

ANALISIS PERBANDINGAN METODE NEURAL NETWORKS DAN NAÏVE BAYES UNTUK KLASIFIKASI KELAYAKAN PEMBERIAN KREDIT (STUDI KASUS: PT. ADIRA FINANCE KOTA TERNATE)

Ahmad Nur Arfandi¹, Assaf Arief², Muhammad Fhadli³, Rosihan⁴

Program Studi Informatika, Fakultas Teknik, Universitas Khairun
Jl.Jati Metro, Kota Ternate

Email: ahmadarfandi58@gmail.com¹, assaf.arief@unkhair.ac.id², mfhadli@unkhair.ac.id³, rosihan@unkhair.ac.id⁴

Abstrak

Kredit merupakan salah satu jalan pintas yang terkadang diambil oleh masyarakat untuk memenuhi kebutuhannya namun memiliki resiko yang besar bagi perusahaan seperti nasabah kredit macet. Klasifikasi *data mining* dipilih untuk dapat membantu memberikan penyelesaian masalah tersebut. Pada penelitian ini menggunakan klasifikasi dengan model algoritma *Neural Network* dan *Naïve Bayes* kemudian membandingkan untuk melihat model algoritma mana yang memiliki akurasi terbaik dalam mengklasifikasikan status kredit nasabah menggunakan Bahasa pemrograman *python* kemudian dilanjutkan dengan pengukuran menggunakan *confusion matrix*. Hasil algoritma *Neural Network* setelah dilakukan pengujian *confusion matrix* menunjukkan nilai akurasi yang sangat tertinggi adalah 85,0% dan algoritma *Naïve Bayes* setelah dilakukan pengujian *confusion matrix* menunjukkan nilai akurasi yang sangat tertinggi adalah 84,7%. Berdasarkan hasil tersebut hal ini menunjukkan bahwa *Neural Network* sedikit lebih baik dalam melakukan klasifikasi terhadap data nasabah kredit dibandingkan dengan *Naïve Bayes*.

Kata kunci: data mining, klasifikasi, resiko kredit, neural network, naïve bayes.

COMPARATIVE ANALYSIS OF NEURAL NETWORKS AND NAÏVE BAYES METHODS FOR CREDIT APPROVAL CLASSIFICATION (CASE STUDY: PT. ADIRA FINANCE TERNATE CITY)

Abstract

Credit is one of the shortcuts sometimes taken by people to fulfill their needs, but it carries significant risks for companies, such as defaulting credit customers. Data mining classification was chosen to help provide a solution to this problem. In this research, classification using the Neural Network and Naïve Bayes algorithm models was employed and compared to determine which algorithm model had the best accuracy in classifying the credit status of customers using the Python programming language. The evaluation was then conducted using a confusion matrix. The results of the Neural Network algorithm, after testing with the confusion matrix, showed the highest accuracy value at 85.0%, while the Naïve Bayes algorithm, after testing with the confusion matrix, showed the highest accuracy value at 84.7%. Based on these results, it can be concluded that the Neural Network performs slightly better in classifying credit customer data compared to Naïve Bayes.

Keywords: data mining, classification, credit risk, neural network, naïve bayes.

1. PENDAHULUAN

Kredit ialah salah satu jalan pintas yang seringkali digunakan oleh masyarakat untuk memenuhi kebutuhannya. PT. Adira Finance adalah salah satu perusahaan yang dimana salah satunya melayani kegiatan pengkreditan seperti motor baru maupun bekas. Oleh karena itu, sangat penting untuk dapat melakukan identifikasi atau evaluasiajuan kredit [3]. Serta pengawasan nasabah yang telah mengajukan kredit melalui data historis yang ada, agar perusahaan dapat memberikan keputusan

yang tepat dan dapat terhindar dari risiko kredit dalam hal ini kredit macet [1]. Selain itu, perilaku kredit macet disebabkan oleh pelaku nasabah yang selalu terlambat melakukan pembayaran.

Data mining digunakan agar dapat membantu memberikan penyelesaian masalah tersebut dengan menggunakan teknik klasifikasi. *Data mining* merupakan suatu proses menemukan hubungan yang berarti, pola dan kecenderungan dengan memeriksa dari sekumpulan data besar yang tersimpan dalam penyimpanan dengan

menggunakan teknik pengenalan pola seperti teknik statistik dan matematika [8].

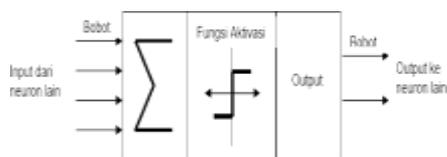
Terdapat beberapa penelitian sejenis dengan melakukan pengklasifikasian nasabah kredit dengan menerapkan teknik data *mining* diantaranya [1] melakukan perbandingan metode *Random Forest* dan *Naive Bayes*. Dan [5] melakukan perbandingan metode *Radial Basis Function* dan *Multilayer Perceptron* terhadap resiko kredit sepeda motor. Ada berbagai algoritma klasifikasi data mining yang dapat digunakan diantaranya: *Random Forest*, *C.45*, *Naive Bayes*, *Radial Basis Function*, *K-nearest neighbors*, *Neural Network* dan lain-lain.

