Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 5, October 2024, pp. 1487-1495
DOI: https://doi.org/10.52436/1.jutif.2024.5.5.2341
p-ISSN: 2723-3863
e-ISSN: 2723-3871

IDENTIFYING POTENTIAL CREDIT CARD PAYMENT DEFAULTS USING GMDKNN WITH LOF AS OUTLIER HANDLING

Liony Puspita Dewi[*1], Yulison Herry Chrisnanto[*2], Rezki Yuniarti[3]

[1,2,3]Informatics, Sains and Informatics Faculty, Universitas Jenderal Achmad Yani, Indonesia
Email: [1]lionypuspita20@if.unjani.ac.id, [2]yhc@if.unjani.ac.id, [3]rezki.yuniarti@lecturer.unjani.ac.id

(Article received: June 20, 2024; Revision: July 10, 2024; published: November 02, 2024)

Abstract

In classifying data, accuracy results are greatly influenced by outliers. The presence of outliers can cause a low level of accuracy in the classification process. The Generalised Mean Distance K-Nearest Neighbor (GMD-KNN) algorithm is a classification technique that shows advantages in terms of flexibility and responsiveness to attribute variations. This research aims to classify credit card data between current and bad payments by handling outliers using the Local Outlier Factor (LOF). The data used is 30,000 credit card transaction data taken from the UCI Machine Learning Repository. This research method uses several stages, namely data collection, data pre-processing carried out to detect and clean outliers with LOF, classification process with GMD-KNN, and evaluation to calculate the accuracy of classification results. As a result, the model shows the best performance at 80%:20% data sharing ratio with k=5 value, achieving 77.60% accuracy, 74.97% precision, 82.57% recall, 78.58% F1-Score, and 77.48% G-Mean.

Keywords: *classification, credit card, GMD-KNN, LOF, outlier.*

IDENTIFIKASI POTENSI KELANCARAN PEMBAYARAN KARTU KREDIT MENGGUNAKAN GMD-KNN DENGAN LOF SEBAGAI PENANGANAN OUTLIER

Abstrak

Dalam mengklasifikasi suatu data hasil akurasi sangat dipengaruhi oleh outlier. Keberadaan outlier dapat menyebabkan tingkat akurasi yang rendah pada proses klasifikasi. Algoritma *Generalized Mean Distance K-Nearest Neighbor* (GMD-KNN) adalah suatu teknik klasifikasi yang memperlihatkan kelebihan dalam hal fleksibilitas dan responsif terhadap variasi atribut. Penelitian ini bertujuan untuk klasifikasi data kartu kredit antara pembayaran lancar dan macet dengan melakukan penanganan outlier menggunakan *Local Outlier Factor* (LOF). Data yang digunakan berupa data transaksi kartu kredit berjumlah 30.000 yang diambil dari *UCI Machine Learning Repository*. Dalam metode penelitian ini menggunakan beberapa tahap, yaitu pengumpulan data, pre-processing data yang dilakukan untuk mendeteksi dan membersihkan outlier dengan LOF, proses klasifikasi dengan GMD-KNN, dan evaluasi untuk menghitung ukuran keakuratan hasil klasifikasi. Hasil dari penelitian ini, model menunjukkan kinerja terbaik pada rasio pembagian data 80%:20% dengan nilai k=5, mencapai akurasi 77,60%, precision 74,97%, recall 82,57%, F1-Score 78,58%, dan G-Mean 77,48%..

Kata kunci: *GMD-KNN, kartu kredit, klasifikasi, LOF, outlier.*

1. PENDAHULUAN

Dalam era perkembangan teknologi informasi, penggunaan analisis data untuk pengambilan keputusan semakin mendominasi berbagai bidang, termasuk kesehatan, finansial, dan kecerdasan buatan [1]. Selama waktu ini, banyak sistem komputer mungkin terpengaruh dan sumber daya berharga mungkin hilang [2].

Tantangan utama dalam analisis data adalah kehadiran outlier yang dapat mengakibatkan kesalahan dalam hasil klasifikasi atau prediksi model. Adanya outlier bisa menjadi sangat penting karena beberapa alasan dan perlu dideteksi [3]. Outlier dapat muncul dalam berbagai bentuk dan memiliki potensi besar untuk memengaruhi interpretasi dan keandalan model [4]. Outlier adalah area dalam analisis data yang umumnya diterapkan dalam bidang keuangan dan internet. Ini berkaitan dengan pengidentifikasian objek yang berbeda dari keseluruhan data, seperti transaksi yang tidak sah [5].

Kartu kredit merupakan salah satu cara pembayaran yang sangat vital dalam kehidupan masyarakat modern, namun semakin banyak kejadian penipuan terjadi dalam penggunaannya [6]. Kartu

kredit dapat melakukan berbagai transaksi dengan mudah, namun bisa saja adanya ancaman yang dapat merugikan atau menipu [7]. Meskipun kartu kredit memiliki berbagai keunggulan, namun juga terdapat risiko, terutama jika pemegang kartu tidak melakukan pembayaran secara rutin dan dapat terjadi pelanggaran dengan pola transaksi yang serupa [7], [8]. Untuk mengurangi penipuan kartu kredit, penting untuk menggunakan aturan ahli dan model berbasis statistik, seperti mesin pendeteksi berbasis aturan, machine learning, dan teknik novelty detection seperti Clustering, metode berbasis klasifikasi, dan tetangga terdekat [9].

