

ANALISIS *CREDIT SCORING* TERHADAP STATUS PEMBAYARAN BARANG ELEKTRONIK DAN *FURNITURE* MENGUNAKAN *BOOTSTRAP AGGREGATING K-NEAREST NEIGHBOR*

Credit Scoring Analysis of Payment Status of Electronic Goods and Furniture Using Bootstrap Aggregating K-Nearest Neighbor

Putri Sri Astuti^{1*}, Memi Nor Hayati², Rito Goejantoro³

^{1,2,3}Prodi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Mulawarman
Jl. Barong Tongkok No.04, Samarinda, 75123, Indonesia

Corresponding author e-mail: ^{1*} putrisastuti1@gmail.com

Abstrak

Klasifikasi merupakan proses pengelompokan objek yang memiliki karakteristik sama ke dalam beberapa kategori. Penelitian ini menerapkan kombinasi algoritma klasifikasi, yaitu *Bootstrap Aggregating K-Nearest Neighbor* pada analisis *credit scoring*. Tujuannya adalah untuk mengklasifikasikan status pembayaran kredit barang elektronik dan *furniture* di PT KB Finansia Multi Finance tahun 2020 dan mengetahui tingkat akurasi yang dihasilkan. Status pembayaran kredit dikelompokkan dalam 2 kategori, yaitu lancar dan tidak lancar. Terdapat 7 variabel bebas yang digunakan sebagai gambaran karakteristik debitur, yaitu usia, jumlah tanggungan, lama tinggal, masa kerja, pendapatan, besar pembayaran, dan jangka pembayaran. Penerapan algoritma klasifikasi pada analisis *credit scoring* diharapkan dapat membantu pihak kreditur dalam mengambil keputusan menerima atau menolak pengajuan kredit dari calon debitur. Hasil penelitian menunjukkan bahwa akurasi yang diperoleh dari algoritma *Bootstrap Aggregating K-Nearest Neighbor* dengan proporsi 90:10, $m=80\%$, $C=73$, dan $K=5$ adalah yang terbaik, yaitu sebesar 92,308%.

Kata Kunci : Klasifikasi, *K-Nearest Neighbor*, *Bootstrap Aggregating*, Kredit

Abstract

Classification is the process of grouping objects that have the same characteristics into several categories. This study applies a combination of classification algorithms, namely *Bootstrap Aggregating K-Nearest Neighbor* in credit scoring analysis. The aim is to classify the credit payment status of electronic goods and furniture at PT KB Finansia Multi Finance in 2020 and determine the level of accuracy produced. Credit payment status is grouped into 2 categories, namely smoothly and not smoothly. There are 7 independent variables that are used to describe the characteristics of the debtor, namely age, number of dependents, length of stay, years of service, income, amount of payment, and payment period. The application of the classification algorithm at the credit scoring analysis is expected to assist creditors in making decisions to accept or reject credit applications from prospective debtors. The results showed that the accuracy obtained from the *Bootstrap Aggregating K-Nearest Neighbor* algorithm with a proportion of 90:10, $m=80\%$, $C=73$, and $K=5$ was the best, which was 92.308%.

Keywords: classification, *k-nearest neighbor*, *bootstrap aggregating*, credit.

Article info:

Submitted: 30th August 2021

Accepted: 19th November 2021

How to cite this article:

P. S. Astuti, M. N. Hayati, and R. Goejantoro, "ANALISIS *CREDIT SCORING* TERHADAP STATUS PEMBAYARAN BARANG ELEKTRONIK DAN *FURNITURE* MENGGUNAKAN *BOOTSTRAP AGGREGATING K-NEAREST NEIGHBO*", *BAREKENG: J. Il. Mat. & Ter.*, vol. 15, no. 04, pp. 735-744, Dec. 2021.



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).
Copyright © 2021 Putri Sri Astuti, Memi Nor Hayati, Rito Goejantoro

