Analisis Perbandingan *K-Means++*, *Mini Batch K-Means*, dan *Fuzzy C-Means* pada Segmentasi Pelanggan

I Putu Satria Dharma Wibawa[a1], Made Agung Raharja[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]wibawa.2308561045@student.unud.ac.id
[2]made.agung@unud.ac.id

Abstract

Customer segmentation is a crucial process for optimizing marketing strategies This study aims to implement and compare three clustering algorithms on customer transaction data using RFMT (Recency, Frequency, Monetary, and Tenure) features. The dataset, obtained from the UCI Machine Learning Repository, underwent several preprocessing stages, including data cleaning, feature extraction, outlier handling, and normalization. Optimal cluster numbers were determined using the elbow method and validated using silhouette score and davies-bouldin index. The results show that mini batch k-means outperforms the other algorithms with the highest silhouette score of 0.4011 and the lowest davies-bouldin index of 0.9521. K-means++ demonstrated better computation time but slightly lower clustering quality, while fuzzy c-means produced less distinct segmentation.

Keywords: *Customer Segmentation, RFMT, K-Means++, Mini Batch K-Means, Fuzzy C-Means, Clustering Evaluation, Transactional Data*

1. Pendahuluan

Dalam era persaingan bisnis yang semakin ketat, pemahaman mendalam terhadap karakteristik setiap pelanggan menjadi salah satu kunci utama dalam menyusun strategi pemasaran yang efektif. Setiap pelanggan memiliki preferensi yang berbeda, sehingga strategi pemasaran seragam tidak lagi efektif untuk menjangkau seluruh segmen pasar secara optimal. Oleh karena itu, segmentasi pelanggan hadir sebagai solusi untuk membagi pasar menjadi kelompok-kelompok pelanggan dengan kebutuhan atau perilaku yang serupa [1].

Untuk melakukan segmentasi pelanggan, pendekatan *data mining* khususnya teknik klasterisasi menjadi salah satu metode yang paling sering digunakan. Salah satu algoritma klasterisasi populer yang sering digunakan adalah *k-means*. Algoritma *k-means* bekerja dengan cara membagi data ke dalam beberapa klaster berdasarkan jarak tiap data ke pusat klaster (*centroid*) [2]. Meski efisien dan mudah diimplementasikan, algoritma *k-means* memiliki beberapa kelemahan, yaitu kepekaannya terhadap inisialisasi klaster awal dan ketidakmampuannya untuk menangani bentuk distribusi data yang sangat kompleks. Untuk menangani kelemahan tersebut, maka dikembangkan beberapa algoritma *k-means* lainnya dengan pendekatan yang berbeda-beda. Penelitian oleh Wicaksono dkk, (2021) menunjukkan bahwa algoritma *fuzzy c-means* dapat melakukan segmentasi pelanggan *e-commerce* secara efektif dengan nilai *silhouette score* 0.703 pada k = 2 [3]. Penelitian oleh Mulyadi dkk, (2024) mengungkapkan bahwa algoritma *mini batch k-means* mampu mengelompokkan data pendistribusian listrik secara efisien dengan *silhouette score* 0.625 untuk k = 2 [4]. Sementara itu, penelitian oleh Zhang dkk, (2022) membandingkan performa algoritma *k-means++* dengan *k-means* biasa pada data ritel dan menyimpulkan bahwa *k-means++* menghasilkan klaster yang lebih stabil dan terdistribusi merata [5].

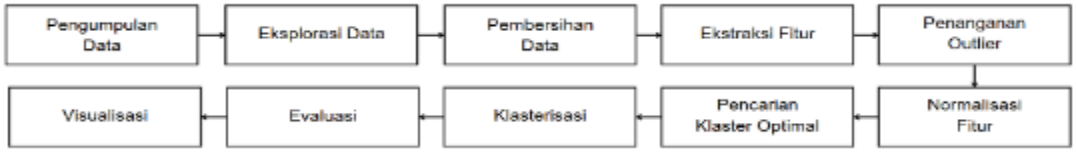
Selain itu, agar hasil dari klasterisasi dapat merepresentasikan perilaku dari setiap pelanggan

