ISSN: 2477-5126
e-ISSN: 2548-9356

Implementasi Pendekatan *Rule-Of-Thumb* untuk Optimasi Algoritma *K-Means Clustering*

M. Nishom[1*)], M. Yoka Fathoni[2]
[1,2]Jurusan Teknik Informatika, Politeknik Harapan Bersama, Tegal
[1,2]Jln. Mataram No.09, Margadana, Tegal, 50272, Indonesia
email : [1]nishom@poltektegal.ac.id, [2]myokafathoni@poltektegal.ac.id

Received: 30 Maret 2018; Revised:12 Mei 2018; Accepted: 13 Mei 2018

***Abstract* - In the big data era, the clustering of data or so-called clustering has attracted great interest or attention from researchers in conducting various studies, many grouping algorithms have been proposed in recent times. However, as technology evolves, data volumes continue to grow and data formats are increasingly varied, thus making massive data grouping into a huge and challenging task. To overcome this problem, various research related methods for data grouping have been done, among them is K-Means. However, this method still has some shortcomings, among them is the sensitivity issue in determining the value of cluster (K). In this paper we discuss the implementation of the rule-of-thumb approach and the normalization of data on the K-Means method to determine the number of clusters or K values dynamically in the data groupings. The results show that the implementation of the approach has a significant impact (related to time, number of iterations, and no outliers) in the data grouping.**

***Abstrak* – Di era big data, pengelompokan data atau biasa disebut *clustering* telah menarik minat atau perhatian yang sangat besar dari para peneliti dalam melakukan berbagai penelitian, banyak algoritma pengelompokan telah diajukan dalam beberapa kurun waktu terakhir. Namun, seiring berkembangnya teknologi, *volume* data terus bertambah dan format data semakin bervariasi (*variety*), sehingga membuat pengelompokan data dengan skala yang sangat besar (*big data*) menjadi sebuah tugas yang sangat besar dan menantang. Untuk mengatasi masalah ini, berbagai penelitian terkait metode untuk pengelompokan data telah dilakukan, diantaranya adalah K-Means. Namun metode ini masih memiliki beberapa kekurangan, diantaranya adalah masalah sensitifitas dalam menentukan nilai *cluster* (*K*). Dalam paper ini kami membahas tentang implementasi pendekatan *rule-of-thumb* dan normalisasi data pada metode *K-Means* untuk menentukan jumlah dari *cluster* atau nilai *K* secara dinamis dalam pengelompokan data. Hasil penelitian menunjukkan bahwa implementasi pendekatan tersebut memiliki dampak yang signifikan (terkait waktu, banyaknya iterasi, serta tidak ada outlier) dalam pengelompokan data.**

***Kata Kunci* – rule-of-thumb, k-means, big data, clustering.**

*) **Corresponding author:** M. Nishom
Email: nishom@poltektegal.ac.id

I. PENDAHULUAN

Seiring dengan kemajuan teknologi, *volume* informasi menjadi semakin besar (*velocity*), bentuk dan format data semakin bervariasi (*variety*), sehingga berdampak pada proses pengelompokan data yang berukuran sangat besar menjadi pekerjaan yang sangat menantang. Salah satu bentuk analisis data yang digunakan untuk mengelompokkan data adalah algoritma klasifikasi dan *clustering* [1]. Dalam rangka untuk menangani masalah ini, banyak peneliti telah mencoba untuk merancang algoritma *clustering* yang efisien. *Clustering* adalah algoritma yang paling sering digunakan dalam bidang penggalian data (*data mining*). Algoritma tersebut dapat membagi data dan mengelompokkan data tersebut ke dalam jenis data yang sama serta jenis data yang memiki kesamaan rendah dalam *cluster*.

Tetapi, dalam melakukan hal tersebut sulit untuk menentukan jumlah *cluster* yang tepat dan selalu membutuhkan kompleksitas waktu yang tinggi. Algoritma *K-means* dengan jumlah *cluster* yang disesuaikan (dinamis), yang secara otomatis dapat menentukan jumlah *cluster* yang optimal memiliki efek yang dan akurasi yang lebih baik. Selain itu, dalam implementasinya, algoritma ini memiliki beberapa kelemahan seperti: seringnya terjadi kekosongan *cluster*, adanya *outlier*, nilai *Sum of Squared Errors (SSE)* yang relatif besar, dan tingkat akurasi yang rendah[2].

