

## **FUZZY CLUSTERING MENGGUNAKAN ALGORITM FIREFLY-FUZZY C-MEANS DENGAN JARAK MAHALANOBIS**

**Joshua Ariel Perkasa<sup>1</sup>, Robert Kurniawan<sup>2</sup>**

Jurusan Statistika, Politeknik Statistika STIS - Jakarta

e-mail: <sup>1</sup>joshuaarielp@gmail.com, <sup>2</sup>robertk@stis.ac.id

### **ABSTRAK**

Fuzzy C-Means (FCM) adalah salah satu teknik clustering yang cukup sering digunakan, tetapi memiliki kelemahan yaitu mudah terjebak ke dalam local optima. Hal ini dikarenakan adanya faktor pengambilan pusat cluster yang awalnya random sehingga terjadi inkonsistensi pada saat memulai FCM. Firefly Algorithm (FA) mampu mengatasi ketidak konsistenan dari FCM. Penelitian ini bertujuan untuk melihat performa dari Firefly Algorithm Fuzzy C-Means (FAFCM) dengan pendekatan jarak mahalanobis dibandingkan dengan jarak euclidean. Algoritme FAFCM ini dibangun dengan 2 jenis jarak tersebut untuk mengakomodir berbagai jenis persebaran data. FAFCM memiliki performa yang lebih baik dikarenakan sebagian besar nilai iterasi dari FAFCM lebih kecil dari FCM. FAFCM Mahalanobis sendiri menunjukkan nilai fungsi objektif paling minimum untuk jenis data hyperspherical sehingga dapat disimpullkan FAFCM Mahalanobis cocok untuk data hyperspherical.

**Kata kunci :** *Clustering, Fuzzy C-Means, Mahalanobis, Firefly Algorithm Optimization-Fuzzy C-Means.*

### **PENDAHULUAN**

*Clustering* merupakan proses pengelompokan data sehingga dapat memberikan informasi yang lebih bermanfaat. Metode *clustering* itu sendiri memiliki beberapa tipe yang dapat diklasifikasikan sebagai berikut *hierarchical versus partitional, exclusives versus overlapping versus fuzzy*, dan *complete versus partial* [6]. *Fuzzy Clustering* memiliki keunggulan dalam pengelompokan. Kemunculan *fuzzy clustering* dipicu dengan adanya ketidakpastian dari *clustering* biasa yang dipengaruhi oleh *noise* (data-data yang terlihat sumbang atau pencilan) dan *fuzzy clustering* bertujuan membuat *clustering* lebih alami [8].

FCM merupakan salah satu metode dari fuzzy clustering. FCM sendiri

memang cukup populer digunakan dalam berbagai bidang aplikasi seperti MRI [7], *image recognition* [8], dan *data clustering* [11]. Namun kelemahan dari FCM adalah sering kali terperangkap dalam optimum lokal dan inkonsistensi akibat pengambilan matriks keanggotaan awal yang dilakukan secara acak.

Optimasi FCM dengan *firefly algorithm* (FA) yang diperkenalkan oleh Xin-she Yang mampu mengatasi kelemahan FCM yaitu seringnya terjebak dalam optimum lokal. FA mengatasi permasalahan FCM dengan mengkonsistenkan pengambilan matriks keanggotaan di awal. FA sendiri merupakan algoritme optimasi yang mengikuti pergerakan kunang-kunang menuju kunang-kunang lainnya.

Pada penelitian ini peneliti menggunakan 2 ukuran jarak. Ukuran

jarak yang digunakan adalah Mahalanobis dan Euclidean. Penggunaan kedua ukuran jarak ini digunakan untuk mengatasi bentuk data yang memiliki persebaran *hyperellipsoidal* dan *hyperspherical*. Dengan adanya kedua ukuran jarak ini FAFCM menjadi bisa lebih fleksibel dalam melakukan pengelompokan pada berbagai jenis data.

Pada penelitian ini akan dilakukan perbandingan antara FAFCM dengan FCM. Parameter FCM yang digunakan sebagai pembandingan pada kedua metode akan ditentukan dengan berbagai percobaan. Setelah dilakukan berbagai percobaan seluruh hasil dan parameter akan dilihat mana yang paling baik dengan menggunakan 7 indeks validitas. Ketujuh indeks validitas ini adalah indeks Partition Coefficient (PC), Modified Partition Coefficient (MPC), Classification Entropy (CE), Partition Index (PI), Fukuyama Sugeno Index (FS), dan Xie-Beni Index (XB).

