ISSN: 2503-054X
Vol. 6 No: 2, Oktober 2021

Implementasi Algoritma *DBScan* dalam Pemngambilan Data Menggunakan *Scatterplot*

Devi Fitrianah
Universitas Mercu Buana
Jakarta, Indonesia
devi.fitrianah@mercubuana.ac.id

Wawan Gunawan
Universitas Mercu Buana
Jakarta, Indonesia
wawan.gunawan@mercubuana.ac.id

Rizky Algian Kurniaputra
Universitas Mercu Buana
Jakarta, Indonesia
41513010053@student.mercubuana.ac.id

Abstract— *Seiring dengan perkembangan teknologi informasi dan komunikasi, semakin banyak data yang digunakan dalam suatu pemecahan masalah. Tetapi, dengan banyaknya data yang ada sangat sulit mencari informasi yang diinginkan. Oleh karena itu, dilakukan data mining untuk mengekstraksi pengetahuan secara otomatis dari data berukuran besar dengan cara mencari pola-pola menarik yang terkandung di dalam data tersebut. Dalam penelitian ini, peneliti menggunakan algoritma DBSCAN dalam penelitiannya. Data yang digunakan adalah data spasial mahasiswa Universitas Mercu Buana. Dari data ini, peneliti mengambil informasi scatterplot yang terbentuk, lalu dengan algoritma DBSCAN untuk melihat cluster yang terbentuk, dan melakukan validasi dengan Silhouette Index. Dari penelitian ini dapat disimpulkan bahwa algoritma DBSCAN berhasil diimplementasikan pada data mahasiswa Universitas Mercu Buana. Dan hasil pengujian dari implementasi algoritma DBSCAN dipengaruhi oleh dua nilai parameter yaitu Minimum Points, dan Epsilon.*

Kata kunci — Data Mining, DBSCAN, Scatterplot, Cluster, Silhouette Index

I. PENDAHULUAN

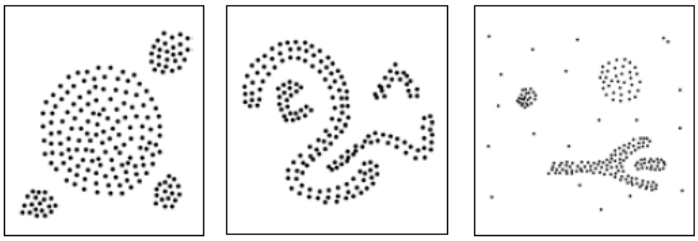
Perkembangan jenis-jenis data mengakibatkan proses clustering semakin bervariasi sesuai dengan jenis data yang akan diproses. Jenis data yang menjadi perhatian dalam penelitian saat ini adalah data-data yang memiliki unsur-unsur spasial. Ditinjau dari jenis data, sumber data dapat dibedakan menjadi sumber data yang mengandung data spasial, sumber data yang mengandung data temporal dan sumber data yang mengandung data spasial dan data temporal (Zhang, 2008). Basis data spasial berisi data-data yang berhubungan taksonomi model untuk ruang, jenis dan operator data spasial, bahasa kueri spasial dan strategi pemrosesan, serta indeks spasial dan teknik pengelompokan [1]. Data temporal berisi data-data kejadian yang terinegrasi berbasis waktu [2]. Data-data tersebut biasanya disimpan ke dalam basis data secara berurut dan periodik.

Dengan kemajuan teknologi informasi seperti saat ini, masyarakat disuguhkan dengan banyaknya informasi yang mudah diakses dengan media yang tersedia [3][4]. Banyak kemajuan dari pengolahan data yang ada untuk mengambil sebuah informasi secara cepat dan efektif. Karena banyaknya data yang ada maka akan sulit dalam mengambil suatu kesimpulan atau informasi jika tidak ada cara pengolahan data yang lebih baik. Maka dari itu, digunakanlah data mining untuk mengolah data yang begitu besar dan mengambil informasi yang terdapat didalamnya menggunakan teknik *spatial data clustering* [5].

Banyak organisasi atau instansi menggunakan data yang digunakan untuk memutuskan sebuah penyelesaian dari suatu masalah untuk mengambil sebuah kesimpulan dari data yang mereka miliki, tidak terkecuali dengan instansi pendidikan seperti Universitas Mercu Buana. Mereka menggunakan data sebagai dasar dalam pengambil keputusan, terutama dalam menentukan startegi pemasaran. Sebuah strategi pemasaran harus efektif dalam pelaksanannya. Maka dari itu, dalam penelitian ini akan membahas penggunaan data mining yaitu algoritma spasial clustering untuk menentukan strategi pemasaran yang lebih baik. Algoritma yang digunakan dalam penelitian ini adalah DBSCAN dengan data yang digunakan adalah data mahasiswa di Universitas Mercu Buana. Dari data tersebut kita akan melihat bagaimana sebaran peminatan mahasiswa berdasarkan asal sekolah serta alamat mereka tinggal dan bagaimana kita bisa mengambil informasi dari kumpulan data tersebut.

Density-based Spatial Clustering of Application with Noise atau lebih dikenal dengan sebutan DBSCAN termasuk ke dalam algoritmas *clustering* berbasis kepadatan (*density-based*). DBSCAN mencari kumpulan data dengan kepadatan yang tinggi untuk dijdikan sebagai *cluster* [6]. Bentuk *cluster* yang dihasilkan oleh DBSCAN bergantung kepada kepadatan tersebut, sehingga dengan algoritma ini dimungkinkan untuk menghasilkan bentuk *cluster* yang sembarang [7]. Suatu *cluster* dalam DBSCAN didefinisikan sebagai sekumpulan maksimum data yang terhubung di dalam kepadatan tersebut (*density-connected*).

