**BULLETIN OF COMPUTER SCIENCE RESEARCH**
**ISSN 2774-3659 (Media Online)**
Vol 5, No 4, June 2025 | Hal 554-563
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.605

# Analisis Perbandingan Metode DBSCAN dan Meanshift dalam Klasterisasi Data Gempa Bumi di Indonesia

**MHD Ade Setiawan, Fitri Insani*, Yelfi Vitriani, Yusra**

Sains dan Teknologi, Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia
Email: [1]ades51648@gmail.com, [2,*]fitri.insani@uin-suska.ac.id, [3]yelfi.vitriani@uin-suska.ac.id, [4]yusra@uin-suska.ac.id
Email Penulis Korespondensi: fitri.insani@uin-suska.ac.id

**Abstrak**−Indonesia merupakan salah satu negara dengan tingkat kerentanan tinggi terhadap gempa bumi karena berada di pertemuan tiga lempeng tektonik utama: Indo-Australia, Eurasia, dan Pasifik. Akibat interaksi lempeng ini, frekuensi aktivitas seismik di berbagai wilayah sangat tinggi. Pemahaman pola distribusi gempa menjadi penting untuk mitigasi risiko bencana. Salah satu pendekatan yang digunakan untuk menganalisis pola tersebut adalah metode klasterisasi, khususnya algoritma *DBSCAN* dan *Meanshift* yang mampu mengelompokkan data spasial tanpa menentukan jumlah klaster di awal. Penelitian ini bertujuan membandingkan efektivitas kedua algoritma dalam mengelompokkan data gempa bumi berdasarkan parameter spasial, yaitu *latitude* dan *longitude*. Evaluasi dilakukan melalui visualisasi klaster dan nilai *Silhouette Score* sebagai metrik *valid*itas klaster. Hasil penelitian menunjukkan bahwa *DBSCAN* menghasilkan klasterisasi lebih optimal dengan nilai *Silhouette Score* sebesar 0.930028, lebih tinggi dibandingkan *Meanshift* sebesar 0.90103. *DBSCAN* juga mampu mendeteksi *outlier* yang relevan dalam analisis gempa, sedangkan *Meanshift* menghasilkan lebih banyak klaster namun kurang terpisah. Dengan menggunakan parameter spasial *latitude* dan *longitude*, *DBSCAN* dinilai lebih efektif dalam mengidentifikasi pola distribusi spasial aktivitas seismik di Indonesia berdasarkan data gempa. Penelitian ini mendukung pengembangan sistem pendukung keputusan untuk mitigasi bencana gempa bumi serta menjadi referensi dalam pemilihan metode klasterisasi yang sesuai untuk data spasial.

**Kata Kunci:** Data Mining; DBSCAN; Gempa Bumi; Meanshift; Silhouette Score

**Abstract**−Indonesia is one of the countries with a high vulnerability to earthquakes due to its location at the convergence of three major tectonic plates: the Indo-Australian, Eurasian, and Pacific plates. As a result of this interaction, seismic activity is highly frequent across various regions. Understanding the distribution patterns of earthquakes is essential for disaster risk mitigation. One approach used to analyze these patterns is clustering, particularly using the DBSCAN and Meanshift algorithms, which can group spatial data without predefining the number of clusters. This study aims to compare the effectiveness of both algorithms in clustering earthquake data based on spatial parameters, namely latitude and longitude. Evaluation was conducted using cluster visualization and the Silhouette Score as the clustering validity metric. The results show that DBSCAN produces more optimal clustering with a Silhouette Score of 0.930028, higher than Meanshift's score of 0.90103. DBSCAN is also capable of detecting relevant outliers in earthquake analysis, while Meanshift generates more clusters but with less separation. Using spatial parameters such as latitude and longitude, DBSCAN is considered more effective in identifying the spatial distribution patterns of seismic activity in Indonesia based on earthquake data. This research supports the development of decision support systems for earthquake disaster mitigation and serves as a reference for selecting appropriate clustering methods for spatial data analysis.

**Keywords**: Data Mining; DBSCAN; Earthquake; Meanshift; Silhouette Score

## 1. PENDAHULUAN

Indonesia merupakan salah satu negara dengan tingkat kerentanan tinggi terhadap bencana gempa bumi karena terletak di wilayah pertemuan tiga lempeng tektonik utama, yaitu Lempeng Indo-Australia, Lempeng Eurasia, dan Lempeng Pasifik. Ketiga lempeng ini terus berinteraksi dan menimbulkan tekanan geologis yang signifikan, sehingga menyebabkan tingginya frekuensi aktivitas seismik di berbagai wilayah nusantara[1]. Interaksi antar-lempeng tersebut tidak hanya memicu gempa bumi berskala kecil, tetapi juga gempa besar yang berpotensi menimbulkan tsunami dan kerusakan infrastruktur secara luas[2]. Pemahaman terhadap pola distribusi gempa bumi menjadi sangat penting dalam upaya mitigasi risiko bencana dan perencanaan pembangunan infrastruktur yang lebih tangguh. Analisis data spasial gempa bumi dapat memberikan wawasan yang berharga dalam mengidentifikasi wilayah dengan tingkat kerentanan tinggi[3], Salah satu pendekatan yang umum digunakan dalam analisis eksploratif data spasial adalah metode klasterisasi. Klasterisasi atau *Clustering* memungkinkan pengelompokan kejadian gempa berdasarkan karakteristik tertentu seperti lokasi geografis, kedalaman, dan magnitudo. Melalui proses ini, pola-pola tersembunyi dalam data dapat diidentifikasi, sehingga memberikan pemahaman yang lebih mendalam terhadap sebaran dan intensitas aktivitas seismik di suatu wilayah[4].

Klasterisasi (*Clustering*) merupakan salah satu teknik dalam data mining yang bertujuan untuk mengidentifikasi kelompok objek dengan kemiripan karakteristik, yang secara alami terbagi dari kelompok lain yang memiliki karakteristik berbeda[5]. Objek-objek dalam satu klaster cenderung lebih homogen dibandingkan dengan objek di klaster lain. Jumlah klaster yang terbentuk sangat bergantung pada banyaknya data serta variasi karakteristik objek yang dianalisis. Melalui hasil klasterisasi ini, informasi yang diperoleh dapat digunakan untuk mengungkap pola distribusi gempa bumi yang signifikan dan mendukung pengambilan keputusan dalam mitigasi bencana maupun perencanaan pembangunan wilayah rawan gempa[6]. Berbagai algoritma klasterisasi telah dikembangkan, namun tidak semua algoritma cocok diterapkan untuk jenis data spasial atau geografis. Dua metode yang banyak digunakan untuk data spasial adalah *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)* dan *Meanshift*. Kedua algoritma ini merupakan metode klasterisasi non-hierarkis yang tidak memerlukan penentuan jumlah klaster di awal proses, menjadikannya fleksibel untuk diterapkan pada data yang tidak diketahui struktur klasternya[7][8]. *DBSCAN* bekerja berdasarkan kepadatan data dan mampu

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 554-563
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.605

mengidentifikasi *outlier* (*noise*), sedangkan *Meanshift* memindahkan pusat klaster secara iteratif ke daerah dengan kepadatan data tertinggi, sehingga efektif untuk data dengan distribusi yang tidak beraturan[9].

