ISSN: 2339-2541
JURNAL GAUSSIAN, Volume 8, Nomor 3, Tahun 2019, Halaman 366-376
Online di: http://ejournal3.undip.ac.id/index.php/gaussian

# PERBANDINGAN KINERJA *MUTUAL K-NEAREST NEIGHBOR* (MKNN) DAN *K-NEAREST NEIGHBOR* (KNN) DALAM ANALISIS KLASIFIKASI KELAYAKAN KREDIT

**Annisa Sugesti[1], Moch. Abdul Mukid[2], Tarno[3]**
[1,2,3] Departemen Statistika FSM Universitas Diponegoro
mamukid@yahoo.com

## ABSTRACT

Credit feasibility analysis is important for lenders to avoid the risk among the increasement of credit applications. This analysis can be carried out by the classification technique. Classification technique used in this research is instance-based classification. These techniques tend to be simple, but are very dependent on the determination of K values. K is number of nearest neighbor considered for class classification of new data. A small value of K is very sensitive to outliers. This weakness can be overcome using an algorithm that is able to handle outliers, one of them is Mutual K-Nearest Neighbor (MKNN). MKNN removes outliers first, then predicts new observation classes based on the majority class of their mutual nearest neighbors. The algorithm will be compared with KNN without outliers. The model is evaluated by 10-fold cross validation and the classification performance is measured by Gemoetric-Mean of sensitivity and specificity. Based on the analysis the optimal value of K is 9 for MKNN and 3 for KNN, with the highest G-Mean produced by KNN is equal to 0.718, meanwhile G-Mean produced by MKNN is 0.702. The best alternative to classifying credit feasibility in this study is K-Nearest Neighbor (KNN) algorithm with K=3.
Keywords: Classification, Credit, MKNN, KNN, G-Mean.

## 1. PENDAHULUAN

Kredit merupakan aset terbesar yang dikelola bank dan juga merupakan kontributor yang paling dominan terhadap pendapatan bank. Pemberian kredit di sisi lain mengandung risiko yang dapat mempengaruhi kondisi keuangan pihak bank dan mengharuskan bank untuk berhati-hati dalam melakukan analisis apakah permohonan yang diajukan calon debitur layak untuk ditolak atau disetujui. Analisis tersebut dapat dilakukan dengan salah satu metode dalam data mining yaitu klasifikasi,.

Menurut Tan dkk (2006) klasifikasi adalah sebuah proses untuk menemukan sebuah model yang menjelaskan dan membedakan konsep atau kelas data dengan tujuan memperkirakan kelas dari suatu objek yang kelasnya tidak diketahui. Permasalahan yang sering ditemui dalam klasifikasi yaitu masalah ketidakseimbangan data, yaitu ketika salah satu kelas memiliki jumlah yang jauh lebih besar dibanding kelas lainnya. Hal tersebut dikhawatirkan menyebabkan menurunnya kinerja klasifikasi pada kelas minoritas, oleh karena itu dibutuhkan solusi untuk mengatasi masalah tersebut salah satunya dengan melakukan *undersampling*. Salah satu teknik klasifikasi yang sering digunakan adalah klasifikasi berbasis *nearest neighbor* karena algoritmanya yang cukup sederhana, namun kinerja teknik tersebut sangat bergantung pada pemilihan nilai K. K yang terlalu besar mengakibatkan tetangga terdekat yang terpilih terlalu banyak dari kelas lain yang sebenarnya tidak relevan karena jarak yang terlalu jauh sedangkan K yang terlalu rendah akan berakibat pada hasil prediksi yang sensitif terhadap keberadaan *outlier* (Prasetyo, 2014), maka dari itu diperlukan sebuah metode klasifikasi yang mampu mengatasi masalah tersebut sehingga hasil klasifikasi yang diperoleh menjadi lebih akurat.

*Mutual KNN Classifier* (MKNN) adalah salah satu *lazy learning algorithm* yang merupakan pengembangan dari *K-Nearest Neighbor Classifier* (KNN). MKNN berbeda

dengan metode KNN, ia pertama-tama menghilangkan *outlier* dengan menggunakan konsep tetangga mutual terdekat (MNN), kemudian membuat prediksi untuk amatan baru berdasarkan MNN-nya. Keuntungannya adalah bahwa hasil prediksi lebih dapat dipercaya karena tetangga "palsu" atau *outlier* telah dikeluarkan sebelum prosedur prediksi (Liu & Zhang, 2012). Teknik validasi model yang digunakan adalah *10-Fold Cross Validation* dan ukuran kinerja klasifiktor yang digunakan adalah sensitifitas, spesifisitas, dan geometric-mean.