Keanggotaan dari setiap profil dihitung berdasarkan rumus jarak. DBSCAN termasuk ke dalam *unsupervised clustering* karena jumlah *cluster* yang dihasilkan ditentukan oleh bentuk persebaran data itu sendiri, bukan diinisialisasi di awal seperti nampak pada gambar 1.



Gambar 1 Contoh *Clustering* dengan algoritma *DBSCAN*

ISSN: 2503-054X
Vol. 6 No: 2, Oktober 2021

Penelitian ini akan terfokus kepada penerapan algoritma spasial clustering pada data mahasiswa, adapun permasalahan-permasalahan spesifik yang hendak diselesaikan adalah bagaimana menggunakan algoritma spasial clustering untuk memetakan cluster daerah asal mahasiswa serta bagaimana menerapkan algoritma clustering spasial yang efektif untuk data yang akan digunakan dalam penelitian. Setiap algoritma yang digunakan dalam analisis *clustering* dapat memberikan hasil yang baik untuk suatu bentuk data, namun dapat memberikan hasil yang tidak baik apabila diimplementasikan terhadap data yang lain. Untuk mengetahui metode yang memberikan akurasi paling baik, diperlukan langkah-langkah tertentu. Langkah-langkah yang umum dilakukan dalam analisis *clustering* adalah sebagai berikut: (a) *pattern representation* (termasuk proses ekstraksi atau proses seleksi fitur), (b) pendefinisian cara menghitungan kedekatan atau jarak dalam pola yang sesuai dengan domain data, (c) *clustering*, (d) proses abstraksi data (opsional), dan (e) peninjauan hasil (opsional) [8].

Setiap algoritma *clustering* memiliki kelebihan dalam menganalisis *cluster* untuk jenis data tertentu, namun belum tentu lebih baik jika diaplikasikan terhadap data lain. Dengan demikian, perlu ada satu metode khusus yang dapat membandingkan secara objektif hasil-hasil analisis *clustering* tersebut. Dalam *clustering*, perbandingan tersebut dikenal sebagai *cluster evaluation* atau *cluster validation*. Evaluasi *clustering* dilakukan dengan memvalidasi hasil analisis *clustering*. Pelabelan keanggotaan terhadap setiap profil dievaluasi dengan menggunakan teknik validasi tertentu. Validasi diperlukan untuk menghitung nilai akurasi dari hasil *cluster* tersebut. Nilai akurasi tersebut diperlukan antara lain untuk membandingkan akurasi hasil *clustering* dengan menggunakan beberapa algoritma terhadap data yang sama.

Pada kasus-kasus tertentu, atribut yang diperlukan tidak secara eksplisit tersedia dalam data dan perlu dilakukan ekstraksi fitur atau atribut terlebih dahulu untuk mendapatkan atribut tersebut [9]. Ekstraksi atribut melakukan pengolahan terhadap data dan menghasilkan atribut baru bedasarkan masukkan dari atribut-atribut yang sudah ada.

Pada penelitian ini data yang digunakan untuk pengujian dalam penelitian ini adalah data mahasiswa Universitas Mercu Buana di Jakarta, yang berfokus kepada analisa domisili asal mahasiswa, sedangkan algoritma spasial clustering yang digunakan yaitu algoritma *Density-based Spatial Clustering of Application with Noise* (DBSCAN) dengan menggunakan aplikasi MATLAB.

II. METODOLOGI PENELITIAN

A. *Spasial Clustering*

Penelitian ini akan berfokus kepada analisis spasial *clustering* yaitu analisis pengelompokkan data spasial menggunakan algoritma *clustering* yang sudah ada. Berbeda dengan klasifikasi, dalam analisis spasial *clustering* tidak memiliki label atau tidak memasukkan label data tersebut ke dalam proses *clustering*. Dengan kata lain, analisis spasial *clustering* merupakan proses *unsupervised*. Untuk data spasial berukuran besar, biaya komputasi spasial *clustering* biasanya lebih mahal karena harus menemukan label-label masing-masing profil dalam data tersebut [10]. Untuk itu diperlukan algoritma komputasi yang efektif dan efisien untuk memproses data spasial yang berukuran besar dengan dimensi data yang tinggi tersebut.

Beberapa kebutuhan-kebutuhan umum yang mendorong penelitian dalam algoritma *clustering* antara lain (1) skalabilitas, (2) kemampuan untuk menangani berbagai jenis atribut, (3) kemampuan menemukan *cluster* dalam data yang acak, (4) hanya membutuhkan pengetahuan minimal terhadap domain pengetahuan asal data untuk menentukan parameter masukkan, (5) kemampuan untuk menangani *noisy data, (6) clustering* bersifat inkremental dan tidak sensitif terhadap urutan masukkan data, (7) kemampuan menangani data berdimensi tinggi, (8) *constaint-based clustering, (9)* hasil *clustering* dapat diinterpretasikan dan digunakan [11].

Dalam melakukan *clustering*, untuk bentuk data tertentu digunakan pendekatan algoritma *clustering* yang berbeda dengan bentuk data yang lain. Secara umum, pendekatan metode *clustering* dibagi menjadi dua pendekatan utama, yaitu pendekatan secara hirarki dan pendekatan secara partisi.

