



Perbandingan Kinerja K-Medoids dan Improved K-Medoids Berbasis Crow Search Algorithm pada Klasterisasi Data Transaksi Penjualan Berdasarkan Silhouette Score dan Efisiensi Komputasi

Melinda Putri Azzahra*, Wahyu Syaifullah J S, Muhammad Nasrudin

Fakultas Ilmu Komputer, Prodi Sains Data, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia

Email: ^{1,*}22083010035@student.upnjatim.ac.id, ²wahyu.s.j.saputra.if@upnjatim.ac.id, ³nasrudin.fasilkom@upnjatim.ac.id

Email Penulis Korespondensi: 22083010035@student.upnjatim.ac.id

Abstrak—Perkembangan sistem transaksi digital menghasilkan data dalam jumlah besar yang perlu diolah menjadi informasi yang dapat mendukung pengambilan keputusan. Salah satu pendekatan yang dapat digunakan untuk menganalisis pola pembelian konsumen adalah klasterisasi. Algoritma K-Medoids dikenal memiliki kemampuan yang baik dalam menangani outlier, namun proses pencarian medoid secara iteratif menyebabkan waktu komputasi yang relatif tinggi. Untuk mengatasi permasalahan tersebut, digunakan metode improved K-Medoids berbasis Crow Search Algorithm (CSA) yang memanfaatkan mekanisme optimasi berbasis metaheuristik dalam pencarian medoid. Penelitian ini bertujuan untuk membandingkan kinerja algoritma K-Medoids dan improved K-Medoids berbasis CSA pada klasterisasi data transaksi penjualan berdasarkan kualitas cluster dan efisiensi waktu komputasi. Data yang digunakan berasal dari sistem Point of Sale (POS) gerai makanan cepat saji yang terdiri atas 18.814 data transaksi. Tahapan penelitian meliputi pra-pemrosesan data, proses klasterisasi menggunakan kedua metode, serta evaluasi hasil menggunakan Silhouette Score dan waktu komputasi. Hasil penelitian menunjukkan bahwa kedua metode menghasilkan jumlah cluster optimal yang sama, yaitu $K = 4$. Algoritma K-Medoids menghasilkan nilai Silhouette Score tertinggi sebesar 0,557791, sedangkan improved K-Medoids berbasis CSA menghasilkan nilai Silhouette Score sebesar 0,537240. Dari sisi efisiensi, improved K-Medoids berbasis CSA mampu menghasilkan waktu komputasi yang jauh lebih singkat dan relatif stabil dibandingkan K-Medoids. Hasil penelitian menunjukkan bahwa terdapat trade-off antara kualitas cluster dan efisiensi waktu komputasi, sehingga pemilihan metode dapat disesuaikan dengan kebutuhan analisis. Kontribusi penelitian ini adalah menyajikan analisis komparatif antara algoritma K-Medoids dan improved K-Medoids berbasis CSA pada data transaksi penjualan dengan mempertimbangkan kualitas cluster dan efisiensi waktu komputasi secara bersamaan. Temuan penelitian ini memberikan rekomendasi dalam pemilihan metode klasterisasi sesuai dengan kebutuhan analisis serta dapat menjadi referensi bagi pengembangan penelitian di bidang analisis data transaksi penjualan.

Kata Kunci: K-Medoids; Crow Search Algorithm (CSA); Klasterisasi; Silhouette Score; Data Transaksi

Abstract—The development of digital transaction systems generates large amounts of data that need to be processed into meaningful information to support decision-making. One approach that can be used to analyze consumer purchasing patterns is clustering. The K-Medoids algorithm is known for its robustness against outliers; however, its iterative medoid search process leads to relatively high computational time. To address this limitation, an improved K-Medoids based on the Crow Search Algorithm (CSA) is employed, utilizing a metaheuristic optimization mechanism to determine optimal medoids. This study aims to compare the performance of the K-Medoids algorithm and the improved K-Medoids based on the CSA in transaction data clustering in terms of cluster quality and computational efficiency. The dataset used was obtained from a Point of Sale (POS) system of a fast-food restaurant and consisted of 18,814 transaction records. The research stages included data preprocessing, clustering using both methods, and performance evaluation based on the Silhouette Score and computation time. The results showed that both methods produced the same optimal number of clusters, namely $K = 4$. The K-Medoids algorithm achieved the highest Silhouette Score of 0.557791, while the improved K-Medoids based on the CSA obtained a Silhouette Score of 0.537240. In terms of efficiency, the improved K-Medoids based on the CSA required significantly shorter and more stable computation times than the conventional K-Medoids algorithm. These findings indicate a trade-off between clustering quality and computational efficiency, implying that the choice of method can be adjusted according to analytical requirements. The main contribution of this study is providing a comparative analysis of the K-Medoids algorithm and the improved K-Medoids based on the CSA on transaction data by jointly evaluating cluster quality and computational efficiency. The findings provide practical recommendations for selecting clustering methods according to analytical requirements and serve as a reference for future research on transaction data clustering.

Keywords: K-Medoid; Crow Search Algorithm (CSA); Clustering; Silhouette Score; Transaction Data

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong peningkatan penggunaan sistem transaksi digital dalam berbagai sektor bisnis, termasuk pada industri makanan cepat saji. Setiap transaksi yang terjadi akan menghasilkan data dalam jumlah besar yang tersimpan secara sistematis. Data transaksi tersebut pada umumnya hanya digunakan sebagai arsip operasional, padahal di dalamnya terdapat informasi penting yang dapat dimanfaatkan untuk memahami pola pembelian konsumen [1]. Permasalahan yang sering terjadi adalah kurang optimalnya pemanfaatan data transaksi untuk menghasilkan informasi yang dapat mendukung pengambilan keputusan, seperti strategi penjualan, maupun pengelompokan konsumen berdasarkan karakteristik pembelian [2]. Oleh karena itu, diperlukan suatu metode yang mampu mengolah data transaksi tersebut menjadi informasi yang lebih bermakna.

