Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 3, No. 3, Maret 2019, hlm. 2208-2214 http://j-ptiik.ub.ac.id

Peringkasan Teks Untuk Deteksi Kejadian Pada Dokumen Twitter Berbahasa Indonesia Dengan Metode Affinity Propagation

Rezky Dermawan[1], Fitra A. Bachtiar[2], Putra Pandu Adikara[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]rezkydermawan97@gmail.com, [2]fitra.bachtiar@ub.ac.id, [3]adikara.putra@ub.ac.id

Abstrak

Karakteristik Twitter di mana pengguna dapat membuat tweet di mana saja dalam waktu yang singkat, memungkinkan munculnya informasi penting sebelum media berita dapat melaporkannya. Namun karena besarnya skala dan keragaman topik yang dibicarakan tweet dalam Twitter, sangat sulit untuk mengetahui kejadian relevan yang berada di suatu tempat. Oleh karena itu, diperlukan sebuah sistem yang dapat mendeteksi kejadian relevan dan memberikan ringkasan tentang kejadian tersebut. Pada penelitian ini dilakukan proses peringkasan teks untuk deteksi kejadian dokumen Twitter berbahasa Indonesia dengan metode *Affinity Propagation*. Melalui proses pengklasteran akan didapatkan kumpulan klaster yang menjadi representasi kejadian pada waktu dan tempat tertentu. Pengujian dilakukan pada dua jenis data, yaitu tematik yang memiliki kejadian khusus dalam rentang waktu tweet dan umum di mana tweet diambil pada rentang waktu acak. Pengujian dilakukan pada parameter Affinity propagation yang kemudian menghasilkan nilai *preference* kuartil 3 dan minimum, nilai *damping factor* 0,3 dan 0,5, nilai *changed limit* 1 dan 2, nilai maksimum iterasi 250 sebagai parameter terbaik. Hasil ringkasan dari proses pengklasteran kemudian dibandingkan dengan hasil ringkasan pakar untuk dievaluasi menggunakan metode ROGUE-N, menghasilkan nilai 0,459 dan 0,4009 secara berturut-turut pada 2 jenis data.

Kata kunci: *twitter, affinity propagation, deteksi kejadian, peringkasan teks*

Abstract

Twitter's qualities where the user could make a tweet anywhere in a brief time frame, make it feasible for critical information to appear before the media even report it. However, it is difficult to comprehend what relevant events are occuring in a specific region because of the sheer size of scale and diverse sort of tweets. Accordingly, there is a need of a framework that could do pertinent event detection and give a summary about that event. In light of the reason expressed over, this research center around text summarization for event detection of Indonesian Twitter archive utilizing Affinity Propagation. Through the process of clustering, the resulting clusters become the representation of events occuring in a specific place and time period. Two kinds of data are used for assessment, first is themathic which has spesific kind of event happening in the time frame of the tweet and second is generic where the tweet are taken from an arbitary time frame. Evaluation are done to the parameters of Affinity Propagation, resulting in preference of quartile 3 dan minimum, damping factor of 0,3 and 0,5, changed limit of 1 and 2, iteration maximum of 250 as the best parameters. The result of tweets summary from the clustering process are then compared with a specialist's summary to be evaluated by ROGUE-N method, scoring 0,459 and 0,4009 respectively on two kinds of data.

Keywords: *twitter, affinity propagation, event detection, text sumarization*

1. PENDAHULUAN

Tiap harinya 313 juta pengguna Twitter mengirimkan lebih dari 500 juta *tweet* (Socialbakers, 2017). Kepopuleran Twitter berasal dari ketersediaannya dalam berbagai jenis alat elektronik (*web*, *smartphone*, dll) dan tren dalam Twitter yang menganjurkan pengguna untuk membentuk banyak kumpulan teman serta membuat *tweet* dalam berbagai jenis subjek, biasanya beberapa kali dalam sehari (Sankaranarayanan, Samet, Teitler,

Lieberman, & Sperling, 2009). Pengaruh Twitter juga telah sampai ke Indonesia, terutama kota Jakarta yang menghasilkan *tweet* terbanyak dari kota lain di seluruh dunia (Semiocast, 2012).