Pembentukan taksonomi berdasarkan beberapa kriteria yang membedakan antara satu algoritma dengan algoritma yang lain.

a. Pendekatan *agglomerative* dibandingkan dengan pendekatan *divisive*
   Dalam pendekatan *agglomerative*, mula-mula setiap profil dianggap sebagai satu *cluster* tersendiri. Dengan menggunakan perhitungan kemiripan, profil-profil yang memiliki kemiripan kemudian digabung sampai memenuhi kriteria tertentu. Sebaliknya, dalam pendekatan *divisive*, mula-mula seluruh profil dianggap sebagai satu *cluster* yang sama. Dengan menggunakan perhitungan kemiripan, *cluster* tersebut kemudian dipecah menjadi beberapa *cluster* sampai memenuhi kriteria tertentu.
b. Pendekatan *monothetic* dibandingkan dengan pendekatan *polythetic*
   Dalam pendekatan *monothetic*, perhitungan kemiripan profil menggunakan atribut yang dimasukkan secara bertahap. Dalam hal ini, setiap kali tambahan atribut dimasukkan, *cluster* kemudian dipecah sesuai masukkan atribut baru tersebut. Berbeda dengan pendekatan *monothetic*, dalam pendekatan *polythetic*, semua atribut dimasukkan dalam perhitungan kemiripan secara bersama-sama. Metode *clustering* yang ada pada umumnya menggunakan pendekatan *polythetic*.
c. Pendekatan *hard clustering* dibandingkan dengan pendekatan *fuzzy clustering*
   Dalam pendekatan *hard clustering*, satu profil menjadi anggota satu *cluster* saja. Sementara itu, dalam pendekatan *fuzzy clustering*, satu profil memiliki derajat keanggotaan di semua *cluster*. Pendekatan *fuzzy clustering* dapat diubah menjadi *hard clustering* dengan memilih derajat keanggotaan paling tinggi sebagai *cluster* data tersebut.
d. Pendekatan *deterministic* dibandingkan dengan pendekatan *sthocastic*
   Dalam pendekatan secara *deterministic*, optimisasi fungsi *squared error* dilakukan dengan teknik tradisional, sedangkan dalam pendekatan secara *stochastic*, optimisasi fungsi *squared error* dilakukan dengan *random search* terhadap *state space* yang terdiri dari keseluruhan label yang mungkin.
e. Pendekatan non-inkremental dibandingkan dengan pendekatan inkremental

ISSN: 2503-054X
Vol. 6 No: 2, Oktober 2021

B. *Algoritma DBSCAN*

Komponen-komponen dalam proses analisis *clustering* yang ada pada algoritmta DBSCAN antara lain:

a. Epsilon

Epsilon kekerabatan dari sebuah profil atau *Eps-neigborhood* dari sebuah profil, $N_{eps}(P)$, didefinisikan sebagai

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

D = basis data yang dianalisis
q = profil lain.
Eps = nilai ambang jarak antarprofil untuk dapat dimasukkan ke dalam *cluster* yang sama. profil p dapat berkerabat dengan profil q (berada dalam satu *cluster* yang sama) jika jarak dari p ke q tidak lebih dari nilai *Eps*.

b. *Minimum Points*

Meskipun p berada dalam *Eps-neigborhood* dari q, akan tetapi jika hanya dua profil itu saja yang berkerabat, maka akan ada kasus dimana terdapat banyak *cluster* dengan jumlah anggota yang sedikit. Untuk mengantisipasi hal tersebut, diperkenalkan istilah *minimum points* atau *MinPts*. *MinPts* merupakan nilai ambang yang merepresentasikan jumlah minimal profil yang berada dalam *Eps-neigborhood* profil p agar dapat terbentuk *cluster*. Dengan nilai ambang ini, maka ada tiga klasifikasi jenis profil di DBSCAN, yaitu profil yang berada di berada di luar daerah padat disebut *outlier*, *profil* yang berada di pangkal daerah padat disebut *border point*, dan *profil* yang berada di dalam daerah padat disebut *core point*.

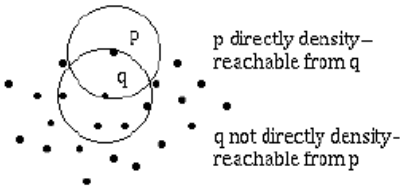
c. *Directly density-reachable*

Sebuah profil p dikatakan *directly density-reachable* terhadap profil q jika

$p \in N_{eps}(q)$, dan

$\left|N_{eps}(q)\right| \geq MinPts$ (q merupakan *core point*)

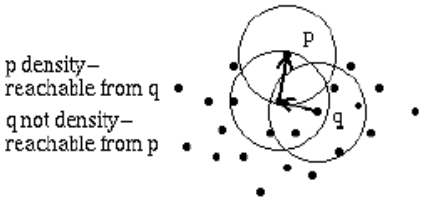
Dari definisi tersebut dapat diketahui bahwa agar profil *p directly density-reachable* terhadap profil q, maka harus memenuhi dua syarat yaitu profil p berada pada *Eps-neighborhood* profil q dan profil q merupakan *core poin*. *Directly density-reachable* bersifat simetris jika p dan q keduanya adalah *core point*. Artinya, jika *p directly density-reachable* terhadap *q,* maka *q directly density-reachable* terhadap *p*.



Gambar 1 Contoh sepasang profil yang *directly density-reachable*

d. *Density-reachable*

Sebuah profil p dikatakan *density-reachable* terhadap profil q jika terdapat rantai $p_1 \ldots . p_n$, dengan $p_1 = p$ dan $p_n = q$, sedemikian sehingga p_{i+1} bersifat *directly density-reachable* terhadap p_i. Dari definisi tersebut dapat diketahui bahwa dua buah profil dikatakan *density-reachable* jika ada satu rantai profil sedemikian sehingga dari profil satu ke profil lain di dalam rantai tersebut bersifat *directly density-reachable*. Sifat *density-reachable* tidak menjamin dua *border point* bersifat *density-reachable*.