Local Outlier Factor (LOF) merupakan salah satu teknik untuk mendeteksi outlier. Teknik ini menentukan tingkat keanehan suatu titik dengan mengukur faktor deviasi objek dalam kumpulan data [10]. Nilai *Local Outlier Factor* (LOF) dari data ditentukan berdasarkan kepadatan lokal di sekitarnya, dan data yang abnormal diidentifikasi berdasarkan nilai pencilan tersebut [4].

Jianping Gou dan kawan-kawan membahas mengenai kinerja klasifikasi pada berbagai dataset dengan membandingkan beberapa metode klasifikasi yang bertujuan untuk mengatasi sensitivitas terhadap parameter ukuran ketetanggaan (k), menyimpulkan bahwa penggunaan metode GMD-KNN dapat meningkatkan kinerja klasifikasi berbasis KNN [11].

Sugriyono dan Maria Ulfah Siregar membahas mengenai mendeteksi dan menghapus outlier menggunakan K-Means dan matriks jarak untuk meningkatkan akurasi klasifikasi, menunjukkan bahwa akurasi klasifikasi algoritma K-NN dipengaruhi oleh pencilan dalam dataset. Akurasi secara signifikan ditingkatkan dengan menghapus outlier dari dataset, yang mengarah ke hasil klasifikasi yang lebih andal [12].

Penelitian ini dilakukan untuk mengklasifikasikan data kartu kredit menggunakan *Generalized Mean Distance K-Nearest Neighbor* (GMD-KNN) dengan mempertimbangkan penanganan outlier menggunakan *Local Outlier Factor* (LOF). Dengan implementasi metode tersebut, penelitian ini diharapkan dapat memberikan kontribusi terhadap ilmu pengetahuan dalam perancangan model untuk mendapatkan akurasi yang tinggi pada hasil klasifikasi dalam menghadapi data yang bervariasi.

2. TINJAUAN LITERATUR

2.2. Data Mining

Data mining melibatkan penggunaan fakta, angka, kecerdasan buatan manusia, dan strategi kecerdasan buatan untuk memisahkan dan mengidentifikasi informasi berguna dari kumpulan data yang besar. Dalam konteks data mining, terdapat berbagai teknik, salah satunya adalah teknik klasifikasi [8]. Dari pengolahan menggunakan data mining, berbagai informasi berharga dapat diperoleh [13]. Prinsip kerja data mining adalah membagi data besar menjadi bagian yang lebih kecil atau ringkasan, dan mengidentifikasi informasi yang berharga serta pengetahuan yang terkait dari berbagai basis data yang besar [14].

2.2. Klasifikasi

Teknik klasifikasi berdasarkan pengamatan dari data dan atributnya, sehingga dapat menempatkan data yang belum diklasifikasikan ke dalam kategori yang sudah ada, sesuai dengan aturan-aturan yang sudah ditetapkan [15]. Teknik klasifikasi dengan menggunakan algoritma biasa dilakukan melalui 3 tahapan, yaitu Perancangan Model untuk menyelesaikan masalah atau yang disebut data latih, Implementasi Model untuk menentuan kelas pada data uji dilakukan berdasarkan model fungsi dan parameter-parameter yang telah ditetapkan pada tahap perancangan, dan Evaluasi Model untuk menilai hasil implementasi model fungsi final parameter yang telah ditetapkan [15]. Klasifikasi digunakan untuk mengelompokkan jenis objek pada data baru. Klasifikasi termasuk dalam model supervised [16].

2.3. Deteksi *Outlier*

Deteksi outlier merupakan proses mendeteksi pencilan dalam dataset, yang dilakukan pada tahap prapemrosesan data analitik [3]. Deteksi outlier adalah masalah yang diteliti pada kedua komunitas yaitu statistik dan penggalian data, tetapi dengan pandangan yang berbeda [17]. Tugas deteksi pencilan adalah mengidentifikasi pola data yang tidak sesuai dengan perilaku yang diharapkan dalam kumpulan data yang ada, seperti pencilan, pengamatan yang tidak konsisten, penyimpangan, dan nilai yang mencolok [4].

2.4. *Local Outlier Factor* (LOF)

Algoritma ini menghitung densitas lokal yang dapat dijangkau dari data, dan menghitung skor faktor pencilan lokal menurut densitas lokal yang dapat dijangkau [4]. Dengan menambahkan batas atau nilai ambang batas yang tinggi ke dalam algoritma Local Outlier Factor (LOF) akan mengurangi jumlah data yang terdeteksi sebagai outlier [7]. Jika nilai LOF suatu data melebihi ambang batas yang ditetapkan, maka data tersebut dianggap sebagai outlier. Sebaliknya, jika nilai LOF kurang dari atau sama dengan ambang batas, maka data tersebut dianggap sebagai data normal atau bukan outlier[7].

2.5. GMD-KNN

Algoritma Generalized Mean Distance K-Nearest Neighbor (GMDKNN) bekerja dengan menyimpan daftar terurut dari k tetangga terdekat untuk setiap kelas [18]. Kemudian dilanjutkan dengan untuk mengubah setiap daftar menjadi vektor rata-

rata lokal, dari mana beberapa perhitungan jarak rata-rata dihitung berulang untuk menghasilkan nilai final untuk jarak setiap kelas ke kueri pengujian. Kelas dengan jarak terkecil yang dianggap sebagai prediksi yang benar untuk kueri pengujian [18]. Dengan pendekatan ini, lebih banyak bobot diberikan kepada tetangga yang lebih dekat, dan kekuatan bobot klasifikasi dapat diatur secara dinamis melalui nilai p yang berbeda [11].

