

## HYBRID K-MEANS DAN PARTICLE SWARM OPTIMIZATION UNTUK CLUSTERING NASABAH KREDIT

Yusuf Priyo Anggodo<sup>1</sup>, Winda Cahyaningrum<sup>2</sup>, Aprilia Nur Fauziah<sup>3</sup>, Irma Lailatul Khoiriyah<sup>4</sup>,  
Oktavianis Kartikasari<sup>5</sup>, Imam Cholissodin<sup>6</sup>

<sup>123456</sup>Fakultas Ilmu Komputer, Universitas Brawijaya, Indonesia

Email: <sup>1</sup>anggodyusuf1950@gmail.com, <sup>2</sup>winda.cahyaningrum1@gmail.com, <sup>3</sup>Nfapril7@gmail.com,  
<sup>4</sup>irma.lailatul31@gmail.com, <sup>5</sup>oktakartikasari5@gmail.com, <sup>6</sup>imamcs@ub.ac.id

(Naskah masuk: 20 Februari 2017, diterima untuk diterbitkan: 7 Mei 2017)

### Abstrak

Kredit merupakan suatu pendapatan terbesar bagi bank. Akan tetapi, bank harus selektif dalam menentukan nasabah yang dapat menerima kredit. Permasalahan ini menjadi semakin kompleks karena ketika bank salah memberikan kredit kepada nasabah dapat merugikan, selian itu banyaknya parameter penentu dalam penentuan nasabah yang kredit. *Clustering* merupakan salah satu cara untuk dapat menyelesaikan permasalahan ini. K-Means merupakan metode yang simpel dan populer dalam menyelesaikan permasalahan *clustering*. Akan tetapi, K-Means murni tidak dapat memberikan solusi optimum sehingga perlu dilakukan *improve* untuk mendapatkan solusi optimum. Salah satu metode optimasi yang dapat menyelesaikan permasalahan optimasi dengan baik adalah *particle swarm optimization* (PSO). PSO sangat membantu dalam proses *clustering* dengan melakukan optimasi pada titik pusat tiap *cluster*. Untuk meningkatkan hasil yang lebih baik pada PSO ada beberapa *improve* yang dilakukan. Pertama penggunaan *time variant inertia* untuk membuat nilai *w* atau *inertia* dinamis tiap iterasinya. Kedua melakukan kontrol kecepatan partikel atau *velocity clamping* untuk mendapatkan posisi terbaik. Selian itu untuk mengatasi konvergensi dini dilakukan *hybrid* PSO dengan *random injection*. Hasil pengujian menunjukan *hybrid* PSO K-Means memberikan hasil terbesar dibandingkan K-Means dan PSO K-Means, dimana *silhouette coefficient* dari K-Means, PSO K-Means, dan *hybrid* PSO K-Means masing-masing 0.57343, 0.792045, 1.

**Kata kunci:** Kredit, Clustering, PSO, K-Means, Random Injection

### Abstract

Credit is a biggest revenue for the bank. However, banks have to be selective in deciding which clients can receive the credit. This issue is becoming increa singly complex because when the bank was wrong to give credit to customers can do harm, that the large number of selian deciding parameter in determing customer credit. Clustering is one way to be able to resolve this issue. K-Means is a simple and popular method in solving clustering. However, the K-Means pure can not provide optimum solutions so that needs to be done to get the optimum solution to improve. One method of optimization that can solve the problems of optimization with particle swarm optimization is good (PSO). PSO is very helpful in the process of clustering to perform optimization on the central point of each cluster. To improve better results on PSO there are some that do improve. The first use of time variant inertia to make the value of each dynamic inertial *w* or iterasinya. Both control the speed of the particle velocity or clamping to get the best position. Selian it to overcome premature convergence do hybrid PSO with random injection. The results of this research provide the optimum results for solving clustering of customer credits. The test results showed the hybrid PSO K-Means provide the greatest results than K-Means and PSO K-Means, where the silhouette's of the K-Means, PSO K-Means, and hybrid PSO K-Means respectively 0.57343, 0.792045, 1.

**Keywords:** Credit, Clustering, PSO, K-Means, Random Injection

## 1. PENDAHULUAN

Kredit menurut undang-undang No.10/1998 pasal 1 (11) tentang Perubahan Atas undang-undang No.7/1992 tentang Perbankan merupakan kesepakatan antara pihak penyedia jasa keuangan dengan pihak lain atas pernyediaan uang atau tagihan yang akan dikembalikan sesuai dengan kesepakatan waktu dan bunga yang telah disepakati bersama (YLBHI, 2007). Kredit timbul dari adanya

kepercayaan antara pemberi kredit dengan penerima kredit bahwa penerima kredit dapat memenuhi perjanjian atau kesepakatan bersama. Penjaminan keamanan transaksi dilakukan dengan melakukan pemilihan nasabah yang akan melakukan kredit secara selektif. Setiap bank memiliki parameter untuk menentukan nasabah yang dapat memperoleh kredit.