## 2. TINJAUAN PUSTAKA

### 2.1. Kredit

UU Perbankan Nomor 10 Tahun 1998 Pasal 1 tentang kredit menjelaskan bahwa kredit adalah Penyediaan uang atau tagihan yang dapat dipersamakan dengan itu, berdasarkan persetujuan atau kesepakatan pinjam meminjam antar bank dengan pihak lain yang mewajibkan pihak peminjam melunasi utangnya setelah jangka waktu tertentu dengan pemberian bunga. Menurut Hermansyah (2008) persetujuan terhadap suatu permohonan kredit dilakukan dengan berpedoman pada Formula 5C, yaitu *Character, Capacity, Capital, Collateral,*dan *Condition of Economy.*

### 2.2. Klasifikasi

Klasifikasi adalah sebuah proses untuk menemukan sebuah model yang menjelaskan dan membedakan konsep atau kelas data dengan tujuan memperkirakan kelas dari suatu objek yang kelasnya tidak diketahui (Tan dkk, 2006).Klasifikasi berbasis *nearest neighbor* dilakukan berdasarkan jarak antara data *testing* dengan data *training* yang dapat dihitung salah satunya dengan jarak Euclidean dengan yang didefinisikan dalam persamaan 1 (Han & Kamber, 2006).

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{p} (diff_{(\mathbf{x}_{il}, \mathbf{x}_{jl})})^2} \quad (1)$$

$i = 1, \dots, n$ ; $j = 1, \dots, n$; $l = 1, \dots, p$

$d(\mathbf{x}_i, \mathbf{x}_j)$ : jarak *Euclid* obyek ke-*i* dan obyek ke-*j* $p$: banyaknya variabel bebas

$diff_{(\mathbf{x}_{il}, \mathbf{x}_{jl})}$ : nilai ketidakmiripan obyek ke-*i* dan objek ke-*j* pada peubah ke-*l*

Menurut Prasetyo (2014), penghitungan nilai ketidaksamaan berdasarkan tipe data untuk tiap variabel dapat diringkas seperti pada Tabel 1.

Tabel 1. Ketidakmiripan Dua Data dengan Satu Atribut

| Tipe Atribut | Formula Jarak |
|---|---|
| Nominal | $diff_{(\mathbf{x}_{i1}, \mathbf{x}_{j1})} = \begin{cases} 0 & Jika\ \mathbf{x}_{i1} = \mathbf{x}_{j1} \\ 1 & Jika\ \mathbf{x}_{i1} \neq \mathbf{x}_{j1} \end{cases}$ |
| Ordinal | $diff_{(\mathbf{x}_{i1}, \mathbf{x}_{j1})} = \frac{\left\| \mathbf{x}_{i1} - \mathbf{x}_{j1} \right\|}{(q-1)}$ <br> $q$ adalah banyaknya pengkategorian dalam x |
| Interval atau Rasio | $diff_{(\mathbf{x}_{i1}, \mathbf{x}_{j1})} = \left\| \mathbf{x}_{i1} - \mathbf{x}_{j1} \right\|$ |

### 2.3. Pemrosesan Awal Data

#### 2.3.1. Deteksi *Missing Value*

Hair (1995) juga mengemukakan bahwa apabila persentase data *missing value* melebihi 30%, maka data boleh dihapus sedangkan jika persentase data *missing value* kurang 30%, maka data missing diimputasi dengan nilai mean jika data kuantitatif dan modus jika data kualitatif.

### 2.3.2. Asumsi Multikolinearitas

Multikolinearitas adalah adanya hubungan linear atau korelasi yang tinggi antar variabel. Pengujian multikolinearitas dilihat dari besaran VIF (Variance Inflation Factor). Nilai *cut off* yang umum dipakai untuk menunjukkan adanya multikolinearitas adalah nilai VIF ≥ 10. Nilai VIF diperoleh dengan persamaan berikut :

$$VIF = \frac{1}{1-R^2} \quad (2)$$

### 2.3.3. Pemilihan Fitur Berbasis Statistik

Pemilihan fitur bertujuan untuk membuang fitur dengan kemampuan diskriminasi yang buruk dan mempertahankan fitur dengan kemampuan diskriminasi yang baik terhadap kelas sehingga mampu mengurangi kompleksitas model dan waktu komputasi. Pemilihan fitur dapat dilakukan dengan uji Mann-Whitney untuk data kontinu dan uji independensi chi-square untuk data kategorik.

