

Penerapan Gravitational Search Algorithm untuk Optimasi Klusterisasi Fuzzy C-Means

Ali Mulyanto

Program Studi Teknik Informatika STMIK Eresha

Email: aliemulyanto@gmail.com

Romi Satria Wahono

Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

Email: romi@brainmatics.com

Abstract: Klusterisasi fuzzy merupakan masalah penting yang merupakan subjek penelitian aktif dalam beberapa aplikasi dunia nyata. Algoritma fuzzy c-means (FCM) merupakan salah satu teknik pengelompokan fuzzy yang paling populer karena efisien, dan mudah diimplementasikan. Namun, FCM sangat mudah terjebak pada kondisi local minimum. Gravitational search algorithm (GSA) merupakan salah satu metode heuristik yang efektif untuk menemukan solusi optimal terdekat. GSA digabungkan ke FCM untuk menemukan pusat kluster yang optimal dengan meminimalkan fungsi objektif FCM. Hasil penelitian menunjukkan bahwa metode yang diusulkan gravitational search algorithm fuzzy c-means (GSA-FCM) dapat menunjukkan hasil yang lebih optimal daripada algoritma FCM.

Keywords: klusterisasi fuzzy, Fuzzy c-means, gravitational search algorithm

1. PENDAHULUAN

Klasifikasi dan klusterisasi merupakan dua bidang garapan yang paling sering ditemui untuk mengekstrak pengetahuan. Analisis kluster merupakan metode *unsupervised learning* dalam analisis data yang digunakan untuk membuat penilaian awal dari struktur data, untuk menemukan struktur yang tersembunyi dalam dataset, dan untuk mengekstrak informasi (Azar, El-Said, & Hassanien, 2013). Sebagai metode *unsupervised learning*, tujuan dari analisis kluster adalah untuk menemukan fitur struktur data melalui metode pemartisan.

Klusterisasi merupakan proses mengelompokkan objek atau pola yang bertujuan untuk menempatkan objek data ke dalam satu himpunan atau kelompok yang saling berhubungan (disebut kluster) sehingga objek dalam setiap kluster memiliki kemiripan satu sama lain. Dengan demikian unsur-unsur dalam kluster memiliki derajat kesamaan yang besar daripada kesamaan data tersebut dengan data dalam kelompok lain (Izakian & Abraham, 2011). Dalam beberapa tahun terakhir, klusterisasi telah banyak diterapkan di berbagai bidang seperti pattern recognition, machine learning, data mining (Izakian & Abraham, 2011), analisis data statistik, dan segmentasi citra (Taherdangkoo & Bagheri, 2013). Teknik klusterisasi yang paling populer adalah metode hirarki dan metode partisi (de Carvalho, Lechevallier, & de Melo, 2012).

Klusterisasi hirarki menemukan urutan partisi yang diawali dari satu data tunggal yang dianggap sebagai sebuah kelompok, dua atau lebih kelompok kecil data bergabung menjadi sebuah kelompok besar dan begitu seterusnya sampai semua data dapat bergabung menjadi sebuah kelompok *singleton* (Pimentel & de Souza, 2013). Metode ini dapat diklasifikasikan lebih lanjut ke dalam metode agglomerative dan metode divisive (Azar et al., 2013). Metode

agglomerative (de Carvalho et al., 2012) menghasilkan urutan partisi bersarang dimulai dengan pengelompokan terendah di mana setiap item data berada dalam kluster yang unik dan berakhir dengan pengelompokan dimana semua item data berada dalam kluster yang sama. Sedangkan metode divisive dimulai dengan semua item data dalam satu kluster dan melakukan prosedur pemisahan sampai kriteria berhenti terpenuhi atau setelah mendapatkan partisi kluster tunggal.

Klusterisasi partisi secara langsung membagi dataset ke beberapa kluster yang tetap menggunakan fungsi objektif yang sesuai. Keuntungan dari metode partisi adalah kemampuannya untuk memanipulasi dataset dalam jumlah yang besar. Pimentel & de Souza (2013) mengelompokkan klusterisasi partisi ke dalam *hard partition* dan *fuzzy partition*. Dalam metode pengelompokan *hard partition*, setiap objek dari kumpulan data harus ditugaskan secara tepat pada satu kluster. Kelemahan utama dari teknik pengelompokan *hard partition* (Azar et al., 2013) adalah bahwa dengan metode ini mungkin akan kehilangan beberapa informasi penting yang mengarah pada pengelompokan tersebut. Sedangkan pengelompokan *fuzzy partition* didasarkan pada gagasan dari keanggotaan parsial dari masing-masing pola dalam sebuah kluster tertentu. Hal ini memberikan fleksibilitas untuk menyatakan bahwa titik data memiliki lebih dari satu kluster pada waktu yang sama dan derajat keanggotaannya jauh lebih halus dari model data. Selain menetapkan titik data ke dalam kluster, menurut (Azar et al. (2013) derajat keanggotaan juga bisa mengungkapkan ambiguitas titik data yang dimiliki sebuah kluster.