1. PENDAHULUAN

Berbagai bidang usaha tentunya memiliki data besar yang berkaitan dengan produk, konsumen, maupun sistem pelayanannya. Sebagai bentuk upaya pengolahan data tersebut menjadi informasi yang lebih bermanfaat, maka muncullah beragam algoritma data *mining*. Pemilihan algoritma yang akan digunakan dapat disesuaikan dengan tugas atau tujuan pengerjaannya, misalnya untuk klasifikasi. Klasifikasi merupakan proses untuk memperkirakan kelompok objek baru, dimana kelasnya belum diketahui, dengan cara menggali model yang dapat menjelaskan atau membedakan kelas pada suatu kelompok data [1]. Dalam proses klasifikasi terdapat dua pekerjaan utama yang dilakukan, yaitu *learning* dan klasifikasi. Pada tahap *learning*, algoritma akan menganalisis data yang kelasnya sudah diketahui. Kemudian dilakukan representasi dalam bentuk *rule* klasifikasi. *Rule* yang diperoleh akan digunakan untuk mengklasifikasikan kelas dari data baru [2]. Algoritma sederhana yang biasa digunakan dalam klasifikasi adalah *K-Nearest Neighbor* (K-NN). K-NN bertujuan mengklasifikasikan data *testing* berdasarkan kedekatan jarak antara data *training* dengan data *testing* tersebut. Hal ini dilakukan dengan mencari beberapa kelompok objek dalam data *training* yang paling mirip dengan objek baru pada data *testing* [3]. Semakin dekat jarak data *training* ke data *testing* mengindikasikan bahwa semakin besar kemiripan yang dimiliki oleh keduanya. Dalam proses klasifikasi sering kali melibatkan kombinasi algoritma dengan tujuan peningkatan akurasi hasil prediksi, seperti *Bootstrap* dan *Aggregating (Bagging)* yang dilakukan oleh Breiman pada tahun 1996. *Bagging* dirancang untuk meningkatkan hasil klasifikasi melalui kombinasi acak pada set data *training* serta dapat mengurangi variansi dan menghindari *overfitting* [4]. Beberapa penelitian telah menerapkan *Bagging* pada algoritma klasifikasi tunggal seperti K-NN dengan pendekatan model *average*.

Bagging K-NN dapat diterapkan di kehidupan sehari-hari, termasuk dalam bidang pengkreditan. Banyaknya jumlah pengajuan kredit, disertai dengan beragamnya karakter debitur, menuntut pihak kreditur memiliki pertimbangan yang tepat. Oleh sebab itu diperlukan suatu analisis untuk memprediksi apakah seorang debitur baru dapat melakukan pembayaran kredit dengan lancar atau tidak. Dalam hal ini berbagai perusahaan di bidang pengkreditan telah menerapkan evaluasi kredit atau dikenal dengan penilaian kredit (*credit scoring*), sebagai bahan pertimbangan untuk menerima pengajuan kredit dari calon debitur. Sejalan dengan kemajuan teknologi informasi yang pesat maka perusahaan pengkreditan mulai menerapkan model statistik dalam mengevaluasi kredit. Salah satunya dengan mengembangkan analisis *credit scoring* dengan menerapkan algoritma klasifikasi. *Credit scoring* adalah tahap mengevaluasi pengajuan kredit dari calon debitur dengan segala atribut yang dimiliki. Model *credit scoring* dibangun menggunakan sampel kredit masa lalu dan kemudian dibagi menjadi 2 kelas, yaitu kredit baik (pembayaran dilakukan tepat waktu) dan kredit bermasalah (pembayaran dilakukan tidak tepat waktu). Kombinasi karakteristik debitur diperoleh berdasarkan pola sampel masa lalu sehingga dapat membedakan debitur yang baik dan buruk [5].

Berdasarkan uraian tersebut, penulis tertarik melakukan penelitian untuk mengklasifikasikan status pembayaran kredit barang elektronik dan *furniture* di PT KB Finansia Multi Finance tahun 2020 dan mengetahui tingkat akurasi yang dihasilkan. Penerapan algoritma klasifikasi pada tahap analisis *credit scoring* diharapkan dapat membantu pihak kreditur dalam mengambil keputusan menerima atau menolak pengajuan kredit dari calon debitur. Dalam penelitian ini digunakan proporsi

2. METODE PENELITIAN

Penelitian ini menerapkan algoritma *Bootstrap Aggregating K-Nearest Neighbor (Bagging K-NN)* pada sampel data debitur yang melakukan pelunasan kredit di tahun 2020. Teknik *purposive sampling* digunakan untuk pengambilan data dengan pertimbangan berkaitan dengan ketersediaan data terbaru.

2.1 K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* (K-NN) termasuk dalam algoritma *supervised learning*. Tujuan algoritma *supervised learning* adalah menemukan pola atau fungsi baru yang dapat memetakan *input* ke *output* yang sesuai [6]. Dasar utama dari algoritma K-NN adalah kedekatan jarak suatu data *testing* ke data *training*. Nilai jarak tersebut digunakan sebagai ukuran kedekatan atau kemiripan. Algoritma ini bekerja dengan mencari nilai K terdekat berdasarkan pola *input* dan kemudian menentukan kelas keputusan [7]. Nilai K menunjukkan jumlah data *training* yang terdekat dengan data *testing*. Misalnya apabila K bernilai 1, maka akan diambil 1 tetangga terdekat dari data *training*. Apabila K bernilai 3, maka akan diambil 3 tetangga

terdekat dari data *training*, begitu seterusnya. Jika dalam K tetangga terdekat terdapat dua kelas berbeda, maka kelas data *testing* akan ditentukan berdasarkan jumlah terbanyak atau melalui *voting* [8].