Penentuan pemberian kredit pada nasabah menjadi satu masalah yang cukup kompleks karena disatu sisi kredit dapat memberi masukan bank yang

cukup besar sedangkan disisi lain dapat merugikan jika nasabah tidak dapat membayar tagihan kredit. Resiko pemberian kredit pada nasabah yang kurang tepat menyebabkan bank tidak memperoleh pendapatan bunga, selain itu jumlah tagihan harus dikompensasi dengan jumlah modal sehingga modal yang ada akan berkurang dan mungkin habis (Suharli dan Oktorina, 2005). Hal tersebut dapat melanggar ketentuan *Capital Adequacy Ratio*. Jika Bank memiliki modal yang cukup, maka tagihan tersebut akan dicadangkan sesuai dengan peraturan Bank Indonesia No. 7/2/PBI/2005. Hal tersebut berpengaruh terhadap dana yang dapat ditempatkan dan menghasilkan bagi bank (Suhardi, 2006).

Ada berbagai pendekatan untuk menyelesaikan permasalahan tersebut, salah satunya adalah *clustering*. Ketepatan dalam membentuk model akan memberikan hasil yang maksimal (Anggodo dan Mahmudy, 2016). K-Means merupakan metode *clustering* yang populer dan bersifat tanpa kendali (Karimov dan Ozbayglu, 2015). Pembentukan model dengan K-Means dari data yang besar dapat memaksimalkan hasil pengelompokan (MacQueen, 1967). Tujuan utama K-Means adalah melakukan *clustering* objek ke *cluster* atau kelompok. Pertama adalah menentukan jumlah *k* yang mempresentasikan jumlah *cluster* yang akan dibentuk, lalu menentukan titik pusat atau *centroid* untuk setiap *cluster* dan dilakukan pengelompokan. Akan tetapi, karena penentuan titik pusat dilakukan secara acak pada tahapan pertama dan mengambil nilai rata-rata untuk langkah selanjutnya memberikan hasil yang kurang maksimal dan diperoleh konvergensi optimum lokal (Niknma dan Amiri, 2010). Permasalahan tersebut dapat diatasi dengan melakukan optimasi nilai titik pusat sehingga didapatkan titik pusat yang paling maksimal untuk melakukan perhitungan K-Means.

Ada berbagai metode meta-heuristik yang dapat digunakan untuk mengatasi permasalahan optimasi (Anggodo *et al*, 2017). PSO merupakan salah satu metode yang dapat melakukan optimasi nilai titik pusat K-Means dengan hasil yang baik (Kuo *et al*, 2011). Penggunaan PSO dapat diterapkan untuk melakukan optimasi nilai titik pusat dengan mempresentasikan solusi permasalahan dalam bilangan real (Alam *et al*, 2015). PSO terinspirasi dari kecerdasan individu dan social dari burung (Kennedy dan Eberhart, 1995). Pada penelitian Liu *et al* (2014) penggunaan PSO dapat melakukan *clustering* segmentasi citra digital. Sedangkan pada permasalahan kompleks penggunaan PSO juga sangat membantu dalam melakukan *clustering* (Li *et al*, 2017; Armano dan Farmani, 2016; Anggodo dan Mahmudy, 2017). Berdasarkan penelitian sebelumnya ditunjukkan penggunaan PSO sangat membantu untuk melakukan optimasi nilai *centroid* sehingga didapatkan tingkat kesalahan *clustering* yang rendah. Penelitian *clustering* nasabah kredit sebelumnya pernah diajukan oleh Wahyuni *et al*

(2016) dengan menggunakan *hybrid* K-Means dan PSO. Akan tetapi, pada penelitian sebelumnya hanya digunakan 2 parameter pada penelitian ini digunakan 13 parameter untuk menentukan seorang nasabah layak menerima kredit atau tidak. Selain itu juga dilakukan penanganan konvergensi dini untuk mendapat solusi terbaik menggunakan *random injection* (Mahmudy, 2015).

Fokus pertama penelitian ini adalah melakukan *clustering* nasabah kredit berdasarkan 13 parameter. Kedua untuk mendapatkan hasil *clustering* yang maksimal dan mencegah konvergensi optimum lokal dilakukan optimasi titik pusat menggunakan PSO. Ketiga penggabungan *random injection* dan PSO untuk penanganan konvergensi dini dan mendapatkan solusi terbaik.