dengan akurat, fitur yang digunakan harus mampu menangkap dimensi-dimensi penting dari setiap perilaku pelanggan dengan baik. Salah satu model fitur yang paling sering digunakan dalam segmentasi pelanggan adalah RFM. Meskipun sudah sering digunakan, fitur RFM masih belum dapat menangkap seluruh dimensi penting dari perilaku setiap pelanggan dengan lengkap. Untuk mengatasi keterbatasan ini, dikembangkan beberapa model RFM lainnya dengan menambahkan atribut-atribut tambahan. Penelitian yang dilakukan oleh Surya dkk, (2022) membandingkan model RFM dan LRFM menggunakan algoritma *k-means* yang menunjukkan bahwa model RFM malah menghasilkan struktur klaster yang lebih baik dibandingkan dengan model LRFM dengan nilai *silhouette score* tertinggi sebesar 0.545 untuk k = 2 [6].

Berdasarkan penjelasan di atas, maka penelitian ini bertujuan untuk membandingkan performa dari algoritma *k-means*++, *mini batch k-means*, dan *fuzzy c-means* dalam melakukan segmentasi pelanggan menggunakan model fitur RFM dengan pendekatan berbeda, yaitu tambahan atribut T (*Tenure*). Perbandingan ini dilakukan untuk mengetahui algoritma mana yang paling efektif dalam membentuk klaster yang representatif terhadap perilaku pelanggan, baik dari segi kualitas klaster maupun efisiensi proses klasterisasi. Dengan menggunakan fitur RFMT, penelitian ini juga diharapkan dapat memberikan wawasan yang lebih mendalam mengenai karakteristik pelanggan melalui dimensi perilaku yang lebih lengkap.

2. Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen komparatif untuk membandingkan tiga algoritma klasterisasi, yaitu *k-means*++, *mini batch k-means*, dan *fuzzy c-means* dalam konteks segmentasi pelanggan berbasis RFMT. Proses penelitian ini terdiri dari sepuluh tahapan utama, yaitu pengumpulan data, eksplorasi data, pembersihan data, ekstraksi fitur, penanganan *outlier*, normalisasi fitur, pencarian klaster optimal, klasterisasi, evaluasi, dan visualisasi. Alur metode penelitian secara umum dapat dilihat pada Gambar 1.



Gambar 1. Metode Penelitian

2.1. Pengumpulan Data

Pengumpulan data adalah tahapan untuk memperoleh data-data relevan yang akan dianalisis. Data yang digunakan pada penelitian ini bersumber dari *UCI Machine Learning Repository*, yaitu dataset *Online Retail II*. Isi dari dataset tersebut memuat data transaksi pelanggan dari sebuah perusahaan *e-commerce* Inggris mulai dari Desember 2009 hingga Desember 2011. Dataset ini terdiri dari 1.067.371 entri data transaksi yang mencakup atribut seperti *InvoiceNo*, *StockCode*, *Description*, *Quantity*, *InvoiceDate*, *UnitPrice*, *CustomerID*, dan *Country* [7].

2.2. Eksplorasi Data

Eksplorasi data adalah tahapan untuk menelaah lebih dalam data yang telah dikumpulkan sebelum dilakukan proses analisis lebih lanjut. Pada tahapan ini akan dilakukan eksplorasi nilai pada tiap atribut secara menyeluruh untuk mengetahui sebaran dan anomali data. Selain itu, akan dilakukan eksplorasi terhadap tipe data dari setiap atribut untuk memastikan kesesuaian antara jenis data dengan konteks penggunaannya. Setelah itu, akan dilakukan pencocokan nilai tiap atribut dengan informasi yang tersedia di *UCI Machine Learning Repository* untuk memastikan kesesuaian format datanya.

2.3. Pembersihan Data

Pembersihan data adalah tahapan untuk meningkatkan kualitas dari data dengan cara menghilangkan elemen-elemen yang dapat mengganggu hasil analisis. Pada tahapan ini akan dilakukan penghapusan terhadap data-data yang tidak sesuai atau tidak relevan, berdasarkan temuan pada tahapan sebelumnya. Kemudian, akan dilakukan juga penanganan terhadap data yang memiliki *missing values* dengan cara menghapus baris data yang memiliki nilai kosong. Setelah itu, akan dilakukan proses pemfilteran untuk memastikan hanya data transaksi valid yang akan dianalisis lebih lanjut. Terakhir, akan dibuatkan atribut baru bernama *TotalSales* yang berfungsi untuk menyimpan total nilai pembelian dari masing-masing transaksi.

