

**ANALISIS KLASTERISASI PRODUK TOKOPEDIA  
DENGAN ALGORITMA *K-MEANS* DAN *K-MEDOIDS***

S K R I P S I



**MUHAMMAD RAIHAN MALIK**

**F1E121145**

**PROGRAM STUDI SISTEM INFORMASI  
JURUSAN TEKNIK ELEKTRO DAN INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS JAMBI**

**2025**

**SURAT PERNYATAAN**

Dengan ini saya menyatakan bahwa skripsi ini benar-benar karya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Tanda tangan yang tertera dalam halaman pengesahan adalah asli. Jika tidak asli, saya siap menerima sanksi sesuai dengan peraturan yang berlaku.

Jambi, 19 November 2025  
Yang menyatakan

MUHAMMAD RAIHAN MALIK  
F1E121145

# **ANALISIS KLASTERISASI PRODUK TOKOPEDIA DENGAN ALGORITMA *K-MEANS* DAN *K-MEDOIDS***

**S K R I P S I**

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana komputer  
pada Program Studi Sistem Informasi



**MUHAMMAD RAIHAN MALIK**

**F1E121145**

**PROGRAM STUDI SISTEM INFORMASI  
JURUSAN TEKNIK ELEKTRO DAN INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS JAMBI**

**2025**

## **PENGESAHAN**

Skripsi dengan judul "**Analisis Klasterisasi Produk Tokopedia dengan Algoritma K-means dan K-medoids**" yang disusun oleh **Muhammad Raihan Malik, NIM: F1E121145** telah dipertahankan di depan tim penguji pada tanggal 19 November 2025 dan dinyatakan lulus.

Susunan Tim Penguji :

Ketua	:	Pradita Eko Prasetyo Utomo, S.Pd., M.Cs.
Sekretaris	:	Benedika Ferdian Hutabarat, S.Komp., M.Kom.
Anggota	:	1. Ulfa Khaira, S.Komp., M.Kom. 2. Akhiyar Waladi, S.Komp., M.Kom.

Disetujui:

Pembimbing Utama

Pembimbing Pendamping

Pradita Eko Prasetyo Utomo, S.Pd., M.Cs.  
NIP. 198710282019031010

Benedika Ferdian Hutabarat, S.Komp., M.Kom.  
NIP. 198702022019031007

Diketahui:

Dekan  
Fakultas Sains dan Teknologi

Ketua Jurusan  
Teknik Elektro dan Informatika

Drs. Jefri Marzal, M.Sc., D.I.T.  
NIP. 196806021993031004

Edi Saputra, S.T., M.Sc.  
NIP. 198501082015041003

## RINGKASAN

Pertumbuhan *e-commerce* yang pesat di Indonesia mendorong kebutuhan akan strategi segmentasi produk yang efektif. Tokopedia sebagai salah satu platform terbesar menyediakan ribuan produk dari berbagai kategori, sehingga diperlukan analisis berbasis data untuk memahami pola karakteristik produk secara lebih terarah. Penelitian ini menerapkan algoritma *K-Means* dan *K-Medoids* untuk mengelompokkan produk Tokopedia berdasarkan atribut numerik *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*, menggunakan *dataset* PRDECT-ID (*Product Reviews Dataset for Emotion Classification Tasks – Indonesian*) dari Mendeley Data. Tahapan penelitian meliputi pembersihan data (*data cleaning*), penanganan *outlier* menggunakan metode *IQR*, normalisasi dengan *Min-Max Scaling*, serta penentuan jumlah klaster optimal melalui metode Elbow yang menghasilkan  $K = 2$ . Implementasi algoritma dilakukan secara iteratif hingga konvergen, kemudian dilakukan evaluasi kuantitatif menggunakan *Davies-Bouldin Index (DBI)* dan *Silhouette Score* yang menunjukkan bahwa *K-Means* lebih unggul dibandingkan *K-Medoids*, dengan nilai *DBI* 0,5717 dan *Silhouette Score* 0,6012 serta waktu komputasi 0,0947 detik, sedangkan *K-Medoids* memperoleh *DBI* 0,5870, *Silhouette Score* 0,5857, dan waktu komputasi 0,1622 detik. Analisis distribusi kategori produk mengungkapkan bahwa kategori Fashion dan Hiburan cenderung terkelompok pada klaster C2, sedangkan kategori Kesehatan dan Otomotif lebih dominan pada klaster C1. Berdasarkan analisis deskriptif, klaster C1 memiliki nilai *Number Sold* dan *Total Review* yang tinggi dengan *Price* relatif lebih rendah, sehingga menggambarkan kelompok produk dengan daya jual tinggi dan performa komersial yang kuat. Strategi yang sesuai adalah mempertahankan volume penjualan dengan menjaga kualitas layanan, memastikan ketersediaan stok, serta meningkatkan *Customer Rating* yang pada klaster ini cenderung lebih rendah. Sebaliknya, klaster C2 memiliki *Customer Rating* tinggi namun *Number Sold* dan *Total Review* yang lebih rendah serta *Price* cenderung lebih tinggi, sehingga mencerminkan kelompok produk dengan persepsi kualitas baik namun tingkat penjualan yang belum optimal. Oleh karena itu, strategi yang direkomendasikan adalah meningkatkan visibilitas dan promosi, misalnya melalui diskon, kampanye iklan digital, atau kolaborasi dengan *influencer*, agar penjualan dapat meningkat tanpa menurunkan persepsi kualitas produk.

## **RIWAYAT HIDUP**



Muhammad Raihan Malik, lahir di Jambi pada tanggal 10 Januari 2003. Penulis merupakan anak ke-1 dari pasangan Bapak Agus Widjianto dan Ibu Melinatriyeni. Pendidikan tinggi penulis ditempuh di Program Studi Sistem Informasi, Jurusan Teknik Elektro dan Informatika, Fakultas Sains dan Teknologi, Universitas Jambi. Penulis resmi terdaftar sebagai mahasiswa sejak tahun 2021.

Selama menjalani studi, penulis menunjukkan minat yang besar terhadap bidang teknologi, khususnya dalam pengembangan aplikasi berbasis web dan pengolahan data. Ketertarikan ini mendorong penulis untuk tidak hanya aktif dalam perkuliahan formal, tetapi juga memperkaya pengetahuan melalui berbagai kegiatan dan pelatihan eksternal di bidang *Web Development* dan *Data Science*.

Sebagai bentuk pengembangan kompetensi, penulis mengikuti pelatihan *Junior Web Developer* yang diselenggarakan oleh Kementerian Komunikasi dan Informatika (Kominfo) melalui program *Vocational School Graduate Academy (VSGA) Digital Talent Scholarship* pada Maret 2023. Pelatihan ini memberikan bekal teknis dalam bidang pemrograman web dan pengembangan antarmuka.

Selain itu, penulis juga melaksanakan kegiatan magang (Praktek Kerja Lapangan) di Badan Pusat Statistik Provinsi Jambi selama dua bulan, terhitung sejak 1 September hingga 31 Oktober 2024. Dalam kegiatan ini, penulis bertanggung jawab pada bidang desain dan pengolahan data statistik visual, yang turut berkontribusi dalam publikasi resmi instansi.

## **PRAKATA**

Bismillahirrahmanirrahim. Alhamdulillahirabbil'alamin, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga skripsi yang berjudul “Analisis Klasterisasi Produk Tokopedia dengan Algoritma *K-means* dan *K-medoids*” ini dapat diselesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW. Penyusunan skripsi ini tidak terlepas dari dukungan, bimbingan, serta doa dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu, baik secara langsung maupun tidak langsung, yaitu:

1. Bapak Agus Widjianto dan Ibu Melinatriyeni, orang tua penulis yang senantiasa mendukung, mendoakan, membantu dan selalu mencoba mengerti kondisi penulis dalam proses penyelesaian skripsi.
2. Bapak Drs. Jefri Marzal, M.Sc., D.I.T., selaku Dekan Fakultas Sains dan Teknologi Universitas Jambi.
3. Ibu Reni Aryani, S.Kom., M.S.I., selaku Koordinator Program Studi Sistem Informasi Universitas Jambi.
4. Bapak Pradita Eko Prasetyo Utomo, S.Pd., M.Cs. dan Bapak Benedika Ferdian Hutabarat, S.Komp., M.Kom., selaku Dosen Pembimbing yang telah memberikan arahan dan bimbingan secara langsung kepada penulis dalam proses penyelesaian skripsi.
5. Ibu Ulfa Khaira, S.Komp., M.Kom., Bapak Akhiyar Waladi, S.Komp., M.Kom., Ibu Dewi Lestari, S.Kom., M.S.I., selaku dosen penguji skripsi, yang telah memberikan saran dan masukan untuk penyelesaian skripsi.
6. Ibu Rizqa Raaiqa Bintana, S.T., M.Kom. selaku Dosen Pembimbing Akademik yang telah memberikan pengarahan selama masa studi.
7. Seluruh dosen dan staf di Program Studi Sistem Informasi Universitas Jambi atas ilmu, wawasan, serta bimbingan yang telah diberikan sepanjang masa studi penulis.
8. Muhammad Vito Alfajr, saudara penulis yang selalu memberikan dukungan moral dan semangat selama penyelesaian skripsi.
9. Alyudha Maryon dan Rahul Marcelino Holis sebagai teman seperjuangan yang selalu bersama dalam menghadapi berbagai situasi di dunia perkuliahan baik secara akademis maupun non akademis.
10. Seluruh teman yang telah menemani penulis selama masa perkuliahan, atas kebersamaan, dukungan, dan semangat belajar yang telah dibangun bersama hingga proses penyelesaian skripsi ini.

11. Segala pihak yang secara langsung maupun tidak langsung telah memberikan dukungan dan bantuan selama proses penyelesaian skripsi, yang tidak dapat disebutkan satu per satu.
12. Terakhir, penulis ingin mengucapkan terima kasih kepada satu sosok yang selama ini diam-diam berjuang tanpa henti, seorang laki-laki sederhana dengan impian yang tinggi, namun sering kali sulit ditebak isi pikiran dan hatinya. Terima kasih kepada penulis skripsi ini, yaitu diriku sendiri, Muhammad Raihan Malik, anak pertama yang sedang melangkah menuju usia 23 tahun, yang dikenal keras kepala namun terkadang sifatnya seperti anak kecil pada umumnya. Terima kasih telah hadir di dunia ini, telah bertahan sejauh ini, dan terus berjalan melewati segala tantangan yang semesta hadirkan. Terima kasih karena tetap berani menjadi dirimu sendiri. Aku bangga atas setiap langkah kecil yang telah kau ambil, atas semua pencapaian yang mungkin tak selalu dirayakan oleh orang lain. Walau terkadang harapanmu tidak sesuai dengan apa yang semesta berikan, tetaplah belajar menerima dan mensyukuri apa pun yang kamu dapatkan. Jangan pernah lelah untuk tetap berusaha, berbahagialah di mana pun kamu berada. Rayakan apa pun dalam dirimu dan jadikan dirimu bersinar di mana pun tempatmu bertumpu. Aku berdoa, semoga langkah dari kaki kecilmu selalu diperkuat, dikelilingi oleh orang-orang yang hebat, serta mimpimu satu per satu akan terwujud.

Semoga skripsi ini dapat memberikan manfaat, khususnya dalam bidang analisis data dan klasterisasi. Penulis menyadari masih terdapat kekurangan dalam penyusunan skripsi ini, sehingga segala bentuk masukan dan saran yang diberikan akan sangat berarti untuk meningkatkan kualitas skripsi ini. Semoga skripsi ini dapat memberikan dampak positif kepada pembaca. Terima kasih atas segala perhatiannya.

Jambi, 19 November 2025  
Yang Menyatakan,

Muhammad Raihan Malik  
NIM. F1E121145

## DAFTAR ISI

	Halaman
PENGESAHAN .....	i
RINGKASAN .....	ii
RIWAYAT HIDUP .....	iii
PRAKATA .....	iv
DAFTAR ISI .....	vi
DAFTAR TABEL .....	viii
DAFTAR GAMBAR .....	ix
DAFTAR LAMPIRAN .....	x
I. PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian .....	3
1.4 Batasan Masalah .....	3
1.5 Manfaat Penelitian .....	4
II. TINJAUAN PUSTAKA .....	5
2.1 <i>E-Commerce</i> dan Tokopedia .....	5
2.2 <i>Data Mining</i> .....	6
2.3 <i>Machine Learning</i> .....	7
2.4 <i>Clustering</i> (Pengelompokan Data) .....	8
2.5 <i>Algoritma K-Means</i> .....	10
2.6 <i>Algoritma K-Medoids</i> .....	12
2.7 Pendekatan Klaster Optimal .....	13
2.8 Evaluasi Hasil <i>Clustering</i> .....	14
2.9 Penelitian Terdahulu .....	16
III. METODOLOGI PENELITIAN .....	19
3.1 Waktu Penelitian .....	19
3.2 Alat dan Bahan Penelitian .....	19
3.3 Tahapan Penelitian .....	20
IV. HASIL DAN PEMBAHASAN .....	38
4.1 Pengumpulan Data .....	38
4.2 <i>Preprocessing</i> Data .....	42
4.3 Transformasi Data .....	50
4.4 Pendekatan Klaster Optimal .....	52
4.5 Implementasi Algoritma .....	55
4.6 Evaluasi <i>Clustering</i> .....	64
4.7 Interpretasi Hasil .....	67

V. KESIMPULAN DAN SARAN.....	72
5.1 Kesimpulan.....	72
5.2 Saran.....	73
DAFTAR PUSTAKA .....	74
DAFTAR LAMPIRAN.....	79

## **DAFTAR TABEL**

Tabel	Halaman
1. Klasifikasi <i>Machine Learning</i> .....	7
2. Perbandingan <i>Hierarchical</i> dan <i>Non-Hierarki</i> .....	9
3. Perbandingan Algoritma <i>Clustering</i> .....	9
4. Evaluasi Kualitas <i>Clustering</i> .....	10
5. Perbandingan Metrik Evaluasi <i>Clustering</i> .....	16
6. Penelitian Terdahulu .....	16
7. Waktu Penelitian .....	19
8. Spesifikasi Perangkat Keras.....	19
9. Perangkat Lunak.....	20
10. Contoh 10 Data Setelah <i>Cleaning</i> .....	24
11. Contoh 10 Data Setelah Transformasi .....	27
12. 10 Data Entri Awal.....	39
13. Sebelum dan Sesudah Standarisasi.....	43
14. <i>Mapping</i> Kategori Asli ke Kategori Utama.....	44
15. Contoh 5 Entri Awal .....	50
16. Nilai Minimum dan Maksimum Sebelum Normalisasi .....	51
17. 1 Data Sebelum Normalisasi.....	52
18. 1 Data Setelah Normalisasi.....	52
19. Nilai Evaluasi <i>DBI</i> dan <i>Silhouette Score</i> .....	66
20. Perbandingan Waktu Komputasi <i>K-Means</i> dan <i>K-Medoids</i> .....	67
21. Jumlah Produk per Klaster Hasil <i>K-Means</i> dan <i>K-Medoids</i> .....	68

## DAFTAR GAMBAR

Gambar	Halaman
1. Logo Tokopedia .....	5
2. Jenis-Jenis <i>Machine Learning</i> .....	8
3. <i>Flowchart</i> Algoritma <i>K-Means</i> .....	11
4. <i>Flowchart</i> Algoritma <i>K-Medoids</i> .....	13
5. Contoh Grafik <i>Elbow Method</i> .....	14
6. <i>Fishbone Diagram</i> .....	21
7. Tampilan Link <i>Dataset</i> .....	38
8. <i>Flowchart Preprocessing</i> .....	42
9. Kode <i>Python</i> Standarisasi Teks .....	43
10. Kode <i>Python</i> Pemetaan Kategori.....	44
11. Kode <i>Python</i> Pembersihan Data Kosong dan Duplikat.....	45
12. Jumlah Data Sebelum dan Sesudah <i>Cleaning</i> .....	45
13. Kode <i>Python</i> Seleksi Kategori.....	46
14. Seleksi Jumlah Data Memadai .....	46
15. Kode <i>Python</i> Penanganan <i>Outlier</i> .....	47
16. <i>Boxplot</i> Harga ( <i>Price</i> ) .....	47
17. <i>Boxplot</i> Number <i>Sold</i> .....	48
18. <i>Boxplot</i> Total <i>Review</i> .....	48
19. <i>Countplot</i> Distribusi Customer Rating.....	48
20. Kode <i>Python</i> Penomoran dan Seleksi Atribut .....	49
21. Data Setelah <i>Cleaning</i> .....	49
22. Kode <i>Python</i> Menghapus dan Menambah Kolom "No" .....	50
23. Kode <i>Python</i> Menentukan Atribut Numerik untuk Dinormalisasi .....	51
24. Kode <i>Python</i> Menerapkan <i>Min-Max Scaling</i> .....	51
25. Hasil <i>Dataset</i> Akhir .....	52
26. Kode <i>Python Elbow Method K-Means</i> .....	53
27. <i>Elbow Method K-Means</i> .....	53
28. Kode <i>Python Elbow Method K-Medoids</i> .....	54
29. <i>Elbow Method K-Medoids</i> .....	54
30. <i>Flowchart K-Means</i> .....	55
31. Kode <i>Python</i> Implementasi <i>K-Means</i> .....	56
32. Iterasi Pertama <i>K-Means</i> .....	57
33. Iterasi Kedua <i>K-Means</i> .....	57
34. Iterasi Ketiga <i>K-Means</i> .....	58
35. Iterasi Keempat <i>K-Means</i> .....	58
36. Iterasi Kelima <i>K-Means</i> .....	59
37. Iterasi Keenam <i>K-Means</i> .....	59
38. Iterasi Konvergen <i>K-Means</i> .....	60
39. <i>Flowchart K-Medoids</i> .....	60
40. Kode <i>Python</i> Implementasi <i>K-Medoids</i> .....	61
41. Iterasi Pertama <i>K-Medoids</i> .....	62
42. Iterasi Kedua <i>K-Medoids</i> .....	62
43. Iterasi Ketiga <i>K-Medoids</i> .....	63
44. Iterasi Keempat <i>K-Medoids</i> .....	63
45. Iterasi Konvergen <i>K-Medoids</i> .....	64
46. Kode <i>Python</i> Evaluasi <i>K-Means</i> .....	65
47. Kode <i>Python</i> Evaluasi <i>K-Medoids</i> .....	66
48. Kode <i>Python</i> Waktu Komputasi <i>K-Means</i> .....	66
49. Kode <i>Python</i> Waktu Komputasi <i>K-Medoids</i> .....	67
50. Analisis Ciri-Ciri Masing-Masing Klaster.....	69
51. Perbandingan Distribusi Produk.....	70

**DAFTAR LAMPIRAN**

Lampiran

Halaman

1. Kode Program Lengkap Tahapan Klasterisasi Produk Tokopedia ..... 79

## I. PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan *e-commerce* di Indonesia sangat pesat dalam satu dekade terakhir, didorong secara signifikan oleh meningkatnya penetrasi internet dan adopsi *smartphone*, yang memudahkan akses belanja digital di kalangan masyarakat luas (V. K. Sari & Nasution, 2024). Akses digital yang semakin inklusif memungkinkan konsumen dari daerah terpencil sekalipun berbelanja daring dengan mudah. Platform *e-commerce* besar seperti Tokopedia dan Shopee pun berkembang sangat cepat misalnya, digitalisasi UMKM melalui Tokopedia dan Shopee mampu meningkatkan pendapatan dan jangkauan pasar para penjual (Purba et al., 2025). Inovasi layanan seperti dompet digital, pengiriman instan, pelacakan pesanan *real-time*, serta antarmuka pengguna yang ramah juga turut mengubah kebiasaan belanja konsumen menjadi lebih mengandalkan platform digital. Secara keseluruhan, tren ini tidak hanya mengubah pola konsumsi masyarakat, tetapi juga mendorong ekonomi digital Indonesia menjadi kekuatan utama di Asia Tenggara diperkirakan mencapai nilai lebih dari USD 130 miliar pada 2025 (Purba et al., 2025).

Dalam ekosistem *e-commerce* yang kompetitif, penyedia platform tidak hanya bersaing soal jumlah produk atau harga, tetapi juga kualitas pengalaman pengguna (*user experience*) dan layanan yang bersifat personal. Salah satu strategi penting adalah penggunaan sistem rekomendasi otomatis berbasis kecerdasan buatan (AI). Dengan menganalisis data interaksi pengguna misalnya riwayat pencarian, kebiasaan belanja, dan pola klik sistem ini dapat menyajikan rekomendasi produk relevan secara *real-time*. Studi menunjukkan bahwa platform *e-commerce* yang menerapkan sistem rekomendasi AI dapat meningkatkan pengalaman pengguna secara signifikan (Didi Riswan et al., 2024). Salah satu metode yang umum diterapkan adalah *Clustering*, yaitu teknik analisis data yang mengelompokkan data berdasarkan kesamaan karakteristiknya. *Clustering* telah banyak diterapkan dalam sistem rekomendasi dan analisis kepuasan pelanggan (Putra et al., 2023). (Nugraha & Hayati, 2024) menerapkan metode ini untuk mengelompokkan kategori produk berdasarkan *Rating* pengguna, sehingga dapat memberikan wawasan bagi platform *e-commerce* dalam menyusun strategi pemasaran yang lebih akurat (Nugraha et al., 2024). Dengan demikian, penelitian terkait analisis klasterisasi produk tidak hanya bermanfaat dalam mendukung sistem rekomendasi, tetapi juga memberikan kontribusi nyata dalam memahami perilaku konsumen serta memetakan tren pasar yang dinamis di sektor *e-commerce* Indonesia.

Sebagai bagian dari metode *unsupervised learning*, *Clustering* bekerja tanpa memerlukan label kelas tertentu, sehingga dapat digunakan untuk mengidentifikasi pola tersembunyi dalam data transaksi dan ulasan pelanggan (Gymnastiar & Bahtiar, 2024). Berbagai penelitian sebelumnya telah menyoroti hubungan antara *rating* dan jumlah penjualan Rahman & Suroyo (2021) menemukan dalam analisis produk elektronik di Shopee bahwa produk dengan penjualan 0–1000 memiliki skor teratas, sedangkan produk dengan penjualan lebih tinggi justru cenderung memiliki rating bintang lebih rendah (Ainur Rahman & Suroyo, 2021). Demikian pula, Harjono et al. (2023) dalam studi klasterisasi tingkat penjualan di platform Panak.id menunjukkan bahwa distribusi penjualan dapat diidentifikasi melalui klasterisasi berdasarkan jumlah *Sold*, memberikan *insight* pengelompokan produk yang kurang laku dan dapat dijadikan acuan strategi penjualan (Harjono et al., 2023). Selain itu, Putra et al. (2021) menunjukkan bahwa penerapan *Clustering* pada kategori produk dapat meningkatkan efisiensi sistem rekomendasi dan berdampak langsung pada tingkat kepuasan pelanggan (Putra et al., 2021).

Berbagai metode *Clustering* telah dikembangkan, dan dua di antaranya yang paling umum digunakan adalah K-Means dan K-Medoids. *K-Means* bekerja dengan membagi data ke dalam sejumlah klaster menggunakan perhitungan jarak *Euclidean*, dan dikenal memiliki efisiensi tinggi dalam menangani *dataset* berukuran besar (Asy Aria et al., 2023). Namun, algoritma ini kurang optimal dalam menangani *outlier* dan memerlukan pemilihan jumlah *Cluster* yang tepat (Hendrastuty, 2024). Sebagai alternatif, *K-Medoids* lebih robust terhadap *outlier* karena memilih titik data aktual sebagai pusat *Cluster* (Yafi et al., 2023). Selain itu, algoritma *K-Medoids* menggunakan *medoid* sebagai pusat *Cluster* yang paling representatif, dengan keunggulan dalam meminimalkan total jarak dalam klaster, meskipun waktu komputasi lebih tinggi dibandingkan *K-Means* saat diterapkan pada *dataset* besar (Meiyanti et al., 2024). Dengan mempertimbangkan kelebihan dan kekurangan kedua algoritma, penelitian ini mengombinasikan *K-Means* dan *K-Medoids* untuk menghasilkan analisis *Clustering* yang lebih komprehensif. Penelitian sebelumnya menunjukkan bahwa pemilihan metode *Clustering* yang tepat sangat memengaruhi hasil segmentasi data.

Penelitian ini yang berjudul "**Analisis Klasterisasi Produk Tokopedia Berdasarkan Rating dan Sold dengan Algoritma K-Means dan K-Medoids**", diharapkan dapat memberikan pemahaman yang lebih mendalam mengenai keunggulan dan efektivitas masing-masing algoritma dalam proses pengelompokan produk.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, terdapat beberapa permasalahan yang menjadi fokus dalam penelitian ini:

1. Bagaimana penerapan teknik *Clustering* untuk mengelompokkan produk di Tokopedia berdasarkan atribut numerik seperti *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*?
2. Bagaimana perbandingan kinerja algoritma *K-Means* dan *K-Medoids* dalam proses *Clustering*, serta bagaimana distribusi kategori produk dalam masing-masing klaster yang terbentuk dapat dianalisis sebagai bagian dari interpretasi hasil?

## 1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Menerapkan teknik *Clustering* untuk mengelompokkan produk Tokopedia berdasarkan atribut numerik yang telah dinormalisasi, yaitu *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*.
2. Membandingkan performa algoritma *K-Means* dan *K-Medoids* dalam menghasilkan klaster yang optimal, serta mengamati distribusi kategori produk dalam tiap klaster sebagai bagian dari interpretasi hasil klasterisasi.

## 1.4 Batasan Masalah

Penelitian ini difokuskan dan diarahkan melalui penetapan beberapa batasan masalah, yaitu:

1. *Dataset* yang digunakan adalah PRDECT-ID (*Product Reviews Dataset for Emotions Classification Tasks – Indonesian*), dipublikasikan oleh Universitas Bina Nusantara melalui platform Mendeley Data pada tahun 2022, dengan lisensi *Creative Commons* (CC BY 4.0).
2. Penelitian ini hanya menggunakan algoritma *K-Means* dan *K-Medoids* sebagai metode *Clustering*.
3. Atribut yang digunakan dalam proses klasterisasi hanya mencakup data numerik: *Price*, *Customer Rating*, *Number Sold*, *Total Review*.
4. Informasi kategori produk (*Main Category*) dianalisis secara deskriptif pada tahap interpretasi hasil untuk memahami sebaran kategori dalam tiap klaster.
5. Penelitian tidak membahas aspek lain seperti *sentiment analysis*, *text mining*, atau pengolahan data ulasan konsumen secara mendalam.

### **1.5 Manfaat Penelitian**

Penelitian ini bertujuan memberikan manfaat sebagai berikut:

1. Manfaat Praktis, Penelitian ini dapat dimanfaatkan oleh pelaku usaha, khususnya seller di platform Tokopedia maupun *e-commerce* lainnya, untuk memahami segmentasi produk berdasarkan atribut numerik yang relevan seperti harga, *rating*, penjualan, dan ulasan.
2. Manfaat Teoritis, Menambah literatur dan pemahaman dalam bidang sistem informasi, khususnya mengenai perbandingan algoritma *Clustering K-Means* dan *K-Medoids* dalam membentuk segmentasi produk berbasis data numerik pada platform *e-commerce*.

## II. TINJAUAN PUSTAKA

### 2.1 E-Commerce dan Tokopedia

*E-commerce* adalah media digital yang memungkinkan proses jual beli melalui internet, dan kini menjadi sarana strategis untuk memperluas jangkauan pasar perusahaan baik berskala besar maupun UMKM (Rahmawati & Fasa, 2025). *E-commerce* melibatkan pembelian, penjualan, serta pemasaran produk dan layanan melalui media berbasis internet, memungkinkan pelaku usaha mengintegrasikan seluruh aktivitas bisnis dalam satu ekosistem digital (Prasetyo et al., 2024). Belanja daring melalui *e-commerce* dilakukan secara digital melalui jaringan internet, didukung oleh metode pembayaran seperti transfer bank, dompet digital, dan pembayaran elektronik lainnya (Wenerda & Hariyanti, 2024). Sejalan dengan definisi dari para pakar sebelumnya, berbagai studi di Indonesia juga menyatakan bahwa *e-commerce* adalah bentuk transaksi bisnis digital yang terjadi antara individu maupun perusahaan, dan dijalankan melalui jaringan komputer atau media elektronik lainnya (Atikah, 2019). Jenis-jenis *e-commerce* diklasifikasikan ke dalam beberapa model utama seperti *Business to Business* (B2B), *Business to Consumer* (B2C), *Consumer to Business* (C2B), dan *Consumer to Consumer* (C2C), yang masing-masing telah banyak diterapkan di Indonesia melalui platform seperti Tokopedia, Bukalapak, dan OLX (Rahmanto, 2022).



**Gambar 1.** Logo Tokopedia  
Sumber : (<https://www.tokopedia.com>)

Tokopedia merupakan salah satu perusahaan *e-commerce* terbesar di Indonesia yang menerapkan model bisnis *marketplace* dan mall online, di mana individu, toko kecil, hingga merek ternama dapat membuka serta mengelola toko daring secara mandiri (Rahmanto, 2022). Sejak didirikan oleh William Tanuwijaya dan Leontinus Alpha Edison pada 6 Februari 2009, Tokopedia terus menunjukkan perkembangan pesat melalui dukungan investasi global serta strategi bisnis digital yang adaptif terhadap kebutuhan pasar lokal (Afianti et al., 2023). Platform ini tidak hanya memfasilitasi transaksi jual beli secara elektronik, namun juga mengusung visi sosial berupa “Membangun Indonesia yang Lebih Baik Lewat Internet”, yang menekankan pentingnya teknologi dalam mendorong kemajuan ekonomi masyarakat (Ahmada, 2021).

Tokopedia berkomitmen mendukung pelaku Usaha Mikro, Kecil, dan Menengah (UMKM) dalam memasarkan produk mereka secara daring dengan memberikan akses terhadap sistem pembayaran digital, promosi online, dan fitur-fitur pemberdayaan berbasis teknologi (Anum et al., 2024). Dalam proses operasionalnya, perusahaan ini juga mengedepankan nilai-nilai seperti integritas, karakter positif, serta fokus pada pelanggan yang diwujudkan dalam pengembangan layanan dan kualitas produk (Ahmada, 2021). Dengan menjadikan internet sebagai instrumen pemberdayaan ekonomi, Tokopedia tidak hanya menciptakan ekosistem digital yang efisien, tetapi juga inklusif dan berdampak luas bagi pertumbuhan bisnis lokal di Indonesia (Rahmanto, 2022).

## **2.2 Data Mining**

*Data mining* merupakan proses pencarian informasi yang bernalih dari kumpulan data besar dengan cara menemukan pola tersembunyi menggunakan berbagai teknik statistik, pembelajaran mesin, dan kecerdasan buatan. Teknik ini krusial karena menyaring informasi relevan untuk pengambilan keputusan. Dalam konteks *e-commerce*, *data mining* digunakan untuk memahami perilaku pelanggan, menyusun strategi penjualan, dan memberikan rekomendasi produk yang tepat. Seperti dijelaskan oleh (Mubarok et al., 2025), penerapan *data mining* pada platform Tokopedia dilakukan melalui tahapan web scraping, ETL (*Extract, Transform, Load*), dan visualisasi data untuk membantu UMKM mengambil keputusan berbasis data. Hal ini didukung oleh (Haryanti et al., 2024) yang menemukan bahwa pemanfaatan *data mining* dapat meningkatkan efektivitas manajemen penjualan dan efisiensi strategi bisnis berbasis analisis historis perilaku konsumen. Di era digital saat ini, pertumbuhan volume data yang sangat cepat dan kebutuhan pasar yang kompetitif menjadi pendorong utama adopsi teknologi *data mining*.