## 2. ANALISIS NASABAH KREDIT

Kredit merupakan suatu pembiayaan yang diberikan bank kepada nasabah untuk jangka pendek (Sheny dan Williams, 2017). Pinjaman yang diberikan untuk keperluan sehari-hari rumah tangga, keperluan membeli rumah, motor, dan lain sebagainya yang bersifat konsumtif. Walaupun kredit bukanlah bantuan likuiditas atau kewajiban perusahaan memenuhi kewajiban jangka pendek akan tetapi kredit juga digunakan untuk keperluan produkti. Berdasarkan hal tersebut kredit menjadi sumber pendapatan paling besar bagi bank, disatu sisi juga memiliki resiko yang besar (Suhardi, 2006). Selain itu pada penelitian Albuquerque *et al* (2015) menunjukkan hasil prediksi nasabah kredit dari tahun ketahun semakin meningkat sehingga bank harus selektif dalam mengambil keputusan untuk mendapatkan pendapatan yang besar dan resiko yang kecil. Pada penelitian ini akan digunakan data dari UCI Dataset. Pada penelitian ini akan digunakan 4121 data dengan 13 parameter lain pekerjaan, status, pendidikan, pernah kredit, kredit rumah, pinjaman bank yang merupakan parameter tergolong kategori. Sedangkan pemasaran sebelumnya, umur, rata-rata gaji dalam tiga bulan, indek konsumsi nasabah dalam satu bulan, indek kepercayaan nasabah dalam satu bulan, rata-rata *euribor* 3 bulan untuk harian, dan nomer perkerja dalam 4 bulan merupakan parameter tergolong numerik. Tabel 1 menunjukkan data nasabah kredit.

**Table 1.** Data nasabah kredit

No	g1	g2	g3	g4	g5	g6	g7	g8	...	g13
1	30	2	2	3	1	2	1	2	...	5099.1
2	39	8	3	4	1	1	1	2	...	5191
-	-	-	-	-	-	-	-	-	...	...
-	-	-	-	-	-	-	-	-	...	...
2120	18	0	0	0	0	0	0	1	...	4963.6
2121	88	11	3	7	2	2	2	3	...	5228.1

Dimana parameter nasabah kredit menunjukkan g1, g2, g3, g4, g5, g6, g7, g8, g9, g10, g11, g12, g13 secara berurut-urut adalah umur, pekerjaan, status,

pendidikan, pernah kredit, kredit rumah, pinjaman bank, pemasaran sebelumnya, rata-rata gaji dalam tiga bulan, indeks konsumsi nasabah dalam satu bulan, indeks kepercayaan nasabah dalam satu bulan, rata-rata *euribor* 3 bulan untuk harian, dan nomor pekerja dalam 4 bulan.

### 3. K-MEANS

Prosedur untuk dalam melakukan *clustering* K-Means sangatlah simpel dan mendapatkan hasil yang terbaik (MacQueen, 1967). Pertama melakukan insialisasi nilai  $k$  yang merupakan jumlah *cluster*, lalu menentukan titik pusat untuk tiap *cluster* secara acak. Setiap objek akan diatur untuk dimasukkan ke *cluster* tertentu, dimana objek-objek dalam satu *cluster* memiliki kemiripan. Penentuan kemiripan objek dengan titik pusat *cluster* diukur dengan *Euclidean Distance* (D) yang ditunjukkan pada persamaan 1. Sebuah objek akan masuk ke *cluster* yang memiliki nilai D terkecil.

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

Dimana,

$p$  = Dimensi data

$x_1$  = Posisi titik 1

$x_2$  = Posisi titik 2

Tahapan algoritma *clustering* K-Means sebagai berikut (Karimi dan Guerro-Zapata, 2015):

1. Insialisasi nilai  $k$  *cluster* dan *centroid* tiap *cluster* secara acak.
2. Menentukan setiap objek masuk *cluster* dengan jarak terdekat berdasarkan nilai *Euclidean Distance*.
3. Melakukan perhitungan ulang nilai *centroid* tiap *cluster* dengan persamaan 2.

$$V_j = \frac{1}{n_j} \sum_{p=0}^n \text{data}_p, \quad (2)$$

dimana  $V_j$  merupakan nilai *centroid* dari *cluster*  $j$ .  $n_j$  merupakan jumlah objek dalam *cluster*  $j$ .  $\text{data}_p$  menunjukkan vektor data ke  $p$ .