a) Uji Mann-Whitney

**Hipotesis**

$H_0 : M_x = M_y$

$H_1 : M_x \neq M_y$

**Statistik Uji**

$$T = S - \frac{n_1(n_1+1)}{2} \quad (3)$$

$n_1$ : jumlah data populasi1 S : jumlah peringkat data yang berasal dari populasi 1

$n_2$ : jumlah data populasi 2 t : jumlah ties

Jika ada ties (nilai yang sama): $$Z = \frac{T - \frac{(n_1 n_2)}{2}}{\sqrt{\frac{n_1 n_2(n_1+n_2+1)}{12} - \frac{n_1 n_2(\sum t^3 - \sum t)}{12(n_1+n_2)(n_1+n_2-1)}}} \quad (4)$$

**Kaidah Pengambilan Keputusan**

Tolak $H_0$ jika $Z_{hit} > Z_\alpha$ atau signifikansi < α

b) Uji Independensi Chi-Square

**Hipotesis**

$H_0$ : Sifat-sifat obyek tidak saling mempengaruhi (independen)

$H_1$ : Sifat-sifat obyek ada yang mempengaruhi (dependen)

**Statistik Uji**

$$\chi^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \quad (5)$$

$O_{ij}$ : Frekuensi Obyek dengan sifat $B_i$ dan $K_j$ atau ($B_i \cap K_j$)

$E_{ij}$: Frekuensi Harapan obyek dengan sifat $B_i$ dan $K_j$ atau ($B_i \cap K_j$)

n : banyaknya baris
m : banyaknya kolom

$$E_{ij} = \frac{(nK_m)(nB_n)}{N} \quad (6)$$

**Kaidah Pengambilan Keputusan**

Tolak $H_0$ apabila nilai $\chi^2 > \chi^2_{\alpha,(n-1)(m-1)}$ atau nilai signifikansi < α

### 2.3.4. Standarisasi Data Interval

Proses standarisasi dilakukan salah satunya dengan menskalakan jangkauan setiap fitur dalam jangkauan [0,1] menggunakan persamaan 8 (Prasetyo, 2014).

$$\hat{x}_{ik} = \frac{x_{ik} - \min(x_k)}{\max(x_k) - \min(x_k)} \quad (7)$$

$\hat{x}_{ik}$ : nilai setelah distandarisasi
$x_{ik}$ : nilai amatan ke-i pada fitur ke-k
$\min(x_k)$ : nilai minimum dari fitur ke-k
$\max(x_k)$ : nilai maximum dari fitur ke-k

### 2.3.5. *Cluster-based Undersampling*

*Cluster-based undersampling* merupakan metode *undersampling* yang dilakukan berdasarkan analisis cluster. Jumlah data dalam suatu *imbalanced dataset* sebanyak N dengan jumlah data mayor dilambangkan $Size_{MA}$ dan jumlah data minor dilambangkan $Size_{MI}$. Pertama-tama dilakukan pengelompokkan keseluruhan *dataset* ke dalam C buah cluster, kemudian dipilih sampel kelas mayor secara random pada masing-masing cluster dengan jumlah yang ditentukan berdasarkan persamaan 8 (Yen & Lee, 2009).

$$SSize^i_{MA} = (m \times Size_{MI}) \times \frac{Size^i_{MA}/Size^i_{MI}}{\sum_{i=1}^{K} Size^i_{MA}/Size^i_{MI}} \quad (8)$$

$SSize^i_{MA}$ : Jumlah sampel data mayor yang dipilih dalam setiap cluster ke-i
$Size^i_{MA}$ : Jumlah data mayor dalam setiap cluster yang terbentuk
$Size^i_{MI}$ : Jumlah data minor dalam setiap cluster yang terbentuk
$m$ : Rasio data kelas mayor yang diharapkan

Algoritma clustering yang dilakukan dalam penelitian ini adalah *Ensemble C-Modes* dengan algoritma sebagai berikut.

Algoritma 1. Clustering Menggunakan C-Modes

1. Input: *dataset* dengan variabel campuran numerik dan kategorik, nilai C sebagai jumlah cluster yang akan dibentuk.
2. Lakukan proses *splitting* atau pemisahan antara variabel numerik dan kategorik
3. Lakukan clustering dengan algoritma C-Means pada variabel numerik, simpan hasil clustering sebagai variabel $Z_1$.
4. Lakukan clustering dengan algoritma C-Modes pada variabel kategorik, simpan hasil clustering sebagai variabel $Z_2$.
5. Lakukan clustering menggunakan algoritma C-Modes pada *dataset* baru yang berisi variabel $Z_1$ dan $Z_2$ pada setiap amatannya, lalu simpan hasilnya sebagai hasil akhir.

