

BAB I

PENDAHULUAN

1.1 Latar Belakang

Analisis data berdimensi tinggi merupakan pusat perhatian dalam pengolahan data modern. Semakin banyak data yang tersedia, metode-metode konvensional sering kali menemui tantangan dalam menghasilkan solusi yang efektif dan efisien. Masalah mendasar ini muncul karena keterbatasan metode untuk mengekstraksi pola-pola yang kompleks dalam data berdimensi tinggi. Padahal, pola-pola ini sering kali mencakup informasi penting yang dapat mempengaruhi hasil secara signifikan. Oleh sebab itu, perkembangan data harus seirama dengan perkembangan metode-metode ilmu komputer dan kecerdasan buatan.

Machine learning telah menjadi bidang kecerdasan buatan yang berkembang pesat saat ini. *Machine learning* merupakan metode yang memungkinkan sistem belajar dari data. Terdapat dua jenis metode dalam *machine learning*, yaitu *supervised learning* dan *unsupervised learning*. *Supervised learning* merupakan pemodelan data berdasarkan label atau klasifikasi yang sudah ada. Algoritma akan berusaha untuk membuat model berdasarkan *output* yang sudah diketahui, yang kemudian akan diuji ke data yang lain. Sedangkan *unsupervised learning* merupakan pemodelan data tanpa adanya acuan atau label awal. Metode ini akan belajar secara bebas untuk menemukan pola atau struktur pada data [1].

Clustering merupakan salah satu teknik dalam *unsupervised learning* yang paling sering digunakan dalam pengolahan data. *Clustering* mengelompokkan data yang memiliki karakteristik yang sama ke dalam satu klaster, kemudian data dengan karakteristik yang lain akan dikelompokkan pada klaster lainnya [2]. Terdapat beberapa jenis *clustering*, yaitu *hierarchical clustering* yang membangun struktur pohon dari kelompok-kelompok data [3], *density-based clustering* yang mengidentifikasi kelompok berdasarkan kepadatan titik-titik data [4], dan *center-based clustering* yang membagi data berdasarkan kedekatan dengan titik-titik pusat [5]. *Center-based clustering* merupakan teknik yang paling sering dijumpai dalam analisis data multidimensi dan menjadi metode yang akan dikaji pada penelitian ini.

K-Means merupakan metode *center-based clustering* yang paling sering digunakan. Kelebihan utama *K-Means* terletak pada kesederhanaan implementasi, efisiensi komputasi, dan kemampuan beradaptasi dengan berbagai jenis data [6][7]. Proses kerja *K-Means* dimulai dengan memilih k titik awal sebagai *initial centroid* dari titik data yang ada. Pada setiap iterasi, jarak setiap titik data ke *centroid* dihitung menggunakan *Euclidean Distance*. Pengelompokan *cluster* setiap titik data akan ditentukan berdasarkan jarak terdekat ke *centroid* [8]. Meskipun sederhana, *K-Means* memiliki kelemahan yaitu sangat sensitif pada penentuan *initial centroid*. Penentuan *initial centroid* yang dilakukan secara acak dapat menghasilkan hasil *clustering* yang tidak optimal dan tidak konsisten [9][10]. Penelitian [11] menjelaskan hal yang sama, penentuan *initial centroid* secara acak menghasilkan konvergensi pada kondisi *local optima*. Hal ini tentu berdampak pada jumlah iterasi dan waktu komputasi yang dibutuhkan. Sehingga perlu dilakukan optimasi pada *K-Means* untuk menghasilkan pengelompokan yang optimal.

Penggunaan teknik optimasi merupakan pendekatan umum dalam mengatasi kelemahan suatu algoritma. Pada algoritma *K-Means*, teknik optimasi sering digunakan untuk mencari *initial centroid* yang optimal, sehingga dapat meningkatkan kualitas dan konsistensi hasil *clustering* [11-22]. Integrasi teknik optimasi ke dalam proses *clustering K-Means* dianggap mampu mengurangi ketergantungan pada penentuan *initial centroid*, mempercepat konvergensi algoritma, dan menghasilkan partisi data yang lebih optimal. Penelitian [16] menjelaskan bahwa penggunaan algoritma *Pillar* dalam menginisialisasi *centroid* pada *K-Means* meningkatkan hasil *clustering*. Hasil tersebut dilihat dari nilai evaluasi DBI dari 0,524765 berkurang menjadi 0,524760 dan jumlah iterasi dari 23 berkurang menjadi 17. Penggunaan metode optimasi juga secara signifikan meningkatkan kualitas *clustering*, terutama untuk dataset yang kompleks atau berdimensi tinggi [15][18].

