

## CLUSTERING THE POTENTIAL RISK OF TSUNAMI USING DENSITY-BASED SPATIAL CLUSTERING OF APPLICATION WITH NOISE (DBSCAN)

Muhammad Tanzil Furqon<sup>1</sup>, Lailil Muflikhah<sup>2</sup>

<sup>1,2</sup>Universitas Brawijaya

Email: <sup>1</sup>m.tanzil.furqon@ub.ac.id, <sup>2</sup>lailil@ub.ac.id

### ABSTRACT

Tsunami is one of the deadliest natural disaster that causing devastating property damage and loss of life. Therefore, this triggers many scientist to do researches in tsunami mitigation disaster, such as analyzing the potential risks caused by tsunami. The process of analyzing the potential risk caused by tsunami can be done by grouping the data of tsunami based on characteristics of the previous tsunami events. DBSCAN (Density-based Spatial Clustering of Application with Noise) is a popular clustering method and can be used to do the task. The algorithm do the clustering processes using density-based concept that able to detect outlier/noise and clusters irregular shapes. It was proved in this research where the evaluation method using Silhouette Coefficient on the DBSCAN clustering result gave highest value 0.96056649 for  $\epsilon$  and  $minPts$  value of .1 and 0.1.

**Keywords:** *clustering, dbscan, tsunami, risk*

### 1. INTRODUCTION

Menurut catatan NOAA (National Oceanic and Atmospheric Administration), lembaga yang bertugas melakukan penelitian, monitoring dan menyampaikan peringatan dini tentang potensi timbulnya tsunami; jumlah korban jiwa terbesar yang ditimbulkan oleh tsunami adalah 316 ribu jiwa yang terjadi pada tahun 2010 di kepulauan Haiti dan Republik Dominika. Sedangkan bencana tsunami yang terjadi di Aceh pada tahun 2004 menempati urutan kedua dengan jumlah korban sekitar 227 ribu jiwa (NGDC 2016).

Berbagai penelitian telah dilakukan dalam rangka menganalisa sumber dan resiko tsunami untuk membuat sistem mitigasi tsunami guna mengantisipasi timbulnya kerugian maupun korban jiwa di masa yang akan datang (Uchida

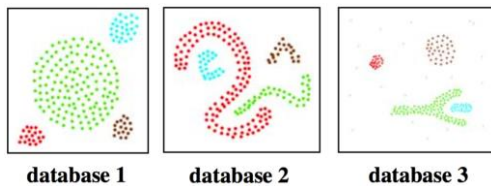
2014, Prabhudesai 2008, Geist 2009, Yadav 2013). Namun diantara beberapa penelitian tersebut lebih fokus pada penyebab terjadinya tsunami dan tidak melakukan pengelompokan wilayah terdampak tsunami berdasarkan potensi resiko yang ditimbulkan. Dengan menggunakan data tsunami dan model algoritma clustering yang ada, diantaranya yang cukup populer adalah DBSCAN, harapannya dapat dihasilkan model output berupa analisa potensi resiko tsunami terhadap beberapa wilayah yang terdampak tsunami.

Di antara beberapa algoritma *clustering*, DBSCAN merupakan algoritma yang masuk dalam kategori *density-based clustering*, yaitu proses pembentukan cluster dilakukan berdasarkan tingkat kedekatan/kepadatan jarak antar obyek dalam dataset tersebut. Beberapa kelebihan yang dimiliki algoritma DBSCAN dibandingkan dengan algoritma *clustering* yang lain diantaranya adalah kemampuannya dalam mendeteksi outlier/noise. Hal tersebut dikarenakan konsep *density-based* yang digunakan, yaitu obyek yang tidak memiliki kedekatan jarak dengan obyek yang lainnya akan dikenali sebagai outlier. Selain itu, dalam DBSCAN tidak perlu mengetahui jumlah cluster ( $k$ ) yang akan terbentuk, sebagaimana yang ada pada k-Means dan k-Medoids. DBSCAN juga mampu mengenali bentuk cluster yang tidak beraturan yang sulit dikenali oleh beberapa algoritma clustering populer lainnya.