## 2.4. K-Nearest Neighbor

K-Nearest Neighbor (KNN) merupakan suatu algoritma dengan prinsip mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (*neighbor*) terdekatnya dalam data pelatihan.

Algoritma 2. Klasifikasi Menggunakan KNN

**Input :** *Training dataset*, amatan x dan nilai K tetangga terdekat;
**Output :** Prediksi label kelas;
(1) Menghitung jarak antara data *training* dengan data data *testing*
(2) Mengurutkan jarak dari yang terkecil hingga terbesar
(3) Dapatkan K tetangga terdekat x
(4) Menetapkan mayoritas kelas dari K tetangga terdekat sebagai kelas data *testing.*

## 2.5. Algoritma *Mutual K-Nearest Neighbor* untuk Deteksi *Outlier*

Algoritma KNN mencari K tetangga terdekat dari data X, namun belum tentu data-data tetangga tersebut adalah tetangga sebenarnya dari X ( tetangga bayangan). K tetangga terdekat yang didapatkan akan diperiksa kembali apakah X juga K tetangga terdekat dari data-data tetangga yang ditemukan, jika data-data tetangga terdekat tersebut tidak mempunyai K tetangga terdekat berupa X maka X dianggap sebagai *outlier* (Liu & Zhang, 2012). Proses perhitungan jarak menggunakan persamaan 1 tergantung pada tipe atribut *dataset* yang diidentifikasi *outlier*-nya.

Algoritma 3. Penghapusan Outlier dengan Prinsip MKNN

**Input :** *Dataset* D dan nilai *nearest neighbor* K;
**Output :** Reduksi D;
(1) Untuk setiap amatan x pada D, lakukan dua langkah berikut:
  (1.1) Dapatkan MNN dari x, Misal M(x) dengan cara:
    Cari k tetangga terdekat x dari D, misal N(x);
    Untuk tiap tetangga terdekat y $\epsilon$ N(x),
    dapatkan k tetangga terdekat y, N(y) dari D;
  (1.2) Jika x $\epsilon$ N(y) maka masukkan y ke dalam M(x);
    Jika M(x) =null maka hapus x dari D;
    *Otherwise*, biarkan x di D.
(2) Kembalikan D sebagai hasil reduksi.

## 2.6. Algoritma *Mutual K-Nearest Neighbor* untuk Klasifikasi

Algoritma MNN untuk klasifikasi dikenal dengan algoritma *Mutual K-Nearest Neighbor Classifier* (MKNN). Algoritma ini mula-mula membuat himpunan label kelas *C(x),* yang nantinya digunakan untuk memprediksi label *x*, selanjutnya mendapatkan K tetangga terdekat dengan teknik KNN konvensional. MKNN kemudian akan mengidentifikasi tetangga mutual terdekat dari *x* dan menyimpan labelnya, lalu mencari K tetangga terdekatnya dari *training set* untuk tiap-tiap tetangga terdekat *y*. *y* adalah tetangga mutual dari *x* apabila salah satu K terdekatnya adalah *x*, kemudian label *y* akan menjadi kandidat kelas dari *x*. Langkah terakhir adalah menentukan label kelas *C(x)* menggunakan strategi mayoritas (Liu & Zhang, 2012).

Algoritma 4. Klasifikasi dengan MKNN

**Input :** *Training dataset*, amatan x dan nilai K tetangga terdekat;
**Output :** Prediksi label c(x);
(1) Inisialisasi parameter relatif, seperti, C(x)=null;
(2) Dapatkan k tetangga terdekat x, Nk(x) dari *training dataset*;
(3) Untuk setiap data tetangga y $\epsilon$ Nk(x), lakukan dua langkah berikut:
  (3.1) Dapatkan K tetangga terdekat y, Nk(y) dari *training dataset*, di mana amatan x termasuk di dalamnya;
  (3.2) Jika x juga K tetangga terdekat Nk(y), Tambahkan label informasi y ke C(x);
  Label kelas C(x) ditentukan sebagai berikut;
  c(x)=arg max $\sum_{ci\,\epsilon\,\mathrm{C(x)}} I(C_y = c)$, di mana I(.) adalah fungsi indikasi.
(4) Kembalikan c(x) sebagai hasil.

