

Optimasi Centroid Awal Algoritma K-Medoids Menggunakan Particle Swarm Optimization Untuk Segmentasi Customer

Initial Centroid Optimization K-Medoids Algorithm Using Particle Swarm Optimization for Customer Segmentation

Danang Bagus Wijaya¹, Edi Noersasongko², Purwanto³

^{1,2,3}Magister Teknik Informatika, Universitas Dian Nuswantoro

Jl. Nakula 1 No. 5-11, Semarang, 50131, Telp: (024) 3517261, Fax : (024) 325 0165

E-mail: ¹p31201902201@mhs.dinus.ac.id, ²edi.noersasongko@dsn.dinus.ac.id,

³purwanto@dsn.dinus.ac.id

Abstrak

Segmentasi pelanggan menjadi strategi yang penting dalam sebuah perusahaan, hal tersebut berpengaruh pada hubungan pelanggan yang baik akan menghasilkan sebuah peningkatan keuntungan. Pengelompokan pelanggan dalam data mining dapat menggunakan beberapa algoritma, tetapi K-Medoids menjadi pilihan yang tepat karena dapat mengurangi noise dan kesensitifan outlier. Tetapi, dalam pemilihan pusat cluster masih secara acak dan berpengaruh dalam hasil clustering sehingga perlu peningkatan pada algoritma k-medoids agar nilai cluster yang dihasilkan dapat optimal. Particle Swarm Optimization menjadi algoritma optimasi yang sering dipakai dan terbukti dapat meningkatkan hasil dari sebuah clustering. Dalam hal ini, optimasi menggunakan Particle Swarm Optimization (PSO) dalam pemilihan pusat cluster awal perlu diterapkan dalam algoritma k-medoids agar hasil dari cluster dapat optimal. Hasil dari pengujian menggunakan K-Medoids menunjukkan nilai Davies-Bouldin Index (DBI) terbaik dengan 3 cluster yaitu 0,283, sedangkan nilai DBI PSO+K-Medoids terbaik terdapat dalam 2 cluster yaitu 0,088. Dari nilai DBI menunjukkan bahwa optimasi PSO pada K-Medoids untuk penentuan centroid awal terbukti meningkatkan hasil dari clustering daripada K-Medoids standar.

Kata kunci: Segmentasi pelanggan, k-medoids, Particle Swarm Optimization, Davies-Bouldin Index

Abstract

Customer segmentation is an important strategy in a company, it affects good customer relationships which will result in increased profits. Grouping customers in data mining can use several algorithms, but K-Medoids is the right choice because it can reduce noise and outlier sensitivity. However, the selection of cluster centers is still random and has an effect on the results of clustering, so it is necessary to improve the k-medoids algorithm so that the resulting cluster value can be optimal. Particle Swarm Optimization is an optimization algorithm that is often used and has been proven to improve the results of a clustering. In this case, optimization using Particle Swarm Optimization (PSO) in the selection of the initial cluster center needs to be applied to the k-medoids algorithm so that the results of the cluster can be optimal. The test results using K-Medoids show the best Davies-Bouldin Index (DBI) value with 3 clusters, namely 0.283, while the best DBI PSO+K-Medoids value is in 2 clusters, namely 0.088. The DBI value shows that PSO optimization on K-Medoids to determine the initial centroid is proven to improve the results of clustering than standard K-Medoids.

Keywords: Customer segmentation, k-medoids, Particle Swarm Optimization, Davies-Bouldin Index

1. PENDAHULUAN

Perusahaan memiliki strategi pemasaran dalam persaingan bisnis, karena hal tersebut memiliki peran penting. Strategi pemasaran berorientasi produk merupakan hal utama, tetapi perusahaan harus memiliki kemampuan dalam strategi pemasaran dengan orientasi pelanggan [1]. Penyusunan strategi pemasaran dengan orientasi pelanggan bisa menggunakan strategi pengelompokan pelanggan sehingga dapat digunakan dalam mengelompokkan pelanggan berdasarkan karakteristik, kebutuhan dan perilaku yang heterogen atau berbeda-beda dalam suatu pasar tertentu untuk menjadikan satuan pasar yang homogen dengan strategi marketing mix merupakan penjelasan dari segmentasi pasar [2]. Segmentasi menjadi kunci pemasaran yang baik. Jika tidak dilakukan segmentasi, dapat membuang waktu, uang, dan penjualan. Pemborosan ini mungkin berlaku baik di sisi pemasar maupun pelanggan. pelanggan yang membeli produk sering kali memiliki karakteristik yang sama [3].

Secara umum, algoritma clustering dibagi menjadi 5 yaitu Partitioning-Based, Hierarchical-Based, Density-Based, Grid-Based, dan Model-Based. Partitioning-Based lebih cocok digunakan untuk segmentasi customer karena dataset yang digunakan lebih cocok dipartisi. Partitioing-Based terdiri dari beberapa algoritma yaitu K-Means, K-Medoids, K-Mode, PAM, CLARANS, CLARA, dan FCM. Dari beberapa metode tersebut, yang sering digunakan yaitu K-Means, K-Medoids dan FCM. K-Means memiliki kekurangan yaitu sensitivitas yang kuat terhadap pencilan, tidak menjamin konvergen ke minimum global sehingga menjadi sensitif terhadap inisialisasi centroid. Penyiapan yang berbeda dapat memberikan hasil yang berbeda. FCM memiliki kelemahan yaitu peka terhadap nilai awal k dan p, dan juga peka terhadap pencilan. Sehingga berpengaruh pada hasil yang berbeda pada penyiapan yang berbeda. K-Medoids dipilih karena memiliki keunggulan dalam pengklasteran dataset campuran atau heterogeny, kompleksitas antar ruang cluster, pemilihan cluster, serta waktu eksekusi [4]. K-medoids memberikan hasil maksimal efisiensi komputasi dan menghasilkan nilai yang akurat daripada algoritma k-means tetapi masih mengalami keterbatasan yaitu pemilihan acak centroid awal dan pertukaran pasangan cluster, K-medoids masih terjebak dalam optima lokal yang menunjukkan kepekaan terhadap outlier dan inisialisasi [5].