2.4. Ekstraksi Fitur

Ekstraksi fitur adalah tahapan untuk membentuk atribut-atribut baru yang merepresentasikan karakteristik penting dari data pelanggan. Dalam penelitian ini, fitur yang akan digunakan adalah RFMT (*Recency*, *Frequency*, *Monetary*, dan *Tenure*). *Recency* dihitung sebagai selisih hari antara tanggal referensi (tanggal terakhir dalam dataset) dengan tanggal transaksi terakhir yang dilakukan oleh masing-masing pelanggan. *Frequency* dihitung sebagai jumlah total transaksi yang dilakukan oleh masing-masing pelanggan selama periode waktu yang diamati. *Monetary* diperoleh dari penjumlahan seluruh nilai transaksi (*TotalSales*) untuk masing-masing pelanggan. *Tenure* dihitung sebagai selisih hari antara tanggal transaksi pertama pelanggan dengan tanggal referensi untuk menggambarkan berapa lama pelanggan telah aktif bertransaksi.

2.5. Penanganan *Outlier*

Penanganan *outlier* adalah tahapan untuk mengidentifikasi dan mengatasi data-data yang memiliki nilai ekstrem atau menyimpang secara signifikan dari mayoritas data lainnya. Dalam penelitian ini, penanganan *outlier* dilakukan terhadap fitur-fitur hasil ekstraksi RFMT dengan menggunakan metode *interquartile range* (IQR). Metode IQR membagi data-data fitur menjadi beberapa kuartil, di mana data yang berada di bawah kuartil pertama dikurangi 1.5 × IQR, atau data yang berada di atas kuartil ketiga ditambah 1.5 × IQR dianggap sebagai *outlier*.

2.6. Normalisasi Fitur

Normalisasi fitur adalah tahapan untuk menyamakan skala antar fitur, sehingga setiap fitur dapat memberikan kontribusi yang seimbang pada proses klasterisasi. Dalam penelitian ini, normalisasi akan dilakukan terhadap keempat fitur RFMT dengan menggunakan metode *z-score normalization*. Metode normalisasi ini mengubah data menjadi distribusi dengan rata-rata 0 dan standar deviasi 1 seperti yang ditunjukkan pada rumus (1).

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

2.7. Pencarian Klaster Optimal

Pencarian klaster optimal merupakan tahapan untuk menentukan jumlah klaster (k) yang paling sesuai, sehingga hasil dari klasterisasi yang dilakukan bisa lebih akurat. Dalam penelitian ini, pencarian klaster optimal dilakukan dengan menggunakan metode *elbow*, yang dilanjutkan dengan evaluasi menggunakan *silhouette score* dan *davies-bouldin index* apabila diperlukan. Metode *elbow* dilakukan dengan menjalankan ketiga algoritma pada berbagai nilai k, untuk menghitung nilai *inertia* pada setiap jumlah klaster. Titik "*elbow*" ditentukan secara manual melalui pengamatan visual, yaitu titik di mana penurunan *inertia* mulai melambat secara signifikan. Titik tersebut dianggap sebagai kandidat jumlah klaster optimal karena menandai batas antara efisiensi pembentukan klaster dan kompleksitas model. Apabila dari grafik *inertia* ditemukan lebih dari satu kemungkinan titik *elbow*, maka masing-masing kandidat jumlah klaster akan dibandingkan menggunakan *silhouette score*. Jika nilai *silhouette score* untuk dua kandidat klaster masih terlalu berdekatan atau belum meyakinkan, maka akan dilakukan perbandingan tambahan menggunakan *davies-bouldin index*.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