Dari permasalahan yang telah dipaparkan maka penulis akan melakukan perbandingan metode algoritma *Neural Network* dan *Naive Bayes* untuk mengetahui algoritma mana yang nantinya memiliki akurasi terbaik dalam melakukan pengklasifikasian status kredit nasabah PT. Adira *Finance* Kota Ternate. Metode yang menghasilkan akurasi yang terbaik akan disarankan untuk digunakan sebagai komponen dalam melakukan klasifikasi nasabah kredit di PT. Adira *Finance* Kota Ternate

2. METODE PENELITIAN

2.1. Algoritma Neural Network

Terdapat beberapa tipe *Neural Network*, namun hampir semua memiliki komponen yang mirip. Sama halnya dengan otak manusia, *Neural Network* juga terdiri dari dan beberapa *neuron*, dan ada hubungan antara *neuron-neuron* tersebut. *Neuron-neuron* tersebut akan menstrasformasikan informasi yang diterima melalui sambungan keluarannya menuju ke *neuron-neuron* yang lainnya. Pada *Neural Network* hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada suatu bobot tersebut. Gambar 1. menunjukkan struktur *neuron-neuron* pada *Neural Network* [6].



Gambar 1. Struktur *Neuron* pada *Neural Network*

2.2. Algoritma Naive Bayes

Naive Bayes merupakan sebuah metode pengklasifikasian dengan menggunakan metode Probabilitas dan statistik yg didasari oleh *teorema bayes* yang ditemukan oleh Thomas Bayes ilmuwan asal inggris pada abad ke-18 [4], yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *Teorema Bayes Cin utama* dan *Naive Bayes Classifier* ini adalah asumsi yg sangat kuat (*all*) akan dependenes dan masing masing kondisi/kejadian [9]. Tahapan dan proses algoritma *naive bayes* sebagai berikut:

1. Menghitung jumlah kelas/label
2. Menghitung jumlah kasus perkelas
3. Kalikan semua variabel kelas
4. Bandingkan hasil perkelas

2.3. Metode Backward Elimination

Backward Elimination merupakan pemodelan dengan memasukkan semua variabel selanjutnya dihilangkan satu per satu variabel yang tidak signifikan dengan melakukan pengujian model dan dilakukan secara terus menerus sampai tidak ada variabel yang tidak signifikan. Dalam prosedur *Backward Elimination*, model dimulai dengan semua atribut yang akan digunakan, dan atribut dengan statistic parsial paling sedikit dihilangkan (menghilangkan atribut-atribut yang tidak relevan) [2]. Tahapan metode *backward* ialah sebagai berikut:

1. Membentuk model dengan meregresikan variabel respon Y dengan seluruh variabel prediktor.
2. Hapus variabel prediktor satu per satu dengan menguji parameter menggunakan F_{tes} parsial. Nilai $F_{partial}$ terkecil dibandingkan dengan F_{tabel} .
 - a. Apabila $F_{parsial} < F_{tabel}$, maka X dihilangkan dari model dan dilanjutkan dengan pembuatan model baru tanpa variabel tersebut.
 - b. Apabila $F_{parsial} > F_{tabel}$, maka proses dihentikan artinya tidak ada variabel yang perlu dihilangkan dan persamaan terakhir tersebut yang digunakan/dipilih.

2.4. Cross Validation

Cross Validation merupakan salah satu teknik validasi dari teknik data *mining* dengan menggunakan seluruh data yang ada sebagai *testing* dan *training* nantinya. Metode ini membagi data menjadi data *training* dan *testing* berulang sebanyak jumlah K dengan data sebanyak 1/K menjadi *testing* dan sisanya sebagai *training* [2]. *Cross Validation* merupakan salah satu cara membagi seluruh data yang digunakan menjadi data *training* dan data *testing*.

