

Clustering Nasabah Bank Berdasarkan Tingkat Likuiditas Menggunakan *Hybrid* Particle Swarm Optimization dengan K-Means

Ida Wahyuni¹, Yudha Alif Auliya², Asyrofa Rahmi³, Wayan Firdaus Mahmudy⁴

Fakultas Ilmu Komputer, Universitas Brawijaya

¹ida.wahyuni8@gmail.com, ²yudha.alif7@gmail.com, ³asyrofarahmi@gmail.com, ⁴wayanfm@ub.ac.id

ABSTRAK. Setiap Bank memiliki layanan dalam meminjamkan modal kepada suatu perusahaan. Namun nominal pinjaman modal tidaklah sedikit. Sehingga untuk mencegah pengembalian modal dapat dilakukan dengan lancar, diperlukan clustering perusahaan berdasarkan analisa likuiditas. Pada penelitian ini, clustering dilakukan menggunakan *hybrid* Particle Swarm Optimization dengan K-Means (PSO-KMeans). Metode *hybrid* tersebut digunakan untuk mendapatkan hasil cluster yang tidak terjebak dalam solusi optimum lokal. Hasil yang diperoleh dari hybrid PSO K-Means menunjukkan hasil yang lebih baik jika dibandingkan dengan menggunakan algoritma K-Means tanpa *hybrid*. Hal ini dibuktikan dengan perolehan centroid terbaik yang ditunjukkan dengan nilai Silhouette Coefficient yang diperoleh *hybrid* PSO K-Means lebih baik dibandingkan K-Means.

Kata Kunci: K-Means, Particle Swarm Optimization, Hybrid PSO K-Means, Clustering, Likuiditas.

1. PENDAHULUAN

Likuiditas merupakan kecukupan sumber kas perusahaan untuk memenuhi kewajiban yang berkaitan dengan kas dalam jangka pendek [1]. Apabila sebuah perusahaan tidak lagi mempunyai tingkat likuiditas yang cukup untuk memenuhi kebutuhan perusahaannya, maka perusahaan tersebut bisa dikatakan akan mengalami kebangkrutan. Analisis likuiditas penting dilakukan, khususnya untuk para nasabah bank yang mempunyai usaha dan mempunyai pinjaman pada bank tersebut. Dengan mengetahui tingkat likuiditas seorang nasabah yang mempunyai pinjaman, maka pihak bank dapat mengetahui apakah pinjaman yang mereka berikan dapat dikembalikan dengan lancar atau tidak. Hal ini penting bagi sebuah lembaga pengelola keuangan khususnya bank agar *return* atau tingkat pengembalian dana dan pendapatan bunga untuk investasi pada surat hutang dapat berjalan dengan lancar [2].

Analisis likuiditas yang bisa dilakukan adalah dengan melakukan pengelompokan terhadap nasabah yang mempunyai pinjaman ke kelompok bangkrut dan tidak bangkrut. Hal ini bisa diketahui dengan beberapa data nasabah yaitu dari lamanya perusahaan tersebut berdiri dan lamanya keterlambatan dalam pembayaran cicilan. Dengan menggunakan kedua data tersebut untuk pengelompokan, maka akan didapatkan data nasabah mana saja yang sekiranya akan bangkrut dan pihak bank dapat mengambil tindakan selanjutnya untuk meminimalisir kerugian.

K-Means merupakan algoritma *clustering* sederhana yang bersifat tanpa arahan (*unsupervised*) [3]. Tujuan *clustering* adalah mengelompokkan obyek ke dalam *kcluster* atau kelompok. Pada *K-Means*, nilai *k* harus ditentukan terlebih dahulu dengan mempertimbangkan ukuran ketidakmiripan dalam mengelompokkan objek yang ada. Selain menentukan jumlah cluster, parameter lain yang harus ditentukan adalah pusat cluster secara acak. Semakin baik penentuan pusat cluster, maka semakin tepat dan cepat proses pengelompokan menggunakan K-Means Clustering. Namun, karena pusat cluster ditentukan secara acak, maka tingkat keakuratannya terkadang kurang baik dan sering terjadi optimum lokal [4]. Hal tersebut dapat diatasi dengan menggunakan sebuah algoritma optimasi untuk memperbaiki pusat cluster pertama, sehingga didapatkan pusat cluster yang paling optimal untuk perhitungan K-Means.

Salah satu algoritma optimasi yang bisa digunakan untuk mengoptimasi pusat cluster pada K-Means adalah Particle Swarm Optimization (PSO) [5]. PSO bisa digunakan pada permasalahan clustering karena menggunakan representasi solusi bilangan real [6]. Particle Swarm Optimization terinspirasi dari fenomena alam dari perilaku burung [7]. Penggabungan algoritma Dynamic Particle Swarm Optimization dan K-Means pernah dilakukan oleh Li, He, & Wen [8] untuk mengoptimasi pusat cluster awal sehingga dapat mencegah terjadinya *local optimum* pada *image segmentation*. Penelitian lain yang menggunakan hybrid K-Means PSO juga pernah dilakukan oleh Cheng, Huang, & Chen [9] untuk menyelesaikan masalah multidimensional dengan mengusulkan K-Means Clustering Particle Swarm Optimization (KCPSO) sebagai solusi. Hybrid K-Means PSO juga pernah digunakan oleh Chiu, et al. [10] untuk mengatasi masalah *market segmentation*.

