

IMPLEMENTASI *FUZZY C-MEAN* DAN ALGORITMA *PARTICLE SWARM OPTIMIZATION* UNTUK *CLUSTERING* KABUPATEN/KOTA DI INDONESIA BERDASARKAN INDIKATOR INDEKS PEMBANGUNAN MANUSIA

I Kadek Sona Dwiguna^{1§}, G. K. Gandhiadi², Luh Putu Ida Harini³

¹Program Studi Matematika, Fakultas MIPA – Universitas Udayana [Email: dwiguna.son@gmail.com]

²Program Studi Matematika, Fakultas MIPA – Universitas Udayana [Email: gandhiadi@unud.ac.id]

³Program Studi Matematika, Fakultas MIPA – Universitas Udayana [Email: ballidah@unud.ac.id]

§Corresponding Author

ABSTRACT

This research is aimed to determine conduct clustering in accordance with the conditions of districts / cities throughout Indonesia based on the IPM indicator and to determine the performance comparison of Fuzzy C-Means using particle swarm optimization compared to ordinary fuzzy c mean. The study uses 514 district / city data in Indonesia based on four IPM indicators. The research show 4 clusters that describe the condition of the Indonesian region and based on the results of cluster validation shows that there are differences in the ordinary Fuzzy C-Means mean algorithm and Fuzzy C-Means using particle swarm optimization.

Keywords: *Fuzzy C Means, IPM Indicator, Particle Swarm Optimization.*

1. PENDAHULUAN

Indonesia adalah negara ke-4 dengan jumlah penduduk terbanyak di dunia. Jumlah penduduk yang besar tidak selalu berbanding lurus dengan kualitas manusia di dalamnya, hal ini dengan seusai dengan laporan yang dipublikasikan oleh *Human Development Report* tahun 2016 yang menempatkan Indonesia di peringkat 113 dari 188 negara berdasarkan nilai *Human Development Index*.

Badan Pusat Statistik (2019), menjelaskan bahwa Indeks Pembangunan Manusia (IPM) merupakan *benchmark* kesuksesan pembangunan manusia yang menjangkau tiga dimensi kehidupan manusia, yaitu umur panjang dan hidup sehat (diukur dengan indikator Umur Harapan Hidup saat lahir dalam tahun), pengetahuan (diukur dengan indikator Harapan Lama Sekolah dan Rata-rata Lama Sekolah dalam tahun), dan standar hidup layak (diukur dengan indikator Pengeluaran per Kapita per bulan dalam ribuan rupiah).

Fuzzy C-means (FCM) merupakan salah satu teknik *clustering* di mana keberadaan setiap data dalam suatu kluster ditentukan oleh nilai derajat

keanggotaan. FCM termasuk algoritma *clustering* yang paling populer, dikarenakan dalam implementasinya sangat mudah dan memiliki kinerja yang terbilang baik, walaupun demikian *Fuzzy C-Means* memiliki kekurangan yaitu dengan dibangkitkannya pusat kluster secara acak menyebabkan sering kali solusi terjebak dalam optimum lokal, hal ini membuat penulis ingin menggunakan salah satu algoritma optimasi yaitu algoritma *Particle Swarm Optimization* untuk membantu *Fuzzy C-Means* yang diharapkan mendapatkan solusi yang lebih baik (Kusumadewi, 2002).

Particle Swarm Optimization merupakan algoritma optimasi yang didasari oleh suatu perilaku sosial dari pergerakan burung, ikan atau makhluk hidup yang berkelompok lainnya. *Particle Swarm Optimization* menyediakan prosedur pencarian dengan berbasis populasi di mana setiap individu disebut dengan partikel. Partikel merupakan kandidat solusi dari permasalahan yang diberikan, setiap iterasi partikel berpindah posisi menuju ke posisi terbaik di mana solusi terbaik ditemukan (Eberhart & Kennedy, 1995).

Adapun tujuan dari penelitian ini, yaitu melakukan *clustering* yang sesuai dengan kondisi kabupaten/kota di seluruh Indonesia berdasarkan indikator IPM kemudian membandingkan kinerja *Fuzzy C-Means* dan algoritma *Particle Swarm Optimization* dibandingkan *Fuzzy C-Means* biasa

Algoritma FCM Menurut Kusumadewi, (2002), adalah suatu teknik klusterisasi data yang mana keberadaan setiap data dalam suatu kluster ditentukan oleh nilai derajat keanggotaan suatu data yang disimbolkan dengan μ_{ik} , elemen di baris ke- i dan kolom ke- k di μ menunjukkan seberapa besar nilai derajat keanggotaan obyek ke- i di kluster ke- k . Derajat keanggotaan μ_{ik} bernilai pada rentang 0 sampai 1. Semakin tinggi derajat keanggotaan maka semakin tinggi kemiripan data dalam satu kelompok. Setiap derajat keanggotaan μ_{ik} haruslah mengikuti kaidah-kaidah sebagai berikut.

$$\mu_{ik} \in [0,1], \forall i = 1,2,\dots,N; \forall k = 1,2,\dots,c \quad (1)$$

$$\sum_{k=1}^c \mu_{ik} = 1; \forall i = 1,2,3,\dots,N \quad (2)$$

$$0 < \sum_{k=1}^n \mu_{ik} < N; \forall j = 1,2,3,\dots,c \quad (3)$$

Dengan N adalah jumlah obyek penelitian dan c adalah jumlah kluster.