2.6. *Confusion Matrix*

Confusion Matrix digunakan untuk mengevaluasi tingkat keakuratan, presisi, dan recall dalam suatu model klasifikasi [15]. Pada Confusion Matrix terdapat informasi yang membandingkan hasil klasifikasi sistem dengan hasil klasifikasi yang seharusnya [19]. Confusion Matrix mencakup 4 (empat) istilah yang menjelaskan klasifikasi hasil pengukuran kinerja, yaitu True Negative (TN), False Positive (FP), True Positive (TP), dan False Negative (FN) [20].

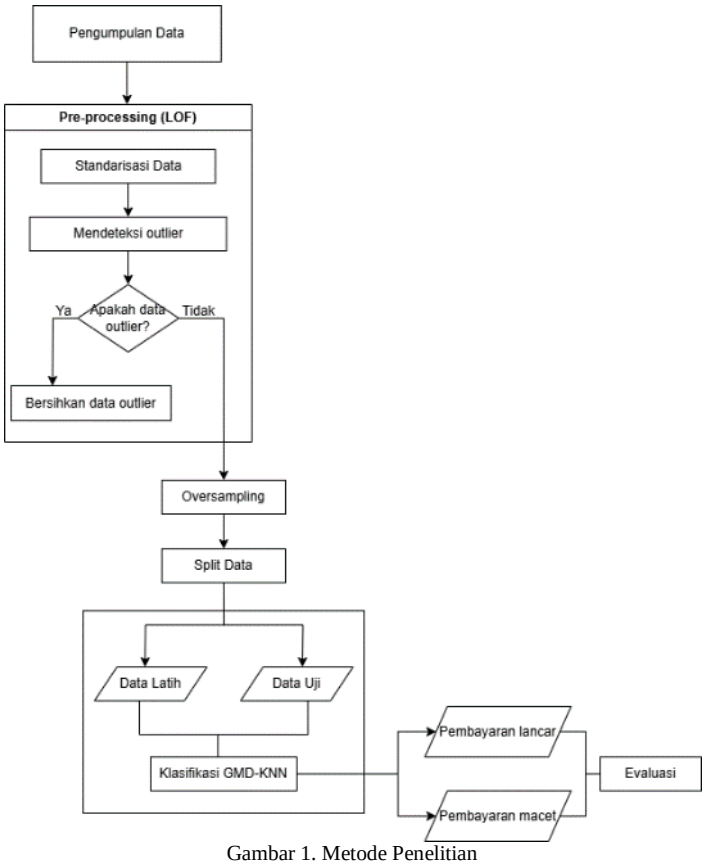
Tabel 1. Gambaran Confusion Matrix

Klasifikasi		**Predicted Class**	
		Negatif	**Positif**
Actual Class	Negatif	True Negatif	False Positif
	Positif	False Negatif	True Positif

Berdasarkan Tabel 1, nilai True Negative (TN) adalah jumlah data tidak berbahaya yang terdeteksi dengan benar, sedangkan False Positive (FP) adalah data tidak berbahaya yang salah terdeteksi sebagai berbahaya. Sementara itu, True Positive (TP) adalah data berbahaya yang terdeteksi dengan benar. False Negative (FN) adalah kebalikan dari True Positive, yaitu data berbahaya yang salah terdeteksi sebagai tidak berbahaya [21].

3. METODE PENELITIAN

Pada penelitian ini terdapat beberapa tahapan yang ditunjukkan pada Gambar 1.



Gambar 1. Metode Penelitian

3.1. Pengumpulan Data

Pengumpulan data ini bertujuan untuk mengumpulkan informasi yang relevan atau yang memiliki hubungan dengan penelitian yang dilakukan. Pengumpulan data ini berupa atribut-atribut dan memiliki kelas atau label untuk dilakukan proses klasifikasi.

3.2. *Preprocessing*

Proses pengolahan data atau *pre-processing* bertujuan untuk membersihkan, mengubah, mengurangi dan mempersiapkan data agar dapat digunakan. Dalam tahap *preprocessing* ini diawali dengan standarisasi data, deteksi *outlier*, dan membersihkan data *outlier*.

Standarisasi data – digunakan untuk mengubah skala pada data numerik, agar data tersebut memiliki skala yang sama. Pada standarisasi ini dilakukan menggunakan *Z-Score*. Berikut persamaan *Z-Score* dilihat pada persamaan (1).

$$Z = \frac{X - \bar{X}}{S_X} \quad (1)$$

Pada persamaan (1), Z adalah standarisasi, X adalah data sebelum standarisasi, $\bar{X}$ adalah rata-rata atribut X, S_X adalah standar deviasi atribut X.

Deteksi *outlier* – proses implementasi metode *Local Outlier Factor* (LOF) untuk mendeteksi outlier dalam dataset. Dalam *Local Outlier Factor* terdapat beberapa tahapan yaitu menghitung *Reachability Distance* (RD), *Local Reachability Density* (LRD), dan menghitung nilai LOF.

$$reach - dist_k(p,o) = \max\big(k - dist(o), dist(p,o)\big) \quad (2)$$

Pada persamaan (2), $reach - dist_k(p,o)$ merupakan *Reachability Distance* (RD) atau jarak jangkauan dari objek p sampai o. $k - dist(o)$ adalah nilai jarak k dari objek o, sedangkan $dist(p,o)$ adalah jarak dari objek p sampai o.

$$lrd_{MinPts(p)} = \frac{1}{\left(\frac{reachdist_{MinPts(p,o)}}{N_{MinPts(p)}}\right)} \quad (3)$$

