

## ANALISIS *CREDIT SCORING* MENGGUNAKAN METODE *BAGGING K-NEAREST NEIGHBOR*

Fatimah<sup>1</sup>, Moch. Abdul Mukid<sup>2</sup>, Agus Rusgiyono<sup>3</sup>

<sup>1</sup>Mahasiswa Departemen Statistika FSM Universitas Diponegoro

<sup>2,3</sup>Staff Pengajar Departemen Statistika FSM Universitas Diponegoro

### ABSTRACT

According to Melayu (2004) credit is all types of loans that have to be paid along with the interest by the borrower according to the agreed agreement. To keep the quality of loans and avoid financial failure of banks due to large credit risks, we need a method to identified any potentially customer's with bad credit status, one of the methods is *Credit Scoring*. One of Statistical method that can predict the classification for *Credit Scoring* called *Bagging k-Nearest Neighbor*. This Method uses *k*-object nearest neighbor between data testing to *B*-bootstrap of the training dataset. This classification will use six independence variables to predict the class, these are Age, Work Year, Net Earning, Other Loan, Nominal Account and Debt Ratio. The result determine  $k = 1$  as the optimal *k*-value and show that *Bagging k-Nearest Neighbor's* accuracy rate is 66,67%.

Key word : Credit scoring, Classification, Bagging k-Nearest Neighbor

### 1. PENDAHULUAN

Istilah kredit berasal dari bahasa Yunani yaitu '*credere*' yang berarti kepercayaan (Herprasetyo, 2012). Menurut Chapra (2000), kredit harus dialokasikan dengan tujuan membantu merealisasikan kemaslahatan sosial secara umum. Tujuan dari kesepakatan kredit menurut Kasmir (2002) adalah memberikan keuntungan bagi bank, membantu usaha nasabah dan membantu pemerintah. Namun jika ditemui kondisi dimana nasabah sudah tidak sanggup membayar sebagian atau seluruh kewajibannya kepada bank seperti yang telah diperjanjikan maka kondisi ini disebut dengan kredit macet (Kuncoro dan Suhardjono, 2002). Semakin banyak jumlah nasabah kredit macet dalam sebuah bank, akan menghambat kinerja bank dan melemahkan pendapatan dari usaha perbankan tersebut. Untuk mencegah terjadinya hal tersebut, dapat dilakukan pertimbangan atas persetujuan pengajuan kredit oleh calon nasabah tersebut ataukah tidak. Metode untuk menilai kelayakan dari calon nasabah kredit diantaranya adalah metode *Credit Scoring*. Metode statistika yang dapat digunakan untuk mengklasifikasikan kondisi status kredit nasabah menggunakan *Credit Scoring* diantaranya adalah metode *k-Nearest Neighbor* (kNN). Kemudian digunakan inovasi untuk meningkatkan keakuratan dalam hal memprediksi suatu klasifikasi, diantaranya seperti yang dikembangkan oleh Breiman (1996) yakni metode klasifikasi yang menggabungkan dua metode yakni *Bootstrap* dan juga *Aggregating (Bagging)*. Maka dalam penulisan ini akan digunakan metode Bagging k-Nearest Neighbor untuk diaplikasikan dalam pengklasifikasian status nasabah kredit lancar dan macet pada nasabah Bank "X".

## 2. TINJAUAN PUSTAKA

### 2.1 Kredit

Menurut Melayu (2004), kredit adalah semua jenis pinjaman yang harus dibayar kembali bersama bunganya oleh peminjam sesuai dengan perjanjian yang telah disepakati. Kredit bermasalah akan berdampak negatif baik bagi kelangsungan hidup bank itu sendiri maupun bagi perekonomian negara, menurut Mahmoedin (2002) kondisi tersebut akan mempengaruhi penghasilan bank, kepercayaan masyarakat kepada bank, menurunkan tingkat kesehatan bank dan juga mempengaruhi modal bank tidak dapat berkembang dengan baik.