Konsep dasar FCM adalah untuk meminimalkan fungsi obyektif $J(U,V)$ yang didefinisikan sebagai berikut.

$$J(U,V) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^w d_{ik}^2 \quad (4)$$

$$d_{ik}^2 = |z_k - v_i|^2 \quad (5)$$

Dengan U adalah matriks *fuzzy* partisi dari data Z , $V = [v_1, v_2, \dots, v_c]$ adalah pusat kluster, c adalah jumlah kluster d_{ik}^2 adalah jarak Euclidean yang digunakan untuk menghitung jarak antara z_k dan v_i , w adalah bobot pada nilai keanggotaan yang bernilai lebih dari 1.

Algoritma FCM bermula dengan menentukan pusat *cluster* yang akan menandai lokasi rata-rata setiap *cluster*. Pada iterasi 0 atau kondisi awal, pusat kluster ini masih belum tepat, dengan demikian diperlukan perbaikan pusat kluster dan derajat keanggotaan secara berulang dengan menggunakan persamaan

sebagai berikut.

$$y_{jk} = \frac{\sum_{k=1}^N (\mu_{ik})^w z_k}{\sum_{k=1}^N (\mu_{ik})^w} \quad (6)$$

$$\mu_{ik} = \frac{\left[\frac{1}{\sum_{j=1}^N d_{ik}^2} \right]^{\frac{1}{w-1}}}{\sum_{k=1}^c \left[\frac{1}{\sum_{j=1}^N d_{jk}^2} \right]^{\frac{1}{w-1}}} \quad (7)$$

Berikut ini adalah langkah-langkah dalam algoritma FCM. (Kusumadewi, 2002)

1. Menentukan jumlah kluster (c), pangkat pembobot (w), maksimum iterasi, error terkecil yang diharapkan (ϵ).
2. Membangkitkan *random* matriks derajat keanggotaan.
3. Menghitung pusat kluster dengan persamaan (3).
4. Memperbaiki matriks U dengan persamaan (7)
5. Menghitung fungsi obyektif dengan persamaan (4).
6. Memeriksa kondisi berhenti (konvergen), yaitu:
($|J(U,V)^t - J(U,V)^{t-1}| < \epsilon$ atau ($t >$ maksimum iterasi).
7. Jika terpenuhi algoritma berhenti, jika tidak maka langkah ke-4 algoritma dilakukan kembali.

Algoritma *Particle Swarm Optimization* adalah algoritma optimasi berbasis pada populasi, dengan kata lain algoritma ini mengeksplorasi populasi suatu solusi untuk menyelidiki ruang pencarian secara bersamaan pada setiap iterasinya. Populasi dalam algoritma PSO disebut dengan istilah *swarm* dan individu-individunya disebut dengan partikel.

Setiap partikel diberi suatu fungsi *fitness* yang unik yang berbeda dengan partikel di *swarm* lainnya sesuai dengan masalah yang bersesuaian, nilai *fitness* suatu partikel ke $-i$ dievaluasi menggunakan fungsi *fitness* tertentu yang dinotasikan dengan $f_i(t)$ dengan t adalah jumlah iterasi.

PSO menyimpan ingatan (*memory*) di mana setiap partikel menyimpan posisi terbaik yang pernah dikunjungi partikel selama pencarian solusi (p_{ibest}) yang didefinisikan sebagai berikut (Parsopoulos & Vrahatis, 2010)

$$p_{ibest}(t) = \arg \min f(X_i(t)), i = 1,2,\dots,m \quad (8)$$

Dengan $\arg \min f(x_i(t))$ adalah posisi partikel x_i dengan $f(x_i(t))$ yang paling minimal hingga iterasi ke- t dan m adalah banyaknya partikel. Setiap iterasi algoritma PSO, maka

setiap partikel akan semakin mendekati nilai minimum global. Dimisalkan g adalah posisi terbaik dengan nilai fungsi terendah diantara p_{ibest} setiap partikel pada t iterasi yang diberikan, maka:

$$g_{best}(t) = \arg \min f(p_{ibest}(t)) \quad (9)$$

Setiap partikel akan memperbarui (*update*) kecepatan dan posisinya masing-masing dengan menggunakan persamaan berikut :

$$vit_i(t+1) = w * vit_i(t) + \varphi_1 * r_1(p_{ibest} - X_i) + \varphi_2 * r_2(g_{best} - X_i) \quad (10)$$

$$x_i(t+1) = x_i + vit_i \quad (11)$$

dengan w adalah koefisien inersia yang ditetapkan sebelumnya dan selanjutnya diturunkan dengan secara bertahap setiap iterasi, jika dituliskan secara matematis adalah sebagai berikut.

