

Segmentasi Pemegang Kartu Kredit melalui Klasterisasi K-Means Menggunakan Bahasa Pemrograman Python

Agil Haubi Zikri¹, Rita Dewi Risanty²

^{1,2}Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jakarta, Jl Cempaka Putih Tengah 27 Jakarta, 10510

rita.dewi@umj.ac.id²

ABSTRAK

Dalam perkembangan ekonomi yang cepat, penggunaan kartu kredit telah menjadi sangat penting. Kartu kredit, sebagai instrumen pembayaran yang diterbitkan oleh bank, memberikan kemampuan kredit kepada pemiliknya untuk melakukan pembelian barang atau layanan. Namun, penggunaannya mengalami penurunan yang signifikan akibat perubahan pola hidup dan dampak pandemi Covid-19. Survei perbankan menunjukkan tren penurunan dalam transaksi kartu kredit, yang dipicu oleh pengurangan pendapatan masyarakat selama pandemi. Untuk mengatasi tantangan ini, bank perlu mengembangkan strategi bisnis yang efektif. Salah satu pendekatan yang dapat digunakan adalah klasterisasi dengan algoritma *K-Means*. Algoritma *K-Means* merupakan teknik klasterisasi yang efektif digunakan untuk mengelompokkan data dengan tujuan memaksimalkan kesamaan antara variabel sambil tetap mempertahankan perbedaan karakteristiknya. Dengan mengimplementasikan algoritma ini dalam bahasa pemrograman *Python*, bank dapat memperoleh pemahaman yang lebih mendalam tentang preferensi dan kebutuhan dari setiap segmen pemegang kartu kredit. Demikian, algoritma *K-Means* dapat membantu dalam pengembangan strategi bisnis yang lebih efektif untuk meningkatkan penggunaan kartu kredit, serta mendapatkan nasabah baru. Selain itu, algoritma ini juga bermanfaat dalam mengatasi dampak penurunan penggunaan kartu kredit yang disebabkan oleh perubahan pola hidup masyarakat dan pandemi Covid-19. Hasil dari penelitian ini adalah pengelompokan data pengguna kartu kredit menjadi tiga kelompok yang dapat digunakan sebagai panduan dalam merumuskan strategi bisnis perbankan.

Kata kunci: Algoritma *K-Means*, Kartu kredit, Klasterisasi, Pandemi Covid-19, *Python*.

ABSTRACT

In the rapid economic development, the use of credit cards has become increasingly crucial. Credit cards, as payment instruments issued by banks, provide credit capabilities to their owners to make purchases of goods or services. However, their usage has experienced a significant decline due to changes in lifestyle patterns and the impact of the Covid-19 pandemic. Banking surveys have indicated a decreasing trend in credit card transactions, triggered by the reduction in people's income during the pandemic. To address this challenge, banks need to develop effective business strategies. One approach that can be employed is clustering using the K-Means algorithm. The K-Means algorithm is an effective clustering technique used to group data with the goal of maximizing the similarity between variables while preserving their distinct characteristics. By implementing this algorithm in the Python programming language, banks can gain a deeper understanding of the preferences and needs of each credit cardholder segment. Thus, the K-Means algorithm can assist in the development of more effective business strategies to increase credit card usage and acquire new customers. Additionally, this algorithm is beneficial in mitigating the decline in credit card usage caused by changes in societal lifestyles

and the Covid-19 pandemic. The result of this research is the categorization of credit card user data into three groups that can be used as a guide in formulating banking business strategies.

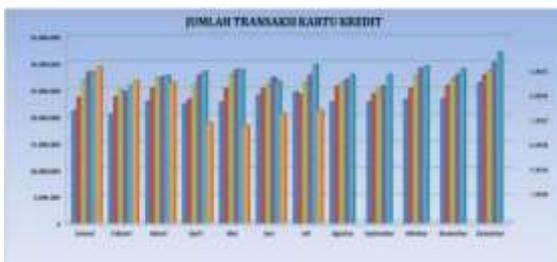
Keywords: Algorithm K-Means, Clustering, Credit Cards, Covid-19 Pandemic, Python.

1. PENDAHULUAN

Dalam perkembangan ekonomi yang cepat, transaksi keuangan menjadi sangat penting dan terjadi setiap waktu. Uang tunai sudah tidak cukup fleksibel, terutama dengan adanya belanja dan transaksi *online* yang membuat segalanya lebih praktis dan cepat (Raharjo, 2021). Karena itu, muncul berbagai cara pembayaran tanpa tunai, salah satunya adalah penggunaan kartu kredit.

Kartu kredit merupakan suatu instrumen pembayaran yang diterbitkan oleh bank tertentu, berbentuk plastik, dan berperan sebagai alat pembayaran kredit yang digunakan oleh pemilik kartu sesuai dengan nama yang tertera di kartu kredit tersebut. Fungsinya adalah untuk melakukan pembelian barang atau layanan dengan kemampuan kredit yang telah ditetapkan oleh penerbit kartu. Selain itu, kartu kredit juga dapat digunakan secara independen untuk melakukan penarikan uang tunai melalui mesin ATM yang sesuai dengan batas kredit yang telah ditentukan (Kurniawan & Barokah, 2020).

Meskipun kartu kredit telah lama menjadi bagian integral dari aktivitas transaksi keuangan, saat ini kita menyaksikan penurunan signifikan dalam penggunaannya. Penurunan ini dipengaruhi oleh berbagai faktor, termasuk perubahan pola hidup masyarakat dan dampak pandemi Covid-19 yang terus berjalan seiring dengan perubahan pesat dalam teknologi dan kondisi ekonomi saat ini (Noviyanto & Sukiman, 2022).



