

Analisis Metode DBSCAN (Density-Based Spatial Clustering of Application with Noise) dalam Mendeteksi Data Outlier

Dedy Armiady

¹ Fakultas Ilmu Komputer, Program Studi Informatika, Universitas Almuslim, Bireuen, Indonesia

Email: dedy.armiady@gmail.com

Email Koresponding: dedy.armiady@gmail.com

Submitted 05-11-2022; Accepted 30-12-2022; Published 30-12-2022

Abstrak

Data outlier merupakan sebuah data berbeda dari sekelompok data yang ada dalam sebuah dataset. Data outlier akan berdampak kepada pembiasan hasil analisis data, apabila tidak ditangani secara baik. Berbagai pendekatan dapat dilakukan untuk melakukan deteksi terhadap data outlier, salah satunya yaitu melalui metode klastering (pengelompokan data). DBSCAN (Density-Based Spatial Clustering of Application with Noise) merupakan salah satu metode klastering yang mampu menemukan data outlier dalam kumpulan data. DBSCAN bekerja dengan cara menentukan klaster berdasarkan kepadatan data, dengan menggunakan parameter epsilon (jangkauan) dan MinPts (minimal point untuk membentuk suatu klaster). Adapun penelitian ini bertujuan untuk menguji beberapa model DBSCAN yang memiliki parameter epsilon dan MinPts yang berbeda. Adapun model yang digunakan terdiri dari 3 model, dengan rincian: Model 1 (eps=0,2, MinPts=5), Model 2 (eps=0,3, MinPts=5) dan Model 3 (eps=0,4, MinPts=5). Dataset yang digunakan merupakan dataset yang digenerate melalui fitur paint data pada tool Orange Data Mining, dengan 2 variabel (x dan y), dengan jumlah record sebanyak 1051 baris data. Adapun hasil yang didapatkan yaitu semua model yang diuji didapati bahwa ada 1 point data yang dianggap sebagai outlier, yaitu data bernilai $x=0,370007$ dan $y=0,410475$. Disamping itu, dari penelitian yang dilakukan ini juga dapat ditarik kesimpulan bahwa nilai epsilon berpengaruh terhadap jumlah klaster yang terbentuk. Semakin tinggi nilai epsilon, maka semakin kecil jumlah klaster yang mungkin terbentuk.

Kata Kunci: Data Outlier; DBSCAN; Data Mining; Epsilon; Klastering; MinPts.

Abstract

Data outlier is data that is different from a group of data in a dataset. Data outlier will have an impact on the refraction of data analysis results, if not handled properly. Various approaches can be taken to detect data outlier, one of which is through the clustering method (grouping data). DBSCAN (Density-Based Spatial Clustering of Application with Noise) is a clustering method that is able to find data outlier in a data set. DBSCAN works by determining clusters based on data density, using the parameters epsilon (range) and MinPts (minimum points to form a cluster). This study aims to test several DBSCAN models that have different epsilon and MinPts parameters. The model used consists of 3 models, with details: Model 1 (eps=0,2, MinPts=5), Model 2 (eps=0,3, MinPts=5) and Model 3 (eps=0,4, MinPts=5). The dataset used is a dataset generated through the paint data feature on the Orange Data Mining tool, with 2 variables (x and y), with a total of 1051 data lines of records. The results obtained are that all the tested models found that there is 1 data point that is considered an outlier, namely the data is worth $x = 0.370007$ and $y = 0.410475$. In addition, from this research, it can also be concluded that the epsilon value affects the number of clusters formed. The higher the epsilon value, the smaller the number of clusters that may be formed.

Keywords: Data Outlier; DBSCAN; Data Mining; Epsilon; Clustering; MinPts.