4. Ulangi langkah ke 2 sampai nilai *centroid* tidak berubah atau sudah melampaui iterasi tertentu.

### 4. PARTICLE SWARM OPTIMIZATION

PSO merupakan metode optimasi bersifat stotastik yang merepresentasikan solusi permasalahan dalam bentuk partikel (Kennedy dan Eberhart, 1995). Partikel dibangkitkan secara acak sebanyak  $n$  partikel, tiap partikel terdiri dari beberapa dimensi posisi  $x_i$  dan kecepatan  $v_i$ . Tiap partikel akan diukur seberapa besar nilai fitness yang diberikan dengan persamaan 3.

$$F = f(x) \quad (3)$$

Dimana  $f(x)$  adalah nilai dari *silhouette coefficient*, yaitu suatu metode untuk mengukur objek yang

terbentuk dalam *cluster*. Semakin besar nilai *silhouette coefficient* maka semakin bagus data yang dikelompokkan. Setiap partikel akan “terbang” menjelajahi ruang pencarian solusi. Perpindahan dari satu posisi ke posisi lainnya sangat dipengaruhi oleh kecepatan tiap partikel, untuk mendapatkan posisi terbaik diperlukan formulasi kecepatan yang dinamis menggunakan persamaan 4 (Eberhart dan Shi, 2000).

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i - x_i) + c_2 \cdot r_2 \cdot (p_{gi} - x_i), \quad (4)$$

dimana  $v_i$  menunjukkan nilai kecepatan untuk dimensi partikel ke  $i$  sampai ke  $n$ ,  $t$  menunjukkan waktu iterasi,  $w$  merupakan nilai vektor inerti yang nilainya didapatkan secara dinamis menggunakan persamaan 5 (Ratnaweera *et al*, 2004).  $p_i$  adalah posisi terbaik yang pernah didapatkan untuk tiap partikel, sedangkan  $p_{gi}$  adalah posisi terbaik yang pernah dicapai oleh keseluruhan partikel.  $c_1$  dan  $c_2$  berurut adalah konstanta *cognitive* dan *social*, yang mana pada penelitian ini bernilai 1.  $r_1$  dan  $r_2$  adalah bilangan yang dibangkitkan secara acak antara  $[0,1]$ . Setelah didapatkan kecepatan maka akan dilakukan *update* posisi menggunakan persamaan 6.

$$w = (w_{max} - w_{min}) \frac{(Iterasi-t)}{Iterasi} + w_{min}, \quad (5)$$

$$x_i^{t+1} = x_i + v_i^{t+1}, \quad (6)$$

Pada perpindahan posisi partikel PSO biasanya terlalu cepat sehingga tidak dapat menemukan solusi optimum. Hal tersebut dapat diatasi dengan melakukan kontrol kecepatan atau *velocity clamping* (Marini dan Walzcak, 2015). Mekanisme kontrol kecepatan dengan melakukan kondisi untuk kecepatan tiap partikel menggunakan persamaan 7.

$$\begin{aligned} &\text{if } (v_{ij}^{t+1} > v_j^{\max}) \text{ maka } v_{ij}^{t+1} = v_j^{\max} \\ &\text{if } (v_{ij}^{t+1} < v_j^{\min}) \text{ maka } v_{ij}^{t+1} = v_j^{\min}, \end{aligned} \quad (7)$$

sedangkan nilai  $v_j^{\max}$  dibangkitkan menggunakan persamaan 8 dan  $v_j^{\min}$  merupakan nilai negatif dari  $v_j^{\max}$ .

$$v_j^{\max} = k \frac{(x_{j,\max} - x_{j,\min})}{2}, \quad k \in (0, 1], \quad (8)$$

Siklus perhitungan nilai kecepatan  $v_i$  dan *update* posisi  $x_i$  akan terus diulangi sampai iterasi selesai. Ketika iterasi sudah selesai, partikel terbaik keluar sebagai solusi optimum.

### 5. RANDOM INJECTION

*Random injection* merupakan suatu metode yang digunakan untuk penanganan konvergensi dini. *Random injection* pertama diajukan untuk menangani permasalahan konvergensi dini pada algoritma genetika (Mahmudy *et al*, 2013).

Permasalahan yang sering terjadi pada PSO adalah ketika iterasi tertentu, nilai solusi yang diberikan tiap partikel sama sebelum tercapai solusi optimum. Pada penelitian Wahyuni *et al* (2016)

penggunaan *hybrid* PSO masih belum memberikan hasil yang solusi optimum karena belum dilakukan penanganan konvergensi dini. Berbagai macam penanganan konvergensi dini salah satunya *random injection*, pada penelitian Mahmudy (2015) penggunaan *random injection* dapat meningkatkan hasil akurasi dari pada penggunaan PSO saja. Mekanisme *random injection* sangat sederhana, dengan memasukan  $p$  partikel acak pada  $q$  iterasi. Penentuan nilai  $p$  partikel dan  $q$  iterasi yang terbaik harus dilakukan pengujian terlebih dahulu.