Gambar 1. Jumlah Transaksi Kartu Kredit (Dudiyanto, 2021)

Survei triwulan III perbankan yang dilakukan oleh Asosiasi Kartu Kredit Indonesia (AKKI) menyatakan terkait jumlah nilai transaksi kartu kredit dari tahun 2015-2020 pada lebih dari 40 bank umum dengan pangsa kredit sekitar 80% dari total kredit, seperti hasil yang terlihat dalam gambar 1, menunjukkan bahwa hingga Maret 2020 transaksi kartu kredit meningkat 7-8%. Namun, angkanya menurun setelah terjadinya pandemi Covid-19 dari Maret hingga Juni. Transaksi sepanjang tahun 2020 terdata mencapai Rp 349,211 miliar atau tumbuh 3,26% dan terus meningkat tapi lambat kalah jauh dibandingkan dengan periode sebelum pandemi Covid-19 yaitu tahun 2019 yang dilihat dengan periode yang sama memiliki nominal Rp 342,68 triliun atau tumbuh 8,9% (Dudiyanto, 2021).

Menurut Meki Dudiyanto saat melakukan penelitian pada hasil survey terkait mengapa jumlah nilai transaksi kartu kredit mengalami penurunan dikarenakan saat terjadinya pandemi Covid-19 pendapatan atau penghasilan berkurang menjadi faktor utama sehingga menyebabkan pembatasan pada pengeluaran untuk membeli barang. Dengan adanya kondisi ini pihak bank perlu membuat strategi bisnis untuk meningkatkan transaksi kartu kredit serta mendapatkan nasabah baru (Dudiyanto, 2021).

Oleh karena itu, diperlukan pendekatan klusterisasi menggunakan algoritma *K-Means* sebagai pemecahan masalah. *K-Means* adalah sebuah algoritma clustering yang sederhana namun efektif dalam membagi data menjadi kelompok-kelompok berdasarkan kesamaan fitur (Dinata et al., 2020).

Dengan menerapkan pendekatan klusterisasi menggunakan algoritma *K-Means* pada data transaksi kartu kredit yang telah tersedia, diharapkan penelitian ini dapat membantu bank-bank dalam mengidentifikasi segmentasi pelanggan serta bagaimana pendekatan untuk mengembangkan strategi bisnis sesuai dengan segmentasi yang telah

dilakukan, serta meningkatkan transaksi kartu kredit dan mendapatkan nasabah baru. Dengan demikian, diharapkan dapat mengatasi dampak penurunan transaksi kartu kredit yang disebabkan oleh perubahan pola hidup masyarakat dan pandemi Covid-19, serta mendukung pertumbuhan ekonomi yang berkelanjutan.

2. LITERATUR REVIEW

Pada penelitian sebelumnya dalam analisis kartu kredit ada beberapa topik penelitian salah satu topik penelitian yang telah dieksplorasi adalah deteksi penipuan dalam transaksi kartu kredit, juga dikenal sebagai *fraud detection*. Penelitian ini mengadopsi pendekatan pengelompokan (*clustering*) dengan menggunakan metode *K-Means*. Hasil dari penelitian ini berhasil mengelompokkan transaksi kartu kredit ke dalam empat kategori, yaitu rendah (*low*), tinggi (*high*), berisiko (*risky*), dan tinggi berisiko (*high risky*) (Vaishali, 2014).

Pada penelitian terdahulu yang dilakukan untuk analisis perilaku untuk memprediksi penggunaan kartu kredit oleh pemilik kartu. Faktor-faktor seperti pekerjaan dan kebiasaan penggunaan kartu kredit dipertimbangkan dalam penelitian tersebut. Penelitian ini berfokus pada hubungan antara pendapatan pelanggan dan penggunaan total batas kredit yang tersedia bagi pelanggan (Dewri et al., 2016).

Kemudian pada topik penelitian lain yang membahas tentang analisis kartu kredit adalah memprediksi status kredit macet dengan menggunakan metode *K-Means* untuk membantu PT Esta Dana Ventura dalam mengevaluasi potensi terjadinya masalah kredit, khususnya dalam konteks permohonan pemberian kartu kredit dari calon nasabah repeat order. Hasil penelitian ini menghasilkan data pengelompokan menjadi 3 kriteria, yaitu nasabah lancar (C0), nasabah sangat lancar (C1), dan nasabah macet (C2) (Fitriani et al., 2023).

3. METODOLOGI PENELITIAN

Algoritma K-Means

Algoritma *K-Means* merupakan salah satu algoritma klastering yang umum digunakan untuk mengelompokkan data

berdasarkan kesamaan karakteristik. Pengelompokan data ini menghasilkan kelompok-kelompok yang disebut klaster. Salah satu keuntungan dari algoritma *K-Means* adalah kemudahan dalam implementasinya serta kompleksitas waktu dan ruang yang relatif rendah. Hal ini membuatnya menjadi algoritma yang efisien dalam perhitungan dan mampu menghasilkan hasil yang memuaskan jika klaster-klaster yang dihasilkan berbentuk kompak dan berbentuk hipersferis, serta dapat memisahkan fitur-fitur dalam ruangnya dengan baik. Algoritma *K-Means* berfungsi dengan membagi data ke dalam sejumlah 'k' klaster yang sudah ditentukan sebelumnya (Luthfi & Wijayanto, 2021).

Langkah-langkah dasar dalam algoritma *K-Means* melibatkan inisialisasi *centroid* awal untuk setiap klaster, kemudian melakukan iterasi untuk memperbarui posisi *centroid* dan mengelompokkan data hingga konvergensi (Ahmed et al., 2020).

1. Tentukan nilai titik k klaster sesuai dengan data yang dimiliki.
2. Pilih titik-titik atau sampel yang menjadi anggota klaster secara acak.
3. Tentukan nilai *centroid* atau titik tengah dari klaster dengan rumus:

$$D(i, j) = \sqrt{(x_{1i} - x_{1j})^2 + (x_{2i} - x_{2j})^2} \quad (1)$$

Dimana X_{ik} = sampel yang ada dalam klaster C_k

4. Tentukan posisi *centroid* baru
5. Kembali ke Langkah 3, apabila masih ada data yang berpindah kelompok atau apabila ada perubahan nilai *centroid* di atas nilai ambang yang ditentukan, atau apabila perubahan nilai pada fungsi objektif yang dibunakan masih di atas nilai ambang yang ditentukan.