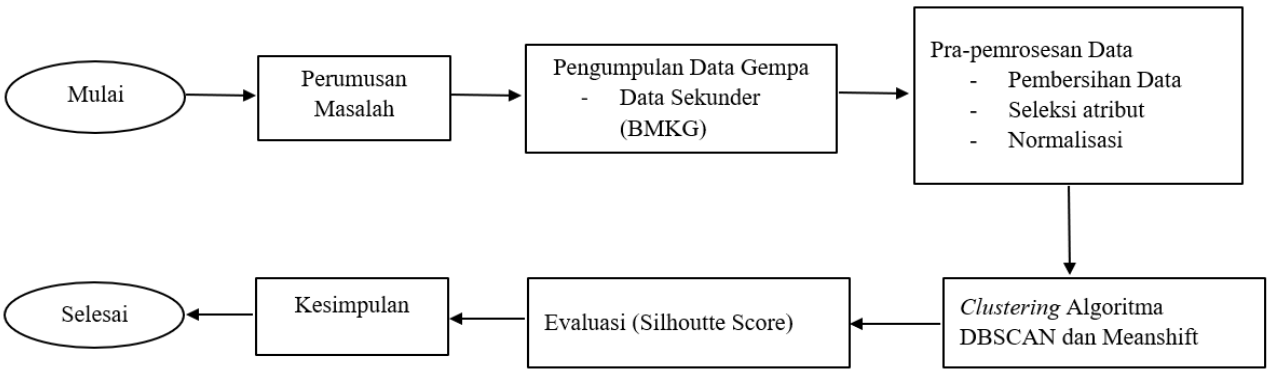
Permasalahan utama dalam penelitian ini adalah bagaimana membandingkan efektivitas algoritma *DBSCAN* dan *Meanshift* dalam mengelompokkan data gempa bumi berdasarkan parameter spasial yaitu *latitude* dan *longitude*. *DBSCAN* memiliki keunggulan dibandingkan *Meanshift* karena mampu mengidentifikasi *noise* atau *outlier* secara eksplisit, serta lebih efektif digunakan pada jumlah data yang besar, sementara *Meanshift* dapat menjadi lebih unggul dalam kondisi data yang lebih sedikit dengan kepadatan titik yang tinggi, maka pada penelitian ini menganalisis dengan parameter *latitude* dan *longitude* saja. Perbandingan ini penting mengingat masing-masing algoritma memiliki karakteristik yang berbeda dalam menangani struktur data dan sebaran spasial. *Silhouette Score* digunakan sebagai tahapan evaluasi dalam analisis klaster karena mampu memberikan ukuran kuantitatif terhadap kualitas dan *valid*itas klaster yang terbentuk. Metode ini mengukur seberapa baik suatu data cocok dengan klasternya sendiri *(intra-cluster similarity)* dibandingkan dengan klaster lainnya *(inter-cluster difference)*, sehingga dapat menunjukkan seberapa jelas batas antar klaster yang dihasilkan. Semakin tinggi nilai *Silhouette Score*, semakin baik objek-objek dalam satu klaster berkumpul dan semakin terpisah dari klaster lain. Selain itu, *Silhouette Score* tidak bergantung pada jenis algoritma klaster tertentu, sehingga fleksibel digunakan pada berbagai pendekatan seperti K-Means, *DBSCAN* , atau hierarchical clustering. Evaluasi ini juga berguna dalam membantu menentukan jumlah klaster yang optimal, dengan memilih jumlah klaster yang menghasilkan nilai *Silhouette Score* tertinggi. Dengan demikian, penggunaan *Silhouette Score* memberikan *valid*asi statistik yang kuat yang melengkapi analisis visualisasi klaster, sehingga menghasilkan evaluasi yang lebih menyeluruh dan objektif[10].

Penelitian ini bertujuan untuk memberikan gambaran yang lebih komprehensif mengenai penerapan algoritma klasterisasi *DBSCAN* dan *Meanshift* pada data gempa bumi di Indonesia. Dengan membandingkan hasil klasterisasi dari kedua algoritma tersebut, diharapkan dapat diperoleh informasi yang lebih objektif mengenai distribusi spasial aktivitas seismik. Ruang lingkup analisis dalam penelitian ini difokuskan pada parameter spasial berupa atribut *latitude* dan *longitude*, yang merepresentasikan lokasi geografis dari setiap kejadian gempa bumi. Hasil penelitian ini diharapkan dapat berkontribusi dalam pengembangan sistem pendukung keputusan untuk mitigasi bencana gempa bumi di Indonesia, serta menjadi referensi dalam pemilihan algoritma klasterisasi yang tepat untuk data spasial serupa.

Penelitian sebelumnya oleh Taufiq et al. (2024) memanfaatkan metode *DBSCAN* untuk memetakan daerah rawan gempa di wilayah Sumatera Barat dan berhasil mengidentifikasi klaster signifikan berdasarkan kepadatan data spasial[4]. Sementara itu, penelitian oleh Cinderatama dan Alhamri (2022) membandingkan *K-Means*, *DBSCAN*, dan *Meanshift* dalam klasifikasi jenis ancaman, dan menyimpulkan bahwa *DBSCAN* dan *Meanshift* lebih unggul dalam menangani distribusi data tidak teratur[11]. Selain itu, Rianti et al. (2024) menerapkan PCA dan algoritma klasterisasi, termasuk *Meanshift*, untuk analisis mutu data spasial dan menunjukkan bahwa pendekatan berbasis densitas memberikan hasil segmentasi yang lebih akurat[12]. Berbeda dari penelitian sebelumnya yang lebih menitikberatkan pada satu algoritma atau menggunakan kombinasi atribut non-spasial, penelitian ini secara spesifik berfokus pada perbandingan dua algoritma non-hierarkis (*DBSCAN* dan *Meanshift*) menggunakan parameter spasial murni (*latitude* dan *longitude*) pada data gempa bumi Indonesia. Kontribusi utama dari penelitian ini adalah memberikan evaluasi empiris berbasis metrik *Silhouette Score*, serta menyajikan analisis visual terhadap distribusi spasial klaster gempa, yang diharapkan dapat memperkuat dasar pengambilan keputusan dalam mitigasi bencana.