Penelitian akan dimulai dengan mengumpulkan data nasabah yang diperlukan sebagai parameter input, yaitu usia perusahaan dan lamanya keterlambatan pembayaran cicilan. Ada seratus dataset yang akan dikelompokkan menjadi dua *cluster* yaitu *cluster* bangkrut = ‘Ya’ dan bangkrut = ‘Tidak’. Pada perhitungan algoritma K-Means, pusat *cluster* pertama dioptimasi terlebih dahulu menggunakan Particle Swarm Optimization (PSO) sehingga hasil clustering akan maksimal dan optimum lokal dapat dicegah.

2. PENELITIAN TERDAHULU

Penelitian sebelumnya yang menggunakan penggabungan metode K-Means dengan Particle Swarm Optimization (PSO) pernah dilakukan oleh Cheng, Huang, & Chen [9] untuk menyelesaikan masalah multidimensional. Penelitian ini proposed approach inherited the paradigm in Particle Swarm Optimization (PSO) to implement a chaotic search around global best position (gbest) and enhanced by K-means clustering algorithm, named KCPSO. Penelitian lain yang menggunakan metode yang sama juga pernah dilakukan oleh Chiu, et al. [10]. Pada kedua penelitian tersebut proses penghitungan *fitness* setiap particle menggunakan rumus Euclidean distance untuk mengetahui baik tidaknya hasil clustering. Karena metode perhitungan *fitness* yang digunakan cukup sederhana maka waktu komputasi yang diperlukan juga singkat. Namun, metode perhitungan *fitness* dengan Euclidean distance dirasa belum terlalu optimal untuk mengukur seberapa baik cluster yang dihasilkan. Oleh karena itu, penelitian ini akan mencoba untuk menerapkan metode hybrid PSO K-Means dengan menerapkan rumus *Silhouette Koefisien* sebagai rumus untuk menghitung *fitness* pada setiap particle dan melihat waktu komputasi yang dibutuhkan serta seberapa baik clustering yang dihasilkan.

3. ANALISIS LIKUIDITAS

Likuiditas merupakan kecukupan sumber kas perusahaan untuk memenuhi kewajiban yang berkaitan dengan kas dalam jangka pendek [1]. Analisis likuiditas khususnya pada dunia perbankan dapat dilakukan dengan mengelompokkan nasabah menjadi dua kelompok yaitu kelompok nasabah yang mempunyai usaha dengan keadaan bangkrut dan tidak bangkrut. Pengelompokan tersebut dapat dilihat dari beberapa data antara lain adalah data lamanya perusahaan tersebut berdiri atau *business age* dan data lamanya keterlambatan dalam pembayaran cicilan atau *days delinquent*. Data *business age* dan *days delinquent* dapat dilihat pada Table I.

Tabel 1.Data Business Age dan Days Delinquent

No	Business Age	Days Delinquent
1	87	2
2	89	2
3	90	2
4	90	2
5	101	2
...
396	373	14
397	29	19
398	349	65
399	237	3
400	329	43

4. K-MEANS CLUSTERING

Seperti yang sudah dijelaskan pada sub-bab sebelumnya, K-Means merupakan algoritma *clustering* sederhana yang bersifat tanpa arahan (*unsupervised*) [3]. Misalkan *D* adalah sebuah dataset dari *n* objek, dan *k* adalah jumlah *cluster* yang akan dibentuk, algoritma partisi mengatur objek-objek tersebut ke dalam partisi *k* ($k \leq n$), dimana setiap partisi menggambarkan sebuah *cluster*. Setiap *cluster* dibentuk untuk mengoptimalkan kriteria partisi, seperti fungsi perbedaan berdasarkan jarak, sehingga objek-objek di dalam sebuah *cluster* adalah mirip, sedangkan objek-objek pada *cluster* yang berbeda adalah tidak mirip dalam hal atribut dataset [11]. Persamaan untuk menghitung jarak antar data pada K-Means menggunakan rumus *Euclidean Distance (D)* yang ditunjukkan pada Persamaan (1).

$$D(x_2, x_1) = \sqrt{\sum_{j=1}^p (x_{2j} - x_{1j})^2} \tag{1}$$

Keterangan:

p = Dimensi data

x_1 = Posisi titik 1

x_2 = Posisi titik 2

Algoritma standart dari K-Means adalah sebagai berikut [12]:

1. Inisialisasi *centroid* K secara acak.
2. Menetapkan setiap objek ke grup dengan *centroid* terdekat. Gunakan rumus *Euclidean Distance* untuk mengukur jarak minimum antara objek data dan setiap *centroid*. Persamaan
3. Hitung Ulang vektor *centroid*, menggunakan Persamaan 2.

$$m_j = \frac{1}{n_j} \sum_{\forall \text{ data}_p \in C_j} \text{data}_p \tag{2}$$

di mana m_j menunjukkan vektor *centroid* dari cluster j , n_j adalah jumlah data vektor dalam cluster j , C_j adalah bagian dari data vektor dari cluster j , dan data_p menunjukkan vektor data.

