



IMPLEMENTASI METODE HYBRID FUZZY C-MEANS DAN FUZZY SWARM UNTUK PENGELOMPOKAN DATA BENANG PERUSAHAAN TEKSTIL

Tiffany Nabarian¹, Muhammad Aris Ganiardi², Reza Firsandaya Malik³

¹Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

²Manajemen Informatika, Politeknik Negeri Sriwijaya

³Sistem Komputer, Universitas Sriwijaya

Jakarta Selatan, DKI Jakarta, Indonesia

tifany.nabarian@dosen.nurulfikri.ac.id, marisg2010@gmail.com, rezafm@unsri.ac.id

Abstract

Thread is the main raw materials for production process in textile companies. The availability of thread consumption data in textile companies can be used to determine the pattern of thread consumption in a certain period. Data clustering is one of method that can be used to form patterns for the data thread. In this research, hybrid Fuzzy C-Means (FCM) and Fuzzy Particle Swarm Optimization (FPSO) clustering algorithms are used, which are the combination of algorithms FCM and FPSO. This hybrid algorithm is able to overcome the weaknesses of the original algorithm, namely FCM. The purpose of this research is to implement the FCM-FPSO hybrid algorithm into an application, then test the performance. The application implemented Unified Process for software engineering method. In this application, the performance of three methods was compared, those methods are FCM, FPSO and Hybrid FCM-FPSO. The result of the implementation is the lowest average objective function is 3441.00 achieved by the Hybrid FCM-FPSO algorithm, then followed by the FCM algorithm with value of 3540.33 and the highest is achieved by the FPSO algorithm with value of 4485.40. This result showed that the application was successfully proved that the FCM-FPSO Hybrid algorithm can produce the best thread clusters.

Keywords: fuzzy, swarm; thread; hybrid FCM-FPSO; clustering; unified process

Abstrak

Salah satu bahan baku utama dalam proses produksi di perusahaan tekstil adalah benang. Ketersediaan data konsumsi benang pada perusahaan tekstil dapat dimanfaatkan untuk mengetahui pola konsumsi benang pada periode tertentu. *Data mining* metode *clustering* adalah salah satu teknik yang dapat digunakan untuk membentuk pola dari data benang tersebut. Pada penelitian ini, digunakan algoritma *clustering Hybrid Fuzzy C-Means (FCM)* dan *Fuzzy Particle Swarm Optimization (FPSO)*, yaitu algoritma kombinasi dari FCM dan FPSO. Algoritma hibrida ini mampu mengatasi kelemahan dari algoritma asalnya, yaitu FCM. Tujuan dari penelitian ini yaitu menguji performa dari metode hibrid FCM-FPSO dengan cara mengimplementasikan pengelompokan data benang perusahaan tekstil ke dalam sebuah aplikasi. Aplikasi dikembangkan dengan menerapkan model *Unified Process (UP)*. Hasil dari implementasi tersebut adalah nilai rata-rata fungsi objektif terendah dicapai oleh algoritma hibrid FCM-FPSO sebesar 3441,00 kemudian diikuti oleh algoritma FCM dengan nilai sebesar 3540,33 dan yang tertinggi dicapai oleh algoritma FPSO dengan nilai sebesar 4485,40. Nilai rata-rata fungsi objektif yang terendah ini menunjukkan bahwa aplikasi yang dibangun berhasil membuktikan keunggulan algoritma *hybrid FCM-FPSO* dalam menghasilkan *cluster* data benang.

