p-ISSN : 2443-2210
e-ISSN : 2443-2229

Jurnal Teknik Informatika dan Sistem Informasi
Volume 6 Nomor 3 Desember 2020

Pengelompokan Komentar *Dataset* Sentipol dengan *Modified K-Means Clustering*

http://dx.doi.org/10.28932/jutisi.v6i3.3006

Ruddy Cahyanto ✉ #1, Antonius Rachmat Chrismanto*2, Danny Sebastian#3

#Prodi Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana
Jl. Dr. Wahidin Sudirohusodo no. 5-25, Yogyakarta

1ruddy.c@ti.ukdw.ac.id

2anton@ti.ukdw.ac.id

3danny.sebastian@staff.ukdw.ac.id

***Abstract* — Clustering is a technique in data mining that groups data sets into similar data clusters. One of the algorithms that is commonly used for clustering is K-Means. However, the K-Means algorithm has several weaknesses, one of them is the random factor in initial centroid selection, so that cluster result is inconsistent even though it is tested with the exact same data. The Modified K-Means algorithm focuses on selecting the initial centroid to overcome inconsistencies of cluster results in the K-Means method. The test was conducted using Sentipol dataset and only focused on comment data. Furthermore, the specified number of clusters is three based on the number of existing comment labels (positive, negative, and neutral). According to testing results, Modified K-Means algorithm produces better purity value than K-Means algorithm. Modified K-Means algorithm produces average of purity value 0,42, while K-Means produces average of purity value 0,391. Meanwhile, from testing related to random factors conducted 5 times with the same attributes and test data, the results of the cluster on the Modified K-Means algorithm did not change, so automatically the resulting purity value was also the same. Whereas in the K-Means algorithm, the cluster results always change in each test, so the result of purity value is also likely to change.**

***Keywords*— clustering; Sentipol *Dataset*; *K-Means*; *Modified K-Means*;**

I. PENDAHULUAN

A. Latar Belakang

Dataset Sentipol merupakan *dataset* sentimen komentar pada kampanye pemilu presiden tahun 2014 yang diambil dari *facebook page* [1]. *Dataset* Sentipol berisi 3400 komentar dari 68 status. Selain itu, komentar-komentar yang ada juga telah diberi label positif, negatif dan netral. Sebelumnya telah dilakukan penelitian mengenai klasifikasi pada *dataset* sentipol, namun belum pernah dilakukan penelitian mengenai *clustering*.

Clustering merupakan salah satu teknik dalam *data mining* yang mengelompokkan suatu himpunan data ke dalam kelompok-kelompok (*clusters*) data yang serupa [2]. Metode yang umum digunakan dalam *clustering* adalah *K-Means*. *K-Means* sering digunakan dalam *clustering* karena kemudahan penggunaannya. Namun, kualitas hasil akhir *clustering* dengan menggunakan algoritma *K-Means* sangat bergantung pada pemilihan *centroid* (titik pusat *cluster*) awal [3]. Pada algoritma *K-Means*, pada umumnya pemilihan *centroid* awal dilakukan secara *random*, sehingga memungkinkan hasil akhir *clustering* berbeda meskipun menggunakan data uji yang sama [4]. Selain itu, jika *centroid* awal yang dipilih buruk, maka *cluster* yang dihasilkan bisa sangat tidak optimal. Berdasarkan kelemahan *K-Means* tersebut, pada penelitian ini akan dilakukan modifikasi pemilihan *centroid* awal dengan menggunakan algoritma *Modified K-Means* [5].

Tujuan utama dari penelitian ini yaitu membangun sistem *clustering* dengan menggunakan *dataset* sentipol dan algoritma *Modified K-Means*, setelah itu dilakukan perbandingan dengan algoritma *K-Means* biasa. Perbandingan dilakukan pada akurasi dan konsistensi dari *cluster* yang dihasilkan. Sedangkan alat ukur pengujian yang akan digunakan adalah nilai *purity* dari *cluster* yang dihasilkan. Nilai *purity* sendiri dapat dihitung dengan cara menjumlahkan dokumen (komentar) yang berada di *cluster* yang tepat lalu dibagi dengan jumlah keseluruhan komentar [6].

Penelitian ini memberikan kontribusi dengan memberikan beberapa manfaat baik dari sisi algoritma maupun *dataset* yang digunakan. Penggunaan algoritma *Modified K-Means* ternyata memang dapat mengatasi inkonsistensi hasil *cluster* pada metode *K-Means* biasa dan menghasilkan nilai *purity* yang lebih baik.

B. Rumusan Masalah

Berdasarkan pada latar belakang masalah yang telah diuraikan, permasalahan tersebut dapat dirumuskan ke dalam beberapa poin, yang pertama yaitu bagaimana penerapan *Modified K-Means* dalam mengklasterisasi data

komentar pada *dataset* sentipol untuk mendapatkan hasil *cluster* yang konsisten? Kemudian yang kedua, bagaimana perbandingan hasil *clustering* pada *Modified K-Means* dan metode *K-Means*? Dan yang terakhir yaitu bagaimana hasil analisis *clustering* data komentar pada *dataset* sentipol berdasarkan nilai *purity* yang dihasilkan?

C. Tujuan Penelitian

Berdasarkan pada rumusan masalah yang ada, terdapat tiga tujuan yang ingin dicapai dari penelitian ini. Tujuan pertama yaitu menerapkan *Modified K-Means* dalam mengklasterisasi data komentar sentipol untuk mendapatkan hasil *cluster* yang konsisten. Kedua, membandingkan hasil *clustering* pada metode *Modified K-Means* dan *K-Means*. Kemudian yang terakhir yaitu menganalisis hasil *clustering* data komentar sentipol berdasarkan nilai *purity* yang dihasilkan.

D. Tinjauan Pustaka

Hingga saat ini, sudah ada beberapa penelitian yang dilakukan mengenai algoritma *K-Means* dan *Modified K-Means*. Penelitian yang pertama yaitu mengenai perbandingan metode *K-Means* dan *Hierarchical Agglomerative Clustering* pada pengelompokan dokumen teks. Fokus penelitian ini yaitu membandingkan tingkat akurasi hasil *clustering* dari kedua metode dengan melakukan jumlah data uji yang bervariasi. Dari penelitian ini, metode *K-Means* dapat menghasilkan akurasi yang lebih baik pada jumlah data uji yang besar [7].

Penelitian selanjutnya lebih berfokus pada efisiensi dari proses *clustering* yang dilakukan. Dengan menggunakan metode *K-Means* yang dimodifikasi pada pemilihan *centroid* awalnya, didapatkan waktu pemrosesan yang lebih efisien dari metode *K-Means* biasa. Selain itu, iterasi yang diperlukan pada proses *clustering* juga semakin sedikit [5].

