**JURNAL LOCUS: Penelitian & Pengabdian**
**Volume 4 No. 8 Agustus 2025**
E-ISSN 2829-7334| P-ISSN 2829-5439
Hompage: https://locus.rivierapublishing.id/index.php/jl

# Klasterisasi Data Penjualan Toko Perak J-Maskus Mengguanakan Algoritma HDBSCAN

**Hendri Diana Rusmana, Wina Witanti, Gunawan Abdillah**
Universitas Jenderal Achmad Yani, Indonesia
Email: hendridiana21@if.unjani.ac.id, witanti@gmail.com, gunawanabdillah03@gmail.com

**Abstrak:**
Di era digital, analisis data penjualan menjadi kunci pengambilan keputusan bisnis. Pebisnis yang mampu mengumpulkan dan menganalisis data secara efektif memiliki keunggulan kompetitif yang signifikan dibandingkan dengan pesaing yang belum memanfaatkan potensi data secara maksimal. Toko Perak J-Maskus menghadapi tantangan dalam mengelola stok akibat ketidakseimbangan persediaan dan permintaan. Penelitian ini bertujuan untuk mengelompokkan produk berdasarkan pola penjualan guna meningkatkan manajemen inventaris dengan menggunakan teknik analisis data yang canggih. Penelitian ini menerapkan algoritma HDBSCAN, yang dipilih karena kemampuannya mengidentifikasi cluster dengan kepadatan bervariasi dan mendeteksi outlier. Dataset yang digunakan terdiri dari 5.765 transaksi (2022–2024) dengan atribut produk dan jumlah terjual. Hasil penelitian menunjukkan parameter optimal yang ditemukan adalah min_samples=5 dan min_cluster_size=5, dengan nilai silhouette score 0.6507 (struktur menengah) dan menghasilkan 206 cluster. Visualisasi t-SNE mengonfirmasi bahwa cluster terpisah dengan jelas. Temuan ini menunjukkan bahwa pengelompokan data penjualan dapat membantu mengidentifikasi produk laris dan mengurangi overstock. Hasil ini dapat diterapkan untuk mengembangkan strategi manajemen stok yang lebih efektif, seperti mengoptimalkan variasi produk dan meningkatkan perputaran inventaris.
**Kata kunci**: Data mining, HDBSCAN, klasterisasi, optimasi stok, silhouette score

***Abstract:***
In the digital era, sales data analysis has become crucial for business decision-making. Business owners who can effectively collect and analyze data gain a significant competitive advantage over those who have not fully leveraged data potential. J-Maskus Silver Store faces challenges in managing inventory due to an imbalance between supply and demand. This research aims to cluster products based on sales patterns to improve inventory management using advanced data analysis techniques. The study applies the HDBSCAN algorithm, selected for its ability to identify clusters with varying densities and detect outliers. The dataset consists of 5,765 transactions (2022–2024), with product attributes and sold quantities. The results show the optimal parameters as min_samples=5 and min_cluster_size=5, resulting in a silhouette score of 0.6507 (medium structure) and 206 clusters. t-SNE visualization confirmed that the clusters were clearly separated. The findings reveal that clustering sales data can help identify fast-moving products and reduce overstock. These results can be applied to develop more effective stock management strategies, such as optimizing product assortment and improving inventory turnover.
***Keywords****: Data mining, HDBSCAN, clustering, stock optimization, silhouette score*

*Corresponding*: Hendri Diana Rusmana
E-mail: hendridiana21@if.unjani.ac.id

## PENDAHULUAN

Di era digital ini, data telah menjadi aset yang berharga bagi setiap jenis bisnis. Informasi yang terkumpul dari berbagai transaksi penjualan, banyaknya produk yang laku dapat memberikan wawasan yang dapat mengarahkan perusahaan dalam membuat keputusan yang lebih tepat dan strategis (Agustia

& Suryani, 2018; Dewantari et al., 2020). Pebisnis yang mampu mengumpulkan dan menganalisis data secara efektif memiliki keunggulan kompetitif yang signifikan dibandingkan dengan pesaing yang belum memanfaatkan potensi data secara maksimal.

Data penjualan dapat digunakan sebagai acuan untuk menghitung keuntungan dan kerugian, dan data ini dapat dimanfaatkan lebih untuk meningkatkan strategi bisnis pemilik toko. Data tersebut tidak hanya berfungsi sebagai arsip, data tersebut dapat dimanfaatkan dan diolah menjadi informasi yang berharga, salah satu cara mendapatkan informasi dari data penjualan adalah dengan menggunakan metode data mining(Lestari & Hafiz, 2020; Rosmayati et al., 2023; Sibarani, 2020; Sulistiyowati et al., 2023; Takdirillah, 2020).

Data mining adalah proses mencari informasi menarik dalam sejumlah data terpilih dengan menggunakan atau dengan teknik tertentu, pengelompokan atau clustering salah satu metode yang terdapat dalam data mining (Hamid Mughal, 2018; Hassani et al., 2020; Nasari & Darma, 2015; Shah & Shah, 2023). Salah satu algoritma pengelompokan berbasis kepadatan pertama, dan yang paling terkenal yaitu DBSCAN.

DBSCAN adalah algoritma pengelompokan berbasis kepadatan yang terkenal dan diusulkan pada tahun 1996, memerlukan parameter jarak sebagai ambang batas untuk menentukan apakah titik data termasuk cluster atau outlier, namun tantangan muncul dalam menentukan nilai terbaik untuk parameter ini karena kurangnya intuisi, dan algoritma ini juga dapat mengalami masalah saat kepadatan cluster bervariasi(Stewart & Al-Khassaweneh, 2022).

Dari algoritma DBSCAN yang diperbarui, lahirlah algoritma HDBSCAN (hierarchical density based spatial clustering of aplication with noise), yaitu konsep algoritma pengembangan dari ordering points to identify the clustering structure (OPTICS), yang menggunakan satu parameter input yang disebut Mpts (MinPoints). Metode hierarki HDBSCAN dapat digunakan sebagai dasar untuk tugas pascapemrosesan, membuat metode hierarki yang ringkas adalah salah satu tugas yang harus diselesaikan(Stewart & Al-Khassaweneh, 2022)(Wahyuni et al., 2021). HDBSCAN adalah metode pengelompokan hirarkis berdasarkan kepatadan yang menentukan hasil pengelompokkan berdasarkan multi-level dalam pemilihan cluster, dengan memperkenalkan konsep stablilitas cluster, hubungan antar jarak dan mendefinisikan keberadaan cluster di berbagai ambang batas untuk mengukur stabilitas cluster(Liu et al., 2021).

Studi yang mengimplementasikan algoritma HDBSCAN untuk dataset pendataan potensi desa(Wahyuni et al., 2021), memiliki kesimpulan algoritma HDBSCAN dengan parameter optimal yaitu minpts = 4, membentuk 2 cluster, cluster 1 memiliki 4 anggota desa dan cluster 2, memiliki 61 desa dengan karakteristik yang berbeda beda, cluster 1 memiliki karakteristik keluarga yang menggunakan listrik PLN beranggotakan 25 keluarga sedangkan cluster 2 memiliki 250 keluarga. Penelitian ini menunjukan algortima HDBSCAN bisa menangani kepadatan data yang bervariatif.

Selanjutnya studi yang memanfaatkan data penjualan untuk mengelompokkan produk menjadi 3 cluster yaitu kurang laris yang berjumlahkan 154 produk, laris yang berjumlahkan 14 produk dan paling laris berjumlahkan 7 produk dengan menggunakan algoritma k-means(Bani Saleh FTID et al., 2023), pada penelitian ini jumlah cluster ditentukan diawal penelitian, dimana jumlah cluster dapat dioptimalkan untuk mendapatkan variasi cluster yang lebih spesifik.