Pengguna Twitter sering menginformasikan kejadian yang ada di sekitar mereka baik dari hal trivial hingga hal penting seperti sebuah kecelakaan. Karakteristik Twitter yang memungkinkan pengguna untuk membuat *tweet* di mana saja dalam waktu yang singkat, memungkinkan munculnya informasi penting sebelum media berita dapat melaporkannya (Sakaki, Okazaki, & Matsuo, 2010). Kejadian yang berpengaruh pada kelancaran kota dapat membantu masyarakat dan pemerintah dalam menentukan keputusan. Namun karena besarnya skala dan keragaman topik yang dibicarakan dalam Twitter, sangat sulit untuk mengetahui kejadian relevan yang berada di suatu tempat (Atefeh & Khreich, 2015). Identifikasi kejadian dari pesan Twitter bukan masalah yang mudah karena pengguna Twitter yang beragam dan berjumlah sangat banyak. Isi pesan Twitter dapat berupa kehidupan personal penggunanya, yang tidak berhubungan dengan kejadian dalam dunia nyata.

Banyak penelitian telah dilakukan untuk mendeteksi kejadian dalam Twitter beserta lokasinya dengan akurat dan efisien. Salah satu penelitian membuat sebuah sistem yang mendeteksi kejadian lokal dengan mengekstrak kata kunci dan kemudian mengklaster kata kunci tersebut berdasarkan kemiripan lokasi saat *tweet* dikirim dengan algoritme *single-pass clustering* (Abdelhaq, Sengstock, & Gertz, 2013). Penelitian lain dengan tujuan yang sama, mendeteksi kejadian dalam Twitter dengan mengelompokkan *tweet* berdasarkan kemiripannya menggunakan metode *hierarchical clustering* dan mengidentifikasi lokasi kejadian berdasarkan lokasi pengguna (Unankard, Li, & Sharaf, 2015). Sebagai bentuk *post-process*, sebuah penelitian juga merekomendasikan peringkasan dokumen untuk memberikan informasi yang ringkas dibandingkan dengan melihat kumpulan *tweet* (Atefeh & Khreich, 2015). Salah satu metode peringkasan teks secara ekstraktif adalah dengan teknik pengelompokkan (Andhale & Bewoor, 2017).

Affinity Propagation merupakan salah satu teknik pengelompokkan yang mengidentifikasi *exemplar*, data yang menjadi titik pusat klaster, dari semua titik data berdasarkan pesan yang disampaikan antar titik hingga kumpulan *exemplar* dan klaster yang layak muncul (Frey & Dueck, 2007). Terdapat penelitian sebelumnya yang memanfaatkan *Affinity Propagation* untuk mengelompokkan *tweet* dalam Twitter (De Villiers, Hoffmann, & Kroon, 2012; Kang, Lerman, & Plangprasopchok, 2010; Rangrej, Kulkarni, & Tendulkar, 2011) dan menyimpulkan bahwa *Affinity Propagation* adalah metode yang menghasilkan klaster paling baik dibandingkan dengan beberapa metode pengklasteran lain.

Fokus dari penelitian ini adalah penggunaan metode *Affinity Propagation* pada peringkasan teks untuk deteksi kejadian dokumen Twitter berbahasa Indonesia. Data yang digunakan dalam penelitian ini adalah *tweet* berbahasa Indonesia yang terdapat pada satu daerah selama rentang waktu tertentu. Keluaran akhirnya berupa kumpulan *tweet* berjumlah sedikit yang merepresentasikan keseluruhan *tweet* pada daerah tersebut secara singkat.