Contoh Penerapan K-Means Clustering

Berikut diberikan contoh perhitungan dari algoritma K-Means untuk mengetahui segmentasi pada data pemegang kartu kredit:

1. Tentukan nilai titik k klaster sesuai dengan data yang dimiliki.

Untuk memudahkan penjelasannya, dari 18 variabel ini dipilih 5 variabel yaitu ['balance', 'purchases', 'cash_advance', 'credit_limit', 'payment'].

Tabel 1. Sampel Data

C.I	B	Pur	C.A	C.L	Pay
C11001	727	503	164	2375	700
C10101	1900	2430	371	2834	5301
C10211	5001	901	2999	8761	7501
C10315	521	3210	455	7654	831
C10254	7231	723	432	2328	2371

Keterangan:

1. C.I = Cust_Id
2. B = Balance
3. Pur = Purchases
4. C.A = Credit_Advance
5. C.L = Credit_Limit
6. Pay = Payments

2. Pilih titik-titik atau sampel yang menjadi anggota klaster secara acak.

Dalam penelitian ini ada 3 cluster yang penjelasannya berada sub-bab 3.1 Analisis Data. Langkah pertama yaitu menentukan titik pusat awal dari setiap cluster secara acak sebagai berikut.

Tabel 2. Titik Pusat Cluster Awal

Cluster	B	Pur	C.A	C.L	Pay
Cluster 1	726	400	380	2400	800
Cluster 2	1150	2200	170	7300	2400
Cluster 3	3890	580	3000	6500	2350

3. Tentukan nilai centroid atau titik tengah dari klaster.

Hitung jarak data terdekat dengan titik pusat awal pada data ke-1 terhadap pusat cluster.

Tabel 3. Data ke-1 dan 2

C.I	B	Pur	C.A	C.L	Pay
C11001	727	503	164	2375	700
C10101	1900	2430	371	2834	5301

Cluster1

$$= \sqrt{(727 - 726)^2 + (503 - 400)^2 + (164 - 380)^2 + (2375 - 2400)^2 + (700 - 800)^2}$$

$$= 260,74$$

Cluster2

$$= \sqrt{(727 - 1150)^2 + (503 - 2200)^2 + (164 - 170)^2 + (2375 - 7300)^2 + (700 - 2400)^2}$$

$$= 5213,33$$

Cluster3

$$= \sqrt{(727 - 3890)^2 + (503 - 580)^2 + (164 - 3000)^2 + (2375 - 6500)^2 + (700 - 2350)^2}$$

$$= 5929,57$$

Jarak dari data ke-2 terhadap pusat cluster 1

Cluster1

$$= \sqrt{(1900 - 726)^2 + (2430 - 400)^2 + (371 - 380)^2 + (2834 - 2400)^2 + (5301 - 800)^2}$$

$$= 4957,34$$

Cluster2

$$= \sqrt{(1900 - 1150)^2 + (2430 - 2200)^2 + (371 - 170)^2 + (2834 - 7300)^2 + (5301 - 2400)^2}$$

$$= 5311,79$$

Cluster3

$$= \sqrt{(1900 - 3890)^2 + (2430 - 580)^2 + (371 - 3000)^2 + (2834 - 6500)^2 + (5301 - 2350)^2}$$

$$= 5876,72$$

Hasil untuk perhitungan selengkapnya untuk iterasi 1 diatas dapat dilihat pada table 4.

Tabel 4. Hasil perhitungan pada iterasi 1

C. I	B	Pu r	C. A	C. L	Pa y	C 1	C 2	C 3	H s l
C 11001	727	503	164	2375	700	260,74	5213,33	5929,57	C 1

C 10 10 1	19 00	24 30	37 1	28 34	53 01	49 57 .3 4	53 11 .7 9	58 76 .7 2	C 1
C 10 21 1	50 01	90 1	29 99	87 61	75 01	10 58 9. 66	72 03 .0 9	57 36 .2 5	C 3
C 10 31 5	52 1	32 10	45 5	76 54	83 1	59 70 .3 6	19 57 .5 9	53 42 .1 4	C 2
C 10 25 4	72 31	72 3	43 2	23 28	23 71	67 03 .2 6	78 59 .6 2	59 41 .7 8	C 3

4. Tentukan posisi centroid baru

Setelah semua data ditempatkan ke dalam *cluster* terdekat, kemudian hitung Kembali pusat *cluster* yang baru berdasarkan rata-rata anggota yang ada pada *cluster* tersebut.

Pembagian ulang *centroid* baru dihitung dengan rumus sebagai berikut:

$$C = \frac{\sum m}{n} \tag{2}$$

Berdasarkan tabel 4, titik pusat *cluster* baru untuk parameter awal (C1) memiliki 5 anggota:

$$C1 = (260,74 + 4957,34 + 10589,66 + 5970,36 + 6703,26) / 5 = 6156,88$$

Cluster ke-2 (C2) memiliki 5 anggota:

$$C2 = (5213,33 + 5311,79 + 7203,09 + 1957,59 + 7859,62) / 5 = 5509,48$$

Cluster ke-3 (C3) memiliki 5 anggota:

$$C3 = (5929,57 + 5876,72 + 5736,25 + 5342,14 + 5941,78) / 5 = 5765,89$$

Tabel 5. Titik Pusat Baru Cluster

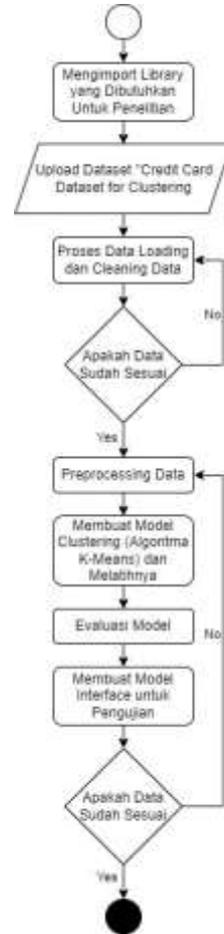
Cluster 1	6159,88	400	380	2400	800
Cluster 2	5509,48	2200	170	7300	2400
Cluster 3	5765,89	580	3000	6500	2350

5. Setelah didapatkan titik pusat yang baru dari setiap *cluster*, lakukan kembali dari langkah ketiga hingga titik pusat dari setiap *cluster* tidak berubah lagi dan tidak

ada lagi data yang berpindah dari satu *cluster* yang lain.

