IT-EXPLORE
Jurnal Penerapan Teknologi Informasi dan Komunikasi
ISSN 2828-7940, e-ISSN 2829-1727, DOI: 10.24246/itexplore.v4i1.2025.pp82-92
Volume 04 Nomor 01 Tahun 2025
Journal homepage: https://ejournal.uksw.edu/itexplore

Analisis perbandingan *machine learning* untuk prediksi kelayakan kredit perbankan pada Bank BRI Tegal

Wresti Andriani[1], Gunawan Gunawan[2], Naella Nabila Putri Wahyuning Naja[3]
[1]Universitas Muhammadiyah Tegal, Tegal Timur, Tegal, Jawa Tengah
[2]Universitas Pancasakti Tegal, Tegal Timur, Tegal, Jawa Tengah
[3]Universitas Negeri Semarang, Gunung Pati, Semarang, Jawa Tengah

Informasi Artikel

Riwayat Artikel:

Dikirim:
22 Desember 2024
Direvisi:
20 Januari 2025
Diterima:
7 Februari 2025

Kata Kunci:

Klasifikasi,
Decission Tree,
Mitigasi Risiko Perbankan,
Prediksi Eligibilitas Kredit

Keywords:

Classification,
Decision Tree,
Banking Risk Mitigation,
Credit Eligibility Prediction

This is an open access article under the CC BY-SA license

ABSTRAK

Prediksi kelayakan kredit adalah langkah penting bagi perbankan untuk mengurangi risiko kredit macet. Penelitian ini membandingkan kinerja empat algoritma klasifikasi, yaitu Support Vector Machine, Naïve Bayes, Random Forest dan Decision Tree. Hasil diperoleh pada metrik akurasi, precision, recall, F1 score, dan AUC-ROC, menunjukkan bahwa Decision Tree memiliki performa terbaik dengan akurasi 42.5%, precision 48.3%, recall 47.5%, F1 score 47.5%, dan AUC 0.60, menandakan kemampuan moderat dalam membedakan kelayakan kredit. Random Forest mencapai akurasi 37.5% dan AUC 0.493, sedangkan Naïve Bayes memiliki kinerja terendah dengan akurasi 27.5% dan AUC 0.425. SVM memberikan hasil lebih baik dibandingkan Naïve Bayes, tetapi tetap lebih rendah dibandingkan Decision Tree. Penelitian ini merekomendasikan penerapan Decision Tree sebagai algoritma dengan optimasi melalui tuning hyperparameter, penambahan fitur relevan, dan penanganan ketidakseimbangan data. Hasil ini diharapkan dapat mendukung pengambilan keputusan perbankan secara lebih efektif dan efisien.

ABSTRACT

Predicting credit worthiness is an important step for banks to reduce the risk of bad credit. This research compares the performance of four classification algorithms, namely SVM, Naïve Bayes, Random Forest and Decision Tree using simulated datasets. The results obtained on the metrics of accuracy, precision, recall, F1 score, and AUC-ROC, show that Decision Tree has the best performance with 42.5% accuracy, 48.3% precision, 47.5% recall, 47.5% F1 score, and AUC 0.60, indicating its ability to is in differentiating credit worthiness. Random Forest achieved an accuracy of 37.5% and an AUC of 0.493, while Naïve Bayes had the lowest accuracy with an accuracy of 27.5% and an AUC of 0.425. SVM gives better results than Naïve Bayes but is still inferior to Decision Tree. This research recommends implementing a Decision Tree as the main model with optimization through hyperparameter tuning, adding relevant features, and handling data accounting. These results are expected to support banking decision making more effectively and efficiently.

