PERBAIKAN INISIALISASI K-MEANS MENGGUNAKAN GRAF HUTAN YANG MINIMUM

***Achmad Maududie*[1]**
***Wahyu Catur Wibowo*[2]**

[1]*Program Studi Sistem Informasi, Universitas Jember*
[2]*Fakultas Ilmu Komputer, Universitas Indonesia,*
[1]*maududie@unej.ac.id,* [2]*wibowo@cs.ui.ac.id*

Abstrak

K-Means adalah salah satu algoritma clustering yang sangat popular karena kesederhanaan dan kemampuannya dalam menangani data dengan skala besar. Namun demikian algoritma ini sangat sensitif terhadap centroid awal. Perbedaan centroid awal akan memberikan perbedaan hasil clustering dan apabila centroid awal yang diberikan adalah centroid yang tidak baik maka dapat dipastikan hasil clusteringnya juga tidak baik. Artikel ini memuat sebuah metode baru yang dikembangkan penulis untuk meningkatkan kualitas centroid awal melalui teknik perbaikan k yang didasarkan pada graf hutan yang minimum (minimum forest graf). Hasil percobaan yang telah dilakukan menunjukkan bahwa metode inisialisasi menggunakan graf hutan yang minimum menghasilkan centroid awal yang lebih baik dan konsisten dibandingkan metode Forgy. Disamping itu jumlah perulangan yang harus dilakukan dalam proses clustering dengan menggunakan metode ini adalah lebih sedikit (rerata 3,2) dibandingkan metode Forgy (rerata 6,4).

Kata Kunci: *Clustering, Graf hutan yang minimum, K-Means*

PENDAHULUAN

Algoritma K-Means merupakan salah satu algoritma populer yang digunakan dalam proses clustering dataset (Celebi, Kingravi, & Vela, 2013; Rokach, 2010, p. 280) karena kesederhanaan algoritmanya (Celebi et al., 2013; Feldman & Sanger, 2007; Jain, Murty, & Flynn, 1999, p. 278; Liao, Liu, Xiao, & Liu, 2013; Rokach, 2010, p. 280). Selain mampu menangani dataset dengan ukur yang besar (Jain et al., 1999, p. 278; Khan & Ahmad, 2013, p. 7444; Reddy & Jana, 2012, p. 395; Rokach, 2010, p. 281), algoritma *partitional clustering* ini juga memiliki tiga keunggulan, yaitu (Jain et al., 1999; Liao et al., 2013, p. 124; Rokach, 2010): kompleksitas waktu, kompleksitas ruang penyimpanan, dan pemrosesannya tidak bergantung pada urutan centroid (titik pusat klaster) yang digunakan. Disamping itu, menurut (Celebi et al., 2013), algoritma K-Means juga bersifat *versatile*, yaitu mudah melakukan modifikasi di setiap tahapan (aspek) dalam algoritma tersebut (misalnya inisialisasi, fungsi perhitungan jaraknya, dan kriteria penghentian iterasi).

Meskipun memiliki sejumlah kelebihan, algoritma K-Means juga memiliki kekurangan, salah satunya adalah sangat sensitif terhadap centroid

awal (Celebi et al., 2013, p. 201; Feldman & Sanger, 2007, p. 86; Reddy & Jana, 2012, p. 395; Rokach, 2010, p. 310; Wu et al., 2007, p. 6), termasuk penentuan nilai *k* (Celebi et al., 2013, p. 201; Jain et al., 1999, p. 278; Reddy & Jana, 2012, p. 395). Inisialisasi centroid ini secara langsung dapat mempengaruhi performa algoritma K-Means secara signifikan (Celebi et al., 2013, p. 201; Rokach, 2010), misalnya munculnya cluster dengan member kosong, konvergensi yang sangat lama, serta besar kemungkinannya berujung pada *bad local minima*. Namun demikian, sifat K-Means yang *versatile* dapat digunakan untuk mengatasi permasalahan yang muncul dalam algoritma ini yaitu melalui *adaptive initialization method* (IM) (Celebi et al., 2013, p. 201).