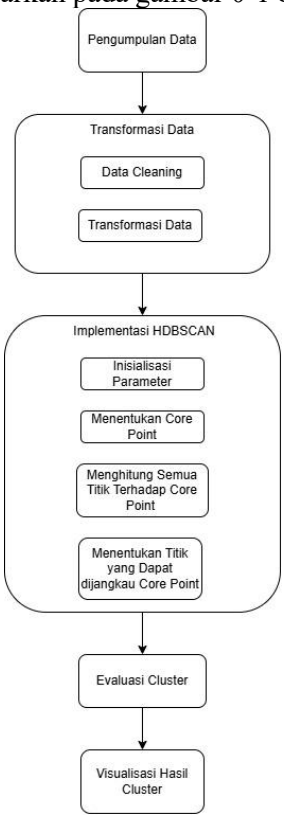
Berdasarkan studi terdahulu, algoritma HDBSCAN telah dibandingkan dengan algoritma AHC (agglomerative hierarchical clustering) untuk menangani data yang mengandung pencilan (outliers), studi tersebut memiliki kesimpulan bahwa dengan dataset yang diteliti, algoritma HDBSCAN terbukti lebih efisien daripada Agglomerative HIierarchical Clustering dalam menangani data dengan pencilan (outliers)(Fitrianto, 2024). Selanjutnya pada studi yang membandingkan algoritma HDBSCAN dengan algoritma K-Means, AHC, dan DBSCAN(*INTISARI CLUSTERING HIERARCHICAL DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE ( HDBSCAN ) UNTUK DATA SPASIAL ( STUDI KASUS : PEMETAAN KAWASAN WISATA TERPADU DI KABUPATEN GUNUNGKIDUL , DAERAH ISTIMEWA YOGYAKARTA ) Oleh LUTHFI SYAFI ABDILLAH* , 2024), dalam studi tersebut dari semua algoritma yang diterapkan, dan memiliki kesimpulan bahwa algoritma HDBSCAN lebih baik dalam mengelompokan lokasi dengan nilai silhouette score, davis-bouldin index, calinski-harabasz index, jarak rata rata dan jarak maksimal dalam cluster dibandingkan dengan algoritma lain. Algoritma HDBSCAN dan DBSCAN juga dibandingkan dalam penelitian lain(Handayani & Isnaini Rosyida, 2023), hasil evaluasi dari penelitian tersebut silhouette coefficient menunjukan bahwa pengelompokan dengan algoritma HDBSCAN lebih baik dari algoritma DBSCAN karena

algoritma HDBSCAN mampu mengumpulkan informasi lebih mendalam dari evaluasi menyeluruh dengan sangat cepat dan efektif.

Tujuan dari penelitian ini adalah mengelompokan produk dengan memanfaatkan data penjualan di toko perak J-Maskus menggunakan algoritma HDBSCAN. Dengan mempertimbangkan hasil dari penelitian sebelumnya, dimana algoritma HDBSCAN menunjukan performa yang lebih baik dibandingkan dengan beberapa algoritma pengelompokkan lain yang telah dibandingkan, dan juga karena kemampuan HDBSCAN dalam menangani kepadatan kelompok yang bervariatif, serta tidak perlunya menentukan jumlah kelompok karena jumlah kelompok akan dibentuk secara otomatis berdasarkan kepadatan data.

## METODE PENELITIAN

Alur dari penelitian ini digambarkan pada gambar 0-1 berikut ini




Gambar 1. Alur penelitian

Data yang dikumpulkan untuk penelitian ini dikumpulkan dengan cara meminta langsung data penjualan selama 6 tahun terakhir kepada pemilik toko perak J-Maskus. Data yang diminta mencakup informasi penjualan selama enam tahun terakhir, mulai dari tahun 2019 sampai tahun 2024, atribut yang akan digunakan dalam penelitian ini adalah tanggal, nama produk, ukuran, warna, dan jumlah terjual. Gambar 0-2 berikut merupakan contoh dari data yang akan digunakan.

### PreProcessing Data

Tujuan dari tahap selanjutnya adalah untuk mengubah data mentah menjadi data berkualitas sehingga dapat digunakan dengan baik pada tahap berikutnya. Pra-Proses mencangkup persiapan dan transformasi data, pada tahap ini data mengalami transformasi, yaitu proses mengubah data menjadi data yang dapat diproses menggunakan algoritma yang akan digunakan. Algoritma HDBSCAN hanya bisa membaca data bertipe numerik, oleh karena itu pada tahap ini akan dilakukan transformasi data yaitu mengubah semua data kategorikal menjadi numerik agar algoritma HDBSCAN bisa berjalan. Data kategorikal yang akan diubah menjadi bentuk numerik yaitu Barang, Ukuran, Model, dan Warna agar algoritma HDBSCAN bisa digunakan untuk pengolahan data tersebut.

### Inisialisasi Parameter

Langkah pertama yang dilakukan setelah data siap digunakan adalah dengan menginisialisasi parameter. Algoritma HDBSCAN tidak parameter memiliki epsilon seperti di DBSCAN, karena algoritma HDBSCAN secara otomatis menentukan jarak optimal berdasarkan kepadatan, tetapi sama seperti DBSCAN pada algoritma HDBSCAN juga perlu memasukan ukuran minimum cluster. Minimum cluste yaitu jumlah minimal cluster yang akan terbentuk, ini akan berpengaruh pada jumlah cluster yang terbentuk. Sebagai contoh jika minimum cluster diinisialisasi dengan 5 maka jumlah cluster yang akan terbentuk tidak akan kurang dari 5 cluster dengan kepadatan point yang berbeda, selain itu ada juga minpts yaitu jumlah point minimal dari sebuah cluster, sebagai contoh jika minpts diinisialisasi dengan 10 maka point minimal dari setiap cluster tidak akan kurang dari 10. Inisialisasi parameter sangat berpengaruh kepada hasil validasi

**Menentukan Core Point**

Langkah selanjutnya setelah menginisialisasi parameter adalah menentukan nilai Xp yang akan menjadi core point secara acak. Core Point adalah titik dimana jumlah tetangga dalam radius terntentu atau melebihi ambang batas.

**Menghitung Semua Jarak Core Point**

Setelah menentukan core point secara acak, tahap selanjutnya adalah menghitung semua jarak titik terhadap Xp dengan Persamaan berikut (1)

$$d_{pq} = \|x_p - x_q\| = \sqrt{\sum_{k=1}^{n} |x_p x_{qk}|} \qquad (1)$$

**Menentukan Titik yang Dapat Dijangkau Core Point**

Setelah core point ditentukan, langkah selanjutnya adalah menentukan titik-titik yang dapat dijangkau terhadap Xp yang berada dalam dcore(Xp). Jika banyaknya titik yang density-reachable terhadap xp lebih dari atau sama dengan Mpts maka titik xp menjadi core point sehingga cluster terbentuk dan dilanjutkan ke titik lain disekitarnya yang merupakan titik yang density-reachable terhadap xp. Jika titik yang density-reachable juga merupakan core point maka cluster yang terbentuk merupakan cluster yang sama dengan cluster untuk core point sebelumnya. Namun jika titik yang dapat dijangkau tersebut adalah border point maka proses dilanjutkan ke titik selain

**Evaluasi Cluster**

Setelah bereksperimen dengan berbagai inisialiasi parameter, langkah selanjutnya adalah validasi untuk melihat kualitas sebuah kelompok untuk mecari parameter yang paling optimal dengan metode silhouette coefficient, metode silhouette coefficient yang menggabungkan cohesion dan separation adalah salah satu cara untuk melihat hasil clustering.

Dengan menggunakan Persamaan (2) dapat menemukan rata-rata data ke-p dengan semua data di cluster yang sama.

$$a_p = \frac{1}{n-1} \, \mathrm{d}_{p,q}, q \neq p \qquad (2)$$

dimana p = 0, 1, 2, …,n.

Persamaan (3) untuk menghitung jarak rata-rata data ke-p dengan semua data yang berbeda pada cluster yang berbeda dan diambil nilai terkecilnya

$$b_p = min\{d_p(p)\}, q \neq p \qquad (3)$$

dengan rumus jarak suatu data ke – p dengan semua data pada cluster yang berbeda adalah

$$d_p = 1\frac{1}{n_i}\sum_{q=1}^{n_i} d_{p,q} \qquad (4)$$

Menghitung nilai Silhoutte Coefficient menggunakan Persamaan (5)

$$SC_1(p) = \frac{b_p - a_p}{mac(a_p b_p)}, p = 1,2,\dots,n \qquad (5)$$

Nilai SC dari sebuah cluster (SC2(i)) didapatkan dengan menghitung rata-rata nilai SC1(p) semua data yang bergabung dalam cluster tersebut dengan menggunakan Persamaan (6)

$$SC_2(i) = \frac{1}{n_i} \sum_{x_{p \in C_i}} SC_1(p) \qquad (6)$$

Setelah itu nilai SC global didapatkan dengan menghitung rata-rata nilai SC2(p) semua data yang bergabung dalam cluster tersebut menggunakan Persamaan (7) berikut

$$SC = \frac{\sum_{i=0}^{c} n_i sc_2(p)}{\sum_{i=0}^{c} n_i} \quad (7)$$

dimana

$a_p$: Rata-rata jarak data ke-p dengan semua data pada satu cluster yang sama

$b_p$: Rata-rata jarak data ke-p dengan semua data pada cluster yang berbeda

$SC_1(p)$: Nilai Silhouette Coefficient pada data ke - p

$SC_2(i)$: Nilai Silhouette Coefficient pada data cluster ke - i

$X_p$: Data pengamatan ke- p

$C_i$: Cluster ke - i

$n_i$: Jumlah data dalam cluster ke i

$C$: Jumlah Clustter

## HASIL DAN PEMBAHASAN

### Analisis Data

Mengumpulkan dataset adalah langkah pertama dalam menjalankan eksperimen. Selanjutnya, data dibersihkan. Pada tahap kedua, data kategorikal akan ditransformasi menjadi data numerikal agar dapat dipahami oleh algoritma HDBSCAN.

