

Pendekatan Metode DBSCAN dan Fuzzy C-Means untuk Klasterisasi Skala Prioritas Stunting

Putri Natasya Lubis^{*1}, Muhammad Siddik Hasibuan²

^{1,2}Program Studi Ilmu Komputer, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara

Email: ¹pnatasyalubis@gmail.com, ²muhammadsiddik@uinsu.ac.id

Abstrak

Stunting merupakan permasalahan kesehatan yang berdampak pada perkembangan fisik dan kognitif anak, serta berkontribusi terhadap kesenjangan sosial dan ekonomi di Indonesia. Penelitian ini bertujuan untuk mengembangkan metode klasterisasi skala prioritas penanganan stunting dengan menggabungkan algoritma DBSCAN (Density-Based Spatial Clustering of Applications with Noise) dan Fuzzy C-Means (FCM). Data diperoleh dari platform Aksi Bangda Kemendagri (2021–2024) dan diolah menggunakan Python. DBSCAN digunakan untuk mengidentifikasi wilayah dengan konsentrasi tinggi kasus stunting dan mendeteksi outlier, sementara FCM membantu menentukan prioritas intervensi berdasarkan tingkat keparahan. Hasil penelitian menunjukkan bahwa DBSCAN menghasilkan dua kluster utama dan sejumlah outlier, sedangkan FCM membagi data menjadi tiga kluster berbasis derajat keanggotaan. Pendekatan ini berpotensi menjadi alat analitik dalam mendukung kebijakan percepatan penurunan angka stunting secara lebih efektif di Indonesia.

Kata kunci: DBSCAN, Fuzzy C-Means, Stunting, Kebijakan Intervensi, Klasterisasi

DBSCAN and Fuzzy C-Means Method Approach for Stunting Priority Scale Clustering

Abstract

Stunting is a health problem that impacts children's physical and cognitive development, and contributes to social and economic disparities in Indonesia. This research aims to develop a priority scale clustering method for handling stunting by combining the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and Fuzzy C-Means (FCM) algorithms. Data was obtained from the Ministry of Home Affairs' Bangda Action platform (2021–2024) and processed using Python. DBSCAN is used to identify areas with a high concentration of stunting cases and detect outliers, while FCM helps determine intervention priorities based on severity level. The research results show that DBSCAN produces two main clusters and a number of outliers, while FCM divides the data into three clusters based on degree of membership. This approach has the potential to become an analytical tool to support policies to accelerate the reduction of stunting rates more effectively in Indonesia.

Keywords: DBSCAN, Clustering, Fuzzy C-Means, Intervention Policy, Stunting

1. PENDAHULUAN

Stunting bukan hanya sekadar masalah kesehatan, tetapi juga ancaman serius bagi masa depan bangsa. Ketika anak-anak mengalami stunting, mereka tidak hanya terhambat secara fisik, tetapi juga dalam kemampuan kognitif, yang berpotensi memengaruhi kualitas hidup mereka sepanjang hayat [1]. Di Indonesia, lebih dari satu dari lima anak berada dalam kondisi ini, menjadikan stunting sebagai salah satu tantangan terbesar yang harus dihadapi. Dampaknya tidak hanya dirasakan pada level individu, tetapi juga pada skala nasional, di mana stunting dapat menghambat perkembangan ekonomi dan memperburuk kesenjangan sosial. Dalam konteks global, Indonesia memiliki komitmen kuat untuk menurunkan angka stunting, namun upaya ini memerlukan strategi yang lebih tajam dan terfokus agar bisa benar-benar berdampak [2].

Dalam menghadapi masalah stunting yang kompleks, menentukan skala prioritas penanganan menjadi langkah penting yang tidak bisa diabaikan. Setiap wilayah memiliki karakteristik dan tantangan unik yang mempengaruhi tingkat keparahan stunting, sehingga diperlukan pendekatan yang cerdas untuk menentukan di mana dan bagaimana intervensi harus dilakukan [3]. Tanpa pemetaan yang tepat, risiko sumber daya yang terbuang sia-sia semakin besar, sementara anak-anak yang paling membutuhkan bantuan justru terlewatkan. Oleh karena itu, pendekatan berbasis data yang komprehensif sangat diperlukan untuk mengidentifikasi wilayah dan kelompok

yang paling rentan, serta menentukan jenis intervensi yang paling efektif. Dengan strategi prioritas yang tepat, kita dapat memastikan bahwa upaya penanganan stunting tidak hanya tepat sasaran, tetapi juga memberikan dampak yang nyata dan berkelanjutan bagi generasi mendatang [4].

Dalam permasalahan diatas untuk menghadapi tantangan tersebut, metode *Clustering* berbasis data seperti DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) dan *Fuzzy C-Means* (FCM) menawarkan solusi potensial dalam mengidentifikasi wilayah prioritas stunting [5]. DBSCAN memungkinkan identifikasi area dengan konsentrasi tinggi kasus stunting, termasuk deteksi wilayah yang mungkin terlewatkan oleh metode lain, sedangkan FCM dapat membantu dalam menentukan prioritas intervensi berdasarkan tingkat keparahan dengan mempertimbangkan keanggotaan ganda dari berbagai kluster. Dengan menggabungkan kedua pendekatan ini, penelitian ini bertujuan untuk mengembangkan model yang mampu memetakan skala prioritas penanganan stunting secara lebih akurat dan dinamis [6]. Model ini diharapkan tidak hanya memberikan wawasan baru dalam upaya penanggulangan stunting, tetapi juga dapat diimplementasikan sebagai alat pendukung kebijakan untuk mempercepat penurunan angka stunting di Indonesia [7].

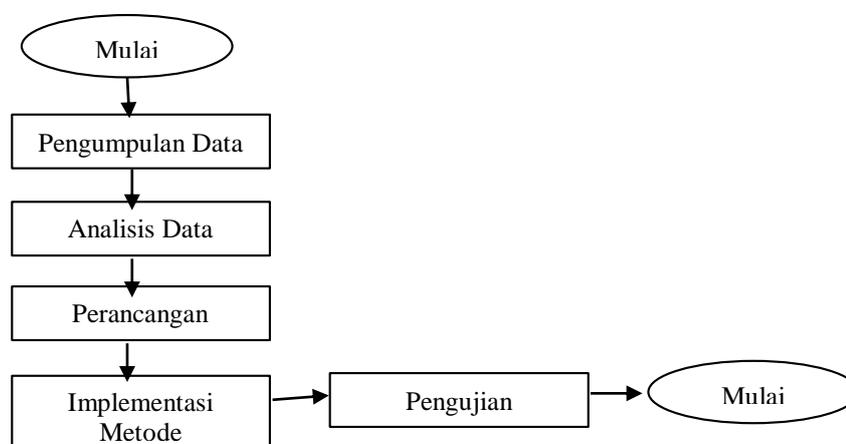
Penelitian seperti ini telah diteliti oleh beberapa ahli, seperti yang diteliti oleh [8] yang berjudul “Klasifikasi Status Stunting Balita menggunakan Algoritma *Fuzzy C-Means* (Studi Kasus Posyandu RW 01 Kelurahan Jepara Surabaya)”. Dalam penelitian ini menggunakan algoritma *Fuzzy C-Means*. algoritma *Fuzzy C-Means* digunakan untuk Klasifikasi Status Stunting Balita. Algoritma FCM dipilih karena kemampuannya dalam mengelompokkan data dengan karakteristik yang mirip namun tetap mempertimbangkan tingkat ketidakpastian dalam data tersebut. FCM membantu dalam mengidentifikasi balita dengan risiko stunting dengan lebih tepat, memungkinkan adanya derajat keanggotaan yang menunjukkan seberapa kuat atau lemahnya seorang balita termasuk dalam kategori stunting [9].

Ada juga yang telah diteliti oleh [10] yang berjudul “Implementation of *Fuzzy C-Means* in *Clustering Stunting Prone Areas*”. Dalam penelitian ini menggunakan metode *Fuzzy C-Means*. Metode *Fuzzy C-Means* digunakan untuk *Clustering Stunting Prone Areas*. Metode FCM dipilih karena kemampuannya dalam menangani data yang memiliki ketidakpastian dan keburaman, sehingga lebih akurat dalam mengelompokkan daerah berdasarkan tingkat kerentanannya terhadap stunting.

Persamaan antara penelitian terdahulu dan penelitian penulis terletak pada penggunaan metode *Fuzzy C-Means* (FCM) untuk mengatasi ketidakpastian dalam data terkait stunting. Kedua penelitian terdahulu menerapkan FCM untuk mengelompokkan atau mengklasifikasikan data stunting, baik itu dalam konteks status stunting pada balita maupun kerentanan wilayah terhadap stunting [11]. Pembahasan yang penulis lakukan adalah dengan menggabungkan metode *Fuzzy C-Means* dengan metode DBSCAN untuk menentukan skala prioritas stunting, menawarkan pendekatan yang lebih komprehensif. DBSCAN memberikan kemampuan tambahan untuk mendeteksi kelompok dengan kepadatan tinggi (*high-density Clusters*) dan *outliers* yang mungkin tidak terdeteksi dengan FCM saja. Ini memungkinkan penentuan skala prioritas intervensi yang lebih tepat berdasarkan pola distribusi dan kepadatan kasus stunting, serta menggunakan bahasa *Python* dalam pengolahan data [12].