## 2. METODOLOGI PENELITIAN

Metodologi penelitian merupakan suatu kerangka sistematis yang disusun untuk mengarahkan jalannya penelitian guna mencapai tujuan yang telah ditetapkan. Pada penelitian ini, metodologi dirancang secara runtut dan terstruktur untuk mendukung proses analisis data gempa bumi di Indonesia menggunakan algoritma klasterisasi *DBSCAN* dan *Meanshift*. Adapun tahapan metodologi dalam penelitian ini dijelaskan sebagai berikut:




**Gambar 1.** Metodologi Penelitian

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 554-563
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.605

Adapun penjelasan dari alur kerja penelitian yang ditampilkan pada Gambar 1 adalah sebagai berikut:

a. Mulai
   Penelitian diawali dengan menetapkan tujuan dan arah penelitian secara umum, yang menjadi dasar bagi seluruh proses selanjutnya.
b. Perumusan Masalah
   Tahap ini berfokus pada identifikasi permasalahan utama yang ingin diselesaikan, yakni bagaimana mengelompokkan data gempa bumi berdasarkan karakteristik spasial dan parameter lainnya untuk memahami pola distribusinya.
c. Pengumpulan Data Gempa
   Data yang digunakan dalam penelitian ini berasal dari sumber sekunder, yaitu Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). Data tersebut mencakup informasi lokasi (lintang, bujur), kedalaman, dan magnitudo gempa bumi yang terjadi di wilayah Indonesia.
d. Pra-pemrosesan Data
   Pada tahap ini dilakukan beberapa proses penting untuk mempersiapkan data sebelum dianalisis, yaitu:
   1. Pembersihan Data: Menghilangkan data duplikat, data kosong, atau data tidak relevan.
   2. Seleksi Atribut: Memilih atribut-atribut penting yang akan digunakan dalam proses klasterisasi.
   3. Normalisasi: Melakukan skala ulang data numerik agar berada dalam rentang yang seragam, sehingga menghindari dominasi atribut tertentu dalam proses pengelompokan.
e. Penerapan Algoritma *DBSCAN* dan *Meanshift*
   Setelah data siap, dua algoritma klasterisasi non-hierarkis, yaitu *DBSCAN* dan *Meanshift*, diterapkan untuk mengelompokkan data berdasarkan pola distribusi spasial. *DBSCAN* mengelompokkan data berdasarkan kepadatan, sedangkan *Meanshift* mencari pusat-pusat kepadatan data untuk membentuk klaster.
f. Evaluasi (*Silhouette Score*)
   Hasil dari masing-masing metode klasterisasi dievaluasi menggunakan *Silhouette Score*, yaitu metrik evaluasi yang mengukur seberapa baik suatu data cocok dengan klasternya sendiri dibandingkan dengan klaster lainnya. Nilai ini menjadi indikator kualitas klaster yang terbentuk.
g. Kesimpulan
   Berdasarkan hasil evaluasi, diambil kesimpulan mengenai metode yang memberikan hasil klasterisasi terbaik serta implikasi dari distribusi klaster gempa bumi terhadap pemetaan risiko dan mitigasi bencana.
h. Selesai
   Tahapan akhir menandai selesainya proses penelitian secara menyeluruh.

### 2.1 *Clustering*

Klasterisasi merupakan salah satu teknik dalam data mining yang bertujuan untuk mengelompokkan sekumpulan data ke dalam kelompok (klaster) berdasarkan kemiripan atau kedekatan karakteristik antar data. Objek dalam satu klaster memiliki karakteristik yang lebih mirip dibandingkan dengan objek pada klaster lain[13]. Metode ini bersifat unsupervised learning, karena tidak memerlukan label kelas untuk proses pelatihan, sehingga sangat cocok digunakan untuk eksplorasi pola tersembunyi dalam data[14].

Dalam konteks data spasial gempa bumi, klasterisasi memungkinkan identifikasi pola distribusi lokasi kejadian gempa berdasarkan atribut seperti koordinat geografis (lintang dan bujur), magnitudo, dan kedalaman. Melalui penerapan metode klasterisasi, wilayah dengan intensitas seismik yang tinggi dapat dipetakan secara lebih akurat, yang berguna dalam penyusunan strategi mitigasi bencana[15]. Beberapa algoritma yang umum digunakan dalam klasterisasi antara lain *K-Means*, *DBSCAN* , *Meanshift*, dan *Agglomerative Clustering*[16]. Namun, tidak semua algoritma cocok untuk data spasial. Algoritma berbasis kepadatan seperti *DBSCAN* dan *Meanshift* lebih unggul dalam menangani bentuk klaster yang tidak beraturan serta *outlier*[17][18].

### 2.2 Algoritma *DBSCAN*

*DBSCAN (Density-Based Spatial Clustering of Applications with Noise)* adalah algoritma klasterisasi berbasis kepadatan yang mampu membentuk klaster berdasarkan area yang memiliki kepadatan tinggi, serta mengidentifikasi *outlier* sebagai *noise*[19]. *DBSCAN* memerlukan dua parameter utama, yaitu ε (epsilon) yang merepresentasikan radius atau jarak maksimum antar titik, dan MinPts sebagai jumlah minimum titik dalam radius tersebut untuk membentuk sebuah klaster. Titik-titik yang berada dalam kepadatan cukup akan diklasifikasikan sebagai *core point*, sedangkan titik-titik lain bisa menjadi border point atau *outlier*[20].

Dalam penerapan algoritma *DBSCAN*, penentuan parameter ε (epsilon) dan MinPts sangat krusial karena akan memengaruhi jumlah dan bentuk klaster yang terbentuk. Untuk menentukan nilai ε yang optimal, penelitian ini menggunakan metode *k-distance graph*, di mana jarak ke tetangga ke-k untuk setiap titik dihitung dan diurutkan. Kemudian grafik jarak tersebut diplot dan titik *'elbow'* atau belokan tajam digunakan sebagai nilai estimasi terbaik untuk ε. Nilai MinPts ditentukan berdasarkan rekomendasi umum, yaitu minimal sebesar ln(n) dengan n adalah jumlah data, atau ditentukan melalui eksperimen beberapa nilai dan evaluasi berdasarkan nilai *Silhouette Score*. Pendekatan ini memungkinkan pengujian sensitivitas hasil terhadap kombinasi parameter yang digunakan, sehingga dapat diperoleh hasil klasterisasi yang optimal.

Salah satu keunggulan utama *DBSCAN* adalah kemampuannya membentuk klaster dengan bentuk arbitrer serta tidak perlu menentukan jumlah klaster di awal seperti pada *K-Means*. Hal ini membuat *DBSCAN* sangat efektif untuk

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 554-563
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.605

data spasial seperti data gempa bumi yang tidak selalu memiliki pola sebaran yang simetris[21]. Secara lebih detail konsep dasar dari algoritma *Clustering DBSCAN* adalah sebagai berikut[22]:

a. Menentukan parameter Eps dan MinPts .
b. Epsilon (ε) dan MinPts perlu ditentukan berdasarkan data. Nilai yang terlalu kecil untuk ε mungkin menghasilkan terlalu banyak *cluster* kecil, sementara nilai yang terlalu besar bisa menggabungkan kluster yang berbeda menjadi satu.
c. Input data yang akan dianalisis.
d. Menghitung jumlah data yang ditentukan oleh parameter radius (Eps). Jika jumlahnya mencukupi (lebih dari atau sama dengan ε), data akan ditandai sebagai inti (*core point*).
e. Menghitung jarak *core point* dengan point yang lain menggunakan jarak *Euclidean*.
Berikut adalah rumus jarak *Euclidean* yang ditujukan pada persamaan sebagai berikut.