2.5. Confusion Matrix

Confusion Matrix ialah metode yang digunakan untuk menentukan akurasi pengenalan klasifikasi. Akurasi yang biasanya dilakukan dengan memilih sampel dari data referensi dan membandingkan kelas pada data referensi ini dengan kelas yang dihasilkan dari klasifikasi sampel [2]. *Confusion Matrik* dapat dilihat pada Tabel 1:

Tabel 1. *Confusion Matrix*

Klasifikasi yang benar	Klasifikasi sebagai	
	+	-
+	<i>true positive</i>	<i>false negative</i>
-	<i>false positive</i>	<i>true negative</i>

Keterangan:

- TP = *true positive*, yang diklasifikasikan positif
- TN = *true negative*, yang diklasifikasikan negatif
- FP = *false positive*, yang diklasifikasikan negatif
- FN = *false negative*, yang diklasifikasikan positif

Untuk menghitung tingkat akurasi pada matriks digunakan rumus :

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \dots\dots\dots (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \dots\dots\dots (3)$$

Untuk nilai AUC pada klasifikasi data *mining* dapat dibagi menjadi beberapa yaitu [2]:

- 0.90 – 1.00 = (*Excellent classification*)
- 0.80 – 0.90 = (*Good classification*)
- 0.70 – 0.80 = (*Fair classification*)
- 0.60 – 0.70 = (*Poor classification*)
- 0.50 – 0.60 = (*Failure*)

3. HASIL DAN PEMBAHASAN

3.1. Dataset

Dataset dalam penelitian ini bersumber dari data historis nasabah kredit motor baru pada PT. Adira Finance Kota Ternate yang berjumlah 1000 *record* yang macet maupun lancar. Memiliki 11 variabel atribut dan 1 variabel label. Berikut *dataset* nasabah kredit yang digunakan pada panelitian ini.

Gambar 2. *Dataset* nasabah kredit

3.2. Data Preprocessing

Dari *dataset* yang peneliti gunakan masih dalam bentuk mentah dan akan diolah terlebih dahulu atau diubah dalam bentuk yang akan lebih mudah dipahami oleh sistem atau komputer.

Pada tahap ini, kita akan mengubah data menjadi bentuk yang sesuai dengan metode analisis. Peneliti akan melakukan normalisasi data untuk menskalakan nilai data dalam rentang tertentu (0.0 hingga 1.0). Proses transformasi data dilakukan dengan teknik *min-max scaling* dan Teknik *Label*

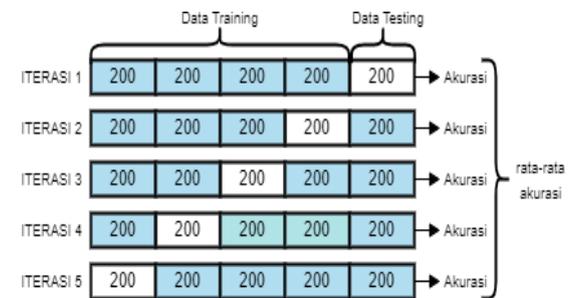
Encoding. Adapun tahapan *Label Encoder* yaitu akan mengubah variabel kategori menjadi variabel numerik yang diwakili oleh integer. Adapun tahapan *min-max scaling* dapat dilakukan dengan rumus pada persamaan 1.

$$x' = \frac{x - \text{Min}_x}{\text{Max}_x - \text{Min}_x} \dots\dots\dots (4)$$

Dimana x merupakan data yang akan dilakukan normalisasi dan x' adalah data yang telah dinormalisasi. Min_x adalah nilai minimum dari keseluruhan data dan max_x adalah nilai maksimum dari keseluruhan data [7].

3.3. Pembagian Data Training dan Testing

Dataset dibagi menjadi 2 yaitu data *training* dan data *testing* secara *random* menggunakan fungsi *KFold* dari *library sklearn* yang nantinya digunakan pada algoritma *Neural Network* dan *Naïve Bayes*. Dapat dilihat pada gambar 3.



Gambar 3. *Pembagian* data *training* dan *testing*

3.4. Implementasi Metode Neural Network

Pada tahap ini akan dilakukan pemodelan dan pengujian performa Algoritma *Neural Network*. peneliti akan mencoba memodelkan menggunakan 1 *hidden layer* dan kedua peneliti akan mencoba menggunakan 2 *hidden layer* dengan menggunakan jumlah *neuron* yang berbeda di tiap *hidden layer*. Tabel percobaan menggunakan 1 *hidden layer* dan 2 *hidden layer* dapat dilihat pada tabel 2 dan tabel 3.

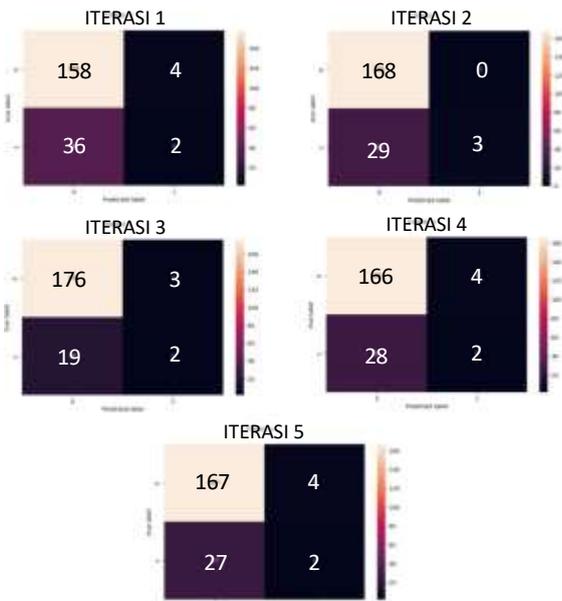
Tabel 2 Percobaan menggunakan 1 *hidden layer*