Beberapa permasalahan tersebut dapat disebabkan oleh penentuan jumlah *cluster* yang tidak tepat, sehingga perlu adanya implementasi sebuah pendekatan baru sebagai solusi dalam menentukan jumlah *cluster* yang dinamis.

II. PENELITIAN YANG TERKAIT

Pendekatan baru untuk mengoptimalkan pemilihan centroid awal pada metode *K-Means Clustering*. Pendekatan ini perlu dilakukan karena kinerja *K-Means Clustering* sangat bergantung pada kebenaran dari centroid awal. Biasanya centroid awal ditentukan secara acak sehingga centroid diharapakan dapat mencapai minimum lokal terdekat, bukan optimum global. Pada penelitian ini digunakan pendekatan baru yang digunakan untuk menentukan satu set lokasi pilar untuk membuat struktur *centroid* yang lebih stabil. Penentuan posisi centroid awal dilakukan dengan menggunakan akumulasi jarak terjauh antara centroid satu dengan *centroid* lain. Pertama, membuat matrik akumulasi jarak antara semua titik data dan rata-rata terbesar. Selanjutnya memilih *centroid* awal dengan cara memilih matrik maksimum akumulasi jarak dari dari titik data. *Centroid* awal berikutnya dipilih dengan memodifikasi matriks akumulasi jarak antara setiap titik data dan semua centroid awal sebelumnya, kemudian memilih titik data yang memiliki jarak maksimum sebagai *centroid* awal baru. Proses berulang ini diperlukan agar semua centroid awal ditunjuk. Pendekatan ini juga memiliki mekanisme untuk menghindari data yang *outlier* yang dipilih sebagai *centroid* awal. Dalam penelitian ini dilakukan 3 jenis analisis untuk pengukuran validitas yaitu; (1) *Variance Analysis*, (2) *Sum of Squared Error*, (3) *Standart Deviation Validity Index*. Hasil percobaan menunjukan efektivitas algoritma yang diusulkan untuk meningkatkan hasil pengomtimalan *K-Means Clustering* [3].

Dalam upaya meningkatkan stabilitas dan tingkat akurasi dari algoritma *K-Means* juga telah dilakukan berbagai penelitian seperti dengan cara mencari nilai *K* yang tepat dengan menentukan *initial seeds* berbasis kerapatan kanopi [4]. Menggabungkan dua metode (yaitu overlapping dan k-harmonic means) untuk menangani kekurangan sensitifitas penentuan centroid awal[5]. Mendeteksi dan menghilangkan atau menghapus outlier pada saat proses klastering [6]. Implementasi pendekatan rekursif dan melakukan pembagian atau partisi terhadap sub-himpunan data, menekan perhitungan jumlah jarak dan meningkatkan kualitas aproksimasi[7]. Implementasi algoritma *K-Means Clustering* terdistribusi berdasarkan metode analisis set pair (SPAB-DKCM) .

Penelitian ini diprogramkan dengan menggunakan model *MapReduce* dan bahasa pemrograman java. Platform atau peron yang digunakan adalah klaster *Hadoop* yang menggunakan enam set sistem operasi Linux (ubuntu versi 12.04), dan JDK versi 1.6.0_21. Percobaan dilakukan menggunakan data set extended iris dan data *set wine* pada UCI. Untuk menggantikan jarak *Euclidean*, digunakan teori analisis *set pair* untuk menghitung kesamaan antara data sampel dan proses iterasi. Algoritma ini lebih baik karena dapat beradaptasi dengan perbandingan kesamaan data multidimensi, dan proses perbandingan kesamaan lebih dekat dengan situasi aktual. Analisis teoritis dan hasil eksperimen menunjukkan bahwa algoritma ini dapat mengurangi jumlah iterasi dari algoritma, meningkatkan efisiensi algoritma, dan memberikan metode yang sederhana dan layak untuk perhitungan kesamaan data multidimensi [8].