**6. SILHOUETTE COEFFICIENT**

*Silhouette coefficient* merupakan suatu metode yang digunakan untuk mengukur seberapa baik objek diletakan dalam *cluster* (Wahyuni et al, 2016). Metode ini merupakan penggabungan dari metode *cohesion* dan *separation* (Anggara et al, 2016) yang digunakan untuk melakukan perhitungan nilai *fitness* pada tiap partikel dalam PSO. Tahapan perhitungan *silhouette coefficient* ditunjukkan pada persamaan 9 sampai 12:

1. hitung jarak rata-rata  $i$  titik *centroid* ke semua objek dalam satu *cluster*.

$$a(i) = \frac{1}{|A|} \sum_{j \in A, j \neq i} C d(i,j), \tag{9}$$

Dimana  $j$  merupakan objek dalam *cluster*  $A$  dan  $d(i, j)$  merupakan jarak objek  $i$  ke  $j$ .

2. hitung jarak rata-rata  $i$  titik *centroid* ke semua objek *cluster* lain, lalu ambil nilai terkecil.

$$d(i,C) = \frac{1}{|A|} \sum_{j \in C} d(i,j), \tag{10}$$

3. Dimana  $d(i,C)$  merupakan jarak objek  $i$  ke objek *cluster* lain.

$$b(i) = \min_{C \neq A} d(i,C), \tag{11}$$

4. hitung nilai *silhouette coefficient* untuk tiap *centroid*.

$$s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))}, \tag{12}$$

Nilai rata-rata *silhouette coefficient* atau  $s(i)$  menunjukkan seberapa tepat data dikelompokan,

**7. HYBRID PSO K-MEANS**

*Siklus Hybrid PSO K-Means Clustering*

Langkah 1: inisialisasi

- Inisialisasi parameter PSO seperti  $n, w_{max}, w_{min}, C1, C2$ .
- inisialisasi partikel yang merepresentasikan nilai *centroid* tiap *cluster* dan kecepatan.

Langkah 2: *update*

- *update* posisi tiap partikel dan hitung nilai *fitness*nya menggunakan K-Means
- Update posisi terbaik untuk tiap partikel.
- Update posisi terbaik keseluruhan partikel.

Langkah 3: *random injection*

- Memasukan  $p$  partikel yang dibangkitkan

secara acak untuk tiap  $q$  iterasi.

Langkah 4: kondisi berhenti

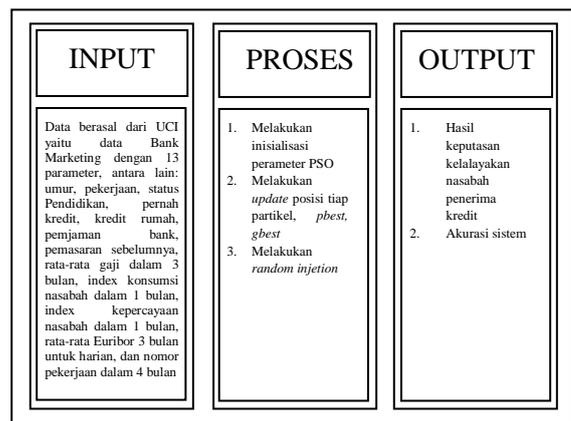
- Jika iterasi belum selesai lakukan langkah 2 dan sudah berhenti.

**Gambar 1.** Siklus metode *hybrid* PSO K-Means

Gambar 1 menunjukkan siklus metode *hybrid* PSO K-Means dalam melakukan *clustering* data nasabah kredit. Salah satu hal yang cukup berpengaruh terhadap hasil *clustering* adalah representasi partikel PSO dalam melakukan optimasi nilai *centroid*. Penelitian ini representasi partikel dalam bentuk kategori numerik dan interger sesuai dengan batasan masing-masing parameter. Terdapat dua baris dan 13 kolom, dimana baris merupakan nilai *centroid* tiap *cluster* dan kolom merupakan dimensi partikel yang merepresentasikan nilai dari parameter yang ditunjukkan pada Tabel 2 dan Gambar 2 menunjukkan alur kerja sistem.

**Table 2.** Representasi partikel

g1	g2	g3	g4	g5	g6	g7	g8	g9	...	g13
35	7	0	1	1	0	1	2	0.9	...	4335.7
47	8	0	4	0	1	1	2	-1.2	...	3049.9



**Gambar 2.** Alur kerja sistem

**8. METODOLOGI**

Proses pengelompokan nasabah kredit berdasarkan 13 parameter pada penelitian ini, akan dimaksimalkan menggunakan metode *hybrid* PSO K-Means. Langkah pertama K-Means adalah membangkitkan titik pusat secara acak. Pada penelitian ini terdapat 2, yaitu “diterima” untuk 1 dan “ditolak” untuk 2. Tabel 3 menunjukkan titik pusat dari 2 klas.

