BAB II
TINJAUAN PUSTAKA

2.1 *State of The Art*

Pada penelitian sebelumnya pada tahun 2023 oleh Aditia Yudhistira dkk., terkait Pengelompokan Data Nilai Siswa Menggunakan Metode K-Means *Clustering*, hasil pengujian menunjukkan bahwa penggunaan 3 *cluster* memberikan hasil yang optimal dalam mengelompokkan data siswa berdasarkan nilai akademik, nilai sikap, dan nilai disiplin. Pengujian dilakukan menggunakan metode *Elbow* dan *Silhouette*, di mana hasil pengujian menunjukkan nilai *Silhouette* sebesar 0,489 yang mendekati nilai optimal [4]. Lalu pada penelitian di tahun 2022 oleh Sarbaini dkk., terkait *Cluster Analysis* Menggunakan Algoritma Fuzzy K-Means Untuk Tingkat Pengangguran Diprovinsi Riau berdasarkan indikator tingkat pengangguran menggunakan metode Fuzzy K-Means. Pengujiannya dilakukan dengan mengukur performa *clustering* melalui indeks performa Fuzzy K-Means. Hasil pengujian menunjukkan bahwa metode Fuzzy K-Means efektif dalam mengelompokkan kabupaten/kota dengan karakteristik pengangguran yang serupa, memungkinkan analisis lebih mendalam tentang pola pengangguran di wilayah tersebut [5]. Selanjutnya pada penelitian sebelumnya di tahun 2020 oleh Tonny Tendean dkk., terkait Analisis *Cluster* Provinsi Indonesia Berdasarkan Produksi Bahan Pangan Menggunakan Algoritma K-Means. Pengujian dilakukan menggunakan algoritma K-Means, menghasilkan 3 *cluster* utama, yaitu *cluster* dengan tingkat produksi bahan pangan tinggi, sedang, dan rendah. Hasil pengujian menunjukkan bahwa provinsi Jawa Barat, Jawa Tengah, dan Jawa Timur termasuk dalam *cluster* dengan produksi bahan pangan tinggi, sementara 27 provinsi lainnya masuk dalam *cluster* dengan produksi rendah [6].

2.2 *Stunting*

Stunting merupakan bentuk kegagalan pertumbuhan (*growth faltering*) akibat ketidak cukupan nutrisi yang berlangsung mulai dari kehamilan sampai usia 24 bulan. Keadaan ini diperparah dengan tidak terimbanginya kejar tumbuh (*catch up growth*) yang memadai. Pada laporan yang dirilis oleh *United Nations Children's Fund* (UNICEF) pada tahun 2021, di perkirakan ada 149,2 juta anak yang menglami Stunting.

Stunting dapat menyebabkan peningkatan risiko terkena penyakit, berdampak buruk terhadap kebugaran fisik, dan gangguan perkembangan. Kesehatan dan asupan gizi merupakan salah satu kebutuhan yang tidak boleh diabaikan dan hendaknya dimulai pada masa *golden age* yaitu dimulai dari masa awal kehamilan hingga anak berusia 2 tahun. Masa balita kerap rawan mengalami berbagai penyakit dan masalah gizi, oleh karena itu pemenuhan terhadap asupan gizi harus tetap diperhatikan [7].

Selain faktor gizi yang dimiliki anak, faktor gizi yang dimiliki oleh ibu juga dapat memengaruhi kemungkinan stunting. seperti keadaan gizi ibu yang buruk selama kehamilan, ukuran tubuh ibu yang lebih pendek, dan kurangnya perhatian pada anak. Faktor lain yang menyebabkan stunting adalah kondisi ekonomi, pekerjaan, mata pencaharian keluarga, kehamilan remaja, jarak kelahiran anak yang pendek, dan infeksi pada balita seperti diare. Anak-anak mengalami kesulitan dalam pertumbuhan karena kurangnya asupan makanan dan penyakit infeksi berulang. Oleh karena itu keadaan ini menyebabkan anak mengalamai gangguan pertumbuhan yang akhirnya berpeluang terjadinya *stunted* [8].