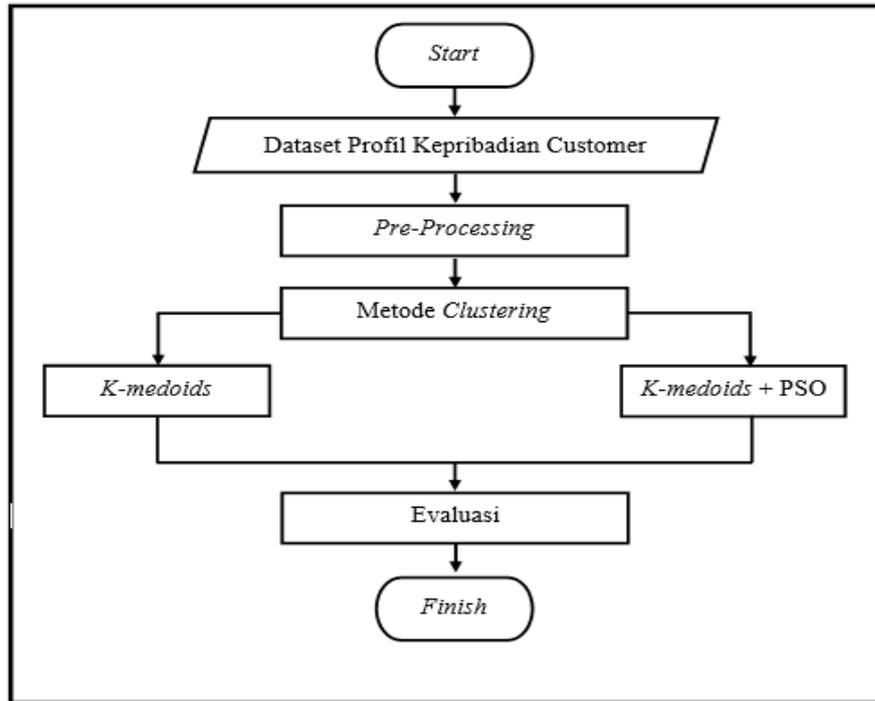
Baru-baru ini, penelitian lebih berkonsentrasi pada integrasi algoritma optimasi yang cocok dengan teknik clustering untuk mengatasi keterbatasan pengelompokan sehingga cluster yang lebih baik diperoleh secara signifikan [6]. Tidak diawasi clustering mencakup berbagai bidang yang mencirikan cocok kelemahan kombinatorial. Untuk mengatasi keterbatasan seperti itu, lebih tinggi sistem berbasis cerdas diperlukan sehingga beroperasi secara efektif untuk menghasilkan solusi optimal global atau mendekati optimal [7].

Kelemahan dari K-Medoids ini yang mendorong penggabungan K-Medoids dengan algoritma optimasi semacam Algoritma Genetika, Simulated Annealing (SA), Ant Colony Optimization (ACO) dan Particle Swarm Optimization (PSO) untuk mendapatkan hasil terbaik. Particle swarm optimization (PSO) adalah teknik optimasi stokastik yang kuat yang merupakan teknik optimasi sederhana dengan lebih sedikit parameter yang mampu menyelesaikan masalah yang kompleks. Keunggulan lain dari PSO yaitu dapat memecahkan berbagai masalah pengoptimalan yang sulit dan dengan tingkat konvergensi yang lebih cepat, tidak membutuhkan fungsi yang dapat dibedakan atau kontinu, secara komputasi tidak mahal baik dari segi kebutuhan memori dan kecepatan, mudah diimplementasikan karena memiliki sangat sedikit parameter algoritme untuk disesuaikan, dapat dengan mudah diparalelkan untuk pemrosesan bersamaan [8]. Dalam beberapa tahun terakhir, Particle Swarm Optimization (PSO) telah berhasil diterapkan untuk sejumlah masalah pengelompokan dunia nyata dengan konvergensi cepat dan efektif untuk data dimensi tinggi [9]. Maka dari itu, tujuan dari penelitian ini adalah mengoptimasi pemilihan centroid awal dari K-Medoids dengan algorithm optimasi Particle Swarm Optimization (PSO) dalam pengelompokan atau clustering. Penggunaan Algoritma PSO ini untuk mengoptimalkan hasil yang maksimal dengan memberikan titik pusat awal untuk clustering. Dengan hal ini, diharapkan hasil penelitian dapat mengoptimalkan hasil clustering yang akan dibentuk untuk segmentasi customer dengan menggunakan dataset profil kepribadian customer.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini memiliki beberapa tahapan, mulai dari pengumpulan dataset sampai ke tahap evaluasi yang digambarkan pada flowchart berikut ini:



Gambar 1 Flowchart penelitian

Eksperimen pada penelitian ini memiliki beberapa tahapan berikut ini:

1. Pengumpulan data
Pertama kali yang dilakukan dalam penelitian yaitu dengan pengumpulan data. Data yang akan pakai dalam penelitian atau eksperimen ini yaitu data profil kepribadian customer, dari Kaggle.com dengan alamat berikut ini <https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis>.
2. Pre-processing
Data Preprocessing yang dilakukan merupakan data cleaning dan menyiapkan data dengan menghapus konsistensi data, redundant dan data tidak komplit yang terdapat dalam data awal. Data preprocessing juga melakukan handling missing [10] data pada data (null) menjadi 0.
3. Usulan Metode
Pada tahap ini membahas tentang metode yang akan dipakai dalam eksperimen. Metode yang diusulkan yaitu optimasi K-Medoids dengan Particle Swarm Optimization (PSO).
4. Evaluasi
Pada tahap ini akan memaparkan pengevaluasian dari eksperimen yang dilakukan dalam penelitian yaitu dengan Davies-Bouldin Index (DBI).