Selain kedua penelitian tersebut, masih terdapat penelitian lain mengenai perbandingan metode *K-Means* dan *Modified K-Means* pada kumpulan data yang besar. Dari hasil pengujian yang dilakukan dengan jumlah data yang bervariasi mulai dari 300, 400, 500, hingga 600 data, algoritma *Modified K-Means* lebih efisien dari segi waktu pemrosesan yang dibutuhkan jika dibandingkan dengan algoritma *K-Means* biasa [8].

Dari beberapa penelitian di atas, algoritma *K-Means* maupun *Modified K-Means* dapat digunakan pada jumlah data uji yang besar. Selain itu, algoritma *Modified K-Means* juga memerlukan waktu pemrosesan yang lebih sedikit dibandingkan *K-Means* biasa. Namun, penelitian ini tidak akan berfokus pada efisiensi dari proses *clustering*, melainkan akan lebih berfokus pada salah satu kelemahan dari metode *K-Means* yaitu konsistensi dan akurasi dari *cluster* yang dihasilkan. Selain itu, salah satu pembeda antara penelitian ini dengan penelitian-penelitian sebelumnya terletak pada *dataset* yang digunakan, di mana sebelumnya belum pernah dilakukan *clustering* pada *dataset* sentipol.

II. METODE PENELITIAN

A. Prosedur Penelitian

Tahapan-tahapan yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Melakukan perumusan serta batasan-batasan masalah terkait penelitian yang dilakukan.
2. Melakukan pemahaman tentang penelitian terkait *clustering* yang menggunakan algoritma *Modified K-Means.*
3. Melakukan perancangan sistem *clustering* yang akan dibangun, meliputi perancangan alur kerja sistem, perancangan antarmuka hingga perancangan *database*.
4. Melakukan perancangan pengujian yang akan dilakukan.
5. Melakukan pengujian konsistensi dan nilai *purity* dari *cluster* yang dihasilkan.
6. Pengambilan kesimpulan berdasarkan hasil pengujian

B. Perancangan Alur Kerja Sistem

Tahapan-tahapan pada sistem *clustering* yang dibangun akan dijelaskan dalam beberapa poin berikut:

1) Input Data: Tahap pertama dalam sistem *clustering* adalah menginputkan data yang akan dilakukan *clustering*. Dalam penelitian ini, data yang digunakan adalah *file dataset* sentipol yang berekstensi *.csv* dan hanya berfokus pada data komentar saja. Setelah *file dataset* dimasukkan ke dalam sistem, sistem akan mengambil komentar dan label komentar untuk kemudian disimpan ke dalam *database*.

2) Normalisasi: Proses selanjutnya adalah normalisasi teks komentar. Normalisasi dilakukan untuk menghilangkan objek-objek yang tidak dibutuhkan seperti alamat *web* (*url*), tagar, *mention user* lain, dan sebagainya. Selain menghapus objek-objek yang tidak dibutuhkan, pada proses normalisasi ini juga dilakukan proses pengubahan kata-kata singkatan serta kata "alay" menjadi kata yang utuh dan baku.

Beberapa normalisasi yang akan dilakukan adalah sebagai berikut [9]:

- *Case folding* atau mengubah huruf kapital ke huruf non-kapital.
- Menghapus *url* atau alamat web.
- Menghapus tagar atau *hashtag*, contohnya #salam2jari, #salam1jari, dll.
- Menghapus *mention*, sebagai contoh @ruddy, @jono, dll.
- Menghapus *emoticon*, contohnya :v, :p, :poop:, dll.
- Menghapus tanda baca.
- Menghapus angka.
- Mengubah kata dengan huruf-huruf yang berulang, misalnya "aduuuuhhh" menjadi "aduh".
- Mengubah singkatan dan kata-kata alay menjadi kata baku, misalnya "mantapz" menjadi "mantap", "yg" menjadi "yang".

Pada proses pengubahan singkatan dan kata-kata alay menjadi kata yang baku, digunakan Kamus Besar Bahasa

Alay (KBBA) yang dibangun sendiri berdasarkan temuan kata-kata alay yang terdapat pada *dataset* sentipol. Untuk memperjelas proses normalisasi yang dilakukan, berikut ini contoh beberapa komentar sebelum dan setelah dilakukan normalisasi seperti yang bisa dilihat pada Tabel I.

TABEL I
CONTOH KOMENTAR SEBELUM DAN SESUDAH NORMALISASI

Sebelum Normalisasi	Sesudah Normalisasi
Enak, di lhat tapa beban	enak di lihat tanpa beban
Tapi ingat 98 gk pak?	tapi ingat nggak pak
yakiiinn..menaaanng???	yakin menang
:*	
http://www.youtube.com/watch?v=HAXObaH8x	

3) Tokenisasi: Proses selanjutnya adalah tokenisasi atau pemenggalan kalimat menjadi kata per kata. Pada proses ini data komentar yang ada di dalam *database* akan dipecah menjadi kata per kata berdasarkan *delimiter* seperti tanda baca (titik, koma, dll), *whitespace* (spasi, *tab*, *new line*), dan lain sebagainya [10].

4) Stopword Removal: Setelah dilakukan tokenisasi dan didapatkan kata per kata dari tiap dokumen, proses selanjutnya adalah *stopword removal* atau proses penghapusan kata-kata yang tidak penting atau kurang memiliki makna. Proses ini sendiri dilakukan dengan cara menghapus kata-kata yang sama dengan kata-kata yang terdapat di *stopword list* [11]. *Stopword list* sendiri berisi kata-kata yang tidak memiliki arti penting dan tidak diperlukan untuk proses selanjutnya dalam *clustering*. Untuk lebih jelasnya, contoh sebagian kata-kata yang termasuk dalam *stopword list* dapat dilihat pada Tabel II. Data *stopword list* ini dibangun sendiri berdasarkan kata-kata umum dari KBBI dan dari *dataset* sumber.

TABEL II
CONTOH KATA DALAM DAFTAR *STOPWORD* BAHASA INDONESIA

Stopword List			
Ada	Bagai	Cara	Dari
Adalah	Bagaikan	Caranya	Daripada
Adanya	Bagaimana	Cukup	Datang
Adapun	Bagaimanakah	Cukupkah	Dekat
Agak	Bagaimanapun	Cukuplah	Demi
Agaknya	Bagi	Cuma	Demikian
Agar	Bagian	Dahulu	Demikianlah
Akan	Bahkan	Dalam	Dengan

5) Feature Generation: Setelah proses tokenisasi dan *stopword removal*, maka telah didapatkan kata-kata yang memiliki makna dan arti penting untuk kemudian dilakukan pengklasteran atau pengelompokkan. Namun, kata-kata yang ada masih belum bisa dilakukan penghitungan jarak untuk kemudian dikelompokkan ke dalam *cluster* yang sesuai. Oleh karena itu, pada tahap ini dilakukan pembobotan pada setiap kata sehingga bisa dilakukan penghitungan jarak berdasarkan bobot masing-masing kata. Pemberian bobot pada tiap kata (*features*) juga dibutuhkan dalam proses *feature selection*, dimana kata-kata yang mewakili keseluruhan kata pada proses *clustering* dipilih berdasarkan bobot tertinggi sesuai dengan persentase *feature selection* yang ditentukan. Selain pada proses *feature selection*, bobot ini juga digunakan pada proses pengelompokan *cluster* untuk menghitung jarak terdekat antar dokumen/data.

