

ANALISIS CLUSTER OTOMATIS MENGGUNAKAN ALGORITMA NOVEL MODIFIED DIFFERENTIAL EVOLUTION

Achmad Yasid¹
ach_yazid@yahoo.com
Universitas Trunojoyo Madura

Budi Dwi Satoto²
budids@gmail.com
Universitas Trunojoyo Madura

ABSTRAK

Analisis *cluster* merupakan salah satu permasalahan pembelajaran tidak terbimbing dan teknik data mining yang penting. Akan tetapi, untuk menentukan jumlah *cluster* akhir merupakan suatu tugas yang menantang. Oleh karena itu, penelitian ini bermaksud mengusulkan algoritma *novel modified differential evolution* (NMDE) dan algoritma *k-means* (NMDE-*k-means*) pada analisis *cluster* otomatis. Algoritma ini dapat menentukan jumlah *cluster* akhir dan melakukan pengelompokan data secara otomatis. Pada prinsipnya Algoritma NMDE akan melakukan pencarian global untuk menemukan jumlah *cluster* dan partisi data, sedangkan algoritma *k-means* akan memperbaiki kinerja algoritma NMDE dalam menentukan *centroid* cluster. Empat data set yang sudah dikenal yaitu Iris, Wine, Glass dan Vowel digunakan untuk memvalidasi algoritma ini. Hasil komputasi menunjukkan bahwa algoritma ini lebih baik dibandingkan dengan empat algoritma cluster otomatis lainnya yaitu *improved automatic clustering based differential evolution* (ACDE), *automatic clustering using differential evolution and k-means* (ACDE-*k-means*) dan algoritma cluster otomatis yang berbasis *particle swarm optimization* (PSO) serta *genetic algorithm* (GA)

Kata Kunci: *cluster* otomatis, *novel modified differential evolution*, *k-means*

I. Latar Belakang

Perkembangan Teknologi dan Sistem Informasi yang sangat cepat telah menyebabkan tersedianya data dan informasi dalam skala besar. Oleh karena itu, agar data dan informasi yang telah tersimpan dapat memberikan nilai tambah, maka perlu diterapkan teknik data mining. Diantara teknik data mining tersebut, analisa *cluster* merupakan salah satu metode yang penting.

Analisa *cluster* merupakan pengelompokan data, item, atau ciri secara tidak terbimbing berdasarkan kemiripan yang dimilikinya. Tujuannya adalah mengelompokkan data dengan kesamaan yang tinggi pada satu kelompok dan data dengan kesamaan yang rendah pada kelompok lainnya[1]. Semakin tinggi kesamaan (homogen) data maka semakin baik hasil dari pengelompokan. Terdapat banyak aplikasi dan bidang ilmiah yang telah menggunakan algoritma pengelompokan seperti segmentasi pasar[2], segmentasi citra[3], bioinformatika[4], *machine learning*[5], pengelompokan dokumen[6] dan lainnya.

Pada umumnya metode pengelompokan, seperti *k-means*, memerlukan pengguna untuk menentukan jumlah kelompok akhir, *K*, pada awal algoritma (*non-automatic clustering*).

Akan tetapi, biasanya pengguna tidak tahu berapa jumlah kelompok akhir yang terbaik. Sedangkan, jumlah kelompok akhir akan menentukan kualitas output dari metode pengelompokan. Semakin kecil jumlah kelompok yang digunakan, maka akan mempermudah pemahaman terhadap struktur data didalamnya, tetapi pola informasi penting yang ada akan terabaikan. Sebaliknya, dengan memberikan jumlah kelompok akhir yang lebih besar, maka kesamaan yang dimiliki antar kelompok akan semakin meningkat pula dan mengabaikan struktur data yang ada[7]. Oleh karena itu penentuan jumlah kelompok akhir merupakan tugas yang sangat menantang dan menjadi salah satu isu penelitian pada metode pengelompokan yang menjanjikan.