2. METODE PENELITIAN

2.1 Rencana Pembahasan



Gambar 1. Diagram Alur Penelitian

Analisis data adalah proses mengolah, membersihkan, mengubah, dan memodelkan data untuk mendapatkan informasi yang bermakna, mendukung pengambilan keputusan, dan menjawab pertanyaan penelitian. Melalui

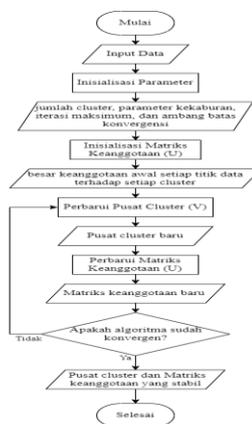
analisis data, pola, tren, dan hubungan dalam data dapat diidentifikasi, yang membantu dalam memahami fenomena tertentu atau memprediksi kejadian di masa depan. Analisis data dalam pendekatan metode DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) dan *Fuzzy C-Means* digunakan untuk mengidentifikasi dan mengelompokkan wilayah atau populasi berdasarkan tingkat prevalensi stunting, dengan tujuan menentukan skala prioritas penanganan. DBSCAN berfungsi untuk mendeteksi pola spasial dan mengidentifikasi kluster stunting yang padat sekaligus mengabaikan *outlier*, sedangkan *Fuzzy C-Means* memungkinkan penentuan keanggotaan ganda setiap data dalam beberapa kluster, memberikan fleksibilitas dalam pengklasifikasian daerah dengan tingkat stunting yang beragam [13].

Tahapan analisis data pada penerapan Metode DBSCAN dan *Fuzzy C-Means* untuk klusterisasi skala prioritas stunting dimulai dengan pengumpulan data dari sumber resmi, seperti dari *platform* Aksi Bangda Kemendagri. Data yang diperoleh mencakup berbagai indikator stunting di berbagai daerah, seperti tingkat prevalensi stunting, akses ke fasilitas kesehatan, gizi anak, dan faktor-faktor sosial ekonomi lainnya. Data ini seringkali dalam bentuk mentah dan memerlukan proses awal berupa data *cleaning* untuk menghilangkan *outlier*, menangani nilai yang hilang (*missing values*), serta memastikan konsistensi dan kesesuaian format data.

Setelah proses *pre-processing* data selesai, langkah selanjutnya adalah transformasi data agar siap untuk digunakan dalam analisis klusterisasi. Pada tahap ini, data yang sudah bersih diubah menjadi bentuk yang dapat diolah oleh algoritma DBSCAN dan *Fuzzy C-Means*. Untuk DBSCAN, data akan melalui identifikasi pola densitas sehingga kluster - kluster yang dihasilkan bisa menggambarkan wilayah-wilayah yang memiliki pola distribusi stunting serupa. Algoritma ini efektif untuk menemukan kluster dengan berbagai bentuk yang tidak teratur dan mampu mengabaikan *noise* atau data yang tidak relevan. Sementara itu, *Fuzzy C-Means* akan digunakan untuk membagi data ke dalam beberapa kluster berdasarkan derajat keanggotaan, di mana setiap daerah tidak hanya diklasifikasikan ke satu kluster, tetapi dapat memiliki keanggotaan parsial di beberapa kluster, yang memberikan fleksibilitas lebih dalam penentuan skala prioritas.

Perancangan adalah proses sistematis yang melibatkan perencanaan dan pengembangan solusi untuk memenuhi kebutuhan atau menyelesaikan masalah. Ini mencakup pembuatan skema atau prototipe untuk memastikan solusi sesuai spesifikasi. Manfaatnya termasuk memastikan kebutuhan terpenuhi, mengurangi risiko, meningkatkan efisiensi, dan memfasilitasi komunikasi antar pihak, serta memungkinkan penyesuaian di masa depan.

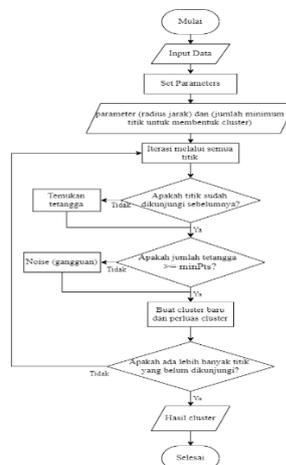
1. *Flowchart Fuzzy C-Means*



Gambar 1. *Flowchart Fuzzy C-Means*

Flowchart dimulai dengan input data dan inisialisasi parameter seperti jumlah *Cluster*, parameter keaburan, dan ambang batas konvergensi. Selanjutnya, inisialisasi matriks keanggotaan yang menunjukkan derajat keanggotaan setiap titik data terhadap setiap *Cluster*. Proses berikutnya adalah menghitung pusat *Cluster* berdasarkan matriks keanggotaan yang ada, diikuti dengan memperbarui matriks keanggotaan dengan menghitung nilai baru berdasarkan pusat *Cluster* yang diperbarui. Periksa konvergensi dengan mengevaluasi apakah perubahan dalam pusat *Cluster* dan matriks keanggotaan di bawah ambang batas yang ditetapkan atau jumlah iterasi maksimum tercapai. Jika konvergensi tercapai, hasil *Clustering* akhir disajikan, menandai akhir dari algoritma. Jika belum konvergen, proses diulang dari langkah menghitung pusat *Cluster*. Parameter FCM yang digunakan dalam analisis adalah nilai epsilon dan minPts. Algoritma ini menghasilkan beberapa *output*: *cntr* (pusat kluster), *u* (matriks keanggotaan), *u0* (nilai awal matriks keanggotaan), *d* (jarak data ke pusat kluster), *jm* (fungsi objektif di setiap iterasi), *p* (iterasi akhir), dan *fpc* (koefisien partisi *fuzzy*).

2. Flowchart algoritma DBSCAN



Gambar 2. Flowchart algoritma DBSCAN

Flowchart algoritma DBSCAN dimulai dengan memasukkan dataset dan menetapkan parameter eps (radius jarak) dan *minPts* (jumlah minimum titik untuk membentuk Cluster). Algoritma kemudian mengiterasi setiap titik dalam dataset, memeriksa apakah titik tersebut sudah dikunjungi. Jika belum, algoritma mencari tetangga dalam radius eps. Jika jumlah tetangga mencukupi, Cluster baru dibentuk dan diperluas; jika tidak, titik tersebut ditandai sebagai *noise*. Proses ini berlanjut hingga semua titik diperiksa, dan algoritma selesai ketika tidak ada lagi titik yang belum dikunjungi. parameter DBSCAN dan FCM yang digunakan dalam analisis adalah jumlah klaster, nilai fuzziness, batas iterasi.

2.2. Implementasi Metode

Implementasi metode merupakan penerapan teknik atau algoritma untuk memecahkan masalah atau mencapai tujuan secara praktis. Dalam konteks ini, pendekatan yang digunakan menggabungkan dua metode Clustering untuk secara efektif mengidentifikasi dan memprioritaskan kasus stunting. Pertama, DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) diterapkan untuk mengelompokkan data berdasarkan kepadatan, mengidentifikasi area dengan konsentrasi tinggi kasus stunting dan mengabaikan outlier. Selanjutnya, hasil Clustering dari DBSCAN digunakan sebagai input untuk Fuzzy C-Means (FCM), yang mengevaluasi derajat keanggotaan setiap kasus dalam Cluster yang telah diidentifikasi, memungkinkan analisis yang lebih mendetail dan penetapan prioritas berdasarkan intensitas dan karakteristik lokal. Kombinasi kedua metode ini memberikan pendekatan menyeluruh untuk menentukan area yang paling membutuhkan intervensi dalam upaya penanggulangan stunting.

2.3. Pengujian

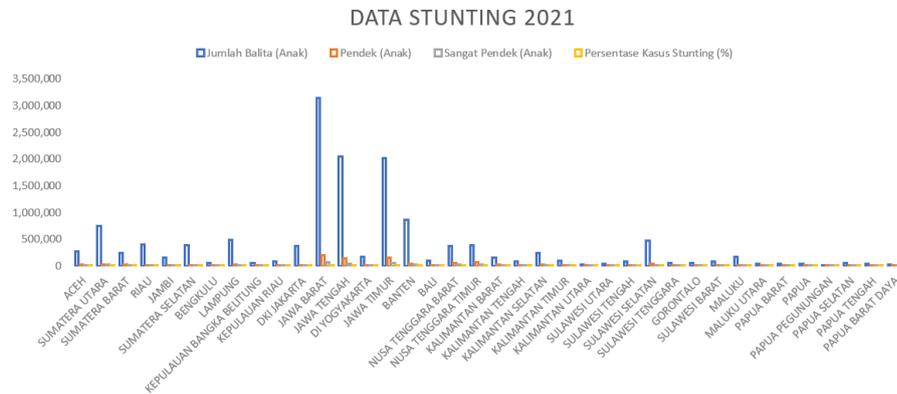
Pengujian adalah proses evaluasi dan verifikasi untuk memastikan bahwa suatu sistem, produk, atau metode berfungsi sesuai dengan spesifikasi dan memenuhi kebutuhan yang ditetapkan. Tujuan utamanya adalah untuk mengidentifikasi dan memperbaiki cacat atau masalah sebelum peluncuran atau implementasi, memastikan akurasi dan keandalan hasil, serta memverifikasi kinerja sistem atau metode dalam berbagai kondisi. Pengujian membantu mengurangi risiko kegagalan, meningkatkan kualitas dan kepuasan pengguna, serta memberikan keyakinan bahwa produk atau metode akan berfungsi secara efektif dalam situasi nyata.