**Table 3.** Representasi partikel

No	g1	g2	g3	g4	g5	g6	g7	g8	g9	...	g13
1	35	7	0	1	1	0	1	2	0.9	...	4335.7
2	47	8	0	4	0	1	1	2	-1.2	...	3049.9

Langkah selanjutnya adalah menghitung kedekatan setiap dengan *centroid* 1 dan 2. Semisal terdapat

data (30, 5, 1, 2, 1, 1, 1, 0, 0.5, 40, -15, 1.2, 3555), sehingga diperoleh jarak dengan kedua *centroid* menggunakan persamaan 1.

$$D(\text{data}, c_1) = \sqrt{\begin{matrix} (35-30)^2 + (7-5)^2 + (0-1)^2 \\ + (1-2)^2 + (1-1)^2 + (0-1)^2 \\ + (1-1)^2 + (2-0)^2 + (0.9-0.5)^2 \\ + (16.428-40)^2 + (-27.2+15)^2 \\ + (1.367-1.2)^2 + (4335.7-3555)^2 \end{matrix}}$$

$D(\text{data}, c_1) = 780.1748$

Dengan cara yang sama sehingga diperoleh nilai  $D(\text{data}, c_2) = 507.4336$ . Karena jarak ke *centroid* 2 lebih pendek maka data tersebut dimasukkan ke dalam *cluster* 2. Langkah selanjutnya adalah melakukan penentuan *cluster* untuk data lainnya, lalu dilakukan perhitungan titik pusat menggunakan persamaan 2 sampai nilai titik pusat tidak berubah atau iterasi berakhir.

Pada PSO proses pertama adalah pembangkitan nilai partikel secara acak sesuai dengan batasan masing dimensi dalam partikel. Tabel 4 menunjukkan hasil pembangkitan nilai partikel. Selain itu inisialisasi nilai kecepatan tiap dimensi pada partikel dengan nilai 0.

**Table 4.** Inisialisasi partikel awal

No.	g1	g2	g3	g4	g5	g6	g7	g8	g9	...	g13
1	35	7	0	1	1	0	1	2	0.9	...	4335.7
	47	8	0	4	0	1	1	2	-1.2	...	3049.9
2	25	5	1	3	0	0	1	1	0.5	...	2000.7
	45	3	0	2	1	1	1	0	1	...	5000.9
-	-	-	-	-	-	-	-	-	-	-	-
20	34	5	0	2	1	0	1	1	-0.5	...	5000.7
	50	8	0	1	0	0	0	1	0.4	...	3000.9

Beberapa parameter yang dilakukan inialisasi diawal seperti jumlah partikel, jumlah iterasi, *cognitif*,  $w_{max}$ , dan  $w_{min}$ . Langkah selanjutnya adalah menghitung nilai *fitness* dari setiap partikel, dimana nilai *fitness* diperoleh dari *silhouette coefficient* pada persamaan 9-12. Tabel 5 menunjukkan hasil perhitungan *fitness* untuk setiap partikel.

**Table 5.** Hasil perhitungan nilai *fitness*

Partikel	<i>Fitness</i>
1	0.2345
2	0.3214
...	...
20	0.1414

Setelah didapatkan nilai *fitness* untuk tiap partikel maka dilakukan upload *pbest* dan *gbest* yaitu nilai *fitness* terbaik pada tiap partikel dan keseluruhan partikel. Selanjutnya adalah tahapan melakukan perubahan kecepatan untuk tiap dimensi pada partikel menggunakan persamaan 4. Akan tetapi, bangkitkan terlebih dahulu nilai  $w$  menggunakan persamaan 5 dengan nilai iterasi 4,  $c_1 = 1$ ,  $c_2 = 1$ ,  $r_1 = 1$ , dan  $r_2 = 1$ .

$$w = (0.9 - 0.4) \frac{(10-1)}{10} + 0.4 = 0.96$$

$$v_{2,1}^2 = 0.96 \cdot 0 + 1.1 \cdot (25 - 30) + 1.1 \cdot (25 - 30) = -10$$

Pada partikel 2 dimensi 1 diperoleh nilai kecepatan -10, dengan cara yang sama maka diperoleh kecepatan pada dimensi tiap partikel yang ditunjukkan pada Tabel 6.

**Table 6.** Kecepatan baru

No.	g1	g2	g3	g4	g5	g6	g7	g8	g9	...	g13
1	-10	2	1	0	0	0	1	1	0.3	...	435
	12	3	1	2	0	1	1	2	0.5	...	304
2	14	4	0	1	1	0	1	0	0.7	...	207
	20	4	0	1	0	1	1	0	0.8	...	500
-	-	-	-	-	-	-	-	-	-	-	-
20	10	2	0	1	1	0	1	1	0.2	...	500
	23	3	0	1	1	0	0	1	0.4	...	300

Setelah didapatkan kecepatan maka langsung adalah melakukan perubahan posisi menggunakan persamaan 6.

$$x_{2,1}^2 = 25 + 14 = 39.$$

Dengan cara yang sama maka diperoleh nilai posisi baru yang ditunjukkan pada Tabel 7.