Jauh atau dekatnya jarak antara data *testing* dan data *training* dapat dihitung dengan perhitungan jarak Euclid. Semakin besar nilai yang diperoleh maka semakin jauh tingkat kemiripan antara data *testing* dan data *training*. Sebaliknya jika semakin kecil nilai yang diperoleh, maka semakin tinggi tingkat kemiripannya [9]. Rumus jarak Euclid dapat dilihat melalui Persamaan (1) [10]:

$$d_{(a,b)} = \sqrt{\sum_{g=1}^p (x_{a,g} - x_{b,g})^2} \quad (1)$$

Keterangan:

$d_{(a,b)}$: jarak Euclid antara data *training* ke-*a* dengan data *testing* ke-*b*

$x_{a,g}$: nilai data *training* ke-*a* pada variabel ke-*g*

$x_{b,g}$: nilai data *testing* ke-*b* pada variabel ke-*g*

p : banyak variabel bebas

Variabel dalam klasifikasi harus distandarisasi apabila memungkinkan untuk menghindari masalah yang akan dihasilkan dari penggunaan nilai skala yang berbeda antar variabel dalam klasifikasi. Setelah dilakukan standarisasi, semua variabel akan berada dalam jangkauan yang sama sehingga proporsi pengaruh pada fungsi dalam klasifikator seimbang [1]. Pencarian nilai rata-rata seperti dalam Persamaan (2) dan juga nilai simpangan baku seperti Persamaan (3) akan digunakan untuk menghitung standarisasi suatu nilai seperti dalam Persamaan (4).

$$\bar{x}_g = \frac{1}{n} \sqrt{\sum_{j=1}^n x_{j,g}} \quad (2)$$

$$s_g = \sqrt{\frac{1}{n-1} \sum_{j=1}^n (x_{(j,g)} - \bar{x}_g)^2} \quad (3)$$

$$x'_{(j,g)} = \frac{x_{(j,g)} - \bar{x}_g}{s_g} \quad (4)$$

Keterangan:

n : jumlah data

$x_{j,g}$: data ke-*j* pada variabel ke-*g*

\bar{x}_g : rata-rata data pada variabel ke-*g*

s_g : simpangan baku variabel ke-*g*

$x'_{(j,g)}$: standarisasi data ke-*j* pada variabel ke-*g*

2.2 Bootstrap Aggregating

Bootstrap Aggregating atau biasa disebut dengan *Bagging* merupakan salah satu teknik yang diusulkan oleh Breiman pada tahun 1996 dan sering digunakan pada beberapa algoritma klasifikasi ataupun regresi untuk mereduksi variansi dari suatu variabel terikat, sehingga dapat memperbaiki proses pendugaan [11]. Penggunaan *Bagging* sangat membantu terutama untuk mengatasi sifat ketidakstabilan klasifikasi dan pohon regresi. Pada banyak gugus data yang dicoba, *Bagging* mampu mengurangi tingkat kesalahan pada kasus klasifikasi [10].

Bootstrap pertama kali digunakan oleh Efron pada tahun 1979. Metode ini didasari teknik *resampling* yaitu pengambilan data sampel secara berulang dan sifatnya saling bebas. Dari pengulangan tersebut akan diestimasi tingkat kesalahannya. Dalam hal ini sampel diambil secara acak dengan pengembalian [12]. Penentuan batas replikasi pada *Bootstrap* ditetapkan apabila hasil replikasi *Bootstrap* sudah menghasilkan nilai akurasi yang tinggi. Peningkatan akurasi dan prediksi akan terjadi jika banyaknya replikasi berkisar pada

50 hingga 100 kali replikasi. Namun jika replikasinya ditingkatkan melebihi 100 kali, biasanya akan menghasilkan nilai akurasi yang tidak lebih besar dari nilai akurasi pada replikasi 50 hingga 100 kali [13]. Penelitian Breiman menunjukkan bahwa pengulangan dapat dilakukan 50 kali untuk kasus klasifikasi dan 25 kali untuk kasus regresi sehingga memperoleh hasil yang memuaskan. Selanjutnya dalam tahapan *Aggregating* dilakukan prediksi gabungan berdasarkan B buah klasifikasi tersebut dengan menggunakan aturan *majority vote* [10]. Adapun tahapan *Bagging* adalah sebagai berikut [14]:

1. Melakukan *Bootstrap* sebanyak m dari sejumlah n data *training*. Hasil replikasi *Bootstrap* X^c diulangi untuk $c = 1, 2, \dots, C$, di mana m adalah banyak data yang diambil dari data *training*, n adalah banyaknya data *training*, dan C adalah banyaknya replikasi *Bootstrap* yang dilakukan.
2. Menggunakan *simple majority vote* yaitu memilih kategori yang paling banyak muncul dari hasil penilaian sebagai aturan untuk pengambilan keputusan akhir.

2.3 Confusion Matrix

Proses prediksi yang dilakukan diharapkan mampu melakukan klasifikasi semua data set dengan benar, tetapi tidak dapat dipungkiri jika kinerja suatu sistem tidak 100% benar sehingga perlu dilakukan pengukuran kinerja klasifikasi. *Confusion matrix* atau matriks konfusi merupakan salah satu cara yang biasanya digunakan untuk mengukur kinerja suatu algoritma klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan dengan hasil klasifikasi sebenarnya [8]. Nilai yang dihasilkan pada *confusion matrix* adalah berupa evaluasi kinerja algoritma yang dapat dilihat dari nilai *accuracy*. Nilai *Accuracy* merupakan persentase jumlah record data yang diklasifikasikan secara benar oleh algoritma [15]. Perhitungan nilai *Accuracy* terdapat pada Persamaan (5).

$$Accuracy = \frac{TP + TN}{P + N} \times 100\% \quad (5)$$

2.4 Variabel Penelitian

Variabel yang digunakan dalam penelitian ini terdiri atas variabel terikat yaitu status pembayaran kredit dari debitur dan tujuh variabel bebas yaitu usia, jumlah tanggungan, lama tinggal, masa kerja, pendapatan, besar pembayaran, dan jangka pembayaran. Status pembayaran kredit yang digunakan dikelompokkan dalam dua kelas, yakni status lancar (L) dan tidak lancar (TL).

2.5 Tahap Analisis

Tahap analisis yang dilakukan adalah sebagai berikut:

- i Standarisasi data menggunakan Persamaan (4)
- ii Pengacakan data
- iii Membagi data *training* dan data *testing*
- iv Menghitung jarak data *training* dan data *testing*
- v Melakukan klasifikasi berdasarkan tetangga terdekat
- vi Memilih proporsi terbaik
- vii Penerapan *Bootstrap*
- viii Penerapan *Aggregating*
- ix Menghitung ketepatan akurasi algoritma

3. HASIL DAN PEMBAHASAN

Data yang digunakan berjumlah 133 data, yaitu D1 hingga D133. Sebelum masuk ke dalam tahap analisis, terlebih dahulu dilakukan standarisasi menggunakan Persamaan (4). tujuannya agar semua data pada setiap variabel berada dalam jangkauan yang sama. Dengan demikian proporsi pengaruh tiap variabel pada fungsi klasifikator dapat seimbang. Setelah standarisasi data, maka tahap selanjutnya adalah pengacakan data. Pengacakan data dalam penelitian ini dilakukan dengan menggunakan *software R*. Adapun tujuan dari pengacakan ini, yaitu agar setiap data memiliki kesempatan yang sama untuk menjadi data *training* atau data *testing*. Data hasil pengacakan akan dibagi menjadi data *training* dan data *testing* dengan menggunakan tiga proporsi, yaitu 50:50, 70:30, dan 90:10. Ketiga proporsi akan diterapkan pada tahap analisis yang sama. Peneliti menggunakan proporsi 90:10 sebagai contoh untuk menguraikan perhitungan dalam analisis.

Penggunaan proporsi 90:10 bermakna bahwa 90% data dari hasil pengacakan yang berada di urutan awal akan menjadi data *training*, sedangkan 10% data di urutan selanjutnya akan menjadi data *testing*. Sehingga diperoleh 120 data *training* dan 13 data *testing*.