Flowchart

Penelitian ini mengadopsi pendekatan prosedur dengan menggambarkan langkah-langkah menggunakan diagram alir atau *flowchart*.



Gambar 2. Flowchart Sistem

Langkah awal adalah dengan mengimport *library* yang dibutuhkan untuk penelitian, kemudian dilanjutkan dengan input *dataset "Credit Card Dataset for Clustering"*. Selanjutnya, dilakukan perulangan pengecekan data untuk memastikan data telah sesuai dengan melalui proses data *loading*, data *cleaning*, dan eksplorasi data.

Setelah data telah disiapkan, dilakukan perulangan pengecekan model untuk memastikan model telah sesuai. Langkah-langkah yang dilibatkan dalam tahap ini meliputi *preprocessing* data, pembuatan model *clustering* menggunakan algoritma *K-Means*, melatih model, evaluasi model, dan pembuatan model inferensi untuk pengujian.

Proses ini berulang hingga model yang dihasilkan sesuai dengan kriteria yang ditentukan.

Dengan menggunakan *flowchart* sebagai panduan, penelitian ini memberikan gambaran jelas tentang proses eksekusi *K-Means* pada dataset "*Credit Card Dataset for Clustering*" sehingga dapat membantu proses dalam memahami dan menerapkan algoritma

klustering *K-Means* dengan lebih terstruktur dan efisien.

4. ANALISIS KEBUTUHAN

Dalam penelitian ini memiliki beberapa kebutuhan fungsional yang dibagi atas dua target utama atau aktor yaitu Admin dan Kepala Marketing Bank. Kebutuhan kedua aktor penting dimana interaksi yang ada akan melibatkan keduanya, yaitu peneliti dan pihak bank.

Adapun beberapa kebutuhan fungsional sistem dijelaskan pada tabel berikut.

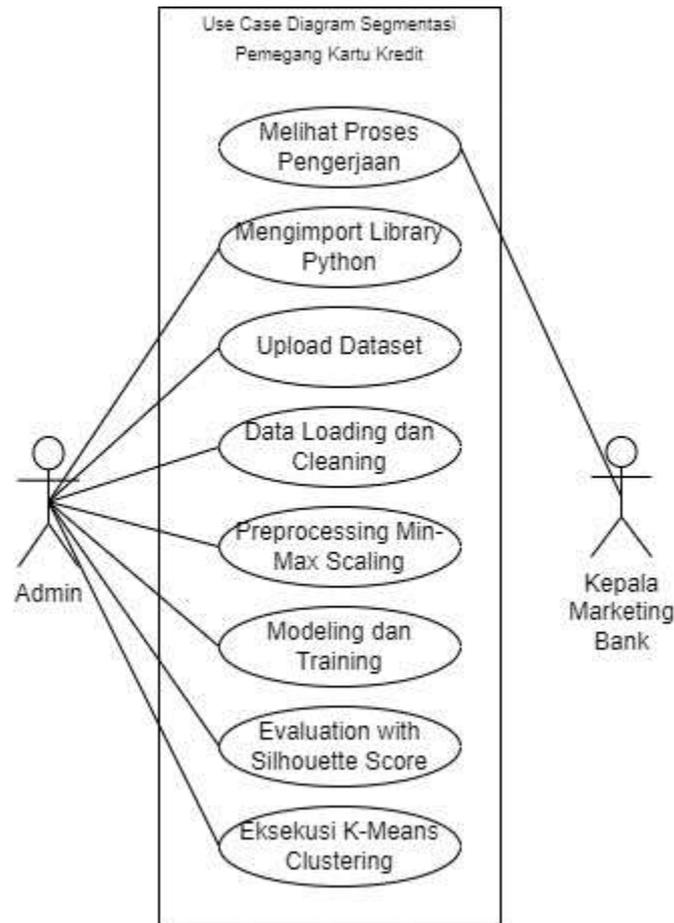
Tabel 6. Kebutuhan Fungsional Sistem

No	Aktor	Kebutuhan Sistem	Deskripsi
1	Admin	Mengimport Library Python	Peneliti dapat mengimport library python
		Upload Dataset	Peneliti dapat mengupload dataset
		Data Loading dan Cleaning	Peneliti dapat melakukan data loading dan cleaning
		Preprocessing Min-Max Scaling	Peneliti dapat melakukan preprocessin Min-Max Scaling
		Modeling dan Training	Peneliti dapat melakukan modeling dan training model
		Evaluation with Silhouette Score	Peneliti dapat melakukan evaluasi model dengan silhouette score
		Eksekusi K-means <i>Clustering</i>	Peneliti dapat melakukan eksekusi hasil klustering menggunakan K-Means model yang telah dibuat
2	Kepala Marketing Bank	Melihat Proses Pengerjaan	Pihak Bank dapat melihat prose pengerjaan yang dilakukan

5. PERANCANGAN FUNGSIONAL SISTEM

Berikut adalah perancangan fungsional sistem segmentasi pemegang kartu kredit pada gambar 3 yang di visualisasikan dalam Use

Case diagram. Terdapat dua aktor yaitu Admin dan Kepala Marketing Bank yang memiliki akses langsung kepada sistem dan menggunakan fungsi-fungsi yang ada dalam sistem.



Gambar 3. Use Case Diagram Segmentasi Pemegang Kartu Kredit

6. PENGUJIAN

Berikut diberikan hasil pengujian segmentasi untuk menilai keterpenuhan spesifikasi fungsi sistem sesuai dengan analisis dan perancangan yang telah dibuat. Tabel berikut menggambarkan hasil uji yang telah dilakukan terhadap target pengguna yaitu

Admin dan Kepala Marketing Bank. Tabel 7 menunjukkan hasil uji yang dilakukan oleh admin untuk segmentasi pengguna kartu kredit dan Tabel 8 menunjukkan hasil uji yang dilakukan oleh Kepala Marketing Bank untuk melihat proses eksekusi sistem.