### 2. ALGORITMA DBSCAN

DBSCAN (Density-based Spatial Clustering of Application with Noise) merupakan salah satu algoritma *clustering* yang mengelompokkan obyek ke dalam *cluster* dimana kategori atau *class* data ujinya belum diketahui (*unsupervised learning*). Algoritma ini cukup populer dan mendapatkan penghargaan *the test of the time award*, yaitu penghargaan yang diberikan untuk algoritma

yang memperoleh perhatian khusus dalam hal teori maupun prakteknya, pada konferensi tentang *data mining* yang diselenggarakan oleh ACM (*Asociacion for Computing Machinery*) pada 2014 (ACM 2014). Dalam penelitian yang lain, Ester M, et al. menyimpulkan bahwa algoritma DBSCAN lebih efisien dalam proses clustering untuk bentuk cluster yang tidak beraturan, seperti yang ditampilkan pada Gambar 1, dibandingkan dengan algoritma CLARANS, bahkan hingga 100 kali lipat lebih efisien (Ester 1996).



Gambar 1. Hasil Clustering DBSCAN (Ester 1996)

DBSCAN menggunakan  $\epsilon$  (epsilon atau jarak *density-reachable object*) dan *minPts* (jumlah tetangga minimal) yang ditentukan diawal dalam membentuk *cluster*. Proses clustering (Gambar 2) diawali dengan memilih secara acak kandidat *core point*, yaitu titik yang berada dalam jangkauan  $\epsilon$  dan memiliki jumlah obyek tetangga sebanyak minimal *minPts*. Obyek yang dipilih sebagai kandidat *core point* tersebut kemudian akan diuji apakah memenuhi syarat sebagai *core point* atau tidak. Jika memenuhi, maka obyek tersebut akan membentuk satu *cluster* baru dengan obyek-obyek tetangganya (*density-reachable object*). Kemudian, proses selanjutnya adalah dengan menjadikan *density-reachable object* yang lainnya sebagai kandidat *core point* berikutnya dan mengujinya hingga seluruh obyek dalam dataset selesai diuji. Obyek yang tidak memenuhi syarat sebagai *core point* dapat dikategorikan menjadi dua, yaitu sebagai *density-reachable object*, yaitu obyek yang jaraknya dengan *core point* kurang dari atau sama dengan  $\epsilon$  tetapi jumlah *density-reachable object* disekitarnya kurang dari *minPts*; dan sebagai *outlier/noise*, yaitu obyek yang jaraknya dengan core point lebih dari  $\epsilon$  dan jumlah *density-reachable object* disekitarnya kurang dari *minPts*. Dalam proses perhitungan

jarak antar obyeknya, algoritma DBSCAN menggunakan fungsi Euclidean Distance sebagaimana ditunjukkan dalam persamaan (1) berikut:

$$d_{ij} = \sqrt{\sum_{a=1}^p (x_{ia} - x_{ja})^2}, i = 1, \dots, n; j = 1, \dots, n. \quad (1)$$

dimana  $x_{ia}$  adalah variabel ke- $a$  dari obyek  $i$  ( $i=1, \dots, n; a=1, \dots, p$ ) dan  $d_{ij}$  adalah nilai Euclidean Distance.

**Algorithm 2** Basic DBSCAN Algorithm

- 1: Begin
- 2: Arbitrary select a point p.
- 3: Retrieve all points density-reachable from p w.r.t Eps and MinPts.
- 4: If p is a core point, a cluster is formed.
- 5: If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database.
- 6: Continue the process until all of the points have been processed.
- 7: End

Gambar 2. Algoritma DBSCAN

Di antara beberapa algoritma *clustering*, DBSCAN merupakan algoritma yang masuk dalam kategori *density-based clustering*, yaitu proses pembentukan cluster dilakukan berdasarkan tingkat kedekatan/kepadatan jarak antar obyek dalam dataset tersebut. Beberapa kelebihan yang dimiliki algoritma DBSCAN dibandingkan dengan algoritma *clustering* yang lain diantaranya adalah kemampuannya dalam mendeteksi outlier/noise. Hal tersebut dikarenakan konsep *density-based* yang digunakan, yaitu obyek yang tidak memiliki kedekatan jarak dengan obyek yang lainnya akan dikenali sebagai outlier. Selain itu, dalam DBSCAN tidak perlu mengetahui jumlah cluster ( $k$ ) yang akan terbentuk, sebagaimana yang ada pada k-Means dan k-Medoids. DBSCAN juga mampu mengenali bentuk cluster yang tidak beraturan yang sulit dikenali oleh beberapa algoritma clustering populer lainnya.

