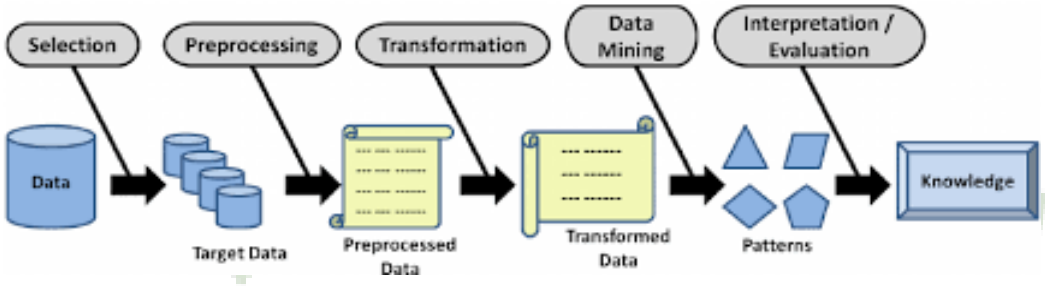
BAB II
TINJAUAN PUSTAKA

2.1 Data Mining

Data mining dikenal sejak tahun 1990-an, ketika adanya suatu pekerjaan yang memanfaatkan data menjadi suatu hal yang lebih penting dalam berbagai bidang, seperti marketing dan bisnis, sains dan teknik, serta seni dan hiburan. Sebagian ahli menyatakan bahwa data mining merupakan suatu langkah untuk menganalisis pengetahuan dalam basis data atau biasa disebut *Knowledge Discovery in Database* (KDD).

Knowledge Discovery In Database (KDD) adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menentukan keteraturan, pola atau hubungan dalam sebuah set data yang berukuran besar. (Mardi 2017)

Keluaran dari data mining banyak digunakan untuk pengambilan keputusan dimasa depan. Gambaran dari proses KDD terlihat seperti gambar berikut:



Gambar 2. 1 Skema Knowledge Discovery in Database (KDD) (Febianto & Palasara,2019)

Dari gambar di atas terlihat bahwa proses KDD terdiri dari:

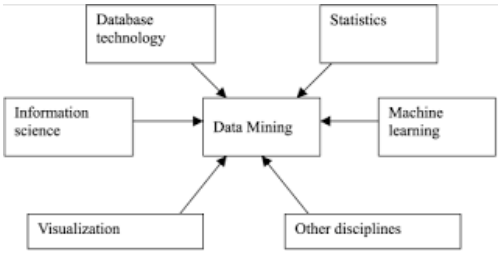
1) Pemahaman data (Data Understanding), yaitu proses memahami data berdasarkan kebutuhan data (Data Requirement). Proses ini meliputi pengumpulan data (initial data collection) dan pendeskripsian data (data decription).

2) Data transformation, yaitu proses mengkonversikan data ke dalam format lain yang sesuai dengan kebutuhan analisa.
3) Data preparation, yaitu preprocessing yang terdiri dari Penyeleksian data (Data Selection) dan pembersihan (cleaning) data, pada proses ini dilakukan pemilihan data yang disesuaikan dengan kebutuhan dan pembersihan data dari data-data yang sifatnya redundansi atau data dengan type data yang salah.
4) Modeling data mining, yaitu proses untuk memperoleh pola dan karakteristik data, dalam fase ini digunakan metode *clustering* yang tujuannya adalah untuk mengelompokkan data kemiskinan berdasarkan dengan karakteristik yang sama ke suatu wilayah dengan karakteristik yang berbeda ke wilayah yang lain. Pada tahapan *clustering* ini pengelompokan data dikelompokkan berdasarkan pengelompokan data berdasarkan wilayah tempat tinggal penduduk miskin dan indikator- indikator kemiskinan.
5) Interpretation/Evaluation, melakukan interpretasi dan evaluasi terhadap masalah yang dihadapi berdasarkan data yang di analisa.

Data mining merupakan proses untuk menemukan pola data dan pengetahuan yang menarik dari kumpulan data yang sangat besar. Sumber data dapat mencakup *database*, data *warehouse*, web, *repository*, atau data yang dialirkan ke dalam sistem dinamis.

Data mining, secara sederhana merupakan suatu langkah ekstraksi untuk mendapatkan informasi penting yang sifatnya implisit dan belum diketahui. Selain itu, data mining mempunyai hubungan dengan berbagai bidang diantaranya statistik, *machine learning* (pembelajaran mesin), *pattern recognition*, *computing algorithms, database technology, dan high performance computing*.

Diagram hubungan data mining disajikan pada gambar berikut:



Gambar 2. 2 Diagram Hubungan Data Mining (K. C. Gouda, 2018)

Secara skematis, Gorunescu (2018) membagi langkah proses pelaksanaan Data Mining dalam tiga aktivitas yaitu:

1. Eksplorasi Data, terdiri dari aktivitas pembersihan data, transformasi data, pengurangan dimensi, pemilihan ciri, dan lain-lain.
2. Model dan Pengujian Validitas Model, merupakan pemilihan terhadap model-model yang sudah dikembangkan yang cocok dengan kasus yang dihadapi. Dengan kata lain, dilakukan pemilihan model secara kompetitif.
3. Penerapan model dengan data baru untuk menghasilkan perkiraan dari kasus yang ada. Tahap ini merupakan tahap yang menentukan apakah model yang telah dibangun dapat menjawab permasalahan yang dihadapi.