2.2 K-Medoids

Kelompok metode partitional clustering dengan memanfaatkan jarak dari nilai minimal antar titik yang berlabel didalam sebuah cluster dan juga titik yang menjadi pusat cluster adalah penjelasan dari K-Medoids [11]

Berikut ini adalah tahapan *K-Medoids*:

1. Menginisialisasi centroid *cluster* sejumlah *k* atau total *cluster*
2. Mengalokasikan tiap data (objek) kedalam cluster yang paling dekat untuk digunakan persamaan ukuran jarak dengan persamaan *Euclidian Distance*:

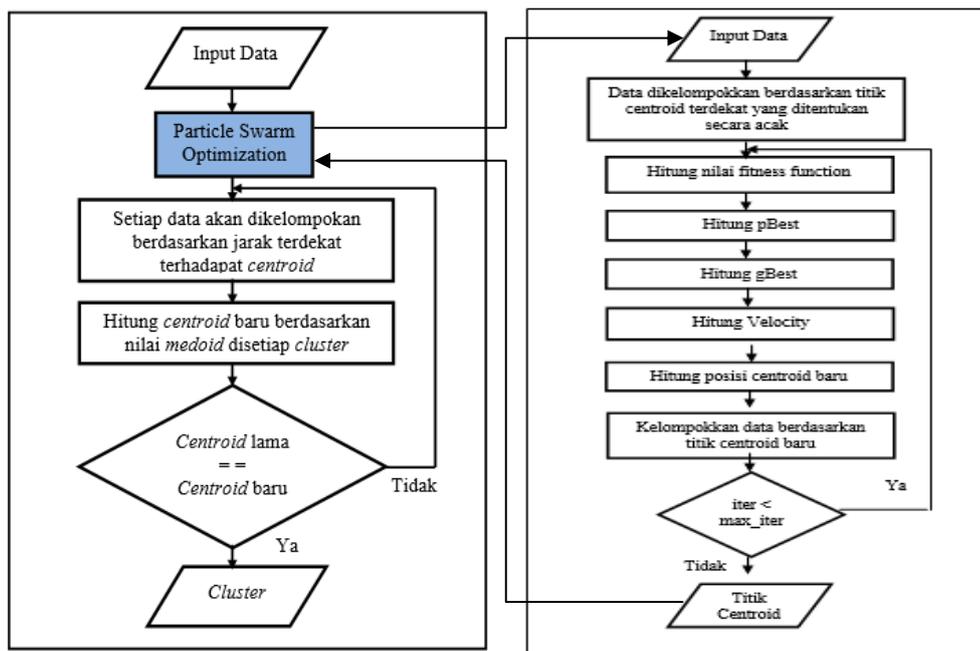
$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \quad (1)$$

Keterangan: - n adalah jumlah atribut
 - x_k dan y_k adalah atribut ke- k dari objek data x dan y

3. Memilih dengan acak objek pada tiap-tiap *cluster* untuk dijadikan kandidat *medoid* yang baru.
4. Menghitung jarak dari tiap objek yang berada pada tiap-tiap *cluster* untuk dijadikan kandidat *medoid* baru.
5. Menghitung total simpangan (S) dengan perhitungan nilai total *distance* yang baru – total *distance* yang lama. Jika $S < 0$, maka tukar objek dengan data *cluster* untuk dibentuk menjadi sekumpulan k objek baru sebagai *medoid*.
6. Mengulangi langkah 3 sampai 5 sehingga tidak terjadi sebuah perubahan *medoid*, sehingga akan didapatkan *cluster* beserta anggota tiap-tiap *cluster*.

2.3 PSO + K-Medoids

Perilaku yang menggambarkan algoritma ini yaitu kumpulan dari serangga, seperti lebah, rayap, burung atau semut adalah pengibaratan Particle swarm optimization (PSO) [8]. Usulan dalam penelitian ini yaitu menggunakan *Particle swarm optimization* (PSO) dalam penentuan centroid awal pada algoritma *K-Medoids*.



Gambar 2 Flowchart PSO+K-Medoids

Berikut ini adalah tahapan PSO+K-Medoids:

1. Menentukan jumlah partikel lalu menginisialisasi posisi awal dan kecepatan tiap partikel.
2. Untuk selanjutnya pada tiap iterasi yang terbentuk, dilakukan perhitungan *fitness function* dengan *cluster* yang tercipta tersebut dengan rumus berikut:

$$s(i) = 1 + decimal(s) \times \frac{a - 1}{2^n - 1} \quad (2)$$

Dimana a adalah bilangan bulat 1 sampai jumlah data, $decimal(s)$ menginterpretasikan strib biner (s) dan mengembalikan angka decimal yang setara.

3. Perhitungan *Pbest* dengan persamaan (3) dan perhitungan *Gbest* dengan persamaan (4).

$$c_2 r_2 [P_{best} - x_j(i - 1)], j = 1, 2, \dots, N \quad (3)$$

$$Gbest(c) = \max_{i=1,2,\dots,n_c} Pbest(i) \quad (4)$$

4. Tahap selanjutnya dengan menghitung kecepatan v dengan persamaan (5) dan menghitung partikel x dengan dasar kecepatan menggunakan (6). Apabila jumlah dari iterasi telah mencapai batas maksimal iterasi akan dihentikan, iterasi yang ditetapkan dalam penelitian ini sebesar 4 iterasi.

$$v_i^k = v_i^k + c_1 r_1 (p_i^k - x_i^k) + c_2 r_2 (p_i^k - x_i^k) \quad (5)$$

$$x_i^k = x_i^k + v_i^k \quad (6)$$

5. Titik centroid yang dihasilkan dari perhitungan *particle swarm optimization* ini nantinya digunakan sebagai pusat *cluster* awal dari *k-medoids*.

