

PENGELOMPOKAN DATA GEMPA BUMI MENGGUNAKAN ALGORITMA DBSCAN

GROUPING EARTHQUAKES DATA USING DBSCAN ALGORITHM

Raisa Rizky Amelia Rahman^{1*}, Arie Wahyu Wijayanto²

¹D-IV Statistika, Politeknik Statistika STIS, Jln. Otto Iskandardinata No. 64C Jatinegara, Jakarta Timur 13330

²Badan Pusat Statistik, Jl. Dr. Sutomo No. 6-8 Sawah Besar, Jakarta Pusat 10710

*E-mail: 211709955@stis.ac.id

Naskah masuk: 18 Januari 2021 Naskah diperbaiki: 3 September 2021 Naskah diterima: 8 September 2021

ABSTRAK

Gempa bumi merupakan bencana alam yang tidak dapat dicegah maupun dihindari. Oleh sebab itu, perlu dilakukan pengelompokan data gempa untuk mendukung upaya minimalisasi dampak yang ditimbulkan. Data yang digunakan dalam penelitian ini adalah data gempa bumi di Indonesia yang bersumber dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). Penelitian ini menggunakan algoritma DBSCAN dalam mengelompokkan data ke dalam beberapa *cluster*. Metode untuk menguji validitas hasil *cluster* adalah dengan menggunakan *Silhouette Coefficient* dan *Gamma Index*. Hasil *clustering* pada penelitian ini memberikan kesimpulan bahwa dengan menggunakan algoritma DBSCAN diperoleh 3 *cluster* wilayah beresiko terjadi gempa bumi berdasarkan karakteristik parameter gempa bumi yang dihasilkan. Kombinasi nilai ϵ dan *MinPts* yaitu 0,28 dan 3 menghasilkan nilai *Silhouette Coefficient* sebesar 0,81091 dan Indeks Gamma sebesar 0,98104 yang menggambarkan bahwa DBSCAN mampu mengelompokkan wilayah beresiko terjadi gempa bumi dengan cukup baik. Hasil penelitian ini dapat digunakan sebagai bahan pertimbangan suatu instansi dalam pengambilan keputusan terkait penanganan (mitigasi) bencana gempa bumi.

Kata kunci: *clustering*, DBSCAN, *gamma index*, gempa bumi, *silhouette coefficient*

ABSTRACT

Earthquakes are natural disasters that cannot be prevented or avoided. Therefore, it is necessary to group the earthquake data as an effort to minimize the impact. The data used in this research were earthquake data in Indonesia sourced from the Meteorology, Climatology, and Geophysics Agency. This study used the DBSCAN algorithm to grouped data into several clusters. The method for testing the validity of the cluster results was *Silhouette Coefficient* and *Gamma Index*. The results of clustering in this study concluded that using the DBSCAN algorithm, 3 clusters of areas with the risk of earthquakes were obtained based on the characteristics of the earthquake parameters. The combination of ϵ and *MinPts* values, namely 0.28 and 3, results in a *Silhouette Coefficient* value of 0,81091 and *Gamma Index* value of 0,98104 indicated that DBSCAN can categorize areas that have the risk for earthquakes quite well. This study's results use as material for an agency's consideration in making decisions related to earthquake disaster mitigation.

Keywords: *clustering*, DBSCAN, *gamma index*, earthquake, *silhouette coefficient*

1. Pendahuluan

Indonesia terletak antara tiga pertemuan lempeng besar, yaitu Lempeng Eurasia, Lempeng Indo-Australia, dan Lempeng Pasifik [1]. Pertemuan dan tumbukan yang terjadi antar lempeng, misalnya Lempeng Eurasia dan Lempeng Indo-Australia, dapat menimbulkan getaran-getaran di permukaan bumi yang kemudian disebut sebagai gempa bumi.

Berdasarkan data yang diperoleh dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG), gempa bumi selalu terjadi di Indonesia setiap hari, namun tidak semuanya dapat dirasakan oleh

penduduk karena magnitudenya kecil dan pusat gempa jauh dari pemukiman penduduk [2]. BMKG juga menyebutkan bahwa Pulau Bali dan sekitarnya merupakan wilayah seismotektonik Indonesia yang dilalui jalur Pegunungan Mediterania dan adanya zona subduksi akibat pertemuan antara Lempeng Eurasia dan Lempeng Indo-Australia. Batas pertemuan ini berupa palung lautan (*Oceanic Trench*) di sebelah selatan gugusan pulau Jawa, Bali dan Nusa Tenggara [3].

Gempa bumi merupakan bencana alam yang tidak dapat dicegah maupun dihindari. Oleh sebab itu, perlu dilakukan pengelompokan wilayah gempa untuk

mendukung upaya minimalisasi dampak yang ditimbulkan [2]. Wilayah terjadinya gempa bumi dapat dikelompokkan dengan memanfaatkan data-data yang diperoleh dari peristiwa gempa bumi. Selanjutnya, dapat dilakukan analisis untuk mengetahui karakteristik hasil pengelompokan wilayah gempa bumi menggunakan teknik clustering. Menurut Tan [4], *clustering* adalah suatu proses untuk mengelompokkan data ke dalam beberapa *cluster* sehingga data dalam satu *cluster* memiliki tingkat kemiripan yang maksimum dan data antar *cluster* memiliki kemiripan yang minimum.