Metode yang digunakan untuk melakukan pembobotan pada penelitian ini adalah TF-IDF (*Term Frequency – Inverse Document Frequency*). Metode ini akan digunakan untuk mendapatkan bobot setiap kata pada dokumen. Persamaan 1 dan 2 merupakan rumus dari penghitungan TF-IDF [12].

$$TF - IDF(w,d) = TF(w,d) * IDF(w) \quad (1)$$

$$IDF(w) = log(|D|/DF(w)) \quad (2)$$

Dimana:
$TF(w,d)$ = jumlah kata yang muncul pada dokumen d
$IDF(w)$ = frekuensi invers dokumen kata w dari jumlah dokumen yang ada
$DF(w)$ = jumlah dokumen yang memuat kata w
D = jumlah seluruh dokumen

6) Features Selection: Setelah pemberian bobot dengan metode TF-IDF dan menggabungkan kata-kata yang sama, proses berikutnya adalah *features selection* atau pemilihan kata-kata yang nantinya akan mewakili keseluruhan kata dalam proses *clustering*. Kata-kata atau *features* yang telah diberi bobot pada proses *features generation* akan dipilih beberapa saja berdasarkan nilai persentase *features selection* yang ditentukan oleh *user*. Pengambilan *features selection* dilakukan agar dimensi vektor kata menjadi lebih kecil. Kata-kata dengan bobot TF-IDF diurutkan secara *descending* dan kemudian diambil sesuai *threshold* tertentu. Persamaan 3 dan 4 menunjukan cara *features selection*. Kata-kata inilah yang nantinya akan digunakan pada proses *clustering*.

$$List\ of\ FS = argmax(desc\_sort(TFIDF(wi))) \quad (3)$$

$$List\ of\ Final\ FS = List\ of\ FS \times \frac{threshold}{100} \quad (4)$$

Tidak ada aturan khusus dalam penentuan *threshold*, *user* bisa menentukan *threshold* dalam skala 0 - 100%. Sebagai contoh, misalnya telah didapatkan tabel *features* yang diurutkan secara *descending* berdasarkan nilai *tf-idf* seperti terlihat pada Tabel III dan *user* menentukan *threshold* sebesar 30%. Maka penghitungannya adalah sebagai berikut:

$$List\ of\ Final\ FS = 10 \times \frac{30}{100}$$
$$List\ of\ Final\ FS = 3$$

Dari perhitungan tersebut, didapatkan jumlah *features* yang terpilih yaitu sebesar 3. Sehingga, mengacu pada Tabel III, maka kata-kata yang dimasukkan ke dalam vektor adalah 3 kata teratas yaitu "Prabowo", "Jokowi", dan "Dua". Sedangkan tujuh kata lainnya tidak dimasukkan ke dalam vektor, sehingga tidak digunakan dalam proses *clustering*.

TABEL III
CONTOH TABEL *FEATURES*

No	*Features*	*TF-IDF*
1	Prabowo	0,0370179
2	Jokowi	0,0321342
3	Dua	0,0272157
4	Indonesia	0,0261607
5	Presiden	0,0235883
6	Satu	0,0227553
7	Pilih	0,0220397
8	Harga	0,0217826
9	Rakyat	0,0197233
10	Tertawa	0,0195362

7) Clustering: Tahap berikutnya adalah proses *clustering*. Proses *clustering* pada tahap ini akan menerapkan dua algoritma yaitu *K-Means* dan *Modified K-Means* dengan tujuan untuk membandingkan dari segi *purity* dan konsistensi dari *cluster* yang dihasilkan.

Proses atau tahapan dari algoritma *K-Means* itu sendiri adalah sebagai berikut [13]:

1. Inisialisasi awal:
   a. Input *dataset* $D = \{d_1, d_2, d_3, \dots, d_n\}$
   b. Set jumlah *cluster* (k).
   c. Ulangi sebanyak jumlah *cluster*:
      - Pilih dokumen (datapoint) pada himpunan D secara random sebagai *centroid* awal.
2. Ulangi hingga nilai *centroid* tidak terjadi perubahan:
   a. Hitung jarak tiap dokumen dengan masing-masing *centroid*.
   b. Masukkan dokumen ke dalam *cluster* yang memiliki jarak *centroid* (titik pusat *cluster*) terdekat.
   c. Hitung ulang *centroid* baru pada setiap *cluster*.
3. Hasil akhir *cluster* telah didapat.

Sementara itu, tahapan dari algoritma *Modified K-Means* adalah sebagai berikut [5]:

1. Inisialisasi awal:
   a. Input *dataset* $D = \{d_1, d_2, d_3, \dots, d_n\}$
   b. Set jumlah *cluster* (k).
   c. Ulangi sebanyak jumlah *cluster*:
      - Hitung semua jarak antar dokumen satu dengan dokumen lainnya pada himpunan D.
      - Temukan pasangan dokumen pada himpunan D yang memiliki jarak terdekat.
      - Masukkan pasangan dokumen tersebut ke dalam himpunan C lalu hapus dari himpunan D.
      - Ulangi hingga jumlah anggota himpunan C mencapai n/k:
        - Temukan dokumen di himpunan D yang memiliki jarak terdekat dengan himpunan C.
        - Masukkan dokumen ke dalam himpunan C lalu hapus dari himpunan D.
      - Hitung rata-rata vektor dari tiap dokumen di himpunan C dan tentukan sebagai *centroid* awal.
2. Ulangi hingga nilai *centroid* tidak terjadi perubahan:
   a. Hitung jarak tiap dokumen dengan masing-masing *centroid*.
   b. Masukkan dokumen ke dalam *cluster* yang memiliki jarak *centroid* (titik pusat *cluster*) terdekat.
   c. Hitung ulang *centroid* baru pada setiap *cluster*.
3. Hasil akhir *cluster* telah didapat.