Pembersihan data adalah langkah penting dalam proses pra-pemroresan data. Kolom yang tidak memiliki nilai akan dihapus, selain menghapus kolom kosong, data kategorikal akan diubah menjadi data numerikal karena algoritma HDBSCAN hanya bisa memproses data numerikal.

### Perancangan Tahapan Penelitian

Perancangan tahapan penelitian untuk mengimplementasikan klasterisasi data penjualan toko perak J-Masus dengan menggunakan algoritma HDBSCAN sangat diperlukan. Tahapan-tahapan ini meliputi pengumpulan data, pra-pemrosesan data, implementasi algoritma HDBSCAN, dan evaluasi hasil. Berikut ini adalah penjelasan dari masing-masing tahapan tersebut.

### Perolehan Data

Dataset yang digunakan dalam penelitian ini adalah data penjualan toko perak J-Maskus. Dataset didapatkan dengan meminta langsung ke pemilik toko perak J-Maskus yang terdiri dari 8 kolom dengan jumlah 5765 data. Kolom tersebut terdiri dari hari, tanggal, barang, ukuran, model, warna, jumlah dan harga.

### Deskripsi Atribut Data

Sebelum memulai tahapan data mining, perlu dilakukan langkah untuk menjelaskan setiap atribut yang terdapat dalam dataset yang telah diperoleh. Langkah ini bertujuan untuk mengidentifikasi atribut-atribut yang relevan dalam mengelompokkan data penjualan. Informasi atribut dari dataset yang digunakan dalam penelitian ini dapat dilihat pada tabel 1.

Tabel 1. Deskripsi Atribut Data

| No | Atribut | Deskripsi |
|---|---|---|
| 1. | Hari | Nama hari barang terjual |
| 2. | Tanggal | Tanggal dari hari barang terjual |
| 3. | Barang | Nama perhiasan yang terjual |
| 4. | Ukuran | Ukuran barang |
| 5. | Model | Model barang |
| 6. | Warna | Warna barang |
| 7. | Jumlah | Jumlah berat barang |
| 8. | Harga | Harga barang |

### Data Penjualan Toko Perak J-Maskus

Berikut ini adalah contoh data penjualan toko perak yang digunakan dalam penelitian ini. Dataset mencangkuo informasi mengenai hari, tanggal, barang, ukuran, model, warna, jumlah dan harga. Rincian data ini dapat dilihat pada tabel 2.

Tabel 2. Data Penjualan Toko Perak J-Maskus

| Hari | Tanggal | Barang | Ukuran | Model | Warna | Jumlah | Harga |
|---|---|---|---|---|---|---|---|
| Selasa | 1/1/2019 | Gelang Kaki | All in | Rante | Putih | 3.7 | Rp 134,555 |
| Selasa | 1/1/2019 | Cincin | Anak | Biasa | Rodium | 5.7 | Rp 263,103 |
| Rabu | 2/1/2019 | Cincin | Anak | Biasa | Rodium | 29.8 | Rp 274,839 |
| Rabu | 2/1/2019 | Kalung | Dewasa | Polos | Rodium | 10 | Rp 110,102 |
| Rabu | 2/1/2019 | Kalung | Dewasa | Koye | Putih | 11.9 | Rp 551,608 |

| Hari | Tanggal | Barang | Ukuran | Model | Warna | Jumlah | Harga |
|---|---|---|---|---|---|---|---|
| Rabu | 2/1/2019 | Liontin | Dewasa | Warna | Kuning | 23.1 | Rp 696,165 |
| Rabu | 2/1/2019 | Liontin | Dewasa | Polos | Kuning | 22.7 | Rp 594,660 |
| Rabu | 2/1/2019 | Cincin | All in | Biasa | Rodium | 25.4 | Rp 477,557 |
| … | … | … | … | … | … | … | … |
| Senin | 30/12/2024 | Liontin | All in | Polos | Rodium | 18.9 | Rp 107,700 |
| Senin | 30/12/2024 | Kalung | Anak | Panjang | Putih | 27.3 | Rp 100,223 |
| Selasa | 31/12/2024 | Kalung | Anak | Koye | Putih | 4 | Rp 566,809 |
| Selasa | 31/12/2024 | Kalung | All in | Polos | Rodium | 22.5 | Rp 625,246 |

**PraProcessing Data**

Tahapan ini bertujuan untuk mempersiapkan data sebelum proses klasterisasi dilakukan. Pra-pemrosesan data melibatkan dua langkah utama yaitu pembersihan data dan transformasi data. Pra-pemrosesan data adalah serangkaian langkah yang dilakukan untuk mempersiapkan data sebelum digunakan dalam analisis pengelompokkan dengan algoritma HDBSCAN.

**Data Cleaning**

Pada langkah ini, perhatian utamanya adalah menangani masalah seperti baris yang kosong. Dalam penelitian ini dengan mengguanakan algoritma HDBSCAN didapatkan hasil yang tidak ada nilai pada atribut dalam dataset yang akan digunakan. Nilai kosong dari setiap atribut dapat dilihat pada gambar 2.

```
Jumlah nilai kosong per kolom:
Hari       0
Tanggal    0
Barang     0
Ukuran     0
Model      0
Warna      0
Jumlah     0
Harga      0
dtype: int64
```

Gambar 2. Nilai kosong

**Data Selection**

Yang dilakukan pada tahapan ini adalah memilih atribut yang akan digunakan untuk proses clustering. Atribut yang akan digunakan adalah barang, ukuran, model, warna dan jumlah. Atribut yang akan digunakan dapat dilihat pada tabel 3.

Tabel 3. Data Selection

| Barang | Ukuran | Model | Warna | Jumlah |
|---|---|---|---|---|
| Gelang Kaki | All in | Rante | Putih | 3.7 |
| Cincin | Anak | Biasa | Rodium | 5.7 |
| Cincin | Anak | Biasa | Rodium | 29.8 |
| Kalung | Dewasa | Polos | Rodium | 10 |
| Kalung | Dewasa | Koye | Putih | 11.9 |
| Liontin | Dewasa | Warna | Kuning | 23.1 |
| Liontin | Dewasa | Polos | Kuning | 22.7 |
| Cincin | All in | Biasa | Rodium | 25.4 |
| … | … | … | … | … |
| Liontin | All in | Polos | Rodium | 18.9 |
| Kalung | Anak | Panjang | Putih | 27.3 |
| Kalung | Anak | Koye | Putih | 4 |
| Kalung | All in | Polos | Rodium | 22.5 |

**Transformasi Data**

Setelah data dibersihkan dan atribut sudah dipilih langkah selanjutnya yaitu transformasi data. Fokus utama pada langkah ini yaitu mengubah data kategorikal menjadi numerikal, karena algoritma HDBSCAN hanya bisa memproses data numerik. Atribut yang akan diubah adalah barang, ukuran, model dan warna. Atribut yang sudah diubah menjadi numerik dapat dilihat pada tabel 4

Tabel 4. Transformasi Data

| Barang | Ukuran | Model | Warna | Jumlah |
|---|---|---|---|---|
| 2 | 0 | 8 | 1 | 3.7 |
| 1 | 1 | 0 | 2 | 5.7 |
| 1 | 1 | 0 | 2 | 29.8 |
| 4 | 2 | 7 | 2 | 10 |
| 4 | 2 | 4 | 1 | 11.9 |
| 5 | 2 | 10 | 0 | 23.1 |
| 5 | 2 | 7 | 0 | 22.7 |
| 1 | 0 | 0 | 2 | 25.4 |
| … | … | … | … | … |
| 5 | 0 | 7 | 2 | 18.9 |
| 4 | 1 | 6 | 1 | 27.3 |
| 4 | 1 | 4 | 1 | 4 |
| 4 | 0 | 7 | 2 | 22.5 |

**Implementasi Algoritma HDBSCAN**

Setelah data siap digunakan, proses selanjutnya adalah melakukan klasterisasi menggunakan algoritma HDBSCAN. Algoritma ini menggunakan dua parameter utama, yaitu min_samples yang merepresentasikan jarak kepadatan antar titik dalam suatu cluster, dan min_cluster_size yang menunjukkan jumlah minimum titik data yang harus ada dalam satu cluster. Pemilihan nilai parameter ini sangat penting karena dapat memengaruhi hasil akhir dari proses pengelompokan data. Pada perhitungan manual yang dilakukan akan menggunakan 8 sampel dengan input nilai min_sample = 5 min_cluster_size = 5 dengan menggunakan data yang sudah melalui proses preprocessing data. Tabel 5 adalah sample data perhitungan manual HDBSCAN.