Menurut (Herlawati dan Handayanto 2020), Aplikasi yang menggunakan Data Mining bermaksud menyelesaikan permasalahan dengan membangun model berdasarkan data yang sudah digali untuk diterapkan terhadap data yang lain. Secara umum ada dua jenis tipologi aplikasi Data Mining:

1. Metode Prediksi, yang bermaksud memprediksi nilai yang akan datang berdasarkan data-data yang telah ada variabelnya seperti klasifikasi, regresi, detikasi anomali, dan lain-lain.
2. Metode Deskriptif, yang bermaksud membantu user agar mudah melihat pola-pola yang berasal dari data yang ada.

Berdasarkan fungsionalitasnya, tugas-tugas *data mining* biasa dikelompokkan ke dalam enam kelompok berikut ini:

1. Klasifikasi (*classification*): men-generalisasi struktur yang diketahui untuk diaplikasikan pada data-data baru.
2. Klasterisasi (*clustering*): mengelompokkan data, yang diketahui label sankelasnya, ke dalam sejumlah kelompok tertentu sesuai dengan ukuran kemiripannya.
3. Regresi (*regression*): menemukan suatu fungsi yang memodelkan data dengan galat (kesalahan prediksi) seminimal mungkin.
4. Deteksi anomaly (*anomaly detection*): mengidentifikasi data yang tidak umum, bias berupa *outlier* (pencilan), perubahan atau deviasi, yang mungkin gat penting dan perlu investigasi lebih lanjut.

5. Pembelajaran aturan asosiasi (*association rule learning*) atau pemodelan kebergantungan (*dependency modeling*): mencari relasi antar variable.
6. Perangkuman (*summarization*): menyediakan representasi data yang lebih sederhana, meliputi visualisasi dan pembuatan laporan.

2.2 Analisis Klaster (Cluster Analysis)

Klasterisasi atau *clustering* adalah proses pengelompokan himpunan data ke dalam beberapa grup atau klaster sedemikian hingga objek-objek dalam suatu klaster memiliki kemiripan yang tinggi, namun sangat berbeda (memiliki Ketidakmiripan yang tinggi) dengan objek-objek di klaster-klaster lainnya (Priyatman, Sajid, dan Haldivany 2019). Sedangkan Menurut (Suyanto 2019) pada bukunya yang berjudul Data Mining untuk Klasifikasi dan Klasterisasi Data, klasterisasi adalah pengelompokan data secara otomatis tanpa diberitahu label kelasnya.

Klasterisasi berbeda dengan klasifikasi dari sisi data, dimana padametode itu data tidak memiliki kelas (sering diistilahkan dengan label atau target), sehingga klasterisasi masuk dalam kategori pembelajaran tak terpandu (*unsupervised learning*) (Herlawati dan Handayanto 2020). Pada dasarnya *clustering* merupakan suatu metode untuk mencari dan mengelompokkan data yang memiliki kemiripan karakteristik (*similarity*) antara satu data dengan data yang lain. *Clustering* merupakan salah satu metode data mining yang bersifat tanpa arahan (*unsupervised*), maksudnya metode ini diterapkan tanpa adanya latihan(*training*) dan tanpa ada guru serta tidak memerlukan target output (Bastian 2018).

Karena itu perlu diketahui teknik-teknik yang digunakan untuk mengukur tingkat kesamaan atau kemiripan, yaitu (Herlawati dan Handayanto 2020):

1. *Minowski Distance* (masuk dalam kelompok ini *Manhattan*, *Eucledian*, dan *Chebysev*)
2. *Tanimoto Measure*
3. *Pearson's r Measure*
4. *Mahalanobis Measure*

Klasterisasi dikelompokkan ke dalam empat kategori: metode-metode berbasis partisi (*partitioning methods*), metode-metode berbasis hierarki (*hierarchical methods*), metode-metode berbasis kepadatan (*density-based methods*), dan metode-metode berbasis kisi (*grid-based methods*) (Suyanto 2019).

2.2.1 Metode Berbasis Partisi (Partitioning Method)

Sesuai dengan Namanya, metode ini bekerja dengan cara membagi atau mempartisi data ke dalam sejumlah kelompok. Metode ini dikenal juga dengan metode berbasis pusat atau metode berbasis *representative*) (Purba, Tamba, dan Saragih 2018). Menurut (Suyanto 2019), pada metode ini, sebuah klaster C_i dipresentasikan sebagai *centroid* atau secara konsep disebut titik pusat klaster. Kualitas klaster C_i dapat diukur menggunakan variasi dalamklaster, yaitu jumlah kesalahan kuadrat atau *sum of squared error* (SSE) antara semua objek dalam klaster C_i dan centroid c_i, yang didefinisikan sebagai

$$SSE = \sum^{k} \quad \sum_{p \in Ci} dist\,(p, ci)$$

Pada prinsipnya, semua metode berbasis partisi berusaha mengoptimasi fungsi objektif tersebut, yaitu meminimalkan variasi dalam klaster atau meminimalkan SSE. Secara komputasional, kompleksitas optimasi variasi dalam klaster tersebut sangatlah tinggi. Jumlah kemungkinan partisi bersifat eksponensial terhadap jumlah klaster dan dimensi ruang. Algoritma- algoritma yang termasuk ke dalam metode berbasis partisi adalah *K-Means, k-modes, k- medoids, fuzzy c-means,* dan banyak lagi variasi lainnya. (Suyanto 2019).