### 2.2 *Credit Scoring*

Penggunaan metode *credit scoring* ini dapat membantu pihak yang memberi pinjaman menentukan apakah pengajuan kredit seorang peminjam dapat disetujui (Morrison, 2004). Sedangkan untuk data nasabah yang dibutuhkan dalam penerapan *credit scoring*, diambil dari masing-masing data aplikasi pinjaman nasabah, selain dengan menggunakan program statistik yang berisi tentang sejarah pinjaman yang bersangkutan (Mester, 1997), data juga berisi antara lain mengenai siklus pembayaran tagihan apakah tepat waktu atau tidak, berapa banyak kredit yang masih atau pernah dimiliki yang bersangkutan.

### 2.3 *Data Mining*

Menurut Prasetyo (2014), *data mining* bertujuan untuk memanfaatkan data dalam basis data dengan mengolahnya sehingga menghasilkan informasi baru yang berguna. Dengan kondisi adanya sisi data yang belum dieksplorasi, dan di sisi lain, kemajuan teknik komputasi dan ilmu komputer sudah tumbuh pesat, keterbutuhan akan eksplorasi informasi baru semakin meningkat. *Data mining* menjadi solusi penyelesaian pencarian informasi yang sebelumnya tidak dapat dideteksi secara tradisional dan hanya menggunakan kemampuan analisis manusia.

#### 2.3.1 *Klasifikasi*

Menurut Prasetyo (2014), klasifikasi adalah proses pembelajaran suatu fungsi tujuan (target)  $f$  yang memetakan tiap himpunan atribut  $x$  ke satu dari label kelas himpunan atribut yang didefinisikan sebelumnya. Fungsi targetan disebut juga model klasifikasi. Menurut Gorunescu (2011), proses klasifikasi data didasarkan pada empat komponen, yaitu:

- a. *Kelas*  
Variabel dependen yang berupa kategorikal yang merepresentasikan 'label' yang terdapat pada objeknya. Contohnya: status kredit macet dan lancar.
- b. *Predictor*  
Variabel independen yang direpresentasikan oleh karakteristik (atribut) data. Contohnya: usia, lama bekerja, total pendapatan bersih, ataupun total pinjaman lain yang dimiliki nasabah kredit.
- c. *Training dataset*  
Satu set data yang berisi nilai dari kedua komponen di atas yang digunakan untuk menentukan kelas yang cocok berdasarkan *predictor*.
- d. *Testing dataset*  
Berisi data baru yang akan diklasifikasikan oleh model yang telah dibuat dan akurasi klasifikasi dievaluasi.

### 2.3.2 Pendekatan Teknik Klasifikasi

Teknik klasifikasi merupakan suatu pendekatan sistematis untuk membangun model klasifikasi dari suatu himpunan data masukan. Tiap teknik menggunakan suatu algoritma pembelajaran (*learning algorithm*) untuk mendapatkan suatu model yang paling memenuhi hubungan antara himpunan atribut dan label kelas dalam data masukan. Tujuan dari algoritma pembelajaran adalah untuk membangun model yang berkemampuan baik, yaitu model yang dapat memprediksi label kelas dari *record* yang tidak diketahui kelas sebelumnya dengan lebih akurat. Biasanya *dataset* yang diberikan dibagi menjadi data *training* dan data *testing*. Data *training* digunakan untuk membangun model dan data *testing* digunakan untuk memvalidasi (Hermawati, 2013). Setelah mendapatkan hasil prediksi, dilakukan proses evaluasi performa dari model yang dibangun, perlu dilakukan pengukuran performa, yaitu pengukuran akurasi (*accuracy*) atau tingkat kesalahan (*error rate*). Jika  $f_{ij}$  menotasikan jumlah *record* dari kelas  $i$  yang berada di kelas  $j$  pada saat pengujian (Hermawati, 2013), maka pengukuran akurasi dapat dituliskan sebagai berikut:

$$\text{Error Rate} = \frac{\text{jumlah\_prediksi\_yang\_salah}}{\text{jumlah\_prediksi\_keseluruhan}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}} \quad (1)$$

Dimana:

$f_{11}$  = jumlah data kelompok 1 yang benar diklasifikasikan ke kelompok 1

$f_{00}$  = jumlah data kelompok 0 yang benar diklasifikasikan ke kelompok 0

$f_{10}$  = jumlah data kelompok 1 yang salah diklasifikasikan ke kelompok 0

$f_{01}$  = jumlah data kelompok 0 yang salah diklasifikasikan ke kelompok 1

### 2.3.3 Analisis *k*-Nearest Neighbor

Metode *k*-nearest neighbor (kNN) termasuk dalam kelompok *instance-based learning*. Algoritma ini juga merupakan salah satu teknik *lazy learning*. Metode kNN dilakukan dengan mencari kelompok  $k$  objek dalam data *training* yang paling dekat dengan objek pada data baru atau data *testing* (Wu dan Kumar, 2009)

#### 2.3.3.1 Jarak Euclidian

Semakin dekat jarak kedua data maka akan semakin besar kemiripannya, dan semakin jauh jarak kedua data maka semakin kecil kemiripannya (Prasetyo, 2014). Jarak yang digunakan untuk mengukur jarak antar data pada tugas akhir ini adalah jarak Euclidian. Diketahui permisalan, obyek A dengan hasil pengukuran adalah  $(x_1, x_2, \dots, x_p)$  dan obyek B dengan hasil pengukuran adalah  $(y_1, y_2, \dots, y_p)$  maka jarak euclidian antara A dan B adalah sebagai berikut:

$$d(A, B) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2} \quad (2)$$

#### 2.3.3.2 Penentuan Parameter

Setelah menghitung jarak Euclidian sebagai nilai kedekatan antara data uji (data *testing*) dengan data latih (data *training*). Langkah selanjutnya adalah menentukan nilai  $k$  pada kNN, yang berarti  $k$ -data terdekat dari data *testing*. Jika  $k=1$ , kelas dari satu data *training* sebagai tetangga terdekat (terdekat pertama) dari data *testing* tersebut akan diberikan sebagai

kelas untuk data *testing*, yaitu kelas 1. Jika  $k=2$ , akan diambil 2 tetangga terdekat dari data *training*. Langkah-langkah penentuan nilai parameter  $k$  adalah sebagai berikut:

1. Tentukan parameter  $k$  (jumlah tetangga paling dekat), menggunakan metode *hold-out*.
2. Hitung jarak Euclidian antara data *testing* dengan data *training* yang ada.
3. Urutkan jarak tersebut dimulai dari kelompok yang memiliki jarak terkecil ke yang terbesar.
4. Tentukan jarak terdekat sebanyak nilai  $k$  yang telah ditentukan.
5. Dengan kategori *nearest neighbor* yang paling banyak, maka dapat ditetapkan kelas terbanyak tersebut sebagai kelas dari masing-masing objek data *testing*.

### 2.3.3.3 Algoritma *k-Nearest Neighbor*

Pada metode *k-nearest neighbor* ini, parameter yang dimaksud adalah nilai  $k$ . Nilai  $k$  yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Terdapat berbagai langkah dalam menetapkan  $k$  terbaik, diantaranya adalah proses *cross validation* dengan menggunakan metode *hold-out* (Bishop, 1995) sebagai berikut:

1. Secara acak, bagi data menjadi 2 bagian, yakni data *training*, dan data *testing*.
2. Tentukan nilai parameter *nearest neighbor*  $k = 1$ .
3. Hitung jarak Euclidian antara data *training* dengan data *testing*.
4. Urutkan nilai jarak tersebut, dari yang nilainya terkecil hingga yang terbesar.
5. Tentukan jarak terdekat sebanyak nilai  $k$  yang telah ditentukan.
6. Tetapkan kelas yang terbanyak muncul dari hasil data tetangga terdekat tersebut, sebagai kelas dari data *testing*.
7. Hitunglah *error rate* hasil klasifikasi kelas data *training* dan data *testing* untuk nilai  $k$  tersebut.
8. Kemudian tentukan nilai  $k=k+1$
9. Lakukan kembali langkah 3 – 7 hingga didapati nilai *error rate* yang paling rendah.
10. Lalu tentukan nilai  $k$  yang mempunyai nilai *error rate* yang paling rendah untuk kedua hasil klasifikasi data *training* dan data *testing* sebagai nilai  $k$  yang terbaik, yang akan digunakan untuk mengklasifikasikan data *testing* menggunakan *Bagging*.