Ada beberapa metode yang digunakan untuk klusterisasi (Oliveira & Pedrycz, 2007) yaitu k-means, possibilistic c-means (PCM) dan fuzzy c-means (FCM). K-means merupakan salah satu teknik klusterisasi terkenal untuk *hard partition*. Algoritma k-means adalah algoritma pengelompokan partisi yang efisiensi dalam mengelompokkan dataset yang besar. Namun menurut (Bai, Liang, & Dang, 2011), penggunaan algoritma k-means terbatas hanya pada data numerik. Klusterisasi PCM merupakan salah satu metode klasifikasi yang kuat terhadap *noise* atau data terisolasi (Hamasuna, Endo, & Miyamoto, 2009). Namun algoritma klusterisasi PCM (Ji, Sun, & Xia, 2011) mengorbankan stabilitas algoritma dan terlalu sensitif terhadap inisialisasi kluster.

2. PENELITIAN TERKAIT

FCM merupakan salah satu metode pengelompokan yang paling terkenal (Wu, 2012) (Maimon & Rokach, 2010), paling banyak digunakan (Zhao, Jiao, & Liu, 2013). Algoritma FCM juga memiliki karakteristik yang kuat untuk ambiguitas dan dapat menyimpan informasi lebih banyak daripada metode *hard c-means* (Yong Zhang, Huang, Ji, & Xie, 2011). Namun algoritma FCM dalam pencarian kluster yang optimal

didasarkan pada fungsi objektif, sehingga mudah terjebak pada kondisi dimana nilai yang dihasilkan bukan nilai terendah dari himpunan solusi atau disebut *local minimum* (Dong, Dong, Zhou, Yin, & Hou, 2009).

Untuk memecahkan masalah pada algoritma FCM, para peneliti telah berhasil menerapkan algoritma evaluasi untuk meningkatkan kinerja FCM seperti ant colony optimization (Kanade & Hall, 2007). Kanade dan Hall (2007) memperkenalkan suatu algoritma dengan konsep peningkatan laju penguapan feromon data yang lebih dekat ke pusat kluster dan mencapai respon yang jauh lebih baik daripada algoritma sebelumnya. Evolutionary Programming Fuzzy C-Means (EPFCM) dimanfaatkan untuk mengoptimalkan fungsi objektif pada FCM (Dong et al., 2009). Algoritma artificial bee colony yang diusulkan oleh (Karaboga & Ozturk, 2010) meniru perilaku lebah madu dalam mencari makanan untuk mengatasi masalah seleksi acak pada pusat kluster awal pada FCM. Fuzzy Particle Swarm Optimization (FPSO) (Izakian & Abraham, 2011) juga telah digunakan untuk mengatasi masalah seleksi acak di titik pusat FCM.

Gravitational search algorithm (GSA) merupakan salah satu metode optimisasi *heuristik* yang efektif dalam analisa kluster yang diusulkan untuk memecahkan masalah pada FCM. Motivasi penggunaan GSA didasarkan pada kesuksesan para peneliti dalam memecahkan masalah optimasi. Kelebihan GSA menurut (Rashedi, Nezamabadi-pour, & Saryazdi, 2009) adalah kemampuan menemukan hasil yang lebih optimal dari algoritma optimasi yang lain. Kelebihan lain dari GSA (Kumar, Chhabra, & Kumar, 2014) terletak pada penggunaan memori yang lebih kecil dari algoritma optimasi lainnya, serta posisi agen yang ikut berpartisipasi dalam memperbaiki iterasi. Pendekatan GSA yang diusulkan oleh (Rashedi, Nezamabadi-pour, & Saryazdi, 2011) telah digunakan untuk memecahkan masalah estimasi parameter untuk *infinite impulse response* (IIR) dan menfilter rasional *non-linier*. Hal ini menunjukkan bahwa GSA dapat memecahkan masalah yang kompleks dan hasil penentiannya menunjukkan bahwa kinerja GSA sebanding dengan algoritma genetic algorithm dan particle swarm optimization. Pendekatan GSA juga diusulkan oleh (Hatamlou, Abdullah, & Nezamabadi-pour, 2012) yang telah digunakan untuk mencari ruang masalah dalam menemukan solusi optimal yang terdekat untuk memecahkan masalah pada algoritma k-means.

Pada penelitian ini, GSA akan diterapkan untuk mengoptimalkan fungsi objektif pada FCM.