4. Ulangi langkah 2 sampai centroid tidak berubah lagi dalam sesuai dengan maksimum jumlah iterasi yang telah ditetapkan.

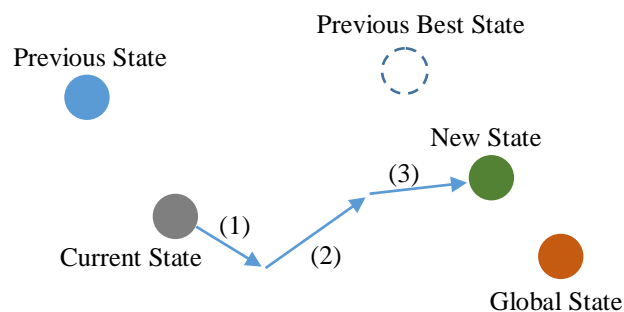
5. PARTICLE SWARM OPTIMIZATION (PSO)

Particle Swarm Optimization adalah salah satu algoritma berbasis populasi yang terinspirasi dari fenomena alam dari perilaku burung dan ikan [7]. Ini adalah metode meta-heuristik berbasis populasi yang mengoptimalkan masalah dengan menginisialisasi kawanan burung secara acak dalam ruang pencarian di mana setiap burung disebut sebagai "particle" dan *population of particles* yang disebut "swarm" [12]. Partikel-partikel bergerak iteratif di dalam ruang pencarian yang dihitung dengan perpindahan posisi partikel dan rumus perubahan kecepatan untuk menemukan posisi terbaik global. Dalam ruang pencarian n -dimensi, posisi dan kecepatan partikel i di t iterasi dilambangkan dengan vektor $X_i(t) = (X_{i1}(t), X_{i2}(t), \dots, X_{in}(t))$ dan vektor $V_i(t) = (V_{i1}(t), V_{i2}(t), \dots, V_{in}(t))$. Solusi ini dievaluasi oleh fungsi biaya atau *cost* untuk setiap partikel pada setiap tahap algoritma untuk memberikan nilai kuantitatif utilitas solusi ini. Setelah itu, catatan posisi terbaik dari setiap partikel berdasarkan nilai biaya disimpan. Posisi terbaik dikunjungi sebelumnya dari partikel i pada tahap saat ini dilambangkan oleh vektor $P_i = (P_{i1}, P_{i2}, \dots, P_{in})$ sebagai *personal best*. Selama proses ini, posisi semua partikel yang memberikan biaya terbaik sampai tahap saat ini juga tercatat sebagai posisi terbaik global yang dilambangkan dengan $G = (g_1, g_2, \dots, g_n)$. Struktur kecepatan dan update posisi digambarkan pada Gambar 1. Setiap iterasi terdiri dari tiga gerakan: dalam gerakan pertama, partikel bergerak sedikit ke arah depan ke arah sebelumnya dengan kecepatan yang sama. Dalam gerakan kedua, bergerak sedikit ke arah posisi terbaik sebelumnya. Akhirnya, dalam gerakan ketiga, bergerak sedikit ke arah posisi global. Pada setiap iterasi, kecepatan dan posisi setiap partikel didefinisikan menurut Persamaan (3) dan (4), masing-masing:

$$V_i(t) = \omega * V_i(t - 1) + c_1 \phi_1 (P_i - X_i(t - 1)) + c_2 \phi_2 (G - X_i(t - 1)) \tag{3}$$

$$X_i(t) = X_i(t - 1) + V_i(t) \tag{4}$$

Dimana ω menunjukkan inersia yang memperkenalkan preferensi untuk partikel agar terus bergerak ke arah yang sama.



Gambar 1. Deskripsi Update Kecepatan dan Posisi di PSO untuk Ruang Parameter 2-Dimensi

6. SILHOUETTE KOEFISIEN

Silhouette Koefisien adalah sebuah teknik yang digunakan untuk mengukur seberapa baik letak objek dalam cluster dengan memberikan sebuah representasi grafis singkat [13]. Metode ini merupakan gabungan dari metode *cohesion* dan *separation*[14]. Pada penelitian ini, *Silhouette Koefisien* akan digunakan untuk menghitung nilai *fitness* yang dimiliki oleh setiap *particle* pada proses PSO. Berikut ini adalah tahapan perhitungan Rumus *Silhouette Koefisien*:

1. Hitung rata-rata jarak dari suatu dokumen misalkan *i* dengan semua dokumen lain yang berpada dalam satu *cluster*

$$\alpha(i) = \frac{1}{|A|} \sum_{j \in A, j \neq i} C d(i, j) \tag{5}$$

Keterangan:

j = dokumen lain dalam satu cluster A.