## 2.7. Ukuran Kinerja Klasifikasi

Sensitivitas mengukur proporsi kelas "ditolak" asli yang diprediksi secara benar sebagai "ditolak", sementara spesifisitas mengukur proporsi kelas "disetujui" asli yang diprediksi secara benar sebagai "disetujui". Evaluasi kinerja metode secara keseluruhan dapat dilakukan dengan menggunakan *geometric mean* (G-*mean*) yang merupakan rata-rata geometric dari *sensitivity* dan *specificity* (Kubat & Matwin, 1997)

$$Specificity = \frac{TN}{(TN+FP)} \times 100\% \quad (9)$$

$$Sensitivity = \frac{TP}{(TP+FN)} \times 100\% \quad (10)$$

$$G - Mean = \sqrt{Sensitivity \times Specificity} \quad (11)$$

## 2.8. *K-Fold Cross Validation* Sebagai Metode Evaluasi Klasifikator

*K-fold cross validation* membagi data secara acak menjadi k partisi atau *fold* yang memiliki ukuran yang sama. Setiap partisi berkesempatan satu kali menjadi data *testing* dan k-1 kali menjadi data *training* (Tan *et al*, 2006). Akurasi klasifikasi model diperoleh dengan cara merata-ratakan akurasi dari setiap iterasi.

## 3. METODE PENELITIAN

Data yang digunakan dalam penulisan penelitian ini adalah data sekunder, yaitu data permohonan produk Kredit Tanpa Agunan (KTA) yang diperoleh dari Bank Mandiri Pusat sejumlah 10613 data dengan process date pada tahun 2009-2010. Software statistik yang digunakan adalah MatLab R2015a, SPSS 16.0, R 3.5.2, dan Microsoft Office Excel 2016. Langkah-langkah analisis yang digunakan pada penelitian ini adalah sebagai berikut:

1. Melakukan penanganan *missing value*
2. Melakukan uji multikolinearitas
3. Melakukan seleksi fitur
4. Melakukan *cluster-based undersampling*
5. Melakukan penanganan *outliers*
6. Membagi *dataset* D menjadi 2 bagian, yakni data *training* dan data *testing*
7. Menstandarisasi data numerik

8. Melakukan Klasifikasi dengan KNN dan MKNN
9. Menghitung dan membandingkan ukuran kinerja klasifikator dari KNN dan MKNN

## 4. HASIL DAN PEMBAHASAN

### 4.1. Asumsi Multikolinearitas

Hasil pegujian multikolinearitas dalam penelitian ini ditampilkan pada Tabel 2.

Tabel 2. Hasil Multikolinearitas

| Variabel | R Square | VIF | Variabel | R Square | VIF |
|---|---|---|---|---|---|
| $X_1$ | 0,538 | 2,165 | $X_8$ | 0,358 | 1,558 |
| $X_2$ | 0,229 | 1,297 | $X_9$ | 0,211 | 1,267 |
| $X_3$ | 0,312 | 1,454 | $X_{10}$ | 0,016 | 1,016 |
| $X_4$ | 0,369 | 1,585 | $X_{11}$ | 0,237 | 1,311 |
| $X_5$ | 0,38 | 1,613 | $X_{12}$ | 0,529 | 2,123 |
| $X_6$ | 0,349 | 1,536 | $X_{13}$ | 0,562 | 2,283 |
| $X_7$ | 0,469 | 1,883 | | | |

Tabel 2 menunjukkan bahwa nilai VIF dari seluruh variabel bebas bernilai lebih kecil dari 10 sehingga dapat disimpulkan bahwa tidak terjadi multikolinearitas.

### 4.2. Seleksi Fitur Berbasis Statistik

a. Uji Independensi Chi-Square

Hipotesis yang digunakan dalam uji ini yaitu:

$H_0$ : Sifat-sifat obyek tidak saling mempengaruhi (independen)

$H_1$ : Sifat-sifat obyek ada yang mempengaruhi (dependen)

dengan $\alpha = 5\%$ diperoleh nilai chi-square dan peluangnya yang diringkas dalam Tabel 3.