3. METODE YANG DIUSULKAN

Metode yang diusulkan untuk mengatasi masalah *local minimum* adalah algoritma berbasis populasi, dimana beberapa calon solusi untuk masalah pengelompokkan diciptakan secara acak. Masing-masing solusi kandidat yang juga disebut massa (agen), menemukan pusat kluster. Setelah membuat solusi acak untuk masalah klusterisasi, calon solusi berinteraksi sebagai massa dalam semesta melalui hukum gravitasi Newton. Dengan cara ini, calon agen yang juga memiliki massa yang besar, menarik massa lain dan menjadi agen untuk menemukan solusi yang lebih baik. Jumlah massa untuk setiap agen akan dihitung dengan fungsi objektif calon agen tersebut. Solusi akan ditemukan ketika agen memiliki nilai fungsi objektif terkecil dan memiliki massa yang besar.

Kontribusi utama dari metode yang diusulkan adalah memanfaatkan algoritma GSA untuk mengoptimalkan fungsi objektif sehingga kinerja pengelompokkan dapat dicapai. Tujuan ini dapat dicapai dengan memperkenalkan sekelompok

fungsi objektif yang direpresentasikan sebagai agen dalam algoritma *gravitational search algorithm*, dimana dimensi agen ditentukan oleh jumlah kluster. Para agen bergerak melalui ruang pencarian menggunakan aturan algoritma dan proses ini terus berlanjut sampai kriteria konvergensi terpenuhi.

Tahapan dari metode yang diusulkan ditunjukkan pada Gambar 1 dan dijabarkan dalam algoritma sebagai berikut:

1. Menyiapkan dataset kemudian melakukan reduksi data dengan tujuan untuk menghilangkan atribut yang tidak diperlukan dalam proses algoritma.
2. Inisialisasi jumlah kluster, nilai *epoch*, nilai maksimal iterasi, nilai gravitasi konstanta, nilai best, nilai worst, Nilai *mass*, nilai *force*, nilai *acceleration*, dan nilai *velocity*.
3. Inisialisasi derajat keanggotaan data pada pusat kluster.
4. Menghitung fungsi *fitness*. *Fitness* merupakan ukuran kinerja individu agar tetap bertahan hidup dalam lingkungannya. Dalam GSA, fungsi *fitness* adalah fungsi objektif dari masalah yang akan dioptimasi. Fungsi *fitness* yang digunakan dalam penelitian ini mengadopsi rumus fungsi objektif yang terdapat pada FCM untuk meminimumkan jarak antara data dengan titik pusat kluster. Fungsi *fitness* ditunjukkan pada persamaan (1).

$$J = \sum_{i=1}^n \sum_{k=1}^c \left(\sum_{j=1}^m (X_{ij} - C_{kj})^2 \right) (\mu_{ik})^w \quad (1)$$

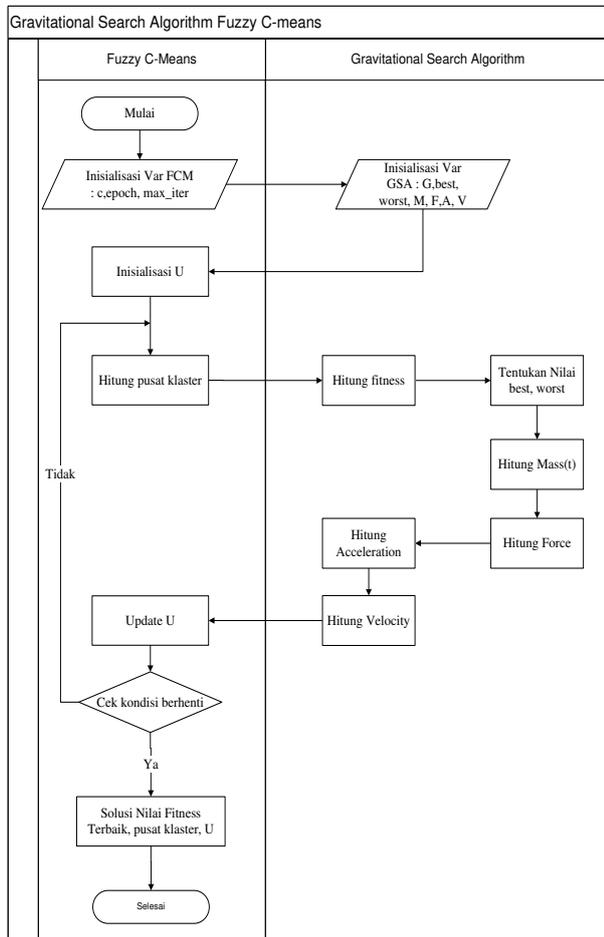
dimana $\|X_{ij} - C_{kj}\|$ menentukan jarak antara objek X_i dan pusat kluster V_k , n adalah banyaknya item data, c adalah banyaknya kluster yang terbentuk, m adalah dimensi pusat kluster, w adalah nilai pembobotan *fuzzyness*, X adalah item data, V adalah pusat kluster, dan μ adalah derajat keanggotaan data pada pusat kluster. Fungsi jarak yang digunakan dalam penelitian ini adalah *Euclidean distance* yang ditunjukkan pada persamaan (2).

