Jurnal Linguistik Komputasional
Indonesian Association for Computational Linguistics
JLK
Vol. 1, No. 1
Maret 2018

Metode Pembobotan Berbasis Topik dan Kelas untuk Berita Online Berbahasa Indonesia

Maryamah[#1], Made Agus Putra Subali[#2], Lailly S. Qolby[#3], Agus Zainal Arifin[#4], M. Ali Fauzi[#5]
[#]*Department of Informatics, Institut Teknologi Sepuluh Nopember*
Surabaya, Indonesia

[1]maryamahfaisol02@gmail.com
[2]madeagusputrasubali@gmail.com
[3]laillys.qolby@gmail.com
[4]agusza@cs.its.ac.id
[5]moch.ali.fauzi@gmail.com

Abstract*— Clustering of news documents manually depends on the ability and accuracy of the human so that it can lead to errors in the grouping process of documents. Therefore, it is necessary to group the news document automatically. In this clustering, we need a weighting method that includes TF.IDF.ICF. In this paper we propose a new weighting algorithm is TF.IDF.ICF.ITF to automatically clustering documents automatically through statistical data patterns so that errors in manual grouping of documents can be reduced and more efficient. K-Means ++ is an algorithm for classification and is the development of the K-Means algorithm in the initial cluster initialization stage which is easy to implement and has more stable results. K-Means ++ classifies documents at the weighting stages of Inverse Class Frequency (ICF). ICF is developed from the use of class-based weighting for the term weighting term in the document. The terms that often appear in many classes will have a small but informative value. The proposed weighting is calculated. Testing is done by using a certain query on some number of best features, the results obtained by TF.IDF.ICF.ITF method gives less optimal results.

***Keywords*— Term Weighting, TF.IDF.ICF.ITF, Indeks Topic, Indeks Class**

Abstrak— Pengelompokan dokumen berita secara manual sangat tergantung pada kemampuan dan ketelitian manusia sehingga dapat menyebabkan terjadinya kesalahan dalam pengelompokan dokumen tersebut. Oleh karena itu, perlu dilakukan pengelompokan dokumen berita secara otomatis. Dalam pengelompokan tersebut diperlukan sebuah metode pembobotan yang meliputi TF.IDF.ICF. Pada *paper* ini kami mengusulkan sebuah algoritma pembobotan yang baru yaitu TF.IDF.ICF.ITF agar dapat mengelompokkan dokumen secara otomatis melalui pola data statistik sehingga kesalahan dalam pengelompokan dokumen secara manual dapat berkurang dan lebih efisien. *K-Means*++ merupakan algoritma untuk klasifikasi dan merupakan pengembangan dari algoritma *K-Means* pada tahap inisialisasi pusat *cluster* awal yang mudah untuk diimplementasikan serta memiliki hasil yang lebih stabil. *K-Means*++ mengelompokan dokumen pada tahap pembobotan kata *Inverse Class Frequency (ICF).* ICF dikembangakan dari penggunaan pembobotan berbasis kelas untuk *term weighting term* pada dokumen. *Term* yang sering muncul pada banyak kelas akan memiliki nilai yang kecil namun informatif. Pembobotan yang diusulkan dihitung $TF \times IDF \times ICF \times ITF$. Pengujian dilakukan dengan menggunakan *query* tertentu pada beberapa jumlah fitur terbaik, hasil yang diperoleh dengan metode TF.IDF.ICF.ITF memberikan hasil yang kurang begitu optimal.

***Kata kunci*— Term Weighting, TF.IDF.ICF.ITF, Indeks Topik, Indeks Kelas**

I. PENDAHULUAN

Kebutuhan akan informasi merupakan suatu hal yang sangat penting. Informasi berupa berita dapat diperoleh tidak hanya dari artikel surat kabar namun juga di dapat dari artikel berita *online*. Popularitas situs berita *online* berbahasa Indonesia saat ini membuat meningkatnya volume berita yang ada. Sehingga perlu dikelompokkan sesuai kategori yang telah di tentukan untuk mempermudah pembaca dalam memilih berita yang ingin dibaca.

