MENGELOMPOKKAN *SHORT MESSAGE SERVICE* (SMS) PENIPUAN MENGGUNAKAN ALGORITME ROCK

CLUSTERING IN SHORT MESSAGE SERVICE (SMS) FRAUD USING ROCK ALGORITHM

Pritasari Palupiningsih
Sistem Informasi, STMIK Indonesia
pritasari.palupiningsih@gmail.com

ABSTRAK

Salah satu media yang digunakan dalam tindak penipuan pada saat ini adalah *Short Message Service* (SMS). Dengan menggunakan SMS sebagai media tindak penipuan, keberadaan penipu akan sulit untuk dilacak. Penelitian ini bertujuan untuk membentuk klaster (kelompok) data SMS dengan algoritme ROCK agar diperoleh informasi pola SMS yang memiliki indikasi tindak penipuan. Selain itu, penelitian ini juga untuk menentukan ukuran banyak klaster dan *threshold* yang memberikan klaster terbaik. Dalam penelitian ini digunakan teknik perbaikan kata dalam SMS. SMS berindikasi tindak penipuan yang dicari polanya adalah SMS yang berisi permintaan pulsa telepon seluler ke nomor telepon seluler tertentu, SMS yang berisi penawaran menjadi agen pulsa, dan SMS yang menunjukkan ketertarikkan pada proses jual beli tanah, rumah, atau mobil. Berdasarkan penelitian, diperoleh bahwa algoritme ROCK dapat diterapkan pada data SMS dan diperoleh klaster terbaik dengan kriteria banyak klaster 5 dengan *threshold* 0.08.

Kata kunci: SMS penipuan, *text mining*, klaster SMS, *threshold*, algoritme ROCK

ABSTRACT

Nowadays the one of media that used for fraud is SMS. By using SMS as a medium for fraud , the location of fraudsters would be difficult to track. This research aims to form a cluster of SMS data using ROCK algorithm to obtain information about the pattern of SMS which indicates the presence of fraud. This research also determine the best size of cluster's number and threshold. Moreover, semantic word correction technique was applied. SMS which have indication of fraud, used in this research are SMS containing request of handphone voucher to certain phone number, offering to be voucher agent, and interest in the activities of buying and selling land. Result from this research is ROCK algorithm can used for clustering SMS and 5 cluster with threshold 0.08 are the best cluster.

Keywords*: fraud SMS, text mining, klaster SMS, threshold, ROCK algorithm*

PENDAHULUAN

Short Message Service (SMS) merupakan salah satu media komunikasi yang banyak digunakan saat ini karena kepraktisannya dan biaya pengiriman yang murah. Namun, seiring dengan semakin populernya penggunaan SMS, ada sekelompok orang yang memanfaatkan SMS untuk melakukan tindak penipuan. Dengan menggunakan SMS sebagai media tindak penipuan, keberadaan penipu akan sulit untuk dilacak. Saat ini, tindak penipuan melalui SMS semakin marak terjadi sehingga masyarakat menjadi resah, apakah SMS yang mereka terima memiliki indikasi tindak penipuan melalui SMS atau tidak.

Ada beberapa macam tindak penipuan yang dilakukan melalui SMS. Salah satu tindak penipuan yang saat ini marak di masyarakat adalah SMS yang menunjukkan ketertarikan pada proses jual beli tanah, rumah, atau mobil.

SMS yang beredar di masyarakat dapat dimanfaatkan untuk memperoleh informasi tentang pola SMS yang berindikasi penipuan. Apabila pola SMS berindikasi penipuan tersebut dapat diketahui, maka masyarakat dapat lebih berhati-hati dalam menindaklanjuti SMS yang diterimanya sehingga dapat dicegah terjadinya penipuan melalui SMS. Untuk itu, dilakukan analisis pola SMS yang berindikasi adanya tindak penipuan dan menggunakan pola SMS tersebut sebagai prediksi terhadap suatu SMS, untuk menentukan ada atau tidaknya indikasi penipuan.