Corresponding Author:

Wresti Andriani
Universitas Muhammadiyah Tegal
Jl. Melati no 27, Slerok, Tegal Timur, Kota Tegal, Jawa Tengah
Email: wresty.andriani@gmail.com

1. PENDAHULUAN

Kelayakan kredit merupakan elemen penting dalam pengelolaan risiko di sektor perbankan, terutama karena teknologi informasi memungkinkan analisis data nasabah yang lebih cepat dan efisien untuk mendukung pengambilan keputusan yang akurat [1]. Di era digital, banyak lembaga keuangan yang mulai mengadopsi pendekatan berbasis data untuk meningkatkan akurasi dalam proses evaluasi kredit [2]. Ketepatan dalam menentukan kelayakan kredit dapat membantu mengurangi potensi kredit macet sekaligus meningkatkan efisiensi pengelolaan portofolio kredit. Proses penilaian kelayakan kredit, yang sebelumnya dilakukan secara manual dengan mempertimbangkan dokumen nasabah dan intuisi petugas, kini dapat ditingkatkan dengan penerapan algoritma data mining [3]. Kesalahan pada penilaian kelayakan kredit dapat berdampak serius, baik berbentuk kredit macet ataupun kerugian keuangan [4].

Berbagai penelitian telah dilakukan untuk mengimplementasikan algoritma klasifikasi dalam prediksi kelayakan kredit. Pada penelitian yang dilakukan oleh Arora & Kaur [5] menggunakan algoritma *Bootstrap-Lasso* (*Bolasso*) untuk memilih fitur yang relevan dan stabil, diikuti dengan berbagai algoritma klasifikasi seperti *K-Nearest Neighbors (KNN)* untuk evaluasi risiko kredit, hasilnya menunjukkan bahwa kombinasi *Bolasso* dan algoritma klasifikasi dapat meningkatkan akurasi prediksi risiko kredit. Metode *KNN* memiliki kelemahan pada ketergantungan parameter k, kompleksitas yang tinggi dan sensitif terhadap ketidakseimbangan data, seperti pada penelitian yang dari Hartatik & Rosyid [6]. kelemahan *Bootstrap Lasso* juga lemah pada ketergantungan data sensitif terhadap hyperparameter juga *overfitting* pada data kecil, seperti pada peneltian dari Ariyoga [7]. Penelitian lain yang dilakukan oleh Sinaga & Yang, [8] dengan menggunakan metode *K*-mean , menemukan jumlah *cluster* yang optimal tanpa inisialisasi dan pemilihan parameter, metode ini juga memiliki kelemahan diantaranya tidak efektif untuk dimensi tinggi, sensitif terhadap skala data serta memiliki ketergantungan pada jumlah *claster*, seperti pada penelitian dari Lestari [9]. Sebagian besar penelitian yang telah dilakukan berfokus pada akurasi tanpa mengkaji kelebihan dan kekurangan model dan implementasinya di perbankan dengan menggunakan dataset dengan fitur terbatas tidak sesuai keadaan sebenarnya.

Pada penelitian ini akan menganalisa perbandingan kinerja empat algoritma klasifikasi yaitu *Decision Tree*, *Support Vector Machine*, *Naïve Bayes* dan *Random Forest* untuk prediksi kelayakan kredit pada bank BRI Tegal. Evaluasi dilakukan berdasarkan metrik akurasi, precision, recall, F1 score, dan AUC-ROC untuk memberikan gambaran menyeluruh mengenai kinerja masing-masing model. Hasil dari penelitian ini diharapkan dapat memberikan rekomendasi model terbaik sekaligus solusi praktis yang dapat diimplementasikan, sehingga dapat mengurangi kerugian yang diakibatkan dari kesalahan dalam mempredikasi kelayakan prediksi perbankan, khususnya pada bank BRI Tegal.

2. KAJIAN PUSTAKA

2.1. *State of The Art*

State of the art atau disingkat *SoTA* seringkali digantikan dengan istilah "*Cutting Edge*" atau "*Leading Edge*" yang berarti level tertinggi dan terkini dari pengembangan sebuah topik penelitian tertentu. Sebagai contoh *state of the art* produk iPhone di tahun 2021 adalah iPhone 13 dengan feature yang lebih baik dari produk-produk iPhone sebelumnya (versi iPhone 12 ke bawah) yakni support teknologi 5G, chip lebih cepat yakni versi A15, dan kualitas kamera lebih baik. Dalam konteks penelitian, *SoTA* adalah bagian/aktivitas dalam penelitian yang memaparkan hasil penelitian paling terkini dari topik yang dipilih, tren penelitian terkini dari topik yang kita pilih, permasalahan terkini dari penelitian dan praktis terkait topik yang dipilih dan standar-standar terkini dari topik yang dipilih..