Langkah selanjutnya untuk menghitung *Local Raechability Densitiy* (LRD) atau jarak kepadatan lokal, dimana $reachdist_{MinPts(p,o)}$ adalah jarak jangkauan dari objek p sampai o, $N_{MinPts(p)}$ adalah jumlah tetangga atau nilai minPts objek p.

$$LOF_{MinPts(p)} = \frac{\left(\frac{lrd_{minPts(o)}}{lrd_{minPts(p)}}\right)}{N_{minPts(p)}} \quad (4)$$

Pada persamaan (4), $LOF_{MinPts(p)}$ merupakan nilai outlier suatu objek p. $lrd_{minPts(o)}$ adalah kepadatan lokal dari objek o, $lrd_{minPts(p)}$ adalah kepadatan lokal dari objek p, $N_{minPts(p)}$ adalah jumlah tetangga atau nilai minPts objek p.

Membersihkan *outlier* – proses penghapusan data berupa *outlier* yang telah diidentifikasi menggunakan *Local Outlier Factor* (LOF).

3.3. *Oversampling*

Ketika salah satu kelas dalam dataset memiliki jumlah sampel yang jauh lebih sedikit dibandingkan kelas lainnya, *oversampling* adalah strategi yang digunakan dalam analisis data untuk menyeimbangkan ketidakseimbangan ini. *Oversampling* menambahkan lebih banyak sampel ke kelas minoritas untuk mencapai representasi yang lebih seimbang antara kelas minoritas dan mayoritas. *Oversampling* ini menggunakan Teknik *Synthetic Minority Over-sampling Technique* (SMOTE).

3.4. *Split Data*

Dataset dibagi menjadi 2 kategori utama, yaitu data latih dan data uji. Dengan membagi data menjadi set pelatihan dan pengujian, dapat lebih mudah memahami seberapa efektif kinerja model mengklasifikasikan data baru..

3.5. Klasifikasi dan Evaluasi

Pada tahap ini dilakukan proses klasifikasi dengan menggunakan *Generalized Mean Distance K-Nearest Neighbor* (GMD-KNN).

$$U_j^{NN} = \left\{U_{i,j}^{NN} \in R^d\right\}_{i=1}^{k} \quad (5)$$

Pada persamaan (2), U_j^{NN} merupakan nilai k *local mean vector* dari masing-masing kelas menggunakan i ($1 \le i \le k$), dimana $U_{i,j}^{NN}$ adalah jarak Euclid tiap kelas, dan k adalah jumlah tetangga terdekat.

$$g(x, U_{r.j}^{NN}) = \left(\frac{1}{r}\sum_{i=1}^{r}\left(d\left(x, u_{i.j}^{NN}\right)^p\right)\right)^{\frac{1}{p}} \quad (6)$$

Langkah selanjutnya untuk menghitung nilai *generalized mean distance* pada tiap kelas menggunakan r ($1 \le r \le k$), dimana

$$G(x, g^j) = \left(\frac{1}{k}\sum_{r=1}^{k}\left(g\left(x, U_{r,j}^{NN}\right)^p\right)\right)^{\frac{1}{p}} \quad (7)$$

Pada persamaan (7), $G(x, g^j)$ merupakan nilai *nested generalized mean distance* yang digunakan untuk memilih kelas dengan nilai terkecil, dimana $g(x, U_{r,j}^{NN})$ merupakan nilai *generalized mean distance* dari tiap kelas.

Evaluasi dilakukan untuk mengukur kinerja model yang dihasilkan dari proses klasifikasi. Evaluasi dilakukan menggunakan *Confusion Matrix* untuk menghitung nilai *Accuracy*, *Precision*, *Recall*, *F1-Score* dan *G-Mean.*

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (8)$$

Pada persamaan (8), digunakan untuk menghitung *Accuracy*, dimana TP merupakan kelas prediksi benar bernilai positif, TN adalah kelas terprediksi netral, FP sebagai kelas positif terprediksi negatif, dan FN adalah kelas negatiif terprediksi positif.

$$Precision = \frac{TP}{(TP+FP)} \quad (9)$$

Pada persamaan (9), digunakan untuk menghitung *Precision*, yang berguna untuk mengukur proporsi kasus positif yang diprediksi dengan benar dibandingkan dengan data aktual.

$$Recall = \frac{TP}{(TP+FN)} \quad (10)$$

Pada persamaan (10), digunakan untuk menghitung *Recall. Recall* merupakan jumlah kasus positif yang sebenarnya dan dikategorikan sebagai positif.

$$F1 - Score = 2\frac{(Precision \times Recall)}{(Precision + Recall)} \quad (11)$$

Pada persamaan (11), digunakan untuk menghitung *F1-Score* adalah ukuran kinerja model yang menggabungkan presisi (precision) dan panggilan (recall) menjadi satu nilai tunggal untuk memberikan gambaran keseimbangan antara kedua metrik tersebut.

$$G - Mean = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{FP+TN}} \quad (12)$$

Pada persamaan (12), digunakan untuk menghitung *G-Mean. G-Mean* merupakan adalah metrik evaluasi yang digunakan untuk menilai kinerja model klasifikasi, terutama dalam situasi di mana ada ketidakseimbangan kelas (imbalance).

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data yang diambil dari UCI Machine Learning Repository. Data tersebut berupa data bulan April - September 2005 yang berjumlah 30000. Data ini terdapat 24 atribut dengan 1 atribut sebagai kelas. Berikut penjelasan dari 24 atribut dijelaskan pada Tabel 2.