Implementasi *data mining* pada platform Tokopedia tidak hanya memungkinkan segmentasi pelanggan atau analisis tren pembelian, tetapi juga dapat diterapkan untuk klasterisasi produk. Teknik seperti ini digolongkan ke dalam metode *unsupervised learning*, yang memungkinkan sistem untuk membentuk pengelompokan tanpa adanya label awal. Proses ini mendukung strategi promosi, pemetaan produk, dan pengelolaan stok, karena hasil klasifikasi otomatis mencerminkan pola preferensi konsumen dengan lebih tepat. Seperti yang dijelaskan oleh (Ningsih, 2024), implementasi *data mining* dalam *e-commerce* mampu menyusun model *Clustering* untuk optimalisasi rekomendasi dan efisiensi pajak berbasis analisis perilaku pengguna dalam ekosistem digital.

### 2.3 Machine Learning

*Machine learning* atau pembelajaran mesin merupakan bagian dari kecerdasan buatan (AI) yang memungkinkan sistem komputer untuk belajar dari data, mengenali pola, dan melakukan prediksi tanpa diprogram secara eksplisit. Teknologi ini bekerja dengan membangun model berdasarkan data historis, kemudian digunakan untuk mengklasifikasikan atau memproses data baru secara otomatis. Pembelajaran mesin kini memainkan peran sentral dalam berbagai penerapan kecerdasan buatan karena kemampuannya untuk beradaptasi terhadap dinamika data dan konteks permasalahan (Asrawi, 2025).

*Machine learning* digunakan untuk membantu proses pengelompokan produk di platform *e-commerce*. Metode yang digunakan adalah *unsupervised learning*, yakni metode pembelajaran mesin yang tidak memerlukan label data awal, tetapi mampu mengelompokkan data berdasarkan kesamaan karakteristik. *K-Means* dan *K-Medoids* merupakan algoritma yang umum digunakan dalam pendekatan ini karena efisien dalam menemukan struktur data secara otomatis. Menurut (Edy et al., 2024), kedua algoritma ini efektif dalam melakukan segmentasi pelanggan tanpa supervisi, terutama ketika diterapkan dalam sistem *marketplace* berbasis data besar.

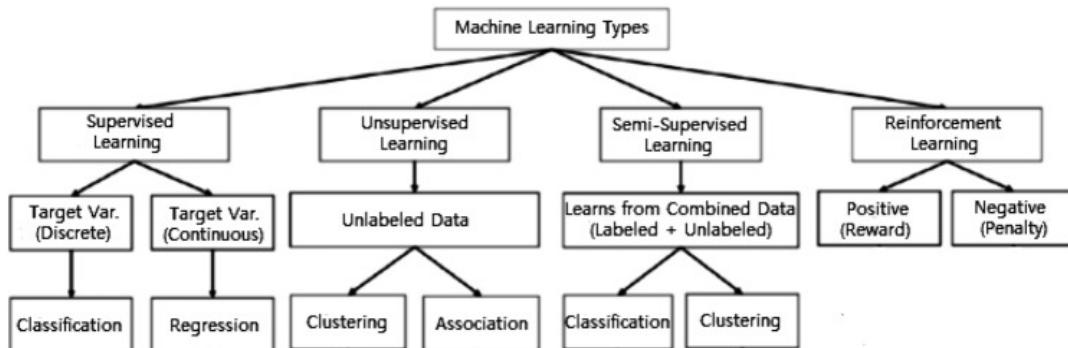
Berdasarkan teknik pembelajarannya, *machine learning* dapat diklasifikasikan menjadi beberapa jenis, antara lain *supervised learning*, *unsupervised learning*, *semi-supervised learning*, dan *reinforcement learning*, yang masing-masing digunakan sesuai dengan struktur dan kebutuhan data. Rincian jenis pembelajaran tersebut ditampilkan pada Tabel 1, yang memaparkan tipe, penjelasan, dan contoh aplikasi dari masing-masing metode *machine learning*.

**Tabel 1.** Klasifikasi *Machine Learning*

Tipe Machine Learning	Penjelasan	Contoh Aplikasi
<i>Supervised Learning</i>	Model dilatih menggunakan data berlabel untuk melakukan prediksi	Prediksi harga, klasifikasi spam email
<i>Unsupervised Learning</i>	Model menemukan struktur tersembunyi dari data tanpa label	<i>Clustering</i> pelanggan, segmentasi produk
<i>Semi-Supervised Learning</i>	Gabungan data berlabel dan tidak berlabel untuk pelatihan	Pengenalan wajah
<i>Reinforcement Learning</i>	Model belajar melalui interaksi dengan lingkungan menggunakan sistem reward	Pengembangan game AI, robotika

(Sumber: Retnoningsih & Pramudita, 2020)

*Unsupervised learning* seperti *Clustering* menjadi penting dalam memahami karakteristik tersembunyi dalam data besar *e-commerce*, karena mampu mengelompokkan produk serupa tanpa intervensi manual. Penelitian oleh (Wijaya et al., 2025) menunjukkan bahwa penerapan teknik *unsupervised learning* sangat bermanfaat dalam menganalisis perilaku pengguna dan fitur aplikasi pada platform *e-commerce*, terutama dalam menyusun strategi peningkatan usability dan pemetaan fitur yang relevan untuk Generasi Z. Hubungan antarjenis pembelajaran *machine learning* dapat dilihat pada Gambar 2, yang menampilkan bagan tipe pembelajaran beserta karakteristik utamanya untuk memberikan pemahaman yang lebih jelas.



**Gambar 2.** Jenis-Jenis *Machine Learning*

(Sumber : Sarker, 2021)

## 2.4 Clustering (Pengelompokan Data)

*Clustering* merupakan salah satu teknik dalam *data mining* yang digunakan untuk mengelompokkan data ke dalam beberapa kelompok atau klaster berdasarkan kesamaan karakteristik antar data. Metode ini termasuk dalam kategori pembelajaran tanpa pengawasan (*unsupervised learning*), karena proses pengelompokannya tidak membutuhkan label kelas pada data yang dianalisis. *Clustering* telah banyak diterapkan di berbagai bidang seperti pemasaran, pendidikan, kesehatan, dan *e-commerce* untuk menemukan pola-pola tersembunyi dalam data (Hermawati et al., 2020). Secara umum, metode *Clustering* terbagi menjadi dua kategori, yaitu *Clustering hierarki* dan *non-hierarki*. *Clustering hierarki* membentuk struktur bertingkat seperti pohon (*dendogram*), sementara metode *non-hierarki* secara langsung mengelompokkan data ke dalam sejumlah klaster yang telah ditentukan sebelumnya berdasarkan nilai parameter klaster (Saputri & Arianto, 2023). Penggunaan metode *Clustering* ini memungkinkan analis untuk memperoleh segmentasi data yang lebih informatif, sehingga hasil analisis dapat digunakan dalam pengambilan keputusan yang lebih tepat sasaran. Perbandingan metode *Hierarchical* dan *Non-Hierarki* ditunjukkan pada Tabel 2.

**Tabel 2.** Perbandingan *Hierarchical* dan *Non-Hierarki*

<b>Metode</b>	<b>Kelebihan</b>	<b>Kekurangan</b>
<i>Hierarchical</i>	Memberikan struktur bertingkat yang informatif	Tidak efisien untuk <i>dataset</i> besar, sensitif terhadap noise dan <i>outlier</i>
<i>Non-Hierarki</i>	Efisien untuk <i>dataset</i> besar	Memerlukan estimasi jumlah klaster di awal

(Sumber: Nur et al., 2023)

Dalam pendekatan *Clustering non-hierarki*, dua algoritma yang sering digunakan adalah *K-Means* dan *K-Medoids*. *K-Means* bekerja dengan menentukan pusat klaster berdasarkan nilai rata-rata (*mean*) dari objek-objek dalam klaster, sedangkan *K-Medoids* menggunakan *medoid*, yaitu objek aktual yang paling representatif dalam suatu klaster. Menurut penelitian oleh Aji & Ahmad, (2024) *K-Means* memiliki keunggulan dalam hal kecepatan dan efisiensi pada *dataset* besar, namun cenderung sensitif terhadap *outlier* dan bentuk distribusi data yang tidak linier (Aji & Ahmad, 2024). Sebaliknya, *K-Medoids* lebih tahan terhadap *outlier* karena menggunakan titik data aktual sebagai pusat klaster, meskipun memiliki waktu komputasi yang lebih tinggi. Pemilihan metode yang sesuai dapat ditentukan berdasarkan karakteristik data dan evaluasi menggunakan metrik seperti *Silhouette Score* dan *Davies-Bouldin Index* untuk mengukur kualitas klaster yang dihasilkan (Hermawati et al., 2020). Perbandingan kedua algoritma dapat dilihat pada Tabel 3.

**Tabel 3.** Perbandingan Algoritma *Clustering*

<b>Algoritma</b>	<b>Pendekatan</b>	<b>Kelebihan</b>	<b>Kekurangan</b>
<i>K-Means</i>	Berbasis <i>centroid</i> (rata-rata)	Cepat dan efektif pada data besar	Rentan terhadap <i>outlier</i> dan bentuk <i>Cluster</i> yang tidak bulat
<i>K-Medoids</i>	Berbasis <i>medoid</i> (titik nyata)	Lebih tahan terhadap <i>outlier</i>	Perhitungan lebih berat dibanding <i>K-Means</i>

(Sumber: Hoerunnisa et al., 2024)

Evaluasi hasil *Clustering* dilakukan menggunakan dua metrik populer, yaitu *Davies Bouldin Index (DBI)* dan *Silhouette Score*, guna memastikan kualitas pengelompokan yang dihasilkan. Kedua metrik ini telah terbukti efektif dalam mengukur validitas klaster berdasarkan kedekatan dan pemisahan antar kelompok data (Syahkur & Hartama, 2024). Definisi serta interpretasi dari kedua metrik tersebut dapat dilihat pada Tabel 4.

**Tabel 4.** Evaluasi Kualitas *Clustering*

<b>Metrik</b>	<b>Definisi</b>	<b>Interpretasi</b>
<i>Davies Bouldin Index</i>	Mengukur tingkat pemisahan dan kerapatan <i>Cluster</i>	Nilai lebih kecil menandakan <i>Clustering</i> lebih baik
<i>Silhouette Score</i>	Mengukur kualitas keseluruhan hasil <i>Clustering</i> dengan mempertimbangkan kohesi dalam klaster dan pemisahan antar klaster.	Nilai mendekati 1 menunjukkan kualitas <i>Cluster</i> yang baik

(Sumber: Syahkur &amp; Hartama, 2024)

## 2.5 Algoritma *K-Means*

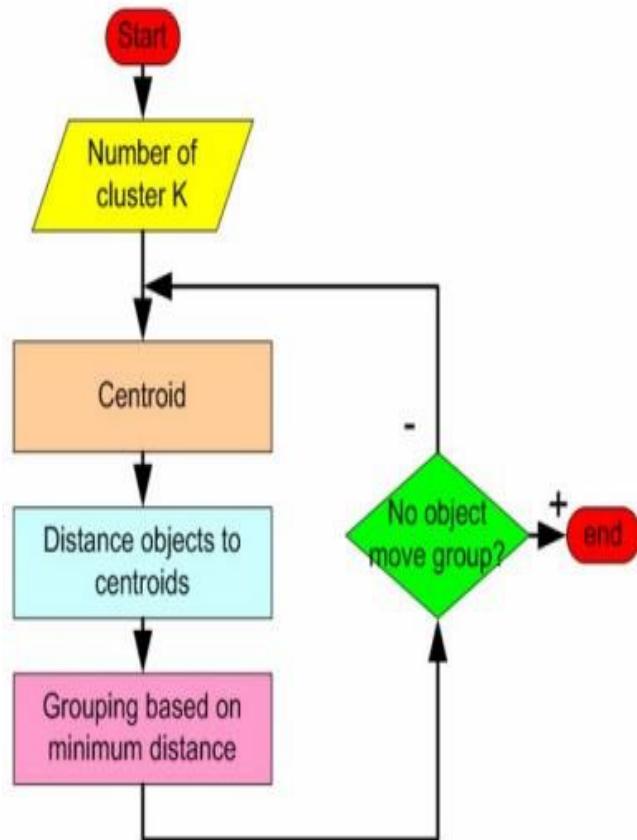
Algoritma *K-Means* merupakan salah satu metode klasterisasi yang populer digunakan dalam analisis data karena kesederhanaannya dan kemampuannya menangani *dataset* berukuran besar secara efisien (Damayanthi et al., 2024). *K-Means* bekerja dengan cara membagi data ke dalam sejumlah klaster yang telah ditentukan sebelumnya (nilai *k*) berdasarkan kesamaan karakteristik antar data. Prosesnya dimulai dengan pemilihan *centroid* awal secara acak, kemudian dilanjutkan dengan menghitung jarak setiap titik data ke masing-masing *centroid* menggunakan rumus *Euclidean Distance*. Data selanjutnya dialokasikan ke klaster dengan *centroid* terdekat, dan *centroid* diperbarui berdasarkan rata-rata posisi seluruh anggota klaster. Tahapan ini diulang secara iteratif hingga *centroid* tidak mengalami perubahan signifikan atau kondisi konvergen tercapai (Wayan & Damayanthi, 2024).

Tujuan utama *K-Means* adalah meminimalkan jumlah kuadrat jarak antara data dan pusat klaster, sehingga data yang tergabung dalam satu klaster memiliki tingkat kemiripan internal yang tinggi (Polgan et al., 2024). Kelebihan algoritma ini terletak pada efisiensi komputasinya, terutama saat diaplikasikan pada *dataset* berukuran besar, serta kemampuan menghasilkan hasil klaster yang interpretatif untuk keperluan analisis lebih lanjut. Namun, *K-Means* juga memiliki kelemahan, seperti sensitivitas terhadap pemilihan *centroid* awal dan kebutuhan untuk menentukan nilai *k* di awal proses (Polgan et al., 2024). Studi-studi terdahulu telah menunjukkan bahwa penerapan *K-Means* yang optimal memerlukan tahapan tambahan, seperti uji validitas klaster (misalnya dengan *Elbow Method*, *Silhouette Score* dan *Davies-Bouldin Index*), untuk memastikan hasil yang diperoleh merepresentasikan struktur data secara akurat . Dengan demikian, *K-Means* tetap menjadi salah satu algoritma fundamental yang banyak digunakan dalam berbagai penelitian *data mining* dan *machine learning*.

### Tahapan Algoritma K-Means

Langkah-langkah kerja *K-Means* dapat dijelaskan sebagai berikut : (Aulia, 2020)

1. Menentukan jumlah *Cluster* (K) yang diinginkan.
  2. Memilih *centroid* awal secara acak dari data yang tersedia.
  3. Menghitung jarak antara setiap data dengan semua *centroid*, umumnya menggunakan rumus *Euclidean Distance*:
- $$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad ..(1)$$
4. Mengelompokkan setiap data ke *centroid* terdekat.
  5. Memperbarui posisi *centroid* dengan menghitung rata-rata seluruh data dalam *Cluster*.
  6. Mengulangi proses perhitungan jarak dan pengelompokan sampai *centroid* stabil.



**Gambar 3.** Flowchart Algoritma K-Means

(Sumber: Wakhidah, 2010)

## 2.6 Algoritma *K-Medoids*

Algoritma *K-Medoids* merupakan salah satu metode *Clustering* berbasis partisi yang bekerja dengan memilih objek nyata dari *dataset* sebagai pusat klaster (*medoid*). Hal ini berbeda dengan *K-Means*, yang menggunakan nilai rata-rata dari anggota klaster sebagai pusatnya. Pendekatan *K-Medoids* memiliki keunggulan dalam hal ketahanan terhadap *outlier* dan *noise*, karena pemilihan *medoid* sebagai pusat klaster menjadikan representasi klaster lebih mencerminkan kondisi nyata data. Dengan demikian, *K-Medoids* dianggap lebih stabil dan akurat dalam situasi data yang tidak bersih, tidak seimbang, atau mengandung nilai ekstrem (Kurmiati et al., 2021).

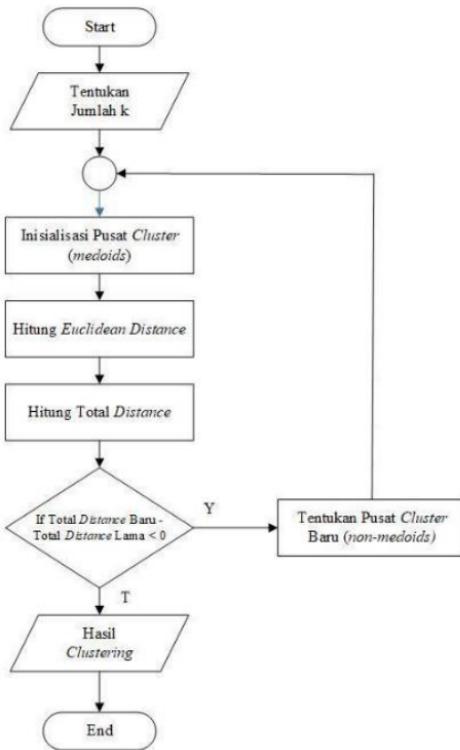
Secara teknis, proses klasterisasi menggunakan *K-Medoids* diawali dengan memilih sejumlah *medoid* secara acak dari kumpulan data. Kemudian, setiap data akan dihitung jaraknya terhadap semua *medoid* menggunakan fungsi jarak seperti *Euclidean Distance*, dan dialokasikan ke *medoid* terdekat. Setelah proses alokasi selesai, algoritma akan mengevaluasi apakah ada kemungkinan pertukaran antara *medoid* yang dipilih dengan data lain dalam klaster, yang dapat menghasilkan penurunan total jarak antar anggota klaster dengan *medoid*-nya. Jika ditemukan pertukaran yang lebih optimal, maka proses tersebut diulang hingga konvergen atau tidak ada perubahan signifikan lagi. Pendekatan ini membuat algoritma lebih tahan terhadap variasi distribusi data, meskipun memiliki beban komputasi yang lebih tinggi dibanding *K-Means* (Alfiah et al., 2020).

Keunggulan utama *K-Medoids* terletak pada kemampuannya menghasilkan klaster yang representatif, terutama saat menangani data *real-world* yang mengandung *noise*, *outlier*, atau bersifat tidak homogen. Selain itu, metode ini lebih fleksibel dalam menginterpretasikan hasil klaster, karena pusat klaster selalu merupakan anggota dari *dataset* itu sendiri, bukan nilai rata-rata yang rentan terhadap nilai ekstrem. Dalam implementasinya, *K-Medoids* sering digunakan pada skenario data yang kompleks, seperti data medis, ekonomi, pemasaran, hingga sistem rekomendasi, karena mampu mempertahankan stabilitas klasterisasi meskipun struktur data tidak ideal. Misalnya, penelitian oleh Hidayat et al., (2022) menunjukkan keberhasilan *K-Medoids* dalam klasterisasi kabupaten di Provinsi Aceh berdasarkan berbagai faktor yang mempengaruhi tingkat kemiskinan, dengan hasil klasterisasi yang stabil, akurat, dan dapat dijadikan dasar pengambilan kebijakan yang lebih tepat sasaran (Hidayat et al., 2022).

### Tahapan Algoritma *K-Medoids*

Berikut adalah langkah-langkah umum dalam algoritma *K-Medoids* (Mirantika et al., 2023):

1. Memilih  $k$  objek secara acak dari data sebagai *medoid* awal.
2. Menghitung jarak setiap data ke semua *medoid* menggunakan rumus *Euclidean Distance*.
3. Mengelompokkan data ke *medoid* terdekat berdasarkan jarak minimum.
4. Memilih calon *medoid* baru dari data *non-medoid* dan menghitung total *cost*.
5. Menukar *medoid* jika *cost* menurun (hasil lebih optimal).
6. Mengulangi proses hingga tidak ada perubahan *medoid* atau *Cluster* stabil.



**Gambar 4.** Flowchart Algoritma *K-Medoids*

(Sumber: Andini & Arifin, 2020)

### 2.7 Pendekatan Klaster Optimal

Menentukan jumlah klaster yang optimal merupakan langkah penting agar hasil klaster yang terbentuk bersifat akurat dan representatif. Salah satu pendekatan yang umum digunakan untuk tujuan ini adalah *Elbow Method*, yang menganalisis perubahan nilai kesalahan klasterisasi untuk mengidentifikasi titik optimal jumlah klaster (S. N. Sari et al., 2024).

### **Elbow Method**

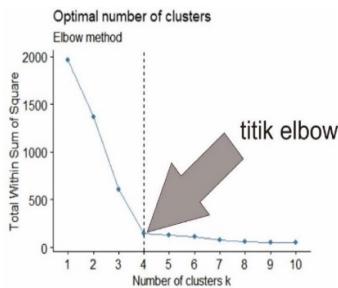
Metode Elbow merupakan teknik yang digunakan untuk menentukan jumlah klaster yang paling sesuai dengan cara mengevaluasi perubahan nilai pada grafik yang membentuk sudut tajam atau "elbow". Titik optimal ditunjukkan ketika penurunan nilai SSE antara jumlah klaster pertama dan kedua paling drastis dibandingkan penurunan berikutnya. Nilai tersebut dihitung menggunakan *Sum of Square Error* (SSE), yang mengukur seberapa jauh penyebaran data dari pusat klasternya. Semakin besar nilai klaster  $K$  yang digunakan, maka nilai SSE cenderung menurun (Sulistiyawan et al., 2021). Adapun rumus untuk menghitung nilai SSE adalah sebagai berikut:

$$SSE = \sum_{k=1}^K \sum_{x_i \in C_k} |x_i - C_k|^2 \quad ..(2)$$

Keterangan:

- $K$ : klaster ke- $c$
- $x_i$ : jarak data objek ke- $i$
- $C_k$ : pusat klaster ke- $i$

Hasil perhitungan SSE untuk berbagai jumlah klaster kemudian divisualisasikan dalam bentuk grafik, sebagaimana ditunjukkan pada Gambar 5.



**Gambar 5.** Contoh Grafik Elbow Method  
(Sumber: Ayu et al., 2019)

### **2.8 Evaluasi Hasil Clustering**

Evaluasi hasil *Clustering* merupakan tahap penting untuk menilai kualitas dan efektivitas pengelompokan. Evaluasi ini bertujuan untuk memastikan bahwa klaster yang terbentuk benar-benar merepresentasikan pola atau struktur dalam data (Hasan, 2024). Dalam penelitian ini, evaluasi dilakukan menggunakan dua metrik yang umum digunakan, yaitu *Davies-Bouldin Index* (*DBI*) dan *Silhouette Score*. *DBI* mengukur rasio antara jarak antar klaster dengan ukuran klaster itu sendiri, di mana nilai *DBI* yang lebih rendah menunjukkan hasil *Clustering* yang lebih baik karena menunjukkan klaster yang lebih terpisah dan kompak (Hasan, 2024). *Silhouette Score* menilai sejauh mana sebuah objek cocok dengan klasternya sendiri dibandingkan dengan klaster lain, dengan nilai mendekati 1 menunjukkan pengelompokan yang optimal (Hasan, 2024).

### **Davies Bouldin Index (DBI)**

*Davies-Bouldin Index (DBI)* merupakan salah satu metode evaluasi yang digunakan untuk menilai kualitas hasil *Clustering* dengan mempertimbangkan dua aspek utama, yaitu kohesi dan separasi. Kohesi mengukur seberapa dekat data dalam suatu klaster terhadap pusat klasternya, sedangkan separasi menilai sejauh mana klaster-klaster yang terbentuk saling berjauhan. Klaster yang ideal ditandai dengan kohesi yang rendah (data dalam klaster saling berdekatan) dan separasi yang tinggi (klaster terpisah dengan jelas). Nilai *DBI* yang mendekati nol menunjukkan kualitas pengelompokan yang semakin baik, karena menggambarkan klaster yang kompak dan terpisah secara optimal (Ayu et al., 2019). Menurut Umagapi & Umaternate, (2023) *DBI* juga digunakan untuk mengevaluasi sejauh mana hasil *Clustering* mampu membedakan kelompok data yang berbeda sekaligus menjaga kedekatan data terhadap pusat klasternya (Umagapi & Umaternate, 2023). Oleh karena itu, semakin kecil nilai *DBI*, maka semakin baik kualitas klaster yang dihasilkan.

Rumus *Davies-Bouldin Index* :

$$DBI = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left( \frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right) \quad ..(3)$$

- $\sigma_i$  : adalah rata-rata jarak antara data dengan pusat klaster  $i$
- $d(c_i, c_j)$  : adalah jarak antar pusat klaster  $i$  dan  $j$

### **Silhouette Score**

*Silhouette Score* merupakan salah satu metode evaluasi yang digunakan untuk menilai kualitas hasil *Clustering* dengan mengukur seberapa dekat suatu data dengan klasternya sendiri dibandingkan dengan klaster lain. Setiap data diberikan nilai *silhouette* yang berkisar antara -1 hingga 1, di mana nilai mendekati 1 menunjukkan bahwa data berada dalam klaster yang sesuai. Semakin tinggi nilai rata-rata *silhouette* dari seluruh data, maka semakin baik struktur klasterisasi yang terbentuk. Menurut (Hasan, 2024), *Silhouette Score* efektif dalam mengevaluasi seberapa optimal data dikelompokkan dalam klaster dan memberikan gambaran yang jelas mengenai kepadatan serta pemisahan antar klaster.

Perhitungan *Silhouette Score*:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad ..(4)$$

- $a(i)$  merupakan rata-rata jarak  $i$  dengan observasi lain dalam satu klaster
- $b(i)$  adalah ratarata jarak antara  $i$  dengan observasi pada klaster terdekat

Berdasarkan Tabel 5 dapat dilihat bahwa setiap metrik evaluasi memiliki karakteristik dan kegunaan masing-masing, sehingga pemilihan metrik yang tepat perlu menyesuaikan dengan algoritma yang digunakan.

**Tabel 5.** Perbandingan Metrik Evaluasi *Clustering*

<b>Metrik Evaluasi</b>	<b>Fungsi Utama</b>	<b>Rentang Nilai</b>	<b>Interpretasi</b>	<b>Cocok untuk</b>
			<b>Nilai Ideal</b>	<b>Algoritma</b>
<i>Davies-Bouldin Index</i>	Mengukur kepadatan & pemisahan antar klaster	$\geq 0$ (bisa negatif)	Semakin rendah, semakin baik	<i>K-Means, K-Medoids</i>
<i>Elbow Method</i>	Menentukan jumlah klaster optimal secara visual berdasarkan SSE	Evaluasi berbasis visual grafik, bukan angka tunggal	Titik ‘siku’ pada grafik	<i>K-Means</i>
<i>Silhouette Score</i>	Menilai kesesuaian data dalam klaster dan jarak ke klaster lain	-1 sampai 1	Mendekati 1: sangat baik, mendekati -1: buruk	<i>K-Means, K-Medoids</i>

## 2.9 Penelitian Terdahulu

**Tabel 6.** Penelitian Terdahulu

<b>No</b>	<b>Penulis</b>	<b>Tahun</b>	<b>Judul</b>	<b>Metode</b>	<b>Hasil</b>
1	Farah Fadhlila Putri Zahardy	2024	Analisis Perbandingan Metode <i>K-Means</i> dan <i>K-Medoids</i> dalam Pengelompokan Mahasiswa Universitas Jambi Berdasarkan Perilaku Penggunaan ChatGPT	<i>K-Means</i> dan <i>K-Medoids</i>	<i>K-Medoids</i> memberikan hasil klaster yang lebih kompak dibanding <i>K-Means</i> pada perilaku penggunaan ChatGPT.
2	Intan Permata Winda Purba	2025	Implementasi <i>Data Mining</i> untuk Analisis Data Penjualan Batik CV. Sogan Batik Rejodani	<i>K-Means Clustering</i>	Metode <i>K-Means</i> berhasil mengelompokkan produk batik berdasarkan pola penjualan untuk membantu pengambilan keputusan manajerial.

3	Nanda Try Luchia et al.	2022	Perbandingan <i>K-Means</i> dan <i>K-Medoids</i> Pada Pengelompokan Data Miskin di Indonesia	<i>K-Means</i> dan <i>K-Medoids</i>	<i>K-Means</i> memberikan nilai <i>DBI</i> 0.041 (lebih optimal) dibanding <i>K-Medoids</i> dalam mengelompokkan data kemiskinan di Indonesia.
4	Endang Retnoningsih & Rully Pramuditaa	2020	Mengenal <i>Machine Learning</i> Dengan Teknik <i>Supervised</i> dan <i>Unsupervised Learning</i> Menggunakan Python	<i>Supervised</i> dan <i>Unsupervised Learning</i> (teoritis)	Memberikan pemahaman dasar mengenai klasifikasi dan <i>Clustering</i> menggunakan Python secara konseptual.
5	Delvina Nur Rahmawati	2023	Perbandingan Algoritma <i>Partitioning</i> dan <i>Hierarchical Clustering</i> dalam Pengelompokan Data Tingkat Pengangguran Terbuka di Pulau Jawa	<i>K-Means</i> , <i>Hierarchical Clustering</i>	Metode Ward ( <i>hierarki</i> , 2 klaster) paling optimal berdasarkan nilai <i>agglomerative coefficient</i> .
6	Romadan syah Siagian	2022	Penerapan Algoritma <i>K-Means</i> dan <i>K-Medoids</i> untuk Segmentasi Pelanggan pada Data <i>E-Commerce</i> Menggunakan Model LRFM	<i>K-Means</i> , <i>K-Medoids</i>	<i>K-Medoids</i> lebih unggul dari <i>K-Means</i> dengan nilai <i>DBI</i> lebih kecil; jumlah klaster optimal adalah 3.
7	Winanda Arsyad	2023	Model Operasional Bisnis <i>E-Commerce</i> (Studi Kasus Tokopedia)	Kualitatif deskriptif	Tokopedia menerapkan model operasional dengan fitur seperti skor toko, deposit, dan laporan GoTo untuk meningkatkan performa dan kualitas layanan.
8	Sekar Dani Nurul Aini	2024	Pengaruh Online <i>Customer Review</i> , Online	Kuantitatif, regresi linier	Online <i>customer review</i> , <i>Rating</i> , dan digital <i>payment</i> berpengaruh positif dan signifikan

---

<i>Customer Rating dan Digital Payment terhadap Keputusan Pembelian Online Generasi Z Shopee</i>	terhadap keputusan pembelian online generasi Z pengguna Shopee.
--	---

---

### III. METODOLOGI PENELITIAN

#### 3.1 Waktu Penelitian

Penelitian ini direncanakan berlangsung dari Maret 2025 hingga September 2025, dengan tahapan-tahapan yang disusun secara sistematis berdasarkan kebutuhan proses klasterisasi data produk Tokopedia. Penentuan waktu dilakukan untuk menjamin setiap proses berjalan efektif, mulai dari pengumpulan data hingga analisis dan penyusunan laporan akhir. Rincian tahapan penelitian dapat dilihat pada Tabel 7.