Dalam makalah ini dijelaskan sebuah metode inisialisasi yang dikembangkan penulis untuk mendapatkan centroid awal yang baik. Metode ini diawali dengan membangkitkan *n* subset data dan kemudian dalam masing-masing subset tersebut dilakukan proses clustering. Penentuan centroid awal untuk melakukan clustering di masing-masing subset didasarkan pada metode *single linkage* untuk membentuk graf hutan (*forest*). Untuk menguji keefektifan metode ini, penulis membandingkan dengan algoritma K-Means yang menggunakan metode inisialisasi Forgy.

METODE PENELITIAN

Secara garis besar algoritma metode inisialisasi untuk menentukan centroid awal yang dikembangkan adalah sebagai berikut.

Algoritma Penentuan Centroid Awal

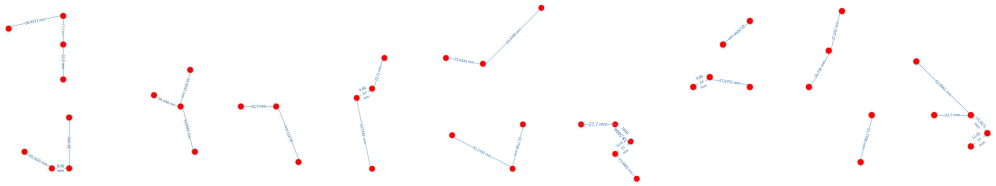
1) Bangkitkan *n* subset data
2) Hitung tingkat kemiripan antar data pointnya (d_{ij}) dalam setiap subset data menggunakan metode *cosine similarity*.
3) Kelompokkan semua data point (x_i) dalam setiap subset berdasarkan tingkat kemiripan yang paling tinggi untuk membentuk graf hutan yang minimum.
4) Bangkitkan centroid untuk setiap komponen graf hutan yang terbentuk (ca_{nk}) dalam setiap subset data yang disebut dengan centroid antara (Ca)
5) Bangkitkan dataset baru yang semua data pointnya adalah semua centroid antara (Ca)
6) Hitung tingkat kemiripan antar data point (d_{ij}) dari dataset baru
7) Kelompokkan semua data point (x_i) dari dataset baru berdasarkan tingkat kemiripan yang paling tinggi hingga membentuk graf hutan yang minimum.
8) Bangkitkan centroid untuk setiap komponen graf hutan yang terbentuk (c_i) dari dataset baru yang merupakan centroid sesungguhnya (disebut "true" centroid).

Dalam algoritma di atas disebutkan bahwa langkah awal adalah membangkitkan *n* subset data. Setiap subset berjumlah *m* data point yang dipilih secara acak dari dataset asalnya. Besarnya nilai *m* adalah 10% dari jumlah data point dalam dataset. Apabila 10% dari jumlah data point tersebut kurang dari empat kali jumlah klaster (*k*), maka nilai *m* adalah empat kali jumlah klaster (4*k*).

Setelah *n* subset data terbentuk, langkah berikutnya adalah melakukan perhitungan tingkat kemiripan antar elemen dalam masing-masing subset menggunakan metode *cosine similarity*.

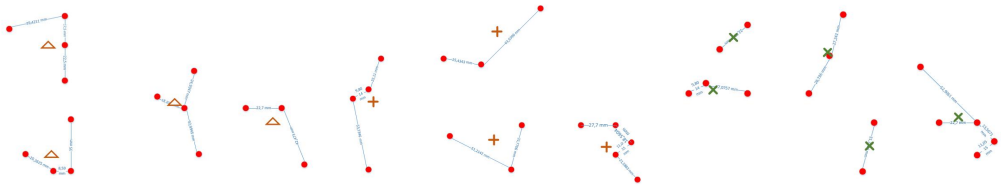
$$Cos(D_1, D_2) = \frac{\sum_{i=1}^{n} d_{1i} d_{2i}}{\sqrt{\sum_{i=1}^{n} d_{1i}^{\,2}} \times \sqrt{\sum_{i=1}^{n} d_{2i}^{\,2}}}$$