Pada *Data Mining*, terdapat banyak algoritma *clustering* yang populer, salah satunya adalah *Density-Based Spatial Clustering of Application with Noise* (DBSCAN). DBSCAN merupakan algoritma yang masuk dalam kategori *density-based clustering*, yaitu proses pembentukan *cluster* dilakukan berdasarkan tingkat kedekatan/kerapatan jarak antar objek dalam dataset tersebut [5]. Penelitian ini menggunakan algoritma DBSCAN dengan beberapa pertimbangan, antara lain kemampuannya dalam mendeteksi *outlier/noise*, tidak perlu mengetahui jumlah *cluster* (k) yang akan terbentuk, dan mampu mengenali bentuk *cluster* yang tidak beraturan. Selain itu, mengacu pada penelitian sebelumnya oleh Kryszkiewicz dan Skonieczny (2005), Ali et al. (2010), dan Schubert et al. (2017), menunjukkan bahwa DBSCAN dengan indeks efektif dan nilai parameter yang wajar, bekerja secara kompetitif dibandingkan dengan metode lain yang lebih *advance*, yang mana semua indeks memberikan kecepatan *runtime* 10 hingga 1.000 kali lipat [6,7,8]. Tujuan dari penelitian ini adalah melakukan pemetaan dan pengelompokan wilayah gempa bumi di Indonesia berdasarkan karakteristik parameter gempa bumi yang dihasilkan, agar suatu hari dapat digunakan sebagai bahan rekomendasi mitigasi bencana gempa bumi terhadap institusi terkait. Algoritma DBSCAN digunakan oleh Muhammad Tanzil Furqon dan Lailil Muflikhah [5] pada *Clustering the Potential Risk of Tsunami Using Density-Based Spatial Clustering of Application with Noise* (DBSCAN). Penelitian tersebut menunjukkan bahwa algoritma DBSCAN dapat diimplementasikan untuk mengelompokkan potensi resiko tsunami ke dalam 81 kelompok. *Clustering* yang dilakukan menggunakan algoritma DBSCAN menghasilkan hasil yang cukup bagus ditandai dengan *Silhouette Coefficient* sebesar 0,96 pada proses pengujian.

Penelitian Scitovski [9] berjudul *A Density-Based Clustering Algorithm for Earthquake Zoning* menunjukkan bahwa kelebihan algoritma DBSCAN untuk zonasi gempa dapat dilihat pada kemampuannya mengenali *cluster* non-konveks. Selain itu, diperoleh hasil bahwa partisi dengan jumlah *cluster* yang paling tepat diperoleh secara otomatis tanpa menggunakan indeks.

Harini et al [10] dalam penelitiannya yang berjudul *The Earthquake Events and Impacts Mapping in Bali*

and Nusa Tenggara Using A Clustering Method didapatkan pemetaan zona potensi gempa pada RC 3 Indonesia. Pemetaan ini menggunakan pendekatan algoritma *clustering* DBSCAN. Hasil percobaan menunjukkan bahwa pengelompokan dilakukan secara efektif dari segi waktu pelaksanaan dan dengan hasil validitas *cluster* yang baik yang ditunjukkan dengan 3 ukuran validitas *cluster*.

2. Metode Penelitian

Sumber Data. Data yang digunakan dalam penelitian ini adalah data gempa bumi di Indonesia yang bersumber dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). Data tersebut terdiri dari 1.038 kejadian gempa dengan 4 variabel. Data yang digunakan dalam proses pengujian dibatasi hanya data gempa bumi yang terjadi selama 1 bulan terakhir, yaitu dari tanggal 5 November sampai dengan 5 Desember 2020. Pada *dataset*, terdapat empat variabel, antara lain:

Tabel 1. Deskripsi Variabel

Variabel	Satuan	Tipe Data
<i>Latitude</i>	Derajat	Numerik
<i>Longitude</i>	Derajat	Numerik
<i>Focal Depth</i>	Kilometer	Numerik
<i>Magnitude</i>	Skala Richter	Numerik

Data Preprocessing. *Preprocessing* data diperlukan untuk memastikan bahwa data telah sesuai agar dapat menghasilkan analisis yang tepat. Tahap perhitungan jarak antar titik data pada algoritma DBSCAN menggunakan fungsi *Euclidean Distance* sehingga perbedaan rentang nilai antar variabel sangat memengaruhi hasil *clustering*. Oleh sebab itu, pada penelitian ini dilakukan proses normalisasi untuk menyamakan rentang nilai seluruh variabel dengan menggunakan fungsi normalisasi *Z-Score* dengan persamaan sebagai berikut:

$$zscore = \frac{x - \mu_x}{\sigma_x} \quad (1)$$

dimana *zscore* merupakan nilai hasil normalisasi, μ_x nilai rata-rata untuk variabel x, dan σ_x simpangan baku variabel x.

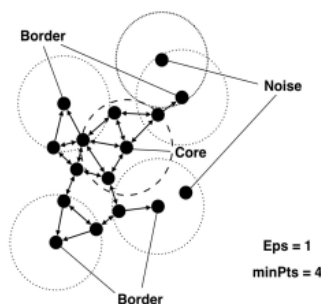
Clustering. *Clustering* adalah tipe dari *unsupervised learning* yang paling umum digunakan dalam *data mining*. *Clustering* dilakukan untuk mengatur data ke dalam himpunan yang terbatas dari kelompok yang konsisten secara semantik berdasarkan perhitungan kemiripan [11]. Proses *clustering* bertujuan untuk mengelompokkan data yang memiliki karakteristik yang sama dalam satu klaster yang sama dan data yang memiliki karakteristik berbeda dikelompokkan ke klaster lain [12].