3. HASIL DAN PEMBAHASAN

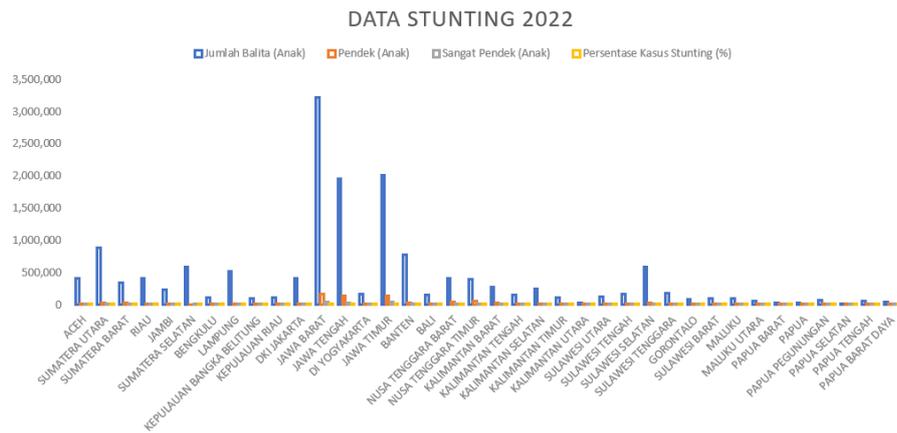
3.1. Representasi Data

Data yang akan digunakan pada penelitian ini diperoleh dari link <https://aksi.bangda.kemendagri.go.id/emonev/DashPrev>, link tersebut merupakan platform Aksi Bangsa Kementerian Dalam Negeri mencakup sebaran stunting di setiap provinsi di Indonesia dari tahun 2021 hingga 2024. Data ini akan digunakan untuk melakukan klusterisasi prioritas stunting, dengan menerapkan metode DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) dan Fuzzy C-Means. Metode DBSCAN berguna untuk mengidentifikasi kelompok provinsi dengan tingkat stunting yang serupa, serta membantu dalam

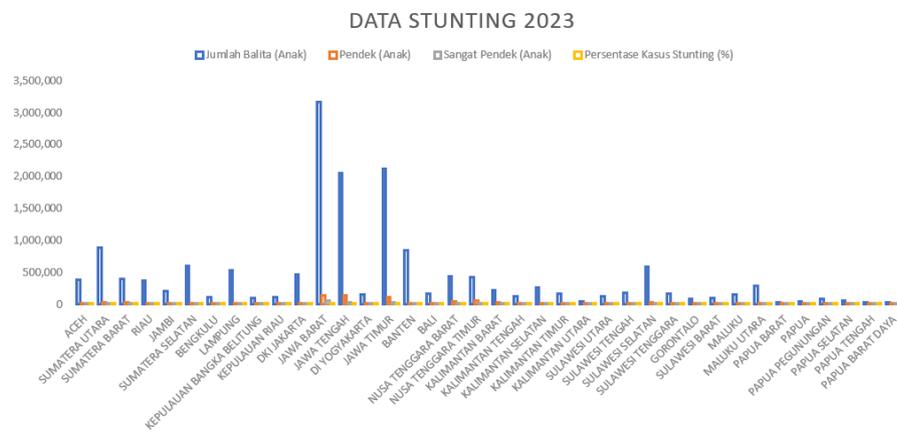
mengeteksi *outlier*, sedangkan *Fuzzy C-Means* akan memungkinkan pengelompokan data dengan pendekatan yang lebih fleksibel, di mana setiap provinsi dapat menjadi anggota dari beberapa kluster dengan tingkat keanggotaan yang berbeda. Dengan analisis ini, diharapkan dapat menghasilkan wawasan yang lebih mendalam mengenai distribusi stunting di Indonesia dan membantu dalam perumusan kebijakan yang lebih efektif untuk menanggulangi masalah stunting.



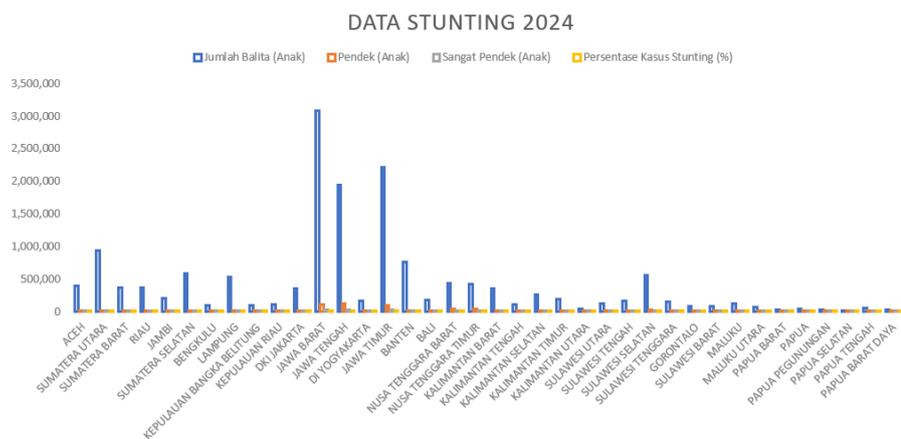
Gambar 3. Grafik Data Stunting 2021



Gambar 4. Grafik Data Stunting 2022



Gambar 5. Grafik Data Stunting 2023



Gambar 6. Grafik Data Stunting 2024

Dari gambar grafik data sebaran stunting diatas yang terdiri dari Provinsi, Jumlah Balita (Anak), Pendek (Anak), Sangat Pendek (Anak), dan Persentase Kasus Stunting (%). Sebelum melakukan klusterisasi prioritas stunting menggunakan metode DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) dan *Fuzzy C-Means*, tahap pertama yang harus dilakukan adalah *pre-processing* data. Proses *pre-processing* ini mencakup pembersihan data untuk menghilangkan nilai yang tidak valid atau hilang, normalisasi untuk memastikan bahwa semua variabel berada dalam rentang yang seragam. Dengan langkah-langkah ini, data akan siap untuk analisis yang lebih mendalam, memungkinkan pengelompokan yang lebih akurat dan pengambilan keputusan yang lebih baik dalam menangani masalah stunting di berbagai provinsi.

3.2. Implementasi Algoritma

Tahapan implementasi algoritma DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) dimulai dengan menentukan dua parameter utama, yaitu *epsilon* (ϵ) yang mengatur jarak maksimum antar titik untuk dianggap sebagai tetangga, dan *minimum points* (*minPts*) yang menetapkan jumlah titik minimum dalam radius ϵ untuk membentuk kluster. Setelah parameter ditentukan, DBSCAN akan mengunjungi setiap titik data, lalu mengelompokkan titik-titik yang berada dalam radius ϵ dengan jumlah tetangga minimal *minPts* sebagai kluster yang sama. Jika ada titik yang tidak memenuhi kriteria ini, titik tersebut akan dianggap sebagai *noise*. Selanjutnya, implementasi algoritma *Fuzzy C-Means* (FCM) dilakukan dengan menetapkan jumlah kluster (*C*) yang diinginkan dan memberikan derajat keanggotaan awal untuk setiap titik terhadap masing-masing kluster. Algoritma FCM kemudian memperbarui pusat kluster (*centroid*) berdasarkan derajat keanggotaan dan jarak antara titik dengan setiap *centroid*, serta menghitung ulang derajat keanggotaan hingga mencapai kondisi konvergensi, yaitu ketika perubahan derajat keanggotaan antar iterasi berada di bawah ambang batas toleransi yang telah ditentukan.

1. Algoritma DBSCAN

Langkah pertama dalam perhitungan manual DBSCAN adalah menentukan data yang akan digunakan. Pada kasus ini, data yang akan dianalisis terdiri dari 38 sampel terkait kasus stunting di Indonesia pada tahun 2024, yang akan digunakan sebagai *input* dalam proses pengelompokan menggunakan algoritma DBSCAN:

Tabel 1. *Dataset*

No	Jumlah Balita (Anak)	Pendek (Anak)	Sangat Pendek (Anak)	Persentase Kasus Stunting (%)
1	0.121	0.135	0.145	0.213
2	0.301	0.143	0.212	0.071
3	0.115	0.172	0.176	0.298
4	0.112	0.046	0.047	0.049
5	0.06	0.024	0.029	0.058
6	0.184	0.038	0.048	0
7	0.027	0.019	0.007	0.111
8	0.166	0.078	0.08	0.062
9	0.024	0.011	0.006	0.084
10	0.031	0.015	0.008	0.08
11	0.11	0.023	0.033	0.009

No	Jumlah Balita (Anak)	Pendek (Anak)	Sangat Pendek (Anak)	Persentase Kasus Stunting (%)
12	1	0.864	1	0.156
13	0.626	1	0.92	0.32
14	0.045	0.067	0.06	0.28
15	0.715	0.752	0.759	0.196
16	0.241	0.113	0.154	0.067
17	0.05	0.03	0.019	0.093
18	0.136	0.315	0.345	0.502
19	0.13	0.329	0.477	0.596
20	0.109	0.166	0.211	0.324
21	0.029	0.065	0.068	0.431
22	0.077	0.114	0.127	0.302
23	0.056	0.093	0.105	0.338
24	0.007	0.01	0.007	0.236
25	0.035	0.006	0	0.013
26	0.047	0.082	0.136	0.4
27	0.177	0.223	0.279	0.262
28	0.043	0.087	0.088	0.396
29	0.02	0.018	0.019	0.178
30	0.021	0.114	0.135	1
31	0.035	0.035	0.029	0.178
32	0.018	0.034	0.028	0.324
33	0.001	0.008	0.003	0.351
34	0.009	0.019	0.012	0.324
35	0.004	0	0.002	0.133
36	0	0.009	0.01	0.502
37	0.01	0.023	0.03	0.387
38	0.001	0.005	0.003	0.276

Langkah selanjutnya ialah menentukan parameter, parameter ϵ (*epsilon*) adalah radius maksimum yang menentukan seberapa dekat dua titik dianggap bertetangga, sedangkan *minPts* adalah jumlah minimum titik dalam radius tersebut agar suatu titik menjadi *core point* dan membentuk *Cluster*:

Setelah parameter ditentukan, DBSCAN menghitung titik-titik yang berada dalam radius ϵ dari setiap titik di *dataset* menggunakan metrik seperti *Euclidean distance*. Perhitungan dimulai dari titik awal, yaitu data pertama (Aceh), dengan menghitung jarak *Euclidean* terhadap seluruh data, termasuk titik awal itu sendiri.