Nilai tingkat kemiripan antar data point digunakan untuk membangun graf hutan yang minimum. Mekanisme membangun graf hutan yang minimum adalah dengan menggunakan data point sebagai simpul dan kemudian membuat sisi dari sebuah simpul ke simpul lain yang memiliki tingkat kemiripan tertinggi. Dalam tahapan ini sangat mungkin sebuah subset memiliki jumlah komponen graf hutan yang berbeda dengan subset lainnya.



a. Graf hutan subset 1 b. Graf hutan subset 2 c. Graf utan subset 3

Gambar 1: Contoh graf hutan yang minimum



a. Centroid antara subset 1 b. Centroid antara subset 2 c. Centroid antara subset 3

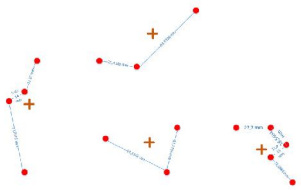
Gambar 2: Contoh centroid antara

Langkah berikutnya adalah menggabungkan seluruh centroid antara dari semua subset untuk membentuk sebuah set data baru dan selanjutnya menghitung tingkat kemiripan antar data pointnya menggunakan metode *cosine similarity*.

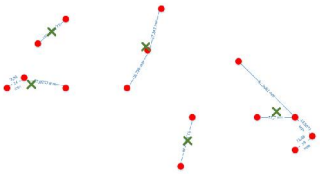
Mengacu pada nilai tingkat kemiripan tersebut, proses berikutnya adalah membangkitkan "*true*" centroid dengan cara membentuk graf hutan yang minimum dari semua centroid antara. Setiap simpul dalam komponen graf hutan tersebut adalah kandidat "*true*" centroid. Dalam tahap ini, jumlah kandidat akan disesuaikan dengan jumlah *k* yang telah ditentukan. Prosedur penyesuaian jumlah kadidat didasarkan pada nilai tingkat kemiripan yang paling besar antara dua simpul. Kedua simpul tersebut digabung untuk membentuk sebuah simpul baru berdasarkan reratanya. Prosedur ini diulang hingga jumlah kandidat sama dengan jumlak *k* yang diinginkan. Gambar 4 memperlihatkan contoh penggabungan simpul berdasarkan tingkat kemiripan yang dimaksud.