Pengelompokan data otomatis (*automatic clustering*) merupakan sebuah teknik yang dapat membantu dalam menentukan jumlah kelompok akhir, *K*, sekaligus melakukan partisi data dengan menggunakan pengukuran validitas kelompok[7, 8]. Saat ini, telah banyak metode pemrograman heuristik seperti *genetic algorithm* (GA)[9], *particle swarm optimization* (PSO)[10], *artificial bee colony* (ABC)[11], *tabu search*[12] dan *differential evolution* (DE)[8, 13] yang telah diaplikasikan untuk

memecahkan pengelompokan data secara otomatis.

Differential evolution merupakan algoritma evolusi untuk optimasi global [14]. Pengkodean populasinya menggunakan bilangan real dan telah banyak berhasil diterapkan pada metode pengelompokan. Pada penelitian yang dilakukan oleh Paterlini dan Krink [15], algoritma DE dapat memberikan hasil yang lebih baik dibandingkan pengelompokan menggunakan algoritma GA dan algoritma PSO pada metode pengelompokan non otomatis. Sedangkan pada pengelompokan otomatis, Das et al. [8] mengusulkan algoritma *automatic clustering based differential evolution* (ACDE) dengan melakukan modifikasi *scale factor*, F , pada DE asli dengan bilangan random pada *range* [1, 0.5] dan menurunkan *crossover rate*, CR , sejalan dengan iterasi. Untuk meningkatkan performa algoritma DE, Zhihua [16] dan Kwedlo mengkombinasikan DE dan algoritma k -means untuk memperbaiki fungsi DE pada saat melakukan pencarian *centroid* (pusat kelompok) terbaik. Untuk meningkatkan performa DE pada permasalahan optimasi yang lebih kompleks, Zou et al. [17] mengajukan *a novel modified differential evolution algorithm for constrained optimization problems* (NMDE). Algoritma NMDE ini akan melakukan penyesuaian *scale factor* dan *crooover rate* berdasarkan informasi dari semua solusi yang berhasil apabila berada pada kondisi stagnan, SP . Kondisi stagnan adalah suatu kondisi dimana dalam beberapa kali iterasi, misalkan 50 iterasi, solusi yang lebih baik tidak dapat diberikan. Nilai SP ini dapat diubah untuk mengetahui nilai SP yang terbaik pada suatu kasus tertentu.

Dengan peningkatan performa algoritma DE yang diberikan oleh algoritma NMDE, maka pada penelitian ini penulis mengajukan penerapan algoritma NMDE yang digabung dengan algoritma k -means (NMDE- k -means) untuk pengelompokan otomatis. Kombinasi NMDE dan k -means akan dapat menyeimbangkan pencarian global yang dilakukan NMDE dan pencarian lokal oleh k -means.

Penelitian ini diorganisasikan sebagai berikut. Bagian dua memberikan penjelasan mengenai dasar teori. Bagian tiga membahas algoritma NMDE- k -means yang diajukan. Sedangkan pada bagian empat membahas mengenai hasil penelitian menggunakan data set Iris, Wine, Glass dan Vowel. Bagian lima membahas mengenai kesimpulan.

II. Dasar Teori

Bagian ini akan membahas latar belakang analisa pengelompokan secara ringkas.

Analisa Pengelompokan

Suatu data set $X = \{x_1, x_2, \dots, x_N\}$ mempunyai anggota N data objek yang mempunyai atribut d dimensi. Metode pengelompokan akan mempartisi data yang tidak mempunyai nama kelompok tersebut kedalam sejumlah K kelompok $C = \{C_1, C_2, \dots, C_K\}$ berdasarkan kesamaan yang dimiliki. Pada metode pengelompokan data non otomatis, pengguna harus menentukan jumlah kelompok akhir yang akan dibentuk. Sedangkan pada pengelompokan otomatis, algoritma dapat memberikan jumlah kelompok akhir tanpa bantuan pengguna berdasarkan pengukuran validitas kelompok tertentu [7, 18]. Oleh karena itu, pemilihan pengukuran validitas kelompok yang tepat akan sangat mempengaruhi hasil dari metode pengelompokan mengingat hasil dari data mining juga dipengaruhi oleh data yang akan diolah.