Analisis klaster mengelompokkan objek-objek data berdasarkan informasi pada data, yang menjelaskan objek dan relasinya. (Han & Kamber 2012). Analisis klaster menghasilkan kelompok data yang bermakna dengan diketahuinya struktur alami kelompok data tersebut.

Penelitian ini bertujuan untuk membentuk klaster data SMS dengan algoritme ROCK dan menentukan ukuran banyak klaster dan *threshold* yang memberikan klaster terbaik. Data SMS yang digunakan pada penelitian ini adalah data SMS yang beredar di masyarakat pada tahun 2014 sampai dengan tahun 2015, baik yang memiliki indikasi tindak penipuan dan tidak memiliki indikasi tindak penipuan. Indikasi tindak penipuan yang ditentukan polanya adalah SMS yang berisi permintaan pulsa telepon seluler ke nomor tertentu, dengan mengatasnamakan orangtua, yaitu mama atau papa. Kemudian SMS yang berisi penawaran menjadi agen pulsa dan SMS yang menunjukkan ketertarikan pada proses jual beli tanah, rumah, atau mobil.

Algoritme ROCK (*Robust Clustering Using Links*)

ROCK adalah algoritme klastering hirarki aglomeratif untuk mengelompokkan data kategorik. Algoritme ROCK membangun *link* untuk menggabungkan klaster dan tidak menggunakan jarak seperti algoritme klastering pada umumnya (Guha *et al.* 2000). Parameter yang digunakan dalam algoritme ROCK diuraikan sebagai berikut.

1. Tetangga

Tetangga objek adalah objek lain yang dianggap paling mirip dengan objek tersebut. Diberikan suatu nilai ambang (θ) yang bernilai antara 1 dan 0. Dua objek $\boldsymbol{x}$ dan $\boldsymbol{y}$, $\mathrm{sim}(\boldsymbol{x},\boldsymbol{y}) \geq \theta$. θ merupakan parameter yang ditentukan oleh pengguna yang dapat digunakan untuk mengontrol seberapa dekat hubungan $\boldsymbol{x}$ dan $\boldsymbol{y}$ sehingga kedua objek tersebut dapat dikatakan sebagai tetangga. Ukuran kemiripan antarpasangan objek dihitung dengan *Jaccard Coefficient*

2. Link

Algoritme ROCK menggunakan informasi *link* sebagai ukuran kemiripan antarobjek. Jika $\boldsymbol{x}$ merupakan tetangga dari $\boldsymbol{z}$ dan $\boldsymbol{z}$ merupakan tetangga dari $\boldsymbol{y}$ maka dikatakan $\boldsymbol{x}$ memiliki *link* dengan $\boldsymbol{y}$ walaupun $\boldsymbol{x}$ bukan tetangga dari $\boldsymbol{y}$. Didefinisikan :

$$\mathrm{Link}(\boldsymbol{x},\boldsymbol{y}) = \sum \text{tetangga yang dimiliki sekaligus oleh } \boldsymbol{x} \text{ dan } \boldsymbol{y} \quad (1)$$

Penghitungan *link* untuk semua kemungkinan pasangan n objek dilakukan dengan matriks tetangga **A**. Matriks tetangga **A** adalah matriks berukuran $n \times n$, dengan $\mathbf{A}[\mathbf{x},\mathbf{y}]$ bernilai 1 jika $\mathbf{x}$ dan $\mathbf{y}$ merupakan tetangga dan bernilai 0 jika $\mathbf{x}$ dan $\mathbf{y}$ bukan tetangga. Banyaknya *link* antar pasangan $\mathbf{x}$ dan $\mathbf{y}$ dapat diperoleh dari hasil kali antara baris ke $\mathbf{x}$ dan kolom ke $\mathbf{y}$ yang rumusnya dinyatakan sebagai

$$\text{link}(x,y) = \sum_{l=1}^{n} \mathbf{A}[x,l] * \mathbf{A}[l,y] \quad (2)$$

Apabila nilai *link* (***x***,***y***) besar, hal ini menunjukkan bahwa kemungkinan x dan y berada dalam klaster yang sama juga besar.