d(i,j) = jarak antar dokumen *i* dengan *j*

2. Hitung rata-rata jarak dari dokumen *i* tersebut dengan semua dokumen di *cluster* lain, dan diambil nilai terkecilnya.

$$d(i, C) = \frac{1}{|A|} \sum_{j \in C} d(i, j) \tag{6}$$

Keterangan:

d(i, C) = jarak rata-rata antar dokumen *i* dengan semua objek pada cluster lain atau C dimana $A \neq C$.

$$b(i) = \min_{C \neq A} d(i, C) \tag{7}$$

3. Nilai *Silhouette Koefisien* nya dihitung menggunakan Persamaan (8) berikut ini:

$$s(i) = \frac{b(i) - \alpha(i)}{\max(\alpha(i), b(i))} \tag{8}$$

Rata-rata *s(i)* dari seluruh data dalam suatu cluster menunjukkan seberapa dekat kemiripan data dalam suatu cluster yang juga menunjukkan seberapa tepat data telah dikelompokkan.

7. METODOLOGI

Proses klasifikasi nasabah bank berdasarkan tingkat likuiditas pada penelitian ini akan dioptimalkan pengelompokannya menggunakan beberapa algoritma yaitu algoritma K-Means dan hybrid PSO-KMeans.

A. K-Means Clustering

Langkah pertama dalam proses clustering K-Means ialah memilih pusat cluster secara acak dari data yang sudah disediakan. Berdasarkan jumlah atribut dari data pada Tabel 1 yang berjumlah 2, maka pusat cluster (centroid) dengan jumlah cluster 2 ditunjukkan pada Table 2

Tabel 2. Penentuan Pusat Cluster

Centroid	Business Age	Days Delinquent
1	90	2
2	237	3

Dengan 2 centroid yang sudah terbentuk yaitu centroid 1 (C1) (90,2) dan centroid 2 (C2) (237,2), maka langkah selanjutnya adalah menghitung kedekatan setiap data pada Tabel 1 dengan kedua centroid. Misal mencari kedekatan jarak data pertama (87, 2) dengan kedua centroid menggunakan persamaan (1) sebagai berikut:

$$D(data_1, c_1) = \sqrt{(x_1 - x_1)^2 + (y_1 - y_1)^2} = \sqrt{(87 - 90)^2 + (2 - 2)^2} = 3$$

$$D(data_1, c_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = \sqrt{(87 - 237)^2 + (2 - 3)^2} = 150$$

Hasil perhitungan *euclidean distance* tersebut memperlihatkan bahwa jarak terdekat pada data 1 adalah dengan centroid pertama (C1). Karena nilai jarak data 1 sebesar 3 dan lebih kecil dibandingkan dengan jarak pada C2 dengan jarak sebesar 150. Sehingga dapat disimpulkan bahwa data 1 masuk ke dalam cluster 1. Untuk memasuki iterasi selanjutnya maka dilakukan perubahan centroid menggunakan persamaan (2). Setelah centroid didapatkan maka proses K-Means dilakukan secara terus menerus sampai batas iterasi maksimum.

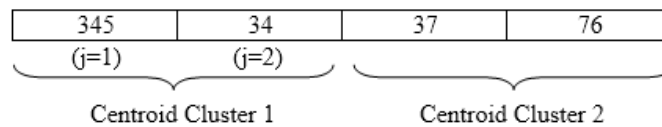
B. Hybrid PSO K-Means Clustering

Siklus PSO K-Means Clustering

- Step1 Inisialisasi partikel yang berisi centroid setiap cluster dan kecepatan partikel
- Step2 Menentukan partikel best dan global best awal
- Step3 Mulai pengulangan dan lakukan :
 Pengulangan sejumlah partikel dan lakukan :
 Setiap data vector lakukan :
 (i) Menghitung menggunakan K-Means Clustering
 (ii) Tentukan cluster setiap vector dengan proses K-Means Clustering
 (iii) Hitung nilai cost menggunakan Silhouette Coefficient
 Pengulangan berhenti ketika partikel yang terakhir
 Update centroid menggunakan persamaan update kecepatan dan update partikel
 Update partikel best dan global best
- Step 4 Pengulangan berhenti ketika iterasi maksimum

C. Inisialisasi Partikel

Dalam proses klasifikasi, setiap partikel pada PSO merepresentasikan kumpulan pusat cluster dengan setiap cluster terdiri dari banyaknya atribut. Sehingga jika cluster dimisalkan $c=\{1, 2, \dots, c\}$ dan atribut diinisialkan $j=\{1, 2, \dots, j\}$ maka setiap partikel memiliki panjang $x=c \times j$. Gambaran sebuah inisialisasi partikel ditunjukkan pada Gambar 2.