$$\|X_i - C_j\| = \sqrt{\sum_{j=1}^c (X_i - C_j)^2} \quad (2)$$

dimana c adalah jumlah kluster, X adalah data, i adalah banyaknya data dan j adalah jumlah kluster. Nilai dari fungsi *fitness* yang dihasilkan kemudian diseleksi untuk menentukan agen terbaik yang memiliki nilai terkecil sebagai calon akhir solusi dengan persamaan (3), dan juga menentukan salah satu agen terburuk dengan persamaan (4). Agen terbaik maupun agen terburuk digunakan untuk menghitung nilai tiap massa.

$$best(t) = \min \{fitness(t)\}, j \in \{1, 2, \dots, S\} \quad (3)$$

$$worst(t) = \max \{fitness(t)\}, j \in \{1, 2, \dots, S\} \quad (4)$$



Gambar 1. Flowchart Algoritma yang Diusulkan

Setiap solusi kandidat di dalam populasi terdiri dari *array* satu dimensi seperti ditunjukkan pada Gambar 2 dengan panjang $d \times k$ yang digunakan untuk menampung semua kandidat solusi, dimana d adalah dimensi objek data dan k adalah jumlah dari kelompok kluster yang diinginkan. Dalam penelitian ini, hukum gravitasi Newton untuk klusterisasi ditunjukkan dengan persamaan (7).

$$F_i^d(t) = \sum_{j \in kbest, j \neq i} \frac{rand_j G(t) (M_j(t) M_i(t))}{(R_{ij}(t) + \epsilon) (x_j^d(t) - x_i^d(t))} \quad (7)$$

dimana $rand_j$ adalah bilangan acak dalam rentang nilai 0 sampai 1, ϵ adalah nilai kecil untuk menghindari pembagian bilangan nol, R_{ij} adalah jarak *euliden* antara dua agen i dan agen j , $kbest$ adalah himpunan pertama agen k dengan nilai *fitness* terbaik dan massa terbesar, dan G adalah nilai konstanta gravitasi yang awalnya diberi nilai 1 dan nilai menurun sampai nilai nol pada iterasi terakhir.

Persamaan (7) digunakan untuk menghitung hasil dari semua gaya yang bekerja pada massa (agen) yang dipilih oleh semua partikel lainnya. Sedangkan persamaan (8) digunakan untuk menghitung percepatan agen.

$$a_i(t) = \frac{F_i(t)}{M_i(t)}, i=1,2,...s \quad (8)$$

dimana F_i dan M_i adalah gaya gravitasi dan massa pada iterasi ke- i di dalam iterasi ke- t

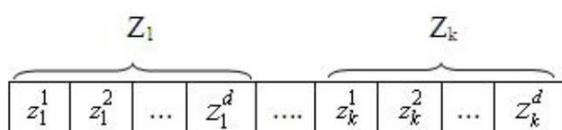
5. Menghitung massa agen menggunakan persamaan (5) dan persamaan (6) berdasarkan fungsi *fitness*, *best* dan *worst*.

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}, i = 1,2,...,S \quad (5)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (6)$$

dimana $M_i(t)$ dan $fit_i(t)$ mewakili nilai massa dan nilai *fitness* pada agen ke- i di iterasi ke- t , N adalah banyaknya data.

6. Menerapkan hukum gravitasi Newton ke dalam algoritma. Hukum gravitasi Newton menjelaskan bahwa setiap partikel di alam semesta menarik setiap partikel lain dengan kekuatan yang berbanding lurus dengan produk dari massa partikel dan berbanding terbalik dengan kuadrat dari jarak antar partikel (Rashedi et al., 2009). Untuk menerapkan hukum gravitasi Newton dalam kluster analisis, digunakan *array* untuk mengkodekan fungsi objektif.



Gambar 2. Contoh Kandidat Solusi Dipetakan Dalam Array Satu Dimensi

7. Update kecepatan agen (*velocity*) dengan persamaan (9) dan kemudian memperbarui derajat keanggotaan data dengan pusat kluster (10)

$$v_i^d(t + 1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (9)$$

$$U_i^d(t + 1) = U_i^d - v_i^d(t + 1) \quad (10)$$

dimana $rand_i$ adalah bilangan acak dalam rentang nilai 0 sampai 1, v_i^d adalah *velocity* ke- i pada dimensi ke- d dalam iterasi ke- t , a_i^d adalah *acceleration* ke- i dimensi ke- d dalam terasi ke- t , dan U_i^d adalah derajat keanggotaan data ke- i pada dimensi ke- d .