3. *Goodness Function*

Algoritme ROCK menggunakan informasi nilai *goodness* sebagai ukuran kemiripan antarklaster, dan menggabungkan objek/klaster yang memiliki kemiripan terbesar. Ukuran *goodness* antara klaster C_i dan C_j dituliskan sebagai

$$g(C_i,C_j) = \text{link}(C_i,C_j) / [(n_i+n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}] \quad (3)$$

dengan

$$\text{link}(C_i,C_j) = \sum_{x \in C_i,\ y \in C_j} \text{link}(x,y)$$

menyatakan banyaknya *cross link* (banyaknya *link)* dari semua kemungkinan pasangan objek yang ada dalam C_i dan C_j, n_i dan n_j masing-masing menyatakan jumlah anggota klaster i dan banyaknya anggota klaster j, dan $f(\theta) = 1 - \theta / 1 + \theta$.

Langkah-langkah dalam algoritme ROCK diuraikan sebagai berikut.

1. Menentukan inisialisasi untuk masing-masing data poin sebagai klaster pada awalnya.
2. Menghitung *similaritas* antar klaster dengan klaster lainnya, menggunakan *jaccard coefficient*.
3. Menentukan nilai matriks tetangga A menggunakan nilai nilai ambang (θ). **A**[***x***,***y***] bernilai 1 jika sim(***x***,***y***) ≥ θ dan bernilai 0 jika sim (***x***,***y***) ≤ θ.
4. Menghitung *link* antar klaster dengan klaster lainnya. Link(T_i, T_j) antar objek diperoleh dari jumlah tetangga antara T_i dan T_j
5. Menghitung nilai *goodness measure* untuk setiap klaster dengan klaster lainnya jika link !=0 yang disebut *local heap*.
6. Memilih nilai maksimum *goodness measure* antarkolom di baris ke *i* yang disebut *global heap.*
7. Mengulangi langkah 5 dan 6 hingga mendapatkan nilai maksimum di *global heap* dan *local heap.*
8. Selama ukuran data>*k*, dengan *k* adalah banyaknya kelas yang ditentukan, dapat dilakukan penggabungan klaster yang memiliki nilai *local heap* terbesar menjadi satu klaster, menambahkan *link* antar klaster yang digabungkan, menghapus klaster yang digabungkan dari *local heap* dan *update* nilai *global heap* dengan nilai hasil penggabungan.
9. Melakukan langkah 8 hingga menemukan banyak klaster yang diharapkan atau tidak ada lagi *link* antara klaster-klasternya.

Evaluasi Klaster

Evaluasi klaster adalah kemampuan untuk mendeteksi ada atau tidaknya suatu struktur tidak acak dalam data. Berikut adalah beberapa aspek penting dalam evaluasi klaster menurut Tan *et al.* (2014).

1. Menentukan kecenderungan klaster dari suatu data.
2. Menentukan jumlah klaster yang tepat.
3. Mengevaluasi seberapa baik hasil analisis klaster tanpa diberikan informasi eksternal.
4. Membandingkan hasil analisis klaster terhadap hasil eksternal yang diketahui, misalnya label kelas eksternal.
5. Membandingkan dua himpunan klaster untuk menentukan klaster yang lebih baik.

Teknik evaluasi klaster dapat digolongkan dalam tiga jenis. Ketiga jenis tersebut diuraikan sebagai berikut.

1. *Unsupervised*

Teknik *unsupervised* mengukur *goodness* dari struktur klaster tanpa informasi eksternal. Ukuran yang digunakan dalam teknik *unsupervised* dibagi menjadi

dua, yaitu *cohesion* dan *separation*. *Cohesion* merupakan ukuran kebaikan klaster yang menentukan seberapa dekat objek-objek dalam klaster. *Separation* merupakan ukuran kebaikan klaster yang menentukan perbedaan atau seberapa jauh suatu klaster dengan klaster lainnya.