### 2.3.3.4 Karakteristik Klasifikasi dengan *k-Nearest Neighbor*

Beberapa karakteristik kNN adalah sebagai berikut:

1. kNN merupakan algoritma yang menggambarkan seluruh data *training* untuk melakukan proses klasifikasi (*complete storage*). Hal ini mengakibatkan proses prediksi yang sangat lama untuk data dalam jumlah yang sangat besar. Tetapi semakin banyak data *training*, maka akan dapat dibuat batas keputusan yang semakin halus.
2. Algoritma kNN tidak membedakan setiap fitur dengan suatu bobot, semua bobot untuk fiturnya bernilai sama untuk satu sama lainnya.
3. Karena kNN masuk kategori *lazy learning* yang menyimpan sebagian atau semua data dan hampir tidak ada proses pelatihan, kNN sangat cepat dalam proses pelatihan (karena memang tidak ada), tetapi sangat lambat dalam proses prediksi.
4. Hal yang rumit adalah menentukan nilai  $k$  yang paling sesuai karena kNN pada prinsipnya memilih tetangga terdekat, parameter jarak juga penting untuk dipertimbangkan sesuai dengan kasus datanya. Jarak Euclidian sangat cocok untuk menggunakan jarak terdekat (lurus) antara dua data.

## 2.4 Metode *Bagging*

Metode *Bagging* merupakan teori yang diusulkan oleh Breiman (1996), yang didasari oleh konsep teori *bootstrap* dan *aggregating* yang menggabungkan kedua manfaat teori tersebut menjadi satu. *Bootstrap* ini diterapkan berdasarkan pada teori *sampling* acak dengan pengembalian (Tibshirani dan Efron, 1993). Oleh karena itu *bootstrap*  $X^b = (X_1^b, X_2^b, \dots, X_n^b)$  (menerapkan pengambilan sampel acak dengan pengembalian) dari data *training* =  $(X_1, X_2, \dots, X_n)$ , dimana penggunaan metode ini dianggap dapat meminimalisir kesalahan dalam pengklasifikasian. Sebagai pengaruhnya, model pada teknik klasifikasi (*classifier*) yang telah dibentuk tersebut kemungkinan mempunyai kinerja yang lebih baik. Pengertian dari proses *aggregating* adalah menggabungkan beberapa *classifier* tersebut. Berikut ini merupakan tahapan penerapan teknik *Bagging*:

1. Lakukan replikasi *bootstrap*  $X^b$  sebanyak  $m$ , dari sejumlah  $n$  data *training*. Ulangi langkah ini untuk  $b = 1, 2, \dots, B$ . Dimana  $m$  adalah banyaknya data yang diambil dari data *training*,  $n$  adalah ukuran sampel dari data *training* dan  $B$  adalah banyaknya replikasi *bootstrap* yang dilakukan.
2. Dengan menggunakan *simple majority vote*, dipilih label yang paling banyak muncul dari hasil penilaian sebagai aturan untuk pengambilan keputusan terakhir.

## 3. METODE PENELITIAN

Data yang digunakan adalah data sekunder nasabah tahun 2011 yang diperoleh dari Bank "X" yang terletak di Provinsi Lampung, dimana terdapat data diri nasabah kredit macet dan kredit lancar di dalamnya. Variabel respon yang digunakan adalah status kredit dari nasabah kredit yang dikategorikan dengan huruf 'L' untuk kredit lancar dan huruf 'M' untuk kredit macet. Variabel prediktor yang digunakan meliputi data usia debitur dalam satuan tahun, lama bekerja dalam satuan tahun, pendapatan bersih dalam satuan rupiah, pinjaman lain dalam satuan rupiah, nominal yang tertera dalam akun tabungan yang dimiliki, rasio hutang terhadap pendapatan. Software atau alat bantu yang digunakan dalam pengolahan data ini adalah Ms.Excel 2010 dan SAS 9.0. Setelah data diperoleh, maka langkah-langkah yang dilakukan dalam menganalisis data adalah:

- a. Menentukan nilai  $k$  terbaik dengan *cross validation* dengan langkah sebagai berikut:
  - 1) Tentukan nilai parameter *nearest neighbor*  $k = 1$ .
  - 2) Hitung jarak Euclidian antara data *training* ke data *testing*
  - 3) Urutkan nilai jarak tersebut, dari yang nilainya terkecil hingga yang terbesar.
  - 4) Tentukan jarak terdekat sebanyak nilai  $k$  yang telah ditentukan.
  - 5) Hitunglah *error rate* hasil klasifikasi kelas data *training* dan data *testing* untuk nilai  $k$  tersebut.
  - 6) Kemudian tentukan nilai  $k = k + 1$
  - 7) Lakukan kembali langkah 2 – 5 hingga diperoleh nilai *error rate* yang paling rendah.
  - 8) Lalu tentukan nilai  $k$  yang mempunyai nilai *error rate* yang paling rendah pada kedua hasil klasifikasi data *training* dan data *testing* sebagai nilai  $k$  yang terbaik, yang akan digunakan untuk mengklasifikasikan data *testing* dengan menggunakan data hasil *bootstrap*.
- b. Melakukan *resampling* pada data *training* menggunakan metode *bootstrap* sebanyak  $B$  kali dengan ukuran sampel sebanyak  $m$ , dimana jumlah  $m$  adalah 80% dari  $n$  data *training*
- c. Menghitung jarak Euclidian setiap data hasil *bootstrap* data *training* kesemua data *testing*

- d. Urutkan objek–objek dimulai dari kelompok yang memiliki jarak terkecil ke yang terbesar.
- e. Menentukan nilai keanggotaan terbesar pada tiap perhitungan jarak terdekat data hasil *bootstrap* dengan data *testing*.
- f. Melakukan *aggregating* dengan aturan *majority vote*, menggunakan nilai keanggotaan dengan jumlah terbesar untuk menentukan kelas dari setiap data yang sudah diperoleh dari hasil *bootstrapping*, untuk menentukan kategori untuk setiap nilai observasi pada data *testing*.
- g. Mengklasifikasikan data observasi menggunakan hasil *aggregating* tersebut.
- h. Melakukan validasi hasil prediksi dengan menghitung nilai keakuratan hasil klasifikasi dengan menghitung nilai *accuracy* dari analisis tersebut.

#### 4. HASIL DAN PENELITIAN

##### 4.1 Penentuan Nilai Parameter Terbaik

Pada metode *k-Nearest Neighbor*, sebelum menghitung jarak Euclidian terdekat dari data yang akan dicari klasifikasi kelasnya, diperlukan proses untuk menentukan nilai parameter terbaik *k* terlebih dahulu. Dalam tugas akhir ini akan digunakan metode *hold-out*. Hasil perbandingan nilai *error rate* nya dapat dilihat pada Tabel 1.

**Tabel 1.**Perbandingan Nilai *Error Rate* Hasil Klasifikasi Kelas Data *Training* dan Data *Testing*

K	Error rate	
	Data <i>Training</i>	Data <i>Testing</i>
1	0	0,313131
2	0,0907	0,363636
3	0,1655	0,393939
4	0,2341	0,515152
5	0,2805	0,565657
6	0,2749	0,434343
7	0,2848	0,414141
8	0,3025	0,464646
9	0,3209	0,484848
10	0,3317	0,515152

Berdasarkan Tabel 1, diperoleh nilai *error rate* terkecil untuk data *training* dan *testing* ada pada nilai  $k = 1$ , yakni *error rate* untuk klasifikasi data *training* adalah sebesar 0, sedangkan *error rate* untuk klasifikasi data *testing* adalah sebesar 0,313131. Maka dapat diambil kesimpulan bahwa nilai *k* terbaik yang digunakan untuk melakukan klasifikasi menggunakan metode *Bagging k-Nearest Neighbor* untuk data *testing* adalah nilai  $k = 1$