Tabel 2. Atribut Data

Atribut	Keterangan
LIMIT_BAL	Jumlah kredit yang diberikan
Gender	Jenis kelamin
Education	Pendidikan
Marriage	Status pernikahan
Age	Usia
PAY_1	Riwayat pembayaran terakhir, status pembayaran pada bulan September
PAY_2	Riwayat pembayaran terakhir, status pembayaran pada bulan Agustus
PAY_3	Riwayat pembayaran terakhir, status pembayaran pada bulan Juli
PAY_4	Riwayat pembayaran terakhir, status pembayaran pada bulan Juni
PAY_5	Riwayat pembayaran terakhir, status pembayaran pada bulan Mei
PAY_6	Riwayat pembayaran terakhir, status pembayaran pada bulan April
BILL_AMT1	Jumlah tagihan bulan September
BILL_AMT2	Jumlah tagihan bulan Agustus
BILL_AMT3	Jumlah tagihan bulan Juli
BILL_AMT4	Jumlah tagihan bulan Juni
BILL_AMT5	Jumlah tagihan bulan Mei
BILL_AMT6	Jumlah tagihan bulan April
PAY_AMT1	Jumlah yang dibayarkan pada bulan September
PAY_AMT2	Jumlah yang dibayarkan pada bulan Agustus
PAY_AMT3	Jumlah yang dibayarkan pada bulan Juli
PAY_AMT4	Jumlah yang dibayarkan pada bulan Juni
PAY_AMT5	Jumlah yang dibayarkan pada bulan Mei
PAY_AMT6	Jumlah yang dibayarkan pada bulan April
Default Payment of Month	Status pembayaran (kelas)

Pada Tabel 2, merupakan keseluruhan atribut data sebanyak 24 dengan 1 sebagai kelas yaitu *Default Payment of Month.*

4.2. *Preprocessing*

Tujuan dari *preprocessing* data adalah agar data transaksi kartu kredit siap digunakan dalam proses klasifikasi dengan melalui tahap pembersihan. Dalam tahap pre-processing ini, data akan melalui beberapa tahapan yaitu standarisasi data, mendeteksi outlier dengan LOF, dan membersihkan data outlier.

Standarisasi Data

Pada standarisasi ini dilakukan menggunakan Z-Score. Data yang telah distandarisasikan akan memiliki rata-rata 0 dan standar deviasi 1. Perhitungan standarisasi dapat dilakukan pada tiap baris dengan menentukan rata-rata dan standar deviasi tiap atribut. Hasil standarisasi dengan menggunakan Z-Score pada setiap atribut seperti pada Tabel 2.

Tabel 3. Hasil Standarisasi Data

ID	LIMIT_ BAL	PAY_1	…	PAY_ AMT1	Default Payment of Month
1	-1,1367	1,7945	…	-0,3419	1
2	-0,3659	-0,8749	…	-0,3419	1
3	-0,5971	0,0148	…	-0,2502	0
4	-0,9054	0,0148	…	-0,2211	0
5	-0,9054	-0,8749	…	-0,2211	0
…	…	…	…	…	…
29996	0,4047	0,0148	…	0,1712	0
29997	-0,1347	-0,8749	…	-0,2310	0
29998	-1,0596	3,5742	…	-0,3419	1
29999	-0,6742	0,9046	…	4,8442	1
30000	-0,9054	0,0146	…	-0,2164	1

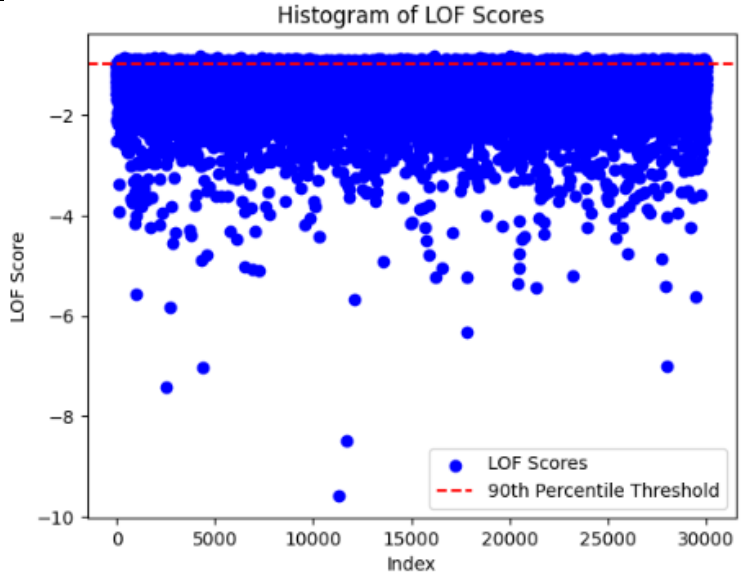
Berdasarkan Tabel 3, dilakukan standarisasi data agar memiliki skala yang sama. Standarisasi dilakukan pada semua atribut kecuali *Default Payment of month* yang merupakan kelas.

Deteksi *outlier*

Pada tahap ini, dilakukan proses deteksi *outlier* menggunakan *Local Outlier Factor* (LOF) terhadap data yang telah distandarisasi. Setelah mendapatkan nilai LOF, kemudian menentukan nilai ambang batas. Berikut histogram nilai ambang batas dengan 90 percentile dapat diliat pada Gambar 2.