**Tabel 7.** Waktu Penelitian

Tahapan Penelitian	Mar	Apr	Mei	Jun	Jul	Ags	Sept
<i>Literature Review</i>							
Pengumpulan Data							
<i>Preprocessing Data</i>							
Transormasi Data							
Penentuan Klaster Optimal							
Implementasi Algoritma							
Evaluasi Hasil <i>Clustering</i>							
Penulisan Skripsi							

#### 3.2 Alat dan Bahan Penelitian

Alat dan bahan yang digunakan bertujuan untuk menunjang proses pengolahan data hingga tahap evaluasi model klasterisasi. Adapun alat terdiri dari perangkat keras dan perangkat lunak yang mendukung proses penelitian, sedangkan bahan penelitian berupa *dataset* produk Tokopedia. Spesifikasi perangkat keras ditampilkan pada Tabel 8, sedangkan perangkat lunak yang digunakan disajikan pada Tabel 9.

**Tabel 8.** Spesifikasi Perangkat Keras

Spesifikasi	Keterangan
Processor	MacBook Air (Chip M1)
OS	macOS
RAM	8 GB
Storage	128 GB SSD

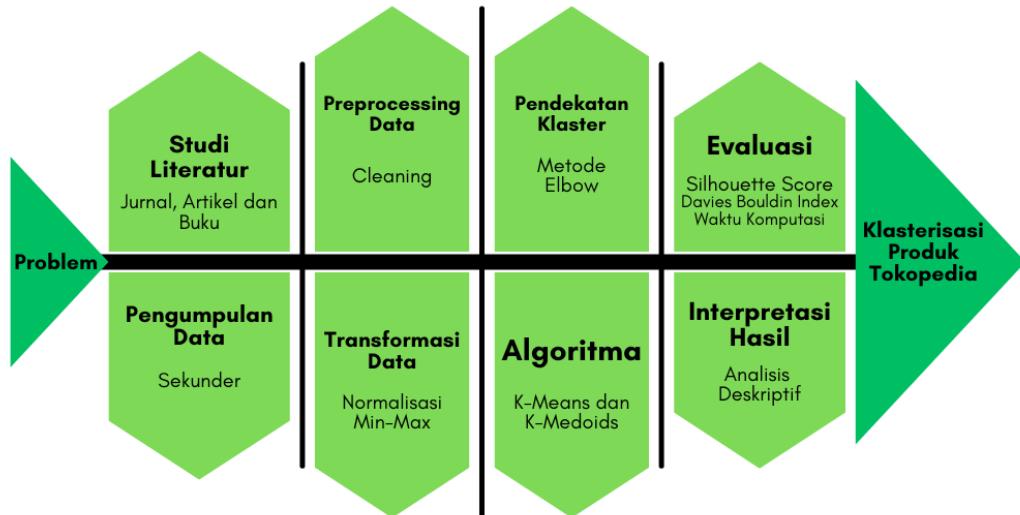
**Tabel 9.** Perangkat Lunak

<b>Nama</b>	<b>Keterangan</b>
<i>Python</i>	Bahasa utama untuk analisis data dan algoritma klasterisasi.
<i>Google Colab</i>	IDE berbasis cloud untuk menjalankan kode <i>Python</i> dan menyimpan hasil.
<i>Pandas</i>	Mengelola dan menganalisis data dalam bentuk tabel.
<i>NumPy</i>	Operasi numerik dan perhitungan jarak antar data.
<i>Matplotlib</i>	Membuat grafik dan visualisasi hasil klasterisasi.
<i>Seaborn</i>	Membuat visualisasi statistik seperti <i>boxplot</i> dan <i>countplot</i> .
<i>Scikit-learn</i>	Untuk normalisasi, evaluasi klaster, dan implementasi <i>K-Means</i> .
<i>PyClustering</i>	Library eksternal untuk implementasi algoritma <i>K-Medoids</i> .
<i>IPython.display</i>	Menampilkan tabel hasil klaster secara interaktif di notebook.
<i>Time</i>	Mengukur waktu proses algoritma klasterisasi.

### 3.3 Tahapan Penelitian

*Fishbone Diagram* pada penelitian ini berfungsi untuk memvisualisasikan alur tahapan serta faktor-faktor penting yang berperan dalam analisis klaster produk Tokopedia. Visualisasi ini membantu menggambarkan hubungan antara langkah-langkah utama dalam penelitian secara sistematis, mulai dari perumusan masalah hingga interpretasi hasil.

Penggunaan diagram ini juga memperjelas bahwa keberhasilan analisis dipengaruhi oleh keterpaduan tiap tahap, seperti studi literatur, pengumpulan data, *preprocessing* data, transformasi, pendekatan klaster, pemilihan metode, evaluasi dan interpretasi hasil. Dengan demikian, *Fishbone Diagram* tidak hanya menjadi alat bantu visual, serta memperkuat metodologi. Alur tahapan penelitian pada *Fishbone Diagram* ditunjukkan pada Gambar 6.



**Gambar 6.** Fishbone Diagram

### Studi Literatur

Proses penelitian ini diawali dengan pengumpulan data mentah dari *dataset* PRDECT-ID yang berisi informasi produk Tokopedia, kemudian dilakukan *assessing* data untuk memahami struktur, tipe variabel, serta kelengkapan data. Selanjutnya dilakukan standarisasi format dan pemetaan atribut untuk menyesuaikan variabel numerik dan kategorikal sesuai kebutuhan analisis. Tahap data *cleaning* mencakup seleksi atribut relevan, standarisasi teks, pemetaan kategori, penanganan nilai kosong, penghapusan data duplikat, serta deteksi dan penanganan *outlier* menggunakan metode *Interquartile Range* (*IQR*). Data yang telah bersih kemudian melalui transformasi *Min-Max Scaling* untuk menormalkan nilai atribut numerik ke rentang 0–1, sehingga variabel dengan skala besar tidak mendominasi proses klasterisasi.

Setelah data siap digunakan, dilakukan penentuan jumlah klaster optimal menggunakan metode Elbow untuk memperoleh nilai *K* yang representatif. Implementasi algoritma melibatkan *K-Means*, yang menggunakan rata-rata atribut (*centroid*) sebagai pusat klaster, dan *K-Medoids*, yang menggunakan titik data aktual (*medoid*) sebagai pusatnya dan lebih tahan terhadap *outlier*. Kedua algoritma dijalankan secara iteratif hingga konvergen, yaitu ketika pusat klaster tidak lagi berubah signifikan. Tahap akhir adalah evaluasi hasil klasterisasi menggunakan metrik *Davies Bouldin Index* (*DBI*) untuk mengukur kemiripan antar klaster, dan *Silhouette Score* untuk menilai ketepatan pengelompokan data. Analisis deskriptif juga dilakukan untuk mengidentifikasi karakteristik unik setiap klaster, termasuk distribusi kategori produk dan strategi pemasaran yang relevan berdasarkan kecenderungan pola pada masing-masing kelompok.

## **Pengumpulan Data**

Pengumpulan data dalam penelitian ini dilakukan dengan memanfaatkan sumber data sekunder yang berasal dari *dataset PRDECT-ID (Product Reviews Dataset for Emotion Classification Tasks – Indonesian)*. *Dataset* ini dikembangkan oleh Sutoyo et al, 2022 dan dipublikasikan melalui platform Mendeley Data pada tahun 2022. *Dataset* tersebut berisi 5.401 entri ulasan produk yang diambil secara langsung dari platform *e-commerce* Tokopedia, sehingga merepresentasikan kondisi pasar dan interaksi konsumen pada lingkungan *e-commerce* di Indonesia. Data yang tersedia mencakup berbagai atribut penting seperti *Category*, *Product Name*, *Location*, *Price*, *Overall Rating*, *Number Sold*, *Total Review*, *Customer Rating*, *Customer Review*, *Sentiment*, dan *Emotion*. Dalam penelitian ini, fokus analisis diarahkan pada atribut numerik seperti *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*, karena atribut tersebut dinilai paling relevan dalam membentuk segmentasi produk berdasarkan performa penjualan dan persepsi konsumen. Keberadaan atribut kategori (*Category*) tetap dimanfaatkan pada tahap interpretasi hasil klasterisasi untuk mengidentifikasi distribusi produk berdasarkan domain bisnis. Sebelum dilakukan analisis lebih lanjut, *dataset* ini terlebih dahulu melalui proses pra-pemrosesan untuk memastikan kualitas data yang digunakan, termasuk pembersihan data, penanganan *outlier*, dan normalisasi.

## **Preprocessing Data**

Tahap *preprocessing* merupakan proses penting untuk memastikan data berada dalam kondisi terbaik sebelum dianalisis. Data yang tidak bersih, memiliki nilai yang hilang, atau mengandung format yang tidak konsisten dapat mengurangi akurasi hasil klasterisasi. Pada penelitian ini, *preprocessing* dilakukan secara sistematis untuk menghapus *inkonsistensi*, menyatukan format, menangani data ekstrem, serta menyeleksi atribut yang relevan. Berikut adalah langkah-langkah yang dilakukan:

**Assessing Data.** Pemeriksaan awal dilakukan untuk memahami kondisi keseluruhan *dataset*, meliputi struktur, jenis atribut, dan distribusi nilai. Tujuannya adalah mengidentifikasi potensi masalah seperti nilai kosong (*missing values*), duplikasi entri, ketidak konsistenan penulisan kategori, serta data ekstrem yang tidak wajar. Pada tahap ini juga dilakukan pengecekan apakah format data sesuai dengan tipe yang diharapkan, misalnya kolom harga (*Price*) berbentuk numerik dan kolom kategori berbentuk teks. Hasil pemeriksaan ini menjadi acuan langkah pembersihan yang diperlukan pada tahap berikutnya.

**Standarisasi Teks.** Kolom berbasis teks, terutama *Product Name* dan *Category*, sering kali memiliki variasi penulisan akibat penggunaan huruf besar kecil yang berbeda, penambahan spasi, atau simbol yang tidak relevan. Standarisasi dilakukan dengan mengubah semua teks menjadi huruf kecil (*lowercase*), menghapus spasi berlebih, dan menyingkirkan karakter khusus yang tidak diperlukan. Tujuan dari proses ini adalah menyamakan format penulisan sehingga produk atau kategori yang sama tidak terdeteksi sebagai entri yang berbeda.

**Pemetaan Kategori ke Main Category.** Kategori pada *dataset* awal sangat beragam dan bersifat spesifik, misalnya men's fashion, beauty, atau computers and laptops. Agar proses analisis menjadi lebih terstruktur, kategori-kategori ini dipetakan ke dalam delapan kelompok utama (*Main Category*), yaitu: Fashion, Kesehatan, Rumah, Elektronik, Hiburan, Otomotif, Travel, dan Umum. Pemetaan ini membantu mengurangi kompleksitas data, mempermudah visualisasi, dan memfasilitasi interpretasi hasil klasterisasi karena setiap kelompok merepresentasikan jenis produk yang memiliki karakteristik serupa.

**Pembersihan Data Kosong dan Duplikat.** Data yang memiliki nilai kosong pada atribut penting seperti *Product Name*, *Category*, *Price*, *Customer Rating*, *Number Sold*, atau *Total Review* dihapus untuk menjaga kelengkapan informasi. Selain itu, baris data yang teridentifikasi sebagai duplikasi juga dihapus, terutama jika nilai seluruh atributnya identik. Pembersihan ini bertujuan untuk mencegah bias pada proses pembentukan klaster, di mana data yang berulang dapat mempengaruhi posisi *centroid* atau *medoid* secara tidak proporsional.

**Seleksi Kategori dengan Jumlah Data Memadai.** Tidak semua kategori produk memiliki jumlah entri yang cukup untuk dianalisis. Oleh karena itu, dilakukan seleksi untuk mempertahankan hanya kategori utama yang memiliki jumlah data memadai agar distribusi antar klaster seimbang. Kategori dengan entri sangat sedikit dihapus dari *dataset* final untuk menghindari dominasi klaster oleh kategori dengan representasi besar.

**Penanganan Outlier pada Atribut Numerik.** Atribut numerik seperti *Price*, *Number Sold*, dan *Total Review* dianalisis untuk mendeteksi *outlier*, yaitu nilai yang terlalu tinggi atau terlalu rendah dibandingkan mayoritas data. Metode *Interquartile Range (IQR)* digunakan untuk menentukan batas bawah dan batas atas data yang dianggap wajar. Nilai yang berada di luar rentang ini dihapus agar tidak mempengaruhi perhitungan jarak pada algoritma klasterisasi. Atribut *Customer Rating* tidak dianalisis menggunakan *IQR* karena menggunakan skala tetap (1–5) yang tidak memiliki variasi ekstrem.

**Tabel 10.** Contoh 10 Data Setelah Cleaning

No	Category	Main Category	Product Name	Price	Customer Rating	Number Sold	Total Review
1.	mother and baby	Kesehatan	kodomo baby tisu basah anti bacterial 50 sheets	37400	2	5309	1636
2.	phones and tablets	Elektronik	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	2114000	1	17400	4629
3.	mother and baby	Kesehatan	transpulmin baby balsam - 20gr	71390	4	4098	1925
4	health	Kesehatan	tim ayam obat herbal komplit 12 macam	30000	3	7474	1311
5.	muslim fashion	Fashion	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capucino	28999	3	11700	2819
6.	men's fashion	Fashion	baju seragam pgri kemeja hem katun pria batik pgri dinas pns panjang - s	119900	3	525	259
7.	sport	Otomotif	basic headband black / bando hitam pria wanita / aksesoris olahraga	4900	2	2806	491
8.	animal care	Rumah	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	79000	1	8243	3964
9.	books	Hiburan	i saw the same dream again	83000	5	234	152
10.	toys and hobbies	Hiburan	tenda bermain anak model castle (biru/pink)	120000	1	1592	1085

Tahap akhir *preprocessing* adalah memilih atribut yang relevan untuk proses klasterisasi, yaitu *Category*, *Main Category*, *Product Name*, *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*. Pemilihan atribut ini didasarkan pada relevansi langsung terhadap karakteristik produk yang menjadi fokus segmentasi. Hasil data setelah melalui tahap *cleaning* ditunjukkan pada Tabel 10, yang memperlihatkan 10 entri contoh pasca *preprocessing*. *Dataset* yang telah dibersihkan kemudian siap digunakan untuk tahap transformasi skala menggunakan metode *Min-Max Scaling* yang akan memastikan setiap atribut memiliki rentang nilai yang sebanding pada tahap analisis berikutnya.

### **Transformasi Data**

Setelah data dibersihkan, langkah berikutnya adalah melakukan transformasi data. Transformasi ini bertujuan untuk menyamakan skala antar atribut numerik sehingga tidak ada satu atribut yang mendominasi perhitungan jarak pada proses klasterisasi. Pada penelitian ini, transformasi diterapkan pada empat atribut numerik, yaitu *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*. Perbedaan rentang nilai antar atribut ini cukup signifikan, misalnya harga produk berada pada kisaran ribuan hingga jutaan rupiah, sedangkan *rating* hanya berada pada skala 1–5. Untuk itu, digunakan teknik *Min-Max Scaling*, yaitu metode normalisasi yang mengubah nilai asli atribut ke dalam rentang 0 hingga 1. Rumus yang digunakan adalah:

$$X_{\text{normalisasi}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad ..(5)$$

Di mana:

- $X$  adalah nilai asli dari suatu atribut
- $X_{\min}$  adalah nilai minimum dalam atribut tersebut
- $X_{\max}$  adalah nilai maksimum dalam atribut tersebut

#### a. Normalisasi Atribut *Price*

1. Data Harga (*Price*): [37400, 2114000, 71390, 30000, 28999, 119900, 4900, 79000, 83000, 120000]
2. Nilai minimum ( $X_{\min}$ ) = 4,900
3. Nilai maksimum ( $X_{\max}$ ) = 2,114,000

Contoh perhitungan untuk baris pertama:

$$X_{\text{Normalise}} \frac{37400 - 4900}{2114000 - 4900} = 0,015$$

Hasil normalisasi *Price*:

[0,015, 1,000, 0,032, 0,012, 0,011, 0,055, 0,000, 0,035, 0,037, 0,055]

b. Normalisasi Atribut *Customer Rating*

1. Data *Rating*: [2, 1, 4, 3, 3, 3, 2, 1, 5, 1]
2. Nilai minimum ( $X_{min}$ ) = 1
3. Nilai maksimum ( $X_{max}$ ) = 5

Contoh perhitungan baris pertama:

$$X_{Normalise} \frac{2 - 1}{5 - 1} = 0,25$$

Hasil normalisasi *Customer Rating*:

[0,25, 0,000, 0,75, 0,50, 0,50, 0,50, 0,25, 0,000, 1,000, 0,000]

c. Normalisasi Atribut *Number Sold*

1. Data *Number Sold*: [5309, 17400, 4098, 7474, 11700, 525, 2806, 8243, 234, 1592]
2. Nilai minimum ( $X_{min}$ ) = 234
3. Nilai maksimum ( $X_{max}$ ) = 17.400

Contoh perhitungan baris pertama:

$$X_{Normalise} \frac{5309 - 234}{17400 - 234} = 0,296$$

Hasil normalisasi *Number Sold*

[0,296, 1,000, 0,225, 0,422, 0,668, 0,017, 0,150, 0,467, 0,000, 0,079]

d. Normalisasi Atribut *Total Review*

1. Data *Total Review*: [1636, 4629, 1925, 1311, 2819, 259, 491, 3964, 152, 1085]
2. Nilai minimum ( $X_{min}$ ) = 152
3. Nilai maksimum ( $X_{max}$ ) = 4.629

Contoh perhitungan baris pertama:

$$X_{Normalise} \frac{1636 - 152}{4629 - 152} = 0,331$$

Hasil normalisasi *Total Review*:

[0,331, 1,000, 0,396, 0,259, 0,596, 0,024, 0,076, 0,851, 0,000, 0,208]

**Tabel 11.** Contoh 10 Data Setelah Transformasi

No	Category	Main Category	Product Name	Price	Customer Rating	Number Sold	Total Review
1	mother and baby	Kesehatan	kodomo baby tisu basah anti bacterial 50 sheets	0,015	0,25	0,296	0,331
2	phones and tablets	Elektronik	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	1,000	0,000	1,000	1,000
3	mother and baby	Kesehatan	transpulmin baby balsam - 20gr	0,032	0,75	0,225	0,396
4	health	Kesehatan	tim ayam obat herbal komplit 12 macam	0,012	0,50	0,422	0,259
5	muslim fashion	Fashion	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capucino	0,011	0,50	0,668	0,596
6	men's fashion	Fashion	baju seragam pgri kemeja hem katun pria batik pgri dinas pns panjang - s	0,055	0,50	0,017	0,024
7	sport	Otomotif	basic headband black / bando hitam pria wanita / aksesoris olahraga	0,000	0,25	0,150	0,076
8	animal care	Rumah	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	0,035	0,000	0,467	0,851
9	books	Hiburan	i saw the same dream again	0,037	1,000	0,000	0,000
10	toys and hobbies	Hiburan	tenda bermain anak model castle (biru/pink)	0,055	0,000	0,079	0,208

Setelah proses transformasi dilakukan, seluruh atribut numerik telah berhasil dinormalisasi ke dalam skala 0 hingga 1 sehingga tidak ada atribut yang mendominasi perhitungan jarak. Tabel 11 menampilkan contoh 10 data setelah dilakukan transformasi menggunakan metode Min-Max Scaling.

### Pendektan Klaster Optimal

Sebelum melakukan klasterisasi menggunakan *K-Means* maupun *K-Medoids*, diperlukan penentuan jumlah klaster (*K*) yang optimal. Jumlah klaster yang tepat akan menghasilkan pengelompokan data yang representatif, dengan perbedaan antar klaster yang jelas. Penentuan jumlah klaster dilakukan menggunakan Metode Elbow, yang merupakan teknik populer dalam analisis klaster.

**Konsep Metode Elbow.** Metode Elbow bekerja dengan mengukur *compactness* klaster, yaitu sejauh mana anggota klaster berkumpul di sekitar pusat klaster. Ukuran ini biasanya dihitung sebagai:

- Inertia* pada *K-Means*, yaitu total kuadrat jarak antar data terhadap *centroid*:

$$Inertia = \sum_{j=1}^K \sum_{x \in C_j} d(x, c_j)^2 \quad ..(6)$$

Keterangan:

1.  $K$  = jumlah klaster yang diuji
  2.  $C_j$  = anggota klaster ke  $j$
  3.  $c_j$  = *centroid* klaster ke  $j$
  4.  $d(x, c_j)^2$  = jarak *Euclidean* antara data  $x$  dan *centroid*  $C_j$
  5. *Inertia* = ukuran seberapa rapat anggota klaster dengan *centroid*, semakin kecil berarti klaster lebih *compact*
- b. Total *Cost* pada *K-Medoids*, yaitu jumlah jarak antar data ke *medoid*:

$$Total Cost = \sum_{j=1}^K \sum_{x \in C_j} d(x, m_j) \quad ..(7)$$

Keterangan:

1.  $m_j$  = *medoid* klaster ke  $j$ , yaitu anggota data yang paling representatif dalam klaster
  2.  $d(x, m_j)$  = jarak antara data  $x$  dan *medoid*  $m_j$
  3. Total *Cost* = ukuran *compactness* klaster untuk *K-Medoids*, semakin kecil berarti anggota klaster lebih dekat ke *medoid*
- c. Jarak *Euclidean* yang digunakan dihitung dengan rumus:

$$D(x, c_j) = \sqrt{\sum_{i=1}^n (x_i - c_{j,i})^2} \quad ..(8)$$

Keterangan:

1.  $n$  = jumlah atribut numerik
2.  $x_i$  = nilai atribut ke  $i$  dari data
3.  $c_{j,i}$  = nilai atribut ke  $i$  dari *centroid* klaster ke  $j$  (*K-Means*) atau *medoid* klaster ke  $j$  (*K-Medoids*)

Konsep Elbow adalah memilih K pada titik di mana penurunan *inertia* atau total *cost* mulai melandai. Titik ini menunjukkan jumlah klaster optimal, penambahan klaster lebih lanjut memberikan perbaikan minimal terhadap kepadatan klaster.

#### Langkah-langkah Metode Elbow pada *K-Means*

1. Menentukan Rentang K

Pilih rentang jumlah klaster yang diuji dari  $K = 1$  hingga  $K = 10$ , agar cakupan kemungkinan jumlah klaster lebih luas.

2. Klasterisasi Sementara

Untuk tiap K, lakukan klasterisasi sementara menggunakan *K-Means*. Hitung *centroid* setiap klaster dan tempatkan data pada klaster terdekat berdasarkan jarak *Euclidean*.

3. Menghitung *Inertia*

Hitung nilai *inertia* untuk setiap K:

$$\text{Inertia } K = \sum_{j=1}^K \sum_{x \in C_j} \sum_{i=j}^n d(x_i - c_{j,i})^2 \quad ..(9)$$

Keterangan:

a. *Inertia K* = ukuran kepadatan klaster untuk jumlah klaster K

b.  $x_i$  = atribut ke i dari data anggota klaster

c.  $c_{j,i}$  = atribut ke i dari *centroid* klaster ke j

d. Semakin kecil *inertia*, anggota klaster lebih dekat dengan *centroid*

4. Membuat Plot *Inertia* vs K

Buat grafik sumbu horizontal = K (1–10) dan sumbu vertikal = *inertia* untuk melihat titik di mana penurunan mulai melandai.

5. Menentukan Titik Elbow

Titik Elbow menunjukkan jumlah klaster optimal. Misalnya, jika penurunan signifikan terlihat dari  $K = 1$  ke  $K = 2$  dan selanjutnya melandai,  $K = 2$  dipilih sebagai jumlah klaster optimal.

#### Langkah-langkah Metode Elbow pada *K-Medoids*

1. Menentukan Rentang K

Tentukan  $K = 1$  hingga  $K = 10$ , sama seperti *K-Means*.

2. Klasterisasi Sementara

Lakukan klasterisasi sementara menggunakan *K-Medoids*. Pilih *medoid* awal dan kelompokkan setiap data ke *medoid* terdekat.

### 3. Menghitung Total Cost

Hitung total *cost* untuk tiap K:

$$\text{Total Cost} = \sum_{j=1}^K \sum_{x \in C_j} d(x - m_j) \quad ..(10)$$

Keterangan:

- a.  $m_j$  = *medoid* klaster ke j, data yang paling representatif
- b.  $d(x - m_j)$  = jarak antara anggota klaster dan *medoid*
- c. Semakin kecil total *cost*, anggota klaster lebih dekat ke *medoid*

### 4. Membuat Plot Total Cost vs K

Grafik sumbu horizontal = K (1–10), sumbu vertikal = total *cost*, untuk menentukan titik Elbow.

### 5. Menentukan Titik Elbow untuk *K-Medoids*

Titik Elbow pada *K-Medoids* ditentukan saat penurunan total *cost* mulai melandai. Untuk perbandingan hasil klasterisasi, jumlah klaster optimal dapat mengikuti hasil Elbow *K-Means*.

## Implementasi Algoritma

**Implementasi *K-Means*.** Tahapan implementasi algoritma *K-Means* pada penelitian ini dilakukan menggunakan data hasil transformasi *Min-Max Scaling* terhadap atribut numerik *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*. Algoritma *K-Means* mengelompokkan data dengan mencari pusat klaster (*centroid*) yang dihitung sebagai nilai rata-rata dari atribut setiap anggota klaster, kemudian memindahkan data ke klaster terdekat secara bertahap hingga posisi *centroid* stabil. Perhitungan dilakukan secara berurutan untuk memudahkan penelusuran proses.

### 1. Menentukan Jumlah Cluster

Tahap awal adalah menentukan berapa banyak klaster yang akan digunakan. Dalam penelitian ini digunakan metode Elbow, di mana nilai *inertia* dihitung untuk berbagai nilai K (misalnya K=1 sampai K=10). *Inertia* adalah jumlah kuadrat jarak setiap titik data ke *centroid* terdekat. Nilai K yang tepat ditentukan pada titik *elbow*, yaitu titik di grafik di mana penurunan *inertia* mulai melandai, sehingga penambahan klaster berikutnya tidak memberikan peningkatan signifikan pada kualitas pengelompokan. Berdasarkan analisis ini, diperoleh nilai optimal K = 2.

2. Menentukan *Centroid* Awal

Setelah jumlah klaster ditentukan, dipilih *centroid* awal secara acak dari data yang telah dinormalisasi. Pemilihan ini penting karena posisi awal *centroid* akan memengaruhi jalannya proses iterasi. Pada penelitian ini, *centroid* awal dipilih sebagai berikut:

- a. C1: Data ke 2 [1.000, 0.000, 1.000, 1.000]
- b. C2: Data ke 3 [0.032, 0.750, 0.225, 0.396]

*Centroid* ini mewakili titik awal dari masing-masing klaster sebelum proses perhitungan jarak dilakukan.

3. Menghitung Jarak *Euclidean*

Jarak setiap produk terhadap masing-masing *centroid* dihitung menggunakan rumus *Euclidean Distance* berikut:

$$D(x, c) = \sum_{i=1}^n (x_i - c_i)^2 \quad ..(11)$$

Keeterangan:

- $D(x, c)$  = Jarak data  $x$  ke *centroid*  $c$
- $x_i$  = Nilai atribut ke  $i$  pada data
- $c_i$  = Nilai atribut ke  $i$  pada *centroid*
- $n$  = Jumlah atribut numerik (4 atribut pada penelitian ini)

4. Menentuan Klaster

Berdasarkan hasil jarak *Euclidean*, setiap data dimasukkan ke klaster yang memiliki jarak terdekat. Pada tahap ini, terbentuk dua klaster sementara yang menjadi dasar perhitungan *centroid* baru pada iterasi berikutnya.

5. Pembaruan *Centroid*

Setelah semua data diberi label klaster, *centroid* diperbarui dengan menghitung rata-rata setiap atribut dari anggota klaster:

$$c_j = \frac{1}{N_i} \sum_{k=1}^{N_i} X_{kj} \quad ..(12)$$

Keterangan :

- $c_j$  = Nilai *centroid* baru untuk fitur ke  $j$
- $N_i$  = Jumlah data dalam *cluster* ke  $i$
- $X_{kj}$  = Nilai fitur ke- $j$  dari produk ke- $k$  dalam *cluster*

## 6. Iterasi Hingga Konvergen

Langkah perhitungan jarak (Langkah 3), penentuan klaster (Langkah 4), dan pembaruan *centroid* (Langkah 5) diulang secara bertahap hingga *centroid* tidak lagi berubah secara signifikan. Iterasi ini memastikan setiap klaster stabil dan representatif bagi data di dalamnya.

## 7. Hasil Akhir Klasterisasi

Setelah *centroid* stabil, diperoleh dua klaster akhir. Setiap klaster memuat data dengan karakteristik serupa berdasarkan atribut *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*. Hasil akhir ini kemudian dianalisis untuk mengetahui profil masing-masing klaster dan hubungannya dengan kategori produk.

**Implementasi *K-Medoids*.** Setelah implementasi *K-Means*, penelitian ini juga menggunakan algoritma *K-Medoids* untuk melakukan klasterisasi pada *dataset* yang sama. Berbeda dengan *K-Means* yang menggunakan nilai rata-rata untuk menentukan pusat klaster, *K-Medoids* memilih *medoid*, yaitu anggota data yang paling representatif atau paling sentral dalam setiap klaster. Metode ini lebih tahan terhadap *outlier*, karena posisi *medoid* selalu berupa data asli.

### 1. Menentukan Jumlah *Cluster*

Jumlah klaster K ditetapkan sama seperti pada *K-Means*, yaitu K = 2, agar hasil klasterisasi dapat dibandingkan.

### 2. Menentukan *Medoid* Awal

Setelah jumlah klaster ditentukan, langkah selanjutnya adalah pemilihan *medoid* awal. *Medoid* awal dipilih dari anggota data secara acak. *Medoid* ini menjadi pusat awal yang digunakan untuk menghitung jarak *Euclidean* antara setiap data dengan *medoid*. Pemilihan *medoid* awal yang representatif dapat mempercepat konvergen, namun *K-Medoids* tetap akan menyesuaikan posisi *medoid* secara bertahap melalui iterasi untuk meminimalkan total jarak klaster.

### 3. Menghitung Jarak *Euclidean*

Setiap data dihitung jaraknya ke semua *medoid* menggunakan rumus *Euclidean*:

$$D(x, m) = \sum_{i=1}^n (x_i - m_i)^2 \quad ..(13)$$

Setelah jarak dihitung, setiap data dimasukkan ke klaster dengan *medoid* terdekat. Proses ini memastikan bahwa anggota klaster memiliki kemiripan atribut yang tinggi dengan *medoid*, sehingga setiap klaster memiliki anggota yang homogen. Penentuan klaster ini dilakukan secara bertahap, sehingga dapat ditelusuri secara sistematis dari awal hingga *medoid* stabil.