Tabel 2. Titik Awal

Titik Awal				
Data ke-2	0.301	0.143	0.200	0.077

Adapun implementasi perhitungan sebagai berikut:

$$d(2,1) = \sqrt{(0.121 - 0.301)^2 + (0.135 - 0.143)^2 + (0.139 - 0.200)^2} \\ \sqrt{+(0.182 - 0.077)^2} = 0.217$$

Berikut ini seluruh perhitungan dalam tabel dibawah ini:
Perhitungan jarak *Euclidean distance* Iterasi ke-1

Tabel 3. Jarak Euclidean distance

Iterasi ke-1			
Data Ke	Jarak		Dalam Radius
(2, 1)	0.217	Data ke-1	Ya
(2, 2)	0.000	Data ke-2	Ya
(2, 3)	0.250	Data ke-3	Ya
(2, 4)	0.261	Data ke-4	Ya
(2, 5)	0.318	Data ke-5	Ya
(2, 6)	0.225	Data ke-6	Ya
(2, 7)	0.356	Data ke-7	Ya

Terpenuhi

Iterasi ke-1			
Data Ke	Jarak		Dalam Radius
(2, 8)	0.193	Data ke-8	Ya
(2, 9)	0.361	Data ke-9	Ya
(2, 10)	0.354	Data ke-10	Ya
(2, 11)	0.283	Data ke-11	Ya
(2, 12)	1.286	Data ke-12	Tidak
(2, 13)	1.190	Data ke-13	Tidak
(2, 14)	0.363	Data ke-14	Ya
(2, 15)	0.933	Data ke-15	Tidak
(2, 16)	0.083	Data ke-16	Ya
(2, 17)	0.330	Data ke-17	Ya
(2, 18)	0.509	Data ke-18	Tidak
(2, 19)	0.640	Data ke-19	Tidak
(2, 20)	0.314	Data ke-20	Ya
(2, 21)	0.472	Data ke-21	Ya
(2, 22)	0.327	Data ke-22	Ya
(2, 23)	0.374	Data ke-23	Ya
(2, 24)	0.408	Data ke-24	Ya
(2, 25)	0.365	Data ke-25	Ya
(2, 26)	0.420	Data ke-26	Ya
(2, 27)	0.250	Data ke-27	Ya
(2, 28)	0.428	Data ke-28	Ya
(2, 29)	0.368	Data ke-29	Ya
(2, 30)	0.967	Data ke-30	Tidak
(2, 31)	0.347	Data ke-31	Ya
(2, 32)	0.424	Data ke-32	Ya
(2, 33)	0.466	Data ke-33	Ya
(2, 34)	0.443	Data ke-34	Ya
(2, 35)	0.388	Data ke-35	Ya
(2, 36)	0.565	Data ke-36	Ya
(2, 37)	0.462	Data ke-37	Ya
(2, 38)	0.431	Data ke-38	Ya

Demikian selanjutnya hingga iterasi ke-10.

Setelah titik baru di tentukan lakukan perhitungan jarak *Euclidean distance* sebagai berikut. Adapun implementasi perhitungan sebagai berikut:

$$d(20,1) = \sqrt{(0.121 - 0.109)^2 + (0.135 - 0.166)^2 + (0.139 - 0.211)^2} \\ \sqrt{+(0.182 - 0.324)^2} = 0.226$$

Berikut ini seluruh perhitungan dalam tabel dibawah ini:

Tabel 4. Perhitungan jarak *Euclidean distance* Iterasi ke-10

Iterasi ke-10			
Data Ke	Jarak		Dalam Radius
(20, 1)	0.163	Data ke-1	Ya
(20, 2)	0.314	Data ke-2	Ya
(20, 3)	0.097	Data ke-3	Ya
(20, 4)	0.330	Data ke-4	Ya
(20, 5)	0.348	Data ke-5	Ya
(20, 6)	0.371	Data ke-6	Ya
(20, 7)	0.347	Data ke-7	Ya
(20, 8)	0.304	Data ke-8	Ya
(20, 9)	0.359	Data ke-9	Ya
(20, 10)	0.358	Data ke-10	Ya
(20, 11)	0.365	Data ke-11	Ya
(20, 12)	1.390	Data ke-12	Tidak

Iterasi ke-10			
Data Ke	Jarak		Dalam Radius
(20, 13)	1.210	Data ke-13	Tidak
(20, 14)	0.197	Data ke-14	Ya
(20, 15)	1.014	Data ke-15	Tidak
(20, 16)	0.300	Data ke-16	Ya
(20, 17)	0.335	Data ke-17	Ya
(20, 18)	0.270	Data ke-18	Ya
(20, 19)	0.413	Data ke-19	Ya
(20, 20)	0.000	Data ke-20	Ya
(20, 21)	0.220	Data ke-21	Ya
(20, 22)	0.107	Data ke-22	Ya
(20, 23)	0.140	Data ke-23	Ya
(20, 24)	0.290	Data ke-24	Ya
(20, 25)	0.415	Data ke-25	Ya
(20, 26)	0.149	Data ke-26	Ya
(20, 27)	0.128	Data ke-27	Ya
(20, 28)	0.175	Data ke-28	Ya
(20, 29)	0.291	Data ke-29	Ya
(20, 30)	0.687	Data ke-30	Tidak
(20, 31)	0.275	Data ke-31	Ya
(20, 31)	0.239	Data ke-32	Ya
(20, 33)	0.278	Data ke-33	Ya
(20, 34)	0.267	Data ke-34	Ya
(20, 35)	0.343	Data ke-35	Ya
(20, 36)	0.326	Data ke-36	Ya
(20, 37)	0.254	Data ke-37	Ya
(20, 38)	0.287	Data ke-38	Ya

Terpenuhi

3.3. Algoritma Fuzzy C-Means

Langkah pertama dalam perhitungan manual *Fuzzy C-Means* adalah menentukan data yang akan digunakan. Pada kasus ini, data yang akan dianalisis terdiri dari 38 sampel terkait kasus stunting di Indonesia pada tahun 2024, yang akan digunakan sebagai *input* dalam proses pengelompokan menggunakan algoritma *Fuzzy C-Means*:

Tabel 5. Dataset (X_{ij})

No	Jumlah Balita (Anak)	Pendek (Anak)	Sangat Pendek (Anak)	Persentase Kasus Stunting (%)
1	0.121	0.135	0.145	0.213
2	0.301	0.143	0.212	0.071
3	0.115	0.172	0.176	0.298
4	0.112	0.046	0.047	0.049
5	0.06	0.024	0.029	0.058
6	0.184	0.038	0.048	0
7	0.027	0.019	0.007	0.111
8	0.166	0.078	0.08	0.062
9	0.024	0.011	0.006	0.084
10	0.031	0.015	0.008	0.08
11	0.11	0.023	0.033	0.009
12	1	0.864	1	0.156
13	0.626	1	0.92	0.32
14	0.045	0.067	0.06	0.28
15	0.715	0.752	0.759	0.196
16	0.241	0.113	0.154	0.067
17	0.05	0.03	0.019	0.093
18	0.136	0.315	0.345	0.502
19	0.13	0.329	0.477	0.596
20	0.109	0.166	0.211	0.324

No	Jumlah Balita (Anak)	Pendek (Anak)	Sangat Pendek (Anak)	Persentase Kasus Stunting (%)
21	0.029	0.065	0.068	0.431
22	0.077	0.114	0.127	0.302
23	0.056	0.093	0.105	0.338
24	0.007	0.01	0.007	0.236
25	0.035	0.006	0	0.013
26	0.047	0.082	0.136	0.4
27	0.177	0.223	0.279	0.262
28	0.043	0.087	0.088	0.396
29	0.02	0.018	0.019	0.178
30	0.021	0.114	0.135	1
31	0.035	0.035	0.029	0.178
32	0.018	0.034	0.028	0.324
33	0.001	0.008	0.003	0.351
34	0.009	0.019	0.012	0.324
35	0.004	0	0.002	0.133
36	0	0.009	0.01	0.502
37	0.01	0.023	0.03	0.387
38	0.001	0.005	0.003	0.276

Langkah selanjutnya ialah penentuan titik *Cluster* awal, penentuan titik awal *Cluster* dalam algoritma *Fuzzy C-Means* biasanya dilakukan dengan memilih titik-titik awal secara acak dari data yang tersedia atau berdasarkan metode tertentu, seperti memilih nilai-nilai ekstrim. Namun, ketentuan penting dalam memilih titik awal *Cluster* adalah bahwa nilai keanggotaan setiap data terhadap seluruh *Cluster* harus memenuhi syarat bahwa $C1+C2+C3=1$. Ini berarti bahwa setiap titik data memiliki derajat keanggotaan terhadap ketiga *Cluster*, dan distribusi keanggotaan tersebut harus dijaga agar selalu total 1. Dalam *Fuzzy C-Means*, titik awal ini akan digunakan untuk menghitung keanggotaan *fuzzy*, di mana jarak antara data dan pusat *Cluster* awal menentukan derajat keanggotaan, yang akan terus diperbarui selama iterasi hingga nilai keanggotaan dan pusat *Cluster* konvergen.