Pengelompokkan berita dapat dilakukan dengan dua cara yaitu secara manual dan otomatis. Dalam pengelompokan dokumen berita secara manual sangat tergantung pada kemampuan dan ketelitian manusia sehingga dapat menyebabkan terjadinya kesalahan dalam pengelompokan dokumen tersebut. Oleh karena itu, suatu otomatisasi dalam pengelompokan dokumen berita yang memiliki banyak kemiripan sangat di perlukan agar proses pencarian dokumen menjadi lebih optimal.

Pada penelitan lainnya mengenai pengelompokan dokumen telah dilakukan dalam penelitian [1] dan [2]. Penelitian tersebut berasumsi seluruh kumpulan data tersedia dan statis. Frekuensi dokumen TF.IDF membutuhkan informasi apriori dari data bahwa data tidak berubah selama proses perhitungan bobot. Pembobotan menggunakan TF.ICF tidak membutuhkan informasi *term* frekuensi dari dokumen lain sehingga dokumen dapat di proses dalam waktu *linear*.

Submitted 22-12-2017; Revised 21-02-2018; Accepted 26-02-2018

Dokumen di representasikan sebagai sebuah vektor yang di bentuk dari nilai *term* yang menjadi indeknya [3]. Dari nilai tersebut pembobotan dilakukan dengan menggunakan TF.IDF dimana TFIDF merupakan pembobotan yang sangat sering digunakan [4]. TF (*Term Frequency*) digunakan untuk mengukur jumlah *term* pada sebuah dokumen. Sedangkan IDF (*Inverse Document Frequency)* digunakan untuk mengukur keinformatifan sebuah *term* dalam sebuah dokumen. Metode pembobotan TFIDF dilakukan hanya mempertimbangkan dokumen inter dan intra dokumen saja tidak mempertimbangkan dokumen pada kelas tertentu sehingga paper tersebut mengusulkan metode yang menerapkan metode pembobotan berbasis kelas [5]. Sehingga fungsi dari ICF (*Inverse Class Frequency*) adalah untuk memperhatikan kemunculan *term* pada kumpulan kategori atau kelas.

Untuk mengoptimalkan proses pembobotan kata pada dokumen berita *online*, maka penelitian ini mengusulkan metode TF.IDF.ICF.ITF karena dengan metode tersebut dokumen berita dapat dibobotkan dengan memperhatikan kelas dan topik berita sesuai dengan jumlah *term* yang paling informatif atau yang penting. Metode ICF (*Inverse Class Frequency)* berfungsi untuk memperhatikan kemunculan *term* pada kumpulan dokumen. *Term* yang sangat jarang muncul adalah *term* yang paling penting. Dari *term* tersebut dokumen dapat di kelompokkan ke dalam topik yang sesuai dengan *term* tersebut.

Metode statistik dapat di gunakan untuk pengelompokan dokumen berita. Algoritma yang digunakan dalam hal ini adalah *K-Means++*. Algoritma ini merupakan algoritma yang mudah untuk diimplementasikan serta memiliki kompleksitas waktu yang *linear*.

II. Related Work

Pada penelitian yang dilakukan [6] dengan judul "*K-Means++: The Advantages of Careful Seeding*" dimana tujuan dari penelitian ini adalah untuk mengusulkan sebuah metode baru pada inisialisasi pusat *cluster* awal pada algoritma *K-Means*. Hasil dari penelitian yang dilakukan algoritma *K-Means++* dapat memberikan akurasi hasil clustering yang lebih optimal.

Pada penelitian yang dilakukan [7] memiliki tujuan untuk mengatasi kelemahan pada inisialisasi pusat *cluster* awal pada algoritma *K-Means* dengan menggunakan algoritma *K-Means++* yang dapat secara berurutan memilih setiap pusat *cluster*, sehingga memperoleh hasil *clustering* yang lebih optimal. Hasil pengujian yang dilakukan menunjukkan bahwa kombinasi *MapReduce* dan *K-Means++* jauh memberikan hasil yang lebih efisien dengan pendekatan yang lebih baik.

