriks. (Trstenjak, Mikac, & Donco, 2013).

$$W_{TF_{(i,j)}} = 1 + log(tf_{(i,j)}) \quad (1)$$

Keterangan:
$W_{TF_{(i,j)}}$: Pembobotan kemunculan setiap kata pada *term* i pada masing-masing dokumen ke j.
$tf_{(i,j)}$: Banyaknya kemunculan kata ke i pada masing-masing dokumen ke j.

$$W_{IDF_{(i,j)}} = log \frac{N_{doc}}{df_{(i,j)}} \quad (2)$$

Keterangan:
$W_{IDF_{(i,j)}}$: Pembobotan kemunculan kata pada *term* i pada keseluruhan dokumen ke j.
N_{doc} : Banyaknya jumlah dokumen.
$df_{(i,j)}$: Banyaknya kemunculan suatu *term* i pada setiap dokumen ke j.

Langkah terakhir setelah mendapatkan nilai dari persamaan 1 dan persamaan 2, menghitung nilai pembobotan TFIDF pada persamaan 3.

$$TFIDF_{(i,j)} = W_{TF_{(i,j)}} \, x \, W_{IDF_{(i,j)}} \quad (3)$$

Keterangan:
$TFIDF_{(i,j)}$: Jumlah keseluruhan bobot setiap term pada setiap dokumen.
$W_{TF_{(i,j)}}$: Pembobotan kemunculan kata pada *term* i pada dokumen ke j.
$W_{IDF_{(i,j)}}$: Pembobotan kemunculan kata pada *term* i pada keseluruhan dokumen ke j.

2.2.3. *Fuzzy C- Means Measure K-Nearest Neighbor* (FCM-KNN)

Penerapan klasifikasi *multilabel* salah satunya dapat dibangkitkan dengan algoritme FCM-KNN yang mengandung proses *clustering* dan klasifikasi. Berikut merupakan proses penerapan klasifikasi *multilabel clustering* pada tahap *training* (Kusumadewi, 2010).

1. Merancang fitur pembobotan TFIDF dalam sekumpulan matriks.

$$\begin{Bmatrix} X_{11} & \cdots & X_{1j} \\ \vdots & \ddots & \vdots \\ X_{k1} & \cdots & X_{kj} \end{Bmatrix} \quad (4)$$

Keterangan :
k : Objek atau dokumen.
j : Fitur atau *term*.

2. Membangkitkan algoritme *Fuzzy C-Means* terdapat beberapa ketentuan yaitu menetapkan jumlah *cluster*, inialisasi nilai *error*, inialisasi bobot *fuzzy* (*fuzz*, inialisasi fungsi objektif awal, dan menetapkan iterasi maksimum.

3. Merancang matriks untuk pengacakan nilai derajat keanggotaan.

$$\mu_{ik} = \begin{matrix} X_{11} & \cdots & X_{1i} \\ \vdots & \ddots & \vdots \\ X_{k1} & \cdots & X_{ki} \end{matrix} \quad (5)$$

Keterangan:
μ_{ik} : Pengacakan nilai derajat keanggotan, *i* representasi dari *cluster* dan *k* representasi dari objek.

4. Menghitung nilai *cluster center*.

$$V_{ij} = \frac{\sum_{k=1}^{n}(\mu_{ik})^m \ . \ X_{kj}}{\sum_{k=1}^{n}(\mu_{ik})^m} \quad (6)$$

Keterangan:
V_{ij} : Nilai *i* representasi dari *cluster center* dan nilai *j* representasi dari *term*.
μ_{ik}: Nilai derajat keanggotaan pada *cluster* ke- *i* dan data ke- *k*.
n : Banyaknya objek.
m : Nilai pembobotan, (pembobotan > 1).
X_{kj}: *k* representasi dari data dan *j* representasi dari bobot TFIDF setiap *term* .