No	Percobaan	<i>Hidden layer 1</i>
1	Pertama	100 <i>Neuron</i>
2	Kedua	10 <i>Neuron</i>
3	Ketiga	6 <i>Neuron</i>

Table 3 Percobaan menggunakan 2 *hidden layer*

No	Percobaan	<i>Hidden layer 1</i>	<i>Hidden layer 2</i>
1	Pertama	100 <i>Neuron</i>	100 <i>Neuron</i>
2	Kedua	10 <i>Neuron</i>	10 <i>Neuron</i>
3	Ketiga	6 <i>Neuron</i>	3 <i>Neuron</i>

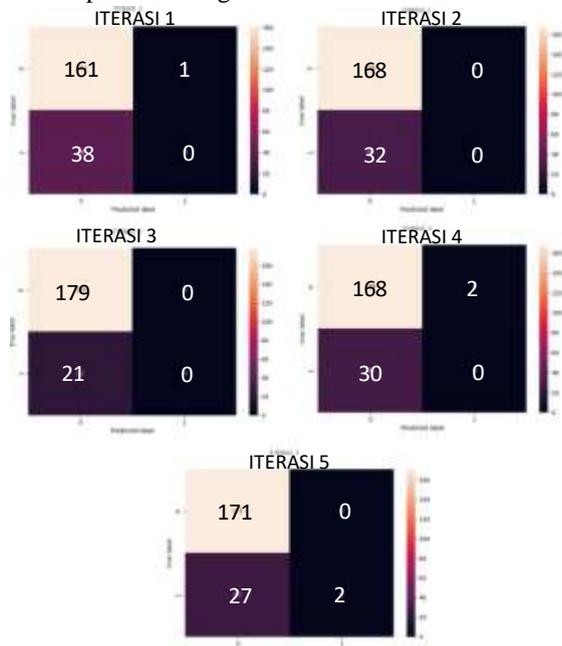
1. Tanpa menggunakan *Backword Elimination*
- a. Menggunakan 1 *hidden layer* percobaan pertama dengan 100 *neuron*



Gambar 4. Confusion Matrix dari Itarasi 1 – 5 pada Model MLP (100)

Jadi, nilai rata-rata dari akurasi *Neural Network* menggunakan 1 *hidden layer* dengan 100 *neuron* dengan menerapkan pembagian data menggunakan *5-fold cross validation* sebesar 0,846 atau sebesar 84.6%.

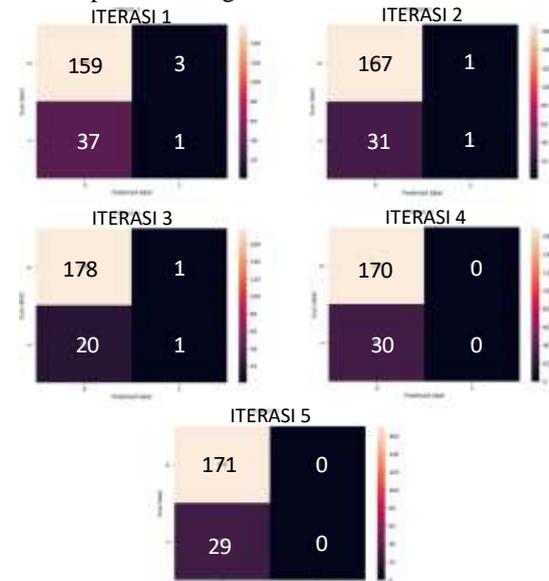
- b. Menggunakan 1 *hidden layer* percobaan pertama dengan 10 *neuron*



Gambar 5. Confusion Matrix dari Itarasi 1 – 5 pada Model MLP (10)

Jadi, nilai rata-rata dari akurasi *Neural Network* menggunakan 1 *hidden layer* dengan 10 *neuron* dengan menerapkan pembagian data menggunakan *5-fold cross validation* sebesar 0,849 atau sebesar 84.9%.