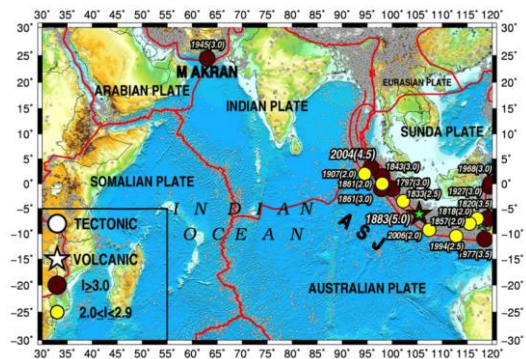
Terlepas dari beberapa keunggulan dari DBSCAN diatas, beberapa kelemahan juga ditemukan, diantaranya adalah penentuan dua nilai input  $\epsilon$  dan *minPts* diawal proses clustering. Memberikan nilai  $\epsilon$  yang terlalu besar berdampak pada semakin luas cakupan cluster yang terbentuk, sedang terlalu kecil

akan menghasilkan jumlah cluster yang banyak dan jarak obyek yang sangat berdekatan satu dengan yang lainnya. Demikian halnya dengan *minPts*, hasil cluster yang terbentuk akan bergantung pada nilai yang ditentukan diawal. Oleh sebab itu diperlukan metode untuk menentukan nilai  $\epsilon$  dan *minPts* yang paling sesuai untuk algoritma DBSCAN agar dapat menghasilkan cluster yang paling baik. Dalam beberapa penelitian digunakan metode optimasi dan komputasi paralel untuk mendapatkan nilai  $\epsilon$  dan *minPts* secara otomatis dan meningkatkan performa komputasi, misalnya dengan menggunakan algoritma evolusi (Karami 2014) dan menggunakan *grid computing* (Ma 2014). Keduanya sangat efektif untuk proses clustering dengan jumlah data yang cukup besar. Sedang Elbatta (Elbatta 2013) melakukan penelitian dengan menggunakan *density threshold* untuk menentukan variasi nilai  $\epsilon$  pada proses *clustering*, dimana masih harus menentukan threshold yang sesuai untuk mendapatkan nilai  $\epsilon$  yang optimal.

### 3. TSUNAMI

Tsunami, dalam bahasa Jepang berarti gelombang pasang dari pelabuhan, merupakan salah satu jenis bencana alam yang menjadi ancaman terbesar manusia. Hampir mayoritas sumber penyebab terjadinya tsunami adalah berasal dari gempa bumi dan pergeseran lempeng tanah/*land slide* (Prabhudesai 2008). Namun demikian, tidak semua gempa bumi atau *land slide* menyebabkan terjadinya tsunami. Bencana tsunami yang diakibatkan oleh gempa bumi telah tercatat beberapa kali terjadi di sepanjang pesisir pantai bagian barat pulau Sumatera, seperti tampak pada Gambar 3 (Geist 2009). Bahkan berdasarkan data NOAA antara tahun 1797-2006 ada lebih dari 100 peristiwa tsunami yang tercatat menimpa negara Indonesia dengan variasi intensitas dan dampak yang diakibatkan. Selain itu, NOAA juga mencatat dampak serta kerugian yang ditimbulkan akibat bencana tsunami, termasuk data jumlah orang yang terluka, hilang, dan meninggal, serta jumlah total rumah yang rusak. Data tersebut tentunya sangat penting untuk proses analisa potensi resiko yang ditimbulkan akibat terjadinya bencana tsunami.

Salah satu model yang digunakan dalam menganalisa potensi resiko tsunami adalah mengelompokkan wilayah/area berdasarkan tingkatan resiko terhadap dampak kerugian yang ditimbulkan, seperti jumlah korban jiwa, korban terluka dan hilang, jumlah rumah yang rusak maupun hancur, serta termasuk intensitas tsunami maupun gempa yang terjadi. Pada penelitian sebelumnya, Yadav et al. (Yadav 2013) menggunakan metode Bayes untuk melakukan prediksi terhadap kemungkinan terjadinya tsunami berdasarkan besar intensitasnya dalam periode waktu tertentu. Sedangkan Uchida et al. (Uchida 2014) menggunakan algoritma Dijkstra untuk melakukan simulasi terhadap gerakan air laut, berikut waktu dan jangkauannya ketika tsunami terjadi. Dari beberapa penelitian tersebut belum ada yang melakukan analisa terkait potensi resiko yang ditimbulkan berdasarkan wilayah yang terdampak tsunami. Oleh sebab itu, dalam penelitian ini, dengan menggunakan salah satu model *clustering*, yaitu DBSCAN, diharapkan dapat mengelompokkan wilayah terdampak tsunami berdasarkan potensi resiko secara akurat sehingga dapat menjadi referensi dalam sistem mitigasi bencana tsunami.