### Fuzzy C-Means

*Fuzzy C-Means* (FCM) merupakan algoritma *Fuzzy Clustering* yang dikembangkan oleh Bezdek pada tahun 1981 [9]. FCM merupakan pengembangan dari *hard c-means* oleh Dunn pada tahun 1973 [10]. Ukuran jarak yang digunakan adalah Euclidean. Berikut adalah fungsi objektif yang terdapat pada fuzzy C-Means:

$$J_w(U, A, X) = \sum_{i=1}^n \sum_{k=1}^c (\mu_{ik})^w \text{dist}^2(x_i, a_k)$$

$$\sum_{k=1}^c \mu_{ik} = 1$$

$$\text{dist}^2(x_i, a_k) = (x_i - a_k) \cdot (x_i - a_k)'$$

Dengan ketentuan :

$$U_{n \times c} = \begin{bmatrix} \mu_{1,1} & \mu_{2,1} & \dots & \mu_{1,c} \\ \mu_{2,1} & \mu_{2,2} & \dots & \mu_{2,c} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n,1} & \mu_{n,2} & \dots & \mu_{n,c} \end{bmatrix}$$

$$\mu_{ik} \in [0, 1]$$

$$A_{c \times m} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,m} \\ a_{2,1} & a_{2,2} & \dots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{c,1} & a_{c,2} & \dots & a_{c,m} \end{bmatrix}$$

$$X_{n \times m} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,m} \\ x_{2,1} & x_{2,2} & \dots & x_{2,m} \\ \vdots & \vdots & \ddots & \dots \\ x_{n,1} & x_{n,2} & \dots & x_{n,m} \end{bmatrix}$$

$$i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m; \\ k = 1, 2, \dots, c$$

$J_w(U, A, X)$  Merupakan fungsi obyektif. Dimana fungsi ini akan menunjukkan seberapa baiknya *cluster* itu terbentuk.  $U$  merupakan matriks yang berisikan nilai keanggotaan setiap data.  $A$  merupakan pusat cluster yang terbentuk.  $X$  adalah data yang akan dilakukan proses FCM.  $\mu_{ik}$  pada rumus diatas menunjukkan nilai keanggotaan objek  $x_i$  pada *cluster*  $k$ . Simbol  $c$  menunjukkan jumlah cluster yang ingin dibentuk. Simbol  $n$  menunjukkan banyaknya jumlah objek pada data. Jumlah atribut ditunjukkan dengan simbol  $m$ . Tingkat kekaburan disimbolkan dengan  $w$ . Vektor pusat cluster dapat dilihat dengan simbol  $a_k$ . Vektor baris data ke-  $i$  dapat dilihat dengan  $x_i$ . Dan jarak dilihat dengan simbol  $\text{dist}^2\{(X)_i, (A)_k\}$ .

Proses pengelompokan akan berakhir saat ditemukan kondisi penghentian algoritma. Kondisi ini penghentian ada dua yaitu berupa iterasi maksimal ( $iter_{max}$ ) dan maksimal tingkat *error* ( $\epsilon_{max}$ ) yang diinginkan. Tingkat *error* ( $\epsilon$ ) dilihat dari  $U$  pada iterasi terakhir dan iterasi sebelumnya dengan rumus ( $\epsilon_{iter} = \max|U_{iter} - U_{iter-1}|$ ).

Berikut adalah tahapan penghitungan yang dilakukan pada algoritma FCM [11] :

1. Tentukan parameter awal pengelompokan yaitu jumlah *cluster*  $c$ , *fuzzifier*  $w$ , dan syarat kondisi penghentian iterasi ( $\epsilon_{max}$  dan/atau  $iter_{max}$ ).

2. Bangkitkan pusat cluster pertama sebagai inisiasi dengan cara random.
3. Hitunglah matriks keanggotaan dengan rumus 6.
4. Perbarui nilai pusat cluster dengan rumus 7
5. Hitunglah error yang dihasilkan.
6. Cek kondisi syarat penghentian algoritme. Jika belum terpenuhi, ulangi ke langkah 3-5. Simbol  $t$  pada rumus 6 menunjukkan iterasi yang sudah dijalankan.

$$a_k(t) = \left\{ \sum_{i=1}^n [\mu_{ik}(t-1)]^w \right\}^{-1} \cdot \sum_{i=1}^n [\mu_{ik}(t-1)]^w \cdot x_i$$

$$\mu_{ik}(t) = \left[ \sum_{l=1}^c \left( \frac{\text{dist}^2(x_i, a_k)}{\text{dist}^2(x_i, a_l)} \right)^{\frac{2}{w-1}} \right]^{-1}$$

### Firefly Algorithm

Algoritme *Firefly* merupakan optimasi yang diperkenalkan oleh Xin-She Yang pada akhir 2007 di *Cambridge University*. Algoritme *firefly* ini mengikuti 3 dasar sifat kunang-kunang yaitu [5] :

- o Semua kunang-kunang merupakan *unisex* yang berarti kunang-kunang tertarik pada kunang-kunang lain bukan karena perbedaan sex
- o Ketertarikan setiap kunang-kunang akan dipengaruhi oleh intensitas cahaya yang dihasilkan dan besarnya pergerakan dipengaruhi oleh jauhnya jarak antar kunang-kunang. Jika intensitas cahaya tidak ada yang lebih besar maupun kecil maka kunang-kunang akan bergerak secara random.
- o Intensitas cahaya pada algoritme kunang-kunang ini adalah sebuah fungsi objektif.

