

Segmentasi Kasus Data Kematian Covid 19 Di Jawa Barat Menggunakan Algoritma DBSCAN

Nissa Aulia Permatasari¹, Yulison Herry Chrisnanto², Ade Kania Ningsih³

Universitas Jendral Achmad Yani

Jln. Terusan Jendral Sudirman, Cibeber, Cimahi Selatan, Cimahi Jawa Barat 40531

Nissaauliakusnya3516@gmail.com¹, y.chrisnanto@gmail.com², Ade.Kanianingsih@lecture.unjani.ac.id³

Abstrak

Beta-coronavirus (SARS-CoV-2) atau virus covid 19 adalah virus yang menyerang saluran pernafasan. Sejauh ini, jumlah kasus positif dan kasus kematian akibat virus ini terus meningkat. Menurut data WHO, jumlah total yang terinfeksi virus ini adalah 6.674.000 dari tanggal 3 januari 2020 sampe dengan tanggal 2 desember 2022 dengan kasus kematian mencapai 159 kasus dari indonesia . Secara global sampai tanggal 2 desember 2022 terdapat 640.395.651 kasus covid 19 yang terkonfirmasi , termasuk dengan kasus kematian nya hingga 6.618.579. peneliti memilih menggunakan algoritma dbscan untuk mengelompokkan hasil data covid 19 yang dihasilkan. Pandemi covid sendiri yang melanda china dan menyebar ke seluruh negara , indonesia menjadi salah satu negara yang terkena covid 19 , masuknya covid 19 ke indonesia menyebabkan beberapa provinsi terjangkit covid 19 salah satu nya merupakan provinsi jawa barat. Algoritma dbscan adalah metode pengelompokan yang menciptakan wilayah terkait kepadatan berdasarkan kepadatan yang terkoneksi. Tujuan dari algoritma dbscan adalah untuk mengidentifikasi area yang mampu diukur dengan jumlah objek yang mendekati. Algoritma dbscan merupakan metode yang dapat digunakan untuk segmentasi pengelompokan data covid 19. Pada penelitian ini dilakukan segmentasi dataset covid 19 di jawa barat menggunakan algoritma dbscan . Dari hasil segmentasi data dapat diperoleh 30.000 data yang menjadi sample data sehingga menghasilkan beberapa cluster yang dapat dibandingkan, dengan membandingkan nilai percobaan 1 dan ke 2, percobaan pertama hasil akurasi sembuh hingga 90% dan pada percobaan kedua 75% begitu juga dengan meninggal , aktif dan total. Maka itu dapat disimpulkan bahwa percobaan pertama mendapatkan angka akurasi sembuh , meninggal , aktif dan total lebih besar dari pada percobaan kedua. Dengan nilai akurasi sebesar 0.0099502487562189.

Kata kunci: Clustering, DBSCAN, Corona, Jawa Barat

PENDAHULUAN

Penyakit Coronavirus 2019 juga dikenal sebagai covid 19 merupakan epidemi pertama yang terdeteksi bulan desember 2019 di kota Wuhan, china. Sebelum WHO atau Organisasi Kesehatan Dunia menamai virus ini sebagai COVID-19, WHO menyebut varian virus ini dengan Coronavirus 2019. (2019). -ncov).(Sohrabi et al., 2020). Dari tanggal 3 januari 2020 sampai dengan 2 desember 2022 , Menurut WHO, total kasus virus ini sebanyak 640.395.651 dan jumlah kematian sebanyak 6.618.579 kasus dari seluruh dunia. Di Indonesia sendiri, total kasus terinfeksi antara 3 Januari 2020 hingga 2 Desember 2022 sebanyak 6.674.000 dengan total 159 kematian (Rasmussen et al., 2020). Penyebaran penyakit yang ditimbulkan virus covid 19. ini mengakibatkan banyak pasien yang terinfeksi meninggal dunia. Menurut beberapa data yang diperoleh, maka dilangsungkan pengelompokan data virus covid 19 kemudian diperkenalkan ke provinsi Indonesia untuk meningkatkan kualitas perawatan penanganan bagi pasien yang terinfeksi virus COVID-19 terutama provinsi jawa barat.(Noviyanto, 2020).