2.2.2 Metode Berbasis Hirarki (Hierarchical Method)

Sesuai dengan namanya, metode klasterisasi hirarki (*hierarchical clustering*) bekerja dengan cara mengelompokkan objek-objek data ke dalam sebuah hirarki klister (Suyanto 2019). *Hierarchical clustering* dapat dilakukan menggunakan dua strategi, yaitu: dari bawah ke atas (*bottom-up*) yang disebut *agglomerative hierarchical clustering* dan dari atas ke bawah (*top-down*) yang disebut *divisive hierarchical clustering* (Exasanti dan Jananto 2021). Strategi *agglomerative*dimulai dengan menganggap setiap objek tunggal sebagai sebuah

klaster, kemudian secara interaktif menggabungkannya untuk membentuk klaster-klaster yang lebih besar. Sebaliknya, strategi *divisive* dimulai dengansebuah klaster bear yang berisi semua objek dalam himpunan data, yang selanjutnya secara interaktif dipecah ke dalam klaster-klaster yang lebih kecil. Dengan kedua strategi tersebut, *hierarchical clustering* terkadang mengalami masalah terkait dengan keputusan untuk menggabungkan ataumemisahkan klaster. (Suyanto, 2019).

Terdapat banyak metode yang termasuk ke dalam *hierarchical clustering*, yang dapat dikelompokkan ke dalam tiga kategori, yaitu: metode algoritma, metode probabilistik, dan metode Bayesian (Suyanto, 2019). Contoh algoritma-algoritma berbasis hierarki adalah metodealgoritma, BIRCH, Chameleon, dan masih banyak lagi.

2.2.3 Metode Berbasis Kepadatan (Density-Based Method)

Menurut Suyanto (2019), metode klasterisasi berbasis partisi maupun hierarki cenderung lemah untuk klaster-klaster yang berbentuk bebas, acak (tidak bulat), dan memiliki derau atau pencilan.

Density-Based *Clustering* merupakan algoritma yang membagi area berdasarkan kepadatan data menjadi cluster. Masing-masing cluster tersebut memiliki bentuk acak dalam database spatial. Algoritma Density-Based *Clustering* menegaskan bahwa sebuah cluster memiliki banyak titik.Neighborhood dari radius yang diberikan dari setiap titik minimal harus memiliki jumlah minimum poin. Jumlah minimum poin yang dimaksud adalah density dari neighborhood harus lebih dari beberapa thresholdtertentu (Rahmi 2019) . Contoh metode berbasis kepadatan, yaitu: *DBSCAN*, OPTICS, dan DENCLUE.

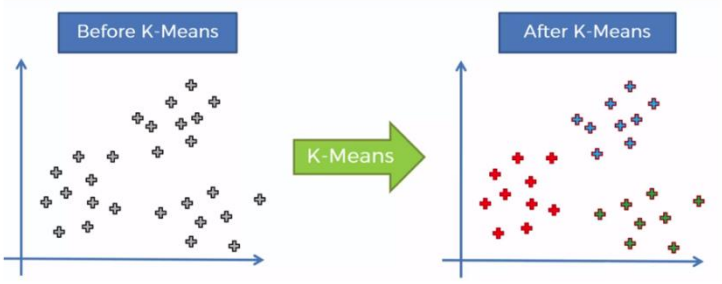
2.2.4 Metode Berbasis Kisi (Grid-Based Method)

Metode berbasis kisi menggunakan pendekatan *space-driven* (disetirrung) dengan mempartisi *embedding space* ke dalam sel-sel yang tidak bergantung pada distribusi objek data. Metode ini menggunakan struktur data kisi multiresolusi. Metode ini memberikan dua keuntungan komputasi yang sangat cepat dan tidak bergantung pada jumlah objek data(melainkan hanya bergantung pada jumlah sel. (Suyanto, 2019). Contoh dari metode berbasis kisi adalah metode STING dan

metode CLIQUE.

2.3 *K-Means Clustering*

K-Means pertama kali dipublikasikan oleh Stuart Lloyd pada tahun 1984 dan merupakan algoritma *clustering* yang banyak digunakan (R. Kurniawan, Hasibuan, and Hasibuan 2023). *K-Means* bekerja dengan mensegmentasi objek yang ada ke dalam kelompok atau yang disebut dengan segmen sehingga objek yang berada dalam masing-masing kelompok lebih serupa satu sama lain dibandingkan dengan objek dalam kelompok yang berbeda (Sutomo dan Dini 2020). *K-Means clustering* merupakan salah satu metode data *clustering* non-hierarki yang mengelompokkan data dalam bentuk satu atau lebih klaster/kelompok (Furqon, Sriani, and Aulia, n.d.). Data-data yang memiliki karakteristik yang sama dikelompokkan dalam satu klaster/ kelompok dan data yang memiliki karakteristik yang berbeda dikelompokkan dengan klaster/kelompok yang lain sehingga data yang berada dalam satu cluster/kelompok memiliki tingkat variasi yang kecil (Aditya, Jovian, dan Sari 2020).



Gambar 2. 3 Visualisasi *K-Means* (Ryan Reza, 2020)

Menurut Suyanto (2019), Ide dasar algoritma *K-Means* sangatlah sederhana, yaitu meminimalkan *Sum of Squared Error* (SSE) antara objek- objek data dengan sejumlah *k centroid*. Algoritma *K-Means* bekerja dengan empat langkah, yang diilustrasikan dalam *pseudocode* di bawah ini. Pertama, dari himpunan data yang akan diklasterisasi, dipilih sejumlah *k* objek secara acak sebagai *centroid* awal (Kurniawan et al., n.d.)

