Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 1, February 2024, pp. 77-87

DOI: https://doi.org/10.52436/1.jutif.2024.5.1.1195
p-ISSN: 2723-3863
e-ISSN: 2723-3871

COMPARISON OF ACCURACY LEVELS OF RANDOM FOREST AND K-NEAREST NEIGHBOR (KNN) ALGORITHMS FOR CLASSIFYING SMOOTH BANK CREDIT PAYMENTS

Bayu Aji Santoso[*1], Kusrini[2], Anggit Dwi Hartanto[3]

[1,2,3]PJJ Masters in Informatics Engineering, Universitas AMIKOM Yogyakarta
Email: [1]bayuajisantoso@students.amikom.ac.id, [2]kusrini@amikom.ac.id, [3]anggit@amikom.ac.id

(Article received: July 6, 2023; Revision: August 1, 2023; Published: February 31, 2024)

Abstract

Providing credit is one of the bank offers offered to customers, but extending credit to customers who are not appropriate can cause problems such as customers who do not pay installments on time and even delay payment of installments for several months until bad credit occurs so that this can be detrimental to the bank. Therefore, in this study a comparative method will be carried out to find out which method is the best in classifying the smoothness of bank credit payments. It is hoped that the results of the research can be used as material for consideration by the bank in the selection of bank credit customers. In this study using a dataset from the UCI Machine Learning Repository, the credit payment data totaled 29,998. The dataset is split by dividing 70% train data and 30% test data with the amount of each data, namely 24000 train data and 6000 test data. Meanwhile, the labels used are Eligible and Ineligible. In this study, implementing the data mining process using the CRISP-DM framework and using the Python programming language. From the results of the evaluation using the confusion matrix, the best accuracy value was obtained for the random forest algorithm, namely 82.22%, precision of 80.44%, recall of 82.22% and f1-score of 80.0%. Meanwhile, the KNN algorithm obtains an accuracy value of 81.55%, a precision of 79.5%, a recall of 81.55% and an f1-score of 79.11%. Based on the results of this evaluation, the Random Forest algorithm has the best accuracy compared to the KNN algorithm in classifying bank credit payments.

Keyword: *Comparison, KNN, Credit, Giving, Random Forest.*

KOMPARASI TINGKAT AKURASI ALGORITMA *RANDOM FOREST* DAN K-*NEAREST NEIGHBOR* (KNN) UNTUK MENGKLASIFIKASI KELANCARAN PEMBAYARAN KREDIT BANK

Abstrak

Pemberian kredit merupakan salah satu penawaran bank yang ditawarkan kepada nasabah, namun pemberian kredit kepada nasabah yang tidak tepat dapat menimbulkan permasalahan seperti nasabah yang tidak membayar angsuran tepat waktu bahkan menunda sampai beberapa bulan pembayaran angsuran hingga terjadinya kredit macet sehingga hal ini dapat merugikan pihak bank. Oleh karena itu, dalam penelitian ini akan melakukan komparasi metode untuk mengetahui metode mana yang terbaik dalam melakukan klasifikasi kelancaran pembayaran kredit bank. Hasil penelitian diharapkan dapat dijadikan bahan pertimbangan pihak bank dalam upaya pemilihan nasabah kredit bank. Penelitian ini menggunakan dataset dari UCI Machine Learning Repository merupakan data pembayaran kredit yang berjumlah 29.998. Selanjutnya dataset dilakukan proses split dengan pembagian 80% data train dan 20% data test dengan jumlah masing-masing data yaitu data train 24000 dan data test 6000. Sedangkan, label yang digunakan yaitu Layak dan Tidak Layak. Kemudian proses sealnjutnya dilakukan pengimplementasian proses *data mining* menggunakan kerangka kerja CRISP-DM dan menggunakan bahasa pemrograman *Python*. Dari hasil evaluasi menggunakan confusion matrix mendapatkan nilai akurasi terbaik pada algoritma random forest yaitu 82.22 %, precision sebesar 80.44 %, recall sebesar 82.22 % dan f1-score sebesar 80.0 %. Sedangkan, pada algoritma KNN mendapatkan nilai accuracy sebesar 81.55 %, precision sebesar 79.5 %, recall sebesar 81.55 % dan f1-score sebesar 79.11 %. Berdasarkan hasil evaluasi tersebut bahwa algoritma Random Forest mendapatkan akurasi terbaik dibandingkan dengan algoritma KNN dalam mengklasifikasi kelancaran pembayaran kredit bank.

Kata kunci: Komparasi, KNN, Kredit, Pemberian, *Random Forest*

1. PENDAHULUAN

Bank merupakan perusahaan yang memiliki data yang besar yang tersimpan di dalam *database* dan diolah menghasilkan sebuah informasi yang saling berkaitan tentang nasabah, data tersebut serta dapat digunakan untuk menjaga hubungan antar bank dengan nasabah yang valid, sehingga berguna untuk menentukan secara individual tentang penawaran produk bank [1]. Kredit merupakan penyediaan uang atau tagihan yang dapat dipersamakan dengan itu berdasarkan persetujuan atau kesepakatan antar pihak bank dengan nasabah [2]. Pemberian kredit merupakan salah satu penawaran bank yang ditawarkan kepada nasabah dengan tujuan untuk mencapai kebutuhan, baik dalam bidang usaha atau pemenuhan kebutuhan sehari-hari. Dalam melakukan pemberian kredit kepada nasabah, pihak bank perlu melakukan penilaian terhadap nasabahnya [3]. Hal tersebut diperlukan agar tidak menimbulkan permasalahan, salah satunya adalah nasabah yang tidak membayar angsuran tepat waktu bahkan menunda sampai beberapa bulan pembayaran angsuran hingga terjadinya kredit macet [4]. Tidak hanya itu, terdapat beberapa kali kasus penipuan bank oleh sejumlah nasabah pemegang kartu kredit karena sebelumnya calon nasabah pemegang kartu kredit tidak mengerti syarat yang diajukan atau bahkan dengan sengaja melakukan hal penipuan untuk kepentingan nasabah sendiri sehinggal hal-hal tersebut dapat merugikan pihak bank [5]. Selain itu dalam proses tersebut pihak bank masih melakukannya dengan cara konvensional yang tentunya sangat tidak efektif apabila jumlah nasabah sangat banyak, sehingga pihak bank memerlukan waktu yang lama dalam melakukan proses evaluasi [6]. Dari permasalahan yang terjadi, diperlukan solusi yang dapat menyelesaikan permasalahan tersebut yaitu dengan menerapkan teknik *data mining* yaitu klasifikasi untuk menentukan kategori nasabah mana yang layak atau tidak layak untuk mengajukan pinjaman berikutnya.