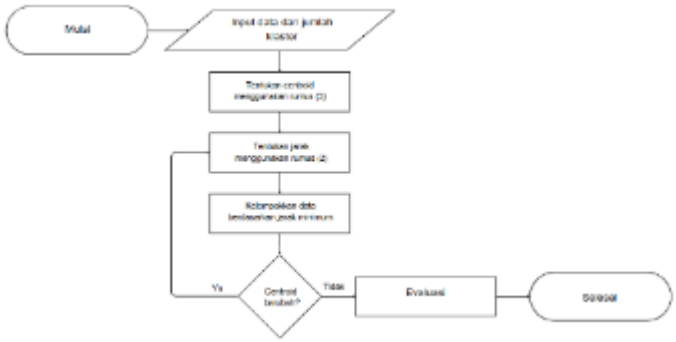
2.8. Klasterisasi

Klasterisasi merupakan tahapan inti dalam penelitian ini, di mana data pelanggan yang telah dinormalisasi dan jumlah klasternya telah ditentukan akan dikelompokkan menggunakan tiga algoritma yang berbeda, yaitu *k-means++*, *mini batch k-means*, dan *fuzzy c-means*. Algoritma *k-means++* memilih *centroid* awal menggunakan rumus jarak *euclidean* (2) dan rumus probabilistik (3), berbeda dengan algoritma *k-means* biasa yang memilih *centroid* awal secara acak. Hal ini menyebabkan pemilihan *centroid* awal menjadi lebih tersebar dan strategis, sehingga hasil dari klasterisasi bisa lebih akurat dan stabil [8].

$$D(x)^2 = minc \in C \ \| x - c \|^2 \quad (2)$$

$$P(x) = \frac{D(x)^2}{\sum x' \in X \ D(x')^2} \quad (3)$$

Flowchart dari algoritma *k-means++* dapat dilihat pada Gambar 2.

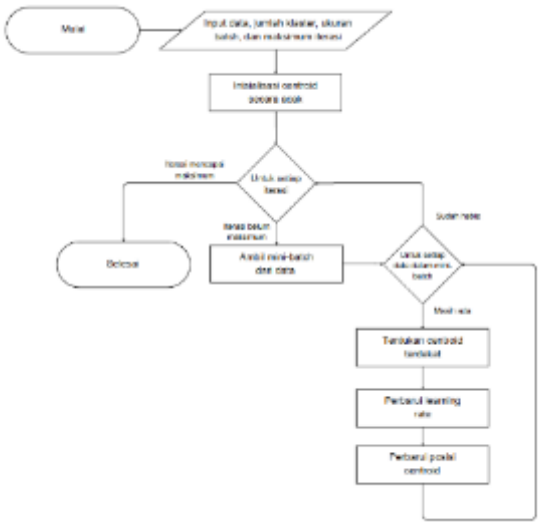


Gambar 2. *Flowchart* Algoritma *K-Means++*

Cara kerja algoritma *k-means++* dimulai dengan memasukkan data dan jumlah klasternya. Setelah itu, tentukan *centroid* awal menggunakan rumus (3), dan tentukan jarak antar data dengan *centroid* menggunakan rumus (2). Selanjutnya, kelompokkan seluruh data berdasarkan jaraknya terhadap *centroid*. Apabila *centroid* berubah, ulangi proses untuk menentukan *centroid* selanjutnya. Apabila *centroid* tidak berubah, lakukan evaluasi terhadap klaster yang sudah dihasilkan.

Mini batch k-means merupakan varian dari algoritma *k-means* yang bekerja dengan menggunakan subset data acak (*mini-batch*) pada setiap iterasi, yang secara signifikan mempercepat proses klasterisasi [9]. *Flowchart* dari algoritma *mini batch k-means* dapat dilihat pada Gambar 3.