Pada penelitian [2] mengusulkan metode Term Frequency – Inverse Corpus Frequency (TFICF) yang diterapkan pada fluid dokumen. Metode tersebut digunakan untuk mencari term terbanyak pada inter dokumen dan mengelempokkan dokumen tersebut menjadi beberapa kelas. Pengelompokkan ini dilakukan dengan metode klasifikasi namun pengelompokan yang dilakukan monitoring untuk memastikan data terkelompok dengan benar. Selain itu pada penelitian lain yang dilakukan [8] mengimplementasikan tahapan pembobotan yang sama namun metode klasifikasi yang digunakan adalah naïve bayes algorithm serta mendapatkan hasil akurasi 60%.

Penelitian yang dilakukan berbasis pembobotan kata menerapkan metode TF.IDF.ICSδF. Perbedaan pembobotan ICF saja adalah penelitian ini mempertimbangkan kerapatan anggota yang terdapat pada kelas sehingga anggota dari tiap kelompok harus seimbang dan meningkatkan bobot dokumen dalam kelas. Hasil yang didapatkan lebih bagus dibandingkan dengan metode TFIDFICF dan TFIDF namun terdapat pada beberapa data lain hasil yang didapatkan kurang dari hasil TFIDF [9].

Penelitian yang dilakukan [10] menjelaskan bahwa metode pembobotan TFIDF rentan terhadap bias sehingga pada beberapa kasus menjadi tidak efektif. Untuk memperbaiki masalah tersebut dilakukan learning algorithm dan mendapatkan hasil lebih baik dalam klasifikasi dan pengelompokan data.

III. Metodologi Penelitian

Pada penelitian ini mengusulkan metode pembobotan TF.IDF.ICF.ITF yang dilakukan untuk *clustering* dokumen secara otomatis sesuai jumlah topik yang ingin dikelompokkan. Terdapat beberapa tahapan proses yang dilakukan pada penelitian ini dan dapat dilihat alurnya pada *Gambar 1*.

1) *Preprocessing*

Pada proses *preprocessing* dilakukan beberapa proses yaitu tokenisasi, *stopword*, dan *stemming* dimana *stemming* yang digunakan adalah sastrawi *stemmer*.

2) *Term Weighting*

Pada tahapan *term weighting* ini kami mengajukan beberapa metode pembobotan yaitu:

a) TF (*Term Frequency*): menghitung banyaknya kemunculan *term* pada dokumen.

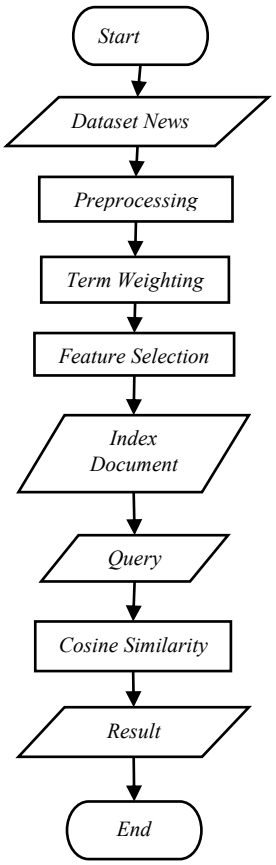
$$\mathbf{TF(d,t) = f(d,t,)} \quad (1)$$

Persamaan 1 merupakan rumus perhitungan metode pembobotan TF dimana *f(d,t)* merupakan frekuensi kemunculan term *t* pada dokumen *d*.

b) IDF (*Inverse Document Frequency*): menghitung banyaknya kemunculan *term* pada dokumen lain.

$$\mathrm{IDF(t)} = 1 + \log(\mathrm{N/df(t)}), \quad (2)$$

Persamaan 2 merupakan rumus perhitungan metode pembobotan IDF dimana N_d merupakan jumlah seluruh dokumen, *df(t)* jumlah dokumen yang mengandung *term t*.