2. *Supervised*

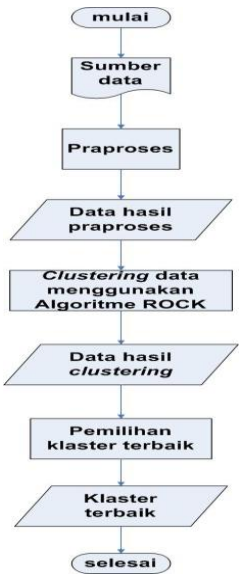
Teknik *supervised* mengukur kecocokan struktur hasil pembentukan klaster dengan struktur eksternal.

3. *Relative*

Teknik *relative* membandingkan klaster yang berbeda. Ukuran evaluasi klaster *relative* merupakan teknik *unsupervised* dan *supervised* yang digunakan untuk perbandingan.

Pada aspek evaluasi klaster kesatu, kedua, dan ketiga termasuk teknik *unsupervised* yang tidak memerlukan informasi eksternal, sedangkan aspek keempat termasuk teknik *supervised* yang memerlukan informasi eksternal. Aspek kelima dapat dilakukan menggunakan teknik *unsupervised* dan *supervised*.

METODE PENELITIAN

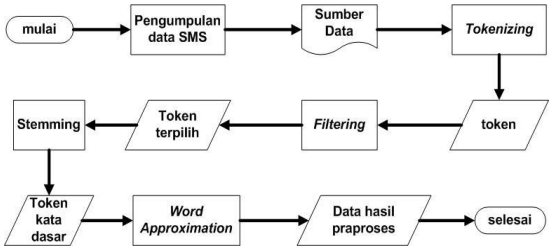


Gambar 1. Metode Penelitan

Terdapat 2 subtahap dari tahapan pembentukan klaster data, yaitu subtahap praproses dan subtahap pembentukan klaster data teks serta pemilihan klaster terbaik.

Struktur Praproses

Tahap praproses dapat dibagi lagi menjadi beberapa tahap. Rincian lengkap tahap praproses dapat dilihat pada Gambar 2.



Gambar 2. Tahapan Praproses

Tahapan praproses data yang dilakukan diuraikan sebagai berikut.

1. *Tokenizing*: Pada proses *tokenizing* dilakukan pemotongan untuk setiap kata, rangkaian angka dan rangkaian angka dengan huruf yang memiliki makna tertentu, yang terdapat dalam SMS. Karakter selain huruf, rangkaian angka, atau rangkaian angka dengan huruf akan dihilangkan. Setiap kata, rangkaian angka, maupun rangkaian angka dengan huruf disebut sebagai token.

   Ilustrasi proses *tokenizing* ditunjukkan pada uraian berikut.

   Data awal

   Tolong belikan dlu Mama pulsa 50 rb di No Mama yang baru ini 081313779293 MAMA mau pakai Nelpon pnting.**!** ini No orng

   Hasil proses *tokenizing*

   tolong belikan dlu mama pulsa nominal di no yang baru ini nomortelepon mau pakai nelpon pnting orng

2. *Filtering*: Proses *filtering* merupakan proses pembuangan token yang termasuk dalam daftar *stop word*. Beberapa kata yang termasuk dalam daftar *stop word* adalah yang, di, ke, dari, adalah, dan, atau, dan lain sebagainya.
   Ilustrasi proses *filtering* ditunjukkan pada uraian berikut.

   Data hasil proses *tokenizing* :

   tolong belikan dlu mama pulsa nominal **di** no **yang** baru ini nomortelepon mau pakai nelpon pnting orng

   Hasil proses *filtering* :

   tolong belikan dlu mama pulsa nominal no baru ini nomortelepon mau pakai nelpon pnting orng

3. *Stemming*: Pada proses *stemming* dilakukan penghapusan awalan dan akhiran yang terdapat pada setiap token yang mengandung imbuhan. Proses ini dilakukan untuk mendapatkan kata dasar setiap token.
   Ilustrasi proses *stemming* ditunjukkan pada uraian berikut.

   Data hasil proses *filtering*

   tolong **belikan** dlu mama pulsa nominal no baru ini nomortelepon mau pakai nelpon pnting orng

   Hasil proses *stemming* :

   tolong beli dlu mama pulsa nominal no baru ini nomortelepon mau pakai nelpon pnting orng

4. *Word Approximation*: Proses *word approximation* merupakan proses perbaikan *token* yang salah ketik atau token yang disingkat (Angkawattanawit *et al*. 2008). Karena *token* yang salah ketik atau token yang disingkat tidak akan memiliki makna. Padahal bisa saja *token* tersebut memiliki makna yang dapat digunakan dalam mengenali kategori suatu SMS.