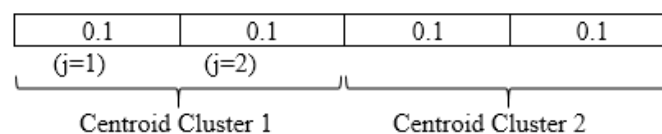
Gambar 2.Inisialisasi Sebuah Partikel

Pada Fig. 2, dua kolom pertama merupakan centroid untuk cluster 1, dua kolom selanjutnya menunjukkan centroid pada cluster 2 dan seterusnya. Pada setiap cluster terdiri dari 2 atribut dengan kolom pertama untuk atribut 1 dan kolom kedua untuk atribut 2. Hal ini berlaku pada keseluruhan centroid.

Nilai untuk suatu partikel diperoleh secara acak dengan range minimal dan maksimal dari keseluruhan data. Dalam permasalahan ini, range yang digunakan adalah 0 sampai 500 pada atribut 1 (j=1) dan antara 0 sampai dengan 100 untuk atribut 2 (j=2).

D. Inisialisasi Kecepatan

Panjang sebuah kecepatan menyerupai panjang sebuah partikel. Sehingga panjang kecepatan merupakan $v=c \times j$. Tujuan dari kecepatan ini merupakan mencari perubahan ke centroid yang lebih optimal. Nilai dari suatu kecepatan merupakan range antara 0 sampai dengan 1. Semakin besar nilai suatu kecepatan maka akan semakin lebar jangka algoritma PSO dalam menemukan suatu solusi. Gambar 3 merupakan gambaran suatu kecepatan.



Gambar 3.Inisialisasi Kecepatan Suatu Partikel

Gambaran kecepatan pada Fig. 3 menunjukkan bahwa nilai 0.1 pada kolom pertama menunjukkan kecepatan pada atribut 1 dan 0,1 pada kolom kedua merupakan kecepatan untuk atribut 2. Kecepatan pada kolom pertama dan kedua merupakan kecepatan untuk centroid pada cluster 1. Hal ini berlaku pada nilai kecepatan pada 2 kolom berikutnya untuk centroid selanjutnya.

Tabel 3.Kecepatan Partikel

P	1	2	3	4
1	0.1	0.1	0.1	0.1
2	0.1	0.1	0.1	0.1
3	0.1	0.1	0.1	0.1
...
20	0.1	0.1	0.1	0.1

E. Inisialisasi Populasi dan Menghitung Nilai Cost

Perhitungan cost dimulai dengan menentukan cluster setiap data pada Tabel 1 dengan mencari jarak terkecil menggunakan rumus *euclidean distance* pada persamaan (1) antara data dan nilai centroid setiap cluster dari partikel yang sudah diinisialisasi sebelumnya. Penentuan cluster untuk setiap data ditunjukkan pada Tabel 4.

Tabel 4.Penentuan Cluster Setiap Data

No	Business Age	Days Delinquent	Cluster
1	87	2	2
2	89	2	2
3	90	2	2
4	90	2	2
5	101	2	2
...
396	373	14	2
397	29	19	2
398	349	65	1
399	237	3	2
400	329	43	1

Suatu partikel dianggap menjadi solusi optimal dengan melihat nilai costnya. Semakin kecil suatu nilai cost maka semakin baik suatu partikel dijadikan sebuah solusi [15]. Namun, nilai cost pada permasalahan clustering ini diperoleh dari nilai silhouette coefficient dengan prinsip semakin besar nilainya (mendekati 1) maka semakin baik suatu centroid digunakan sebagai solusi clustering. Sehingga perhitungan cost diperoleh dari meminimalkan fungsi silhouette coefficient menggunakan persamaan (9).

$$f(x) = C - f(x) \tag{9}$$

Dimana $f(x)$ merupakan nilai cost yang diperoleh dari hasil akhir silhouette coefficient yang diminimalkan. Kemudian C sebagai konstanta awal untuk mengurangi nilai $f(x)$ selaku nilai akhir silhouette coefficient. Dalam penelitian ini, nilai C yang akan digunakan adalah 1 karena nilai $f(x)$ hanya berada dalam range -1 sampai dengan 1.

Proses perhitungan silhouette coefficient diawali dengan menghitung rata-rata jarak suatu data (i) dengan data lain dalam satu cluster a(i). Selanjutnya menghitung nilai b(i) yaitu rata-rata jarak data (i) pada perhitungan sebelumnya dengan data (i) pada cluster yang lain. Karena jumlah cluster ditentukan sejumlah dua maka proses perhitungan jarak dengan cluster lain hanya satu. Tahap terakhir adalah perhitungan nilai s(i) sehingga didapatkan hasil masing-masing perhitungan pada Tabel 5.

Tabel 5.Hasil Setiap Tahap Silhouette Coefficient

No	a(i)	b(i)	s(i)
1	39.12399	57.81842	0.056247
2	39.57504	57.51214	0.047609
3	39.81984	57.36059	0.04292
4	39.81984	57.36059	0.04292
5	43.05241	55.35947	-0.01899
...
96	57.71303	33.15072	-0.69213
97	50.38618	32.4401	-0.51058
98	46.2676	53.36912	-0.08056
99	71.50459	42.68811	-0.56389
100	41.4213	56.36921	0.01225

Nilai silhouette coefficient merupakan rata-rata hasil perhitungan s(i) dan menghasilkan nilai -0.1542. Karena nilai silhouette coefficient harus diminimalkan untuk mendapatkan nilai cost, maka nilai cost yang didapatkan adalah 1.15412 yang dihiitung menggunakan persamaan (7).