Tabel 5. Implementsai Perhitungan Manual

| No | Barang | Ukuran | Model | Warna | Jumlah |
|---|---|---|---|---|---|
| 1 | 2 | 0 | 8 | 1 | 3.7 |
| 2 | 1 | 1 | 0 | 2 | 5.7 |
| 3 | 1 | 1 | 0 | 2 | 29.8 |
| 4 | 4 | 2 | 7 | 2 | 10 |
| 5 | 4 | 2 | 4 | 1 | 11.9 |
| 6 | 5 | 2 | 10 | 0 | 23.1 |
| 7 | 5 | 2 | 7 | 0 | 22.7 |
| 8 | 1 | 0 | 0 | 2 | 25.4 |

Titik 1 ke Titik 2:

$$d_(12) = \sqrt{((2.0 - 1.0)^2 + (0.0 - 1.0)^2 + (8.0 - 0.0)^2 + (1.0 - 2.0)^2 + (3.7 - 5.7)^2)}$$
$$= \sqrt{(1 + 1 + 64 + 1 + 4)}$$
$$= \sqrt{71.00}$$
$$\approx 8.426$$

Titik 1 ke Titik 3:

$$d_(13) = \sqrt{((2.0 - 1.0)^2 + (0.0 - 1.0)^2 + (8.0 - 0.0)^2 + (1.0 - 2.0)^2 + (3.7 - 29.8)^2)}$$
$$= \sqrt{(1 + 1 + 64 + 1 + 681.21)}$$
$$= \sqrt{748.21}$$
$$\approx 27.353$$

Titik 1 ke Titik 4:

$$d_(14) = \sqrt{((2.0 - 4.0)^2 + (0.0 - 2.0)^2 + (8.0 - 7.0)^2 + (1.0 - 2.0)^2 + (3.7 - 10.0)^2)}$$
$$= \sqrt{(4 + 4 + 1 + 1 + 39.69)}$$
$$= \sqrt{49.69}$$
$$\approx 7.049$$

Titik 1 ke Titik 5:

$$d_(15) = \sqrt{((2.0 - 4.0)^2 + (0.0 - 2.0)^2 + (8.0 - 4.0)^2 + (1.0 - 1.0)^2 + (3.7 - 11.9)^2)}$$

$= \sqrt{(4 + 4 + 16 + 0 + 67.24)}$
$= \sqrt{91.24}$
$\approx 9.552$

Titik 1 ke Titik 6:

$d_(16) = \sqrt{((2.0 - 5.0)^2 + (0.0 - 2.0)^2 + (8.0 - 10.0)^2 + (1.0 - 0.0)^2 + (3.7 - 23.1)^2)}$
$= \sqrt{(9 + 4 + 4 + 1 + 376.36)}$
$= \sqrt{394.36}$
$\approx 19.858$

Titik 1 ke Titik 7:

$d_(17) = \sqrt{((2.0 - 5.0)^2 + (0.0 - 2.0)^2 + (8.0 - 7.0)^2 + (1.0 - 0.0)^2 + (3.7 - 22.7)^2)}$
$= \sqrt{(9 + 4 + 1 + 1 + 361)}$
$= \sqrt{376.00}$
$\approx 19.391$

Titik 1 ke Titik 8:

$d_(18) = \sqrt{((2.0 - 1.0)^2 + (0.0 - 0.0)^2 + (8.0 - 0.0)^2 + (1.0 - 2.0)^2 + (3.7 - 25.4)^2)}$
$= \sqrt{(1 + 0 + 64 + 1 + 470.89)}$
$= \sqrt{536.89}$
$\approx 23.171$

Titik 2 ke Titik 3:

$d_(23) = \sqrt{((1.0 - 1.0)^2 + (1.0 - 1.0)^2 + (0.0 - 0.0)^2 + (2.0 - 2.0)^2 + (5.7 - 29.8)^2)}$
$= \sqrt{(0 + 0 + 0 + 0 + 580.81)}$
$= \sqrt{580.81}$
$\approx 24.100$

Titik 2 ke Titik 4:

$d_(24) = \sqrt{((1.0 - 4.0)^2 + (1.0 - 2.0)^2 + (0.0 - 7.0)^2 + (2.0 - 2.0)^2 + (5.7 - 10.0)^2)}$
$= \sqrt{(9 + 1 + 49 + 0 + 18.49)}$
$= \sqrt{77.49}$
$\approx 8.803$

Titik 2 ke Titik 5:

$d_(25) = \sqrt{((1.0 - 4.0)^2 + (1.0 - 2.0)^2 + (0.0 - 4.0)^2 + (2.0 - 1.0)^2 + (5.7 - 11.9)^2)}$
$= \sqrt{(9 + 1 + 16 + 1 + 38.44)}$
$= \sqrt{65.44}$
$\approx 8.089$

Titik 2 ke Titik 6:

$d_(26) = \sqrt{((1.0 - 5.0)^2 + (1.0 - 2.0)^2 + (0.0 - 10.0)^2 + (2.0 - 0.0)^2 + (5.7 - 23.1)^2)}$
$= \sqrt{(16 + 1 + 100 + 4 + 302.76)}$
$= \sqrt{423.76}$
$\approx 20.585$

Titik 2 ke Titik 7:

$d_(27) = \sqrt{((1.0 - 5.0)^2 + (1.0 - 2.0)^2 + (0.0 - 7.0)^2 + (2.0 - 0.0)^2 + (5.7 - 22.7)^2)}$
$= \sqrt{(16 + 1 + 49 + 4 + 289)}$
$= \sqrt{359.00}$
$\approx 18.947$

Titik 2 ke Titik 8:

$d_(28) = \sqrt{((1.0 - 1.0)^2 + (1.0 - 0.0)^2 + (0.0 - 0.0)^2 + (2.0 - 2.0)^2 + (5.7 - 25.4)^2)}$
$= \sqrt{(0 + 1 + 0 + 0 + 388.09)}$
$= \sqrt{389.09}$
$\approx 19.725$

Titik 3 ke Titik 4:

$d_(34) = \sqrt{((1.0 - 4.0)^2 + (1.0 - 2.0)^2 + (0.0 - 7.0)^2 + (2.0 - 2.0)^2 + (29.8 - 10.0)^2)}$
$= \sqrt{(9 + 1 + 49 + 0 + 392.04)}$
$= \sqrt{451.04}$

$\approx 21.238$

Titik 3 ke Titik 5:

$d_(35) = \sqrt{((1.0 - 4.0)^2 + (1.0 - 2.0)^2 + (0.0 - 4.0)^2 + (2.0 - 1.0)^2 + (29.8 - 11.9)^2)}$
$= \sqrt{(9 + 1 + 16 + 1 + 320.41)}$
$= \sqrt{347.41}$
$\approx 18.639$

Titik 3 ke Titik 6:

$d_(36) = \sqrt{((1.0 - 5.0)^2 + (1.0 - 2.0)^2 + (0.0 - 10.0)^2 + (2.0 - 0.0)^2 + (29.8 - 23.1)^2)}$
$= \sqrt{(16 + 1 + 100 + 4 + 44.89)}$
$= \sqrt{165.89}$
$\approx 12.880$

Titik 3 ke Titik 7:

$d_(37) = \sqrt{((1.0 - 5.0)^2 + (1.0 - 2.0)^2 + (0.0 - 7.0)^2 + (2.0 - 0.0)^2 + (29.8 - 22.7)^2)}$
$= \sqrt{(16 + 1 + 49 + 4 + 50.41)}$
$= \sqrt{120.41}$
$\approx 10.973$

Titik 3 ke Titik 8:

$d_(38) = \sqrt{((1.0 - 1.0)^2 + (1.0 - 0.0)^2 + (0.0 - 0.0)^2 + (2.0 - 2.0)^2 + (29.8 - 25.4)^2)}$
$= \sqrt{(0 + 1 + 0 + 0 + 19.36)}$
$= \sqrt{20.36}$
$\approx 4.512$

Titik 4 ke Titik 5:

$d_(45) = \sqrt{((4.0 - 4.0)^2 + (2.0 - 2.0)^2 + (7.0 - 4.0)^2 + (2.0 - 1.0)^2 + (10.0 - 11.9)^2)}$
$= \sqrt{(0 + 0 + 9 + 1 + 3.61)}$
$= \sqrt{13.61}$
$\approx 3.689$

Titik 4 ke Titik 6:

$d_(46) = \sqrt{((4.0 - 5.0)^2 + (2.0 - 2.0)^2 + (7.0 - 10.0)^2 + (2.0 - 0.0)^2 + (10.0 - 23.1)^2)}$
$= \sqrt{(1 + 0 + 9 + 4 + 171.61)}$
$= \sqrt{185.61}$
$\approx 13.624$

Titik 4 ke Titik 7:

$d_(47) = \sqrt{((4.0 - 5.0)^2 + (2.0 - 2.0)^2 + (7.0 - 7.0)^2 + (2.0 - 0.0)^2 + (10.0 - 22.7)^2)}$
$= \sqrt{(1 + 0 + 0 + 4 + 161.29)}$
$= \sqrt{166.29}$
$\approx 12.895$