$$w(t) = w_{max} - \left(\frac{w_{max} - w_{min}}{T_{max}} \right) * t \quad (12)$$

dengan w_{max} dan w_{min} adalah nilai koefisien inersia yang maksimum dan minimum, T_{max} adalah maksimum iterasi dan t adalah iterasi yang sedang berlangsung. Ketika w bernilai maksimum, proses pencarian terjadi secara global ke segala arah. Sementara itu, ketika w diturunkan, pencarian menjadi lebih ke fokus ke suatu area tertentu yang potensial (Suyanto, 2017). φ_1, φ_2 adalah koefisien akselerasi berupa konstanta positif pada umumnya bernilai 2 (Eberhart & Kennedy, 1995) dan r_1, r_2 adalah bilangan acak pada interval $[0,1]$.

Setelah kecepatan dan posisi partikel diperbarui mengikuti partikel optimum, algoritma PSO terus dilakukan berulang hingga diperoleh semua partikel (Izakian & Abraham, 2011) menuju solusi yang optimum. (Izakian & Abraham, 2011).

Berikut ini adalah langkah-langkah algoritma *Particle Swarm Optimization*

1. Menginisialisasi posisi awal setiap partikel (x_i) dan kecepatan awal setiap partikel (vit_i) secara acak dengan $i = 1, 2, \dots, l$ dan l adalah banyaknya partikel.
2. Menghitung nilai fungsi *fitness* (fungsi tujuan) untuk setiap partikel $f(x_i)$.
3. Menentukan p_{ibest} dan g_{best} awal setiap partikel.
4. Memperbarui kecepatan setiap partikel (vit_i) dengan persamaan (10)
5. Memperbarui posisi setiap partikel (x_i) dengan persamaan (11)

6. Menghitung kembali fungsi *fitness* (fungsi tujuan) $f(x_i)$, jika $f(x_i) \leq f(p_{ibest})$ maka $p_{ibest} = x_i$, setelah mendapatkan p_{ibest} baru, maka didapatkan g_{best} baru.

$$g_{best} = \begin{cases} p_{ibest}, & f(p_{ibest}) < f(g_{best}) \\ g_{best}, & f(p_{ibest}) \geq f(g_{best}) \end{cases}$$

7. Jika iterasi mencapai maksimum atau konvergen, maka algoritma berhenti, jika tidak maka langkah ke-4 algoritma dilakukan kembali.

Permasalahan dari FCM adalah solusi yang diperoleh sering kali terjebak dalam optimum lokal. (Wei Pang, 2004) dalam jurnalnya mengusulkan modifikasi algoritma *Particle Swarm Optimization* untuk permasalahan TSP (*Travel Salesman Problem*) yang kemudian dikenal dengan *Fuzzy Particle Swarm Optimization* (FPSO). Pada FPSO, solusi optimum adalah fungsi obyektif pada FCM, dalam FPSO, pencarian solusi dilakukan dengan memanggil l sejumlah partikel, dengan partikel adalah kandidat solusi dalam permasalahan FCM. Posisi partikel dalam FPSO dinotasikan dengan x_i . Posisi partikel dalam FPSO merepresentasikan matriks derajat keanggotaan U dalam FCM, dengan kata lain x_i merupakan matriks yang berisikan q jumlah data yang diproses dan c jumlah kluster.

Konsep dasar FCM adalah meminimalkan fungsi obyektif dengan kata lain semakin kecil $J(U, V)$ semakin baik solusi yang diperoleh, hal ini berarti posisi partikel ke- i dengan nilai $f(x_i)$ yang terendah adalah posisi partikel ke- i yang terbaik dan g_{best} adalah $f(p_{ibest})$ yang terkecil. Setelah diperoleh p_{ibest} dan g_{best} keseluruhan partikel dalam ruang solusi bergerak mengikuti posisi partikel dengan $f(x_i)$ yang terbaik, pergerakan partikel ini dihitung dengan cara sebagai berikut. (Izakian & Abraham, 2011).