Untuk menentukan jarak antar dokumen digunakan rumus *Euclidean Distance* berikut [13] seperti pada Persamaan 5:

$$d(i,j) = \sqrt{\left(|x_{i1} - x_{j1}|^2 + \cdots + |x_{in} - x_{jn}|^2\right)} \quad (5)$$

Dimana:

$d(i,j)$ = jarak dokumen ke - i ke dokumen ke - j
x_{in} = nilai bobot kata ke - n di dokumen ke - i
x_{jn} = nilai bobot kata ke - n di dokumen ke - j

Sedangkan untuk menghitung nilai *centroid* baru pada tiap *cluster*, digunakan rumus pada Persamaan 6 [14]:

$$c_i = \left(\frac{1}{m_i}\right) \sum_{x \in C_i} x \quad (6)$$

Dimana:

c_i = nilai *centroid* dari *cluster* C_i
m_i = jumlah dokumen yang berada dalam *cluster* ke - i
x = nilai vektor dari sampel dokumen ke - i

C. Rancangan Pengujian

Dataset sebanyak 3400 komentar akan diambil beberapa saja untuk dijadikan sampel. Pengambilan sampel dilakukan secara seimbang antara label satu dan label lainnya. Proses pengambilan data dilakukan dengan cara mencari label yang memiliki jumlah komentar paling sedikit, lalu jumlah tersebut akan dijadikan sebagai patokan banyaknya jumlah komentar yang akan dipilih pada label yang lainnya. Misalnya label yang memiliki jumlah komentar paling sedikit adalah label netral dengan jumlah komentar sebanyak 100, maka label positif dan negatif masing-masing akan diambil 100 komentar sebagai data sampel. Proses pengambilan komentar dilakukan secara *random*.

Kemudian pengujian dilakukan menggunakan jumlah persentase *features selection* yang meningkat setiap tahapnya yaitu mulai dari 10%, 20%, 30% sampai dengan 100%. Selain itu, dalam pengujian ini juga akan membandingkan dua algoritma *clustering* yang digunakan yaitu *K-Means* dan *Modified K-Means*.

Alat ukur pengujian yang akan digunakan adalah nilai *purity* dari *cluster* yang dihasilkan. *Purity* merupakan suatu ukuran evaluasi yang sederhana dan transparan. *Purity* dihitung dengan cara menjumlahkan dokumen yang berada di *cluster* yang tepat lalu dibagi dengan *N* (jumlah keseluruhan dokumen) [6].

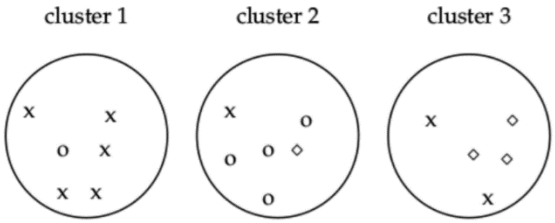
Purity bisa dihitung dengan menggunakan rumus pada Persamaan 7 [15]:

$$Purity(P, \mathrm{C}) = \frac{1}{N}\sum_{k} \max_{j} \left|p_k \cap c_j\right| \quad (7)$$

Dimana:

P = himpunan *cluster* $\{p_1, p_2, p_3, \dots, p_k\}$
C = himpunan kelas (*cluster*) acuan $\{C_1, C_2, C_3, \dots, C_j\}$
p_k = himpunan data/dokumen pada *cluster* p_k
c_j = himpunan data/dokumen pada kelas acuan c_j

Sebagai contoh, misalnya diperoleh hasil *clustering* seperti pada Gambar 1 berikut [15]:



Gambar 1. Contoh hasil *cluster*

Pada Gambar 1, dapat dilihat bahwa mayoritas kelas dan jumlah anggota dari mayoritas kelas pada ketiga *cluster* tersebut adalah x = 5 (*cluster* 1), o = 4 (*cluster* 2) dan ⋄ = 3 (*cluster* 3). Sedangkan total anggota pada keseluruhan *cluster* sama dengan N = 17. Sehingga, penghitungan nilai *purity*-nya adalah sebagai berikut:

$$Purity = \frac{1}{17} \times (5 + 4 + 3) = 0{,}71$$

Clustering yang buruk mempunyai nilai *purity* mendekati 0, sedangkan *clustering* yang sempurna memiliki nilai *purity* sebesar 1 [15].

Selain menganalisis kualitas *cluster* dari kedua metode berdasarkan besarnya nilai *purity*, penelitian ini juga akan membandingkan konsistensi hasil *clustering* dari kedua metode. Konsistensi yang dimaksud adalah untuk melihat apakah *cluster* yang dihasilkan berubah jika dilakukan beberapa kali pengujian *clustering* dengan atribut-atribut yang sama persis. Oleh karena itu, pada penelitian ini akan dilakukan pengujian *clustering* pada masing-masing metode sebanyak 5 kali, dengan ketentuan yang sama dan data sampel yang sama persis.

III. HASIL DAN ANALISIS

Bagian ini akan membahas mengenai hasil pengujian serta analisis *cluster* yang dihasilkan oleh kedua metode yaitu *K-Means* dan *Modified K-Means*.

A. Hasil Pengujian

Bagian ini membahas tentang hasil pengujian dari dua metode yaitu *K-Means* dan *Modified K-Means.*

Tabel IV merupakan hasil pengujian pada metode *K-Means*. Dari 10 pengujian dengan jumlah persentase *features selection* yang meningkat, rata-rata nilai *purity* dari hasil *cluster* yang diperoleh sebesar 0,391 atau 39,1%.

TABEL IV
HASIL PENGUJIAN *K-MEANS*

***Feature Selection* (%)**	***Purity***	**Jumlah Komentar**								
		***Cluster* 1**			***Cluster* 2**			***Cluster* 3**		
		+	**-**	**N**	**+**	**-**	**N**	**+**	**-**	**N**
10	0,44	91	77	45	0	0	2	1	0	0
20	0,41	100	87	58	0	0	1	1	0	0
30	0,40	1	2	3	98	85	65	6	7	0
40	0,39	104	99	73	0	0	2	4	0	0
50	0,39	112	101	80	0	1	2	1	3	2
60	0,38	0	1	2	114	104	90	1	0	0
70	0,37	0	1	0	5	3	4	111	104	92
80	0,37	113	117	90	0	0	1	2	1	3
90	0,39	100	112	103	0	0	1	17	4	0
100	0,37	0	1	0	105	112	107	14	7	0

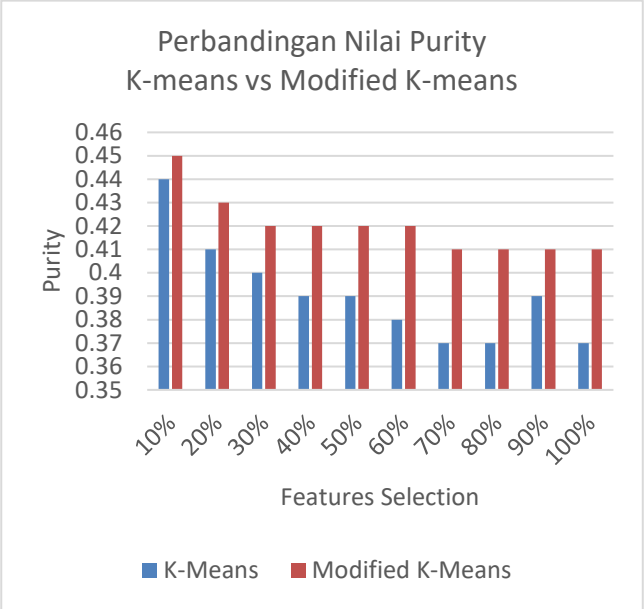
Sementara itu, Tabel V merupakan hasil dari pengujian pada metode *Modified K-Means*. Rata-rata nilai *purity* dari *cluster* yang dihasilkan sebesar 0,42 atau 42%.