3. HASIL DAN PEMBAHASAN

3.2 Dataset Kepribadian Customer

Dataset yang akan digunakan dalam penelitian ini adalah dataset publik yang diperoleh dari Kaggle.com. Teknik kuantitatif adalah teknik yang digunakan dalam pengumpulan data, dengan mengambil dataset tentang profil customer atau kepribadian pelanggan dari <https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis>. Berikut adalah dataset yang akan digunakan dalam penelitian:

Tabel 1 Dataset Kepribadian Customer

Attribut	Type Data	Attribut	Type Data	Attribut	Type Data
ID	Numeric	MntFruits	Numeric	AcceptedCmp3	Binary
Year_Birth	Numeric	MntMeatProducts	Numeric	AcceptedCmp4	Binary
Education	Nominal	MntFishProducts	Numeric	AcceptedCmp5	Binary
Marital_Status	Nominal	MntSweetProducts	Numeric	AcceptedCmp1	Binary
Income	Numeric	MntGoldProds	Numeric	AcceptedCmp2	Binary
Kidhome	Numeric	NumDealsPurchases	Numeric	Complain	Binary
Teenhome	Numeric	NumWebPurchases	Numeric	Z_CostContact	Numeric
Dt_Customer	Date	NumCatalogPurchases	Numeric	Z_Revenue	Numeric
Recency	Numeric	NumStorePurchases	Numeric	Response	Binary
Mntwines	Numeric	NumWebVisitsMonth	Numeric		

3.3 Pre-Processing

Data Preprocessing yang dilakukan merupakan data cleaning dan menyiapkan data dengan menghapus konsistensi data, redundant dan data tidak komplit yang terdapat dalam data awal. Data preprocessing juga melakukan handling missing [10] data pada data (null) menjadi 0. Berikut adalah tahapan dalam proses pre-processing:

3.3.1 Menambah *Attribute* dan Menghapus *Attribute* yang tidak diperlukan

Penambahan dan menghapus *attribute* dilakukan agar proses clustering pada dataset bisa diolah dengan baik karena beberapa *attribute* yang kurang tepat, berikut adalah *attribute* yang ditambahkan dan dihapus.

Tabel 2 *attribute* yang ditambahkan dan dihapus

Attribut Baru	Attribut yang dihapus	Attribut yang dihapus	Attribut yang dihapus
Age	Year_Birth	Education	Marital_Status
Number of Child	Kidhome	Teenhome	Dt_Customer
Total Spend	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5
Total campaign acc	AcceptedCmp1	AcceptedCmp2	Complain
Total Purchases	Z_CostContact	Z_Revenue	Response

3.3.2 Mengatasi Missing Value

Dataset yang memiliki Missing Value tidak dapat diolah karena sebagian besar data mining tidak dapat bekerja dengan kumpulan data yang hilang nilainya [12]. Data yang mengalami missing value berjumlah 24 record, jadi dataset yang akan diolah berjumlah 2216 record

3.3.3 Normalisasi

Metode normalisasi digunakan untuk menggabungkan skor pencocokan heterogeny [13]. Metode normalisasi data yang akan digunakan pada dataset diatas adalah simple feature scaling atau range transformation, kisaran nilai dari normalisasi antara 0 dan 1. Berikut ini adalah tabel dari hasil normalisasi:

Tabel 3 Dataset setelah preprocessing

ID	Age	Income	Number_of_Child	Recency	MntWines	...	Total_purchases
5524	5524	0.38	0.35	0.00	0.59	...	0.69
2174	2174	0.41	0.28	0.67	0.38	...	0.13
4141	4141	0.30	0.43	0.00	0.26	...	0.63
...
9405	0.41	0.32	0.67	0.40	0.06	...	0.25

Inisialisasi dilakukan untuk memberi inisial tiap *attribute* agar mudah dalam perhitungan clustering, berikut adalah *attribute* dan inisialnya:

Tabel 4 Inisialisasi atribut

Attribute	Inisial	Attribute	Inisial	Attribute	Inisial
ID	Id	MntFruits	X6	NumCatalogPurchases	X13
Age	X1	MntMeatProducts	X7	NumStorePurchases	X14
Income	X2	MntFishProducts	X8	NumWebVisitsMonth	X15
Number_of_Child	X3	MntSweetProducts	X9	Total_spend	X16
Recency	X4	NumDealsPurchases	X11	total_campaign_acc	X17
MntWines	X5	NumWebPurchases	X12	Total_purchases	X18

3.4 Clustering K-Medoids

Pada metode K-medoids ini tahapan awal yaitu menentukan jumlah cluster, lalu mencari pusat cluster secara random, kemudian menghitung jarak setiap data terhadap pusat cluster dengan Euclidean distance. Kemudian mengelompokkan data berdasarkan jarak objek ke pusat cluster terdekat. Kemudian menghitung nilai simpangan, ulangi proses sampai nilai $S < 0$.

3.4.1 Penentuan Jumlah Cluster

Eksperimen dilakukan dengan menentukan jumlah cluster yang sudah ditentukan yaitu 4 cluster. Hasil evaluasi cluster nanti dihitung dengan menggunakan Davies Bouldin Index. Apabila mendekati nilai 0 maka cluster tersebut dinyatakan baik atau optimal.