$$vit_i(t+1) = w * vit_i(t) + \varphi_1 r_1(p_{ibest} - x_i) + \varphi_2 r_2(g_{best} - x_i) \quad (13)$$

$$x_i(t+1) = x_i \oplus vit_i \quad (14)$$

Setelah semua posisi partikel telah diperbarui, tinggi kemungkinan aturan $\sum_{k=1}^c \mu_{pk} = 1, \forall p$ untuk setiap posisi partikel tidak dipenuhi. Pada kondisi seperti ini, maka dilakukan normalisasi matriks untuk setiap partikel. Pertama semua nilai μ_{pk} yang bernilai negatif menjadi nol. Jika semua nilai dalam baris matriks bernilai nol maka bangkitkan bilangan acak dengan interval $[0,1]$. Matriks baru yang telah dinormalkan ditunjukkan sebagai

berikut.(Izakian & Abraham, 2011).

$$x_i^{normal} = \begin{bmatrix} \frac{\mu_{11}}{\sum_{i=1}^c \mu_{i1}} & \dots & \frac{\mu_{c1}}{\sum_{i=1}^c \mu_{i1}} \\ \vdots & \ddots & \vdots \\ \frac{\mu_{1n}}{\sum_{i=1}^c \mu_{in}} & \dots & \frac{\mu_{cn}}{\sum_{i=1}^c \mu_{in}} \end{bmatrix} \quad (15)$$

Izakian & Abraham (2011) dalam jurnalnya mengintegrasikan algoritma FCM ke dalam algoritma *Fuzzy Particle Swarm Optimization* (FPSO) untuk membentuk algoritma klusterisasi yang disebut dengan algoritma FCM-PSO. Dalam algoritma FCM-PSO, kinerja algoritma FCM dan algoritma FPSO dapat dipertahankan dan bisa ditingkatkan guna mendapatkan hasil kluster yang lebih baik.

Algoritma FCM-PSO

1. Menentukan data yang akan dilakukan klusterisasi z_{ij} yaitu data sampel ke- i dengan $i = (1,2,3 \dots N)$ dan variabel ke- j dengan $j = (1,2,3, \dots, n)$.
2. Menentukan jumlah kluster (c), pangkat pembobot (w), maksimum iterasi, error terkecil yang diharapkan (ε).
3. Algoritma FPSO
 - i. Menghitung pusat kluster masing-masing partikel menggunakan persamaan (6).
 - ii. Menghitung nilai $J(U, V)$ dan $f(x_i)$ masing-masing partikel menggunakan persamaan (4).
 - iii. Menentukan p_{ibest} setiap partikel menggunakan persamaan (8).
 - iv. Menentukan g_{best} menggunakan persamaan (9).
 - v. Memperbarui kecepatan (vit_i) seluruh partikel menggunakan persamaan (13).
 - vi. Memperbarui posisi (x_i) seluruh partikel dengan persamaan (14).
 - a. Cek kondisi posisi setiap partikel (x_i).
 - b. Jika $\sum_{k=1}^c \mu_{pk} \neq 1, \forall p$ lakukan normalisasi matriks dengan menggunakan persamaan (15)
 - c. Jika $\sum_{k=1}^c \mu_{pk} = 1, \forall p$, algoritma dilanjutkan.
 - vii. Melakukan kembali langkah ke-3 hingga $t \leq$ maksimum iterasi atau $g_{best}(t) - g_{best}(t-1) < \varepsilon$.
4. Algoritma FCM
 - i. Menghitung pusat kluster masing-masing partikel menggunakan persamaan (6).

- ii. Menghitung fungsi jarak *euclidean* masing-masing partikel menggunakan persamaan (5).
 - iii. Memperbarui fungsi keanggotaan masing-masing partikel menggunakan persamaan (7).
 - iv. Menentukan p_{ibest} setiap partikel menggunakan persamaan (8).
 - v. Menentukan nilai g_{best} menggunakan persamaan (9).
 - vi. Melakukan kembali Langkah 4 hingga $t \leq$ maksimum iterasi atau $g_{best}(t) - g_{best}(t-1) < \varepsilon$.
5. Melakukan ulang langkah ke-3 hingga $t \leq$ maksimum iterasi atau tidak terdapat perubahan nilai g_{best} selama 2 iterasi berturut-turut.

Validitas *Cluster* digunakan untuk membandingkan kinerja pengelompokan *Fuzzy C-Means* dan algoritma optimasi *Particle Swarm Optimization* dengan *Fuzzy C-Means* biasa.

1. *Partition Coefficient* (PC)

Indeks ini digunakan untuk menghitung jumlah *overlapping* antar kelompok. Indeks ini dirumuskan sebagai berikut. (Balasko, Abonyi, & Feil, 2005)

$$PC(C) = 1/N \sum_{i=1}^c \sum_{k=1}^N \mu_{ik}^2$$

dengan N merupakan jumlah objek penelitian, c adalah jumlah kluster, dan μ_{ik} adalah nilai keanggotaan objek ke- k dengan pusat kelompok ke- i . Nilai PC yang paling besar menunjukkan jumlah kelompok yang optimal.