. Kedua, setiap objek yang bukan *centroid* dimasukkan ke klaster terdekat berdasarkan ukuran jarak tertentu. Ketiga, setiap*centroid* diperbarui berdasarkan rata-rata dari objek yang ada di dalam setiap klaster. Keempat, langkah kedua dan ketiga tersebut diulang-ulang (diiterasi) sampai semua *centroid* stabil atau konvergen, dalam arti semua *centroid* yang dihasilkan dalam iterasi saat ini sama dengan semua *centroid* yang dihasilkan pada iterasi sebelumnya.

Menurut (Asmiatun, Wakhidah, dan Putri 2020), Tahapan dalam klasterisasidengan *K-Means* antara lain:

1. Menentukan jumlah klaster "K" yang akan dibagi.
2. Menentukan nilai centroid Dalam menentukan nilai centroid untuk awal iterasi, nilai awal centroid dilakukan secara acak. Sedangkan iterasi, maka digunakan rumus sebagai berikut

   $$\bar{v}_{ij} = \frac{1}{N_i} \sum_{k=0}^{Ni} Xkj$$

   dimana:

   V*ij* adalah centroid/ rata-rata cluster ke-I untuk variabel ke-j

   N*i* adalah jumlah data yang menjadi anggota cluster ke-i

   i,k adalah indeks dari cluster

   j adalah indeks dari variabel

   X*kj* adalah nilai data ke-k yang ada di dalam cluster tersebut untuk variabel ke-j
3. Menghitung jarak antara titik centroid dengan titik tiap objek Untuk menghitung jarak tersebut dapat menggunakan Euclidean Distance,

   yaitu

   $D_e = \sqrt{(Xi - Si)^2 + (yi - ti)^2}$

   dimana:

   De adalah Euclidean Distance

   i adalah banyaknya objek,

   (x,y) merupakan koordinat object dan

   (s,t) merupakan koordinat centroid.

4. Pengelompokan object

   Untuk menentukan anggota cluster adalah dengan memperhitungkan jarak minimum objek. Nilai yang diperoleh dalam keanggotaan data pada distance matriks adalah 0 atau 1, dimana nilai 1 untuk data yang dialokasikan ke cluster dan nilai 0 untuk data yang dialokasikan ke cluster yang lain.

5. Kembali ke tahap 2, lakukan perulangan hingga nilai centroid yang dihasilkan tetap dan anggota cluster tidak berpindah ke cluster lain.

Kelebihan Algoritma *K-Means* diantaranya adalah mampu mengelompokkan objek besar dan pencilan obyek dengan sangat cepat sehingga mempercepat proses pengelompokan. Kekurangan Algoritma *K-Means* yaitu sangat sensitive pada pembangkitan titik pusat awal secara random (Bastian 2018).

Tabel 2. 1 Kasus *K-Means clustering*

Siswa	A	B
1	1	1
2	2	1
3	4	3
4	5	4

Iterasi – 1 :

Diketahui:

1. Jumlah data siswa (S) = 4
2. Ditentukan nilai K = 2
3. Centroid cluster 1 (CC-1) = {1,1}
4. Centroid cluster 2 (CC-2) = {2,1}

Hitung jarak data - Centroid:

Jarak data - centroid C-1:

1. $d\,(S_1,C_1) = \sqrt{(\mathbf{1}-\mathbf{1})^2 + (\mathbf{1}-\mathbf{1})^2} = \mathbf{0}$
2. $d\,(S_2,C_1) = \sqrt{(\mathbf{2}-\mathbf{1})^2 + (\mathbf{1}-\mathbf{1})^2} = 1$
3. $d\,(S_3,C_1) = \sqrt{(\mathbf{4}-\mathbf{1})^2 + (\mathbf{3}-\mathbf{1})^2} = 3{,}61$
4. $d\,(S_4,C_1) = \sqrt{(\mathbf{5}-\mathbf{1})^2 + (\mathbf{4}-\mathbf{1})^2} = 5$

Jarak data – centroid C-2:

1. $d\,(S_1,C_2) = \sqrt{(\mathbf{1}-\mathbf{2})^2 + (\mathbf{1}-\mathbf{1})^2} = 1$
2. $d\,(S_2,C_2) = \sqrt{(\mathbf{2}-\mathbf{2})^2 + (\mathbf{1}-\mathbf{1})^2} = 0$
3. $d\,(S_3,C_2) = \sqrt{(\mathbf{4}-\mathbf{2})^2 + (\mathbf{3}-\mathbf{1})^2} = 2{,}83$
4. $d\,(S_4,C_2) = \sqrt{(\mathbf{5}-\mathbf{2})^2 + (\mathbf{4}-\mathbf{1})^2} = 4{,}24$

Mengelompokkan sesuai jarak terpendek

Tabel 2. 2 Hasil Pengelompokan Data Dengan Centroid

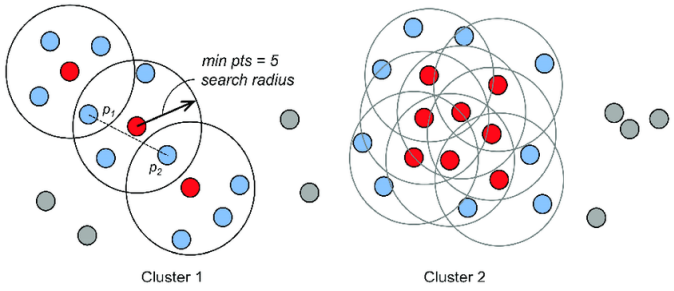
SISWA	A	B	d(S,C1)	d(S,C2)	C-1	C-2
1	1	1	0	1	√	
2	2	1	1	0		√
3	4	3	3,61	2,83		√
4	5	4	5	4,24		√

Inilah hasil proses pengelompokan berdasarkan jarak yang terdekat antara data dengan centroid.