3.1 Evaluasi Model yang Diusulkan

Evaluasi tingkat keakuratan kluster terdiri dari dua (Maimon & Rokach, 2010) yaitu evaluasi yang hanya menggunakan nilai-nilai keanggotaan μ_{ij} dari partisi fuzzy data dan yang kedua evaluasi yang melibatkan kedua matriks U dan kumpulan data itu sendiri. Validasi algoritma yang menggunakan nilai-nilai keanggotaan μ_{ij} terdiri dari dua yaitu *partition coefficient* dan *classification entropy*, sedangkan yang melibatkan kedua matriks U dan kumpulan data yaitu *Xie-Beni index*. Pada penelitian ini, evaluasi tingkat keakuratan kluster menggunakan *partition coefficient*, *classification entropy* dan *Xie-Beni Index*. Ketiga evaluasi tingkat keakuratan kluster tersebut merupakan evaluasi keakuratan kluster yang paling banyak digunakan (Yunjie, Zhang & Wang, 2008).

3.1.1 Partition Coefficient (PC)

Partition coefficient (PC) mengukur *overlapping* antar klaster. PC diusulkan oleh Bezdeck di dalam (Maimon & Rokach, 2010) dengan persamaan (11)

$$PC = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^c (\mu_{ij})^2 \tag{11}$$

dimana N adalah banyaknya data dalam dataset. Semakin dekat nilai Partition coefficient ke 0, berarti *crisp clustering*. Nilai indeks yang mendekati batas atas 1 menunjukkan tidak adanya struktur pengelompokkan dalam kumpulan data atau ketidakmampuan algoritma tersebut.

3.1.2 Classification Entropy (CE)

Classification entropy mengukur *fuzziness* klaster (Azar et al., 2013). CE mirip dengan PC dimana evaluasi hanya menggunakan matriks keanggotaan saja. CE dihitung menggunakan persamaan (12).

$$CE(c) = -\frac{1}{N} \sum_{i=1}^c \sum_{j=1}^N \mu_{ij} \cdot \text{Log}(\mu_{ij}) \tag{12}$$

dimana N adalah banyaknya data dalam dataset, c adalah jumlah klaster. Ketika teknik klasterisasi dievaluasi, semakin dekat indeks PC dengan nilai 1 dan semakin dekat indeks CE dengan nilai 0, maka semakin baik pengelompokkan tersebut (Azar et al., 2013).

3.1.3 Xie-Beni Index (XB)

Xie-Beni Index diusulkan oleh Xie dan Beni di dalam (Maimon & Rokach, 2010). Xie mengukur keseluruhan kekompakan rata-rata pemisahan data antar klaster. Validasi Xie-Beni Index ditunjukkan dengan persamaan (13).

$$V_{xb} = \frac{\sum_{i=1}^c \sum_{j=1}^N \mu_{ij}^2 \|x_j - v_i\|^2}{N * \min_{i \neq j} \|x_i - v_j\|^2} \tag{13}$$

dimana c adalah jumlah klaster, N adalah banyaknya item data pada dataset, μ adalah derajat keanggotaan data pada pusat klaster, x adalah item data, v adalah pusat klaster.

Dalam persamaan (13), pembilang adalah jumlah dari kekompakan tiap klaster fuzzy dan penyebut adalah pemisahan minimal antara cluster fuzzy. Partisi fuzzy optimal diperoleh dengan meminimalkan nilai XB. Jika nilai XB yang dihasilkan semakin kecil, maka hasil klaster dinilai semakin baik (Azar et al., 2013).

4. HASIL EKSPERIMEN

Dalam eksperimen ini digunakan komputer dengan spesifikasi prosesor Intel B940 2.00 GHz, memori 2 GB dan sistem operasi Windows 7 Ultimate 32 bit serta pemrograman matematika untuk menguji alg. Dataset untuk pengujian metode gravitational search algorithm fuzzy c-means (GSA-FCM) menggunakan enam dataset seperti yang ditunjukkan pada Tabel 1 yang dapat diunduh secara bebas pada laman (<http://archive.ics.uci.edu/ml/datasets>).

Tabel 1. Dataset untuk Pengujian

Nama Dataset	Jumlah Atribut	Jumlah Sampel Data
Iris	5	150
Wine	13	178
Glass	10	214
Contraceptive Method Choice (CMC)	9	1.473
Seeds	7	210
Perfume Data	2	560

Nilai parameter untuk jumlah klaster diatur pada nilai 3, untuk *epoch* diatur pada nilai 10^{-5} , *max_iter* pada nilai 100. Dari hasil pengujian menunjukkan bahwa algoritma GSA-FCM mampu keluar dari kondisi *local minimum* yaitu dengan menghasilkan fungsi objektif yang lebih optimal dari algoritma FCM seperti yang ditunjukkan pada Tabel 2.