##### 4.2 Penentuan Klasifikasi Kelas untuk Observasi Pertama Data *Testing*

Pada metode ini akan digabungkan dua metode untuk mengklasifikasikan kelas dari data *testing*, yakni metode *Bagging* dan *k-Nearest Neighbor*. Sebelum diterapkan metode *k-Nearest Neighbor* akan dilakukan terlebih dahulu metode *Bagging (Bootstrap Aggregating)*. Pertama, proses *bootstrap* yakni pengambilan sampel ulang disertai pengembalian pada data *training*

sebanyak 50 kali dengan jumlah data pada masing-masing hasil *bootstrap* sebanyak 720 data (80%) menggunakan bantuan *software* SAS 9.0. Kedua, proses *aggregating* yaitu pendugaan kelas secara gabungan berdasarkan dugaan data *testing* menggunakan data hasil *bootstrap* yang telah ada dengan aturan *majority vote* (suara terbanyak).

#### 4.2.1 Perhitungan Jarak Euclidian antara Data *Testing* dengan Data *Bootstrap*

Langkah menghitung jarak Euclidian data *bootstrap* dengan data *testing* ini dilakukan hingga observasi ke 720 pada data *bootstrap* pertama. Sehingga didapati hasil perhitungan jarak Euclidian semua observasi seperti yang dapat dilihat pada Tabel 2.

**Tabel 2.** Jarak Euclidian data *Bootstrap* pertama ke data *Testing*

No.	Usia	Lama kerja	Pinjaman lain	Pendapatan bersih	Nominal akun	Rasio hutang	Status kredit	Jarak
1	40	7	0	3.920.069	164.824,2	24,027	Lancar	6.479.309
2	45	14	0	2.222.741	3.600.000	47,901	Macet	2.858.089
3	30	2	0	11.601.567	918.311,4	21,988	Lancar	10.821.890
.	.	.	.	.	.	.	.	.
718	40	15	913.296	4.129.724	14.098.273	55,464	Lancar	7.907.909
719	40	15	913.296	4.129.724	14.098.273	55,464	Lancar	7.907.909
720	40	15	913.296	4.129.724	14.098.273	55,464	Lancar	7.907.909

#### 4.2.2 Penentuan Tetangga Terdekat Sebanyak $k$ pada Setiap data *Bootstrap*

Setelah melakukan perhitungan jarak Euclidian antar data, akan ditentukan tetangga terdekat sebanyak  $k$ , dimana  $k = 1$  adalah nilai terbaik parameter  $k$ , untuk memutuskan kelas dari observasi pertama data *testing* berdasarkan data hasil *bootstrap*. Selanjutnya menghitung jarak euclidian terdekat.

**Tabel 3.** Urutan Jarak dari Nilai yang Terkecil Hingga Terbesar untuk Data *Bootstrap* Pertama

No.	Usia	Lama Kerja	Pinjaman Lain	Pendapatan Bersih	Nominal Akun	Rasio Hutang	Status Kredit	Jarak
715	38	9	408.412	2.173.496	6.391.048	23,647	Lancar	134.460,8
92	50	6	535.991	2.630.509	6.480.000	22,876	Lancar	350.156,6
288	32	1	0	2.581.924	6.215.350	29,853	Lancar	580.969,4
82	44	17	0	2.581.747	6.068.012	29,186	Lancar	647.579
477	43	13	0	2.687.930	6.819.995	19,117	Lancar	726.834,5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
384	53	17	493.090	3.843.320	1,77E+08	30,015	Macet	1,7E+08
294	35	10	0	4.328.601	2,04E+08	34,815	Lancar	1,97E+08
351	34	2	1.188.659	5.756.717	9,94E+08	30,676	Lancar	9,87E+08
352	34	2	1.188.659	5.756.717	9,94E+08	30,676	Lancar	9,87E+08
472	45	3	2.360.289	45.639.712	2,88E+09	13,339	Lancar	2,87E+09

Pada Tabel 3 tersebut, dapat dilihat untuk nilai  $k=1$ , jarak terdekat pertama dari data *bootstrap* pertama ke data *testing* adalah data observasi ke-715, dengan nilai jarak = 134.460,8. Sehingga prediksi kelas menggunakan data *bootstrap* pertama untuk observasi

pertama data *testing* masuk ke dalam kelas kredit Lancar, sesuai dengan kelas dari data tetangga terdekat pertama observasi ke-715.