Tabel 6. Titik awal *Cluster* (u_{ik})

No	C1	C2	C3	Jumlah (C1+C2+C3)
1	0.2	0.35	0.45	1
2	0.25	0.4	0.35	1
3	0.15	0.4	0.45	1
4	0.3	0.35	0.35	1
5	0.1	0.4	0.5	1
6	0.2	0.3	0.5	1
7	0.25	0.45	0.3	1
8	0.15	0.35	0.5	1
9	0.3	0.4	0.3	1
10	0.2	0.3	0.5	1
11	0.15	0.45	0.4	1
12	0.1	0.35	0.55	1
13	0.25	0.4	0.35	1
14	0.2	0.45	0.35	1
15	0.1	0.4	0.5	1
16	0.3	0.3	0.4	1
17	0.2	0.4	0.4	1
18	0.25	0.35	0.4	1
19	0.15	0.4	0.45	1
20	0.1	0.3	0.6	1
21	0.2	0.4	0.4	1
22	0.15	0.35	0.5	1
23	0.1	0.45	0.45	1
24	0.3	0.35	0.35	1
25	0.15	0.4	0.45	1
26	0.2	0.35	0.45	1
27	0.1	0.45	0.45	1

No	C1	C2	C3	Jumlah (C1+C2+C3)
28	0.25	0.35	0.4	1
29	0.2	0.4	0.4	1
30	0.1	0.3	0.6	1
31	0.25	0.4	0.35	1
32	0.15	0.45	0.4	1
33	0.1	0.3	0.6	1
34	0.3	0.3	0.4	1
35	0.15	0.4	0.45	1
36	0.2	0.35	0.45	1
37	0.25	0.35	0.4	1
38	0.15	0.4	0.45	1

Langkah selanjutnya ialah menghitung pusat *Cluster* dengan rumus:

$$v_{kj} = \frac{\sum_{i=1}^n ((u_{ik})^w \times x_{ij})}{\sum_{i=1}^n (u_{ik})^w}$$

Tabel 7. Perhitungan $(u_{ik})^w$ Iterasi 1

No	$(U_{ik})^w$			
	U_{ik}^2	U_{ik}^2	U_{ik}^2	U_{ik}^2
1	0.040	0.123	0.203	0.203
2	0.063	0.160	0.123	0.123
3	0.023	0.160	0.203	0.203
4	0.090	0.123	0.123	0.123
5	0.010	0.160	0.250	0.250
6	0.040	0.090	0.250	0.250
7	0.063	0.203	0.090	0.090
8	0.023	0.123	0.250	0.250
9	0.090	0.160	0.090	0.090
10	0.040	0.090	0.250	0.250
11	0.023	0.203	0.160	0.160
12	0.010	0.123	0.303	0.303
13	0.063	0.160	0.123	0.123
14	0.040	0.203	0.123	0.123
15	0.010	0.160	0.250	0.250
16	0.090	0.090	0.160	0.160
17	0.040	0.160	0.160	0.160
18	0.063	0.123	0.160	0.160
19	0.023	0.160	0.203	0.203
20	0.010	0.090	0.360	0.360
21	0.040	0.160	0.160	0.160
22	0.023	0.123	0.250	0.250
23	0.010	0.203	0.203	0.203
24	0.090	0.123	0.123	0.123
25	0.023	0.160	0.203	0.203
26	0.040	0.123	0.203	0.203
27	0.010	0.203	0.203	0.203
28	0.063	0.123	0.160	0.160
29	0.040	0.160	0.160	0.160
30	0.010	0.090	0.360	0.360
31	0.063	0.160	0.123	0.123
32	0.023	0.203	0.160	0.160
33	0.010	0.090	0.360	0.360
34	0.090	0.090	0.160	0.160
35	0.023	0.160	0.203	0.203
36	0.040	0.123	0.203	0.203
37	0.063	0.123	0.160	0.160

(U _{ik}) ^w				
No	U _{ik} ^2	U _{ik} ^2	U _{ik} ^2	U _{ik} ^2
	38	0.023	0.160	0.203
Σ(Total)		1.530	5.433	7.423

Kemudian hasil tiap (u_{ik})^w kita kalikan terhadap data x_{ij} sebagai berikut:

Tabel 8. Perhitungan (u_{ik})^w * x_{ij} terhadap C1 Iterasi 1

Menghitung (U _{ik}) ^w * X _{ij}					
No	U _{ik} ^2 * X _{ij}				
1	0.0048	0.0054	0.0056	0.0073	0.0073
2	0.0188	0.0089	0.0125	0.0048	0.0048
3	0.0026	0.0039	0.0038	0.0054	0.0054
4	0.0101	0.0041	0.0044	0.0058	0.0058
5	0.0006	0.0002	0.0003	0.0007	0.0007
6	0.0074	0.0015	0.0019	0.0010	0.0010
7	0.0017	0.0012	0.0006	0.0061	0.0061
8	0.0037	0.0018	0.0018	0.0016	0.0016
9	0.0021	0.0010	0.0010	0.0076	0.0076
10	0.0012	0.0006	0.0005	0.0032	0.0032
11	0.0025	0.0005	0.0007	0.0009	0.0009
12	0.0100	0.0086	0.0100	0.0016	0.0016
13	0.0391	0.0625	0.0575	0.0200	0.0200
14	0.0018	0.0027	0.0024	0.0112	0.0112
15	0.0071	0.0075	0.0076	0.0020	0.0020
16	0.0216	0.0101	0.0138	0.0060	0.0060
17	0.0020	0.0012	0.0008	0.0037	0.0037
18	0.0085	0.0197	0.0216	0.0314	0.0314
19	0.0029	0.0074	0.0107	0.0134	0.0134
20	0.0011	0.0017	0.0021	0.0032	0.0032
21	0.0012	0.0026	0.0027	0.0172	0.0172
22	0.0017	0.0026	0.0028	0.0068	0.0068
23	0.0006	0.0009	0.0011	0.0034	0.0034
24	0.0006	0.0009	0.0006	0.0212	0.0212
25	0.0008	0.0001	0.0000	0.0003	0.0003
26	0.0019	0.0033	0.0055	0.0160	0.0160
27	0.0018	0.0022	0.0028	0.0026	0.0026
28	0.0027	0.0054	0.0055	0.0247	0.0247
29	0.0007	0.0013	0.0007	0.0071	0.0071
30	0.0002	0.0012	0.0014	0.0100	0.0100
31	0.0022	0.0026	0.0018	0.0111	0.0111
32	0.0005	0.0009	0.0006	0.0073	0.0073
33	0.0001	0.0001	0.0000	0.0035	0.0035
34	0.0011	0.0016	0.0011	0.0292	0.0292
35	0.0001	0.0001	0.0001	0.0030	0.0030
36	0.0004	0.0004	0.0004	0.0201	0.0201
37	0.0019	0.0013	0.0019	0.0242	0.0242
38	0.0001	0.0002	0.0001	0.0062	0.0062
Σ(Total)	0.1682	0.1782	0.1885	0.3508	0.3508

Tabel 9. Perhitungan (u_{ik})^w terhadap C2 Iterasi 1

Menghitung (U _{ik}) ^w * X _{ij}					
No	U _{ik} ^2 * X _{ij}				
1	0.0148	0.0165	0.0171	0.0223	0.0223
2	0.0482	0.0228	0.0320	0.0123	0.0123
3	0.0184	0.0275	0.0271	0.0381	0.0381
4	0.0138	0.0056	0.0060	0.0078	0.0078

Menghitung (U_{ik})^w * X_{ij}					
No	U _{ik} ^2 * X _{ij}				
5	0.0097	0.0039	0.0047	0.0110	
6	0.0166	0.0034	0.0043	0.0023	
7	0.0055	0.0038	0.0020	0.0196	
8	0.0204	0.0095	0.0096	0.0088	
9	0.0038	0.0018	0.0017	0.0134	
10	0.0028	0.0014	0.0010	0.0073	
11	0.0223	0.0046	0.0066	0.0081	
12	0.1225	0.1058	0.1225	0.0191	
13	0.1001	0.1600	0.1471	0.0512	
14	0.0092	0.0136	0.0121	0.0567	
15	0.1144	0.1204	0.1215	0.0313	
16	0.0216	0.0101	0.0138	0.0060	
17	0.0080	0.0048	0.0031	0.0149	
18	0.0167	0.0386	0.0423	0.0615	
19	0.0208	0.0526	0.0763	0.0953	
20	0.0098	0.0149	0.0190	0.0292	
21	0.0047	0.0104	0.0109	0.0690	
22	0.0094	0.0140	0.0155	0.0370	
23	0.0114	0.0189	0.0213	0.0684	
24	0.0009	0.0012	0.0009	0.0289	
25	0.0056	0.0009	0.0000	0.0021	
26	0.0057	0.0101	0.0167	0.0490	
27	0.0357	0.0452	0.0565	0.0531	
28	0.0053	0.0106	0.0108	0.0485	
29	0.0030	0.0051	0.0030	0.0284	
30	0.0021	0.0105	0.0122	0.0900	
31	0.0056	0.0066	0.0046	0.0284	
32	0.0042	0.0080	0.0058	0.0657	
33	0.0006	0.0013	0.0003	0.0316	
34	0.0011	0.0016	0.0011	0.0292	
35	0.0004	0.0007	0.0004	0.0213	
36	0.0012	0.0014	0.0012	0.0615	
37	0.0036	0.0025	0.0036	0.0474	
38	0.0004	0.0011	0.0004	0.0441	
Σ(Total)	0.7001	0.7717	0.8348	1.3200	

Langkah selanjutnya setelah didapat titik *Cluster* ialah menghitung fungsi objektif dengan rumus sebagai berikut:

$$L = \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^m (x_{ij} - v_{kj})^2 x (u_{ik})^w$$

Berdasarkan rumus diatas maka didapat hasil terlihat pada tabel dibawah ini:

Tabel 10. Fungsi Objektif Iterasi 1

Fungsi Objektif					
N	(x _{ij} - v _{kj}) ² + + (x _{ij} - v _{kj}) ² * (u _{ik}) ^w	(x _{ij} - v _{kj}) ² + + (x _{ij} - v _{kj}) ² * (u _{ik}) ^w	(x _{ij} - v _{kj}) ² + + (x _{ij} - v _{kj}) ² * (u _{ik}) ^w	Σ(Total)	
1	0.0001	0.0005	0.0020	0.0026	
2	0.0041	0.0095	0.0080	0.0216	
3	0.0001	0.0002	0.0005	0.0008	
4	0.0034	0.0064	0.0086	0.0184	
5	0.0005	0.0103	0.0209	0.0317	
6	0.0023	0.0065	0.0225	0.0314	
7	0.0029	0.0137	0.0078	0.0244	
8	0.0007	0.0050	0.0136	0.0193	