2. AFFINITY PROPAGATION

Pada penelitiannya, Frey dan Dueck pada tahun 2007 memperkenalkan sebuah metode yang secara bersamaan mempertimbangkan semua data sebagai *exemplar* (Frey & Dueck, 2007). Pesan bernilai saling ditukarkan antar data hingga kumpulan *exemplar* dan klaster yang layak muncul. *Affinity Propagation* menerima sebagai masukan kumpulan nilai kemiripan antar data, di mana kemiripan $s(i,k)$ mengindikasikan seberapa cocok data dengan indeks *k* untuk menjadi *exemplar* data indeks *i*. Daripada memerlukan banyak kelompok ditentukan, *Affinity Propagation* menerima masukan sebuah nilai untuk *s(k,k)* pada setiap data *k* disebut sebagai *preference*. Banyaknya *exemplar* yang muncul (jumlah kelompok) dipengaruhi oleh nilai masukan *preference*, tetapi juga muncul dari prosedur pertukaran pesan. Nilai *preference* yang digunakan bisa saja didapat dari median nilai kemiripan untuk mendapatkan klaster berjumlah sedang, atau nilai minimum untuk mendapatkan klaster berjumlah kecil.

Setiap data memiliki nilai *responsibility* dan *availability* dengan data lain. Pada awalnya *availability* diinisialisasi dengan 0, lalu *responsibility* dihitung dengan Persamaan (1).

$$r(i,k) \leftarrow s(i,k) - \max_{k's.t.k' \neq k} \{a(i,k') + s(i,k')\} \quad (1)$$

Keterangan:

s(i,k): Nilai *similarity* antara data *i* dengan data *k*
a(i,k'): Nilai *availability* antara data *i* dengan data *k'*, di mana *k'≠k*
s(i,k'): Nilai *similarity* antara data *i* dengan data *k'*, di mana *k'≠k*

Kemudian *availability* dihitung dengan Persamaan (2).

$$a(i,k) \leftarrow \min\left\{0, r(k,k) + \sum_{i's.t.i' \neq \{i,k\}} \max\{0, r(i',k)\}\right\} \quad (2)$$

Khusus jika nilai i sama dengan k maka *availability* dihitung dengan Persamaan (3).

$$a(k,k) \leftarrow \sum_{i's.t.i' \neq k} \max\{0, r(i',k)\} \quad (3)$$

Keterangan:
r(k,k): Nilai *Responsibility* antara data *k* dengan data *k*
r(i',k): Nilai *Responsibility* antara data *i'* dengan data k, di mana *i'≠k*

Ketika memperbaharui pesan, penting agar nilainya diperkecil dengan sebuah faktor pengecil (*damping factor*) untuk menghindari hasil yang bergelombang atau tidak konvergen. Proses ini dihitung menggunakan Persamaan (4) dan (5).

$$a(i,k)_t = \lambda a(i,k)_{t-1} + (1-\lambda) a(i,k) \quad (4)$$

$$r(i,k)_t = \lambda r(i,k)_{t-1} + (1-\lambda) r(i,k) \quad (5)$$

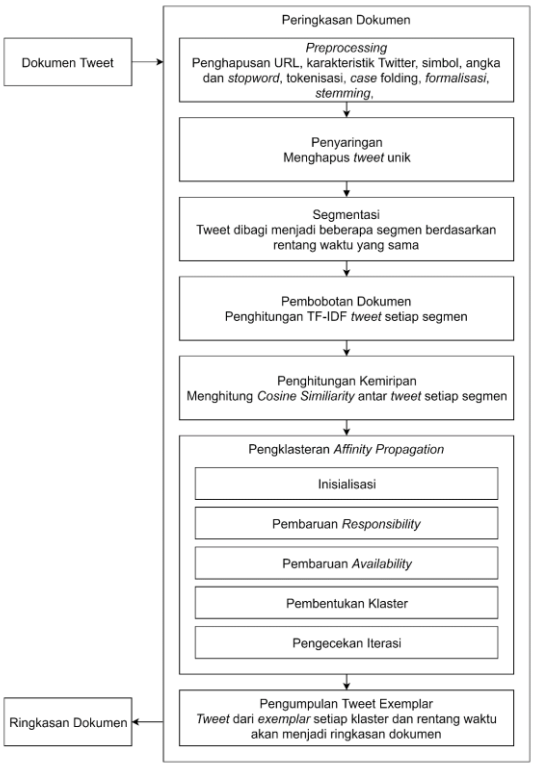
Keterangan:
λ : Nilai *damping factor*, antara 0 sampai 1
a(i,k): Nilai *availability* sekarang antara data *i* dengan data *k*
$a(i,k)_{t-1}$: Nilai *availability* iterasi sebelum antara data *i* dengan data *k*
r(i,k): Nilai *responsibility* sekarang antara data *i* dengan data *k*
$r(i,k)_{t-1}$: Nilai *responsibility* iterasi sebelum antara data *i* dengan data *k*