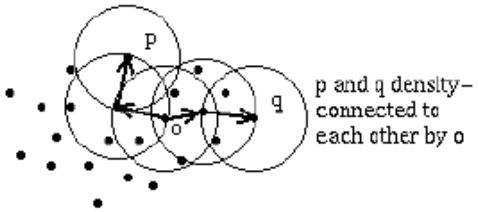


Gambar 2 Contoh sepasang profil yang *density-reachable*

e. *Density-connected*

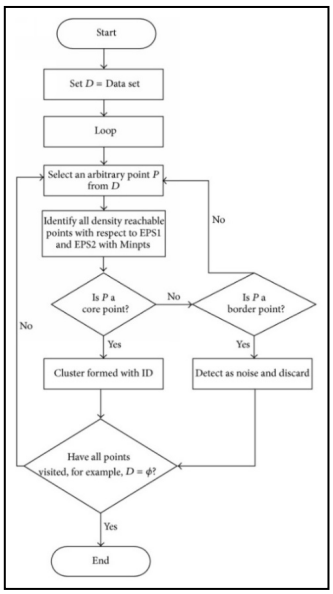
Sebuah profil p dikatakan *density-connected* terhadap profil q jika terdapat profil o sedemikian sehingga profil p dan profil q bersifat *density-reachable* terhadap poin o. Dengan demikian, setidaknya dua profil di dalam satu *cluster* bersifat *density-connected*. *Density-connected* bersifat simetris dan refleksif. Artinya, jika profil p bersifat *density-connected* terhadap poin q, maka profil q bersifat *density-connected* terhadap profil p.

ISSN: 2503-054X
Vol. 6 No: 2, Oktober 2021



Gambar 3 Contoh sepasang profil yang *density-connected*

Nilai *eps* dan *MinPts* harus diketahui untuk dapat menjalankan algoritma DBSCAN. Algoritma DBSCAN dimulai dengan memilih satu profil *p* secara acak, kemudian mencari profil-profil lain yang *density-reachable* terhadap profil *p*. Jika *p* merupakan *core point*, maka terbentuk suatu *cluster*. Akan tetapi jika *p* adalah *border poin*t, maka DBSCAN akan mengambil profil lain dari basis data. Dalam proses tersebut, terdapat kemungkinan dua *cluster* bergabung jika kedua *custer* tersebut dekat dengan flowchart seperti pada gambar 5.



Gambar 4 Flowchart Algoritma DBSCAN

Berdasarkan gambar 5, dalam algoritma DBSCAN dilakukan input data yang akan digunakan, kemudian dilakukan proses *loop.* Pada proses *loop* ini dilakukan pengecekan tiap nilai yang *density reachable.* Jika data yang dicek ini merupakan titik inti maka akan dibentuk *cluster.* Jika tidak,maka akan dicek apakah titik itu merupakan titik tepi, jika iya maka titik itu akan melakukan proses pengecekan kembali. Jika tidak, maka akan dimasukkan kedalam kategori *noise.* Proses ini berlanjut hingga semua titik diperiksa.

C. *Metode Silhouette Index*

Dalam penelitian ini, metode validasi hasil analisi spasial *clustering* yang digunakan adalah metode *Silhouette index*. Metode validasi dengan mengukur kesamaan atau ketidaksamaan antara profil yang ada dalam satu *cluster* dengan *cluster* lain hasil analisis *clustering*. Pengukuran kesamaan atau ketidaksamaan tesebut dihitung dengan rumus jarak. Jika diketahui data yang dianalisis $X = \{x_1,\ x_2, \ldots., x_n\}, n =$ jumlah profil dalam data, dan diketahui *A* adalah salah satu *cluster* yang dihasilkan, maka dapat dicari:

$a(i)$ = jarak rata-rata profil ke$-i$ terhadap profil lain yang ada dalam *cluster A*

Jika diketahui terdapat *cluster C* dimana $C \neq A$, maka dapat dicari

$d(i, C) =$ jarak rata-rata profil ke$-i$ terhadap profil yang ada dalam *cluster C*.

Jika terdapat lebih dari satu *cluster* lain selain *cluster A*, maka perlu dicari *cluster* yang paling dekat dengan *cluster A* dengan menghitung

$$b(i) = \min_{C \neq A} d(i, C)$$

Silhouette index terhadap profil ke$-i$ yaitu $s(i)$ didefinisikan sebagai berikut

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

Rentang nilai $s(i)$ yang diperoleh dari perhitungan tersebut adalah $-1 \leq s(i) \leq 1$. Nilai $s(i)$ lebih dekat ke 1 menunjukkan bahwa data tersebut sudah *well clustered*. Artinya, kemiripan data tersebut dengan data lain yang ada dalam *cluster* yang sama jauh lebih besar dibandingkan dengan kemiripan data tersebut dengan data dari *cluster* lain yang berdekatan. Nilai $s(i)$ lebih dekat ke 0 menunjukkan bahwa data tersebut termasuk dalam *intermediate case*. Artinya, kemiripan data tersebut dengan data lain yang ada dalam *cluster* yang sama relatif sama besar dibandingkan dengan kemiripan data tersebut dengan data dari *cluster* lain yang berdekatan.