Salah satu pendekatan yang dapat digunakan untuk menganalisis pola pembelian konsumen adalah *data mining*, khususnya metode klasterisasi. Klasterisasi merupakan proses pengelompokan data ke dalam beberapa kelompok berdasarkan kemiripan karakteristik tertentu [3]. Dengan melakukan klasterisasi, data transaksi dapat dikelompokkan menjadi beberapa segmen yang memiliki pola pembelian yang serupa, sehingga dapat membantu pelaku usaha dalam memahami perilaku konsumen. Salah satu algoritma klasterisasi yang cukup banyak digunakan adalah *K-Medoids*.

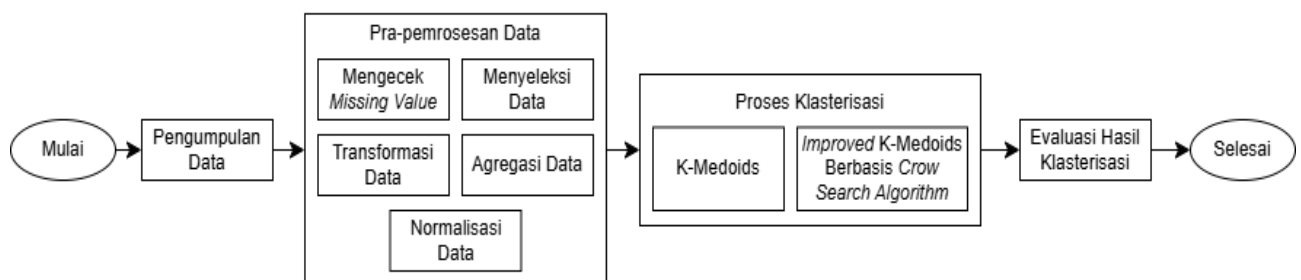
Algoritma ini memiliki keunggulan dibandingkan *K-Means*, karena lebih tahan terhadap *outlier* dan menggunakan objek data aktual sebagai pusat *cluster* (*medoid*), sehingga hasil klusterisasi lebih representatif [4]. Meskipun demikian, algoritma K-Medoids memiliki kelemahan pada proses pencarian *medoid* yang optimal. Proses iteratif yang dilakukan untuk mencari kombinasi *medoid* terbaik cenderung membutuhkan waktu komputasi yang cukup tinggi, terutama pada dataset dengan ukuran besar [5]. Hal ini menjadi kendala dalam penerapan K-Medoids pada data transaksi yang terus bertambah. Untuk mengatasi permasalahan tersebut, diperlukan pendekatan optimasi yang mampu meningkatkan efisiensi proses tanpa mengurangi kualitas hasil klusterisasi.

Beberapa penelitian sebelumnya telah mencoba mengatasi permasalahan tersebut dengan mengkombinasikan algoritma klusterisasi dengan metode optimasi berbasis *metaheuristic*. Penelitian oleh Fitriani *et al.* [6], menunjukkan bahwa integrasi K-Medoids dengan *Particle Swarm Optimization* (PSO) mampu meningkatkan kualitas klusterisasi melalui pemilihan *medoid* yang lebih optimal. Selain itu, penelitian oleh Sabrina dan Setiawan [7] mengemukakan bahwa algoritma K-Medoids memiliki keunggulan dalam menghasilkan *cluster* yang lebih stabil serta mampu menangani *outlier* dengan lebih baik dibandingkan metode lain. Penelitian lain oleh Yosia dan Siregar [8] melakukan analisis perbandingan antara K-Means dan K-Medoids pada data penjualan dan menunjukkan bahwa K-Medoids menghasilkan kualitas *cluster* yang lebih baik berdasarkan evaluasi *Silhouette Score* dan *Davies-Bouldin Index*. Selanjutnya, penelitian oleh Nitesh Sureja *et al.* [9] mengusulkan metode *improved* K-Medoids berbasis *Crow Search Algorithm* (CSA) yang mampu menyeimbangkan proses eksplorasi dan eksploitasi dalam pencarian solusi serta meningkatkan performa klusterisasi secara signifikan. Namun, penelitian tersebut masih menggunakan dataset umum dan lebih berfokus pada peningkatan kualitas hasil klusterisasi. Selain itu, sebagian besar penelitian sebelumnya belum banyak membahas perbandingan kualitas *cluster* dan efisiensi waktu komputasi secara bersamaan pada data transaksi penjualan yang memiliki karakteristik dinamis dan terus bertambah.

Berdasarkan penelitian sebelumnya, masih terdapat keterbatasan dalam kajian yang membandingkan kinerja algoritma K-Medoids dan *improved* K-Medoids berbasis CSA pada data transaksi penjualan dengan mempertimbangkan kualitas *cluster* dan efisiensi waktu komputasi secara simultan. Oleh karena itu, penelitian ini bertujuan untuk membandingkan kinerja kedua metode berdasarkan nilai *Silhouette Score* sebagai indikator kualitas *cluster* dan waktu komputasi sebagai indikator efisiensi proses. Kontribusi utama penelitian ini meliputi tiga aspek. Pertama, penelitian ini menyajikan perbandingan langsung antara algoritma K-Medoids dan *improved* K-Medoids berbasis CSA pada data transaksi penjualan. Kedua, evaluasi dilakukan menggunakan dua indikator, yaitu kualitas *cluster* berdasarkan *Silhouette Score* dan efisiensi waktu komputasi sehingga memberikan gambaran mengenai *trade-off* antara kedua metode. Ketiga, hasil penelitian memberikan rekomendasi dalam pemilihan metode klusterisasi sesuai dengan kebutuhan analisis, baik yang berorientasi pada kualitas hasil klusterisasi maupun efisiensi komputasi.

