E-ISSN : 2988-1986
https://ejournal.cahayailmubangsa.institute/index.php/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 10 No 2 Tahun 2025

PERBANDINGAN TEORITIS DAN EKSPERIMEN ALGORITMA K-MEANS DAN K-MEDOIDS DALAM KLASTERISASI DATA

Arya Pratama Putra[1*], Jihan Tshivana[2], Elkin Rilvani[3]

[1,2,3]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Pelita Bangsa, Cikarang.

E-mail: arlodut2706@gmail.com, Jihantshivana@gmail.com, elkin.rilvani@pelitabangsa.ac.id

ABSTRACT

This study presents a comparative analysis of two popular clustering algorithms, K-Means and K-Medoids, focusing on clustering quality, computational efficiency, and robustness to outliers. Using the Wine Dataset, which contains various chemical properties of wines, we evaluated the performance of both algorithms with multiple evaluation metrics, including Silhouette Score, Calinski-Harabasz Score, Davies-Bouldin Score, and processing time. The results indicate that K-Means outperforms K-Medoids in computational efficiency, with faster execution times and higher Silhouette and Calinski-Harabasz scores. However, K-Medoids demonstrates greater robustness to outliers and noise, producing more stable clustering results. This study suggests that K-Means is more suitable for large, relatively clean datasets, while K-Medoids is recommended for datasets with significant noise or outliers. The findings provide valuable insights for selecting the optimal clustering algorithm based on data characteristics and application requirements.

Keywords:K-Means, K-Medoids, Clustering, Outlier, Computational Efficiency

ABSTRAK

Penelitian ini menyajikan analisis perbandingan antara dua algoritma clustering yang populer, K-Means dan K-Medoids, dengan fokus pada kualitas clustering, efisiensi komputasi, dan ketahanan

Article History
Received: Agustus 2025
Reviewed: Agustus 2025
Published: Agustus 2025

Plagiarism Checker No 235
Prefix DOI :
10.8734/Kohesi.v1i2.365

Copyright : Author
Publish by : Kohesi

terhadap outlier. Menggunakan Wine Dataset, yang berisi berbagai sifat kimia dari anggur, kami mengevaluasi kinerja kedua algoritma dengan menggunakan beberapa metrik evaluasi, termasuk Silhouette Score, Calinski-Harabasz Score, Davies-Bouldin Score, dan waktu pemrosesan. Hasil penelitian menunjukkan bahwa K-Means unggul dalam hal efisiensi komputasi, dengan waktu eksekusi yang lebih cepat dan skor Silhouette serta Calinski-Harabasz yang lebih tinggi dibandingkan K-Medoids. Namun, K-Medoids menunjukkan ketahanan yang lebih baik terhadap outlier dan noise, menghasilkan hasil clustering yang lebih stabil. Penelitian ini menyarankan bahwa K-Means lebih cocok digunakan untuk dataset besar yang relatif bersih, sementara K-Medoids lebih disarankan untuk dataset yang mengandung banyak noise atau outlier. Penelitian ini memberikan wawasan yang berharga dalam memilih algoritma clustering yang optimal berdasarkan karakteristik data dan kebutuhan aplikasi.

Kata Kunci: K-Means, K-Medoids, Clustering, Outlier, Efisiensi Komputasi

1. PENDAHULUAN

Clustering merupakan salah satu teknik dalam pembelajaran tanpa pengawasan (unsupervised learning) yang sangat penting dalam analisis data eksploratori. Tujuan utama dari teknik ini adalah mengelompokkan data ke dalam sejumlah grup atau klaster berdasarkan kesamaan karakteristik antar objek. Teknik clustering sangat bermanfaat dalam berbagai bidang seperti bioinformatika, pemasaran, pengolahan citra, dan deteksi anomali. Salah satu tantangan utama dalam clustering adalah memilih algoritma yang tepat sesuai dengan karakteristik data serta tujuan analisis. Oleh karena itu, pemahaman terhadap berbagai metode clustering sangat penting dalam konteks penelitian maupun implementasi praktis.

Algoritma K-Means merupakan salah satu metode clustering yang paling populer dan banyak digunakan karena kesederhanaan serta efisiensinya dalam menangani dataset berukuran besar. Algoritma ini bekerja dengan menentukan centroid dari setiap klaster berdasarkan rata-rata posisi data dalam klaster tersebut. Meskipun cepat dan relatif mudah diimplementasikan, K-Means sangat sensitif terhadap data outlier dan tidak mampu menangani klaster yang

berbentuk non-sferikal dengan baik [1]. Ketergantungan terhadap rata-rata juga membuat hasil clustering dapat berubah signifikan ketika data mengandung noise atau nilai ekstrim.

Algoritma K-Medoids hadir sebagai alternatif dari K-Means dengan menawarkan pendekatan yang lebih robust terhadap outlier dan noise. Berbeda dari K-Means yang menggunakan centroid matematis, K-Medoids menggunakan objek data yang sebenarnya sebagai pusat klaster (medoid). Pendekatan ini membuat K-Medoids lebih stabil dalam menghadapi data yang tidak berdistribusi normal atau memiliki skala yang berbeda antar fitur [2]. Namun, metode ini cenderung lebih lambat secara komputasi karena proses pemilihan medoid yang membutuhkan perhitungan jarak antar semua pasangan data.

Penelitian yang membandingkan performa K-Means dan K-Medoids telah dilakukan dalam berbagai domain. Studi sebelumnya menunjukkan bahwa pemilihan algoritma clustering sangat tergantung pada distribusi data, keberadaan outlier, dan kebutuhan efisiensi komputasi [3]. Dalam aplikasi pengelompokan dokumen, K-Medoids mampu memberikan hasil yang lebih akurat dalam hal interpretasi semantik, sementara K-Means unggul dalam waktu pemrosesan [4]. Oleh karena itu, evaluasi sistematis terhadap kedua algoritma perlu dilakukan dalam berbagai kondisi data yang dikendalikan.

Penggunaan metrik evaluasi clustering menjadi penting dalam menilai kualitas hasil klasterisasi secara objektif. Beberapa metrik populer yang digunakan antara lain Silhouette Score, Calinski-Harabasz Index, dan Davies-Bouldin Index. Silhouette Score mengukur seberapa mirip suatu objek dengan klaster miliknya dibandingkan dengan klaster lain, sedangkan Calinski-Harabasz dan Davies-Bouldin mengukur kompaksi dan pemisahan antar klaster [5]. Ketiga metrik ini dapat memberikan gambaran menyeluruh mengenai kinerja algoritma clustering pada sebuah dataset.

Dataset sintetik sering digunakan dalam eksperimen algoritma clustering karena memiliki struktur dan distribusi yang dapat dikontrol dengan baik. Dengan data sintetik, peneliti dapat memastikan bahwa faktor-faktor seperti jumlah klaster, jarak antar klaster, dan kehadiran noise dapat dikendalikan untuk menguji performa algoritma secara adil [6]. Oleh karena itu, eksperimen ini menggunakan dataset sintetik sebagai dasar untuk melakukan perbandingan terstruktur antara algoritma K-Means dan K-Medoids.

Tujuan utama dari penelitian ini adalah untuk melakukan analisis komparatif antara algoritma K-Means dan K-Medoids dalam konteks clustering data sintetik. Evaluasi dilakukan dengan mempertimbangkan tiga aspek utama, yaitu kualitas hasil klasterisasi yang diukur menggunakan metrik evaluasi clustering, efisiensi waktu eksekusi algoritma, dan ketahanan

terhadap outlier. Dengan demikian, hasil penelitian ini diharapkan dapat memberikan panduan praktis dalam memilih algoritma clustering yang sesuai untuk berbagai kondisi data.