5. Perhitungan nilai jarak biasanya dapat menggunakan *Euclidian Distance* yang mana mengukur jarak antara pusat cluster dengan data. Sedangkan khusus untuk data yang berupa dokumen teks, metode efektif untuk menghitung kemiripan antar dokumen dapat menggunakan *cosine simmilarity*, sebelum menghitung jarak antara dokumen dengan pusat klaster.

$$dist(d1, d2) = 1 - \cos\theta \quad (7)$$

Keterangan:
$dist(d1, d2)$: Jarak antara dokumen satu dengan dokumen lainnya.
$\cos\theta$: Nilai kesamaan antara dokumen dengan pusat *cluster*.

6. Nilai fungsi objektif awal di inialiasi nilai 0, sedangkan pada iterasi awal dan seterusnya menggunakan Persamaan 8.

$$Pn = \sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{ik})^w(d_{ik})^2 \quad (8)$$

Keterangan:
Pn : Nilai fungsi objektif.
μ_{ik} : Nilai *i* representasi dari *cluster* dan *k* representasi dari nilai data terhadap nilai derajat keanggotaan.
d_{ik} : Nilai *k* representasi dari data dan nilai *i* representasi dari *cluster* terhadap nilai jarak.

7. Memperbaiki atau *update* nilai derajat keanggotaan.

$$\mu_{ik(new)} = \left[\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}\right]^{-1} \quad (9)$$

Keterangan:
$\mu_{ik(new)}$: Nilai *i* representasi dari *cluster* dan nilai *k* terhadap nilai derajat keanggotaan.
d_{ik}: Nilai *i* representasi dari *cluster* dan nilai *k* representasi dari data terhadap nilai jarak.
d_{jk}: Nilai *j* rerpresentasi dari *cluster* lainnya dan nilai *k* representasi dari data terhadap nilai jarak.
m: Nilai bobot.

8. Proses pemberhentian iterasi.

$$|P_n - P_{n-1}| < \varepsilon \quad (10)$$

Keterangan:
Pn : Fungsi objektif
ε : Nilai *error*.

9. Nilai derajat keanggotan paling besar dengan membandingkan pada *cluster* yang dibentuk menandakan data masuk kedalam *cluster* tersebut.

10. Menghitung *sigma output label k* tetangga

terdekat.

$$n_j^t = \sum_{r=v1}^{v_k} y_{jr} \quad (11)$$

Keterangan :
n_j^t : sigma label dokumen.
y_{jr} : Nilai j representasi dari label dan nilai r representasi dari dokumen terhadap *output* label.

11. Menghitung nilai *prior probability* digunakan untuk menghitung nilai peluang label saat peluang bernilai 1 dan peluang bernilai 0.

$$P(H_j = 1) = \frac{S + \sum_{i=1}^{l} y_{ji}}{2S + l} \quad (12)$$

$$P(H_j = 0) = 1 - P(H_j = 1) \quad (13)$$

Keterangan :
$P(H_j = 1)$: Nilai j representasi dari label terhadap peluang label 1.
S : Nilai konstanta
y_{ji} : nilai l*abel* ke-*j* data ke-*i*
$P(H_j = 0)$: peluang *label* ke-*j* bernilai 0

12. Menghitung nilai *likelihood* bertujuan untuk mencari peluang label saat bernilai 1 dan bernilai 0, yang mana didasarkan pada inialisasi nilai *k* tetangga terdekat.

$$\delta_{ei}(j) = \begin{Bmatrix} 1 \; if \; e = n_j^i \\ 0 \; if \; e \neq n_j^i \end{Bmatrix} \quad (14)$$

$$Z(e,j) = \sum_{i=1}^{l} y_{ji} \delta_{ei}(j) \quad (15)$$

$$\tilde{Z}(e,j) = \sum_{i=1}^{l} \tilde{y} j i_{\delta_{ei}}(j) \quad (16)$$

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

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

Keterangan :
$P(E = e | H_j = 1)$: Nilai j representasi dari label dan nilai E = e representasi dari sigma label terhadap peluang label bernilai 1.
$P(E = e | H_j = 0)$: Nilai j representasi dari label dan nilai E = e representasi dari sigma label terhadap peluang label bernilai 0.
e : 0,1,2.., k sigma label k tetangga
S : Nilai Konstanta.
y_{ji} : Nilai j representasi dari label dan nilai i representasi dari data pada *output label*.
$\tilde{y}ji$: Nilai j representasi dari label dan nilai i representasi dari data pada *output label* negasi.
k : Nilai informasi ketetanggan terdekat.