ISSN: 2503-054X
Vol. 6 No: 2, Oktober 2021

Data yang berada pada *intermediate case* dapat dipindah ke *cluster* yang berdekatan. Nilai $s(i)$ lebih dekat ke -1 menunjukkan bahwa data tersebut *misclassified*. Artinya, kemiripan data tersebut dengan data lain yang ada di *cluster* yang sama dibandingkan dengan data di *cluster* yang berdekatan jauh lebih mirip terhadap data dari *cluster* yang berdekatan. Setelah menghitung masing-masing nilai *Silhouette index* dalam data, maka dapat diketahui S yaitu rata-rata nilai *Silhouette index* dari data:

$$S = \frac{1}{mc}\sum_{i=1}^{c}\sum_{j=1}^{m} s(ij)$$

$i = 1, \dots, c$ menyatakan *cluster* ke-i, $j = 1, \dots, m$ menyatakan data ke$-j$ dalam *cluster* ke$-i$, dan $s(ij)$ menyatakan nilai *Silhouette index* data ke$-j$ dalam *cluster* ke$-i$. Nilai dari S menyatakan akurasi hasil analisi *clustering.*

III. Pembahasan dan Hasil

Implementasi Algoritma DBSCAN

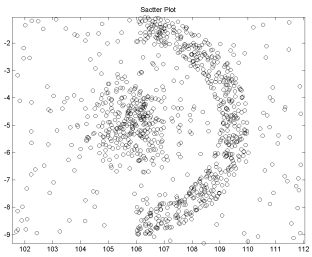
Proses implementasi tahap pengumpulan data dilakukan dengan memasukan data kordinat alamat yang terdapat dalam data mahasiswa Universitas Mercu Buana. Pertama, data alamat dimasukkan kedalam aplikasi google maps. Maka, akan didapatkan data koordinat (longitude dan latitude). Setelah diperoleh data yang diperlukan, langkah selanjutnya dalam penelitian ini adalah proses spasial *clustering*. Algoritma yang digunakan dalam implementasi analisis spasial *clustering* adalah algoritma *Density-base Spatial Clustering of Application with Noise* (DBSCAN).

Peneliti menggunakan fungsi *Scatter Plot* untuk melihat bentukan awal data yang akan dibentuk *clusternya.* Data yang telah ada dijadikan sebagai inputan dalam fungsi *Scatter Plot* agar peneliti dapat melihat secara jelas bagaimana bentuk penyebaran data yang telah disiapkan. Selanjutnya penggunaan algoritma spasial *clustering* adalah tahap validasi hasil spasial *clustering*. Validasi dilakukan terhadap setiap hasil analisis *clustering*. Metode validasi *clustering* yang digunakan dalam penelitian ini adalah metode *Silhouette index*. Implementasi metode *Silhouette index* menggunakan *toolbox* yang sudah ada di Matlab yaitu fungsi *silhouette*. Bagian ini hanya akan ditunjukkan bagian penting dari implementasi proses validasi hasil *clustering* dengan *Silhouette index*.

Pengujian Algoritma DBSCAN

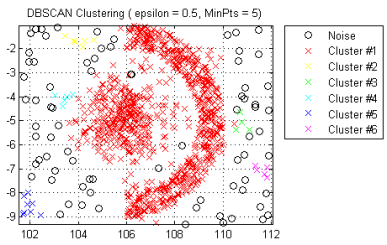
Proses pengujian dilakukan dengan menggunakan aplikasi MATLAB. Pertama penguji melakukan hasil Scatter Plot pada data *Clustering* untuk melihat bentukan awal yang telah ada. Selanjutnya dilakukan pengujian algoritma DBSCAN pada data yang telah ada. Pengujian dilakukan dengan proses berulang dengan menggunakan nilai parameter yang berbeda (epsilon & minPts). Berikut ini hasil pengujian yang telah dilakukan oleh peneliti.

Pertama, peneliti melakukan implementasi kode *Scatterplot* untuk melihat penyebaran data yang telah disiapkan. Berikut hasil uji implementasi fungsi *Scatterplot* seperti Nampak pada gambar 6.

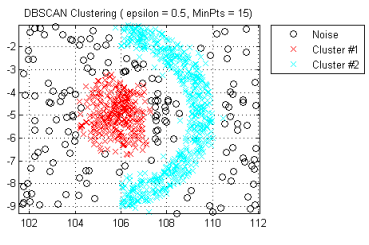


Gambar 6 Hasil *Scatterplot*

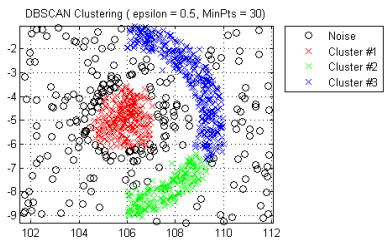
Pada gambar 6, garis absis atau x merupakan nilai dari garis bujur. Dan garis ordinat atau y merupakan nilai dari garis lintang. Rentang nilai dari garis yang digunakan berasal dari nilai minimum dan maksimum dari data yang telah disiapkan sebelumnya. Berdasarkan gambar diatas, secara kasat mata peneliti bisa menentukan bahwa ada 2 *cluster* yang terbentuk. Selanjutnya, peneliti melakukan pengujian DBSCAN pada data yang telah disiapkan peneliti. Pengujian idilakukan dalam beberapa kali percobaan, dimana tiap percobaan peneliti menggunakan nilai parameter yang berbeda (epsilon & minPts) seperti pada gambar 7