Segmentasi sebelumnya dilakukan terhadap diagnosis COVID-19 pada citra X-Ray paru namun proses segmentasi dapat mempengaruhi hasil klasifikasi dan identifikasi pasien COVID-19. Hasil dari proses Segmentasi tidak secara signifikan meningkatkan performa model, khususnya akurasi klasifikasi. Namun, segmentasi meningkatkan keandalan dan kualitas model yang dikembangkan.(Raras et al., 2023) Segmentasi adalah proses pembagian atau pengelompokan suatu hal menjadi beberapa bagian yang lebih kecil dan homogen berdasarkan karakteristik atau kriteria tertentu.

Dalam algoritma DBSCAN (Density-Based Spatial Clustering of Applications with Noise), segmentasi digunakan sebagai metode untuk mengelompokkan data spasial menjadi cluster berdasarkan kepadatan data. DBSCAN bekerja dengan mengidentifikasi titik-titik yang berdekatan dalam ruang data dan membentuk cluster berdasarkan kepadatan titik-titik tersebut. Titik-titik yang memiliki kepadatan yang cukup tinggi akan menjadi bagian dari cluster, sedangkan titik-titik yang memiliki kepadatan rendah atau terisolasi akan dianggap sebagai noise atau outlier. Dalam konteks DBSCAN, segmentasi digunakan untuk memisahkan data menjadi kelompok-kelompok yang homogen berdasarkan kepadatan spasialnya. Algoritma ini dapat membantu

mengidentifikasi pola atau struktur dalam data yang mungkin sulit ditemukan dengan metode clustering lainnya. Dengan menggunakan DBSCAN, kita dapat melakukan segmentasi data spasial seperti segmentasi citra berwarna, segmentasi objek pada citra, atau segmentasi data geografis. Algoritma ini dapat membantu dalam berbagai aplikasi seperti pengenalan pola, pengolahan citra, dan analisis data spasial. (Raras et al., 2023)

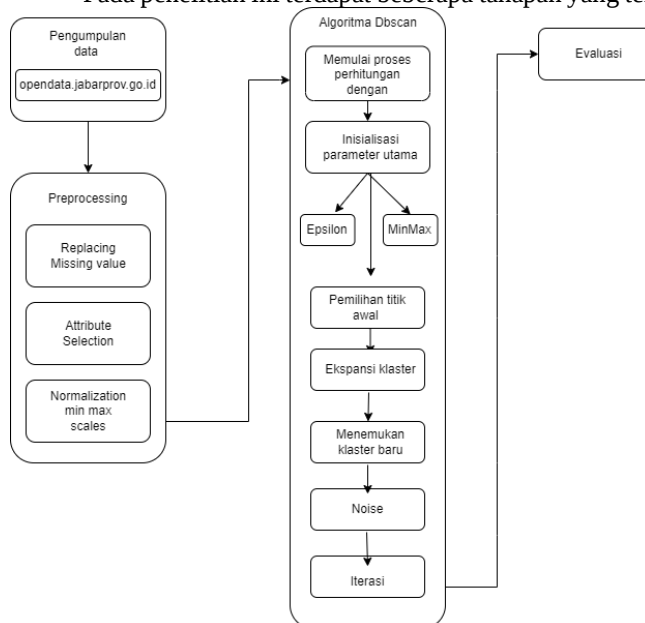
Penelitian ini menggunakan metode Algoritma DBSCAN, Menurut (David, 2020), algoritma DBSCAN mampu mengelompokkan data yang berdekatan secara relatif, dan sering diterapkan pada data yang banyak dan mengandung noise, hal ini disebabkan karena DBSCAN tidak akan memasukan data yang dianggap noise kedalam cluster manapun. DBSCAN memerlukan dua parameter input dengan melakukan proses clustering yaitu Eps dan MinPTS. Epsilon merupakan jarak maksimal antara dua data dalam satu cluster yang diizinkan, dan minimum point merupakan banyaknya data minimal dalam jarak epsilon agar terbentuk suatu cluster. Algoritma DBSCAN sendiri mempunyai performa yang baik diantaranya, DBSCAN dapat bekerja dengan sangat cepat dan efisien dalam menangani dataset yang besar dan dapat menghasilkan waktu eksekusi yang lebih cepat dibandingkan algoritma lainnya. Selain itu algoritma DBSCAN juga dapat menangani dataset dengan noise atau outlier secara efektif, dan mampu mengidentifikasi juga memberi label pada titik data yang tidak termasuk kedalam cluster sebagai noise (David, 2020). Algoritma DBSCAN pun mempunyai kelemahan dalam menentukan nilai epsilon yang sesuai agar dapat menghasilkan cluster yang optimal, Selain itu DBSCAN juga kurang cocok untuk dataset dengan kepadatan yang berbeda-beda, karena algoritma ini memungkinkan kesulitan dalam mengelompokkan titik data dengan kepadatan yang berbeda.