Kontribusi penelitian ini terletak pada penyediaan perbandingan empiris berbasis eksperimen terkontrol yang didukung oleh visualisasi hasil dan metrik evaluasi numerik. Penelitian ini juga menyajikan perspektif teoretis terhadap kekuatan dan kelemahan masing-masing algoritma, serta memberikan implikasi praktis dalam penggunaan algoritma clustering pada data nyata. Hasil dari eksperimen ini diharapkan dapat memperkaya literatur terkait dan menjadi referensi bagi peneliti maupun praktisi dalam memilih metode clustering yang optimal.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan unsupervised learning untuk membandingkan performa algoritma K-Means dan K-Medoids dalam melakukan proses clustering terhadap data tanpa label. Keduanya merupakan algoritma populer dalam pengelompokan data yang secara umum bertujuan untuk meminimalkan jarak intra-klaster dan memaksimalkan jarak antar-klaster. K-Means dikenal karena efisiensinya dalam skala besar, sedangkan K-Medoids dipilih karena ketahanannya terhadap keberadaan outlier. Penelitian ini dilaksanakan secara eksperimental menggunakan data sintetik dua dimensi yang dibangkitkan dengan distribusi normal multivariat. Tahapan dalam penelitian ini dijelaskan sebagai berikut:

1. **Pengumpulan dan Pembuatan Data**

   Data yang digunakan dalam penelitian ini bersifat sintetik dan dibangkitkan secara programatik dengan menggunakan pustaka make_blobs dari Scikit-learn. Dataset terdiri atas 300 titik data dua dimensi yang dikelompokkan menjadi tiga klaster. Setiap klaster memiliki pusat distribusi (centroid) yang berbeda, serta variasi (standard deviation) yang sama agar tidak terjadi bias dalam pemisahan antar-klaster. Pembuatan data sintetik seperti ini sering digunakan dalam penelitian komputasi karena memungkinkan kendali penuh atas parameter distribusi data [1].

2. **Pra-Pemrosesan Data**

   Data yang telah dibangkitkan kemudian dinormalisasi menggunakan teknik standardisasi. Proses ini dilakukan dengan StandardScaler dari pustaka Scikit-learn agar seluruh fitur memiliki distribusi dengan rata-rata nol dan standar deviasi satu. Normalisasi ini sangat penting dalam clustering berbasis jarak seperti K-Means maupun K-Medoids karena jarak Euclidean yang digunakan sangat dipengaruhi oleh skala fitur [2]. Tanpa proses ini, fitur dengan rentang nilai yang lebih besar dapat mendominasi proses pengelompokan.

3. **Implementasi Algoritma K-Means**

Algoritma K-Means diterapkan dengan nilai K = 3 sesuai jumlah klaster pada data sintetik. Proses diawali dengan inisialisasi centroid secara acak, dilanjutkan dengan iterasi pengelompokan data berdasarkan jarak ke centroid terdekat, serta pembaruan centroid berdasarkan rata-rata posisi anggota dalam tiap klaster. Proses ini berlangsung secara iteratif hingga posisi centroid tidak lagi berubah secara signifikan atau telah mencapai batas maksimum iterasi. Implementasi algoritma menggunakan pustaka Scikit-learn dengan parameter default dan pengaturan random seed agar hasil replikasi konsisten [3].

4. **Implementasi Algoritma K-Medoids**

Sebagai pembanding, algoritma K-Medoids juga diterapkan dengan nilai K = 3. Tidak seperti K-Means, algoritma ini memilih medoid (contoh nyata dari data) sebagai pusat klaster. Proses dilakukan dengan pendekatan Partitioning Around Medoids (PAM) menggunakan pustaka scikit-learn-extra. Algoritma akan menentukan konfigurasi medoid yang meminimalkan total jarak antara titik data dan medoid yang mewakilinya dalam setiap klaster. Karena menggunakan data aktual sebagai pusat, algoritma ini lebih tahan terhadap outlier namun lebih mahal secara komputasional [4].

5. **Visualisasi Hasil Clustering**

Hasil dari kedua algoritma divisualisasikan menggunakan diagram sebar dua dimensi. Visualisasi ini dilakukan agar pemisahan klaster dapat diamati secara langsung oleh peneliti dan pembaca. Titik-titik data diberi warna berdasarkan label klaster yang dihasilkan oleh masing-masing algoritma. Selain itu, pusat klaster juga divisualisasikan untuk mempermudah interpretasi posisi relatif antar klaster. Proses visualisasi ini sangat membantu dalam interpretasi hasil dan penyampaian informasi secara intuitif [5].

6. **Evaluasi Kualitas Clustering**

Kualitas dari hasil pengelompokan dinilai menggunakan tiga metrik evaluasi utama, yaitu Silhouette Score, Calinski-Harabasz Index, dan Davies-Bouldin Score. Metrik Silhouette mengukur konsistensi internal suatu klaster, sedangkan Calinski-Harabasz mengukur rasio dispersi antar dan intra klaster. Sementara itu, Davies-Bouldin Score digunakan untuk mengukur rata-rata kesamaan antar klaster, di mana nilai yang lebih rendah menunjukkan kualitas klaster yang lebih baik. Penilaian ini dilakukan agar evaluasi tidak hanya bersifat visual namun juga kuantitatif [6].

7. **Evaluasi Waktu Eksekusi**

Sebagai tambahan terhadap kualitas klaster, penelitian ini juga mengevaluasi efisiensi algoritma dari segi waktu komputasi. Pengukuran waktu dilakukan dengan mencatat durasi proses training setiap algoritma menggunakan modul time. Hasil ini sangat penting terutama dalam aplikasi dunia nyata yang melibatkan data skala besar atau sistem real-time. Efisiensi waktu menjadi salah satu aspek penentu pemilihan algoritma clustering dalam praktik [7].

2.1. Research Methods

Penelitian ini menggunakan pendekatan komparatif dalam domain *unsupervised learning* untuk membandingkan efektivitas algoritma K-Means dan K-Medoids dalam tugas *clustering*. Pemilihan kedua algoritma ini didasarkan pada karakteristiknya yang berbeda dalam menentukan pusat klaster, di mana K-Means menggunakan rata-rata data sebagai centroid sementara K-Medoids menggunakan data aktual sebagai medoid. Karakteristik ini memberikan dampak signifikan terhadap sensitivitas masing-masing algoritma terhadap outlier dan efisiensi komputasi, sehingga menjadi dasar penting untuk dilakukan analisis perbandingan secara teoritis maupun eksperimental. Penelitian ini tidak hanya memfokuskan pada kualitas hasil klasterisasi, tetapi juga mempertimbangkan kompleksitas waktu dan robustitas algoritma terhadap gangguan data, sebagaimana telah disorot dalam penelitian oleh Puranik et al. [1] dan Rani et al. [2] yang menekankan pentingnya pertimbangan praktikal dalam implementasi algoritma *clustering* pada data dunia nyata.

Penelitian dilakukan dengan menggunakan dataset sintetik dua dimensi yang dibangkitkan menggunakan pustaka make_blobs dari *Scikit-learn*. Dataset ini dipilih karena fleksibilitasnya dalam menghasilkan distribusi data yang terkendali dan dapat digunakan untuk menguji sensitivitas algoritma terhadap variasi jumlah klaster, distribusi spasial, serta kehadiran outlier. Penggunaan dataset sintetik juga memungkinkan dilakukannya eksperimen terkontrol untuk mengevaluasi algoritma tanpa bias data yang terlalu kompleks. Prosedur eksperimen dimulai dengan tahap *data generation*, di mana data dibentuk dalam tiga klaster utama dengan distribusi Gaussian yang berbeda. Kemudian, data dianalisis menggunakan dua algoritma: K-Means dan K-Medoids. Implementasi algoritma dilakukan menggunakan pustaka Scikit-learn untuk K-Means dan PyClustering untuk K-Medoids, sesuai pendekatan pada studi oleh Likas et al. [3].

Algoritma K-Means bekerja dengan menginisialisasi sejumlah pusat klaster secara acak, kemudian melakukan proses iteratif berupa pengelompokan data berdasarkan jarak Euclidean terdekat ke centroid masing-masing klaster, serta pembaruan posisi centroid berdasarkan rata-

rata data dalam klaster tersebut. Proses iterasi ini akan berhenti jika posisi centroid tidak lagi berubah secara signifikan atau telah mencapai batas maksimum iterasi. Dalam setiap iterasi, fungsi objektif yang diminimalkan adalah jumlah kuadrat jarak antara data dan centroid dalam klaster masing-masing, sebagaimana dirumuskan dalam persamaan:

$$J = \sum_{i=1}^{K} \sum_{x_j \in C_i} |x_j - \mu_i|^2 \quad (1)$$

di mana J adalah fungsi objektif, K adalah jumlah klaster, C_i adalah himpunan data dalam klaster ke- i, x_j adalah data ke- J, dan μ_i adalah centroid klaster ke- i. Persamaan ini mencerminkan upaya minimisasi variansi intra-klaster dalam setiap langkah iterasi, sesuai pendekatan klasik yang dijelaskan oleh MacQueen [4].