Sesuai penjelasan sifat-sifat diatas yang menyatakan *attractiveness* (ketertarikan) setiap kunang-kunang dipengaruhi oleh jarak dan intensitas cahaya, rumus *attractiveness* adalah sebagai berikut:

$$\beta = \beta_0 e^{-\gamma r^2}$$

Dimana  $\beta_0$  adalah *attractiveness* pada saat  $r = 0$ . Simbol  $\gamma$  pada rumus.17 adalah koefisien penyerapan cahaya. Intensitas  $I$  cahaya merupakan fungsi objektif yang ada (pada optimasi dapat diganti dengan fungsi objektif dari algoritme yang akan di optimasi).  $I_0$  adalah intensitas pertama pada iterasi = 0.

$$I = I_0 e^{-\gamma r^2}$$

Jarak  $r$  merupakan ukuran jarak antara dua kunang-kunang dan ukuran jarak yang digunakan adalah ukuran jarak Euclidean.

$$r_{ij} = ||fa_i - fa_j|| \quad \text{atau}$$

$$\sqrt{\sum_{i=1}^n \sum_{j=1}^n (fa_i - fa_j)^2}$$

Dimana  $fa_{ij}$  adalah populasi kunang-kunang yang diinisiasi. Setiap kunang-kunang akan bergerak menuju kunang-kunang lainnya dengan menggunakan rumus pergerakan dibawah ini:

$$fa_i = fa_i + \beta$$

$$= \beta_0 e^{-\gamma r_{ij}^2} (fa_i - fa_j) + \alpha \cdot N_i(0, 1)$$

Penginisiasian kunang-kunang dapat dilakukan dengan rumus dibawah ini :

$$fa_{ij}^{(0)} = (ub_i - lb_i) \cdot rand(0, 1) + lb_i$$

Pada algoritme optimasi rumus inisiasi sendiri dapat diubah sesuai dengan inisiasi algoritme yang akan dioptimasi. Nilai  $\alpha$  sendiri akan berubah-ubah sesuai dengan jumlah iterasi. Berikut adalah pergerakan nilai  $\alpha$ :

$$\Delta = 1 - 10^{-4} / 0.9^{\frac{1}{MAX\_GEN}}$$

$$\alpha^{(t+1)} = 1 - \Delta \cdot \alpha^{(t)}$$

Pada rumus perubahan diatas **MAX\_GEN** adalah nilai parameter yang diinput oleh pengguna dimana iterasi akan dibatasi jumlahnya oleh parameter ini.

*Firefly Algorithm* berjalan dengan tahapan sebagai berikut:

1. Masukkan syarat penghentian, nilai  $\gamma$ , dan nilai *Attractiveness*.
2. Bangkitkan populasi kunang-kunang.
3. Hitunglah nilai  $\alpha$  yang baru dengan rumus diatas eq.(1)
4. Setiap kunang-kunang yang ada akan dihitung nilai intensitasnya dengan fungsi objektif rumus (3). Kemudian lihat manakah yang paling baik nilainya dan diurutkan
5. Hitunglah pergerakan setiap kunang-kunang dimana kunang-kunang akan bergerak menuju cahaya kunang-kunang yang lebih terang. Pergerakan ini akan dihitung dengan rumus (20). Setelah dihitung pergerakannya maka akan didapatkan populasi kunang-kunang yang baru.
6. Lihatlah pada kriteria penghentian apakah sudah tercapai atau belum. Jika belum maka kembali ke tahap no 3.

### Index Validitas

Selain menggunakan fungsi objektivitas kita juga dapat melihat seberapa baiknya sebuah metode pengelompokan dengan indeks validitas. Pada penelitian ini akan digunakan 7 index validitas yaitu *Partition Coefficient* (PC), *Classification Entropy* (CE), *Modified Partition Coefficient* (MPC), *Partition Index* (PI), *Fukuyama Sugeno* (FS), *Xie-Beni* (XB), dan Kwon. Ketujuh index validitas ini nantinya digunakan untuk menentukan parameter fuzzifier yang digunakan untuk perbandingan metode. Berikut adalah penjelasan setiap index validitas beserta kegunaan setiap indeks validitas

- *Partition Coefficient* (PC)

PC merupakan indeks validitas yang dibuat oleh [9] dimana berfungsi untuk tingkat tumpang tindih antar-*cluster*. Berikut adalah rumus dari PC:

$$PC = \frac{1}{n} \sum_{k=1}^c \sum_{i=1}^n (\mu_{ik})^2$$

(Wu & Yang, 2005). Rentang nilai yang dimiliki adalah  $\frac{1}{c} \leq PC \leq 1$ . Semakin besar nilai PC maka hasil *cluster* yang dibentuk semakin baik

- *Classification Entropy* (CE)

Begitu juga dengan PC, CE juga diperkenalkan oleh Bezdek (1981). Fungsinya mirip dengan PC untuk melihat tingkat tumpang tindih antar-*cluster*. Berikut adalah rumus dari CE:

$$CE = -\frac{1}{n} \sum_{k=1}^c \sum_{i=1}^n \mu_{ik} \cdot \ln(\mu_{ik})$$

(Wu & Yang, 2005). Rentang nilai yang dimiliki **CE** adalah  $0 < CE(c) < \ln(c)$ . Nilai yang semakin kecil membentuk *cluster* yang baik.