   Proses *word approximation* dilakukan menggunakan algoritme Damerau Levenshtein. Ukuran jarak Damerau Levenshtein adalah jarak antara dua string yang dihitung dari jumlah minimum operasi yang diperlukan untuk mengubah satu string ke string yang lain, di mana operasi yang dimaksud didefinisikan sebagai penyisipan, penghapusan, atau penggantian satu karakter, atau penukaran dari dua karakter yang berdekatan.

   Data hasil proses *stemming* dan daftar kata dasar dalam Bahasa Indonesia digunakan sebagai masukan pada proses ini. Pada proses *word approximation* dilakukan perbandingan kata dengan memperhatikan empat jenis kesalahan pengetikan, yaitu :

   a. penyisipan sebuah huruf,
   b. penghapusan sebuah huruf,
   c. penggantian sebuah huruf dengan huruf lain, dan
   d. penukaran dua karakter yang berdekatan.

   Tahapan yang dilakukan pada proses *word approximation* dengan algoritme Damerau Levenshtein diuraikan sebagai berikut.

   a. Menghitung jarak antara token dengan setiap kata yang terdapat pada daftar kata dasar dengan menggunakan ukuran jarak Damerau Levenshtein.
   b. Mencari nilai jarak yang paling kecil.
   c. Mengganti token tersebut dengan kata dari daftar kata dasar, dimana jarak antara token dengan kata tersebut merupakan jarak paling kecil.

   Ilustrasi proses *word approximation* ditunjukkan pada uraian berikut.

Data hasil proses *stemming*

tolong beli **dlu** mama pulsa nominal **no** baru ini nomortelepon mau pakai **nelpon pnting orng**

Hasil proses *word approximation*

tolong beli dulu mama pulsa nominal nomor baru ini nomortelepon mau pakai telepon penting orang

Pembentukan Klaster Data Teks dan Pemilihan Klaster Terbaik

Proses pembentukan klaster data dilakukan untuk mendapatkan pengelompokkan seluruh data SMS yang ada. Proses ini dilakukan menggunakan data hasil tahap praproses. Pada penelitian ini dilakukan percobaan membentuk klaster menggunakan algoritme ROCK (Guha *et al*. 1999) dengan merubah masukan banyak klaster dan *threshold*. Banyaknya klaster yang digunakan adalah 2, 3, 4, dan 5. Nilai *threshold* yang digunakan antara 0.01 sampai 0.18. Selanjutnya dilakukan pemilihan klaster terbaik berdasarkan ukuran kebaikan klaster. Ukuran kebaikan klaster yang digunakan adalah *cohesion* dan *separation*. *Cohesion* adalah jarak antara suatu objek terhadap objek lain dalam klaster dan *separation* adalah jarak antara suatu objek dengan objek lain di dalam klaster yang berbeda. Semakin tinggi nilai *cohesion* dan semakin rendah nilai *separation*, semakin baik klaster tersebut terhadap klaster lainnya. Selain itu, waktu pemrosesan dan banyak klaster yang memiliki anggota SMS penipuan juga digunakan untuk memilih klaster terbaik. Pengelompokkan klaster terbaik yang akan menjadi kategori kelas dari data SMS.