Gambar 3. Peta Pusat Gempa Pemicu Tsunami dari Tahun 1797-2006 (Geist 2009)

## 4. EKSPERIMEN

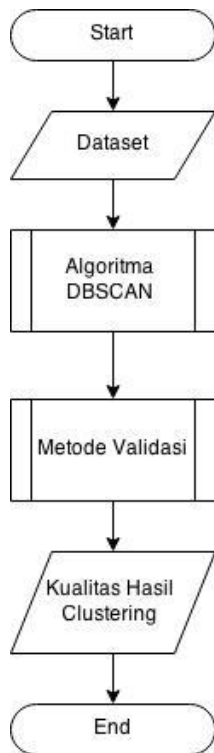
### 4.1. Gambaran Sistem

Secara garis besar, sistem memiliki dua proses utama, yaitu proses *clustering* menggunakan algoritma DBSCAN dan proses validasi sekaligus perhitungan kualitas hasil *clustering* (Gambar 4). Pada proses *clustering*

dataset akan dikelompokkan ke dalam sejumlah *cluster* tertentu menggunakan algoritma DBSCAN. Hasil proses tersebut kemudian divalidasi dengan menggunakan metode validasi *Silhouette Coefficient* (Kaufman 2009) 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  $A$ .



Gambar 4. Diagram Alir Sistem

#### 4.2. Preprocessing

Sebelum data dikelompokkan menggunakan algoritma DBSCAN, maka data harus dipastikan sudah siap untuk diproses. Preprocessing data diperlukan untuk memastikan bahwa data telah sesuai sehingga nantinya dapat menghasilkan analisa yang

tepat. Proses preprocessing dilakukan dengan melihat karakteristik data, dimana data tsunami yang digunakan memiliki 5 fitur dengan rentang nilai yang berbeda. Pada proses *clustering* dengan menggunakan algoritma DBSCAN, tahap perhitungan jarak antar titik data menggunakan fungsi *Euclidean Distance*. Perbedaan rentang nilai antar fitur akan sangat mempengaruhi hasil proses *clustering*. Fitur yang memiliki rentang nilai yang sangat besar akan mendominasi hasil perhitungan jarak antar titik data. Oleh sebab itu, perlu dilakukan proses normalisasi untuk menyamakan rentang nilai seluruh fitur yang digunakan dengan menggunakan fungsi Normalisasi Min-max sesuai dengan persamaan (3) berikut:

$$new_v = \frac{v - \min_x}{\max_x - \min_x} \cdot (new\_max_x - new\_min_x) + new\_min_x \quad (3)$$

dimana  $new_v$  merupakan nilai hasil normalisasi  $v$ ,  $\min_x$  nilai minimum untuk fitur  $x$ ,  $\max_x$  nilai maksimum fitur  $x$ ,  $new\_max_x$  nilai maksimum rentang nilai baru fitur  $x$ , dan  $new\_min_x$  nilai minimum rentang nilai fitur  $x$ .

Proses preprocessing berikutnya adalah memastikan tidak terdapat *missing value* didalam dataset yang digunakan. Dalam data tsunami yang diunduh dari situs NOAA, terdapat cukup banyak *missing value* pada beberapa fitur data. Beberapa metode dalam data mining dapat digunakan untuk menangani data dengan *missing value*, diantaranya dengan tidak menggunakan/menghapus data dengan *missing value* dari dataset, dengan mengganti dengan nilai tertentu (imputasi), yaitu dengan menggunakan nilai rata-rata fiturnya, dengan nilai modulus (nilai yang sering muncul), atau dengan menggunakan pendekatan algoritma *machine learning* untuk memprediksi *missing value*, seperti  $k$ -NN,  $k$ -Means, dsb. Pada beberapa penelitian (Rahman 2013, Mehala 2009) disimpulkan bahwa proses imputasi dengan menggunakan algoritma *machine learning* memperoleh hasil yang lebih baik dibandingkan dengan menggunakan nilai rata-rata ataupun modulus. Oleh sebab itu, pada penelitian ini digunakan algoritma  $k$ -NN untuk