Tabel 6 merupakan inialisasi populasi yang diperoleh dari inialisasi partikel beserta hasil akhir perhitungan costnya. Sehingga dalam satu populasi terdiri dari kumpulan partikel sejumlah 20 dengan jumlah populasi sudah ditentukan diawal perhitungan.

Tabel 6.Ilustrasi Pembentukan Populasi

P	1	2	3	4	Cost
1	345	34	37	76	1.15412
2	25	74	108	62	1.1723
3	256	98	10	57	1.1743
...
20	45	63	76	94	1.1837

F. Menentukan Partikel Best dan Global Best Awal

Partikel best ditentukan sejumlah partikel posisi yang diinisialisasikan pada awal perhitungan. Partikel best pada urutan tertentu merupakan partikel terbaik yang dibandingkan dengan partikel urutan tersebut pada iterasi-iterasi sebelumnya. Perbandingan setiap urutan partikel dilihat berdasarkan nilai cost yang telah diperoleh. Sehingga partikel best awal atau partikel best pada iterasi awal akan selalu bernilai sama dengan posisi partikel saat diinisialisasikan yaitu Tabel 5. Karena pada tahap ini, setiap urutan partikel tidak memiliki partikel perbandingan dengan partikel pada iterasi sebelumnya.

Lain halnya dengan global best. Seperti namanya yaitu ‘global’, maka global best didapatkan dari membandingkan keseluruhan partikel dari semua iterasi yang sudah dilalui berdasarkan nilai costnya. Sehingga pada iterasi awal ini, global best diperoleh dari partikel dalam satu populasi yang memiliki nilai cost paling kecil. Berdasarkan populasi dalam Tabel 5, partikel dengan nilai cost terkecil dan paling baik adalah partikel ke 1 sehingga partikel inilah yang ditentukan sebagai global best. Penentuan particle best dan global best awal ini akan digunakan dalam mencari perubahan kecepatan maupun posisi partikel guna mencari centroid baru yang lebih baik dan lebih optimal. Hasil global best awal ditunjukkan pada Gambar 4.

345	34	37	36
-----	----	----	----

Gambar 4.Global Best Awal

G. Update Kecepatan

Proses selanjutnya merupakan update kecepatan. Guna mencari posisi terbaik dalam ruang pencarian, secara iteratif kecepatan mengalami suatu perubahan. Dengan menggunakan persamaan (1), suatu perubahan kecepatan akan menunjukkan nilai yang berbeda.

Dimisalkan diketahui sejak awal bahwa $\omega = 0.5$ selaku inersia, $c_1 = c_2 = 1$ sebagai learning factor kemudian nilai $\varphi_1 = 0.4$ dan $\varphi_2 = 0.5$ merupakan nilai random dengan range 0 sampai dengan 1. Selain itu, partikel best dan global best yang digunakan adalah seperti yang sudah dijelaskan pada subbab menentukan partikel best dan global best awal sebelumnya. Sehingga untuk mencari perubahan kecepatan partikel awal dengan

kecepatan sebelumnya yang ditunjukkan pada Fig. 3, maka didapatkan perubahan kecepatan pada partikel 1 dan kolom 1 sebesar 0.05.

$$V_1(t) = (0.5 * 0.1) + 1 * 0.4 * (345 - 345) + 1 * 0.5 * (345 - 345)$$

$$V_1(t) = 0.05$$

Penjabaran perhitungan tersebut dilakukan pada semua kecepatan partikel dan pada semua kolom-kolomnya. Sehingga setelah dihitung keseluruhan kecepatan, maka dilihat perubahan kecepatan keseluruhan partikel pada Tabel 7.

Tabel 7.Update Kecepatan

P	1	2	3	4
1	0.05	0.05	0.05	0.05
2	160.05	-19.95	-35.45	7.05
3	44.55	-31.95	13.55	9.55
...
20	150.05	-14.45	-19.45	-8.95

H. Update Posisi Partikel

Setelah melalui proses perubahan kecepatan, maka dilakukan proses perubahan posisi partikel untuk menemukan posisi partikel terbaik berdasarkan nilai posisi partikel sebelumnya dan perubahan kecepatan yang dijelaskan pada subbab update kecepatan sebelumnya.

Proses perhitungan posisi partikel dilakukan menggunakan persamaan (2). Dimisalkan mencari perubahan posisi partikel awal yang ditunjukkan pada Fig. 2, maka pada kolom 1 perubahan posisi yang dihasilkan adalah senilai 345.05.

$$X_1(t) = 345 + 0.05 = 345.05$$

Sama halnya dengan perubahan kecepatan, proses perubahan posisi partikel juga dilakukan pada semua kolom dan keseluruhan partikel. Sehingga setelah melalui proses perhitungan didapatkan hasil perubahan posisi partikel dalam satu populasi diperlihatkan dalam Tabel 8.