Gambar. 1 Metodologi Penelitian

c) ICF (*Inverse Class Frequency*): merupakan metode yang diusulkan dimana metode ini menghitung kemunculan *term* pada dokumen dan mengelompokkan dokumen berita dengan metode statistika. Pengelompokkan ini dilakukan secara otomatis untuk mengganti pengelompokkan secara manual. Metode pengelompokkan menggunakan salah satu metode *clustering* yaitu *K-Means++*.

$$ICF(t) = 1+\log(Nc / cf(t)) \quad (3)$$

Persamaan 3 merupakan rumus perhitungan metode pembobotan TF dimana N_c merupakan jumlah seluruh kelas, *cf(t)* jumlah kelas yang mengandung *term t*.

d) ITF (*Inverse Topic Frequency*): merupakan metode yang digunakan untuk menghitung kemunculan *term* pada suatu dokumen yang sebelumnya telah terbagi menjadi berbagai topik.

$$IBF(t) = 1+\log(N_b / tf(t)) \quad (4)$$

Persamaan 1 merupakan rumus perhitungan metode pembobotan ITF dimana N_b merupakan jumlah seluruh buku, *tf(t)* jumlah topik yang mengandung *term t*.

e) TF.IDF.ICF.ITF: merupakan kombinasi dari keempat metode tersebut dengan cara dikalikan.

$$TF.IDF.ICF.ITF\ (d,t) = TF(d,t) \times IDF(t) \times ICF(t) \times ITF(t) \quad (5)$$

Persamaan 5 merupakan kombinasi dari rumus perhitungan sebelunya dimana d_i merupakan dokumen ke *i* dan *n* adalah jumlah keseluruhan dokumen. Metode pembobotan ini akan digunakan pada penelitian ini.

3) *Feature Selection*

Feature selection dilakukan dengan melakukan menghitung rata-rata dari nilai *term weighting* sebelumnya dan diurutkan dari nilai terbesar, dimana pada penelitian ini akan diambil 5000, 1000, 500, dan 250 fitur terbaik.

4) *Cosine Similarity*

Cosine similarity digunakan untuk mengetahui seberapa mirip *query* dengan dokumen yang ada. Jika *query* menghasilkan kemiripan dengan suatu kelas maka *query* akan dikelompokkan menjadi kelas tersebut.

$$\cos(q,d_j) = \frac{\sum_{t_k}[TFIDFICFITF(t_k,q)] \bullet [TFIDFICFITF(t_k,d_j)]}{\sqrt{\sum |TFIDFICFITFq|^2} \bullet \sqrt{\sum |TFIDFICFITFd_j|^2}}, \quad (6)$$

Persamaan 6 merupakan perhitungan cosine similarity query input dengan dokumen dimana *cos(q,d$_j$)* merupakan nilai kosinus antara *query* testing dan dokumen training *j*, sedangkan *TF.IDF.ICF.ITF(t_k,q)* dan *TF.IDF.ICF.ITF(t_k,d_j)* adalah hasil pembobotan *TF.IDF.ICF.ITF* kata t_k pada *query* dan dokumen training *j*. | *TF.IDF.ICF.ITF*q| dan | *TF.IDF.ICF.IBFd$_j$*| adalah panjang dari vektor *query* q dan dokumen. Sebagai contoh $||d_i||^2$ = (*TF.IDF.ICF.ITFt*$_1^2$+ *TF.IDF.ICF.ITFt*$_2^2$ +...+*TF.IDF.ICF.ITFt*$_k^2$)$^{1/2}$.*TF.IDF.ICF.ITFt$_k$* adalah bobot kata ke-t_k pada vektor dokumen d_i.