mengganti *missing value* yang ada pada dataset tsunami seperti tampak pada Gambar 5.

```

Input :  $D$ , the set of training objects, the test object,  $z$ , which is a vector of
attribute values, and  $L$ , the set of classes used to label the objects
Output :  $c_z \in L$ , the class of  $z$ 
foreach object  $y \in D$  do
  | Compute  $d(z, y)$ , the distance between  $z$  and  $y$ ;
end
Select  $N \subseteq D$ , the set (neighborhood) of  $k$  closest training objects for  $z$ ;
 $c_z = \operatorname{argmax}_{v \in L} \sum_{y \in N} I(v = \operatorname{class}(c_y))$ ;
where  $I(\cdot)$  is an indicator function that returns the value 1 if its argument is true and
0 otherwise.

```

Gambar 5. Algoritma k-NN

Algoritma  $k$ -NN ( $k$ -Nearest Neighbour) merupakan algoritma klasifikasi dalam data mining yang digunakan untuk mengelompokkan data berdasarkan jumlah tetangga terdekatnya ( $k$ ). Nilai  $k$  ditentukan diawal proses klasifikasi, dimana untuk mendapatkan nilai  $k$  yang sesuai dilakukan dalam proses *training*. Proses klasifikasi dilakukan dengan menghitung jarak antara titik yang diuji dengan seluruh obyek yang ada dalam dataset. Proses selanjutnya adalah mencari sejumlah  $k$  obyek terdekat dengan data uji dengan cara mengurutkan obyek-obyek dalam dataset berdasarkan jarak terdekat dengan data uji. Tahap terakhir adalah menentukan kelas data dari data uji dengan melakukan voting atau menghitung jumlah kelas terbanyak dari  $k$  tetangga terdekat tersebut, dimana data uji akan diklasifikasikan berdasarkan jumlah voting kelas terbanyak dari  $k$  tetangga terdekatnya.

#### 4.2. Dataset Tsunami

Data yang digunakan dalam penelitian ini adalah data tsunami yang diunduh dari halaman situs NOAA (NGDC 2016). Data tersebut terdiri dari 2521 *records* dengan 45 fitur. Data yang akan digunakan dalam proses pengujian dibatasi hanya data tsunami yang terjadi pada 100 tahun terakhir, yaitu dari tahun 1915 – 2015 yang berjumlah 1046 *records*. Dari 45 fitur yang ada, hanya akan digunakan sebanyak 5 fitur yang berkorelasi dengan fokus dan tujuan utama dari penelitian ini, yaitu untuk memetakan potensi resiko yang ditimbulkan akibat tsunami. Fitur-fitur tersebut adalah *region\_code*, *distance\_from\_source*, *death\_description*, *damage\_description*, dan *houses\_destroyed\_description*.

### 5. HASIL DAN PEMBAHASAN

Eksperimen dilakukan dengan mengevaluasi kualitas hasil *clustering* algoritma DBSCAN menggunakan *Silhouette Coefficient* berdasarkan kombinasi masukan dua parameter yaitu  $\epsilon$  dan *minPts*. Hasil pengujian ditunjukkan pada table 1 berikut:

Tabel 1. Hasil Pengujian DBSCAN

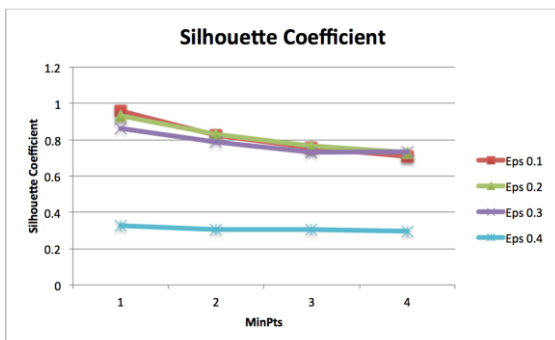
Ep s	Min Pts	Silhouette Coefficient	Jml Clust er	Jml Noi se	Jml Ter- clust er
0.1	1	0.96056649	81	0	355
0.1	2	0.827227086	36	44	311
0.1	3	0.755389076	25	68	287
0.1	4	0.708639518	20	83	272
0.2	1	0.933609473	60	0	355
0.2	2	0.830803492	30	35	320
0.2	3	0.764958847	23	57	298
0.2	4	0.730559072	19	69	286
0.3	1	0.861535215	56	0	355
0.3	2	0.786764287	31	25	330
0.3	3	0.731783924	20	43	312
0.3	4	0.735393967	18	58	297
0.4	1	0.325343599	10	0	355
0.4	2	0.308079386	3	7	348
0.4	3	0.303753325	2	9	346
0.4	4	0.296555435	1	12	343

Hasil *cluster* terbaik diperoleh dengan kombinasi input  $\epsilon = 0.1$  dan *minPts* = 1 dengan nilai *Silhouette Coefficient* mendekati nilai maksimum 1, yaitu 0.96056649. Hasil tersebut mengindikasikan bahwa obyek-obyek dalam dataset tercluster dengan baik, yaitu memiliki jarak *inter-cluster* (jarak antar satu *cluster* dengan *cluster* lainnya) tinggi dan jarak *intra-cluster* (jarak antar obyek dengan obyek lainnya didalam satu *cluster* yang sama) rendah. Hal tersebut menggambarkan bahwa setiap *cluster* terpisah cukup jauh dengan *cluster* yang lain sedang setiap anggota *cluster* memiliki kerapatan yang tinggi dengan obyek-obyek lainnya yang berada dalam satu *cluster* yang sama.



### 5.1. Pengaruh Nilai $\epsilon$ dan $minPts$ terhadap *Silhouette Coefficient*

Berdasarkan hasil *clustering* algoritma DBSCAN pada Gambar 6 dapat dilihat bahwa perubahan nilai masukan  $\epsilon$  dan  $minPts$  sangat berpengaruh terhadap kualitas hasil *clustering*. Nilai  $\epsilon$  dan  $minPts$  yang semakin besar akan semakin menurunkan kualitas hasil *clustering*. Hal tersebut dikarenakan pengaruh dari nilai  $\epsilon$  yang semakin besar maka akan semakin luas cakupan kerapatan suatu *cluster*, sehingga kemungkinan obyek yang berjauhan letaknya akan dikelompokkan dalam *cluster* yang sama sehingga semakin memperbesar rasio jarak intra-*cluster* hasil *clustering*. Sedang pengaruh nilai  $minPts$  terhadap hasil *cluster* adalah dengan semakin besar nilai  $minPts$  maka akan semakin sulit suatu obyek yang meskipun jaraknya saling berdekatan tetapi jika jumlah obyeknya tidak memenuhi syarat  $minPts$  (jumlah minimal obyek dalam satu *cluster*) maka tidak dimasukkan dalam satu *cluster* yang sama.

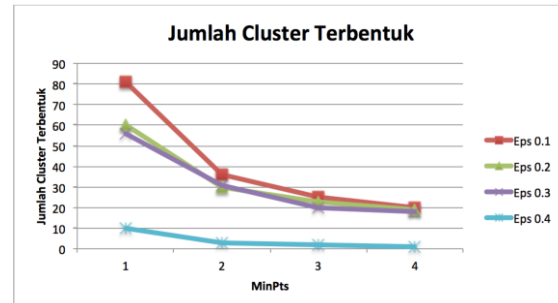


Gambar 6. Grafik Pengaruh Nilai  $\epsilon$  dan  $minPts$  terhadap Nilai *Silhouette Coefficient*

### 5.2. Pengaruh Nilai $\epsilon$ dan $minPts$ terhadap Jumlah *Cluster* Terbentuk

Nilai  $\epsilon$  yang semakin besar menggambarkan jarak cakupan kerapatan suatu obyek dengan obyek yang lain semakin luas sehingga kemungkinan menghasilkan *cluster* akan semakin besar. Hal tersebut dapat dilihat pada Gambar 7 dimana semakin besar nilai  $\epsilon$  maka semakin sedikit jumlah *cluster* yang terbentuk karena semakin banyak obyek yang terangkul dalam masing-masing *cluster*. Sedangkan semakin besar nilai  $minPts$  akan semakin memperkecil kemungkinan terbentuknya suatu *cluster*. Hal tersebut