- *Modified Partition Coefficient* MPC

Dilihat dari rumus PC dan CE yang digunakan dapat dikatakan nilai akan menjadi monoton dikarenakan tidak adanya penimbang jumlah *cluster* [4]. Maka dibuatlah MPC oleh [12]. Berikut adalah rumus MPC:

$$MPC = 1 - \frac{c}{c-1} (1 - PC)$$

Rentang nilai yang dimiliki  $0 \leq MPC \leq 1$  dan semakin besar nilai MPC akan menghasilkan *cluster* yang lebih baik.

- *Partition Index* (PI)

Indeks ini berfungsi untuk melihat rasio antara jumlah kepadatan dan pemisahan antar-*cluster* [1]. PI berguna untuk membandingkan beberapa proses *clustering* yang memiliki ukuran *cluster* yang sama.

$$PI = \sum_{k=1}^c \frac{\sum_{i=1}^n (\mu_{ik})^w \cdot \text{dist}^2(x_i, a_k)}{n_i \cdot \sum_{l=1}^c \{\text{dist}^2(a_l, a_k)\}}$$

Semakin kecil nilai PI, semakin baik partisi *cluster* yang dibuat.

- *Fukuyama Sugeno*

Indeks ini diperkenalkan oleh Fukuyama dan Sugeno pada tahun 1989. Fungsi dari indeks validitas FS adalah sebagai berikut [4] :

$$\begin{aligned}
 FS &= J_w(U, A, X) - K_w(U, A, X) \\
 FS &= \sum_{k=1}^c \sum_{i=1}^n (\mu_{ik})^w \{dist^2(x_i, a_k)\} \\
 &\quad - \sum_{k=1}^c \sum_{i=1}^n (\mu_{ik})^w dist^2(a_k, \bar{a})
 \end{aligned}$$

Nilai indeks *FS* semakin besar maka menghasilkan *cluster* semakin baik.

- Xie-Beni

*Xie-Beni* menghitung ke validitasan proses *clustering* adalah dengan membandingkan antara kepadatan dan keterpisahan pada sebuah *clustering* berikut adalah rumus *Xie-Beni* [3] :

$$XB = \frac{\sum_{k=1}^c \sum_{i=1}^n (\mu_{ik})^2 dist^2(x_i, a_k)}{n \cdot \min_{kl} \{dist^2(a_k, a_l)\}}$$

Semakin besar nilai *XB* maka hasil *clustering* semakin baik [4].

- Kwon

Indeks ini dikembangkan oleh [2]. Indeks ini digunakan untuk mengatasi kecendrungan beberapa indeks validitas yang cenderung monoton turun mengikuti jumlah banyaknya *cluster* dan *fuzzifier*. Indeks ini merupakan modifikasi dari *XB* dengan adanya pertimbangan nilai dari data.

**Kwon**

$$= \frac{\sum_{k=1}^c \sum_{i=1}^n (\mu_{ik})^2 dist^2(x_i, a_k) + \frac{1}{c} \sum_{k=1}^c dist^2(a_k, \bar{a})}{\min_{kl} \{dist^2(a_k, a_l)\}}$$

$$\text{dengan } \bar{a} = \sum_{i=1}^n \frac{x_i}{n}$$

Semakin kecil nilai *Kwon* maka semakin baik partisi *cluster* yang dihasilkan.

## METODE PENELITIAN

### Sumber Data dan Variabel Penelitian

Data yang digunakan adalah data bangkitan dengan sebaran data hyperelipsoidal dan Hyper spherical. Kedua data ini diperoleh dari software R melalui fungsi ‘*rnorm*’ pada package *stat(norm{stat})* untuk data hyperelipsoidal dan fungsi ‘*runif*’ untuk data hyper spherical. Jumlah data bangkitan yang digunakan adalah sebanyak 200 observasi dengan 2 variabel dan 4 variabel.

### Metode Analisis

Metode yang akan digunakan adalah FAFCM dan FCM dengan kedua ukuran jarak. Adapun metode FAFCM adalah sebagai berikut:

#### *Firefly Algorithm optimization – Fuzzy C-Means (FAFCM)*

FAFCM dikatakan memiliki optimisasi yang bagus untuk menghindari lokal optimum dimana seorang pengguna pasti ingin mendapatkan global optimum, hal ini membuat FCM yang sering terjebak pada lokal optima menjadi kekurangan yang cukup kerap terjadi. Janmenjoy Nayak, dkk melakukan penelitian dan mengembangkan FCM dengan melakukan optimasi terlebih dahulu dengan algoritme *firefly*. Penelitian tersebut menghasilkan hasil yang baik bahkan nilai fungsi obyektif yang dihasilkan lebih dibandingkan optimasi lainnya seperti PSO-FCM maupun FCM itu sendiri [11]. Pada penelitian ini pun akan menggunakan algoritma yang telah dikembangkan oleh Janmenjoy Nayak, dkk (serta terdapat penambahan pada penentuan jarak. Pada penelitian ini akan ditambahkan penentuan jarak menggunakan jarak Mahalanobis.