TABEL V
HASIL PENGUJIAN *MODIFIED K-MEANS*

***Feature Selection* (%)**	***Purity***	**Jumlah Komentar**								
		***Cluster* 1**			***Cluster* 2**			***Cluster* 3**		
		+	**-**	**N**	**+**	**-**	**N**	**+**	**-**	**N**
10	0,45	66	72	39	20	15	0	1	1	6
20	0,43	77	81	52	17	4	0	7	2	7
30	0,42	81	87	59	17	4	0	7	3	9
40	0,42	84	91	64	17	4	0	7	4	11
50	0,42	89	98	72	17	4	0	7	3	12
60	0,42	90	98	78	18	4	0	7	3	14
70	0,41	91	101	85	18	4	0	7	3	11
80	0,41	92	105	88	18	4	0	7	4	11
90	0,41	92	108	91	18	4	0	7	4	13
100	0,41	96	110	93	18	5	0	7	4	14

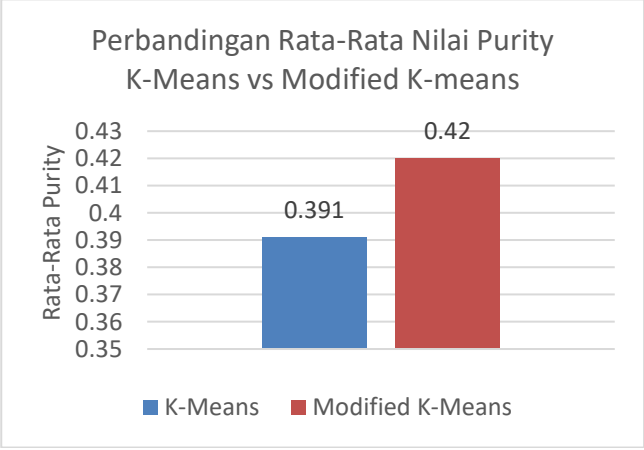
B. Analisis Perbandingan Nilai Purity K-Means dan Modified K-Means

p-ISSN : 2443-2210
e-ISSN : 2443-2229



Gambar 2. Grafik perbandingan nilai *purity K-Means* dan *Modified K-Means*

Gambar 2 merupakan grafik perbandingan antara metode *K-Means* dan *Modified K-Means* secara menyeluruh berdasarkan pada peningkatan *features selection*. Dari hasil pengujian tersebut, dapat dilihat bahwa *Modified K-Means* memiliki nilai *purity* yang lebih tinggi dibanding hasil *clustering* pada metode *K-Means* biasa. Sementara itu, jika di rata-rata, *Modified K-Means* menghasilkan rata-rata nilai *purity* sebesar 0,42 atau 42%, sedangkan metode *K-Means* menghasilkan rata-rata nilai *purity* sebesar 0,391 atau 39,1% sehingga terjadi peningkatan kualitas *cluster* sebesar 2,9%. Untuk lebih jelasnya, dapat dilihat pada grafik rata-rata nilai *purity* pada Gambar 3.



Gambar 3. Grafik perbandingan rata-rata nilai *purity* antara *K-Means* dan *Modified K-Means*

Dari hasil pengujian ini, dapat disimpulkan bahwa metode *Modified K-Means* mampu menghasilkan kualitas *cluster* yang lebih baik berdasarkan pada nilai *purity* yang dihasilkan.

C. Pengujian Konsistensi Hasil Cluster

Setelah menganalisis kualitas *cluster* dari kedua metode berdasarkan besarnya nilai *purity*, analisis selanjutnya yaitu membandingkan kualitas hasil *clustering* dari kedua metode berdasarkan konsistensi *cluster* yang dihasilkan. Analisis yang dimaksud adalah untuk melihat apakah *cluster* yang dihasilkan berubah jika dilakukan beberapa kali *clustering* dengan atribut-atribut yang sama persis. Pengujian akan dilakukan sebanyak 5 kali.

Tabel VI menunjukkan rangkuman dari 5 kali pengujian yang telah dilakukan pada metode *K-Means*. Pada Tabel VI, dapat dilihat bahwa selain nilai *purity* yang dihasilkan, anggota-anggota dari *cluster* yang dihasilkan juga selalu berubah setiap kali dilakukan pengujian. Dari hasil ini, maka dapat disimpulkan bahwa metode *K-Means* menghasilkan *cluster* yang tidak konsisten atau selalu berubah-ubah. Perubahan *cluster* yang dihasilkan juga mempengaruhi nilai *purity* yang juga ikut berubah.

TABEL VI
HASIL PENGUJIAN KONSISTENSI HASIL *CLUSTERING* PADA METODE *K-MEANS*

Pengujian ke-	***Purity***	**Jumlah Komentar**								
		***Cluster* 1**			***Cluster* 2**			***Cluster* 3**		
		+	-	N	+	-	N	+	-	N
1	0,43	72	78	44	0	2	1	15	8	0
2	0,41	1	1	2	0	0	1	86	87	42
3	0,41	80	82	43	0	0	1	7	6	1
4	0,40	1	2	0	0	0	1	86	86	44
5	0,40	85	86	42	0	0	1	2	2	2

Sementara itu, rangkuman pengujian konsistensi pada metode *Modified K-Means* dapat dilihat pada Tabel VII. Pada Tabel VII, dapat dilihat bahwa dari 5 kali pengujian yang dilakukan, baik nilai *purity* maupun *cluster* yang dihasilkan tidak pernah berubah. Hal ini terjadi karena pada *Modified K-Means*, *centroid* awal ditentukan menggunakan algoritma *modified centroid Selection* dimana tidak terdapat faktor *random*, sehingga selalu menghasilkan *centroid* awal yang sama. Sedangkan pada metode *K-Means*, *centroid* awal ditentukan secara *random*, sehingga *centroid* awal memungkinkan untuk berubah-ubah dan mempengaruhi hasil *clustering* yang juga berubah-ubah.

TABEL VII
HASIL PENGUJIAN KONSISTENSI HASIL *CLUSTERING* PADA METODE *K-MEANS*

Pengujian ke-	***Purity***	**Jumlah Komentar**								
		***Cluster* 1**			***Cluster* 2**			***Cluster* 3**		
		+	-	N	+	-	N	+	-	N
1	0,45	66	72	39	20	15	0	1	1	6
2	0,45	66	72	39	20	15	0	1	1	6
3	0,45	66	72	39	20	15	0	1	1	6
4	0,45	66	72	39	20	15	0	1	1	6
5	0,45	66	72	39	20	15	0	1	1	6

Dari pengujian ini, dapat disimpulkan jika metode *Modified K-Means* menghasilkan *cluster* yang konsisten karena tidak ada faktor *random*, sedangkan metode *K-Means* menghasilkan *cluster* dan nilai *purity* yang tidak konsisten atau berubah-ubah yang juga ikut mempengaruhi kualitas *cluster* yang dihasilkan.