2.4 *DBSCAN Clustering*

DBSCAN merupakan metode *clustering* yang membangun cluster berdasarkan kepadatan, cluster yang tidak termasuk objek dianggap noise (Inayah dkk. 2022). Tidak seperti *K-Means clustering*, *DBSCAN* tidak perlu menentukan jumlah kelompok secara manual. Namun, diperlukan jumlah minimum tetangga untuk dipertimbangkan dalam kelompok dan jarak maksimum yang diperbolehkan antara titik mana pun untuk menjadi bagian dari kelompok yang sama. Dalam pengguna tertentu, ditentukan jarak di sekitar sampel, *DBSCAN* akan menghitung

jumlah tetangga. Ketika jumlah tetangga antar jarak melebihi ambang batas, *DBSCAN* akan mengelompokkan titik data sebagai satu kelompok.



Gambar 2. 4 Algoritma pengelompokan menggunakan radius pencarian dan jumlah titik minimum untuk mendefinisikan sebuah cluster (Tonini dkk, 2019)

Langkah-langkah untuk melengkapi algoritma *DBSCAN* menurut (Inayah dkk. 2022) adalah sebagai berikut.

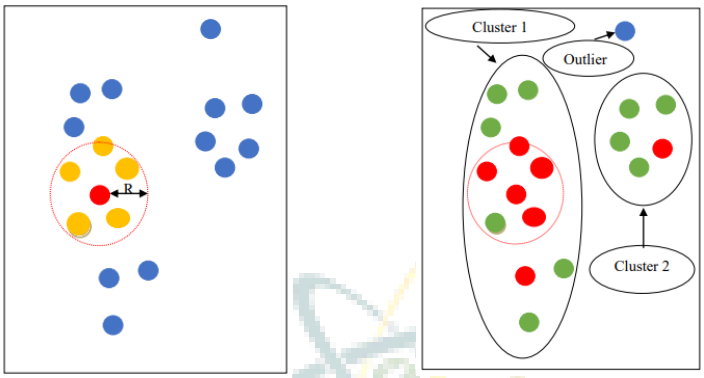
1. Inisialisasi parameter Min Pts dan dan parameter Eps.
2. Tentukan titik awal atau p secara acak.
3. Hitung Eps atau semua jarak titik yang kepadatan atau density *reachable* terhadap p menggunakan rumus jarak *Euclidean* berikut.

$$D\,i\,j \;=\; \sqrt{\sum_{a}^{p} (\,X\,ia \;\; - X\,ja\,)^2}$$

   Dimana X_{ia} merupakan variable ke-a dari obyek i (i=1, …, n; a=1, …, p) dan d_{ij} adalah nilai *euclidean distance.*
4. Terbentuk sebuah cluster ketika titik yang memenuhi Eps lebih dariMinPts dan titik p sebagai core point.
5. Lakukan pengulangan langkah 3 – 4 hingga dilakukan proses pada semua titik. Jika p merupakan titik border dan tidak ada titik yang density reachable terhadap p, maka proses dilanjutkan ke titik yang lain.

Salah satu keuntungan *DBSCAN* dibanding *K-Means* adalah *DBSCAN* tidak terbatas pada jumlah cluster yang ditetapkan saat inisialisasi. Algoritmanya akan menentukan jumlah cluster berdasarkan kepadatan suatu daerah. Algoritma *DBSCAN* dibangun dalam konsep kebisingan (noise). Pada umumnya digunakan

untuk mendeteksi outliers dalam data, seperti aktivitas kecurangan dalam kartu kredit, ecommerce, atau klaim asuransi (Mohammad Mahmudur Rahman Khan, 2022).



Gambar 2. 5 Kasus *DBSCAN* (Inayah dkk. 2022)

Parameter :

R = 2 unit dan M = 5

Langkah 1 :

Pilih poin secara acak

Langkah 2 :

Poin pertama merupakan poin Inti karena dalam radius R sebesar 2 unit, terdapat 6 poin (termasuk poin pertama)

Langkah 3 :

Ulangi langkah 1 dan 2 hingga Semua poin sudah dikategorikan seluruh poin sudah dikategorikan.

Langkah 4 : Menentukan jumlah Cluster

2.5 Normalisasi

Salah satu topik cukup komplek dalam dunia manjemen database adalah proses untuk menormalisasi tabel-tabel dalam database relasional. Dengan normalisasi kita ingin mendesain database relasional yang terdiri dari tabel-tabel berikut:

1) Berisi data yang diperlukan
2) Memiliki sesedikit mungkin redundasi
3) Mengakomodasi banyak nilai untuk tipe data yang diperlukan

4) Mengedefinisikan update
5) Menghindari kemungkinan kehilalangan data secara tidak segaja atau tidak diketahui.

Alasan utama dari normalisasi database minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya *"insertion anomalies", "deletion anomalies", dan "update anomalies"*. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada database yang tidak normal.