Pada penelitian ini akan dilakukan proses penentuan titik dan jumlah *centroid* dengan menggunakan FA dimana hasil terbaik (*global best centroid*) akan digunakan sebagai titik cluster pada FCM, sehingga FCM tidak akan melakukan penentuan *centroid* secara acak. Berikut

adalah tahapan melakukan analisis data menggunakan FAFCM:

Tahapan pertama (Firefly)

1. Menginput parameter:
  - a. Data yang akan dilakukan pengelompokan (data bangkitan, data iris, data *single outlayer*, data Pendidikan)
  - b. Jumlah *fireflies* yang diinginkan
  - c. Nilai *attractiveness* ( $\beta_0$ )
  - d. Nilai *Light absorption coefficient* ( $\gamma$ )
  - e. Kriteria error untuk menghentikan proses atau jumlah iterasi maksimal
2. Membangkitkan  $n$  *firefly* secara random dengan setiap "C" cluster dan "d" dimensi
3. Melakukan iterasi hingga kriteria error tercapai:
  - a. Tentukan nilai intensitas cahaya dari setiap *fireflies* menggunakan fungsi tujuan dari FCM.  $I_i$
  - b. Hitung nilai ketertarikan dari setiap *fireflies*.  $\beta_r$
  - c. Dekatkan *fireflies* dari *firefly* yang bercahaya lebih redup menuju *firefly* dengan cahaya lebih terang menggunakan rumus pergerakannya.

Urutkan peringkat dari setiap *fireflies*, gunakan *fireflies* terbaik dengan nilai error sesuai kriteria dan perbaharui nilai intensitas cahaya untuk dilakukan pengulangan selanjutnya.

Tahapan kedua (FCM)

1. Setelah mendapatkan optimum global cluster dengan metode *firefly* maka dapat kita gunakan centroid setiap cluster yang akan kita gunakan sebagai centroid pada FCM. (anggap ini sebagai centroid pada iterasi  $b = 0$ )
2. Lakukan iterasi dibawah ini dan hentikan iterasi jika  $\|P_{(b+1)} - P_{(b)}\| < \epsilon$  FCM:
  - a. Hitung matriks partisi  $W_b$
  - b. Lakukan pembaharuan centroid  $P_{(b+1)}$

3. Hasil akan berupa cluster.

Metode Analisis

1. Melakukan pencarian parameter untuk membandingkan kedua metode dengan indeks validitas
2. Membandingkan metode FAFCM dan FCM

**HASIL PENELITIAN**

1. Pengelompokan data bangkitan 200 observasi 2 variabel

Pengelompokan dilakukan dengan analisis FAFCM sama seperti 20 observasi yaitu dilakukan sebanyak lima kali untuk setiap parameter *cluster* dan *fuzzifier*. Setelah melakukan lima kali percobaan akan diambil nilai rata-rata dari indeks validitas.

**Tabel 1.** Parameter cluster dan fuzzifier pada analisis FAFCM beserta indeks validitas data bangkitan 200 hyperelipsoidal sejumlah 2 variabel dengan jarak Euclidean

C	2	5	5	5
m	1,5	1,5	1,6	4
PC	0,886	0,832	0,779	0,274
MPC	0,773	0,79	0,724	0,093
CE	0,199	0,334	0,442	1,458
XB	0,207	0,195	0,172	0,01
FS	-13,8	-144	-133,1	-6
PI	0,414	0,091	0,086	0,005
Kwon	38,084	32,467	28,373	60,447

**Tabel 2.** Parameter cluster dan fuzzifier pada analisis FAFCM beserta indeks validitas data bangkitan 200 hyperelipsoidal sejumlah 2 variabel dengan jarak Mahalanobis

C	2	4	5	5	5
m	1,5	1,5	1,5	2	4
PC	0,914	0,896	0,885	0,673	0,28
MP	0,829	0,861	0,856	0,591	0,1
C					
CE	0,155	0,204	0,223	0,684	1,444
XB	1,017	0,364	0,241	0,17	0,017
FS	147,35	102,64	143,16	99,314	4,787

PI	1,873	0,168	0,098	0,071	0,007
Kwo n	177,257	63,909	42,174	35,162	97,885

**Tabel 3.** Parameter cluster dan fuzzifier pada analisis FAFCM beserta indeks validitas data bangkitan 200 hyperspherical sejumlah 2 variabel dengan jarak Euclidean