a. Centroid antara 1 d. Graf hutan yang minimum hasil penggabungan centroid

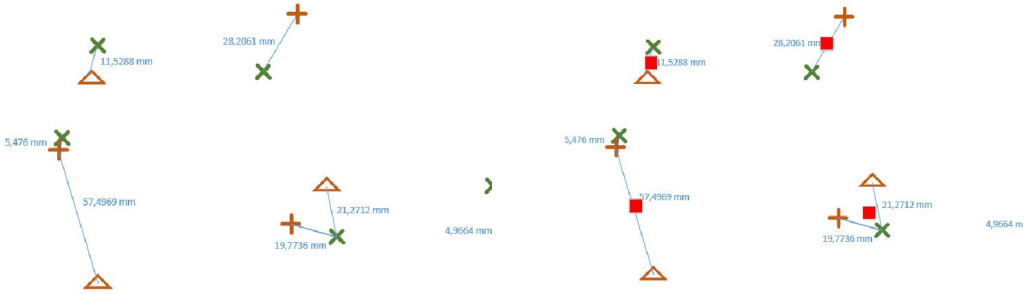
b. Centroid antara 2



c. Centroid antara 3

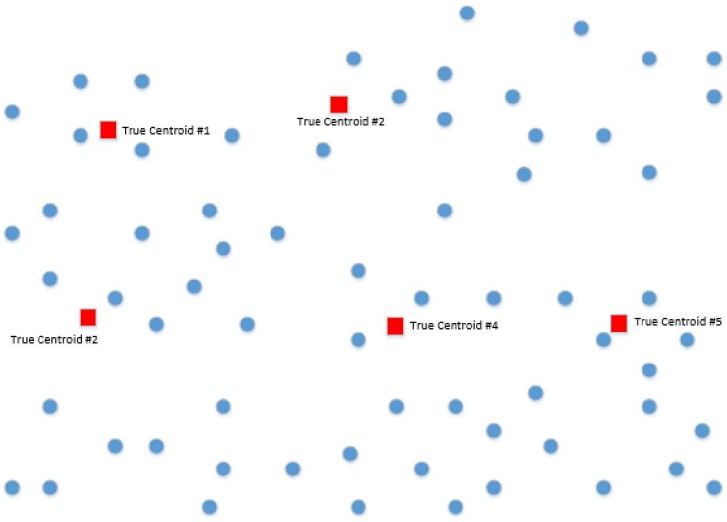


Gambar 3. Contoh hasil pembangkitan dataset baru



a. Centroid kandidat

b. Pembangkitan "true" centroid dengan $k = 5$



c. True centroid dengan dataset asli

Gambar 4: Contoh hasil pembangkitan dataset baru

True centroid yang dibangkitkan pada tahapan ini merupakan luaran dari metode inisialisasi yang digunakan sebagai centroid awal dalam algoritma K-Means. Dengan demikian, proses selanjutnya adalah proses clustering

biasa yang menggunakan algoritma tersebut yang secara umum terdiri dari empat langkah sebagai berikut (Jain et al., 1999, p. 278; Liao et al., 2013, p. 124; Reddy & Jana, 2012, p. 396).

1. Membangkitkan centroid awal sejumlah k (c_1, c_2,…, c_k), $k \leq n$.
2. Menempatkan setiap data point (x_i, i = 1, 2, …, n) pada salah satu cluster (c_j, j = 1, 2, …, k) menurut jarak terdekatnya terhadap centroid.
3. Menghitung centroid baru berdasarkan data yang menjadi anggota tiap centroidnya, dengan nilai $c_i^* = \frac{1}{n_i}\sum_{x_j \in C_i} x_j$ untuk $i = 1, 2, \ldots, k$.
4. Untuk $c_i^* - c_i, \forall i - 1, 2, \ldots, k$ atau apabila jumlah pengulangannya ρ maka berhenti. Namun apabila tidak, maka kembali ke langkah ke dua.

Untuk menguji metode perbaikan inisialisasi K-Means ini, penulis menggunakan tiga set data training yang dibangkitkan dari dokumen teks yang diunduh dari internet. Dua set data yang dimaksud memiliki jumlah dokumen dan tema yang berbeda, sedangkan set data yang ketiga merupakan gabungan set data pertama dan kedua. Tema setiap dokumen ditentukan secara manual oleh penulis berdasarkan pemahaman isinya.

Proses clustering untuk masing-masing set data dilakukan menggunakan algoritma K-Means dengan dua metode inisialisasi, yaitu metode Forgy dan metode yang dikembangkan oleh penulis, yang masing-masing metode akan dijalankan sebanyak 10 kali untuk setiap set datanya. Jumlah pengulangan (p) dalam setiap proses clustering adalah 10, sedangkan kesamaan centroid (sebelumnya dengan saat ini) dilihat dari nilai *Sum Square Error* (SSE) yang dihitung menggunakan jarak euclideance.

Kualitas luaran proses clustering diukur menggunakan rerata *Root Mean Square Error* (RMSE) untuk setiap centroidnya terhadap data point di setiap clusternya. Disamping itu juga dicatat jumlah tema yang muncul dalam setiap clusternya serta pengulangan yang dibutuhkan untuk mencapai konvergensi dalam proses clustering juga akan dibandingkan.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{n}}$$

HASIL DAN PEMBAHASAN

1. Sintetik data

Seperti yang dijelaskan di atas, dokumen yang digunakan untuk menguji metode inisialisasi yang dikembangkan dibentuk dari dokumen internet. Jumlah seluruh dokumen adalah 115 dokumen yang terdiri dari 5 tema (lihat table 1).