"*Insertion anomaly"* adalah sebuah kesalahan dalam peneempatan informasi entry data baru ke seluruh tempat dalam database di mana informasi tersebut perlu disimpan. Dalam database yang telah dinormalisasi, proses pemasukan suatu informasi baru hanya perlu di masukkan ke dalam suatu tempat.

"Deletion anomaly" adalah sebuah kesalahan dalam penghapusan suatu informasi dalam database harus dilakukan dengan penghapusan informasi tersebut dari beberapa tempat dalam database. Dalam database yang telah dinormalisasi, penghapusan suatu informasi hanya diperlu dilakukan dalam satu tempat dalam database tersebut.

Sedangkan dalam melakukan update satu informasi, kesalahan juga dapat terjadi ketika kita harus melakukan update keseluruh tempat yang menyimpan informasi tersebut. Kesalahan ini disebut dengan *"update anomaly".*

Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Pada dasarnya desain logika basis data relasional dapat menggunakan prinsip normalisasi maupun transformasi dari model E-R ke bentuk fisik (Kusrini, 2018).

2.6 Min Max Scaller

Normalisasi Min-Max Scaller Min-max scaller normalization adalah metode normalisasi dengan melakukan transformasi linear terhadap data sehingga menghasilkan keseimbangan nilai pada data(D. A. Nasution, H. H. Khotimah, and N. Chamidah). Metode ini dapat dirumuskan seperti pada rumus dibawah ini.

$$Z = \frac{X-min}{Max-Min}$$

Dimana:

X adalah nilai datanya.

min adalah nilai terkecil dari atribut tersebut.

max adalah nilai terbesar dari atribut tersebut. Python

2.7 Python

Menurut Handayanto dan Herlawati (2020), bahasa Python diluncurkan pertama kali pada tahun 1991 di *Scitching Mathematicsch Centrum*, Belanda. Bahasa ini dirancang oleh Guido van Rossum yang mengambil nama Python dari grup komedi terkenal asal Inggris, Monty Python. Menurutnya, Bahasa inisangat diminati karena karakteristiknya yang *open source* dengan lisensi GPL – *compatible*. Beberapa literatur mengungkapkan beberapa kelebihan Python, diantaranya: *Readability*, efisien, multifungsi, interoperabilitas yang baik, dan dukungan komunitas yang kuat.

Python adalah bahasa pemrograman yang bersifat open source. Bahasa pemrograman ini dioptimalisasikan untuk software quality, developer productivity, program portability, dan component integration. Python telah digunakan untuk mengembangkan berbagai macam perangkat lunak, seperti internet scripting, systems programming, user interfaces, product customization, numeric programming dll. (Lutz, 2018). Bahasa pemrograman Python memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak. Berikut adalah beberapa fitur yang ada pada bahasa pemrograman Python (Lutz, 2018):

1. *Multi Paradigm Design*
2. *Open Source*
3. *Simplicity*
4. *Library Support*
5. *Portability*
6. *Extendable*
7. *Scalability*

2.8 Indeks Silhouette

Indeks Silhouette merupakan gabungan dari Metode separasi dan kohesi. (Kodinariva & Makwana, 2018). Indeks validitas Silhouette menghitung rata-rata nilai setiap titik pada himpunan data. perhitungan nilai setiap titik adalah selisih nilai separation dan compactness yang dibagi denganmaksimum antara keduanya. (Aditya, dkk. 2020).

Menurut Qadrini (2020), tahapan untuk menghitung nilai Silhoutte Coeffisient adalah sebagai berikut :

1. Untuk setiap objek i, hitung rata-rata jarak dari objek I dengan seluruh objek yang berada dalam satu kelompok. Akan didapatkan nilai rata-rata yang disebut ai .
2. Untuk setiap objek i, hitung rata-rata jarak dari objek i dengan objekyang berada di kelompok lainnya. Dari semua jarak rata-rata tersebut ambil nilai yang paling kecil. Nilai ini disebut bi .
3. Setelah itu maka untuk objek i memiliki nilai Silhouette Coefficient
4. hasil perhitungan nilai Silhoutte Coeffisient dapat bervariasi antara -1 hingga 1. Hasil pengelompokan dikatakan baik jika nilai Silhoutte Coeffisient bernilai positif ($ai< bi$) dan ai mendekati 0, sehingga akan menghasilkan nilai Silhoutte Coeffisient yang maksimum yaitu 1 saat ai = 0. Maka dapat dikatakan,jika Si = 1 berarti objek i sudah berada dalam kelompok yang tepat. Jika nilai
5. Si = 0 maka objek i berada di antara dua kelompok sehingga objek tersebut tidak jelas harus dimasukan ke dalam kelompok A atau kelompok B. Akan tetapi, jika $Si = -1$ artinya struktur kelompok yang dihasilkan overlapping, sehingga objek i lebih tepat dimasukan ke dalam kelompok yang lain. Nilai rata-rata Silhouette Coeffisient dari tiap objek dalam suatu kelompok adalah suatu ukuran yang menunjukan seberapa ketat data dikelompokan dalam kelompok tersebut. (Qadrini, 2020)
6. Berikut adalah nilai Silhouette Coefficient berdasarkan (Kaufman & Rousseeuw) adalah:

Tabel 2. 3 Tabel Keterangan Nilai Silhoutte

Nilai SC	Keterangan
$0{,}7 < SC \leq 1$	Struktur Kuat
$0{,}5 < SC \leq 0{,}7$	Struktur Medium
$0{,}25 < SC \leq 0{,}5$	Struktur Lemah
$SC \leq 0{,}25$	Tidak ada Struktur

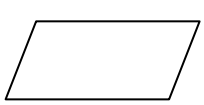
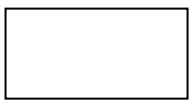
2.9 *Flowchart*

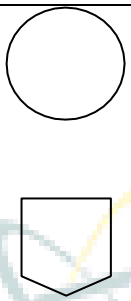
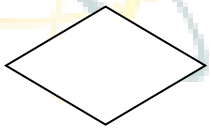
Flowchart adalah gabungan kata *flow* dan *chart*. *Flow* berarti aliran, dan *chart* berarti bagan atau *diagram*. Sehingga pengertian dari *flowchart* adalah bagan berupa aliran yang saling terhubung.