Berdasarkan Gambar 2, terdapat titik berwarna biru yang merupakan skor LOF dan garis merah sebagai ambang batas yang berada pada nilai -1,0. Pada gambar tersebut skor LOF di bawah -1,0 sampai -10 merupakan data normal. Sehingga nilai LOF di atas ambang batas atau -1,0 dianggap sebagai outlier.

DOI: https://doi.org/10.52436/1.jutif.2024.5.5.2341
p-ISSN: 2723-3863
e-ISSN: 2723-3871



Gambar 2. Histrogram Nilai Ambang Batas

Selanjutnya mengidentifikasi outlier berdasarkan nilai threshold. Jika LOF Score lebih besar dari threshold yaitu -1,0, maka data tersebut merupakan outlier. Outlier yang teridentifikasi sebanyak 4898 data dapat dilihat pada Tabel 3.

Tabel 4. Data Outlier

ID	LIMIT_BAL	PAY_1	…	LOF Score	Outlier
1	70000	1	…	-0,9821	True
2	20000	0	…	-0,9796	True
3	120000	-1	…	-0,9905	True
4	70000	2	…	-0,9754	True
5	100000	0	…	-0,9684	True
…	…	…	…	…	…
4894	360000	-1	…	-0,9611	True
4895	130000	0	…	-0,9937	True
4896	80000	2	…	-0,9645	True
4897	220000	0	…	-0,9743	True
4898	50000	0	…	-0,9802	True

Pada Tabel 4, data *outlier* teridentifikasi sebanyak 4898. Terdapat kolom *LOF Score* bernilai lebih dari -0,1, dan kolom *Outlier* bernilai *True* yang artinya data tersebut merupakan *outlier*.

Membersihkan *Outlier*

Setelah mendeteksi outlier, kemudian dilakukan penghapusan data yang berupa outlier terdapat pada Tabel 3, hingga data normal memiliki sebanyak 25102 data dapat dilihat pada Tabel 4.

Tabel 5. Data Normal

ID	LIMIT_BAL	PAY_1	…	LOF Score	Outlier
1	20000	2	…	-1,1306	False
2	1200000	-1	…	-1,6778	False
3	90000	0	…	-1,3228	False
4	50000	0	…	-1,0324	False
5	50000	-1	…	-1,5002	False
…	…	…	…	…	…
25098	10000	0	…	-1,2998	False
25099	100000	0	…	-1,0182	False
25100	150000	-1	…	-1,2430	False
25101	30000	4	…	-1,4995	False
25102	80000	1	…	-1,4966	False

Pada Tabel 5, data normal yang didapati sebanyak 25102. Terdapat kolom LOF Score dengan nilai di bawah -0,1, dan kolom *Outlier* berisi *False* yang artinya data tersebut bukan *outlier*.

4.3. *Oversampling*

Pada tahap ini, dilakukan oversampling untuk menyeimbangkan data. Berikut merupakan jumlah data dari setiap label yang sudah dilakukan teknik *oversampling*, dapat dilihat pada Tabel 5.

Tabel 6. Hasil *Oversampling*

	0	1
Sebelum	19656	5446
Sesudah	19656	19656

Dari Tabel 6, hasil *Oversampling* meningkat pada kelas 1. Dengan jumlah awal sebanyak 5446 menjadi 19656. Sehingga jumlah kelas 0 dan 1 mendapatkan keseimbangan data.

4.4. Splitting

Pada pengujian model ini, eksperimen akan dilakukan menggunakan tiga skenario pengujian, yaitu: skenario pertama dengan membagi data menjadi 60% data latih dan 40% data uji, skenario kedua dengan 70% data latih dan 30% data uji, serta skenario ketiga dengan 80% data latih dan 20% data uji. Ketiga skenario ini dirancang untuk mengamati stabilitas kinerja model pada berbagai pembagian data. Dengan membagi skenario pengujian, dapat diidentifikasi model yang memiliki kinerja terbaik dan memberikan prediksi atau hasil yang akurat. Dalam kasus ini, setiap pembagian data telah melalui proses oversampling, dapat dilihat pada Tabel 6.

Tabel 7. Splitting Data

Rasio	Data Latih	Data Uji	Total Data
60% : 40%	23587	15725	
70% : 30%	27518	11794	39312
80% : 20%	31449	7863	

Berdasarkan Tabel 7, dilakukan *splitting data* sebanyak 3 rasio, ini dilakukan untuk mengetahui hasil pengujian dengan banyaknya rasio.

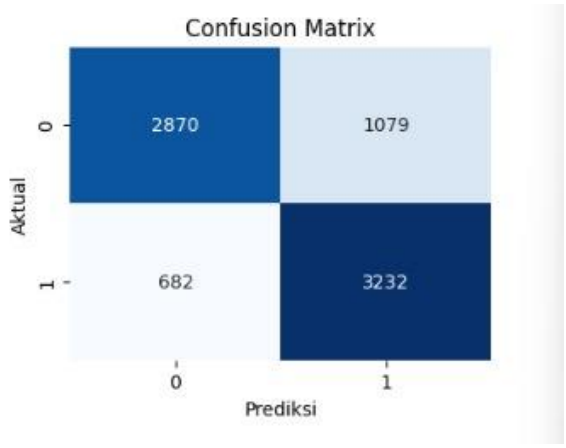
4.5. Klasifikasi dan Evaluasi

Pada hasil pengujian model GMD-KNN dengan menggunakan confusion matrix, akurasi tertinggi dicapai oleh k=5 pada rasio 80% data latih dan 20% data uji. Pada pengujian ini dilakukan dengan menggunakan parameter p=1 pada tiap masing-masing pengujian.