1. PENDAHULUAN

Data outlier merupakan suatu observasi yang memiliki nilai ekstrim dibandingkan sebagian besar nilai lain dalam suatu dataset [1]–[3]. Maksud dari nilai ekstrim adalah sebuah data memiliki nilai yang jauh berbeda dari kumpulan data yang ada. Misalnya data suhu udara pada jaringan *Internet of Things* yang dikumpulkan oleh sensor temperatur yang biasanya memiliki rentang 16°C hingga 45°C , ternyata didapatkan satu record data dari sensor tersebut ada yang bernilai 95°C , maka data tersebut dianggap tidak masuk akal dan dianggap sebagai data outlier [4], [5]. Data outlier memiliki dampak yang buruk terhadap hasil dari analisis data. Misalnya dalam penelitian dan pengolahan data penelitian, adanya data outlier akan menyebabkan bias pada hasil penelitian yang dilakukan [6]. Oleh karena itu, data outlier harus mendapatkan penanganan khusus, misalnya dilakukan transformasi terhadap data outlier tersebut dengan menurunkan value nya agar tidak terlalu jauh dengan kumpulan data yang ada, atau menghilangkan record tersebut apabila value nya terlalu jauh dari sebagian besar observasi yang ada [7]–[9].

Beberapa metode deteksi outlier telah diusulkan, seperti dengan menggunakan regression model, histograms, kernel function, Local Outlier Factor (LOF) dan lain sebagainya [10]. Salah satu metode lain untuk mendeteksi outlier yang dapat digunakan yaitu dengan melalui pendekatan berbasis clustering [11]–[14]. Dalam penelitian ini digunakan metode DBSCAN (Density-Based Spatial Clustering of Application with Noise). Metode DBSCAN merupakan salah satu metode pengelompokan data dan termasuk dalam algoritma unsupervised learning [15]. Pada metode DBSCAN, data dikelompokkan dengan menerapkan konsep kepadatan data (density). Pengelompokan data pada metode DBSCAN dilakukan dengan menggunakan parameter epsilon (jarak antar point) dan MinPts (minimal point untuk membentuk suatu klaster). Langkah awal metode DBSCAN adalah mengambil random point untuk dijadikan sebagai centroid (pusat klaster) sementara, kemudian dihitung jarak ke titik terdekat dengan euclidean distance sesuai dengan parameter epsilon yang ditentukan [16]. Jika point-point yang ada di dekat pusat klaster sementara dapat dijangkau berdasarkan nilai epsilon, maka point tersebut dianggap sebagai anggota klaster sementara tersebut [17]. Dan apabila data lainnya yang ada disekitar

klaster tersebut terjangkau oleh epsilon, dan klaster sementara tersebut memiliki jumlah point minimal sesuai dengan nilai MinPts yang ditentukan, maka satu klaster baru terbentuk dengan anggota yang terdiri dari point-point yang terjangkau oleh epsilon yang ditentukan. Langkah selanjutnya adalah mengulang langkah-langkah tersebut hingga semua data dikelompokkan ke dalam klaster masing-masing. Adapun data yang tidak terjangkau oleh nilai epsilon dan tidak masuk dalam klaster manapun, serta tidak bisa dimasukkan ke dalam klaster baru karena tidak mencapai minimal point (MinPts) maka ditetapkan sebagai data outlier [18].

Dalam penelitian ini, akan dilakukan analisis terhadap metode DBSCAN dalam mendeteksi data outlier. Penelitian ini menggunakan dataset yang di generate dengan menggunakan tools Orange Data Mining. Dataset berupa data acak yang terdiri dari dua variabel, dengan tujuan untuk memudahkan dalam melakukan visualisasi data. Penelitian ini menggunakan 3 model DBSCAN dengan nilai parameter epsilon dan MinPts yang berbeda untuk melihat perbedaan hasil klaster data serta melihat pola deteksi outlier yang didapatkan.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Adapun tahapan yang dilalui untuk melakukan penelitian ini adalah sebagai berikut:

a. Studi Pustaka

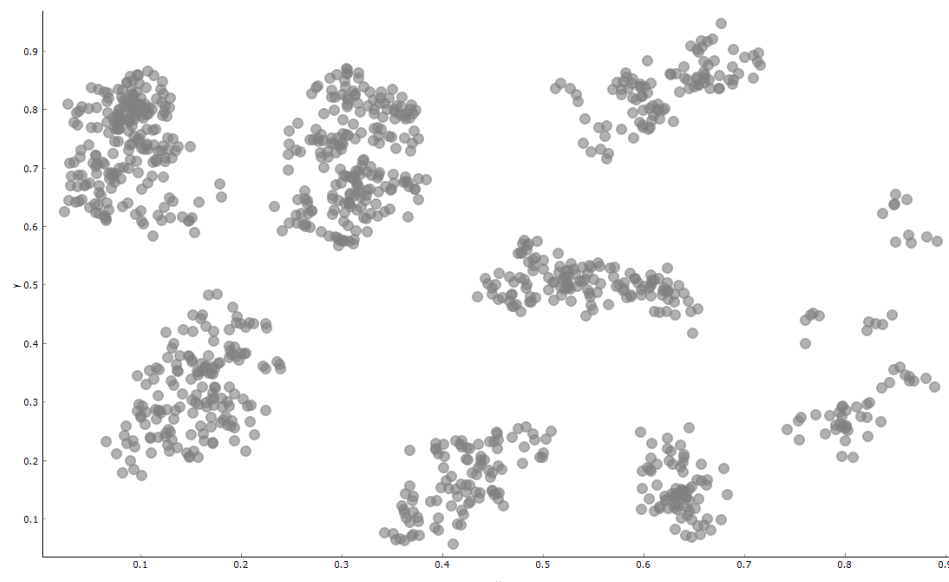
Pada tahapan ini, penulis melakukan studi pustaka untuk mencari informasi ilmiah terkait data outlier melalui berbagai sumber, baik buku, jurnal ilmiah, publikasi hasil konferensi dan sumber-sumber bacaan lainnya. Pada tahapan ini, informasi terkait pendekatan-pendekatan yang digunakan untuk mendeteksi data outlier juga dikumpulkan.

b. Identifikasi Masalah dan Penentuan Fokus Penelitian

Pada tahap ini, penulis melakukan identifikasi masalah untuk mengetahui permasalahan apa saja yang ada terkait dengan data outlier. Pada tahap ini dilakukan analisis dan pemetaan terhadap masalah yang ditimbulkan dengan adanya outlier. Pada tahap ini juga ditentukan fokus penelitian, dimana dalam penelitian ini akan dilakukan analisis terhadap beberapa model DBSCAN sesuai dengan parameter input epsilon dan MinPts untuk menemukan pola klaster dan temuan terhadap data outlier.

c. Pengumpulan Data

Adapun terkait dataset yang digunakan dalam penelitian ini menggunakan dataset yang digenerate dengan menggunakan tool Orange Data Mining melalui fitur Paint Data. Data yang digenerate yaitu data dengan total record sebanyak 1050 record dengan 2 variabel x dan y, dimana tujuannya yaitu agar mudah dalam melakukan visualisasi data. Adapun visualisasi data hasil generate dapat dilihat pada gambar berikut:



Gambar 1. Dataset yang digunakan

Grafik tersebut merupakan dataset yang divisualisasikan dalam bentuk scatter plot, dimana data yang digenerate secara visual membentuk beberapa klaster.

d. Analisis Model DBSCAN

Pada tahap ini, dilakukan eksperimen dengan menguji pengelompokan dataset, dimana pada tahap ini dibangun 3 model DBSCAN dengan nilai parameter input epsilon dan MinPts berbeda. Adapun rincian parameter untuk tiap model dapat dilihat pada tabel berikut:

Tabel 1. Rincian parameter untuk tiap model

Model DBSCAN	Nilai Epsilon	Nilai MinPts
Model 1	0,2	5
Model 2	0,3	5
Model 3	0,4	5

2.2 Metode DBSCAN

DBSCAN (Density-Based Spatial Clustering of Application with Noise) merupakan algoritma klastering berbasis kepadatan [19]. DBSCAN menggunakan konsep density-connected, dimana klaster dibentuk berdasarkan kepadatan data yang saling terhubung dengan menggunakan paramter epsilon dan MinPts [20]. Namun kelebihan metode ini dibandingkan metode lain seperti K-Means adalah kemampuannya dalam mendeteksi data outlier [21], [22]. Data yang terjangkau oleh epsilon dan memenuhi syarat minimal point untuk sebuah klaster, maka data-data tersebut dimasukkan kedalam klaster tersebut. Namun data yang tidak terjangkau, ataupun tingkat kepadatan datanya tidak sesuai dengan paramter MinPts, maka akan dianggap sebagai data outlier. Pada metode DBSCAN terdapat beberapa definisi sebagai berikut:

- a. Epsilon-Neighborhood dari satu titik

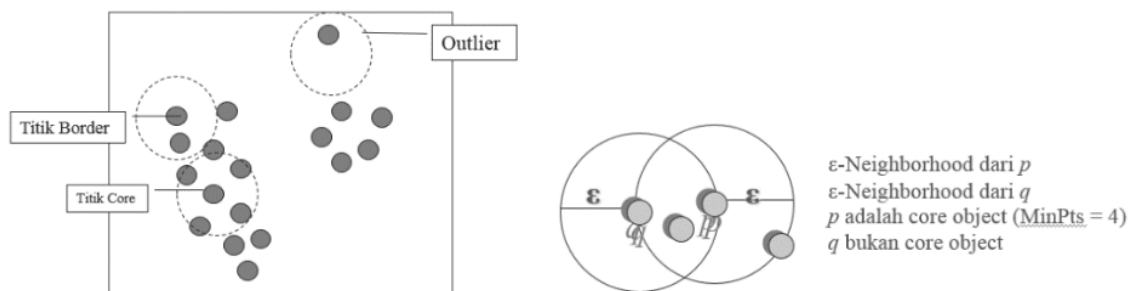
$$N_{Eps}(p) = \{q \in D \mid dist(p, q) < eps\} \quad (1)$$

Suatu data yang masuk ke dalam suatu klaster, harus mempunyai minimal satu data lain yang terletak lebih dekat ke data tersebut, berdasarkan nilai epsilon [18]

- b. Directly Density-Reachable

$$p \in N_{Rps}(Q) \quad (2)$$

Border point masih dianggap sebagai anggota suatu klaster jika border point memiliki epsilon neighborhood dari sebuah q (core point) [18]. Adapun ilustrasinya dapat dilihat pada gambar berikut:



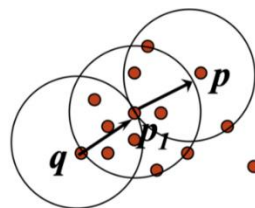
Gambar 2. Border Point dan Core Point

Supaya q point menjadi core point, point tersebut harus memiliki MinPts (minimal point untuk membentuk klaster). Adapun kondisinya ditunjukkan melalui persamaan berikut:

$$|N_{Eps}(q)| \geq MinPts \quad (3)$$

- c. Density-Reachable

Suatu data harus bersifat density-reachable dari suatu data q, berdasarkan parameter epsilon dan MinPts [18]. Adapun ilustrasinya dapat dilihat pada gambar berikut:

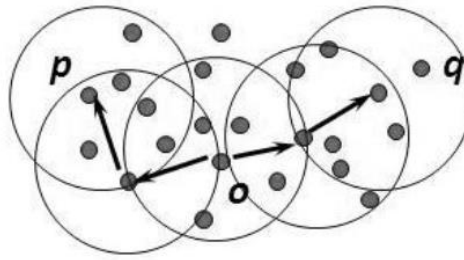


Gambar 3. Density-Reachable

Jika suatu rantai data p_1, p_2, \dots, p_n , $p_1=q$ dan $p_n=p$, dimana p_{i+1} merupakan density reachable langsung dari p_i .

- d. Density-Connected

Apanila dua border-point akan dimasukkan ke dalam suatu klaster, namun dua border-point tersebut tidak membagi core-point spesifik, maka titik tersebut tidak dianggap density-reachable antar satu dengan lainnya [18]. Adapun gambarannya dapat dilihat pada gambar berikut:



Gambar 4. Density-Connected

Kondisi seperti ini harus ada core-point q yang menjadi density reachable dari kedua border-point tersebut. Sebuah titik p merupakan density connected ke titik q, sesuai dengan parameter epsilon dan MinPts jika terdapat satu titik o dimana p dan q merupakan density reachable dari o sesuai dengan parameter epsilon dan MinPts.

e. **Klaster**

Point q merupakan bagian dari klaster, apabila satu titik merupakan anggota klaster C dan point q merupakan density reachable dari point p berdasarkan epsilon dan MinPts yang ditentukan [18].

f. **Noise**

Merupakan data yang tidak masuk dalam klaster manapun, atau data yang tidak memenuhi syarat untuk dimasukkan ke dalam sebuah klaster sesuai dengan parameter epsilon dan MinPts [18].