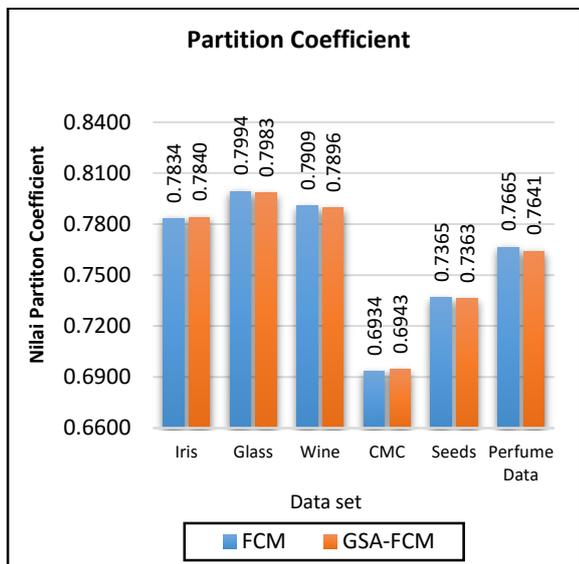
Tabel 2. Komparasi Algoritma FCM dengan GSA-FCM

No	Dataset	FCM		GSA-FCM	
		Iterasi	Fungsi Objektif	Iterasi	Fungsi Objektif
1	Iris	24	6.058,6900	19	6.058,5599
2	Wine	40	69.393,0820	55	69.090,4074
3	Glass	44	1.796.125,9370	50	1.792.180,9158
4	CMC	24	18.137,3141	65	18.142,8910
5	Seeds	16	441,2240	9	443,8684
6	Perfume Data	35	6.954,6250	23	6.947,7642

Tabel 3 dan Gambar 3 menunjukkan validasi hasil klasterisasi untuk dataset iris, glass, wine, CMC, seeds dan perfume data dengan menggunakan pengukuran partition coefficient. Validasi hasil klasterisasi pada data set iris dan CMC menunjukkan nilai yang lebih mendekati pada nilai satu. Hal ini menunjukkan bahwa algoritma yang diusulkan memiliki kinerja yang lebih baik dari algoritma FCM. Sedangkan validasi hasil klasterisasi pada data set glass, wine, seeds dan perfume data menghasilkan nilai yang lebih mendekati nilai nol. Hal ini menunjukkan bahwa kinerja algoritma FCM pada ketiga data set tersebut memiliki kinerja yang lebih baik dari algoritma yang diusulkan.

Tabel 3. Validasi Hasil Klasterisasi dengan Partition Coefficient

Dataset	Partition Coefficient	
	FCM	GSA-FCM
Iris	0,7834	0,7840
Glass	0,7994	0,7983
Wine	0,7909	0,7896
CMC	0,6934	0,6943
Seeds	0,7365	0,7363
Perfume Data	0,7665	0,7641

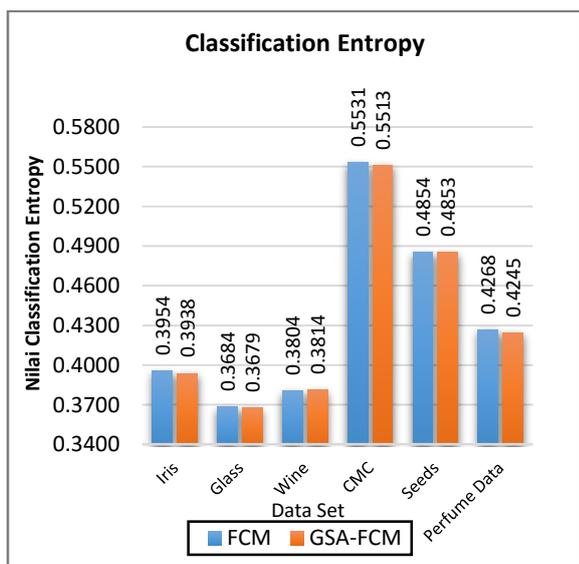


Gambar 3. Pengukuran Validasi Kluster dengan PC

Tabel 4 dan Gambar 4 menunjukkan validasi hasil klusterisasi untuk dataset iris, glass, wine, CMC, seeds, perfume data dengan menggunakan pengukuran classification entropy. Dari enam data set yang diuji, lima dataset yaitu iris, glass, CMC, seeds dan perfume data menghasilkan nilai yang lebih mendekati nol. Hal ini menunjukkan bahwa kinerja algoritma yang diusulkan lebih baik dari algoritma FCM.

Tabel 4. Validasi Hasil Klusterisasi dengan Classification Entropy

Dataset	Classification Entropy	
	FCM	GSA-FCM
Iris	0,3954	0,3938
Glass	0,3684	0,3679
Wine	0,3804	0,3814
CMC	0,5531	0,5513
Seeds	0,4854	0,4853
Perfume Data	0,4268	0,4245