Pada saat kapanpun, *availability* dan *responsibility* bisa digabung untuk mengidentifikasi *exemplar*. Proses penentuan *exemplar* ini dihitung menggunakan Persamaan (6). Prosedur pertukaran pesan bisa dihentikan setelah sejumlah iterasi yang ditentukan, ketika perubahan pada pesan berada di bawah suatu batas, atau ketika keputusan lokal tetap konstan setelah beberapa iterasi.

$$exemplar(i) = \arg\max_k \{r(i,k) + a(i,k)\} \quad (6)$$

Keterangan:
exemplar(i): Data yang menjadi *exemplar* data *i*
r(i,k): Nilai *responsibility* sekarang antara data *i* dengan data *k*
a(i,k): Nilai *availability* sekarang antara data *i* dengan data *k*

3. METODOLOGI



Gambar 1. Alir Diagram Metodologi

Peringkasan dokumen teks untuk deteksi kejadian pada dokumen Twitter berbahasa Indonesia dengan metode *Affinity Propagation* yang akan dibangun menerima masukan berupa *tweet* yang memiliki informasi geolokasi pada suatu daerah dalam rentang waktu tertentu. Keluaran dari sistem ini adalah kumpulan *tweet* yang lebih sedikit dan dapat merepresentasikan kejadian yang ada. Tahapan proses implementasi algoritme yang terdapat pada Gambar 1 dijelaskan sebagai berikut.

3.1. Pre-processing

Sebelum dapat diterapkan proses pembelajaran seperti pengklasteran, kumpulan dokumen *tweet* perlu melewati tahap *pre-processing* terlebih dahulu. Tujuan dari tahap *pre-processing* adalah mengubah kumpulan dokumen menjadi kumpulan data yang dapat dianalisis lebih lanjut. Namun karena sifat pesan Twitter yang tidak terstruktur dan kompleks, tahapan *pre-processing* standar kurang efektif jika digunakan pada pesan Twitter. Pada umumnya, pesan Twitter memiliki kata salah eja, singkatan, simbol, URL, dan lainnya yang tidak relevan untuk

pengolahan dokumen. Oleh karena itu, perlu suatu tahapan *pre-processing* khusus untuk pesan Twitter dalam bahasa Indonesia.

3.2. Penyaringan

Proses penyaringan bertujuan untuk mengurangi jumlah *tweet* yang akan diproses sehingga datanya lebih sedikit dan relevan. *Tweet* yang dihapus ditentukan berdasarkan 2 kriteria, yaitu minimal kata dan kemunculan kata populer. Pada minimal kata, sebuah *tweet* harus memiliki jumlah kata dari hasil pre-processing sebanyak yang ditentukan. Pada kemunculan kata populer, sebuah *tweet* harus memiliki sebuah kata populer atau kata yang muncul pada *tweet* lain sebanyak yang ditentukan.

3.3. Segmentasi

Proses segmentasi membagi kumpulan *tweet* menjadi beberapa segmen atau kelompok berdasarkan rentang waktu yang ditentukan. Hal ini bertujuan untuk membagi klaster kedalam rentang waktu yang kecil agar kejadiannya lebih spesifik serta memudahkan proses pengklasteran dalam mengolah data. Setiap proses berikutnya akan dijalankan pada setiap segmen.