Tabel 8. Perbandingan Evaluasi Pengujian

Rasio	nilai k	Accuracy	Precision	Recall	F1-Score	G-Mean
60% : 40%	k=5	76,21%	74,28%	80,85%	77,43%	76,02%
	k=9	76,42%	75,26%	79,37%	79,37%	76,33%
	k=15	75,94%	75,63%	77,19%	76,40%	75,91%
70% : 30%	k=5	76,95%	74,96%	81,50%	78,09%	76,77%
	k=9	76,61%	75,29%	79,82%	77,49%	76,52%
	k=15	76,51%	75,90%	78,26%	77,06%	77,47%
80% : 20%	k=5	77,60%	74,97%	82,57%	78,58%	77,48%
	k=9	77,29%	75,41%	80,68%	77,96%	77,22%
	k=15	77,45%	76,15%	79,63%	77,86%	77,43%

Berdasarkan Tabel 8, dilakukan pengujian pada rasio 60% : 40%, 70% :30%, 80% : 20%, dengan nilai k sebanyak 5, 9, dan 15 pada masing-masing rasio dengan parameter p=1. Nilai k yang lebih rendah cenderung memberikan recall yang lebih tinggi, namun nilai k yang lebih tinggi memberikan presisi yang sedikit lebih baik.



Gambar 3. Confusion Matrix rasio 80% : 20% k=5

Pada Gambar 3, merupakan hasil *Confusion Matrix* dari rasio data 80% : 20% sebanyak k=5. Dengan TP bernilai 3232, FP sebanyak 1079, FN sebanyak 682, dan TN sebanyak 2870.

Accuracy

Berikut persamaan yang digunakan untuk menghitung *Accuracy* menggunakan persamaan (8).

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$
$$Accuracy = \frac{(3232+2870)}{(3232+2870+1079+682)}$$
$$Accuracy = \frac{6102}{7863} = 0{,}7760$$

Precision

Berikut persamaan yang digunakan untuk menghitunng *Precision* menggunakan persamaan (9).

$$Precision = \frac{TP}{(TP+FP)}$$
$$Precision = \frac{3232}{(3232+1079)}$$
$$Precision = \frac{3232}{4311} = 0{,}7497$$

Recall

Berikut persamaan yang digunakan untuk menghitung *Recall* menggunakan persamaan (10).

$$Recall = \frac{TP}{(TP+FN)}$$
$$Recall = \frac{3232}{(3232+682)}$$
$$Recall = \frac{3232}{3914} = 0{,}8257$$

F1-Score

Berikut persamaan yang digunakan untuk menghitung *F1-Score* menggunakan persamaan (11).

$$F1-Score = 2\frac{(Precision \times Recall)}{(Precision+Recall)}$$
$$F1-Score = 2\frac{(0{,}7497 \times 0{,}8257)}{(0{,}7497+0{,}8257)}$$
$$F1-Score = 2\frac{0{,}6190}{1{,}5754} = 0{,}7858$$

G-Mean

Berikut persamaan yang digunakan untuk menghitung *G-Mean* menggunakan persamaan (12).

$$G-Mean = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{FP+TN}}$$
$$G-Mean = \sqrt{\frac{3232}{3232+682} \times \frac{2870}{2870+1079}}$$
$$G-Mean = \sqrt{0{,}8257 \times 0{,}7268} = 0{,}7748.$$

5. DISKUSI

Sugriyono dan Maria Ulfah Siregar melakukan deteksi dan menghapus outlier untuk meningkatkan akurasi klasifikasi, menunjukkan bahwa akurasi klasifikasi algoritma K-NN dipengaruhi oleh pencilan dalam dataset. Hasilnya pada prapemrosesan menggunakan metode K-means dan Euclidean untuk klasifikasi dengan penanganan outlier menghasilkan akurasi hasil klasifikasi sebesar 98,42 % (meningkat 26,14 %).

Berdasarkan penjelasan dan hasil dalam penelitian ini jika deteksi *outlier* dengan LOF dan pembersihan data *outlier* secara signifikan meningkatkan kualitas data yang digunakan untuk pelatihan dan pengujian. Teknik oversampling menggunakan SMOTE berhasil menyeimbangkan jumlah data antara kelas mayoritas dan minoritas, yang berdampak pada peningkatan akurasi dan stabilitas model.

Hasil pada penelitian ini, menunjukkan peningkatan kinerja dengan bertambahnya jumlah data latih, menghasilkan akurasi 77,60% pada rasio 80% data latih dan 20% data uji serta k=5 dengan evaluasi *Confusion Matrix* yang dapat dilihat pada Gambar 3. Penggunaan algoritma GMD-KNN dengan penyesuaian parameter k dan p, dapat meningkatkan akurasi dan keandalan model dalam mengidentifikasi potensi kelancaran pembayaran kartu kredit.

6. KESIMPULAN

Berdasarkan penelitian yang sudah dilakukan, pada klasifikasi data kartu kredit menggunakan *Generalized Mean Distance K-Nearest Neighbor* dengan *Local Outlier Factor* sebagai penanganan outlier. Metode *Local Outlier Factor* (LOF) terbukti efektif dalam mendeteksi *outlier* dalam data transaksi kartu kredit. Dengan menghapus data yang terdeteksi sebagai *outlier*, kualitas data meningkat, yang berdampak positif pada performa model klasifikasi. Hasil pada model menunjukkan peningkatan kinerja seiring dengan peningkatan jumlah data latih yaitu menghasilkan akurasi 77,60% pada rasio 80% data latih dan 20% data uji pada k=5. Namun, nilai k dan p yang optimal bervariasi berdasarkan rasio pembagian data. Kedepannya penulis akan melakukan hal mendalam tentang klasifikasi data dengan penanganan outlier dalam penentuan parameter k dan p yang lebih baik.