Penelitian dalam upaya peningkatan performa algoritma K-Means juga telah dilakukan dengan merancang algoritma pengelompokan paralel berbasis *MapReduce*. Algoritma pengelompokkan paralel sangat cocok untuk sistem yang terdistribusi dengan jaringan interkoneksi yang handal. Namun, dalam sistem terdistribusi skala besar algoritma paralel menjadi tidak berguna saat terjadi kegagalan komunikasi tunggal atau *latency* yang tinggi di jalur komunikasi [9].

Pendekatan *MapReduce K-Means* (*MRK-means*) digunakan untuk solusi optimalisasi teknik *clustering K-means* agar proses eksekusi menjadi lebih optimal. Berbeda dengan implementasi k-means berbasis *MapReduce* pada umumnya, MRK-means hanya membaca dataset sekali dan karena itu MRK-means dapat menjadikan proses klastering menjadi lebih cepat. Kompleksitas waktu dari MRK-means adalah linier lebih rendah daripada iterasi k-means. Karena penggunaan algoritma k-means++, MRK-means menghasilkan cluster dengan kualitas yang lebih tinggi. Secara teoritis, hasil MRK-means O(log 2 k) – kompetitif untuk pengelompokan yang optimal di kasus yang buruk, mengingat k sebagai jumlah dari klaster. Untuk evaluasi performa dari MRK-means, analisis secara kompleks (kompleksitas waktu, I/O, dan ruang memori) dikalkukan dengan menggunakan kumpulan data yang telah disintesis dan data asli yang diambil dari repositori UCI *machine learning* [10].

Penelitian terkait sensitifitas dalam menentukan jumlah *cluster* (K) pada algoritma *K-Means* juga telah dilakukan. Pada tahapan awal, algoritma *K-Means* dijalankan seperti tahapan pada umumnya, kemudian proses dilanjutkan dengan melakukan perhitungan di dalam dan diantara klaster, selanjutnya jika dalam proses tersebut ditemukan *distance* atau jarak dari intra adalah lebih kecil dan jarak inter lebih besar, maka algoritma tersebut akan melakukan perhitungan kelompok baru dengan cara menambah nilai *k* dengan nilai satu($k = k + 1$) pada setiap perulangan atau iterasi sampai memenuhi batas validitas klaster [11].

III. METODE PENELITIAN

A. Dataset

Dataset yang digunakan sebagai data uji adalah data kependudukan dan data kesehatan provinsi Jawa Tengah tahun 2018 yang diunduh dari situs resmi BPS (Badan Pusat Statistik) yang meliputi data jumlah penduduk, jumlah penderita diare, jumlah penderita demam berdarah, jumlah penderita malaria, jumlah penderita HIV, dan data jumlah penderita Aids. Dalam penelitian ini, data diproses diolah dengan aplikasi yang dikembangkan dengan bahasa pemrograman Java, sedangkan untuk *Database Management System* (DBMS) yang digunakan adalah MySQL.

ISSN: 2477-5126
e-ISSN: 2548-9356

B. Clustering

Pengelompokan data atau biasa disebut *clustering* merupakan suatu cara atau metode yang biasa digunakan untuk mengklaster atau mengelompokan kumpulan suatu data ke dalam sebuah himpunan atau klaster, mencari dan mengelompokkan data yang memiliki kesamaan atau kemiripan dalam sifat atau ciri-ciri (*similarity*) antara data satu dengan data yang lainnya dalam sebuah dataset. Sifat dari metode ini adalah *unsupervised* atau tanpa arahan (artinya metode ini diimplementasikan tanpa adanya *training* atau latihan dan tanpa ada *teacher*) serta tidak memerlukan target atau sasaran luaran. Tujuan algoritma klaster adalah menciptakan klaster yang koheren secara internal, tetapi jelas berbeda satu sama lain. Dengan kata lain, data dalam sebuah klaster harus semirip mungkin dan data dalam satu klaster harus sebeda mungkin dari data dalam klaster lainnya [12].