3.4.2 Penentuan Pusat Cluster

Pada iterasi 1 melakukan penentuan pusat cluster yang dilakukan secara random. Record data 10 dan 474. Dipilih menjadi pusat cluster awal sehingga didapatkan pusat cluster seperti pada tabel 4.6 berikut:

Tabel 5 Pusat cluster awal

Data	Id	X1	X2	X3	...	X18	Cluster
10	5899	0.45	0.02	0.67	...	0.03	C1
474	5172	0.19	0.39	0.33	...	0.47	C2

3.4.3 Penentuan Jarak data dengan Pusat Cluster

Setelah mengetahui pusat *cluster* kemudian dihitung jarak pusat cluster terhadap data dengan rumus:

$$De = \sqrt{(xi - si)^2 + (yi - ti)^2}$$

Untuk record data ke 1 dengan pusat cluster 1 jaraknya =

$$\sqrt{(0,38 - 0,45)^2 + (0,35 - 0,02)^2 + (0,00 - 0,67)^2 + \dots (0,69 - 0,03)^2} = 1,78$$

Untuk record data ke 1 dengan pusat cluster 2 jaraknya =

$$\sqrt{(0,38 - 0,45)^2 + (0,35 - 0,02)^2 + (0,00 - 0,67)^2 + \dots (0,69 - 0,47)^2} = 1,18$$

Demikian seterusnya untuk record data yang lainnya.

3.4.4 Kelompokkan data berdasarkan dengan jarak terdekat dngan pusat cluster

Setelah didapatkan hasil Euclidean distance nya maka ditentukan jarak terpendek dengan pusat cluster. Untuk Euclidean pada record data ke 1 jarak terpendek antara 1,78 dan 1,18 adalah 1,18. Maka letak untuk record data ke 1 adalah di cluster 2. Demikian seterusnya untuk record data yang lainnya. Sehingga didapatkan data seperti pada data seperti berikut:

Tabel 6 Cluster iterasi 1

No	Id	C1	C2	Kedekatan	Cluster
1	5524	1.78	1.18	1.18	2
2	2174	0.89	1.02	0.89	1
3	4141	1.71	0.77	0.77	2
...
2216	9405	0.87	0.89	0.87	1

3.4.5 Menghitung simpangan awal

Menghitung nilai simpangan dari jarak Euclidean distance dari jarak tiap cluster, simpangan awal.

Tabel 7 Simpangan awal

C1	C2	Total Simpangan
2966,54	2201,70	5168.25

Hasil dari simpangan awal dari cluster yang terbentuk adalah 5168.25, kemudian tahap selanjutnya yaitu menentukan pusat cluster non-medoids.

3.4.6 Menghitung Simpangan Iterasi 1 dan Iterasi 2

Menghitung nilai simpangan dari jarak Euclidean distance dari jarak tiap cluster. Simpangan dari iterasi 1 dan iterasi 2 dibandingkan, jika $S < 0$ maka dilanjutkan, jika sebaliknya maka dihentikan.

Tabel 8 Simpangan iterasi 2

C1	C2	Total Simpangan
2517.84	2249.92	4767.76

$$S = \text{iterasi 2} - \text{Iterasi 1} = 4767.76 - 5168.25 = -400.48$$

Dengan hasil tersebut, maka iterasi dihentikan dan menghasilkan cluster sebagai berikut:

3.4.7 Cluster yang terbentuk

Iterasi 2 menjadi hasil akhir dari clustering dengan hasil cluster sebagai berikut:

Tabel 9 Cluster terbentuk

No	Id	X1	X2	X3	...	X18	Cluster
1	5524	0.38	0.35	0.00	...	0.69	2
2	2174	0.41	0.28	0.67	...	0.13	1
3	4141	0.30	0.43	0.00	...	0.63	2
...
2216	9405	0.41	0.32	0.67	...	0.25	2

3.5 Clustering PSO+K-Medoids

Penentuan pusat cluster menggunakan Particle Swarm Optimization sehingga tidak perlu dilakukan secara random. Proses awal PSO adalah menentukan jumlah partikel yang akan digunakan, lalu inialisasi posisi awal partikel, hitung fitness setelah itu menghitung Pbest dan Gbest, lalu update posisi dilakukan untuk menentukan posisi terbaru tiap partikel berdasarkan hasil update kecepatan sebelumnya. Setelah didapatkan nilai kecepatan, maka dilanjutkan dengan perhitungan posisi partikel tersebut sehingga mendapatkan nilai Pbest dan Gbest terbaru, kemudian update Pbest dan Gbest, yaitu dengan fitness dari update posisi. Nilai Pbest dan Gbest digunakan untuk iterasi selanjutnya sampai batas iterasi yang ditentukan. Hasil diperoleh menjadikan titik centroid awal [14].

3.5.1 Penentuan Jumlah Partikel

Pada pemilihan jumlah artikel atau N yang digunakan sebaiknya ukurannya tidak terlalu besar maupun terlalu kecil, agar ada banyak kemungkinan posisi menuju solusi optimal. Jika terlalu besar juga akan membuat perhitungan jadi lebih panjang, sehingga diambil jumlah partikel secara acak yaitu N=20.

3.5.2 Penentuan Posisi, Kecepatan, Pbest dan Gbest awal partikel

Pada penentuan posisi awal didapatkan dengan melakukan nilai rata-rata dari 20 jumlah partikel. Dengan rumus sebagai berikut:

$$\text{Partikel 1} = \bar{X} = \sum_{i=1}^{18} \frac{X_i}{18} = \frac{0,28+0,38+0,33+\dots+0,63}{18} = 0,25$$

$$\text{Partikel 2} = \bar{X} = \sum_{i=1}^{18} \frac{X_i}{18} = \frac{0,46+0,47+0,33+\dots+0,75}{18} = 0,41$$

$$\text{Partikel 3} = \bar{X} = \sum_{i=1}^{18} \frac{X_i}{18} = \frac{0,29+0,43+0,00+\dots+0,63}{18} = 0,30$$

$$\text{Partikel 4} = \bar{X} = \sum_{i=1}^{18} \frac{X_i}{18} = \frac{0,25+0,55+0,00+\dots+0,47}{18} = 0,32$$

...

...