5) *K-Means++*

Algoritma *K-Means++* merupakan perkembangan dari algoritma *K-Means* dimana algoritma *K-Means* banyak digunakan untuk mengelompokkan data karena termasuk algoritma yang mudah dan sederhana dalam di implementasikan. Namun algoritma *K-Means* memiliki kekurangan yaitu waktu eksekusi metode tergolong lama dan memiliki akurasi yang rendah pada beberapa kasus. Dengan adanya kelemahan tersebut algoritma *K-Means* mengalami perkembangan yaitu algoritma *K-Means++* yang dapat mengatasi kekurangan dari algoritma *K-Means*. Perbedaan mendasar algoritma *K-Means++* adalah inisialisasi *centroid* awal dengan kondisi tertentu sedangkan algoritma *K-Means* secara *random*.

Hasil *clustering* yang baik adalah dapat mengelompokkan data ke dalam kelompok sehingga data dalam satu kelompok memiliki kemiripan yang maksimum dan data antar kelompok memiliki kemiripan yang minimum. Secara intuitif, merupakan pilihan bijak untuk

memilih pusat *cluster* awal yang jauh dari satu sama lainnya. Algoritma *K-Means++* mengikuti ide ini, namum titik terjauh tidak selalu dipilih untuk dijadikan pusat *cluster*. *K-Means++* sangat sederhana dan mudah untuk diimplementasikan. Sebenarnya, kecuali pada pusat *cluster* awal yang dipilih secara *random* dari *data point*, setiap pusat *cluster* berikutnya dipilih berdasarkan perhitungan jarak terdekat antara *data point* dengan pusat *cluster* yang telah terpilih sebelumnya. Berikan $D(x)$ menjdai *euclidean distance* antara x dan pusat *cluster* terdakat yang telah dipilih. Detail metode *k-means++* digambarkan pada *Algoritma 1*.

Algoritma 1: K-Means++

Input: k, jumlah *cluster*
$X = \{x_1, x_2, x_3, \dots, x_n\}$, Dataset
Output: $C = \{c_1, c_2, c_3, \dots, c_k\}$

1 $C \leftarrow \emptyset$
2 Pilih pusat *cluster* awal x dari *datasets* X secara *random*, $C = C \cup \{x\}$
3 **Repeat**
4 Pilih $x \in X$ dengan probabilitas $D(x)^2 / \sum_{x \in X} D(x)^2$
5 $C = C \cup \{x\}$
6 **Until** pusat *cluster* k terpilih sepenuhnya

Pada perhitungan ini akan membandingkan hasil pengelompokan dokumen yang dilakukan otomatis dengan pengelompokan menggunakan hasil pengelompokan berita secara manual. Setelah melakukan perhitungan bobot secara terpisah hasil akan digabungkan dengan mengkalikan semua hasil dan akan menjadi bobot dinal dari dokumen. TF.IDF.ICF.ITF merupakan kombinasi dari keempat metode tersebut dengan cara mengalikan satu sama lain.

IV. HASIL & PEMBAHASAN

Data yang digunakan dalam uji coba ini menggunakan artikel berita *online* berbahasa Indonesia. Jumlah artikel yang dikumpulkan sebesar 11.245 yang terbagi menjadi 13 topik berita, tetapi data yang digunakan hanya 286 artikel berita yang dipilih secara acak dimana terdapat 22 artikel disetiap topik yang digunakan. Hasil pengelompokkan dokumen menggunakan metode *K-Means++* memberikan hasil *rand index* sebesar 51% dan untuk *homogeneity* sebesar 70%. Hasil jumlah data setiap *cluster* yang berhasil dikelompokkan terlihat pada *Tabel 1*.