Tabel 7. Pengujian sistem segmentasi oleh Admin

No	Deskripsi	Prosedur pengujian	Masukan	Hasil pengujian	Kesimpulan
1	Mengimport Library Python	Periksa apakah library sudah terpasang pada code editor	Perintah Import Pandas, NumPy, Matplotlib.pyplot, Seaborn, dan Warnings	Pass	Library telah terimpor dan siap digunakan
2	Upload Dataset	Menjalankan code perintah pd.read_csv(url) dan df.info	Dataset Kartu Kredit	Pass	Data telah terupload dan terbaca oleh sistem
3	Data Loading dan Cleaning	Menjalankan perintah untuk mengecek informasi data serta	Dataset Kartu Kredit	Pass	Dataset berhasil diloading untuk melihat informasinya serta

		menjalankan perintah code untuk membersihkan data dari missing value dan outlier			berhasil dibersihkan dari outlier
4	Preprocessing Min-Max Scaling	Menjalankan code perintah untuk scalling menggunakan Min-Max	Dataset Kartu Kredit hasil loading dan cleaning	Pass	Data berhasil discalling
5	Modeling dan Training	Menjalankan perintah Modeling K-Means dan perintah training model	Dataset Kartu Kredit hasil preprocessing Min-Max Scalling	Pass	Model dataset berhasil terbuat dan juga berhasil terlatih
6	Evaluation with Silhouette Score	Menjalankan perintah uji skor silhouette untuk evaluasi clustering	Dataset Kartu Kredit hasil Modeling dan Training	Pass	Skor silhouette menunjukkan kualitas clustering yang baik
7	Eksekusi K-means Clustering	Menggunakan model K-means untuk clustering	Dataset Kartu Kredit hasil evaluasi menggunakan skor silhouette	Pass	Model berhasil mengelompokkan 3 pengguna kartu kredit

Tabel 8. Pengujian sistem oleh Kepala Marketing Bank

No	Deskripsi	Prosedur pengujian	Masukan	Hasil pengujian	Kesimpulan
1	Melihat Proses Pengerjaan	Melihat hasil tiap pengerjaan yang dilakukan oleh Admin	-	Pass	Kepala Marketing Bank melihat seluruh proses yang dilakukan oleh admin dalam pengerjaan segmentasi pemegang kartu kredit

7. PEMBAHASAN

Data yang digunakan berasal dari data pengguna kartu kredit aktif suatu bank yang bisa diakses melalui link berikut: <https://www.kaggle.com/datasets/arjunbhasin2013/ccdata>

Data “Credit Card Dataset for Clustering” berisi informasi perilaku sekitar 8950 pemegang kartu kredit aktif selama enam bulan terakhir. Dataset ini mencakup 18 variabel perilaku, termasuk saldo akun, frekuensi, pembelian, batas kredit, dan lainnya. Output yang diharapkan untuk segmentasi memiliki 3 cluster yaitu:

1. Cluster 0

- 'balance' berada pada kategori "Low Balance" karena rata-rata saldo yang relatif rendah.
- 'purchases' berada pada kategori "Low Purchases" karena rata-rata

jumlah pembelian yang cenderung rendah.

- 'cash_advance' berada pada kategori "Low Cash Advance" karena rata-rata jumlah uang muka yang diberikan oleh pelanggan cenderung rendah.
- 'credit_limit' berada pada kategori "Medium Credit Limit" karena rata-rata batas kredit yang diberikan kepada pelanggan berada pada tingkat menengah.
- 'payments' berada pada kategori "Medium Payments" karena rata-rata jumlah pembayaran yang dilakukan oleh pelanggan berada pada tingkat menengah.

2. Cluster 1

- 'balance' berada pada kategori "Medium Balance" karena rata-rata saldo yang berada pada tingkat menengah.

- 'purchases' berada pada kategori "High Purchases" karena rata-rata jumlah pembelian yang dilakukan oleh pelanggan cenderung tinggi.
- 'cash_advance' berada pada kategori "Low Cash Advance" karena rata-rata jumlah uang muka yang diberikan oleh pelanggan cenderung rendah.
- 'credit_limit' berada pada kategori "High Credit Limit" karena rata-rata batas kredit yang diberikan kepada pelanggan cenderung tinggi.
- 'payments' berada pada kategori "High Payments" karena rata-rata jumlah pembayaran yang dilakukan oleh pelanggan cenderung tinggi.

3. Cluster 2

- 'balance' berada pada kategori "High Balance" karena rata-rata saldo yang relatif tinggi.
- 'purchases' berada pada kategori "Low Purchases" karena rata-rata jumlah pembelian yang cenderung rendah.
- 'cash_advance' berada pada kategori "High Cash Advance" karena rata-rata jumlah uang muka yang diberikan oleh pelanggan cenderung tinggi.
- 'credit_limit' berada pada kategori "High Credit Limit" karena rata-rata batas kredit yang diberikan kepada pelanggan cenderung tinggi.
- 'payments' berada pada kategori "High Payments" karena rata-rata jumlah pembayaran yang dilakukan oleh pelanggan cenderung tinggi.

Disini dibahas hasil penggunaan bahasa pemrograman *Python* untuk kebutuhan *data science* dalam mengclustering dataset "Credit Card Dataset for Clustering".

1. Tahap persiapan dimulai dengan mengimpor pustaka-pustaka yang diperlukan untuk pengolahan data menggunakan metode *Unsupervised*

Learning dengan Klustering. Pada Gambar 4, terdapat pustaka yang dibutuhkan dalam proses pengolahan dan pembuatan model *machine learning* untuk penelitian. Tahap ini memiliki tingkat penting yang tinggi karena dalam pengembangan *data science* menggunakan bahasa pemrograman *Python*, ketergantungan terhadap pustaka-pustaka tersebut sangatlah signifikan.

```
# data Manipulasi dan Analisis
import pandas as pd
import numpy as np

# visualisasi Data
import matplotlib.pyplot as plt
import seaborn as sns

# machine Learning dan Pemodelan
from sklearn.preprocessing import
MinMaxScaler
from scipy import stats
from sklearn.cluster import
KMeans
from sklearn.model_selection
import train_test_split

# lainnya
import warnings
warnings.filterwarnings("ignore")
```

Gambar 4. Sintaks Import Library yang Dibutuhkan

Tabel di bawah ini berisi tentang informasi daftar pustaka yang diperlukan dalam pengolahan dan pembuatan model *machine learning* pada penelitian sesuai dengan sintaks koding yang telah dibuat diatas.