#### 4. Evaluasi *Medoid* dan Perhitungan Total Cost

Setelah data ditempatkan dalam klaster, dilakukan evaluasi *medoid* baru untuk setiap klaster. *Medoid* baru dipilih dari anggota klaster yang meminimalkan total jarak ke seluruh anggota klaster. Rumus total cost per klaster adalah:

$$\text{Total Cost} = \sum_{i=1}^n \min(D_{iM1}, D_{iM2}) \quad ..(14)$$

*Medoid* baru dipilih berdasarkan total cost terkecil, sehingga setiap iterasi menghasilkan klaster yang lebih representatif dan total jarak intra-klaster lebih rendah. Proses evaluasi *medoid* ini dilakukan secara bertahap untuk semua klaster hingga tidak ada perubahan *medoid* lagi.

#### 5. Iterasi Hingga Konvergen

Langkah perhitungan jarak, penentuan klaster, dan evaluasi *medoid* diulang secara bertahap hingga konvergen. Konvergen terjadi ketika *medoid* tidak berubah lagi atau perubahan total cost antar iterasi sudah sangat kecil. Metode iteratif ini memastikan bahwa setiap anggota data berada pada klaster yang paling tepat dan pusat klaster mencerminkan anggota yang paling representatif. Iterasi bertahap memungkinkan peneliti untuk menelusuri proses pengelompokan secara rinci, mengetahui perubahan posisi *medoid*, dan memverifikasi bahwa pembagian klaster stabil.

#### 6. Hasil Akhir Klasterisasi

Setelah proses konvergen tercapai, diperoleh pembagian akhir klaster yang stabil dengan *medoid* sebagai pusat klaster. Hasil ini menunjukkan kelompok produk yang memiliki karakteristik serupa berdasarkan atribut numerik *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*. Penggunaan *medoid* sebagai pusat klaster membuat hasil *K-Medoids* lebih *robust* terhadap *outlier* dibandingkan *K-Means*. Hasil akhir ini kemudian dianalisis untuk memahami karakteristik setiap klaster, sehingga dapat digunakan sebagai dasar untuk pengambilan keputusan atau strategi yang relevan terhadap kategori produk dalam penelitian ini.

### Evaluasi Hasil Klasterisasi

Tahap evaluasi dilakukan untuk menilai kualitas hasil *Clustering* yang dihasilkan oleh algoritma *K-Means* dan *K-Medoids*. Evaluasi bertujuan memastikan bahwa hasil pengelompokan tidak hanya secara matematis terbentuk, tetapi juga memiliki kualitas yang baik, ditandai dengan kohesi internal yang tinggi dan separasi antar *Cluster* yang kuat. Evaluasi dilakukan menggunakan dua metode, yaitu: *Davies-Bouldin Index (DBI)*, dan *Silhouette Score*.

**Davies-Bouldin Index (DBI).** *Davies-Bouldin Index (DBI)* adalah metrik evaluasi yang digunakan untuk menilai kualitas hasil klasterisasi dengan membandingkan tingkat kemiripan antar klaster terhadap tingkat kepadatan dalam klaster itu sendiri. Semakin rendah nilai *DBI*, semakin baik kualitas klaster, karena ini berarti setiap klaster memiliki kohesi internal yang tinggi dan separasi antar klaster yang kuat.

Rumus *DBI*:

$$DBI = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left( \frac{S_i + S_j}{M_{ij}} \right) \quad ..(15)$$

Keterangan:

- a.  $K$  = jumlah klaster yang dihasilkan.
- b.  $S_i$  = rata-rata jarak setiap anggota klaster  $i$  terhadap pusat klasternya (*centroid* pada *K-Means*, *medoid* pada *K-Medoids*) menggambarkan kohesi atau tingkat kerapatan internal klaster.
- c.  $S_j$  = rata-rata jarak anggota klaster  $j$  terhadap pusat klaster  $j$ .
- d.  $M_{ij}$  = jarak antara pusat klaster  $i$  dan pusat klaster  $j$  menggambarkan separasi atau tingkat keterpisahan antar klaster.
- e. Rasio  $\left( \frac{S_i + S_j}{M_{ij}} \right)$  menunjukkan tingkat “kemiripan” dua klaster. Semakin kecil rasio, semakin berbeda kedua klaster tersebut.

#### Langkah-langkah Perhitungan *DBI*

1. Hitung Kohesi ( $s$ )

Untuk setiap klaster, hitung rata-rata jarak semua anggota klaster terhadap pusatnya:

Pada *K-Means*, pusat klaster adalah *centroid*, yaitu rata-rata nilai atribut semua anggota klaster. Pada *K-Medoids*, pusat klaster adalah *medoid*, yaitu data yang paling representatif (memiliki jarak total terpendek terhadap anggota klaster lainnya).

2. Hitung Separasi ( $M$ )

Untuk setiap pasangan klaster  $i$  dan  $j$ , hitung jarak antara pusat klaster  $i$  dan pusat klaster  $j$ . Jarak ini biasanya dihitung menggunakan *Euclidean Distance*, dan semakin besar jarak ini, semakin terpisah kedua klaster.

3. Hitung Rasio Kedekatan Antar Klaster

Untuk setiap klaster  $i$ , hitung rasio  $(\frac{s_i+s_j}{M_{ij}})$  terhadap semua klaster  $j$  yang berbeda dari  $i$ . Ambil nilai rasio terbesar untuk setiap klaster  $i$ , karena ini menunjukkan pasangan klaster yang paling “mirip” atau paling dekat.

4. Hitung Nilai  $DBI$

Hitung rata-rata dari semua nilai maksimum rasio yang diperoleh pada Langkah 3 untuk seluruh klaster. Nilai ini merupakan  $DBI$  akhir yang digunakan untuk mengevaluasi kualitas klasterisasi. Semakin kecil nilai  $DBI$ , semakin baik kualitas klaster, dengan nilai mendekati 0 menunjukkan bahwa klaster sangat kompak dan terpisah jelas.

**Silhouette Score.** *Silhouette Score* adalah metrik evaluasi yang mengukur seberapa mirip suatu objek dengan anggota klasternya sendiri (kohesi) dibandingkan dengan objek di klaster terdekat lainnya (separasi). Nilai *Silhouette* berada pada rentang -1 hingga 1, dengan interpretasi:

- Nilai mendekati +1 menunjukkan objek sesuai dengan *Cluster* nya
- Nilai mendekati 0 berarti objek berada di antara dua *Cluster*
- Nilai mendekati -1 menunjukkan objek kemungkinan salah *Cluster*

Rumus *Silhouette Score* :

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad ..(16)$$

Keterangan:

- $a(i)$ = rata-rata jarak objek  $i$  ke semua objek lain di klaster yang sama (kohesi internal).
- $b(i)$  = jarak rata-rata objek  $i$  ke semua objek di klaster terdekat yang berbeda (separasi eksternal).

Langkah-langkah Perhitungan *Silhouette Score*

1. Hitung Kohesi Internal  $a(i)$

Untuk setiap data  $i$ , hitung rata-rata jarak ke semua anggota klaster yang sama.

Pada *K-Means*, jarak dihitung dari data ke sesama anggota menggunakan *Euclidean distance* berbasis koordinat atribut. Pada *K-Medoids*, jarak dihitung dari data ke sesama anggota menggunakan jarak ke *medoid* atau antar titik data aktual.

2. Hitung Separasi Eksternal  $b(i)$

Untuk setiap data  $i$ , hitung jarak rata-rata ke semua anggota di klaster terdekat yang berbeda. Klaster terdekat adalah yang memiliki nilai jarak rata-rata terkecil.

3. Hitung Nilai *Silhouette*  $s(i)$

Gunakan rumus:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad ..(17)$$

Nilai ini dihitung untuk setiap data.

4. Hitung Nilai *Silhouette* Rata-rata

Ambil rata-rata nilai  $s(i)$  dari seluruh data. Nilai ini menjadi *Silhouette Score* akhir yang digunakan untuk mengevaluasi kualitas klasterisasi. Semakin tinggi nilainya, semakin baik kualitas klaster.

Pada Bab 4, perhitungan dan analisis *Davies-Bouldin Index (DBI)* dan *Silhouette Score* akan dilakukan secara terpisah untuk hasil klasterisasi menggunakan algoritma *K-Means* dan *K-Medoids*. Hal ini bertujuan untuk mendapatkan gambaran yang jelas mengenai performa masing-masing algoritma, serta memastikan evaluasi yang dilakukan bersifat objektif dan tidak saling mempengaruhi.

**Waktu Komputasi.** Waktu komputasi digunakan sebagai salah satu parameter tambahan untuk membandingkan efisiensi algoritma *K-Means* dan *K-Medoids* dalam proses klasterisasi data produk Tokopedia. Pengukuran dilakukan dengan mencatat durasi eksekusi masing-masing algoritma, dimulai dari tahap inisialisasi hingga mencapai batas konvergen.

Pengukuran dilakukan menggunakan fungsi `time.time()` dalam bahasa pemrograman *Python* dengan mencatat selisih waktu sebelum dan sesudah proses klasterisasi dijalankan. Hasil pengukuran ini disajikan dalam satuan detik dengan presisi empat angka di belakang koma, dan menjadi bagian dari analisis perbandingan performa kedua algoritma dalam penelitian ini.

### **Interpretasi Hasil Klasterisasi**

Setelah seluruh proses *preprocessing* dan transformasi selesai dilakukan, langkah selanjutnya adalah menginterpretasikan hasil klasterisasi yang diperoleh dari penerapan algoritma *K-Means* dan *K-Medoids*. Proses interpretasi dilakukan dengan beberapa tahapan berikut:

1. Pengelompokan Data Berdasarkan Hasil Klasterisasi

Data yang telah melalui proses klasterisasi dikelompokkan ke dalam masing-masing klaster berdasarkan label yang dihasilkan oleh algoritma.

2. Analisis Ciri-Ciri Masing-Masing Klaster

Rata-rata nilai dari atribut numerik (seperti *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*) dihitung untuk setiap klaster guna mengetahui karakteristik dominan dari setiap kelompok produk.

3. Distribusi Kategori Produk dalam Tiap Klaster

Informasi atribut Kategori yang sebelumnya tidak digunakan dalam proses klasterisasi dimanfaatkan kembali untuk melihat bagaimana sebaran kategori produk dalam masing-masing klaster.

4. Evaluasi Kuantitatif Klasterisasi

Kualitas klasterisasi dinilai menggunakan metrik evaluasi seperti *Silhouette Score* dan *Davies-Bouldin Index (DBI)* untuk masing-masing algoritma, sebagai dasar perbandingan performa.

5. Pengukuran Waktu Komputasi Algoritma

Lama waktu proses klasterisasi diukur untuk setiap algoritma, yaitu *K-Means* dan *K-Medoids*, sebagai indikator efisiensi proses. Nilai waktu komputasi ini diperoleh dengan mencatat durasi mulai hingga klasterisasi selesai pada masing-masing metode.

## IV. HASIL DAN PEMBAHASAN

Bab ini menjelaskan secara rinci langkah-langkah penelitian yang telah dilaksanakan hingga diperoleh hasil akhir, sesuai dengan metode yang telah dijabarkan pada bab sebelumnya.

### 4.1 Pengumpulan Data

Penelitian ini menggunakan data sekunder yang bersumber dari *dataset* PRDECT-ID (*Product Reviews Dataset for Emotions Classification Tasks - Indonesian*). *Dataset* ini dikembangkan oleh tim peneliti Universitas Bina Nusantara dan dipublikasikan melalui platform Mendeley Data pada tahun 2022 dengan lisensi Creative Commons (CC BY 4.0) Sutoyo et al, 2022. *Dataset* tersebut memuat 5.401 entri ulasan produk asli dari platform e-commerce Tokopedia, dan mencakup atribut seperti *Category*, *Product Name*, *Location*, *Price*, *Overall Rating*, *Number Sold*, *Total Review*, *Customer Rating*, *Customer Review*, *Sentiment*, dan *Emotion*. Informasi ini digunakan sebagai dasar untuk membentuk segmentasi produk berdasarkan karakteristik numerik yang relevan. Tampilan link *dataset* dapat dilihat pada Gambar 7, sementara file *dataset* disimpan dalam folder utama pada direktori Dokumen laptop pribadi penulis.

The screenshot shows the Mendeley Data page for the "Product Reviews Dataset for Emotions Classification Tasks - Indonesian (PRDECT-ID) Dataset". Key details include:

- Mendeley Data** logo and navigation links: Find Research Data, My Data, and a user icon.
- Dataset metrics**: Views: 1884, Downloads: 611.
- Latest version**: Version 1, Published: 20 May 2022, DOI: 10.17632/574v66hf2v.1.
- Description**: PRDECT-ID Dataset is a collection of Indonesian product review data annotated with emotion and sentiment labels. The data were collected from one of the giant e-commerce in Indonesia named Tokopedia. The dataset contains product reviews from 29 product categories on Tokopedia that use the Indonesian language. Each product review is annotated with a single emotion, i.e., love, happiness, anger, fear, or sadness. The group of annotators does the annotation process to provide emotion labels by following the emotions annotation criteria created by an expert in clinical psychology. Other attributes related to the product review are also extracted, such as Location, Price, Overall Rating, Number Sold, Total Review, and Customer Rating, to support further research.
- Files**: A file named "PRDECT-ID Dataset.csv" is listed, 1.2 MB, with a download button.
- Institutions**: Bina Nusantara University.
- Categories**: Natural Language Processing, Text Processing, Consumer Emotion, Text Mining, Sentiment Analysis.
- Licence**: CC BY 4.0.

**Gambar 7.** Tampilan Link *Dataset*  
Sumber : (<https://data.mendeley.com/datasets/574v66hf2v/1>)

Sebagai gambaran kondisi awal sebelum dilakukan *preprocessing*, Tabel 12 menyajikan 10 data teratas dari dataset PRDECT-ID. Data ini masih dalam bentuk asli (raw) sehingga masih terdapat variasi penulisan, atribut teks yang panjang, serta potensi duplikasi maupun *outlier* yang kemudian perlu dilakukan tahap pembersihan.

**Tabel 12.** 10 Data Entri Awal

No	Category	Product Name	Location	Price	Overall Rating	Number Sold	Total Review	Customer Rating	Customer Review	Sentiment	Emotion
1.	Computers and Laptops	Wireless Keyboard i8 Mini TouchPad Mouse 2.4G Handheld PC Android TV	Jakarta Utara	53500	49	5449	2369	5	Alhamdulilla h berfungsi dengan baik. Packaging aman. Respon cepat dan ramah. Seller dan kurir amanah	Positive	Happy
2.	Computers and Laptops	PAKET LISENSI WINDOWS 10 PRO DAN OFFICE 2019 ORIGINAL + BONUS	Kota Tangerang Selatan	72000	49	2359	1044	5	barang bagus dan respon cepat, harga bersaing dengan yg lain.	Positive	Happy
3.	Computers and Laptops	SSD Midasforce 128 Gb - Tanpa Caddy	Jakarta Barat	21300	5	12300	3573	5	barang bagus, berfungsi dengan baik, seler ramah, pengiriman cepat	Positive	Happy

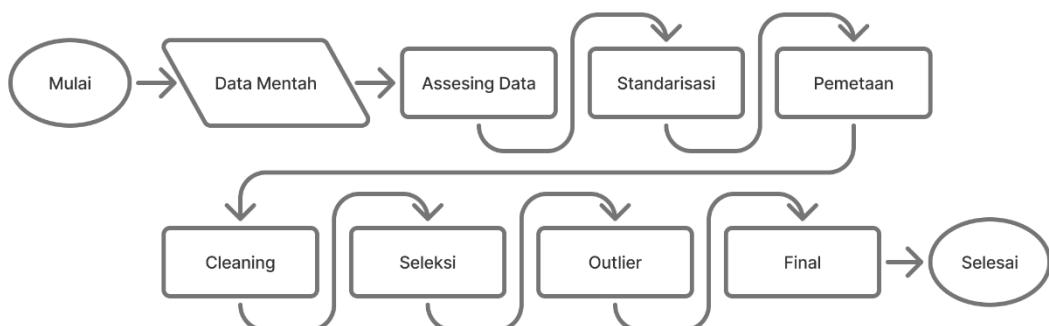
4.	Computers and Laptops	ADAPTOR CHARGER MONITOR LCD LED TV LG merek LG 19V ORIGINAL	Jakarta Timur	55000	47	2030	672	5	bagus sesuai harapan penjual nya juga ramah. trimakasih pelapak ??	Positive	Happy
5.	Computers and Laptops	ADAPTOR CHARGER MONITOR LCD LED TV LG merek LG 19V ORIGINAL	Jakarta Timur	55000	47	2030	672	5	Barang Bagus, pengemasan Aman, dapat Berfungsi dengan Baik	Positive	Happy
6.	Computers and Laptops	ADAPTOR CHARGER MONITOR LCD LED TV LG merek LG 19V ORIGINAL	Jakarta Timur	55000	47	2030	672	5	barang bagus, seller ramah..	Positive	Happy
7.	Computers and Laptops	ADAPTOR CHARGER MONITOR LCD LED TV LG merek LG 19V ORIGINAL	Jakarta Timur	55000	47	2030	672	5	barang bagus, seller ramah..	Positive	Happy

8.	Computers and Laptops	Kepala Colokan Listrik Charger Macbook untuk colokan Indonesia	Jakarta Selatan	85000	49	1339	718	5	mantap paten joss	Positive	Happy
9.	Computers and Laptops	AFL Bidirectiona 1 HDMI Switcher 1-In 2-Out & 2-In 1-Out - AFU	Jakarta Barat	13650	5 0	1201	632	5	Works fine. Respon seller cepat, barang berfungsi dengan baik. Recommanded.	Positive	Happy
10.	Computers and Laptops	AFL Bidirectiona 1 HDMI Switcher 1-In 2-Out & 2-In 1-Out - AFU	Jakarta Barat	13650	5 0	1201	632	5	barang bagus.. segel.. utuh.. original.. berfungsi dengan bener.. seller respon nya ajib.. cepat amir... uda gt kasi sarannya pas banget lg dengan kebutuhan.. mantabbb	Positive	Happy

## 4.2 Preprocessing Data

Proses selanjutnya adalah membersihkan data mentah yang telah dikumpulkan. Pada tahap awal, dilakukan *assessing data* untuk meninjau kondisi awal *dataset* sehingga dapat diketahui kualitas, kelengkapan, dan potensi masalah yang perlu ditangani. Tahap *preprocessing* ini bertujuan untuk memastikan data bersih, terstruktur, dan siap diproses. Langkah-langkah *preprocessing* meliputi: standarisasi penulisan pada kolom *Product Name* dan *Category*, pemetaan kategori ke dalam 8 kelompok *Main Category*, penghapusan nilai kosong dan duplikat, seleksi kategori dengan entri produk yang memadai, serta penanganan *outlier* pada atribut numerik *Price*, *Number Sold*, dan *Total Review* menggunakan metode *Interquartile Range (IQR)*.

Seluruh tahapan diimplementasikan menggunakan *Python* pada Google Colab, dengan bantuan pustaka *pandas*, *numpy*, dan *openpyxl*. Sebelum membaca file *.xlsx*, dilakukan instalasi dependensi dengan perintah: !pip install openpyxl. Kemudian file data diunggah ke Google Colab menggunakan perintah files.upload() dari pustaka google.colab. Setelah semua data dibersihkan, hasilnya disimpan kembali ke direktori lokal dalam bentuk file bernama ‘tokopedia\_cleaned.xlsx’. Seluruh alur tahapan *preprocessing* yang telah dijelaskan dapat dilihat secara ringkas pada Gambar 8, yang menampilkan proses mulai dari data mentah hingga data siap digunakan untuk tahap analisis berikutnya.



**Gambar 8.** Flowchart Preprocessing

### Assesing Data

*Dataset* awal berjumlah 5.401 entri produk hasil pengumpulan data dari Tokopedia. Pada tahap *assessing data*, dilakukan pemeriksaan awal untuk memahami kondisi keseluruhan *dataset*. Pemeriksaan ini mencakup pengecekan struktur dan format data, identifikasi nilai kosong, deteksi duplikasi entri, serta pengamatan terhadap kemungkinan adanya ketidak konsistensi penulisan, nilai ekstrem (*outlier*), maupun data yang tidak relevan. Hasil dari tahap ini menjadi acuan dalam menentukan langkah pembersihan dan penyesuaian pada proses analisis selanjutnya.

## Standarisasi Teks

Langkah pertama adalah melakukan standarisasi penulisan pada kolom *Product Name* dan *Category*. Semua huruf dikonversi menjadi huruf kecil (*lowercase*), dan spasi di awal atau akhir *string* dihapus. Hal ini dilakukan untuk mencegah perbedaan format yang tidak perlu yang dapat menyebabkan duplikasi atau klasifikasi yang keliru, di mana proses standarisasi ini ditunjukkan pada Gambar 9 dan hasil perbandingan sebelum serta sesudahnya dapat dilihat pada Tabel 13.



```
# Standarisasi teks kolom kategori dan produk

df['Product Name'] = df['Product
Name'].astype(str).str.strip().str.lower()
df['Category'] = df['Category'].astype(str).str.strip().str.lower()
```

**Gambar 9.** Kode Python Standarisasi Teks

**Tabel 13.** Sebelum dan Sesudah Standarisasi

<b>Product Name (Sebelum)</b>	<b>Category (Sebelum)</b>	<b>Product Name (Sesudah)</b>	<b>Category (Sesudah)</b>
GoldRET Tali	Household	goldret tali	household
Sepeda Pengikat		sepeda pengikat	
Barang Luggage		barang luggage	
Mount		mount	
BASEUS C1 MINI	Automotive	baseus c1 mini	automotive
CAR VACUUM		car vacuum	
CLEANER...		cleaner...	
Klip Orange	Camera	klip orange	camera
Backdrop clamp		backdrop clamp	
Studio Fotografi		studio fotografi	
Gangsing	Toys and Hobbies	gangsing	toys and hobbies
Beyblade Tornado		beyblade tornado	
Burst...		burst...	
Selang Pianika	Movies and Music	selang pianika	movies and music
Yamaha		yamaha	

## Pemetaan Kategori ke Kategori Utama

Kategori produk yang sebelumnya sangat beragam dan spesifik dipetakan ke dalam delapan kategori utama (*Main Category*), yaitu: Fashion, Kesehatan, Rumah, Elektronik, Hiburan, Otomotif, Travel, dan Umum. Pemetaan ini bertujuan untuk menyederhanakan klasifikasi dan memastikan bahwa analisis dilakukan pada kelompok produk yang memiliki karakteristik umum dan representatif, dengan proses pemetaan ditunjukkan pada Gambar 10 serta hasil pemetaan kategori asli ke kategori utama disajikan pada Tabel 14.

```
# Mapping kategori ke kategori utama (Main Category)
kategori_map = {
    "men's fashion": "Fashion", "women's fashion": "Fashion",
    "kids and baby fashion": "Fashion", "muslim fashion": "Fashion",
    "beauty": "Kesehatan", "body care": "Kesehatan",
    "health": "Kesehatan", "mother and baby": "Kesehatan",
    "household": "Rumah", "kitchen": "Rumah", "food and drink": "Rumah",
    "animal care": "Rumah", "party supplies and craft": "Rumah",
    "electronics": "Elektronik", "phones and tablets": "Elektronik",
    "computers and laptops": "Elektronik", "camera": "Elektronik",
    "gaming": "Hiburan", "toys and hobbies": "Hiburan",
    "movies and music": "Hiburan", "books": "Hiburan",
    "automotive": "Otomotif", "carpentry": "Otomotif",
    "sport": "Otomotif", "office & stationery": "Otomotif",
    "tour and travel": "Travel", "other products": "Umum"
}
df['Main Category'] = df['Category'].map(kategori_map).fillna('Umum')
```

**Gambar 10.** Kode Python Pemetaan Kategori**Tabel 14.** Mapping Kategori Asli ke Kategori Utama

Kategori Asli	Kategori Utama
men's fashion	Fashion
women's fashion	Fashion
kids and baby fashion	Fashion
muslim fashion	Fashion
beauty	Kesehatan
body care	Kesehatan
health	Kesehatan
mother and baby	Kesehatan
household	Rumah
kitchen	Rumah
food and drink	Rumah
animal care	Rumah
party supplies and craft	Rumah
electronics	Elektronik
phones and tablets	Elektronik
computers and laptops	Elektronik
camera	Elektronik
gaming	Hiburan
toys and hobbies	Hiburan
movies and music	Hiburan
books	Hiburan
automotive	Otomotif
carpentry	Otomotif
sport	Otomotif
office & stationery	Otomotif
tour and travel	Travel
other products	Umum

### Pembersihan Data Kosong dan Duplikat

Setelah data dibersihkan dari inkonsistensi format, langkah selanjutnya adalah menghapus baris dengan nilai kosong di kolom penting seperti *Product Name*, *Category*, *Customer Rating*, *Price*, *Number Sold*, dan *Total Review*. Selain itu, duplikasi data secara utuh juga dihapus untuk menjaga integritas analisis, dengan proses penghapusan ini diimplementasikan melalui kode pada Gambar 11 dan dampaknya terhadap jumlah data ditunjukkan pada Gambar 12.

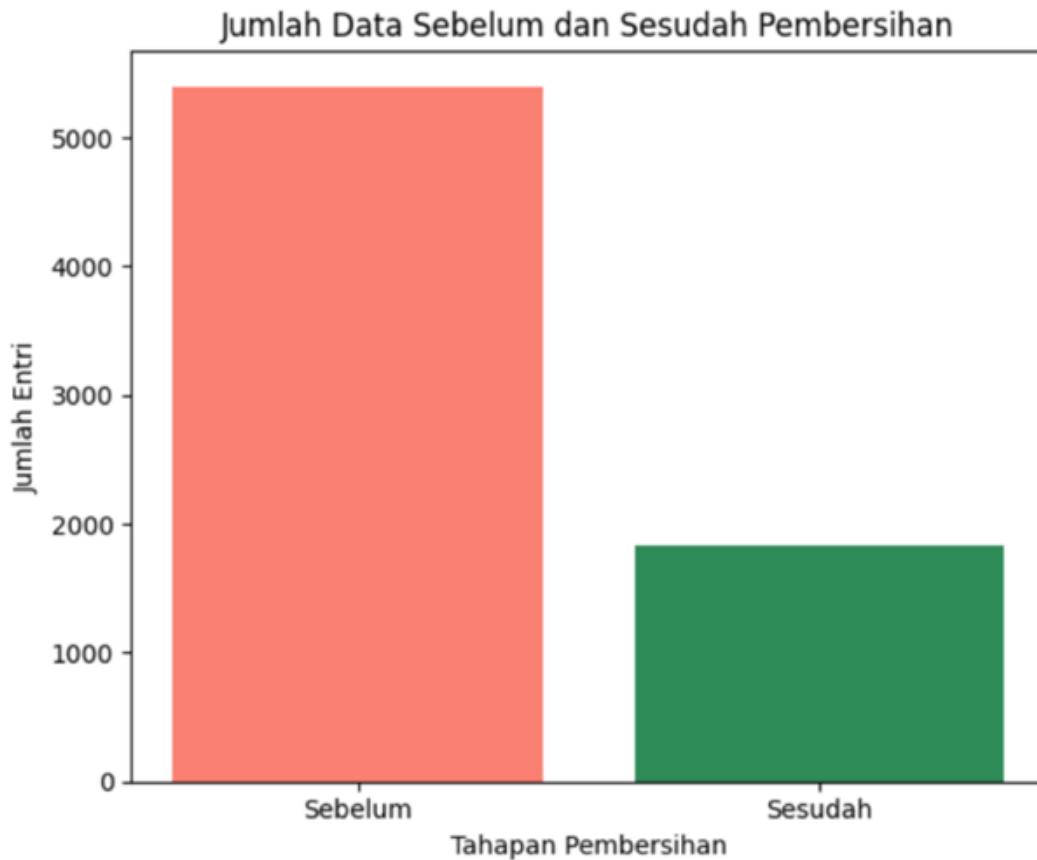
```
● ● ●

# Hapus nilai kosong di kolom penting
df = df.dropna(subset=['Category', 'Product Name', 'Customer Rating', 'Number Sold', 'Price', 'Total Review'])

# Ubah kolom numerik ke float
df['Price'] = pd.to_numeric(df['Price'], errors='coerce')
df['Customer Rating'] = pd.to_numeric(df['Customer Rating'], errors='coerce')
df['Number Sold'] = pd.to_numeric(df['Number Sold'], errors='coerce')
df['Total Review'] = pd.to_numeric(df['Total Review'], errors='coerce')
df = df.dropna(subset=['Price', 'Customer Rating', 'Number Sold', 'Total Review'])

# Hapus duplikat
df = df.drop_duplicates(subset=['Category', 'Product Name', 'Price', 'Customer Rating', 'Number Sold', 'Total Review'])
```

**Gambar 11.** Kode Python Pembersihan Data Kosong dan Duplikat



**Gambar 12.** Jumlah Data Sebelum dan Sesudah *Cleaning*

### Seleksi Kategori dengan Jumlah Data Memadai

Untuk menghindari bias pada kelompok dengan representasi terlalu kecil, hanya kategori utama dengan jumlah produk yang memadai ( $\geq 20$  entri) yang disertakan dalam analisis. Kategori dengan jumlah data terlalu sedikit dieliminasi dari *dataset* final, dengan proses seleksi ini dilakukan melalui kode pada Gambar 13 dan hasil distribusi kategori utama setelah penyaringan ditampilkan pada Gambar 14.