C	2	5	2	2	5
m	1,5	1,5	2	4	4
PC	0,947	0,815	0,84	0,598	0,269
MP C	0,894	0,768	0,68	0,196	0,087
CE	0,098	0,358	0,276	0,589	1,458
XB	0,106	0,307	0,092	0,033	0,053
FS	- 499, 6	- 810, 1	- 439, 1	- 129, 7	- 34,26 2
PI	0,213	0,097	0,183	0,065	0,005
Kwo n	20,248	51,306	18,601	30,363	319,145

**Tabel 4.** Parameter cluster dan fuzzifier pada analisis FAFCM beserta indeks validitas data bangkitan 200 hyperspherical sejumlah 2 variabel dengan jarak Mahalanobis

C	2	4	5	2	5
m	1,5	1,5	1,5	1,7	4
PC	0,865	0,855	0,838	0,804	0,256
MP C	0,73	0,807	0,798	0,607	0,069
CE	0,227	0,288	0,325	0,32	1,492
XB	0,062	0,161	0,221	0,06	0,017
FS	- 567, 6	- 887, 94	- 933, 62	- 496, 86	- 28,19 2
PI	0,255	0,109	0,101	0,261	0,006
Kwo n	10,905	29,177	40,169	10,702	101,562

2. Pengelompokan data bangkitan 200 observasi 4 variabel

Pengelompokan dilakukan dengan analisis FAFCM sama seperti 200 observasi dengan 2 variabel maka penelitian dilakukan sebanyak lima kali

untuk setiap parameter *cluster* dan *fuzzifier*. Setelah melakukan lima kali percobaan akan diambil nilai rata-rata dari indeks validitas yang dihasilkan. Dari seluruh nilai indeks validitas yang dihasilkan maka akan ditampilkan setiap parameter jumlah cluster dan fuzzifier dengan minimal satu indeks validitas adalah yang terbaik. Setelah itu dilihat manakah parameter yang memiliki nilai indeks validitas terbaik dan terbanyak. Setelah didapatkan parameter tersebut maka akan dilakukan perbandingan antar metode.

**Tabel 5.** Parameter cluster dan fuzzifier pada analisis FAFCM beserta indeks validitas data bangkitan 200 hyperellipsoidal sejumlah 4 variabel dengan jarak Euclidean

C	2	4	5	4	4	5
M	1,5	1,5	1,5	1,8	4	4
PC	0,89	0,861	0,844	0,713	0,298	0,242
MP C	0,779	0,814	0,805	0,617	0,064	0,052
CE	0,194	0,276	0,312	0,572	1,299	1,518
XB	0,348	0,168	0,236	0,149	0,024	0,03
FS	63, 21	- 164	- 196	- 129	- 2,75	- 2,263
PI	0,695	0,148	0,109	0,126	0,014	0,007
Kw on	59, 6	28, 9	39, 8	26, 41	85,687	169,697

**Tabel 6.** Parameter cluster dan fuzzifier pada analisis FAFCM beserta indeks validitas data bangkitan 200 hyperellipsoidal sejumlah 4 variabel dengan jarak Mahalanobis

C	2	5	5	5
W	1,5	1,5	2	4
PC	0,744	0,597	0,318	0,2
MPC	0,488	0,496	0,147	4,03E-09
CE	0,404	0,814	1,364	1,609
XB	1,313	4,705	9,113	2,05E+07
FS	444,22	159,73	112,29	6,368
PI	1,032	0,341	0,279	Inf
Kwo n	207,659	598,751	1830,157	1,03E+11

**Tabel 7.** Parameter cluster dan fuzzifier pada analisis FAFCM beserta indeks validitas data bangkitan 200 hyperspherical sejumlah 4 variabel dengan jarak Euclidean

C	2	3	2	2	4
W	1,5	1,5	1,9	4	4
PC	0,967	0,815	0,87	0,582	0,291
MP	0,934	0,723	0,74	0,165	0,055
CE	0,072	0,317	0,241	0,607	1,301
XB	0,106	1,34	0,098	0,036	4,10E+07
FS	-1008,	1276,7	873,4	173,4	-21,685
PI	0,213	0,192	0,197	0,073	0,009
Kwon	20,682	236,701	19,376	33,743	1,53E+13

**Tabel 8.** Parameter cluster dan fuzzifier pada analisis FAFCM beserta indeks validitas data bangkitan 200 hyperspherical sejumlah 4 variabel dengan jarak Mahalanobis

C	2	5	5
W	1,5	1,5	1,9
PC	0,731	0,587	0,348
MPC	0,461	0,483	0,185
CE	0,421	0,837	1,307
XB	0,113	0,565	0,65
FS	-455,571	-852,832	-280,926
PI	0,497	0,251	0,222
Kwon	17,962	74,565	117,07

Dari seluruh percobaan diatas maka nilai parameter fuzzifier yang terbaik adalah 1,5 penelitian akan dilanjutkan dengan melihat nilai iterasi dan Jm dengan menggunakan parameter fuzzifier 1,5 pada setiap metode dan data bangkitan berikutnya akan dilakukan perbandingan metode antara FAFCM dan FCM berikut dengan kedua ukuran jarak yang digunakan. Perbandingan ini menggunakan parameter fuzzifier 1,5 sesuai dengan uji coba data sebelumnya. Untuk parameter cluster sendiri digunakan jumlah 2 hingga 5 cluster untuk melihat setiap nilai Jm yang terdapat pada

pembentukan 2 hingga 5 cluster dengan parameter fuzzifier 1,5.