2. METODOLOGI PENELITIAN

Tahapan penelitian yang dilakukan pada penelitian ini dapat dilihat pada Gambar 1. Proses penelitian dimulai dari pengumpulan data transaksi penjualan, dilanjutkan dengan tahap pra-pemrosesan data, proses klusterisasi menggunakan algoritma K-Medoids dan *improved* K-Medoids berbasis *Crow Search Algorithm* (CSA), kemudian dilakukan evaluasi hasil klusterisasi menggunakan *Silhouette Score* dan waktu komputasi untuk membandingkan kinerja kedua metode yang digunakan..



Gambar 1. Alur Penelitian

2.1 Pengumpulan Data

Penelitian ini menggunakan data sekunder yang diperoleh dari sistem *Point of Sale* (POS) gerai makanan cepat saji. Data yang digunakan merupakan data transaksi penjualan yang telah diizinkan untuk diakses dan dianalisis dalam konteks penelitian ini. Proses pengumpulan data dilakukan selama beberapa periode operasional dengan tetap menjaga kerahasiaan informasi internal perusahaan dan tidak menampilkan identitas pribadi pelanggan.

2.2 Pra-pemrosesan Data

Pada tahap pra-pemrosesan data, dilakukan beberapa langkah untuk memastikan kualitas data sebelum proses analisis. Tahap awal diawali dengan pengecekan *missing value* pada dataset untuk memastikan tidak terdapat data kosong yang dapat memengaruhi hasil analisis, kemudian dilanjutkan dengan seleksi atribut untuk memilih variabel yang relevan



dengan tujuan penelitian serta mengeliminasi atribut yang tidak digunakan dalam proses analisis. Selanjutnya, dilakukan transformasi tipe data untuk mengonversi atribut jumlah pembelian dan nilai transaksi ke dalam format numerik agar dapat diproses secara matematis, khususnya pada metode klasterisasi berbasis jarak. Setelah itu, dilakukan agregasi data berdasarkan nomor transaksi untuk membentuk representasi perilaku pelanggan yang menghasilkan variabel utama berupa total jumlah item yang dibeli dan total nilai transaksi sehingga data menjadi lebih ringkas dan terfokus. Tahap akhir adalah normalisasi data menggunakan metode *Min-Max Scaling*, di mana setiap variabel ditransformasikan ke dalam rentang nilai antara 0 hingga 1. Proses ini bertujuan untuk menyamakan skala antar variabel agar tidak terjadi dominasi nilai tertentu dalam proses pengelompokan data, terutama pada metode klasterisasi berbasis jarak [13].

2.3 Proses Klasterisasi

2.3.1 K-Medoids

Algoritma K-Medoids dengan pendekatan *Partitioning Around Medoids* (PAM) digunakan dalam penelitian ini sebagai metode dasar klasterisasi untuk mengelompokkan pola transaksi penjualan konsumen. Metode ini dipilih karena lebih *robust* terhadap keberadaan *outlier* pada data transaksi, serta menggunakan objek data aktual sebagai pusat *cluster* (*medoid*) [10]. Berbeda dengan K-Means yang menggunakan nilai rata-rata (*centroid*) sebagai pusat *cluster*, K-Medoids menentukan pusat *cluster* berdasarkan objek data yang paling representatif di dalam *cluster*. Penggunaan objek data aktual sebagai *medoid* menyebabkan algoritma ini lebih stabil dalam menangani data yang memiliki *noise* atau distribusi yang tidak rata. Secara umum, tujuan utama algoritma K-Medoids adalah meminimalkan total jarak antara setiap data dengan *medoid cluster* terdekatnya. Fungsi objektif pada algoritma K-Medoids dapat dirumuskan sebagai berikut [10]:

$$E = \sum_{j=1}^K \sum_{xi \in C_j} d(xi, m_j) \quad (1)$$

dimana E menunjukkan total fungsi objektif K-Medoids yang diperoleh dari penjumlahan jarak seluruh data terhadap *medoid* pada masing-masing *cluster*. Variabel C_j menyatakan himpunan data pada *cluster* ke- j , sedangkan m_j merupakan *medoid* pada *cluster* tersebut. Nilai K menunjukkan jumlah *cluster* yang digunakan dalam proses klasterisasi, dan $d(x_i, m_j)$ merupakan jarak antara data ke- i dengan *medoid cluster* ke- j .

Dalam penelitian ini, perhitungan jarak antar data dilakukan menggunakan *Euclidean Distance* yang dirumuskan sebagai berikut [11]:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2)$$

dimana $d(x, y)$ menyatakan jarak *Euclidean* antara data x dan y . Variabel x_i dan y_i menunjukkan nilai atribut ke- i dari masing-masing data, sedangkan n merupakan jumlah atribut atau fitur yang digunakan dalam proses perhitungan jarak.

Proses klasterisasi menggunakan algoritma K-Medoids diawali dengan menentukan jumlah *cluster* (K) yang akan digunakan. Selanjutnya, sejumlah objek dipilih secara acak sebagai *medoid* awal. Jarak antara setiap data dengan *medoid* dihitung menggunakan *Euclidean Distance*, kemudian masing-masing data dikelompokkan ke dalam *cluster* dengan jarak terdekat [14]. Setelah proses pengelompokan selesai, dilakukan perhitungan total *cost* untuk mengevaluasi kualitas *medoid* yang digunakan. Tahap berikutnya adalah melakukan pertukaran kandidat *medoid* dan menghitung kembali total *cost* yang dihasilkan. Proses tersebut dilakukan secara iteratif hingga tidak terdapat perubahan *medoid* atau jumlah iterasi maksimum telah tercapai, sehingga diperoleh hasil klasterisasi yang stabil [15].