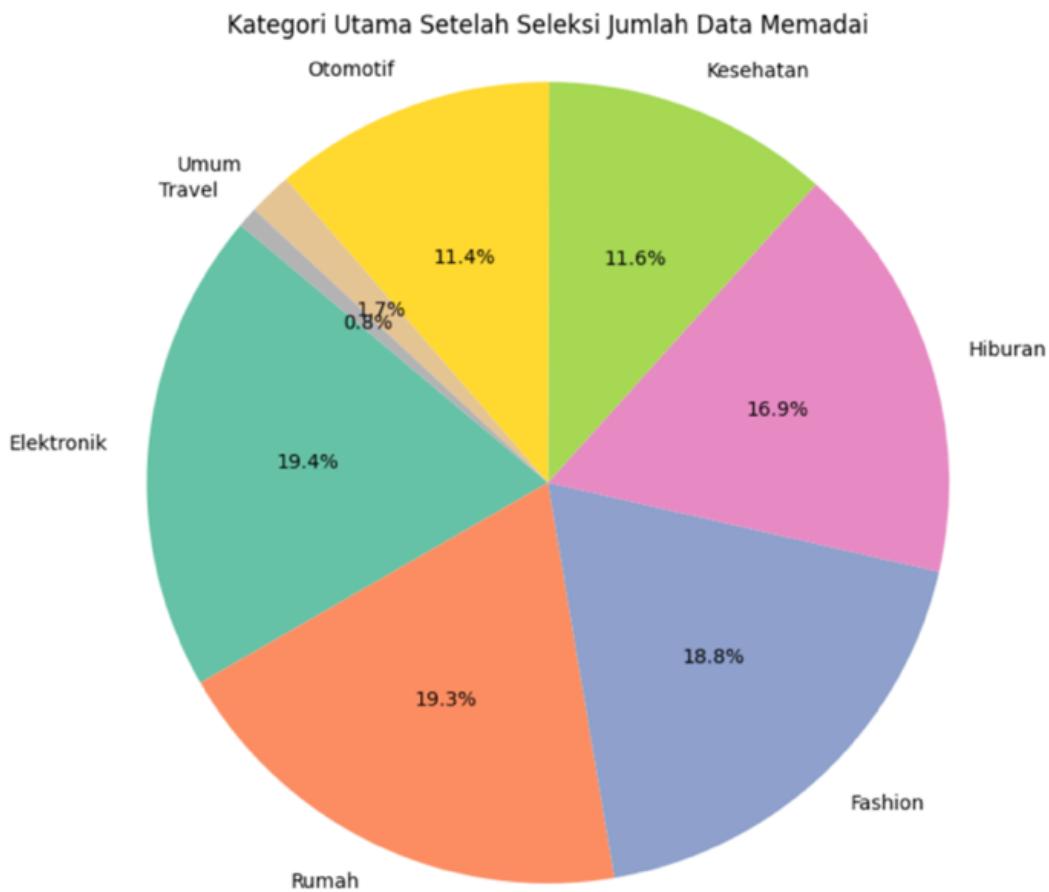
```

● ● ●

# Filter kategori yang jumlah datanya minimal 20
kategori_ok = df['Category'].value_counts()
[df['Category']
.value_counts() >= 20].index
df = df[df['Category'].isin(kategori_ok)]

```

**Gambar 13.** Kode Python Seleksi Kategori



**Gambar 14.** Seleksi Jumlah Data Memadai

### Penanganan Outlier

Atribut numerik seperti *Price*, *Number Sold*, dan *Total Review* dianalisis menggunakan metode *Interquartile Range (IQR)*. Data yang berada di luar rentang  $Q1 - 1.5 \times IQR$  dan  $Q3 + 1.5 \times IQR$  dihapus dari *dataset* karena dianggap sebagai *outlier* yang dapat mengganggu pembentukan *centroid* dan *medoid* pada proses klasterisasi. Proses deteksi dan penghapusan *outlier* diterapkan menggunakan metode *IQR* sebagaimana ditunjukkan pada Gambar 15, sedangkan dampak pembersihan pada ketiga atribut tersebut terlihat pada *boxplot* Gambar 16, Gambar 17, dan Gambar 18 yang menunjukkan berkurangnya nilai ekstrem setelah pembersihan. Sementara itu, atribut *Customer Rating* tidak disertakan karena bersifat ordinal tetap (skala 1–5) dan distribusinya tetap stabil, sebagaimana terlihat pada Gambar 19.

```

● ● ●

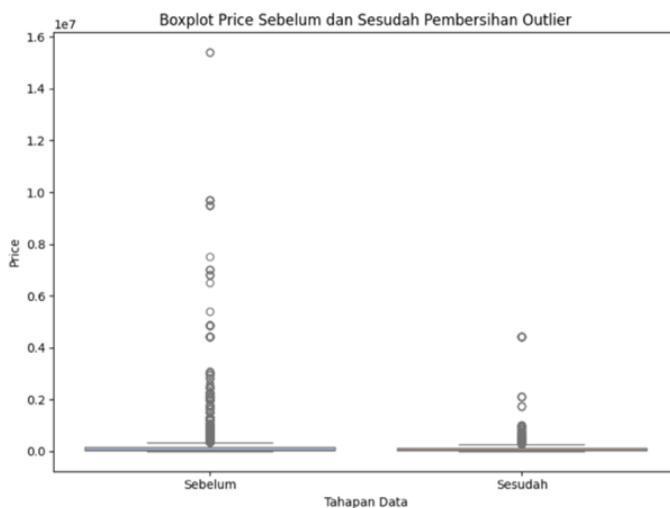
# Bersihkan outlier per kategori dengan metode IQR
cleaned_dfs = []
for cat in df['Category'].unique():
    subset = df[df['Category'] == cat].copy()
    for col in ['Customer Rating', 'Number Sold', 'Price', 'Total
Review']:
        Q1 = subset[col].quantile(0.25)
        Q3 = subset[col].quantile(0.75)
        IQR = Q3 - Q1
        lower = Q1 - 1.5 * IQR
        upper = Q3 + 1.5 * IQR
        subset = subset[(subset[col] >= lower) & (subset[col] <= upper)]
    cleaned_dfs.append(subset)

df_cleaned = pd.concat(cleaned_dfs).reset_index(drop=True)

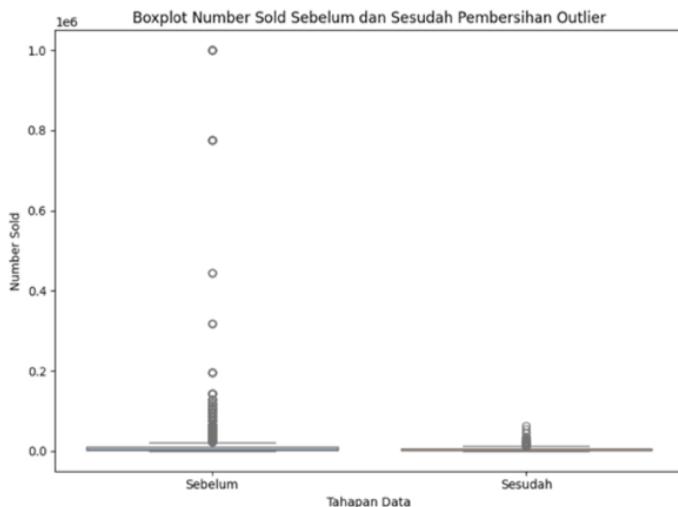
# Visualisasi distribusi Customer Rating setelah cleaning
plt.figure(figsize=(6, 4))
sns.countplot(x='Customer Rating', data=df_cleaned, palette='pastel')
plt.title("Distribusi Customer Rating Setelah Pembersihan Outlier")
plt.xlabel("Customer Rating")
plt.ylabel("Jumlah Produk")
plt.tight_layout()
plt.show()

```

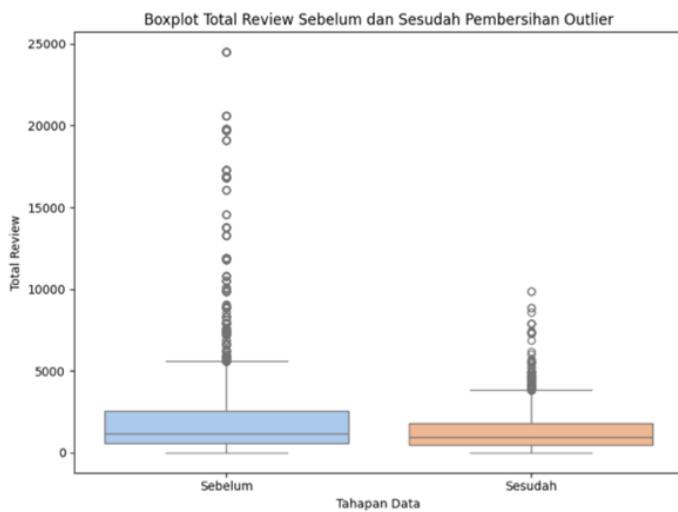
**Gambar 15.** Kode Python Penanganan Outlier



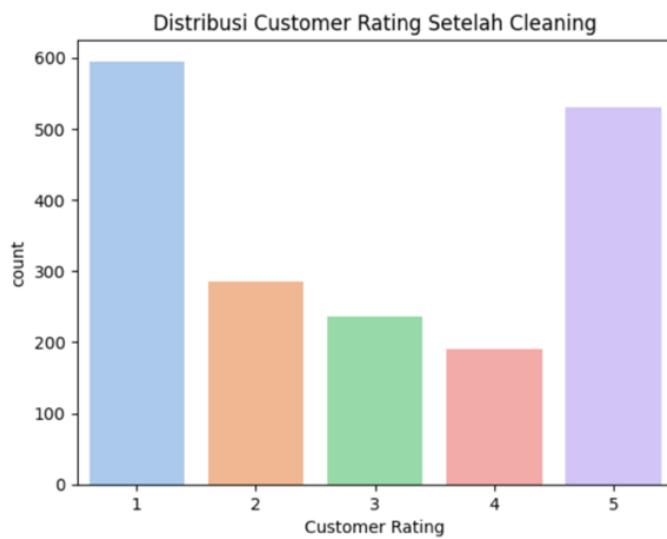
**Gambar 16.** Boxplot Harga (*Price*)



**Gambar 17.** Boxplot Number Sold



**Gambar 18.** Boxplot Total Review



**Gambar 19.** Countplot Distribusi Customer Rating

## Penomoran dan Seleksi Atribut

Setelah proses pembersihan data selesai, dilakukan tahap akhir penyusunan struktur *dataset* untuk memastikan bahwa hanya kolom yang dibutuhkan saja yang akan digunakan dalam analisis. Beberapa atribut dipilih secara khusus, seperti *No*, *Category*, *Main Category*, *Product Name*, *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*, yang dianggap paling representatif untuk menggambarkan performa dan jenis produk secara menyeluruh. Nilai pada kolom numerik kemudian dibulatkan ke angka bulat dengan fungsi `.round(0).astype(int)` agar lebih mudah dianalisis dan divisualisasikan tanpa mengurangi makna data, sebagaimana ditunjukkan pada Gambar 20 yang menampilkan potongan kode *Python* untuk penomoran ulang serta seleksi atribut.

```
● ● ●

# Buat ulang nomor urut berdasarkan data hasil cleaning
df_cleaned['No'] = range(1, len(df_cleaned) + 1)

# Pilih kolom akhir yang akan digunakan
df_final = df_cleaned[['No', 'Category', 'Main Category', 'Product Name', 'Price', 'Customer Rating', 'Number Sold', 'Total Review']].copy()

# Bulatkan angka dan ubah ke integer
for col in ['Price', 'Customer Rating', 'Number Sold', 'Total Review']:
    df_final[col] = df_final[col].round(0).astype(int)

# Tampilkan 10 entri acak setelah cleaning
print("Data setelah cleaning (10 entri acak):")
display(HTML(df_final.sample(10, random_state=42).to_html(index=False)))
```

**Gambar 20.** Kode *Python* Penomoran dan Seleksi Atribut

Data setelah cleaning (10 entri acak):							
No	Category	Main Category	Product Name	Price	Customer Rating	Number Sold	Total Review
1507	mother and baby	Kesehatan	kodomo baby tisu basah anti bacterial 50 sheets	37400	2	5309	1636
1434	phones and tablets	Elektronik	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	2114000	1	17400	4629
1484	mother and baby	Kesehatan	transpulmin baby balsam - 20gr	71390	4	4098	1925
1716	health	Kesehatan	tim ayam obat herbal komplit 12 macam	30000	3	7474	1311
998	muslim fashion	Fashion	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capucino	28999	3	11700	2819
1190	men's fashion	Fashion	baju seragam pgri kemeja hem katun pria batik pgri dinas pns panjang - s	119900	3	525	259
576	sport	Otomotif	basic headband black / bando hitam pria wanita / aksesoris olahraga	4900	2	2806	491
332	animal care	Rumah	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	79000	1	8243	3964
619	books	Hiburan	i saw the same dream again	83000	5	234	152
110	toys and hobbies	Hiburan	tenda bermain anak model castle (biru/ pink)	120000	1	1592	1085

**Gambar 21.** Data Setelah *Cleaning*

Setelah semua tahap dilakukan, jumlah data yang semula berisi 5.401 entri berkurang menjadi 1.838 entri, namun jumlah tersebut masih cukup memadai untuk tahap selanjutnya. *Output* dari proses ini berupa data yang sudah tersusun rapi dan siap untuk transformasi lebih lanjut, sebagaimana ditunjukkan pada Gambar 21 yang memperlihatkan sebagian data setelah dilakukan proses *cleaning*. File kemudian disimpan dengan nama ‘tokopedia\_cleaned\_final.xlsx’ dan diletakkan dalam direktori utama penelitian pada perangkat lokal penulis. Struktur ini digunakan sebagai *input* pada tahapan berikutnya yaitu normalisasi.

### 4.3 Transformasi Data

Setelah data bersih diperoleh, langkah berikutnya adalah melakukan transformasi terhadap atribut numerik utama agar memiliki skala yang seragam. Atribut yang digunakan adalah: *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*. Normalisasi dilakukan dengan metode *Min-Max Scaling* menggunakan pustaka *sklearn.preprocessing.MinMaxScaler*, yang mengubah nilai numerik ke rentang 0–1. Sebelum transformasi, kolom penomoran “No” yang lama dihapus dan digantikan dengan penomoran baru untuk menjaga urutan data. Setelah proses normalisasi selesai, hasilnya digabungkan kembali dengan kolom deskriptif (*Main Category*, *Category*, *Product Name*) dan disimpan dalam file ‘tokopedia\_normalized\_full.xlsx’ yang ditempatkan di direktori Dokumen laptop pribadi penulis.

#### Menghapus Kolom “No” Lama dan Menambahkan Ulang

Langkah awal dalam transformasi adalah menghapus kolom *No* lama yang sebelumnya digunakan untuk penomoran sementara selama proses pembersihan. Setelah dihapus, kolom *No* ditambahkan kembali secara otomatis dari baris pertama hingga terakhir, sebagaimana ditunjukkan pada Gambar 22 yang memperlihatkan potongan kode *Python* untuk menghapus kolom lama dan menambahkan kolom *No* baru. Hasil penerapan kode tersebut kemudian dapat dilihat pada Tabel 15, yang menampilkan contoh 5 entri awal setelah kolom *No* diperbarui dan disusun kembali secara berurutan.



```
# Hapus kolom 'No' lama jika ada
if 'No' in df_final.columns:
    df_final = df_final.drop(columns='No')

# Tambahkan ulang kolom No sesuai jumlah data
df_final.insert(0, 'No', range(1, len(df_final) + 1))
```

**Gambar 22.** Kode *Python* Menghapus dan Menambah Kolom “No”

**Tabel 15.** Contoh 5 Entri Awal

No	Category	Main Category	Product Name	Price	Customer Rating	Number Sold	Total Review
1	computers and laptops	Elektronik	wireless keyboard i8 mini touchpad...	53500	5	5449	2369
2	computers and laptops	Elektronik	paket lisensi windows 10 pro...	72000	5	2359	1044
3	computers and laptops	Elektronik	ssd midasforce 128 gb	213000	5	12300	3573
4	computers and laptops	Elektronik	adaptor charger monitor lcd led tv...	55000	5	2030	672
5	computers and laptops	Elektronik	adaptor charger monitor lcd led tv...	55000	5	2030	672

## Menentukan Atribut Numerik untuk Dinormalisasi

Dari dataset yang telah dibersihkan, dipilih empat atribut numerik yang akan dinormalisasi, yaitu *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*. Keempat kolom ini merupakan fitur kuantitatif yang paling mencerminkan performa dari masing-masing produk, dan digunakan sebagai dasar utama dalam proses klasterisasi. Proses pemilihan atribut numerik tersebut dilakukan menggunakan kode *Python* sebagaimana ditunjukkan pada Gambar 23, sedangkan Tabel 16 menampilkan nilai minimum dan maksimum dari setiap atribut sebelum dilakukan normalisasi.



```
# Ambil kolom numerik
fitur_numerik = df_final[['Price', 'Customer Rating', 'Number Sold', 'Total Review']]
```

**Gambar 23.** Kode *Python* Menentukan Atribut Numerik untuk Dinormalisasi

**Tabel 16.** Nilai Minimum dan Maksimum Sebelum Normalisasi

	<b>Price</b>	<b>Customer Rating</b>	<b>Number Sold</b>	<b>Total Review</b>
Min	176	1	14	6
Max	4439000	5	62200	9855

## Menerapkan Min-Max Scaling

Proses transformasi data dilakukan menggunakan teknik *Min-Max Scaling* melalui pustaka *sklearn*. Setiap atribut numerik seperti *Price*, *Customer Rating*, *Number Sold*, dan *Total Review* dikonversi ke dalam rentang 0–1 berdasarkan nilai minimum dan maksimum tiap atribut. Proses ini dilakukan secara otomatis menggunakan *MinMaxScaler*, kemudian hasil normalisasi dikombinasikan kembali dengan kolom referensi seperti *No*, *Category*, *Main Category*, dan *Product Name*. Kode penerapan normalisasi ditunjukkan pada Gambar 24, sementara Tabel 17 menampilkan contoh data sebelum proses normalisasi.



```
# Terapkan Min-Max Scaling
scaler = MinMaxScaler()
fitur_scaled = scaler.fit_transform(fitur_numerik)

# Buat DataFrame hasil normalisasi (presisi penuh)
df_normalized = pd.DataFrame(fitur_scaled, columns=fitur_numerik.columns)

# Gabungkan dengan kolom referensi, termasuk 'Main Category'
df_normalized_final = pd.concat([
    df_final[['No', 'Category', 'Main Category', 'Product Name']].reset_index(drop=True),
    df_normalized
], axis=1)

# Tampilkan 10 entri acak
print("Data setelah Min-Max Scaling (tanpa pembulatan):")
display(HTML(df_normalized_final.sample(10, random_state=42).to_html(index=False)))
```

**Gambar 24.** Kode *Python* Menerapkan *Min-Max Scaling*

**Tabel 17.** 1 Data Sebelum Normalisasi

No	Category	Main Category	Product Name	Price	Customer Rating	Number Sold	Total Review
1	computers and laptops	Elektronik	wireless keyboard i8 mini touchpad...	53500	5	5449	2369

Hasil penerapan normalisasi terhadap data tersebut dapat dilihat pada Tabel 18, yang menunjukkan data setelah dinormalisasi.

**Tabel 18.** 1 Data Setelah Normalisasi

No	Category	Main Category	Product Name	Price	Customer Rating	Number Sold	Total Review
1	computers and laptops	Elektronik	wireless keyboard i8 mini touchpad mouse 2.4g handheld pc android tv	0,0120 13092	1	0,0873 99093	0,2399 22835
<hr/>							
No	Category	Main Category	Product Name	Price	Customer Rating	Number Sold	Total Review
1507	mother and baby	Kesehatan	kodomo baby tisu basah anti bacterial 50 sheets	0.008386	0.25	0.085148	0.165499
1434	phones and tablets	Elektronik	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	0.476213	0.00	0.279581	0.469388
1484	mother and baby	Kesehatan	transpulmin baby balsam - 20gr	0.016043	0.75	0.065674	0.194842
1716	health	Kesehatan	tim ayam obat herbal komplit 12 macam	0.006719	0.50	0.119963	0.132501
998	muslim fashion	Fashion	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capucino	0.006493	0.50	0.187920	0.285613
1190	men's fashion	Fashion	baju seragam pgri kemeja hem katun pria batik pgri dinas pns panjang - s	0.026972	0.50	0.008217	0.025688
576	sport	Otomotif	basic headband black / bando hitam pria wanita / aksesoris olahraga	0.001064	0.25	0.044898	0.049244
332	animal care	Rumah	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	0.017758	0.00	0.132329	0.401868
619	books	Hiburan	i saw the same dream again	0.018659	1.00	0.003538	0.014824
110	toys and hobbies	Hiburan	tenda bermain anak model castle (biru/ pink)	0.026995	0.00	0.025375	0.109554

**Gambar 25.** Hasil Dataset Akhir

Setelah seluruh nilai numerik berhasil dinormalisasi ke rentang 0–1, data memiliki skala seragam sehingga perhitungan jarak lebih mudah dan tidak ada atribut yang mendominasi hasil klasterisasi. Data hasil transformasi kemudian digabung dengan atribut referensi (*No*, *Category*, *Main Category*, dan *Product Name*) menggunakan metode *concat* pada pustaka *pandas* agar tiap entri tetap dapat dikenali. Hasil akhirnya disimpan sebagai ‘tokopedia\_normalized\_full.xlsx’ di direktori utama laptop pribadi penulis, menjadi data akhir tahap normalisasi dan *input* untuk implementasi algoritma klasterisasi selanjutnya, sebagaimana ditunjukkan pada Gambar 25.

#### 4.4 Pendekatan Klaster Optimal

Setelah data selesai ditransformasi dan dinyatakan siap, tahap berikutnya adalah menentukan jumlah klaster yang paling optimal sebelum algoritma klasterisasi dijalankan. Penelitian ini menggunakan pendekatan *Elbow Method* pada dua algoritma berbeda, yaitu *K-Means* dan *K-Medoids*, dengan menghitung nilai *inertia* (untuk *K-Means*) dan total *cost* (untuk *K-Medoids*) pada rentang K = 1 hingga K = 10. Seluruh proses perhitungan dilakukan menggunakan pustaka *sklearn.cluster.KMeans* untuk *K-Means* dan *pyclustering.cluster.kmedoids* untuk *K-Medoids*. Pemasangan pustaka *pyclustering* dilakukan terlebih dahulu di Google Colab menggunakan perintah *!pip install pyclustering*.

```

● ● ●

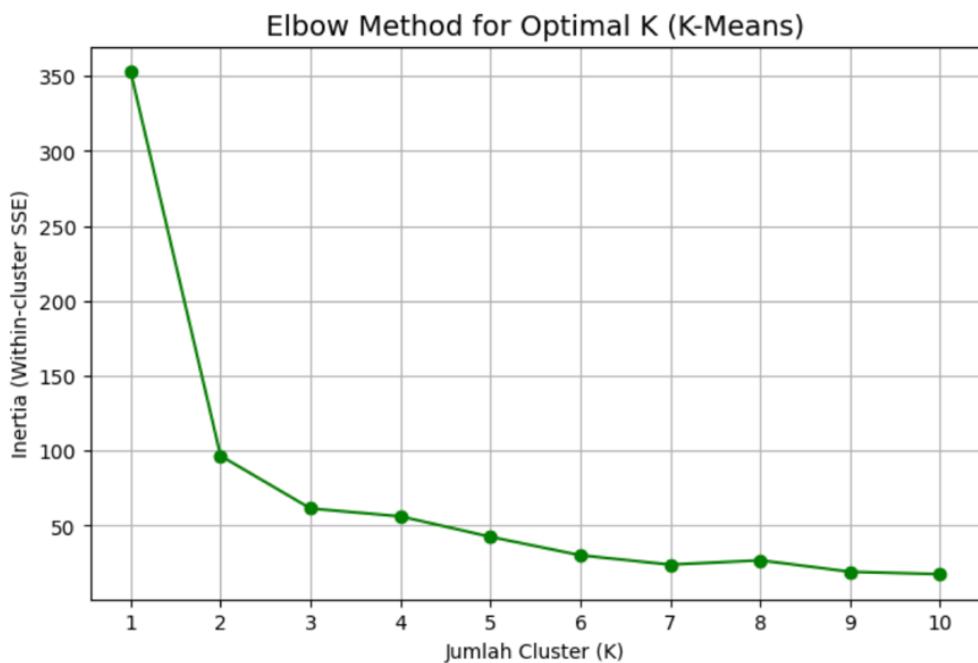
# Ambil fitur numerik untuk clustering
X = df[['Customer Rating', 'Number Sold', 'Total Review']]

# Hitung inertia untuk K = 1 hingga 10
inertia = []
K = range(1, 11)
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X)
    inertia.append(kmeans.inertia_)

# Plot Elbow Method
plt.figure(figsize=(8, 5))
plt.plot(K, inertia, marker='o', linestyle='-', color='green')
plt.title('Elbow Method for Optimal K (K-Means)', fontsize=14)
plt.xlabel('Jumlah Cluster (K)')
plt.ylabel('Inertia (Within-cluster SSE)')
plt.xticks(K)
plt.grid(True)
plt.show()

```

**Gambar 26.** Kode Python Elbow Method K-Means



**Gambar 27.** Elbow Method K-Means

```

● ● ●

# Hitung total cost K-Medoids untuk K = 1 sampai 10
K = range(1, 11)
total_cost = []

# Buat distance matrix terlebih dahulu
distance_matrix = calculate_distance_matrix(X)

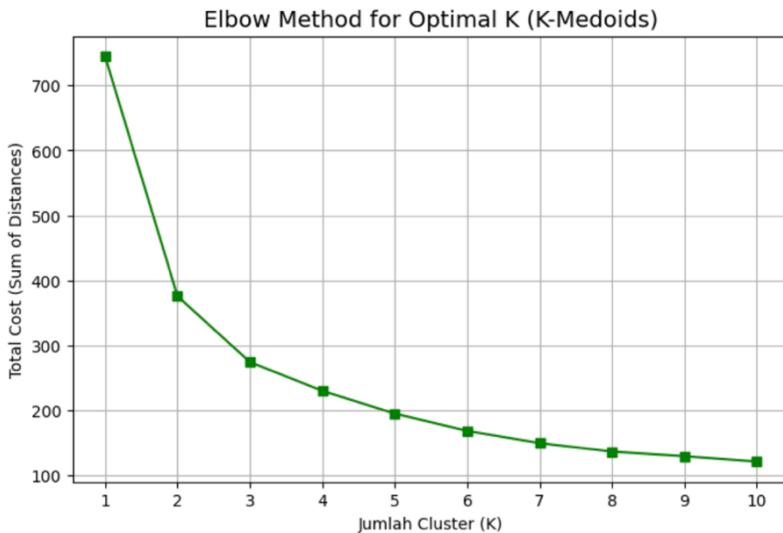
for k in K:
    # Pilih medoid awal secara acak
    initial_medoids = list(np.random.choice(len(X), k, replace=False))
    kmmedoids_instance = kmmedoids(distance_matrix, initial_medoids,
                                    data_type='distance_matrix')
    kmmedoids_instance.process()
    clusters = kmmedoids_instance.get_clusters()
    medoids = kmmedoids_instance.get_medoids()

    # Hitung total cost (jumlah jarak tiap titik ke medoid-nya)
    cost = 0
    for idx, cluster in enumerate(clusters):
        medoid_idx = medoids[idx]
        cost += np.sum([distance_matrix[i][medoid_idx] for i in cluster])
    total_cost.append(cost)

# Plot Elbow Method K-Medoids
plt.figure(figsize=(8, 5))
plt.plot(K, total_cost, marker='s', linestyle='-', color='green')
plt.title('Elbow Method for Optimal K (K-Medoids)', fontsize=14)
plt.xlabel('Jumlah Cluster (K)')
plt.ylabel('Total Cost (Sum of Distances)')
plt.xticks(K)
plt.grid(True)
plt.show()

```

**Gambar 28.** Kode Python Elbow Method K-Medoids



**Gambar 29.** Elbow Method K-Medoids

Hasil visualisasi grafik Elbow menunjukkan bahwa penurunan paling tajam terjadi pada  $K = 2$ , baik pada *inertia K-Means* (Gambar 27) maupun total *cost K-Medoids* (Gambar 29). Setelah  $K = 2$ , penurunan nilai metrik menjadi melandai, menandakan titik tekuk (elbow) berada pada angka tersebut. Oleh karena itu, jumlah klaster optimal yang digunakan dalam implementasi algoritma ditetapkan sebanyak dua klaster ( $K = 2$ ).

Adapun keseluruhan proses perhitungan ditunjukkan secara berurutan pada beberapa gambar, mulai dari potongan kode *Python* perhitungan *inertia K-Means* (Gambar 26), visualisasi grafik *Elbow Method K-Means* (Gambar 27), potongan kode *Python* perhitungan *total cost K-Medoids* (Gambar 28), serta visualisasi grafik *Elbow Method K-Medoids* (Gambar 29).

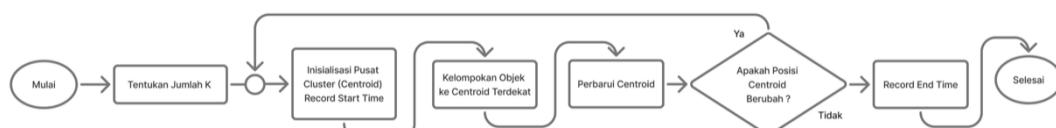
#### 4.5 Implementasi Algoritma

Setelah jumlah klaster optimal ditentukan dan data dinormalisasi, tahap berikutnya adalah implementasi klasterisasi menggunakan dua metode *K-Means* dan *K-Medoids*. *K-Means* mengelompokkan data berdasarkan jarak ke *centroid* hasil rata-rata klaster, sedangkan *K-Medoids* memakai titik representatif dari data aktual sebagai pusat. Proses dijalankan dengan skrip *Python* di Google Colab menggunakan pustaka *sklearn.cluster.KMeans* untuk *K-Means* dan *sklearn.metrics.pairwise\_distances* untuk perhitungan jarak pada *K-Medoids*, dengan *dataset* hasil transformasi yang telah disimpan secara lokal.

Masing-masing algoritma dijalankan secara iteratif hingga mencapai kondisi konvergen, yaitu ketika posisi *centroid* atau *medoid* tidak mengalami perubahan lagi. Hasil akhir dari proses *K-Means* kemudian disimpan dalam file ‘hasil\_kmeans\_k2\_final.xlsx’, sedangkan hasil *K-Medoids* disimpan dalam file ‘hasil\_kmedoids\_k2\_final.xlsx’. Kedua file tersebut mencakup informasi lengkap berupa label klaster, atribut numerik hasil transformasi, serta metadata produk yang tetap dipertahankan untuk keperluan interpretasi selanjutnya.

##### Implementasi *K-Means*

Implementasi metode *K-Means* dilakukan dengan jumlah klaster optimal  $K = 2$  menggunakan empat atribut numerik hasil normalisasi *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*. Tahapan proses ditunjukkan pada *flowchart* pada Gambar 30, sedangkan penerapan kode *Python* untuk menghasilkan klasterisasi ditampilkan pada Gambar 31. Data diacak ulang dengan *random\_state=42*, lalu dua *centroid* awal dipilih secara acak menggunakan *np.random.seed(42)*. Setiap iterasi menghitung jarak *Euclidean*, mengelompokkan data ke *centroid* terdekat, dan memperbarui posisi *centroid* berdasarkan rata-rata klaster hingga konvergen.



Gambar 30. Flowchart *K-Means*

```

K = 2

# Baca file dan acak datanya satu kali
file_path = "tokopedia_normalized_full.xlsx"
df = pd.read_excel(file_path)
df = df.sample(frac=1, random_state=42).reset_index(drop=True)
df['No'] = np.arange(1, len(df) + 1)

# Ambil fitur numerik
X = df[['Price', 'Customer Rating', 'Number Sold', 'Total Review']].values

# Inisialisasi centroid awal secara acak (tetap)
np.random.seed(42)
centroids = X[np.random.choice(len(X), K, replace=False)]

iteration = 0

# Mulai waktu komputasi
start_time = time.time()

while True:
    print(f"\nIterasi {iteration + 1}")

    # Hitung jarak ke setiap centroid
    distances = np.zeros((X.shape[0], K))
    for i in range(K):
        distances[:, i] = np.linalg.norm(X - centroids[i], axis=1)

    labels = np.argmin(distances, axis=1)

    # Buat DataFrame iterasi
    df_iter = df[['No', 'Product Name', 'Category', 'Main Category', 'Price', 'Customer Rating', 'Number Sold', 'Total Review']].copy()
    for i in range(K):
        df_iter[f'Ke C{i+1}'] = distances[:, i]
    df_iter['Cluster'] = ['C' + str(i+1) for i in labels]

    # Tampilkan 10 data pertama
    print("Tabel Hasil Iterasi (10 Data Awal):")
    display(HTML(df_iter.head(10).to_html(index=False)))

    # Tampilkan centroid saat ini
    print("Centroid pada Iterasi Ini:")
    for i in range(K):
        print(f"C{i+1}: {centroids[i]}")

    # Update centroid
    new_centroids = np.zeros_like(centroids)
    for i in range(K):
        cluster_points = X[labels == i]
        if len(cluster_points) > 0:
            new_centroids[i] = cluster_points.mean(axis=0)
        else:
            new_centroids[i] = centroids[i]

    # Cek konvergensi
    if np.allclose(centroids, new_centroids):
        print("\nKonvergen pada iterasi ke-{iteration + 1}")
        print("\nTabel Final Setelah Konvergen (iterasi ke-{iteration + 1})")
        display(HTML(df_iter.to_html(index=False)))

        print("Centroid Final:")
        for i in range(K):
            print(f"C{i+1}: {centroids[i]}")

    # Akhiri timer dan tampilkan waktu komputasi
    end_time = time.time()
    print(f"\nWaktu Komputasi K-Means: {end_time - start_time:.4f} detik")

    # Simpan hasil akhir ke Excel
    filename = f"hasil_kmeans_k{K}_final.xlsx"
    df_iter.to_excel(filename, index=False)
    files.download(filename)

    # Simpan label akhir berdasarkan centroid konvergen
    labels_kmeans = np.argmin(np.linalg.norm(X[:, np.newaxis] - centroids, axis=2), axis=1)
    break

    centroids = new_centroids
    iteration += 1

```

Gambar 31. Kode Python Implementasi K-Means

Pada iterasi pertama, seluruh data dihitung jaraknya terhadap kedua *centroid* menggunakan rumus *Euclidean* dan dikelompokkan ke dalam dua klaster, C1 dan C2. Setelah pengelompokan, posisi *centroid* diperbarui berdasarkan rata-rata nilai atribut dari anggota klaster, sebagaimana ditunjukkan pada Gambar 32.