Tabel 9. Penjelasan Library yang Digunakan

Pustaka/Fungsi	Penjelasan
pandas	Ini adalah pustaka Python yang menyediakan struktur data dan alat analisis data yang fleksibel dan efisien. Biasanya digunakan untuk manipulasi data dan analisis data dalam Python.
numpy	Ini adalah pustaka Python yang digunakan untuk komputasi numerik. Ia memiliki dukungan kuat untuk array berdimensi besar dan matriks matematika, serta fungsi matematika tingkat tinggi untuk beroperasi pada struktur data ini.

matplotlib.pyplot	Ini adalah pustaka plotting Python yang digunakan untuk membuat visualisasi data 2D dan 3D, seperti grafik batang, pie chart, histogram, scatterplot, dan lainnya.
seaborn	Ini adalah pustaka visualisasi data Python berbasis matplotlib yang menyediakan antarmuka tingkat tinggi untuk membuat grafik statistik yang menarik dan informatif.
sklearn.preprocessing.MaxScaler	Ini adalah metode preprocessing yang mengubah fitur dengan penskalaan setiap fitur ke rentang yang diberikan, biasanya antara nol dan satu. Hal ini penting dalam banyak algoritma machine learning.
scipy.stats	Ini adalah modul scipy yang berisi sejumlah besar distribusi probabilitas dan fungsi statistik.
sklearn.cluster.KMeans	K-Means adalah algoritma pengelompokan populer yang mencoba untuk mempartisi kumpulan data menjadi K kluster pra-penentuan (subkelompok) non-overlapping. Setiap data point milik kluster dengan rata-rata terdekat (centroid kluster).
sklearn.model_selection.train_test_split	Fungsi ini digunakan untuk membagi kumpulan data menjadi dua bagian (pelatihan dan pengujian). Hal ini penting dalam memvalidasi performa model.
warnings.filterwarnings("ignore")	Fungsi ini digunakan untuk mengabaikan peringatan yang dikeluarkan oleh berbagai modul atau pustaka dalam Python.

2. Setelah mengimpor pustaka yang diperlukan, langkah selanjutnya adalah tahap upload dataset “Credit Card Dataset for Clustering”.

```
# read dataset
df = pd.read_csv('CC
General.csv')
```

Gambar 5. Sintaks Untuk Upload Dataset

Sintaks yang telah ditulis diatas digunakan untuk membaca dataset dari file CSV menggunakan library Pandas di Python. ‘pd’ adalah alias yang digunakan untuk Pandas. ‘read_csv’ adalah fungsi Pandas yang digunakan untuk membaca file CSV. ‘CC General.csv’ adalah nama file CSV

yang akan dibaca, dan hasilnya akan disimpan dalam variable ‘df’.

3. Tahap selanjutnya yaitu melakukan *data loading* dan *data cleaning* terhadap dataset yang sudah diupload. Perlu dilakukan pencarian informasi dasar terkait dataset dan melanjutkan ke tahap pembersihan data sebelum akhirnya nanti akan diproses ke tahap eksploitasi data.

```
# melihat informasi dari dataset
df.info()
```

Gambar 6. Sintaks Untuk Mengetahui Informasi Dataset

➔ Dengan sintaks “df.info()” membuat sistem membaca informasi yang dimiliki oleh dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8950 entries, 0 to 8949
Data columns (total 18 columns):
 #   Column                Non-Null Count  Dtype
---  ---                ---
 0   CUST_ID               8950 non-null   object
 1   BALANCE               8950 non-null   float64
 2   BALANCE_FREQUENCY    8950 non-null   float64
 3   PURCHASES            8950 non-null   float64
 4   ONEOFF_PURCHASES    8950 non-null   float64
 5   INSTALLMENTS_PURCHASES 8950 non-null   float64
 6   CASH_ADVANCE         8950 non-null   float64
 7   PURCHASES_FREQUENCY 8950 non-null   float64
 8   ONEOFF_PURCHASES_FREQUENCY 8950 non-null   float64
 9   PURCHASES_INSTALLMENTS_FREQUENCY 8950 non-null   float64
10  CASH_ADVANCE_FREQUENCY 8950 non-null   float64
11  CASH_ADVANCE_TRX     8950 non-null   int64
12  PURCHASES_TRX       8950 non-null   int64
13  CREDIT_LIMIT         8948 non-null   float64
14  PAYMENTS            8950 non-null   float64
15  MINIMUM_PAYMENTS    8637 non-null   float64
16  PRC_FULL_PAYMENT    8950 non-null   float64
17  TENURE              8950 non-null   int64
dtypes: float64(14), int64(3), object(1)
memory usage: 2.2+ MB
```

Gambar 7. Informasi Dataset

➔ Berdasarkan informasi yang ditemukan dengan menggunakan sintaks “df.info()”, dataset *Credit Card* terdiri dari 18 atribut dan 8.950 baris.

➔ Atribut-atribut tersebut terdiri dari 14 atribut dengan tipe data *float*, 3 atribut dengan tipe data *integer*, dan 1 atribut dengan tipe data objek.

```
# menampilkan jumlah missing value pada dataset
df.isna().sum()
```

Gambar 8. Mencari Informasi Missing Value

➔ Dengan menggunakan sintaks “df.isna().sum()” dimana “df.isna()” akan mencari atribut

yang memiliki missing value dan “sum()” akan menjumlahkan total dari missing value yang ditemukan.

```
balance          0
balance_frequency 0
purchases        0
oneoff_purchases 0
installments_purchases 0
cash_advance     0
purchases_frequency 0
oneoff_purchases_frequency 0
purchases_installments_frequency 0
cash_advance_frequency 0
cash_advance_trx 0
purchases_trx   0
credit_limit    1
payments       0
minimum_payments 313
prc_full_payment 0
tenure         0
dtype: int64
```

Gambar 9. Missing Value pada Dataset

- ➔ Ditemukan adanya *missing values* pada atribut *credit_limit* (1 *missing value*) dan atribut *minimum_payments* (313 *missing values*).