State of the art atau disingkat *SoTA* seringkali digantikan dengan istilah "*Cutting Edge*" atau "*Leading Edge*" yang berarti level tertinggi dan terkini dari pengembangan sebuah topik penelitian tertentu. Sebagai contoh *state of the art* produk iPhone di tahun 2021 adalah iPhone 13 dengan feature yang lebih baik dari produk-produk iPhone sebelumnya (versi iPhone 12 ke bawah) yakni support teknologi 5G, chip lebih cepat yakni versi A15, dan kualitas kamera lebih baik. Dalam konteks penelitian, *SoTA* adalah bagian/aktivitas dalam penelitian yang memaparkan hasil penelitian paling terkini dari topik yang dipilih, tren penelitian terkini dari topik yang kita pilih, permasalahan terkini dari penelitian dan praktis terkait topik yang dipilih dan standar-standar terkini dari topik yang dipilih.

2.2. Landasan Teori

Landasan teori yang akan dipakai dalam penelitian ini meliputi klasifikasi pada *data mining*. Klasifikasi adalah salah satu teknik utama dalam data mining untuk memprediksi kelas atau kategori suatu objek berdasarkan fitur-fiturnya [10]. Proses ini melibatkan model yang dilatih menggunakan dataset berlabel untuk mempelajari pola yang digunakan untuk prediksi data baru. Metode klasifikasi yang digunakan pada penelitian ini yaitu *Decision Tree*, *Support Vector Machine*, *Naïve Bayes*, dan *Random Forest*. Pada penilaian kelayakan kredit, klasifikasi bertujuan untuk mengkategorikan calon peminjam ke dalam kelompok "layak" atau "tidak layak" mendapatkan pinjaman berdasarkan atribut tertentu seperti pendapatan, riwayat kredit, dan jumlah pinjaman [11].

Algoritma *Decision Tree* yaitu algoritma berbasis pohon yang membagi dataset menjadi kumpulan data yang lebih kecil (*subset*), berdasarkan fitur tertentu hingga setiap subset termasuk ke dalam satu kelas. *Decision Tree* sering digunakan karena interpretasinya (pemahaman) yang mudah dan kemampuannya menangani data numerik atau jenis data yang direpresentasikan dalam bentuk angka dan dapat digunakan untuk perhitungan matematis maupun data kategorikal yaitu jenis data yang terdiri dari kategori atau kelompok yang biasanya tidak memiliki urutan atau nilai numerik yang berarti [12]. Namun, algoritma ini cenderung *overfitting* pada dataset yang kecil tanpa pruning [13]. *Overfitting* adalah sebuah model yang mempelajari data latih terlalu mendetail hingga menangkap *noise* atau pola yang tidak relevan. Akibatnya, dapat menghasilkan performa buruk ketika diterapkan pada data baru (data uji) karena tidak mampu menggeneralisasi dengan baik, sedangkan pruning adalah teknik pada algoritma *Decision Tree*, untuk mengurangi ukuran dengan menghilangkan cabang yang tidak relevan atau terlalu rumit yang bertujuan untuk mencegah *overfitting* dan meningkatkan kemampuan generalisasi model pada data baru [14].

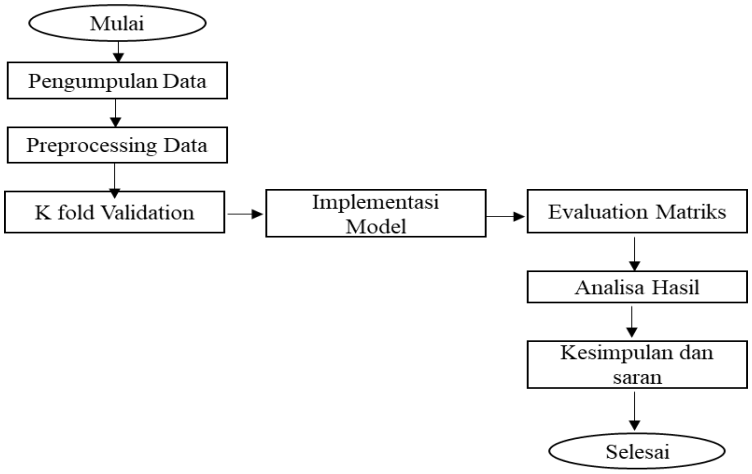
Support Vector Machine (*SVM*) adalah metode klasifikasi yang menggunakan *hyperplane* untuk memisahkan data ke dalam kelas yang berbeda. Algoritma ini efektif untuk data berdimensi tinggi dan mampu menangani hubungan non-linear melalui kernel trick. Namun, *SVM* membutuhkan *tuning* parameter yang baik untuk performa optimal dan kurang efisien pada dataset besar [15]. *Kernel Trick* adalah metode, untuk memproyeksikan data ke dimensi yang lebih tinggi tanpa harus menghitung koordinat di ruang dimensi tinggi sehingga pemisahan data dapat dipisahkan secara linear [16] .