DAFTAR PUSTAKA

[1] P. A. Ariawan, "Optimasi Pengelompokan Data Pada Metode K-means dengan Analisis Outlier," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 5, no. 2, pp. 88–95, Sep. 2019, doi: 10.25077/teknosi.v5i2.2019.88-95.

[2] Juozas Auskalnis, Nerijus Paulauskas, and Algirdas Baskys, "Application of Local Outlier Factor Algorithm to Detect Anomalies in Computer Network," *Elektronika IR Elektrotechnika, ISSN 1392-1215*, vol. 24, Dec. 2018, doi: 10.1109/CISDA.2009.5356528.

[3] A. Smiti, "A critical overview of outlier detection methods," *Computer Science Review*, vol. 38. Elsevier Ireland Ltd, Nov. 01, 2020. doi: 10.1016/j.cosrev.2020.100306.

[4] D. Zou *et al.*, "Outlier detection and data filling based on KNN and LOF for power transformer operation data classification," *Energy Reports*, vol. 9, pp. 698–711, Sep. 2023, doi: 10.1016/j.egyr.2023.04.094.

[5] S. P. Maniraj, A. Saini, S. Deep Sarkar, and S. Ahmed, "Credit Card Fraud Detection using Machine Learning and Data Science," *International Journal of Engineering Research & Technology (IJERT)*, 2019.

[6] E. Strelcenia and S. Prakoonwit, "Improving Classification Performance in Credit Card Fraud Detection by Using New Data Augmentation," *AI (Switzerland)*, vol. 4, no. 1, pp. 172–198, Mar. 2023, doi: 10.3390/ai4010008.

[7] S. Sugidamayatno and D. Lelono, "Outlier Detection Credit Card Transactions Using Local Outlier Factor Algorithm (LOF)," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 13, no. 4, p. 409, Oct. 2019, doi: 10.22146/ijccs.46561.

[8] E. H. Yulianti, O. Soesanto, and Y. Sukmawaty, "Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit," *JOMTA Journal of Mathematics: Theory and Applications*, vol. 4, no. 1, 2022.

[9] Rishikeshan, S. K. Sakala, S. Prasath, and M. Anitha, "Credit Card Fraud Detection Using Isolation Forest and Local Outlier Factor," *International Journal of Scientific Research in Engineering and Management (IJSREM)* , vol. 06, no. 06, Jun. 2022, doi: 10.55041/ijsrem14371.

[10] H. Xu, L. Zhang, P. Li, and F. Zhu, "Outlier detection algorithm based on k-nearest neighbors-local outlier factor," *J Algorithm Comput Technol*, vol. 16, Mar. 2022, doi: 10.1177/17483026221078111.

[11] J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao, and H. Yang, "A generalized mean distance-based k-nearest neighbor classifier," *Expert Syst Appl*, vol. 115, pp. 356–372, Jan. 2019, doi: 10.1016/j.eswa.2018.08.021.

[12] S. Sugriyono and M. U. Siregar, "Preprocessing kNN algorithm classification using K-means and distance matrix with students' academic performance dataset," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 4, Oct. 2020, doi: 10.14710/jtsiskom.2020.13874.

[13] D. Rosita, dan Syamsuddin Mallala, S. Informasi, S. Widya Cipta Dharma, T. Informatika, and P. Korespondensi, "Komparasi Data Mining Naive Bayes dan Neural Network Memprediksi Masa Studi Mahasiswa S1," vol. 7, no. 3, pp. 443–452,

2020, doi: 10.25126/jtiik.202072093.

[14] D. P. Utomo and M. Mesran, "Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut Pada Data Set Penyakit Jantung," *Jurnal Media Informatika Budidarma*, vol. 4, no. 2, p. 437, Apr. 2020, doi: 10.30865/mib.v4i2.2080.

[15] F. Wafiyah, N. Hidayat, and R. S. Perdana, "Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 10, 2017.

[16] S. Sumayah, F. Sembiring, and W. Jatmiko, "Analysis Of Sentiment Of Indonesian Community On Metaverse Using Support Vector Machine Algorithm," *Jurnal Teknik Informatika (JUTIF)*, vol. 4, no. 1, 2023, doi: 10.20884/1.jutif.2023.4.1.417.

[17] S. Chawla and A. Gionis, "k-means-: A unified approach to clustering and outlier detection".

[18] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-10358-x.

[19] N. Triadha Pitaloka, "PCOS Disease Classification Using Feature Selection Rfecv And Eda With Knn Algorithm Method," *Jurnal Teknik Informatika (JUTIF)*, vol. 4, no. 4, pp. 693–701, 2023, doi: 10.52436/1.jutif.2023.4.4.693.

[20] F. Rahmad, Y. Suryanto, and K. Ramli, "Performance Comparison of Anti-Spam Technology Using Confusion Matrix Classification," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, Aug. 2020.

[21] S. Tiwari, V. Sapra, and A. Jain, "Heartbeat sound classification using Mel-frequency cepstral coefficients and deep convolutional neural network," in *Advances in Computational Techniques for Biomedical Image Analysis: Methods and Applications*, Elsevier, 2020, pp. 115–131.