Tabel 1 merupakan tabel yang berisi jumlah data yang digunakan pada ekperiment penelitian ini. Data tersebut memiliki total 286 dari semua artikel. Pengujian dilakukan menggunakan *query* yang memberikan hasil pencarian relevan lebih dari satu artikel. Pengujian dilakukan menggunakan beberapa variasi *feature selection*, yaitu 5000, 1000, 500, 250 fitur terbaik. Dengan membandingkan antara kemiripan *query* yang diberikan oleh pengguna dan hasil *feature selection* di setiap metode pembobotan kata. Untuk 5000 fitur terbaik TF.IDF digambarkan pada *Gambar 2*, sedangkan untuk TF.IDF.ICF digambarkan pada *Gambar 3*, dan TF.IDF.ICF.ITF digambarkan pada *Gambar 4*. Hasil yang diperoleh akan diukur *precision*, *recall*, dan *f-measure*. Hasil uji coba metode TF.IDF.ICF.ITF akan dibandingkan dengan metode TF.IDF dan TF.IDF.ICF.

TABEL 1
JUMLAH DATA SETIAP CLUSTER

Cluster	***Total***
0	22
1	27
2	27
3	29
4	15
5	21
6	25
7	18
8	31
9	15
10	22
11	22
12	12

Hasil pengujian dengan 5000 fitur terbaik dapat dilihat pada *Tabel 2*. 1000 fitur terbaik dapat dilihat pada *Tabel 3*. Sedangkan untuk 500 fitur terbaik terlihat pada *Tabel 4* dan untuk 250 fitur terbaik pada *Tabel 5*. Dari ketiga hasil tersebut metode TF.IDF.ICF.ITF memiliki *precision*, *recall*, dan *f-measure* yang lebih rendah pada semua variasi fitur. Hasil perbandingan nilai *f-measure* keseluruhan metode pembobotan digambarkan pada *Gambar 5*.

TABEL 2.
PENGUJIAN PADA 5000 FITUR

Metode	**Hasil Pengujian**		
	Precision	*Recall*	*F Measure*
TF.IDF	40%	100%	57%
TF.IDF.ICF	38%	83%	53%
TF.IDF.ICF.ITF	33%	66%	44%

Tabel 2 merupakan hasil pengujian yang dilakukan dengan mengambil 5000 fitur terbaik. Fitur tersebut diuji dengan perbandingan tiga metode yaitu TF.IDF, TF.IDF.ICF, TF.IDF.ICF.ITF. Pada Gambar 2 - 4 menunjukkan 10 fitur terbaik yang terdapat pada masing-masing metode. 10 fitur yang digunakan pada masing-masing metode memiliki fitur yang tidak sama dengan sehingga hasil pengujian akan memiliki perbedaan. Ketiga metode tersebut memiliki hasil recall yang lebih tinggi dibandingkan dengan precision dan F Measure. Namun pada ketiga metode tersebut metode TF.IDF memiliki nilai yang lebih tinggi dibandingkan dengan dua metode yang lain.

#	Term	TF.IDF
1	gram	41.7582225632
2	kuartal	38.8319011534
3	sapi	37.6056507898
4	iran	36.8454904969
5	cabai	35.2318147942
6	vietnam	33.3775087524
7	astra	32.330040562
8	fitch	31.9327584307
9	persib	31.6679164546
10	prabowo	31.6331285183

Gambar 2. 5000 Fitur Terbaik TF.IDF

#	Term	TF.IDF.ICF
1	gram	46.5162944284
2	iran	41.0437892015
3	cabai	39.2462458797
4	fitch	35.5712839747
5	gugus	33.6129115129
6	gugus	33.6129115129
7	makam	33.6129115129
8	ecu	32.8350313612
9	sapi	30.5701358123
10	vanili	30.0987787478

Gambar 3. 5000 Fitur Terbaik TF.IDF.ICF

#	Term	TF.IDF.ICF.ITF
1	gram	51.8165169525
2	iran	45.7204561345
3	cabai	43.7180947007
4	fitch	39.6243953166
5	gugus	37.4428793315
6	gugus	37.4428793315
7	ecu	36.5763649076
8	nguyen	33.5283344987
9	vanili	33.5283344987
10	cabai	32.2133329374