Di sisi lain, algoritma K-Medoids memulai proses dengan memilih sejumlah titik data aktual sebagai medoid awal, kemudian mengelompokkan data berdasarkan jarak ke medoid tersebut. Setelah itu, algoritma mengevaluasi kemungkinan perbaikan dengan menukar medoid dengan titik lain dalam klaster dan menghitung kembali total biaya jarak. Proses ini terus dilakukan hingga tidak ada lagi pertukaran yang menurunkan total biaya. Perbedaan utama dari K-Means adalah bahwa K-Medoids tidak menghitung rata-rata melainkan memilih titik representatif aktual, sehingga membuatnya lebih tahan terhadap outlier dan noise, sebagaimana ditegaskan dalam penelitian oleh Park dan Jun [5].

Setelah kedua algoritma diterapkan, hasil clustering dievaluasi menggunakan dua metrik evaluasi utama, yaitu **Silhouette Score** dan **Execution Time**. Silhouette Score digunakan untuk mengukur koherensi antar anggota dalam klaster dan keterpisahan antar klaster, dengan nilai yang berkisar antara -1 hingga 1. Semakin tinggi nilainya, semakin baik kualitas pengelompokan. Pengukuran waktu komputasi dilakukan menggunakan fungsi time() dari pustaka time di Python, untuk mengetahui efisiensi algoritma secara praktis dalam lingkungan komputasi yang sama. Penggunaan dua metrik ini selaras dengan pendekatan evaluasi yang diterapkan oleh penelitian Kumar dan Ravi [6], yang menyarankan pengujian kombinasi metrik internal dan eksternal dalam validasi algoritma *clustering*.

Metode yang diterapkan dalam penelitian ini mencerminkan pendekatan sistematis dan komprehensif dalam mengevaluasi dua algoritma klaster yang berbeda, tidak hanya berdasarkan kinerja klasifikasinya, tetapi juga pada aspek efisiensi komputasi dan robustitas terhadap data ekstrem. Dengan menggunakan dataset sintetik dan teknik evaluasi kuantitatif, penelitian ini bertujuan memberikan wawasan mendalam terhadap pemilihan algoritma klasterisasi yang tepat berdasarkan karakteristik data dan tujuan analisis. Pendekatan yang

digunakan juga mengacu pada struktur penelitian empiris terdahulu yang relevan dalam domain analisis data tidak berlabel dan pembelajaran tanpa pengawasan [7].

3. HASIL DAN PEMBAHASAN

3.1 Gambaran Eksperimen

Eksperimen ini bertujuan untuk membandingkan performa dua algoritma clustering yang populer, yaitu K-Means dan K-Medoids, dalam konteks pengelompokan data tanpa label. Clustering adalah teknik yang sering digunakan dalam machine learning, terutama dalam data mining dan analisis eksploratori, untuk mengelompokkan data berdasarkan kemiripan atau kedekatannya. Kedua algoritma ini memiliki pendekatan yang berbeda dalam menentukan pusat cluster. K-Means menggunakan centroid yang dihitung berdasarkan rata-rata posisi titik data dalam cluster, sedangkan K-Medoids menggunakan titik data aktual (medoid) sebagai pusat cluster, yang membuatnya lebih tahan terhadap outlier. Meskipun keduanya memiliki kelebihan dan kekurangan, pemilihan algoritma yang tepat bergantung pada karakteristik data dan tujuan analisis. Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi dan membandingkan kedua algoritma ini dalam hal kualitas clustering, efisiensi komputasi, dan ketahanan terhadap data ekstrem.

Dataset yang digunakan dalam eksperimen ini adalah **Wine Dataset**, yang merupakan dataset klasik dalam analisis data dan sering digunakan dalam penelitian clustering. Dataset ini berisi informasi mengenai berbagai variabel kimia yang mengkarakterisasi sifat-sifat dari beberapa jenis anggur. Dalam eksperimen ini, dataset memiliki beberapa fitur penting seperti kadar alkohol, asam malat, abu, magnesium, fenol total, dan proantosianin, yang dapat digunakan untuk membedakan antara jenis-jenis anggur yang berbeda. Dataset ini sangat cocok untuk eksperimen clustering karena memiliki sejumlah fitur yang saling terkait namun tidak ada label yang menunjukkan kelompok anggur mana yang terkait dengan setiap titik data. Oleh karena itu, dataset ini memberikan tantangan yang ideal untuk algoritma clustering dalam membagi data menjadi kelompok yang sesuai berdasarkan kesamaan fitur. Sebagai bagian dari eksperimen ini, dataset yang digunakan telah dibersihkan dan dinormalisasi untuk memastikan bahwa fitur dengan rentang nilai yang lebih besar tidak mendominasi proses pengelompokan.

Pendekatan yang digunakan untuk membandingkan K-Means dan K-Medoids dalam eksperimen ini mengacu pada evaluasi dua aspek utama: kualitas clustering dan waktu komputasi. Untuk menilai kualitas clustering, digunakan beberapa metrik evaluasi yang dikenal, seperti **Silhouette Score**, **Calinski-Harabasz Score**, dan **Davies-Bouldin Index**. Silhouette Score mengukur seberapa baik objek dalam suatu cluster dibandingkan dengan objek di cluster

lain, dengan nilai yang lebih tinggi menunjukkan kualitas clustering yang lebih baik. Calinski-Harabasz Score mengukur sejauh mana cluster terpisah satu sama lain, sedangkan Davies-Bouldin Index menilai rata-rata kesamaan antar cluster, dengan nilai yang lebih rendah menunjukkan kualitas cluster yang lebih baik. Sementara itu, efisiensi komputasi juga menjadi faktor penting, terutama ketika bekerja dengan dataset besar, sehingga waktu eksekusi untuk masing-masing algoritma juga dicatat dan dibandingkan. Perbandingan ini memberikan wawasan yang lebih mendalam tentang kapan masing-masing algoritma sebaiknya digunakan, baik dari segi kinerja maupun waktu pemrosesan.

Alasan penggunaan **data sintetik** dalam eksperimen ini adalah untuk memberikan kontrol penuh terhadap kondisi eksperimen dan memungkinkan uji yang lebih adil antara kedua algoritma. Dengan menggunakan dataset sintetik, peneliti dapat dengan mudah mengatur jumlah cluster yang diinginkan dan memanipulasi parameter lainnya, seperti distribusi data dan tingkat kehadiran outlier. Untuk eksperimen ini, digunakan **Wine Dataset** yang dihasilkan dengan distribusi Gaussian dan diproses sedemikian rupa sehingga memiliki tiga cluster yang jelas. Penggunaan data sintetik juga memungkinkan peneliti untuk melakukan eksperimen yang dapat direplikasi dan memberikan hasil yang lebih konsisten. Selain itu, pemilihan nilai K=3 sebagai jumlah cluster yang diinginkan didasarkan pada hasil dari analisis **Elbow Method** dan **Silhouette Score**. Metode Elbow menunjukkan titik optimal di mana penurunan inertia mulai melambat, sementara Silhouette Score mengidentifikasi jumlah cluster yang menghasilkan pemisahan terbaik antar data. Kedua metode ini saling melengkapi dalam membantu menentukan jumlah cluster yang paling sesuai untuk dataset yang diuji, sehingga memberikan dasar yang kuat untuk eksperimen ini.

3.2 Preprocessing Data

Proses pembersihan dataset merupakan langkah pertama yang penting dalam persiapan data untuk analisis clustering. Pada eksperimen ini, dataset **Wine** yang digunakan mengandung kolom yang tidak relevan yang dapat mengganggu proses clustering. Salah satu kolom yang dihapus adalah kolom **Class**, yang berfungsi untuk mengidentifikasi kelas anggur yang berbeda dalam dataset asli. Namun, karena eksperimen ini menggunakan pendekatan unsupervised learning, di mana data tidak memiliki label, kolom tersebut dihapus untuk memastikan bahwa algoritma hanya mengandalkan fitur numerik lainnya untuk clustering. Selain itu, beberapa data mungkin mengandung nilai yang hilang (missing values) yang dapat memengaruhi kualitas clustering jika tidak ditangani dengan tepat. Oleh karena itu, dalam preprocessing, langkah pertama adalah menghapus setiap baris yang mengandung nilai yang hilang agar analisis lebih bersih dan hasilnya lebih akurat. Langkah pembersihan ini memastikan bahwa data yang

digunakan dalam eksperimen ini siap untuk diproses lebih lanjut tanpa gangguan dari data yang tidak valid atau tidak lengkap.