Metode klasifikasi yang dapat digunakan seperti *Decision Tree, Naïve Bayes, Artificial Neural Network* (ANN), Regresi, *Support Vector Machine, Random Forest* dan K-*Nearest Neighbor* (KNN) dan metode lainnya [7]. Penggunaan metode klasifikasi dalam penelitian ini yaitu metode K-*Nearest Neighbor* (KNN) dan *Random Forest*. KNN merupakan metode yang terawasi dimana hasil dari uji data yang baru diklasifikasikan berdasarkan kelas mayoritas [8]. Akan tetapi, terdapat permasalahan pada KNN yaitu perolehan nilai akurasi yang cenderung lebih rendah dibanding metode klasifikasi lainnya. Hal ini dibuktikan oleh penelitian terdahulu yang dilakukan oleh Ginting, et. al. dengan mendapatkan akurasi KNN lebih rendah yaitu sebesar 66.35% [9]. Selain itu, hal ini juga dibuktikan oleh penelitian sebelumnya yang dilakukan oleh gultom dan simanjuntak dengan menghasilkan akurasi KNN lebih rendah dibanding dengan akurasi *random forest* yaitu tingkat akurasi KNN sebesar 76.78% dan *random forest* sebesar 86.56% pada klasifikasi pengobatan kutil [10]. Oleh karena itu, dalam penelitian ini akan melakukan percobaan menggunakan algoritma KNN dan *Random Forest* dalam mengklasifikasi kelancaran pembayaran kredit bank untuk mengetahui performa dari *Random Forest* dan KNN dengan dataset yang digunakan dari UCI *Machine Learning Repository*.

UCI *Machine Learning Repository* merupakan kumpulan *database*, teori domain, dan generator data yang digunakan oleh komunitas *machine learning* untuk menganalisis secara empiris algoritma *machine learning* [11] [12].

Dari uraian diatas, dalam penelitian ini melakukan komparasi metode *Random Forest* dan KNN dalam melakukan klasifikasi kelancaran pembayaran kredit bank. Alasan menggunakan metode *Random Forest* yaitu menghasilkan akurasi yang lebih tinggi dari metode lain, dapat mengatasi data dalam jumlah yang besar secara efisien, dan tidak terdapat pemangkasan variabel seperti pada algoritma pohon klasifikasi tunggal [13]. Sedangkan, metode KNN yaitu dapat menghasilkan data yang lebih akurat dan efektif apabila memliki *training* data yang cukup besar [14]. Selain itu, alasan lain pemilihan algoritma KNN dan *random forest* adalah pada penelitian-penelitian yang telah dijelaskan algoritma *Random Forest* dan KNN dapat mengeksplorasi tipe data dengan bentuk numerik maupun kategorik. Algoritma *random forest* dan KNN dapat dikatakan *apple to apple* karena kedua algoritma tersebut sering digunakan pada kasus klasifikasi yang dapat digunakan untuk perbandingan [15]. Oleh karena itu, dalam penelitian ini akan melakukan komparasi metode untuk mengetahui metode mana yang terbaik dalam melakukan klasifikasi kelancaran pembayaran kredit bank. Hasil penelitian diharapakan dapat dijadikan bahan pertimbangan pihak bank dalam upaya pemilihan nasabah kredit bank.

1.1 CRISP-DM

Metode CRISP-DM (*Cross-Industry Standard Process Model for Data mining*) merupakan metode atau kerangka kerja yang banyak diterapkan dalam *data mining*. Terdapat 6 (enam) tahap yang dapat di lakukan berikut.

1. *Business Understanding*

   Business Understanding adalah proses menentukan tujuan bisnis, memahami situasi dan kondisi pada saat penelitian dan menetapkan

sebuah tujuan dari penelitian yang dilakukan ke dalam permasalahan yang diselesaikan dengan *data mining*.

2. Data *Understanding*

   Data *understanding* adalah tahap persiapan, melakukan pengecekan terhadap data yang digunakan, mengumpulkan data awal serta melakukan identifikasi pada kualitas data. Dalam data understanding, data yang digunakan akan melalui proses deskripsi dari setiap fiturnya.

3. Data *Preparation*

   Data *preparation* merupakan proses yang dilakukan setelah data telah dikumpulkan. Pada tahap ini, data akan melalui proses identifikasi, pemilihan data, pembersihan data dan transformasi data.

4. *Modelling*

   Modeling merupakan tahap implementasi algoritma yang akan digunakan untuk melakukan pencarian, identifikasi, serta menghasilkan pola yang akan digunakan pada data penelitian.

5. *Evaluation*

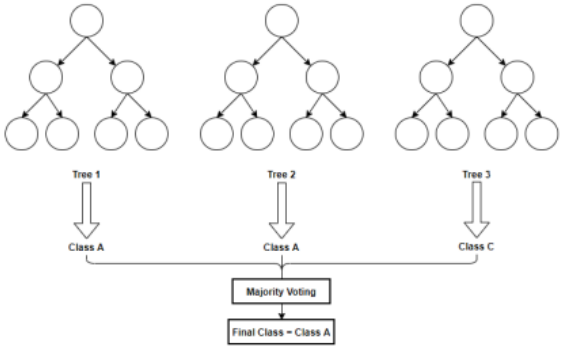
   Evaluation adalah suatu proses untuk melakukan pengukuran hasil evaluasi dari model yang telah diimplementasikan sebelumnya di tahap *modelling*. Hasil evaluasi tersebut mengGambarkan proses dari *data mining* yang telah dilakukan dan mengukur model yang paling baik untuk digunakan.

6. *Deployment*

   Deployment atau penyebaran merupakan proses menggunakan model yang dihasilkan sebelumnya. Dalam tahap ini terdapat 2 jenis kegiatan penyebaran yaitu melakukan perencanaan dan pemantauan dari penyebaran hasil yang dilakukan. Jenis kegiatan selanjutnya menyelesaikan tugas penutup dengan membuat laporan akhir dan melakukan tinjauan proyek. Kedua kegiatan ini dapat dilakukan semua atau menyelesaikan salah satunya.

1.2 *Random Forest*

Random Forest merupakan salah satu metode CART (*Classification and Regression Tree*) dalam *data mining* dan tidak memerlukan asumsi apapun. Metode ini menggunakan konsep pohon keputusan (*decision tree*). Model ini dibentuk dari banyak § pohon seperti hutan (*forest*) dengan menerapkan metode *bootstrap aggregating* (*bagging*) dan *random feature selection* [16]. Ilustrasi dari pohon keputusan dan pengambilan keputusan dengan *random forest* terdapat pada Gambar 1.



Gambar 1. Ilustrasi *Random Forest*

1.3 K-*Nearest Neighbor*

*K-Nearest Neighbo*r (KNN) adalah sebuah metode yang digunakan sebagai klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Metode KNN terbagi menjadi dua fase, yaitu fase pembelajaran (*training*) dan fase klasifikasi atau pengujian (*testing*). Algoritma KNN cukup mudah untuk diimplementasikan karena bekerja menurut jarak terdekat dari *query instance* ke sample latih untuk menentukan tetangga terdekatnya. Jarak dekat atau jauhnya tetangga dihitung dengan menggunakan jarak *Euclidean Distance* yang dapat dilihat pada persamaan 1 [3].

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (1)$$

Berikut merupakan langkah – langkah untuk menghitung algoritma KNN [3]:

1. Menentukan nilai parameter k.
2. Menghitung jarak antara data training dan data testing dengan menggunakan perhitungan jarak *Euclidean Distance*.
3. Mengurutkan jarak euclid yang terbentuk dari yang paling kecil.
4. Menentukan jarak terdekat sampai urutan ke k.
5. Mengumpulkan label class Y (klasifikasi *Nearest Neighbor*).
6. Menghitung jumlah kelas dari tetangga terdekat yang terbanyak dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi.