2.3.2 Improved K-Medoids Berbasis Crow Search Algorithm (CSA)

Metode *improved* K-Medoids berbasis *Crow Search Algorithm* (CSA) digunakan dalam penelitian ini sebagai pengembangan dari algoritma K-Medoids untuk mengoptimalkan proses pemilihan kandidat *medoid* dalam klasterisasi pola pembelian konsumen [9]. Pendekatan ini memanfaatkan mekanisme pencarian berbasis populasi untuk memperoleh solusi *medoid* yang lebih baik dan mengurangi risiko terjebak pada *local optimum*. Pada metode ini, setiap solusi dalam CSA merepresentasikan kombinasi *medoid* yang akan dievaluasi menggunakan fungsi objektif K-Medoids berdasarkan total jarak antar data dengan *medoid* terdekatnya.

Crow Search Algorithm (CSA) merupakan algoritma optimasi metaheuristik yang terinspirasi dari perilaku alami burung gagak dalam menyimpan dan melindungi makanan. Dalam CSA, setiap burung gagak direpresentasikan sebagai satu solusi kandidat yang memiliki posisi saat ini dan memori terbaik yang pernah diperoleh selama proses pencarian solusi [9]. Proses pencarian dilakukan secara iteratif melalui mekanisme eksplorasi dan eksploitasi menggunakan parameter *Awareness Probability* (AP) dan *Flight Length* (fl). Secara matematis, proses pembaruan posisi solusi pada CSA dapat dirumuskan sebagai berikut [12]:

$$x_m^{Iter+1} = x_m^{Iter} + r \cdot fl \cdot (m_k^{Iter} - x_m^{Iter}) \quad (3)$$

dimana x_m^{Iter} menunjukkan posisi solusi ke- m pada iterasi ke- $Iter$, sedangkan m_k^{Iter} merupakan posisi memori terbaik dari solusi yang diikuti. Variabel r merupakan bilangan acak dalam rentang $[0,1]$, sedangkan fl merupakan parameter *flight length* yang mengontrol jarak perpindahan solusi dalam ruang pencarian.

Proses klasterisasi diawali dengan penentuan jumlah *cluster* (K) yang diuji pada beberapa variasi nilai. Selanjutnya, CSA digunakan untuk mengoptimasi kandidat *medoid* dengan membentuk sejumlah populasi solusi secara acak. Setiap solusi kemudian dievaluasi menggunakan fungsi *fitness* berdasarkan total jarak pada algoritma K-Medoids



sesuai dengan Persamaan 1. Tahapan CSA meliputi inialisasi populasi solusi, penyimpanan solusi terbaik dalam memori, serta pembaruan posisi solusi berdasarkan parameter *Awareness Probability* (AP) dan *Flight Length* (fl). Jika solusi yang dihasilkan memiliki nilai *fitness* yang lebih baik dibandingkan solusi sebelumnya, maka memori solusi akan diperbarui. Proses ini dilakukan secara iteratif hingga mencapai jumlah iterasi maksimum atau tidak terjadi peningkatan nilai *fitness* yang signifikan. Solusi terbaik akhir ditentukan menggunakan fungsi minimasi berikut [9]:

$$S^* = \arg \min_{S_c} E(S_c) \quad (4)$$

dimana S^* merupakan solusi terbaik akhir yang memiliki nilai *fitness* minimum. Solusi tersebut kemudian digunakan sebagai *medoid* optimal dalam proses pembentukan *cluster* akhir menggunakan algoritma K-Medoids. Dengan mekanisme tersebut, pendekatan *improved* K-Medoids berbasis CSA mampu menghasilkan proses pencarian *medoid* yang lebih efisien karena tidak hanya bergantung pada inialisasi awal. Penerapan CSA memungkinkan eksplorasi berbagai kombinasi *medoid* secara lebih luas, sehingga proses klasterisasi menjadi lebih stabil dan memiliki efisiensi waktu komputasi yang lebih baik dibandingkan K-Medoids tanpa optimasi.

2.4 Evaluasi Kinerja Klasterisasi

Evaluasi kinerja klasterisasi dalam penelitian ini dilakukan untuk membandingkan hasil algoritma K-Medoids dan *improved* K-Medoids berbasis *Crow Search Algorithm* (CSA) dalam mengelompokkan pola pembelian konsumen. Evaluasi difokuskan pada dua aspek utama, yaitu kualitas *cluster* dan efisiensi waktu komputasi. Kualitas hasil klasterisasi diukur menggunakan *Silhouette Score*, yang digunakan untuk mengevaluasi seberapa baik suatu data berada dalam *cluster* yang sesuai dibandingkan dengan *cluster* lainnya. Nilai *Silhouette Score* berada pada rentang -1 hingga 1, di mana nilai yang mendekati 1 menunjukkan kualitas hasil klasterisasi yang semakin baik [3]. Perhitungan *Silhouette Score* secara matematis dinyatakan pada Persamaan 5.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (5)$$

dimana $a(i)$ merupakan rata-rata jarak antara data ke- i dengan seluruh data dalam *cluster* yang sama, sedangkan $b(i)$ merupakan rata-rata jarak terkecil antara data ke- i dengan data pada *cluster* lain. Nilai *Silhouette Score* berada pada rentang -1 hingga 1, di mana nilai yang mendekati 1 menunjukkan bahwa data telah terkelompok dengan baik dalam *cluster* yang sesuai.

Selain kualitas *cluster*, penelitian ini juga mempertimbangkan aspek efisiensi melalui waktu komputasi. Waktu komputasi digunakan untuk mengukur lamanya proses algoritma dalam membentuk *cluster* mulai dari inialisasi hingga diperoleh hasil akhir. Semakin kecil waktu komputasi yang dibutuhkan, maka semakin efisien algoritma tersebut dalam memproses data. Hasil evaluasi kemudian digunakan untuk menganalisis kualitas hasil klasterisasi dan efisiensi proses komputasi, sehingga dapat diketahui metode yang lebih sesuai berdasarkan kebutuhan analisis.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini terdiri atas 18814 baris dan 13 atribut data transaksi, di mana setiap baris merepresentasikan satu aktivitas transaksi pelanggan. Informasi yang terdapat pada dataset meliputi tanggal transaksi, nomor transaksi, kode menu, nama menu, jumlah pembelian, harga satuan, total harga, serta beberapa atribut pendukung lainnya. Data transaksi tersebut diperoleh dari sistem *Point of Sale* (POS) pada salah satu gerai makanan cepat saji dan digunakan sebagai dasar dalam proses klasterisasi pola pembelian konsumen.