Pada persamaann 11 untuk menemukan kemiripan antar dokumen yang mana digunakan untuk mendapatkan nilai k tetangga terdekat yaitu menggunakan nilai Persamaan 19.

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

Keterangan:
$cos\,\theta$: nilai kemiripan antar dokumen
$W_{i,p}$: nilai dokumen ke- p dari pembobotan dokumen ke- i.
$W_{i,q}$: nilai dokumen ke- q dari pembobotan dokumen ke- i.

Pada proses *testing* melakukan perkalian antara *prior* dan *likelihood* berdasarkan *sigma output label* pada data uji.

$$A = P(E = e | H_j = 0) \; P(H_j = 0) \quad (20)$$

$$B = P(E = e | H_j = 1) \; P(H_j = 1) \quad (21)$$

$$y_{ji} = \begin{Bmatrix} 0, & if \; B < A \\ 1, & if \; B > A \\ R[0,1], & if \; B = A \end{Bmatrix} \quad (22)$$

Keterangan:
y_{ji} : Nilai j representasi dari label dan nilai i representasi dari dokumen terhadap *output label*
$P(E = e | H_j = 1)$: Peluang nilai *likelihood* bernilai 1 pada masing-masing label jika E sigma *output* data uji sama seperti nilai e pada peluang nilai *likelihood*.
$P(E = e | H_j = 0)$: Peluang nilai *likelihood* bernilai 0 pada masing-masing label jika E sigma *output* data uji sama seperti nilai e pada peluang nilai *likelihood*.

$P(H_j = 0)$: Peluang *prior* saat bernilai 0.
$P(H_j = 1)$: Peluang *likelihood* saat bernilai 1.
e : 0,1,2,..,*k* representasi dari sigma label.
A : Perkalian nilai peluang *likelihood* dan nilai *prior* saat bernilai 0.
B: Perkalian nilai peluang *likelihood* dan nilai *prior* saat bernilai 1.

2.3. Evaluasi

Pengujian yang digunakan pada penelitian ini yaitu evaluasi *Silhouette Coefficient*, F1 *Measure*, dan *Break Even Point*.

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

Metode evaluasi mendapatkan korelasi antara nilai rata - rata *precision* dan nilai rata – rata *recall* direpresentasikan pada evaluasi F1 (Guns, Lioma, & Larsen, 2012). *Break Even Point* (BEP) merupakan metode evaluasi yang memiliki kondisi ketika nilai *recall* sama seperti nilai *precision* (Sebastiani, 2002). Berikut Persamaan 2.26 – 2.29 menjelaskan persamaan umum dari F1 dan BEP.

$$MicroP = \frac{\sum_{i=1}^{c} TP_i}{\sum_{i=1}^{c}(TP_i + FP_i)} \quad (23)$$

$$MicroR = \frac{\sum_{i=1}^{c} TP_i}{\sum_{i=1}^{c}(TP_i + FN_i)} \quad (24)$$

$$F1\ Measure = \frac{2 \times MicroP \times MicroR}{MicroP + MicroR} \quad (25)$$

$$BEP = \frac{MicroP + MicroR}{2} \quad (26)$$

Keterangan:
c : Jumlah label.
TP : Jumlah label dokumen yang positif dan teridentifikasi sebagai positif.
FP : Jumlah label dokumen yang negatif dan teridentifikasi sebagai positif.
FN : Jumlah label dokumen yang positif dan teridentifikasi sebagai negatif.
$MicroP$: *micro average precision*.
$MicroR$: *micro average recall*.