Titik 4 ke Titik 8:

$d_(48) = \sqrt{((4.0 - 1.0)^2 + (2.0 - 0.0)^2 + (7.0 - 0.0)^2 + (2.0 - 2.0)^2 + (10.0 - 25.4)^2)}$
$= \sqrt{(9 + 4 + 49 + 0 + 237.16)}$
$= \sqrt{299.16}$
$\approx 17.296$

Titik 5 ke Titik 6:

$d_(56) = \sqrt{((4.0 - 5.0)^2 + (2.0 - 2.0)^2 + (4.0 - 10.0)^2 + (1.0 - 0.0)^2 + (11.9 - 23.1)^2)}$
$= \sqrt{(1 + 0 + 36 + 1 + 125.44)}$
$= \sqrt{163.44}$
$\approx 12.784$

Titik 5 ke Titik 7:

$d_(57) = \sqrt{((4.0 - 5.0)^2 + (2.0 - 2.0)^2 + (4.0 - 7.0)^2 + (1.0 - 0.0)^2 + (11.9 - 22.7)^2)}$
$= \sqrt{(1 + 0 + 9 + 1 + 116.64)}$
$= \sqrt{127.64}$
$\approx 11.298$

Titik 5 ke Titik 8:

$$d(58) = \sqrt{((4.0 - 1.0)^2 + (2.0 - 0.0)^2 + (4.0 - 0.0)^2 + (1.0 - 2.0)^2 + (11.9 - 25.4)^2)}$$
$$= \sqrt{(9 + 4 + 16 + 1 + 182.25)}$$
$$= \sqrt{212.25}$$
$$\approx 14.569$$

Titik 6 ke Titik 7:

$$d(67) = \sqrt{((5.0 - 5.0)^2 + (2.0 - 2.0)^2 + (10.0 - 7.0)^2 + (0.0 - 0.0)^2 + (23.1 - 22.7)^2)}$$
$$= \sqrt{(0 + 0 + 9 + 0 + 0.16)}$$
$$= \sqrt{9.16}$$
$$\approx 3.027$$

Titik 6 ke Titik 8:

$$d(68) = \sqrt{((5.0 - 1.0)^2 + (2.0 - 0.0)^2 + (10.0 - 0.0)^2 + (0.0 - 2.0)^2 + (23.1 - 25.4)^2)}$$
$$= \sqrt{(16 + 4 + 100 + 4 + 5.29)}$$
$$= \sqrt{129.29}$$
$$\approx 11.371$$

Titik 7 ke Titik 8:

$$d(78) = \sqrt{((5.0 - 1.0)^2 + (2.0 - 0.0)^2 + (7.0 - 0.0)^2 + (0.0 - 2.0)^2 + (22.7 - 25.4)^2)}$$
$$= \sqrt{(16 + 4 + 49 + 4 + 7.29)}$$
$$= \sqrt{80.29}$$
$$\approx 8.960$$

Hasil dari perhitungan manual Berikut versi paragraf dengan gaya bahasa yang lebih santai dan sesuai untuk tulisan mahasiswa: Titik-titik yang punya jarak yang pendek atau kecil menunjukkan bahwa data tersebut punya kemiripan dalam beberapa fitur, seperti jumlah barang, ukuran, model, warna, dan jumlah total. Karena mirip, titik-titik ini kemungkinan besar akan masuk ke dalam cluster yang sama. Hal ini sejalan dengan konsep pada algoritma HDBSCAN yang mengelompokkan data berdasarkan kemiripan. Sebaliknya, kalau jaraknya jauh, berarti datanya cukup berbeda satu sama lain. Titik-titik seperti ini kemungkinan akan masuk ke cluster yang berbeda, atau bahkan dianggap sebagai noise atau data yang menyimpang, apalagi kalau tidak punya cukup tetangga dekat sesuai dengan parameter min_samples yang ditentukan.

**Perancangan Perangkat Lunak**

Pada tahapan perancangan perangkat lunak, fokus utama berada pada bagaimana merancang aplikasi yang dapat membantu proses clustering data menggunakan algoritma HDBSCAN. Perancangan ini mencakup penentuan fitur-fitur yang akan digunakan, tampilan antarmuka user serta alur kerja dari input data hingga hasil clustering.

**Analisis Kebutuhan Fungsional**

Kebutuhan fungsional dalam aplikasi ini mencakup seluruh fungsi utama yang harus dimiliki agar aplikasi dapat berjalan sesuai tujuan. Fungsi utama yang paling penting adalah kemampuan untuk melakukan proses clustering menggunakan algoritma HDBSCAN, di mana user dapat memasukkan data dan menentukan parameter seperti min_samples dan min_cluster_size. Selain itu, aplikasi juga harus mampu menampilkan hasil clustering dalam bentuk visual seperti plot atau grafik, agar user lebih mudah memahami hasil pengelompokan data. Analisis kebutuhan fungsional dapat dilihat pada tabel 6.

Tabel 6. Kebutuhan Fungsional

| No | Kebutuhan Fungsional | Aktor | Deskripsi |
|---|---|---|---|
| 1. | Dataset | User | Pada bagian “load data” user diminta untuk memasukan data berbentuk csv. Setelah user memasukan data ke dalam aplikasi user dapat melihat data yang telah berhasil dimasukan. |
| 2. | Preprocessing Data | User | Setalah dataset berhasil ditampilkan, user diminta melakukan preprocessing data agar user data dapat diolah oleh aplikasi |
| 3 | Inisialisasi Parameter | User | Untuk mengolah data dengan algoritma HDBSCAN user akan diminta memasukan parameter HDBSCAN yaitu min_sample dan min_cluster_size |

**Analisis Kebutuhan Non Fungsional**

Analisis kebutuhan non-fungsional, yang merupakan langkah penting dalam perancangan perangkat lunak, memastikan bahwa sistem tidak hanya memenuhi tujuan fungsionalnya, tetapi juga beroperasi dengan efisien, andal, dan memberikan pengalaman user yang optimal. Komponen non-fungsional perangkat lunak termasuk performa, dan kemudahan penggunaan. Aplikasi ini membutuhkan pemrosesan data yang cepat, dan pengalaman antar muka yang mudah digunakan.

**Perangkat Keras**

Perangkat keras yang digunakan untuk mendukung sistem mencakup seperangkat komputer atau laptop dengan spesifikasi sebagai berikut :

a. Processor : AMD Ryzen 5 5400U
b. RAM : 8 GB
c. VGA : 4 GB
d. Penyimpanan: 512 GB

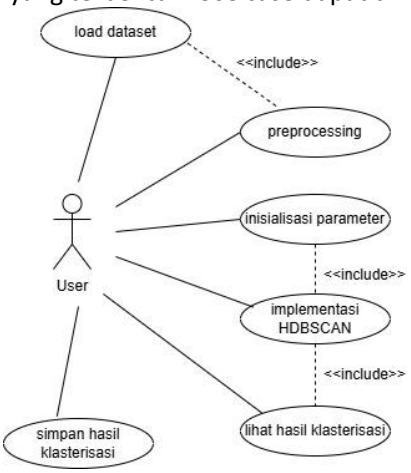
**Perangkat Lunak**

Perangkat lunak yang dibutuhkan untuk menjalankan sistem mencakup sebagai berikut :

Sistem Operasi : Windows 11
Pemrograman : Python
Teks Editor : Visual Studio Code

**Use Case Diagram**

Use case diagram menggambarkan interaksi aktor antara user dan fungsionalitas utama aplikasi. User pertama-tama akan menginput data yang akan digunakan untuk proses clustering. Setelah data dimasukkan, user dapat menentukan parameter penting dalam algoritma HDBSCAN, seperti min_cluster_size dan min_samples, untuk mengontrol bagaimana proses clustering akan dijalankan. Setelah parameter ditentukan, user dapat memulai proses clustering dengan menekan tombol untuk menjalankan algoritma.Setelah proses clustering selesa user dapat melihat hasilnya dalam bentuk visualisasi plot yang menunjukkan pengelompokan data, serta evaluasi skor yang memberikan gambaran tentang kualitas dan jumlah cluster yang terbentuk. Use case dapat dilihat pada gambar 3.1 berikut




Gambar 3. Use Case Diagram

Tabel 7 merupakan deskripsi dari use case diagram diatas

Tabel 7. Deskripsi Use Case Diagram

| No | Use Case | Deskripsi |
|---|---|---|
| 1 | Load dataset | User dapat mengunggah data dari file dataset berformat csv. |
| 2 | Preprocessing Data | User dapat melakukan preprocessing data, data akan dibersihkan dan ditransformasi menjadi data numerik |
| 3 | Inisialisasi Parameter | User dapat memasukan parameter min_cluster_size dan min_sample |

| No | Use Case | Deskripsi |
|---|---|---|
| 4 | Implementasi HDBSCAN | User dapat menjalankan proses klasterisasi menggunakan algoritma HDBSCAN |
| 5 | Lihat Hasil Klasterisasi | User dapat melihat visualisasi hasil clustering, melihat jumlah cluster yang terbentuk dan user dapat melihat skor hasil klasterisasi dengan menggunakan Silhouette Score. |
| 6 | Simpan Hasil Klasterisasi | User dapat menyimpan file hasil klasterisasi |

**Use Case Scenario**

Setiap skenario penggunaan yang ditampilkan dalam diagram skenario disebut skenario penggunaan. Situasi ini melibatkan interaksi user dengan sistem, yang mencakup tindakan user , respons sistem, dan alur logis yang terjadi selama proses. Dengan mencatat skenario use case secara menyeluruh, pengembang dapat memastikan bahwa setiap fungsi sistem dirancang sesuai dengan kebutuhan user dan keadaan yang diantisipasi. Selain itu, skenario ini berfungsi sebagai panduan untuk tahap implementasi dan pengujian perangkat lunak.