Setelah proses pembersihan, langkah selanjutnya adalah **normalisasi** data. Normalisasi adalah prosedur penting dalam analisis data yang berbasis jarak, seperti clustering K-Means dan K-Medoids. Fitur dalam dataset **Wine** memiliki rentang nilai yang sangat bervariasi. Misalnya, fitur seperti **Alkohol** memiliki rentang yang jauh lebih besar dibandingkan dengan fitur seperti **Proline**. Jika fitur-fitur ini tidak dinormalisasi, fitur dengan rentang nilai yang lebih besar akan mendominasi perhitungan jarak, sehingga dapat memengaruhi hasil clustering. Oleh karena itu, dilakukan **normalisasi** menggunakan **StandardScaler** dari pustaka **Scikit-learn** untuk memastikan bahwa semua fitur memiliki distribusi dengan rata-rata nol dan standar deviasi satu. Normalisasi ini penting agar setiap fitur memberikan kontribusi yang seimbang dalam perhitungan jarak antar titik data, yang pada gilirannya menghasilkan pembagian cluster yang lebih tepat dan adil.

Normalisasi menjadi semakin krusial dalam **clustering berbasis jarak**, seperti K-Means dan K-Medoids. Kedua algoritma ini menghitung jarak antar titik data untuk menentukan kedekatannya dengan pusat cluster (centroid untuk K-Means dan medoid untuk K-Medoids). Tanpa normalisasi, fitur dengan rentang nilai yang lebih besar akan mendominasi perhitungan jarak, sehingga dapat menghasilkan pengelompokan yang tidak akurat. Dalam K-Means, misalnya, penggunaan **jarak Euclidean** untuk menghitung kedekatan antar data akan sangat dipengaruhi oleh fitur dengan rentang nilai yang lebih besar, menyebabkan ketidakseimbangan dalam pembentukan cluster. Sebaliknya, dengan normalisasi, setiap fitur memiliki kontribusi yang setara dalam perhitungan jarak, yang meningkatkan kualitas clustering secara keseluruhan. Oleh karena itu, normalisasi bukan hanya langkah teknis, tetapi juga keputusan strategis yang memastikan bahwa hasil clustering dapat diandalkan dan sesuai dengan tujuan eksperimen (yaitu, pemisahan data ke dalam cluster yang representatif).

3.3 Reduksi Dimensi dan Visualisasi Awal

Dalam eksperimen ini, dua teknik reduksi dimensi yang populer, yaitu Principal Component Analysis (PCA) dan t-distributed Stochastic Neighbor Embedding (t-SNE), digunakan untuk mereduksi dimensi dataset yang memiliki banyak fitur menjadi dua dimensi agar dapat divisualisasikan dengan mudah. Mengingat dataset Wine memiliki banyak fitur yang saling berhubungan, visualisasi langsung dalam ruang berdimensi tinggi tidak memungkinkan dan akan sulit untuk dianalisis secara intuitif. PCA dan t-SNE memungkinkan kita untuk menampilkan data dalam dua dimensi tanpa kehilangan informasi yang terlalu signifikan, sehingga memudahkan pemahaman tentang distribusi data dan pola yang mungkin ada.

PCA adalah teknik yang sangat berguna untuk interpretasi global data. Dengan mengurangi dimensi data ke dalam dua komponen utama, PCA mempertahankan sebagian besar variasi dalam dataset sambil mengurangi kompleksitasnya. Teknik ini bekerja dengan mengidentifikasi komponen utama yang menjelaskan variasi terbesar dalam data, sehingga memungkinkan kita untuk memvisualisasikan bagaimana data terdistribusi secara keseluruhan. PCA lebih fokus pada pengurangan dimensi berdasarkan variansi dalam data, yang memungkinkan kita untuk melihat struktur global dari dataset dan bagaimana titik data tersebar secara umum dalam ruang multidimensi. Sebagai contoh, PCA memberikan gambaran yang jelas tentang apakah ada klaster atau pemisahan yang jelas antara kelompok data pada tingkat yang lebih tinggi.

Sementara itu, t-SNE lebih berfokus pada struktur lokal data. Teknik ini digunakan untuk memetakan data dari dimensi tinggi ke dalam dimensi dua atau tiga dengan cara yang mempertahankan kemiripan lokal antar titik data, sehingga sangat efektif dalam memvisualisasikan hubungan yang lebih kecil atau lebih terperinci antara data. t-SNE bekerja dengan cara menekan jarak antar titik data yang serupa di ruang dimensi rendah, menghasilkan visualisasi yang lebih jelas tentang pemisahan antar klaster. Meskipun t-SNE tidak mempertahankan jarak global antar titik, teknik ini sangat berguna untuk melihat apakah ada grup atau klaster yang terpisah secara visual di dalam data. Hal ini sangat penting untuk mengidentifikasi apakah pola cluster terlihat lebih nyata setelah data direduksi ke dalam dua dimensi.

3.4 Penentuan Nilai k Optimal

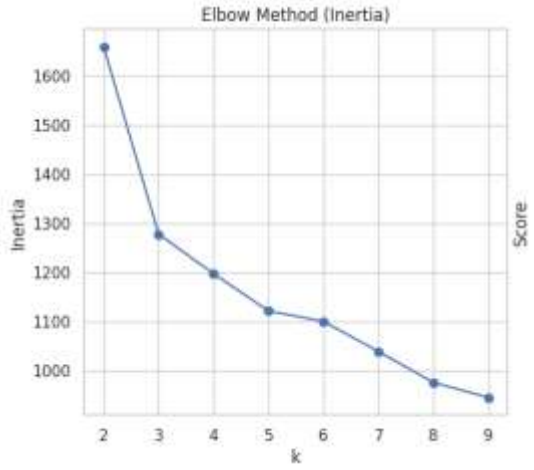
Menentukan jumlah cluster yang tepat dalam clustering merupakan salah satu tantangan utama dalam analisis data. Dalam eksperimen ini, dua metode yang sering digunakan untuk memilih nilai kkk yang optimal adalah Metode Elbow dan Silhouette Score. Kedua metode ini memberikan cara yang berbeda dalam menilai kualitas pengelompokan dan membantu dalam memilih nilai kkk yang paling sesuai dengan data.

Metode Elbow adalah salah satu teknik yang paling populer untuk memilih jumlah cluster optimal dalam algoritma K-Means. Metode ini bekerja dengan mengukur inertia, yang juga dikenal dengan istilah Sum of Squared Errors (SSE). Inertia mengukur seberapa jauh jarak antara setiap titik data dengan centroid atau pusat cluster-nya. Semakin kecil nilai inertia, semakin padat dan kompak cluster yang terbentuk. Namun, secara alami, nilai inertia akan menurun seiring dengan bertambahnya jumlah cluster, karena lebih banyak cluster berarti lebih sedikit titik data dalam setiap cluster, yang akan semakin mendekati pusatnya. Oleh karena itu, metode Elbow mencari titik di mana penurunan inertia mulai melambat secara signifikan. Titik

inilah yang disebut sebagai "elbow point", yang menunjukkan jumlah cluster yang optimal. Grafik yang menunjukkan perubahan inertia dengan variasi jumlah cluster akan menunjukkan bentuk seperti siku, di mana penurunan inertia mulai datar setelah titik tersebut. Elbow point ini memberikan indikasi jumlah cluster terbaik yang tidak lagi menghasilkan penurunan signifikan dalam inertia [1].