2.3.2 *Silhouette Coefficient*

Mengukur kualitas setiap objek yang mana menilai seberapa tepat objek tersebut berada dalam suatu *cluster* representasi dari evaluasi *Silhouette Coefficient* (Wahyuni et al, 2016).

$$a_i = \frac{1}{n_x - 1} \sum_{j=1}^{n_x} dist\left(doc_{(i,x)}, doc_{(j,x)}\right) \quad (27)$$

Keterangan:
a_i : Jarak rata-rata dokumen ke *i* dengan keseluruhan dokumen dalam satu *cluster*.
$dist\left(doc_{(i,x)}, doc_{(j,x)}\right)$: Jarak antara dokumen ke *i* dengan dokumen ke *j* pada *cluster* yang sama.

$$b_i = max\left\{\frac{1}{m_n} \sum_{r=1}^{m_n} dist\left(doc_{(i,x)}, doc_{(r,n)}\right)\right\} \quad (28)$$

Keterangan:
b_i : Jarak rata-rata dokumen ke *i* dengan keseluruhan dokumen yang berada di *cluster* lainnya.
$dist\left(doc_{(i,x)}, doc_{(r,n)}\right)$: Jarak antara dokumen ke *i* dengan dokumen ke *j* pada *cluster* yang berbeda.

2. Kemudian dari Persamaan tersebut dapat menghitung Nilai S_i dengan persamaan 26.

$$S_i = \begin{cases} 1 - \frac{b_i}{a_i} & jika\ a_i > b_i \\ 0 & jika\ a_i = b_i \\ \frac{a_i}{b_i} - 1 & jika\ a_i < b_i \end{cases}$$

Keterangan:
S_i : Nilai evaluasi pada setiap objek.
a_i : Nilai *i* representasi dari setiap objek terhadap rata-rata dari nilai kemiripan antar dokumen berada dalam *cluster* yang sama.
b_i : Nilai *i* representasi dari setiap objek terhadap rata-rata dari nilai kemiripan antar dokumen berada dalam *cluster* yang berebeda.

3. HASIL DAN PEMBAHASAN

Skenario pengujian klasifikasi multilabel yaitu pada proses clustering dalam tahap training menetapkan nilai error terkecil yaitu 1,0 x e^(-10), menetapkan maksimum iterasi yaitu berjumlah 100, dan mengatur komposisi dari clustering yang dibentuk yaitu mulai dari cluster 2 hingga cluster 6, serta nilai bobot fuzzy mulai dari nilai bobot 2 hinggan nilai 4, setiap cluster yang dibentuk dan perubahan bobot nantinya akan

diinialisasi k tetangga mulai dari nilai 1 hingga nilai 10. Pengujian dilakukan sebanyak 5 kali pada setiap cluster, bobot, dan inialisasi jumlah k untuk mendapatkan nilai rata-rata dari evaluasi F1 dan BEP, hal tersebut dilakukan karena proses pembangkitan clustering pada Fuzzy C-Means yaitu mengacak nilai derajat keanggotaan.