**Use Case Scenario Load Dataset**

Use Case Scenario Load dataset menjelaskan proses user dapat mengunggah file csv ke dalam sistem. Use Case Scenario Load data dapat ditemukan padatabel 8 berikut.

Tabel 8. Use Case Scenario Load Dataset

| | |
|---|---|
| ID | UC-1 |
| Nama | Mengunggah data csv |
| Deskripsi | Skenario ini menjelaskan user dapat mengunggah data csv ke dalam sistem. |
| Aktor | User |
| Trigger | Menekan tombol Load Dataset |
| Pre Condition | Dataset belum ditambahkan ke dalam sistem |
| Post Condition | Dataset berhasil ditambahkan kedalam sistem |
| Aksi Aktor | Reaksi Sistem |
| User masuk ke dalam aplikasi | |
| | Menampilkan halaman utama |
| Menekan tombol Load Dataset | |
| | Membuka file explorer |
| Memilih File dengan format csv | |
| | Membaca dan menampilkan data ke halaman utama |

**Use Case Scenario Preprocessing Data**

Usecase Scneario Preprocessing data menjelaskan bagaimana interaksi antara user dengan sistem dalam melakukan pengolahan dataset Use Case Scenario Preprocessing data dapat ditemukan pada tabel 9 berikut

Tabel 9. Use Case Scenario Preprocessing Data

| | |
|---|---|
| ID | UC-2 |
| Nama | Melakukan Preprocessing data |
| Deskripsi | Skenario ini menjelaskan user dapat mengolah data |
| Aktor | User |
| Trigger | Menekan tombol Preprocessing Data |
| Pre Condition | Dataset belum diolah |

| | |
|---|---|
| Post Condition | Dataset sudah sudah diolah |
| Aksi Aktor | Reaksi Sistem |
| User masuk ke dalam aplikasi | |
| | Menampilkan halaman utama |
| Memasukan Dataset | |
| | Membaca dan menampilkan dataset |
| Menekan tombol Preprocessing Data | |
| | Menampilkan dataset yang sudah diolah |

**Use Case Scenario Inisialisasi Parameter**

Use Case Scenario Inisialisasi Parameter menjelaskan tentang bagaimana interaksi user dengan sistem pada saat user menginisialisasi parameter min_cluster_size dan min_sample. Use case inisialisasi parameter dapat dilihat pada tabel 10 berikut.

Tabel 10. Use Case Scenario Inisialisasi Parameter

| | |
|---|---|
| **ID** | UC-3 |
| **Nama** | Menginisalisasi Parameter |
| **Deskripsi** | Skenario ini menjelaskan *user* melakukan inisialisasi parameter |
| **Aktor** | *User* |
| ***Trigger*** | Mengisi *min_sample* dan *min_cluster_size* |
| ***Pre Condition*** | Parameter *default min_sample = 5, min_cluster_size_5* |
| ***Post Condition*** | Parameter diisi oleh *user* |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. User masuk ke dalam aplikasi | |
| | 2. Menampilkan halaman utama |
| 3. Memasukan Dataset | |
| | 4. Membaca dan menampilkan dataset |
| 5. Mengisi *min_sample* dan *min_cluster_size* | |
| | 6. Menampilkan *min_sample* dan *min_cluster_size* yang diisi oleh *user* |

Use Case Scenario Implementasi HDBSCAN

Use Case Scenario Implementasi HDBSCAN menjelaskan tentang bagaimana interaksi user dengan sistem pada saat melakukan clustering dengan algoritma HDBSCAN. Use case Implementasi HDBSCAN dapat dilihat pada 11 berikut

Tabel 1. Use Case Scenario Implementasi HDBSCAN

| | |
|---|---|
| **ID** | UC-4 |
| **Nama** | Implementasi HDBSCAN |
| **Deskripsi** | Skenario ini menjelaskan *user* melakukan *clustering* dengan algoritma HDBSCAN |
| **Aktor** | *User* |
| ***Trigger*** | Menekan tombol jalankan HDSBCAN |
| ***Pre Condition*** | Clustering belum dilakukan |
| ***Post Condition*** | Clustering sudah dilakukan |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. User masuk ke dalam aplikasi | |
| | 2. Menampilkan halaman utama |

| | |
|---|---|
| 3. Memasukan Dataset | |
| | 4. Membaca dan menampilkan dataset |
| 5. Menginisialisasi Parameter | |
| 6. Menampilkan *min_sample* dan *min_cluster_size* yang diisi oleh *user* | |
| *7.* Memilih visualisasi hasil *clustering* | |
| | 8. Menampilkan visualisasi informasi hasil *clustering* |

**Use Case Scenario Simpan Hasil Klasterisasi**

Use case Scenario simpan hasil klasterisasi menjelaskan tentang bagaimana interaksi user dan sistem dalam menyimpan file hasil klasterisasi. Tabel 12 adalah use case scenario simpan hasil klasterisasi

Tabel 2. Use Case Scenario Simpan Hasil Klasterisasi

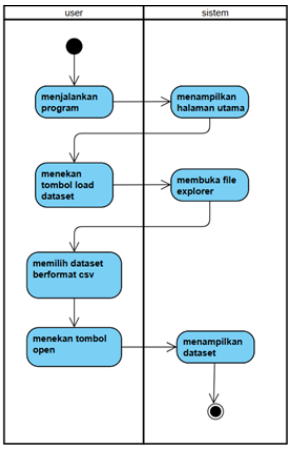
| | |
|---|---|
| ID | UC-5 |
| Nama | Simpan Hasil Clustering |
| Deskripsi | Skenario ini menjelaskan user menyimpan hasil clustering ke fromat csv |
| Aktor | User |
| Trigger | Menekan tombol Simpan Hasil |
| Pre Condition | Hasil belum disimpan |
| Post Condition | Hasil sudah disimpan |
| Aksi Aktor | Reaksi Sistem |
| User masuk ke dalam aplikasi | |
| | Menampilkan halaman utama |
| Memasukan Dataset | |
| | Membaca dan menampilkan dataset |
| Menginisialisasi Parameter | |
| Menampilkan min_sample dan min_cluster_size yang diisi oleh user | |
| Memilih visualisasi hasil clustering | |
| | Menampilkan visualisasi dan informasi hasil clustering |
| Menekan tombol simpan hasil | |
| Aksi Aktor | Reaksi Sistem |
| | Meembuka file explorer |
| Memasukan nama file | |
| Menekan tombol simpan | |
| | Menyimpan file |

**Activity Diagram**

Activity Diagram digunakan untuk menunjukkan alur proses utama. Alur proses ini mencakup input data, preprocessing data, klasterisasi data dengan algotitma HDBSCAN, dan menyimpan hasil. Diagram ini membantu memahami bagaimana setiap komponen sistem bekerja secara terstruktur dan memastikan bahwa alur kerja perangkat lunak dirancang dengan efisien. Aktivitas diagram juga berfungsi sebagai pedoman penting untuk implementasi dan pengujian perangkat lunak.

**Load Dataset**

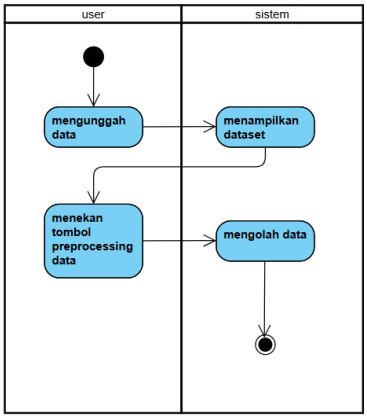
Activity diagram load dataset menggambarkan proses yang terjadi saat pengguna mengunggah dan menampilkan dataset. Activity diagram load dataset dapat dilihat pada gambar 4 berikut.