Algoritma *Naïve Bayes* adalah metode berbasis probabilistik yang mengasumsikan setiap fitur bersifat independen. Algoritma ini dikenal cepat dan efisien pada dataset kecil. *Naive Bayes* cocok untuk aplikasi dengan banyak fitur kategorikal [14]. *Random Forest* adalah metode *ensemble learning* yaitu dengan menggabungkan banyak pohon keputusan untuk meningkatkan akurasi prediksi. Dengan menggunakan teknik *bagging* dan *random feature selection*, *Random Forest* mampu mengurangi *overfitting* dan menghasilkan model yang lebih *robust* atau kemampuan model untuk memberikan performa yang baik meskipun terdapat ketidaksempurnaan dalam data atau lingkungan pelatihan. dibandingkan dengan *Decision Tree* [18].

Kelayakan kredit dalam perbankan adalah proses evaluasi untuk menentukan apakah calon peminjam memiliki kemampuan dan kemauan untuk melunasi pinjaman mereka [19]. Proses ini melibatkan analisis atribut seperti pendapatan, riwayat kredit, usia, status pekerjaan, dan jumlah pinjaman [20]. Pada beberapa dekade terakhir, teknologi data mining dan *machine learning* telah banyak digunakan untuk meningkatkan efisiensi dan akurasi proses ini.

3. METODE PENELITIAN

Penelitian ini menggunakan metode kuantitatif dengan penerapan algoritma *machine learning* untuk klasifikasi kelayakan kredit perbankan. Tahapan pada penelitian ini dapat dilihat pada gambar 1 seperti di bawah ini.



Gambar 1. Diagram alur metodelogi penelitian

Tahapan pada Gambar 1 dapat dijelaskan bahwa penelitan ini dimulai dengan

3.1. Pengumpulan data

Data yang digunakan pada penelitian ini adalah data sekunder yang diperoleh dari data Bank BRI Tegal, yang diperoleh secara online. Data diperoleh sebanyak 200 dataset dan terdiri dari 7 variabel, seperti pada Tabel 1

Tabel 1. Data Pelanggan BRI Tegal

ID	Nama	Pendapatan	Jumlah Kredit	Durasi Kredit	Skor Kredit	Status Pembayaran
1	Puti Carla Rahayu, S.E.	6249610	85178298	36	669	Menunggak
2	Drs. Juli Haryanti, M. Ak	7047323	87136908	36	680	Gagal Bayar
3	Cornelia Zulaika, S. Ked	8391416	79917504	6	775	Menunggak
4	Nilam Prasasta	7446925	24312461	36	611	Gagal Bayar
5	R. Farhunnisa Mangunsong, S.H.	7175761	127139216	18	807	Gagal Bayar
…	…	…	…	…	…	…
200	Kairav Rajata	2985952	61802499	6	552	Gagal Bayar

Pada Tabel 1 menjelaskan, ID pelanggan merupakan identifikasi unik untuk setiap pelanggan, nama untuk nama pelanggan, pendapatan (Rp) adalah pendapatan bulanan pelanggan, jumlah kredit (Rp) adalah total jumlah kredit yang diajukan, durasi kredit (bulan) adalah lama waktu kredit yang disepakati, skor kredit adalah angka yang mencerminkan kelayakan kredit pelanggan dan status pembayaran adalah status apakah kredit pelanggan "lancar", "menunggak", atau "gagal bayar".