**Tabel 9.** Jm keempat metode pada data hyperelipsoid 200 observasi dengan 2 variabel

Cluster	2	3	4	5
Jm FAFCM	113,190	80,975	57,2475	41,7887
Jm FCM	113,190	81,235	57,6502	41,7896
Jm FAFCM Mahalanobis	191,977	122,46	68,4112	51,5454
Jm FCMCM	33,67	22,636	42,445	23,9298

**Tabel 10.** Iterasi keempat metode pada data hyperelipsoid 200 observasi dengan 2 variabel

Iterasi FAFCM	14,8	90	97,4	101
Iterasi FCM	19,2	87,2	101	97,2
Iterasi FAFCM Mahalanobis	17	40,6	24,6	49,6
Iterasi FCMCM	17,4	18	51,8	31,2

**Tabel 11.** Jm keempat metode pada data hyperspherical 200 observasi dengan 2 variabel

Cluster	2	3	4	5
Jm FAFCM	393,530	292,001	219,079	173,55
Jm FCM	393,530	296,212	219,089	173,55
Jm FAFCM Mahalanobis	212,547	126,577	86,9076	70,065
Jm FCMCM	399,543	291,238	184,733	145,35

**Tabel 12.** Iterasi keempat metode pada data hyperspherical 200 observasi dengan 2 variabel

Cluster	2	3	4	5
Iterasi FAFCM	7,8	78,6	91,4	51,2
Iterasi FCM	10	79,6	88,2	51,2
Iterasi FAFCM Mahalanobis	20,4	33,8	34,2	73,6
Iterasi FCMCM	26	44,6	68,6	54,8



**Tabel 13.** Jm keempat metode pada data hyperellipsoidal 200 observasi dengan 4 variabel

Cluster	2	3	4	5
Jm FAFCM	191,301 09	123,5 35	80,41 7	64,04 24
Jm FCM	191,301 09	123,5 42	80,41 7	64,04 24
Jm FAFCM Mahalanobis	529,289 22	416,9 19	345,4 5	298,8 70
Jm FCMCM	191,301 09	123,5 35	80,41 7	64,04 24

**Tabel 14.** Iterasi keempat metode pada data hyperellipsoidal 200 observasi dengan 4 variabel

Cluster	2	3	4	5
Iterasi FAFCM	23, 6	65, 8	35, 2	39, 2
Iterasi FCM	26, 4	78, 2	38, 8	41, 6
Iterasi FAFCM Mahalanobis	23, 8	68	84	83, 8
Iterasi FCMCM	31	81, 2	44	43, 2

**Tabel 15.** Jm keempat metode pada data hyperspherical 200 observasi dengan 4 variabel

Cluster	2	3	4	5
Jm FAFCM	775,10 77	643, 48	522, 37	464,81 409
Jm FCM	775,10 77	643, 48	522, 37	464,81 411
Jm FAFCM Mahalanobis	533,05 42	422, 36	349, 4	302,74 520
Jm FCMCM	903,15 86	737, 42	638, 62	571,19 166

**Tabel 16.** Iterasi keempat metode pada data hyperspherical 200 observasi dengan 4 variabel

Cluster	2	3	4	5
Iterasi FAFCM	6,2	62, 2	65, 4	93
Iterasi FCM	8,8	82, 4	73, 6	95, 6
Iterasi FAFCM Mahalanobis	23, 4	80, 4	82, 4	78, 8
Iterasi FCMCM	10 1	10 1	10 1	10 1

Pada penelitian ini dilakukan uji coba pada setiap parameter jumlah cluster [2:5] dan fuzzifier [1,5:4 dengan penambahan 0,1] dengan mengulang setiap set parameter sebanyak 5 kali dan meratakan nilai indeks validitas dari setiap

perulangan. Setelah dilakukan uji coba pada seluruh persebaran data bangkitan fuzzifier yang terbaik digunakan untuk persebaran ini adalah 1,5. Parameter fuzzifier 1,5 selanjutnya dijadikan sebagai parameter fuzzifier pada setiap metode dengan jumlah cluster 2 hingga 5.