1.4 *Data mining*

Data mining merupakan penggabungan dari beberapa disiplin ilmu yang dimanfaatkan untuk menggali informasi dari sekumpulan data. Informasi yang diperoleh selanjutnya dapat dimanfaatkan dalam proses penarikan kesimpulan, manajemen informasi, pengendalian proses dan sebagainya [17].

1.5 *K-Fold Cross Validation*

K-fold cross validation atau disebut estimasi rotasi adalah sebuah teknik validasi model untuk menilai bagaimana hasil statistik analisis akan menggeneralisasi kumpulan data independen. Teknik ini utamanya digunakan untuk melakukan prediksi model dan memperkirakan seberapa akurat sebuah model prediktif ketika dijalankan dalam praktiknya. Salah satu teknik dari validasi silang adalah k-*fold cross validation*, yang mana memecah data menjadi K bagian set data dengan ukuran yang sama. Penggunaan k-fold=5 *cross-validation* untuk menghilangkan bias pada data. Pelatihan dan pengujian dilakukan sebanyak K [18].

1.6 *Google Colaboratory*

Secara lebih teknis, *Colab* merupakan layanan *notebook Jupyter* yang dihosting dan dapat digunakan tanpa penyiapan, serta menyediakan akses gratis ke resource komputasi termasuk GPU. *Resource Colab* tidak dijamin dan sifatnya terbatas, serta batas penggunaannya terkadang berfluktuasi. Hal ini diperlukan agar *Colab* dapat menyediakan *resource* secara gratis. Tujuan jangka panjang pihak *Google* adalah untuk terus menyediakan versi gratis *Colab*, dan di saat yang bersamaan berkembang secara berkelanjutan untuk memenuhi kebutuhan pengguna *Google* [19].

1.7 *Python*

Python merupakan bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* dikenal sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif [20].

1.8 *Confusion Matrix*

Confusion matrix merupakan perhitungan yang digunakan untuk mempresentasikan tingkat akurasi dari suatu klasifikasi. Dasar perhitungan confusion matrix dengan membandingkan total data yang diklasifikasikan pada kelas yang bener dengan total seluruh data yang ada pada matrix [21]. Berikut *confusion matrix*, dapat dilihat pada tabel 1.

Tabel 1. *Confusion Matrix*

		Prediction Class	
		Positive	*Negative*
Actual Class	*Positive*	*True Positive (TP)*	*False Negative (FN)*
	Negative	*False Positive (FP)*	*True Negative (TN)*

Dimana pada tabel 1 dijelaskan bahwa:

TN : Model memprediksi data ada di kelas negatif dan yang sebenarnya data memang ada di kelas negatif

TP : Model memprediksi data ada di kelas positif dan yang sebenarnya data memang ada di kelas positif

FN : Model memprediksi data ada di kelas negatif dan yang sebenarnya data memang ada di kelas positif

FP : Model memprediksi data ada di kelas positif dan yang sebenarnya data memang ada di kelas negatif.

Accuracy atau biasa disebut akurasi mengukur jumlah total prediksi yang benar dibandingkan dengan total data. Rumus untuk menghitung *accuracy* ditunjukkan pada persamaan 2.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \times 100 \quad (2)$$

Precision adalah perbandingan antara jumla prediksi data kelas positif yang merupakan kelas positif dengan banyak data yang diprediksi positif. Sedangkan *recall* adalah perbandingan antara jumlah prediksi data kelas positif yang merupakan kelas positif dengan banyaknya data yang sebenarnya positif [21]. Adapun rumus yang digunakan untuk menghitung *precision* dapat menggunakan persamaan 3, sedangkan untuk menghitung *recall* dapat menggunakan persamaan 4.

$$precision = \frac{\text{TP}}{\text{TP+FP}} \quad (3)$$

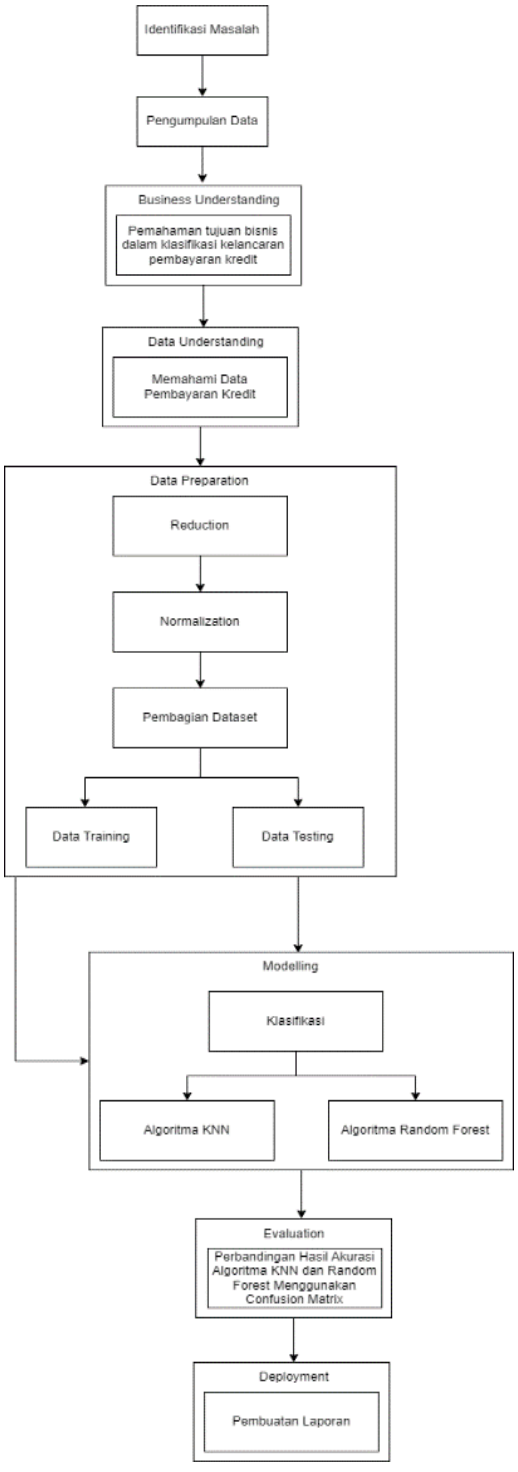
$$recall = \frac{\text{TP}}{\text{TP+FN}} \quad (4)$$

Dan untuk persamaan ke 4 *F1-Score* merupakan *harmonic mean* dari *precision* dan *recall*. Nilai terbaik *F1-Score* adalah 1.0 sedangkan nilai terburuknya adalah 0 [21]. Adapun rumus yang digunakan untuk menghitung *F1-Score* dapat dilihat pada persamaan ke 5 berikut ini.

$$F1 - Score = \frac{2 \times \text{Precision x Recall}}{\text{Precision+Recall}} \quad (5)$$

2. METODE PENELITIAN

Adapun alur dari penelitia yang akan dilakukan dapat dilihat pada Gambar 2.