3.2. *Preprocessing* data

Preprocessing data adalah tahapan untuk memastikan data yang akan diproses bersih, konsisten, dan siap untuk diproses [21]. *Preprocessing* data diterapkan adalah data yang hilang (missing data) caranya yaitu menghilangkan baris atau kolom yang *missing values* [22]. Kemudian proses normalisasi data, untuk merubah data pada variabel dengan rentang yang berbeda, seperti pendapatan dan jumlah kredit, dengan menggunakan teknik normalisasi *min-max scaling* [23], rumusnya adalah:

$$X^1 = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

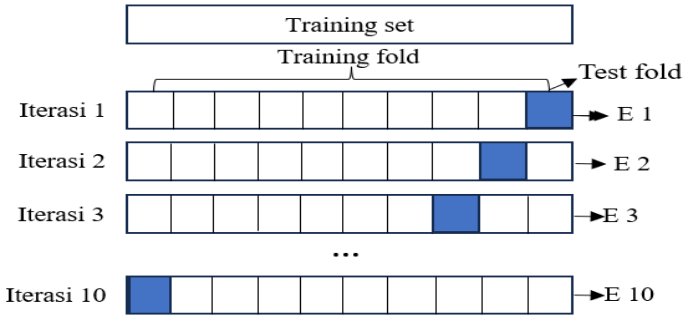
Dimana: X adalah nilai asli yang dinormalisasi, X_{min} adalah nilai minimum pada data, X_{max} adalah nilai maksimum dalam data dan X^1 adalah hasil normalisasi, pada rentang 0 dan 1.

Label target pada variabel status pembayaran dikonversi menjadi nilai numerik menggunakan teknik label *Encoding*, kelas lancar = 0, menunggak = 1, dan gagal bayar = 2.

3.3. *K-Fold Validation*

K-Fold Validation adalah teknik (*cross-validation*) yang dipakai dalam evaluasi kinerja model *machine*

learning. Teknik ini membagi dataset menjadi beberapa lipatan (*folds*) untuk memastikan bahwa model dilatih dan diuji secara bergantian [24]. Pada penelitian ini akan menggunakan k=10, maka dataset akan dibagi menjadi 10 lipatan, dan proses *K-Fold Validation* akan dijalankan 10 kali, setiap lipatan menjadi test data satu kali dan selebihnya menjadi data latih. Pada banyak penelitian menunjukkan K=10 adalah pilihan optimal untuk berbagai jenis dataset dan sering memberikan estimasi performa yang akurat [25]. Gambar Diagram *K-Fold Validation* dengan k=10 sebagai berikut:



Gambar 2. Diagram *K-Fold Validation* dengan k=10

Gambar 2 menunjukkan proses *K-Fold Cross Validation*, data dibagi menjadi beberapa *fold*, dan setiap f*old* menjadi *test set* (blok biru), fold lainnya menjadi *training set* (blok putih), untuk mengurangi *bias*.

3.4. Implementasi Model

Model yang digunakan dalam penelitian ini adalah *Decision Tree*, yang rumusnya adalah:

1. *Entropy (H),* digunakan untuk mengukur ketidakteraturan dalam dataset. Semakin tinggi *entropi*, semakin tidak teratur dataset tersebut, rumusnya:

$$H(S) = -\sum_{i=1}^{n} p_i \log_2(p_i) \quad (2)$$

   Dimana: *S* adalah dataset, *n* adalah jumlah kelas dalam dataset dan p_i adalah *probabilitas* elemen dalam kelas *i*

2. *Information Gain (IG)* digunakan untuk mengukur ketidakteraturan *(entropy)* setelah membagi dataset menjadi *fitur* tertentu. Fitur dengan IG tertinggi dipilih untuk membagi dataset, rumusnya adalah:

$$IG\ (S,A) = H(S) - \sum \frac{S_v}{S}.\ H\ (S_v) \quad (3)$$

   Dimana: *S* adalah *dataset*, *A* adalah *fitur* yang diuji, S_v adalah *subset* data ketika *A* bernilai *v* dan *S* adalah jumlah *elemen* dalam dataset *S.*

Selain itu menggunakan *Support Vector Machine*, dengan menggunakan rumus:

1. Persamaan *Hyperplane*, untuk mengetahui keputusan untuk memisahkan data, rumusnya adalah:

$$w.\ x + b = 0 \quad (4)$$

   Dimana: *w* adalah vektor bobot, *x* adalah vektoe fitur input, *b* adalah *bias* (konstanta).