Tabel 3. Statistik Uji dalam Uji Independensi Chi-Square

| Variabel | Pearson Chi-Square | Df | Signifikansi |
|---|---|---|---|
| Status perkawinan | 0,272 | 2 | 0,873 |
| Jenis kelamin | 17,889 | 1 | 0,000 |
| Status kepemilikan rumah | 5,042 | 2 | 0,080 |
| Pendidikan | 12,553 | 1 | 0,000 |
| Pekerjaan | 9,294 | 2 | 0,010 |
| Wilayah Tempat Tinggal | 52,620 | 5 | 0,000 |
| Jumlah anak | 0,632 | 2 | 0,729 |

Berdasarkan Tabel 3 dapat disimpulkan bawa variabel yang berpengaruh pada keputusan akhir adalah Jenis Kelamin, Pendidikan, Pekerjaan, dan Wilayah Tempat Tinggal.

b. Uji Mann-Whitney

Hipotesis yang digunakan dalam uji ini yaitu:

$H_0 : M_x = M_y$

$H_1 : M_x \neq M_y$

dengan $\alpha = 5\%$ diperoleh nilai signifikansi yang diringkas dalam Tabel 4.

Tabel 4. Statistik Uji dalam Uji Mann-Whitney

| Variabel | Signifikansi |
|---|---|
| Lama bekerja | 0,790 |
| Lama perusahaan | 0,008 |
| Usia | 0,005 |
| Durasi pinjaman | 0,164 |
| Pendapatan | 0,047 |
| Jumlah Pinjaman yang Diajukan | 0,001 |

Berdasarkan Tabel 4 dapat disimpulkan bawa variabel yang berpengaruh pada keputusan akhir adalah Lama Perusahaan, Usia, Pendapatan, dan Jumlah Pinjaman yang Diajukan.

### 4.1 *Cluster-based Undersampling* dengan *Ensemble C-Modes*

Jumlah cluster yang dibentuk dalam proses ini ialah C=20. Hasil cluster dari proses *splitting* hingga hasil akhir cluster ditampilkan dalam Tabel 5.

Tabel 5. Hasil Proses Clustering dengan *Ensemble C-Modes*

| Indeks Amatan | Hasil Cluster C-Modes | Hasil Cluster C-Means | Hasil Cluster *Ensemble C-Modes* |
|---|---|---|---|
| 1 | 8 | 5 | 5 |
| 2 | 4 | 7 | 4 |
| 3 | 1 | 5 | 1 |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |
| 10613 | 1 | 12 | 6 |

Jumlah data mayor yang diambil secara keseluruhan dalam penelitian ini adalah 257 data. Data mayor tersebut diambil dari masing-masing cluster secara random dengan jumlah yang berbeda sesuai dengan persamaan 8.

Tabel 6. Jumlah Pengambilan Data Mayor Setiap Cluster

| Cluster | Jumlah Data Mayor | Jumlah Data Minor | Jumlah Pengambilan Data Mayor Setiap Cluster |
|---|---|---|---|
| 1 | 3540 | 65 | 14 |
| 2 | 231 | 33 | 2 |
| 3 | 823 | 17 | 12 |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |
| 20 | 26 | 1 | 7 |

### 4.3. Penghapusan *Outlier* dengan Algoritma MKNN

Jumlah *outlier* yang terdeteksi dalam penelitian ini berdasarkan Algoritma 3 adalah 53 outlier untuk K=3, 8 outlier untuk K=9, dan tidak ada outlier untuk K=13. Hasil tersebut menunjukkan bahwa semakin besar nilai K maka semakin sedikit *outlier* yang terdeteksi. Data yang digunakan setelah *outlier removal* dengan K=3 adalah 461 data, dengan K=9 adalah 506 data, dan dengan K=13 adalah 514 data.

### 4.4. Klasifikasi dengan Algoritma MKNN

Proporsi pembagian *training* dan *testing* adalah 9:1 dan digunakan *10-fold Cross Validation* untuk evaluasi klasifikator sehingga akan dilakukan 10 kali percobaan untuk setiap nilai K. Ukuran kinerja masing-masing model diperoleh dari rata-rata nilai G-Mean dari keseluruhan percobaan dalam model. *Dataset* yang digunakan adalah *dataset* yang telah dihapus *outlier*-nya menggunakan metode penghapusan *outlier* MKNN dengan nilai K yang sesuai. Matriks konfusi untuk percobaan pertama ditampilkan dalam Tabel 7.