**Table 7.** Posisi baru

No.	g1	g2	g3	g4	g5	g6	g7	g8	g9	...	g13
1	25	9	1	1	1	0	2	3	1.2	...	477
											0.7
	59	11	1	6	0	2	2	4	-0.7	...	335
											3.9
2	39	9	1	4	1	0	2	1	1.2	...	220
											7.7
	65	7	0	3	1	2	2	0	1.8	...	550
											0.9
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
20	44	7	0	3	2	0	2	2	-0.3	...	550
											0.7
	73	11	0	2	1	0	0	2	0.8	...	330
											0.9

Proses PSO akan terus dijalankan sampai iterasi berakhir, *gbest* keluar sebagai solusi yang diberikan.

**9. HASIL**

Proses pengujian digunakan untuk mengetahui metode yang menghasilkan solusi terbaik. K-Means dan PSO merupakan metode statistik atau bersifat *random* sehingga akan dilakukan simulasi sebanyak 5 kali dan hasil rata-rata akurasi dan waktu komputasi akan dibandingkan.

*Clustering* nasabah bank akan diujikan sebanyak 4121 *record* nasabah. Data tersebut akan dikelompokkan ke *cluster* 1 (“diterima”) atau *cluster* 2 (“ditolak”). Parameter terbaik PSO yang akan digunakan adalah iterasi = 10, populasi = 10,  $v_{min} = -0.6$ ,  $v_{max} = 0.6$ ,  $w_{max} = 0.9$ ,  $w_{min} = 0.4$ . Sedangkan parameter *random injection* terbaik adalah  $p = 0.3$ ,  $q = 2$ . Hasil pengujian yang telah didapatkan ditunjukkan pada Tabel 8.

**Table 8.** Hasil perbandingan

Algoritma	Rata-rata Akurasi	Rata-rata Waktu komputasi (s)
K-Means	0.57343	0.58
PSO K-Means	0.792045	1.55
Hybrid PSO K-Means	1.00	1.67

Berdasarkan hasil pengujian parameter menunjukkan bahwa penggunaan *hybrid* PSO K-Means memberikan hasil akurasi yang maksimal sehingga metode pada penelitian ini dapat menentukan *cluster* nasabah kredit dengan sangat baik.

## 10. KESIMPULAN

*Clustering* permasalahan nasabah kredit telah diselesaikan secara baik menggunakan *hybrid particle swarm optimization* K-Means. Serangkaian pengujian menunjukkan bahwa penggunaan *hybrid* PSO K-Means dengan *random injection* memberikan hasil yang mendekati solusi optimum dan waktu komputasi yang relatif cepat, yaitu kurang dari 2 menit untuk jumlah data yang besar. *Hybrid* PSO K-Means memberikan hasil solusi yang lebih baik dibandingkan dengan PSO K-Means dan K-Means. Penggunaan *hybrid* PSO K-Means dapat melakukan *clustering* data dengan benar sebesar 80%.

Pada penelitian ini pada iterasi ke 10 sudah mengalami konvergensi secara global untuk keseluruhan partikel, hal ini karena sebagian besar parameter merupakan tipe data kategori yang jumlah kategorinya tidak lebih dari 5. Untuk mendapatkan hasil yang lebih baik lagi perlu dilakukan seleksi parameter. Salah satu cara seleksi parameter dengan merepresentasikan seleksi parameter kedalam segmen baru dalam partikel PSO (Novitasari *et al*, 2016).

## DAFTAR PUSTAKA

- ANGGODO, Y. P. & MAHMUDY, W. F., 2016. Peramalan butuhan hidup minimum menggunakan automatic clustering dan fuzzy logical relationship. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 3, no. 2, pp. 94-102.
- ANGGODO, Y. P. & MAHMUDY, W. F., 2017. Automatic clustering and optimized fuzzy logical relationships for minimum living needs forecasting. *Journal of Environmental Engineering and Sustainable Technology (JEEST)*, vol. 4, no. 1, pp. 1-7.
- ANGGODO, Y. P., ARIYANI, A. K., ARDI, M. K., & MAHMUDY, W. F. 2017. Optimization of Multi-Trip Vehicle Routing Problem with Time Windows using Genetic

Algorithm. *Journal of Environmental Engineering and Sustainable Technology* vol. 3, no.2, pp. 92-97.