C. K-Means Clustering

Algoritma *K*-means merupakan suatu algoritma yang biasa digunakan untuk menemukan kelompok dari dokumen atau objek yang *non-overlapping* [13]. Algoritma klaster K-Means juga disebut sebagai suatu algoritma yang sangat efektif untuk mengelompokkan kumpulan data [14]. Algoritma K-Means clustering juga sangat sederhana untuk diimplementasikan dan dijalankan, proses clustering relatif cepat, mudah beradaptasi serta umum penggunaanya dalam paraktek [15]. Proses algoritma ini ide atau alurnya cukup sederhana. Di tahap awal, terlebih dahulu ditentukan jumlah kelompok atau *cluster* yang akan akan dipakai. Kemdian dilanjutkan dengan memilih dokumen pertama atau elemen pertama dalam sebuah *cluster* untuk digunakan sebagai *centroid point cluster* (titik tengah klaster). Selanjutnya, dilakukan iterasi atau pengulangan langkah-langkah dalam menentukan jarak dokumen atau objek dengan centroid, sampai terjadi ke-stabilan (semua kelompok objek telah konvergen) [16].

Adapun tahapan dalam algoritma *K-Means* dengan implementasi pendekatan *rule-of-thumb* ditunjukkan pada Gambar 1. Pertama, sebelum masuk ke proses *clustering*, diperlukan analisis terhadap tipe data yang ada, apakah data tersebut perlu untuk dinormalisasi atau tidak. Misalnya, terkait pencatatan tingkat atau jumlah kematian dari data penduduk di negara Indonesia pada setiap bulan yang didasarkan pada jenis usia atau umur. Biasanya, akan terdapat tiga dimensi atau kelompok data, yaitu jumlah kematian 0-jutaan, penduduk usia 1 sampai 12 bulan, dan umur 0 sampai 100 tahun. Jika jarak dari masing-masing dimensi dibentangkan, maka akan terjadi ketidak-seimbangan pada dimensi ketiga (jumlah kematian). Dalam kasus ini, maka diperlukan normalisasi data sebelum proses klastering. Normalisasi yang dapat digunakan adalah normalisasi *Min-Max*. Cara ini diimplementasikan dengan merubah nilai atau data asli ke bentuk *linier* dilakukan dengan menggunakan persamaan (1).

$$x' = \frac{x - nilai_{min}}{nilai_{max} - nilai_{min}} \quad (1)$$

dimana,
x = data per kolom
$nilai_{min}$ = nilai terkecil dari data per kolom
$nilai_{max}$ = nilai terbesar dari data perkolom

Kedua, menentukan jumlah *cluster* (*K*). Proses ini menginisialisasi nilai awal K sebagai jumlah *cluster* yang akan dipartisi secara dinamis. Dalam menentukan jumlah K digunakan pendekatan *rule-of-thumb* dengan menggunakan persamaan (2) sebagai berikut:

$$k = \sqrt{\frac{n}{2}} \quad (2)$$

dimana,
n = jumlah objek yang akan di kelompokkan
k = jumlah *cluster*

Ketiga, menentukan *centroid* awal *(initial centroid)*. Banyak metode yang dapat digunakan, seperti dengan metode *random* (mengambil secara acak). Sedangkan untuk menentukan *centroid* baru, dapat dilakukan dengan menghitung nilai rata-rata dari total nilai objek dalam *cluster* baru. Rumus yang digunakan adalah persamaan (3) sebagai berikut:

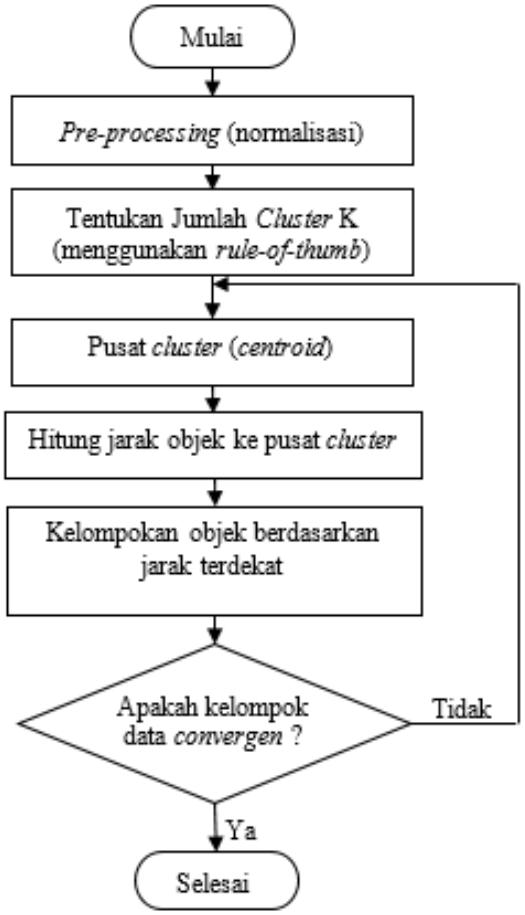
$$C_i = \frac{\sum_{i=1}^{n} x_i \in s_i}{n} \quad (3)$$