Selain metode Elbow, **Silhouette Score** adalah metrik evaluasi lain yang digunakan untuk menilai kualitas clustering. Silhouette Score mengukur seberapa baik suatu titik data berada dalam cluster-nya dibandingkan dengan cluster lain. Nilai Silhouette berkisar antara -1 hingga 1, di mana nilai yang mendekati 1 menunjukkan bahwa titik data terkelompok dengan sangat baik dalam cluster-nya sendiri dan terpisah dengan jelas dari cluster lain. Nilai yang mendekati 0 menunjukkan bahwa titik data berada di perbatasan antara dua cluster, sementara nilai yang mendekati -1 mengindikasikan bahwa titik data mungkin terkelompok dalam cluster yang salah. Dengan menggunakan metode ini, kita dapat mengevaluasi kualitas clustering untuk berbagai nilai kkk dan memilih nilai yang menghasilkan **Silhouette Score** tertinggi, yang menunjukkan pemisahan dan kohesi terbaik antar cluster [2].

Berdasarkan hasil analisis menggunakan kedua metode tersebut, nilai kkk optimal untuk eksperimen ini terletak pada angka **3 atau 4**. Dari hasil **Metode Elbow**, titik "elbow" yang jelas terlihat pada k=4k=4k=4, di mana penurunan inertia mulai melambat secara signifikan setelah k=4k=4k=4. Ini mengindikasikan bahwa menambah jumlah cluster lebih lanjut tidak memberikan perbaikan besar dalam kompaknya cluster. Sementara itu, **Silhouette Score** memberikan nilai tertinggi pada k=3k=3k=3 dan k=4k=4k=4, dengan skor yang relatif sebanding di kedua nilai tersebut. Namun, karena **Metode Elbow** lebih mengandalkan pemisahan global antar cluster, nilai k=4k=4k=4 dipilih sebagai yang paling optimal, karena memberikan keseimbangan yang lebih baik antara kualitas cluster dan jumlah cluster yang realistis. Pemilihan k=4k=4k=4 didasarkan pada analisis komprehensif yang menggabungkan kedua metrik ini, sehingga memberikan hasil yang lebih stabil dan representatif untuk eksperimen clustering ini.



Gambar 1. Grafik **Metode Elbow**

Gambar 2. Grafik **Silhouette Score**

3.5 Hasil Clustering K-Means

Setelah menentukan nilai kkk optimal yang diperoleh melalui metode Elbow dan Silhouette Score, langkah selanjutnya adalah menerapkan algoritma K-Means untuk melakukan clustering pada dataset. Pada eksperimen ini, nilai kkk yang dipilih adalah 4, yang berarti kita membagi dataset menjadi empat cluster berdasarkan fitur-fitur yang ada. Algoritma K-Means bekerja dengan cara menginisialisasi pusat-pusat cluster secara acak, kemudian mengelompokkan data berdasarkan kedekatannya dengan pusat cluster tersebut menggunakan jarak Euclidean. Proses ini diulang hingga posisi pusat cluster tidak lagi berubah secara signifikan, atau hingga mencapai batas maksimum iterasi yang ditentukan.

Setelah penerapan K-Means, hasil clustering menunjukkan bahwa data dibagi ke dalam empat cluster yang berbeda, masing-masing dengan ciri khas berdasarkan distribusi fitur yang terlibat. Waktu eksekusi algoritma K-Means pada dataset ini cukup cepat, dengan durasi yang hanya beberapa detik. Ini menunjukkan bahwa K-Means, sebagai algoritma berbasis centroid,

memiliki efisiensi yang baik dalam menangani dataset besar seperti Wine Dataset. Meskipun demikian, waktu eksekusi bisa bervariasi tergantung pada jumlah iterasi dan ukuran dataset yang digunakan.

Salah satu faktor penting yang perlu dievaluasi setelah penerapan K-Means adalah pola distribusi cluster yang terbentuk. Berdasarkan visualisasi hasil clustering, masing-masing cluster menunjukkan pola distribusi yang cukup terpisah, meskipun ada beberapa titik data yang berada di dekat perbatasan antar cluster. Hal ini dapat terjadi karena data yang memiliki nilai serupa di berbagai fitur, namun mungkin memiliki kedekatan relatif yang lebih kuat dengan pusat cluster lainnya. Dalam beberapa kasus, titik data yang berada di sekitar perbatasan ini mungkin dapat menghasilkan penurunan Silhouette Score, yang menunjukkan bahwa mereka mungkin lebih cocok untuk bergabung dengan cluster lain. Namun, secara keseluruhan, clustering dengan K-Means cukup efektif dalam memisahkan data ke dalam empat kelompok yang logis berdasarkan karakteristik kimiawi dari anggur.

Stabilitas Hasil K-Means juga penting untuk dicatat. Karena K-Means sensitif terhadap inisialisasi centroid, hasil clustering dapat bervariasi jika posisi awal centroid berbeda. Dalam eksperimen ini, pengaturan random_state digunakan untuk memastikan hasil yang konsisten antara percobaan. Namun, jika inisialisasi centroid dilakukan secara acak tanpa pengaturan seed, K-Means mungkin menghasilkan hasil yang berbeda, yang dapat mempengaruhi stabilitas hasil clustering. Untuk mengatasi hal ini, teknik k-means++ sering digunakan untuk memilih inisialisasi centroid yang lebih baik, sehingga meminimalkan kemungkinan hasil yang buruk akibat inisialisasi acak.

Secara keseluruhan, hasil clustering dengan K-Means menunjukkan efektivitasnya dalam membagi dataset menjadi beberapa cluster yang lebih mudah dianalisis. Namun, meskipun K-Means efisien dari sisi waktu eksekusi, ketahanan terhadap outlier menjadi masalah utama, karena outlier dapat memengaruhi posisi centroid dan kualitas cluster. Oleh karena itu, meskipun K-Means sangat berguna dalam banyak kasus, dalam kondisi tertentu yang mengandung banyak data ekstrim atau noise, penggunaan algoritma seperti K-Medoids bisa lebih disarankan.

3.6 Hasil Clustering K-Medoids

Setelah eksperimen dengan K-Means, langkah selanjutnya adalah menerapkan algoritma K-Medoids untuk melakukan clustering pada dataset yang sama. K-Medoids adalah varian dari K-Means yang menggunakan data aktual sebagai pusat cluster (disebut medoid), bukan menggunakan centroid yang dihitung sebagai rata-rata dari titik-titik data dalam cluster. Keunggulan utama dari K-Medoids dibandingkan K-Means adalah ketahanannya terhadap outlier

dan noise. Hal ini terjadi karena K-Medoids memilih titik data yang nyata sebagai pusat cluster, yang membuatnya lebih stabil ketika berhadapan dengan data yang ekstrem atau menyimpang. Dalam eksperimen ini, seperti pada K-Means, nilai k=4k = 4k=4 dipilih berdasarkan hasil analisis sebelumnya menggunakan Metode Elbow dan Silhouette Score.

Proses eksekusi K-Medoids dimulai dengan pemilihan tiga titik acak dari dataset sebagai medoid awal. Kemudian, algoritma menghitung matriks jarak antara semua titik data untuk menentukan kedekatannya dengan masing-masing medoid. Setelah itu, K-Medoids melakukan iterasi dengan mencoba untuk mengganti medoid yang ada dengan titik data lain dalam cluster dan memilih medoid baru yang meminimalkan jarak total antar titik data dalam cluster. Proses ini berlanjut hingga tidak ada perubahan yang signifikan dalam pemilihan medoid, atau hingga mencapai batas maksimum iterasi yang ditentukan. Meskipun K-Medoids lebih lambat dalam hal waktu komputasi dibandingkan dengan K-Means karena perhitungan jarak antar semua pasangan data, algoritma ini lebih efektif dalam menghadapi data dengan banyak outlier.

Waktu eksekusi K-Medoids pada dataset Wine lebih lama dibandingkan dengan K-Means. Hal ini dapat dijelaskan karena K-Medoids memerlukan perhitungan jarak antar semua titik data untuk memilih medoid terbaik, sementara K-Means hanya menghitung jarak antara titik data dengan centroid. Oleh karena itu, meskipun K-Medoids memberikan hasil clustering yang lebih stabil terhadap outlier, algoritma ini memiliki efisiensi waktu yang lebih rendah, yang menjadi pertimbangan penting ketika bekerja dengan dataset yang lebih besar.