Flowchart adalah sebuah bagan-bagan yang memiliki arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma. *Flowchart* adalah representasi grafis dari suatu algoritma. Programmer sering menggunakan *flowchart* sebagai alat perencanaan program untuk memecahkan sebuah masalah dengan menggunakan simbol-simbol yang saling terhubung untuk menunjukkan aliran informasi dan pemrosesan. *Flowchart* atau bagan alir digunakan sebagai alat bantu komunikasi untuk menyampaikan informasi agar lebih mudah dibaca dan dimengerti (Sahmiati dkk. 2022).

Simbol *flowchart* dan fungsinya dapat dilihat pada table berikut:

Tabel 2. 4 Simbol *Flowchart* (https://www.dicoding.com/blog/*flowchart*)

No.	Nama	Simbol	Fungsi
1	"Simbol *input / output"*		Simbol ini digunakan untuk mewakili data input / output
2	"Simbol proses"		Simbol digunakan untuk mewakili suatu proses

3	"Simbol garis alir"		Simbol garis alir (*flow lines symbol*) digunakan untuk menunjukkan arus dari proses
4	"Simbol penghubung"		Simbol ini digunakan untuk menunjukkan sambungan dari bagan alir yang terputus pada halaman yang sama atau lainnya
5	"Simbol keputusan"		Simbol keputusan (*decision symbol*) digunakan untuk penyelesaian suatu kondisi
6	"Simbol proses terdefinisi"		(*predefinied process symbol*) digunakan untuk menunjukkan suatu operasi yang rinciannya ditujukan di tempat lain
7	"Simbol persiapan"		Simbol persiapan (*preparation symbol*) digunakan untuk memberi nilai awal untuk suatu besaran
8	"Simbol titik terminal"		Simbol titik terminal (*terminal point symbol*) digunakan untuk menunjukkan awal dan akhir dari suatu proses
9	"Document"		Mencetak output dalam format dokumen (melalui printer).

Berikut beberapa keuntungan dan kekurangan dari penggunaan *flowchart*, diantaranya:

Tabel 2. 5 Keuntungan dan Kekurangan *Flowchart* (https://www-aplustopper-com)

Keuntungan *Flowchart*	Kekurangan *Flowchart*
Cara paling efisien untuk mengkomunikasikan logika sistem.	*Flowchart* sulit dibentuk pada program yang besar dan kompleks.
Sebagai panduan dalam proses perancangan program	Tidak mempunyai jumlah detail yang tepat dan sesuai
Membantu proses *debugging*	*Flowchart* sulit untuk diproduksi
Dapat membantu menganalisis program dengan lebih mudah sekaligus sebagai dokumentasi.	*Flowchart* sulit untuk dimodifikasi

2.10 Penelitian Terdahulu

Metode *Clustering DBSCAN* dan *K-Means Clustering* ini sebelumnya telah digunakan pada beberapa penelitian lain dengan studi kasus dan objek analisis yang berbeda. Penelitian terdahulu yang dijadikan sebagai kajian pustaka dalam penelitian ini, yaitu:

Tabel 2. 6 Penelitian Terdahulu

No	Judul Penelitian	Penulis	Hasil Penelitian
1.	Implementasi *DBSCAN* dalam *Clustering* Data Minat Mahasiswa Setelah Pandemi Covid 19	A Kristianto	Penelitian ini bertujuan untuk mencari faktor penentu minat mahasiswa dalam berkuliah setelah pandemi menggunakan algoritma *DBSCAN*. Data yang digunakan yaitu data primer, diperoleh dengan menyebarkan kuesioner

No	Judul Penelitian	Penulis	Hasil Penelitian
			kepada beberapa mahasiswa dengan berbagai latar belakang dengan jumlah lebih kurang 400 responden, yang menghasilkan 385 kuesioner yang valid. Hasil penelitian ini dapat disimpulkan bahwa algoritma *DBSCAN* menunjukkan performa yang baik dalam proses *clustering* dengan data yang besar. Ditemukan faktor penentu pemilihan minat metode perkuliahan mahasiswa adalah semester, IPK, dan kelompok ilmu (Eksakta atau Non-Eksakta).
2.	Perbandingan *Clustering* Karyawan Berdasarkan Nilai Kinerja Dengan Algoritma *K-Means* Dan Fuzzy C-Means	Anissa Enggar Pramitasari, Yessica Nataliani	Penelitian ini bertujuan untuk mengevaluasi kinerja karyawan untuk meningkatkan produktivitas karyawan. Metode yang digunakan adalah Metode *K-Means*. Data yang digunakan yaitu data penilaian kinerja pada bagian produksi sejumlah 25 orang. Dari hasil analisis yang sudah diperoleh dapat disimpulkan bahwa