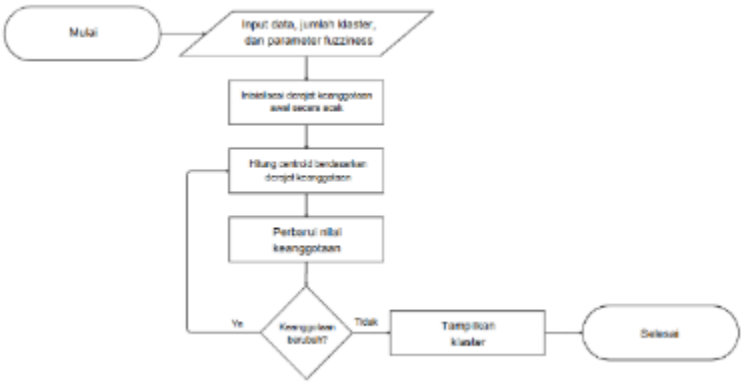
p-ISSN: 2986-3929
e-ISSN: 3032-1948



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

Cara kerja algoritma *mini batch k-means* dimulai dengan memasukkan data, jumlah klaster, ukuran *batch*, dan jumlah maksimum iterasinya. Setelah itu, dilakukan inisialisasi *centroid* awal secara acak. Kemudian, untuk setiap iterasi ambil subset dari data sesuai dengan ukuran *batch*-nya. Untuk setiap data yang ada di dalam subset, tentukan jarak *centroid* terdekat, perbarui *learning rate*, dan perbarui posisi *centroid*-nya. Apabila data dalam subset sudah semuanya diproses, lanjut ke iterasi berikutnya untuk mengambil subset data yang baru. Apabila seluruh data telah diproses atau telah mencapai maksimum iterasi, proses klasterisasi diakhiri.

Fuzzy c-means menerapkan pendekatan *soft clustering*, di mana setiap data tidak hanya dimiliki oleh satu klaster secara mutlak, melainkan memiliki derajat keanggotaan terhadap setiap klaster [10]. Flowchart dari algoritma fuzzy c-means dapat dilihat pada Gambar 4.



Gambar 4. *Flowchart* Algoritma *Fuzzy C-Means*

Cara kerja algoritma *fuzzy c-means* dimulai dengan memasukkan data, jumlah klaster, dan parameter *fuzziness*-nya. Kemudian, dilakukan inisialisasi awal derajat keanggotaan untuk setiap data secara acak. Selanjutnya, berdasarkan derajat keanggotaan tersebut, hitung posisi *centroid*-nya. Setelah itu, perbarui kembali derajat keanggotaan setiap data berdasarkan posisi *centroid*. Apabila derajat keanggotaannya berubah, ulang kembali proses tersebut. Apabila derajat keanggotaannya tidak berubah, tampilkan hasil klasterisasinya.

2.9. Evaluasi

Evaluasi merupakan tahapan untuk mengukur kualitas hasil klasterisasi yang dihasilkan oleh masing-masing algoritma. Dalam penelitian ini, evaluasi dilakukan dengan menggunakan tiga metrik utama, yaitu *silhouette score*, *davies-bouldin index*, dan *execution time*. *Silhouette score* digunakan untuk menilai seberapa baik objek dikelompokkan dalam klaster. *Davies-bouldin index* digunakan untuk mengukur kepadatan dan pemisahan antar klaster. *Execution time* digunakan untuk mengukur durasi waktu komputasi yang dibutuhkan oleh masing-masing algoritma dalam proses klasterisasi.

2.10. Visualisasi

Visualisasi merupakan tahapan akhir dalam proses klasterisasi yang bertujuan untuk menyajikan hasil segmentasi pelanggan secara visual agar lebih mudah dipahami. Visualisasi hasil klasterisasi dilakukan melalui dua pendekatan utama. Pendekatan pertama menggunakan grafik tiga dimensi (3D) yang menampilkan distribusi data dalam ruang fitur *Recency*, *Frequency*, dan *Monetary* untuk masing-masing algoritma. Pendekatan kedua melibatkan pembuatan *heatmap* statistik deskriptif terhadap nilai *Tenure* pada tiap klaster. Statistik yang digunakan mencakup nilai *mean*, median, minimum, dan maksimum, yang dihitung berdasarkan pembagian klaster dari masing-masing algoritma.

3. Hasil dan Diskusi

Hasil dan diskusi berisi penyampaian temuan penelitian yang diperoleh dari proses analisis data, Dalam bagian ini, akan dipaparkan hasil eksperimen yang telah diperoleh secara sistematis, baik dalam bentuk tabel, grafik, maupun uraian deskriptif.

3.1. Pengolahan Data

Tabel 1. merupakan dataset bersih yang telah melalui proses eksplorasi dan pembersihan data. Setelah diolah, jumlah data transaksi yang relevan dan layak untuk diklasterisasi tersisa sebanyak 802.649 entri, atau sekitar 75% dari total data awal.