Kata kunci: fuzzy, swarm; benang; hibrid FCM-FPSO; clustering; unified process

1. PENDAHULUAN

Pada Persaingan di dunia industri mengharuskan pihak manajemen perusahaan untuk selalu memikirkan strategi terbaik sehingga proses produksi dapat terus berlangsung. Kebutuhan akan data dan informasi menjadi sesuatu yang diutamakan dalam dunia industri. Melimpahnya ketersediaan data industri merupakan aset strategis yang dimiliki oleh setiap perusahaan [1]. Oleh karena itu, dibutuhkan suatu teknik khusus agar data industri yang tersedia dapat dimanfaatkan untuk pengaturan strategi bisnis. *Data mining* merupakan salah satu teknik ilmiah yang dapat dimanfaatkan oleh perusahaan untuk mencapai tujuan tersebut. Terdapat beragam metode dalam teknik data mining, *clustering* adalah salah satunya. Tujuan dari metode *clustering* ini yaitu untuk dapat melihat pola karakteristik yang terbentuk dari setiap data sehingga akan memudahkan dalam pengaturan strategi perusahaan. Metode *clustering* yang diimplementasikan pada penelitian ini adalah metode hibrida antara algoritma *Fuzzy C-Means* dan *Fuzzy Particle Swarm Optimization*. Pada penelitian ini, data yang akan dianalisis adalah data konsumsi benang pada salah satu perusahaan tekstil di kota Batam. Data benang dipilih karena merupakan salah satu bahan baku utama pada proses produksi di perusahaan tekstil tersebut sehingga hasil *clustering* dapat bermanfaat secara signifikan untuk strategi perusahaan.

Beberapa penelitian di bidang bisnis telah mengimplementasikan teknik *data mining* metode *fuzzy clustering*, diantaranya untuk memecahkan permasalahan di koperasi simpan pinjam [2], transaksi swalayan [3], sampai penjualan makanan pada bisnis restoran [4]. Pada penelitian ini dilakukan analisis terhadap data benang pada perusahaan tekstil dengan membangun aplikasi pengelompokan data benang menggunakan metode *unified process*. Selain itu, juga dilakukan perbandingan performa diantara tiga algoritma, diantaranya *Fuzzy C-Means*, *Fuzzy Particle Swarm Optimization*, serta hibrida dari kedua algoritma tersebut.

2. TINJAUAN PUSTAKA

Pada bagian ini dapat dijelaskan landasan teori yang dibahas yang bersifat khusus, penelitian terkait yang menjadi dasar penelitian, hipotesis penelitian (jika ada), dan model penelitian.

2.1 *Data mining dan Fuzzy Clustering*

Data mining adalah proses berulang di mana setiap proses kemajuannya didefinisikan sebagai penemuan, baik melalui metode manual maupun otomatis. *Data mining* berguna untuk analisis eksplorasi, yaitu di saat belum ada gagasan yang telah ditentukan sebelumnya dan belum diketahui

hasil yang ingin didapatkan. *Data mining* merupakan teknik pencarian informasi baru, berharga, dan *nontrivial* dalam volume data yang besar. Hasil terbaik bisa dicapai dengan menyelaraskan pengetahuan pakar (manusia) dalam menggambarkan tujuan dan masalah dengan kemampuan pencarian pada komputer.

Metode *data mining* yang diimplementasikan pada penelitian ini adalah metode *clustering*. *Clustering* adalah teknik pengelompokan yang tidak membutuhkan fase pelatihan atau pembelajaran dalam pemanfaatannya (*unsupervised*). Tujuan dari metode *clustering* adalah mengelompokkan sejumlah data atau objek ke dalam *cluster* (himpunan bagian, kelompok, atau kelas) [5]. Pengelompokan harus memiliki nilai kemiripan (*homogeneity*) antar objek di dalam suatu *cluster* dan ketidakmiripan (*heterogeneity*) antar objek pada *cluster* yang berbeda. Konsep jarak dapat diartikan sebagai konsep ketidakmiripan. Apabila jarak dua data atau objek titik cukup dekat, maka dua objek itu dapat dikatakan mirip. Semakin jauh jarak, maka semakin tinggi juga nilai ketidakmiripannya [6]. Metode *clustering* secara umum diklasifikasikan menjadi dua, yaitu *hard clustering* dan *soft (fuzzy) clustering*. Metode *hard clustering* menetapkan setiap objek ke grup tunggal, sementara metode *fuzzy clustering* memperkenalkan derajat keanggotaan antara objek dan grup yang berbeda dari dataset [5].