Iterasi 1 Tabel Hasil Iterasi (10 Data Awal):												
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster		
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.751153	0.769003	C1		
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	1.160744	1.226759	C1		
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.256702	0.314110	C1		
4	tim ayan obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.503691	0.525810	C1		
5	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capucino	muslim fashion	Fashion	0.006493	0.50	0.187920	0.285613	0.535374	0.596644	C1		
6	baju seragam pgri kemeja hem katun pria batik pgri dinas panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.516380	0.500719	C2		
7	basic headband black / bando hitam pria wanita / aksesoris olahraga	sport	Otomotif	0.001064	0.25	0.044898	0.049244	0.756620	0.751110	C2		
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	1.035903	1.079832	C1		
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	0.141584	0.016615	C2		
10	tenda bermain anak model castle (biru/ pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	1.001379	1.004993	C1		

Centroid pada Iterasi Ini:  
C1: [0.03600593 1. 0.06715338 0.14011575]  
C2: [0.00286653 1. 0.0086193 0.01391084]

**Gambar 32.** Iterasi Pertama K-Means

Nilai *centroid* hasil iterasi pertama ini selanjutnya digunakan sebagai acuan dalam proses iterasi berikutnya, hingga posisi *centroid* tidak mengalami perubahan signifikan dan mencapai kondisi konvergen. Nilai *centroid* pada akhir iterasi ini digunakan untuk langkah perhitungan selanjutnya.

- a. C1: [0.03600593 1. 0.06715338 0.14011575]
- b. C2: [0.00286653 1. 0.0086193 0.01391004]

Pada iterasi kedua, Klasterisasi berlanjut menggunakan *centroid* terbaru dari iterasi pertama. Karena perubahan persebaran data, nilai *centroid* bergeser mengikuti komposisi klaster yang terbentuk, sebagaimana terlihat pada Gambar 33. Beberapa data berpindah klaster karena jaraknya kini lebih dekat ke pusat klaster yang berbeda.

Iterasi 2 Tabel Hasil Iterasi (10 Data Awal):												
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster		
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.194902	0.302244	C1		
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	0.699806	0.851628	C1		
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.314193	0.282467	C2		
4	tim ayan obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.093961	0.136735	C1		
5	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capucino	muslim fashion	Fashion	0.006493	0.50	0.187920	0.285613	0.141075	0.297097	C1		
6	baju seragam pgri kemeja hem katun pria batik pgri dinas panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.204404	0.027099	C2		
7	basic headband black / bando hitam pria wanita / aksesoris olahraga	sport	Otomotif	0.001064	0.25	0.044898	0.049244	0.249003	0.269430	C1		
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	0.486869	0.640701	C1		
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	0.597997	0.483544	C2		
10	tenda bermain anak model castle (biru/ pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	0.454559	0.522089	C1		

Centroid pada Iterasi Ini:  
C1: [0.03925735 0.43917112 0.10521998 0.1945681 ]  
C2: [0.02121901 0.51748252 0.02103612 0.04088108]

**Gambar 33.** Iterasi Kedua K-Means

Sebagai contoh, produk “transpT Tulmin baby balsam” yang sebelumnya termasuk dalam klaster C1, kini berpindah ke klaster C2. Dan produk lainnya tetap berada di klaster asal, mencerminkan kedekatan yang masih relevan dengan pusat klaster sebelumnya.

- a. C1: [0.03925735 0.43917112 0.10521998 0.1945681 ]
- b. C2: [0.02121901 0.51748252 0.02103612 0.04088108]

Pada iterasi ketiga, proses klasterisasi dilanjutkan dengan menggunakan *centroid* hasil perhitungan dari iterasi sebelumnya. Kedua *centroid*, C1 dan C2, kembali mengalami penyesuaian posisi berdasarkan rata-rata nilai atribut dari data dalam masing-masing klaster. Langkah ini terus dilakukan untuk meminimalkan variasi internal dalam klaster. Hasil klasterisasi pada tahap ini ditampilkan pada Gambar 34.

Iterasi 3 Tabel Hasil Iterasi (10 Data Awal):												
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster		
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.098764	0.582008	C1		
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	0.589091	1.034258	C1		
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.597284	0.134288	C2		
4	tim ayam obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.350312	0.337519	C2		
5	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capuccino	muslim fashion	Fashion	0.006493	0.50	0.187920	0.285613	0.374922	0.407102	C1		
6	baju seragam pgri kemeja hem katun pria batik pgri dinas prns panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.387677	0.332908	C2		
7	basic headband black / bando hitam pria wanita / aksesoris olahraga	sport	Otomotif	0.001064	0.25	0.044898	0.049244	0.169192	0.577211	C1		
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	0.275246	0.886884	C1		
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	0.866463	0.195965	C2		
10	tenda bermain anak model castle (biru/ pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	0.181940	0.825240	C1		

Centroid pada Iterasi Ini:  
C1: [0.02470986 0.15380658 0.09561745 0.17668471]  
C2: [0.04069406 0.82456647 0.04642483 0.08762757]

**Gambar 34.** Iterasi Ketiga *K-Means*

Terjadi beberapa perpindahan data, seperti produk “tim ayam obat herbal komplit” dan “baju seragam PGRI” yang sebelumnya berada di klaster C1 kini berpindah ke klaster C2. Sementara itu, sebagian besar produk lainnya tetap berada di klaster asalnya.

- C1: [0.02470986 0.15380658 0.09561745 0.17668471]
- C2: [0.04069406 0.82456647 0.04642483 0.08762757]

Pada iterasi keempat, distribusi data tidak mengalami banyak perubahan dibandingkan iterasi sebelumnya. Kedua *centroid* mengalami sedikit penyesuaian posisi karena adanya komposisi ulang klaster. Proses ini mengindikasikan bahwa kondisi klaster mulai stabil. Rincian perubahannya terlihat pada Gambar 35.

Iterasi 4 Tabel Hasil Iterasi (10 Data Awal):												
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster		
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.150169	0.596611	C1		
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	0.591593	1.037383	C1		
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.650551	0.126603	C2		
4	tim ayam obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.401677	0.350353	C2		
5	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capuccino	muslim fashion	Fashion	0.006493	0.50	0.187920	0.285613	0.432073	0.406436	C2		
6	baju seragam pgri kemeja hem katun pria batik pgri dinas prns panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.427094	0.357571	C2		
7	basic headband black / bando hitam pria wanita / aksesoris olahraga	sport	Otomotif	0.001064	0.25	0.044898	0.049244	0.189617	0.597664	C1		
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	0.269299	0.894386	C1		
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	0.913905	0.194496	C2		
10	tenda bermain anak model castle (biru/ pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	0.126605	0.843619	C1		

Centroid pada Iterasi Ini:  
C1: [0.02415526 0.10091991 0.08547511 0.15671367]  
C2: [0.04041499 0.84282585 0.05927562 0.11252131]

**Gambar 35.** Iterasi Keempat *K-Means*

Beberapa perubahan klaster terjadi, seperti produk “hijab voal segiempat premium” yang sebelumnya berada di klaster C1 kini berpindah ke klaster C2. Sementara itu, sebagian besar produk lainnya tetap berada pada klaster sebelumnya, menunjukkan kecenderungan struktur klaster yang mulai menetap.

- C1: [0.02415526 0.10091991 0.08547511 0.15671367]
- C2: [0.04041499 0.84282585 0.05927562 0.11252131]

Pada iterasi kelima, proses klasterisasi kembali dilakukan dengan menggunakan *centroid* hasil perhitungan dari iterasi sebelumnya. Perubahan posisi *centroid* masih terjadi, meskipun pergeserannya semakin kecil dibandingkan iterasi sebelumnya. Hasil pengelompokan pada tahap ini ditampilkan pada Gambar 36.

Iterasi 5 Tabel Hasil Iterasi (10 Data Awal):												
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster		
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.169387	0.580875	C1		
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	0.592989	1.022010	C1		
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.669773	0.110915	C2		
4	tim ayam obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.420478	0.334625	C2		
5	hijab voal segiempat premium - emikawa jilbab kerudung terbaru korea - capucino	muslim fashion	Fashion	0.006493	0.50	0.187920	0.285613	0.451891	0.389566	C2		
6	baju seragam pgri kemeja hem katun pria batik pgri dinas pns panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.442850	0.345547	C2		
7	basic headband black / bando hitam pria wanita / aksesoris olahraga	sport	Otomotif	0.001064	0.25	0.044898	0.049244	0.201146	0.583615	C1		
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	0.268424	0.877597	C1		
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	0.931545	0.211274	C2		
10	tenda bermain anak model castle (biru/ pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	0.108421	0.828765	C1		

Centroid pada Iterasi Ini:  
C1: [0.02350404 0.0819161 0.08284176 0.15115063]  
C2: [0.04030134 0.82774869 0.06285991 0.11960265]

**Gambar 36.** Iterasi Kelima K-Means

Dari hasil pengelompokan, sebagian besar data tetap berada dalam klaster sebelumnya, yang menunjukkan kestabilan struktur klaster. Namun, terdapat sedikit penyesuaian posisi data pada beberapa titik, meskipun tidak sebanyak iterasi sebelumnya.

- C1: [0.02350404 0.0819161 0.08284176 0.15115063]
- C2: [0.04030134 0.82774869 0.06285991 0.11960265]

Pada iterasi keenam, proses klasterisasi dilakukan kembali dengan menggunakan *centroid* dari iterasi kelima. Hasil pengelompokan menunjukkan bahwa tidak terjadi perpindahan data antar klaster seluruh entri tetap berada di klaster yang sama seperti sebelumnya. Meskipun tidak ada perubahan afiliasi klaster, nilai *centroid* masih mengalami sedikit pergeseran, sebagaimana ditunjukkan pada Gambar 37.

Iterasi 6 Tabel Hasil Iterasi (10 Data Awal):												
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster		
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.170471	0.580055	C1		
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	0.593836	1.020864	C1		
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.670797	0.109471	C2		
4	tim ayam obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.421430	0.333774	C2		
5	hijab voal segiempat premium - emikawa jilbab kerudung terbaru korea - capucino	muslim fashion	Fashion	0.006493	0.50	0.187920	0.285613	0.453368	0.388167	C2		
6	baju seragam pgri kemeja hem katun pria batik pgri dinas pns panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.443222	0.345386	C2		
7	basic headband black / bando hitam pria wanita / aksesoris olahraga	sport	Otomotif	0.001064	0.25	0.044898	0.049244	0.201393	0.583122	C1		
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	0.269605	0.876450	C1		
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	0.932212	0.212711	C2		
10	tenda bermain anak model castle (biru/ pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	0.108767	0.828130	C1		

Centroid pada Iterasi Ini:  
C1: [0.02354793 0.08096591 0.08210225 0.14972725]  
C2: [0.04022589 0.82706374 0.06358168 0.12097744]

**Gambar 37.** Iterasi Keenam K-Means

Karena masih terdapat perubahan meskipun kecil pada posisi *centroid*, proses iterasi dilanjutkan satu kali lagi untuk memastikan apakah hasil klasterisasi benar-benar telah mencapai kondisi konvergen atau belum.

- C1: [0.02354793 0.08096591 0.08210225 0.14972725]
- C2: [0.04022589 0.82706374 0.06358168 0.12097744]

Setelah perhitungan ulang pada iterasi keenam, posisi *centroid* tidak berubah dibandingkan iterasi sebelumnya. Hal ini menunjukkan bahwa proses klasterisasi telah mencapai konvergen, di mana seluruh data telah stabil pada klaster terdekatnya, sebagaimana ditunjukkan pada Gambar 38.

Konvergen pada iterasi ke-6												
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster		
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165498	0.170471	0.580055	C1		
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	0.593836	1.020864	C1		
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.670797	0.109471	C2		
4	tim ayam obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.421430	0.333774	C2		
5	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capucino	muslim fashion	Fashion	0.006493	0.50	0.187920	0.285613	0.453368	0.388167	C2		
6	baju seragam pgr kemeja hem katun pria batik pgr dinas pramuka - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.443222	0.345386	C2		
7	basic headband black / bando hitam pria wanita / aksesoris olahraga	sport	Otomotif	0.001064	0.25	0.044898	0.049244	0.201393	0.583122	C1		
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	0.269605	0.876450	C1		
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	0.932212	0.212711	C2		
10	tenda bermain anak model castle (biru/pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	0.106767	0.828130	C1		

Centroid Final (Tidak berubah):  
C1: [0.02354793 0.08096591 0.08210225 0.14972725]  
C2: [0.04022589 0.82706374 0.06358168 0.12097744]

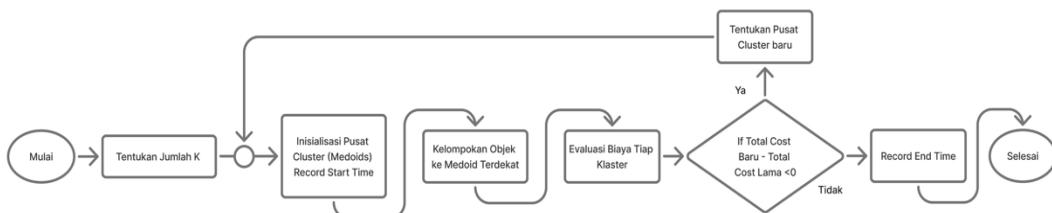
**Gambar 38.** Iterasi Konvergen *K-Means*

Dengan demikian, iterasi keenam menjadi hasil akhir dari proses *K-Means* pada data ini. Dua *centroid* akhir yang terbentuk menjadi representasi masing-masing klaster, yaitu:

- C1: [0.02354793 0.08096591 0.08210225 0.14972725]
- C2: [0.04022589 0.82706374 0.06358168 0.12097744]

### Implementasi *K-Medoids*

Implementasi *K-Medoids* dilakukan dengan jumlah klaster yang sama seperti *K-Means*, yaitu  $K = 2$ , baik karena nilai tersebut merupakan hasil optimal maupun untuk mempermudah perbandingan. Data diacak dengan *random\_state*=42, lalu *medoid* awal dipilih secara acak menggunakan *np.random.seed(42)*. Jarak setiap data ke *medoid* dihitung menggunakan *pairwise\_distances*, dan data dikelompokkan ke *medoid* terdekat. Pada tiap iterasi dihitung total *cost* sebagai jumlah jarak terdekat seluruh data, kemudian *medoid* baru dipilih dengan *cost* terkecil hingga posisi *medoid* stabil (konvergen). Alur proses ditunjukkan pada Gambar 39, sedangkan implementasi kode *Python* dapat dilihat pada Gambar 40.



**Gambar 39.** Flowchart *K-Medoids*

```

● ● ●

# Baca dan acak data
file_path = "tokopedia_normalized_full.xlsx"
df = pd.read_excel(file_path)
df = df.sample(frac=1, random_state=42).reset_index(drop=True)
df['No'] = np.arange(1, len(df) + 1)

# Ambil fitur numerik
X = df[['Price', 'Customer Rating', 'Number Sold', 'Total Review']].values

# Inisialisasi jumlah cluster dan medoid awal
K = 2
np.random.seed(42)
medoids = np.random.choice(len(X), K, replace=False)

iteration = 0

# Mulai waktu komputasi
start_time = time.time()

while True:
    print(f"\n Iterasi {iteration + 1}")

    # Hitung jarak dari semua data ke setiap medoid
    distance_matrix = pairwise_distances(X, X[medoids])
    labels = np.argmin(distance_matrix, axis=1)

    # Hitung total cost untuk iterasi ini
    total_cost = np.sum(np.min(distance_matrix, axis=1))

    # Buat DataFrame hasil iterasi
    df_iter = df[['No', 'Product Name', 'Category', 'Main Category',
                  'Price', 'Customer Rating', 'Number Sold', 'Total Review']].copy()
    for i in range(K):
        df_iter[f'Ke C{i+1}'] = distance_matrix[:, i]
    df_iter['Cluster'] = ['C' + str(i+1) for i in labels]

    # Tampilkan 10 data pertama hasil iterasi
    print("\n Tabel Hasil Iterasi (10 Data Awal):")
    display(HTML(df_iter.head(10).to_html(index=False)))

    # Tampilkan total cost setelah tabel
    print(f"\n Total Cost Iterasi {iteration + 1}: {total_cost:.6f}")

    # Tampilkan medoid saat ini
    print("Medoid pada Iterasi Ini:")
    for i in range(K):
        print(f"C{i+1} (Index {medoids[i]}): {X[medoids[i]]}")

    # Evaluasi medoid baru untuk tiap cluster
    new_medoids = []
    for i in range(K):
        cluster_indices = np.where(labels == i)[0]
        if len(cluster_indices) == 0:
            new_medoids.append(medoids[i])
            continue
        intra_distances = pairwise_distances(X[cluster_indices],
                                              X[cluster_indices])
        cost = intra_distances.sum(axis=1)
        best_index = cluster_indices[np.argmin(cost)]
        new_medoids.append(best_index)

    new_medoids = np.array(new_medoids)

    # Cek konvergensi
    if np.array_equal(medoids, new_medoids):
        print(f"\n Konvergen pada iterasi ke-{iteration + 1}")
        print("Medoid Final (Tidak berubah):")
        for i in range(K):
            print(f"C{i+1} (Index {medoids[i]}): {X[medoids[i]]}")

    # Akhiri waktu komputasi
    end_time = time.time()
    print(f"\n Waktu Komputasi K-Medoids: {end_time - start_time:.4f} detik")

    # Tampilkan tabel akhir
    print("\n Tabel Final Setelah Konvergen (10 Data Awal):")
    display(HTML(df_iter.head(10).to_html(index=False)))
    print(f"\n Total Cost Final: {total_cost:.6f}")

    # Simpan hasil akhir ke Excel
    filename = f"hasil_kmedoids_K{K}_final.xlsx"
    df_iter.to_excel(filename, index=False)
    files.download(filename)

    # Simpan label akhir berdasarkan medoid konvergen
    labels_kmedoids = np.argmin(pairwise_distances(X, X[medoids]), axis=1)
    break

# Perbarui medoid
medoids = new_medoids
iteration += 1

```

Gambar 40. Kode Python Implementasi K-Medoids

Pada iterasi pertama, jarak data dihitung dengan *Euclidean distance* dan dikelompokkan ke dalam dua klaster (C1 dan C2). Evaluasi dilakukan untuk mengganti *medoid* dengan titik yang paling mendekati rata-rata jarak anggotanya. Proses ini bertujuan menurunkan total *cost* agar klaster lebih optimal. Hasil iterasi pertama ditunjukkan pada Gambar 41.

Iterasi 1												
Tabel Hasil Iterasi (10 Data Awal):												
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster		
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.751153	0.769003	C1		
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	1.160744	1.226759	C1		
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.256702	0.314110	C1		
4	tim ayam obat herbal komplit 12 macam		health	0.006719	0.50	0.119963	0.132501	0.503691	0.525810	C1		
5	hijab voal sejgiemt premium - emikoawa jilbab kerudung terbaru korea - capucino		muslim fashion	0.006493	0.50	0.187920	0.285613	0.535374	0.596644	C1		
6	baju seragam pgri kemeja hem katun pria batik pgri dinas prns panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.516380	0.500719	C2		
7	basic headband black / bando hitam pria wanita / aksesoris olahraga		sport	0.001064	0.25	0.044898	0.049244	0.756620	0.751710	C2		
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	1.035903	1.079832	C1		
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	0.141584	0.016615	C2		
10	tenda bermain anak model castle (biru/ pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	1.001379	1.004993	C1		

Total Cost Iterasi 1: 1052.206135

Medoid pada Iterasi 1:  
C1 (Index 1506): [0.03600593 1. 0.06715338 0.14011575]  
C2 (Index 1433): [0.00286653 1. 0.0086193 0.01391004]

**Gambar 41.** Iterasi Pertama *K-Medoids*

*Medoid* hasil pembaruan digunakan sebagai acuan untuk iterasi selanjutnya. Selama posisi *medoid* berubah atau total *cost* belum minimum, proses akan terus berlanjut. Hasil total *cost* yang diperoleh pada iterasi ini cukup tinggi karena posisi *medoid* masih belum optimal.

Total Cost Iterasi Pertama: 1052.206135

- a. C1 (Index 1506): [0.03600593 1. 0.06715338 0.14011575]
- b. C2 (Index 1433): [0.00286653 1. 0.0086193 0.01391004]

Pada iterasi kedua menghasilkan distribusi klaster yang lebih efisien dengan beberapa data berpindah klaster, seperti produk "baju seragam PGRI kemeja hem batik pria" dari C1 ke C2. Hasil iterasi kedua dapat dilihat pada Gambar 42.

Iterasi 2												
Tabel Hasil Iterasi (10 Data Awal):												
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster		
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.009261	0.288090	C1		
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	0.634530	0.841988	C1		
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.501224	0.297735	C2		
4	tim ayam obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.254684	0.139694	C2		
5	hijab voal sejgiemt premium - emikoawa jilbab kerudung terbaru korea - capucino		muslim fashion	0.006493	0.50	0.187920	0.285613	0.291263	0.299545	C1		
6	baju seragam pgri kemeja hem katun pria batik pgri dinas prns panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.301588	0.017745	C2		
7	basic headband black / bando hitam pria wanita / aksesoris olahraga		sport	0.001064	0.25	0.044898	0.049244	0.131307	0.252779	C1		
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	0.342404	0.627462	C1		
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	0.771268	0.500855	C2		
10	tenda bermain anak model castle (biru/ pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	0.266375	0.504754	C1		

Total Cost Iterasi 2: 583.776829

Medoid pada Iterasi 2:  
C1 (Index 406): [0.01140482 0.25 0.09143537 0.17159102]  
C2 (Index 1710): [0.02248884 0.5 0.01530891 0.04132399]

**Gambar 42.** Iterasi Kedua *K-Medoids*

Penurunan total *cost* cukup signifikan dibandingkan iterasi pertama, menandakan bahwa struktur klaster makin efisien. Oleh karena itu, *medoid* kembali diperbarui untuk menyempurnakan distribusi data. Proses tetap dilanjutkan hingga struktur benar-benar stabil.

Total Cost Iterasi Kedua: 583.776829

- a. C1 (Index 406): [0.01140482 0.25      0.09143537 0.17159102]
- b. C2 (Index 1710): [0.02248884 0.5      0.01530891 0.04132399]

Pada iterasi ketiga, klasterisasi mulai menunjukkan kestabilan, *medoid* diperbarui berdasarkan data yang representatif dari tiap klaster. Hanya satu data yang berpindah, yaitu “*tim ayam obat herbal komplit*” ke klaster C1, sebagaimana ditunjukkan pada Gambar 43.

Iterasi 3													
Tabel Hasil Iterasi (10 Data Awal):													
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster			
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.254914	0.754652	C1			
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	0.615008	1.184272	C1			
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.753767	0.271025	C2			
4	tim ayam obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.502988	0.507041	C1			
5	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capucino	muslim fashion	Fashion	0.006493	0.50	0.187920	0.285613	0.540473	0.554439	C1			
6	baju seragam pgri kerjeja hem katun pria batik pgri dinas pns panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.512313	0.505939	C2			
7	basic headband black / bando hitam pria wanita / aksesoris olahraga	sport	Otomotif	0.001064	0.25	0.044898	0.049244	0.261193	0.751510	C1			
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	0.289561	1.050357	C1			
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	1.007523	0.088939	C2			
10	tenda bermain anak model castle (biru/ pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	0.044698	1.000420	C1			

Total Cost Iterasi 3: 403.837041

Medoid pada Iterasi Ini:  
C1 (Index 1336): [0.01685672 0.      0.06773229 0.11960605]  
C2 (Index 117): [0.02203827 1.      0.04780819 0.0918875 ]

**Gambar 43.** Iterasi Ketiga *K-Medoids*

Perubahan kecil ini tetap berkontribusi dalam menurunkan total cost. Struktur klaster yang terbentuk pun mulai jelas, dengan mayoritas data tetap berada dalam klaster asalnya. Evaluasi tetap dilanjutkan untuk memastikan hasil yang optimal.

Total Cost Iterasi Ketiga: 403.837041

- a. C1 (Index 1336): [0.01685672 0.      0.06773229 0.11960605]
- b. C2 (Index 117): [0.02203827 1.      0.04780819 0.0918875 ]

Pada iterasi keempat, sebagian besar data tetap berada pada klaster yang sama seperti iterasi sebelumnya. Ini menunjukkan bahwa sistem mulai mencapai kestabilan. Satu-satunya perubahan hanya terjadi pada posisi *medoid* C2, yang diperbarui ke titik data lain dalam klaster yang sama. Hasil pembaruan ini dapat dilihat pada Gambar 44.

Iterasi 4													
Tabel Hasil Iterasi (10 Data Awal):													
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster			
1	kodomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.254914	0.755392	C1			
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	0.615008	1.188967	C1			
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.753767	0.273607	C2			
4	tim ayam obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.502988	0.507989	C1			
5	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capucino	muslim fashion	Fashion	0.006493	0.50	0.187920	0.285613	0.540473	0.557444	C1			
6	baju seragam pgri kerjeja hem katun pria batik pgri dinas pns panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.512313	0.504972	C2			
7	basic headband black / bando hitam pria wanita / aksesoris olahraga	sport	Otomotif	0.001064	0.25	0.044898	0.049244	0.261193	0.751057	C1			
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	0.289561	1.052474	C1			
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	1.007523	0.081722	C2			
10	tenda bermain anak model castle (biru/ pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	0.044698	1.000507	C1			

Total Cost Iterasi 4: 403.815028

Medoid pada Iterasi Ini:  
C1 (Index 1336): [0.01685672 0.      0.06773229 0.11960605]  
C2 (Index 1154): [0.01685672 1.      0.04401312 0.08579551]

**Gambar 44.** Iterasi Keempat *K-Medoids*

Meskipun tidak ada perpindahan data, proses tetap dilanjutkan karena total *cost* masih mengalami penurunan, meskipun sangat kecil. Evaluasi terus dilakukan untuk memastikan *medoid* yang digunakan sudah paling optimal.

Total Cost Iterasi Keempat: 403.815028

- a. C1 (Index 1336): [0.01685672 0. 0.06773229 0.11960605]
- b. C2 (Index 1154): [0.01685672 1. 0.04401312 0.08579551]

Setelah perhitungan ulang, posisi *medoid* tidak mengalami perubahan dibandingkan dengan iterasi sebelumnya. Hal ini menunjukkan bahwa proses klasterisasi telah mencapai konvergen, di mana seluruh data telah stabil pada klaster terdekatnya. Distribusi data konsisten dan struktur klaster tidak mengalami gangguan. Tidak ada lagi penurunan total *cost* yang signifikan, sebagaimana ditunjukkan pada Gambar 45.

Medoid Final (Tidak berubah):										
C1 (Index 1336):		[0.01685672 0. 0.06773229 0.11960605]								
C2 (Index 1154):		[0.01685672 1. 0.04401312 0.08579551]								
Tabel Final Setelah Konvergen (10 Data Awal):										
No	Product Name	Category	Main Category	Price	Customer Rating	Number Sold	Total Review	Ke C1	Ke C2	Cluster
1	kedomo baby tisu basah anti bacterial 50 sheets	mother and baby	Kesehatan	0.008386	0.25	0.085148	0.165499	0.254914	0.755392	C1
2	samsung galaxy a12 4/128 gb garansi resmi sein - hitam	phones and tablets	Elektronik	0.476213	0.00	0.279581	0.469388	0.615008	1.188967	C1
3	transpulmin baby balsam - 20gr	mother and baby	Kesehatan	0.016043	0.75	0.065674	0.194842	0.753767	0.273607	C2
4	tim ayam obat herbal komplit 12 macam	health	Kesehatan	0.006719	0.50	0.119963	0.132501	0.502988	0.507989	C1
5	hijab voal segiempat premium - emikoawa jilbab kerudung terbaru korea - capucino	muslim fashion	Fashion	0.006493	0.50	0.187920	0.285613	0.540473	0.557444	C1
6	baju seragam pgri kemeja hem katun pria batik pgri dinira pns panjang - s	men's fashion	Fashion	0.026972	0.50	0.008217	0.025688	0.512313	0.504972	C2
7	basic headband black / bando hitam pria wanita / aksesoris olahraga	sport	Otomotif	0.001064	0.25	0.044898	0.049244	0.261193	0.751057	C1
8	obat kutu hewan anjing & kucing bahan alami racoon / flea remover	animal care	Rumah	0.017758	0.00	0.132329	0.401868	0.289561	1.052474	C1
9	i saw the same dream again	books	Hiburan	0.018659	1.00	0.003538	0.014824	1.007523	0.081722	C2
10	tenda bermain anak model castle (biru/pink)	toys and hobbies	Hiburan	0.026995	0.00	0.025375	0.109554	0.044698	1.000507	C1

Total Cost Final: 403.815028

**Gambar 45.** Iterasi Konvergen *K-Medoids*

*Medoid* pada tahap akhir ini ditetapkan sebagai *medoid* final dari masing-masing klaster. Hasil ini menunjukkan representasi data yang efisien terhadap pusat klasternya. Proses dihentikan karena struktur sudah optimal dan tidak diperlukan iterasi tambahan.

Total Cost Final: 403.815028 Medoid Final (Tidak berubah):

- C1 (Index 1336): [0.01685672 0. 0.06773229 0.11960605]
- C2 (Index 1154): [0.01685672 1. 0.04401312 0.08579551]

#### 4.6 Evaluasi Clustering

Setelah proses klasterisasi selesai, langkah selanjutnya adalah mengevaluasi kualitas pengelompokan menggunakan metrik kuantitatif, yaitu *Davies-Bouldin Index (DBI)* dan *Silhouette Score*. Evaluasi dilakukan menggunakan pustaka *sklearn.metrics* dan diterapkan langsung pada label klaster hasil algoritma *K-Means* dan *K-Medoids*. Dataset yang digunakan berasal dari file hasil akhir, yakni ‘hasil\_kmeans\_k2\_final.xlsx’ dan ‘hasil\_kmedoids\_k2\_final.xlsx’. Proses evaluasi ini difokuskan pada nilai numerik tanpa visualisasi tambahan, guna menilai kepadatan dan pemisahan antar klaster yang terbentuk.