Fungsi Objektif						
N	$(x_{ij} - v_{kj})^2 + \dots + (x_{ij} - v_{kj})^2 *$	$(x_{ij} - v_{kj})^2 + \dots + (x_{ij} - v_{kj})^2 *$	$(x_{ij} - v_{kj})^2 + \dots + (x_{ij} - v_{kj})^2 *$	$(x_{ij} - v_{kj})^2 + \dots + (x_{ij} - v_{kj})^2 *$	$\Sigma(\text{Total})$	
o	$(u_{ik})^w$	$(u_{ik})^w$	$(u_{ik})^w$	$(u_{ik})^w$		
9	0.0047	0.0118	0.0085	0.0250		
10	0.0020	0.0065	0.0231	0.0316		
11	0.0012	0.0143	0.0144	0.0299		
12	0.0212	0.2454	0.5902	0.8568		
13	0.1055	0.2520	0.1865	0.5441		
14	0.0005	0.0046	0.0034	0.0085		
15	0.0118	0.1736	0.2615	0.4468		
16	0.0040	0.0040	0.0086	0.0166		
17	0.0016	0.0095	0.0124	0.0235		
18	0.0103	0.0164	0.0178	0.0444		
19	0.0069	0.0421	0.0468	0.0958		
20	0.0002	0.0010	0.0021	0.0033		
21	0.0021	0.0094	0.0088	0.0202		
22	0.0001	0.0009	0.0020	0.0031		
23	0.0002	0.0039	0.0038	0.0078		
24	0.0032	0.0066	0.0080	0.0178		
25	0.0018	0.0166	0.0260	0.0443		
26	0.0014	0.0043	0.0062	0.0119		
27	0.0004	0.0050	0.0038	0.0093		
28	0.0021	0.0046	0.0056	0.0124		
29	0.0012	0.0075	0.0097	0.0184		
30	0.0060	0.0527	0.1960	0.2547		
31	0.0014	0.0062	0.0063	0.0140		
32	0.0007	0.0090	0.0078	0.0176		
33	0.0005	0.0059	0.0252	0.0316		
34	0.0037	0.0050	0.0098	0.0186		
35	0.0011	0.0112	0.0178	0.0300		
36	0.0043	0.0146	0.0232	0.0421		
37	0.0031	0.0074	0.0098	0.0204		
38	0.0009	0.0093	0.0136	0.0238		

Demikian seterusnya hingga iterasi ke-9.

Hasil analisis *Clustering* menggunakan metode *Fuzzy C-Means* menunjukkan bahwa:

1. *Cluster* ini memiliki anggota dengan derajat keanggotaan tertinggi di C1. Data yang tergolong dalam *Cluster* 1 adalah data ke-1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 14, 16, 17, 24, 25, 29, 31, 35, dan 38., yang menunjukkan bahwa data-data ini memiliki kesamaan karakteristik yang kuat dan lebih dekat satu sama lain dibandingkan dengan data dari *Cluster* lainnya.
2. *Cluster* ini mencakup data dengan derajat keanggotaan tertinggi di C2. Data yang tergolong dalam *Cluster* 2 adalah data ke-3, 18, 19, 20, 21, 22, 23, 26, 27, 28, 30, 32, 33, 34, 36, dan 37, yang juga menunjukkan kesamaan tertentu, meskipun berbeda dari *Cluster* 1.
3. *Cluster* ini mencakup data dengan derajat keanggotaan tertinggi di C3. Data yang tergolong dalam *Cluster* 3 adalah data ke-12, 13, dan 15, yang mungkin mewakili pola atau karakteristik yang berbeda dibandingkan dengan dua *Cluster* sebelumnya.

3.4. Penerapan Model

Setelah merancang dan membangun sistem, tahap selanjutnya adalah penerapan dan evaluasi untuk menguji sejauh mana sistem memenuhi tujuan awal serta menilai kinerja dan keandalannya dalam kondisi nyata. Dalam penelitian ini, digunakan dua metode *Clustering*, yaitu DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) dan *Fuzzy C-Means* (FCM), yang diterapkan menggunakan *Python*. DBSCAN dipilih karena mampu menemukan kluster dengan bentuk tidak beraturan dan secara otomatis mengidentifikasi *outlier*, sedangkan FCM memungkinkan setiap data memiliki derajat keanggotaan pada beberapa kluster sekaligus, sehingga lebih fleksibel dalam menangkap karakteristik tumpang tindih antar kluster. Implementasi kedua metode ini memanfaatkan pustaka *Python* seperti *Scikit-learn* dan *Scikit-fuzzy*, yang memfasilitasi analisis dan visualisasi hasil dengan efisien. Evaluasi dilakukan untuk membandingkan performa kedua algoritma, mengidentifikasi

kelebihan dan kekurangannya, serta menentukan metode mana yang paling sesuai dengan karakteristik data dan kebutuhan penelitian, sehingga dapat memberikan dasar untuk optimalisasi sistem atau penerapan lebih lanjut.

1. Tampilan *Input Data*

```

1 # 1. Membaca Data
2 file_path = 'DataTasya1.xlsx'
3 data = pd.read_excel(file_path)
    
```

Gambr 7. Tampilan *Input Data*

Kode di atas berfungsi untuk membaca data dari file Excel yang disimpan di lokasi 'DataTasya1.xlsx' dan memuatnya ke dalam variabel bernama data menggunakan fungsi `pd.read_excel()` dari library pandas, di mana pd adalah alias standar untuk pandas. Fungsi ini secara khusus dirancang untuk membaca file Excel berformat .xlsx dan mengubahnya menjadi *dataframe*, yaitu struktur data tabular di pandas yang menyerupai tabel pada *spreadsheet*. *Dataframe* ini menyediakan kemampuan untuk analisis dan manipulasi data secara efisien di *Python*, seperti filter, agregasi, dan transformasi data, menjadikannya alat yang sangat berguna dalam pengolahan data.

2. Tampilan *Pre-processing Data*

```

1 # membersihkan data
2 data.columns = data.iloc[0] # Menetapkan ulang kolom
3 data = data.drop(0).reset_index(drop=True) # Menghapus baris pertama
4 data.columns = ['No', 'Provinsi', 'Jumlah Balita (Anak)', 'Pendeak (Anak)', 'Sangat Pendeak (Anak)', 'Persentase Kasus Stunting (%)']
5 data = data.drop(columns=['No', 'Provinsi']).dropna() # Menghapus kolom tidak perlu dan baris kosong
6 data = data.astype(float) # Mengonversi ke tipe numerik
7
8 # Menampilkan data untuk memastikan
9 print(data.head())
10
11 # 2. Normalisasi data
12 scaler = StandardScaler()
13 data_scaled = scaler.fit_transform(data)
    
```

Gambar 8. Tampilan *Pre-processing Data*

Kode di atas digunakan untuk membersihkan dan menyiapkan data untuk analisis. Pertama, baris pertama *dataset* digunakan untuk menetapkan nama kolom dengan mengatur ulang kolom (`data.columns = data.iloc[0]`), kemudian baris pertama dihapus, dan indeks direset. Setelah itu, kolom yang tidak diperlukan seperti 'No' dan 'Provinsi' dihapus, serta baris yang memiliki nilai kosong juga dihapus menggunakan `dropna()`. Selanjutnya, data dikonversi ke tipe numerik dengan menggunakan `astype(float)` agar dapat diproses lebih lanjut. Untuk memastikan langkah ini berjalan dengan benar, lima baris pertama dari data yang telah dibersihkan ditampilkan dengan `print(data.head())`. Pada tahap selanjutnya, normalisasi data dilakukan menggunakan objek `StandardScaler()` dari `sklearn`, yang akan menstandarkan data dengan mengubahnya ke skala yang memiliki rata-rata 0 dan standar deviasi 1, dan hasil normalisasi disimpan dalam `data_scaled`.

3. Tampilan Implementasi Metode DBSCAN

```

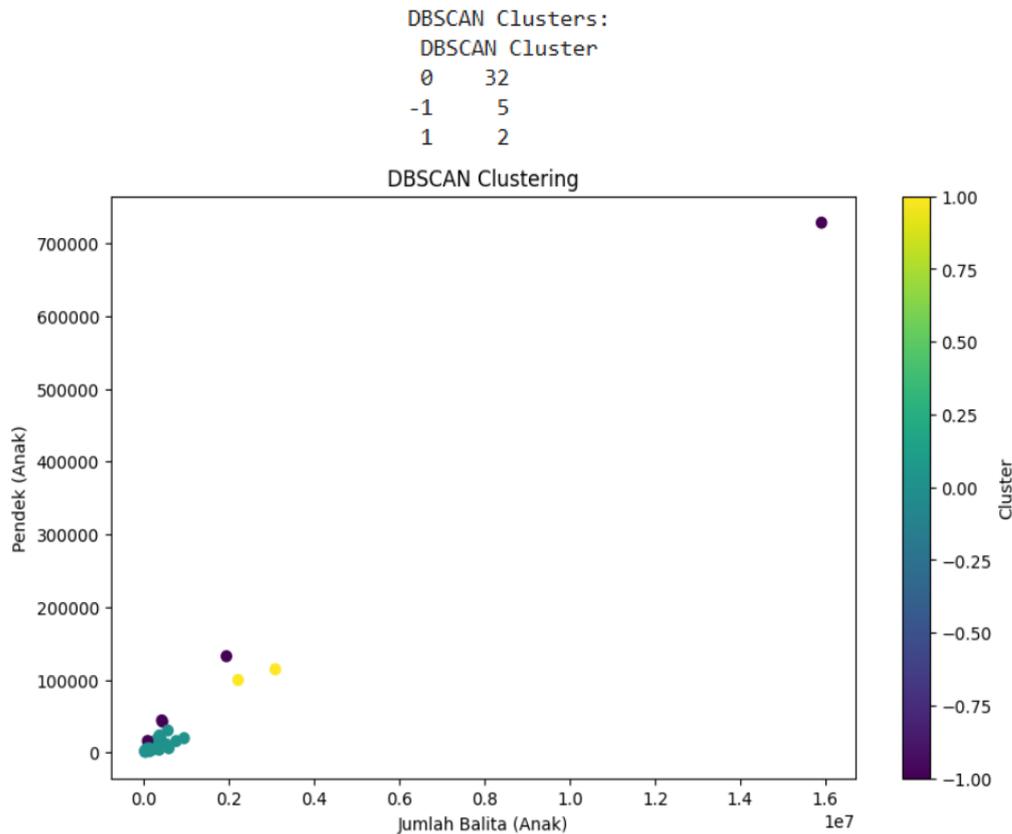
1 # 3. Implementasi DBSCAN
2 dbscan = DBSCAN(eps=0.5, min_samples=2)
3 dbscan_labels = dbscan.fit_predict(data_scaled)
    
```