- ANGGARA, M., SUJIANI, H. & NASUTION, H., 2016. Pemilihan Distance Measure Pada K-Means Clustering Untuk Pengelompokan Member Di Alvaro Fitness, *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 1, no. 1, pp. 1-6.
- ALAM, S., DOBBIE, G. & REHMAN, S. S., 2015. Analysis of particle swarm optimization based hierarchical data clustering approaches, *Swarm and Evolutionary Computation*, vol. 25, pp. 36-51.
- ARMANO, G. & FARMANI, M. R., 2016. Multiobjective clustering analysis using particle swarm optimization. *Expert Systems With Applications*, vol. 55, pp. 184-193
- EBERHART, R.C. & SHI, Y., 2000. Comparing inertia weight and constriction factors in particle swarm optimization. *IEEE Congress on Evolutionary Computation*, 16-19 Juni, San Diego, CA, US.
- KARAMI, A. & GUERRERO-ZAPATA, M., 2015. A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks. *Neurocomputing*, vol. 149, no. PC, pp. 1253-1269,
- KARIMOV, J. & OZBAYOGLU, M., 2015. *Clustering Quality Improvement of k-means Using a Hybrid Evolutionary Model*, *Procedia Comput. Sci.*, vol. 61, pp. 38-45.
- KENNEDY, J. & EBERHART, R., 1995. Particle swarm optimization. *IEEE International Conference on Neural Networks*, 27 Nov. – 1 Dec., Perth, Western, Australia, vol. 4, pp. 1942-1948.
- KUO, R. J., WANG, M. J. & HUANG, T. W., 2011. An application of particle swarm optimization algorithm to clustering analysis. *Soft Comput.*, vol. 15, no. 3, pp. 533-542.
- LIU, L., JIAO, L., ZHAO, J., SHANG, R. & GONG, M., 2017. Quantum-behaved discrete multi-objective particle swarm optimization for complex network clustering. *Pattern Recognition*, vol. 63, pp. 1-14.
- LIU, R., CHEN, Y., JIAO, LICHENG. & LI, Y., 2014. A particle swarm optimization based simultaneous learning framework for clustering and clustering. *Pattern Recognition*, vol. 47, pp. 2143-2152.
- MACQUEEN, J., 1967. Some methods for clustering and analysis of

multiobservations. *Proceedings of the fifth symposium on mathematical statistic and probability*, 21 Juni – 18 Juli, California, USA, vol. 1, pp. 281-297.

- MAHMUDY, W. F., 2015. Improved particle swarm optimization untuk menyelesaikan permasalahan part type selection dan mechine loading pada flexible manufacturing system (FMS). *Konf. Nas. Sist. Inf. (KNSI)*, Universitas Klabat, Airmadidi, Minahasa Utara, Sulawesi Utara, 26-28 Februari, pp. 1003–1008.
- MAHMUDY, W. F., MARIAN, R. M. & LUONG, L. H. S. 2013. Optimization of part type selection and loading problem with alternative production plans in flexible manufacturing system using hybrid genetic algorithms – Part 1: modelling and representation. *5th International Conference on Knowledge and Smart Technology (KST)*, Chonburi, Thailand, pp. 75-80.
- MARINI, F. DAN WALCZAK, B., 2015., Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153-165.
- NIKNAM, T. & AMIRI, B., 2010. An efficient hybrid approach based on PSO, ACO, and K-Means for cluster analysis. *Applied soft computing*, vol. 10, pp. 183-197.
- NOVITASARI, D., CHOLISSODIN, I. & MAHMUDY, W. F., 2016. Hybridizing PSO with SA for optimizing SVR applied to software effort estimation. *TELKOMNIKA*, vol. 14, no. 1, pp. 245-253.
- SHENOY, J. & WILLIAMS, R., 2017. Trade credit and the joint effect of supplier and customer financial characteristics. *J. Finan. Intermediation*, vol. 29, pp. 68-80.
- SUHARLI, M. & OKTORINA, M., 2005. Memprediksi tingkat pengembalian investasi pada equity securities melalui rasio profitabilitas, likuiditas, dan utang pada perusahaan publik di Jakarta. *Semen. Nas. Akunt. VII*, pp. 288-296.
- RATNAWEERA, A., HALGAMUGE, S. K. DAN WATSON, H. C., 2004. Self-organizing hierarchical particle swarm optimization with time-varying acceleration coefficients, *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240-255.
- SUHARDI, G., 2006. Resiko dalam pemberian kredit perbankan. *Jurnal Hukum Projustitia*, vol. 24, no. 1, pp. 96-111.
- WAHYUNI, I., AULIYA, Y. A., RAHMI, A. & MAHMUDY, W. F., 2016. Clustering nasabah bank berdasarkan tingkat likuiditas menggunakan hybrid particle swarm optimization. *Jurnal Ilmiah Teknologi dan Informasi ASIA (JITIKA)*, vol. 10, no. 2, pp. 24-33.
- YLBHI, 2007. *Panduan Bantuan Hukum di Indonesia: Pedoman Anda Memahami dan Menyelesaikan Masalah Hukum*. Jakarta: Yayasan Obor Indonesia. <https://books.google.co.id/books?id=Y1oghffVI2cC>.