2.3 *Machine Learning*

Machine Learning (ML) adalah bagian dari *Artificial intelligence (AI)* yang berfokus pada pembelajaran data, dan pengembangan sistem yang mampu belajar secara mandiri tanpa harus berulang kali diprogram manusia. ML memungkinkan komputer untuk menemukan pola berdasarkan data yang telah diproses. ML membutuhkan data yang valid ketika proses *training* sebelum digunakan ketika *testing* untuk hasil *output* yang optimal [9]. Ada beberapa teknik yang dimiliki oleh ML, yaitu *Supervised Learning* dan *Unsupervised Learning*.

1. *Supervised Learning*
   Supervised learning adalah bidang interdisipliner yang luas yang dibangun atas konsep dari ilmu komputer, statistik, ilmu kognitif, teknik, teori pengoptimalan dan banyak disiplin matematika dan sains [10]. Algoritma *supervised learning* bergantung pada data input berlabel untuk mempelajari fungsi yang menghasilkan output yang sesuai ketika diberi data baru tanpa label. Algoritma knn mengasumsikan bahwa hal serupa ada dalam jarak dekat. Dengan kata lain, hal-hal serupa dekat satu sama lain [11].

2. *Unsupervised Learning*
   Algoritma *unsupervised learning* adalah salah satu tipe algoritma ML yang

digunakan untuk menarik kesimpulan dari dataset [12]. *Unsupervised learning* bekerja dengan mengelompokkan data berdasakan kedekatannya atau disebut dengan *clustering*. Metode ini bekerja dengan menganalisis data yang tidak berlabel untuk menemukan pola tersembunyi dan menentukan korelasinya [13].

2.4 *Clustering*

Clustering merupakan teknik mengelompokkan data-data (objek) ke dalam beberapa *cluster* atau kelompok, lalu objek yang serupa disatukan ke dalam *cluster* yang sama, sedangkan yang berbeda harus menjadi bagian dari *cluster* yang berbeda. Tujuan utama dari *clustering* adalah untuk memaksimalkan kumpulan data yang memiliki sifat atau komposisi yang seragam atau konsisten dan data yang memiliki sifat yang beragam dan tidak seragam di seluruh bagian *cluster* yang berbeda [14].

Pada teknik *clustering* targetnya adalah untuk kasus pendistribusian ke dalam suatu kelompok, hingga derajat tingkat keterhubungan antar anggota *cluster* yang sama adalah kuat dan lemah antara anggota *cluster* yang berbeda [15]. Teknik *Clustering* digunakan untuk mengelompokkan kecamatan berdasarkan jumlah kasus stunting. Dengan ini, kecamatan yang memiliki kesamaan dalam hal jumlah anak dengan pertumbuhan merah (indikator stunting) dikelompokkan bersama. Ini membantu mengidentifikasi pola dalam data yang mungkin tidak terlihat secara langsung.

2.5 *K-Means*

K-Means merupakan algortima untuk menentuan jumlah kelompok dengan mendefinisikan nilai *centroid* awalnya. Algoritma K-Means menggunakan proses secara berulang-ulang untuk mendapatkan basis data *cluster*. Dalam proses algoritma K-Means membutuhkan jumlah *cluster* awal yang diinginkan sebagai masukan dan menghasilkan jumlah *cluster* akhir sebagai *output*. Jika algoritma diperlukan untuk menghasilkan *cluster K* maka akan ada *K* awal dan *K* akhir. Metode K-Means akan memilih pola *K* sebagai titik awal *centroid* secara acak. Jumlah iterasi untuk mencapai *cluster centroid* akan dipengaruhi oleh calon *cluster centroid* awal secara acak dimana jika posisi *centroid* baru tidak berubah. Nilai *K* yang dipilih menjadi pusat awal, akan dihitung dengan menggunakan rumus *Euclidean Distance* yaitu mencari jarak terdekat antara titik *centroid* dengan data/objek. Data yang memiliki jarak terdekat dengan *centroid* akan membentuk sebuah *cluster*. Berikut merupakan tahapan algoritma K-Means [16].