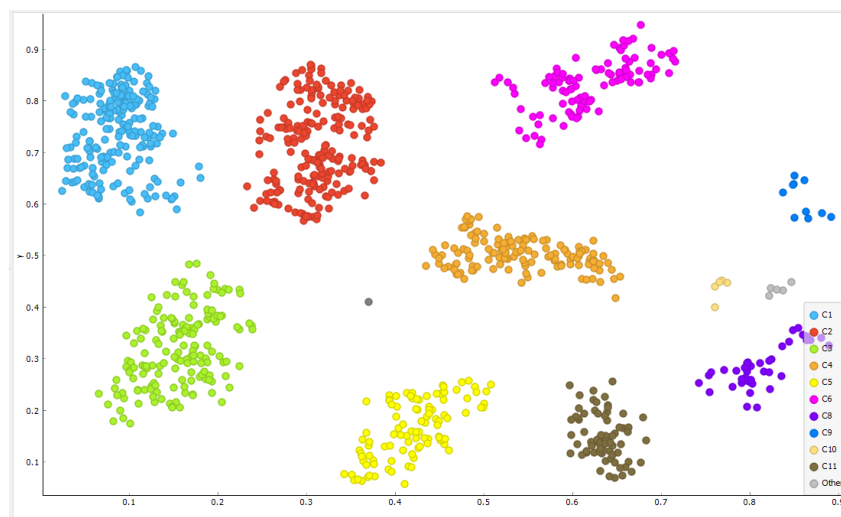
3. HASIL DAN PEMBAHASAN

3.1 Hasil Penelitian

Penelitian ini dilakukan dengan tahapan yang telah dijelaskan pada bagian sebelumnya, hasil yang didapatkan yaitu adanya dataset yang di generate dengan tool Orange Data Mining. Adapun analisis datanya juga menggunakan tool Orange Data Mining, dimana hasil yang didapatkan berikutnya berupa 3 model DBSCAN dengan parameter yang berbeda seperti yang telah disebutkan pada bagian sebelumnya. Kemudian pengujian tiap model dengan menggunakan dataset yang ada dilakukan sehingga didapatkan hasil sebagai berikut:

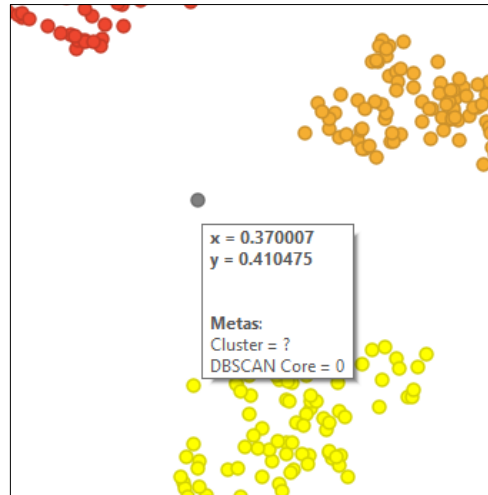
3.1.1 Hasil Klaster dengan Model 1

Model pertama menggunakan nilai epsilon = 0,2 dan MinPts = 5 dengan menggunakan jarak euclidean, maka didapatkan hasil klaster seperti pada gambar berikut:



Gambar 5. Hasil Klaster Model 1

Pada gambar hasil klaster model 1, dapat dilihat bahwa model 1 DBSCAN dengan epsilon = 0,2 dan MinPts = 5 membentuk total 11 klaster (C1-C11) yang ditandai dengan warna data point yang berbeda. Adapun pada hasil klaster tersebut, klaster 7 (C7) hanya memiliki 5 point sebagai anggota klasternya, hal ini dikarenakan MinPts = 5, sehingga 5 point tersebut yang density-reachable dapat dianggap sebagai salah satu klaster. Namun berbeda jika MinPts = 6, maka semua anggota klaster C7 akan dianggap sebagai outlier. Disamping itu juga ada point yang berwarna grey yang dianggap sebagai outlier dikarenakan jaraknya yang jauh dari kumpulan data lainnya seperti pada gambar berikut:

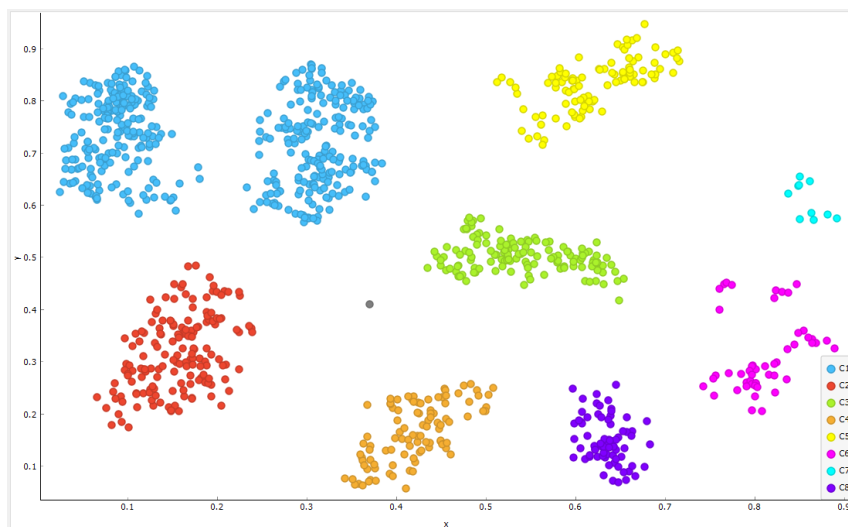


Gambar 6. Data Outlier

Pada gambar tersebut satu point yang berwarna grey dianggap sebagai data outlier, hal ini dikarenakan data tersebut tidak terjangkau oleh core point atau border point yang ada berdasarkan nilai epsilon = 0,2. Adapun pada kluster C9 yang berwarna biru tua dapat diperhatikan bahwa ada sekitar 9 data menjadi satu kluster, dikarenakan MinPts = 5 memenuhi kualifikasi untuk data-data tersebut sehingga DBSCAN melakukan assign data tersebut sebagai kluster C8. Namun apabila parameter MinPts dirubah menjadi 4, maka yang terjadi adalah jumlah kluster menjadi semakin banyak.

3.1.2 Hasil Kluster dengan Model 2

Adapun pada model 2 menggunakan nilai parameter epsilon = 0,3 serta MinPts = 5, dimana didapatkan hasil kluster yang dapat dilihat pada gambar berikut:

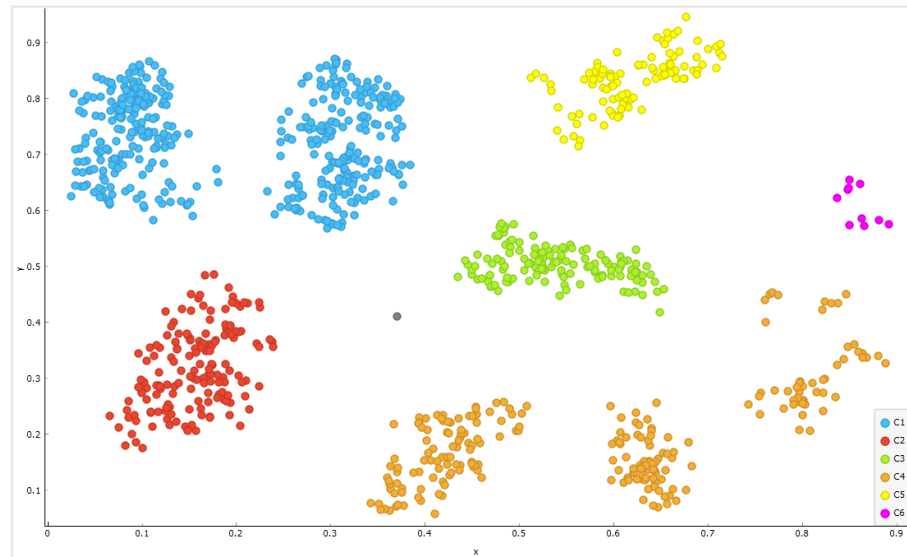


Gambar 7. Hasil Kluster Model 2

Pada gambar tersebut dapat dilihat hasilnya yaitu dengan menggunakan model 2 DBSCAN dengan parameter epsilon = 0,3 dan MinPts = 5, maka kluster yang dibentuk sejumlah 8 kluster (C1-C8). Pada hasil kluster tersebut juga terdapat data outlier yang sama dengan hasil pada kluster sebelumnya dengan model DBSCAN 1. Namun perbedaannya dengan hasil sebelumnya terletak pada kluster C2, dimana kelompok data C2 pada model 1 berubah menjadi kelompok kluster C1 pada hasil kluster 2. Hal ini disebabkan oleh pengaruh nilai epsilon yang semakin besar, sehingga border point yang ada pada C1 ikut menjangkau point yang ada pada kluster C2, sehingga semua data pada kluster C2 sekarang menjadi anggota dari kluster C1. Perbedaan lain juga terletak pada kluster C7, C8 dan C10, dimana pada model 1 data-data tersebut di assign menjadi 3 kluster (C7, C8 dan C10), namun pada model 2 data-data tersebut diassign menjadi 1 kluster, yaitu kluster C6.