```
# mengisi missing value dengan nilai mean
df = df.fillna(df.mean())
```

Gambar 10. Sintaks Mengatasi Missing Value

- ➔ Diputuskan untuk mengatasi *missing value* tersebut dengan mengisinya dengan nilai mean dengan perintah sintaks “df = df.fillna(df.mean())” diaman “df.fillna” merupakan fungsi untuk mengisi nilai *missing value* dengan nilai yang di tentukan yaitu dengan *mean* menggunakan sintaks “df.mean()”.
4. Langkah selanjutnya dalam *preprocessing* adalah melakukan *scaling* data menggunakan metode *Min-Max Scaling*. *Scaling* data ini bertujuan untuk menormalkan nilai dan tipe data dalam rentang yang sama, sehingga tidak terjadi bias akibat perbedaan tinggi rendahnya angka dalam data. Dengan melakukan *scaling* data menggunakan metode *Min-Max Scaling*, proses pemodelan akan menjadi lebih baik dan efisien.

```
# Membuat objek MinMaxScaler
```

```
scaler = MinMaxScaler()

# Melakukan penskalaan fitur dan menyimpan hasilnya pada variabel baru
df_scaled = scaler.fit_transform(df)
```

Gambar 11. Sintaks Untuk Scalling Dataset

- ➔ “scaler = MinMaxScaler()” sintaks ini membuat objek *MinMaxScaler* dari *library* “sklearn.preprocessing”. Objek ini akan digunakan untuk melakukan penskalaan fitur pada dataset.
- ➔ “df_scaled = scaler.fit_transform(df)” sintaks ini melakukan dua tugas sekaligus. Metode *fit* menghitung parameter penskalaan (yaitu nilai minimum dan maksimum setiap fitur) yang diperlukan untuk proses penskalaan. Sedangkan metode *transform* mengaplikasikan penskalaan ke dataset berdasarkan parameter yang dihitung. Jadi, secara singkat, *fit_transform* menghitung parameter penskalaan dan kemudian langsung menerapkannya pada data. Hasil dari penskalaan fitur ini disimpan dalam variabel *df_scaled*.

5. Setelah melakukan *preprocessing* pada data, langkah selanjutnya adalah membuat model *K-means* serta melatihnya.

```
# Model K-means
kmeans = KMeans(n_clusters=3)

# melatih model K-means
kmeans.fit(df_scaled)
```

Gambar 12. Sintaks Load Model K-Means dan Melatihnya

- ➔ “kmeans = KMeans(n_clusters=3)” sintaks ini adalah inisialisasi model *K-Means* dari *library* *sklearn.cluster*. Mengatur jumlah *cluster* yang diinginkan (dalam hal ini 3) sebagai parameter.
- ➔ “kmeans.fit(df_scaled)” dengan sintaks ini akan melatih atau “fit”

model *K-Means* pada data yang telah diskalakan “df_scaled”. Proses pelatihan ini melibatkan penemuan *centroid cluster* dan penugasan setiap titik data ke *cluster* terdekat.

6. Proses selanjutnya adalah evaluasi model.

```
from sklearn.metrics import
silhouette_score

# Evaluasi K-means
kmeans_silhouette =
silhouette_score(df_scaled,
kmeans.labels_)

print("Silhouette Score K-
means:", kmeans_silhouette)
```

Gambar 13. Sintaks Evaluasi Model

- “from sklearn.metrics import silhouette_score” sintaks ini akan mengimpor fungsi *silhouette_score* dari modul *sklearn.metrics*. Skor siluet digunakan untuk mengevaluasi kualitas model klustering. Skor ini berkisar antara -1 hingga 1, di mana nilai yang lebih tinggi menunjukkan bahwa sampel lebih dekat ke klaster mereka sendiri daripada klaster lain.
- “kmeans_silhouette = silhouette_score(df_scaled, kmeans.labels_)” sintaks ini menghitung skor siluet untuk model *K-Means*. Parameter pertama yang diberikan adalah data (df_scaled), dan parameter kedua adalah label klaster yang ditentukan oleh model *K-Means* (kmeans.labels_).
- “print(“Silhouette Score K-means:”, kmeans_silhouette)” sintaks mencetak skor siluet untuk model *K-Means*.

Silhouette Score K-means: 0.29738856808365083

Gambar 14. Hasil Evaluasi Menggunakan Silhouette

Dalam proses evaluasi model klustering, salah satu metrik yang digunakan adalah *Silhouette Score*. *Silhouette Score* mengukur seberapa baik tiap sampel berada dalam kelompoknya sendiri dibandingkan dengan kelompok lainnya. Semakin tinggi nilai *Silhouette Score*, semakin baik pemisahan antar kelompok.

Evaluasi model klustering menggunakan algoritma *K-Means* yang telah dilatih, diperoleh hasil *Silhouette Score K-means* 0.4030866767042894 yang berarti model memiliki hasil yang baik baik.

7. Tahap terakhir adalah melakukan model *inferensi* untuk melakukan pengelompokan atau *clustering* pada model yang telah dibuat.