1. Tentukan *K* sebagai jumlah *cluster* yang akan dibentuk.
2. Tentukan *K Centroid* (titik pusat *cluster*) awal secara acak.
3. Hitung jarak setiap objek ke masing-masing *centroid* dari masing-masing *cluster*. Untuk menghitung jarak antara objek dengan *centroid* dapat menggunakan *Euclidian Distance*.

$$d(x, y) = ||x - y|| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2, i = 1, 2, 3, \dots, n} \qquad (2.3)$$

dimana:
x_1 = objek x ke$-i$

y_i = daya y ke$-i$

n = banyaknya objek

Namun, karena hanya menggunakan satu fitur yaitu jum- lah anak pertumbuhan merah atau jumlah anak yang mengalami stunting saja, maka ini dianggap sebagai satu dimensi (yaitu, ketika n=1), rumus ini disederhanakan menjadi:

$$d(x_i, \mu_j) = |x_i - \mu_j| \qquad (2.2)$$

Dimana:

x_i = titik data ke-i dalam dataset.

μ_j = centroid dari cluster ke-j atau nilai rata-rata dari semua titik data dalam cluster tersebut.

4. Alokasikan masing-masing objek ke dalam *centroid* yang paling dekat
5. Lakukan iterasi, kemudian tentukan posisi *centroid* baru.

2.6 *K-Nearest Neighbors* (KNN)

Algoritma KNN adalah algoritma ML yang sederhana, yang biasa digunakan untuk klasifikasi. KNN juga dapat digunakan untuk *clustering*, dengan mengelompokkan data ke dalam *cluster* berdasarkan kedekatan atau kemiripan data. Dengan cara membentuk *cluster* berdasarkan jarak rata-rata ke tetangga terdekat (*distances*

dari tetangga ke-4 hingga ke-10), yang kemudian diinterpretasikan menjadi beberapa kelompok. KNN digunakan untuk tugas *clustering* seperti pengelompokan kecamatan berdasarkan jumlah anak yang mengalami stunting di setiap *cluster*nya. Cara kerja KNN klasifikasi adalah dengan mengklasifikasikan objek berdasarkan mayoritas tetangga terdekatnya. Nilai *K* merupakan jumlah tetangga terdekat yang akan diolah saat melakukan klasifikasi. Berbeda dengan KNN *clustering*, dimana KNN tidak mengklasifikasikan data ke dalam kelas tertentu, melainkan mengelompokkan data ke dalam *cluster* berdasarkan jarak antara titik data. Dengan menggunakan rata-rata jarak dari tetangga terdekat untuk menentukan *cluster* di mana data akan ditempatkan. Dalam KNN klasifikasi, KNN bergantung pada data *input* berlabel untuk mempelajari fungsi dan menghasilkan *output* yang sesuai ketika diberi data baru tanpa label. Namun dalam KNN *clustering*, dengan menggunakan kuantil dari jarak, KNN dapat mengelompokkan data ke dalam *cluster* yang kemudian diberi label. Berikut merupakan tahapan algoritma KNN Clustering:

1. Menentukan kedekatan antara titik-titik data dalam ruang fitur menggunakan *Euclidian Distance*.

   $$d(x,y) = \left||x-y|\right| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2, i = 1,2,3,\dots,n} \qquad (2.3)$$

   Dalam KNN, algoritma menghitung jarak antara satu data point dengan data point lainnya untuk menentukan tetangga terdekatnya, namun karena hanya menggunakan satu fitur, rumus Euclidean distance hanya berupa selisih absolut antara dua nilai tersebut. Berikut merupakan rumus euclidian distance yang telah disederhanakan:

   $$d(x_i, x_j) = |x_i - x_j| \qquad (2.4)$$

   Dimana:

   x_i = adalah titik data untuk mencari tetangga terdekatnya.

   x_j = adalah salah satu ”tetangga” yang dibandingkan jaraknya dengan Xi.

2. Setelah mendapatkan jarak ke tetangga terdekat, kemudian menghitung jarak rata-rata dari tetangga ke-4 hingga ke-10 untuk setiap titik data.

   $$distances_{4\ to\ 10} = \frac{1}{6}\sum_{i=4}^{0} distances_i \qquad (2.3)$$

3. Hitung batas kuantil untuk pengelompokan. Untuk menghitung kuantil ke-p dari data:

   - Mengurutkan data dengan urutan menaik.
   - Hitung Indeks Kuantil:

   $$k = \lceil p \times (N - 1) \rceil \tag{2.6}$$

   di mana p adalah persentase kuantil (misalnya, 0.33 untuk kuantil ke-33 dan 0.66 untuk kuantil ke 66), dan N adalah jumlah total data.