Tabel 1. Dataset

	tanggal	nomor	kode menu	menu master	...	toko	device
0	2025-01-01	1735685093	3	SAYAP	...	NaN	NaN
1	2025-01-01	1735685093	7827	SAYAP+NASI+MINUMAN	...	NaN	NaN
				ALLVARIANT			
2	2025-01-01	1735686274	5	PAHA ATAS	...	NaN	NaN
...	...	...	...	...	...	...	...
18812	2025-03-28	1743162791	7808	AYAM BEKAKAK	...	ABU	NaN
						SOFYAN	
18813	2025-03-28	1743162791	107818	BEEF CHEESE BURGER	...	ABU	NaN
						SOFYAN	

Berdasarkan Tabel 1, setiap transaksi memuat informasi yang berkaitan dengan aktivitas pembelian pelanggan. Banyaknya data transaksi yang tersedia memungkinkan proses pengelompokan dilakukan secara lebih representatif sehingga karakteristik pola pembelian konsumen dapat diidentifikasi dengan lebih baik. Dataset tersebut selanjutnya digunakan pada tahap pra-pemrosesan sebelum dilakukan proses klasterisasi.



3.2 Pra-pemrosesan Data

Sebelum dilakukan proses klusterisasi, terlebih dahulu dilakukan tahap pra-pemrosesan data untuk memastikan kualitas data yang digunakan. Tahap awal dilakukan dengan pemeriksaan *missing value* untuk mengetahui keberadaan data kosong yang dapat memengaruhi hasil analisis. Berdasarkan hasil pemeriksaan, seluruh atribut yang digunakan dalam penelitian telah terisi dengan lengkap sehingga tidak diperlukan proses penanganan *missing value*. Selanjutnya dilakukan seleksi atribut dengan memilih atribut nomor transaksi, menu_toko, qty, unit_price, dan total_price. Pemilihan atribut tersebut dilakukan karena atribut tersebut memiliki keterkaitan langsung dengan perilaku pembelian konsumen. Atribut yang tidak relevan terhadap tujuan penelitian tidak digunakan agar proses analisis menjadi lebih terfokus.

Tahap berikutnya adalah transformasi tipe data untuk memastikan atribut qty, unit_price, dan total_price berada dalam format numerik sehingga dapat diproses secara matematis. Setelah itu dilakukan agregasi data berdasarkan nomor transaksi untuk memperoleh representasi transaksi yang lebih ringkas. Proses agregasi menghasilkan dua variabel utama, yaitu total jumlah item yang dibeli dan total nilai transaksi pada setiap transaksi. Tahap terakhir adalah normalisasi menggunakan metode *Min-Max Scaling*. Normalisasi dilakukan karena kedua variabel yang digunakan memiliki rentang nilai yang berbeda. Dengan normalisasi, setiap variabel ditransformasikan ke dalam rentang nilai antara 0 hingga 1 sehingga tidak terjadi dominasi salah satu variabel dalam proses perhitungan jarak pada metode klusterisasi berbasis *Euclidean Distance*.

3.3 Proses Klusterisasi

3.3.1 K-Medoids

Proses klusterisasi menggunakan algoritma K-Medoids dilakukan dengan variasi jumlah *cluster* (K) dari 2 hingga 7. Pada setiap pengujian digunakan parameter *maximum iteration* sebesar 50 dan *random state* sebesar 42 untuk menjaga konsistensi hasil. Proses dimulai dengan pemilihan sejumlah objek secara acak sebagai *medoid* awal. Selanjutnya, jarak antara setiap data terhadap *medoid* dihitung menggunakan *Euclidean Distance*, kemudian setiap data dikelompokkan ke dalam *cluster* dengan *medoid* terdekat.

Setelah proses pengelompokan selesai, dilakukan perhitungan total *cost* untuk mengevaluasi kualitas *medoid* yang digunakan. Tahap berikutnya adalah melakukan pertukaran kandidat *medoid* dengan objek lainnya untuk memperoleh kombinasi *medoid* yang menghasilkan nilai *cost* lebih kecil. Proses tersebut dilakukan secara iteratif hingga tidak terdapat perubahan *medoid* atau jumlah iterasi maksimum telah tercapai.

Tabel 2. Jumlah *Cluster* Terbaik K-Medoids

K	4.000000
Silhouette Score	0.557791
Cost	14.663275
Waktu (detik)	111.630863

Berdasarkan Tabel 2, jumlah *cluster* terbaik diperoleh pada K = 4 dengan nilai *Silhouette Score* sebesar 0,557791. Nilai tersebut merupakan nilai tertinggi dibandingkan jumlah *cluster* lainnya sehingga dipilih sebagai jumlah *cluster* yang paling optimal. Selain itu, nilai *cost* sebesar 14,663275 menunjukkan bahwa total jarak antara data dan *medoid* relatif kecil sehingga *cluster* yang terbentuk mampu merepresentasikan anggota *cluster* dengan baik. Meskipun menghasilkan kualitas *cluster* yang baik, algoritma K-Medoids membutuhkan waktu komputasi sebesar 111,630863 detik. Hal tersebut disebabkan karena proses pencarian *medoid* dilakukan secara iteratif melalui evaluasi berbagai kemungkinan pertukaran objek hingga diperoleh kombinasi *medoid* terbaik.