Gambar 4. 5000 Fitur Terbaik TF.IDF.ICF.ITF

TABEL 3.
PENGUJIAN PADA 1000 FITUR

Metode	**Hasil Pengujian**		
	Precision	*Recall*	*F Measure*
TF.IDF	40%	100%	57%
TF.IDF.ICF	40%	100%	57%
TF.IDF.ICF.ITF	25%	50%	33%

Table 3 merupakan hasil pengujian dengan mengambil 1000 fitur terbaik dari hasil yang diuji menggunakan tiga metode. Hasil recall ketiga metode lebih tinggi dibandingkan dengan yang lain. Dibandingkan dengan pengujian 5000 fitur terbaik metode TF.IDF stabil dengan ketiga hasil evaluasi yang sama dan metode TF.IDF.ICF memiliki peningkatan hasil. Namun metode yang diusulkan yaitu TF.IDF.ICF.ITF mengalami penurunan hasil pada ketiganya.

TABEL 4
PENGUJIAN PADA 500 FITUR

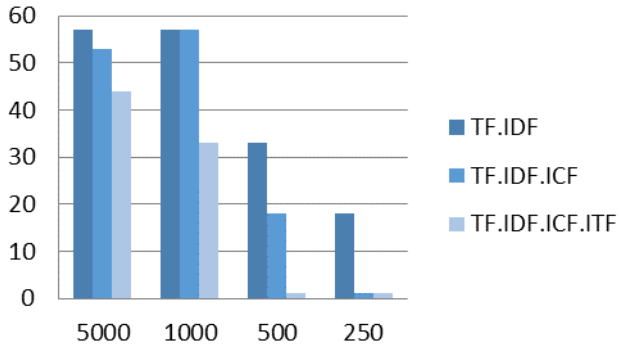
Metode	**Hasil Pengujian**		
	Precision	*Recall*	*F Measure*
TF.IDF	25%	50%	33%
TF.IDF.ICF	14%	25%	18%
TF.IDF.ICF.ITF	0%	0%	0%

Tabel 4 merupakan hasil pengujian dengan menggunakan 500 fitur terbaik. Hasil yang didapatkan semakin menurun pada ketiga metode. Bahkan pada metode TF.IDF.ICF.ITF hasil yang didapatkan sangat tidak sesuai dengan query yang dimasukkan.

TABEL 5
PENGUJIAN PADA 250 FITUR

Metode	**Hasil Pengujian**		
	Precision	*Recall*	*F Measure*
TF.IDF	14%	25%	18%
TF.IDF.ICF	0%	0%	0%
TF.IDF.ICF.ITF	0%	0%	0%

Tabel 5 merupakan hasil pengujian dengan 250 fitur terbaik. Hasil yang didapatkan semakin menurun dan paling rendah dibandingkan dengan jumla fitur yang lain. Menurunnya hasil dari masing-masing metode disebabkan karena semakin sedikit fitur yang diambil maka informasi penting yang ada semakin sedikit sehingga pada pengujian dengan query yang diinputkan akan menurunkan hasil precision, recall dan F Measure.



Gambar 5. Perbandingan Nilai *F-Measure*

Gambar 5 menunjukkan bahwa hasil pembobotan TFIDF lebih tinggi dibandingkan dengan metode pembobotan lain. Menurut penelitian [8] propose method term weighting TF.IDF.ICSδF mendapatkan hasil yang bagus pada namun hasil hanya bagus beberapa dokumen sedangkan dengan menggunakan metode TFIDF hasil yang didapatkan pada semua dokumen relative stabil.