Berdasarkan beberapa penelitian yang menggunakan algoritma dbscan telah dilakukan, diantaranya penelitian yang berjudul segmentasi pelanggan E-money dengan menggunakan algoritma DBSCAN di provinsi Jawa Barat, yang membahas tentang Berdasarkan k-dist graph parameter Eps (ϵ) dan MinObj yang optimal yang MinObj=2 dan 3, $\epsilon=0,25$ dan $\epsilon=0,3$. Dengan menggunakan Eps (ϵ) sebesar 0,25 dan MinObj sebesar 3 menghasilkan nilai Silhouette Coefficient sebesar 0,26 dan 17 noise. Parameter tersebut membentuk 2 segmen yaitu segmen 1 dan segmen 2. Pengguna E-money dalam segmen 2 adalah pelanggan potensial karena memiliki rata-rata nominal saldo dan banyaknya transaksi lebih besar dari rata-rata seluruh data. (Rohalidyawati et al., 2020) Lalu selanjutnya pada penelitian yang membandingkan algoritma DBSCAN dan K-means dengan menghasilkan Pengujian validitas cluster terhadap algoritma DBSCAN dan K-Means menggunakan nilai Silhouette Index (SI) sudah dilakukan. klusterisasi menggunakan DBSCAN yang memiliki nilai SI terbaik yaitu 0.3624 dengan nilai Eps 0,2 dan nilai MinPts 3. Sementara dari hasil klusterisasi menggunakan K-Means, nilai SI terbaik diperoleh percobaan k=8 dengan nilai 0,6902. Maka pada penelitian ini, algoritma K-Means memiliki nilai validitas cluster lebih baik dibandingkan algoritma DBSCAN. Dengan demikian, pada diperoleh cluster paling optimal yaitu percobaan menggunakan algoritma K-Means dengan nilai k=8. (Adha et al., 2021)

Maka dari permasalahan dan penjelasan yang telah disebutkan diatas. dalam penelitian ini akan menerapkan algoritma DBSCAN pada segmentasi kasus data covid 19 terutama di Jawa Barat. Fokus dalam penelitian ini hanya pada data covid-19 di Jawa Barat

METODE PENELITIAN

Pada penelitian ini terdapat beberapa tahapan yang telah dilakukan, seperti yang tertera pada gambar 1.1



1.1.1. Pengumpulan Dataset

Dataset didapatkan melalui website opendata.jabarpov.go.id dalam web dari tahun 2020 – 2023. Pada hasil pengambilan data yang dapat diperoleh adalah sebesar 30.000 data yang tersimpan pada ekstensi CSV. Informasi yang diperoleh dari data tersebut berisi tentang tanggal, konfirmasi total, konfirmasi sembuh, konfirmasi meninggal dan konfirmasi aktif secara harian selama rentan waktu 3 tahun lamanya.

1.1.2. Preprocessing

Preprocessing (pre-processing) dalam konteks pemrosesan data mengacu pada serangkaian langkah atau teknik yang diterapkan pada data mentah sebelum masuk ke tahap analisis atau pemodelan. Tujuan dari preprocessing adalah untuk membersihkan, mengubah format, mengatur ulang, atau mengekstraksi informasi yang relevan dari data sehingga dapat diolah lebih lanjut dengan lebih efektif dan akurat. Dalam fase ini akan ada beberapa yaitu sebagai berikut :

a. Normalization min max scaler

Normalization menggunakan metode Min-Max Scaler adalah salah satu teknik yang umum digunakan dalam preprocessing data untuk mengubah skala data numerik ke dalam rentang yang terbatas, biasanya antara 0 dan 1. Tujuan dari Normalization menggunakan Min-Max Scaler adalah untuk membuat distribusi data menjadi seragam dan mempertahankan hubungan relatif antara nilai-nilai tersebut.