### 4.2.3 Penentuan Kelas dari Observasi Pertama Data *Testing* Berdasarkan Hasil *Majority Vote*

Setelah melihat hasil prediksi dari 50 data *bootstrap*, langkah selanjutnya adalah menghitung suara terbanyak dari kelas yang muncul tersebut yang telah direkap juga pada Tabel 4.

Hasilnya, untuk observasi pertama data *testing* diklasifikasikan ke dalam status kredit Lancar, karena hasil perhitungan kelas Lancar muncul lebih banyak yakni sebanyak 50 kali, dibandingkan dengan kelas Macet yang tidak muncul sama sekali atau nol.

Lakukan juga secara berulang untuk observasi kedua hingga terakhir pada data *testing*, yakni menentukan kelas dari observasi berdasarkan suara terbanyak dari kelas yang muncul pada hasil prediksi kelas status kredit berdasarkan data *bootstrap* tersebut.

**Tabel 4.** Hasil Prediksi Kelas untuk Status Kredit Berdasarkan Jarak dengan Data *Bootstrap*

n	Hasil Prediksi berdasarkan jarak dengan Data Hasil <i>Bootstrap</i> ke-																				L	M	Majority Vote		
	1	2	3	4	5	6	7	8	9	10	...	40	41	42	43	44	45	46	47	48				49	50
1	L	L	L	L	L	L	L	L	L	L		L	L	L	L	L	L	L	L	L	L	L	50	0	L

### 4.3 Ketepatan Hasil Prediksi Gabungan Menggunakan *Bagging k-Nearest Neighbor*

Pada Tabel 5 telah disajikan data hasil penentuan kelas berdasarkan suara terbanyak untuk 99 observasi data *testing*. Dari 99 hasil prediksi kelas status kredit data *testing* tersebut selanjutnya akan dihitung *error rate* klasifikasi. Setelah di hitung terdapat 33 hasil prediksi yang berbeda dengan data asli kelas dari status kredit pada data *testing*, dan terdapat 66 data yang sesuai dengan data asli kelas dari status kredit tersebut.

**Tabel 5.** Hasil *Majority Vote* untuk Penentuan Kelas Observasi Pertama Data *Testing*

n	Status	Hasil Prediksi berdasarkan jarak dengan Data Hasil <i>Bootstrap</i>																				L	M	Majority Vote	
		1	2	3	4	5	6	7	8	9	10	....	41	42	43	44	45	46	47	48	49				0
1	L	L	L	L	L	L	L	L	L	L	L	...	L	L	L	L	L	L	L	L	L	L	50	0	Lancar
2	L	L	L	M	L	L	L	L	L	L	L	...	L	L	L	L	L	M	L	L	L	L	45	5	Lancar
3	L	L	L	L	L	L	L	L	L	L	L	...	M	L	L	L	L	L	L	L	L	L	45	5	Lancar
4	M	L	L	L	L	L	L	L	L	L	L	...	L	M	L	M	L	L	L	L	L	L	47	3	Lancar
5	L	M	M	M	M	M	M	M	M	M	L	...	L	M	M	M	L	M	M	M	M	L	12	38	Macet
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
95	L	L	L	L	M	L	L	L	L	L	L	...	L	L	L	L	L	L	L	L	L	L	47	2	Lancar
96	L	L	L	L	L	L	L	L	L	L	L	...	L	L	L	L	L	L	L	L	L	L	49	0	Lancar
97	M	L	L	L	L	L	L	L	L	L	L	...	L	L	L	L	L	L	L	L	L	L	49	0	Lancar
98	L	M	M	L	M	M	M	M	M	M	M	...	M	M	M	L	M	M	M	M	M	M	3	46	Macet
99	L	L	L	L	L	L	L	L	L	L	L	...	L	L	L	L	L	L	L	L	L	L	49	0	Lancar