2. Hasil clustering menggunakan algoritma K-means

Tabel 2 merupakan hasil clustering set data dengan menggunakan metode Forgy dan metode yang dikembangkan oleh penulis.

Dari Tabel 2 terlihat bahwa RMSE setiap cluster dalam setiap hasilnya (run) sangat sering berubah-ubah. Hal ini terjadi karena inisial centroid yang dihasilkan tidak konsistensi (karena dipilih secara acak). Hal ini berbeda dengan hasil dari metode yang dikembangkan. Nilai RMSE (Tabel 3) terlihat bahwa hanya pada run ke 8 saja nilai tersebut tidak konsisten. Apabila dilihat dari jumlah tema, hasil clustering dengan metode Forgy juga menunjukkan seringnya muncul dua tema atau lebih dalam satu cluster (34%) sedangkan metode yang dikembangkan hanya muncul dua kali (4%). Apabila kedua indikator tersebut dipadukan, berupa rerata RMSE dikali jumlah tema tiap cluster, maka terlihat nilai bahwa

metode Forgy hanya konsisten sebanyak 2 kali, sedangkan metode yang dikembangkan konsisten sebanyak 9 kali. Disamping itu banyaknya nilai rerata yang besar menunjukkan banyak jumlah tema yang berbeda yang tergabung dalam satu cluster.

Dari Tabel 4 nampak bahwa pengulangan yang dibutuhkan serta hasil yang didapat (rerata RMSE dikalikan jumlah tema tiap cluster) dalam setiap melakukan clustering sangat berbeda. Metode Forgy membutuhkan rerata pengulangan sebanyak 6.4 sedangkan metode baru menggunakan pengulangan sebanyak 3,2. Meskipun terdapat nilai pengulangan yang bernilai kecil pada metode Forgy, yaitu 4, namun rerata RMSE dikalikan jumlah tema tiap clusternya bernilai besar yang menunjukkan adanya cluster yang memiliki dua atau lebih tema yang berbeda. Atau dengan kata lain, ada tema yang tidak relevan tergabung dalam satu cluster.

Tabel 1. Tema dokumen dalam set data

No	Tema	Jumlah dokumen
1	Agne Monica	15
2	Hadi Poernomo & KPK	30
3	SmartPhone	25
4	MH370	25
5	Sengketa Pilpres	20

Table 2. RMSE & jumlah tema tiap cluster menggunakan metode Forgy

Run	Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5		Rerata RMSE x Jml. Tema
	RMSE	Jml. Tema	RMSE	Jml. Tema	RMSE	Jml. Tema	RMSE	Jml. Tema	RMSE	Jml. Tema	
1	652,24	1	486,57	1	1362,49	1	1057,15	1	1105,23	1	932,736
2	1073,53	3	486,57	1	1340,2	1	42,18	1	1199,08	1	1257,724
3	652,24	1	486,57	1	1362,49	1	1057,15	1	1105,23	1	932,736
4	1041,31	2	486,57	1	137,15	1	885,63	2	1105,23	1	1116,566
5	917,01	3	1274,04	2	1181,92	2	1281,54	1	674,8	1	1923,858
6	652,24	1	898,31	2	1125,23	1	1322,49	1	1105,23	1	1200,362
7	1086,67	2	245,48	1	486,57	1	1362,48	1	1122,13	2	1302,426
8	898,07	3	1237,95	2	517,42	3	1078,98	3	1259,85	3	2747,772
9	1010,69	2	486,57	1	1325,52	1	1056,72	1	1105,23	1	1199,084
10	652,24	1	928,83	2	1366,49	1	781,72	2	1094,17	1	1306,8