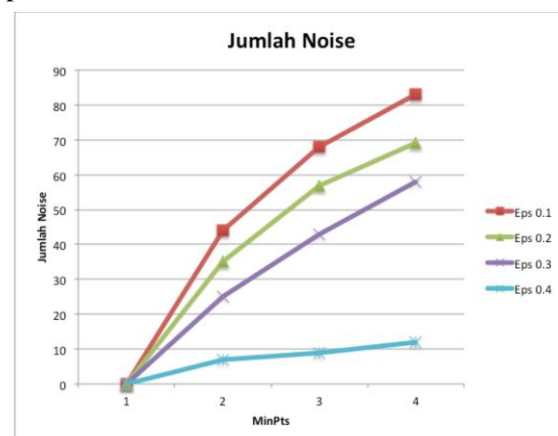
dikarenakan syarat jumlah minimal obyek yang berdekatan dalam *cluster* semakin besar sehingga membutuhkan lebih banyak obyek untuk membentuk suatu *cluster*, demikian pula sebaliknya.



Gambar 7. Grafik Pengaruh Nilai  $\epsilon$  dan  $minPts$  terhadap Jumlah *Cluster* Terbentuk

### 5.3. Pengaruh Nilai $\epsilon$ dan $minPts$ terhadap Jumlah Noise

Noise atau *outlier* merupakan obyek yang memiliki karakteristik berbeda dibandingkan dengan sebagian besar obyek dalam dataset. Salah satu kelebihan dari algoritma DBSCAN adalah mampu dengan baik mendeteksi noise karena konsep dari algoritma DBSCAN adalah mengelompokkan obyek berdasarkan kerapatannya (density-based) dengan obyek yg lain, sehingga akan mengabaikan obyek dengan karakteristik yang tidak mirip dengan obyek disekitarnya. Berdasarkan hasil *clustering* dataset tsunami pada Tabel 1 sebelumnya, maka analisa pengaruh nilai  $\epsilon$  dan  $minPts$  terhadap jumlah noise dapat dilihat pada Gambar 8 berikut:



Gambar 8. Grafik Pengaruh Nilai  $\epsilon$  dan  $minPts$  terhadap Jumlah Noise

Dari Gambar 8 dapat dilihat bahwa semakin besar nilai *minPts* maka akan semakin tinggi jumlah noise yang dihasilkan. Hal tersebut dikarenakan kemampuan membentuk *cluster* dalam proses *clustering* menggunakan DBSCAN semakin kecil, sehingga obyek-obyek yang tidak dapat membentuk *cluster* akan dikategorikan sebagai noise. Hal tersebut juga tergambar dari pengaruh nilai  $\epsilon$  yang berkebalikan dengan *minPts*. Semakin besar nilai  $\epsilon$  justru mengakibatkan semakin menurunnya jumlah noise yang dihasilkan. Hal tersebut disebabkan oleh pengaruh dari nilai  $\epsilon$  yang semakin besar akan semakin memperluas cakupan area *cluster* sehingga kemungkinan pembentukan *cluster* akan semakin besar dan kemungkinan suatu obyek masuk kedalam suatu *cluster* juga semakin besar.

#### 5.4. Pembahasan dan Analisa Potensi Resiko Tsunami

Berdasarkan hasil analisa proses *clustering* dengan menggunakan algoritma DBSCAN pada bab sebelumnya, maka dapat disimpulkan bahwa algoritma DBSCAN dapat dengan baik mengelompokkan daerah terdampak tsunami berdasarkan karakteristik potensi resiko yang ditimbulkan. Hal tersebut dapat dilihat dari tingginya nilai *Silhouette Coefficient* dari hasil *clustering* dataset tsunami. Hasil yang terbaik diperoleh dengan kombinasi nilai  $\epsilon$  dan *minPts* yaitu 0.1 dan 1. Maka dengan menggunakan hasil dari *clustering* tersebut (Gambar 5.4), dapat kemudian dilakukan pemetaan potensi resiko tsunami terhadap masing-masing area terdampak tsunami berdasarkan karakteristik obyek-obyek yang ada dalam satu *cluster*. Dari hasil *clustering* menggunakan algoritma DBSCAN, maka didapatkan 81 area yang berbeda (yang digambarkan dalam bentuk *cluster*) berdasarkan karakteristik potensi resiko tsunami yang ditimbulkan. Proses selanjutnya adalah memberikan label pada cluster berdasarkan karakteristik data yang ada didalam masing-masing cluster untuk menentukan tingkat potensi resiko tsunami sehingga dapat digunakan dalam proses pengambilan keputusan atau kebijakan terkait penanganan atau mitigasi bencana tsunami.