Gambar 2 Alur Penelitian

Dari Gambar 2, alur penelitian dapat dijelaskan bahwa terdapat beberapa tahapan yang dilakukan dalam penelitian ini yaitu:

a) Identifikasi Masalah
Tahap identifikasi masalah merupakan tahap mengidentifikasi permasalahan yang terjadi, dalam penelitian ini permasalahan terkait metode KNN yang memperoleh akurasi yang cenderung rendah dibandingkan metode klasifikasi yang lainnya. Oleh karena itu, dalam penelitian ini akan melakukan percobaan menggunakan algoritma KNN dan *Random Forest* dalam mengklasifikasi kelancaran pembayaran kredit bank untuk mengetahui performa dari *Random Forest* dan KNN dengan dataset yang digunakan dari UCI *Machine Learning Repository*.

b) Pengumpulan Data
Pengumpulan data merupakan proses mengumpulkan data yang digunakan dalam penelitian dengan melakukan studi literatur untuk mendapatkan teori, penelitian terdauhulu sebagai bahan referensi dalam penelitian ini. Kemudian, pengumpulan dataset yang diambil dari UCI *Machine Learning Repository* terkait data pembayaran kredit dengan rentang waktu tertentu.

c) *Business Understanding*
Tahap selanjutnya berdasarkan metode proyek *data mining* yaitu CRISP-DM. Tahapan dari metode CRISP-DM yang pertama yaitu business understanding merupakan proses pemahaman tujuan bisnis, dalam penelitian ini bertujuan untuk melakukan komparasi akurasi dari algoritma *Random Forest* dan KNN dalam mengklasifikasi kelancaran pembayaran kredit bank.

d) Data *Understanding*
Data *understanding*, proses pemahaman data yang digunakan dengan menganalisis serta mengevaluasi kualitas data yang digunakan.

e) Data *Preparation*
Data *preparation*, proses persiapan data yang dilakukan dalam penelitian ini terdiri dari data reduction menggunakan *Principal Component Analysis* (PCA) untuk pemilihan *feature*, kemudian normalisasi yaitu proses penyeragaman nilai yang memiliki range 0 dan 1 jika menggunakan *min max normalization*. Dilanjutkan dengan pembagian dataset yaitu membagi data menjadi data training dan data testing.

f) *Modelling*
Tahap *modelling* yaitu pembentukan model klasifikasi menggunakan metode *Random Forest* dan KNN untuk melakukan perbandingan. Dalam penelitian ini diimplementasikan kedalam bahasa pemrograman *Python* dengan tools *Google Colab*.

g) *Evaluation*
Pada tahap *evaluation* dilakukan evaluasi metode yang digunakan menggunakan confusion matrix dan model K-*Fold Cross Validation*. *Confusion matrix* akan menghasilkan nilai *accuracy*, *precision*, *recall* dan *f1-score* dari algoritma *Random Forest* dan

KNN, sedangkan model K-*Fold Cross Validation* menghasilkan memperkirakan seberapa akurat sebuah model prediktif ketika dijalankan dalam praktiknya.

h) *Deployment*
Tahap terkahir yaitu tahap deployment, melakukan penyebaran hasil penelitian dengan membuat laporan hasil penelitian yang berisi hasil komparasi akurasi kedua metode tersebut dalam bentuk ilmu pengetahuan

3. HASIL DAN PEMBAHASAN

3.1 *Business Understanding*

Pada tahap ini merupakan tahap pemahaman bisnis, penjelasan terhadap permasalahan yang terjadi sehingga dapat menentukan tujaun penelitian. Pemahaman masalah dalam penelitian ini yaitu kelancaran pembayaran kredit bank. Dalam pemberian kredit kepada nasabah yang tidak tepat dapat menimbulkan permasalahan seperti nasabah yang tidak membayar angsuran tepat waktu bahkan menunda sampai beberapa bulan pembayaran angsuran hingga terjadinya kredit macet sehingga hal ini dapat merugikan pihak bank. Oleh karena itu, tujuan penelitian yaitu menerapkan *data mining* untuk melakukan klasifikasi kelancaran pembayaran kredit bank menggunakan algoritma *Random Forest* dan K-*Nearest Neighbor* (KNN) sehingga dapat diketahui performa tingkat akurasi dari algoritma tersebut.

3.2 *Data Understanding*

Data *understanding* merupakan tahap untuk memahami data yang digunakan. Dalam penelitian ini menggunakan dataset yang dilakukan dari UCI *Machine Learning Repository* merupakan data pembayaran kredit dengan rentang waktu tertentu. *Dataset* tersebut memiliki 29.998 data dan terdiri dari 2 *class* dengan nama *class* yaitu "Y" serta terdiri dari *class* "Dibayarkan" memiliki 6.636 data, dan pada *class* "Tidak Dibayarkan" memiliki 23.364 data. Berikut pada Tabel 2 merupakan metadata yang ada pada *dataset* yang digunakan.

Tabel 2. *Metadata* Dataset

	Feature	Tipe Data	Keterangan
X 1	LIMIT_BAL	Numerik	Total Kredit yang diberikan
X 2	SEX	Kategorik	Jenis Kelamin Customer
X 3	EDUCATION	Kategorik	Jenjang Pendidikan
X 4	MARRIAGE	Kategorik	Status
X 5	AGE	Numerik	Umur (Tahun)
X 6	PAY_0	Kategorik	Riwayat Pembayaran Semptember 2005
X 7	PAY_2	Kategorik	Riwayat Pembayaran Agustus 2005
X 8	PAY_3	Kategorik	Riwayat Pembayaran Juli 2005
X 9	PAY_4	Kategorik	Riwayat Pembayaran Juni 2005
X 10	PAY_5	Kategorik	Riwayat Pembayaran Mei 2005
X 11	PAY_6	Kategorik	Riwayat Pembayaran April 2005
X 12	BILL_AMT1	Numerik	Jumlah Pembayaran Semptember 2005
X 13	BILL_AMT2	Numerik	Jumlah Pembayaran Agustus 2005
X 14	BILL_AMT3	Numerik	Jumlah Pembayaran Juli 2005
X 15	BILL_AMT4	Numerik	Jumlah Pembayaran Juni 2005
X 16	BILL_AMT5	Numerik	Jumlah Pembayaran Mei 2005
X 17	BILL_AMT6	Numerik	Jumlah Pembayaran April 2005
X 18	PAY_AMT1	Numerik	Jumlah yang sudah dibayar Semptember 2005
X 19	PAY_AMT2	Numerik	Jumlah yang sudah dibayar Agustus 2005
X 20	PAY_AMT3	Numerik	Jumlah yang sudah dibayar Juli 2005
X 21	PAY_AMT4	Numerik	Jumlah yang sudah dibayar Juni 2005
X 22	PAY_AMT5	Numerik	Jumlah yang sudah dibayar Mei 2005
X 23	PAY_AMT6	Numerik	Jumlah yang sudah dibayar April 2005
Y	default payment next month	Kategorik	Keterangan Dibayar atau tidak

Dataset pembayaran kredit bank yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.

Tabel 3. Dataset Pembayaran Kredit Bank

	X1	X2	⋯	X23	Y
ID	LIMIT_BAL	SEX	...	PAY_ AMT6	default payment next month
1	20000	2	⋯	0	1
2	120000	2	⋯	2000	1
3	90000	2	⋯	5000	0
4	50000	2	⋯	1000	0
5	50000	1	⋯	679	0
…	…	…	…	…	…
29996	220000	1	⋯	1000	0
29997	150000	1	⋯	0	0
29998	30000	1	⋯	3100	1
29999	80000	1	⋯	1804	1
30000	50000	1	⋯	1000	1

3.3 Data *Preparation*

Data *preparation* merupakan proses melakukan persiapan data dengan menyesuaikan dataset agar dapat sesuai dengan kebutuhan pada saat tahap pemodelan.