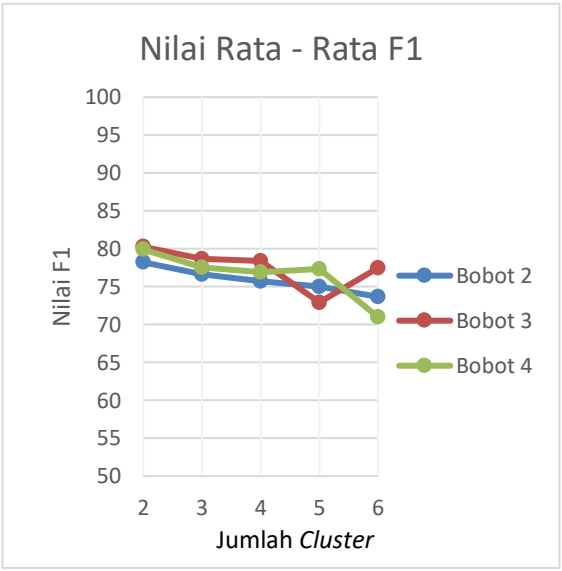
3.1 Pengujian F1 dan BEP

Bedasarkan pengujian F1 dan BEP dari hasil pengujian dari *cluster* yang dibentuk dari *cluster* 2 – *cluster* 6 , nilai bobot 2 – nilai bobot 4, dan *k* bernilai 1 hingga *k* bernilai 10, dari hasil evaluasi F1 dan BEP menunjukkan inialisasi nilai *k* tetangga yang bernilai 1 pada setiap cluster yang dibentuk dan nilai bobot yang diuji, menunjukan hasil nilai F1 dan BEP yang lebih baik jika dibandingkan dengan inialisasi k tetangga lainnya. Nilai k = 1 menunjukkan bahwa suatu data hanya memiliki 1 tetangga yang paling dekat bedasarkan nilai kemiripannya, hal tersebut juga menyebabkan bahwa saat melakukan prediksi untuk data uji hanya dipengaruhi oleh data yang benar-benar paling mirip dengannya, berbeda saat nilai k tetangga semakin besar data tidak hanya memperhatikan data paling mirip dengannya akan tetapi data juga dapat memperhatikan data yang tidak mirip dengannya.

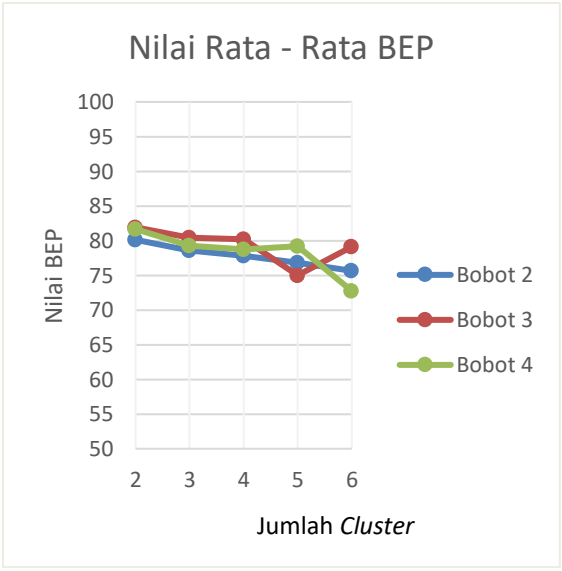
Bedasarkan pengujian F1 dan BEP dari hasil klasifikasi *multilabel* FCM-KNN, didapatkan nilai rata-rata keseluruhan nilai *k* tetangga terdekat yang telah diuji dari pengaruh *cluster* yang dibentuk dan nilai bobot yang diberikan, yang mana hasil rata-rata *k* tetangga terdekat terhadap evaluasi F1 dan BEP ditunjukan pada Tabel 1.

Tabel 1. *K* Tetangga Terhadap F1 dan BEP

No	Cluster	Bobot	F1 (%)	BEP (%)
1	2	2	78,22	80,12
2	3	2	76,63	78,56
3	4	2	75,73	77,81
4	5	2	75,04	76,83
5	6	2	73,66	75,65
6	2	3	80,26	81,92
7	3	3	78,66	80,45
8	4	3	78,42	80,16
9	5	3	72,92	75,00
10	6	3	77,46	79,09
11	2	4	79,91	81,67
12	3	4	77,55	79,30
13	4	4	76,88	78,73
14	5	4	77,32	79,21
15	6	4	70,99	72,75