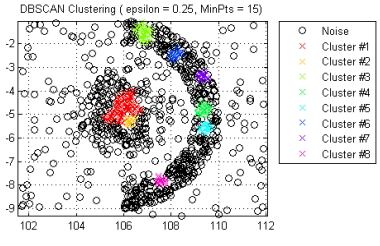
a

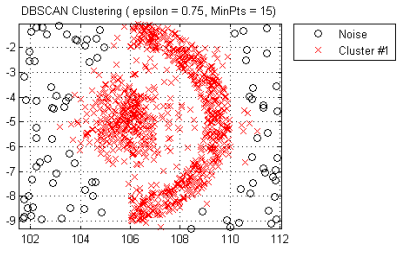


b



c





ISSN: 2503-054X
Vol. 6 No: 2, Oktober 2021

d e

Gambar 7 Hasil Uji Implentasi Algoritma DBSCAN

Berdasarkan gambar 7(a), terlihat hasil uji penggunaan algoritma DBSCAN dengan nilai parameter epsilon = 0.5, dan MinPts = 5 menghasilkan enam buah *cluster* disertai *noise*. Dimana *cluster* 1 mendominasi dibandingkan dengan *cluster* lainnya. Sedangkan pada gambar 7(b), terlihat hasil uji penggunaan algoritma DBSCAN dengan nilai parameter epsilon = 0.5, dan MinPts = 15, menghasilkan dua buah *cluster* disertai *noise*. Dimana hasil pengujian ini lebih mendekati dengan yang peneliti amati pada *Scatterplot* yaitu menghasilkan 2 *cluster*. Pada gambar 7(c), terlihat hasil uji penggunaan algoritma DBSCAN dengan nilai parameter epsilon = 0.5, dan MinPts = 30, menghasilkan tiga buah *cluster* disertai *noise*. Pada gambar 7(d) terlihat hasil uji penggunaan algoritma DBSCAN dengan nilai parameter epsilon = 0.25, dan MinPts = 15, menghasilkan delapan buah *cluster* disertai *noise*. Dimana tiap *cluster* terbentuk dalam kelompok yang lebih kecil. Pada gambar 7(e) terlihat hasil uji penggunaan algoritma DBSCAN dengan nilai parameter epsilon = 0.75, dan MinPts = 15, menghasilkan hanya satu buah *cluster* disertai *noise*.

Berdasarkan pengujian yang dihasilkan pada gambar 7, dapat dilihat bahwa dengan berubahnya nilai parameter yang telah ditentukan, dapat mengubah bentuk dan banyaknya *cluster* yang terbentuk. Hasil pengujian ini dapat dilihat pada tabel 1.

Table 1 **Tabel Jumlah *Cluster* Yang Terbentuk**

Gambar 8.	Epsilon	MinPts	*Cluster* Yang Terbentuk
a	0.5	5	6
b	0.5	15	2
c	0.5	30	3
d	0.25	15	8
e	0.75	15	1

Pada tabel 1 dapat kita lihat jika nilai MinPts yang semakin naik, jumlah *cluster* yang terbentuk mengalami penurunan. Hal ini karena ketika nilai MinPts kecil, maka akan terbentuk banyak *cluster.* Sedangkan ketika nilai MinPts mengalami kenaikan, maka *cluster* yang terbentuk akan semakin berkurang, karena nilai MinPts yang dijadikan sebagai parameter *cluster* semakin besar. Sebaliknya jika semakin besar nilai epsilon, jumlah *cluster* yang terbentuk mengalami penurunan. Karena epsilon merupakan parameter jarak antar satu data dengan data yang lain. Jika, epsilon kecil maka jarak satu titik ke titik yang lain semakin pendek. Sehingga akan terbentuk banyak *cluster.* Tetapi, jika epsilon semakin besar, maka cakupan data untuk mengambil jarak antar data lainnya semakin besar. Sehingga, akan terbentuk *cluster* yang lebih sedikit dengan bentukan yang besar.

Tahap selanjutnya, peneliti melakukan uji waktu yang digunakan untuk mengukur waktu program berjalan pada algoritma DBSCAN dengan nilai parameter yang berbeda. Pengujian ini menggunakan *tools* tic dan toc. Hasil pengujian ini dapat dilihat pada tabel 2.

Table 2 **Tabel Hasil Uji Waktu Jalannya Program**

Gambar 8	Epsilon	MinPts	Waktu (detik)
a	0.5	5	0.194570
b	0.5	15	0.175188
c	0.5	30	0.129221
d	0.25	15	0.055377
e	0.75	15	0.198502

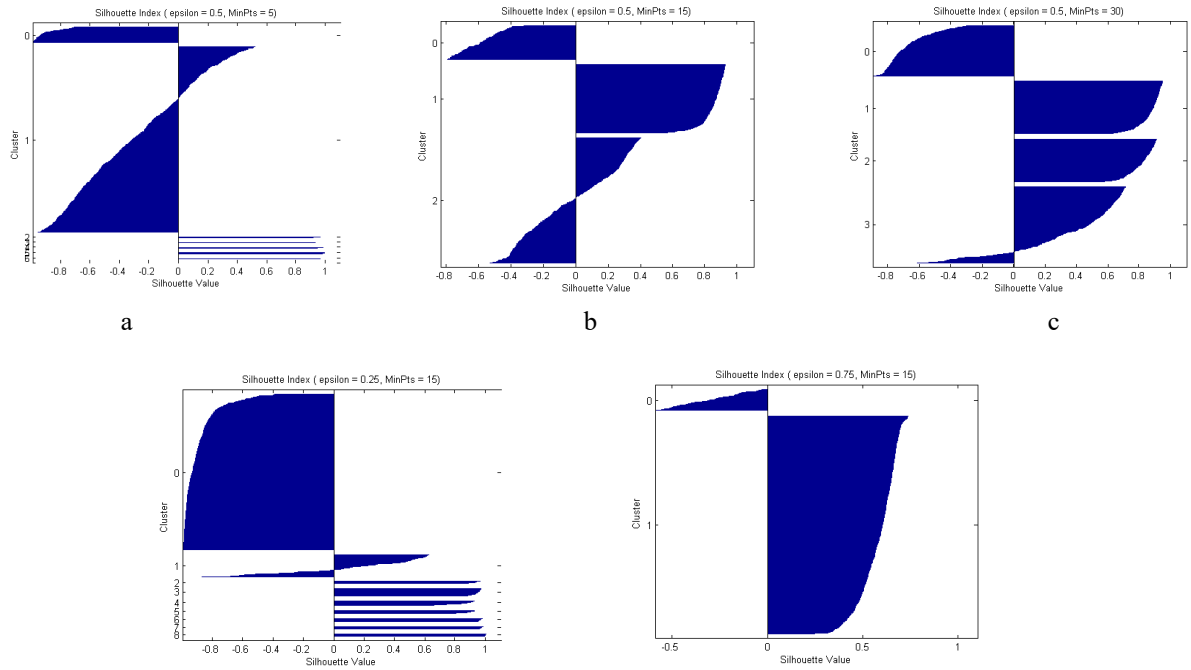
Pada tabel 2 merupakan hasil uji waktu jalannya program algoritma DBSCAN pada MATLAB dengan nilai MinPts yang berbeda. Dapat dilihat dengan nilai MinPts yang semakin naik, maka waktu yang dibutuhkan untuk jalannya program akan lebih sedikit. Hal ini dikarenakan, karena pemprosesan data oleh computer yang terbilang lebih cepat ketika menggunakan nilai MinPts yang lebih besar. Jika nilai MinPts kecil, maka program akan memproses secara perlahan dengan menggunakan lima data. Berbeda dengan nilai MinPts yang lebih besar, maka program akan langsung melakukan proses dengan menggunakan puluhan data. Sebaliknya dengan nilai epsilon yang semakin naik, maka waktu yang dibutuhkan untuk jalannya program akan lebih banyak. Hal ini dikarenakan, jika nilai epsilon kecil, maka proses akan lebih cepat karena jarak yang digunakan sebagai parameter lebih pendek. Dibandingkan dengan nilai epsilon yang lebih besar, dengan cakupan jangkauan yang lebih bear, maka dibutuhkan waktu yang lebih banyak.