3.4. Pembobotan Dokumen

Proses pembobotan dokumen bertujuan untuk mendapatkan representasi data numerik dari kumpulan *tweet* yang telah diproses sebelumnya. Bobot dokumen dihitung menggunakan metode TF-IDF, ditampilkan dalam Persamaan (7).

$$w_{t,d} = tf_{t,d} \times \log(\frac{N}{df_t}) \quad \textbf{(7)}$$

Keterangan:
$tf_{t,d}$: Frekuensi kemunculan term *t* pada tweet *d*
df_t: Frekuensi kemunculan term *t* pada seluruh *tweet*

3.5. Penghitungan *Cosine Similarity*

Proses penghitungan kemiripan bertujuan untuk mengubah data numerik tiap *tweet* dari proses sebelumnya menjadi data jarak antar *tweet* sehingga dapat diproses oleh pengklasteran *Affinity Propagation*. Perhitungan jarak dokumen yang digunakan pada penelitian ini adalah *Cosine Similarity* seperti yang ditunjukkan pada Persamaan (8).

$$sim(a,b) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^{\,2}} \sqrt{\sum_{i=1}^{n} b_i^{\,2}}} \quad (8)$$

Keterangan:
a_i: Nilai ke *i* dari vektor *a*
b_i: Nilai ke i dari vektor *b*

3.6. Pengklasteran *Affinity Propagation*

Proses pengklasteran *Affinity Propagation* merupakan proses utama yang mengubah data numerik dari proses pengolahan data teks menjadi informasi yang berguna, yaitu kumpulan klaster *tweet* beserta *exemplar*-nya. Hasil *Cosine Similarity* dari proses sebelumnya menjadi masukan dalam pengklasteran *Affinity Propagation*. Kemudian melalui proses pengklasteran yang telah dijelaskan tentang *Affinity Propagation*. Setelah melalui beberapa proses iterasi akan didapat kumpulan klaster *tweet* yang optimal untuk setiap segmen. *Exemplar* dari setiap klaster pada seluruh segmen akan menjadi hasil ringkasan sistem.

4. HASIL DAN PEMBAHASAN

4.1 Pengujian Pengklasteran

Pengujian pertama yang dilakukan dalam penelitian ini adalah menentukan parameter yang dapat menghasilkan hasil pengklasteran paling baik berdasarkan nilai *Silhouette Coefficient*. Semua pengujian parameter menggunakan data yang sama yaitu hasil pengolahan data teks dari data *tweet* tematik sebanyak 32.256 dan umum sebanyak 102.214. Parameter yang digunakan untuk pengolahan data *tweet* tematik adalah minimal kata 5, minimal kemunculan kata 80, dan rentang waktu segmen 3 jam. Parameter yang digunakan untuk pengolahan data *tweet* umum adalah minimal kata 5, minimal kemunculan kata 255, dan rentang waktu segmen 6 jam. Data umum memiliki rentang waktu segmen yang lebih kecil karena *tweet* per harinya lebih padat dibandingkan dengan data tematik. Selain itu juga, dengan memperkecil jumlah *tweet* yang diproses setiap pengklasteran dapat mempercepat proses pengujian karena keterbatasan waktu.

Setelah melalui proses penyaringan, terdapat 26.115 *tweet* pada data tematik dan 34.926 *tweet* pada data umum. Tahapan ini menghilangkan 19% dan 66% secara berturut *tweet* yang tidak relevan pada data tematik dan umum. Setelah melalui proses segmentasi, terdapat 60 segmen pada data tematik dan 46 segmen pada data umum. Setiap segmen tersebut dihitung pembobotan dan *Cosine Similarity*-nya untuk digunakan pada pengujian parameter berikut. Pada percobaan pertama nilai parameter yang digunakan dalam pengklasteran *tweet* adalah *damping factor* 0,5,

changed limit 2, maksimum iterasi 10. Untuk percobaan selanjutnya akan menggunakan nilai parameter terbaik dari pengujian sebelumnya.

4.1.1. Pengujian Parameter *Preference*

Nilai *preference* didapatkan dengan mengambil salah satu nilai dari matriks *Cosine Similarity* yang telah dirata dan diurutkan. Semakin kecil nilai *preference* maka klaster yang dihasilkan lebih banyak, sebaliknya juga jika nilai *preference* lebih besar menghasilkan jumlah klaster yang lebih banyak. Parameter yang diuji adalah nilai minimum, kuartil 1 (Q1), kuartil 2 atau median (Q2), dan kuartil 3 (Q3) dari matriks *Cosine Similarity* tersebut. Nilai *Silhouette Coefficient* terbaik didapatkan dengan menggunakan nilai *preference* Q3 data tematik dan nilai *preference* minimum untuk data umum. Hasil penghitungan *Silhouette Coefficient* untuk pengujian parameter *preference* ditampilkan pada Tabel 1.