Gambar 4. Activity Diagram Load Dataset

**Preprocessing Data**

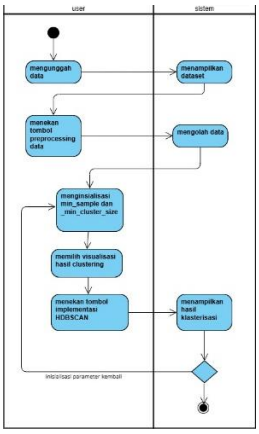
Activity diagram preprocessing data menggambarkan proses yang terjadi saat pengguna melakukan pengolahan data. Activity diagram preprocessing data dapat dilihat pada gambar 5 berikut.




Gambar 5. Activity Diagram Preprocessing Data

**Implementasi Algoritma HDBSCAN**

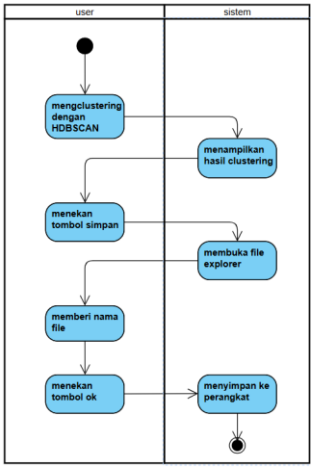
Activity diagram implementasi algoritma HDBSCAN menggambarkan proses yang terjadi saat pengguna mengimplementasikan algoritma HDBSCAN untuk clustering. Activity diagram implementasi algoritma HDBSCAN dapat dilihat pada gambar 6 berikut.




Gambar 6. Activity Diagram Implementasi HDBSCAN

**Menyimpan Hasil Clustering**

Activity diagram menyimpan hasil clustering menggambarkan proses yang terjadi saat pengguna menyimpan file hasil dari clustering. Activity diagram menyimpan hasil clustering dapat dilihat pada gambar 3.5 berikut.



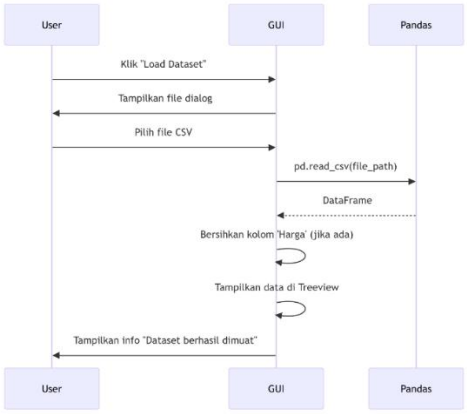

Gambar 7. Activity Diagram Menyimpan Hasil Clustering

**Sequence Diagram**

Sequence Diagram menggambarkan interaksi antar objek atau komponen dalam suatu sistem secara berurutan (sequence) berdasarkan alur waktu. Diagram ini fokus pada pesan yang dikirim dan diterima antara berbagai entitas, serta urutan kejadian dalam sebuah skenario tertentu

**Load Dataset**

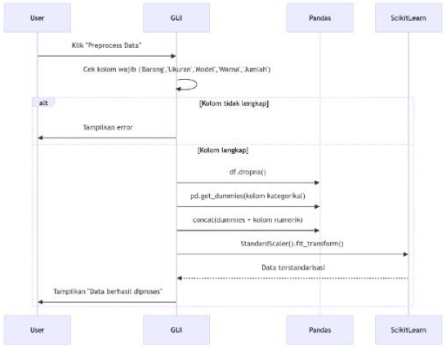
Proses Load Dataset dimulai ketika pengguna mengklik tombol "Load Dataset" pada antarmuka aplikasi. Sistem kemudian menampilkan dialog pemilihan file untuk memungkinkan pengguna memilih file CSV. Setelah file dipilih, aplikasi menggunakan library Pandas untuk membaca data (pd.read_csv). Jika dataset memiliki kolom b sistem membersihkan formatnya (menghilangkan 'Rp' dan koma) sebelum mengonversinya ke tipe data numerik. Data yang telah dibaca kemudian ditampilkan dalam Treeview untuk preview, dan pengguna mendapat notifikasi bahwa dataset berhasil dimuat beserta jumlah baris data. Gambar 3.6 adalah sequence diagram untu load dataset.




Gambar 8. Sequence Diagram Load Dataset

**Preprocessing Data**

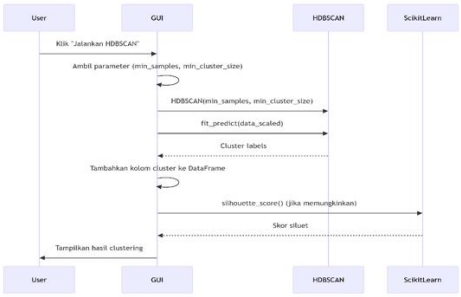
Proses Preprocessing Data dilakukan ketika pengguna menekan tombol "Preprocess Data". Pertama, sistem memeriksa kelengkapan kolom wajib ('Barang', 'Ukuran', 'Model', 'Warna', 'Jumlah'). Jika ada kolom yang tidak ditemukan, aplikasi menampilkan pesan error. Jika semua kolom tersedia, data dibersihkan dari nilai kosong (dropna), lalu kolom kategorikal di-encode menggunakan one-hot encoding (pd.get_dummies). Selanjutnya, data numerik (seperti 'Jumlah') digabungkan dengan data hasil encoding. Data kemudian distandarisasi menggunakan StandardScaler dari Scikit-Learn untuk memastikan konsistensi skala. Jika berhasil, pengguna mendapat notifikasi bahwa data siap untuk klasterisasi. Sequence diagram dari preprocessing data dapat dilihat pada gambar 9.




Gambar 9. Sequence Diagram Preprocessing Data

**Implementasi Algoritma HDBSCAN**

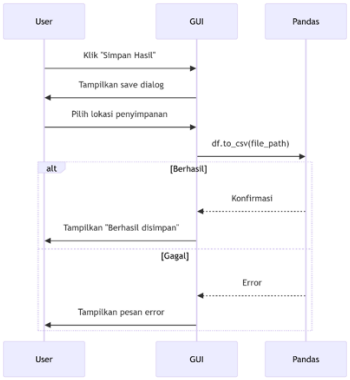
Proses Implementasi HDBSCAN dijalankan saat pengguna mengklik tombol "Jalankan HDBSCAN". Sistem mengambil parameter min_samples dan min_cluster_size yang diinput pengguna. Algoritma HDBSCAN dipanggil dengan parameter tersebut, lalu data yang telah diproses (data_scaled) digunakan untuk melakukan klasterisasi (fit_predict). Hasil cluster (label) ditambahkan ke DataFrame asli. Jika memungkinkan (jumlah cluster > 1), sistem menghitung Silhouette Score untuk mengukur kualitas cluster. Hasil cluster ditampilkan dalam Treeview, dan informasi seperti jumlah cluster, noise, serta silhouette score ditampilkan di antarmuka. Jika klasterisasi gagal (misalnya, terlalu sedikit data), sistem memberi tahu pengguna. Gambar dari sequence diagram impementasi algoritma HDBSCAN dapat dilihat pada gambar 10 berikut.




Gambar 10. Sequence Diagram Implementasi Algoritma HDBSCAN

**Menyimpan Hasil Clustering**

Proses Simpan Hasil Clustering dimulai ketika pengguna menekan tombol "Simpan Hasil". Sistem menampilkan dialog penyimpanan file untuk memilih lokasi dan nama file output. Jika pengguna menentukan lokasi, DataFrame yang berisi hasil cluster disimpan dalam format CSV menggunakan df.to_csv(). Jika penyimpanan berhasil, pengguna mendapat konfirmasi bahwa data telah tersimpan. Jika gagal (misalnya, lokasi tidak valid atau akses ditolak), sistem menampilkan pesan error. Proses ini memastikan pengguna dapat menyimpan hasil analisis untuk penggunaan lebih lanjut. Sequence diagram dari menyimpan hasil clustering dapat dilihat pada gambar 11 berikut.




Gambar 11. Seqiemce Diagram Menyimpan Hasil Clustering

**Perancangan Antar Muka**

Perancangan antarmuka adalah proses merancang tampilan dan interaksi antara user dengan suatu aplikasi agar mudah digunakan (usability), fungsional, dan estetis. Fokus utamanya adalah pada pengalaman pengguna (UX/User Experience) dan tampilan visual (UI/User Interface). Perancangan antar muka dapat dijadikan sebagai acuan untuk pembuatan aplikasi

**Antar Muka Dashboard**

Halaman Dashboard adalah tampilan awal program, terdapat 4 button yaitu load data, preprocessing data, implmentasi hdbscan dan simpan. Selain itu terdapat juga 2 texteditor untuk user menginputkan min_sample dan min_cluster_size. User juga dapat memilih visualisasi hasil clustering. Terdapat 2 frame untuk menampilkan data yang sudah diunggah oleh user dan untuk menampilkan visualisasi hasil clustering. Gambar 12 berikut merupakan peracangan dari halaman utama




Gambar 12. Perancangan Antar Muka Dashboard

**Antar Muka Load Data**

Halaman load data merupakan halaman dashboard yang berisi data yang sudah diunggah oleh user. Data akan ditampilkan pada halaman dashboard seperti pada gambar 3.11 berikut.