3.3.2 Improved K-Medoids Berbasis Crow Search Algorithm (CSA)

Pada metode *improved* K-Medoids berbasis *Crow Search Algorithm* (CSA), pengujian dilakukan menggunakan jumlah *cluster* yang sama, yaitu K = 2 hingga K = 7. Parameter yang digunakan meliputi *population size* sebesar 20, *maximum iteration* sebesar 50, *awareness probability* (AP) sebesar 0,1, serta *flight length* (fl) sebesar 2. Proses diawali dengan pembentukan sejumlah solusi awal yang merepresentasikan kandidat *medoid*. Setiap solusi kemudian dievaluasi menggunakan fungsi *fitness* berdasarkan total jarak antara data dan *medoid*. Selanjutnya dilakukan pembaruan posisi solusi menggunakan mekanisme CSA hingga diperoleh solusi terbaik yang digunakan sebagai *medoid* akhir dalam pembentukan *cluster*.

Tabel 3. Jumlah *Cluster* Terbaik K-Medoids Berbasis CSA

K	4.000000
Silhouette Score	0.537240
Fitness	15.224862
Waktu (detik)	1.836520

Berdasarkan Tabel 3, jumlah *cluster* terbaik juga diperoleh pada K = 4 dengan nilai *Silhouette Score* sebesar 0,537240. Nilai tersebut menunjukkan bahwa *cluster* yang dihasilkan masih memiliki kualitas yang cukup baik meskipun sedikit lebih rendah dibandingkan algoritma K-Medoids. Selain itu, nilai *fitness* sebesar 15,224862 menunjukkan bahwa

solusi yang diperoleh melalui proses optimasi CSA mampu menghasilkan kombinasi *medoid* yang mendekati solusi optimal. Dari sisi efisiensi, metode ini hanya membutuhkan waktu komputasi sebesar 1,836520 detik sehingga menunjukkan bahwa penggunaan CSA mampu mempercepat proses pencarian *medoid* dibandingkan K-Medoids konvensional.

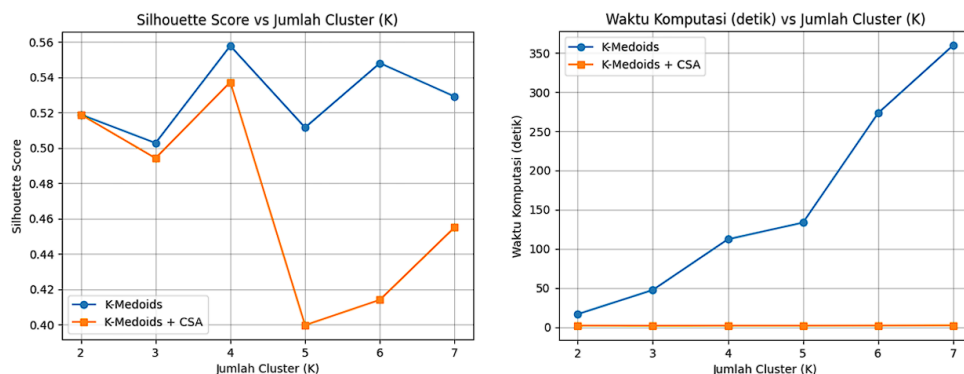
3.4 Evaluasi Kinerja Klusterisasi

Evaluasi kinerja dilakukan untuk membandingkan algoritma K-Medoids dan *improved* K-Medoids berbasis *Crow Search Algorithm* (CSA) berdasarkan kualitas *cluster* dan efisiensi waktu komputasi. Pengujian dilakukan menggunakan variasi jumlah *cluster* dari K = 2 hingga K = 7.

Tabel 4. Hasil Evaluasi Kinerja Klusterisasi

K	<i>Silhouette Score</i> K-Medoids	Waktu (detik) K-Medoids	<i>Silhouette Score Improved</i> K-Medoids berbasis CSA	Waktu (detik) <i>Improved</i> K-Medoids berbasis CSA
2	0.518869	15.897214	0.518869	1.764985
3	0.502705	47.847260	0.494112	1.750353
4	0.557791	111.630863	0.537240	1.836520
5	0.511474	129.506872	0.399568	1.938836
6	0.547979	250.072287	0.413924	1.970505
7	0.529053	340.949317	0.455007	2.120880

Berdasarkan Tabel 4, algoritma K-Medoids menghasilkan nilai *Silhouette Score* yang lebih tinggi dibandingkan *improved* K-Medoids berbasis *Crow Search Algorithm* (CSA) pada sebagian besar variasi jumlah *cluster*. Nilai *Silhouette Score* tertinggi pada algoritma K-Medoids diperoleh pada K = 4 dengan nilai sebesar 0,557791, sedangkan metode *improved* K-Medoids berbasis CSA memperoleh nilai *Silhouette Score* tertinggi sebesar 0,537240 yang juga diperoleh pada K = 4. Dari sisi kualitas *cluster*, hasil tersebut menunjukkan bahwa kedua metode menghasilkan jumlah *cluster* terbaik yang sama, yaitu K = 4. Akan tetapi, algoritma K-Medoids mampu menghasilkan *cluster* yang sedikit lebih baik dibandingkan metode *improved* K-Medoids berbasis CSA. Selain kualitas *cluster*, Tabel 4 juga menunjukkan adanya perbedaan yang cukup signifikan dari sisi waktu komputasi. Waktu komputasi algoritma K-Medoids meningkat seiring bertambahnya jumlah *cluster* dan mencapai 340,949317 detik pada K = 7. Sebaliknya, metode *improved* K-Medoids berbasis CSA memiliki waktu komputasi yang relatif stabil pada kisaran 1,75 hingga 2,12 detik.