   - Ambil Nilai Kuantil:

   $$Q_p = \text{sorted\_data}[k] \tag{2.7}$$

4. Gunakan batas kuantil yang telah dihitung untuk mengelompokkan data ke dalam cluster yang sesuai:

   $$\text{cluster} = \text{np.digitize}(\text{distances}_{4 \text{ to } 10}, \text{bins} = \text{quantiles}) \tag{2.8}$$

5. setelah mengelompokan data kedalam setiap cluster, kemudian setiap cluster tersebut diurutkan terlebih dahulu. Ini dilakukan untuk memastikan bahwa urutan setiap cluster sesuai dengan rata-rata nilai dalam masing-masing cluster. Jika tidak diurutkan, cluster dengan nilai yang lebih rendah atau lebih tinggi bisa memiliki label yang tidak sesuai

6. Setelah setiap cluster diurutkan, barulah menetapkan label sesuai dengan kategori yang diinginkan (’Rendah’, ’Sedang’, ’Tinggi’). Ini memastikan bahwa label diberikan sesuai dengan urutan yang benar.

2.7 Perbandingan K-Means dengan KNN

Dalam konteks pengelompokan data (*clustering*) cara kerja dari kedua metode yaitu K-Means dan KNN memiliki perbedaan yang mendasar dalam pengelompokan data. Perbedaan mendasar antara kedua algoritma ini dalam hal prinsip kerja, proses iterasi, dan penentuan jumlah *cluster* memberikan keuntungan dan tantangan yang berbeda tergantung pada jenis data yang dihadapi. Berikut ini adalah tabel 2.1 yang merangkum perbandingan cara kerja antara K-Means dan KNN dalam konteks *clustering*:

Tabel 2.1. Perbandingan Cara Kerja K-Means dan KNN dalam Pengklasteran Data

Aspek	**K-Means *Clustering***	**KNN *Clustering***
Tujuan	Mengelompokkan data menjadi K kluster berdasarkan kesamaan karakteristik menggunakan centroid yang dioptimalkan iteratif.	Mengelompokkan data berdasarkan jarak rata-rata ke tetangga terdekat (4 hingga 10 tetangga).
Prinsip Kerja	Menentukan K pusat *cluster* (centroid) dan meminimalkan jarak antar data ke centroid tersebut secara iteratif.	Menghitung jarak antara data dan tetangga terdekat, kemudian menentukan *cluster* berdasarkan distribusi jarak dengan pendekatan kuantil.
Proses Inisialisasi	Memulai dengan memilih K centroid secara acak, kemudian mengelompokkan data ke centroid terdekat.	Tidak ada inisialisasi *cluster* secara langsung, tetapi data diurutkan berdasarkan jarak tetangga terdekat dan dibagi menjadi beberapa *cluster*.
Proses Iterasi	Secara iteratif memperbarui posisi centroid berdasarkan rata-rata posisi data dalam kluster sampai konvergen.	Tidak ada iterasi eksplisit, *clustering* dilakukan sekali setelah menghitung jarak rata-rata ke tetangga terdekat.
Penentuan Jumlah *Cluster*	Ditetapkan sebelumnya (K yang harus dipilih oleh pengguna).	Jumlah *cluster* ditentukan berdasarkan pembagian jarak tetangga menjadi kuantil tertentu.
Evaluasi	Dapat dievaluasi menggunakan metrik seperti *Silhouette Score*, inertia, atau Davies-Bouldin Index.	Evaluasi dapat dilakukan dengan metrik yang sama seperti *Silhouette Score*, tetapi bergantung pada pemilihan tetangga dan jarak rata-rata.

2.8 *Silhouette Score*

Silhouette score adalah proses evaluasi yang digunakan untuk mengukur seberapa baik suatu algoritma *clustering* mampu memisahkan data menjadi kelompok-

kelompok yang berbeda. Metode ini melibatkan perhitungan *silhouette score* untuk setiap titik data, yang mencerminkan seberapa baik titik tersebut cocok dengan kelompoknya sendiri dibandingkan dengan kelompok lainnya.