$$d(P,C) = \sqrt{\sum_{i=1}^{n}(xpi - xci)^2} \quad (1)$$

Jarak *Euclidean* antara dua titik data, misalnya titik $P$ dan titik pusat klaster $C$, dihitung sebagai $d(P,C)$, yang merepresentasikan jarak langsung di ruang berdimensi $n$. Dalam perhitungan ini, $xpi$ menyatakan nilai fitur ke-$i$ pada titik $P$, sedangkan $xci$ merupakan nilai fitur ke-$i$ pada titik pusat klaster $C$. Jumlah dimensi data yang digunakan dalam perhitungan dilambangkan dengan $n$, sehingga setiap komponen fitur dibandingkan satu per satu untuk menghasilkan total jarak antara kedua titik tersebut.

f. Buat *cluster* baru dengan menambahkan data ke dalam *cluster*.
g. Melakukan identifikasi pada data yang ditandai sebagai *core point*
h. Melanjutkan proses sampai semua point telah diproses.
i. Jika ada data yang tidak masuk ke dalam *cluster* manapun akan ditandai sebagai *noise*.

### 2.3 Algoritma *Meanshift*

*Meanshift* adalah algoritma klasterisasi berbasis kepadatan yang bekerja dengan cara menggeser titik pusat *(centroid)* ke arah area dengan densitas tertinggi berdasarkan *kernel density estimation* (KDE)[11]. Proses ini dilakukan secara iteratif hingga pusat klaster stabil. Tidak seperti *K-Means*, *Meanshift* tidak memerlukan penentuan jumlah klaster di awal dan sangat fleksibel dalam menemukan jumlah klaster secara otomatis berdasarkan distribusi data[12].

Sementara itu, untuk algoritma *Meanshift*, penentuan nilai *bandwidth* (h) dilakukan secara otomatis dengan menggunakan fungsi estimate_*bandwidth*() dari pustaka *Scikit-learn*. Fungsi ini menghitung *bandwidth* berdasarkan estimasi distribusi data dan menggunakan parameter kuantil tertentu untuk mengatur sensitivitas pencarian tetangga. Selain estimasi otomatis, penelitian ini juga melakukan pengujian beberapa nilai *bandwidth* untuk melihat pengaruhnya terhadap hasil klasterisasi dan nilai *Silhouette Score*. Dengan demikian, pemilihan parameter pada kedua algoritma tidak hanya berdasarkan nilai *default*, tetapi melalui proses eksplorasi dan evaluasi sistematis agar hasil yang diperoleh lebih *valid* dan representatif.

Algoritma Mean shift bekerja dengan cara berikut:

a. Tentukan jenis kernel yang akan digunakan serta lebar *bandwidth* (h) yang akan mengatur jangkauan pencarian tetangga. Fungsi kernel.

$$\left(K(x)\right) = \left(\frac{1}{h}\right)\varphi\left(\frac{x}{h}\right) \quad (2)$$

di mana h merupakan *bandwidth* yang mempengaruhi jarak atau radius dari pusat *cluster* saat menghitung perpindahanrata-rata. Fungsi Gaussian atau Epanechnikov (φ) memungkinkan penentuan bobot pada setiap titik data Sesuai denganjaraknya dari pusat *cluster*.

b. Lakukan inisialisasi pusat-pusat massa awal yang akan menjadi *centroid*
c. Pada setiap *centroid*, hitung ulang titik pusat massa baru denganrumus Mean-Shift.

$$m(x) = \frac{\sum(K(x-xi)xi}{\sum K(x-xi)} \quad (3)$$

Perhitungan pusat massa baru $(x)$ dihitung berdasarkan total dari fungsi kernel $((x -))$ dibagi dengan total dari nilai fungsi kernel $(K(x - xi))$ di dalam radius kernel yang diberikan, di mana $x$ adalah pusat massa saat ini dan $xi$ adalah titik data dalam radius kernel.

d. Ulangi langkah perhitungan pusat massa baru untuk setiap *centroid* sampai *centroid* tidak berubah lagi atau memenuhi kriteria berhenti yang telah ditetapkan, seperti mencapai batas jumlah iterasi yang ditentukan.
e. Setelah proses konvergensi, tiap *centroid* akan merepresentasikan mode atau pusat dari kelompok data yang diidentifikasi.

### 2.4 Evaluasi Klasterisasi

Dalam penelitian ini, evaluasi kualitas hasil klasterisasi dilakukan menggunakan metode *Silhouette Score*. *Silhouette Score* merupakan metrik evaluasi yang digunakan untuk mengukur seberapa baik suatu objek data ditempatkan dalam klaster yang sesuai, dengan mempertimbangkan kedekatan objek tersebut dengan klaster lain[23]. Nilai *Silhouette Score*

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 554-563
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.605

berkisar antara -1 hingga 1, di mana nilai mendekati 1 menunjukkan bahwa objek data sangat cocok dengan klaster yang ditentukan dan kurang cocok dengan klaster lainnya[24]. Berikut adalah rumus *Silhoutte Score:*

$$S = \frac{b-a}{\max(a,b)} \quad (4)$$

*Silhouette Score*, yang disimbolkan dengan $S$, merupakan metrik evaluasi yang digunakan untuk menilai kualitas hasil klasterisasi. Dalam perhitungan ini, nilai $a$ merepresentasikan jarak rata-rata antara suatu sampel dengan seluruh titik lain yang berada dalam klaster yang sama. Sementara itu, $b$ menunjukkan jarak rata-rata antara sampel tersebut dengan seluruh titik yang berada di klaster terdekat namun berbeda. Nilai $S$ kemudian dihitung berdasarkan selisih dan rasio dari kedua nilai tersebut, sehingga dapat menggambarkan seberapa baik suatu data cocok berada dalam klasternya dan seberapa terpisah data tersebut dari klaster lain.

# 3. HASIL DAN PEMBAHASAN

Penelitian ini menguraikan proses perbandingan dua algoritma klasterisasi, yaitu *DBSCAN* dan *Meanshift*, dalam mengidentifikasi pola konsentrasi aktivitas gempa bumi di Indonesia. Dataset yang digunakan terdiri dari 600 entri data gempa dengan atribut spasial utama, yaitu *latitude* dan *longitude*. Proses klasterisasi dilakukan setelah data mentah melalui tahap seleksi dan praproses. Algoritma *DBSCAN* digunakan untuk membentuk klaster berdasarkan kepadatan titik data, dengan mempertimbangkan parameter epsilon (ε) dan jumlah minimum tetangga (MinPts). Sementara itu, algoritma *Meanshift* membentuk klaster secara otomatis berdasarkan puncak kepadatan data melalui pendekatan kernel density estimation dan parameter *bandwidth*.