### **Davies-Bouldin Index (DBI) K-Means.**

Hasil evaluasi menggunakan *DBI* pada algoritma *K-Means* menunjukkan nilai sebesar 0.5717. Mengindikasikan bahwa klaster yang terbentuk memiliki pemisahan yang cukup baik antar satu sama lain. Semakin kecil nilai *DBI*, semakin baik performa klasterisasi karena menunjukkan bahwa klaster saling berjauhan dan kompak di dalamnya.

### **Silhouette Score K-Means.**

Sementara itu, hasil *Silhouette Score* untuk algoritma *K-Means* menunjukkan nilai sebesar 0.6012. Skor ini berada dalam kategori baik, karena mendekati nilai maksimum 1. Artinya, sebagian besar data berada cukup dekat dengan anggota dalam klaster yang sama dan jauh dari klaster lain. Ini menunjukkan bahwa struktur klaster cukup jelas dan data telah dikelompokkan dengan representatif.

Hasil evaluasi kuantitatif algoritma *K-Means* ditunjukkan pada Gambar 46, yang memuat potongan kode *Python* beserta perhitungan nilai *Davies-Bouldin Index* dan *Silhouette Score* sebagai dasar analisis kualitas klasterisasi.

```
● ● ●

# Hitung evaluasi
dbi_score = davies_bouldin_score(X, labels)
sil_score = silhouette_score(X, labels)

# Tampilkan hasil
print("Evaluasi Hasil K-Means:")
print(f"Davies-Bouldin Index : {dbi_score:.4f}")
print(f"Silhouette Score      : {sil_score:.4f}")
```

**Gambar 46.** Kode Python Evaluasi *K-Means*

### **Davies-Bouldin Index (DBI) K-Medoids.**

Hasil evaluasi menggunakan *DBI* pada algoritma *K-Medoids* menunjukkan nilai sebesar 0.5870. yang menunjukkan bahwa hasil pengelompokan yang terbentuk juga memiliki struktur yang layak. Nilai tersebut menggambarkan bahwa masing-masing klaster cukup terisolasi dari klaster lainnya dan memiliki penyebaran internal yang relatif merata. Nilai ini mencerminkan performa klasterisasi yang cukup stabil.

### **Silhouette Score K-Medoids.**

Sementara itu, *Silhouette Score* untuk *K-Medoids* adalah sebesar 0.5857. Nilai ini juga menunjukkan hasil klasterisasi yang memadai. Skor di atas 0.5 menandakan bahwa sebagian besar data berada dalam klaster yang sesuai dan memiliki jarak yang signifikan terhadap klaster lainnya. Dengan demikian, struktur pengelompokan yang dihasilkan tergolong baik dan dapat digunakan untuk analisis lanjutan.

Hasil evaluasi kuantitatif algoritma *K-Medoids* ditunjukkan pada Gambar 47, yang menampilkan potongan kode *Python* beserta perhitungan nilai *Davies-Bouldin Index* dan *Silhouette Score* sebagai dasar penilaian kualitas klasterisasi.

```
● ● ●

# Hitung evaluasi
dbi_score = davies_bouldin_score(X, labels)
sil_score = silhouette_score(X, labels)

print("\nEvaluasi Hasil K-Medoids:")
print(f"Davies-Bouldin Index : {dbi_score:.4f}")
print(f"Silhouette Score : {sil_score:.4f}")
```

**Gambar 47.** Kode *Python* Evaluasi *K-Medoids*

Ringkasan hasil evaluasi *Davies-Bouldin Index* dan *Silhouette Score* untuk kedua algoritma ditunjukkan pada Tabel 19, yang memperlihatkan perbandingan performa *K-Means* dan *K-Medoids* secara kuantitatif.

**Tabel 19.** Nilai Evaluasi *DBI* dan *Silhouette Score*

Algoritma	Davies-Bouldin Index	Silhouette Score
<i>K-Means</i>	0.5717	0.6012
<i>K-Medoids</i>	0.5870	0.5857

### Waktu Komputasi

Selain mengevaluasi kualitas klaster secara struktural, penelitian ini juga mengukur efisiensi proses dari masing-masing algoritma melalui waktu komputasi. Pengukuran dilakukan dengan memanfaatkan pustaka *time*, yang digunakan untuk mencatat durasi mulai dari awal proses iterasi hingga algoritma mencapai kondisi konvergen. Pengukuran ini bertujuan untuk memberikan gambaran tambahan mengenai performa teknis kedua algoritma dalam skenario *dataset e-commerce* dengan ukuran sedang sebagaimana digunakan dalam penelitian ini.

**Waktu Komputasi Algoritma *K-Means*.** Pada algoritma *K-Means*, waktu komputasi dihitung berdasarkan durasi antara sebelum dan sesudah proses *fit()* dijalankan pada data numerik. Hasil pengukuran menunjukkan bahwa waktu yang dibutuhkan untuk menyelesaikan proses klasterisasi adalah sekitar 0.0947 detik, sebagaimana ditunjukkan pada Gambar 48.

```
● ● ●

start_time_kmeans = time.time() # Mulai timer

# Proses K-Means dimulai
kmeans = KMeans(n_clusters=2, init='random', n_init=1, random_state=42)
kmeans.fit(X)

end_time_kmeans = time.time() # Selesai timer
print(f"Waktu komputasi K-Means: {end_time_kmeans - start_time_kmeans:.4f} detik")
```

**Gambar 48.** Kode *Python* Waktu Komputasi *K-Means*

**Waktu Komputasi Algoritma *K-Medoids*.** Sedangkan pada algoritma *K-Medoids*, waktu eksekusi diukur selama proses iteratif pembaruan *medoid* dan evaluasi total *cost* berlangsung. Durasi komputasi dicatat sejak sebelum iterasi dimulai hingga *medoid* tidak mengalami perubahan. Berdasarkan hasil eksekusi, proses klasterisasi selesai dalam waktu sekitar 0.1622 detik, sebagaimana ditunjukkan pada Gambar 49.



```

start_time_kmedoids = time.time() # Mulai timer

# Proses K-Medoids dimulai (dengan pairwise distances)
while True:
    distance_matrix = pairwise_distances(X, X[medoids])
    labels = np.argmin(distance_matrix, axis=1)

    # Evaluasi medoid baru
    ...
    if np.array_equal(medoids, new_medoids):
        break
    medoids = new_medoids
    iteration += 1

end_time_kmedoids = time.time() # Selesai timer
print(f"Waktu Komputasi K-Medoids: {end_time_kmedoids - start_time_kmedoids:.4f} detik")

```

**Gambar 49.** Kode Python Waktu Komputasi *K-Medoids*

Perbandingan waktu komputasi antara *K-Means* dan *K-Medoids* ditunjukkan pada Tabel 20. Dari hasil tersebut terlihat bahwa *K-Means* membutuhkan waktu eksekusi yang lebih singkat, sedangkan *K-Medoids* relatif lebih lama karena proses iteratifnya lebih kompleks.

**Tabel 20.** Perbandingan Waktu Komputasi *K-Means* dan *K-Medoids*

Algoritma	Waktu Eksekusi (detik)	Keterangan
<i>K-Means</i>	0.0947	Lebih cepat
<i>K-Medoids</i>	0.1622	Proses lebih kompleks

#### 4.7 Interpretasi Hasil

Setelah seluruh rangkaian proses mulai dari pengumpulan data, *preprocessing*, transformasi, hingga penentuan jumlah klaster optimal selesai dilakukan, tahapan selanjutnya adalah menginterpretasikan hasil klasterisasi. Klasterisasi telah diterapkan pada dataset produk Tokopedia menggunakan dua algoritma, yakni *K-Means* dan *K-Medoids*, dengan jumlah klaster optimal yang ditentukan berdasarkan pendekatan *Elbow Method*, yaitu  $K = 2$ . Implementasi algoritma dilakukan secara iteratif hingga mencapai konvergen, dan hasil akhirnya telah disimpan di dalam direktori utama laptop pribadi penulis dalam bentuk file *output* sesuai dengan label klaster masing-masing.

Interpretasi hasil bertujuan untuk memberikan pemahaman yang lebih mendalam mengenai struktur dan karakteristik dari masing-masing kelompok produk yang terbentuk berdasarkan hasil klasterisasi. Melalui proses ini, peneliti dapat mengidentifikasi ciri khas tiap klaster, pola distribusi atribut, serta hubungan antara kategori produk dengan performa numeriknya. Berikut ini adalah tahapan interpretasi yang dilakukan:

### **Pengelompokan Data Berdasarkan Hasil Klasterisasi**

Setelah proses klasterisasi dilakukan menggunakan metode *K-Means* dan *K-Medoids*, seluruh data produk berhasil dikelompokkan ke dalam dua klaster utama, yaitu klaster C1 dan C2. Setiap klaster merepresentasikan karakteristik tertentu dari produk berdasarkan kemiripan fitur yang telah dinormalisasi, seperti harga, kategori, *rating*, jumlah ulasan, dan jumlah produk terjual. Proses pengelompokan ini bertujuan untuk memisahkan produk ke dalam kelompok homogen yang dapat dianalisis lebih lanjut. Hasil pembagian jumlah produk pada masing-masing klaster ditunjukkan pada Tabel 21.

**Tabel 21.** Jumlah Produk per Klaster Hasil *K-Means* dan *K-Medoids*

<b>Metode</b>	<b>Klaster</b>	<b>Jumlah Produk</b>
<i>K-Means</i>	C1	880
<i>K-Means</i>	C2	957
<i>K-Medoids</i>	C1	973
<i>K-Medoids</i>	C2	864

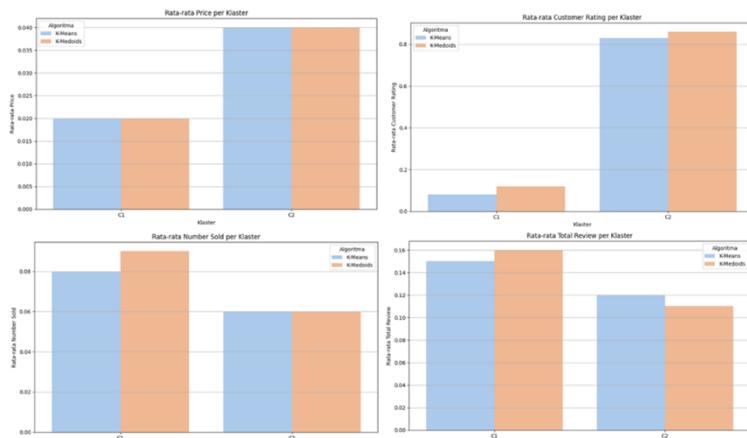
Perbedaan jumlah anggota tiap klaster mencerminkan perbedaan cara penentuan pusat *K-Means* menggunakan *centroid* (rata-rata titik), sedangkan *K-Medoids* memilih objek aktual sebagai pusat klaster (*medoid*). Selanjutnya, data yang telah dikelompokkan ini dapat digunakan untuk analisis lebih lanjut, seperti identifikasi karakteristik produk dalam masing-masing klaster, evaluasi performa penjualan antar klaster, serta strategi pemasaran yang lebih terarah sesuai segmen.

### **Analisis Ciri-Ciri Masing-Masing Klaster**

Analisis karakteristik tiap klaster dilakukan dengan algoritma *K-Means* dan *K-Medoids* melalui statistik deskriptif pada atribut *Price*, *Customer Rating*, *Number Sold*, dan *Total Review*. Keempat atribut ini dipilih karena mencerminkan aspek penting performa produk *e-commerce*, meliputi harga, kepuasan konsumen, intensitas pembelian, dan interaksi ulasan. Perbandingan rata-rata tiap atribut antar klaster divisualisasikan pada Gambar 50 untuk mengungkap pola segmentasi yang terbentuk.

Berdasarkan hasil visualisasi terlihat bahwa klaster C1 memiliki rata-rata *Number Sold* dan *Total Review* yang lebih tinggi dibandingkan C2 baik pada *K-Means* maupun *K-Medoids*. Hal ini menunjukkan bahwa C1 cenderung berisi produk yang populer, sering dibeli, dan banyak mendapatkan perhatian pengguna. Nilai *Price* pada klaster ini relatif lebih rendah dibandingkan C2, yang mengindikasikan bahwa produk dengan harga lebih terjangkau cenderung memiliki volume penjualan dan ulasan yang tinggi. Namun, nilai *Customer Rating* pada C1 lebih rendah dibandingkan C2, yang berarti tingginya aktivitas pembelian tidak selalu diikuti oleh kepuasan pengguna. Sebaliknya, klaster C2 memiliki *Customer Rating* lebih tinggi, namun angka *Number Sold* dan *Total Review* lebih rendah, serta *Price* yang sedikit lebih tinggi. Pola ini menggambarkan bahwa C2 berisi produk dengan kualitas yang diakui pengguna, namun belum menjangkau pasar luas atau masih relatif baru di pasaran. Sebagai contoh, produk “*obat herbal komplit 12 macam*” masuk ke C1 karena memiliki jumlah terjual dan ulasan tinggi walaupun berada pada rentang harga menengah, sedangkan produk “*wireless keyboard i8 mini touchpad mouse*” termasuk ke C2 karena memiliki *rating* tinggi namun jumlah ulasan dan transaksi yang lebih rendah.

Secara keseluruhan, analisis ini menunjukkan bahwa klaster C1 merepresentasikan produk dengan performa komersial tinggi yang ditunjukkan oleh volume penjualan dan ulasan yang besar serta harga yang kompetitif, sedangkan klaster C2 menonjol pada kualitas produk dengan tingkat kepuasan pengguna lebih tinggi. Informasi ini dapat dimanfaatkan untuk menyusun strategi promosi dan pengelolaan produk. Pada C1, strategi yang tepat adalah menjaga kualitas layanan dan meminimalkan keluhan pelanggan untuk mempertahankan pangsa pasar, sedangkan pada C2, disarankan untuk meningkatkan promosi dan visibilitas produk agar penjualannya dapat meningkat tanpa mengorbankan kualitas yang telah diakui pengguna.

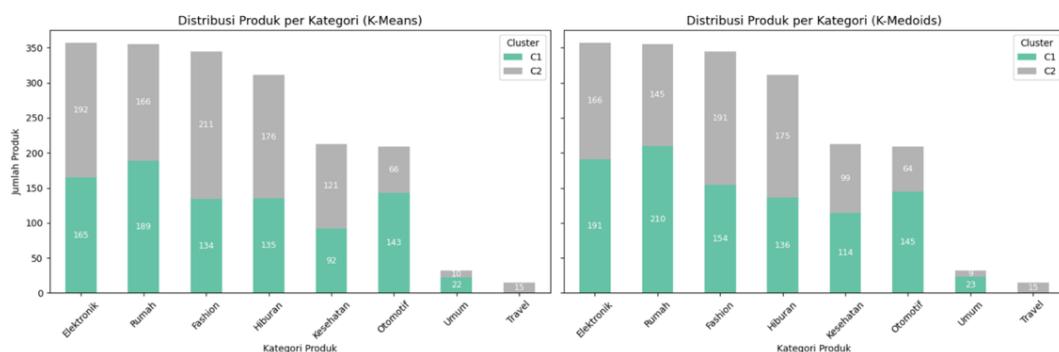


**Gambar 50.** Analisis Ciri-Ciri Masing-Masing Klaster

### Distribusi Kategori Produk dalam Tiap Klaster

Tahap ini memanfaatkan kembali atribut kategori produk yang sebelumnya tidak digunakan dalam pembentukan klaster untuk melihat pola distribusi di setiap klaster. Atribut *Main Category* digunakan untuk mengamati sebaran jenis produk pada klaster hasil algoritma *K-Means* dan *K-Medoids*, sehingga dapat diketahui kecenderungan suatu kategori menempati klaster tertentu sekaligus memberikan konteks bisnis yang relevan dari segmentasi yang terbentuk. Distribusi hasil pengelompokan kategori produk tersebut dapat dilihat pada Gambar 51.

Berdasarkan grafik distribusi, kategori Elektronik dan Rumah menunjukkan persebaran relatif seimbang antara C1 dan C2 pada kedua algoritma, meskipun pada hasil *K-Medoids* terlihat sedikit dominan di C1. Kategori Fashion dan Hiburan cenderung mendominasi C2, yang mengindikasikan potensi kualitas produk yang baik namun belum mencapai volume penjualan tinggi. Untuk kategori ini, strategi yang tepat adalah meningkatkan promosi, memperluas kanal distribusi, dan memanfaatkan kampanye diskon atau bundling untuk menarik pembelian. Sebaliknya, kategori Kesehatan dan Otomotif lebih kuat pada klaster C1, yang menunjukkan performa komersial tinggi dari sisi penjualan dan ulasan, namun dengan *rating* yang lebih rendah. Strategi yang disarankan adalah menjaga kualitas layanan dan meningkatkan kepuasan pelanggan agar performa tetap stabil. Kategori Umum dan Travel cenderung lebih dominan di C2, menandakan tingkat penjualan yang lebih rendah, sehingga diperlukan strategi peningkatan visibilitas produk, optimalisasi deskripsi dan foto produk, serta pemanfaatan program gratis ongkir untuk menarik minat pembeli. Pola distribusi ini memberikan wawasan yang dapat dimanfaatkan pelaku usaha dan analis *e-commerce* untuk merancang strategi pemasaran yang lebih terarah. Dengan mengetahui kategori mana yang unggul di klaster performa tinggi maupun rendah, langkah pengembangan produk dan promosi dapat disesuaikan dengan perilaku pasar dan segmentasi konsumen yang terpetakan melalui hasil klasterisasi.



**Gambar 51.** Perbandingan Distribusi Produk

### **Evaluasi Kuantitatif Klasterisasi**

Melalui analisis lebih lanjut terhadap hasil evaluasi kuantitatif yang telah diperoleh, dapat disimpulkan bahwa algoritma *K-Means* menunjukkan performa yang lebih optimal dibandingkan *K-Medoids*. Nilai *DBI* yang lebih rendah dan *Silhouette Score* yang lebih tinggi pada *K-Means* menjadi indikator kuat bahwa klaster yang terbentuk memiliki pemisahan yang lebih tegas serta kekompakan internal yang lebih baik antar anggota dalam setiap klaster. Dengan struktur pengelompokan yang lebih solid dan kemampuan dalam membedakan pola data secara konsisten, *K-Means* memberikan hasil yang lebih representatif dalam mengelompokkan produk berdasarkan karakteristiknya. Maka dari itu, *K-Means* dinilai lebih unggul dan lebih tepat digunakan sebagai pendekatan segmentasi pada data *e-commerce* dalam penelitian ini.

### **Pengukuran Waktu Komputasi Algoritma**

Dari hasil pengukuran waktu komputasi yang dilakukan sebelumnya, *K-Means* menunjukkan efisiensi yang lebih tinggi dengan waktu eksekusi sebesar 0.0947 detik. Durasi ini sedikit jauh lebih cepat dibandingkan *K-Medoids* yang membutuhkan waktu 0.1622 detik untuk mencapai konvergen. Perbedaan ini mencerminkan bahwa *K-Means* memiliki keunggulan dari sisi kecepatan karena proses optimasinya lebih sederhana dan tidak memerlukan evaluasi total cost di setiap iterasi, seperti pada *K-Medoids*. Oleh karena itu, dalam aspek efisiensi waktu, *K-Means* lebih sesuai digunakan pada skenario data skala menengah seperti penelitian ini.

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, beberapa kesimpulan yang dapat diambil adalah sebagai berikut:

1. Proses klasterisasi terhadap data produk Tokopedia berhasil dilakukan menggunakan dua algoritma, yaitu *K-Means* dan *K-Medoids*. Keduanya mampu membentuk dua klaster utama berdasarkan atribut numerik hasil normalisasi, seperti harga, *rating*, jumlah ulasan, dan total penjualan. Selain menghasilkan segmentasi yang relevan, hasil klasterisasi juga menunjukkan bahwa algoritma *K-Means* lebih unggul dalam membentuk struktur klaster yang terarah, termasuk dalam hal distribusi kategori produk. Hal ini terlihat dari pola distribusi kategori yang lebih jelas, di mana kategori seperti Fashion dan Hiburan cenderung terkonsentrasi di klaster C2, sedangkan Kesehatan dan Otomotif mendominasi klaster C1. Konsistensi distribusi kategori produk pada hasil *K-Means* memberikan gambaran segmentasi yang lebih kuat secara kontekstual, menjadikannya lebih unggul tidak hanya dari segi teknis, tetapi juga dalam mengidentifikasi kecenderungan kategori dominan dan pola harga dalam setiap klaster. Selain itu, klaster pertama (C1) cenderung merepresentasikan produk dengan performa komersial tinggi ditinjau dari volume penjualan, eksposur ulasan, serta harga yang lebih kompetitif, sedangkan klaster kedua (C2) menonjol dalam hal kualitas produk berdasarkan penilaian pengguna melalui *rating* yang lebih tinggi dan harga yang relatif lebih tinggi pula. Dengan demikian, C1 dapat diidentifikasi sebagai klaster produk berorientasi pasar massal dengan harga kompetitif, sedangkan C2 merupakan klaster produk berorientasi kualitas dengan segmen harga menengah ke atas.
2. Evaluasi hasil klasterisasi menggunakan metode *Davies-Bouldin Index* dan *Silhouette Score* menunjukkan bahwa *K-Means* mampu memberikan kualitas klaster yang lebih baik dibandingkan *K-Medoids*. Klaster yang dihasilkan *K-Means* lebih kompak, terpisah dengan jelas, dan diperoleh dengan waktu komputasi yang lebih efisien. Hal ini menjadikan *K-Means* lebih tepat digunakan dalam analisis segmentasi produk *e-commerce* berskala menengah, baik dari segi efektivitas maupun efisiensi.

## 5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran yang dapat dipertimbangkan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Untuk produk pada klaster C1 yang memiliki jumlah terjual (*Number Sold*) dan ulasan (*Total Review*) tinggi dengan harga yang kompetitif, strategi yang tepat adalah mempertahankan performa penjualan dengan menjaga kualitas layanan, memastikan ketersediaan stok, serta meminimalkan keluhan pelanggan. Mengingat *rating* pada klaster ini relatif lebih rendah, peningkatan kualitas produk dan layanan tetap diperlukan agar kepuasan pelanggan meningkat. Sementara itu, pada klaster C2 yang memiliki *Customer Rating* tinggi namun angka penjualan dan ulasan relatif rendah serta harga yang sedikit lebih tinggi, strategi yang disarankan adalah meningkatkan visibilitas dan promosi produk, misalnya melalui program diskon, kampanye iklan, atau kolaborasi dengan *influencer*, agar volume penjualan dapat meningkat tanpa menurunkan kualitas yang telah diakui konsumen.
2. Analisis distribusi kategori produk dalam tiap klaster dapat diperdalam lebih lanjut, misalnya dengan mengaitkannya pada perilaku konsumen atau tren penjualan, agar hasil interpretasi tidak hanya bersifat deskriptif tetapi juga memiliki nilai strategis dalam konteks bisnis.

## DAFTAR PUSTAKA

- Afianti, Y., Ayu Ramadhani, N., Risnandyaa Rahmi, A., & Madiistriyanto, H. (2023). Pemasaran Digital Efektif Dalam Platform Tokopedia: Studi Kasus. *Journal of Comprehensive Science (JCS)*, 2(7), 1324–1328. <https://doi.org/10.5918/jcs.v2i7.455>
- Ainur Rahman, & Suroyo, H. (2021). Analisis Data Produk Elektronik Di E-Commerce Dengan Metode Algoritma K-Means Menggunakan Python. *Journal of Advances in Information and Industrial Technology*, 3(2), 11–18. <https://doi.org/10.52435/jaiit.v3i2.158>
- Alfiah, F., Farizi, D. Al, & Widodo, E. (2020). Analisis Clustering K-Medoids Berdasarkan Indikator Kemiskinan di Jawa Timur Tahun 2020. *Jurnal Ilmiah Sains*, 22(April), 1–7.
- Andini, A. D., & Arifin, T. (2020). Implementasi Algoritma K-Medoids Untuk Klasterisasi Data Penyakit Pasien Di Rsud Kota Bandung. *Jurnal Responsif: Riset Sains Dan Informatika*, 2(2), 128–138. <https://doi.org/10.51977/jti.v2i2.247>
- Anum, P. L., Sabila, J., Fransiska, R. M., & Damayanti, P. (2024). Pemberdayaan Aplikasi Digital Shop / Tokopedia Untuk Memaksimalkan Penjualan UMKM Gang Perwira Medan. *Jurnal Pengabdian West Science*, 03(12), 1288–1297.
- Ardi Hizban Ahmada. (2021). Strategi Tokopedia Dalam Meningkatkan Kualitas Sumber Daya Manusia Untuk Meningkatkan Produktivitas Dan Laba Bisnis. *JENIUS (Jurnal Ilmiah Manajemen Sumber Daya Manusia)*, April, 3–8.
- Asrawi, H. (2025). Implementasi Long Short Term Memory Pada Klasifikasi Teks. *Jurnal Ilmiah Sains, Teknologi, Ekonomi, Sosial Dan Budaya*, 9, 4–10.
- Asy Aria, T., Julkarnain, M., & Hamdani, F. (2023). Penerapan Algoritma K-Means Clustering Untuk Data Obat. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, 4(1), 649–657.
- Atikah, I. (2019). Pengaturan Hukum Transaksi Jual Beli Online (E-Commerce) Di Era Teknologi. *Muamalatuna*, 10(2), 1. <https://doi.org/10.37035/mua.v10i2.1811>
- Aulia, S. (2020). Klasterisasi Pola Penjualan Pestisida Menggunakan Metode K-Means Clustering (Studi Kasus Di Toko Juanda Tani Kecamatan Hutabayu Raja). *Djtechno: Jurnal Teknologi Informasi*, 1(1), 1–5. <https://doi.org/10.46576/djtechno.v1i1.964>
- Ayu, D., Cahya, I., Ayu, D., & Pramita, K. (2019). Analisis Perbandingan Metode Elbow dan Sillhouette pada Algoritma Clustering K-Medoids dalam Pengelompokan Produksi Kerajinan Bali. *Jurnal Matrix*, 9(3).

- Didi Riswan, Heri Eko Rahmadi Putra, & Risfan Nazar Saputra. (2024). Pengembangan Sistem Rekomendasi Berbasis Kecerdasan Buatan Untuk Meningkatkan Pengalaman Pengguna Di Platform E-Commerce. *Jurnal Komputer Teknologi Informasi Dan Sistem Informasi (JUKTISI)*, 2(3), 572–580. <https://doi.org/10.62712/juktisi.v2i3.145>
- Edy, M. R., Alif, A. A. N., & Hidayat, A. (2024). Pengembangan Aplikasi Monitoring Pelanggaran Siswa Berbasis Website Pada SMA Negeri 1 Parepare. *Jurnal MediatIK*, 1(1), 1–8. <https://doi.org/10.59562/mediatik.v6i2.1392>
- Farhan Nugraha, M., & Hayati, U. (2024). CLUSTERING DATA INDONESIAN FOOD DELIVERY MENGGUNAKAN METODE K-MEANS PADA GOFOOD PRODUCT LIST. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 3).
- Gymnastiar, S., & Bahtiar, A. (2024). Penerapan Algoritma K-Means Clustering Untuk Mengelompokan Data Kejadian Kekeringan Di Kabupaten Cirebon. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(2), 2325–2331. <https://doi.org/10.36040/jati.v8i2.8948>
- Harjono, S. W., Utami, N. W., & Putri, I. G. A. P. D. (2023). Klasterisasi Tingkat Penjualan pada Startup Panak.id dengan Algoritma K-Means. *Jurnal Ilmiah Teknologi Informasi Asia*, 17(1), 55–66. <https://doi.org/10.32815/jitika.v17i1.888>
- Haryanti, M. F., Fauzi, A., Jelita, A. A., Setiyowati, A., Octarina, A., Putra Edina, E., Zahra Aulia, R., & Fitriana, S. (2024). Pengaruh Data Mining, Strategi Perusahaan Terhadap Laporan Kinerja Perusahaan. *Jurnal Manajemen Dan Bisnis*, 3(1), 71–90.
- Hasan, Y. (2024). Pengukuran Silhouette Score dan Davies-Bouldin Index pada Hasil Cluster K-Means dan Dbscan. *KAKIFIKOM (Kumpulan Artikel Karya Ilmiah Fakultas Ilmu Komputer)*, 06(01), 60–74.
- Hendrastuty, N. (2024). Penerapan Data Mining Menggunakan Algoritma K-Means Clustering Dalam Evaluasi Hasil Pembelajaran Siswa. *Jurnal Ilmiah Informatika Dan Ilmu Komputer (Jima-IIkom)*, 3(1), 46–56.
- Hermawati, A., Jumini, S., Astuti, M., Ismail, F., & Rahim, R. (2020). Unsupervised Data Mining with K-Medoids Method in Mapping Areas of Student and Teacher Ratio in Indonesia. *TEM Journal*, 9(4), 1614–1618. <https://doi.org/10.18421/TEM94>
- Hidayat, Putra, Alfitrah, W. (2022). Implementasi Clustering K-Medoids dalam Pengelompokan Kabupaten. *Indonesian Journal of Applied Statistics*, 5(2), 121–130.

- Hoerunnisa, A., Dwilestari, G., Dikananda, F., Sunana, H., & Pratama, D. (2024). Komparasi Algoritma K-Means Dan K-Medoids Dalam Analisis Pengelompokan Daerah Rawan Kriminalitas Di Indonesia. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(1), 103–110. <https://doi.org/10.36040/jati.v8i1.8249>
- Kurmiati, D., Fauzi, M. Z., & Falegas, A. (2021). Klasterisasi Daerah Rawan Gempa Bumi di Indonesia Menggunakan Algoritma K-Medoids. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 1(April), 47–57.
- Meiyanti, R., Munauwar, M. M., Fitria, R., & Kautsar, H. Al. (2024). Implementasi Algoritma K-Medoid pada Clustering Sayuran Unggulan di Kabupaten Aceh Utara. *TEKNIKA*, 19(x), 327–337.
- Mirantika, N., Syamfithriani, T. S., & Trisudarmo, R. (2023). Implementasi Algoritma K-Medoids Clustering Untuk Menentukan Segmentasi Pelanggan. *Jurnal Nuansa Informatika*, 17(1), 2614–5405.
- Mubarok, Adjani, Hutama, Mutoffar, I. (2025). BIG DATA ANALYTICS DAN MACHINE LEARNING UNTUK MEMPREDIKSI PERILAKU KONSUMEN DI E-COMMERCE. *JIRE (Jurnal Informatika & Rekayasa Elektronika)*, 8(1), 159–167.
- Ningsih, R. &. (2024). ANALISIS POTENSI E-COMMERCE MELALUI IMPLEMENTASI DATA MINING DALAM PERPAJAKAN: SEBUAH STUDI KOMPARASI. *Juremi: Jurnal Riset Ekonomi*, 4(1), 155–168.
- Nugraha, M. F., Martano, M., & Hayati, U. (2024). Clustering Data Indonesian Food Delivery Menggunakan Metode K-Means Pada Gofood Product List. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(3), 3484–3492. <https://doi.org/10.36040/jati.v8i3.9727>
- Nur, N., Iqram, M., & Inayah, N. (2023). Perbandingan K-Means dan Hierarchical Clustering dalam Pengelompokan Daerah Beresiko Stunting. *JURNAL INOVTEK POLBENG*, 356–367.
- Pambudhi Ganang Aji , Homaidi Ahmad, S. F. (2024). KOMPARASI ALGORITMA K-MEANS DENGAN K-MEDOIDS DALAM KLASTERISASI WILAYAH RAWAN BENCANA DI KABUPATEN SITUBONDO. *Jurnal Teknik Elektro Dan Informatika*, 19(September), 173–179.
- Polgan, J. M., Rivaldo, M. D., Wahyu, G., Wibowo, N., Mulyo, H., Studi, P., Informatika, T., Nahdlatul, U. I., Jepara, U., Tua, K., & Jeron, T. (2024). Implementasi Algoritma K-Means untuk Klasterisasi Data Hasil Tangkapan Ikan di Karimunjawa. *Jurnal Minfo Polgan*, 13, 1045–1056.