Algoritma DBSCAN. *Density-Based Spatial Clustering of Application with Noise* (DBSCAN) adalah algoritma untuk mengelompokkan data berdasarkan kepadatan. Konsep kepadatan yang dimaksud dalam DBSCAN adalah jumlah minimal data dalam radius ϵ , data tersebut termasuk dalam kategori kepadatan yang diinginkan.

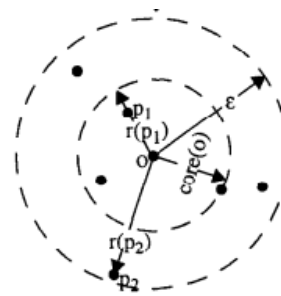
Konsep kepadatan pada DBSCAN dapat dijelaskan pada Gambar 1, yang mana terdapat tiga macam status dari setiap data, yaitu inti (*core*), batas (*border*), dan *noise*. Sebuah data akan dimasukkan sebagai inti jika jumlah data tetangga dan dirinya sendiri pada radius ϵ berjumlah \geq MinPts [13]. Nilai radius ϵ dan MinPts ini ditetapkan secara mandiri. Untuk data yang jumlah tetangga dan dirinya sendiri dalam radius ϵ kurang dari MinPts, tetapi tetangganya menjadi inti karena kehadirannya, data tersebut dikategorikan sebagai batas. Jika jumlah tetangga dan dirinya sendiri dalam radius ϵ kurang dari MinPts dan tidak ada tetangga yang menjadi inti karena kehadirannya, data tersebut dikategorikan sebagai noise [14].

DBSCAN memiliki beberapa keuntungan, antara lain (1) tidak memerlukan informasi tentang banyaknya kelompok yang akan dibentuk; (2) dapat memperoleh kelompok berbentuk sembarang; (3) dapat menemukan *noise*; dan (4) hanya membutuhkan dua parameter yang tidak sensitif terhadap urutan titik dalam basis data [15].

Algoritma OPTICS. Ketidakmampuan untuk menemukan cluster dengan kepadatan yang bervariasi adalah kelemahan utama dari DBSCAN yang dihasilkan dari kombinasi ukuran *neighborhood* tertentu dengan *density threshold* tunggal [16]. Pada tahun 1999, beberapa penulis DBSCAN mengembangkan OPTICS [17] untuk mengatasi masalah ini. OPTICS meminjam konsep *core density-reachable* dari DBSCAN, tetapi OPTICS merupakan algoritma pengurutan tambahan yang darinya hasil pengelompokan hierarkis dapat diturunkan. OPTICS membutuhkan parameter yang sama dengan DBSCAN, yaitu ϵ dan minPts. Namun, parameter ϵ secara teoritis tidak diperlukan dan hanya digunakan untuk tujuan mengurangi kompleksitas *runtime* dari algoritma.



Gambar 1. Konsep Kepadatan pada DBSCAN



Gambar 2. Core-distance(o), reachability-distances $r(p1,o)$, $r(p2,o)$ untuk MinPts = 4

OPTICS mengandung dua konsep tambahan yang disebut *core-distance* dan *reachability-distance* [16] seperti yang ditunjukkan pada Gambar 2. *Core-distance* suatu objek p adalah jarak terkecil ϵ' antara p dan sebuah objek di ϵ -neighborhood sedemikian rupa sehingga p akan menjadi objek inti sehubungan dengan ϵ' jika tetangga ini terkandung dalam $N_{\epsilon'}(p)$. *Reachability-distance* suatu objek p terhadap objek lain o adalah jarak terkecil sedemikian sehingga p dapat dicapai langsung oleh kerapatan dari o (jika o adalah objek inti). Dalam hal ini, *reachability-distance* tidak boleh lebih kecil dari jarak inti o karena untuk jarak yang lebih kecil tidak ada objek yang dapat dijangkau kepadatan secara langsung dari o [17]. Untuk membangun pengelompokan yang berbeda secara bersamaan, objek harus diproses dalam urutan tertentu, yaitu *cluster* dengan kepadatan lebih tinggi (ϵ lebih rendah) akan diselesaikan terlebih dahulu [18].

Cluster Validation. Untuk menghindari ketidaksesuaian hasil dan memastikan bahwa hasil *cluster* mencerminkan populasi secara umum, maka hasil dari *clustering* harus tervalidasi. Metode validasi hasil *clustering* terdiri dari indeks internal dan eksternal. Indeks internal digunakan untuk mengukur kebaikan struktur pengelompokan tanpa memperhatikan informasi eksternal, sedangkan indeks eksternal digunakan untuk mengukur sejauh mana label *cluster* cocok dengan label kelas yang disediakan secara eksternal.

Salah satu metode untuk menguji validitas hasil *cluster* adalah dengan menggunakan *Silhouette Coefficient* [19]. Nilai dari *Silhouette Coefficient* berada berkisar antara -1 sampai dengan 1. Jika nilai *Silhouette Coefficient* hampir 1 maka objek tersebut berada di *cluster* yang tepat, jika berada di sekitar 0 maka objek tersebut bisa berada di antara 2 *cluster*, dan jika hasilnya negatif maka objek tersebut kemungkinan berada di *cluster* yang salah [20]. *Silhouette Coefficient* [21] untuk mengetahui kualitas hasil *clustering* seperti pada persamaan (2) berikut:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (2)$$

dimana $s(i)$ merupakan nilai *Silhouette Coefficient*, $a(i)$ merupakan rata-rata jarak antara titik i dengan seluruh titik dalam A (*cluster* dimana titik i berada), $b(i)$ adalah rata-rata jarak antara titik i ke seluruh titik dalam *cluster* selain A .