Tabel 1. Dataset Bersih

	InvoiceNo	Stock Code	Description	Quantity	InvoiceDate	Unit Price	CustomerID	Country	Total Sales
1	489434	85048	15CM CHR…	12	2009-12-01 07:45:00	6.95	13085	United Kingdom	83.40
2	489434	79323P	PINK CHE...	12	2009-12-01 07:45:00	6.75	13085	United Kingdom	81.00
3	489434	79323W	WHITE CH...	12	2009-12-01 07:45:00	6.75	13085	United Kingdom	81.00
...	...	...	...	...	...	...	...	...	
802649	581587	22138	BAKING SE…	3	2011-12-09 12:50:00	4.95	12680	France	14.85

3.2. Ekstraksi Fitur dan Penanganan *Outlier*

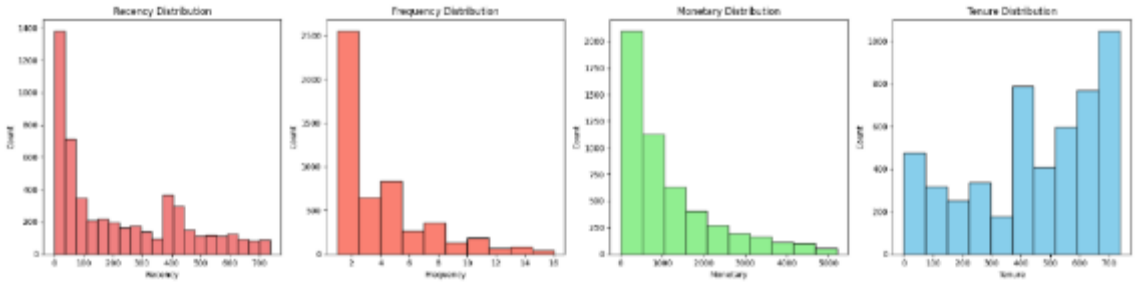
Tabel 2 merupakan data fitur RFMT dari total 5.852 pelanggan. Melalui agregasi, diperoleh informasi berupa tanggal transaksi terakhir, total nilai transaksi (*Monetary*), jumlah transaksi unik (*Frequency*), dan tanggal transaksi pertama. Selanjutnya, dihitung fitur *Recency* sebagai selisih hari antara tanggal referensi dengan tanggal transaksi terakhir, serta fitur *Tenure* sebagai selisih hari antara tanggal referensi dengan tanggal transaksi pertama. Hasil tahapan ekstraksi fitur dapat dilihat pada Tabel 2.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Tabel 2. Fitur RFMT

	CustomerID	Recency	Frequency	Monetary	Tenure
1	12346	325	3	77352.96	646
2	12347	1	8	5633.32	403
3	12348	74	5	1658.40	437
…	…	…	…	…	…
5852	18287	42	7	4132.99	571

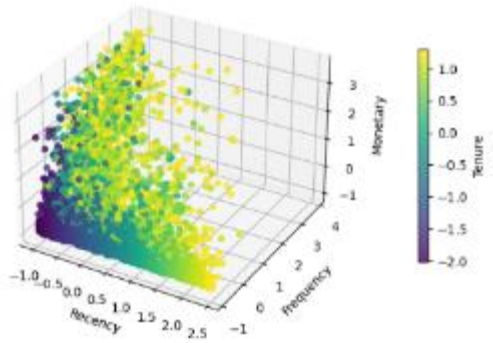
Setelah dilakukan ekstraksi fitur, perlu dilakukan penanganan *outlier* dilakukan untuk mengidentifikasi dan mengatasi data-data fitur yang memiliki nilai ekstrem. Sebaran data untuk setiap fitur setelah melalui tahapan penanganan *outlier* dapat dilihat pada Gambar 5.



Gambar 5. Sebaran Data Fitur Setelah Penanganan *Outlier*

3.3. Normalisasi Fitur

Tahapan normalisasi fitur dilakukan untuk menyamakan skala antar fitur sehingga tidak ada fitur yang mendominasi dalam proses analisis. Normalisasi dilakukan menggunakan metode *z-score normalization*, yang juga dikenal sebagai standardisasi. Hasil normalisasi fitur dapat dilihat pada Gambar 6.