- Prasetyo, D. Y., Yunita, F., Thaher, S., Studi, P., Informasi, S., Indragiri, U. I., Studi, P., Sipil, T., & Indragiri, U. I. (2024). E-COMMERCE FOR WEBSITE-BASED BAROKAH TEMPEH CHIPS PRODUCTS USING A USER-CENTRIC INTERFACE MODEL. *Jurnal Sistem Informasi (TEKNOFILE)*, 2(11), 847–857.
- Purba, D. S., Dwi Permatasari, P., Tanjung, N., Rahayu, P., Fitriani, R., Wulandari, S., Universitas, ), Negeri, I., Utara, S., Muslim, U., & Al Washliyah, N. (2025). Analisis Perkembangan Ekonomi Digital Dalam Meningkatkan Pertumbuhan Ekonomi Di Indonesia. *Jurnal Masharif Al-Syariah: Jurnal Ekonomi Dan Perbankan Syariah*, 10(1), 126–139.
- Putra, R. F., Zebua, R. S. Y., Budiman, B., Rahayu, P. W., Bangsa, M. T. A., AZulfadhlilah, M., Andiyan, A. (2023). *Data Mining: Algoritma dan Penerapannya*.
- Putra, Y. D., Sudarma, M., Bagus, I., & Swamardika, A. (2021). Cluster ing History Data Penjualan Menggunakan. *Majalah Ilmiah Teknologi Elektro*, 20(2).
- Rahmanto, S. B. T. (2022). *Model Bisnis E-Commerce*. eureka media aksara.
- Rahmawati, L. H., & Fasa, M. I. (2025). TRANSFORMASI DIGITAL : PERAN E-COMMERCE ( SHOPEE ) DALAM MENINGKATKAN DAYA SAING UMKM DI INDONESIA DIGITAL TRANSFORMATION : THE ROLE OF E-COMMERCE ( SHOPEE ) IN IMPROVING THE COMPETITIVENESS OF UMKM IN INDONESIA. *JIIC : JURNAL INTELEK INSAN CENDIKIA, April*, 6704–6712.
- Retnoningsih, E., & Pramudita, R. (2020). Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python. *Bina Insani Ict Journal*, 7(2), 156. <https://doi.org/10.51211/biict.v7i2.1422>
- Saputri, F. W., & Arianto, D. B. (2023). PERBANDINGAN PERFORMA ALGORITMA K-MEANS, K- MEDOIDS, DAN DBSCAN DALAM PENGEROMBOLAN PROVINSI DI INDONESIA BERDASARKAN INDIKATOR KESEJAHTERAAN MASYARAKAT. *JURNAL TEKNOLOGI INFORMASI*, 17(2), 138–151.
- Sari, S. N., Pratama, B. G., & Prastowo, R. (2024). Penggunaan Metode Elbow untuk Pemilihan Jumlah Klaster dalam Identifikasi Bahan Material Shelter Modular. *ReTII*, 2024(November), 157–163.
- Sari, V. K., & Nasution, M. I. P. (2024). Dampak E-commerce Terhadap Perkembangan Digital. *Jurnal Akademik Ekonomi Dan Manajemen*, 1(4), 18–24.
- Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3), 160. <https://doi.org/10.1007/s42979-021-00592-x>

- Sulistiyawan, E., Hapsery, A., & Arifahanum, L. J. A. (2021). PERBANDINGAN METODE OPTIMASI UNTUK PENGELOMPOKAN PROVINSI BERDASARKAN SEKTOR PERIKANAN DI INDONESIA (Studi Kasus Dinas Kelautan dan Perikanan Indonesia). *Jurnal Gaussian*, 10(1), 76–84. <https://doi.org/10.14710/j.gauss.v10i1.30936>
- Sutoyo, R. C. A. A. S. A. E. W. M. I. S. S. J. P. A. J. R. ; R. R. ; H. M. A. ; A. D. ; P. F. P. (2022). Product Reviews Dataset for Emotions Classification Tasks - Indonesian (PRDECT-ID) Dataset. *Procedia Computer Science*, 179. <https://doi.org/10.1016/j.procs.2021.01.058>
- Syahkur, M. R., & Hartama, D. (2024). Evaluasi Jumlah Cluster pada Algoritma K-Means ++ Menggunakan Silhouette dan Elbow dengan Validasi Nilai DBI dalam Mengelompokkan Gizi Balita. *Jurnal Sains Dan Teknologi*, 13(3), 487–496.
- Umagapi, I. T., & Umaternate, B. (2023). Uji Kinerja K-Means Clustering Menggunakan Davies-Bouldin Index Pada Pengelompokan Data Prestasi Siswa. *PROSIDING SISFOTEK*, 303–308.
- Wakhidah, N. (2010). Clustering Menggunakan K-Means Algorithm. *Jurnal Transformatika*, 8(1), 33–39. <https://doi.org/10.26623/transformatika.v8i1.45>
- Wayan, N., & Damayanthi, R. (2024). *Penerapan Metode K-means Pada Klasterisasi Provinsi di Indonesia Berdasarkan Indikator Indeks Kebahagiaan*. 14(1), 61–74. <https://doi.org/10.24843/JMAT.2024.v14.i01.p172>
- Wenerda, I., & Hariyanti, N. (2024). Penggunaan Dompet Digital dalam Transaksi Daring bagi Millennial Moms selama Pandemi Covid-19. *Jurnal Ilmu Komunikasi*, 21(3), 465. <https://doi.org/10.31315/jik.v21i3.5572>
- Wijaya, J., Lim, A., & Adnas, D. A. (2025). Analysis of E-Commerce Applications in Generation Z with the System Usability Scale Approach Analisis Aplikasi E-Commerce pada Generasi Z dengan Pendekatan System Usability Scale. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 5(April), 645–655.
- Yafi, M., Goejantoro, R., & Dani, A. T. R. (2023). Pengelompokan Algoritma K-Medoids Dengan Principal Component Analysis (PCA) (Studi Kasus : Kabupaten/Kota di Pulau Kalimantan Berdasarkan Indikator Kemiskinan Tahun 2021). *Prosiding Seminar Nasional Matematika Dan Statistika*, 3(1), 183–195.

## DAFTAR LAMPIRAN

### Lampiran 1. Kode Program Lengkap Tahapan Clustering Produk Tokopedia

```
INFORMASI NOTE BOOK
NAMA Muhammad Raihan Malik
NIM F1E121145
JUDUL ANALISIS KLASTERISASI PRODUK TOKOPEDIA DENGAN ALGORITMA K-MEANS DAN K-MEDOIDS
PEMBIMBING I Pradita Eko Prasetyo Utomo, S.Pd., M.Cs.
PEMBIMBING II Benedika Ferdian Hutabarat, S.Komp., M.Kom.

SETUP
!pip install openpyxl
!pip install pyclustering

Tampilkan output tersembunyi

# Modul bawaan
import time

# Modul pihak ketiga
import numpy as np
import pandas as pd

# Visualisasi
import matplotlib.pyplot as plt
import seaborn as sns

# Google Colab
from google.colab import files

# Display
from IPython.display import display, HTML

# Preprocessing & Evaluasi
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import silhouette_score, davies_bouldin_score, pairwise_distances
from sklearn.cluster import KMeans

# PyClustering
from pyclustering.cluster.kmedoids import kmedoids
from pyclustering.utils import calculate_distance_matrix

uploaded = files.upload()

Tampilkan output tersembunyi
```

#### ▼ Preprocessing Data

```
# Baca file Excel
df = pd.read_excel(list(uploaded.keys())[0])

# Tambahkan kolom nomor urut
df.insert(0, 'No', range(1, len(df) + 1))

# Tampilkan 10 data awal (semua kolom, horizontal)
print("Contoh data mentah (10 entri pertama, semua atribut):")
display(HTML(df.head(10).to_html(index=False)))

# Standarisasi teks kolom kategori dan produk
df['Product Name'] = df['Product Name'].astype(str).str.strip().str.lower()
df['Category'] = df['Category'].astype(str).str.strip().str.lower()

# Mapping kategori ke kategori utama (Main Category)
kategori_map = {
    "men's fashion": "Fashion",
    "women's fashion": "Fashion",
    "kids and baby fashion": "Fashion",
    "muslim fashion": "Fashion",
    "beauty": "Kesehatan",
    "body care": "Kesehatan",
    "health": "Kesehatan",
    "mother and baby": "Kesehatan",
    "household": "Rumah",
    "kitchen": "Rumah",
    "food and drink": "Rumah",
    "animal care": "Rumah",
```

```

"party supplies and craft": "Rumah",
"electronics": "Elektronik",
"phones and tablets": "Elektronik",
"computers and laptops": "Elektronik",
"camera": "Elektronik",
"gaming": "Hiburan",
"toys and hobbies": "Hiburan",
"movies and music": "Hiburan",
"books": "Hiburan",
"automotive": "Otomotif",
"carpentry": "Otomotif",
"sport": "Otomotif",
"office & stationery": "Otomotif",
"tour and travel": "Travel",
"other products": "Umum"
}
df[['Main Category']] = df['Category'].map(kategori_map).fillna('Umum')

# Hapus nilai kosong di kolom penting
df = df.dropna(subset=['Category', 'Product Name', 'Customer Rating', 'Number Sold', 'Price', 'Total Review'])

# Ubah kolom numerik ke float
df['Price'] = pd.to_numeric(df['Price'], errors='coerce')
df['Customer Rating'] = pd.to_numeric(df['Customer Rating'], errors='coerce')
df['Number Sold'] = pd.to_numeric(df['Number Sold'], errors='coerce')
df['Total Review'] = pd.to_numeric(df['Total Review'], errors='coerce')
df = df.dropna(subset=['Price', 'Customer Rating', 'Number Sold', 'Total Review'])

# Hapus duplikat
df = df.drop_duplicates(subset=['Category', 'Product Name', 'Price', 'Customer Rating', 'Number Sold', 'Total Review'])

# Filter kategori yang jumlah datanya minimal 20
kategori_ok = df['Category'].value_counts()[df['Category'].value_counts() >= 20].index
df = df[df['Category'].isin(kategori_ok)]

# Bersihkan outlier per kategori dengan IQR
cleaned_dfs = []
for cat in df['Category'].unique():
    subset = df[df['Category'] == cat].copy()
    for col in ['Customer Rating', 'Number Sold', 'Price', 'Total Review']:
        Q1 = subset[col].quantile(0.25)
        Q3 = subset[col].quantile(0.75)
        IQR = Q3 - Q1
        lower = Q1 - 1.5 * IQR
        upper = Q3 + 1.5 * IQR
        subset = subset[(subset[col] >= lower) & (subset[col] <= upper)]
    cleaned_dfs.append(subset)

df_cleaned = pd.concat(cleaned_dfs).reset_index(drop=True)

# Buat ulang nomor urut berdasarkan data hasil cleaning
df_cleaned['No'] = range(1, len(df_cleaned) + 1)

# Pilih kolom akhir yang akan digunakan
df_final = df_cleaned[['No', 'Category', 'Main Category', 'Product Name', 'Price', 'Customer Rating', 'Number Sold', 'Total Review']]

# Bulatkan angka dan ubah ke integer
for col in ['Price', 'Customer Rating', 'Number Sold', 'Total Review']:
    df_final[col] = df_final[col].round(0).astype(int)

# Tampilkan 10 entri acak setelah cleaning
print("Data setelah cleaning (10 entri acak):")
display(HTML(df_final.sample(10, random_state=42).to_html(index=False)))

# Simpan ke Excel dan unduh otomatis
df_final.to_excel("tokopedia_cleaned_final.xlsx", index=False)

from google.colab import files
files.download("tokopedia_cleaned_final.xlsx")

```

[Tampilkan output tersembunyi](#)

#### Visualisasi Barchart

```

# Load dataset mentah dan hasil cleaning
raw_df = pd.read_excel("NEW DATASET TOKOPEDIA SKRIPSI.xlsx")
cleaned_df = pd.read_excel("tokopedia_cleaned_final.xlsx")

# Hitung jumlah entri
count_before = len(raw_df)
count_after = len(cleaned_df)

```

```

# Visualisasi jumlah data sebelum dan sesudah
plt.figure(figsize=(6, 5))
plt.bar(['Sebelum', 'Sesudah'], [count_before, count_after], color=['salmon', 'seagreen'])
plt.title('Jumlah Data Sebelum dan Sesudah Pembersihan')
plt.ylabel('Jumlah Entri')
plt.xlabel('Tahapan Pembersihan')
plt.tight_layout()

# Simpan gambar
plt.savefig("pembersihan_data_kosong_duplikat.png")
plt.show()

```

Tampilkan output tersembunyi

#### Visualisasi Pie Chart

```

# Load dataset hasil cleaning
df = pd.read_excel("tokopedia_cleaned_final.xlsx")

# Hitung jumlah produk per kategori utama
category_counts = df['Main Category'].value_counts().sort_values(ascending=False)

# Set warna-warna dan gaya grafik
colors = sns.color_palette('Set2', len(category_counts))

# Buat pie chart
plt.figure(figsize=(8, 8))
plt.pie(
    category_counts.values,
    labels=category_counts.index,
    autopct='%.1f%%',
    colors=colors,
    startangle=140
)
plt.title("Kategori Utama Setelah Seleksi Jumlah Data Memadai")
plt.axis('equal') # Membuat pie chart berbentuk lingkaran proporsional

# Simpan grafik ke file PNG
plt.savefig("piechart_kategori_setelah_seleksi.png", dpi=300)

# Tampilkan di layar
plt.show()

```

Tampilkan output tersembunyi

#### Visualisasi Boxplot dan Countplot

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load data mentah dan hasil cleaning
raw_data = pd.read_excel("NEW DATASET TOKOPEDIA SKRIPSI.xlsx")
cleaned_data = pd.read_excel("tokopedia_cleaned_final.xlsx")

# Daftar atribut numerik yang digunakan dalam penelitian
numerical_columns = ['Price', 'Number Sold', 'Total Review']

# Konversi seluruh kolom numerik ke tipe numerik (koersif)
for col in numerical_columns:
    raw_data[col] = pd.to_numeric(raw_data[col], errors='coerce')
    cleaned_data[col] = pd.to_numeric(cleaned_data[col], errors='coerce')

# Tambahkan kolom sumber untuk identifikasi visualisasi
raw_data['Sumber'] = 'Sebelum'
cleaned_data['Sumber'] = 'Sesudah'

# Buat visualisasi boxplot untuk setiap atribut numerik
for col in numerical_columns:
    # Gabungkan data sebelum dan sesudah untuk atribut saat ini
    df_plot = pd.concat([
        raw_data[[col, 'Sumber']],
        cleaned_data[[col, 'Sumber']]
    ])

    # Buat plot
    plt.figure(figsize=(8, 6))
    sns.boxplot(x='Sumber', y=col, data=df_plot, palette='pastel')

```

```

plt.title(f"Boxplot {col} Sebelum dan Sesudah Pembersihan Outlier")
plt.xlabel("Tahapan Data")
plt.ylabel(col)
plt.tight_layout()

# Simpan setiap gambar ke file PNG
filename = f"boxplot_{col.lower().replace(' ', '_')}_before_after.png"
plt.savefig(filename, dpi=300)

# Tampilkan
plt.show()

# Buat countplot untuk Customer Rating setelah cleaning
plt.figure(figsize=(6, 4))
sns.countplot(x='Customer Rating', data=cleaned_data, palette='pastel')
plt.title("Distribusi Customer Rating Setelah Cleaning")
plt.xlabel("Customer Rating")
plt.ylabel("Jumlah Produk")
plt.tight_layout()
plt.savefig("countplot_customer_rating_cleaned.png", dpi=300)
plt.show()

```

Tampilkan output tersembunyi

#### ▼ Transformasi

```

# Langsung baca file hasil cleaning
file_path = "tokopedia_cleaned_final.xlsx"
df_final = pd.read_excel(file_path)

# Perbarui kolom 'No' agar berurutan dari awal
df_final['No'] = range(1, len(df_final) + 1)

# Ambil kolom numerik
fitur_numerik = df_final[['Price', 'Customer Rating', 'Number Sold', 'Total Review']]

# Terapkan Min-Max Scaling
scaler = MinMaxScaler()
fitur_scaled = scaler.fit_transform(fitur_numerik)

# Buat DataFrame hasil normalisasi
df_normalized = pd.DataFrame(fitur_scaled, columns=fitur_numerik.columns)

# Gabungkan dengan kolom referensi
df_normalized_final = pd.concat([
    df_final[['No', 'Category', 'Main Category', 'Product Name']].reset_index(drop=True),
    df_normalized
], axis=1)

# Tampilkan 10 entri acak
print("Data setelah Min-Max Scaling (tanpa pembulatan):")
display(HTML(df_normalized_final.sample(10, random_state=42).to_html(index=False)))

# Simpan hasil normalisasi
df_normalized_final.to_excel("tokopedia_normalized_full.xlsx", index=False)

```

Tampilkan output tersembunyi

#### ▼ Pendekatan Klaster Optimal

##### ▼ Elbow Methode K-Means

```

# Load data normalisasi
df = pd.read_excel('tokopedia_normalized_full.xlsx')

# Ambil fitur numerik untuk clustering
X = df[['Customer Rating', 'Number Sold', 'Total Review']]

# Hitung inertia untuk K = 1 hingga 10
inertia = []
K = range(1, 11)
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X)
    inertia.append(kmeans.inertia_)


```

```
# Plot Elbow Method
plt.figure(figsize=(8, 5))
plt.plot(K, inertia, marker='o', linestyle='-', color='green')
plt.title('Elbow Method for Optimal K (K-Means)', fontsize=14)
plt.xlabel('Jumlah Cluster (K)')
plt.ylabel('Inertia (Within-cluster SSE)')
plt.xticks(K)
plt.grid(True)
plt.show()

Tampilkan output tersembunyi
```

#### ▼ Elbow Methode K-Medoids

```
# Load data normalisasi
df = pd.read_excel('tokopedia_normalized_full.xlsx')
X = df[['Customer Rating', 'Number Sold', 'Total Review']].values

# Hitung total cost K-Medoids untuk K = 1 sampai 10
K = range(1, 11)
total_cost = []

# Buat distance matrix terlebih dahulu
distance_matrix = calculate_distance_matrix(X)

for k in K:
    # Pilih medoid awal secara acak
    initial_medoids = list(np.random.choice(len(X), k, replace=False))
    kmmedoids_instance = kmmedoids(distance_matrix, initial_medoids, data_type='distance_matrix')
    kmmedoids_instance.process()
    clusters = kmmedoids_instance.get_clusters()
    medoids = kmmedoids_instance.get_medoids()

    # Hitung total cost (jumlah jarak tiap titik ke medoid-nya)
    cost = 0
    for idx, cluster in enumerate(clusters):
        medoid_idx = medoids[idx]
        cost += np.sum([distance_matrix[i][medoid_idx] for i in cluster])
    total_cost.append(cost)

# Plot Elbow Method K-Medoids
plt.figure(figsize=(8, 5))
plt.plot(K, total_cost, marker='s', linestyle='-', color='green')
plt.title('Elbow Method for Optimal K (K-Medoids)', fontsize=14)
plt.xlabel('Jumlah Cluster (K)')
plt.ylabel('Total Cost (Sum of Distances)')
plt.xticks(K)
plt.grid(True)
plt.show()

Tampilkan output tersembunyi
```

#### ▼ Implementasi K-Means (K2) Optimal dan Waktu Komputasi

```
K = 2

# Baca file dan acak datanya satu kali
file_path = "tokopedia_normalized_full.xlsx"
df = pd.read_excel(file_path)
df = df.sample(frac=1, random_state=42).reset_index(drop=True)
df['No'] = np.arange(1, len(df) + 1)

# Ambil fitur numerik
X = df[['Price', 'Customer Rating', 'Number Sold', 'Total Review']].values

# Inisialisasi centroid awal secara acak (tetap)
np.random.seed(42)
centroids = X[np.random.choice(len(X), K, replace=False)]

iteration = 0

# Mulai waktu komputasi
start_time = time.time()

while True:
    print(f"\nIterasi {iteration + 1}")

    iteration += 1
```

```

# Hitung jarak ke setiap centroid
distances = np.zeros((X.shape[0], K))
for i in range(K):
    distances[:, i] = np.linalg.norm(X - centroids[i], axis=1)

labels = np.argmin(distances, axis=1)

# Buat DataFrame iterasi
df_iter = df[['No', 'Product Name', 'Category', 'Main Category', 'Price', 'Customer Rating', 'Number Sold',
for i in range(K):
    df_iter['C{i+1}'] = distances[:, i]
df_iter['Cluster'] = ['C' + str(i+1) for i in labels]

# Tampilkan 10 data pertama
print("Tabel Hasil Iterasi (10 Data Awal):")
display(HTML(df_iter.head(10).to_html(index=False)))

# Tampilkan centroid saat ini
print("Centroid pada Iterasi Ini:")
for i in range(K):
    print(f"C{i+1}: {centroids[i]}")

# Update centroid
new_centroids = np.zeros_like(centroids)
for i in range(K):
    cluster_points = X[labels == i]
    if len(cluster_points) > 0:
        new_centroids[i] = cluster_points.mean(axis=0)
    else:
        new_centroids[i] = centroids[i]

# Cek konvergensi
if np.allclose(centroids, new_centroids):
    print("\nKonvergen pada iterasi ke-{iteration + 1}")
    print("\nTabel Final Setelah Konvergen (iterasi ke-{iteration + 1})")
    display(HTML(df_iter.head(10).to_html(index=False)))

    print("Centroid Final:")
    for i in range(K):
        print(f"C{i+1}: {centroids[i]}")

    # Akhiri timer dan tampilkan waktu komputasi
    end_time = time.time()
    print(f"\nWaktu Komputasi K-Means: {end_time - start_time:.4f} detik")

    # Simpan hasil akhir ke Excel
    filename = f"hasil_kmeans_k{K}_final.xlsx"
    df_iter.to_excel(filename, index=False)
    files.download(filename)

    # Simpan label akhir berdasarkan centroid konvergen
    labels_kmeans = np.argmin(np.linalg.norm(X[:, np.newaxis] - centroids, axis=2), axis=1)
    break

centroids = new_centroids
iteration += 1

```

Tampilkan output tersembunyi

#### ▼ Implementasi K-Medoids (K2) Optimal dan Waktu Komputasi

```

# Baca dan acak data
file_path = "tokopedia_normalized_full.xlsx"
df = pd.read_excel(file_path)
df = df.sample(frac=1, random_state=42).reset_index(drop=True)
df['No'] = np.arange(1, len(df) + 1)

# Ambil fitur numerik
X = df[['Price', 'Customer Rating', 'Number Sold', 'Total Review']].values

# Inisialisasi jumlah cluster dan medoid awal
K = 2
np.random.seed(42)
medoids = np.random.choice(len(X), K, replace=False)

iteration = 0

# Mulai waktu komputasi

```

```

start_time = time.time()

while True:
    print(f"\n Iterasi {iteration + 1}")

    # Hitung jarak dari semua data ke setiap medoid
    distance_matrix = pairwise_distances(X, X[medoids])
    labels = np.argmin(distance_matrix, axis=1)

    # Hitung total cost untuk iterasi ini
    total_cost = np.sum(np.min(distance_matrix, axis=1))

    # Buat DataFrame hasil iterasi
    df_iter = df[['No', 'Product Name', 'Category', 'Main Category',
                  'Price', 'Customer Rating', 'Number Sold', 'Total Review']].copy()
    for i in range(K):
        df_iter[f'Ke C{i+1}'] = distance_matrix[:, i]
    df_iter['Cluster'] = ['C' + str(i+1) for i in labels]

    # Tampilkan 10 data pertama hasil iterasi
    print("\n Tabel Hasil Iterasi (10 Data Awal):")
    display(HTML(df_iter.head(10).to_html(index=False)))

    # Tampilkan total cost setelah tabel
    print(f"\n Total Cost Iterasi {iteration + 1}: {total_cost:.6f}")

    # Tampilkan medoid saat ini
    print("Medoid pada Iterasi Ini:")
    for i in range(K):
        print(f"C{i+1} (Index {medoids[i]}): {X[medoids[i]]}")

    # Evaluasi medoid baru untuk tiap cluster
    new_medoids = []
    for i in range(K):
        cluster_indices = np.where(labels == i)[0]
        if len(cluster_indices) == 0:
            new_medoids.append(medoids[i])
            continue
        intra_distances = pairwise_distances(X[cluster_indices], X[cluster_indices])
        cost = intra_distances.sum(axis=1)
        best_index = cluster_indices[np.argmin(cost)]
        new_medoids.append(best_index)

    new_medoids = np.array(new_medoids)

    # Cek konvergensi
    if np.array_equal(medoids, new_medoids):
        print("\n Konvergen pada iterasi ke-(iteration + 1)")
        print("Medoid Final (Tidak berubah):")
        for i in range(K):
            print(f"C{i+1} (Index {medoids[i]}): {X[medoids[i]]}")

        # Akhir waktu komputasi
        end_time = time.time()
        print(f"\n Waktu Komputasi K-Medoids: {end_time - start_time:.4f} detik")

        # Tampilkan tabel akhir
        print("\n Tabel Final Setelah Konvergen (10 Data Awal):")
        display(HTML(df_iter.head(10).to_html(index=False)))
        print(f"\n Total Cost Final: {total_cost:.6f}")

        # Simpan hasil akhir ke Excel
        filename = f"hasil_kmedoids_k{K}_final.xlsx"
        df_iter.to_excel(filename, index=False)
        files.download(filename)

        # Simpan label akhir berdasarkan medoid konvergen
        labels_kmedoids = np.argmin(pairwise_distances(X, X[medoids]), axis=1)
        break

    # Perbarui medoid
    medoids = new_medoids
    iteration += 1

```

Tampilkan output tersembunyi

## ✓ Evaluasi Clustering

▼

### Davies-Bouldin Index (DBI) dan Silhouette Score K-Means

```
print("\nEvaluasi K-Means")
dbi_kmeans = davies_bouldin_score(X, labels_kmeans)
sil_kmeans = silhouette_score(X, labels_kmeans)

print(f"Davies-Bouldin Index (DBI): {dbi_kmeans:.4f}")
print(f"Silhouette Score: {sil_kmeans:.4f}")

Tampilkan output tersembunyi
```

#### ▼ Davies-Bouldin Index (DBI) dan Silhouette Score K-Medoids

```
print("\nEvaluasi K-Medoids")
dbi_kmedoids = davies_bouldin_score(X, labels_kmedoids)
sil_kmedoids = silhouette_score(X, labels_kmedoids)

print(f"Davies-Bouldin Index (DBI): {dbi_kmedoids:.4f}")
print(f"Silhouette Score: {sil_kmedoids:.4f}")

Tampilkan output tersembunyi
```

#### ▼ Visualisasi Analisis Deskriptif

##### ▼ Ciri Ciri Masing Klaster

```
# Baca file hasil
kmeans_df = pd.read_excel("hasil_kmeans_k2_final.xlsx")
kmedoids_df = pd.read_excel("hasil_kmedoids_k2_final.xlsx")

# Hitung rata-rata
kmeans_avg = kmeans_df.groupby("Cluster")[["Price", "Customer Rating", "Number Sold", "Total Review"]].mean().round(2)
kmedoids_avg = kmedoids_df.groupby("Cluster")[["Price", "Customer Rating", "Number Sold", "Total Review"]].mean().round(2)

# Tambah label algoritma
kmeans_avg["Algoritma"] = "K-Means"
kmedoids_avg["Algoritma"] = "K-Medoids"

# Gabung dua dataframe
combined_avg = pd.concat([kmeans_avg, kmedoids_avg]).reset_index()

# Visualisasi per atribut
atribut = ["Price", "Customer Rating", "Number Sold", "Total Review"]

for col in atribut:
    plt.figure(figsize=(10, 6))
    sns.barplot(data=combined_avg, x="Cluster", y=col, hue="Algoritma", palette="pastel")
    plt.title(f"Rata-rata {col} per Klaster")
    plt.xlabel("Klaster")
    plt.ylabel(f"Rata-rata {col}")
    plt.grid(axis='y')
    plt.tight_layout()
    plt.show()
```

Tampilkan output tersembunyi

#### ▼ Distribusi Kategori Produk dalam Tiap Klaster

```
# Baca file hasil klasterisasi
df_kmeans = pd.read_excel("hasil_kmeans_k2_final.xlsx")
df_kmedoids = pd.read_excel("hasil_kmedoids_k2_final.xlsx")

# Ambil 8 Main Category teratas dari K-Means sebagai standar
top8 = df_kmeans['Main Category'].value_counts().nlargest(8).index.tolist()

df_kmeans = df_kmeans[df_kmeans['Main Category'].isin(top8)]
df_kmedoids = df_kmedoids[df_kmedoids['Main Category'].isin(top8)]

# Crosstab jumlah produk per klaster untuk tiap kategori
kmeans_counts = pd.crosstab(df_kmeans['Main Category'], df_kmeans['Cluster'])
kmedoids_counts = pd.crosstab(df_kmedoids['Main Category'], df_kmedoids['Cluster'])
```

```
# Urutkan agar sesuai
kmeans_counts = kmeans_counts.loc[top8]
kmmedoids_counts = kmmedoids_counts.loc[top8]

# Fungsi plotting dengan label
def plot_with_labels(ax, data, title):
    bars = data.plot(kind='bar', stacked=True, ax=ax, colormap='Set2')
    ax.set_title(title, fontsize=13)
    ax.set_xlabel('Kategori Produk')
    ax.set_ylabel('Jumlah Produk')
    ax.tick_params(axis='x', rotation=45)
    for container in ax.containers:
        for bar in container:
            height = bar.get_height()
            if height > 0:
                ax.text(bar.get_x() + bar.get_width()/2, bar.get_y() + height/2,
                        f'{int(height)}', ha='center', va='center', fontsize=9, color='white')

# Plot berdampingan
fig, axes = plt.subplots(1, 2, figsize=(16, 6), sharey=True)
plot_with_labels(axes[0], kmeans_counts, "Distribusi Produk per Kategori (K-Means)")
plot_with_labels(axes[1], kmmedoids_counts, "Distribusi Produk per Kategori (K-Medoids)")

plt.suptitle('Perbandingan Distribusi Produk Berdasarkan Klaster', fontsize=15)
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```

Tampilan output tersembunyi