Teknik optimasi merupakan proses mencari solusi optimal dari suatu masalah. Solusi tersebut diperoleh dengan meminimalkan atau memaksimalkan suatu fungsi objektif. Terdapat dua metode optimasi, yaitu *deterministic* dan *metaheuristic* [19]. Metode *deterministic* menggunakan pendekatan yang terstruktur dan matematis untuk masalah optimasi yang sederhana atau terbatas. Sedangkan metode

metaheuristic menggunakan fenomena alam atau perilaku biologis untuk masalah optimasi yang kompleks. Metode ini sangat efektif untuk masalah optimasi global, di mana ruang pencarian memiliki banyak optimum lokal. Metode *metaheuristic* yang sering digunakan adalah *Particle Swarm Optimization* (PSO) [12-15].

Particle Swarm Optimization (PSO) merupakan teknik optimasi yang menggunakan fenomena kawanan burung atau ikan. Kemampuan algoritma PSO dalam menemukan solusi yang bersifat global didasarkan pada kecerdasan kognitif yang dipengaruhi oleh kecerdasan kolektif kawanan [23]. Pada metode PSO, ketika satu partikel menemukan jalur optimal menuju solusi akhir, partikel lainnya akan mengikuti jalur tersebut, meskipun pada awalnya mereka berada pada lokasi yang berbeda. Penelitian [22] menjelaskan penggunaan algoritma PSO dalam menginisialisasi *centroid* pada algoritma *K-Medoids* mampu memberikan pusat *initial centroid* secara optimal dan hasil dari *clustering* menjadi lebih baik. Hasil pengujian menunjukkan secara umum dengan percobaan empat *cluster*, *PSO+K-Medoids* lebih unggul dibandingkan *K-Medoids* murni. Optimasi mampu meningkatkan nilai *Davies-Bouldin Index* (DBI) dari 0,283 menjadi 0,088.

Keandalan metode *Particle Swarm Optimization* (PSO) untuk menentukan solusi yang optimal menjadi alasan penelitian ini untuk menguji pengaruh dari metode PSO dalam menentukan *initial centroid* terhadap kualitas hasil *clustering* algoritma *K-Means*. Di samping itu, PSO ternyata menunjukkan sensitivitas terhadap pemilihan parameter, terutama *inertia weight*, yang dapat mempengaruhi keseimbangan antara eksplorasi global dan eksploitasi lokal dalam proses pencarian solusi [13, 24-27]. Ketika partikel bergerak terlalu cepat atau lambat dalam ruang solusi maka akan mengurangi efektivitas dari algoritma PSO. Ada beberapa penelitian yang telah mengoptimasi nilai *inertia weight* yang awalnya statis menjadi dinamis, diantaranya penggunaan *Exponential Decay Inertia Weight* (EPSO) [24], *Inertia Weight Contraction Factor* (IWCF-PSO) [13][25], dan *Linearly Decreasing Weight* (LDW-PSO) [26] [27]. Penelitian-penelitian tersebut memungkinkan *inertia weight* berubah secara dinamis selama iterasi. Namun, modifikasi-modifikasi tersebut sering kali menghasilkan perubahan *inertia weight* yang terlalu tajam dan inkonsisten. Hal tersebut dapat menyebabkan ketidakstabilan dalam perilaku partikel. Oleh karena ini, pada penelitian ini akan menggunakan salah satu fungsi

invers trigonometri yang lebih stabil sehingga dapat menjaga keseimbangan antara eksplorasi dan eksploitasi dalam pencarian solusi. Metode ini akan disebut *Improved Particle Swarm Optimization* (IPSO) pada penelitian ini. Metode IPSO akan diuji dengan 15 *benchmark function*. Kemudian hasil numerik metode IPSO akan dikomparasi dengan metode EPSO, IWCF-PSO dan LDW-PSO .

Pengembangan metode *Improved Particle Swarm Optimization* (IPSO) kemudian diaplikasikan untuk menentukan *initial centroid* optimal dari algoritma *K-Means*. Metode *IPSO-K-Means* diintegrasikan dengan data yang terdiri dari dua tipe, yaitu sepuluh data *n*-dimensi hasil *generate* secara acak dan lima data dari UCI Machine Learning Repository. Untuk mengevaluasi kinerja algoritma yang diajukan, algoritma *IPSO-K-Means* akan dibandingkan dengan algoritma *Regular K-Means* menggunakan metrik evaluasi *Silhouette Score*, *Davies-Bouldin Index* (DBI), dan *running time*. Dengan memodifikasi *inertia weight* pada algoritma *Particle Swarm Optimization* (PSO) dengan fungsi invers trigonometri, penelitian ini diharapkan dapat menghasilkan algoritma optimasi yang lebih baik dan mampu mengatasi kelemahan *K-Means* dalam menentukan *initial centroid*.