Gambar 2. Grafik Evaluasi F1



Gambar 3. Grafik Evaluasi BEP

Bedasarkan hasil pengujian terbaik dari rata-rata keseluruhan inialisasi *k* tetangga terdekat

dari *cluster* yang dibentuk dan nilai bobot yang diberikan, bahwa dengan pembentukkan 2 *cluster* dengan pembobotan 3 memiliki nilai rata-rata yang cukup baik saat melakukan pencarian *k* tetangga terdekat mulai dari *k* = 1 hingga *k* = 10, yaitu nilai evaluasi F1 yang didapat bernilai 80,26 % dan BEP yaitu bernilai 81,92 %, hal tersebut dikarenakan saat data dikelompokkan menjadi 2 *cluster* pada saat data melakukan pencarian *k* tetangga tetangga terdekat, kemungkinan data melihat *k* tetangga lebih banyak sehingga saat pencarian *k* tetangga terdekat mulai dari *k* = 1 hingga *k* = 10 nilai evaluasi F1 dan nilai evaluasi BEP yang didapat menjadi optimal.

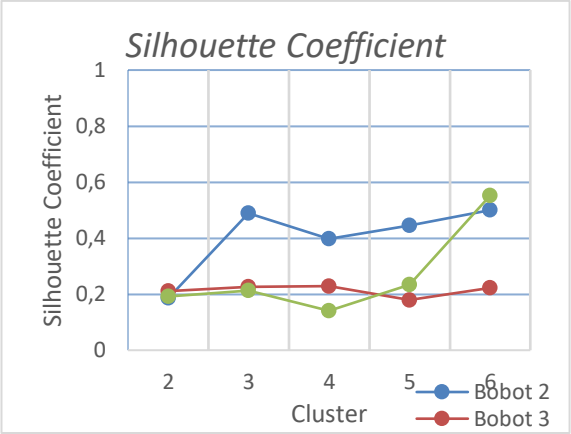
Bedasarkan rata-rata dengan pembentukan 2 *cluster* dengan pembobotan 3, dari 5 percobaan dengan rata – rata evaluasi F1 bernilai 80,26 % dan BEP yaitu bernilai 81,92 % didapatkan hasil pengujian terbaik yaitu saat *k* = 1 dengan evaluasi F1 bernilai 93,33 % dan Evaluasi BEP bernilai 93,75 %, dari hasil proses klasifkasi 20 dokumen uji jika dilihat dari segi akurasi dari perhitungan manual, 18 dokumen terklasifikasi sesuai label aktualnya, sedangkan 2 dokumen tidak terklasifikasi secara benar.

3.2 Pengujian *Silhouette Coefficient*

Skenario pengujian klasifikasi *multilabel* pada proses *clustering* pada tahap *training*, yaitu menetapkan nilai error terkecil yang diharapkan yaitu 1,0 x e^{-10}, menetapkan maksimum iterasi yaitu berjumlah 100, dan mengatur komposisi dari *clustering* yang dibentuk yaitu mulai dari *cluster* 2 hingga *cluster* 6, serta nilai bobot *fuzzy* mulai dari nilai bobot 2 hinggan nilai 4. Pengujian dilakukan dengan mengambil nilai rata-rata dari 5 kali percobaan, hal tersebut dilakukan karena proses pembangkitan *clustering* pada *Fuzzy C-Means* yaitu mengacak nilai derajat keanggotaan. Tabel 6.5 merupakan hasil pengujian menggunakan evaluasi *Silhouette Coefficient*.

Tabel 2. Pengujian *Silhouette Coefficient*

No	Jumlah *Cluster*	Nilai Bobot	Nilai *Silhouette*
1	2 *Cluster*	2	0,1856242
2	3 *Cluster*	2	0,4881274
3	4 *Cluster*	2	0,3971330
4	5 *Cluster*	2	0.4451827
5	6 *Cluster*	2	0.4998842
6	2 *Cluster*	3	0.209968
7	3 *Cluster*	3	0,2266004
8	4 *Cluster*	3	0,2287835
9	5 *Cluster*	3	0,1793087
10	6 *Cluster*	3	0,2220530
11	2 *Cluster*	4	0,1911900
12	3 *Cluster*	4	0,2119470
13	4 *Cluster*	4	0,1407078
14	5 *Cluster*	4	0,2328595
15	6 *Cluster*	4	0,5515284