Secara keseluruhan iterasi yang dihasilkan oleh FAFCM memiliki iterasi yang lebih kecil dibandingkan dengan jumlah iterasi yang digunakan pada FCM namun masih terdapat iterasi dengan jumlah lebih besar disbanding dengan FCM. Pada tabel 12 iterasi FCM lebih kecil pada pembentukan *cluster* 4 namun Jm yang dihasilkan oleh FAFCM menunjukkan nilai yang lebih kecil dibandingkan dengan FCM. Begitu juga dengan tabel 10 dimana nilai Iterasi FAFCM menunjukkan nilai yang lebih kecil dibandingkan dengan FCM namun memiliki nilai Jm yang lebih baik dibandingkan dengan FCM. Dengan melihat tabel-tabel diatas dapat disimpulkan bahwa FAFCM mampu menemukan global optimum lebih cepat dibandingkan dengan FCM dan FAFCM mampu menemukan nilai Jm lebih kecil yang berarti mampu mendapatkan letak global optimum dan membuktikan FCM dapat terjebak pada local optimum.

Pada FAFCM mahalanobis nilai Jm pada data Hyperellipsoidal menunjukkan nilai paling besar dan nilai Jm yang paling minimum pada data Hyperspherical. Pada tabel 16 nilai Jm yang dihasilkan adalah paling minimum dan begitu juga dengan nilai iterasinya. Penelitian ini menunjukan bahwa FAFCM Mahalanobis lebih cocok pada data yang memiliki persebaran data Hyperspherical. Pada tabel 16 FAFCM Mahalanobis bahkan mampu menemukan nilai Jm paling minimum dengan iterasi yang paling minimum diantara metode lainnya.

## KESIMPULAN

Berdasarkan penjelasan sebelumnya dapat disimpulkan FAFCM mampu menemukan lokasi global optima lebih cepat. FAFCM juga mampu menemukan nilai  $J_m$  lebih kecil dibandingkan dengan FCM ini menunjukkan bahwa FAFCM mampu menemukan global optimum dan membuktikan bahwa FCM dapat terjebak pada local optima. Untuk perbedaan jarak dengan perbedaan persebaran data dapat disimpulkan bahwa FAFCM Mahalanobis menunjukkan nilai  $J_m$  yang lebih kecil diantara metode lainnya serta memiliki iterasi yang cenderung lebih singkat dibandingkan dengan metode lainnya.

## SARAN

FAFCM memiliki pergerakan randomisasi yang dapat diatur. Pada penelitian peneliti menggunakan pergerakan randomisasi dengan distribusi uniform.

Pada bukunya Xin-She Yang mengatakan bahwa pergerakan randomisasi pada FAFCM dapat diganti dengan berbagai macam distribusi. Peneliti menyarankan untuk melihat perilaku algoritme *firefly* terhadap perbedaan persebaran data yang disesuaikan dengan pergerakan randomisasi.

## DAFTAR PUSTAKA

- [1] Abonyi, J., & Feil, B. (2010). *Cluster Analysis for Data Mining and System Identification*. Boston, Berlin, Jerman: Springer. Retrieved Februari 6, 2016
- [2] Kwon, S. H. (1998, Oktober 29). Cluster Validity Index for Fuzzy Clustering. *Electronics Letters*, 2176–2177. doi:10.1049/el:19981523
- [3] Muranishi, M., Honda, K., & Notsu, A. (2014). Xie-Beni-Type Fuzzy Cluster Validation in Fuzzy Co-clustering of Documents and Keywords. *Soft Computing in Artificial Intelligence*, 29-38.
- [4] Wu, K.-L., & Yang, M.-S. (2005). A Cluster Validity Index for Fuzzy Clustering. *Pattern Recognition Letters* 26, 1275-1291.
- [5] Yang, X.-S. (2014). *Cuckoo Search and Firefly Algorithm*. New York: Springer.
- [6] Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining*. London: Pearson
- [7] Rajendran, A., & Dhanasekaran, R. (2012). Brain Tumor Segmentation on MRI Brain Images with Fuzzy Clustering and GVF Snake Model. *Int J Comput Commun*, 530-539.
- [8] Hoppner, F., Klawonn, F., Kruse, R., & Runkler, T. (2000). *Fuzzy Cluster Analysis : Method for Classification, Data Analysis, and Image Recognition*. New York: Wiley.
- [9] Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithm*. New York: Plenum Press. doi: 10.1007/978-1-4757-0450-1.
- [10] Liu, H.-C., Jeng, B.-C., Yih, J.-M., & Yu, Y.-K. (2009). Fuzzy C-Means Algorithm Based on Standard Mahalanobis Distances. *Proceedings of the 2009 International Symposium on Information Processing (ISIP'09)* (pp. 422427). Huangshan, P. R. China: Academy Publisher. Retrieved Februari 7, 2016.
- [11] Nayak, J., Nanda, M., Nayak, K., Naik, B., & Behera, H. S. (2014). An Improved Firefly Fuzzy C-Means (FAFCM) Algorithm for Clustering Real World Data Sets. *Advanced Computing and Informatics, Networking and Informatics – Volume 1 (ICACNI-2014)*, 354-363.
- [12] Dave, R.N., 1996. Validating Fuzzy Partition Obtained Through C-Shells Clustering. *Pattern Recognition Lett.* 17, 613-623