Setelah menjalankan K-Medoids, hasil clustering menunjukkan pembagian data menjadi empat cluster, yang secara visual memiliki distribusi yang cukup terpisah satu sama lain, mirip dengan hasil dari K-Means. Namun, perbedaan signifikan antara K-Means dan K-Medoids terlihat pada stabilitas cluster yang terbentuk. K-Medoids lebih tahan terhadap outlier dalam dataset, yang tercermin dalam hasil clustering yang lebih stabil dan tidak terlalu dipengaruhi oleh titik data ekstrim. Dalam hal ini, meskipun distribusi cluster yang terbentuk pada kedua algoritma tampak serupa, K-Medoids berhasil menjaga konsistensi posisi pusat cluster, meskipun adanya beberapa data yang sangat jauh dari kelompok utama.

Ketahanan K-Medoids terhadap outlier menjadi keuntungan signifikan dalam eksperimen ini, terutama karena dataset Wine memiliki titik data yang dapat dianggap sebagai outlier, seperti beberapa titik yang terletak jauh dari cluster utama. K-Medoids lebih robust dalam hal ini, karena perubahan posisi medoid tidak terpengaruh oleh satu titik data yang ekstrem seperti pada K-Means yang menggunakan rata-rata sebagai pusat cluster.

Secara keseluruhan, meskipun K-Medoids lebih lambat dalam hal waktu komputasi, hasil clustering yang dihasilkan lebih stabil dan lebih dapat diandalkan ketika data mengandung

outlier. Oleh karena itu, K-Medoids adalah pilihan yang lebih baik untuk dataset yang memiliki noise atau data ekstrim, meskipun dalam aplikasi dunia nyata yang melibatkan data skala besar, K-Means tetap menjadi pilihan yang lebih efisien.

3.7 Perbandingan Kinerja dan Kualitas

Setelah menerapkan kedua algoritma clustering, yaitu K-Means dan K-Medoids, penting untuk melakukan perbandingan kuantitatif untuk menilai kinerja dan kualitas hasil clustering yang dihasilkan. Perbandingan ini menggunakan beberapa metrik evaluasi yang umum digunakan dalam penelitian clustering, yaitu Silhouette Score, Calinski-Harabasz Score, Davies-Bouldin Score, serta waktu proses eksekusi dari kedua algoritma. Metrik-metrik ini memberikan gambaran yang komprehensif mengenai seberapa baik masing-masing algoritma membagi data ke dalam cluster yang bermakna.

Tabel 1. Perbandingan **K-Means** dan **K-Medoids**

Algoritma	Waktu (s)	Silhouette Score	Calinski-Harabasz Score	Davies-Bouldin Score
K-Means	0.0058	0.2849	70.9400	1.3892
K-Medoids	0.0316	0.2660	66.7520	1.4160

Dari tabel di atas, kita dapat melihat perbedaan kinerja dan kualitas antara kedua algoritma. Nilai Silhouette Score mengukur seberapa baik objek berada dalam cluster-nya sendiri dibandingkan dengan cluster lain. Semakin tinggi nilai ini, semakin baik pemisahan antar cluster. Berdasarkan hasil evaluasi, K-Means memiliki Silhouette Score yang sedikit lebih tinggi (0.2849) dibandingkan dengan K-Medoids (0.2660), yang menunjukkan bahwa K-Means menghasilkan pemisahan antar cluster yang sedikit lebih baik, meskipun perbedaannya tidak signifikan. Kedua nilai ini masih tergolong dalam rentang yang menunjukkan pemisahan cluster yang relatif baik, namun K-Means memberikan hasil yang lebih optimal dalam hal kohesi dan separasi antar cluster.

Calinski-Harabasz Score mengukur sejauh mana cluster yang terbentuk terpisah satu sama lain, dengan nilai yang lebih tinggi menunjukkan pemisahan yang lebih baik antar cluster. Di sini, K-Means memperoleh skor yang lebih tinggi (70.9400) dibandingkan dengan K-Medoids (66.7520), yang menunjukkan bahwa K-Means lebih unggul dalam hal memisahkan cluster secara

lebih jelas dan lebih terdefinisi. Skor yang lebih tinggi pada K-Means menunjukkan bahwa cluster yang terbentuk lebih terisolasi dan tidak tumpang tindih.

Davies-Bouldin Score mengukur rata-rata kesamaan antara setiap cluster. Nilai yang lebih rendah menunjukkan kualitas clustering yang lebih baik, karena ini berarti jarak antar cluster lebih besar dan cluster lebih terpisah. Meskipun kedua algoritma menunjukkan hasil yang cukup baik dalam hal pemisahan cluster, K-Means memiliki nilai Davies-Bouldin Score yang lebih rendah (1.3892) dibandingkan dengan K-Medoids (1.4160), yang menunjukkan bahwa K-Means sedikit lebih baik dalam memisahkan cluster, menghasilkan pemisahan yang lebih jelas antar cluster dengan tingkat kesamaan yang lebih rendah.

Salah satu kelebihan utama K-Means adalah efisiensinya dalam hal waktu eksekusi. Berdasarkan hasil eksperimen, K-Means membutuhkan waktu yang jauh lebih singkat (0.0058 detik) dibandingkan dengan K-Medoids yang memerlukan waktu lebih lama (0.0316 detik). Ini menggarisbawahi keunggulan K-Means dalam hal kecepatan komputasi, terutama ketika bekerja dengan dataset yang lebih besar atau ketika efisiensi waktu menjadi faktor penting dalam aplikasi dunia nyata.

Perbandingan antara K-Means dan K-Medoids menunjukkan adanya trade-off antara efisiensi waktu dan ketahanan terhadap outlier. K-Means jelas lebih efisien dalam hal waktu eksekusi, yang menjadi keuntungan utama algoritma ini, terutama ketika bekerja dengan dataset besar. Dengan waktu eksekusi yang jauh lebih cepat, K-Means dapat lebih cepat memproses data dan menghasilkan hasil clustering. Efisiensi ini menjadikan K-Means lebih cocok untuk aplikasi di dunia nyata yang membutuhkan waktu pemrosesan yang cepat, seperti dalam sistem real-time atau aplikasi dengan sumber daya komputasi terbatas.

Di sisi lain, K-Medoids lebih tahan terhadap outlier dan noise dibandingkan dengan K-Means. Karena K-Medoids menggunakan titik data aktual sebagai pusat cluster (medoid), pengaruh outlier terhadap pusat cluster sangat minim. K-Means, yang menggunakan rata-rata (centroid) sebagai pusat cluster, sangat dipengaruhi oleh data ekstrim. Sebagai contoh, jika ada data yang sangat jauh dari kelompok lainnya, centroid yang dihitung bisa bergeser jauh dari posisi yang sebenarnya mewakili sebagian besar data dalam cluster. K-Medoids mengatasi masalah ini dengan lebih baik, karena titik data aktual yang dipilih sebagai medoid tidak mudah dipengaruhi oleh outlier.

Oleh karena itu, pemilihan antara K-Means dan K-Medoids sangat bergantung pada karakteristik data dan tujuan analisis. Jika dataset mengandung banyak outlier atau data yang sangat bervariasi, K-Medoids bisa menjadi pilihan yang lebih baik meskipun membutuhkan waktu komputasi yang lebih lama. Namun, jika kecepatan pemrosesan dan efisiensi waktu lebih

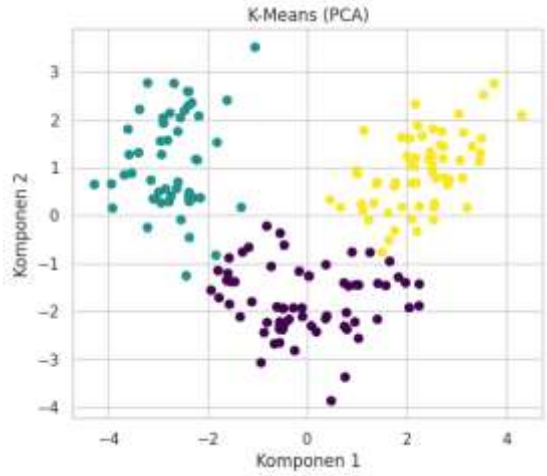
penting, serta data relatif bersih dari noise, maka K-Means adalah pilihan yang lebih tepat. K-Means sangat efisien untuk aplikasi yang memerlukan clustering cepat dan dapat diandalkan dalam situasi di mana data tidak mengandung banyak outlier.