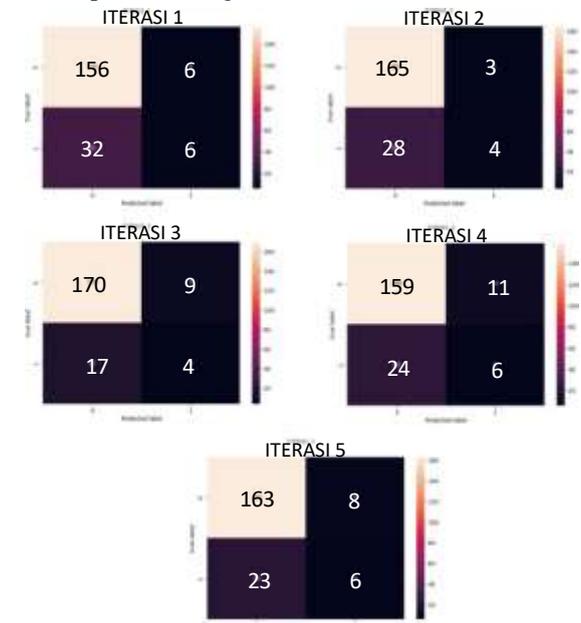
- c. Menggunakan 1 *hidden layer* percobaan pertama dengan 6 *neuron*



Gambar 6. Confusion Matrix dari Itarasi 1 – 5 pada Model MLP (6)

Jadi, nilai rata-rata dari akurasi *Neural Network* menggunakan 1 *hidden layer* dengan 6 *neuron* dengan menerapkan pembagian data menggunakan *5-fold cross validation* sebesar 0,848 atau sebesar 84.8%.

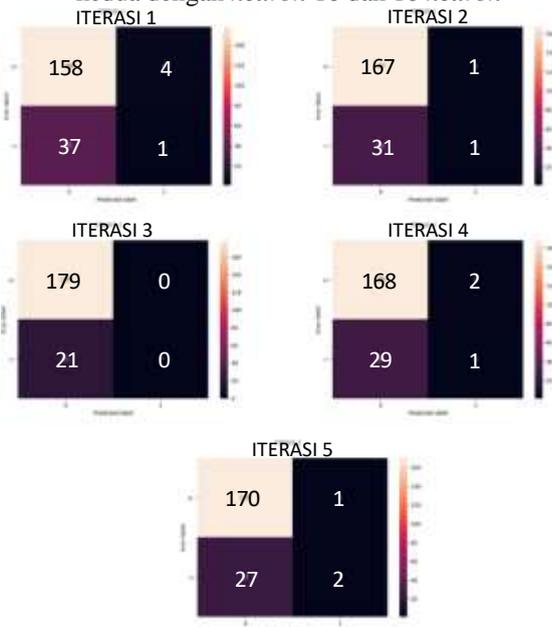
- d. Menggunakan 2 *hidden layer* percobaan pertama dengan *neuron* 100 dan 100 *neuron*



Gambar 7. Confusion Matrix dari Itarasi 1 – 5 pada Model MLP (100, 100)

Jadi, nilai rata-rata dari akurasi *Neural Network* menggunakan 100 *neuron* pada *hidden layer* 1 dan 100 *neuron* pada *hidden layer* 2 dengan menerapkan pembagian data menggunakan *5-fold cross validation* sebesar 0,839 atau sebesar 83.9%.

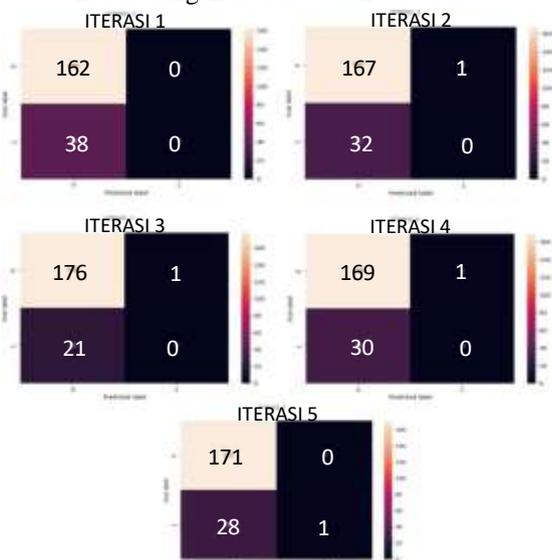
e. Menggunakan 2 *hidden layer* percobaan kedua dengan *neuron* 10 dan 10 *neuron*



Gambar 8. Confusion Matrix dari Itarasi 1 – 5 pada Model MLP (10, 10)

Jadi, nilai rata-rata dari akurasi *Neural Network* menggunakan 10 *neuron* pada *hidden layer* 1 dan 10 *neuron* pada *hidden layer* 2 dengan menerapkan pembagian data menggunakan *5-fold cross validation* sebesar 0,847 atau sebesar 84.7%.

f. Menggunakan 2 *hidden layer* percobaan kedua dengan *neuron* 6 dan 3 *neuron*



Gambar 9. Confusion Matrix dari Itarasi 1 – 5 pada Model MLP (6,3)

Jadi, nilai rata-rata dari akurasi *Neural Network* menggunakan 6 *neuron* pada *hidden layer* 1 dan 3 *neuron* pada *hidden layer* 2 dengan menerapkan pembagian data menggunakan *5-fold cross validation* sebesar 0,849 atau sebesar 84.9%.