1.1.3. Segmentasi Dengan Algoritma DBSCAN

Dari data yang telah melalui tahapan pengumpulan data dan preprocessing maka tahapan selanjutnya adalah segmentasi dengan menggunakan algoritma dbscan dengan langkah sebagai berikut :

- Inisialisasi: Tentukan parameter untuk algoritma DBSCAN. Parameter utama adalah epsilon (ϵ), yang merupakan jarak maksimum antara dua titik yang dianggap tetangga, dan MinPts (minimum points), yang merupakan jumlah minimum titik tetangga dalam suatu wilayah untuk memenuhi definisi kluster.
- Pemrosesan titik data: Ambil titik data pertama dan periksa apakah titik tersebut telah dikunjungi atau diklasifikasikan sebelumnya. Jika sudah, lanjutkan ke titik data berikutnya. Jika belum, tandai titik tersebut sebagai dikunjungi.
- Cari tetangga: Temukan semua titik tetangga yang berada dalam jarak ϵ dari titik data yang sedang diproses. Jika jumlah tetangga yang ditemukan lebih besar atau sama dengan MinPts, titik data tersebut dianggap sebagai titik inti (core point). Jika jumlah tetangga kurang dari MinPts, titik data tersebut dianggap sebagai titik batas (border point).
- Ekspansi kluster: Jika titik data saat ini adalah titik inti, buat kluster baru dan tambahkan titik tersebut ke kluster. Lanjutkan dengan menemukan dan menambahkan titik-titik tetangga yang terkait ke kluster. Terus cari tetangga baru dan tambahkan mereka ke kluster jika mereka memenuhi kriteria tetangga ϵ dan MinPts. Ulangi proses ini hingga tidak ada tetangga baru yang dapat ditambahkan ke kluster.
- Pergi ke titik data berikutnya: Setelah selesai memperluas kluster saat ini, pergi ke titik data berikutnya yang belum dikunjungi dan ulangi langkah-langkah 3 hingga 5.
- Noise: Jika sebuah titik data tidak memenuhi kriteria sebagai titik inti atau batas, tandai titik tersebut sebagai noise atau outlier. Ini berarti titik tersebut tidak termasuk dalam kluster apa pun.
- Ulangi proses: Ulangi langkah-langkah 2 hingga 6 untuk semua titik data yang belum dikunjungi atau belum diklasifikasikan.

HASIL DAN PEMBAHASAN

3.1 Analisis Sistem Segmentasi Kasus Covid 19

Analisis sistem suatu proses untuk mengevaluasi serta menganalisis sistem segmentasi, pada proses ini terdiri dari pengumpulan dataset Covid 19 yang berisi dengan jumlah pasien positif, negatif, meninggal dan sembuh dari setiap kabupaten dan kota yang ada di Jawa Barat. Setelah dataset dikumpulkan lalu dilakukan dengan preprocessing data seperti menghapus data konfirmasi yang masih 0, selanjutnya dilakukan inisialisasi data dengan menentukan nilai epsilon (ϵ) dan minPTS (jumlah

minimum tetangga) yang akan digunakan pada algoritma selanjutnya menentukan titik data secara acak, lalu mengecek apakah jumlah tetangga dalam radius epsilon (ϵ) dari titik tersebut memenuhi syarat untuk menjadi titik inti (core point) dengan jumlah tetangga lebih besar atau sama dengan MinPts. Lalu jika sudah selanjutnya adalah jika titik tersebut adalah titik inti, lakukan pengunjungan tetangga-tetangga dalam radius epsilon (ϵ) dari titik tersebut dan tambahkan tetangga-tetangga ini ke dalam kluster yang sesuai. Jika tetangga-tetangga ini juga merupakan titik inti, ulangi proses pengunjungan tetangga-tetangga mereka sampai tidak ada tetangga baru yang memenuhi syarat. Jika sudah selanjutnya membangun model DBSCAN dengan parameter yang telah ditentukan untuk membangun model segmentasi. DBSCAN akan mengelompokkan titik data berdasarkan kepadatan mereka dan mengenali titik noise atau outlier.

1.3. Perhitungan Dengan DBSCAN.

1.3.1. Pengumpulan Data

Data yang digunakan pada penelitian ini merupakan data kasus covid-19 pada 28 kabupaten yang ada pada provinsi Jawa Barat. Atribut yang digunakan adalah total kasus (TC), total kematian (TD), total sembuh (TS), total meninggal (TM), total daily growth (TDG), total aktif growth (TAG), total meninggal growth (TMG), total sembuh growth (TSG) pada tanggal 22 April 2020. Adapun sumber data pada penelitian ini berasal dari halaman web <https://opendata.jabarprov.go.id/> berikut hasil cleaning data Tabel 3.4 sebagai berikut:

Tabel 3.1 Contoh Data Mentah

Tanggal	nama_kab_kota	TC	TD	TS	TM	TDG	TAG	TMG	TSG
22/04/2020	Kabupaten Bandung	57	2	5	50	1	0	0	1
22/04/2020	Kabupaten Bandung Barat	41	1	1	39	0	0	0	0
22/04/2020	Kabupaten Bekasi	107	25	9	73	1	2	0	-1
22/04/2020	Kabupaten Bogor	92	26	6	60	0	0	0	0
22/04/2020	Kabupaten Ciamis	5	0	0	5	0	0	0	0
22/04/2020	Kabupaten Cianjur	11	0	1	10	0	0	0	0
22/04/2020	Kabupaten Cirebon	5	2	0	3	0	0	0	0
22/04/2020	Kabupaten Garut	7	1	0	6	0	0	0	0
22/04/2020	Kabupaten Indramayu	12	0	0	12	0	0	0	0
22/04/2020	Kabupaten Karawang	12	8	0	4	0	0	0	0
22/04/2020	Kabupaten Kuningan	7	1	2	4	0	0	0	0
22/04/2020	Kabupaten Majalengka	2	0	1	1	0	0	0	0
22/04/2020	Kabupaten Pangandaran	2	0	0	2	0	0	0	0
22/04/2020	Kabupaten Purwakarta	21	2	0	19	0	0	0	0
22/04/2020	Kabupaten Subang	28	0	1	27	0	0	0	0
22/04/2020	Kabupaten Sukabumi	23	1	0	22	0	0	0	0
22/04/2020	Kabupaten Sumedang	10	1	0	9	0	0	0	0
22/04/2020	Kabupaten Tasikmalaya	1	0	0	1	0	0	0	0
22/04/2020	Kota Bandung	269	24	31	214	0	1	0	-1
22/04/2020	Kota Banjar	5	0	0	5	0	0	0	0
22/04/2020	Kota Bekasi	285	36	6	243	7	0	0	7
22/04/2020	Kota Bogor	97	39	11	47	0	0	0	0
22/04/2020	Kota Cimahi	64	4	1	59	0	0	0	0
22/04/2020	Kota Cirebon	4	0	1	3	0	0	0	0
22/04/2020	Kota Depok	308	71	23	214	0	2	1	-3
22/04/2020	Kota Sukabumi	36	0	0	36	0	0	0	0
22/04/2020	Kota Tasikmalaya	20	1	0	19	0	0	0	0

1.3.2. Preprocessing

Preprocessing merupakan salah satu tahapan yang penting pada proses *mining*. Data yang digunakan dalam *proses mining* tidak selamanya dalam kondisi yang *ideal* untuk diproses.

Terkadang pada data tersebut terdapat berbagai permasalahan yang dapat mengganggu hasil dari proses *mining* itu sendiri seperti diantaranya adalah *missing value*, *data redundant*, *outliers*, atau pun format data yang tidak sesuai dengan sistem. Oleh karena itu untuk mengatasi permasalahan tersebut dibutuhkan tahap *Preprocessing*. *Preprocessing* merupakan salah satu tahapan menghilangkan permasalahan – permasalahan yang dapat mengganggu hasil daripada proses data.

1.3.2.1. Normalisasi MinMax

Metode normalisasi melakukan tranformasi atribut numeric dalam skala yang lebih kecil yaitu dengan batas terendah 0 dan batas teratas 1. Hasil dari normalisasi dengan nilai minimum 0 dan maksimal 1 dapat dilihat pada tabel 3.2

Tabel 3. 2 Tabel Perhitungan Normalisasi MinMax

konfirmasi_ total	konfirmasi_se mbuh	konfirmasi_men inggal	konfirmasi_ aktif	konfirmasi_ total
Kabupaten Bandung	0,542748	0,484745	0,262621	0,716228
Kabupaten Bandung Barat	0,251003	0,236397	0,080021	0,150529
Kabupaten Bekasi	0,899302	0,886169	0,227175	0,083403
Kabupaten Bogor	0,977278	0,949844	0,034909	0,348716
Kabupaten Ciamis	0,123537	0,114018	0,097207	0,07216
Kabupaten Cianjur	0,104487	0,100195	0,042965	0,014096
Kabupaten Cirebon	0,29606	0,28122	0,435016	0,04296
Kabupaten Garut	0,321109	0,296727	0,594522	0,130055
Kabupaten Indramayu	0,180195	0,158724	0,329216	0,17117
Kabupaten Karawang	0,573387	0,526329	1	0,328746
Kabupaten Kuningan	0,121444	0,113199	0,075188	0,060748
Kabupaten Majalengka	0,089986	0,084069	0,064984	0,030542
Kabupaten Pangandaran	0	-0,00249	0,005371	0
Kabupaten Purwakarta	0,126402	0,118193	0,245435	0,007048
Kabupaten Subang	0,09312	0,078386	0,079484	0,152542
Kabupaten Sukabumi	0,114111	0,106101	0,103115	0,048666
Kabupaten Sumedang	0,081672	0,062154	0,012889	0,241987
Kabupaten Tasikmalaya	0,049221	0,020044	0,089689	0,356603
Kota Bandung	1	0,924397	0,162728	1
Kota Banjar	0,000784	-0,00277	0	0,016949