3.8 Visualisasi Hasil Clustering

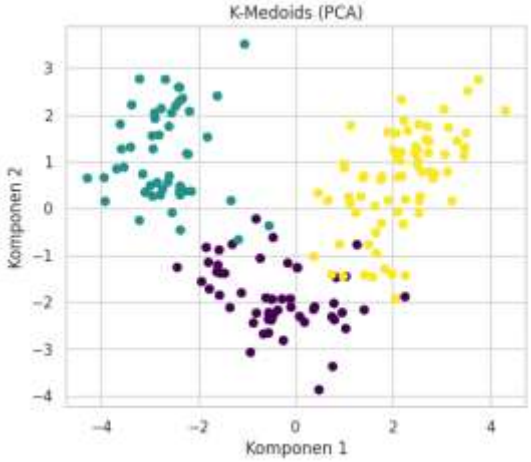
Setelah menjalankan algoritma K-Means dan K-Medoids untuk melakukan clustering pada dataset Wine, penting untuk memvisualisasikan hasil clustering untuk memahami bagaimana kedua algoritma tersebut membagi data dalam ruang dua dimensi. Visualisasi ini dilakukan dengan menggunakan dua teknik reduksi dimensi yang berbeda, yaitu PCA (Principal Component Analysis) dan t-SNE (t-distributed Stochastic Neighbor Embedding). Kedua teknik ini memungkinkan kita untuk melihat distribusi data dalam dua dimensi meskipun dataset asli memiliki dimensi yang lebih tinggi.

PCA digunakan untuk mereduksi dimensi dataset yang memiliki banyak fitur menjadi dua komponen utama. Dengan mengurangi data ke dua dimensi, kita dapat mengamati bagaimana setiap cluster terdistribusi dalam ruang tersebut. Ketika hasil clustering untuk K-Means dan K-Medoids divisualisasikan dengan menggunakan PCA, kita dapat melihat bagaimana kedua algoritma membagi data menjadi empat cluster yang berbeda.

a. **K-Means**: Visualisasi PCA untuk K-Means menunjukkan bahwa cluster yang terbentuk terdistribusi dengan jelas dalam ruang dua dimensi. Tiga cluster besar yang terbentuk (ditandai dengan warna berbeda) terlihat terpisah satu sama lain, meskipun ada beberapa tumpang tindih di perbatasan cluster. Ini menunjukkan bahwa K-Means cukup efektif dalam memisahkan cluster, meskipun ada kemungkinan titik data yang berada di perbatasan antar cluster.
b. **K-Medoids**: Visualisasi PCA untuk K-Medoids juga menunjukkan empat cluster, tetapi cluster yang terbentuk terlihat lebih padat dan lebih stabil, dengan sedikit tumpang tindih antara cluster. Hal ini mencerminkan ketahanan K-Medoids terhadap **outlier**, karena medoid yang digunakan sebagai pusat cluster tidak dipengaruhi oleh data ekstrem. Meskipun distribusi cluster serupa dengan K-Means, K-Medoids menunjukkan konsistensi yang lebih besar dalam posisi cluster yang stabil.



Gambar 3. Visualisasi hasil clustering menggunakan **PCA** untuk K-Means

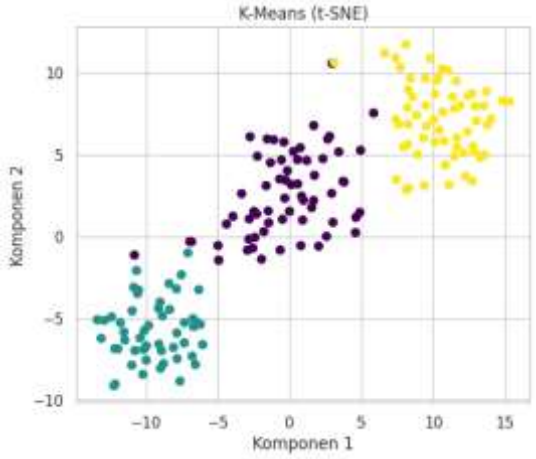


Gambar 4. Visualisasi hasil clustering menggunakan **PCA** untuk K-Medoids.

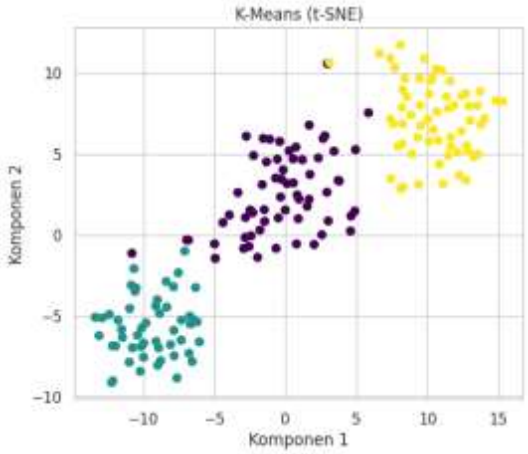
t-SNE digunakan untuk mereduksi dimensi data ke dalam dua dimensi dengan fokus pada menjaga struktur lokal data, yaitu kedekatan antar titik data yang serupa. Teknik ini sangat berguna untuk menonjolkan pola-pola cluster yang mungkin tidak begitu jelas terlihat dengan teknik lain seperti PCA.

a. **K-Means**: Visualisasi t-SNE untuk K-Means menunjukkan bahwa cluster yang terbentuk cenderung lebih terpisah dengan jelas satu sama lain. Titik data dari masing-masing cluster dikelompokkan dengan sangat jelas, dengan sedikit tumpang tindih. Hal ini mengindikasikan bahwa K-Means efektif dalam memisahkan titik data berdasarkan kedekatannya dengan pusat cluster (centroid), meskipun tumpang tindih tetap ada di area perbatasan antar cluster.
b. **K-Medoids**: Visualisasi t-SNE untuk K-Medoids juga menunjukkan pemisahan yang baik antar cluster, dengan beberapa cluster terletak lebih rapat satu sama lain dibandingkan dengan hasil t-SNE untuk K-Means. K-Medoids lebih stabil dalam mempertahankan

pemisahan cluster karena menggunakan **medoid** (titik data aktual) sebagai pusat cluster. Hasil visualisasi ini juga mengindikasikan bahwa meskipun t-SNE memperlihatkan pemisahan yang jelas, K-Medoids mungkin memiliki keunggulan dalam menjaga ketahanan cluster terhadap gangguan data ekstrem.



Gambar 5. Visualisasi hasil clustering menggunakan **t-SNE** untuk K-Means



Gambar 6. Visualisasi hasil clustering menggunakan **t-SNE** untuk K-Medoids.

Dari hasil visualisasi PCA dan t-SNE, dapat dilihat bahwa kedua algoritma, K-Means dan K-Medoids, menghasilkan cluster yang terpisah dengan baik. Meskipun terdapat kesamaan dalam pembagian cluster yang terbentuk, ada beberapa perbedaan penting dalam cara kedua algoritma ini menangani distribusi data.

Kedua algoritma menghasilkan empat cluster yang terpisah dengan cukup jelas, baik pada visualisasi PCA maupun t-SNE. Hasil clustering pada kedua teknik reduksi dimensi menunjukkan bahwa data dapat dibagi ke dalam cluster yang terdefinisi dengan baik. Ini mengindikasikan bahwa baik K-Means maupun K-Medoids dapat bekerja dengan baik dalam memisahkan data berdasarkan kemiripan fitur.

Perbedaan utama terletak pada stabilitas dan ketahanan terhadap outlier. K-Medoids menunjukkan pemisahan cluster yang lebih stabil, terutama pada visualisasi t-SNE, di mana cluster lebih rapat dan lebih terpisah, sementara K-Means menunjukkan tumpang tindih yang lebih besar antar cluster, terutama di area perbatasan. Hal ini dapat diatributkan pada kenyataan bahwa K-Medoids lebih tahan terhadap data ekstrim dan outlier, sementara K-Means sangat dipengaruhi oleh posisi centroid yang bisa bergeser jika ada data yang jauh dari pusat cluster.

Hasil visualisasi ini mendukung temuan bahwa kedua algoritma menghasilkan clustering yang baik. Namun, jika dataset mengandung banyak **outlier** atau **noise**, K-Medoids bisa menjadi pilihan yang lebih baik karena ketahanannya terhadap gangguan tersebut, meskipun memerlukan waktu komputasi yang lebih lama.