Validasi internal lainnya yang dapat digunakan untuk melihat kebaikan struktur *cluster* yang terbentuk adalah Indeks Gamma Baker-Hubert. Indeks Gamma Baker-Hubert merupakan adaptasi dari indeks Γ korelasi antara dua vektor data A dan B dengan ukuran yang sama [22]. Misalkan kita memiliki dua vektor $A = \{a_1, a_2, \dots, a_n\}$, $B = \{b_1, b_2, \dots, b_n\}$. Kemudian, dua indeks yang diberikan (i,j) dikatakan *concordant* jika setiap kali $a_i < a_j$ diikuti $b_i < b_j$. Jika tidak, kedua indeks tersebut dikatakan *discordant*. Selanjutnya, dihitung jumlah pasangan *concordant* (s^+) dan jumlah pasangan *discordant* (s^-). Indeks gamma [23] kemudian didefinisikan sebagai:

$$\Gamma = \frac{s^+ - s^-}{s^+ + s^-} \quad (3)$$

Angka s^+ mewakili jarak antara dua titik yang termasuk dalam *cluster* yang sama (yaitu pasangan yang nilai vektor B adalah 0) lebih kecil dari jarak antara dua titik yang tidak tergolong dalam *cluster* yang sama (yaitu pasangan yang nilai vektor B adalah 1). Angka s^- merepresentasikan berapa kali terjadi situasi yang berlawanan, artinya jarak antara dua titik yang berada dalam *cluster* yang sama (nilai 0 di B) lebih besar dari jarak antara dua titik yang tidak termasuk dalam *cluster* yang sama (nilai 1 di B). Indeks gamma berada pada rentang -1 sampai 1 yang berarti semakin besar nilai indeks gamma maka semakin baik struktur *cluster* yang terbentuk tersebut [23].

Selanjutnya, kita juga dapat membandingkan partisi yang dibuat menggunakan dua algoritma, yang mana dapat digunakan untuk menguji keefektifan *cluster* yang diproduksi. Langkah-langkah ini termasuk dalam indeks eksternal karena kita membandingkan dua partisi. Misalkan kita memiliki kumpulan data, $X = \{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ dan dua partisi, $P_1 = \{X_1, X_2, \dots, X_r\}$ dan $P_2 = \{Y_1, Y_2, \dots, Y_s\}$ dengan mendefinisikan:

yy: Jumlah titik data yang berada di *cluster* yang sama di kedua partisi.

yn: Jumlah titik data yang berada di *cluster* yang sama di partisi P_1 , tetapi di *cluster* berbeda di partisi P_2 .

ny: Jumlah titik data yang berada di *cluster* berbeda di partisi P_1 , tetapi di *cluster* yang sama di partisi P_2 .

nn: Jumlah titik data yang berada di *cluster* berbeda di kedua partisi.

Indeks Rand [24] didefinisikan sebagai:

$$R = \frac{yy+nn}{yy+ny+yn+nn} \quad (4)$$

Secara intuitif, indeks Rand bertindak sebagai ukuran akurasi untuk *cluster*, dan berada di antara 0 dan 1.

Semakin besar nilai indeks Rand (semakin mendekati 1) maka semakin akurat pengelompokan data antara partisi 1 dan partisi 2 [25].

Implementasi Sistem. Bagian ini menjelaskan tentang bagaimana alur kerja dari sistem *clustering* dengan metode DBSCAN. Terdapat 4 proses utama sistem *clustering* dalam penelitian ini, yaitu:

1. Penentuan Nilai ϵ dan MinPts

Merupakan proses penentuan ϵ dan MinPts yang paling optimal. Proses ini penting karena merupakan proses awal yang diperlukan untuk menjalankan sistem. Nilai ϵ ditentukan dengan menggunakan algoritma *k-Nearest Neighbor*, sedangkan nilai MinPts melalui proses *trial and error*.

2. *Clustering* menggunakan DBSCAN

Pada proses ini, nilai ϵ dan MinPts yang telah ditentukan dimasukkan ke dalam sistem. Sistem secara otomatis akan mengelompokkan objek-objek dalam dataset berdasarkan masukan dari nilai ϵ dan MinPts. Jika suatu objek berada dalam radius ϵ , objek tersebut termasuk dalam kategori kepadatan yang diinginkan.

3. *Cluster Validation*

Cluster validation diperlukan untuk mengetahui apakah *cluster* yang terbentuk sudah cukup baik dalam mengelompokkan objek-objek dalam dataset. *Cluster validation* dilakukan dengan cara menghitung *Silhouette Coefficient* dari *cluster* yang terbentuk.