dimana,
C_i = *centroid* baru ke *i*
s_i = objek ke *i*
x_i = nilai pada objek ke *i*
n = jumlah data pada tiap kelompok

Keempat, menghitung jarak objek dengan *centroid*. Untuk menghitung jarak antara objek dengan pusat *cluster* (*centroid*) dapat dilakukan dengan menggunakan beberapa pendekatan. Pada penelitian ini digunakan rumus *Euclidean Distance*. Perhitungan ini menghitung nilai kuantitatif dari tingkat kemiripan atau ketidakmiripan data (*proximity measure*) yang dapat menghasilkan jarak dari objek dengan pusat *cluster* (*centroid*). Berikut merupakan rumus *Euclidean Distance* yang digunakan untuk menghitung jarak objek dengan pusat *cluster*:

$$d(x, y) = |x - y| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (4)$$

dimana,
d = jarak antara *x dan* y
x = data pusat klaster
y = data pada atribut
i = setiap data
n = jumlah data,
x_i = data pada pusat klaster ke *i*
y_i = data pada setiap data ke *i*



Gbr. 1 Flowchart Algoritma *K-Means*

Kelima, pengelompokkan objek berdasarkan jarak terdekat (jarak terkecil dari semua jarak objek dengan *centroid*). Sebelum pengelompokan objek dilakukan, pertama harus dilakukan perhitungan untuk menentukan jarak (*distance*) yang bernilai minimum. Setelah didapatkan nilai minimum, dilakukan pengelompokkan objek. Tahapan akhir adalah melakukan uji konvergensi antara kelompok data baru dan kelompok data pada proses sebelumnya, jika kelompok data yang baru adalah sama dengan kelompok data sebelumnya (*convergen*), maka proses *clustering* selesai. Jika tidak, maka lakukan iterasi dimulai dari penentuan pusat klaster baru.

IV. HASIL DAN PEMBAHASAN

Dalam penelitian ini, pengujian untuk mengetahui perbedaan hasil dari sisi efektivitas iterasi, lamanya waktu eksekusi, dan jumlah outlier antara penggunaan metode K-Means pada umumnya dan metode K-Means telah dilakukan dengan mengimplementasikan pendekatan *rule-of-thumb* untuk menentukan jumlah *cluster* secara dinamis. Dari hasil perhitungan menggunakan pendekatan *rule-of-thumb*, jumlah *cluster* yang dihasilkan adalah 4. Pada penelitian ini telah dilakukan pengujian dengan jumlah *cluster* seperti terlihat pada Tabel I.

TABEL I
JUMLAH *CLUSTER* YANG AKAN DIUJI

Jumlah *Cluster*	**Pendekatan**
3	Manual
4	*Rule-of-thumb*
5	Manual
6	Manual
7	Manual

TABEL II
HASIL PENGUJIAN

Jumlah *Cluster*	**(N) Uji**	**Pende-katan**	**Iterasi (Rata-Rata)**	**Waktu (Rata-Rata)**	**Outlier**
3	100 kali	Manual	4	0.28821	Tidak ada
4	100 kali	*Rule-of-thumb*	**4**	**0.28338**	Tidak ada
5	100 kali	Manual	5	0.33281	Tidak ada
6	100 kali	Manual	5	0.34432	Tidak ada
7	100 kali	Manual	4	0.31131	Tidak ada

Pengujian dilakukan sebanyak 100 kali untuk menemukan hasil yang optimal. Dari hasil pengujian didapatkan bahwa penggunaan pendekatan *rule-of-thumb* terbukti dapat mengurangi jumlah iterasi dan waktu yang diperlukan selama proses *clustering* relatif lebih cepat. Selain itu, di sana juga tidak ditemukan outlier. Hasil pengujian secara detail dapat dilihat pada Tabel II.