Gambar 2. Visualisasi Perbandingan Kinerja Klusterisasi

Berdasarkan Gambar 2, terlihat bahwa algoritma K-Medoids cenderung menghasilkan nilai *Silhouette Score* yang lebih tinggi dibandingkan *improved* K-Medoids berbasis CSA. Namun demikian, metode *improved* K-Medoids berbasis CSA memiliki keunggulan yang signifikan dari sisi efisiensi waktu komputasi. Hasil penelitian ini menunjukkan adanya *trade-off* antara kualitas *cluster* dan efisiensi waktu komputasi. Algoritma K-Medoids memiliki keunggulan dalam menghasilkan kualitas *cluster* yang lebih baik, sedangkan *improved* K-Medoids berbasis CSA memiliki keunggulan dari sisi efisiensi proses. Hasil penelitian ini sejalan dengan penelitian yang dilakukan oleh Nitesh Sureja et al. [9], yang menunjukkan bahwa *Crow Search Algorithm* mampu meningkatkan efisiensi proses pencarian *medoid*. Akan tetapi, pada penelitian ini diperoleh bahwa peningkatan efisiensi waktu komputasi tidak selalu diikuti dengan peningkatan kualitas *cluster*. Dengan demikian, apabila kualitas *cluster* menjadi prioritas utama maka algoritma K-Medoids lebih sesuai digunakan. Sebaliknya, apabila efisiensi waktu komputasi menjadi pertimbangan utama, maka *improved* K-Medoids berbasis CSA dapat menjadi alternatif yang lebih baik.

3.5 Pembahasan

Hasil penelitian menunjukkan bahwa algoritma K-Medoids menghasilkan kualitas *cluster* yang lebih baik dibandingkan *improved* K-Medoids berbasis *Crow Search Algorithm* (CSA), yang ditunjukkan oleh nilai *Silhouette Score* tertinggi sebesar 0,557791. Di sisi lain, *improved* K-Medoids berbasis CSA mampu menghasilkan waktu komputasi yang jauh lebih singkat, yaitu sekitar 1,84 detik, dibandingkan K-Medoids yang membutuhkan waktu lebih dari 111 detik. Temuan



tersebut menunjukkan bahwa penerapan CSA memberikan peningkatan pada aspek efisiensi komputasi, meskipun kualitas *cluster* yang dihasilkan sedikit lebih rendah dibandingkan metode K-Medoids konvensional.

Hasil penelitian ini sejalan dengan penelitian yang dilakukan oleh Sureja et al. [9], yang mengembangkan *improved* K-Medoids berbasis CSA untuk meningkatkan proses pencarian *medoid*. Penelitian tersebut menunjukkan bahwa penggunaan CSA mampu meningkatkan efisiensi proses optimasi melalui mekanisme eksplorasi dan eksploitasi dalam ruang pencarian. Pada penelitian ini, peningkatan efisiensi tersebut juga terlihat dari waktu komputasi yang jauh lebih singkat dibandingkan algoritma K-Medoids. Namun, berbeda dengan penelitian Sureja et al. yang lebih menitikberatkan pada peningkatan proses optimasi, penelitian ini menunjukkan bahwa peningkatan efisiensi komputasi tidak selalu diikuti oleh peningkatan kualitas *cluster* berdasarkan nilai *Silhouette Score*.

Selain itu, temuan penelitian ini juga memiliki kesamaan dengan penelitian Pratama et al. [11] yang membandingkan algoritma K-Means, K-Medoids, dan DBSCAN pada proses klusterisasi kualitas hidup Indonesia. Penelitian tersebut menunjukkan bahwa setiap algoritma memiliki karakteristik performa yang berbeda bergantung pada karakteristik data dan indikator evaluasi yang digunakan. Hasil tersebut mendukung temuan penelitian ini bahwa tidak terdapat satu algoritma yang unggul pada seluruh aspek evaluasi. Pada penelitian ini, K-Medoids lebih unggul dalam menghasilkan kualitas *cluster*, sedangkan *improved* K-Medoids berbasis CSA lebih unggul dalam efisiensi waktu komputasi.

Temuan penelitian ini juga mendukung penelitian Yosia dan Siregar [8] yang menunjukkan bahwa algoritma K-Medoids mampu menghasilkan kualitas *cluster* yang baik pada data penjualan berdasarkan evaluasi menggunakan *Silhouette Score*. Perbedaannya, penelitian ini tidak hanya mengevaluasi kualitas *cluster*, tetapi juga membandingkan efisiensi waktu komputasi melalui penerapan CSA sebagai metode optimasi. Dengan demikian, penelitian ini memberikan sudut pandang yang lebih luas mengenai *trade-off* antara kualitas hasil klusterisasi dan efisiensi proses komputasi.

Berdasarkan hasil perbandingan tersebut, dapat diketahui bahwa pemilihan metode klusterisasi perlu disesuaikan dengan kebutuhan analisis. Apabila tujuan utama adalah memperoleh kualitas *cluster* yang optimal, maka algoritma K-Medoids merupakan pilihan yang lebih tepat. Sebaliknya, apabila analisis dilakukan pada data berukuran besar atau membutuhkan waktu komputasi yang cepat, maka *improved* K-Medoids berbasis CSA dapat menjadi alternatif yang lebih sesuai karena mampu mempertahankan kualitas *cluster* yang baik dengan efisiensi komputasi yang jauh lebih tinggi.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma K-Medoids dan *improved* K-Medoids berbasis *Crow Search Algorithm* (CSA) mampu digunakan untuk mengelompokkan data transaksi penjualan dan menghasilkan jumlah *cluster* terbaik yang sama, yaitu sebanyak empat *cluster*. Hasil evaluasi menunjukkan bahwa algoritma K-Medoids menghasilkan kualitas *cluster* yang lebih baik berdasarkan nilai *Silhouette Score*, sehingga mampu membentuk kelompok data yang memiliki tingkat kemiripan yang lebih tinggi di dalam *cluster* serta pemisahan yang lebih jelas antar *cluster*. Di sisi lain, *improved* K-Medoids berbasis CSA menunjukkan keunggulan dari aspek efisiensi waktu komputasi, di mana proses pembentukan *cluster* dapat dilakukan dalam waktu yang jauh lebih singkat dan relatif stabil pada berbagai variasi jumlah *cluster*. Temuan tersebut menunjukkan bahwa penerapan CSA mampu meningkatkan efisiensi proses pencarian *medoid*, meskipun peningkatan tersebut tidak selalu diikuti oleh peningkatan kualitas *cluster* yang dihasilkan. Dengan demikian, pemilihan metode klusterisasi dapat disesuaikan dengan kebutuhan analisis yang dilakukan. Apabila kualitas hasil pengelompokan menjadi prioritas utama, maka algoritma K-Medoids lebih sesuai digunakan. Sebaliknya, apabila efisiensi waktu komputasi menjadi pertimbangan utama, maka *improved* K-Medoids berbasis CSA dapat menjadi alternatif yang lebih tepat. Kontribusi penelitian ini terletak pada penyediaan analisis komparatif yang menunjukkan bahwa penerapan CSA mampu meningkatkan efisiensi waktu komputasi secara signifikan, sementara algoritma K-Medoids tetap menghasilkan kualitas *cluster* yang lebih baik berdasarkan nilai *Silhouette Score*. Hasil penelitian ini dapat menjadi referensi bagi peneliti maupun praktisi dalam menentukan metode klusterisasi yang sesuai dengan karakteristik data dan tujuan analisis, serta menjadi dasar bagi penelitian selanjutnya untuk mengembangkan metode optimasi lain pada proses klusterisasi data transaksi penjualan.