4. Perbandingan antara DBSCAN dan OPTICS

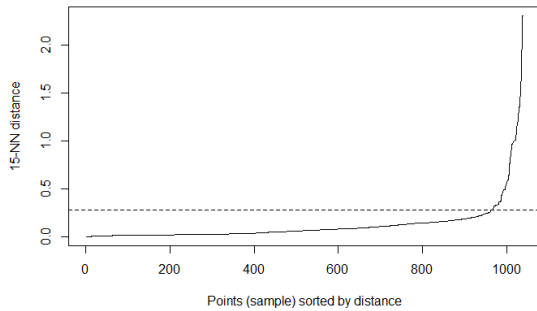
Membandingkan algoritma DBSCAN dan OPTICS diperlukan untuk mengetahui apakah dataset yang digunakan memiliki kepadatan yang sama atau berbeda. Jika hasil *clustering* menggunakan DBSCAN lebih baik dibandingkan hasil *clustering* menggunakan OPTICS, ditandai dengan hasil validasi yang lebih baik, dataset tersebut lebih cocok dikelompokkan menggunakan algoritma DBSCAN dan mengindikasikan bahwa dataset memiliki kepadatan yang cenderung seragam.

5. Pembahasan dan Analisis Gempa Bumi

Membahas hasil *clustering* dengan algoritma yang lebih baik (DBSCAN atau OPTICS) pada dataset gempa bumi yang digunakan dalam penelitian ini

3. Hasil dan Pembahasan

Penentuan Nilai ϵ dan MinPts. Pada penelitian ini, nilai ϵ yang akan digunakan dalam analisis ditentukan dengan menggunakan algoritma *k-Nearest Neighbor* dengan $k = 15$.



Gambar 3. 15-NN pada Penentuan Nilai ϵ

Tabel 2. Kombinasi Input Nilai ϵ dan MinPts

ϵ	MinPts	Silhouette	Cluster
0,28	1	0,5639	10
0,28	2	0,6675	6
0,28	3	0,8109	3
0,28	4	0,8093	3
0,28	5	0,7965	3

Berdasarkan gambar 3, terlihat bahwa *tipping point* berada pada titik 0,28 sehingga nilai ϵ optimum yang digunakan dalam analisis selanjutnya sebesar 0,28. Kemudian, dalam menentukan nilai MinPts, dapat dilakukan secara *trial and error* dengan memperhatikan nilai *Silhouette Coefficient* yang dihasilkan. Pada tabel 2 berikut, ditampilkan beberapa nilai *Silhouette Coefficient* dari beberapa percobaan.

Tabel 2 menjelaskan bahwa pada suatu nilai ϵ tertentu, penambahan nilai MinPts akan menghasilkan pola *diminishing return* pada kualitas *cluster* yang ditunjukkan oleh *Silhouette Coefficient*. Penambahan nilai MinPts akan meningkatkan kualitas *cluster*, tetapi pada titik tertentu justru akan menurunkan kualitas *cluster*. Sedangkan semakin besar nilai minPts akan semakin memperkecil kemungkinan terbentuknya suatu *cluster*. Hal tersebut dikarenakan syarat jumlah minimal objek yang berdekatan dalam *cluster* semakin besar sehingga membutuhkan lebih banyak objek untuk membentuk suatu *cluster*, demikian pula sebaliknya. Hal ini sejalan dengan hasil penelitian yang dilakukan oleh Furqon dan Muflikah pada tahun 2016.

Clustering menggunakan algoritma DBSCAN. Setelah didapatkan nilai ϵ dan MinPts optimum yang ditandai dengan nilai *Silhouette Coefficient* paling tinggi, selanjutnya dilakukan proses clustering menggunakan algoritma DBSCAN. Nilai ϵ dan MinPts tersebut ($\epsilon = 0,28$ dan MinPts = 3) digunakan sebagai input untuk menjalankan proses clustering pada DBSCAN.

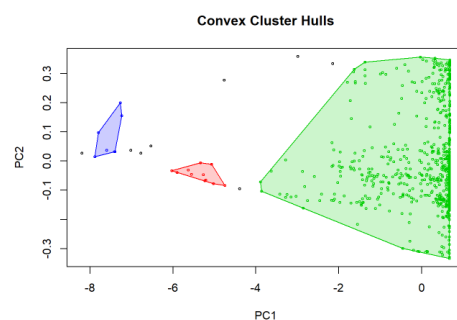
Tabel 3 merupakan hasil *clustering* yang telah dilakukan. Diketahui bahwa jumlah kejadian gempa dalam *dataset* adalah sebanyak 1.038 kejadian gempa (Pts = 1038). *Clustering* tersebut menghasilkan 3 *cluster* dengan *noise* sebanyak 10 kejadian gempa (*cluster* 0). Tiap-tiap *cluster* terbentuk dari jumlah kejadian gempa yang berbeda, yaitu 14 kejadian gempa pada *cluster* 1, 1.006 kejadian gempa pada *cluster* 2, dan 8 kejadian gempa pada *cluster* 3. Plot dari hasil *clustering* dapat dilihat pada gambar 5 berikut.

Clustering menggunakan algoritma OPTICS. Setelah didapatkan nilai ϵ dan MinPts optimum yang ditandai dengan nilai *Silhouette Coefficient* paling tinggi, selanjutnya dilakukan proses clustering menggunakan algoritma OPTICS. Nilai ϵ dan MinPts tersebut ($\epsilon = 0,28$ dan MinPts = 3) digunakan sebagai input untuk menjalankan proses clustering pada OPTICS.

Tabel 4 merupakan hasil *clustering* yang telah dilakukan. *Clustering* yang dilakukan dengan algoritma OPTICS tersebut menggunakan parameter yang sama dengan DBSCAN, yaitu $\epsilon = 0,28$ dan MinPts = 3 dengan ϵ *threshold* terbesar bernilai 0.1. Pada gambar tersebut diketahui bahwa jumlah kejadian gempa dalam *dataset* adalah sebanyak 1.038 kejadian gempa. *Clustering* tersebut menghasilkan 24 *cluster* dengan *noise* sebanyak 64 kejadian gempa.