HASIL DAN PEMBAHASAN

Pembentukan Klaster Data SMS

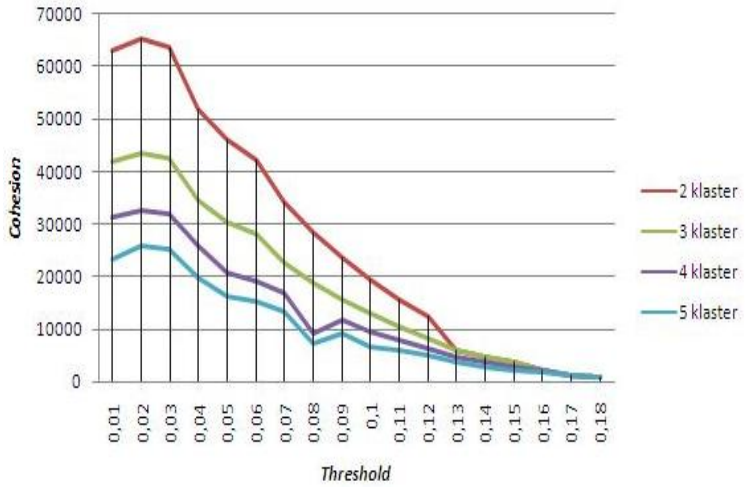
Data yang digunakan pada penelitian ini 140 data SMS. Hasil yang diperoleh dari tahap praproses dapat dilihat pada Tabel 1.

Tabel 1 Hasil Tahapan Praproses

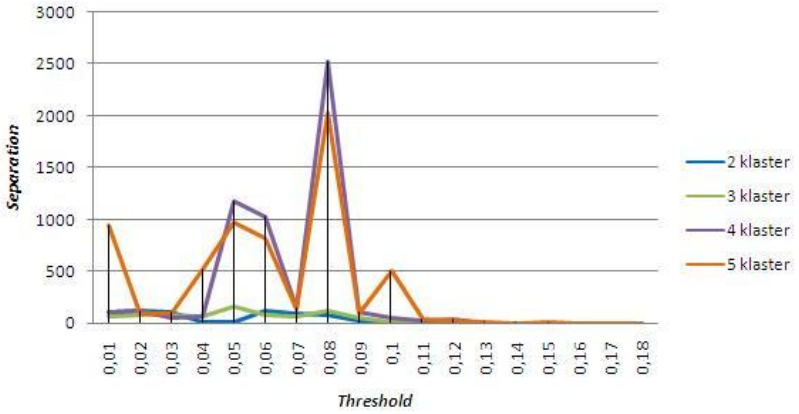
Tahap	Banyak *token* yang diperoleh
Tokenizing	2990
Filtering	2701
Stemming	2444
Word Approximation	2441

Banyak *token* pada masing-masing tahap merupakan keseluruhan jumlah token yang terdapat pada 140 data SMS. Dengan demikian terdapat duplikasi *token* dengan kata yang sama, dimana token dengan kata sama tetapi terdapat pada SMS yang berbeda akan dihitung lebih dari satu kali. Sedangkan banyak *token* jika kemunculan setiap token hanya dihitung satu kali adalah 267 *token*.

Pada Gambar 3 ditunjukkan plot *cohesion* dan *threshold* untuk banyak klaster 2, 3, 4, dan 5. Dan Gambar 4 ditunjukkan plot *separation* untuk jumlah klaster 2, 3, 4, dan 5. Dari Gambar 3 dapat dilihat bahwa jumlah klaster 2 memiliki nilai *cohesion* paling tinggi dibandingkan dengan jumlah klaster lain. Kemudian dari Gambar 4 dapat dilihat bahwa jumlah klaster 2 juga memiliki nilai *separation* paling rendah dibandingkan dengan jumlah klaster yang lain. Akan tetapi, *range* nilai yang terbentuk sangat lebar. Sedangkan jumlah klaster 5 memiliki *range* nilai *cohesion* dan *separation* yang tidak terlalu lebar tetapi cukur beragam. Untuk itu, jumlah klaster yang dipilih sebagai klaster terbaik adalah 5 klaster.