Tabel 8.Perubahan Posisi Partikel

P	1	2	3	4	Cost
1	345.05	34.05	37.05	76.05	1.1482
2	185.05	54.05	72.55	69.05	1.1596
3	300.55	66.05	23.55	66.55	1.1476
...
20	195.05	48.55	56.55	85.05	1.1629

I. Update Partikel Best dan Global Best

Prinsip penentuan partikel best dan global best setiap iterasi lebih dari 1 adalah membandingkan nilai-nilainya dengan hasil nilai pada iterasi sebelumnya. Dalam penentuan partikel best ini, maka nilai cost pada partikel best iterasi sebelumnya dibandingkan dengan nilai cost pada perubahan partikel yang sudah dihitung. Sehingga, melalui Tabel 6 dan Tabel 8 dapat dihasilkan partikel best terbaru yang ditunjukkan pada Tabel 9.

Tabel 9.Perubahan Partikel Best

P	1	2	3	4	Cost
1	345.05	34.05	37.05	76.05	1.1482
2	185.05	54.05	72.55	69.05	1.1596
3	300.55	66.05	23.55	66.55	1.1476
...
20	195.05	48.55	56.55	85.05	1.1629

Sehingga dimisalkan pada partikel urutan 1. Nilai cost pada partikel pertama pada Tabel 5 senilai 1.15412 dibandingkan dengan nilai cost partikel pertama pada Tabel 7 yaitu 1.1482. Sehingga partikel best pada partikel pertama adalah partikel dengan nilai cost terkecil yaitu partikel pertama pada Tabel 7 yaitu 1.1482

Kemudian dalam penentuan global best, keseluruhan partikel pada perubahan posisi partikel yang baru maupun partikel-partikel pada iterasi-iterasi sebelumnya dibandingkan nilai costnya. Proses pencarian global best ditunjukkan pada Tabel 10.

Tabel 10. Proses Update Global Best Berdasarkan Cost

P	Update Partikel	Partikel Iterasi - 1
1	1.1482	1.15412
2	1.1596	1.1723
3	1.1476	1.1743
...
20	1.1629	1.1837

Berdasarkan proses pencarian global best pada Tabel 10 maka didapatkan nilai cost terkecil dari keseluruhan partikel yaitu sebesar 1.1482 dengan global best yang diartikan sebagai hasil centroid terbaik. Hasil centroid terbaik ditunjukkan pada Gambar 5.

345.05	34.05	37.05	76.05
--------	-------	-------	-------

Gambar 5. Hasil Centroid terbaik

Proses PSO clustering tetap berlangsung sampai batas iterasi maksimum. Setelah ditemukan hasil centroid terbaik yang diperoleh dari global best, maka cluster terbaik dari keseluruhan data juga diperoleh. Salah satu kelemahan dalam PSO adalah seringnya terjebak dalam solusi optimum lokal sehingga terkadang bisa menyebabkan nilai cost yang dihasilkan mengalami konvergensi dini. Namun hal tersebut tidak begitu mempengaruhi hasil clustering, karena PSO disini hanya bertugas untuk mengoptimasi pusat cluster pada K-Means.

8. HASIL

Proses clustering dilakukan menggunakan dua algoritma untuk membuktikan algoritma mana yang lebih akurat dalam permasalahan clustering. Kedua algoritma yang digunakan tersebut adalah K-Means Clustering dan Hybrid PSO K-Means Clustering. Bagus atau tidaknya hasil clustering dilihat dari nilai *Silhouette Coefficient*. Semakin besar nilai *Silhouette Coefficient* maka semakin baik hasil clustering yang dihasilkan.

Clustering dilakukan untuk data nasabah bank yang berjumlah 400 *record* customer dengan dua atribut yaitu *business age* dan *days delinquent*. Data-data tersebut akan dibuat menjadi dua cluster, yaitu cluster “Ya” dan cluster “Tidak”. Cluster “Ya” menunjukkan nasabah tersebut diprediksikan akan mengalami kebangkrutan dan pihak bank tidak akan memberikan pinjaman lagi kepada nasabah tersebut, dan cluster “Tidak” menunjukkan nasabah tersebut diprediksikan tidak akan mengalami kebangkrutan dan pihak bank akan memberikan pinjaman lagi kepada nasabah tersebut. Percobaan dilakukan sebanyak 20 kali kemudian dihitung rata-rata dari nilai *Silhouette Coefficient* pada setiap uji coba, sehingga didapatkan hasil seperti ditunjukkan pada Tabel 11.

Tabel 11. Analisa Hasil

Algoritma	Average of Computation Time (s)	Average of Silhouette Coefficient
K-Means	12.75	0.516242
Hybrid PSO K-Means	13	0.535799

Dapat dilihat pada Tabel 10 hasil rata-rata *Silhouette Coefficient* menggunakan K-Means lebih kecil dari pada menggunakan Hybrid PSO K-Means. Waktu komputasi yang tidak jauh berbeda. Hal ini menunjukkan bahwa dengan proses hybridisasi K-Means dengan PSO menghasilkan hasil clustering yang lebih baik dari metode K-Means.