2.2 *Fuzzy C-Means (FCM)*

Metode FCM diajukan oleh James C. Bezdek pada tahun 1981. Pada metode FCM setiap data dapat menjadi anggota pada lebih dari satu *cluster*. FCM membagi himpunan sejumlah n objek $o = \{o_1, o_2, \dots, o_n\}$ pada dimensi R^d ke dalam c ($1 < c < n$) *cluster fuzzy* dengan $Z = \{z_1, z_2, \dots, z_c\}$ sebagai pusat *cluster* atau *centroid*. *Fuzzy clustering* dari sejumlah objek dideskripsikan dengan matriks fuzzy μ dengan n jumlah baris dan c jumlah kolom, untuk n merupakan jumlah dari objek data dan c jumlah dari *cluster*. Elemen baris ke- i dan kolom ke- j pada μ , μ_{ij} , mengindikasikan derajat keanggotaan dari objek ke- i terhadap kolom ke- j [7]. Algoritma dari FCM adalah sebagai berikut [7].

1. Tentukan nilai m ($m > 1$); inialisasi nilai derajat keanggotaan μ_{ij} , $i = 1, 2, \dots, n$; $j = 1, 2, \dots, c$.
2. Hitung pusat *cluster* z_j , $j = 1, 2, \dots, c$ menggunakan Persamaan (1).

$$z_j = \frac{\sum_{i=1}^n \mu_{ij}^m o_i}{\sum_{i=1}^n \mu_{ij}^m} \quad (1)$$

3. Hitung jarak *Euclidean* d_{ij} , $i = 1, 2, \dots, n$; $j = 1, 2, \dots, c$
4. Perbarui nilai derajat keanggotaan μ_{ij} , $i = 1, 2, \dots, n$; $j = 1, 2, \dots, c$ menggunakan Persamaan (2).

$$\mu_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{ik}}\right)^{\frac{2}{m-1}}} \quad (2)$$

5. Jika nilai fungsi objektif berkurang, dilanjutkan kembali ke langkah 2, jika tidak maka algoritma dihentikan.

2.3 Fuzzy Particle Swarm Optimization (FPSO)

Algoritma *particle swarm optimization* adalah algoritma komputasi evolusioner yang diusulkan oleh Kennedy dan Eberhart di tahun 1995. Terinspirasi oleh metafora interaksi sosial, algoritma PSO dikembangkan untuk menyimulasikan perilaku sosial kerumunan ikan atau burung, di mana setiap partikel mewakili kandidat solusi dari masalah penelitian [8]. Pang et al. pada tahun mengusulkan algoritma modifikasi dari PSO untuk permasalahan *Traveling Salesman Problem* (TSP) pada tahun 2004, metode ini dinamakan FPSO [7]. Pada penelitian tersebut, kecepatan dan posisi partikel didefinisikan kembali untuk merepresentasikan relasi *fuzzy* diantara variabel. Dalam pencarian posisinya yang optimal, nilai *velocity* (kecepatan) dan posisi partikel harus selalu diperbaiki. Nilai *fitness* dari setiap posisi partikel diketahui menggunakan fungsi *fitness*. Nilai kecepatan partikel ditentukan oleh dua nilai posisi terbaik, *personal best position* (*pbest*) dan *global best position* (*gbest*). *Pbest* merupakan posisi terbaik dari partikel dan *gbest* merupakan posisi terbaik dari *swarm* (kelompok). Algoritma dari FPSO adalah sebagai berikut [7].

1. Berikan nilai awal parameter, termasuk ukuran populasi P , c_1 , c_2 , w dan jumlah iterasi terbesar.
2. Buatlah sebuah kelompok dengan P partikel (X , V , $pbest$, dan $gbest$ pada matriks $n \times c$).
3. Berikan nilai awal untuk variabel X , V , $pbest$ untuk partikel dan $gbest$ untuk kelompok.
4. Tentukan *centroid cluster* pada setiap partikel menggunakan Persamaan (1).
5. Tentukan nilai *fitness* dari setiap partikel menggunakan Persamaan (3).

$$f(X) = \frac{K}{J_m} \quad (3)$$

6. Tentukan nilai *pbest* untuk masing-masing partikel.
7. Tentukan nilai *gbest* untuk masing-masing kelompok.
8. Perbaiki nilai matriks kecepatan untuk masing-masing partikel menggunakan Persamaan (4).

$$V(t+1)w \otimes V(t) \oplus (c_1 r_1) \otimes (pbest(t) \ominus X(t)) \oplus (c_2 r_2) \otimes (gbest(t) \ominus X(t)) \quad (4)$$

dengan,

$V(t)$	=	kecepatan partikel pada iterasi t
w	=	bobot insersia
c_1, c_2	=	koefisien akselerasi
r_1, r_2	=	nilai random antara 0 dan 1
$X(t)$	=	posisi partikel pada iterasi t
$pbest(t)$	=	posisi terbaik partikel sampai iterasi t
$gbest(t)$	=	posisi terbaik kelompok sampai iterasi t