Metode Silhouette Index

Tahap ketiga dalam pengujian ini adalah melakukan validasi hasil spasial *clustering*. Validasi dilakukan terhadap setiap hasil analisis *clustering*. Dengan menggunakan *Silhouette index*, maka akan tampak nilai kerapatan tiap *cluster.* Nilai pada *Silhouette*

ISSN: 2503-054X
Vol. 6 No: 2, Oktober 2021

index memiliki rentang nilai dari minus satu (-1) hingga satu (1). Dimana semakin mendekati angka satu maka *cluster* merupakan cluster yang baik. Berikut hasil uji implementasi fungsi *Silhouette index* pada gambar 8.



Gambar 8 Hasil Uji Validasi *Silhouette index* Algoritma DBSCAN

Berdasarkan gambar 8(a), terlihat hasil validasi *Silhouette index* algoritma DBSCAN dengan nilai parameter epsilon = 0.5, dan MinPts = 5, di mana *cluster* satu yang dominan memiliki grafik yang lebih besar dengan nilai kerapatan yang rendah. Sedangkan, *cluster* lainnya memiliki kerapatan yang tinggi dengan grafik yang kecil dikarenakan *cluster* tersebut memiliki anggota yang sedikit dibandingkan *cluster* satu. Pada gambar 8(b), terlihat hasil validasi *Silhouette index* algoritma DBSCAN dengan nilai parameter epsilon = 0.5, dan MinPts = 15, di mana *cluster* satu dan *cluster* dua memiliki anggota yang cukup banyak sehingga grafiknya terllihat besar. Dengan nilai *cluster* satu memiliki nilai kerapatan yang baik *cluster* dua memiliki nilai kerapatan yang lebih sedikit dari *cluster* dua. Pada gambar 8(c), terlihat hasil validasi *Silhouette index* algoritma DBSCAN dengan nilai parameter epsilon = 0.5, dan MinPts = 30, di mana terdapat tiga *cluster* yang memiliki besar grafik yang seimbang dengan nilai kepadatan *cluster* yang cukup baik. Pada gambar 8(d), terlihat hasil validasi *Silhouette index* algoritma DBSCAN dengan nilai parameter epsilon = 0.25, dan MinPts = 15, di mana terdapat 8 *cluster* yang hampir keseluruhan terbilang *cluster* kecil karena bentukan grafik yang kecil, tetapi memiliki nilai kerapatan yang baik. Sebagian besar data masuk dalam kategori *noise.* Pada gambar 8(e), terlihat hasil validasi *Silhouette index* algoritma DBSCAN dengan nilai parameter epsilon = 0.75, dan MinPts = 15 hanya memiliki satu *cluster*, dengan kepadatan yang baik.

Selanjutnya, peneliti melakukan perhitungan rata-rata nilai dari tiap *Silhouette index* dengan mengambil total nilai dari seluruh nilai *Silhouette index* dan dibagi jumlah data nilai yang digunakan. Hasil perhitungan ini dapat dilihat pada tabel 3.

Table 3 **Tabel Nilai Rata-Rata *Silhouette Index***

Gambar 9	**Epsilon**	**MinPts**	**Nilai *Silhouette Index***
a	0.5	5	0.2818
b	0.5	15	0.5629
c	0.5	30	0.6883
d	0.25	15	0.7439
e	0.75	15	0.5806

Pada table 3 dapat dilihat dengan nilai MinPts yang semakin naik, maka nilai rata-rata *Silhouette Index* mengalami kenaikan pula. Maka *cluster* yang terbentuk semakin baik. Karena, nilai MinPts yang akan terbentuk banyak *cluster.* Dan *cluster* terbesar didalamnya memiliki kerapatan yang kurang baik. Sedangkan dengan nilai MinPts yang besar, maka kerapatan dalam satu *cluster* yang terbentuk lebih baik dan lebih *compact*. Sebaliknya dapat dilihat dengan nilai epsilon yang semakin naik, nilai rata-rata *Silhouette Index* mengalami kenaikan dan penurunan sehingga tidak menentu perubahan nilainya. Hal ini dikarenakan dengan nilai epsilon yang kecil maka jarak antar data akan lebih dekat dengan begitu hasil *cluster* yang dibentuk akan lebih rapat. Sedangkan dengan angka epsilon yang lebih besar maka akan ada penurunan nilai yang signifikan, dikarenakan jarak antar data yang lebih besar. Dari hal itu, maka *cluster* yang terbentuk akan lebih renggang.