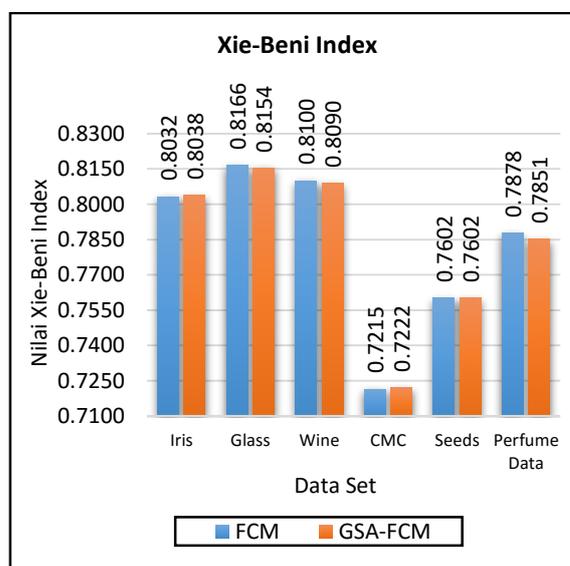
Gambar 4. Pengukuran Validasi Kluster dengan CE

Tabel 5 dan Gambar 5 menunjukkan evaluasi hasil klusterisasi menggunakan Xie-Beni Index. Dari hasil

pengujian untuk dataset glass, wine dan perfume data menunjukkan kinerja algoritma yang diusulkan lebih baik dari algoritma FCM. Sedangkan pada data set iris dan CMC menunjukkan kinerja algoritma FCM masih lebih baik dibandingkan algoritma yang diusulkan. Sedangkan pada data set seeds, kinerja kedua algoritma menunjukkan kinerja yang seimbang.

Tabel 5 Validasi Hasil Klusterisasi dengan Xie-Beni Index

Dataset	Xie-Beni Index	
	FCM	GSA-FCM
Iris	0,8032	0,8038
Glass	0,8166	0,8154
Wine	0,8100	0,8090
CMC	0,7215	0,7222
Seeds	0,7602	0,7602
Perfume Data	0,7878	0,7851



Gambar 5. Pengukuran Validasi Kluster dengan XB

Secara keseluruhan, validasi hasil klusterisasi menggunakan PC, CE dan XB menunjukkan bahwa algoritma yang diusulkan lebih baik dari algoritma FCM.

5. KESIMPULAN

Dari penelitian yang dilakukan, fuzzy c-means dengan optimasi gravitational search algorithm terbukti memiliki kinerja yang lebih baik dari algoritma FCM. Pengujian algoritma dilakukan pada data set iris, wine, glass, CMC, seeds dan perfume data. Empat data set yaitu iris, wine, glass dan perfume data menghasilkan fungsi objektif yang lebih optimal dari algoritma FCM. Hal ini membuktikan bahwa algoritma yang diusulkan dapat keluar dari kondisi *local minimum*.

Evaluasi validitas kluster dengan menggunakan Partition Coefficient (PC), Classification Entropy (CE) dan Xie-Beni Index, juga membuktikan bahwa gravitational search algorithm fuzzy c-mans (GSAFCM) mampu menghasilkan kualitas kluster yang lebih optimal. Hal ini dibuktikan dengan nilai CE dan Xie-Beni Index yang lebih mendekati ke nilai nol. Sedangkan pada nilai PC, hasil klusterisasi dari algoritma yang diusulkan lebih optimal pada dataset iris dan CMC.

Meskipun model yang diusulkan sudah memberikan hasil yang lebih baik, namun untuk penelitian selanjutnya dapat dilakukan:

1. Dalam eksperimen dengan data set yang besar, algoritma GSAFCM mengalami konvergensi terlalu dini, sehingga masih perlu ditambahkan operator baru untuk lebih mengeksplorasi solusi.
2. Penentuan nilai konstan pada parameter G dalam algoritma GSA-FCM yang kurang tepat dapat mempengaruhi hasil klasterisasi, sehingga diperlukan suatu metode baru agar penentuan nilai G lebih akurat.