Setelah selesai menentukan kelas untuk semua observasi data *testing*, langkah yang terakhir adalah menghitung ketepatan klasifikasi kelas status kredit tersebut. Cara menghitungnya adalah dengan menghitung *error rate* terlebih dahulu seperti pada Persamaan (1). Selanjutnya cara menghitung *error rate* nya adalah sebagai berikut:

$$Error\ Rate = \frac{Jumlah\ prediksi\ kelas\ yang\ salah}{jumlah\ data\ seluruhnya} = \frac{33}{99} = 0,333333$$

Dengan diketahuinya *error rate* akan dapat dihitung nilai *accuracy* (ketepatan), yakni:

$$Accuracy = 1 - Error\ Rate = 1 - 0,333333 = 0,666667$$

Dengan nilai *accuracy* 0,666667 berarti ketepatan hasil klasifikasi dengan *Bagging k-Nearest Neighbor* adalah sebesar 0,666667 atau 66,7%.

## 5. KESIMPULAN

Berdasarkan hasil dari pembahasan, diperoleh kesimpulan bahwa metode *Bagging k-Nearest Neighbor* dapat digunakan untuk mengklasifikasi status nasabah kredit lancar dan macet pada Bank 'X' di provinsi Lampung. Hasil pengklasifikasian tersebut, selanjutnya digunakan untuk memprediksi calon nasabah yang akan menggunakan layanan jasa kredit. Dalam penelitian ini, diperoleh hasil ketepatan klasifikasi nasabah kredit dengan metode *Bagging k-Nearest Neighbor* sebesar 66,67%.

## 6. DAFTAR PUSTAKA

- [1] Breiman, L. 1996. *Bagging Predictors. Machine Learning* 24 123-140.
- [2] Bishop, C. M. 1995. *Neural Networks For Pattern Recognition*. Clarendon Press: Oxford
- [3] Chapra, M. U. 2000. *Sistem Moneter Islam*, terj. Ikhwan Abidin Basri, *Towards a Just Monetary System*. Jakarta: Gema Insani.
- [4] Gorunescu, F. 2011. *Data Mining: Concepts, Model and Techniques*, Prof. Janusz Kacprzyk and Prof. Lakhmi C. Jain, Eds. Berlin. Jerman: Springer
- [5] Hermawati, F. A. 2013. *Data Mining*. Yogyakarta: ANDI.
- [6] Herprasetyo, B. 2012. *Sukses Ubah Kartu Kredit jadi Modal Usaha*. Tulungagung: Adora media
- [7] Kasmir. 2002. *Bank dan Lembaga Keuangan Lainnya. Edisi Revisi 2002*. Jakarta: PT. Rajagrafindo Persada.
- [8] Kuncoro, M dan Suhardjono. 2002. *Manajemen Perbankan: Teori Dan Aplikasi*. Jogjakarta: BPFE
- [9] Mahmoedin. 2002. *Melacak Kredit Bermasalah*. Jakarta: Pustaka Sinar Harapan.
- [10] Mester, L. J. 1997. *What's The Point of Credit Scoring? Business Review (September)* 3-16.
- [11] Melayu, H. 2004. *Dasar-dasar Perbankan*. Jakarta: CV. Haji Masagung.
- [12] Morrison, J. 2004. *Introduction to Survival Analysis in Business, The Journal of Business Forecasting Methods & Systems* 23 (1) 18-22.
- [13] Prasetyo, E. 2014. *DATA MINING: Mengolah Data Menjadi Informasi Menggunakan Matlab*. Yogyakarta: ANDI.

- [14] Tibshirani, R. J dan Efron, B. 1993. *An Introduction to the Bootstrap*. London: Chapman dan Hall.
- [15] Wu, X. dan Kumar, V. 2009. *The Top Ten Algorithms in Data Mining*. London: CRC Press Taylor & Analitic Machine Intelligence 13: 841-847.