1.3.3. Segmentasi Dengan DBSCAN

Langkah – langkah yang dibutuhkan dalam segmentasi kasus covid 19 menggunakan algoritma dbscan sebagai berikut

1. Tentukan parameter Epsilon dan MinPts

Eps = 4 dan MinPts = 4

Untuk menghitung jarak titik core point dengan point lain pada DBSCAN bisa dilakukan dengan menggunakan rumus jarak biasa yaitu Euclidean Distance :

$$Jarak = \sqrt{(x - xp)^2 + (y - yp)^2}$$

Dimana :

x = koordinat sumbu x titik tujuan

y = koordinat sumbu y titik tujuan

xp = koordinat pusat sumbu x

yp = koordinat pusat sumbu y

Iterasi 1

Pada iterasi 1 titik pusat ditentukan secara acak dengan titik pusat adalah kabupaten garut dengan wp = 92 , xp = 26 , yp = 6 , zp = 60. Berikut merupakan perhitungan jarak setiap titik dengan titik pusat.

$$AD : \sqrt{(57 - 92)^2 + (2 - 26)^2 + (5 - 6)^2 + (50 - 60)^2} = 642$$

$$BD : \sqrt{(41 - 92)^2 + (1 - 26)^2 + (1 - 6)^2 + (39 - 60)^2} = 1.040$$

$$CD : \sqrt{(107 - 92)^2 + (25 - 26)^2 + (9 - 6)^2 + (73 - 0)^2} = 194$$

$$DD : \sqrt{(92 - 92)^2 + (26 - 26)^2 + (6 - 6)^2 + (60 - 60)^2} = 0$$

$$ED : \sqrt{(5 - 92)^2 + (0 - 26)^2 + (0 - 6)^2 + (5 - 60)^2} = 3.650$$

$$FD : \sqrt{(11 - 92)^2 + (0 - 26)^2 + (1 - 6)^2 + (10 - 60)^2} = 3.120$$

$$GD : \sqrt{(5 - 92)^2 + (2 - 26)^2 + (0 - 6)^2 + (3 - 60)^2} = 3.174$$

$$HD : \sqrt{(7 - 92)^2 + (7 - 26)^2 + (0 - 6)^2 + (6 - 60)^2} = 3.228$$

$$ID : \sqrt{(12 - 92)^2 + (0 - 26)^2 + (0 - 6)^2 + (12 - 60)^2} = 2.936$$

Setelah selesai proses perhitungan maka dapat dilihat dalam bentuk tabel

Jarak	Hasil
AD	642
BD	1,040
CD	194
DD	0
ED	3.650
FD	3.120
GD	3.174
HD	3.228
ID	2.936

Tabel 3. 3 Data Jarak Iterasi I

Density reachable didapatkan jika titik termasuk dalam radius dengan nilai jarak yang sudah ditentukan kurang atau sama dengan eps yang sudah ditentukan sebelumnya.

Berikut tabel titik yang sudah terseleksi dengan ketentuan nilai Eps = 4

Titik	Kota / Kab	TC	TD	TS	TM	Jarak Ke Titik D
H	Kabupaten Garut	7	1	0	6	3.228
D	Kabupaten Bogor	92	26	6	60	0
F	Kabupaten Cianjur	11	0	1	10	3.120
G	Kabupaten Cirebon	5	2	0	3	3.174
I	Kabupaten Indramayu	12	0	3,04	12	2.936

Tabel 3. 4 Titik terseleksi dengan ketentuan Nilai Eps I

Neighborhood Core Object tercapai dikarenakan total e-neighborhood telah melingkupi syarat karena total titik lebih dari sama dengan MinPts = 4. Titik selanjutnya dipilih berdasarkan jarak terjauh dari core object yaitu titik G