Tabel 7. Matriks Konfusi Percobaan Pertama MKNN K=3

| **Kelas Asli** | **Kelas Prediksi** | |
|---|---|---|
| | **Ditolak** | **Disetujui** |
| **Ditolak** | 17 | 5 |
| **Disetujui** | 11 | 13 |

Ukuran-ukuran ketepatan klasifikasi dapat dihitung berdasarkan Tabel 7 dengan perhitungan berikut.

$$Sensitivity = \frac{TP}{(TP+FN)} \times 100\% = \frac{17}{(17+5)} \times 100\% = 77{,}27\%$$

$$Specificity = \frac{TN}{(TN+FP)} \times 100\% = \frac{13}{(13+11)} \times 100\% = 54{,}17\%$$

$$G - Mean = \sqrt{Sensitivity \times Specificity} = \sqrt{77{,}27\% \times 54{,}17\%} = 64{,}7\%$$

Sensitifitas sebesar 77,27% menunjukkan bahwa terdapat 77,27% pemohon dengan kelas asli ditolak yang diprediksikan secara benar untuk ditolak. Spesifisitas sebesar 54,17% berarti terdapat 54,17% pemohon dengan kelas asli disetujui yang diprediksikan secara benar untuk disetujui. Perhitungan di atas dilakukan pada setiap percobaan, lalu dicari rata-rata G-Mean untuk menunjukkan kinerja klasifikasi secara keseluruhan pada masing-masing K sebagaimana tercantum pada Tabel 8.

Tabel 8. Ukuran Kinerja Klasifikasi dengan MKNN

| **K** | **Rata-rata Nilai G-Mean** |
|---|---|
| 3 | 0,689 |
| 9 | **0,702** |
| 13 | 0,692 |

Tabel 8 menunjukkan bahwa nilai K yang optimal adalah K=9, oleh karena itu prediksi label kelas dalam sub bab ini akan dilakukan dengan nilai K tersebut. Misalkan diperoleh informasi dari pemohon bernama A yang merupakan seorang perempuan warga Banjarmasin berusia 27 tahun dengan pendidikan terakhir S1. Beliau telah bekerja sebagai konsultan di sebuah perusahaan yang berusia 7 tahun dengan pendapatan Rp. 3.800.000,- per bulan dan jumlah pinjaman yang diajukan adalah sebesar Rp. 13.000.000,-. Berdasarkan informasi tersebut akan diprediksi apakah permohonan kredit yang diajukan A layak untuk disetujui atau tidak. Hasil perhitungan jarak menunjukkan bahwa sembilan tetangga terdekat dari pemohon A adalah amatan ke-267, 437, 395, 160, 183, 96, 88, 117 dan 108. Langkah selanjutnya adalah mengevaluasi amatan mana yang merupakan tetangga mutual dari data pemohon A.

Tabel 9. Hasil Evaluasi Tetangga Mutual

| Amatan | Evaluasi | Kelas |
|---|---|---|
| 267 | Tetangga Mutual | Disetujui |
| 437 | Tetangga Mutual | Disetujui |
| 395 | Tetangga Mutual | Disetujui |
| 160 | Bukan Tetangga Mutual | Ditolak |
| 183 | Bukan Tetangga Mutual | Ditolak |
| 96 | Tetangga Mutual | Ditolak |
| 88 | Bukan Tetangga Mutual | Ditolak |
| 117 | Bukan Tetangga Mutual | Ditolak |
| 108 | Bukan Tetangga Mutual | Ditolak |

Tabel 9 menunjukkan bahwa dari sembilan tetangga terdekat hanya ditemukan empat buah tetangga mutual yaitu amatan 267, 437, 395 dan 96. Keputusan akhir bagi pemohon A kemudian diprediksi berdasarkan kelas terbanyak dari tetangga mutualnya, yaitu kelas ke-dua atau disetujui.

### 4.5. Klasifikasi dengan Algoritma KNN

*Dataset* yang digunakan adalah *dataset* yang telah dihapus *outlier*-nya menggunakan metode penghapusan *outlier* MKNN dengan nilai K yang sesuai. Ukuran kinerja klasifikasi dengan KNN secara keseluruhan pada masing-masing K adalah sebagai berikut.

Tabel 20. Ukuran Kinerja Klasifikasi dengan KNN

| K | Rata-rata Nilai G-Mean |
|---|---|
| 3 | **0,718** |
| 9 | 0,697 |
| 13 | 0,686 |

Tabel 10 menunjukkan bahwa nilai K yang optimal adalah K=3, oleh karena itu prediksi label kelas dalam sub bab ini akan dilakukan dengan nilai K tersebut. Data yang akan diprediksi label kelasnya adalah data pemohon A sebagaimana tercantum pada sub bab 4.5. Berdasarkan informasi yang diperoleh akan diprediksi apakah permohonan kredit yang diajukan oleh pemohon A layak untuk disetujui atau tidak. Proses prediksi diawali dengan perhitungan jarak terhadap *training*. Hasil perhitungan jarak menunjukkan bahwa 3 tetangga terdekat dari data pemohon A adalah amatan 238, 398, dan 358 dan ketiganya termasuk ke dalam kelas disetujui. Algoritma KNN memprediksi label kelas berdasarkan mayoritas kelas dari tetangga terdekat, dengan demikian berdasarkan hasil yang diperoleh dapat diprediksi bahwa pengajuan kredit debitur A termasuk ke kelas ke-dua atau disetujui.