REFERENCES

- [1] S. Muharni and S. Andriyanto, "Penentuan Pola Penjualan Menggunakan Algoritma Apriori," *Digit. Transform. Technol.*, vol. 4, no. 1, pp. 60–71, Mar. 2024, doi: 10.47709/digitech.v4i1.3679.
- [2] M. Marindah and K. Setiawan, "Implementasi Data Mining Menggunakan Algoritma Apriori Terhadap Pola Pembelian Konsumen di Marketplace Shopee Jaktimstore," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 7, no. 5, pp. 1473–1481, Sep. 2024, doi: 10.31539/intecomsv7i5.11644.
- [3] R. Hidayati, A. Zubair, A. H. Pratama, and L. Indana, "Analisis Silhouette Coefficient pada 6 Perhitungan Jarak K-Means Clustering," *Techno.Com*, vol. 20, no. 2, pp. 186–197, May 2021, doi: 10.33633/tc.v20i2.4556.
- [4] S. I. Murpratiwi, I. G. Agung Indrawan, and A. Aranta, "Analisis Pemilihan Cluster Optimal dalam Segmentasi Pelanggan Toko Retail," *J. Pendidik. Teknol. Dan Kejuru.*, vol. 18, no. 2, p. 152, Sep. 2021, doi: 10.23887/jptk-undiksha.v18i2.37426.
- [5] D. Cheng, Y. Zhang, and R. Jin, "Careful Seeding for K-Medoids Clustering with Incremental K-Means++ Initialization," *arXiv*, 2024, doi: 10.48550/arXiv.2207.02404.



- [6] A. N. Fitriani, K. M. Hindrayani, and T. Trimono, "Enhanced Clustering Using PSO-KMedoids for Government Aid Distribution," *Jurnal Sistem Informasi dan Informatika (Simika)*, vol. 8, no. 2, pp. 294–305, Aug. 2025, doi: 10.47080/simika.v8i2.3930.
- [7] S. M. Sabrina and T. H. Setiawan, "Clustering Analysis of Provinces in Indonesia Based on the 2023 Human Development Index Indicators Using the K-Medoids Algorithm," *Jurnal Matematika UNAND*, vol. 14, no. 1, pp. 93–102, Jan. 2025, doi: 10.25077/jmua.14.1.93-102.2025.
- [8] Yosia and B. Siregar, "Comparative Analysis of K-Means and K-Medoids Algorithms for Product Sales Clustering and Customer," *Journal of Mathematics and Computer Statistics*, vol. 7, no. 2, pp. 360–370, Oct. 2024, doi: 10.35580/jmathcos.v7i2.4053.
- [9] N. Sureja, B. Chawda, and A. Vasant, "An Improved K-Medoids Clustering Approach Based on the Crow Search Algorithm," *Journal of Computational Mathematics and Data Science*, vol. 3, p. 100034, Jun. 2022, doi: 10.1016/j.jcmds.2022.100034.
- [10] E. Luthfi and A. W. Wijayanto, "Analisis Perbandingan Metode Hierarchical, K-Means, dan K-Medoids Clustering dalam Pengelompokan Indeks Pembangunan Manusia Indonesia," *INOVASI*, vol. 17, no. 4, pp. 761–773, Dec. 2021, doi: 10.30872/jinv.v17i4.10106.
- [11] M. R. P. Pratama, M. I. Fieldi, M. S. Albani, M. Al Fachrozi, F. R. Aderiyana, K. D. Tania, and A. Meiriza, "Perbandingan Algoritma K-Means, K-Medoid, dan DBSCAN untuk Clustering Kualitas Hidup Indonesia dalam Perspektif Knowledge Management dan Data Discovery," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 4, pp. 5903–5910, Aug. 2025, doi: 10.36040/jati.v9i4.13916.
- [12] R. Ranjan and J. Chhabra, "Automatic Data Clustering Using Dynamic Crow Search Algorithm," *EAI Endorsed Transactions on Context-Aware Systems and Applications*, p. 173982, May 2022, doi: 10.4108/eai.17-5-2022.173982.
- [13] A. K. Singh and S. Gupta, "Comparative Study of Feature Scaling Techniques for Distance-Based Machine Learning Algorithms," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 4, pp. 1060–1068, 2023.
- [14] G. Erda, R. K. Usdika, R. Pitri, and Z. Erda, "Implementation of the K-Medoids Method in Clustering Human Development Indexes in Indonesia," *Parameter: Journal of Statistics*, vol. 3, no. 2, pp. 58–67, 2023, doi: 10.22487/27765660.2023.v3.i2.16906.
- [15] L. Nigro and P. Fränti, "Two Medoid-Based Algorithms for Clustering Sets," *Algorithms*, vol. 16, no. 7, p. 349, Jul. 2023, doi: 10.3390/a16070349.