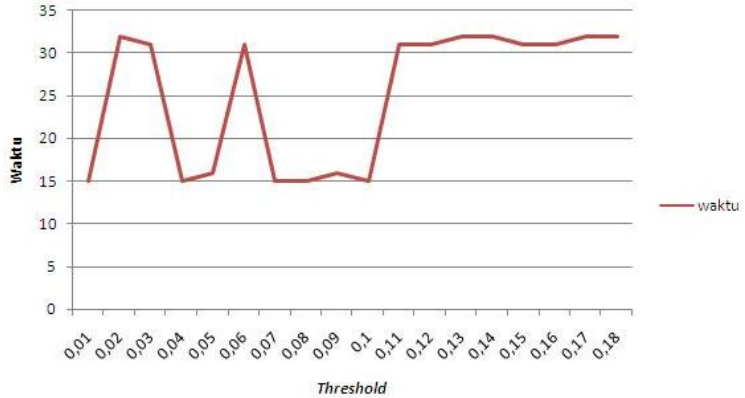


Gambar 3. Plot *Cohesion* dan threshold dengan banyak klaster 2, 3, 4, dan 5

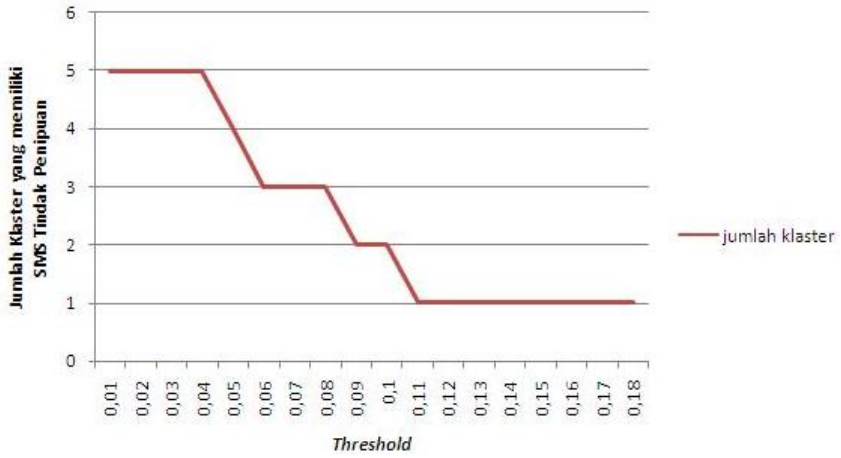


Gambar 4. Plot *Separation* dan *threshold* dengan banyak klaster 2, 3, 4, dan 5

Selanjutnya berdasarkan waktu proses pembentukan klaster dan banyak klaster yang memiliki data SMS penipuan untuk menentukan *threshold* diperoleh 5 klaster yang memberikan hasil terbaik. Pada Gambar 5 ditunjukkan waktu proses berdasarkan *threshold* dengan banyak klaster 5 dan banyak klaster yang memiliki data SMS penipuan dengan banyak klaster 5 ditunjukkan pada Gambar 6.



Gambar 5. Plot waktu dan *threshold* pada proses pembentukan 5 klaster



Gambar 6. Plot banyak klaster dengan SMS penipuan untuk 5 klaster

Berdasarkan nilai *cohesion*; *separation*; waktu proses; dan banyak klaster yang memiliki data SMS penipuan 5 klaster, diperoleh nilai *threshold* 0,08; nilai *cohesion* sebesar 7436,2; nilai *separation* 2025,37; waktu proses 15 ms, diperoleh bahwa klaster terbaik yang digunakan dalam pemberian label kelas pada *dataset*. Label kelas yang digunakan adalah c1, c2, c3, c4, dan c5. Dimana anggota klaster c1, dan c2 didominasi SMS berindikasi penipuan. Sedangkan anggota klaster c3, c4, dan c5 didominasi SMS yang tidak memiliki indikasi penipuan.

SIMPULAN

Algoritme ROCK dapat digunakan untuk mengelompokkan data SMS. Banyak klaster 5 dengan *threshold* 0.08 merupakan klaster terbaik.

DAFTAR PUSTAKA

Guha S, Rastogi R, & Shim K. 2000. ROCK*: A Robust Clustering Algorithm for Categorical Attributes*. In Proc.of the 15th Int.Conf.on Data Engineering.

Han J & Kamber M. 2012. *Data Mining Concepts and Techniques, Third Edition*. Waltham : Morgan Kaufmann Publisher.

Tan P.N, Steinbach M, & Kumar V. 2014. *Introduction to Data Mining*, Boston : Pearson Education, Inc.