2. Menggunakan *Backward Elimination*

Pada tahapan ini mengeliminasi variabel atribut atau fitur satu persatu, setelah itu di uji menggunakan metode *Neural Network*. Tahapan yang dilakukan seperti tahapan sebelumnya namun menghilangkan satu variabel atribut atau fitur kemudian diuji. Hasil akurasi rata-rata tiap menghilangkan satu variabel atribut atau fitur dapat dilihat pada Tabel 3.

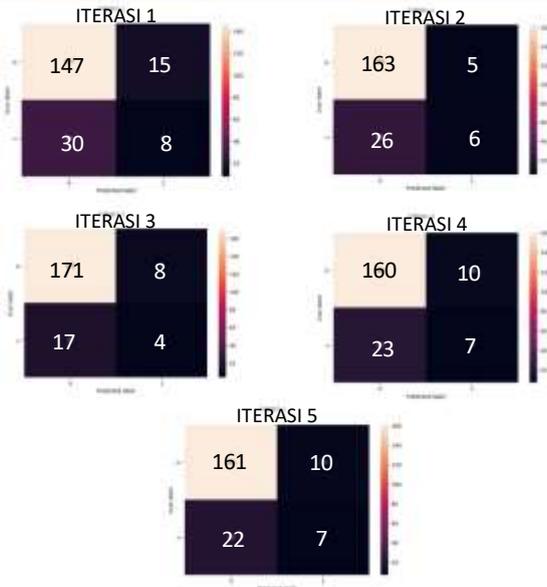
Table 3. Mengeliminasi *K_features* Satu Persatu pada Model *Neural Network*

<i>K_features</i>	Nilai rata-rata akurasi tiap percobaan					
	1 <i>hidden layer</i>			2 <i>hidden layer</i>		
	(100)	(10)	(6)	(100, 100)	(10,10)	(6,3)
10	0,850	0,845	0,850	0,846	0,850	0,850
9	0,850	0,848	0,850	0,849	0,850	0,850
8	0,850	0,849	0,849	0,846	0,850	0,848
7	0,847	0,850	0,849	0,842	0,850	0,850
6	0,847	0,849	0,850	0,839	0,850	0,850
5	0,849	0,849	0,850	0,843	0,850	0,850
4	0,849	0,843	0,850	0,838	0,850	0,850
3	0,850	0,850	0,850	0,834	0,847	0,850
2	0,844	0,850	0,850	0,831	0,850	0,850
1	0,847	0,850	0,849	0,838	0,846	0,850

Pada *K_features* = 10, variabel atribut atau fitur yang digunakan yaitu penghasilan, pendidikan, pekerjaan, tanggungan, lama tinggal, DP net, gender, status pernikahan, status tempat dan tenor kemudian diuji. *K_features* = 9, variabel atribut atau fitur yang digunakan yaitu penghasilan, pendidikan, pekerjaan, tanggungan, lama tinggal, DP net, gender, status pernikahan dan status tempat kemudian diuji. *K_features* = 8, variabel atribut atau fitur yang digunakan yaitu penghasilan, pendidikan, pekerjaan, tanggungan, lama tinggal, DP net, gender dan status pernikahan kemudian diuji. *K_features* = 7, variabel atribut atau fitur yang digunakan yaitu penghasilan, pendidikan, pekerjaan, tanggungan, lama tinggal, DP net dan gender kemudian diuji. *K_features* = 6, variabel atribut atau fitur yang digunakan yaitu penghasilan, pendidikan, pekerjaan, tanggungan, lama tinggal dan DP net kemudian diuji. *K_features* = 5, variabel atribut atau fitur yang digunakan yaitu penghasilan, pendidikan, pekerjaan, tanggungan dan lama tinggal kemudian diuji. *K_features* = 4, variabel atribut atau fitur yang digunakan yaitu penghasilan, pendidikan, pekerjaan dan tanggungan kemudian diuji. *K_features* = 3, variabel atribut atau fitur yang digunakan yaitu penghasilan, Pendidikan dan pekerjaan kemudian diuji. *K_features* = 2, variabel atribut atau fitur yang digunakan yaitu penghasilan dan Pendidikan kemudian diuji. *K_features* = 1, variabel atribut atau fitur yang digunakan yaitu hanya penghasilan kemudian diuji.

3.5. Implementasi Metode *Naïve Bayes*

1. Tanpa menggunakan *Backword Elimination*



Gambar 10. Confusion Matrix dari Itarasi 1 – 5 pada Model *Naïve Bayes*

Jadi, nilai rata-rata dari akurasi dengan menerapkan pembagian data menggunakan *5-fold cross validation* pada *Naïve Bayes* sebesar 0,834 atau sebesar 83%.