Pada dasarnya metode pengelompokan dibagi menjadi dua: metode *hierarchical* dan *partitional* [1]. Metode *hierarchical* bekerja dengan mode *agglomerative* atau *divisive*. Metode *agglomerative* menganggap setiap data adalah sebuah kelompok kemudian menggabungkan dua kelompok yang saling berdekatan kedalam kelompok yang sama secara berulang sampai suatu kriteria tertentu terpenuhi. Mode *divisive* merupakan kebalikan dari mode *agglomerative* yaitu dimulai dari satu kelompok besar yang dipecah menjadi beberapa kelompok yang lebih kecil sampai suatu kriteria terpenuhi. Sedangkan metode *partitional* mencoba mencari kelompok secara simultan.

Algoritma pengelompokan juga dapat dibagi menjadi pengelompokan *crisp* dan *fuzzy*. Pada pengelompokan algoritma *crisp*, sebuah data hanya boleh menjadi anggota dari satu kelompok saja. Sedangkan pada algoritma pengelompokan *fuzzy*, sebuah data dapat menjadi anggota dari beberapa kelompok dengan nilai derajat keanggotaan tertentu. Penelitian ini membatasi pembahasan pada pengelompokan otomatis untuk metode *partitional*.

Pengukuran Validitas Pengelompokan

Indek pengukuran validitas kelompok mempunyai dua peranan penting: menentukan jumlah kelompok akhir dan membagi data

kedalam beberapa kelompok berdasarkan kesamaan. Ada banyak pengukuran validitas pengelompokan yang telah diajukan pada penelitian sebelumnya. Namun pada penelitian ini hanya membahas pengukuran validitas *VI Index* untuk pengelompokan otomatis metode *partitional*.

1. *VI Index*

Indek pengukuran ini bekerja dengan menghitung perbedaan antara *intra* dan *inter* kelompok [7].

$$VI = (c \times N(0,1) + 1) \times \frac{\text{intra}}{\text{inter}}$$

dimana,

- 1) $(c \times N(0,1) + 1)$ adalah nilai hukuman supaya tidak menghasilkan jumlah kelompok yang terlalu kecil,
- 2) c adalah nilai konstan yang bernilai 30.
- 3) $N(0,1)$ adalah fungsi Gaussian.

Intra adalah jarak rata-rata antara pusat kelompok, m_k , dengan sebuah data, x , dalam suatu kelompok. Hitung jarak sebuah data ke semua pusat kelompok. Kemudian dapatkanlah jarak terpendek, jumlahkan semua jarak terpendek yang diperoleh tersebut dan bagi dengan total data, N_p .

$$\text{intra} = \frac{1}{N_p} \sum_{k=1}^K \sum_{x \in C_k} \|x - m_k\|^2$$

Sedangkan *inter* adalah jarak terkecil antara pusat kelompok satu dengan pusat kelompok lainnya.

$$\text{inter} = \min \{d(\bar{m}_k, \bar{m}_{kk})\}$$

$$\forall k = 1, 2, \dots, K-1 \text{ and } kk = k+1, \dots, K.$$

Algoritma *k-means*

k-means merupakan algoritma pengelompokan yang sangat terkenal. Berikut ini merupakan langkah-langkah algoritma *k-means* oleh Macqueen [19].

1. Pilihlah sejumlah k data awal secara acak untuk dijadikan pusat kelompok.
2. Kelompokkan semua data pada pusat kelompok yang terdekat
3. Hitung ulang pusat kelompok pada setiap kelompok yang ada dengan mendapatkan rata-rata semua data yang ada pada kelompok yang sama.
4. Ulangi langkah 2 dan 3 sampai suatu kriteria terpenuhi.

Pengelompokan Otomatis

Metode pengelompokan otomatis merupakan salah satu algoritma pengelompokan yang tidak memerlukan input jumlah kelompok akhir dari pengguna dan dapat mempartisi data secara otomatis pada kelompok yang mempunyai kesamaan lebih besar. Pada awal proses, algoritma ini akan secara acak menentukan jumlah kelompok awal beserta pusat kelompoknya. Selanjutnya secara berulang, jumlah kelompok dan pusat kelompok tersebut akan diperbarui dengan menggunakan bantuan pengukuran validitas sehingga menghasilkan jumlah kelompok akhir dan anggota bagi setiap kelompok. Jumlah kelompok yang mungkin didapatkan berkisar antara 2 dan k_{\max} [20]. Untuk mencapai tujuan tersebut, metode heuristic banyak diterapkan pada pengelompokan otomatis ini.