3.3.1 *Feature Selection*

Dalam implementasi kedalam bahasa pemrograman *Python* terdapat *library* yang digunakan yaitu numpy, seaborn, matplotlib, PCA, Min Max Scaler, KNeighborsClassifier, RandomForestClassifier, accuracy_score, confusion_matrix, precision_score, recall_score dan f1_score.

Hasil *feature selection* menggunakan PCA dapat dilihat pada Gambar 3. dari hasil tersebut didapatkan informasi jumlah *feature* sebanyak 23 dan hasil *reduction* sebanyak 20 *feature*. Hasil data *reduction* ditampilkan dalam bentuk tabel yang berisi data dari setiap *feature*, *feature* tersebut yaitu MARRIAGE, AGE, PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6, BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_AMT6, PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5 dan PAY_AMT6.

```
Original number of features: 23
Reduced number of features : 20
```

	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
0	-0.980259	0.521612	-0.309457	0.125423	3.632171	1.135527	-9.902204	-126.809629	832.973648	450.097645	714.585844	-189.499608	79.013084	-861.737802	22.858177	-356.372601	138.240596	472.952106	75538.153664	166488.191086
1	-0.227876	2.467209	0.386503	0.306074	-0.845945	-1.917522	-8.956800	453.880852	-6.875099	1052.860787	483.598389	626.282074	1261.720752	-1516.058527	-696.602606	4050.710334	1567.195890	-2734.447477	-9780.671251	114226.976816
2	0.046229	0.030648	0.004681	0.050489	-0.262034	-0.252556	-0.668003	398.884282	-2911.542027	-3126.191011	4599.925596	2283.836424	1612.055662	-6986.944318	-177.028488	-3361.177076	5102.262357	2678.469790	33471.387331	98432.362180
3	0.014328	0.053599	-0.026412	0.008502	-0.325648	0.370170	2.599444	-1404.697693	-3644.742566	5514.330363	-1815.577038	-547.526631	837.225566	2367.908856	-6192.886363	637.932441	-4420.222964	10047.716972	95224.273104	71230.675903
4	-0.771190	0.462268	0.191095	-0.359462	-0.916790	0.631038	22.798985	2647.059824	224.382714	4062.262493	-625.666837	-7678.722530	2722.672426	-11838.007527	-4668.730523	-8197.837873	-34257.290425	-24604.948086	68729.185233	114834.618365
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29995	-0.001521	0.229562	-0.529002	0.866364	-0.680842	-0.562591	1.948856	-849.217492	-13448.294881	15958.406940	-35405.948109	-10491.810171	-6323.562456	5659.860060	-10261.573899	15152.671330	-76676.272865	141623.472655	55465.636023	-200018.431424
29996	-0.022674	0.489383	-0.533281	0.871481	-1.108602	0.614858	7.732285	-1606.078294	-182.921024	1802.454943	-1204.739251	-3650.254915	-1922.187948	-3220.507763	6764.647370	5995.152582	-1717.464417	-4753.925193	-35089.032854	97718.568024
29997	0.716095	0.866649	-0.568850	-0.431336	4.102088	-2.873917	3.007409	-1299.435036	-1985.427891	6139.941402	-3685.951075	-1273.800749	-4584.941924	-5604.399788	16149.639767	-5723.740233	7149.554021	-26243.133381	77857.496539	140206.128048
29998	-0.052529	-0.923155	-1.083915	0.775161	0.741621	0.231198	6.668380	-1544.915893	-10716.733178	-26006.824518	-16251.952760	10867.649618	-49383.714787	92784.294514	-1029.747369	-8431.917084	-44989.240431	-17722.587980	74908.303672	41427.740774
29999	-0.042558	0.065927	0.022157	-0.228475	-0.333326	0.271701	11.622813	-6179.929215	571.030049	-474.772680	-5973.828992	-1781.452516	4285.984246	-252.990411	3648.483547	6804.331031	-8688.849832	14117.466645	95514.144309	70860.311209

30000 rows × 20 columns

Gambar 3. Hasil *Feature Selection* Menggunakan PCA

3.3.2 Normalisasi Data

Normalisasi data merupakan proses penyeragaman nilai yang memiliki *range* 0 dan 1 jika menggunakan *min max normalization. Source code* proses *min max normalization*. Hasil normalisasi yaitu terdapat 30000 baris dan 20 kolom.

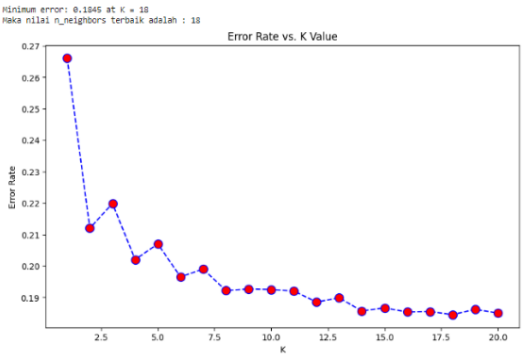
3.3.3 *Split Data*

Pada tahap ini dilakukan *split* data yaitu proses membagi data menjadi 2 kelompok yaitu data *train* dan data *test.* Data *training* digunakan untuk pelatihan model dan data *test* digunakan untuk pengujian model. Pembagian dataset menggunakan perbandingan 80% data *train* dan 20% data *test* dengan jumlah masing-masing data yaitu data *train* 24000 dan data *test* 6000.

3.4 *Modelling*

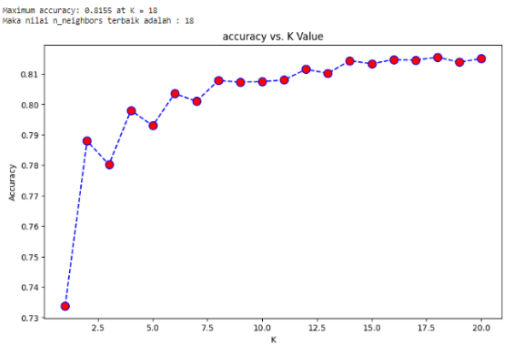
3.4.1 Algoritma KNN

Algoritma *K-Nearest Neighbor* merupakan algoritma yang dilakukan untuk klasifikasi kelancaran pembayaran kredit bank menggunakan dataset yang telah dilakukan *preparation*. Dalam algoritma KNN memilih nilai "K" adalah langkah awal untuk melatih model. Dari Gambar 4, nilai K=18 menunjukkan tingkat kesalahan yang relatif lebih rendah dengan nilai *error rate* yaitu 0.1845. Jadi, untuk model ini menggunakan K=18, Selanjutnya masukkan jalur dataset pelatihan dalam model KNN untuk melatih model dan menghitung jarak dengan sendirinya.



Gambar 4. Hasil *Min imum Error Rate* dan Nilai K Pada KNN

Dari Gambar 5, nilai K=18 menunjukkan tingkat kesalahan yang relatif lebih tinggi dengan nilai *error rate* yaitu 0.8155. Jadi, untuk model ini menggunakan K=18, Selanjutnya masukkan jalur dataset pelatihan dalam model KNN untuk melatih model dan menghitung jarak dengan sendirinya.