9. KESIMPULAN

Berdasarkan pengujian yang dilakukan Dapat dilihat pada Tabel 11 hasil rata-rata *Silhouette Coefficient* menggunakan K-Means lebih kecil dari pada menggunakan Hybrid PSO K-Means. Waktu komputasi dari kedua metode juga tidak jauh berbeda. Hal ini menunjukkan bahwa dengan proses hybridisasi K-Means dengan PSO menghasilkan hasil clustering yang lebih baik dari pada hanya menggunakan metode K-Means dengan waktu komputasi yang tidak terlalu lama jika dibandingkan dengan penelitian sebelumnya [8]. Penelitian selanjutnya dilaksanakan dengan menambahkan suatu mekanisme pada PSO untuk mencegah konvergensi dini sehingga didapatkan solusi yang lebih baik [16].

DAFTAR PUSTAKA

- [1] I. G. K. A. Ulupui, "Analisis pengaruh rasio likuiditas, leverage, aktivitas, dan profitabilitas terhadap return saham," *J. Ilm. Akunt. dan Bisnis*, pp. 1–20, 2007.
- [2] M. Suharli and M. Oktorina, "Memprediksi Tingkat Pengembalian Investasi pada Equity Securities Melalui Rasio profitabilitas, Likuiditas, dan utang pada Perusahaan Publik di Jakarta," *Semin. Nas. Akunt. VIII*, pp. 288–296, 2005.
- [3] J. Karimov and M. Ozbayoglu, "Clustering Quality Improvement of k-means Using a Hybrid Evolutionary Model," *Procedia Comput. Sci.*, vol. 61, pp. 38–45, 2015.
- [4] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis," *Appl. Soft Comput. J.*, vol. 10, no. 1, pp. 183–197, 2010.
- [5] R. J. Kuo, M. J. Wang, and T. W. Huang, "An application of particle swarm optimization algorithm to clustering analysis," *Soft Comput.*, vol. 15, no. 3, pp. 533–542, 2011.
- [6] W. F. Mahmudy, "Optimasi Part Type Selection and Machine Loading Problems Pada Fms Menggunakan Metode Particle Swarm Optimization," *Konf. Nas. Sist. Inf. (KNSI), STMIK Dipanegara, Makassar*, vol. 27 Februari, pp. 1718–1723, 2014.
- [7] G. Armarno and M. R. Farmani, "Multiobjective clustering analysis using particle swarm optimization," *Expert Syst. Appl.*, vol. 55, pp. 184–193, 2016.
- [8] H. Li, H. He, and Y. Wen, "Dynamic Particle Swarm Optimization and K-means Clustering Algorithm for Image Segmentation," *Opt. - Int. J. Light Electron Opt.*, vol. 126, no. 24, pp. 4817–4822, 2015.
- [9] M. Y. Cheng, K. Y. Huang, and H. M. Chen, "K-means particle swarm optimization with embedded chaotic search for solving multidimensional problems," *Appl. Math. Comput.*, vol. 219, no. 6, pp. 3091–3099, 2012.
- [10] C.-Y. Chiu, Y.-F. Chen, I.-T. Kuo, and H. C. Ku, "An Intelligent Market Segmentation System Using K-Means and Particle Swarm Optimization," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 4558–4565, 2009.
- [11] J. Han and K. Micheline, *Data Mining: Concepts and Techniques*, Second Edi., no. Second Edition. Morgan Kaufmann Publishers, 2006.
- [12] A. Karami and M. Guerrero-Zapata, "A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks," *Neurocomputing*, vol. 149, no. PC, pp. 1253–1269, 2015.
- [13] S. N. Gama, I. Cholissodin, and M. T. Furqon, "Clustering Portal Jurnal International untuk Rekomensari Publikasi Berdasarkan Kualitas Cluster Menggunakan Kernel K-Means," *Progr. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*.
- [14] M. Anggara, H. Sujiani, and H. Nasution, "Pemilihan Distance Measure Pada K-Means Clustering Untuk Pengelompokkan Member Di Alvaro Fitness," vol. 1, no. 1, pp. 1–6, 2016.
- [15] M. S. Couceiro, N. M. F. Ferreira, and J. a T. Machado, "Fractional Order Darwinian Particle Swarm Optimization," vol. 283, pp. 36–54, 2016.
- [16] W. F. Mahmudy, U. Brawijaya, O. Article, and K. Nasional, "Improved Particle Swarm Optimization untuk Menyelesaikan Permasalahan Part Type Selection dan Machine Loading pada Flexible Manufacturing System (FMS)," *Konf. Nas. Sist. Informasi, Univ. Klatat, Airmadidi, Minahasa Utara, Sulawesi Utara*, vol. 26–28 Febr, pp. 1003–1008, 2015.