ISSN: 2503-054X
Vol. 6 No: 2, Oktober 2021

IV. KESIMPULAN DAN SARAN

Kesimpulan

Penelitian ini telah berhasil mengimplementasikan metode analisis mengenai aplikasi algoritma spasial clustering pada data mahasiswa Universitas Mercu Buana. Algoritma yang dilakukan dalam penelitian adalah *Density-based Spatial Clustering of Application with Noise*. Pengujian untuk mengevaluasi implementasi tersebut dilakukan dengan eksperimen terhadap data mahasiswa Universitas Mercu Buana. Dari hasil pengujian, diperoleh kesimpulan sebagai berikut:

1. Pengujian penggunaan algoritma DBSCAN pada data mahasiswa Universitas Mercu Buana berhasil dilakukan dalam menentukan *cluster* dan *noise* berdasarkan alamat pada data mahasiswa Universitas Mercu Buana.
2. Pengunaan *Scatterplot* diperlukan dalam menentukan gambaran awal *cluster* yang terbentuk.
3. Hasil *cluster* yang terbentuk sangat dipengaruhi oleh nilai *epsilon* dan *Minimum Points*. Dengan nilai *epsilon* yang semakin naik maka *cluster* yang terbentuk mengalami penurunan. Dengan nilai *Minimum Points* yang semakin naik maka *cluster* yang terbentuk mengalami perubahan yang kurang menentu.
4. Perhitungan waktu berjalannya program juga dipengaruhi oleh nilai *epsilon* dan *Minimum Points*. Dengan nilai *epsilon* yang semakin naik maka waktu yang dibutuhkan lebih banyak. Sebaliknya jika *Minimum Points* yang semakin naik, maka waktu yang dibutuhkan lebih sedikit.
5. Perhitungan nilai *Silhouette Index* dari tiap pengujian juga menghasilkan nilai yang berbeda. Dengan nilai *epsilon* yang semakin naik, maka nilai *Silhouette Index* mengalami perubahan yang tidak menentukan. Jika *Minimum Points* yang semakin naik, maka nilai *Silhouette Index* mengalami kenaikan.

Saran

Penelitian mengenai analisa algoritma spasial *clustering* pada data mahasiswa Universitas Mercu Buana masih memiliki kekurangan. Kekurangan-kekurangan tersebut antara lain data yang digunakan untuk eksperimen hanya satu jenis data sehingga hasilnya tidak dapat dibandingkan dengan data lain. Untuk penelitian berikutnya, penelitian dapat dilakukan dengan menggunakan berbagai jenis data sehingga performa algoritma dapat dibandingkan tidak hanya untuk satu data tetapi juga perbandingan antar data. selain itu, dapat diimplementasikan algoritma-algoritma *clustering* yang lain untuk menghasilkan algoritma yang lebih baik penanganannya terhadap suatu data

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Mercu Buana yang telah memberi dukungan *financial* terhadap penelitian ini.

DAFTAR PUSTAKA

[1] S. Shekhar, S. Chawla, S. Ravada, A. Fetterer, X. Liu, and C. T. Lu, "Spatial databases accomplishments and research needs," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 1, pp. 45–55, 1999, doi: 10.1109/69.755614.

[2] Y. T. Fan, J. Y. Yang, D. H. Zhu, and K. L. Wei, "A time-based integration method of spatio-temporal data at spatial database level," *Math. Comput. Model.*, vol. 51, no. 11–12, pp. 1286–1292, 2010, doi: 10.1016/j.mcm.2009.10.032.

[3] S. Tentang, W. E. B. E. Di, and K. Kota, "IMPLEMENTASI TEKNOLOGI INFORMASI DAN KOMUNIKASI (STUDI TENTANG WEB E-GOVERNMENT DI KOMINFO KOTA MANADO) oleh," *e-journal "Acta Diurna,"* vol. VI, no. 3, 2017.

[4] H. S. Wahyudi and M. P. Sukmasari, "TEKNOLOGI DAN KEHIDUPAN MASYARAKAT," *Sosiologi, J. Anal.*, vol. 3, no. 1, 2014.

[5] A. S. Devi, I. K. G. D. Putra, and I. M. Sukarsa, "Implementasi Metode Clustering DBSCAN pada Proses Pengambilan Keputusan," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 6, no. 3, p. 185, 2015, doi: 10.24843/lkjiti.2015.v06.i03.p05.

[6] G. H. Shah, "An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets," *3rd Nirma Univ. Int. Conf. Eng. NUiCONE 2012*, pp. 6–8, 2012, doi: 10.1109/NUICONE.2012.6493211.

[7] T. Wang, C. Ren, Y. Luo, and J. Tian, "NS-DBSCAN: A density-based clustering algorithm in network space," *ISPRS Int. J. Geo-Information*, vol. 8, no. 5, 2019, doi: 10.3390/ijgi8050218.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review. ACM Comput Surv," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[9] D. Dwi Purwanto, "Decision Support System untuk Penentuan Pemberian Beasiswa Prestasi di Perguruan Tinggi," vol. 2, no. 01, p. 9, 2016.

[10] C. Clustering, "Cluster analysis," *Data Handl. Sci. Technol.*, vol. 20, no. PART 2, pp. 57–86, 1998, doi: 10.1016/S0922-3487(98)80040-3.

[11] J. Han, J. Pei, and M. Kamber, *Data Mining, Southeast Asia Edition 2nd Edition*. 2006.