Table 3. RMSE jumlah tema tiap cluster menggunakan metode yg dikembangkan

Run	Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5		Rerata RMSE x Jml. Tema
	RMSE	Jml. Tema	RMSE	Jml. Tema	RMSE	Jml. Tema	RMSE	Jml. Tema	RMSE	Jml. Tema	
1	652,24	1	486,57	1	1362,48	1	1057,15	1	1105,23	1	932,734
2	652,24	1	486,57	1	1362,48	1	1057,15	1	1105,23	1	932,734
3	652,24	1	486,57	1	1362,48	1	1057,15	1	1105,23	1	932,734
4	652,24	1	486,57	1	1362,48	1	1057,15	1	1105,23	1	932,734
5	652,24	1	486,57	1	1362,48	1	1057,15	1	1105,23	1	932,734
6	652,24	1	486,57	1	1362,48	1	1057,15	1	1105,23	1	932,734
7	652,24	1	486,57	1	1362,48	1	1057,15	1	1105,23	1	932,734
8	623,09	2	1071,25	2	1233,63	1	1261,26	1	1057,15	1	1388,144
9	652,24	1	486,57	1	1362,48	1	1057,15	1	1105,23	1	932,734
10	652,24	1	486,57	1	1362,48	1	1057,15	1	1105,23	1	932,734

Table 4. Pengulangan yang dibutuhkan dan rerata RMSE x jumlah tema

Run	Metode Forgy		Metode Baru	
	Pengulangan	Rerata RMSE x Jml. Tema	Pengulangan	Rerata RMSE x Jml. Tema
1	7	932,736	4	932,734
2	10	1257,724	3	932,734
3	8	932,736	4	932,734
4	10	1116,566	3	932,734
5	4	1923,858	3	932,734
6	5	1200,362	3	932,734
7	4	1302,426	3	932,734
8	4	2747,772	3	1388,144
9	5	1199,084	3	932,734
10	7	1306,8	3	932,734

SIMPULAN DAN SARAN

Hasil percobaan yang telah dilakukan menunjukkan bahwa metode inisialisasi menggunakan graf hutan yang minimum menghasilkan centroid awal yang lebih baik dan konsisten dibandingkan metode Forgy. Disamping itu jumlah perulangan yang harus dilakukan dalam proses clustering dengan menggunakan metode ini juga menunjukkan hasil yang lebih baik (rerata 3,2) dibandingkan metode Forgy (rerata 6,4). Namun demikian, untuk memastikan kualitas dari metode ini perlu adanya penelitian yang lebih lanjut

yang melibatkan jumlah data yang jauh lebih besar dengan tingkat kemiripan tema yang lebih tinggi.

DAFTAR PUSTAKA

Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications, 40(1), 200–210. doi:10.1016/j.eswa.2012.07.021

Feldman, R., & Sanger, J. (2007). The Text Mining Handbook. New York: Cambridge University Press.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM Computing Surveys, 31(3), 264–323. doi:10.1145/331499.331504

Khan, S. S., & Ahmad, A. (2013). Cluster center initialization algorithm for K-modes clustering. Expert Systems with Applications, 40(18), 7444–7456. doi:10.1016/j.eswa.2013.07.002

Liao, K., Liu, G., Xiao, L., & Liu, C. (2013). A sample-based hierarchical adaptive K-means clustering method for large-scale video retrieval. Knowledge-Based Systems, 49, 123–133. doi:10.1016/j.knosys.2013.05.003

Reddy, D., & Jana, P. K. (2012). Initialization for K-means Clustering using Voronoi Diagram. Procedia Technology, 4, 395–400. doi:10.1016/j.protcy.2012.05.061

Rokach, L. (2010). A survey of Clustering Algorithms. In O. Maimon & L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook (2nd ed.). Springer Science+Business Media, LLC. doi:10.1007/978-0-387-09823-4_14

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., … Steinberg, D. (2007). Top 10 algorithms in data mining. Knowledge and Information Systems, 14(1), 1–37. doi:10.1007/s10115-007-0114-2