...

$$\text{Partikel 20} = \bar{X} = \sum_{i=1}^{18} \frac{X_i}{18} = \frac{0,48+0,45+0,00+\dots+0,78}{18} = 0,40$$

Tabel 10 Partikel awal

No	Id	X1	X2	X3	...	X18	X ⁻
1	7446	0.28	0.38	0.33	...	0.63	0.25
2	6565	0.46	0.47	0.33	...	0.75	0.41
3	10383	0.29	0.43	0.00	...	0.63	0.30
4	1592	0.25	0.55	0.00	...	0.47	0.32
5	7987	0.26	0.40	0.33	...	0.47	0.26
6	6422	0.41	0.53	0.00	...	0.50	0.30
7	8932	0.26	0.39	0.33	...	0.66	0.37
8	2379	0.42	0.42	0.00	...	0.38	0.31
9	10172	0.08	0.42	0.00	...	0.63	0.32
10	7030	0.40	0.40	0.33	...	1.00	0.35
11	7999	0.40	0.46	0.00	...	0.50	0.29
12	203	0.20	0.49	0.00	...	0.66	0.33
13	10928	0.37	0.41	0.33	...	0.47	0.24
14	7761	0.17	0.23	0.33	...	0.22	0.16
15	4297	0.26	0.13	0.33	...	0.16	0.14
16	2445	0.21	0.17	0.33	...	0.16	0.14
17	8686	0.17	0.14	0.33	...	0.16	0.14
18	7937	0.14	0.14	0.00	...	0.19	0.44
19	4127	0.28	0.47	0.33	...	1.00	0.12
20	6203	0.48	0.45	0.00	...	0.78	0.40

Dari tabel tersebut diketahui posisi awal partikel dengan nama record x awal. Untuk iterasi awal yaitu iterasi 0 kecepatan awal adalah 0 untuk semua partikel dikarenakan partikel masih berapa ditempat awal belum ke solusi yang diinginkan. Pbest awal adala sama dengan nilai posisi awal partikel.

3.5.3 Penentuan Fitness

Penentuan nilai fitness digunakan untuk menentukan solusi mencari titik centroid awal. Dengan rumus $k=1+\text{decimal}(s) \times \frac{a-1}{2^n-1}$

Dimana a adalah bilangan bulat antara 1 sampai jumlah data, decimal(s) menginterpretasikan string biner (s) dan mengembalikan angka decimal yang setara. Maka nilai fitness setiap partikel adalah

$$\text{Partikel 1} = 1 + \text{decimal}(s) \times \frac{19-1}{2^{0,25}-1} = 98$$

Dan seterusnya sampai partikel ke 20.

3.5.4 Penentuan Pbest dan Gbest

Pbest diperoleh dari hasil pengurangan dari $X|$ (nilai rata-rata) dengan V (kecepatan).

Partikel 1 = $\bar{X}+V = 0,25 - 0 = 0,25$

Partikel 2 = $\bar{X}+V = 0,41 - 0 = 0,41$

Partikel 3 = $\bar{X}+V = 0,30 - 0 = 0,30$

...

Partikel 20 = $\bar{X}+V = 0,40 - 0 = 0,40$

Gbest merupakan nilai terbaik seluruh data. Gbest didapatkan dari nilai minimal dari Pbest untuk iterasi 1, setelah update kecepatan nilai Gbest diperoleh dari nilai Pbest minimal.

Gbest = min Pbest = 0,14

Maka didapatkan Gbest untuk iterasi 1 adalah 0,14 dengan posisi partikel 19.

Tabel 11 Iterasi 1 PSO

Partikel	\bar{X}	V	a-1	2xi-1	Fitness	Pbest	Gbest	Posisi partikel
1	0.25	0	18	0.19	98	0.25	0,12	19
2	0.41	0	18	0.33	56	0.41		
3	0.30	0	18	0.23	80	0.30		
4	0.32	0	18	0.25	74	0.32		
5	0.26	0	18	0.20	92	0.26		
6	0.30	0	18	0.23	79	0.30		
7	0.37	0	18	0.29	63	0.37		
8	0.31	0	18	0.24	76	0.31		
9	0.32	0	18	0.25	73	0.32		
10	0.35	0	18	0.27	67	0.35		
11	0.29	0	18	0.22	83	0.29		
12	0.33	0	18	0.25	72	0.33		
13	0.24	0	18	0.18	101	0.24		
14	0.16	0	18	0.12	156	0.16		
15	0.14	0	18	0.10	179	0.14		
16	0.14	0	18	0.10	173	0.14		
17	0.14	0	18	0.10	175	0.14		
18	0.44	0	18	0.36	52	0.44		
19	0.12	0	18	0.09	209	0.12		
20	0.40	0	18	0.32	58	0.40		

Pada tabel diatas diketahui iterasi 1 dari PSO menghasilkan Gbest 0,12 dan posisi partikel di 19. Dengan ini pusat awal cluster untuk $K=1$ yaitu pada data dengan id 4127 dan urutan record ke 1653.

3.5.5 Penentuan Update Kecepatan

Langkah selanjutnya adalah menghitung update kecepatan setiap partikel. Proses ini merupakan penentuan kemana arah partikel akan berjalan. Rumus update kecepatan partikel adalah: $v_i^k = v_i^k + c_1r_1(p_i^k - x_i^k) + c_2r_2(p_i^k - x_i^k)$

Dimana c_1 dan c_2 masing-masing adalah learning rates untuk kemampuan individu dan pengaruh kawanan, r_1 dan r_2 adalah bilangan random yang bernilai interval antara 0 dan 1. Jadi parameter c_1 dan c_2 menunjukkan bobot dari memory (posisi) sebuah partikel terhadap memory (posisi) dari kelompok. Nilai dari c_1 dan c_2 adalah 2 sehingga perkalian c_1r_1 dan c_2r_2 memastikan bahwa partikel-partikel akan mendekati target sekitar setengah selisihnya, Maka nilai update kecepatan setiap partikel adalah:

Partikel 1: $0 + \text{Random } 0-1 \times 2 (0,25-0,25) + \text{random } 0-1 \times 2 (0,12-0,25) = -0.2289$

Selanjutnya update Gbest yang ditentukan dengan cara sbagai berikut.