Tabel 3. Output DBSCAN Menggunakan MinPts= 3 dan $\epsilon= 0,28$

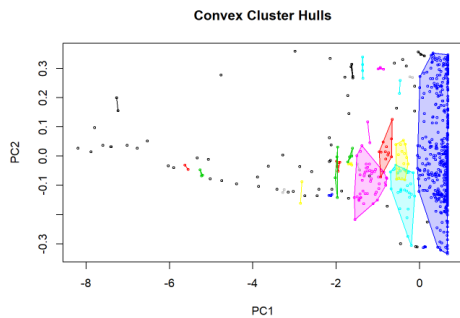
Output	Keterangan	Border	Seed	Jumlah
0	Noise	10	0	10
1	Cluster 1	0	14	14
2	Cluster 2	0	1006	1006
3	Cluster 3	1	7	8



Gambar 4. Cluster Plot DBSCAN

Tabel 4. Hasil Clustering dengan OPTICS

Parameter	
MinPts	3
ϵ	0,28
E threshold	0,1
Output	
Cluster	24
Noise	64



Gambar 5. Cluster Plot OPTICS

Cluster Validation. Penelitian ini menggunakan validasi internal dan eksternal, yaitu Koefisien Silhouette dan Indeks Gamma untuk validasi internal, serta Indeks Rand untuk validasi eksternal. Hasil *clustering* DBSCAN dan OPTICS masing-masing diuji validitasnya dengan tiga koefisien/indeks tersebut. Tabel 5 berikut menyajikan informasi yang dihasilkan dari DBSCAN maupun OPTICS.

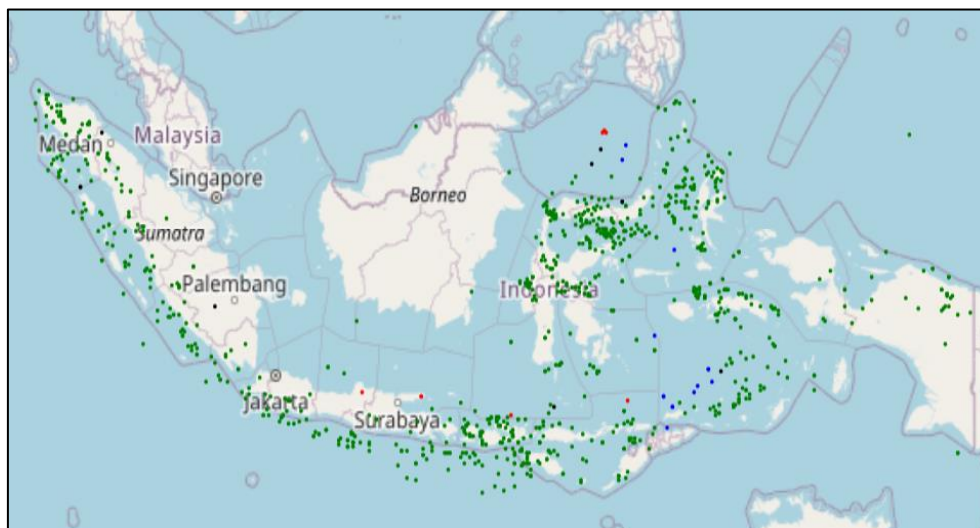
Tabel 5. Perbandingan hasil clustering antara algoritma DBSCAN dan OPTICS

	DBSCAN	OPTICS
Cluster terbentuk	3	24
Noise	10	64
Koef. Silhouette	0.81091	0.58126
Indeks Gamma	0.98104	0.96108
Indeks Rand	0.36112	0.48050

Tabel 5 memberikan informasi bahwa algoritma DBSCAN dapat mengelompokkan data di dalam *dataset* dengan lebih baik dibandingkan algoritma OPTICS. Hal ini terlihat dari jumlah *cluster* yang terbentuk lebih sedikit pada DBSCAN sehingga terbentuk dengan algoritma DBSCAN lebih sedikit dibandingkan OPTICS sehingga mudah untuk

dianalisis dan diinterpretasikan. Selain itu, *noise* dari DBSCAN juga lebih sedikit dibandingkan OPTICS. Kemudian validitas internal pada DBSCAN lebih baik daripada OPTICS karena memiliki indeks internal yang lebih tinggi, baik Koefisien Silhouette maupun Indeks Gamma. Koefisien/indeks internal tersebut dapat diartikan bahwa struktur *cluster* yang terbentuk sudah cukup baik, yaitu kemiripan karakteristik antar titik di dalam *cluster* sudah cukup tinggi. Walaupun demikian, Indeks Rand dari DBSCAN masih lebih kecil daripada OPTICS. Hal ini berarti bahwa akurasi pengelompokan data antara partisi 1 dan partisi 2 pada OPTICS lebih tinggi dibandingkan pada DBSCAN. Oleh sebab itu, analisis selanjutnya pada penelitian ini mengacu pada hasil *clustering* dengan algoritma DBSCAN karena alasan-alasan yang telah dipaparkan sebelumnya.