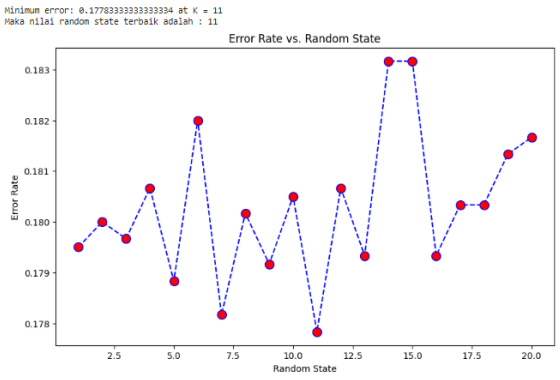


Gambar 5. Hasil *Maximum Error Rate* dan Nilai K Pada KNN

3.4.2 Algoritma *Random Forest*

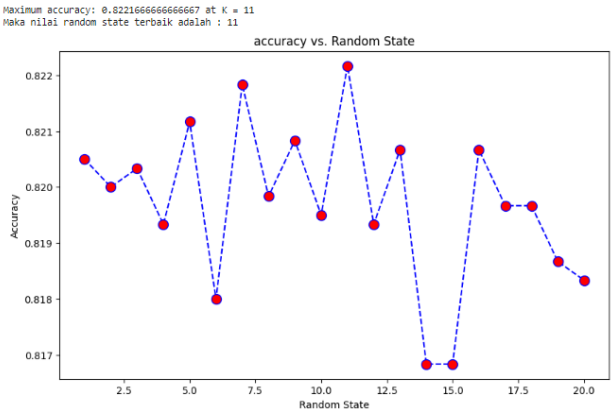
Algoritma *Random Forest* merupakan algoritma yang dilakukan untuk klasifikasi kelancaran

pembayaran kredit bank menggunakan dataset yang telah dilakukan *preparation*.. Dalam algoritma *Random Forest* memilih nilai *random state* adalah langkah awal untuk melatih model. Dari Gambar 6, nilai *random state*=11 menunjukkan tingkat kesalahan yang relatif lebih rendah dengan nilai *error rate* yaitu 0.17783333333333334. Jadi, untuk model ini menggunakan *random state* =11, Selanjutnya masukkan jalur dataset pelatihan dalam model *Random Forest* untuk melatih model dan menghitung jarak dengan sendirinya.



Gambar 6. Hasil *Minimum Error Rate* dan Nilai *Random State* Pada *Random Forest*

Selanjutnya, menentukan nilai *error rate maximum* pada model *Random Forest.* Dari Gambar 7, nilai *random state* = 11 menunjukkan tingkat kesalahan yang relatif lebih tinggi dengan nilai *error rate* yaitu 0.8221666666666667. Jadi, untuk model ini menggunakan *random state* = 11, Selanjutnya masukkan jalur dataset pelatihan dalam model *Random Forest* untuk melatih model dan menghitung jarak dengan sendirinya.

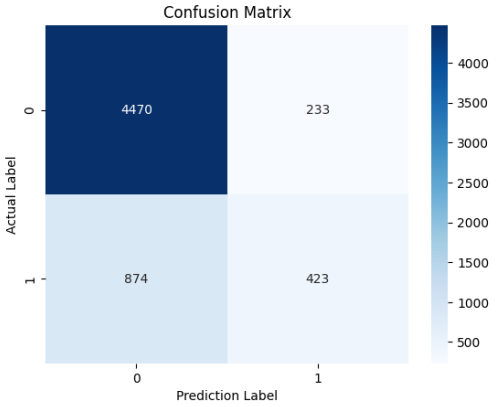


Gambar 7. Hasil *Maximum Error Rate* dan Nilai *Random State* Pada *Random Forest*

3.5 *Evaluation*

3.5.1 Evaluasi Algoritma KNN

Pada tahap ini dilakukan evaluasi algoritma KNN menggunakan *confusion matrix* untuk mengetahui performa tingkat akurasi dari model. Sebelum melakukan evaluasi, pada tahap ini melakukan prediksi menggunakan algoritma KNN dengan data *test* yang telah disiapkan. Kemudian, dilanjutkan proses evaluasi menggunakan *confusion matrix* yang menghasilkan nilai *accuracy*, *precision*, *recall* dan f1_*score*. Hasil prediksi data menggunakan algoritma KNN Dalam penelitian pada algoritma KNN mendapatkan nilai *accuracy* sebesar 81.55 %, *precision* sebesar 79.5 %, *recall* sebesar 81.55 % dan f1-*score* sebesar 79.11 %. Dari nilai yang diperoleh *confusion matrix* untuk algoritma KNN dilakukan visualisasi dalam bentuk grafik dari grafik tersebut akan ditampilkan nilai prediksi dan aktual pada label Layak dan Tidak Layak. Hasil visualisasi *confusion matrix* pada algoritma KNN ditampilkan pada Gambar 8. Dari grafik tersebut didapatkan informasi *True Positive* (TP), *False Positive* (FP), *False Negative* (FN) dan *True Negative* (TN).



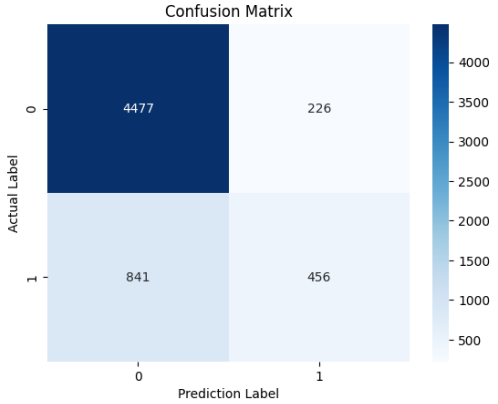
Gambar 8. Hasil Visualisasi Metode KNN

Berdasarkan Gambar 8 dapat dijelaskan bahwa algoritma KNN dengan diperoleh hasil prediksi pada kelas Layak dengan jumlah data 4703 data dengan label Layak yang diuji, terdapat 4470 data yang terklasifikasi dengan benar dan kesalahan prediksi sebesar 233 data yang masuk kedalam kelas Tidak Layak. Sedangkan, pada kelas Tidak Layak yang diuji dengan total 1297 dengan label Tidak Layak yang diuji, terdapat 423 data sudah terklasifikasi dengan benar dan kesalahan prediksi sebesar 874 data yang masuk kedalam kelas Layak. Dari nilai *confusion matrix* tersebut didapatkan nilai *accuracy* sebesar 81.55 %, *precision* sebesar 79.5 %, *recall* sebesar 81.55 % dan f1-*score* sebesar 79.11 %.

3.5.2 Evaluasi Algoritma *Random Forest*

Pada tahap ini dilakukan evaluasi algoritma *Random Forest* menggunakan *confusion matrix* untuk mengetahui performa tingkat akurasi dari model. Sebelum melakukan evaluasi, pada tahap ini melakukan prediksi menggunakan algoritma KNN dengan data *test* yang telah disiapkan. Kemudian, dilanjutkan proses evaluasi menggunakan *confusion matrix* yang menghasilkan nilai *accuracy*, *precision*, *recall* dan f1_*score*. Dalam penelitian pada

algoritma *Random Forest* mendapatkan nilai *accuracy* sebesar 82.22 %, *precision* sebesar 80.44 %, *recall* sebesar 82.22 % dan f1-*score* sebesar 80.0 %. Dari nilai yang diperoleh *confusion matrix* untuk algoritma *Random Forest* dilakukan visualisasi dalam bentuk grafik. Dari grafik tersebut akan ditampilkan nilai prediksi dan aktual pada label Layak dan Tidak Layak. Hasil visualisasi *confusion matrix* pada algoritma *Random Forest* ditampilkan pada Gambar 9. Dari grafik tersebut didapatkan informasi *True Positive* (TP), *False Positive* (FP), *False Negative* (FN) dan *True Negative* (TN).