Gbest (2) = Min Pbest = 0,02

Nilai Gbest iterasi 2 diperoleh dari nilai Pbest minimal yaitu pada posisi partikel 1 sehingga iterasi 2 didapat pada tabel berikut:

Tabel 12 Iterasi 2 PSO

Partikel	X ⁻	V	a-1	2xi-1	Fitness	Pbest	Gbest	Posisi partikel
1	0.25	-0.23	18	0.19	98	0.02	0.02	1
2	0.41	-0.23	18	0.33	56	0.18		
3	0.30	-0.18	18	0.23	80	0.12		
4	0.32	-0.20	18	0.25	74	0.12		
5	0.26	-0.03	18	0.20	92	0.23		
6	0.30	-0.04	18	0.23	79	0.26		
7	0.37	-0.45	18	0.29	63	-0.08		
8	0.31	-0.04	18	0.24	76	0.27		
9	0.32	-0.04	18	0.25	73	0.28		
10	0.35	-0.05	18	0.27	67	0.30		
11	0.29	-0.20	18	0.22	83	0.09		
12	0.33	-0.29	18	0.25	72	0.04		
13	0.24	-0.10	18	0.18	101	0.14		
14	0.16	-0.02	18	0.12	156	0.14		
15	0.14	0.00	18	0.10	179	0.14		
16	0.14	-0.02	18	0.10	173	0.12		
17	0.14	-0.02	18	0.10	175	0.12		
18	0.12	-0.01	18	0.09	202	0.12		
19	0.44	-0.58	18	0.36	51	-0.14		
20	0.40	-0.11	18	0.32	58	0.29		

Pada tabel diatas diketahui iterasi 2 dari PSO yang menghasilkan Gbest 0,02 dan posisi partikel 1. Dengan ini pusat awal cluster untuk K=2 yaitu pada data dengan id 7446 dan urutan record ke 6.

3.5.6 Penentuan Pusat Cluster

Pada PSO yang telah dilakukan, kemudian melakukan penentuan pusat *cluster* yang dilakukan dengan hasil beberapa partikel. Record data 1653 dan 6 dipilih menjadi pusat cluster awal sehingga didapatkan pusat cluster seperti pada tabel 4.20 berikut:

Tabel 13 Pusat cluster awal PSO

Data	Id	X1	X2	X3	...	X18	Cluster
1653	4127	0.28	0.47	0.33	...	1.00	C1
6	7446	0.28	0.38	0.33	...	0.63	C2

3.5.7 Penentuan Jarak Data dengan Pusat Cluster

Setelah mengetahui pusat *cluster* kemudian dihitung jarak pusat cluster terhadap data dengan rumus:

$$De = \sqrt{(xi - si)^2 + (yi - ti)^2}$$

Untuk record data ke 1 dengan pusat cluster 1 jaraknya =

$$\sqrt{(0,38 - 0,28)^2 + (0,35 - 0,47)^2 + (0,00 - 0,33)^2 + \dots + (0,69 - 1,00)^2} = 0,97$$

Untuk record data ke 1 dengan pusat cluster 2 jaraknya =

$$\sqrt{(0,38 - 0,28)^2 + (0,35 - 0,38)^2 + (0,00 - 0,33)^2 + \dots + (0,69 - 0,63)^2} = 1,18$$

Demikian seterusnya untuk record data yang lainnya.

3.5.8 Kelompokkan data berdasarkan dengan jarak terdekat dngan pusat cluster

Setelah didapatkan hasil Euclidean distance nyam aka ditentukan jarak terpendek dengan pusat cluster. Untuk Euclidean pada record data ke 1 jarak terpendek antara 0,97 dan 1,18 adalah 0,97. Maka letak untuk record data ke 1 adalah di cluster 1. Demikian seterusnya untuk record data yang lainnya. Sehingga didapatkan data seperti pada data seperti berikut:

Tabel 14 Cluster Iterasi 1

No	Id	C1	C2	Kedekatan	Cluster
1	5524	0.97	1.18	0.97	1
2	2174	1.82	1.06	1.06	2
3	4141	1.15	0.59	0.59	2
...
2216	9405	1.60	0.90	0.90	2

3.5.9 Menghitung simpangan awal

Menghitung nilai simpangan dari jarak Euclidean distance dari jarak tiap cluster, simpangan awal.

Tabel 15 Simpangan Iterasi 1 PSO

C1	C2	Total Simpangan
3167.08	2144.41	5311.49

Hasil dari simpangan awal dari cluster yang terbentuk adalah 5311.49, kemudian tahap selanjutnya yaitu menentukan pusat cluster non-medoids.

3.5.10 Menghitung Simpangan Iterasi 1 dan Iterasi 2

Menghitung nilai simpangan dari jarak Euclidean distance dari jarak tiap cluster. Simpangan dari iterasi 1 dan iterasi 2 dibandingkan, jika $S < 0$ maka dilanjutkan, jika sebaliknya maka dihentikan.

Tabel 16 Simpangan Iterasi 2 PSO

C1	C2	Total Simpangan
3167.08	2144.41	5311.49

$$S = \text{iterasi 2} - \text{Iterasi 1} = 4939.05 - 5311.49 = -372.43$$

Dengan hasil tersebut, maka iterasi dihentikan dan mendapatkan hasil akhir cluster.