V. KESIMPULAN

Penentuan jumlah *cluster* pada metode *K-Means* dengan menggunakan pendekatan *rule-of-thumb* telah berhasil diterapkan dan memiliki dampak yang signifikan dalam proses *clustering*. Hasil penelitian menunjukkan bahwa jumlah iterasi

selama proses *clustering* menjadi lebih sedikit sehingga berdampak pada lebih cepatnya waktu yang dibutuhkan dalam proses pengelompokkan data, yaitu rata-rata 0.28338 detik untuk setiap posesnya. Saran yang dapat diberikan untuk penelitian selanjutnya adalah dalam proses menentukan titik pusat (*centroid*) seharusnya dilakukan dengan metode tertentu (bukan cara random) untuk meningkatkan efektifitas dan akurasi.

DAFTAR PUSTAKA

[1] F. Fanny, Y. Muliono, and F. Tanzil, "A Review of News Classification using k-NN, Naive Bayes and Support Vector Machine Classifiers," *J. Pengemb. IT*, vol. 3, pp. 55–60, 2018.

[2] K. Singh, D. Malik, and N. Sharma, "Evolving limitations in K-means algorithm in data mining and their removal," vol. 12, no. April, pp. 105–109, 2011.

[3] A. R. B. and Y. Kiyoki, "A Pillar Algorithm for K-Means Optimization by Distance Maximization for Initial Centroid Designation," *Comput. Intell. Data Min.*, pp. 61–88, 2009.

[4] G. Zhang, C. Zhang, and H. Zhang, "Improved K-means Algorithm Based on Density Canopy," *Knowledge-Based Syst.*, 2018.

[5] S. Khanmohammadi, N. Adibeig, and S. Shanehbandy, "An improved overlapping k-means clustering method for medical applications," *Expert Syst. Appl.*, vol. 67, pp. 12–18, 2017.

[6] G. Gan and M. K. P. Ng, "K-Means Clustering With Outlier Removal," *Pattern Recognit. Lett.*, vol. 90, pp. 8–14, 2017.

[7] M. Capó, A. Pérez, and J. A. Lozano, "An efficient approximation to the K-means clustering for massive data," *Knowledge-Based Syst.*, vol. 117, pp. 56–69, 2017.

[8] S. L. and Q. Yunfeng, "Optimization of the Distributed K-means Clustering Algorithm Based on Set Pair Analysis," *Int. Congr. Image Signal Process.*, pp. 1593–1598, 2015.

[9] M. O. S. and E. Torunski, "A Parallel K-Medoids Algorithm for Clustering based on MapReduce," *Int. Conf. Mach. Learn. Appl.*, pp. 502–507, 2016.

[10] S. Shahrivari and S. Jalili, "Single-pass and linear-time k-means clustering based on MapReduce," *Inf. Syst.*, vol. 60, pp. 1–12, 2016.

[11] Widiarini and R. Satria Wahonono, "Algoritma Cluster Dinamik Untuk Optimasi Cluster Pada Algoritma K-Means Dalam Pemetaan Nasabah Potensial Algoritma Cluster Dinamik Untuk Optimasi Cluster Pada Algoritma K-Means Dalam," *J. Intell. Syst.*, vol. 1, no. 1, pp. 32–35, 2015.

[12] C. D. Manning, *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.

[13] X. Wu, B. Wu, J. Sun, S. Qiu, and X. Li, "A hybrid fuzzy K-harmonic means clustering algorithm," *Appl. Math. Model.*, no. November, 2014.

[14] D. T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*. Canada: Wiley-Interscience, 2014.

[15] E. Prasetyo, *Data Mining, Mengolah Data Menjadi Informasi Menggunakan Matlab*. Yogyakarta: ANDI, 2014.

[16] J. Oyelade, O. Oladipupo, and I. C. Obagbuwa, "Application of k Means Clustering algorithm for prediction of Students Academic Performance," *Int. J. Comput. Sci. Inf. Secur.*, vol. 7, no. 1, pp. 292–295, 2010.