Iterasi II

Pada iterasi II titik pusat didapat dari iterasi I yaitu G dengan $W_p = 5$, $x_p = 2$, $y_p = 0$ dan $z_p = 3$, Berikut tabel data perhitungan jarak seluruh point terhadap titik pusat iterasi II

Jarak	Hasil
AG	2.286
BG	1.336
CG	5.612
DG	3.948
EG	8
FG	60
GG	0
HG	11
IG	92

Tabel 3. 5 Data Jarak Iterasi II

Titik	Kota / Kab	TC	TD	TS	TM	Jarak Ke Titik G
H	Kabupaten Garut	7	1	0	6	11
D	Kabupaten Bogor	92	26	6	60	3.948
F	Kabupaten Cianjur	11	0	1	10	60
G	Kabupaten Cirebon	5	2	0	3	0
I	Kabupaten Indramayu	12	0	0	12	92

Tabel 3. 6 Titik terseleksi dengan ketentuan Nilai Eps II

Dapat dilihat titik D adalah titik terjauh akan tetapi tidak dapat dipilih karena sudah pernah dikunjungi dan titik terjauh selanjutnya adalah yaitu F dan H tidak dapat dipilih karena masuk border point di core object pada iterasi sebelumnya maka titik yang dapat dipilih adalah titik I.

Iterasi III

Pada iterasi III titik pusat didapat dari iterasi II yaitu I dengan $W_p = 12$, $x_p = 0$, $y_p = 0$ dan $z_p = 12$, Berikut tabel data perhitungan jarak seluruh point terhadap titik pusat iterasi III

Jarak	Hasil
AI	1.158
BI	760
CI	3.818
DI	3.089
EI	42
FI	4
GI	78
HI	80
I	0

Tabel 3. 7 Data Jarak Iterasi III

Titik	Kota / Kab	TC	TD	TS	TM	Jarak Ke Titik G
H	Kabupaten Garut	7	1	0	6	80
D	Kabupaten Bogor	92	26	6	60	3.089
F	Kabupaten Cianjur	11	0	1	10	4
G	Kabupaten Cirebon	5	2	0	3	78
I	Kabupaten Indramayu	12	0	0	12	0

Tabel 3. 8 Titik terseleksi dengan ketentuan Nilai Eps III

Dari hasil tersebut tidak ada lagi titik yang dapat dijadikan core object. Maka akan ditentukan titik lainnya dalam database.

Iterasi IV

Pada iterasi IV titik pusat ditentukan kembali dari awal yaitu titik A dengan $W_p = 57$, $x_p = 2$, $y_p = 5$ dan $z_p = 50$, Berikut tabel data perhitungan jarak seluruh point terhadap titik pusat iterasi IV

Jarak	Hasil
AA	0
BA	102
CA	1.124
DA	812
EA	2.002
FA	1.575
GA	2.182
HA	1.912
IA	1.428

Tabel 3. 9 Data jarak Iterasi IV

Titik	Kota / Kab	TC	TD	TS	TM	Jarak Ke Titik A
A	Kabupaten Bandung	57	2	5	50	0
B	Kabupaten Bandung Barat	41	1	1	39	102
C	Kabupaten Bekasi	107	25	9	73	1.124
E	Kabupaten Ciamis	5	0	0	5	2.002

Tabel 3. 10 Titik terseleksi dengan ketentuan Nilai Eps IV

Iterasi V

Pada iterasi V titik pusat didapat dari iterasi IV yaitu titik B dengan $W_p = 41$, $x_p = 1$, $y_p = 1$ dan $z_p = 39$, Berikut tabel data perhitungan jarak seluruh point terhadap titik pusat iterasi V

Jarak	Hasil
AB	154
BB	0
CB	1.862
DB	1.142
EB	1.131
FB	812
GB	1.262
HB	1.056
IB	702

Tabel 3. 11 Data Jarak Iterasi V

Titik	Kota / Kab	TC	TD	TS	TM	Jarak Ke Titik A
A	Kabupaten Bandung	57	2	5	50	154
B	Kabupaten Bandung Barat	41	1	1	39	0
C	Kabupaten Bekasi	107	25	9	73	1.862
E	Kabupaten Ciamis	5	0	0	5	1.142

Tabel 3. 12 Titik terseleksi dengan ketentuan Nilai Eps V

Dari hasil density reachable yang didapat sama seperti iterasi sebelumnya, maka tidak ada titik yang dapat dipilih kembali dan proses selesai. Berikut tabel hasil dari iterasi I sampai V