3.5.11 Cluster yang terbentuk

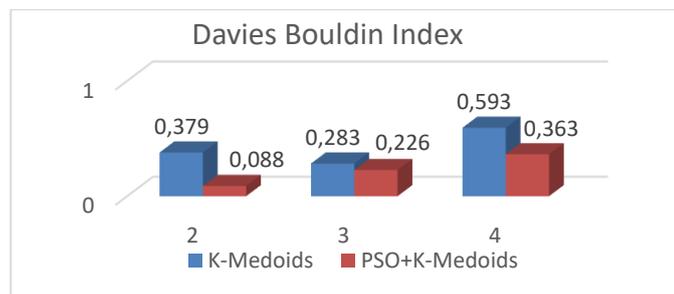
Iterasi 2 menjadi hasil akhir dari clustering dengan hasil cluster sebagai berikut:

Tabel 17 Cluster terbentuk PSO

No	Id	X1	X2	X3	...	X18	Cluster
1	5524	0.38	0.35	0.00	...	0.69	1
2	2174	0.41	0.28	0.67	...	0.13	2
3	4141	0.30	0.43	0.00	...	0.63	2
...
2216	9405	0.41	0.32	0.67	...	0.25	2

3.6 Perbandingan Hasil nilai DBI metode K-Medoids dengan PSO+K-Medoids

Pada tahap ini dilakukan perbandingan hasil Davies Bouldin Index (DBI) [15] dari metode K-Medoids dan PSO+K-Medoids. Dengan hasil nilai DBI yang diperoleh dengan $K=4$ diuji coba K-Medoids dan PSO+K-Medoids dengan dataset Profil Kepribadian Customer yang telah dilakukan menghasilkan nilai DBI yang akan diperlihatkan dalam grafik berikut:



Gambar 3 Grafik perbandingan DBI

Pada gambar diatas menjelaskan pada clustering dengan K-Medoids menghasilkan nilai DBI $K=2$ yaitu 0,379, sedangkan $K=3$ adalah 0,283, untuk $K=4$ adalah 0,593. Sedangkan untuk clustering menggunakan optimasi PSO+K-Medoids pada $K=2$ menghasilkan DBI 0,088, untuk $K=3$ yaitu 0,226, sedangkan $K=4$ adalah 0,363. Dari hasil clustering dengan nilai evaluasi DBI menghasilkan optimasi PSO pada K-Medoids menjadi yang terbaik karena nilai DBI lebih rendah dari pada hasil clustering K-Medoids standar.

4. KESIMPULAN DAN SARAN

Pada penelitian ini dilakukan pengujian model dengan menggunakan metode K-Medoids dalam penentuan pusat centroid awal menggunakan Particle Swarm Optimization dengan tujuan untuk mengoptimalkan kinerja algoritma K-Medoids sehingga dapat bermanfaat untuk clustering data menjadi lebih baik. PSO dapat langsung memberikan pusat centroid awal secara optimal dan hasil dari clustering menjadi lebih baik. Hasil pengujian menunjukkan secara umum dengan percobaan 4 cluster, PSO+K-Medoids lebih unggul dibandingkan K-Medoids standar. Meski demikian, masih perlu dilakukan penelitian lebih lanjut dalam penentuan jumlah cluster terbaik, supaya hasil clustering dapat optimal dan menjadi rekomendasi terbaik dalam clustering

DAFTAR PUSTAKA

- [1] B. E. Adiana, I. Soesanti and A. E. Permanasari, "Analisis Segmentasi Pelanggan Menggunakan Kombinasi Rfm Model dan Teknik Clustering," *JUTEI (Jurnal Terapan Teknologi Informasi)*, pp. 23-32, 2018.
- [2] R. W. B. S. Berahmana, F. A. Mohammed and K. Chairuang, "Customer Segmentation Based on RFM Model Using K-Means, K-Medoids, and DBSCAN Methods," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 11, no. 1, p. 32–43, 2020.
- [3] P. Kotler, B. Molan, B. Sarwiji and K. Lane, *Marketing Management, Thirteen Edition*, Jakarta: Erlangga, 2009.
- [4] A. A. D. Sulistyawati and M. Sadikin, "Penerapan Algoritma K-Medoids Untuk Menentukan Segmentasi Pelanggan," *SISTEMASI: Jurnal Sistem Informasi*, vol. 10, no. 3, pp. 516-526, 2021.
- [5] D. Yu, L. Guojun, M. Guo and X. Liu, "An improved K-Medoids based on step increasing and optimizing medoids," *Expert System with Applications*, vol. 92, pp. 464-473, 2018.
- [6] S. K. Majhi and S. Biswal, "Optimal cluster analysis using hybrid K-Means and Ant Lion Optimizer," *Karbala International Journal of Modern Science*, vol. 4, no. 4, pp. 347-360, 2018.
- [7] C. Mageshkuma, S. Karthik and P. Arunachalam, "Hybrid metaheuristic algorithm for improving the efficiency of data clustering," *Cluster Computing*, 2018.
- [8] J. Kennedy and R. Eberhart, "Particle swarm optimization," *In Proceedings of the IEEE International Conference on Neural Networks. IEEE Service Center, Piscataway*, 1995.
- [9] C. P. Lekshmy and N. Abdul, "An Improved Clustering Algorithm based on K-Means and Harmony Search Optimization," *IEEE*, 2011.
- [10] D. T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*, Wiley, 2005.
- [11] D. F. Pramesti, M. T. Furqon and C. Dewi, "Implementasi Metode K-Medoids Clustering Untuk Pengelompokan Data Potensi Kebakaran Hutan/Lahan Berdasarkan Persebaran Titik Panas (Hotspot)," *Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 9, pp. 723-732, 2017.
- [12] Gautam, Chandan and R. Vadlamani, "Data imputation via evolutionary computation, clustering and a neural network," *Neurocomputing*, vol. 156, pp. 134-142, 2015.
- [13] J. Peshawa, A. Muhammad and F. H. Rezhna, "Data Normalization and Standardization: A Technical Report," *Machine Learning Technical Reports*, vol. 1, no. 1, pp. 1-6, 2014.
- [14] S. S. Rao, *Engineering Optimization, Theory and Practice*. fourth edition, New York: John Wiley & Sons, 2009.
- [15] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *EEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, no. 2A, 1979.