3.1.3 Hasil Kluster dengan Model 3

Adapun model 3 DBSCAN menggunakan nilai parameter epsilon = 0,4 serta MinPts = 5, dimana hasil kluster untuk model ini dapat dilihat pada gambar berikut:



Gambar 8. Hasil Kluster Model 3

Pada gambar tersebut dapat dilihat hasil kluster dengan nilai epsilon = 0,4 dan MinPts = 5, dimana pada model ini menghasilkan 6 kluster (C1-C6). Adapun perbedaan hasil pada model 3 ini dengan model 2 yaitu pada jumlah klusternya, dimana semakin besar nilai epsilon yang diberikan maka semakin kecil jumlah kluster yang mungkin terbentuk. Perbedaan yang paling signifikan adalah pada kluster C4, dimana pada hasil model 2 terbentuk 3 kluster (C4, C6 dan C8), namun pada model 3 semua data tersebut dianggap sebagai anggota kluster yang sama yaitu kluster C4. Hal ini dikarenakan nilai epsilon yang semakin besar berpengaruh kepada density-reachable yang semakin luas, sehingga secara langsung berpengaruh kepada pengurangan jumlah kluster yang terbentuk.

4. KESIMPULAN

Berdasarkan penelitian dan pengujian model daripada DBSCAN dengan parameter epsilon dan MinPts yang berbeda, maka dapat diambil kesimpulan bahwa DBSCAN dapat dipakai untuk mendeteksi data outlier dikarenakan konsep density-reachable yang digunakan dapat mengelompokkan suatu data sebagai data outlier berdasarkan jarak dengan point-point lainnya. Adapun data yang dianggap sebagai data outlier dalam dataset yang digunakan ini berjumlah 1 data dengan nilai variable $x=0,370007$ dan $y=0,410475$, dimana data tersebut ditandai dengan warna grey pada scatter plot, dimana data tersebut terdeteksi sebagai outlier pada semua model DBSCAN yang diuji. Kesimpulan lainnya yaitu, terdapat perbedaan hasil kluster antara model DBSCAN yang digunakan, yaitu pada jumlah kluster yang semakin berkurang, berdasarkan nilai epsilon-nya. Dengan kata lain, semakin bertambah nilai epsilon, maka kemungkinan berkurangnya jumlah kluster semakin besar. Hal ini dikarenakan jangkauan dari core point atau border point ke point-point lainnya dalam dataset menjadi lebih besar. Namun berdasarkan pengamatan penulis pada grafik hasil kluster, metode DBSCAN masih dinilai kurang optimal dalam menentukan kluster, dikarenakan ada kluster yang titik awalnya dengan titik akhirnya dianggap terlalu jauh jaraknya, hanya dikarenakan border point yang menjangkau point-point lain yang ada disekitarnya. Sehingga berdasarkan hal tersebut, penulis menyarankan untuk penelitian berikutnya perlu dikembangkan metode yang memberikan nilai epsilon dinamis sesuai dengan kluster yang akan dibentuk.

REFERENCES

- [1] L. Sun, K. Zhou, X. Zhang, and S. Yang, "Outlier Data Treatment Methods Toward Smart Grid Applications," IEEE Access, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2852759.
- [2] O. Alghushairy, R. Alsini, T. Soule, and X. Ma, "A review of local outlier factor algorithms for outlier detection in big data streams," Big Data and Cognitive Computing, vol. 5, no. 1. 2021. doi: 10.3390/bdcc5010001.
- [3] L. Lyutikova and E. Shmatova, "Using a logical derivative to analyze data outlier," in Procedia Computer Science, 2020, vol. 169. doi: 10.1016/j.procs.2020.02.187.
- [4] H. Ghallab, H. Fahmy, and M. Nasr, "Detection outliers on internet of things using big data technology," Egyptian Informatics Journal, vol. 21, no. 3, 2020, doi: 10.1016/j.eij.2019.12.001.
- [5] R. Zhu et al., "KNN-Based Approximate Outlier Detection Algorithm over IoT Streaming Data," IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2977114.
- [6] T. Schamberger, F. Schuberth, J. Henseler, and T. K. Dijkstra, "Robust partial least squares path modeling," Behaviormetrika, vol. 47, no. 1, 2020, doi: 10.1007/s41237-019-00088-2.
- [7] J. Frost, "Guidelines for Removing and Handling Outliers in Data," Statistics by Jim, 2018.
- [8] T. v. Pollet and L. van der Meij, "To Remove or not to Remove: the Impact of Outlier Handling on Significance Testing in Testosterone Data," Adaptive Human Behavior and Physiology, vol. 3, no. 1, 2017, doi: 10.1007/s40750-016-0050-z.