Analisa kami mengenai kurang optimalnya metode yang diusulkan, dikarenakan hasil akurasi *rand index* dan *homogeneity* pada tahap ICF atau saat proses *clustering* memberikan akurasi yang kurang optimal. Kedua, data yang digunakan terdiri dari tiga belas topik dengan kemiripan yang cukup tinggi disetiap artikel. Dalam melakukan pembobotan kata pada dokumen memperhatikan topik berita itu sendiri (ITF) dan hasil proses clustering menggunakan algoritma Kmeans++ pada pembobotan ICF. Namun ketika antar dokumen memiliki tingkat kemiripan yang rendah maka akan dikenali dengan tepat dan hasil akurasi akan meningkat. Sehingga pada penelitian selanjutnya diharapkan dapat mempertimbangkan kemiripan term antar dokumen. Pembagian topik yang lebih umum dapat meningkatkan hasil karena term dalam dokumen akan berbeda atau memiliki kemiripan yang rendah.

V. KESIMPULAN

Penelitian ini mengusulkan metode TF.IDF.ICF.ITF karena dengan tujuan metode yang diusulkan dapat memperhatikan topic dan kelas pada dokumen berita. Pengujian dilakukan menggunakan *query* yang memberikan hasil pencarian relevan lebih dari satu artikel. Hasil yang diberikan metode TF.IDF.ICF.ITF memberikan hasil yang kurang optimal. Analisa kami mengenai kurang optimalnya metode yang diusulkan, dikarenakan hasil akurasi pada proses *clustering* memberikan hasil yang kurang optimal dan data yang digunakan terdiri dari tiga belas topik dengan kemiripan yang cukup tinggi disetiap artikel. Pada penelitian selanjutnya akan dilakukan uji coba dengan data yang lebih sesuai dengan pembobotan TF.IDF.ICF.ITF. Dengan memperhatikan hasil ICF atau *proses clustering* dengan akurasi yang baik.

ACKNOWLEDGMENT

Kami mengucapkan terima kasih kepada seluruh peneliti pada bidang *text mining* yang memberikan referensi dan memotifasi dalam penulisan *paper* ini.

REFERENSI

[1] A. Z. Arifin and A. N. Novan, "Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering," *Pros. Semin. Intell. Technol. its Appl. (SITIA), Tek. Elektro, Inst. Teknol. Sepuluh Nop. Surabaya*, 2002.

[2] J. W. Reed, J. Yu, T. E. Potok, B. A. Klump, M. T. Elmore, and A. R. Hurson, "TF-ICF: A new term weighting scheme for clustering dynamic data streams," *Proc. - 5th Int. Conf. Mach. Learn. Appl. ICMLA 2006*, no. May 2014, pp. 258 - 263, 2006.

[3] M. A. Fauzi *et al.*, "Term Weighting Berbasis Indeks Buku Dan Kelas Untuk Perangkingan Dokumen Berbahasa Arab," *Lontar Komput.*, vol. 5, no. 2, pp. 110 - 117, 2015.

[4] Salton, G. 1989. Automatic Text Processing: The Transformation, Analysis, and Retrieval of. Reading: Addison-Wesley

[5] Ren, Fuji, and Mohammad Golam Sohrab. 2013. "Class-indexing-based term weighting for automatic text classification." Information Sciences 109-125

[6] D. Arthur and S. Vassilvitskii, "K-Means++: The Advantages of Careful Seeding," *Proceedings of the 18th Annu. ACM - SIAM Symposium on Discrete Algorithms*, pp. 1027 - 1035, 2007.

[7] Y. Xu, W. Qu, Z. Li, G. Min, K. Li and Z. Liu, "Efficient K-Means++ Approximation with MapReduce," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no.12, pp. 3135 - 3144, 2014.

[8] Shathi S.P, Hossain Md.Delowar, Nadim MD, Riayadh Sayed G.R, Sultana Tangina, "Enhancing Performance of Naïve Bayes in Text Classification by Introduction an Extra Weight using less Number of Training Examples", no. December, pp. 12–13, 2016.

[9] Kurniawati and A. Syauqi, "Term weighting based class indexes using space density for Al-Qur'an relevant meaning ranking," *2016 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2016*, pp. 460–463, 2017.

[10] Chen. C.H., "Improved TF.IDF in Big News Retrieval: An Empirical Study," *Pattern Recognition Letters,* vol. 93, pp. 113 - 122, 2017.