Gambar 4. Grafik *Silhouette Coefficient*

Berdasarkan pada Gambar 1 dari hasil pengujian *Silhouette Coefficient*, maka dapat dilakukan analisis Pada pengujian *Silhouette Coefficient* bahwa dari pembentukan 2 *cluster* hingga 6 *cluster* dan pengujian bobot yang dimulai dari 2 -4, didapat pengujian terbaik yaitu pada *cluster* 6 dengan pembobotan 4 yaitu sebesar 0,551528, akan tetapi saat dilakukan pengujian rata - rata saat mencari *k* tetangga terdekat mulai dari *k* = 1 hingga *k* = 10 pada tabel 6.2 didapatkan nilai F1 sebesar 70,99 % dan BEP sebesar 72,95 %, nilai evaluasi tersebut memiliki rata – rata terkecil jika dibandingkan dengan pembentukan *cluster* dan pembobotan yang lain, hal tersebut dikarenakan saat dilakukan pembentukkan *cluster* 6 data cenderung lebih sedikit melihat *k* tetangga terdekatnya sehingga saat pencarian *k* tetangga terdekat mulai dari *k* = 1 hingga *k* = 10 relatif lebih kecil terhadap hasil evaluasi pada proses klasifikasi. Berbeda saat data di *cluster* 2 ruang data saat dikelompokkan

jika dilihat dari rata – rata dari *k* tetangga mulai *k* = 1 hingga *k* = 10 nilai evaluasi yang didapat sebesar 80,26 % untuk F1 dan 81,92 % untuk BEP, akan tetapi dalam pengujian *silhouette coefficient* dalam penerapan *single cluster* data cenderung tidak cocok untuk dikelompokkan menjadi 2 *cluster* karena mendapatkan nilai *Silhouette Coefficient* sebesar 0,1929926, akan tetapi secara performa dalam mencari nilai *k* = 1 hingga *k* = 10 dapat optimal karena data dapat melihat *k* tetangganya lebih banyak.

Proses *clustering* masih kurang optimal karena pada pengujian bobot yang diberi mempengaruhi terbentuknya nilai derajat keanggotaan, pembobotan yang diberi mulai dari bobot 2, 3, dan 4 masih kurang optimal karena derajat keanggotaan yang terbentuk pada setiap *cluster* masih relatif sama saat proses *clustering*, sehingga nilai *silhouette* yang didapat cenderung tidak seimbang antar *cluster*, pembobotan dapat dilakukan dengan cara mengecilkan nilai bobot *fuzzy* mulai dari range 1,1. Nilai *Silhouette Coefficient* yang baik tidak menjamin untuk mendapatkan klasifikasi yang baik jika dilihat pencarian kesuluruhan *k* tetangga terdekat hal tersebut dikarenakan *k* tetangga yang dilihat semakin sedikit.

4. KESIMPULAN

Multilabel otomatis pada artikel dapat dilakukan dengan menggunakan klasifikasi *multilabel* dengan metode *Fuzzy C- Means* dan *K Nearest Neighbor*. Data yang digunakan untuk klasifikasi *multilabel* adalah artikel *online* yang terdapat pada situs *kompasiana.com* sebanyak 100 artikel yang terdiri dari 3 *label.* Dari 100 data yang digunakan, sebanyak 80 artikel digunakan sebagai data latih untuk proses *training*, sedangkan 20 artikel digunakan sebagai data uji. Sebelum dilakukan proses klasifikasi *multilabel* artikel *online*, masing – masing pada tahap *training* dan tahap *testing* melakukan *text processing* dan pembobotan tfidf, pada tahap *training* data akan dilakukan *clustering*, menghitung matrik *output* label bedasarkan *k* tetangga terdekat, melakukan perhitungan *prior*, dan perhitungan *likelihood*, sedangkan untuk tahap *testing* data uji dilakukan proses *clustering*, menghitung sigma matrik *output* label bedasarkan *k* tetangga terdekat, melakukan proses pelabelan menggunakan perkalian nilai *prior* dan nilai *likelihood* bedasarkan sigma *output* label *k* tetangga terdekat pada data uji.