Iterasi I	Iterasi II	Iterasi III	Iterasi IV	Iterasi V
H	H	D	A	A
D	D	F	B	B
F	F	G	C	C
G	G	I	E	E

I	I
---	---

Tabel 3. 13 Data Iterasi I Sampai V

Berdasarkan tabel dapat dilihat secara keseluruhan core point yaitu D, G, I, A, B. Titik D, G dan I saling berkaitan maka dianggap satu cluster. Tidak ada data noise karena seluruh titik masuk ke dalam cluster. Dari perhitungan diatas dengan menggunakan algoritma DBSCAN maka hasil cluster seperti pada tabel 3.14 berikut :

Cluster I	Cluster II
D	A
F	B
G	C
H	E
I	
J	

Tabel 3. 14 Data Cluster

Dari kedua cluster dapat dibandingkan, maka dapat disimpulkan wilayah yang terkena covid 19 lebih banyak di Jawa Barat berada pada cluster I yaitu alternatif D, F, G, H, I dan J

KESIMPULAN

Dalam studi ini, dilakukan segmentasi kasus COVID-19 menggunakan algoritma DBSCAN (Density-Based Spatial Clustering of Applications with Noise). DBSCAN adalah algoritma yang dapat mengelompokkan data berdasarkan kerapatan spasial, sehingga cocok untuk menganalisis data geografis yang terkait dengan penyebaran COVID-19.

Pada penelitian ini, data kasus COVID-19 dari berbagai lokasi digunakan untuk melakukan segmentasi. Algoritma DBSCAN mampu mengidentifikasi kelompok kasus yang memiliki kerapatan tinggi di sekitar titik pusat atau klaster, serta mengenali titik-titik yang berada di luar klaster dan dianggap sebagai noise.

Percobaan pertama penelitian ini menggunakan nilai Eps = 0,5 dan Min Pts = 15 dan Silhouette = 0.25, dengan 4 parameter yaitu sembuh, aktif, meninggal dan total, lalu menghasilkan sebanyak 7 cluster lalu pada percobaan kedua penelitian menaikkan angkanya dengan Eps = 10 dan Min Pts = 20 dan Silhouette = 0.50, pada percobaan kedua cluster yang dihasilkan ada 3 cluster, maka itu dapat disimpulkan bahwa percobaan pertama mendapatkan angka akurasi sembuh, meninggal, aktif dan total lebih besar dari pada percobaan kedua. Dengan nilai akurasi sebesar 0.0099502487562189.

REFERENSI

- Adha, R., Nurhaliza, N., Sholeha, U., & Mustakim, M. (2021). Perbandingan Algoritma DBSCAN dan K-Means Clustering untuk Pengelompokan Kasus Covid-19 di Dunia. *SITEKIN: Jurnal Sains, Teknologi Dan Industri*, 18(2), 206–211.
- David. (2020). DBSCAN CLUSTERING.
- Noviyanto. (2020). Penerapan Data Mining dalam Mengelompokkan Jumlah Kematian. *Jurnal Informatika Dan Komputer*, 22(2), 183–188.
- Raras, C., Widiawati, A., Setiawan, P., Komarasary, D., Studi, P., Informasi, T., ... Amikom, U. (2023). *Pengaruh Segmentasi terhadap Diagnosis COVID-19 pada Citra X-Ray Paru*. 9(1), 171–180.
- Rasmussen, S. A., Smulian, J. C., Lednicky, J. A., Wen, T. S., & Jamieson, D. J. (2020). Coronavirus Disease 2019 (COVID-19) and pregnancy: what obstetricians need to know. *American Journal of Obstetrics and Gynecology*, 222(5), 415–426.
<https://doi.org/10.1016/j.ajog.2020.02.017>
- Rohalidyawati, W., Rahmawati, R., & Mustafid, M. (2020). Segmentasi Pelanggan E-Money Dengan Menggunakan Algoritma DbSCAN (Density Based Spatial Clustering Applications

With Noise) Di Provinsi Dki Jakarta. *Jurnal Gaussian*, 9(2), 162–169.

<https://doi.org/10.14710/j.gauss.v9i2.27818>

Sohrabi, C., Alsafi, Z., O'Neill, N., Khan, M., Kerwan, A., Al-Jabir, A., ... Agha, R. (2020). World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19). *International Journal of Surgery*, 76(February), 71–76.

<https://doi.org/10.1016/j.ijsu.2020.02.034>