9. Perbaiki matriks posisi untuk setiap partikel menggunakan Persamaan (5).

$$X(t+1) = X(t) \oplus V(t+1) \quad (5)$$

10. Jika telah mencapai iterasi maksimum atau *gbest* tidak mengalami penambahan nilai, hentikan algoritma, jika tidak lanjutkan kembali ke langkah empat.

2.4 Algoritma Hibrid FCM -FPSO

Algoritma FCM-FPSO adalah algoritma hibrida dari algoritma FCM dan FPSO. Algoritma ini mengatasi kelemahan dari algoritma FCM. Algoritma FCM meski lebih cepat daripada algoritma FPSO karena hanya menggunakan sedikit fungsi evaluasi, namun FCM dapat terperangkap dalam kondisi optimum lokal [7]. Algoritma FCM-FPSO mengintegrasikan algoritma FCM dengan algoritma FPSO untuk membentuk suatu algoritma *clustering*. Algoritma FCM-FPSO mengaplikasikan FCM pada setiap partikel di kelompok untuk setiap iterasi ketika nilai *fitness* partikel ditingkatkan. Algoritma FCM-FPSO disajikan sebagai berikut [7].

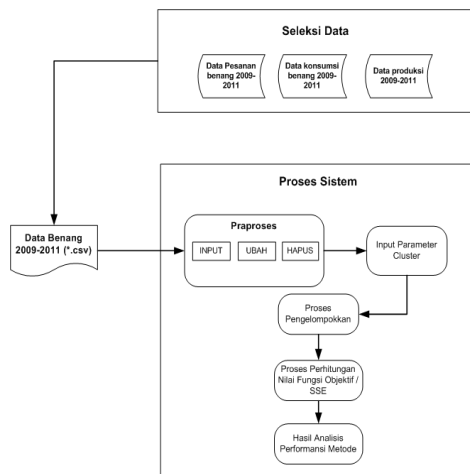
1. Inisialisasi parameter dari FPSO dan FCM (ukuran nilai P , c_1 , c_2 , w dan m).
2. Ciptakan sebuah kelompok dengan partikel P (X , V , $pbest$, dan $gbest$ pada matriks $n \times c$).
3. Inisialisasi X , V , $pbest$ untuk masing-masing partikel dan $gbest$ untuk kelompok.
4. Algoritma FPSO:
 - a. Tentukan *centroid cluster* untuk masing-masing partikel menggunakan Persamaan (1).
 - b. Tentukan nilai *fitness* dari masing-masing partikel menggunakan Persamaan (3).
 - c. Tentukan nilai *pbest* untuk masing-masing partikel.
 - d. Tentukan nilai *gbest* untuk masing-masing kelompok.
 - e. Perbaiki nilai matriks kecepatan untuk masing-masing partikel menggunakan Persamaan (4).

- f. Perbaiki nilai matriks posisi untuk masing-masing partikel menggunakan Persamaan (5).
 - g. Jika telah mencapai iterasi maksimum atau $gbest$ tidak mengalami penambahan nilai, hentikan algoritma, jika tidak lanjutkan kembali ke langkah empat.
5. Algoritma FCM:
 - a. Tentukan *centroid cluster* untuk masing-masing partikel menggunakan Persamaan (1).
 - b. Hitunglah jarak *Euclidean* $d_{ij}, i = 1, 2, \dots, n; j = 1, 2, \dots, c$.
 - c. Perbarui nilai derajat keanggotaan $\mu_{ij}, i = 1, 2, \dots, n; j = 1, 2, \dots, c$ untuk masing-masing partikel menggunakan Persamaan 2.
 - d. Tentukan nilai $pbest$ untuk setiap partikel.
 - e. Tentukan nilai $gbest$ untuk setiap kelompok.
 - f. Jika kondisi berhenti algoritma FCM tidak terpenuhi, lanjutkan kembali ke langkah lima.
 6. Jika kondisi berhenti FCM-FPSO tidak terpenuhi, lanjutkan kembali ke langkah empat.