Nilai *silhouette score* yang tinggi menandakan bahwa *clustering* tersebut baik, sementara nilai yang rendah menunjukkan bahwa ada penyebaran yang tidak sesuai dalam satu atau lebih kelompok. Oleh karena itu, pengujian *silhouette score* dapat membantu dalam mengevaluasi performa algoritma *clustering* yang digunakan [17]. *Silhouette score* dihitung untuk setiap titik data dengan membandingkan jarak rata-rata *intra-cluster* (a) dengan jarak antar *cluster* (b). Pengujian menggunakan *Silhouette Score* dapat menggunakan rumus:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2.9}$$

dimana:

- $a(i)$ adalah jarak rata-rata antara titik data *i* dan semua titik data lain di dalam cluster yang sama.
- $b(i)$ adalah jarak rata-rata antara titik data i dan semua titik data di cluster tetangga terdekat.

Nilai *silhouette score* berkisar antara -1 hingga 1. Nilai positif menunjukkan bahwa titik data tersebut lebih cocok dengan kelompoknya sendiri, nilai 0 menunjukkan bahwa titik data tersebut berada di batas antara dua *cluster*, dan nilai negatif menunjukkan bahwa titik data tersebut lebih cocok dengan *cluster* lain [18]. 0.7 sampai 1.0 merupakan hasil yang sangat baik seperti *cluster* sangat jelas dan objek-objek sangat konsisten dalam kelompoknya. 0.25 sampai 0.5 merupakan hasil yang cukup baik, tapi mungkin masih ada tumpang tindih antar *cluster*. Kurang dari 0.25 merupakan hasil yang menunjukkan klasterisasi yang kurang baik atau terlalu banyak tumpang tindih antar klasternya.

2.9 *Datamining*

Datamining merupakan suatu cara untuk melakukan analisis terhadap data yang disajikan dalam *database*. *Datamining* dapat diterapkan untuk mengetahui pola data yang memiliki karakteristik masing-masing yang dapat memberikan informasi penting dari data tersebut. Pada penerapan *datamining*, terkadang terdapat dua dataset yang memiliki dimensi yang berbeda. Seperti dataset dimensi yang

tinggi dan rendah. Dataset dengan dimensi yang tinggi memiliki jumlah *attribu- te* yang banyak dan dapat berpengaruh terhadap proses penerapan teknik *datamining*, Seperti beberapa *attribute feature* yang tidak memiliki relevansi dengan *attribute class*, Sehingga berdampak buruk terhadap algoritma yang digunakan. Sedangkan Dataset dengan dimensi rendah memiliki jumlah fitur yang lebih sedikit dan memiliki relevansi dengan *attribute class*, sehingga lebih mudah untuk dianalisis dan dipahami [19]. Dalam penelitian ini dataset yang digunakan termasuk kedalam dataset berdimensi rendah, karena hanya melibatkan satu fitur utama (jumlah_anak_pertumbuhan_merah). Dalam konteks ini, analisis menjadi lebih langsung dan tidak memerlukan teknik khusus untuk mengatasi masalah yang sering muncul pada dataset berdimensi tinggi tersebut.

2.10 *Python*

Python adalah sebuah bahasa pemrograman berbasis objek yang dapat berinteraksi secara langsung dengan suatu sistem. *Python* mendukung berbagai model pemrograman, termasuk pemrograman dengan langkah-langkah yang dieksekusi secara berurutan. Bahasa pemrograman Python dirancang khusus untuk membuat program dengan waktu yang sesingkat mungkin, kemudahan pengembangan, dan beroperasi secara baik dengan sistem.

Python dapat digunakan untuk membuat aplikasi mandiri atau pemrograman. Dengan struktur yang rapi dan fokus pada kemudahan pemahaman, Python memungkinkan programmer untuk menulis kode tingkat tinggi yang mudah dipahami. Dengan banyaknya pengembangan pustaka seperti NumPy, Pandas, dan TensorFlow, Python menjadi bahasa yang mudah digunakan untuk mengembangkan aplikasi dengan cepat dan efektif. [20].