3.1 Perhitungan Jarak Euclid

Jarak Euclid digunakan untuk menghitung jarak data *training* ke setiap data *testing*. Perhitungan jarak Euclid dilakukan dengan menggunakan Persamaan (1). Jarak antara semua data *training* dengan data *testing* pertama yaitu D90 akan dihitung seperti berikut.

$$d_{1,1} = \sqrt{((-0,997) - (-0,450))^2 + ((-1,829) - (-0,036))^2 + \dots + ((0,365) - (0,365))^2}$$

$$= 2,559$$

$$d_{2,1} = \sqrt{((-0,450) - (-0,450))^2 + ((1,159) - (-0,036))^2 + \dots + ((0,365) - (0,365))^2}$$

$$= 1,646$$

⋮

$$d_{120,1} = \sqrt{((-0,778) - (-0,450))^2 + ((-0,634) - (-0,036))^2 + \dots + ((-0,663) - (0,365))^2}$$

$$= 2,053$$

Perhitungan jarak Euclid dilanjutkan hingga data *training* ke-120 (D109) dengan data *testing* ke-13 (D25).

3.2 Klasifikasi Status Pembayaran Kredit

Jarak Euclid untuk masing-masing data *testing* diurutkan berdasarkan kedekatannya dengan data *training*. Setelah itu data *testing* diklasifikasikan berdasarkan nilai K, dimana dalam penelitian ini digunakan K=1, K=3, K=5, K=7, dan K=9. Nilai K berfungsi sebagai batas K-NN atau batas K tetangga terdekat. Apabila K awal yang digunakan adalah 1 (1-NN), maka hasil klasifikasi data *testing* akan didasari pada kelas dari 1 tetangga terdekat saja atau *rank* pertama. Apabila menggunakan batas 3 tetangga terdekat (3-NN), maka hasil klasifikasi data *testing* akan didasari pada 3 tetangga terdekat, yaitu *rank* pertama sampai dengan *rank* ketiga dan kemudian dilakukan *voting* dari 3 kelas tetangga terdekat tersebut. Begitu pun seterusnya sampai menggunakan batas 9 tetangga terdekat (9-NN). Peneliti menggunakan data *testing* pertama (D90) sebagai contoh untuk menguraikan *rank* jarak Euclid yang dapat dilihat pada **Tabel 1**.

Tabel 1. Rank Jarak Euclid Data Training dengan Data Testing D90

Rank	Data Training		Data testing	Batas K-NN	Hasil Klasifikasi Data Testing
	Sampel	Klasifikasi	D90 $d_{(a,90)}$		
1	D18	L	0,729	1-NN	L
2	D126	TL	1,139		
3	D94	L	1,157	3-NN	L
4	D19	L	1,186		
5	D110	L	1,191	5-NN	L
6	D46	L	1,272		
7	D23	L	1,340	7-NN	L
8	D60	L	1,389		
9	D120	TL	1,412	9-NN	L
⋮	⋮	⋮	⋮		
120	D34	TL	6,427		

3.3 Pemilihan Proporsi Terbaik

Hasil klasifikasi kategori status pembayaran kredit untuk semua data *testing* akan dibandingkan dengan kategori sebenarnya (*actual*), sehingga dapat diketahui jumlah kesalahan klasifikasi di setiap nilai K. Semakin kecil jumlah kesalahan klasifikasi maka semakin besar kemungkinan nilai K tersebut menjadi nilai K optimal.

Selanjutnya untuk memilih proporsi terbaik dapat dilakukan dengan melihat dari nilai akurasi klasifikasi yang dihasilkan menggunakan Persamaan (5). Secara keseluruhan akurasi untuk setiap nilai K pada masing-masing proporsi dapat dilihat dalam **Tabel 2**. Kemudian akurasi tersebut akan dibandingkan dan dipilih yang terbaik (nilai akurasi tertinggi) untuk digunakan dalam tahap selanjutnya.

Tabel 2. Akurasi Setiap Nilai K pada Proporsi yang Berbeda Proporsi Data *Training* dan Data *Testing*

K-NN	<i>Testing</i>		
	50:50	70:30	90:10
1-NN	66,667%	69,231%	76,923%
3-NN	66,667%	69,231%	76,923%
5-NN	68,182%	69,231%	76,923%
7-NN	74,242%	66,667%	84,615%
9-NN	75,758%	71,795%	92,308%

Berdasarkan **Tabel 2** diketahui bahwa akurasi tertinggi terdapat pada penggunaan proporsi 90:10, yaitu sebesar 92,308%. Hal ini berarti data yang akan digunakan dalam tahap selanjutnya adalah data dengan pembagian proporsi 90:10, yakni terdapat 120 data training dan 13 data testing.