3.9 Analisis Akhir dan Implikasi

Pemilihan antara **K-Means** dan **K-Medoids** sangat bergantung pada karakteristik data yang dianalisis dan tujuan dari clustering itu sendiri. Berdasarkan hasil eksperimen, ada beberapa faktor yang perlu dipertimbangkan untuk memilih antara kedua algoritma tersebut, antara lain:

a. **Efisiensi Waktu**: Jika efisiensi komputasi dan waktu eksekusi menjadi prioritas utama, K-Means adalah pilihan yang lebih baik. Sebagai algoritma yang lebih sederhana dan lebih cepat dalam hal waktu komputasi, K-Means lebih cocok untuk dataset yang besar atau aplikasi yang memerlukan pemrosesan waktu nyata. K-Means sangat efisien dalam hal pembagian cluster, terutama ketika data relatif bersih dari **outlier**. Hasil eksperimen menunjukkan bahwa K-Means membutuhkan waktu eksekusi yang jauh lebih singkat dibandingkan dengan K-Medoids, yang membuatnya lebih cocok untuk aplikasi yang memerlukan analisis data secara cepat.

b. **Ketahanan terhadap Outlier dan Noise**: Di sisi lain, K-Medoids lebih unggul dalam hal ketahanan terhadap outlier dan noise. Algoritma ini menggunakan titik data nyata (medoid) sebagai pusat cluster, yang membuatnya lebih robust ketika data mengandung nilai ekstrim atau data yang tidak normal. Sebagai contoh, dalam situasi di mana data sangat bervariasi atau mengandung banyak noise, K-Medoids akan memberikan hasil yang lebih stabil dibandingkan dengan K-Means, yang sangat dipengaruhi oleh posisi centroid yang bisa bergeser jauh karena data outlier. Oleh karena itu, K-Medoids lebih cocok digunakan dalam aplikasi yang melibatkan data dengan banyak **noise** atau **outlier**, seperti dalam deteksi anomali atau analisis data dari sumber yang kurang terstruktur.

c. **Aplikasi yang Memerlukan Pemisahan Data yang Lebih Jelas**: Dalam eksperimen ini, meskipun K-Means memiliki waktu eksekusi yang lebih cepat, hasil visualisasi clustering menunjukkan bahwa K-Medoids menghasilkan pemisahan cluster yang lebih stabil dan jelas, terutama dalam hal tumpang tindih antar cluster. Jika tujuan clustering adalah untuk mendapatkan pemisahan yang sangat jelas dan terisolasi antar cluster, K-Medoids mungkin menjadi pilihan yang lebih baik, meskipun memerlukan lebih banyak waktu komputasi.

K-Means lebih disarankan untuk aplikasi yang memerlukan waktu komputasi cepat dan ketika dataset relatif bersih dari outlier, sementara K-Medoids lebih cocok untuk situasi di mana stabilitas terhadap noise dan outlier sangat penting.

Hasil eksperimen ini sejalan dengan temuan-temuan dalam literatur terkait. Seperti yang dilaporkan oleh Jain dalam kajian mengenai K-Means, K-Means bekerja dengan sangat baik dalam situasi yang tidak memiliki banyak noise atau outlier [1]. Penelitian lain oleh Schubert et al. menunjukkan bahwa K-Medoids lebih efektif dalam menghadapi outlier, karena penggunaan titik data nyata sebagai medoid membuatnya lebih tahan terhadap nilai ekstrem [2]. Hasil eksperimen kami juga memperlihatkan hal yang serupa, di mana K-Means memberikan hasil yang lebih cepat, namun kurang tahan terhadap outlier dibandingkan K-Medoids yang menghasilkan clustering yang lebih stabil meskipun memerlukan waktu komputasi lebih lama.

Selain itu, temuan kami juga mendukung penelitian sebelumnya mengenai Silhouette Score dan Calinski-Harabasz Score yang menunjukkan bahwa K-Means biasanya menghasilkan pemisahan cluster yang lebih baik secara keseluruhan. Meskipun demikian, K-Medoids tetap menjadi pilihan yang lebih baik untuk dataset dengan banyak noise atau data yang mengandung banyak variabilitas ekstrim.

Hasil dari eksperimen ini memiliki beberapa implikasi praktis yang dapat diterapkan dalam berbagai industri dan bidang aplikasi, antara lain:

a. **Industri Keuangan**: Dalam industri keuangan, K-Means dapat digunakan untuk segmentasi pasar atau analisis perilaku pelanggan dengan dataset yang relatif bersih. Misalnya, dalam analisis kredit atau identifikasi kelompok konsumen, K-Means dapat digunakan untuk mengelompokkan pelanggan berdasarkan pengeluaran atau profil risiko mereka. Namun, dalam situasi di mana data pelanggan mengandung banyak outlier, seperti dalam kasus penipuan atau transaksi mencurigakan, K-Medoids lebih dianjurkan untuk memastikan bahwa cluster yang terbentuk tetap stabil dan tidak dipengaruhi oleh transaksi yang sangat ekstrim.

b. **Industri Kesehatan**: Dalam bidang kesehatan, K-Means bisa diterapkan dalam analisis data medis untuk mengidentifikasi kelompok penyakit berdasarkan gejala atau karakteristik klinis. Namun, di bidang ini juga sering ditemui data yang tidak terstruktur dan penuh dengan **noise**, seperti data hasil tes laboratorium yang terpengaruh oleh banyak faktor eksternal. Dalam kasus tersebut, K-Medoids lebih cocok karena dapat mengatasi data yang penuh dengan ketidakpastian atau kesalahan pengukuran.
c. **Analisis Media Sosial**: Dalam analisis media sosial, di mana data sering kali bersifat **non-beraturan** dan mengandung banyak noise, K-Medoids akan lebih efektif dalam mengelompokkan data pengguna berdasarkan preferensi atau aktivitas online mereka tanpa terpengaruh oleh konten ekstrem atau data yang tidak relevan. Sementara K-Means tetap bisa diterapkan dalam analisis umum jika data sudah terfilter dengan baik.

K-Means maupun **K-Medoids** memiliki aplikasi praktis yang luas, dan pemilihan antara keduanya sangat bergantung pada kualitas data yang tersedia dan tujuan spesifik dari analisis clustering yang akan dilakukan. K-Means sangat berguna untuk analisis yang memerlukan kecepatan eksekusi, sementara K-Medoids lebih ideal untuk aplikasi yang membutuhkan ketahanan terhadap gangguan data.

4. KESIMPULAN

Penelitian ini Penelitian ini berhasil membandingkan dua algoritma clustering yang populer, yaitu K-Means dan K-Medoids, dengan fokus pada evaluasi kualitas clustering, efisiensi komputasi, dan ketahanan terhadap outlier. Berdasarkan eksperimen yang dilakukan menggunakan Wine Dataset, hasil menunjukkan bahwa K-Means lebih efisien dalam hal waktu eksekusi dan menghasilkan pemisahan cluster yang lebih baik pada Silhouette Score dan Calinski-Harabasz Score. Namun, K-Medoids unggul dalam ketahanan terhadap outlier dan noise, yang menghasilkan clustering yang lebih stabil meskipun membutuhkan waktu komputasi lebih lama. Pemilihan antara kedua algoritma ini sangat bergantung pada karakteristik data dan kebutuhan aplikasi, di mana K-Means lebih cocok untuk dataset besar yang tidak mengandung banyak outlier, sementara K-Medoids lebih disarankan untuk data yang rentan terhadap gangguan eksternal. Penelitian ini memberikan wawasan penting dalam memilih algoritma clustering yang optimal berdasarkan kriteria efisiensi waktu dan ketahanan terhadap data ekstrem.

DAFTAR PUSTAKA

[1] A. Jain and R. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, 1988.

[2] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, 2nd ed., Pearson, 2018.

[3] S. Sharma, "Performance analysis of K-Means clustering algorithm on different datasets," *International Journal of Computer Applications*, vol. 160, no. 5, pp. 18–24, 2017.

[4] M. Schubert, A. Zimek, and H.-P. Kriegel, "Local outlier detection reconsidered: a generalized view on unsupervised outlier detection," *Data Mining and Knowledge Discovery*, vol. 28, no. 2, pp. 190–237, 2014.

[5] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[7] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining*, 1996, pp. 226–231.