1.2 Rumusan Masalah

Efektivitas algoritma *Particle Swarm Optimization* (PSO) sangat dipengaruhi oleh parameter *inertia weight* dalam menentukan kemampuan partikel menjelajahi ruang solusi secara global dan mengeksplorasi potensi solusi secara lokal. *Inertia weight* pada dasarnya bernilai konstan, tetapi perkembangan algoritma PSO menunjukkan bahwa *inertia weight* yang berubah secara dinamis memberikan hasil yang lebih baik. Salah satu modifikasi *inertia weight* adalah *Exponential Decay Inertia Weight* (EDIW). Namun, fungsi EDIW juga menghadapi risiko *overflow* pada iterasi tinggi atau dimensi yang besar. Untuk mengatasi masalah tersebut, penelitian ini memodifikasi *inertia weight* dengan fungsi yang memiliki sifat yang sama namun lebih stabil, yaitu invers trigonometri. Pengembangan modifikasi ini tidak hanya bertujuan meningkatkan performa PSO, tetapi juga diaplikasikan untuk mengatasi kelemahan algoritma *K-Means* dalam menentukan *initial centroid*. Pada dasarnya *K-Means* menentukan *initial centroid* secara acak, namun menyebabkan hasil *clustering* yang inkonsisten dan tidak optimal.

1.3 Pertanyaan Penelitian

1. Bagaimana kinerja modifikasi *inertia weight* pada algoritma *Particle Swarm Optimization* (PSO) dalam meningkatkan hasil komputasi?
2. Bagaimana perbandingan hasil komputasi algoritma *Particle Swarm Optimization* dengan *Arcus Tangent Inertia Weight*, *Exponential Decay Inertia weight*, *Inertia Weight Contraction Factor*, dan *Linearly Decreasing Weight*?
3. Bagaimana kinerja algoritma *Improved Particle Swarm Optimization* (IPSO) dalam mengoptimasi algoritma *K-Means* untuk menentukan *initial centroid*?
4. Bagaimana perbandingan kualitas *clustering* antara algoritma *K-Means* dengan *initial centroid* yang ditentukan menggunakan *Improved Particle Swarm Optimization* (IPSO) dan *K-Means* dengan *initial centroid* secara acak?

1.4 Batasan Masalah

1. *Benchmark function* yang digunakan sebagai fungsi uji metode IPSO adalah *Six-humpback Camel Function*, *Two-Dimensional Rastrigin Function*, *Two-Dimensional Function*, *Treccani Function*, *Goldstein and Price Function*, *Shekel's Function*, *Branin Function*, *N-Dimensional Sine-Square Function*, *Modified Griewank Function*, *Ackley Function*, *Random Test Function*, *Both Function*, *Beale Function*, dan *Mc Cormick Function*.
2. Data yang digunakan terdiri dari sepuluh data hasil *generate* secara acak, dan data Mice Protein Expression, Facebook Live Seller, Gas Turbine Emission 2011, Gas Turbine Emission 2012, dan Gas Turbine Emission 2013 yang diambil dari UCI Machine Learning Repository.
3. Jumlah *cluster* ditentukan di awal penelitian.

1.5 Tujuan Penelitian

1. Mengajukan fungsi *inertia weight* baru untuk meningkatkan kinerja algoritma *Particle Swarm Optimization* (PSO).
2. Melakukan komparasi performa komputasi fungsi *inertia weight* baru (IPSO) dengan *Exponential Decay Inertia Weight* (EPSO), *Inertia Weight*

Contraction Factor (IWCF-PSO), dan *Linearly Decreasing Weight* (LDW-PSO).

3. Mengaplikasikan algoritma *Improved Particle Swarm Optimization* (IPSO) untuk menentukan *initial centroid* pada algoritma *K-Means*.
4. Melakukan perbandingan hasil *clustering* antara algoritma *K-Means* murni dengan kombinasi *K-Means* dan *Improved Particle Swarm Optimization* (IPSO).

1.6 Manfaat Penelitian

1. Mendapatkan hasil komputasi yang lebih baik pada algoritma *Particle Swarm Optimization* (PSO) dengan memanfaatkan fungsi *inertia weight* baru.
2. Mendapatkan hasil *clustering* yang optimal pada algoritma *K-Means* dengan memanfaatkan *initial centroid* dari proses algoritma *Improved Particle Swarm Optimization* (IPSO).
3. Menjadi referensi untuk penelitian selanjutnya untuk modifikasi *inertia weight* pada algoritma PSO dan penentuan *initial centroid* pada algoritma *K-Means*.