Gambar 13. Perancangan Antar Muka Load Data

**Antar Muka Preprocessing Data**

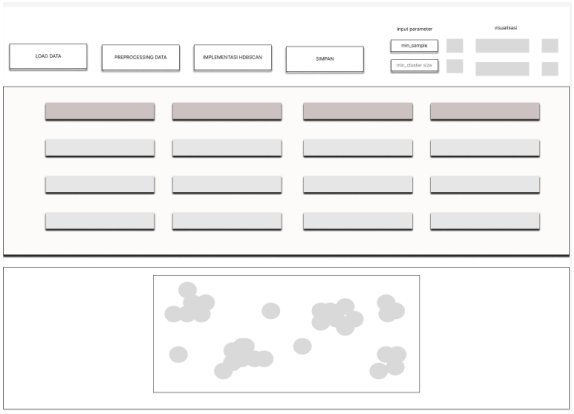
Setelah data diunggah, atribut yang akan dipakai yang memiliki tipe kategorikal akan diubah menjadi numerikal. Antar muka preprocessing data dapat dilihat pada gambar 14.



Gambar 14. Perancangan Antar Muka Preprocessing Data

**Antar Muka Implementasi Algoritma HDBSCAN**

Antar muka implementasi HDBSCAN akan menampilkan visualisasi cluster dan informasi mengenai shilouette score, jumlah cluster yang terbentuk, jumlah noise. Berikut adalah gambar 3.13 antar muka implementasi algoritma HDBSCAN.




Gambar 15. Perancangan Antar Muka Implementasi Algoritma HDBSCAN

## KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan algoritma HDBSCAN berhasil mengelompokkan data penjualan Toko Perak J-Maskus berdasarkan atribut seperti barang, ukuran, model, warna, dan jumlah terjual. Dari 7 eksperimen parameter, parameter paling optimal berdasarkan silhouette score diperoleh pada nilai min_samples=5 dan min_cluster_size=5, dengan silhouette score sebesar 0.6507, yang menunjukkan struktur pengelompokan termasuk struktur tingkat menengah (intermediate structure). Hasil klasterisasi menghasilkan 206 cluster, mencerminkan variasi pola penjualan yang kompleks. Cluster 33 yang termasuk kedalam cluster 33 adalah cincin dengan ukuran all in, model biasa dan warna rodium. Cluster 33 merupakan cluster paling laku, dan cluster -1 atau outlier merupakan produk yang paling tidak laku. Produk yang termasuk kedalam outlier adalah liontin ukuran dewasa dengan model polos dan warna kuning, gelang tangan ukuran dewasa dengan model mainan dan warna kuning, cincin dengan ukuran all in model biasa berwarna putih, gelang kaki anak bermodel golong dengan warna rodium. Beberapa cluster menunjukkan karakteristik unik yang menarik untuk diteliti lebih lanjut. Cluster 4, berisi Cincin Dewasa Biasa Putih, termasuk dalam kategori premium dengan harga sering di atas Rp500.000, Cluster 117, Liontin Anak Polos Kuning, menawarkan produk dengan harga lebih terjangkau dan warna cerah yang sesuai untuk pasar anak-anak.

## DAFTAR PUSTAKA

Agustia, Y. P., & Suryani, E. (2018). Pengaruh Ukuran Perusahaan, Umur Perusahaan, Leverage, Dan Profitabilitas Terhadap Manajemen Laba (Studi Pada Perusahaan Pertambangan yang Terdaftar di Bursa Efek Indonesia Periode 2014-2016). *Jurnal ASET (Akuntansi Riset)*, *10*(1). https://doi.org/10.17509/jaset.v10i1.12571

Bani Saleh FTID, U., Bani Saleh, U., Nur, M., Irwan, S., & Pranata, R. (2023). Klasterisasi Penjualan Produk Fantech Menggunakan Metode K-Means Pada Toko Giben Advance Premium. *JURNAL TRIDI Teknologi Informatika & Komputer*, *1*(1), 76–90.

Dewantari, N. L. S., Cipta, W., & Susila, G. P. A. J. (2020). PENGARUH UKURAN PERUSAHAAN DAN LEVERAGE SERTA PROFITABILITAS TERHADAP NILAI PERUSAHAAN PADA PERUSAHAAN FOOD AND BEVERAGES DI BEI. *Prospek: Jurnal Manajemen Dan Bisnis*, *1*(2). https://doi.org/10.23887/pjmb.v1i2.23157

Fitrianto, N. (2024). *PERBANDINGAN ALGORITMA HDBSCAN DAN AGGLOMERATIVE HIERARCHICAL CLUSTERING DALAM KLASTERISASI PADA DATA YANG MENGANDUNG*. *08*(02), 122–135.

Hamid Mughal, M. J. (2018). Data mining: Web data mining techniques, tools and algorithms: An overview. *International Journal of Advanced Computer Science and Applications*, *9*(6). https://doi.org/10.14569/IJACSA.2018.090630

Handayani, F. D., & Isnaini Rosyida. (2023). Clustering Review Pengguna Aplikasi Zenius pada Layanan Google Play Store Menggunakan Metode DBSCAN dan HDBSCAN. *Emerging Statistics and Data Science Journal*, *1*(2), 178–191. https://doi.org/10.20885/esds.vol1.iss.2.art19

Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., & Yeganegi, M. R. (2020). Text mining in big data analytics. *Big Data and Cognitive Computing*, *4*(1). https://doi.org/10.3390/bdcc4010001

*INTISARI CLUSTERING HIERARCHICAL DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE ( HDBSCAN ) UNTUK DATA SPASIAL ( STUDI KASUS : PEMETAAN KAWASAN WISATA TERPADU DI KABUPATEN GUNUNGKIDUL , DAERAH ISTIMEWA YOGYAKARTA ) Oleh LUTHFI SYAFI ABDILLAH* . (2024).

Lestari, A. F., & Hafiz, M. (2020). PENERAPAN ALGORITMA APRIORI PADA DATA PENJUALAN BARBAR WAREHOUSE. *INOVTEK Polbeng - Seri Informatika*, *5*(1). https://doi.org/10.35314/isi.v5i1.1317

Liu, J., Sun, S., & Chen, C. (2021). Big data Analysis of Regional Meteorological Observation Based : on Hierarchical Density Clustering Algorithm HDBSCAN. *Proceedings - 2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology, AINIT 2021*, 111–116. https://doi.org/10.1109/AINIT54228.2021.00032

Nasari, F., & Darma, S. (2015). *Seminar Nasional Teknologi Informasi dan Multimedia 2015 PENERAPAN K-MEANS CLUSTERING PADA DATA PENERIMAAN MAHASISWA BARU (STUDI KASUS : UNIVERSITAS POTENSI UTAMA)*. 6–8.

Rosmayati, I., Wahyuningsih, W., Harahap, E. F., & Hanifah, H. S. (2023). Implementasi Data Mining pada Penjualan Kopi Menggunakan Algoritma Apriori. *Jurnal Algoritma*, *20*(1). https://doi.org/10.33364/algoritma/v.20-1.1259

Shah, N., & Shah, K. (2023). Introduction to Data Mining. In *Practical Data Mining Techniques and Applications*. https://doi.org/10.1201/9781003390220-1

Sibarani, A. J. P. (2020). Implementasi Data Mining Menggunakan Algoritma Apriori Untuk Meningkatkan Pola Penjualan Obat. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *7*(2). https://doi.org/10.35957/jatisi.v7i2.195

Stewart, G., & Al-Khassaweneh, M. (2022). An Implementation of the HDBSCAN* Clustering Algorithm. *Applied Sciences (Switzerland)*, *12*(5), 1–21. https://doi.org/10.3390/app12052405

Sulistiyowati, R., Legis, S., & Krisna, D. Y. (2023). Strategi Penjualan Dengan Mengetahui Pola Pembelian Pelanggan Terhadap Data Penjualan Menggunakan Algoritma Apriori. *Jurnal Informatika & Komputasi*, *17*(1), 12–27.

Takdirillah, R. (2020). Penerapan Data Mining Menggunakan Algoritma Apriori Terhadap Data Transaksi Sebagai Pendukung Informasi Strategi Penjualan. *Edumatic : Jurnal Pendidikan Informatika*, *4*(1). https://doi.org/10.29408/edumatic.v4i1.2081

Wahyuni, N. A., Hayati, M. N., & Rizki, A. (2021). Metode Hierarchical Density-Based Spatial Clustering of Application with Noise ( HDBSCAN ) Pada Wilayah Desa / Kelurahan Tertinggal di Kabupaten Kutai Kartanegara ( Studi Kasus : Data Hasil Pendataan Potensi Desa ( PODES ) Tahun 2018 ) Hierarchical Densit. *Eksponensial*, *12*(1), 47–52.