Gambar 9 Hasil Visualisasi Metode *Random Forest*

Berdasarkan Gambar 9 dapat dijelaskan bahwa algoritma *Random Forest* dengan diperoleh hasil prediksi pada kelas Layak dengan jumlah data 4703 data dengan label Layak yang diuji, terdapat 4477 data yang terklasifikasi dengan benar dan kesalahan prediksi sebesar 226 data yang masuk kedalam kelas Tidak Layak. Sedangkan, pada kelas Tidak Layak yang diuji dengan total 1297 dengan label Tidak Layak yang diuji, terdapat 456 data sudah terklasifikasi dengan benar dan kesalahan prediksi sebesar 841 data yang masuk kedalam kelas Layak. Dari nilai *confusion matrix* tersebut didapatkan nilai *accuracy* sebesar 82.22 %, *precision* sebesar 80.44 %, *recall* sebesar 82.22 % dan f1-*score* sebesar 80.0 %.

Dari hasil evaluasi menggunakan *confusion matrix* mendapatkan nilai akurasi terbaik pada algoritma *random forest* yaitu 82.22 %, *precision* sebesar 80.44 %, *recall* sebesar 82.22 % dan f1-*score* sebesar 80.0 %. Sedangkan, pada algoritma KNN mendapatkan nilai *accuracy* sebesar 81.55 %, *precision* sebesar 79.5 %, *recall* sebesar 81.55 % dan f1-*score* sebesar 79.11 %. Berdasarkan hasil evaluasi tersebut bahwa algoritma *Random Forest* mendapatkan akurasi terbaik dibandingkan dengan algoritma KNN dalam mengklasifikasi kelancaran pembayaran kredit bank dikarenakan pendekatan model *random forest* termasuk *ensemble* model yang menggabungkan banyak pohon keputusan secara *parallel*, sedangkan KNN algoritma berbasis *instance* yang menggunakan jarak untuk klasifikasi data. Selain itu dari segi skalabilitas *random forest* dapat menangani dataset yang lebih besar dengan lebih efisien dibandingkan KNN.

Tabel 4. Komparasi Tingkat Akurasi Algoritma KNN dan Random Forest

Confusion Matrix	**Algoritma**	
	KNN	***Random Forest***
Accuracy	81.55 %	82.22 %
Precision	79.5 %	80.44 %
Recall	81.55 %	82.22 %
F1-Score	79.11 %	80.0 %

Selain menggunakan evaluasi menggunakan *confusion matrix*, algoritma KNN dan *Random Forest* juga melakukan evaluasi *K-fold cross validation*. Dimana *K-fold cross* validation ini digunakan untuk menguji kinerja model *Random Forest* dengan membagi data menjadi k subset (*fold*) yang lebih kecil. Adapun hasil yang diperoleh dari evaluasi hasil *k-fold cross validation* algoritma KNN dapat dilihat pada Tabel 5.

Tabel 5. Komparasi Tingkat Akurasi Algoritma KNN dan Random Forest

n_fold	***Accuracy***
2	0.77550
3	0.77670
4	0.77850
5	0.77767
6	**0.77910**
7	0.77797
8	0.77867
9	0.77847
10	0.77790

Sedangkan untuk hasil yang diperoleh dari evaluasi hasil *k-fold cross validation* algoritma *Random Forest* dapat dilihat pada Tabel 6.

Tabel 6. Komparasi Tingkat Akurasi Algoritma KNN dan Random Forest

n_fold	***Accuracy***
2	0.81443
3	0.81527
4	0.81567
5	0.81453
6	0.81567
7	0.81637
8	0.81597
9	**0.81730**
10	0.81587

Berdasrkan tabel 5 dan tabel 6 dapat disimpulkan bahwabahwa nilai akurasi terbaik *K-Nearestt Neighbor Classifier* diperoleh pada n_fold = 6 sebesar 77.91%. sedangkan untuk nilai akurasi terbaik *Random Forest* diperoleh pada n_fold = 9 sebesar 81.73%.

4. DISKUSI

Tujuan penelitian ini adalah membandingkan dua jenis algoritma yaitu algoritma KNN dan

Algroitma *Random Forest*. Pada penelitian sebelumnya yang di lakukan oleh [22] melakukan prediksi terhadap kelancaran pembayaran kredit dengan mengkombinasikan metode Naïve Bayes dan K-Nearest Neighbor. Prediksi tingkat kelancaran pembayaran kredit ini dilakukan dengan mengkombinasikan algoritma *Naïve Bayes* dan *K-Nearest Neighbo*r agar dapat memprediksi kelancaran pembayaran kredit kedepannya, hal ini dapat dilihat dari hasil prediksi yang diperoleh sebesar 80%. Namun kasus pada penelitian kali ini berbeda dengan penelitian sebelumnya dimana komparasi algoritma KNN dan algoritma random forest akan di lakukan klasifikasi untuk mengetahui metode mana yang terbaik dalam melakukan klasifikasi kelancaran pembayaran kredit bank. Hasil penelitian diharapkan dapat dijadikan bahan pertimbangan pihak bank dalam upaya pemilihan nasabah kredit bank

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dalam melakukan komparasi tingkat akurasi algoritma KNN dan *Random Forest* untuk mengklasifikasi kelancaran pembayaran kredit bank didapatkan kesimpulan sebagai berikut:

1. Tingkat akurasi algoritma *Random Forest* dalam mengklasifikasikan kelancaran nilai *accuracy* 82.22 %, *precision* sebesar 80.44 %, *recall* sebesar 82.22 % dan f1-*score* sebesar 80.0 %. Sedangkan evaluasi menggunakan *K-fold cross validation* memperoleh nilai akurasi sebesar 81.73% pada nilai n_folds = 9
2. Tingkat akurasi algoritma KNN dalam mengklasifikasikan kelancaran pembayaran kredit bank yaitu mendapatkan nilai *accuracy* sebesar 81.55 %, *precision* sebesar 79.5 %, *recall* sebesar 81.55 % dan f1-*score* sebesar 79.11 %. Dari hasil evaluasi menggunakan *k-fold cross validation*, akurasi terbaik *K-Nearestt Neighbor Classifier* diperoleh pada n_fold = 6 sebesar 77.91%. sedangkan untuk nilai akurasi terbaik *Random Forest* diperoleh pada n_fold = 9 sebesar 81.73%. Sehingga berdasarkan hasil evaluasi tersebut dapat disimpulkan bahwa algoritma *Random Forest* mendapatkan akurasi terbaik dibandingkan dengan algoritma KNN dalam mengklasifikasi kelancaran pembayaran kredit bank.
3. Hasil analisis menunjukkan bahwa algoritma *random forest* memiliki nilai akurasi yang lebih tinggi dibandingkan dengan KNN dikarenakan pendekatan model *random forest* termasuk *ensemble* model yang menggabungkan banyak pohon keputusan secara *parallel*, sedangkan KNN algoritma berbasis *instance* yang menggunakan jarak untuk klasifikasi data. Selain itu dari segi skalabilitas *random forest* dapat menangani dataset yang lebih besar dengan lebih efisien dibandingkan KNN.