Pembahasan dan Analisis Gempa Bumi. Hasil analisis proses *clustering* dengan menggunakan algoritma DBSCAN menunjukkan bahwa algoritma DBSCAN dapat dengan baik mengelompokkan wilayah beresiko terjadi gempa bumi berdasarkan karakteristik parameter gempa bumi yang dihasilkan. Hal tersebut dapat dilihat dari tingginya nilai *Silhouette Coefficient* dan Indeks Gamma dari hasil *clustering dataset* gempa bumi. Hasil yang terbaik diperoleh dengan kombinasi nilai ϵ dan \minPts yaitu 0,28 dan 3. Dengan menggunakan hasil dari *clustering* tersebut, dapat kemudian dilakukan pengelompokan data gempa bumi berdasarkan karakteristik objek-objek yang ada dalam satu *cluster*. Pada gambar 6 disajikan hasil pengelompokan data kejadian gempa bumi di Indonesia berdasarkan parameter gempa bumi yang dihasilkan.



Keterangan: Biru = Cluster 1 Hijau = Cluster 2 Merah Cluster 3 Hitam = Noise

Gambar 6. Pengelompokan Data Gempa Bumi dengan Algoritma DBSCAN

Dari hasil *clustering* menggunakan algoritma DBSCAN, maka didapatkan 3 area yang berbeda (yang digambarkan dalam bentuk *cluster*). Pada gambar 6, tiap-tiap *cluster* ditunjukkan dengan warna yang berbeda, yaitu warna biru untuk *cluster* 1, hijau untuk *cluster* 2, dan merah untuk *cluster* 3, sedangkan wilayah yang diberi warna hitam disebut *noise*. *Noise* merupakan *random error* yang keberadaannya cenderung kurang berarti dalam suatu *dataset*. Dari tabel 6 berikut diketahui bahwa gempa bumi paling sering terjadi pada kedalaman (*depth*) 10-50 km dan magnitudo sebesar 2,5-3,49 SR yaitu pada wilayah gempa bumi yang tergolong sebagai *cluster* 2.

Tabel 6. Karakteristik Parameter Gempa Bumi

Cluster	Depth (km)		Magnitudo (SR)		Jumlah Data	Label (Resiko)
	Min	Max	Min	Max		
1	319	497,7	4,01	5	8	Tinggi
2	10	318,5	3,5	4	650	Sedang
3	498	589,2	2,5	3,49	6	Rendah

Proses selanjutnya adalah memberikan label pada *cluster* berdasarkan karakteristik parameter gempa bumi yang ada di dalam masing-masing *cluster*. Wilayah potensial gempa bumi dapat dikelompokkan menjadi 3 *cluster*, yaitu wilayah beresiko tinggi, sedang, dan rendah. Hal ini ditentukan berdasarkan kekuatan gempa (magnitudo) yang sering terjadi pada suatu *cluster*. *Cluster* 1 selanjutnya disebut sebagai wilayah beresiko tinggi karena 8 dari 14 titik gempa berkekuatan 4,01-5 SR, *cluster* 2 sebagai wilayah beresiko sedang karena 650 dari 1.006 titik gempa berkekuatan 3,5-4 SR, dan *cluster* 3 sebagai wilayah beresiko rendah karena 6 dari 8 titik gempa berkekuatan 2,5-3,49 SR. Pemberian label pada *cluster* penting dilakukan agar dapat digunakan dalam proses pengambilan keputusan atau kebijakan terkait penanganan atau mitigasi bencana gempa bumi.

4. Kesimpulan

Hasil *clustering* pada penelitian ini memberikan kesimpulan bahwa dengan menggunakan algoritma DBSCAN, maka diperoleh 3 *cluster* wilayah beresiko terjadi gempa bumi. Tiap-tiap *cluster* terbentuk dari jumlah *kejadian gempa* yang berbeda, yaitu 14 *kejadian gempa* pada *cluster* 1, 1.006 *kejadian gempa* pada *cluster* 2, dan 8 *kejadian gempa* pada *cluster* 3. Kombinasi nilai ϵ dan *minPts* yaitu 0,28 dan 3 menghasilkan nilai *Silhouette Coefficient* sebesar 0,8109115 dan Indeks Gamma sebesar 0,98104 yang menggambarkan bahwa DBSCAN mampu mengelompokkan wilayah beresiko terjadi gempa bumi dengan cukup baik. Wilayah beresiko terjadi gempa bumi dapat dikelompokkan berdasarkan kekuatan gempa menjadi 3 *cluster*, yaitu wilayah

beresiko tinggi (4-5 SR), sedang (3,5-4 SR), dan rendah (2,5-3,5 SR).

Saran. Berdasarkan kesimpulan yang diperoleh di atas, penulis menyarankan agar kedepannya dapat dilakukan penelitian menggunakan algoritma lain dengan metode *density-based clustering* dan membandingkannya dengan algoritma DBSCAN. Selain itu, dapat ditambahkan jumlah data yang digunakan agar diperoleh akurasi dan kualitas *overall value* yang tinggi dari *cluster* yang dihasilkan. Bagi instansi terkait, penelitian ini dapat dijadikan bahan pertimbangan dalam menentukan kebijakan mitigasi bencana gempa bumi, khususnya di wilayah yang beresiko tinggi terhadap gempa bumi.