2.5 Model Pengembangan Perangkat Lunak metode *Unified Process*

Unified Process (UP) adalah metode proses pengembangan perangkat lunak yang menyajikan pengembangan perangkat lunak sebagai aktivitas dari empat fase. Keempat fase tersebut terdiri dari tahapan *inception* (analisis kebutuhan), *elaboration* (pendefinisian arsitektur), *construction* (implementasi), dan *transition* (rilis aplikasi). Pada fase *inception* dilakukan proses analisis bisnis pada sistem. Fase *elaboration* dilakukan proses pendefinisian arsitektur sistem. Selanjutnya adalah fase *construction*, yaitu fase mengimplementasikan sistem, Selanjutnya, fase keempat yaitu *transition*. Pada fase ini dilakukan proses *deploy* aplikasi pada lingkungan *customer* [9].

3. METODE PENELITIAN



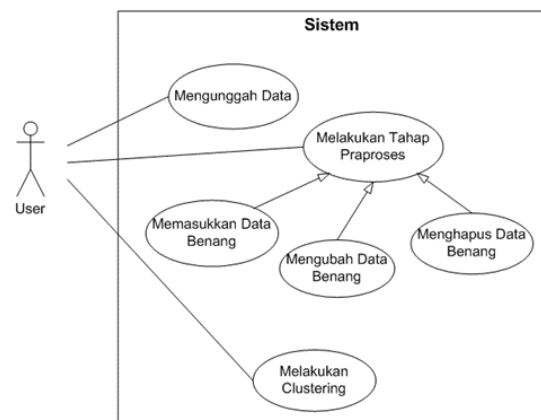
Gambar 1. Tahapan Penelitian

Tahap pertama dilakukan seleksi data, yaitu melakukan proses pemilihan dan penyimpanan data benang yang akan disimpan ke dalam suatu *database* atau dokumen yang berbeda, dalam hal ini dokumen yang dihasilkan berupa file *comma separated values* (CSV). File ini berisi data benang dengan atribut *id_warna*, warna benang, panjang benang, jumlah konsumsi benang, jumlah produk yang menggunakan benang dan jumlah pemesanan benang. Data yang dihimpun merupakan data produksi dan konsumsi benang dari tahun 2009 sampai dengan 2011. Data benang (n) yang dikelompokkan yaitu tahun 2009 sebanyak 96 data, tahun 2010 sebanyak 57 data, tahun 2011 sebanyak 177 data dan data keseluruhan dari tahun 2009 s.d. 2011 sebanyak 209 data. Data berisi *field* *id_benang*, warna, panjang, jumlah stok, jumlah konsumsi, dan jumlah pemesanan.

Tahap pra proses merupakan proses pembersihan data benang, mencakup penghapusan duplikasi data, mengecek data benang yang tidak konsisten serta memperbaiki kesalahan data, seperti kesalahan tulis pada atribut data. Pada aplikasi yang dikembangkan ini disediakan beberapa sub proses dari pra proses, yang terdiri dari proses menambah data benang, mengubah data benang dan menghapus data benang. Selanjutnya, tahap transformasi adalah implementasi data ke dalam kode program agar sesuai dengan tahapan *data mining* yang digunakan. Proses *data mining* pada penelitian ini adalah tahapan pengimplementasian metode *clustering* pada kode program. Tahapan terakhir adalah evaluasi, yaitu tahapan mendapatkan *knowledge* yang dihasilkan dari proses *data mining*.

4. HASIL DAN PEMBAHASAN

Pengembangan aplikasi pengelompokan data benang (PPDB) menngadaptasi metode *unified process*. Pada fase *inception* dilakukan analisis kebutuhan bisnis dari sistem. Kebutuhan fungsional dari aplikasi PPDB ini dapat dilihat pada Gambar 2.



Gambar 3. Diagram Use case Aplikasi PPDB

Terdapat enam *use case* yang didefinisikan pada aplikasi PPDB. *Use case* tersebut yaitu, mengunggah data, melakukan tahap praproses, memasukkan atau menambah data, mengubah data, menghapus data, serta melakukan *clustering*. *Use case* memasukkan, mengubah, dan menghapus data merupakan generalisasi dari *use case* melakukan tahapan praproses.