3.4 Penerapan *Bootstrap*

Tahap ini didasari pada pengambilan sampel secara berulang (*resampling*) disertai pengembalian, sehingga memungkinkan sampel terambil lebih dari satu kali. Hal pertama yang perlu dilakukan yaitu mengambil sampel secara acak dari data *training* untuk menjadi data *Bootstrap* sebanyak $m\%$ dari jumlah data *training*. Proporsi 90:10 terpilih pada sub bab 4.5 karena memiliki nilai akurasi tertinggi, sehingga jumlah data *Bootstrap* yang akan diambil adalah $(m\% \times 120)$ data. Kemudian data *bootstrap* dijadikan sebagai data *training* untuk mengklasifikasikan kembali 13 data *testing* dengan menggunakan tahapan yang sama seperti pada tahap K-NN. Setelah itu tahap *resampling* hingga tahap klasifikasi diulangi sebanyak C kali. Dalam hal ini, akan diperoleh hasil klasifikasi data *testing* sebanyak C .

Penelitian ini menggunakan $m=50\%$, $m=60\%$, $m=70\%$, $m=80\%$, $m=90\%$, dan $m=100\%$. Adapun nilai C dipilih secara acak sebanyak 10 dari rentang 30 hingga 120. Berdasarkan hasil pengacakan diperoleh 10 nilai C yang digunakan, yaitu 31, 45, 51, 57, 67, 73, 79, 85, 99, dan 107. Proses klasifikasi menggunakan nilai $K=1$, $K=3$, $K=5$, $K=7$, dan $K=9$. Penggunaan nilai m , C , dan K dilakukan satu per satu dalam tahap klasifikasi. Kemudian dianalisis serta dipilih yang terbaik berdasarkan akurasi yang diperoleh. Peneliti menetapkan penggunaan $m=80\%$, $C=73$, dan $K=5$ sebagai contoh untuk menguraikan tahapan *Bootstrap* hingga perhitungan akurasi. Dengan demikian jumlah data *Bootstrap* yang diambil adalah sebanyak $(80\% \times 120)$ data, yakni sebanyak 96 data. Pengulangan dilakukan sebanyak 73 kali dengan menggunakan batas 5-NN dalam proses klasifikasi. Data *Bootstrap* pertama dapat dilihat pada **Tabel 3**.

Tabel 3. Data *Bootstrap* pertama

No	Sampel	Variabel							
		Y	X_1	X_2	X_3	X_4	X_5	X_6	X_7
1	D14	TL	0,206	-0,036	-0,713	0,563	1,007	2,074	1,907
2	D3	L	-0,122	1,159	0,199	-0,905	-0,85	-0,268	-1,177
3	D51	TL	0,206	1,159	2,113	0,563	-0,85	0,155	0,365
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
96	D73	TL	0,863	1,159	1,110	0,122	1,316	-0,506	-1,177

Data *Bootstrap* pertama akan berperan sebagai data *training* untuk mengklasifikasikan data *testing*. Tahapan klasifikasi dilakukan sama seperti pada tahapan klasifikasi K-NN tunggal.

3.4.1 Perhitungan Jarak

Perhitungan jarak Euclid antara data *training Bootstrap* pertama dengan data *testing* D90 menggunakan Persamaan (1) dilakukan seperti berikut.

$$\begin{aligned}
 d_{1,1} &= \sqrt{((0,206) - (-0,450))^2 + ((-0,036) - (-0,036))^2 + \dots + ((-1,907) - (0,365))^2} \\
 &= 3,457 \\
 d_{2,1} &= \sqrt{((-0,122) - (-0,450))^2 + ((1,159) - (-0,036))^2 + \dots + ((-1,177) - (0,365))^2} \\
 &= 2,545 \\
 &\vdots \\
 d_{120,1} &= \sqrt{((0,863) - (-0,450))^2 + ((1,159) - (-0,036))^2 + \dots + ((-1,177) - (0,365))^2} \\
 &= 3,586
 \end{aligned}$$

Perhitungan jarak Euclid dilanjutkan hingga data *training Bootstrap* pertama ke-96 (D73) dengan data *testing* ke-13 (D25).

3.4.2 Klasifikasi Status Pembayaran Kredit

Jarak Euclid yang telah diperoleh untuk masing-masing data *testing* diurutkan berdasarkan kedekatannya dengan data *training Bootstrap* pertama. Setelah itu dilakukan klasifikasi status pembayaran menggunakan batas 5-NN. Peneliti menggunakan data *testing* D90 sebagai contoh untuk menampilkan *rank* jarak Euclid dan hasil klasifikasi yang dapat dilihat pada **Tabel 4**.