Setelah proses klasterisasi selesai, hasil dari kedua algoritma dibandingkan berdasarkan struktur dan jumlah klaster yang terbentuk, serta dievaluasi menggunakan *Silhouette Score* untuk mengukur kualitas masing-masing hasil. Meskipun keduanya menggunakan data dan parameter spasial yang sama, hasil yang diperoleh menunjukkan perbedaan dalam bentuk, kepadatan, dan pemisahan klaster. Temuan ini menunjukkan bahwa karakteristik masing-masing algoritma berpengaruh terhadap interpretasi data spasial gempa bumi.

Secara umum, *DBSCAN* lebih unggul dalam menangani dataset besar dan kompleks karena kemampuannya dalam mengidentifikasi klaster dengan bentuk tidak beraturan serta mendeteksi outlier secara otomatis. Sementara itu, *Meanshift* lebih sesuai untuk dataset yang relatif kecil dan terdistribusi merata, karena mampu membentuk klaster tanpa perlu menentukan jumlah klaster di awal. Oleh karena itu, dalam penelitian ini dilakukan pengukuran dan evaluasi performa kedua algoritma berdasarkan atribut spasial, untuk menilai efektivitas masing-masing metode dalam mendeteksi konsentrasi wilayah aktivitas gempa di Indonesia.

## 3.1 Lingkungan dan Alat Pengujian

Penelitian ini dilaksanakan menggunakan platform *Google colaboratory (Google colab)*, yaitu layanan komputasi awan berbasis *Jupyter Notebook* yang disediakan oleh *Google. Google colab* dipilih karena mendukung pemrosesan data secara efisien, memiliki akses ke GPU/TPU secara gratis, serta mendukung integrasi langsung dengan *Google Drive*, tempat penyimpanan dataset penelitian.

Bahasa pemrograman yang digunakan adalah Python versi 3.10, dengan pustaka-pustaka pendukung sebagai berikut:

a. Pandas: Untuk membaca, membersihkan, dan memanipulasi data dalam bentuk tabel.
b. NumPy: Untuk perhitungan numerik dan manipulasi array.
c. *Scikit-learn*: Untuk implementasi algoritma klasterisasi seperti *DBSCAN* dan *Meanshift*, serta evaluasi hasil menggunakan *Silhouette Score*.
d. Matplotlib dan Seaborn: Untuk visualisasi data dan hasil klasterisasi dalam bentuk grafik dua dimensi.

Dataset penelitian disimpan dan diakses melalui *Google Drive*, yang di-mount langsung ke lingkungan *Google colab*. Pendekatan ini memungkinkan proses pengolahan data secara efisien dan aman di lingkungan berbasis *cloud*, tanpa bergantung pada kapasitas lokal perangkat keras.

## 3.2 Implementasi

Proses implementasi pada penelitian ini dilakukan dalam beberapa tahapan utama mulai dari pemrosesan awal data, penerapan algoritma klasterisasi, hingga evaluasi performa model. Seluruh tahapan dilakukan menggunakan bahasa pemrograman Python di lingkungan *Google colaboratory*.

a. Import dan Pemanggilan Dataset
   Langkah pertama yang dilakukan adalah mengimpor seluruh pustaka yang dibutuhkan, seperti *pandas, numpy, matplotlib, seaborn, dan Scikit-learn.*

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 554-563
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.605

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import DBSCAN, MeanShift
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import silhouette_score
```

**Gambar 2.** Import Library

Dataset diunggah ke *Google Drive*, kemudian diakses melalui Colab dengan teknik mount menggunakan drive.mount() dari pustaka *google colab*.

```
from google.colab import drive
drive.mount('/content/drive/')

# Membaca dataset dari file Excel yang disimpan di Google Drive
df = pd.read_excel("/content/drive/MyDrive/Colab Notebooks/datasetgempa.xlsx")
```

**Gambar 3.** Mount *Google Drive*

Dataset yang digunakan dibaca menggunakan pustaka pandas dan ditampilkan dengan fungsi head() dan info() untuk eksplorasi awal terhadap struktur dan isi data.

b. Pra-Pemrosesan Data

Tahap pra-pemrosesan data dilakukan untuk membersihkan dan menyesuaikan format data agar dapat digunakan oleh algoritma *Clustering*. Proses ini meliputi konversi seluruh data menjadi tipe string, penghapusan karakter non-numerik, transformasi ke tipe numerik, serta penanganan nilai hilang *(NaN)* dengan menghapus kolom dan baris yang tidak relevan.

```
df_clean = df.copy()
for col in df_clean.columns:
    df_clean[col] = (
        df_clean[col]
        .astype(str)                        # pastikan semuanya string dulu
        .str.replace(r'[^\d\.\-]', '', regex=True)  # hapus semua kecuali angka, minus, titik
        .replace('', np.nan)                # ganti string kosong dengan NaN
    )

    # Ubah ke float
    df_clean[col] = pd.to_numeric(df_clean[col], errors='coerce')

# Drop kolom yang gagal dikonversi semua (isinya NaN semua)
df_clean = df_clean.dropna(axis=1, how='all')

# Drop baris dengan NaN (jika ada)
df_clean = df_clean.dropna()

# Cek hasil konversi
print("\nSetelah konversi:")
print(df_clean.dtypes)
print(df_clean.head())
```

**Gambar 4.** Proses Cleaning

Setelah proses pembersihan selesai, data dinormalisasi menggunakan metode *Min-Max Scaling*. Tujuan dari normalisasi adalah untuk menyamakan skala antar fitur numerik, sehingga algoritma *Clustering* dapat bekerja secara optimal tanpa dipengaruhi oleh perbedaan skala nilai antar kolom.

Proses normalisasi ini dilakukan menggunakan pustaka *MinMaxScaler* dari *Scikit-learn*. Gambar berikut menunjukkan potongan kode yang digunakan dalam proses normalisasi data:

```
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(df_clean)
```

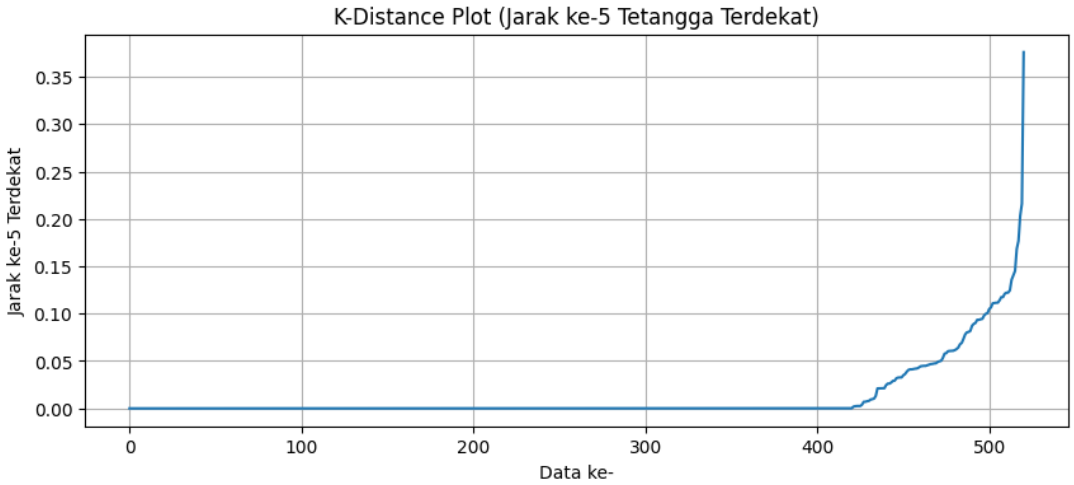
**Gambar 5.** Proses Normalisasi

c. Klasterisasi Menggunakan *DBSCAN*

Sebelum menerapkan algoritma *DBSCAN*, terlebih dahulu dilakukan penentuan nilai parameter ε (epsilon) menggunakan metode k-distance graph. Metode ini bertujuan untuk menentukan nilai ambang batas jarak yang optimal antara titik-titik data berdasarkan kepadatan lokal. Dalam penerapannya, jarak ke tetangga ke-k dari setiap titik data dihitung dan diurutkan. Hasil pengurutan ini kemudian diplot ke dalam bentuk grafik, yang dikenal sebagai

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 554-563
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.605

grafik k-distance. Titik elbow atau belokan tajam pada grafik menunjukkan perubahan signifikan pada gradien, yang secara visual menunjukkan batas antara area dengan kepadatan tinggi dan rendah. Titik ini digunakan sebagai acuan dalam memilih nilai ε karena dianggap paling tepat untuk memisahkan klaster padat dari *noise*.
Gambar 6 berikut menunjukkan hasil visualisasi grafik k-distance yang digunakan untuk menentukan nilai parameter ε dalam penelitian ini:




**Gambar 6.** Grafik *K-distance*

Berdasarkan grafik tersebut, nilai ε (epsilon) ditentukan sebesar 0.2, yaitu tepat sebelum terjadi kenaikan tajam pada grafik. Parameter MinPts (minimum samples) ditentukan menggunakan pendekatan logaritmik ln(n), di mana $n = 600$, menghasilkan nilai mendekati 6. Namun, untuk menjaga sensitivitas terhadap data spasial, nilai MinPts disesuaikan menjadi 5. Setelah parameter ditetapkan, proses klasterisasi dilakukan menggunakan algoritma *Density-Based Spatial Clustering of Applications with Noise* (*DBSCAN*). Implementasi dilakukan menggunakan pustaka *sklearn.cluster* dari *Python*, dengan nilai eps = 0.2 dan min_samples = 5. *DBSCAN* akan mengelompokkan titik-titik data yang memiliki kepadatan tinggi dan secara otomatis mendeteksi outlier (*noise*), yaitu titik yang tidak memiliki cukup tetangga dalam radius ε.
Gambar berikut menampilkan cuplikan kode program dalam proses implementasi *DBSCAN* menggunakan Python:

```
dbscan = DBSCAN(eps=0.2, min_samples=5)
dbscan_labels = dbscan.fit_predict(X_scaled)
```

**Gambar 7.** Proses *DBSCAN*

d. Klasterisasi Menggunakan *Meanshift*
*Meanshift* merupakan algoritma klasterisasi berbasis kepadatan yang tidak memerlukan penentuan jumlah klaster di awal. Algoritma ini bekerja dengan cara menggeser pusat data (*centroid*) secara iteratif menuju area dengan kepadatan tertinggi, menggunakan pendekatan *Kernel Density Estimation* (KDE). Proses ini berlangsung hingga posisi pusat klaster mencapai kondisi konvergen, yaitu ketika tidak ada lagi perubahan signifikan pada lokasi pusat. Karakteristik ini menjadikan *Meanshift* sangat fleksibel dalam mengelompokkan data yang memiliki distribusi tidak seragam.
Dalam penelitian ini, algoritma *Meanshift* diimplementasikan menggunakan pustaka *sklearn.cluster* dari *Python*. Nilai *bandwidth*, yang merupakan parameter utama dalam algoritma ini, dihitung secara otomatis menggunakan fungsi estimate_*bandwidth*() dari *Scikit-learn*. Berdasarkan hasil estimasi, diperoleh nilai *bandwidth* sebesar 0.2, yang kemudian digunakan langsung dalam proses klasterisasi tanpa dilakukan pengujian nilai alternatif lainnya. Nilai ini merepresentasikan radius pencarian tetangga dan sangat berpengaruh terhadap jumlah serta bentuk klaster yang terbentuk.
Gambar berikut menampilkan cuplikan kode program dalam proses implementasi algoritma *Meanshift* menggunakan Python:

```
meanshift = MeanShift(bandwidth=0.2)  # bandwidth=0.2 → ubah sesuai kebutuhan
meanshift_labels = meanshift.fit_predict(X_scaled)
```

**Gambar 8.** Proses *Meanshift*

e. Evaluasi Klasterisasi dengan *Silhouette Score*
Untuk mengevaluasi kualitas hasil klasterisasi, digunakan metrik *Silhouette Score*. Berikut adalah implementasi perhitungan nilai *Silhouette Score* menggunakan Python:

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 554-563
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.605

```
if len(set(dbscan_labels)) > 1 and len(set(dbscan_labels)) - (1 if -1 in dbscan_labels else 0) > 1:
    silhouette_dbscan = silhouette_score(X_scaled, dbscan_labels)
else:
    silhouette_dbscan = "Silhouette tidak valid (hanya 1 klaster atau noise semua)"
```

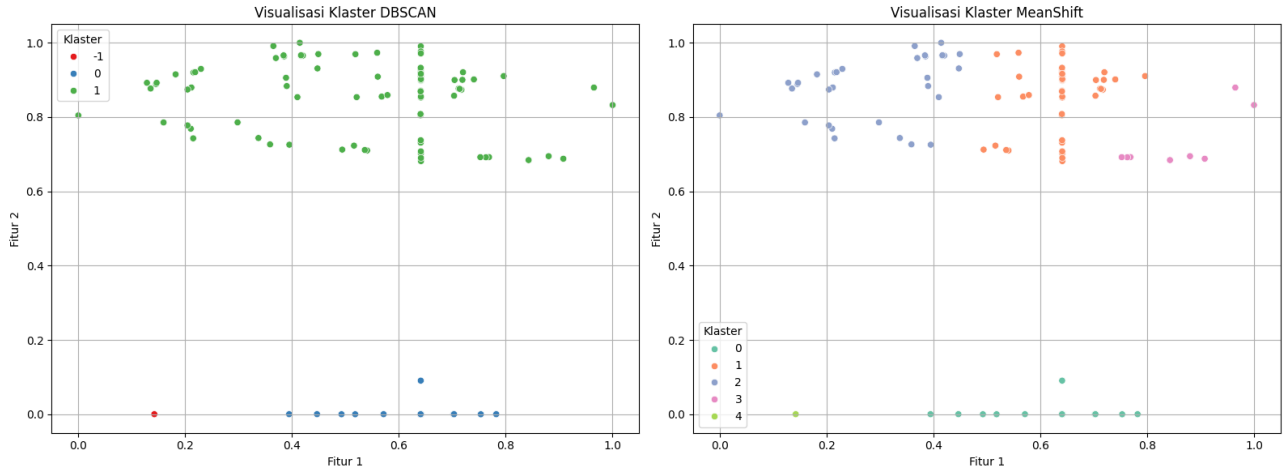
**Gambar 9.** *Silhoutte Score DBSCAN*