Differential Evolution

Algoritma DE diajukan oleh Storn dan Price [14]. Seperti algoritma evolusi lainnya, populasi awal, $V_{i,d}(t)$, dibangkitkan secara acak. Sebuah individu vektor (kromosom), dari suatu populasi pada suatu waktu (generasi), t , mempunyai atribut-atribut, d , (dimensi) yang dapat diformulasikan pada persamaan (4).

$$V_{i,d}(t) = v_{i,1}(t), v_{i,2}(t), \dots, v_{i,d}(t) \quad (4)$$

Mutasi adalah proses untuk membangkitkan parameter baru dengan menambahkan perbedaan bobot dua buah vektor populasi kepada populasi vektor ketiga. *Mutant vector*, $Z_{i,d}(t+1)$, yang diilustrasikan pada persamaan (5) diperoleh dari tiga vektor secara acak, misalkan $V_{j,d}(t)$, $V_{k,d}(t)$, dan $V_{l,d}(t)$, dari generasi yang sama (iterasi i, j, k dan l) berdasarkan *scale factor*. F .

$$Z_{i,d}(t+1) = V_{j,d}(t) + F(V_{k,d}(t) - V_{l,d}(t)) \quad (5)$$

Trial vector, $U_{j,i,d}(t+1)$, pada persamaan (6) dibangkitkan untuk meningkatkan keragaman parameter vektor dengan menggunakan operator mutasi.

$$U_{j,i,d}(t+1) = \begin{cases} Z_{i,d}(t+1) & \text{if } \text{rand}_j(0,1) \leq CR \text{ or } j = \text{rand}(d) \\ V_{i,d}(t) & \text{if } \text{rand}_j(0,1) > CR \text{ or } j \neq \text{rand}(d) \end{cases} \quad (6)$$

Proses seleksi dilakukan dengan membandingkan fungsi tujuan dengan nilai *offspring* yang dihasilkan algoritma DE.

Apabila nilai *offspring* baru, $U_i(t+1)$, lebih baik (minimum) dari fungsi tujuan, $f(V_i(t))$, maka *offspring* tersebut akan menjadi vektor populasi baru pada generasi berikutnya. Sebaliknya, apabila nilai *offspring* tidak lebih baik, maka *parent*, $V_i(t)$, akan tetap digunakan pada generasi selanjutnya. Aturan tersebut dapat dituliskan pada persamaan (7) berikut ini.

$$V_i(t+1) = \begin{cases} U_i(t+1), & \text{if } f(U_i(t+1)) > f(V_i(t)) \\ V_i(t), & \text{if } f(U_i(t+1)) \leq f(V_i(t)) \end{cases} \quad (7)$$

Novel Modified Differential Evolution

Zou et al. [17] mengajukan *a novel modified differential evolution algorithm for constrained optimization problems* (NMDE) untuk permasalahan optimasi yang lebih kompleks. Peningkatan performa DE diupayakan dengan melakukan peningkatan *offspring* yang sukses pada saat proses seleksi. Apabila sebuah solusi ke-*i* ($i= 1, 2, \dots, M$) pada suatu generasi tidak mendapatkan nilai yang lebih baik dari *parent*, maka nilai *F* dan *CR* akan dibangkitkan kembali berdasarkan informasi dari semua nilai *F* dan *CR* yang sukses dengan nilai interval pada persamaan (8). Nilai δ_f (δ_{CR}) ditentukan diawal algoritma dan F_m (CR_m) dihitung menggunakan persamaan (9,10).

$$[F_m - \delta_F, F_m + \delta_F][CR_m - \delta_{CR}, CR_m + \delta_{CR}]$$

$$F_m = \frac{\sum_{F \in S_F} F}{N_{S_F}}$$

$$CR_m = \frac{\sum_{CR \in S_{CR}} CR}{N_{S_{CR}}}$$

dimana,

- 1) S_F adalah himpunan nilai *F* yang sukses pada suatu iterasi
- 2) S_{CR} adalah himpunan nilai *CR* yang sukses pada suatu iterasi
- 3) N_{S_F} adalah total *F* yang sukses pada suatu iterasi
- 4) $N_{S_{CR}}$ adalah total *CR* yang sukses pada suatu iterasi

III. Algoritma Novel Modified Differential Evolution dan k-means untuk Pengelompokan Otomatis

Algoritma NMDE-*k*-means untuk pengelompokan otomatis yang diajukan pada penelitian ini adalah sebagai berikut :

- 1) Inisialisasi setiap kromosom mengandung *k* (dibangkitkan secara acak) sejumlah

pusat kelompok yang dipilih secara acak dan sejumlah, *k*, *activation threshold* [0,1] secara acak pula.