Pada fase *elaboration*, didefinisikan arsitektur dari sistem berupa kelas diagram. Terdapat 14 kelas yang digunakan oleh aplikasi PPDB. Deskripsi dari keempat belas tersebut disajikan pada Tabel 1.

Tabel 2. Deskripsi Kelas Aplikasi PPDB

N O	Nama Kelas	Tipe Kelas	Deskripsi
1	V_main	Boundary	Kelas ini berfungsi untuk menghubungkan pengguna dengan form praproses dan <i>clustering</i> . Kelas ini juga digunakan untuk proses mengunggah data.
2	C_unggah	Control	Kelas ini berisi <i>method</i> untuk menampilkan form praproses dan <i>clustering</i> . Kelas ini juga berisi <i>method</i> untuk mengunggah data
3	V_praproses	Boundary	Kelas ini merupakan antarmuka yang menampilkan tabel data benang. Kelas ini berhubungan dengan C_praproses dan TabelBenang.
4	C_praproses	Control	Kelas ini berisi <i>method</i> untuk mengolah data benang pada tabel benang. <i>Method</i> tersebut yaitu insert, ubah dan hapus.
5	Benang	Entity	Kelas ini berisi <i>setter</i> dan <i>getter</i> terhadap data benang (id, warna, panjang, konsumsi, produk dan pesanan)
6	TabelBenang	Control	Kelas ini berisi <i>method</i> untuk menampilkan data benang ke dalam tabel pada interface V_praproses.
7	V_cluster	Boundary	Merupakan antarmuka yang menerima inputan parameter dan menampilkan hasil <i>clustering</i> .
8	C_cluster	Control	Kelas ini berisi <i>method</i> untuk mengelompokkan data dan perhitungan nilai best, avg dan worst setiap metode.
9	Cluster	Entity	Kelas ini berisi <i>setter</i> dan <i>getter</i> terhadap data <i>cluster</i>
10	TabelCluster	Control	Kelas ini berisi <i>method</i> untuk menampilkan data <i>cluster</i> ke dalam bentuk tabel pada interface V_cluster.
11	FCM	Control	Merupakan kelas yang berisi atribut dan <i>method</i> pengelompokan yang

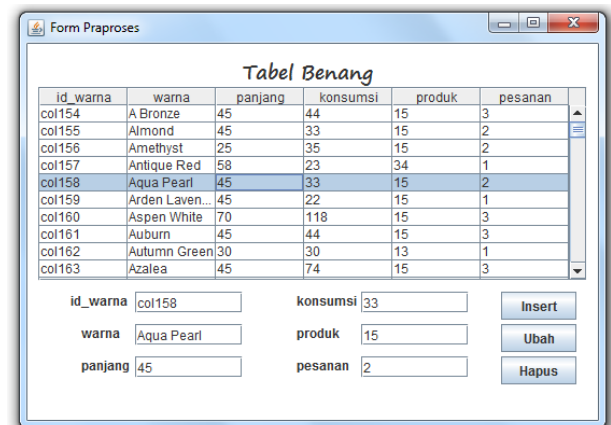
N O	Nama Kelas	Tipe Kelas	Deskripsi
			digunakan pada algoritma FCM.
12	FPSO	Control	Merupakan kelas yang berisi atribut dan <i>method</i> pengelompokan yang digunakan pada algoritma FPSO. Kelas ini merupakan turunan dari kelas FCM
13	FCMFPSO	Control	Merupakan kelas yang berisi atribut dan <i>method</i> pengelompokan yang digunakan pada algoritma FCM-FPSO. Kelas ini merupakan turunan dari kelas FPSO
14	Parameter	Entity	Kelas ini berisi nilai parameter yang merupakan inputan dari form V_Cluster.

Selanjutnya adalah fase *construction*. Implementasi sistem dilakukan menggunakan bahasa pemrograman Java. Tampilan awal dari aplikasi PPDB dapat dilihat pada Gambar 4. Pada halaman awal, *user* dapat menggenerate file *.csv untuk mengimpor data di file ke dalam aplikasi.