DAFTAR PUSTAKA

[1] P. Subarkah, E. P. Pambudi dan S. O. N. Hidayah, "Perbandingan Metode Klasifikasi Data Mining untuk Nasabah Bank Telemarketing," *Matrik: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, pp. 139-148, 2020.

[2] Farida Gultom dan T. S. , "Prediksi Tingkat Kelancaran Pembayaran Kredit Bank Dengan Menggunakan Algoritma Naïve Bayes Dan K-Nearest Neighbor," *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, vol. 4, pp. 1-5, 2020.

[3] D. Hartono, L. W. Santoso dan S. Rostianingsih, "Sistem Pendukung Keputusan Pemberian Kredit berdasarkan Klasifikasi Kelancaran Pembayaran Kredit," *JURNAL INFRA*, vol. 10, no. 2, pp. 226-232, 2022.

[4] E. Ndruru dan T. Zebua, "Penerapan Algoritma Nearest Neighbor Dalam Memprediksi Kelayakan Penerimaan Kartu Kredit Pada Bank Cimb Niaga," *Jurnal METHODIKA*, vol. 5, no. 1, pp. 1-6, 2019.

[5] W. Hidayat dan A. M. Utami, "Penerapan Metode Algoritma C4.5 Untuk Menentukan Kelayakan Calon Nasabah Pemegang Kartu Kredit Bank Mega Card center Kuningan," *TeknoIS : Jurnal Ilmiah Teknologi Informasi dan Sains*, vol. 12, no. 1, pp. 31-48, 2022.

[6] E. Wijaya, F. A. Tarigan dan Michael, "Aplikasi Prediksi Penentuan Kelancaran Pembayaran Koperasi Dengan Algoritma C5.0," *Jurnal TIMES*, vol. 10, no. 1, pp. 31-38, 2021.

[7] M. Maulidah, W. Gata, R. Aulianita dan C. I. Agustyaningrum, "Algoritma Klasifikasi Decision Tree Untuk Rekomendasi Buku Berdasarkan Kategori Buku," *JURNAL ILMIAH EKONOMI DAN BISNIS*, pp. 89 - 96, 2020.

[8] Marsono, A. H. Nasyuha, S. N. Arif, M. Zunaidi dan N. Y. L. Gaol, "Implementasi Algoritma K-Nearest Neighbor Dalam Mendiagnosis Kurap Pada Kucing," *Journal of Computer System and Informatics (JoSYC)*, vol. 4, no. 1, pp. 61-65, 2022.

[9] A. K. B. Ginting, M. S. Lydia dan E. M. Zamzami, "Peningkatan Akurasi Metode K-Nearest Neighbor dengan Seleksi Fitur Symmetrical Uncertainty," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, pp. 1714-1719, 2021.

[10] U. Erdiansyah, A. I. Lubis dan K. Erwansyah, "Komparasi Metode K-Nearest Neighbor dan Random Forest Dalam Prediksi Akurasi Klasifikasi Pengobatan Penyakit Kutil," *JURNAL MEDIA INFORMATIKA BUDIDARMA,* vol. 6, pp. 208-214, 2022.

[11] I. Ubaedi dan Y. M. Djaksana, "Optimasi Algoritma C4.5 Menggunakan Metode Forward Selection Dan Stratified Sampling Untuk Prediksi Kelayakan Kredit," *JSiI | Jurnal Sistem Informasi,* pp. 17-26, 2022.

[12] E. Fauziah dan A. F. Zulfikar, "Penerapan Metode Decision TreeMenggunakan Algoritma Iterative Dichotomiser3 (ID3) Untuk Klasifikasi Resiko Penyakit Jantung," *OKTAL : Jurnal Ilmu Komputer dan Science,* vol. 2, no. 4, pp. 1207-1219, 2023.

[13] A. Ramadhan, B. Susetyo dan I. , "Penerapan Metode Klasifikasi Random Forest Dalam Mengidentifikasi Faktor Penting Penilaian Mutu Pendidikan," *Jurnal Pendidikan dan Kebudayaan,* pp. 169-182, 2019.

[14] C. A. Rahardja, T. Juardi dan H. Agung, "Implementasi Algoritma K-Nearest Neighbor Pada Website Rekomendasi Laptop," *Jurnal Buana Informatika,* pp. 75-84, 2019.

[15] V. Angkasa dan J. J. Pangaribuan, "Komparasi Tingkat Akurasi Random Forest Dan Knn Untuk Mendiagnosis Penyakit Kanker Payudara," *Information System Development,* vol. 7, no. 1, pp. 49-62, 2022.

[16] Q. Iman dan A. W. Wijayanto, "Klasifikasi Rumah Tangga Penerima Beras Miskin (Raskin)/Beras Sejahtera (Rastra) di Provinsi Jawa Barat Tahun 2017 dengan Metode Random Forest dan Support Vector Machine," *JUSTIN (Jurnal Sistem dan Teknologi Informasi),* vol. 9, pp. 178-184, 2021.

[17] D. Sartika dan I. Saluza, "Penerapan Metode Principal Component Analysis (PCA) Pada Klasifikasi Status Kredit Nasabah Bank Sumsel Babel Cabang KM 12 Palembang Menggunakan Metode Decision Tree," *GENERIC Jurnal Ilmu Komputer dan Teknologi Informasi,* vol. 14, pp. 45-49, 2022.

[18] H. Azis, Purnawansyah dan F. Fattah, "Performa Klasifikasi K-NN dan Cross-validation pada Data Pasien Pengidap Penyakit Jantung," *Jurnal Ilmiah,* vol. 12, no. 2, -ISSN 2548-7779, pp. .81-86, 2020.

[19] G. I. E. Soen, M. dan R. , "Implementasi Cloud Computing dengan Google Colaboratory Pada Aplikasi Pengolah Data Zoom Participants," *JITU : Journal Informatic Technology And Communication,* pp. 24-30, 2022.

[20] R. N. Melinda, L. M. Ningrum, I. B. Suryabrata, G. S. B. A. Dwipa dan T. P. Sukoco, "Program Perhitungan RAB Pekerjaan Struktur Baja (WF BEAM) Menggunakan Bahasa Python," *TIERS Information Technology Journal,* pp. 31-38, 2021.

[21] D. Sartika dan I. Saluza, "Penerapan Metode Principal Component Analysis (PCA) Pada Klasifikasi Status Kredit Nasabah Bank Sumsel Babel Cabang KM 12 Palembang Menggunakan Metode Decision Tree," *GENERIC Jurnal Ilmu Komputer dan Teknologi Informasi,* vol. 15, pp. 45-49, 2022.

[22] F. Gultom dan T. Simanjuntak, "Prediksi Tingkat Kelancaran Pembayaran Kredit Bank Dengan Menggunakan Algoritma Naïve Bayes Dan K-Nearest Neighbor," *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi,* vol. 4, pp. 1-45, 2020.