Tabel 1. Hasil Pengujian Parameter *Preference*

Parameter *Preference*	*Silhouette Coefficient*	
	Tematik	Umum
Minimum	0,01095	0,07260
Q1	0,01095	0,07260
Q2	0,01095	0,07260
Q3	0,01156	0,07260

Dapat dilihat dari hasil pengujian pada Tabel 1 bahwa hasil *Silhouette Coefficient* tidak banyak berubah, hal ini dikarenakan banyaknya pasangan *tweet* yang tidak mirip sama sekali. Semakin banyak pasangan *tweet* yang tidak mirip maka semakin banyak nilai 0 yang ada di dalam matriks *Cosine Similarity*. Berdasarkan hasil ini dapat dilihat *Cosine Similarity* untuk data tematik memiliki lebih dari 50% nilai 0 dan untuk data umum memiliki lebih dari 75% nilai 0. Oleh karena itu, *preference* selalu bernilai 0 untuk minimum, Q1, dan Q2 pada data tematik dan bernilai 0 untuk minimum, Q1, Q2, dan Q3 untuk data umum.

4.1.2. Pengujian Parameter *Damping factor*

Nilai *Silhouette Coefficient* terbaik didapatkan dengan menggunakan nilai *damping factor* 0,5 untuk data tematik dan nilai *damping factor* 0,3 untuk data umum. Hasil penghitungan *Silhouette Coefficient* untuk pengujian parameter *damping factor* ditampilkan pada Tabel 2.

Berdasarkan hasil pengujian damping factor, nilai yang terlalu rendah atau terlalu tinggi mengurangi kualitas klaster yang dihasilkan. Nilai damping factor yang kecil bisa lebih akurat tapi memerlukan iterasi yang lebih banyak, sedangkan nilai yang lebih besar kesulitan mencapai konvergen karena perubahan yang besar setiap iterasi.

Tabel 2. Hasil Pengujian Parameter *Damping Factor*

Parameter *Damping factor*	*Silhouette Coefficient*	
	Tematik	Umum
0	-0,93325	-0,95778
0,1	-0,32380	-0,47750
0,2	-0,05268	0,05328
0,3	-0,03205	0,07827
0,4	-0,00750	0,07677
0,5	0,01156	0,07260
0,6	0,00307	0,03469
0,7	-0,06631	-0,04347
0,8	-0,35099	-0,34587
0,9	-0,65030	-0,67190

4.1.3. Pengujian Parameter *Changed limit*

Nilai *Silhouette Coefficient* terbaik didapatkan dengan menggunakan nilai *changed limit* 1 untuk data tematik dan nilai *changed limit* 2 untuk data umum. Hasil penghitungan *Silhouette Coefficient* untuk pengujian parameter *changed limit* ditampilkan pada Tabel 3.

Tabel 3. Hasil Pengujian *Changed limit*

Parameter *Changed limit*	*Silhouette Coefficient*	
	Tematik	Umum
1	0,01156	0,07769
2	0,01156	0,07827
3	0,01156	0,07827
4	0,01156	0,07827
5	0,01156	0,07827

4.1.4. Pengujian Parameter Maksimum Iterasi

Nilai *Silhouette Coefficient* terbaik didapatkan dengan menggunakan nilai maksimum iterasi 250 untuk data tematik dan nilai maksimum iterasi 250 untuk data umum. Hasil penghitungan *Silhouette Coefficient* untuk pengujian parameter maksimum iterasi ditampilkan pada Tabel 4.