Gambar 9. Hasil *Clustering* Algoritma DBSCAN

## 6. KESIMPULAN

Berdasarkan hasil eksperimen penelitian ini, Algoritma DBSCAN dapat diimplementasikan untuk mengelompokkan potensi resiko tsunami ke dalam 81 kelompok tingkat potensi resiko tsunami. *Clustering* menggunakan algoritma DBSCAN menghasilkan hasil yang cukup bagus dengan cukup tingginya nilai *Silhouette Coefficient* pada proses pengujian sehingga bisa digunakan sebagai bahan rekomendasi untuk analisa potensi resiko tsunami.

## 7. DAFTAR PUSTAKA

- A. KARAMI AND R. JOHANSSON, "Choosing dbscan parameters automatically using differential evolution," *Int. J. Comput. Appl.*, vol. 91, no. 7, 2014.
- B. MEHALA, P. R. J. THANGAIAH, and K. VIVEKANANDAN, "Selecting scalable algorithms to deal with missing values," *Int. J. Recent Trends Eng.*, vol. 1, no. 2, pp. 80–83, 2009.
- E. L. GEIST and T. PARSONS, "Assessment of source probabilities for potential tsunamis affecting the U.S. Atlantic coast," *Mar. Geol.*, vol. 264, no. 1–2, pp. 98–108, Aug. 2009.
- K. UCHIDA, S. NOGAMI, M. TAKEMATSU, and J. HONDA, "Tsunami Simulation Based on Dijkstra Algorithm," 2014, pp. 114–119.
- L. KAUFMAN and P. J. ROUSSEEUW, *Finding groups in data: an introduction to cluster analysis*, vol. 344. John Wiley & Sons, 2009.
- L. MA, L. GU, B. LI, S. YI QIAO, and J. WANG, "G-DBSCAN: An Improved

- DBSCAN Clustering Method Based On Grid,” *Adv Sci Technol Lett*, vol. 74, pp. 23–28, 2014.
- M. ESTER, H.-P. KRIEGEL, J. SANDER, and X. XU, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Kdd*, 1996, vol. 96, pp. 226–231.
- M. M. RAHMAN and D. N. DAVIS, “Machine Learning-Based Missing Value Imputation Method for Clinical Datasets,” in *IAENG Transactions on Engineering Technologies*, vol. 229, G.-C. Yang, S. Ao, and L. Gelman, Eds. Dordrecht: Springer Netherlands, 2013, pp. 245–257.
- M. T. ELBATTA and W. M. ASHOUR, “A dynamic method for discovering density varied clusters,” *Int J. Signal Process. Image Process. Pattern Recognit.*, vol. 6, no. 1, pp. 123–134, 2013.
- NGDC/WDS, “Global Historical Tsunami Database”. [Online]. Available: [https://www.ngdc.noaa.gov/hazard/tsu\\_db.shtml](https://www.ngdc.noaa.gov/hazard/tsu_db.shtml). [Accessed: 01-Mar-2016]
- R. B. S. YADAV, T. M. TSAPANOS, J. N. TRIPATHI, and S. CHOPRA, “An evaluation of tsunami hazard using Bayesian approach in the Indian Ocean,” *Tectonophysics*, vol. 593, pp. 172–182, May 2013.
- R. G. PRABHUDESAI, A. JOSEPH, P. MEHRA, Y. AGARVADEKAR, S. TENGALI, and V. KUMAR, “Detection of 12 th September 2007 Sumatra Tsunami at Goa and Kavaratti Island,” in *OCEANS 2008-MTS/IEEE Kobe Techno-Ocean*, 2008, pp. 1–8.
- S. ACM, “2014 SIGKDD Test of Time Award.” [Online]. Available: <http://www.kdd.org/blog/2014-sigkdd-test-time-award>. [Accessed: 11-Mar-2015].