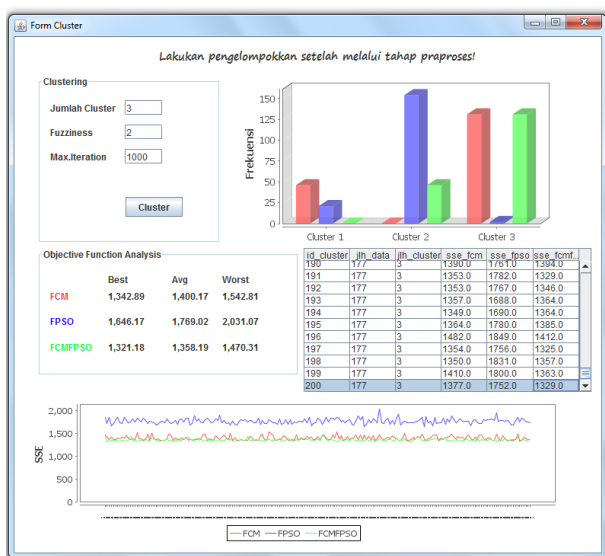
Gambar 5. Tampilan Awal Aplikasi PPDB

Data yang diimpor kemudian ditampilkan pada *form* praproses yang disajikan pada Gambar 6.



Gambar 7. Form Praproses pada Aplikasi PPDB

Tampilan selanjutnya adalah form *clustering* yang disajikan pada Gambar 8. Pada proses *input* parameter, nilai awal yang perlu dimasukkan pada sistem adalah jumlah *cluster* (c), nilai bobot *fuzzy* (m), dengan syarat nilai c dan m harus lebih besar dari satu ($c > 1$ dan $m > 1$) dan jumlah iterasi maksimum berlangsungnya proses pengelompokan (t). Ketiga nilai tersebut perlu diinisialisasi oleh pengguna karena merupakan parameter yang berpengaruh pada setiap algoritma yang digunakan. Nilai *random* dan koefisien akselerasi ditentukan oleh sistem, begitu juga untuk inisialisasi matriks-matriks parameter lainnya. Sementara itu, untuk dimensi matriks ditentukan oleh banyaknya baris data benang (n) dan kolom jumlah *cluster* yang dibentuk (c).



Gambar 9. Form Clustering

Enam kebutuhan fungsional yang digambarkan pada diagram *use case* diuji berdasarkan skenario pengujian metode *black-box*. Skenario terdiri dari *input* dan *output* yang diharapkan pada sistem. Tabel 3. menyajikan skenario pengujian untuk kebutuhan fungsional “Melakukan *Clustering*”. Hasil pengujian menunjukkan bahwa semua skenario yang diuji menghasilkan keluaran yang sesuai dengan harapan.

Tabel 4. Skenario Pengujian Use Case Melakukan Clustering

ID	Prosedur <i>Input</i>	<i>Output</i>	Kesimpulan
U-3-101	Memilih Menu Clustering -> Menekan menu Clustering	Tampilan form Clustering	Diterima
U-3-102	Mengisi inisialisasi parameter dengan huruf “a” -> Menekan menu Clustering	Notifikasi kesalahan <i>input</i>	Diterima
U-3-103	Mengisi jumlah cluster & fuzziness dengan nilai < 2 -> Menekan menu Clustering	Notifikasi kesalahan <i>input</i>	Diterima

U-3-104	Melakukan pengelompokan -> Menekan menu Clustering	Hasil cluster disimpan & grafik ditampilkan	Diterima
---------	--	---	----------

Pengelompokan data benang pada aplikasi PPDB dilakukan sebanyak 200 kali untuk setiap data benang (pertahun dan keseluruhan). Jumlah *cluster* (c) yaitu sebanyak 3 *cluster* untuk tahun 2011 dan data keseluruhan, dan sebanyak 5 *cluster* untuk tahun 2009 dan 2010. Nilai *fuzziness* $m = 2$, nilai *random* (r_1, r_2) dalam rentang $[0,1]$, nilai koefisien akselerasi (c_1, c_2) adalah 2 dan nilai bobot inersia (w) dalam rentang $[0,9, 0,1]$. Hasil perhitungan nilai fungsi objektif pada setiap metode disajikan pada Tabel 5.