REFERENSI

- Alata, M., Molhim, M., Ramini, A., & Arabia, S. (2013). Using GA for Optimization of the Fuzzy C-Means Clustering Algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, 5(3), 695–701.
- Azar, A. T., El-Said, S. A., & Hassanien, A. E. (2013). Fuzzy and hard clustering analysis for thyroid disease. *Computer Methods and Programs in Biomedicine*, 111(1), 1–16.
- Bai, L., Liang, J., & Dang, C. (2011). An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data. *Knowledge-Based Systems*, 24(6), 785–795.
- Dawson, C. W. (2009). *Projects in Computing and Information Systems*.
- De Carvalho, F. D. a. T., Lechevallier, Y., & de Melo, F. M. (2012). Partitioning hard clustering algorithms based on multiple dissimilarity matrices. *Pattern Recognition*, 45(1), 447–464.
- Dong, H., Dong, Y., Zhou, C., Yin, G., & Hou, W. (2009). A fuzzy clustering algorithm based on evolutionary programming. *Expert Systems With Applications*, 36(9), 11792–11800.
- Hamasuna, Y., Endo, Y., & Miyamoto, S. (2009). On tolerant fuzzy c-means clustering and tolerant possibilistic clustering. *Soft Computing*, 14(5), 487–494.
- Hatamlou, A., Abdullah, S., & Nezamabadi-pour, H. (2011). Application of Gravitational Search Algorithm, 337–346.
- Hatamlou, A., Abdullah, S., & Nezamabadi-pour, H. (2012). A combined approach for clustering based on K-means and gravitational search algorithms. *Swarm and Evolutionary Computation*, 6, 47–52.
- Izakian, H., & Abraham, A. (2011). Fuzzy C-means and fuzzy swarm for fuzzy clustering problem. *Expert Systems With Applications*, 38(3), 1835–1838.
- Ji, Z., Sun, Q., & Xia, D. (2011). Computerized Medical Imaging and Graphics A modified possibilistic fuzzy c -means clustering algorithm for bias field estimation and segmentation of brain MR image. *Computerized Medical Imaging and Graphics*, 35(5), 383–397.
- Kanade, P., & Hall, L. (2007). Fuzzy ants and clustering. *Systems, Man and Cybernetics, Part A: ...*, 37(5), 758–769.
- Karaboga, D., & Ozturk, C. (2010). Fuzzy clustering with artificial bee colony algorithm. *Scientific Research and Essays*, 5(14), 1899–1902.
- Kumar, V., Chhabra, J. K., & Kumar, D. (2014). Automatic cluster evolution using gravitational search algorithm and its application on image segmentation. *Engineering Applications of Artificial Intelligence*, 29, 93–103.
- Maimon, O., & Rokach, L. (2010). Data Mining and Knowledge Discovery Handbook. *Zhurnal Eksperimental'noi I Teoreticheskoi Fiziki*.
- Oliveira, J. V. De, & Pedrycz, W. (2007). *Advances in Fuzzy Clustering and its Applications*. (J. Valente de Oliveira & W. Pedrycz, Eds.). Chichester, UK: John Wiley & Sons, Ltd.
- Pimentel, B. a., & de Souza, R. M. C. R. (2013). A multivariate fuzzy c-means method. *Applied Soft Computing*, 13(4), 1592–1607.
- Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. *Information Sciences*, 179(13), 2232–2248.
- Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2011). Filter modeling using gravitational search algorithm. *Engineering Applications of Artificial Intelligence*, 24(1), 117–122.

- Ross, T. J. (2004). *Fuzzy Logic with Engineering Application*. Wiley.
- Sabzekar, M., & Naghibzadeh, M. (2013). Fuzzy c-means improvement using relaxed constraints support vector machines. *Applied Soft Computing*, 13(2), 881–890. d
- Satapathy, S., & Patnaik, S. (2011). Data clustering using modified fuzzy-PSO (MFPSO). *Multi-Disciplinary Trends in Artificial Intelligence*, 7080, 136–146.
- Taherdangkoo, M., & Bagheri, M. H. (2013). A powerful hybrid clustering method based on modified stem cells and Fuzzy C-means algorithms. *Engineering Applications of Artificial Intelligence*, 26(5-6), 1493–1502.
- Wu, K. (2012). Analysis of parameter selections for fuzzy c -means. *Pattern Recognition*, 45(1), 407–415.
- Zhang, Y., Huang, D., Ji, M., & Xie, F. (2011). Image segmentation using PSO and PCM with Mahalanobis distance. *Expert Systems with Applications*, 38(7), 9036–9040.
- Zhang, Y., & Wang, W. (2008). A cluster validity index for fuzzy clustering, 178, 1205–1218.
- Zhao, F., Jiao, L., & Liu, H. (2013). Kernel generalized fuzzy c-means clustering with spatial information for image segmentation. *Digital Signal Processing*, 23(1), 184–199.

BIOGRAFI PENULIS



Ali Mulyanto. Memperoleh gelar S.Kom dari Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Muhammadiyah Jakarta dan M.Kom dari program pasca sarjana program studi Teknik Informatika STMIK Eresha (d/a STTBI Benarif). Pernah bekerja sebagai ketua Program Studi Manajemen Informatika, kemudian sebagai ketua Program Studi Teknik Informatika di STMIK Cikarang. Saat ini bekerja sebagai dosen dan wakil ketua I bidang akademik di STMIK Cikarang.



Romi Satria Wahono. Mendapatkan gelar B.Eng and M.Eng di bidang ilmu komputer dari Saitama University, Jepang, dan gelar Ph.D di bidang software engineering dari Universiti Teknikal Malaysia Melaka. Saat ini menjadi pengajar dan peneliti pada Program Pascasarjana Ilmu Komputer di Univeristas Dian Nuswantoro. Selain itu juga sebagai founder dan CEO PT Brainmatics Cipta Informatika., sebuah perusahaan pengembang perangkat lunak di Indonesia. Tertarik pada penelitian di bidang software engineering dan machine learning. Professional member dari ACM, PMI dan IEEE Computer Society.