Tabel 4. Hasil Pengujian Maksimum Iterasi

Parameter	*Silhouette Coefficient*	
	Tematik	Umum
50	0,08979	0,13361
100	0,13906	0,14020
150	0,14407	0,14601
200	0,14306	0,14785
250	0,14465	0,15140

Berdasarkan hasil pengujian ini, nilai maksimum iterasi yang didapatkan belum termasuk parameter dengan *Silhouette Coefficient* tertinggi. Hal ini karena masih ada kemungkinan mendapatkan hasil *Silhouette Coefficient* yang lebih baik jika menggunakan nilai maksimum iterasi lebih dari 250. Namun karena keterbatasan waktu dan sumber daya pengujian ini hanya dapat mencoba hingga 250.

4.2 Pengujian Hasil Ringkasan

Pengujian kedua yang dilakukan pada penelitian ini adalah mengevaluasi hasil ringkasan yang didapatkan dari klaster terbaik berdasarkan hasil pengujian pengklasteran. Pengujian ini dilakukan dengan metode ROUGE-N, yaitu memanfaatkan pakar untuk mendapatkan ringkasan kemudian dibandingkan dengan ringkasan sistem.

Tabel 5. Hasil Pengujian ROGUE-N

Klaster	ROGUE-N	
	Tematik	Umum
1	0,1	0,166
2	1	1
3	0,117	0,15
4	1	0,083
5	0,181	0,214
6	1	0,125
7	1	0.062
8	0,117	1
9	0,009	0,214
10	0,066	1
Rata-rata	0,459	0,4009

Pada pengujian dengan metode ROGUE-N akan diambil 10 klaster secara acak dari masing-masing hasil pengklasteran, yaitu 5864 klaster untuk data tematik dan 6550 klaster data umum. Dari total 20 klaster tersebut, seorang pakar akan diminta untuk mengambil sebuah *tweet* dari masing-masing klaster yang dapat merepresentasikan klaster tersebut. Hasil *tweet* yang dipilih oleh pakar kemudian akan dibandingkan dengan hasil dari sistem dengan metode ROGUE-N untuk mendapatkan tingkat akurasi peringkasan sistem. Pakar yang melakukan pengujian dalam penelitian ini adalah seorang yang telah melalui edukasi tingkat lanjut dan berfokus pada Bahasa Indonesia. Hasil pengujian ROGUE-N antara pakar dan sistem dapat dilihat pada Tabel 5.5

Dari hasil pengujian pada Tabel 5, terdapat hasil ROGUE-N klaster bernilai 1, artinya *tweet* yang dipilih oleh sistem untuk menjadi ringkasan adalah sama dengan yang dipilih oleh pakar. Pada data tematik terdapat 4 klaster yang memiliki hasil ringkasan yang sama dengan pakar. Pada data umum terdapat 3 klaster yang memiliki hasil ringkasan yang sama dengan pakar. Klaster yang hasil ringkasannya tidak sama dengan pakar memiliki nilai ROGUE-N yang rendah, hal ini karena hanya ada satu atau beberapa kata yang sesuai antara tweet dari sistem dengan tweet dari pakar. Rata-rata dari nilai ROGUE-N seluruh klater yang diujikan adalah 0,459 untuk data tematik dan 0,4009 untuk data umum.

Hasil ringkasan dari data tematik memiliki nilai rata-rata ROGUE-N yang lebih tinggi dibandingkan dengan data umum, hal ini karena banyaknya klaster dengan topik pembicaraan yang jelas yaitu tentang ASIAN Games. Nilai evaluasi yang termasuk rendah pada kedua jenis data disebabkan karena banyaknya klaster dengan kumpulan *tweet* dengan maksud berbeda namun dalam klaster yang sama. Ini terjadi karena kemiripan *tweet* dikelompokkan berdasarkan kata yang telah di-*stemming*, sehingga *tweet* dengan makna berbeda akan dikelompokkan hanya karena satu atau dua kata dasar yang mirip.