Tabel 4. Rank Jarak Euclid dan Hasil Klasifikasi Data Testing D90

Rank	Data Training		Data Testing	Batas K-NN	Hasil Klasifikasi Data Testing
	Sampel	Klasifikasi	D90 $d_{(a,90)}$		
1	D94	L	1,157		
2	D19	L	1,186		
3	D19	L	1,186		
4	D110	L	1,191		
5	D23	L	1,340	5-NN	L
⋮	⋮	⋮	⋮		
96	D34	TL	6,427		

Hasil klasifikasi menunjukkan status pembayaran untuk data *testing* D90 menggunakan 5-NN adalah lancar. Selanjutnya dilakukan *resampling* kedua dan seterusnya sampai dengan 73 kali. Data hasil *resampling* dijadikan sebagai data *training Bootstrap* yang digunakan untuk mengklasifikasikan setiap data *testing*. Dalam hal ini satu data *testing* memiliki 73 hasil klasifikasi berdasarkan data *training Bootstrap*.

3.5 Penerapan Aggregating

Setelah memperoleh 73 hasil klasifikasi untuk setiap data *testing*, maka langkah selanjutnya adalah menerapkan *Aggregating*. Langkah ini dilakukan untuk menentukan kategori status pembayaran akhir dari tiap data *testing* berdasarkan suara terbanyak (*majority vote*). Hasil *majority vote* dari *Aggregating* dapat dilihat pada **Tabel 5**.

Tabel 5. Hasil *Aggregating Data Testing* dengan 5-NN

Data <i>Testing</i>	Hasil Klasifikasi Pengulangan <i>Bootstrap</i> ke-								L	TL	<i>Majority Vote</i>
	1	2	3	4	...	71	72	73			
D90	L	L	L	L	...	L	L	L	68	5	L
D116	L	L	L	TL	...	L	L	TL	57	16	L
D133	TL	L	L	L	...	L	L	TL	64	9	L
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
D25	TL	TL	L	L	...	TL	TL	TL	41	32	L

Berdasarkan 73 hasil *Bootstrap* menggunakan 5-NN diperoleh bahwa data *testing* D90 diklasifikasikan dalam kategori lancar sebanyak 68 kali dan dalam kategori tidak lancar sebanyak 5 kali. Sehingga dengan menerapkan *majority vote* dalam *Aggregating* maka klasifikasi akhir untuk data *testing* D90 adalah status pembayaran kredit lancar. Data *testing* D116 diklasifikasikan dalam kategori lancar sebanyak 57 kali dan kategori tidak lancar sebanyak 16 kali. Dengan demikian maka klasifikasi akhir untuk data *testing* D116 adalah status pembayaran kredit lancar. Data *testing* D133 diklasifikasikan dalam kategori lancar sebanyak 64 kali dan kategori tidak lancar sebanyak 9 kali. Klasifikasi akhir untuk data *testing* D133 adalah status pembayaran kredit lancar. Cara yang sama juga dilakukan untuk menentukan klasifikasi akhir dari data *testing* yang lainnya hingga data *testing* D25.

3.6 Ketepatan Akurasi Algoritma

Setelah memperoleh hasil klasifikasi untuk setiap data *testing* melalui tahap *Aggregating*, maka selanjutnya dilakukan perhitungan akurasi menggunakan Persamaan (5). Setelah memperoleh nilai akurasi, maka selanjutnya adalah mengulangi tahapan *Bootstrap* hingga perhitungan akurasi dengan menggunakan nilai m , C , dan K yang lain. Akurasi yang diperoleh akan dianalisis secara keseluruhan dan dipilih yang terbaik untuk mengklasifikasikan data *testing*. Peneliti melakukan analisis untuk memilih nilai m , C , dan K terbaik dengan beberapa pertimbangan, yaitu peningkatan akurasi dari nilai yang diperoleh pada Tabel 4.10 dan kekonsistenan akurasi yang ditinjau secara keseluruhan di setiap nilai m , C , dan K . Kekonsistenan ini menggambarkan bahwa hasil klasifikasi yang diperoleh dapat lebih dipercaya karena tidak berubah untuk setiap nilai m , C , dan K yang digunakan.

Berdasarkan pertimbangan tersebut, peneliti memilih penggunaan $K=5$ dengan nilai $m=80\%$ dan $C=73$ yang terbaik. Dengan demikian penerapan algoritma *Bagging* K-NN menggunakan proporsi 90:10, $m=80\%$, $C=73$, dan $K=5$ dalam analisis *credit scoring* barang elektronik dan *furniture* di PT KB Finansia Multi Finance menghasilkan akurasi terbaik, yakni sebesar 92,308%. Algoritma ini dapat mengklasifikasikan status pembayaran kredit dari 12 debitur secara tepat dari 13 debitur yang status pembayarannya tepat diklasifikasikan, yaitu 10 debitur dengan status pembayaran lancar dan 2 debitur dengan status pembayaran tidak lancar. Adapun 1 debitur lainnya tidak tepat diklasifikasikan, yaitu debitur yang seharusnya memiliki status pembayaran lancar diklasifikasikan dalam kategori status pembayaran tidak lancar.