D. Analisis Nilai Purity

Dari hasil pengujian pada kedua metode yang bisa dilihat pada Tabel III dan Tabel IV, rata-rata nilai *purity* yang dihasilkan kurang dari 0,5 atau kurang dari 50%. Rendahnya nilai *purity* ini disebabkan oleh *dataset* itu sendiri, lebih spesifiknya karena dari pemberian label komentar yang didasarkan pada pembuat status (Jokowi atau Prabowo). Hal ini bisa dibuktikan dengan adanya beberapa komentar yang cenderung identik namun memiliki label yang berbeda. Misalnya komentar "salam 2 jari", jika komentar ditujukan pada status pasangan nomor urut 1, maka labelnya akan negatif, sedangkan jika ditujukan pada status pasangan nomor urut 2, maka labelnya positif. Contoh kasus ini bisa dilihat pada Gambar 4 yang merupakan hasil dari salah satu *cluster* yang sebenarnya sangat identik namun memiliki label yang berbeda.

Id Komentar	Komentar	Label
1053	salam 2 jari :-)	negatif
1310	Slam 2 jarii	negatif
1316	Salam 2jari	negatif
1367	salam 2 jari	negatif
2384	wonosobo hadir slm 2 jri	positif
2515	Sukseses selalu pak ,,salam 2 jari	positif
2603	Salam 2 jari	positif
2604	salam 2 jari	positif
2608	salam 2 jari	positif
2612	Salam 2 jari	positif
2640	Slm 2 jari	positif

Gambar 4. Contoh *cluster* yang memiliki komentar identik dengan label berbeda

Selain masalah pada pelabelan, berdasarkan pada hasil pengamatan, penyebab kecilnya nilai *purity* yang dihasilkan kemungkinan terletak pada jumlah *cluster* yang ditetapkan. Pada penelitian ini, jumlah *cluster* yang ditetapkan sebanyak 3 sesuai dengan jumlah label yang ada. Namun, pada implementasinya, dalam satu *cluster* masih terdapat komentar-komentar yang masih bisa dipecah atau dikelompokkan lagi ke dalam *cluster* yang berbeda.

Id Komentar	Komentar	Label
305	hidup prabowo	positif
660	maju trus Pa Prabowo	positif
673	Nice 1	positif
751	Like prabowo	positif
911	Sudah dr dulu wowo...	negatif
1012	MANTAPrabowo	positif
1120	7DC06813 di invite ya	netral
1184	Kau'lah pilih'an ku Pa PRABOWO	positif
1208	Insya Allah jokowi menang,slm 2 jari..	positif
1360	Bgmn teknis bertanya-nya?	netral
1403	Bravo prabowo ?!	positif
1516	Mantapnya pak prabowo	positif
1620	nomor satu	positif
1669	Duit nya Dari mana tapi pak?	negatif
1884	Saya rasa pak kala gk tau bicara, hidup prabowo,	negatif
1947	no.1	positif
2015	Jokowi Yes	positif
2547	Jakarta	netral
2648	semangat terus Pak,salam 2 jari?	positif
2719	salam dua jari	positif
2783	Salam 2 jari	positif
2806	Salam 2 jari	positif
2952	anu mungkin ea	netral
2968	SalM 2 jari	positif

Gambar 5. Contoh *cluster* yang masih bisa dikelompokkan lagi

Gambar 5 merupakan contoh komentar-komentar dalam salah satu *cluster* yang masih bisa dikelompokkan lagi berdasarkan kemiripannya. Misalnya komentar-komentar seperti “hidup prabowo”, “like prabowo” dan sejenisnya bisa berada dalam *cluster* sendiri, lalu komentar seperti “salam 2 jari” juga bisa dikelompokkan ke dalam *cluster* yang berbeda. Dari pengamatan tersebut, kurangnya jumlah *cluster* yang ditetapkan kemungkinan menjadi salah satu penyebab buruknya kualitas *cluster* yang dihasilkan.

Sebagai pembuktian hasil analisis mengenai penyebab kecilnya nilai *purity* yang dihasilkan, dilakukan dua pengujian yaitu yang pertama adalah pengujian ulang dengan menggunakan data label yang sudah diperbaiki secara manual berdasarkan interpretasi sendiri, kemudian selanjutnya adalah pengujian dengan menggunakan jumlah *cluster* yang meningkat dari 3, 4, 5, 6 dan 7.

Berdasarkan pada hasil analisis sebelumnya mengenai perbandingan dari kedua metode, maka pengujian ini akan menggunakan metode *Modified K-Means*. Pengujian pada data label yang telah diperbaiki akan dibandingkan dengan pengujian sebelumnya yang dapat dilihat pada Tabel V, dimana perbandingan ini dilakukan untuk melihat apakah perbaikan label berpengaruh pada nilai *purity* yang dihasilkan. Sedangkan pada pengujian jumlah *cluster* bertujuan untuk melihat apakah dengan meningkatnya jumlah *cluster* juga bisa meningkatkan kualitas *cluster* yang diukur dari nilai *purity* yang dihasilkan.

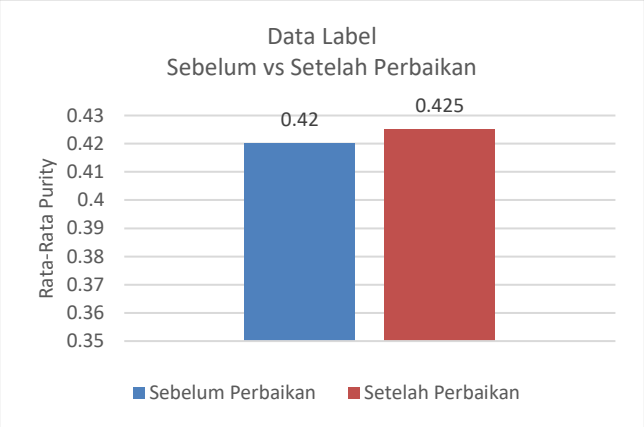
1) Pengujian pada Perbaikan Label: Hasil pengujian dengan menggunakan data label yang telah diperbaiki bisa dilihat pada Tabel VIII. Dari 10 pengujian yang dilakukan berdasarkan persentase *feature selection*, rata-rata nilai *purity* yang dihasilkan yaitu sebesar 0,425. Hasil ini lebih besar dari hasil pengujian pada data label yang belum diperbaiki yaitu sebesar 0,42 yang bisa dilihat pada Tabel V.

p-ISSN : 2443-2210
e-ISSN : 2443-2229

TABEL VIII
HASIL PENGUJIAN *MODIFIED K-MEANS* DENGAN PERBAIKAN LABEL DAN JUMLAH *CLUSTER* 3

***Feature Selection Threshold* (%)**	***Purity***
10	0,47
20	0,43
30	0,43
40	0,42
50	0,42
60	0,41
70	0,41
80	0,42
90	0,42
100	0,42

Gambar 6 merupakan grafik perbandingan nilai *purity* hasil *clustering* pada metode *Modified K-Means* yang menggunakan data label sebelum dan setelah diperbaiki. Dari hasil perbandingan tersebut, dapat dilihat jika perbaikan label dapat meningkatkan nilai *purity*, namun tidak begitu signifikan. Dimana peningkatan yang terjadi hanya sebesar 0,5%.