2. *Xie and Beni's Index* (XB)

Indeks ini digunakan untuk mengukur rasio total variasi dalam kluster dan pemisahan antar kluster. Indeks ini dirumuskan sebagai berikut. (Balasko, Abonyi, & Feil, 2005)

$$XB = \frac{\sum_{i=1}^c \sum_{k=1}^N \mu_{ik}^m \|x_k - v_k\|^2}{N \min_{i,k} \|v_k - v_i\|^2}$$

dengan N adalah jumlah obyek penelitian, c adalah jumlah kelompok μ_{ik} adalah nilai keanggotaan obyek ke- k dengan pusat kelompok ke- i , m adalah pembobot dalam FCM, $\|x_k - v_k\|^2$ adalah jarak *euclidean* data (x_k) dengan pusat kluster v_i , dan $\|v_k - v_i\|^2$ adalah jarak *euclidean* antar pusat kluster, kluster yang lebih baik ditunjukkan dengan nilai XB yang lebih rendah.

3. Partition Index (PI)/Separation dan Compactness (SC)

PI atau SC adalah rasio antara jumlah kepadatan dan pemisahan antar kluster. PI atau SC digunakan untuk membandingkan hasil pengelompokan yang berbeda yang memiliki jumlah kluster yang sama. Indeks ini dirumuskan dalam persamaan berikut (Balasko, Abonyi, & Feil, 2005)

$$SC = \frac{\sum_{i=1}^c \frac{\sum_{k=1}^N (\mu_{ik})^m \|x_k - v_i\|^2}{N_i \sum_{j=1}^c \|v_j - v_i\|^2}}$$

Dengan N adalah jumlah obyek penelitian, N_i adalah obyek penelitian kelompok ke- i , c merupakan banyaknya kluster, μ_{ik} merupakan nilai keanggotaan obyek ke- k dengan pusat kelompok ke- i , m merupakan pembobot dalam FCM, $\|x_k - v_i\|^2$ adalah jarak *euclidean* data (x_k) dengan pusat kluster v_i , dan $\|v_k - v_i\|^2$ adalah jarak *euclidean* antar pusat kluster. Partisi kluster yang lebih baik ditunjukkan dengan nilai SC yang lebih rendah

4. Separation Index (S)

Pemisahan antar kluster pada *Separation Index* (S), didefinisikan sebagai minimum jarak antar pusat kluster. Indeks ini dirumuskan dalam persamaan berikut (Bezdek, 1996).

$$S = \frac{\sum_{i=1}^c \frac{\sum_{k=1}^N (\mu_{ik})^m \|x_k - v_i\|^2}{N_i \min_{i,j} \|v_j - v_i\|^2}}$$

Nilai S yang lebih rendah mengindikasikan partisi kelompok yang lebih baik.

Indikator IPM menurut (Badan Pusat Statistik, 2019) ada empat, yaitu

1. Umur Harapan Hidup saat Lahir (UHH)
Merupakan rata-rata taksiran jumlah tahun yang dapat dicapai oleh seseorang selama hidup.
2. Rata-rata Lama Sekolah (RLS)
Menggambarkan jumlah tahun pada penduduk usia 25 tahun ke atas dalam menjalani pendidikan formal.
3. Angka Harapan Lama Sekolah (HLS)
Menggambarkan lamanya sekolah dalam satuan tahun yang diharapkan akan diperoleh oleh anak rentang umur tertentu di masa mendatang.
4. Pengeluaran per Kapita Disesuaikan
Merupakan pengeluaran yang ditentukan dari nilai pengeluaran per kapita dan paritas daya beli (*Purchasing Power Parity*-PPP).

2. METODE PENELITIAN

Sumber Data. Data yang akan digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari BPS tahun 2019, yaitu data indikator IPM di 514 Kabupaten/Kota di Indonesia.

Variabel Penelitian. Variabel penelitian yang digunakan diperoleh dari BPS tahun 2019, variabel merupakan indikator IPM yang digunakan oleh BPS, meliputi Umur Harapan Hidup-UHH (Z_1), Rata-rata Lama Sekolah-RLS (Z_2), Angka Harapan Lama Sekolah-HLS (Z_3), dan Pengeluaran per kapita yang disesuaikan-PPK (Z_4).

Teknik analisis data. Berikut ini adalah langkah-langkah yang dilakukan dalam penelitian.