Pada proses pengujian berdasarkan evaluasi *F1* dan *BEP* didapatkan nilai rata-rata dari 5 percobaan dan rata – rata pencarian *k* tetangga mulai dari *k* = 1 hingga *k* = 10, sebesar 80,26 % dan 81,92 % dengan pembentukkan *cluster* 2 dan pembobotan 3, yang mana dari 5 percobaan tersebut didapatkan evaluasi F1 terbaik yaitu saat *k* = 1 dengan hasil evaluasi F1 sebesar 93,33 % dan BEP sebesar 93,75 %, dari segi akurasi 18 dokumen uji terklasifikasikan secara benar dan 2 dokumen tidak terklasifikasikan secara benar. Secara performa nilai F1 dan BEP dari rata keseluruhan k tetangga yang diuji, nilai evaluasi *Silhouette Coefficient* yang baik tidak menjamin untuk mendapatkan performa yang baik dalam klasifikasi, hal tersebut dikarenakan *k* tetangga yang dilihat semakin sedikit.

5. DAFTAR PUSTAKA

Afriyanto, B. R., Kurniawati, Y. L., 2013. Kategorisasi Dokumen Teks Secara Multi Label Menggunakan *Fuzzy C-Means* dan *K-Nearest Neighbors* Pada Artikel Berbahasa Indonesia. Jurnal Ilmiah Teknologi Informasi. Institut Teknologi Sepuluh November.

Guns, R., Lioma, C., Larsen, B., 2012. *The Tipping Point: F-score as a Function of The Number of Retrieved Items*. *Information Processing* and *Management*, 48(2012), 1171 – 1180. Elsevier.

Kompasiana.com, 2018. Tentang Kompasiana . Diakses dari https://www.kompasiana.com/tentang-kompasiana. [Diakses 10 Februari 2018].

Kusumadewi, S., Purnomo, H., 2010. Aplikasi Logika Fuzzy Untuk Pendukung Keputusan. Yogyakarta: Graha Ilmu.

Nasrullah, R., 2014. "*Selling" Self-Image In The Era Of New Media*. Univeritas Gajah Mada, [online] tersedia di: <https://jurnal.ugm.ac.id/jurnal-humaniora/article/view/4642/4111> [Diakses 20 Februari 2018]

Priandini, N., Zaman, B., Purwanti, E., 2017. *Categorizing Document By Fuzzy C-Means and K-nearest Neighbors Approach*. International Conference on Mathematics: Pure, Applied and Computation.

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

Tala, F. Z., 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. M.S. thesis. M.Sc. Thesis. Master of Logic Project. Institute for Logic, Language and Computation*. Universiteti van Amsterdam The Netherlands.

Trstenjak, B., Mikac, S., Dzenana, D., 2013. *KNN with TF-IDF Based Framework for Text Categorization*. Procedia Engineering 69 (2014) 1356 – 1364. Elsevier.

Wahyuni, I., Auliya, Y. A., Rahmi, A., Mahmudy, W. F., 2016. Clustering Nasabah Bank Berdasarkan Tingkat Likuiditas Menggunakan Hybrid Particle Swarm Optimization dengan K-Means. Jurnal Ilmiah Teknologi dan Informasi ASIA (JITIKA) 10(2): 24-33.

Zhang, M. L., & Zhou, Z. H., 2007. MLKNN: *A Lazy Learning Approach to Multi-Label Learning*. Pattern Recognition, 40(7), 2038 – 2048. Elsevier.