- [9] H. Torkey, E. Ibrahim, E. E. D. Hemdan, A. El-Sayed, and M. A. Shouman, "Diabetes classification application with efficient missing and outliers data handling algorithms," *Complex and Intelligent Systems*, vol. 8, no. 1, 2022, doi: 10.1007/s40747-021-00349-2.
- [10] J. Zhang, "Advancements of Outlier Detection: A Survey," *ICST Transactions on Scalable Information Systems*, vol. 13, no. 1, 2013, doi: 10.4108/trans.sis.2013.01-03.e2.
- [11] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Trans Knowl Discov Data*, vol. 10, no. 1, 2015, doi: 10.1145/2733381.
- [12] D. Phamtoan, K. Nguyenhuu, and T. Vovan, "Fuzzy clustering algorithm for outlier-interval data based on the robust exponent distance," *Applied Intelligence*, vol. 52, no. 6, 2022, doi: 10.1007/s10489-021-02773-w.
- [13] H. Tong and C. Tortora, "Model-based clustering and outlier detection with missing data," *Adv Data Anal Classif*, vol. 16, no. 1, 2022, doi: 10.1007/s11634-021-00476-1.
- [14] M. J. Bah, H. Wang, L. H. Zhao, J. Zhang, and J. Xiao, "EMM-CLOUDS: An Effective Microcluster and Minimal Pruning CLustering-Based Technique for Detecting Outliers in Data Streams," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/9178461.
- [15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.
- [16] M. Ramadhani and D. Fitriana, "Implementation of data mining analysis to determine the tuna fishing zone using DBSCAN algorithm," *Int J Mach Learn Comput*, vol. 9, no. 5, 2019, doi: 10.18178/ijmlc.2019.9.5.862.
- [17] H. T. Lee, J. S. Lee, H. Yang, and I. S. Cho, "An AIS data-driven approach to analyze the pattern of ship trajectories in ports using the DBSCAN algorithm," *Applied Sciences (Switzerland)*, vol. 11, no. 2, 2021, doi: 10.3390/app11020799.
- [18] F. Huang et al., "Research on the parallelization of the DBSCAN clustering algorithm for spatial data mining based on the Spark platform," *Remote Sens (Basel)*, vol. 9, no. 12, 2017, doi: 10.3390/rs9121301.
- [19] M. M. Putri, C. Dewi, E. Permata Siam, G. Asri Wijayanti, N. Aulia, and R. Nooraeni, "Comparison of DBSCAN and K-Means Clustering for Grouping the Village Status in Central Java 2020," *Jurnal Matematika, Statistika & Komputasi*, vol. 17, no. 3, 2021.
- [20] A. Kristianto, "Analisa Performa K-Means dan DBSCAN dalam Clustering Minat Penggunaan Transportasi Umum," *Elkom : Jurnal Elektronika dan Komputer*, vol. 14, no. 2, 2021, doi: 10.51903/elkom.v14i2.551.
- [21] M. Verma, M. Srivastava, N. Chack, A. K. Diswar, and N. Gupta, "A Comparative Study of Various Clustering Algorithms in Data Mining," *International Journal of Engineering Research and Applications* www.ijera.com, vol. 2, no. 3, 2012.
- [22] T. H. F. Khan, N. N. Alleema, N. Yadav, S. Mishra, and A. Shahi, "Text document clustering using K-means and dbscan by using machine learning," *Int J Eng Adv Technol*, vol. 9, no. 1, 2019, doi: 10.35940/ijeat.A2040.109119.