Gambar 6. Grafik perbandingan rata-rata nilai *purity* sebelum dan setelah perbaikan label

2) Pengujian pada Penambahan Jumlah Cluster: Pengujian jumlah *cluster* bertujuan untuk melihat apakah dengan meningkatnya jumlah *cluster* juga bisa meningkatkan kualitas *cluster* yang diukur dari nilai *purity* yang dihasilkan. Pengujian dilakukan dengan jumlah *cluster* 3, 4, 5, 6 dan 7.

Dari hasil pengujian sebelumnya mengenai perbaikan label, maka pengujian mengenai penambahan jumlah *cluster* akan menggunakan data label yang telah diperbaiki. Pengujian pada jumlah *cluster* sebesar 3 menggunakan hasil pengujian sebelumnya yang bisa dilihat pada Tabel VIII.

Sementara itu, hasil pengujian menggunakan jumlah *cluster* sebesar 4 dapat dilihat pada Tabel IX. Hasil pengujian pada jumlah *cluster* sebesar 4 menghasilkan rata-rata nilai *purity* sebesar 0,433, dimana hasil ini lebih besar dari rata-rata nilai *purity* yang dihasilkan pada jumlah *cluster* sebesar 3 dengan rata-rata *purity* sebesar 0,425.

TABEL IX
HASIL PENGUJIAN *MODIFIED K-MEANS* DENGAN PERBAIKAN LABEL DAN JUMLAH *CLUSTER* 4

***Feature Selection Threshold* (%)**	***Purity***
10	0,47
20	0,45
30	0,44
40	0,44
50	0,42
60	0,42
70	0,42
80	0,42
90	0,43
100	0,42

Kemudian hasil pengujian menggunakan jumlah *cluster* sebesar 5 dapat dilihat pada Tabel X Hasil pengujian pada jumlah *cluster* sebesar 5 menghasilkan rata-rata nilai *purity* sebesar 0,435, dimana hasil ini lebih besar dari rata-rata nilai *purity* yang dihasilkan pada jumlah *cluster* sebesar 4 dengan rata-rata *purity* sebesar 0,433.

TABEL X
HASIL PENGUJIAN *MODIFIED K-MEANS* DENGAN PERBAIKAN LABEL DAN JUMLAH *CLUSTER* 5

***Feature Selection Threshold* (%)**	***Purity***
10	0,49
20	0,44
30	0,44
40	0,44
50	0,43
60	0,42
70	0,42
80	0,42
90	0,42
100	0,43

Lalu, Tabel XI menunjukkan hasil pengujian menggunakan jumlah *cluster* sebesar 6. Hasil pengujian pada jumlah *cluster* sebesar 6 menghasilkan rata-rata nilai *purity* sebesar 0,438, dimana hasil ini lebih besar dari rata-rata nilai *purity* yang dihasilkan pada jumlah *cluster* sebesar 5 dengan rata-rata *purity* sebesar 0,435.

TABEL XI
HASIL PENGUJIAN *MODIFIED K-MEANS* DENGAN PERBAIKAN LABEL DAN JUMLAH *CLUSTER* 6

***Feature Selection Threshold* (%)**	***Purity***
10	0,51
20	0,45
30	0,44
40	0,45
50	0,46
60	0,42

***Feature Selection Threshold* (%)**	***Purity***
70	0,42
80	0,41
90	0,41
100	0,41

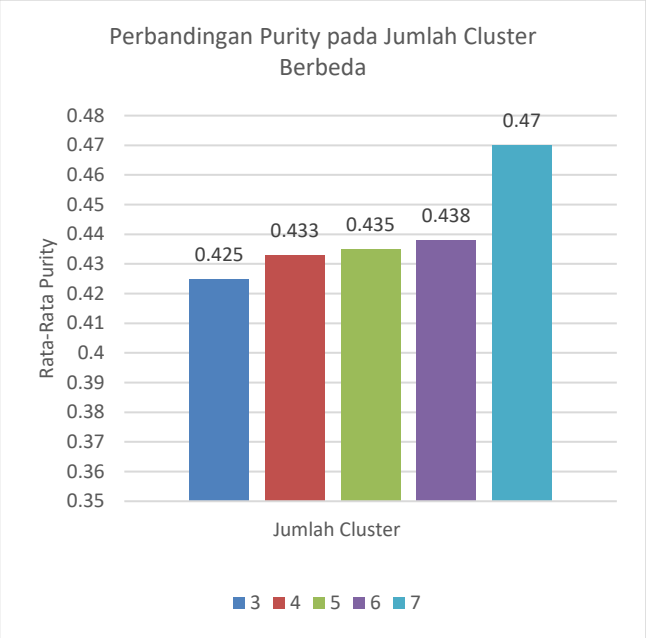
Selanjutnya, Tabel XII menunjukkan hasil pengujian menggunakan jumlah *cluster* sebesar 7. Hasil pengujian pada jumlah *cluster* sebesar 7 menghasilkan rata-rata nilai *purity* sebesar 0,47, dimana hasil ini lebih besar dari rata-rata nilai *purity* yang dihasilkan pada jumlah *cluster* sebesar 6 dengan rata-rata *purity* sebesar 0,438.

TABEL XII
HASIL PENGUJIAN *MODIFIED K-MEANS* DENGAN PERBAIKAN LABEL DAN JUMLAH *CLUSTER* 7

***Feature Selection Threshold* (%)**	***Purity***
10	0,51
20	0,48
30	0,50
40	0,49
50	0,48
60	0,45
70	0,45
80	0,45
90	0,44
100	0,45

Dari hasil pengujian dengan menggunakan jumlah *cluster* yang berbeda yang bisa dilihat pada Tabel VIII sampai Tabel XII, dilakukan perbandingan rata-rata nilai *purity* yang dihasilkan. Perbandingan tersebut bisa dilihat pada grafik perbandingan rata-rata nilai *purity* pada Gambar 7.