Gambar 6. Sebaran Data Pelanggan Berdasarkan Fitur RFMT Setelah Nomalisasi

Berdasarkan Gambar 6, tahapan normalisasi fitur telah berhasil dilakukan dengan baik. Hasil normalisasi menunjukkan bahwa rentang nilai dari setiap fitur sebagian besar berada dalam kisaran -1 hingga 3.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

3.4. Pencarian Klaster Optimal

Tahapan pencarian klaster optimal dilakukan untuk menentukan jumlah klaster (k) yang paling sesuai, sehingga hasil dari klasterisasi bisa lebih akurat. Dalam penelitian ini, pencarian klaster dilakukan dengan menggunakan metode *elbow*, yang dilanjutkan dengan evaluasi menggunakan *silhouette score* dan *davies-bouldin index* apabila diperlukan. Perbandingan ketiga metode tersebut dapat dilihat pada Gambar 7.

Gambar 7. Pencarian Klaster Optimal

Berdasarkan Gambar 7., didapatkan dua kandidat klaster optimal, yaitu 4 dan 5. Berdasarkan *silhouette score* dan *davies-bouldin index* kedua kandidat tersebut, disimpulkan bahwa klaster yang paling optimal adalah 4.

3.5. Klasterisasi

Tahapan klasterisasi dilakukan dengan menggunakan algoritma *k-means++*, *mini batch k-means*, dan *fuzzy c-means* berdasarkan jumlah klaster yang telah ditentukan. Performa dari masing-masing algoritma ketika melakukan klasterisasi dapat dilihat pada Tabel 3.

Tabel 3. Performa Ketiga Algoritma Klasterisasi

Algoritma	Silhouette Score	Davies-Bouldin Index	Execution Time
K-Means++	0.3950	0.9573	0.0158s
Mini Batch K-Means	0.4011	0.9521	0.3864s
Fuzzy C-Means	0.3936	0.9577	0.1131s

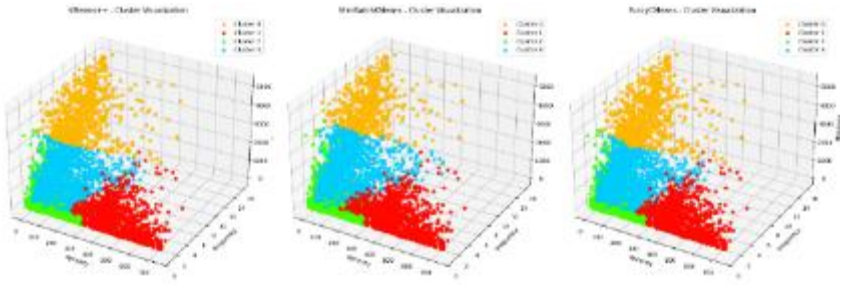
3.6. Evaluasi

Berdasarkan performa ketiga algoritma klasterisasi yang dapat dilihat pada Tabel 3., *mini batch k-means* menunjukkan performa terbaik dengan nilai *silhouette score* tertinggi sebesar 0.4011 dan *davies-bouldin index* terendah sebesar 0.9521. Meskipun secara teori algoritma ini dirancang untuk efisiensi waktu komputasi dengan kompromi terhadap akurasi, pada kasus ini justru menghasilkan kualitas segmentasi tertinggi dikarenakan mekanisme *mini batch* yang lebih tahan terhadap *noise* dan fluktuasi lokal pada data berskala besar. Di sisi lain, *k-means++* yang secara teoritis memiliki akurasi tinggi berkat inisialisasi *centroid* yang lebih baik, justru menghasilkan performa sedikit lebih rendah dengan *silhouette score* sebesar 0.3951 dan *davies-bouldin index* sebesar 0.9574. Namun, dari sisi efisiensi waktu, *k-means++* unggul dengan waktu eksekusi tercepat, yaitu hanya 0.0158 detik. *Fuzzy c-means*, menghasilkan performa terendah dengan *silhouette score* sebesar 0.3937 dan *davies-bouldin index* sebesar 0.9578. Hal ini menunjukkan bahwa pembentukan klaster oleh f*uzzy c-means* kurang efektif dalam memisahkan pelanggan ke dalam kelompok yang berbeda secara tegas. Pendekatan ini justru tidak memberikan keuntungan signifikan karena fitur RFMT cenderung memiliki struktur yang jelas dan segmentasi yang eksplisit.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

3.7. Visualisasi

Visualisasi hasil klasterisasi dilakukan melalui dua pendekatan utama. Pendekatan pertama menggunakan grafik tiga dimensi (3D) yang menampilkan distribusi data dalam ruang fitur *Recency*, *Frequency*, dan *Monetary* untuk masing-masing algoritma. Pendekatan kedua melibatkan pembuatan *heatmap* statistik deskriptif terhadap nilai *Tenure* pada tiap klaster. Visualisasi dapat dilihat pada Gambar 8. dan Gambar 9.