```
if len(set(meanshift_labels)) > 1:
    silhouette_meanshift = silhouette_score(X_scaled, meanshift_labels)
else:
    silhouette_meanshift = "Silhouette tidak valid (hanya 1 klaster)"
```

**Gambar 10.** *Silhoutte Score Meanshift*

### 3.3 Hasil Implementasi dan Evaluasi Klasterisasi

Gambar 11 berikut menunjukkan hasil visualisasi klaster yang dibentuk oleh dua metode klasterisasi yang berbeda, yaitu *DBSCAN* dan *Meanshift*. Visualisasi ini memberikan gambaran mengenai bagaimana masing-masing metode mengelompokkan data ke dalam klaster yang berbeda berdasarkan distribusi data.




**Gambar 11.** Hasil Klaster *DBSCAN* dan *Meanshift*

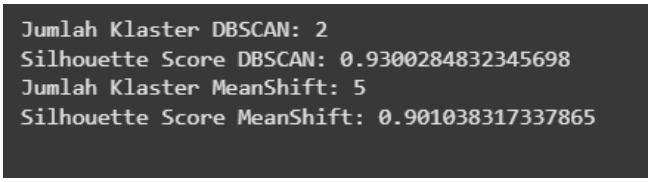
Hasil klasterisasi menggunakan algoritma *DBSCAN* menunjukkan terbentuknya dua klaster utama, serta sejumlah data yang terdeteksi sebagai outlier. Secara geografis, Klaster 0 mengelompokkan titik-titik gempa yang terkonsentrasi di sepanjang zona subduksi Sumatera dan Jawa, wilayah yang memang dikenal memiliki aktivitas seismik tinggi. Sementara itu, Klaster 1 merepresentasikan area aktivitas gempa di wilayah Maluku dan Sulawesi, yang juga merupakan zona kompleks pertemuan lempeng aktif. Data yang dikategorikan sebagai outlier (label -1) sebagian besar merupakan gempa bumi terisolasi, yang muncul di wilayah seperti Kalimantan atau perairan terpencil. Titik-titik ini menarik untuk dianalisis lebih lanjut karena menyimpang dari pola utama dan tidak tergabung dalam klaster manapun.

Sementara itu, hasil klasterisasi menggunakan algoritma *Meanshift* menunjukkan terbentuknya tiga klaster, tanpa adanya deteksi outlier. Secara spasial, Klaster 0 mencakup wilayah Sumatera bagian selatan dan barat Jawa, Klaster 1 mencakup aktivitas di wilayah timur Indonesia seperti Maluku dan Papua, dan Klaster 2 mengelompokkan titik-titik di wilayah Sulawesi dan sekitarnya. Meskipun menghasilkan jumlah klaster yang lebih banyak dibandingkan *DBSCAN*, hasil *Meanshift* menunjukkan bahwa pemisahan antar klaster kurang jelas secara visual, dengan beberapa klaster tampak tumpang tindih, terutama di wilayah peralihan zona lempeng.

Dari sisi evaluasi kuantitatif, *Meanshift* menghasilkan nilai Silhouette Score sebesar 0.901038, yang mengindikasikan bahwa hasil klasterisasi tergolong baik. Namun, pemisahan antar klaster tidak sejelas *DBSCAN*, terutama karena algoritma ini memindahkan *centroid* berdasarkan kepadatan lokal, tanpa mempertimbangkan batasan *noise*. Akibatnya, *Meanshift* cenderung membentuk lebih banyak klaster pada area dengan variasi kepadatan kecil, meskipun secara visual hasilnya kurang terstruktur.

Sebaliknya, *DBSCAN* menghasilkan Silhouette Score lebih tinggi, yaitu sebesar 0.930028, yang menandakan bahwa pemisahan antar klaster sangat jelas dan klaster yang terbentuk memiliki kepadatan internal yang tinggi. Keunggulan ini sesuai dengan karakteristik *DBSCAN* yang mampu menangani *noise*, mengenali bentuk klaster arbitrer, dan tidak terpengaruh oleh distribusi *non-linear*. Dengan membentuk klaster berdasarkan kepadatan lokal dan mengabaikan titik-titik outlier, *DBSCAN* mampu memisahkan wilayah-wilayah seismik aktif secara alami, tanpa memaksakan bentuk atau jumlah klaster tertentu. Hasil ini menjadikan *DBSCAN* lebih representatif dalam mengelompokkan data spasial gempa bumi di Indonesia.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 5, No 4, June 2025 | Hal 554-563
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v5i4.605




**Gambar 12.** Hasil *Silhoutte Score*

Hasil evaluasi menunjukkan bahwa:

a. *DBSCAN* menghasilkan *Silhouette Score* sebesar 0.930028, yang menandakan bahwa kualitas pemisahan klaster cukup baik, dengan jarak antar klaster yang cukup jelas.
b. *Meanshift* menghasilkan *Silhouette Score* sebesar 0.901038, yang juga menunjukkan pemisahan klaster yang cukup baik, meskipun sedikit lebih rendah dibandingkan *DBSCAN*.
c. Dari hasil ini, dapat disimpulkan bahwa kedua metode memiliki performa yang cukup sebanding dalam mengelompokkan data, namun *DBSCAN* sedikit lebih unggul dalam hal kualitas klasterisasi berdasarkan nilai *Silhouette Score*.

Berdasarkan nilai evaluasi tersebut, *DBSCAN* unggul dengan *Silhouette Score* lebih tinggi sebesar 0.930028 dibandingkan *Meanshift* yang memperoleh 0.901038. Selisih sebesar 0.029 atau hampir 3% ini dapat dianggap signifikan dalam konteks data spasial, terutama ketika hasil klasterisasi digunakan untuk keperluan penting seperti pengambilan keputusan mitigasi bencana. *DBSCAN* mampu menghasilkan klaster yang lebih kompak, terpisah dengan jelas, dan secara efektif mengabaikan *noise*, sehingga hasil analisis lebih mudah untuk diinterpretasikan.

Di sisi lain, *Meanshift* tetap menjadi pilihan yang baik untuk data dengan struktur yang lebih halus dan sedikit outlier. Namun, untuk jenis data seperti aktivitas gempa bumi yang memiliki distribusi tidak merata dan kemungkinan *noise* tinggi, *DBSCAN* terbukti lebih efektif. Dengan demikian, perbedaan nilai evaluasi ini memperkuat klaim bahwa *DBSCAN* lebih cocok diterapkan pada data spasial gempa bumi di Indonesia.