- 2) Dapatkan semua pusat kelompok yang aktif $v_{i,k}T_k$ pada setiap kromosom dengan menggunakan aturan pada persamaan (11)
- 3) Mulai dari $t=1$ sampai t_{max} lakukan
 - a. Hitung jarak untuk setiap data vektor kepada semua pusat kelompok yang aktif dari kromosom ke-*i*
 - b. Masukkan vektor data pada kelompok yang mempunyai jarak terdekat.
 - c. Untuk semua vektor solusi Terapkan algoritma NMDE
 - a) Perbarui keanggotaan populasi dengan menggunakan algoritma DE gunakan persamaan (5,6)
 - b) Gunakan pengukuran validitas pengelompokan pada persamaan (1) sebagai fungsi tujuannya.
 - c) Lakukan pengecekan *offspring*. Jika nilai *offspring* lebih besar dari fungsi tujuan dan iterasi lebih besar atau sama dengan 50, maka perbaharui *F* dan *CR* menggunakan persamaan (9,10)
 - d. Terapkan algoritma *k*-means untuk menyeimbangkan pencarian pusat kelompok untuk kromosom yang aktif.
- 4) Tampilkan hasil dari kromosom yang mempunyai nilai fungsi tujuan terkecil.

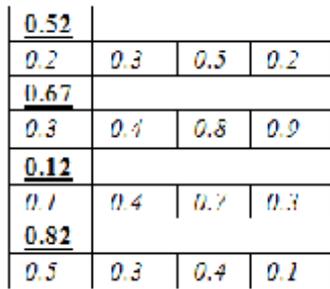
Pada penelitian ini, skema kromosom DE yang digunakan berdasarkan pada penelitian Kuo et al [13]. Setiap kromosom, $v_i(t)$, terdiri dari dimensi dan *activation threshold*, $v_{i,k}T_k + (v_{i,k} \times d_n)$. Nilai dari *activation threshold* diacak pada range [0,1] yang berfungsi sebagai parameter kontrol untuk mengetahui suatu kelompok aktif atau tidak aktif.

IF $v_{i,k}T_k > 0.5$, THEN the *k*th cluster center $v_{i,k}m_k$ is ACTIVE
 ELSE $v_{i,k}m_k$ is INACTIVE

(11)

	$v_{i,1}T_1$	$v_{i,1}d_1$	$v_{i,1}d_2$...	$v_{i,1}d_n$
$v_{i,1}$	$v_{i,1}m_1$	$v_{i,2}d_1$			
	$v_{i,2}T_2$	$v_{i,2}d_2$			
$v_{i,2}$	$v_{i,2}m_2$	$v_{i,2}d_1$	$v_{i,2}d_2$		$v_{i,2}d_n$
...
	$v_{i,k}T_k$				
$v_{i,k}$	$v_{i,k}m_k$	$v_{i,k}d_1$	$v_{i,k}d_2$...	$v_{i,k}d_n$

(a)



Gambar 1. (a) detail skema kromosom, (b) Sebuah vektor dengan empat buah *activation threshold* (bold underlined) dengan empat pusat kelompok (underlined)

IV. Hasil Percobaan

Empat data set yaitu Iris, Wine, Glass dan Vowel digunakan untuk memvalidasi algoritma NMDE-*k*-means. Pengukuran akurasi algoritma juga dilakukan dengan menjalankan algoritma sebanyak 30 kali. Parameter DE yang digunakan mengacu pada penelitian Zou et al. Sedangkan jumlah kelompok awal, *k*, menggunakan rumus $\sqrt{N_{pop}}$. Lebih jauh lagi, hasil NMDE-*k*-menas dibandingkan dengan algoritma *improved automatic clustering based differential evolution* (ACDE), *automatic clustering using differential evolution and k-means* (ACDE-*k*-means) dan algoritma cluster otomatis yang berbasis *particle swarm optimization* (PSO) serta *genetic algorithm* (GA).