```
# Melakukan clustering dengan K-
means
kmeans =
KMeans(n_clusters=n_clusters,
random_state=42)
kmeans.fit(df_scaled)

# Mendapatkan label hasil
clustering
kmeans_labels = kmeans.labels_

# Menambahkan kolom label ke
dalam dataset
df['KMeans_Labels'] =
kmeans_labels

# Menghitung jumlah data dalam
setiap kelompok hasil segmentasi
kmeans_counts =
df['KMeans_Labels'].value_counts(
)

# Menampilkan urutan segmentasi
pelanggan
kmeans_sorted =
kmeans_counts.sort_values(ascendi
ng=False)
print(kmeans_sorted)
```

Gambar 15. Model Inference

- “kmeans = KMeans(n_clusters=n_clusters, random_state=42)” sintak ini menginisialisasi objek *K-Means*. Jumlah klaster ditentukan oleh

variabel `n_clusters` dan `random_state` ditetapkan menjadi 42 untuk hasil yang konsisten.

- `"kmeans.fit(df_scaled)"` sintaks untuk Melatih model *K-Means* pada data yang telah diskalakan (`df_scaled`).
- `"kmeans_labels = kmeans.labels_"` sintaks untuk mendapatkan label kluster untuk setiap sampel data setelah proses klustering.
- `"df['KMeans_Labels'] = kmeans_labels"` sintaks Menambahkan label kluster sebagai kolom baru di DataFrame asli.
- `"kmeans_counts = df['KMeans_Labels'].value_counts()"` sintaks untuk menghitung jumlah sampel dalam setiap kluster.
- `"kmeans_sorted = kmeans_counts.sort_values(ascending=False)"` sintaks untuk mengurutkan kluster berdasarkan jumlah sampel di dalamnya, dari yang paling banyak sampai yang paling sedikit.
- `"print(kmeans_sorted)"` sintaks untuk menampilkan urutan kluster berdasarkan jumlah sampel.

8. HASIL

Berikut pada gambar 16 adalah hasil yang didapatkan dari segmentasi pemegang kartu kredit dengan menggunakan algoritma *K-Means* dan bahasa pemrograman *Python* untuk eksekusinya.

```
0    5440
1    1645
2     1291
Name: KMeans_Labels, dtype: int64
```

Gambar 16. Hasil Model Inferensi

Berdasarkan hasil segmentasi diatas, dapat ditarik beberapa kesimpulan:

- *Cluster 0* adalah kluster dengan jumlah pelanggan terbanyak (5440 pelanggan).
- *Cluster 1* adalah kluster dengan jumlah pelanggan kedua terbanyak (1645 pelanggan).

- *Cluster 2* adalah kluster dengan jumlah pelanggan terkecil (1291 pelanggan).

9. KESIMPULAN

Dari hasil penelitian ini dapat ditarik beberapa kesimpulan:

1. Penerapan Metode: Dengan menggunakan metode *K-means Clustering*, kita dapat mengidentifikasi dan mengelompokkan pelanggan menjadi tiga kluster utama berdasarkan pola perilaku mereka dalam menggunakan kartu kredit. Metode ini efektif dalam menangani data numerik dan dapat memberikan *insight* penting tentang segmen pelanggan yang berbeda.
2. Karakteristik Segmen: Segmen yang diidentifikasi adalah "Cluster 0", "Cluster 1", dan "Cluster 2". Setiap segmen ini memiliki karakteristik dan perilaku yang berbeda dalam menggunakan kartu kredit mereka.
3. Manfaat Bisnis: Pengetahuan ini sangat berharga untuk sektor bank dalam merancang strategi bisnis kartu kredit yang tepat dan personalisasi layanan. Dengan data segmentasi penelitian ini, segmentasi dapat merancang program rewards atau penawaran khusus untuk "Pelanggan Berbelanja" untuk mendorong lebih banyak pengeluaran, atau program konseling keuangan untuk membantu "Pelanggan Berutang" mengelola utang mereka.

UCAPAN TERIMAKASIH

Saya mengucapkan terima kasih kepada Program Studi Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Jakarta yang telah memberikan kesempatan untuk saya dapat mengikuti program di PT. Hactivate Teknologi Indonesia serta terima kasih kepada Ibu Rita Dwi Risanty yang telah membimbing saya sampai terciptanya jurnal ini.

DAFTAR PUSTAKA

- Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means Algorithm: A Comprehensive Survey and Performance Evaluation. *Electronics*, 9(8), 1295. <https://doi.org/10.3390/electronics9081295>
- Dewri, L. V., Islam, Md. R., & Saha, N. K. (2016). Behavioral Analysis of Credit Card Users in Developing Country: A Case of Bangladesh. *International Journal of Business and Management*, 11(4), 299. <https://doi.org/10.5539/ijbm.v11n4p299>
- Dinata, R. K., Safwandi, S., Hasdyna, N., & Azizah, N. (2020). Analisis K-Means Clustering pada Data Sepeda Motor. *INFORMAL: Informatics Journal*, 5(1), 10. <https://doi.org/10.19184/isj.v5i1.17071>
- Dudiyanto, M. (2021). PERTUMBUHANKARTU KREDIT DI INDONESIA DALAM PANDEMI COVID'19. *Jurnal Manajemen*, 11(1), 34–41.
- Fitriani, M. N. R., Priyatna, B., Huda, B., Hananto, A. L., & Tukino, T. (2023). Implementasi Metode K-Means Untuk Memprediksi Status Kredit Macet. *Jurnal Sistem Komputer Dan Informatika (JSON)*, 4(3), 554. <https://doi.org/10.30865/json.v4i3.5953>
- Kurniawan, Y. I., & Barokah, T. I. (2020). Klasifikasi Penentuan Pengajuan Kartu Kredit Menggunakan K-Nearest Neighbor. *Jurnal Ilmiah Matrik*, 22(1), 73–82. <https://doi.org/10.33557/jurnalatrik.v22i1.843>
- Luthfi, E., & Wijayanto, A. W. (2021). Analisis perbandingan metode hierarchical, k-means, dan k-medoids clustering dalam pengelompokan indeks pembangunan manusia Indonesia. *INOVASI*, 17(4), 761–773. <https://doi.org/10.30872/jinv.v17i4.10106>
- Noviyanto, D. A., & Sukiman, S. J. F. X. (2022). ANALISIS STRATEGI PEMASARAN PRODUK KARTU KREDIT DALAM MENARIK MINAT DAN KEPUASAN MASYARAKAT GENERASI MILENIAL MENJADI CALON NASABAH DI ERA PANDEMI COVID 19. *Soetomo Business Review*, 3(1), 125–165.
- Raharjo, B. (2021). *FINTECH TEKNOLOGI FINANSIAL PERBANKAN DIGITAL*. Penerbit Yayasan Prima Agus Teknik.
- Vaishali, V. (2014). Fraud Detection in Credit Card by Clustering Approach. *International Journal of Computer Applications*, 98(3), 29–32. <https://doi.org/10.5120/17164-7225>