5. KESIMPULAN

Pada penelitian ini dilakukan pengujian menggunakan data Twitter berupa *tweet* tematik dan umum. Memanfaatkan proses penyaringan jumlah *tweet* yang diproses dapat dikurangi dari 32.256 menjadi 26.115 untuk data tematik dan 102.214 menjadi 34.926 untuk data umum. Setelah melalui proses segmentasi , setiap *tweet* dalam setiap segmen kemudian dilakukan penghitungan TF-IDF dan *Cosine Similarity* untuk digunakan pada proses pengklasteran *Affinity Propagation*.

Pengujian dilakukan untuk mendapatkan parameter *Affinity Propagation* yang menghasilkan klaster optimal. Parameter yang optimal untuk data tematik dan umum adalah *preference* kuartil 3 dan minimum, *damping factor* 0,5 dan 0,3, *changed limit* 1 dan 2, serta maksimum iterasi 250. Hasil klaster terbaik tersebut memiliki nilai *Silhouette Coefficient* 0,14465 dan 0,1554. Hasil ringkasan dari pengklasteran *Affinity Propagation* menggunakan parameter terbaik kemudian dibandingkan dengan ringkasan dari seorang pakar. Data tematik memiliki nilai rata-rata ROGUE-N 0,459 dan data umum memiliki nilai rata-rata ROGUE-N 0,4009.

Berdasarkan penelitian peringkasan teks untuk deteksi kejadian pada Twitter berbahasa Indonesia dengan metode *Affinity Propagation* yang telah dilakukan, terdapat beberapa kelemahan yang dapat menjadi fokus penelitian selanjutnya. Pada penelitian selanjutnya disarankan untuk memanfaatkan koordinat dari *tweet* serta Named Entity Recognition (NER) untuk mendeteksi nama tempat dalam *tweet* dalam menentukan lokasi spesifik sehingga dapat ditampilkan pada peta. Sebuah sistem yang mendeteksi *spam* dapat digunakan untuk lebih mengurangi *tweet* yang tidak relevan. Disarankan juga membangun sistem yang dapat menyampaikan kejadian secara *real-time* untuk mendapatkan informasi kejadian lebih cepat

6. DAFTAR PUSTAKA

Abdelhaq, H., Sengstock, C., & Gertz, M. (2013). EvenTweet: Online Localized Event Detection from Twitter. *Proc. VLDB Endow.*, *6*(12), 1326–1329.

Andhale, N., & Bewoor, L. A. (2017). An overview of text summarization techniques. *Proceedings - 2nd International Conference on Computing, Communication, Control and Automation, ICCUBEA 2016*.

Atefeh, F., & Khreich, W. (2015). A Survey of Techniques for Event Detection in Twitter. *Comput. Intell.*, *31*(1), 132–164.

De Villiers, F., Hoffmann, M., & Kroon, S. (2012). Unsupervised construction of topic-based twitter lists. *Proceedings - 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust and 2012 ASE/IEEE International Conference on Social Computing, SocialCom/PASSAT 2012*, 283–292.

Frey, B. J., & Dueck, D. (2007). Clustering by Passing Messages Between Data Points. *Science*, *315*(5814), 972–976.

Kang, J. H., Lerman, K., & Plangprasopchok, A. (2010). Analyzing Microblogs with Affinity Propagation. In *Proceedings of the First Workshop on Social Media Analytics* (hal. 67–70). New York, NY, USA: ACM.

Rangrej, A., Kulkarni, S., & Tendulkar, A. V. (2011). Comparative Study of Clustering Techniques for Short Text Documents. *Proceedings of the 20th international conference companion on World wide web*, 111–112.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. *Proceedings of the 19th International Conference on World Wide Web*, 851–860.

Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D., & Sperling, J. (2009). TwitterStand: News in Tweets. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, 42.

Semiocast. (2012). Twitter reaches half a billion accounts (140M in the US). Diambil 7 Maret 2017, dari http://semiocast.com/publications/2012_07_30_Twitter_reaches_half_a_billion_accounts_140m_in_the_US

Socialbakers. (2017). All Twitter statistics in one place. Diambil 20 Mei 2017, dari https://www.socialbakers.com/statistics/twitter/

Unankard, S., Li, X., & Sharaf, M. A. (2015). Emerging event detection in social networks with location sensitivity. *World Wide Web*, *18*(5), 1393–1417.