Tabel 1. Informasi data set *Becnhmark*

Set data	Jumlah Data	Dimensi	Jumlah Kelompok	Jumlah anggota pada setiap kelompok
Iris	150	4	3	50,50,50
Wine	178	13	3	59,71,48
Glass	214	9	6	70,76,17,13,9,29
Vowel	871	3	6	72,89,172,151, 207,180

Tabel 2. Parameter DE

Parameter	Nilai
Jumlah populasi	10 x dimensi
<i>F</i> (ACDE)	0.5 : 1.0
<i>F</i> (NMDE- <i>k</i> -means)	$\delta_F = 0.2$
<i>CR</i> (ACDE)	0.5 : 1.0
<i>CR</i> (NMDE- <i>k</i> -means)	$\delta_{CR} = 0.1$
<i>SP</i> (NMDE- <i>k</i> -means)	50
$k_{min} ; k_{max}$	2; $\sqrt{N_{pop}}$

Tabel 3. Jumlah kelompok

Set data	Algoritma	Rata-rata jumlah kelompok \pm SD	VI Index \pm SD
Iris	NMDE- <i>k</i> -means	3.0123 \pm 0.1654	0.2635 \pm 0.0193

	ACDE	2.8333 \pm 0.4611	0.0526 \pm 0.0221
	ACDE- <i>k</i> -means	3.0667 \pm 0.2537	0.2460 \pm 0.0089
	DCPG	3 \pm 0	0.3587 \pm 0.0144
Wine	NMDE- <i>k</i> -means	3.0562 \pm 0.4153	0.1423 \pm 0.0241
	ACDE	2.8667 \pm 0.4342	0.0264 \pm 0.0087
	ACDE- <i>k</i> -means	3.2333 \pm 0.4302	0.1172 \pm 0.0218
	DCPG	3.7667 \pm 0.4229	0.1172 \pm 0.0218
Glass	NMDE- <i>k</i> -means	6.0267 \pm 0.1645	0.3435 \pm 0.0162
	ACDE	5.5333 \pm 0.6288	0.1854 \pm 0.0322
	ACDE- <i>k</i> -means	6.0667 \pm 0.2537	0.3574 \pm 0.0103
	DCPG	5.9333 \pm 0.7717	0.4699 \pm 0.0484
Vowel	NMDE- <i>k</i> -means	6.3765 \pm 0.8674	0.4324 \pm 0.0123
	ACDE	6.2667 \pm 1.0806	0.5040 \pm 0.0506
	ACDE- <i>k</i> -means	6.4333 \pm 0.7739	0.4560 \pm 0.0154
	DCPG	5.7667 \pm 0.8584	0.4930 \pm 0.0308

Keterangan : nilai yang lebih baik dicetak tebal

Pada penelitian ini, fokus utama adalah pada penentuan jumlah kelompok dan bilai *VI index* oleh setiap algoritma. Sehubungan dengan jumlah kelompok akhir, nilai terbaik berdasarkan rata-rata jumlah kelompok akhir yang mendekati data benchmark. Pada tabel 3, tampak bahwa algoritma NMDE-*k*-mean menghasilkan jumlah kelompok yang terbaik pada data set Wine dan Glass.

V. Kesimpulan

Pada penelitian ini, algoritma NMDE-*k*-means telah diterapkan pada pengelompokan otomatis. Performa pencarian pusat kelompok yang dilakukan algoritma DE ditingkatkan dengan memberikan nilai *F* dan *CR* baru dari informasi nilai *F* dan *CR* yang sukses pada suatu generasi apabila terjadi kondisi stagnan. Sehingga pada proses seleksi mampu memberikan solusi baru yang lebih beragam dan dapat terhindar dari lokal optima. Disamping itu, algoritma *k*-means akan membantu keseimbangan algoritma NMDE dalam proses evolusi pencarian pusat data kelompok terbaik. Hasil pengukuran oleh *VI Index* menunjukkan algoritma NMDE-*k*-means lebih baik dari empat algoritma pengelompokan otomatis lainnya.