1. Menentukan data yang akan dilakukan klusterisasi z_{ij} yaitu data sampel ke- i dengan $i = (1,2,3 \dots N)$ dan variabel ke- j dengan $j = (1,2,3, \dots, n)$. Dalam penelitian ini N merupakan jumlah data Kabupaten/Kota di Indonesia yaitu 514 dan n merupakan variabel yang digunakan yaitu 4 indikator IPM yaitu UHH (Z_1), RLS(Z_2), AHS(Z_3), dan PPK(Z_4).
2. Menormalisasi data x_{ij} dengan *z-score*.

$$\hat{z}_{ij} = (z_{ij} - \frac{\sum_{i=1}^N z_{ij}}{N}) / std$$

untuk setiap i , dengan \hat{z}_{ij} adalah data sampel yang telah di normalisasi dan z_{ij} adalah data sampel yang belum di normalisasi.

3. Menentukan jumlah kluster (c), pangkat pembobot (w), maksimum iterasi, error terkecil yang diharapkan (ϵ). Jumlah kluster yang digunakan adalah 4 mengikuti jumlah pengelompokan yang dilakukan oleh BPS, pangkat pembobot adalah 2, maksimum iterasi kondisi berhenti pada sub iterasi FPSO adalah tercapai 1000 iterasi atau nilai *gbest* tidak berubah selama 200 iterasi beruntun., maksimum iterasi pada sub iterasi FCM adalah 5 dan kondisi berhenti keseluruhan algoritma FCM-PSO adalah nilai *gbest* tidak berubah selama 2 iterasi beruntun.
4. Algoritma FPSO
 - i. Menghitung pusat kluster masing-masing partikel menggunakan persamaan (6).
 - ii. Menghitung nilai $J(U, V)$ dan $f(x_i)$ masing-masing partikel menggunakan persamaan (4).
 - iii. Menentukan p_{ibest} setiap partikel menggunakan persamaan (8).

- iv. Menentukan g_{best} menggunakan persamaan (9).
 - v. Memperbarui kecepatan (vit_i) seluruh partikel menggunakan persamaan (13).
 - vi. Memperbarui posisi (x_i) seluruh partikel dengan persamaan (14).
 - a. Memeriksa kondisi posisi setiap partikel (x_i)
 - b. Jika $\sum_{k=1}^c \mu_{pk} \neq 1, \forall p$ dilakukan normalisasi matriks dengan menggunakan persamaan (15)
 - c. Jika $\sum_{k=1}^c \mu_{pk} = 1, \forall p$, algoritma dilanjutkan
 - vii. Melakukan ulang langkah ke-4 hingga $t \leq$ maksimum iterasi atau $g_{best}(t) - g_{best}(t - 1) < \epsilon$
5. Algoritma FCM
- i. Menghitung pusat kluster masing-masing partikel menggunakan persamaan (6).
 - ii. Menghitung jarak *euclidean* masing-masing partikel menggunakan persamaan (5).
- iii. Memperbarui fungsi keanggotaan setiap partikel menggunakan persamaan (7).
 - iv. Menentukan p_{ibest} setiap partikel menggunakan persamaan (8).
 - v. Menentukan nilai g_{best} menggunakan persamaan (9).
 - vi. Melakukan kembali langkah 5 hingga $t \leq$ maximum iterasi atau $g_{best}(t) - g_{best}(t - 1) < \epsilon$.
6. Melakukan kembali langkah ke-4 hingga $t \leq$ maximum iterasi atau tidak terdapat perubahan nilai g_{best} selama 2 iterasi beruntun.
7. Melakukan uji validitas untuk kelompok yang diperoleh dari *Fuzzy C-Means* dan *Particle Swarm Optimization* dengan *Fuzzy C-Means* tanpa optimasi.
8. Menginterpretasi hasil pengelompokan yang diperoleh.

3. HASIL DAN DISKUSI

Penelitian ini dikerjakan untuk mengetahui hasil *clustering* algoritma *hybrid Fuzzy C-Means* dengan algoritma optimasi *Particle Swarm Optimization* pada data 514 kabupaten/kota di Indonesia. Hasil *clustering* data tersebut diharapkan nantinya dapat dijadikan sebagai acuan sejauh mana pembangunan nasional di Indonesia, sehingga diharapkan kabupaten/kota yang berada di kluster terbawah bisa mendapat perhatian lebih oleh pemerintah.

Berikut ini adalah data kabupaten/kota di Indonesia.