### 4.6. Perbandingan Kinerja Klasifikasi Algoritma MKNN dan KNN

Perbandingan ukuran kinerja klasifikasi G-Mean dari masing-masing algoritma adalahsebagai berikut:

Tabel 31. Perbandingan Rata-rata Nilai G-Mean

| K | MKNN | KNN |
|---|---|---|
| 3 | 0,689 | **0,718** |
| 9 | **0,702** | 0,697 |
| 13 | 0,692 | 0,686 |

Tabel 11 menunjukkan bahwa MKNN bekerja lebih baik daripada KNN ketika nilai K cenderung tinggi yaitu K=9 dan K=13, namun secara keseluruhan rata-rata nilai G-Mean tertinggi dihasilkan oleh KNN dengan K=3. Hal tersebut menunjukkan bahwa penanganan kasus klasifikasi kelayakan kredit dalam penelitian ini paling tepat dilakukan dengan algoritma KNN pada K=3.

## 5. PENUTUP

### 5.1. KESIMPULAN

Kesimpulan yang diperoleh dari penelitian ini adalah sebagai berikut:

1) Penanganan outlier dengan prinsip tetangga mutual menghasilkan hasil yang berbeda-beda tergantung pada nilai K yang digunakan, semakin besar nilai K maka semakin sedikit outlier yang dideteksi.
2) Penanganan ketidakseimbangan kelas dalam dataset dapat dilakukan dengan teknik cluster-based undersampling salah satunya menggunakan algoritma Ensemble C-Modes dengan rasio jumlah kelas yang dihasilkan dalam penelitian ini sebesar 1:1.
3) Rata-rata nilai G-Mean yang dihasilkan dari klasifikasi dengan K=9 menggunakan Mutual K-Nearest Neighbor (MKNN) yaitu 0,702. Nilai tersebut lebih rendah dari rata-rata nilai G-Mean KNN pada K=3 senilai 0,718.
4) Penanganan kasus klasifikasi kelayakan kredit dalam penelitian ini paling tepat dilakukan dengan algoritma KNN pada K=3.

### 5.2. SARAN

Saran bagi penelitian selanjutnya adalah sebagai berikut:

1) Penggunaan dataset dengan tipe numerik saja atau kategorik saja dan struktur dataset yang berbeda sehingga dapat dilihat bagaimana kinerja MKNN dalam kondisi yang berbeda.
2) Penggunaan algoritma-algoritma clustering lain serta membandingkan kinerja klasifikasi pada data sebelum dan sesudah undersampling sehingga dapat diketahui bagaimana pengaruh yang diberikan dari proses resampling tersebut.

## DAFTAR PUSTAKA


Hermansyah. (2008). *Hukum Perbankan Nasional Indonesia.* Jakarta: Kencana.

Hair, J. F., J.R, A., Tatham, R., & Black, W. (1998). *Multivariate Data Analysis* (5th ed.). USA: Prentie-Hall Inc.

Han, J., Kamber, M., & Pei, J. (2006). *Data Mining: Concept and Techniques.* Waltham: Morgan Kaufmann Publisher.

Kubat, M., & Matwin, S. (1997). Addressing the Curse of Imbalanced Training Sets: one-sided selection. *Fourteenth International Conference on Machine Learning* , (pp. 179-186).

Liu, H., & Zhang, S. (2012). Noisy Data Elimination Using Mutual k-Nearest Neighbor for Classification Mining. *The Journal of System and Software*, 1067-1074.

Prasetyo, E. (2014). *Data Mining: Mengolah Data Menjadi Informasi Menggunakan Matlab.* Yogyakarta: ANDI.

Tan, P., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining.* Boston: Pearson Education.

Yen, S.-J., & Lee, Y.-S. (2009). Cluster-based Under-sampling Approaches for Imbalanced Data Distribution. *Expert Systems with Applications, 36*, 5718-5727.