Gambar 8. Visualisasi Klasterisasi Berdasarkan RFM pada Ketiga Algoritma

Berdasarkan Gambar 8. Fitur RFM pada data pelanggan telah diklasterisasi menjadi 4 klaster. Warna kuning melambangkan klaster 0, warna merah melambangkan klaster 1, warna hijau melambangkan klaster 2, dan warna biru melambangkan klaster 3.

Gambar 9. Visualisasi Klasterisasi Berdasarkan Tenure pada Ketiga Algoritma

Berdasarkan Gambar 9. Fitur T pada data pelanggan telah diklasterisasi menjadi 4 klaster. Ketiga algoritma memperlihatkan pola statistik yang cukup mirip, menunjukkan konsistensi dalam segmentasi.

4. Kesimpulan

Penelitian ini telah berhasil melakukan segmentasi pelanggan menggunakan algoritma *k-means++*, *mini batch k-means*, dan *fuzzy c-means* dengan memanfaatkan fitur RFMT. Setelah melalui tahapan pembersihan, ekstraksi fitur, penanganan *outlier*, normalisasi, dan pencarian jumlah klaster optimal, ditemukan bahwa algoritma *mini batch k-means* memberikan performa klasterisasi terbaik dengan *silhouette score* sebesar 0.4011 dan *davies-bouldin index* sebesar 0.9521. Sementara itu, algoritma *k-means++* mencatat waktu komputasi tercepat dalam hanya 0.0158 detik. Sedangkan, algoritma *fuzzy c-means* menghasilkan segmentasi yang kurang tegas dan tidak menunjukkan keunggulan signifikan dalam kasus ini. Secara keseluruhan, algoritma *mini batch k-means* paling sesuai digunakan untuk segmentasi pelanggan dalam dataset berskala besar dengan karakteristik yang cukup kompleks.

Daftar Pustaka

[1] P. Kotler and G. Armstrong, *Principles of Marketing*, 18th ed. Pearson Education, 2021.
[2] J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate

Observations," *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.
[3] A. Wicaksono, R. Hidayat, and F. A. Permadi, "Segmentasi Pelanggan Menggunakan Fuzzy C-Means Clustering Berdasarkan RFM Model pada E-Commerce," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 3, pp. 233–239, 2021.
[4] S. Mulyadi, F. Insani, S. Agustian, and L. Afriyanti, "Pengelompokan Data Pendistribusian Listrik Menggunakan Algoritma Mini Batch K-Means Clustering: Grouping Electricity Distribution Data Using The Mini Batch K-Means Clustering Algorithm", *MALCOM*, vol. 4, no. 3, pp. 1051-1062, Jun. 2024.
[5] Y. Zhang, J. Zhou, and W. Chen, "Comparative Analysis of K-Means and K-Means++ Clustering on Retail Customer Data," *Journal of Data Science and Analytics*, vol. 3, no. 2, pp. 55–66, 2022.
[6] I. K. A. Surya, M. A. Raharja, I. K. A. Mogi, A. Muliantara, I. G. A. Wibawa, and I. G. N. A. C. Putra, "Pengelompokan Pelanggan Toko Kerajinan Menggunakan K-Means dengan Model RFM dan LRFM," *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, vol. 11, no. 1, p. 22, 2022, doi: https://doi.org/10.24843/jlk.2022.v11.i01.p03.
[7] D. Chen. "Online Retail II," *UCI Machine Learning Repository*, 2012. https://doi.org/10.24432/C5CG6D
[8] N. Nugroho and F. D. Adhinata, "Penggunaan Metode K-Means dan K-Means++ Sebagai Clustering Data Covid-19 di Pulau Jawa," *Teknika*, vol. 11, no. 3, pp. 170–179, Oct. 2022, doi: https://doi.org/10.34148/teknika.v11i3.502.
[9] J. Bejar, K-Means vs Mini Batch K-Means: A Comparison. *UPCommons*, 2013.
[10] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. Boston, *MA: Springer*, 1981.