Gambar 9. Tampilan Implementasi Metode DBSCAN

Kode di atas menggunakan algoritma DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) untuk melakukan *Clustering* pada data yang telah diskalakan. Pertama, objek `DBSCAN` dibuat dengan parameter `eps=0.5`, yang menunjukkan radius maksimum untuk satu titik agar dianggap tetangga, dan

``min_samples=2``, yang berarti jumlah minimum titik dalam radius tersebut agar dapat membentuk kluster. Setelah itu, algoritma DBSCAN diterapkan pada data terstandarisasi melalui metode ``fit_predict``, yang menghasilkan label kluster untuk setiap data. Hasil kluster ini kemudian ditambahkan ke dalam `dataframe`data`` sebagai kolom baru bernama `'DBSCAN Cluster'`. Terakhir, kode ini mencetak jumlah titik di setiap kluster, termasuk titik yang dianggap *noise* (biasanya ditandai dengan label -1).

4. Tampilan Hasil DBSCAN



Gambar 10. Tampilan Hasil DBSCAN

Hasil DBSCAN menunjukkan bahwa sebagian besar data terkumpul di Kluster 0 dengan 121 data, sementara kluster-kluster kecil (1 hingga 10) masing- masing hanya memiliki 2 hingga 4 data. Sebanyak 6 data teridentifikasi sebagai *outlier* (Kluster -1) karena tidak memenuhi kriteria kepadatan. Pola ini menggambarkan adanya satu kluster dominan, beberapa kluster kecil, dan sejumlah data yang menyimpang.

5. Tampilan Implementasi Metode *Fuzzy C-Means*

```

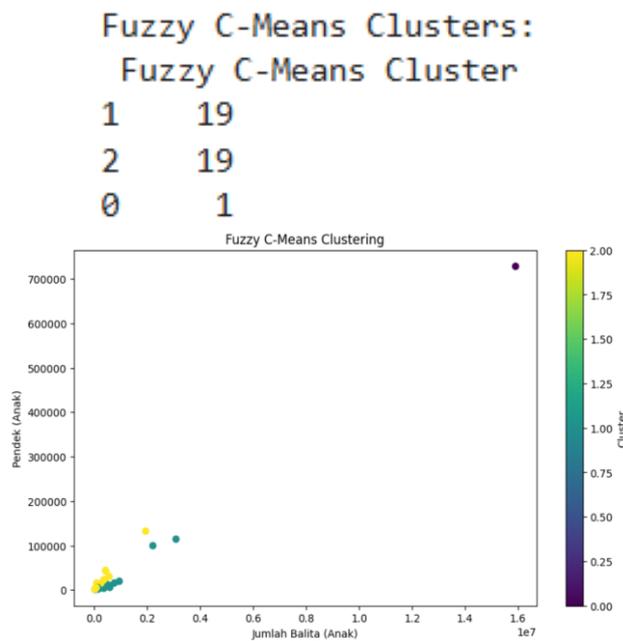
1
2 # 4. Implementasi Fuzzy C-Means
3 # Transpose data karena skfuzzy c-means bekerja dengan data berbentuk (features, samples)
4 data_transposed = np.transpose(data_scaled)
5 n_clusters = 3
6
7 # Menjalankan algoritma Fuzzy C-Means
8 cnr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(
9     data_transposed, n_clusters, 2, error=0.005, maxiter=1000, init=None)
10
11 # Menentukan hasil klustering untuk tiap data
12 fcm_labels = np.argmax(u, axis=0)
13 data['Fuzzy C-Means Cluster'] = fcm_labels
14
15 print("Fuzzy C-Means Clusters:\n", data['Fuzzy C-Means Cluster'].value_counts())

```

Gambar 11. Tampilan Implementasi Metode *Fuzzy C-Means*

Kode di atas menerapkan algoritma *Fuzzy C-Means* untuk melakukan klusterisasi pada data yang telah diskalakan. Pertama, data *ditranspos* agar sesuai dengan format yang diharapkan oleh fungsi *fuzz.Cluster.cmeans*, yaitu (fitur, sampel). Algoritma *Fuzzy C-Means* kemudian dijalankan dengan 3 kluster menggunakan parameter default lain seperti eksponen *fuzzification* 2, tingkat error 0.005, dan maksimum iterasi 1000. Algoritma ini menghasilkan beberapa *output*: *cntr* (pusat kluster), *u* (matriks keanggotaan), *u0* (nilai awal matriks keanggotaan), *d* (jarak data ke pusat kluster), *jm* (fungsi objektif di setiap iterasi), *p* (iterasi akhir), dan *fpc* (koefisien partisi *fuzzy*). Setelah itu, label kluster untuk tiap data ditentukan berdasarkan nilai keanggotaan tertinggi di matriks *u*, dan label tersebut ditambahkan ke dalam *dataset* sebagai kolom baru bernama "*Fuzzy C-Means Cluster*". Terakhir, distribusi data pada tiap kluster dicetak menggunakan *value_counts()*.

6. Tampilan Hasil *Fuzzy C-Means*



Gambar 12. Tampilan Hasil *Fuzzy C-Means*

Hasil dari klusterisasi *Fuzzy C-Means* menunjukkan bahwa data dibagi menjadi tiga kluster, dengan jumlah anggota yang berbeda di setiap kluster. Kluster 0 berisi 72 data, Kluster 2 terdiri dari 67 data, dan Kluster 1 hanya memiliki 12 data. Distribusi ini mencerminkan bagaimana algoritma *Fuzzy C-Means* mengelompokkan data berdasarkan derajat keanggotaan dari tiap data ke pusat kluster. Perbedaan jumlah anggota pada setiap kluster menunjukkan bahwa beberapa kluster mungkin lebih dominan atau lebih padat dibanding yang lain, dengan kluster 2 sebagai yang terkecil, menunjukkan data yang mungkin lebih unik atau berbeda dari yang lain dalam *dataset*.

3.5. Pengujian

Pengujian dalam penelitian ini menggunakan metode *Blackbox*, yang bertujuan untuk menilai apakah program telah memenuhi persyaratan fungsional yang telah ditentukan. Pengujian dilakukan berdasarkan spesifikasi sistem, mencakup berbagai fungsi, opsi menu, serta kompatibilitas model yang digunakan. Pendekatan ini melibatkan menjalankan program yang telah dikembangkan dan memeriksa kesesuaiannya dengan persyaratan yang telah ditetapkan sebelumnya. Untuk memperjelas hasil pengujian, disertakan tabel kinerja sistem yang menggambarkan sejauh mana program memenuhi standar dan tujuan yang diharapkan, memberikan gambaran konkret dari evaluasi yang dilakukan.

Tabel 11. Pengujian *Blackbox*

No	Skenario Kerja	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1.	Menginput <i>Dataset</i> pada <i>Jupyter Notebook</i> .	User dapat menginput <i>Dataset</i> pada <i>Jupyter Notebook</i> .	User berhasil menginput <i>Dataset</i> pada <i>Jupyter Notebook</i> .	Validasi

No	Skenario Kerja	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
2.	Sistem melakukan <i>Pre-processing</i> Data pada <i>Jupyter Notebook</i> .	Data dapat di <i>Pre-processing</i> pada <i>Jupyter Notebook</i> .	Data berhasil di <i>Pre-processing</i> pada <i>Jupyter Notebook</i> .	Validasi
3.	Implementasi algoritma DBSCAN pada data oleh <i>Jupyter Notebook</i> .	algoritma DBSCAN dapat diimplementasikan pada data oleh <i>Jupyter Notebook</i> .	algoritma DBSCAN berhasil diimplementasikan pada data oleh <i>Jupyter Notebook</i> .	Validasi
4.	Sistem melakukan <i>Clustering</i> pada data menggunakan algoritma DBSCAN pada <i>Jupyter Notebook</i> .	Sistem dapat melakukan <i>Clustering</i> pada data menggunakan algoritma DBSCAN pada <i>Jupyter Notebook</i> .	Sistem berhasil melakukan <i>Clustering</i> pada data menggunakan algoritma DBSCAN pada <i>Jupyter Notebook</i> .	Validasi
5.	Implementasi algoritma <i>Fuzzy C-Means</i> pada data oleh <i>Jupyter Notebook</i> .	algoritma <i>Fuzzy C-Means</i> dapat diimplementasikan pada data oleh <i>Jupyter Notebook</i> .	algoritma <i>Fuzzy C-Means</i> berhasil diimplementasikan pada data oleh <i>Jupyter Notebook</i> .	Validasi
6.	Sistem melakukan <i>Clustering</i> pada data menggunakan algoritma <i>Fuzzy C-Means</i> pada <i>Jupyter Notebook</i> .	Sistem dapat melakukan <i>Clustering</i> pada data menggunakan algoritma <i>Fuzzy C-Means</i> pada <i>Jupyter Notebook</i> .	Sistem berhasil melakukan <i>Clustering</i> pada data menggunakan algoritma <i>Fuzzy C-Means</i> pada <i>Jupyter Notebook</i> .	Validasi

4. DISKUSI

Berdasarkan implementasi algoritma DBSCAN dan *Fuzzy C-Means* pada data penyebaran stunting tahun 2024 yang berjumlah 38 data, diperoleh pemetaan kluster dengan pendekatan yang berbeda. Dengan parameter $\text{eps}=0.5$ dan $\text{min_samples}=2$, DBSCAN menghasilkan Kluster 0 dengan 32 data, Kluster 1 dengan 2 data, dan 5 data lainnya dianggap sebagai noise dan diberi label -1. Sementara itu, *Fuzzy C-Means*, dengan parameter eksponen *fuzzification* 2, tingkat *error* 0.005, dan maksimum iterasi 1000, membagi data menjadi tiga kluster: Kluster 1 dan Kluster 2 masing-masing berisi 19 data, serta Kluster 0 dengan hanya 1 data. Metode *Fuzzy C-Means* juga menghasilkan pusat kluster dan matriks keanggotaan, di mana label kluster ditentukan berdasarkan nilai keanggotaan tertinggi. Hasil ini menunjukkan bagaimana kedua algoritma berhasil memetakan pola penyebaran stunting, meskipun menghasilkan pembagian kluster yang berbeda sesuai dengan karakteristik dan parameter masing-masing. Hasil ini menunjukkan bahwa kedua algoritma mampu mengelompokkan data secara efektif, meskipun menggunakan pendekatan yang berbeda.