## 4. KESIMPULAN

Berdasarkan hasil implementasi dan evaluasi metode klasterisasi *DBSCAN* dan *Meanshift*, dapat disimpulkan bahwa kedua algoritma mampu mengelompokkan data dengan cukup baik. Visualisasi hasil klasterisasi menunjukkan bahwa kedua metode berhasil membentuk klaster yang cukup jelas dan terpisah. Evaluasi menggunakan *Silhouette Score* menunjukkan bahwa metode *DBSCAN* memiliki nilai skor sebesar 0.930028, sedikit lebih tinggi dibandingkan dengan *Meanshift* yang memiliki nilai 0.901038. Hal ini mengindikasikan bahwa hasil klasterisasi *DBSCAN* memiliki kualitas pemisahan antar klaster yang lebih baik. Selain itu, *DBSCAN* juga memiliki keunggulan dalam mendeteksi *outlier* atau data yang tidak termasuk dalam klaster manapun, yang dapat berguna dalam proses analisis lebih lanjut. Sementara itu, *Meanshift* menghasilkan klaster yang lebih banyak tanpa adanya deteksi *outlier*, sehingga lebih cocok digunakan pada data yang cenderung bersih dan terdistribusi secara merata. Secara keseluruhan, *DBSCAN* lebih unggul dalam konteks dataset ini, baik dari segi nilai evaluasi maupun kemampuan identifikasi *outlier*, meskipun perbedaan performanya tidak terlalu signifikan.

## REFERENCES

[1] R. R. A. Rahman and A. W. Wijayanto, "Pengelompokan Data Gempa Bumi Menggunakan Algoritma *DBSCAN*," *J. Meteorol. dan Geofis.*, vol. 22, no. 1, p. 31, 2021, doi: 10.31172/jmg.v22i1.738.
[2] Admin, "Pengertian Bencana," *Dinas Sosial*, 2021. https://dinsos.bulelengkab.go.id/informasi/detail/artikel/pengertian-bencana-35#:~:text=Gempa bumi tektonik,kecil hingga yang sangat besar.
[3] S. farisi Y, "Analisis Tingkat Kerentanan Fisik Dan Sosial Bencana Gempabumi Di" Swara Bhumi, vol. v, no. 9, 2020.
[4] R. M. Taufiq, R. Firdaus, F. Handayani, P. F. Muarif, and R. R. Rizqy, "Density-Based Clustering untuk Pemetaan Daerah Rawan Gempa Bumi di Wilayah Sumatera Barat Menggunakan Metode *DBSCAN*," Jurnal Fasilkom, vol. 14, no. 3, pp. 817–822, 2024.
[5] M. Firdaus, "Data Mining," 2023.
[6] I. N. Simbolon and P. D. Friskila, "Analisis Dan Evaluasi Algoritma *Dbscan* (Density-Based Spatial Clustering Of Applications With *Noise*) Pada Tuberkulosis." JITET, vol. 12, no. 3, 2024.
[7] Sachinsoni, "Clustering Like a Pro: A Beginner's Guide to *DBSCAN*," *Medium*, 2023. https://medium.com/%40sachinsoni600517/clustering-like-a-pro-a-beginners-guide-to-*DBSCAN*-6c8274c362c4
[8] "*Meanshift*," *Wikipedia*. https://en.wikipedia.org/wiki/Mean_shift?utm_source=chatgpt.com
[9] Admin, "Kupas Tuntas Algoritma Data Science dengan Mean-Shift Algorithm," *DQLab*, 2021. https://dqlab.id/kupas-tuntas-algoritma-data-science-dengan-mean-shift-algorithm
[10] A. N. Tahiyat, B. Maulana, A. E. Saputra, and L. Efrizoni, "Klasterisasi Lagu Populer dan Eksplorasi Subgenre Spotify 2024 dengan K-Medoids," 2025.
[11] T. A. Cinderatama and Y. Y. , Rinanza Zulmy Alhamri, "Implementasi Metode K-Means, *DBSCAN*, dan *Meanshift* untuk analisis jenis ancaman," *J. INOVTEK POLBENG - SERI Inform.*, vol. 7, no. 1, 2022.
[12] R. Rianti, R. Andarsyah, and R. M. Awangga, "Penerapan PCA dan Algoritma Clustering untuk Analisis Mutu.pdf," *NUANSA Inform.*, 2024.
[13] D. R. S. S. Wijayatia, Rika, "Clustering Data Campuran Numerik Dan Kategorik," PRISMA, vol 6, 2023.
[14] A. M. Afinda, "Supervised vs Unsupervised Learning: Mana yang Paling Cocok untuk Data Kamu?," 2024.

https://www.dicoding.com/blog/supervised-vs-unsupervised-learning-mana-yang-sesuai-untuk-data-kamu/
[15] C. Ramadhani and I. M. B. Suksmadana, "Pengklasteran Kejadian Gempa Wilayah Indonesia Menggunakan Algoritma k-Means," *Dielektrika*, vol. 8, no. 2. pp. 62–67, 2022
[16] C. K. Saputra, "Perbandingan Hasil Segmentasi Citra dengan Metode K-means, Agglomerative Hierarchical, dan *DBSCAN*." 2023
[17] A. N. A. Maulidhia, R. Basya, Indri Ika Widyastuti Friska Intan Sukarno, and S. T. T. Brian, "Implementasi Perbandingan Algoritma k-Means dan DB-Scan Pada Beban Listrik Rumah Tangga.pdf." INTEGER: Journal of Information Technology, 2025.
[18] M. I. Awal Lidya Musaffak, Kartika Maulida Hindrayani, "Penerapan Metode Mean Shift Clustering Untuk Mengelompokkan Wilayah Berdasarkan Pengelolaan Sampah," JATI (Jurnal Mahasiswa Teknik Informatika), vol. 9, no. 3, 2025.
[19] Z. L. M. Nabila Dea Azahraa, M. F. Farizqia, Iqbal, and Kharisudina, "Segmentasi Pelanggan dengan Algoritma *DBSCAN* dan KMeans.pdf," *Prism. Pros. Semin. Nas. Mat.*, 2025.
[20] J. Riyono, C. E. Pujiastuti, A. L. Riyana, And Putri, "Clustering Negara Berdasarkan Skor Pengendalian Konsumsi Tembakau," *J. Tek. Inform. Kaputama*, vol. 8, no. 1, 2024.
[21] C. Alkahfi, "Pengenalan Algoritma Clustering *DBSCAN*," *sains.data*, 2024. https://sainsdata.id/machine-learning/13080/pengenalan-algoritma-clustering-*DBSCAN*/#:~:text=Keunggulan dan Keterbatasan *DBSCAN*,hasil berbeda tergantung titik awal
[22] R. Azwarini, "Metode Clustering *DBSCAN* (Density-Based Spatial Clustering of Applications with *Noise*)," *exsight.id*. https://exsight.id/blog/2024/10/30/clustering-*DBSCAN*/
[23] RIZUAN, "Penerapan Algoritma Mean-Shift Pada Clustering Penerima Bantuan Pangan Non Tunai Tugas Akhir," *J. Comput. Syst. Informatics*, 2023.
[24] T. Joshi, "Evaluating Clustering Algorithm — *Silhouette Score*," *Medium*, 2021. https://tushar-joshi-89.medium.com/silhouette-score-a9f7d8d78f29