Daftar Pustaka

- [1] Pasau, G. and Tanauma, A., 2011. Pemodelan sumber gempa di Wilayah Sulawesi Utara sebagai upaya mitigasi bencana gempa bumi. *Jurnal Ilmiah Sains*, 11(2), pp.202-209.
- [2] Rahmi, E., 2018. Penerapan Algoritma DBSCAN untuk Pengelompokan Wilayah Gempa Bumi di Indonesia. Universitas Islam Negeri Sultan Syarif Kasim Riau.
- [3] Denpasar, P. I. B. B. W. I. B., 2017. Balai Besar Meteorologi, Klimatologi, dan Geofisika Wilayah III. [Online] Available at: <http://balai3.denpasar.bmkg.go.id/tentang-gempa> [Accessed 9 December 2020].
- [4] Tan, P. N., Steinbach, M., & Kumar, V. 2006. *Introduction to Data Mining*, Pearson Education. Inc., New Delhi.
- [5] Furqon, M. T., & Muflikhah, L. 2016. *Clustering the Potential Risk of Tsunami Using Density-Based Spatial Clustering of Application with Noise (DBSCAN)*. *Journal of Environmental Engineering and Sustainable Technology*, 3(1), 1-8.
- [6] Kryszkiewicz M., Skonieczny Ł. 2005. *Faster Clustering with DBSCAN*. In: *Kłopotek M.A., Wierchoń S.T., Trojanowski K. (eds) Intelligent Information Processing and Web Mining. Advances in Soft Computing*, vol 31. Springer, Berlin, Heidelberg.
- [7] Ali, T., Asghar, S. and Sajid, N.A., 2010, *June. Critical analysis of DBSCAN variations. In 2010 International Conference on Information and Emerging Technologies (pp. 1-6). IEEE.*
- [8] Schubert, E., Sander, J., Ester, M., Kriegel, H.P. and Xu, X., 2017. *DBSCAN revisited, revisited: why and how you should (still) use DBSCAN*. *ACM Transactions on Database Systems (TODS)*, 42(3), pp.1-21.
- [9] Scitovski, S. 2017. *A Density-Based Clustering Algorithm for Earthquake Zoning*. *Computers & Geosciences*, 110, 90-95.

- [10] Harini, S., Fahmi, H., Dwi Mulyanto, A., & Khudzaifah, M. 2020. *The Earthquake Events and Impacts Mapping in Bali and Nusa Tenggara Using A Clustering Method*.
- [11] Kim, J. H., Choi, J. H., Yoo, K. H., & Nasridinov, A. 2019. AA-DBSCAN: An Approximate Adaptive DBSCAN for Finding Clusters with Varying Densities. *The Journal of Supercomputing*, 75(1), 142-169.
- [12] Sani, A. 2018. Penerapan Metode K-Means Clustering pada Perusahaan. *Jurnal Ilmiah Teknologi Informasi*, 353, 1-7.
- [13] Safitri, D., Wuryandari, T., & Rahmawati, R. 2017. Metode DBSCAN Untuk Pengelompokan Kabupaten/Kota di Provinsi Jawa Tengah Berdasarkan Produksi Padi Sawah dan Padi Ladang. *Jurnal Statistika Universitas Muhammadiyah Semarang*, 5(1).
- [14] Prasetyo, E. 2012. *Data Mining-Konsep dan Aplikasi Menggunakan Matlab*, Nikodemus WK, Ed. Yogyakarta, Indonesia: Andi Offset.
- [15] Fahamsyah, M. 2020. Metode DBSCAN Clustering untuk Analisis Pola Penyebaran Petir di Pasuruan (*Doctoral Dissertation*, Universitas Islam Negeri Maulana Malik Ibrahim).
- [16] Hahsler, M., Piekenbrock, M. and Doran, D., 2019. *DBSCAN: Fast density-based clustering with R*. *Journal of Statistical Software*, 91(1), pp.1-30.
- [17] Ankerst, M., Breunig, M.M., Kriegel, H.P. and Sander, J., 1999. *OPTICS: ordering points to identify the clustering structure*. *ACM Sigmod kejadian gempa*, 28(2), pp.49-60.
- [18] Kanagala, H.K. and Krishnaiah, V.J.R., 2016, January. A comparative study of K-means, DBSCAN and OPTICS. In *2016 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-6). IEEE.
- [19] Larose, D. T., & Larose, C. D. 2014. *Discovering Knowledge in Data: An Introduction to Data Mining* (Vol. 4). John Wiley & Sons.
- [20] Simamora, D. A. S., Furqon, M. T., & Priyambadha, B. 2017. *Clustering Data Kejadian Tsunami yang Disebabkan oleh Gempa Bumi dengan Menggunakan Algoritma K-Medoids*. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer* E-ISSN, 2548, 964x.
- [21] Kaufman, L., & Rousseeuw, P. J. 2009. *Finding Groups in Data: An Introduction to Cluster Analysis* (Vol. 344). John Wiley & Sons.
- [22] Yedida, R., 2019. *Beginning with ML: Machine learning for beginners, by a beginner*. [Online] Available at: <https://beginningwithml.wordpress.com/2019/04/26/11-6-evaluating-clusters/> [Accessed 13 Januari 2021].
- [23] Desgraupes, B., 2013. Clustering indices. *University of Paris Ouest-Lab Modal'X*, 1, p.34.
- [24] W. M. Rand. 1971. "Objective criteria for the evaluation of clustering methods". *Journal of the American Statistical Association*. American Statistical Association. 66 (336): 846–850.
- [25] Lawrence Hubert and Phipps Arabie. 1985. "Comparing partitions". *Journal of Classification*. 2 (1): 193–218.