Berdasarkan penelitian yang dilakukan oleh [14] menunjukkan bahwa penggunaan metric cosine dengan fitur yang direduksi merupakan skenario terbaik diantara yang lain dengan nilai silhouette coefficient sebesar 0.88382763, nilai intracuster sebesar 0.08688583 dan nilai intercluster sebesar 0.74671096. Penelitian ini juga menunjukkan bahwa rata-rata outlier sebesar 89.96%. Dari penggunaan DBSCAN, dapat dilihat bahwa kualitas kluster sangat sensitif terhadap metric tetapi tidak sensitif terhadap kuantitas kluster dimana perubahan metric tidak menyebabkan perubahan kluster yang berarti. Di sisi lain, penggunaan DBSCAN memberikan korelasi yang positif antara epsilon dan intracuster dengan nilai sebesar 0.97669 dan juga menunjukkan korelasi yang negatif antara epsilon dan silhouette dengan nilai 0.9789. Selain itu, fitur yang direduksi memiliki kualitas kluster yang lebih baik dibandingkan yang tidak direduksi untuk semua metric.

Sedangkan pada penelitian yang dilakukan oleh [15] menunjukkan bahwa Implementasi pada penelitian ini menggunakan data mapping profil penyedia layanan faskes dengan atribut nama dati, kepemilikan, rawat inap, prolans, closed population, jam pelayanan, nilai terakhir, dan jumlah tenaga medis yang meliputi jumlah dokter, dokter gigi, perawat, bidan, apoteker, dan asisten apoteker. Klasterisasi data menggunakan metode FCM dan aplikasi Matlab dapat menghasilkan kelompok profil pelayanan faskes di BPJS KC Bandar Lampung yang sesuai dengan standar rekredensialing. Penerapan metode FCM ini menghasilkan dua kluster, dengan hasil kluster pertama sebanyak 479 dan kluster dua sebanyak 580. Dari hasil klasterisasi menggunakan metode FCM dimulai dari tahap training diperoleh hasil akurasi menghasilkan nilai PCI 0.50002 dan PEI 0.99998 yang berarti tingkat akurasi dari nilai keanggotaan dari kluster cukup baik.

5. KESIMPULAN

Berdasarkan penelitian yang sudah dilakukan, penulis menyimpulkan sebagai berikut:

1. Implementasi algoritma DBSCAN dan *Fuzzy C-Means* berhasil diterapkan untuk menganalisis 38 data stunting di Indonesia pada tahun 2024, dengan hasil yang menunjukkan pemetaan data ke dalam kluster berdasarkan karakteristik dan kepadatan data. Algoritma DBSCAN dengan parameter $\epsilon=0.5$ dan $MinPts=2$ berhasil membentuk dua kluster utama, di mana *Cluster 1* mencakup data ke-1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 36, 37, dan 38, serta *Cluster 2* yang terdiri dari data ke-12, 13, dan 15, sementara data ke-30 dianggap sebagai noise. Di sisi lain, algoritma *Fuzzy C-Means* membagi data ke dalam tiga kluster dengan pembobotan $w=2$, iterasi maksimum 10, dan *error* terkecil 0.05. *Cluster 1* meliputi data ke-1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 14, 16, 17, 24, 25, 29, 31, 35, dan 38, menunjukkan kesamaan karakteristik yang kuat. *Cluster 2* mencakup data ke-3, 18, 19, 20, 21, 22, 23, 26, 27, 28, 30, 32, 33, 34, 36, dan 37, dengan pola tertentu yang berbeda dari *Cluster 1*. Sementara itu, *Cluster 3* mencakup data ke-12, 13, dan 15, yang memiliki karakteristik berbeda dibandingkan dua kluster lainnya. Hasil ini menunjukkan bahwa kedua algoritma mampu mengelompokkan data secara efektif, meskipun menggunakan pendekatan yang berbeda.
2. Berdasarkan implementasi algoritma DBSCAN dan *Fuzzy C-Means* pada data penyebaran stunting tahun 2024 yang berjumlah 38 data, diperoleh pemetaan kluster dengan pendekatan yang berbeda. Dengan parameter $\epsilon=0.5$ dan $min_samples=2$, DBSCAN menghasilkan Kluster 0 dengan 32 data, Kluster 1 dengan 2 data, dan 5 data lainnya dianggap sebagai noise dan diberi label -1. Sementara itu, *Fuzzy C-Means*, dengan parameter eksponen *fuzzification 2*, tingkat *error 0.005*, dan maksimum iterasi 1000, membagi data menjadi tiga kluster: Kluster 1 dan Kluster 2 masing-masing berisi 19 data, serta Kluster 0 dengan hanya 1 data. Metode *Fuzzy C-Means* juga menghasilkan pusat kluster dan matriks keanggotaan, di mana label kluster ditentukan berdasarkan nilai keanggotaan tertinggi. Hasil ini menunjukkan bagaimana kedua algoritma berhasil memetakan pola penyebaran stunting, meskipun menghasilkan pembagian kluster yang berbeda sesuai dengan karakteristik dan parameter masing-masing.

DAFTAR PUSTAKA

- [1] Saputri, R.A. 2019. Upaya pemerintah daerah dalam penanggulangan stunting di provinsi kepulauan bangka Belitung. pp. 152–168.
- [2] Wati, S. K., Kusyanti, A., & Fitriyah, E. T. (2021). Pengaruh faktor ibu (pengetahuan ibu, pemberian ASI-eksklusif & MP-ASI) terhadap kejadian stunting pada anak. *Journal of Health Science Community*, 2(1), 40-52.
- [3] Hisbullah, R., & Hasibuan, M. S. (2023). Pendekatan Bayes-HDSS dalam Menentukan Status Pantauan Gizi Balita. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 10(5), 1071-1082.
- [4] Ulinnuha, N. (2020). Provincial Clustering in Indonesia Based on Plantation Production Using Fuzzy C-Means. *ITSMART: Jurnal Teknologi dan Informasi*, 9(1), 8-12.
- [5] 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.
- [6] Wulandari, L., & Yogantara, B. O. (2022). Algorithm analysis of K-means and fuzzy C-means for clustering countries based on economy and health. *Faktor Exacta*, 15(2).
- [7] Koto, S. Z., Marsono, M., & Syahra, Y. (2019). Analisa Data Mining Untuk Pengelompokan Pemegang Sertipikat Hak Atas Tanah dengan Algoritma K-Means Clustering di Kota Medan. *Jurnal Cyber Tech*, 2(11).
- [8] Christyanti, R. D., Sulaiman, D., Utomo, A. P., & Ayyub, M. (2022). Implementation of Fuzzy C-Means in Clustering Stunting Prone Areas. *International Journal of Natural Science and Engineering*, 6(3), 110-121.
- [9] Pane, S. F., & Saputra, Y. A. (2020). *Big Data: Classification Behavior Menggunakan Python* (Vol. 1). Kreatif.
- [10] Damayanti, D. K. D., & Jakfar, M. (2023). Klasifikasi Status Stunting Balita Menggunakan Algoritma Fuzzy C-Means (Studi Kasus Posyandu RW 01 Kelurahan Jeparo Surabaya). *MATHunesa: Jurnal Ilmiah Matematika*, 11(03), 533-542.
- [11] Saputri, R. A. (2019). Upaya pemerintah daerah dalam penanggulangan stunting di Provinsi Kepulauan Bangka Belitung. *Jdp (Jurnal Dinamika Pemerintahan)*, 2(2), 152-168.
- [12] Pimentel, J. F., Murta, L., Braganholo, V., & Freire, J. (2021). Understanding and improving the quality and reproducibility of Jupyter notebooks. *Empirical Software Engineering*, 26(4), 65.

- [13] Asyrofi, R. R., & Asyrofi, R. (2023). Implementasi Aplikasi Jupyter Notebook Sebagai Analisis Kreteria Plagiasi Dengan Teknik Simantik. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 8(2), 627-637.
- [14] Khan, Z. V. V., Alamsyah, D., & Widhiarso, W. (2022). Klasterisasi Topik Skripsi Informatika dengan Metode DBSCAN. *Jurnal Algoritme*, 3(1), 82-90.
- [15] Herlinda, V., Darwis, D., & Dartono, D. (2021). Analisis Clustering Untuk Recredesialing Fasilitas Kesehatan Menggunakan Metode Fuzzy C-Means. *Jurnal Teknologi Dan Sistem Informasi*, 2(2), 94-99.