Pada Gambar 7, dapat dilihat bahwa penambahan jumlah *cluster* yang ditentukan dapat meningkatkan rata-rata nilai *purity* yang dihasilkan. Kemudian peningkatan yang paling signifikan terjadi pada jumlah *cluster* sebesar 7, dimana mengalami peningkatan sebesar 4,5% dibandingkan dengan jumlah *cluster* semula yang sebesar 3. Dari hasil pengujian ini maka dapat disimpulkan bahwa beberapa penyebab kecilnya nilai *purity* yang dihasilkan yaitu karena masalah pada pelabelan dan kurangnya jumlah *cluster* yang ditetapkan.



Gambar 7. Grafik perbandingan rata-rata nilai *purity* pada jumlah *cluster* yang berbeda

Pengujian pada perbaikan label dan penambahan jumlah *cluster* ini dilakukan bukan untuk mencari nilai *purity* yang terbaik, melainkan hanya untuk menganalisis penyebab buruknya kualitas *cluster* yang dihasilkan. Sehingga, masih diperlukan penelitian lebih lanjut untuk mencari jumlah *cluster* yang terbaik.

IV. KESIMPULAN

Dari penelitian dan analisis yang telah dilakukan, dapat ditarik beberapa kesimpulan. Yang pertama yaitu dari segi konsistensi hasil *cluster*, dimana metode *Modified K-Means* mampu menghasilkan *cluster* yang lebih konsisten. Dari 5 kali pengujian dengan data yang sama, metode *Modified K-Means* yang dalam penerapannya menggunakan algoritma *modified centroid Selection* dalam menentukan *centroid* awal tidak mengalami perubahan sama sekali, baik dari segi *purity* maupun *cluster* yang dihasilkan. Sementara pada metode *K-Means*, selalu terjadi perubahan *cluster* pada setiap pengujian yang dilakukan sehingga mempengaruhi nilai *purity* yang dihasilkan.

Sementara itu, dari segi akurasi atau kualitas *cluster*, penerapan metode *Modified K-Means* pada pengelompokan data komentar sentipol menghasilkan *cluster* yang lebih baik berdasarkan pada nilai *purity* yang dihasilkan. Dimana pada penggunaan algoritma *Modified K-Means* menghasilkan rata-rata nilai *purity* sebesar 0,42 atau 42% sementara *K-Means* menghasilkan rata-rata nilai *purity* sebesar 0,391 atau 39,1%. Namun, rata-rata persentase nilai *purity* yang dihasilkan masih dibawah 50%. Hal ini disebabkan karena masalah pelabelan pada *dataset* dan kurangnya jumlah *cluster* yang ditetapkan. Dari hasil pengujian dengan label yang telah diperbaiki secara manual, terjadi

peningkatan nilai *purity* sebesar 0,5%. Sedangkan pada pengujian penambahan jumlah *cluster*, dari jumlah *cluster* sebesar 3 sampai dengan 7, selalu terjadi peningkatan nilai *purity* setiap kali jumlah *cluster* ditambah.

Secara umum, algoritma *Modified K-Means* yang memodifikasi pemilihan *centroid* awal, bisa digunakan untuk mengoptimalkan kinerja dari algoritma *K-Means* yang mana kualitas *cluster* yang dihasilkan bergantung pada pemilihan *centroid* awal. Selain itu, berdasarkan hasil analisis dari penelitian ini, masih bisa dilakukan penelitian lebih lanjut dalam menentukan jumlah *cluster* yang terbaik untuk lebih mengoptimalkan akurasi atau kualitas *cluster* yang dihasilkan.

DAFTAR PUSTAKA

[1] A. Rachmat & Y. Lukito, "Sentipol: *Dataset* Sentimen Komentar pada Kampanye Pemilu Presiden Indonesia 2014 dari Facebook Page," *Konferensi Nasional Teknologi Informasi dan Komunikasi*, 2016, pp. 218-228.

[2] J. Han & M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed., San Francisco: Elsevier, 2006.

[3] A. Alrabea, A. V. Senthilkumar, H. Al-Shalabi & A. Bader, "Enhancing *K-Means* Algorithm with Initial *Cluster* Centers Derived from Data Partitioning along the Data Axis with PCA," *Journal of Advances in Computer Networks*, vol. 1, no. 2, pp. 137-142, Jun. 2013.

[4] A. C. Fabregas, B. D. Gerardo & B. T. Tanguilig III, "Enhanced Initial *Centroid*s for *K-Means* Algorithm," *International Journal of Information Technology and Computer Science*, vol. 9, no. 1, pp. 26-33, Jan. 4, 2017.

[5] S. Sujatha & A. S. Sona, "New Fast *K-Means Clustering* Algorithm using Modified *Centroid Selection* Method," *International Journal of Engineering Research & Technology*, vol. 2, no. 2, pp. 1-9, Feb. 2013.

[6] C. D. Manning, P. Raghavan & H. Schutze, *An Introduction to Information Retrieval*, Cambridge: Cambridge University Press, 2009.

[7] Rudy, "Perbandingan Metode *K-Means* Dan *Hierarchical Agglomerative Clustering* untuk Pengelompokan Dokumen Teks," Undergraduate thesis, Duta Wacana Christian University, 2009.

[8] S. S. Raghuwanshi & P. N. Arya, "Comparison of *K-Means* and Modified K-Mean Algorithms for Large Data-set," *International Journal of Computing, Communications and Networking*, vol. 1, no. 3, pp. 106-110, 2012.

[9] I. Safeek & M. R. Kalideen, "Preprocessing on Facebook Data for Sentiment Analysis," *7th International Symposium*, 2017, pp. 69-78.

[10] J. Joseph & J. R. Jeba, "Information Extaction Using Tokenization and *Clustering* Methods," *International Journal of Recent Technology and Engineering*, vol. 8, no. 4, pp. 3690-3692, Nov. 2019.

[11] J. K. Raulji & J. R. Saini, "Stop-Word *Removal* Algorithm and its Implementation for Sanskrit Language," *International Journal of Computer Applications*, vol. 150, no. 2, pp. 15-17, Sept. 2016.

[12] H. Christian, M. P. Agus & D. Suhartono, "Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF_IDF)," *Computer, Mathematics and Engineering Applications*, vol. 7, no. 4, pp. 285-294, Dec. 2016.

[13] T. Widiyaningtyas, M. I. W. Prabowo & M. A. M Pratama, "Implementation of *K-Means Clustering* Method to Distribution of High School Teachers," in *4th International Conference on Electrical Engineering, Computer Science and Informatics*, 2017, pp. 49-54.

[14] F. L. Sibuea & A. Sapta, "Pemetaan Siswa Berprestasi Menggunakan Metode *K-Means Clustering*," *Jurnal Teknologi dan Sistem Informasi*, vol. 4, no. 1, pp. 85-92, Dec. 2017.

[15] S. M. Kim, M. I. Pena, M. Moll, G. Giannakopoulos, G. N. Bennett & L. E. Kavraki, "An Evaluation of Different *Clustering* Methods and Distance Measures Used for Grouping Metabolic Pathways," *8th International Conference on Bioinformatics and Computational Biology*, Las Vegas, 2016.