Tabel 6. Hasil Clustering Data Benang

Data Benang (n,c)		2009 (96, 5)	2010 (57, 5)	2011 (177, 3)	Mix (209, 3)
FCM	<i>Worst</i>	385,1	278,5	1342,8	3413,4
	<i>Average</i>	348,1	293,2	1400,1	3540,3
	<i>Best</i>	8	2	7	3
FPSO	<i>Worst</i>	490,9	378,2	1646,1	4213,1
	<i>Average</i>	517,9	408,6	1769,0	4485,4
	<i>Best</i>	2	2	1	3801,4
FCMFPSO	<i>Worst</i>	561,6	450	2031,0	4991,5
	<i>Average</i>	374,7	250,5	1321,1	3326,7
	<i>Best</i>	5	2	8	6
O	<i>Worst</i>	345,1	282,6	1358,1	3441
	<i>Average</i>	7	2	9	3657,6
	<i>Best</i>	319,8	315,0	1470,3	4

Tabel 7. menunjukkan metode FCM-FPSO menghasilkan nilai fungsi objektif paling minimum atau terkecil dari kedua metode lainnya. Nilai rata-rata fungsi objektif terendah dicapai oleh algoritma hibrid FCM-FPSO sebesar 3441,00 kemudian diikuti oleh algoritma FCM dengan nilai sebesar 3540,33 dan yang tertinggi dicapai oleh algoritma FPSO dengan nilai sebesar 4485,40. Semakin kecil nilai fungsi objektif yang dihasilkan maka semakin baik *cluster* yang terbentuk [7]. Hal ini menunjukkan *cluster* yang dihasilkan oleh metode FCM-FPSO adalah yang terbaik dibandingkan kedua metode lainnya yang telah diuji.

5. KESIMPULAN

Penelitian ini menguji performansi metode hibrid FCM-FPSO dengan mengimplementasikannya ke sebuah aplikasi pengelompokan data benang pada perusahaan tekstil. Metode dengan nilai performa terbaik untuk pengelompokan data benang, yaitu metode hibrid FCM-FPSO, dengan nilai rata-rata fungsi objektif terendah mencapai angka 3441,00, paling kecil di antara dua algoritma lainnya. Hal ini menunjukkan bahwa aplikasi yang dikembangkan

berhasil menguji performa metode FCM-FPSO. Hasil *cluster* metode FCM-FPSO yang didapatkan dari aplikasi PPDB menjadi rekomendasi untuk perusahaan tekstil untuk menyusun strategi pembelian benang untuk periode yang akan datang.

DAFTAR PUSTAKA

- [1] K. Mehmed, *Data Mining: Concepts, Models, Methods, and Algorithms*, 3rd ed. Hoes Lane: IEEE Press & Wiley, 2020.
- [2] R. Rustiyan and M. Mustakim, "Penerapan Algoritma Fuzzy C Means untuk Analisis Permasalahan Simpanan Wajib Anggota Koperasi," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 2, p. 171, 2018.
- [3] S. Rani, A. Puspita, and F. Nhita, "Implementasi Algoritma Fuzzy C-Means untuk Mengkategorikan Tingkat Penjualan Produk pada Data Transaksi Swalayan," *e-Proceeding Eng.*, vol. 5, no. 3, pp. 8206–8217, 2018.
- [4] F. Agustini, "Implementasi Algoritma Fuzzy C-Means Studi Kasus Penjualan di Sushigroove Restaurant," *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 3, no. 1, pp. 127–132, 2019.
- [5] T. M. Silva Filho, B. A. Pimentel, R. M. C. R. Souza, and A. L. I. Oliveira, "Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization," *Expert Syst. Appl.*, vol. 42, no. 17–18, pp. 6315–6328, 2015.
- [6] C. Skiadas and J. Bozeman, *Data Analysis and Applications 1*, 2nd ed. Wiley, 2019.
- [7] H. Izakian and A. Abraham, "Fuzzy C-means and fuzzy swarm for fuzzy clustering problem," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 1835–1838, 2011.
- [8] W. Liu, Z. Wang, X. Liu, N. Zeng, and D. Bell, "A Novel Particle Swarm Optimization Approach for Patient Clustering from Emergency Departments," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 632–644, 2019.
- [9] I. Sommerville, *Software Engineering-Sommerville(Tenth Edition)*, 10th ed. Pearson Education, 2016.