2. Menggunakan *Backward Elimination*

Table 4. Mengeliminasi *K_features* Satu Persatu pada Model *Naïve Bayes*

No.	<i>K_features</i>	akurasi
1	['Penghasilan', 'Pendidikan', 'Pekerjaan', 'Tanggungan', 'Lama_tinggal', 'DP_net', 'Gender', 'Status_pernikahan', 'Status_tempat' dan 'Tenor']	0,837
2	['Penghasilan', 'Pendidikan', 'Pekerjaan', 'Tanggungan', 'Lama_tinggal', 'DP_net', 'Gender', 'Status_pernikahan' dan 'Status_tempat']	0,836
3	['Penghasilan', 'Pendidikan', 'Pekerjaan', 'Tanggungan', 'Lama_tinggal', 'DP_net', 'Gender' dan 'Status_pernikahan']	0,835
4	['Penghasilan', 'Pendidikan', 'Pekerjaan', 'Tanggungan', 'Lama_tinggal', 'DP_net', dan 'Gender']	0,833
5	['Penghasilan', 'Pendidikan', 'Pekerjaan', 'Tanggungan', 'Lama_tinggal' dan 'DP_net']	0,835
6	['Penghasilan', 'Pendidikan', 'Pekerjaan', 'Tanggungan' dan 'Lama_tinggal']	0,838

7	['Penghasilan', 'Pendidikan', 'Pekerjaan', dan 'Tanggungan']	0,842
8	['Penghasilan', 'Pendidikan', dan 'Pekerjaan']	0,847
9	['Penghasilan', dan 'Pendidikan']	0,846
10	['Penghasilan']	0,847

Pada tabel 4. menampilkan nilai rata-rata akurasi tiap menghilangkan variabel atribut atau fitur. baris ke-1 dimulai dengan menghilangkan variabel atribut OTR kemudian di uji. Baris ke-2 dengan menghilangkan variabel atribut OTR dan tenor kemudian di uji. Baris ke-3 dengan menghilangkan variabel atribut OTR, tenor dan status tempat kemudian di uji. Baris ke-4 dengan menghilangkan variabel atribut OTR, tenor, status tempat dan status pernikahan kemudian di uji. Baris ke-5 dengan menghilangkan variabel atribut OTR, tenor, status tempat, status pernikahan dan gender kemudian di uji. Baris ke-6 dengan menghilangkan variabel atribut OTR, tenor, status tempat, status pernikahan, gender dan DP net kemudian di uji. Baris ke-7 dengan menghilangkan variabel atribut OTR, tenor, status tempat, status pernikahan, gender, DP net dan lama tinggal kemudian di uji. Baris ke-8 dengan menghilangkan variabel atribut OTR, tenor, status tempat, status pernikahan, gender, DP net, lama tinggal dan tanggungan kemudian di uji. Baris ke-9 dengan menghilangkan variabel atribut OTR, tenor, status tempat, status pernikahan, gender, DP net, lama tinggal, tanggungan dan pekerjaan kemudian di uji. Baris ke-10 dengan menghilangkan variabel atribut OTR, tenor, status tempat, status pernikahan, gender, DP net, lama tinggal, tanggungan, pekerjaan dan Pendidikan kemudian di uji. Hasil akurasi paling tertinggi yang didapatkan sebesar 0.847 atau sebesar 84.7%.

3.6. Analisis

Tabel 5. Perbandingan Kinerja Evaluasi Neural Network dan Naïve Bayes dengan Backward Elimination

Metode	Keterangan	Akurasi tanpa Backward Elimination	Akurasi dengan Backward Elimination									
			1	2	3	4	5	6	7	8	9	10
Naïve Bayes	-	0,834	0,837	0,836	0,835	0,833	0,835	0,838	0,842	0,847	0,846	0,847
Neural Network	Hidden (100)	0,846	0,847	0,844	0,85	0,849	0,849	0,847	0,847	0,85	0,85	0,85
	Hidden (100,100)	0,839	0,838	0,831	0,834	0,838	0,843	0,839	0,842	0,846	0,849	0,846
	Hidden (10)	0,849	0,845	0,848	0,849	0,85	0,849	0,849	0,843	0,85	0,85	0,85
	Hidden (10,10)	0,847	0,846	0,85	0,847	0,85						
	Hidden (6)	0,848	0,849	0,85	0,85	0,85	0,85	0,85	0,849	0,849	0,85	0,85
	Hidden (6,3)	0,849	0,85	0,85	0,848	0,85						

Dari perbandingan Kinerja Evaluasi Neural Network dan Naïve Bayes dan dengan menerapkan fitur Backward Elimination akurasi yang di dapatkan paling tertinggi sebesar 0,850.

Metode Neural Network mendapatkan akurasi yang paling tertinggi dibandingkan dengan akurasi yang didapatkan metode Naïve Bayes. Pada model Neural Network menggunakan 2 hidden layer dengan jumlah 6 neuron dan 3 neuron yang paling banyak mendapatkan akurasi tertinggi dibandingkan dengan percobaan yang lain.