Tabel 1. Data Kabupaten/Kota di Indonesia

No.	Kabupaten / Kota	AHH (Z1) (Tahun)	RLS (Z2) (Tahun)	HLS (Z3) (Ribu Rupiah/Tahun)	PPK(Z4) (Tahun)
1	SIMEULUE	65.22	13.51	7210.00	9.08
2	ACEH SINGKIL	67.36	14.30	8715.00	8.52
3	ACEH SELATAN	64.27	14.41	8187.00	8.59
4	ACEH TENGGAR A	68.04	13.99	8067.00	9.65
5	ACEH TIMUR	68.67	13.02	8600.00	7.86
6	ACEH TENGAH	68.82	14.26	10782.00	9.69
...
512	INTAN JAYA	65,09	7.36	5593.00	2.64
513	DEIYAI	64,63	9.80	4958.00	3.00
514	KOTA JAYAPURA	70,00	15.00	15176.00	11.55

Berdasarkan data yang ditunjukkan pada tabel 1 selanjutnya akan dilakukan normalisasi data, kemudian dengan mengikuti tahapan-tahapan algoritma FCM-PSO akan diperoleh g_{best} untuk seluruh partikel di setiap iterasinya. g_{best} mempresentasikan solusi dari FCM yaitu fungsi obyektif paling minimum. Selanjutnya algoritma FCM-PSO akan terus berjalan hingga kondisi berhentinya terpenuhi. Dengan bantuan *software Matlab* berikut adalah nilai g_{best} untuk setiap iterasinya dan perbandingan dengan algoritma FCM tanpa optimasi.

Tabel 2. Fungsi Obyektif FCM-PSO

iterasi	FCM – PSO
1	491.405469745084
2	434.504609958592
3	433.820162036769
4	433.789060324425
5	433.785768537181
6	433.785202111597
7	433.785080865956
8	433.785055463270
9	433.785054126373
10	433.785054068165

Tabel 3. Fungsi Obyektif FCM

iterasi	FCM
1	682.8043733617
2	512.4185957680
3	511.0442316092
⋮	⋮
52	433.7850972358
53	433.7850875734

Perbandingan *Fuzzy C-Means* dengan *Fuzzy C-Means* dan *Particle Swarm Optimization*

Berdasarkan tabel 2 diperoleh fungsi obyektif pada algoritma *Fuzzy C-Means* dengan algoritma optimasi *Particle Swarm Optimization* setelah 10 iterasi mendapatkan fungsi obyektif terakhir sebesar 433.78505. Sedangkan pada tabel 3 diperoleh FCM tanpa optimasi setelah 53 iterasi dilakukan diperoleh fungsi obyektif terakhir sebesar 433.78508 hal ini menunjukkan FCM-PSO tidak memiliki hasil yang berbeda secara signifikan walaupun hasil yang diperoleh dari algoritma FCM-PSO sedikit lebih baik, langkah selanjutnya adalah menguji validitas klaster dengan menggunakan indeks *Partition Coefficient* (PC), *Xie and Beni's Index* (XB), *Partition Index* (PI)/*Separation dan Compactness* (SC) dan *Separation Index* (S). Hasil pengujian dilakukan dengan *Matlab* yang bisa dilihat pada tabel di bawah ini.

Tabel 4. Perbandingan Algoritma FCM-PSO dan Algoritma FCM

Validitas	FCM-PSO	FCM
PC	0.43642120	0.43641980
SC	0.41558799	0.41558872
S	0.00134610	0.00134612
XB	1.94088680	1.94054017

Berdasarkan tabel 4 dapat dilihat bahwa berdasarkan kedua uji validitas yang dilakukan, hasil partisi menggunakan metode FCM-PSO lebih unggul di 3 dari 4 uji validitas yang dilakukan walaupun tidak signifikan.

Interpretasi hasil *clustering* kabupaten/kota di Indonesia

Software matlab memerlukan 10 iterasi untuk mendapatkan posisi partikel terbaik sebelum memperoleh solusi optimal pada fungsi tujuan senilai 433.785054068165. Hasil kedua yang diperoleh adalah pusat 197lister atau *center*.

$$V = \begin{bmatrix} 70.030 & 12.888 & 10940.97 & 8.371 \\ 65.364 & 12.592 & 8459.01 & 7.575 \\ 72.230 & 14.461 & 14196.16 & 10.559 \\ 69.202 & 12.289 & 9509.899 & 7.321 \end{bmatrix}$$

Berdasarkan dari pusat klaster (V) yang diperoleh, maka Kabupaten/Kota di Indonesia dikelompokkan menjadi 4 kelompok sebagai berikut:

1. Kelompok pertama (*Cluster* ke-1), berisi kabupaten/kota yang memiliki angka harapan hidup saat lahir sekitar 70.030 tahun; memiliki harapan lama sekolah sekitar 12.888 tahun ; memiliki rata-rata pengeluaran per kapita 10940.97 ribu rupiah per tahun; dan memiliki rata-rata lama sekolah 8.371 tahun.
2. Kelompok kedua (*Cluster* ke-2), berisi kabupaten/kota yang memiliki angka harapan hidup saat lahir sekitar 65.364 tahun; memiliki harapan lama sekolah sekitar 12.592 tahun ; memiliki rata-rata pengeluaran perkapita 8459.01 ribu rupiah per tahun; dan memiliki rata-rata lama sekolah 7.575 tahun.
3. Kelompok ketiga (*Cluster* ke-3), berisi kabupaten/kota yang memiliki angka harapan hidup saat lahir sekitar 72.23 tahun; memiliki harapan lama sekolah sekitar 14.461 tahun ; memiliki rata-rata pengeluaran perkapita 14196.16 ribu rupiah per tahun; dan memiliki rata-rata lama sekolah 10.559 tahun.
4. Kelompok keempat (*Cluster* ke-4), berisi kabupaten/kota yang memiliki angka harapan hidup saat lahir sekitar 69.202 tahun; memiliki harapan lama sekolah sekitar 12.289 tahun ; memiliki rata-rata pengeluaran perkapita 9509.899 ribu rupiah per tahun; dan memiliki rata-rata lama sekolah 7.321 tahun.
Berikut adalah hasil pengelompokan kabupaten/kota di Indonesia berdasarkan derajat keanggotaannya.

Tabel 5. Hasil kluster kabupaten/kota di Indonesia

Klaster 1	Klaster 2	Klaster 3	Klaster 4
Aceh Tenggara	Simeulue	Aceh Besar	Aceh Timur
Aceh Tengah	Aceh Singkil	Kota Banda Aceh	Nias
Aceh Barat	Aceh Selatan	Kota Sabang	Nias Selatan
Pidie	Aceh Barat Daya	Kota Langsa	Nias Utara
Bireuen	Gayo Lues	Kota Lhokseumawe	Nias Barat
Aceh Utara	Kota Subulussalam	Toba Samosir	Indragiri Hilir
...
Mimika	Deiyai	Kota Jayapura	Intan Jaya

Berdasarkan pengelompokan 4 kluster tersebut bisa diambil kesimpulan kelompok ke 2 yang berisikan 138 kabupaten/kota memiliki nilai indikator terkecil. 138 kabupaten/kota tersebut 69.5% merupakan kabupaten/kota yang berada di kawasan Indonesia timur.

4. SIMPULAN DAN SARAN

Simpulan

- Metode *Fuzzy C-Means* dapat dipadukan dengan optimasi *Particle Swarm Optimization* dalam proses clustering. FCM-PSO membagi data kabupaten/kota di Indonesia menjadi 4 kluster, proses algoritma berhenti pada iterasi ke-10 dengan fungsi obyektif diperoleh sebesar 433.7850541 hasil pengelompokan menunjukkan kabupaten/kota di Indonesia bagian timur cenderung masuk ke dalam kluster kedua yaitu kluster dengan nilai indikator yang paling kecil, dengan demikian diharapkan pembangunan nasional lebih ditingkatkan lagi pada wilayah di Indonesia bagian timur.
- Penggunaan algoritma optimasi *Particle Swarm Optimization* dalam *Fuzzy C-Means* tidak terlalu berpengaruh secara signifikan walaupun 3 dari 4 uji validitas, algoritma FCM-PSO lebih unggul dari FCM.

Saran

Saran yang bisa diberikan pada penelitian tentang *clustering* Kabupaten/Kota di Indonesia

adalah menggunakan optimasi yang lain seperti *Ant-Colony*, *Genetic Algorithm* yang kemudian dipadukan dengan algoritma FCM.

DAFTAR PUSTAKA

- Badan Pusat Statistik. (2019). *Indeks Pembangunan Manusia 2019*. Jakarta: Badan Pusat Statistik.
- Balasko, B., Abonyi, J., & Feil, B. (2005). *Fuzzy Clustering and Data Analysis Toolbox For Use with Matlab*. Department of Process Engineering University of Veszprem.
- Bezdek, J. C. (1996). Validity-guided (Re) Clustering with applications to image segmentation. *IEEE Transactions on Fuzzy Systems*, 112-123.
- Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In Proceedings of the sixth international symposium on micro machine and human. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39-43.
- Izakian, H., & Abraham, A. (2011). Fuzzy C-means and fuzzy swarm for fuzzy clustering problem. *Expert Systems with Applications*, 1835-1838.
- Kusumadewi, S., & Purnomo, H. (2010). *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Yogyakarta: Graha Ilmu.
- Parsopoulos, K. E., & Vrahatis, M. N. (2010). *Particle Swarm Optimization and Intelligence: Advances and Applications*. New York: Information Science Reference.
- Parsopoulos, K., & Vrahatis, M. (2010). *Particle swarm optimization and intelligence : advances and applications*. Information Science Reference.
- Wei Pang, K.-P. W. (2004). Fuzzy Discrete Particle Swarm Optimization for Solving Traveling Salesman Problem. *The Fourth International Conference on Computer and Information Technology*, 796-800.