Referensi

[1] Jain, A.K., Murty, M.N., dan Flynn, P.J. 1999. Data clustering: a review. *ACM computing surveys (CSUR)*. **31**(3): 264-323.

[2] Kuo, R.J., Akbaria, K., dan Subroto, B. 2012. Application of particle swarm optimization and perceptual map to tourist market segmentation. *Expert Systems with Applications*. **39**(10): 8726-8735.

[3] Frigui, H. dan Krishnapuram, R. 1999. A robust competitive clustering algorithm with applications in computer vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. **21**(5): 450-465.

[4] Higham, D.J., Kalna, G., dan Kibble, M. 2007. Spectral clustering and its use in

- bioinformatics. *Journal of Computational and Applied Mathematics*. **204**(1): 25-37.
- [5] Al-Omary, A.Y. dan Jamil, M.S. 2006. A new approach of clustering based machine-learning algorithm. *Knowledge-Based Systems*. **19**(4): 248-258.
- [6] Guerrero-Bote, V.P., López-Pujalte, C., de Moya-Anegón, F., dan Herrero-Solana, V. 2003. Comparison of neural models for document clustering. *International Journal of Approximate Reasoning*. **34**(2-3): 287-305.
- [7] Kuo, R.J., Syu, Y.J., Chen, Z.-Y., dan Tien, F.C. 2012. Integration of particle swarm optimization and genetic algorithm for dynamic clustering. *Information Sciences*. **195**(0): 124-140.
- [8] Das, S., Abraham, A., dan Konar, A. 2008. Automatic Clustering Using an Improved Differential Evolution Algorithm. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*. **38**(1): 218-237.
- [9] He, H. dan Tan, Y. 2012. A two-stage genetic algorithm for automatic clustering. *Neurocomputing*. **81**(0): 49-59.
- [10] Das, S., Abraham, A., dan Konar, A. 2008. Automatic kernel clustering with a Multi-Elitist Particle Swarm Optimization Algorithm. *Pattern Recognition Letters*. **29**(5): 688-699.
- [11] Kuo, R.J., Huang, Y.D., Lin, C.-C., Wu, Y.-H., dan Zulvia, F.E. 2014. Automatic kernel clustering with bee colony optimization algorithm. *Information Sciences*. **283**(0): 107-122.
- [12] Shih-Ming, P. dan Kuo-Sheng, C. 2007. Evolution-Based Tabu Search Approach to Automatic Clustering. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*. **37**(5): 827-838.
- [13] Kuo, R.J., Suryani, E., dan Yasid, A., 2013. *Automatic Clustering Combining Differential Evolution Algorithm and k-Means Algorithm*, in *Proceedings of the Institute of Industrial Engineers Asian Conference 2013*, Y.-K. Lin, Y.-C. Tsao, and S.-W. Lin, Editors. Springer Singapore. Hal. 1207-1215.
- [14] Storn, R. dan Price, K. 1997. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. of Global Optimization*. **11**(4): 341-359.
- [15] Paterlini, S. dan Krink, T. 2006. Differential evolution and particle swarm optimisation in partitioned clustering. *Comput. Stat. Data Anal.* **50**(5): 1220-1247.
- [16] Kwedlo, W. 2011. A clustering method combining differential evolution with the K-means algorithm. *Pattern Recognition Letters*. **32**(12): 1613-1621.
- [17] Zou, D., Liu, H., Gao, L., dan Li, S. 2011. A novel modified differential evolution algorithm for constrained optimization problems. *Computers & Mathematics with Applications*. **61**(6): 1608-1623.
- [18] Chou, C.-H., Su, M.-C., dan Lai, E. 2004. A new cluster validity measure and its application to image compression. *Pattern Anal. Appl.* **7**(2): 205-220.
- [19] MacQueen, J. 1967. *Some methods for classification and analysis of multivariate observations*. in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press.
- [20] Abdule-Wahab, R., Monmarché, N., Slimane, M., Fahdil, M., dan Saleh, H., *A 2006. Scatter Search Algorithm for the Automatic Clustering Problem*, in *Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining*, P. Perner, Editor. Springer Berlin Heidelberg. Hal. 350-364.