4. KESIMPULAN

1. Untuk klasifikasi status kredit, dapat dilakukan menggunakan metode Neural Network dan Naïve Bayes karena memiliki kemampuan untuk melakukan supervised learning, dengan melatih dan mempelajari data untuk mendapatkan output yang diharapkan.
2. Hasil akurasi terbaik pada metode Neural Network menggunakan 1 hidden layer dari 3 kali percobaan dengan neuron yang berbeda terdapat pada percobaan kedua dengan menggunakan 10 neuron dengan nilai sebesar 0,849 atau sebesar 84.9%.
3. Hasil akurasi terbaik pada metode Neural Network menggunakan 2 hidden layer dari 3 kali percobaan dengan neuron yang berbeda terdapat pada percobaan ketiga dengan dengan 6 neuron pada hidden layer 1 dan 3 neuron pada hidden layer 2 dengan nilai sebesar 0,849 atau sebesar 84.9%.
4. Berdasarkan hasil perhitungan selisih antara pemodelan Neural Network dan Naïve Bayes

tanpa Backward Elimination menunjukkan nilai akurasi pada model Naïve Bayes paling tinggi dengan nilai sebesar 0,834 atau sebesar 83.4%. Kemudian nilai akurasi pada model Neural Network paling tinggi dengan nilai sebesar 0,849 atau sebesar 84.9%. Hal ini menunjukkan bahwa Neural Network sedikit lebih baik dalam melakukan klasifikasi terhadap data nasabah kredit di Kantor Adira Finance dengan selisih rata-rata nilai akurasi sebesar 0.015 atau sebesar 1.5%.

5. Berdasarkan hasil perhitungan selisih antara pemodelan Neural Network dan Naïve Bayes dengan fitur Backward Elimination menggunakan 10 variabel atribut / fitur menunjukkan nilai akurasi pada model Naïve Bayes paling tinggi dengan nilai sebesar 0,847 atau sebesar 84.7%. Kemudian nilai akurasi pada model Neural Network paling tinggi dengan nilai sebesar 0,850 atau sebesar 85.0%. Hal ini menunjukkan bahwa Neural Network sedikit lebih baik dalam melakukan klasifikasi dalam kasus ini dengan selisih rata-rata nilai akurasi sebesar 0.003 atau sebesar 0.03%.

5. DAFTAR PUSTAKA

- [1] Bawono, B., & Wasono, R. (2019). Perbandingan Metode Random Forest dan Naive Bayes. Jurnal Sains Dan Sistem Informasi, 3(7), 343–348. <http://prosiding.unimus.ac.id>
- [2] Ghani, A. D., Salman, N., & Mustikasari. (2019). Algoritma k-Nearest Neighbor Berbasis Backward Elimination Pada Client

- Telemarketing. Prosiding Seminar Ilmiah Sistem Informasi Dan Teknologi Informasi, 8(2), 141–150.
- [3] Hadiano, N., Novitasari, H. B., & Rahmawati, A. (2019). Klasifikasi Peminjaman Nasabah Bank Menggunakan Metode Neural Network. *Jurnal Pilar Nusa Mandiri*, 15(2), 163–170. <https://doi.org/10.33480/pilar.v15i2.658>
- [4] Handayani, F., & Pribadi, F. S. (2015). Implementasi Algoritma Naive Bayes Classifier dalam Pengklasifikasian Teks Otomatis Pengaduan dan Pelaporan Masyarakat melalui Layanan Call Center 110. *Jurnal Teknik Elektro*, 7(1), 19–24.
- [5] Lubis, C. P. (2022). Perbandingan Metode Radial Basis Function dan Multilayer Perceptron Terhadap Resiko Kredit Sepeda Motor. *Infosys (Information System) Journal*, 7(1), 25. <https://doi.org/10.22303/infosys.7.1.2022.25-33>
- [6] S. Kusumadewi. (2004). Membangun Jaringan Syaraf Tiruan Menggunakan Matlab Dan Excel Link (F. W. Nurwiyati (ed.); pertama). Graha Ilmu.
- [7] Siregar, B. (2021). Metode Statistika. https://rstudio-pubs-static.s3.amazonaws.com/805666_02a47e7c20174d8ba01da72f16c951db.html#
- [8] Sumiah, A., & Mirantika, N. (2020). Perbandingan Metode K-Nearest Neighbor dan Naive Bayes untuk Rekomendasi Penentuan Mahasiswa Penerima Beasiswa pada Universitas Kuningan. *Buffer Informatika (Jurnal Ilmiah Teknik Informatika)*, 6(April).
- [9] Widiyanto, M. H. (2019). Algoritma Naive Bayes. Binus University. Tersedia di: <https://binus.ac.id/bandung/2019/12/algoritma-naive-bayes/>. Diakses tanggal, 22.