4. KESIMPULAN

Analisis *credit scoring* terhadap status pembayaran kredit barang elektronik dan *furniture* di PT KB Finansia Multi Finance Tahun 2020 menggunakan metode *Bagging* K-NN, dengan proporsi 90:10, $m=80\%$, $C=73$, dan $K=5$ dapat mengklasifikasikan 10 debitur yang memiliki status pembayaran kredit lancar secara tepat dan juga 2 debitur yang memiliki status pembayaran kredit tidak lancar secara tepat. Sedangkan 1 debitur lainnya tidak tepat diklasifikasikan, dimana debitur seharusnya memiliki status pembayaran kredit lancar namun hasil klasifikasi menunjukkan status pembayaran kredit tidak lancar. Adapun tingkat akurasi yang diperoleh dari hasil analisis *credit scoring* dengan menggunakan algoritma *Bagging* K-NN adalah 92,308%.

DAFTAR PUSTAKA

- [1] A. Siregar and A. Puspabhuana, *Data Mining: Pengolahan Data menjadi Informasi dengan RapidMiner*, Surakarta: CV Kekata Group, 2017.
- [2] E. Prasetyo, *Data Mining: Mengolah Data Menjadi Informasi Menggunakan Matlab*, Yogyakarta: Andi, 2014.
- [3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, San Francisco: Morgan Kaufman Publisher, 2011.
- [4] R. S. Wahono and N. Suryana, "Combining Particle Swarm Optimization Based Feature Selection and Bagging Technique for Software Defect Prediction," *International Journal of Software Engineering and Its Applications*, vol. 7, no. 5, pp. 153-166, 2013.
- [5] Y. Y. W., "Perbandingan Performansi Algoritma Decision Tree C5.0, Car, dan Chaid: Kasus Prediksi Status Resiko Kredit di Bank X," in *Seminar Nasional Aplikasi Teknologi Informasi*, Yogyakarta, 2007.
- [6] F. S. Pamungkas, B. D. Prasetya and I. Kharisudin, "Perbandingan Metode Klasifikasi Supervised Learning pada Data Bank Customers Menggunakan Python," *PRISMA Prosiding Seminar Nasional Matematika*, pp. 689-694, 2020.
- [7] E. Prasetyo, *Data Mining: Konsep dan Aplikasi Menggunakan Matlab*, Yogyakarta: Andi, 2012.
- [8] Suyanto, *Machine Learning Tingkat Dasar dan Lanjut*, Bandung: Informatika, 2018.
- [9] J. Riany, M. Fajar and M. P. Lukman, "Penerapan Deep Sentiment Analysis pada Angket Penilaian Terbuka Menggunakan K-Nearest Neighbor," *Jurnal SISFO*, vol. 6, no. 1, pp. 147-156, 2016.
- [10] A. M. Mukid, T. Wuryandari, D. Ratnaningrum and R. S. Rahayu, "Bagging Classification Trees untuk Prediksi Resiko Preeklampsia (Studi Kasus: Ibu Hamil Kategori Penerima Jampersal di RSUD Dr. Moewardi Surakarta)," *Media Statistika*, vol. 8, no. 2, pp. 111-120, 2015.
- [11] C. D. Sutton, "Classification and Regression Trees, bagging, and Boosting," *Handbook of Statistics*, vol. 24, pp. 303-329, 2005.
- [12] R. J. Tibshirani and B. Efron, *An Introduction to the Bootstrap*, New York: Chapman and Hall, 1993.
- [13] Mustafa, "Perancangan Aplikasi Prediksi Kelulusan Tepat Waktu Bagi Mahasiswa Baru dengan Teknik Data Mining (Studi Kasus: Data Akademik Mahasiswa STMIL Dipanegara)," *Jurnal Citec*, vol. 1, no. 3, 2014.
- [14] T. Hestie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York: Springer-Verlag, 2001.
- [15] M. F. Rahman, M. I. Darmawidjadja and D. Alamsah, "Klasifikasi untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN)," *Jurnal Informatika*, vol. 11, no. 1, pp. 36-45, 2017.