No	Judul Penelitian	Penulis	Hasil Penelitian
			pengelompokan karyawan dari kepala bagian tidak sesuai dengan data penilaian kinerja karyawan yang berupa angka. Hasil pengelompokan dari data nilai kinerja karyawan menggunakan algoritma *K-Means* dengan algoritma FCM berbeda.
3.	Penerapan *Clustering DBSCAN* Untuk Pertanian Padi Di Kabupaten Karawang	Betha Nurina Sari, Aji Primajaya	Penelitian ini bertujuan untuk pemetaan atau pengelompokan daerah sesuai karakteristiknya tersebut. Metode yang digunakan yaitu *Clustering DBSCAN*. Data yang digunakan pada penelitian ini adalah data laporan hasil pertanian di Kabupaten Karawang mulai tahun 2010 sampai tahun 2015 yang diambil dari Dinas Pertanian dan Kehutanan Kabupaten Karawang bagian pangan. Hasil cluster menunjukkan perbedaan mengenai

No	Judul Penelitian	Penulis	Hasil Penelitian
			tingkat curah hujan, luas lahan yang mempengaruhi jumlah produksi, dan jumlah serangan hama serta jenis hama yang menyerang lahan pertanian.
4.	Perbandingan Kinerja *K-Means* Dengan *DBSCAN* Untuk Metode *Clustering* Data Penjualan Online Retail	Bena Siti Ashari, Steven Christ Otniel, Rianto	Tujuan dari penelitian ini untuk melakukan pengelompokan data penjualan menggunakan metode *K-Means* dan *DBSCAN*. Data yang digunakan menggunakan data set sebanyak 500 data dan memiliki 3 atribut: deskripsi, kuantitas barang per transaksi dan harga barang per unit. Hasil penelitian ini yaitu pengelompokan data dari data set Retail Online dengan menggunakan *K-Means* dengan jumlah kelompok tiga menghasilkan kelompok 1 dengan 103 data, kelompok 2 dengan 261 data, kelompok 3 dengan 134

No	Judul Penelitian	Penulis	Hasil Penelitian
			data. Sedangkan pengelompokan menggunakan metode *DBSCAN* dengan nilai Epsilon 1.005 dan Minimal Points: 11 menghasilkan kelompok 1 dengan 30 data, kelompok 2 dengan 47 data dan kelompok 3 dengan 347 data dan sisa 74 data adalah noise yang tidak masuk ke dalam cluster mana pun.
5.	Perbandingan Algoritma *DBSCAN* dan *K-Means Clustering* untuk Pengelompokan Kasus Covid-19 di Dunia	Rimelda Adha, Nana Nurhaliza , Ummi Soleha , Mustakim	Penelitian dilakukan untuk mengelompokkan negara-negara yang memiliki pola kasus Covid-19 di dunia. Penelitian ini menggunakan algoritma *DBSCAN* dan *K-Means*. Data yang digunakan diperoleh melalui halaman website resmi World Health Organization (WHO) pada alamat https://covid19.who.int. Berdasarkan hasil pengujian validasi cluster terhadap hasil klasterisasi data kasus Covid-19 di dunia menggunakan algoritma *DBSCAN* dan *K-*

No	Judul Penelitian	Penulis	Hasil Penelitian
			Means, maka pada penelitian ini *K-Means* lebih unggul daripada *DBSCAN* dengan nilai SI terbaik yaitu 0,6902 dengan nilai k = 8. Pola dari hasil penelitian dapat dijadikan sebagai acuan dalam menggambarkan model *clustering* Covid-19 di Dunia.
6.	Implementasi Metode *DBSCAN* Pada Pengelompokan Kabupaten/Kota Di Pulau Kalimantan Berdasarkan Indikator Kesejahteraan Rakyat	Risman, Syaripuddin , Suyitno	Tujuan dalam penelitian ini adalah untuk mengetahui berapa cluster yang terbentuk pada pengelompokan menggunakan metode *DBSCAN* dan mengetahui kombinasi parameter yang optimal berdasarkan nilai Davies Bouldin Index (DBI). Data Penelitian ini adalah data sekunder yaitu data Indikator Kesejahteraan Rakyat di 42 Kabupaten/Kota yang ada di Pulau Kalimantan tahun 2017. Dari hasil analisis cluster menggunakan metode *DBSCAN* didapatkan bahwa kombinasi parameter yang optimal adalah ε = 2 dan MinPts = 2 dengan nilai DBI

No	Judul Penelitian	Penulis	Hasil Penelitian
			sebesar 0.605 yang menghasil 5 cluster.

Kesimpulan dari penelitian terdahulu ialah Metode *Clustering* dan *DBSCAN* dapat menyelesaikan permasalahan dalam data mining dengan mengelompokkan berbagai data untuk setiap tujuan penelitian. Metode *DBSCAN* memiliki performa yang baik dalam proses *clustering*, terutama dengan data yang cukup banyak. Metode *DBSCAN* sering dibandingkan dengan algoritma *K-Means*, namun perbedaan kedua metode ini terletak pada jumlah *cluster* inputan yang dibutuhkan. Metode *DBSCAN* hanya membutuhkan *epsilon* dan minPts untuk mengelola data *cluster* yang dibutuhkan. Metode *DBSCAN* memiliki keunggulan berupa performa untuk menangkap *cluster* yang memiliki beragam bentuk. Namun sayangnya metode ini kurang cocok digunakan pada data dengan tingkat kerapatan yang beragam.