2. *Margin* adalah jarak *hyperplane* dan titik terdekat dari masing-masing kelas, untuk meningkatkan *generalisasi*, rumusnya adalah:

$$Margin = \frac{2}{\|w\|} \quad (5)$$

   Dimana: *||w||: Norm* (panjang) dari vektor *w*

3. Fungsi *Optimasi Maximizing Margin,* dengan mencari *w* dan *b* yang memaksimalkan *margin*, yang setara dengan meminimalkan $||w||^2$ dengan syarat semua data terklasifikasi dengan benar, rumusnya adalah:

$$Minimize = \frac{1}{2}\|w\|^2 \quad (6)$$

Model ketiga yang digunakan *Naïve Bayes*, dengan menggunakan rumus:

1. *Teorema Bayes*, yang digunakan untuk menghitung probabilitas suatu kelas *C* diberikan *X:*

$$PC = \frac{P(CIX).\ P(C)}{P(X)} \quad (7)$$

Dimana: *P(CIX)* adalah probabilitas kelas *C* yang diberikan data *X*, $P(XIC)$ adalah probabilitas kelas *X* yang diberikan data *C*, *P(C)* adalah probabilitas awal kelas C dan *P(X*) adalah probabilitas data *X.*

2. Fungsi *Likelihood*, yang bergantung pada tipe data, $P\{X_i|C\}$ dihitung sebagai *frekwensi relatif*, dengan rumus:

$$P(X_i|C) = \frac{Jumlah\ data\ dengan\ X_i\ pada\ kelas\ C}{Jumlah\ total\ data\ dalam\ kelas\ C} \quad (8)$$

Model keempat yang digunakan dalam perbandingan ini adalah *metode Random Forest* yang menggabungkan banyak *Decision Trees* untuk meningkatkan *akurasi* dan *stabilitas* model, rumusnya yaitu:

1. *Bagging (Bootstrap Aggregating)* untuk membuat *dataset* latih baru dengan *sampling* ulang dari dataset asli secara acak dengan pengembalian, rumusnya adalah:

$$S_i = Bootstrap\ sample\ untuk\ S \quad (9)$$

Dimana: S_i adalah total dataset untuk setiap pohon dan *S* adalah dataset asli.

2. *Split Criterion (Impurity Reduction),* setiap pohon memilih fitur terbaik untuk membagi data atas *kriteria* tertentu yaitu:
   a. *Gini Index*, rumusnya:

$$Gini\ (S) = 1 - \sum_{i=1}^{n} {p_i}^2 \quad (10)$$

Dimana: *S* adalah dataset, *n* adalah jumlah kelas dan p_i adalah *proporsi* data di kelas *i*.

   b. *Entropy* rumusnya:

$$H(S) = -\sum_{i-1}^{n} p_i\ log_2(p_i) \quad (11)$$

   c. *Information Gain*, rumusnya:

$$IG\ (S, A) = H(S) - \frac{S_v}{S} H\ (S_v) \quad (12)$$

Fitur dengan nilai *gini index* atau *entopy* terbaik dipilih untuk membagi dataset

3.5. *Evaluation* Matriks

Evaluasi model *klasifikasi* menggunakan bermacam *metrik*, pada penelitian ini menggunakan *evaluasi* a*kurasi* adalah *proporsi prediksi* benar terhadap total *prediksi*, rumusnya adalah:

$$Akurasi = \frac{TP + TN}{TP+TN+FP+FN}\ 100\% \quad (13)$$

Dimana: *TP* adalah *prediksi* benar untuk kelas positif, *TN* adalah prediksi benar untuk kelas negatif, *FP* adalah *prediksi* salah, dimana kelas sebenarnya negatif tapi terprediksi positif dan *FN prediksi* salah, namun sebenarnya positif tapi terprediksi negatif [26] *Precision* adalah *proporsi prediksi positif* yang benar, rumusnya adalah:

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

Dimana: TP adalah jumlah prediksi benar untuk kelas positif dan FP adalah jumlah prediksi salah [27].

Recall adalah kemampuan model mendeteksi semua sampel positif, rumus untuk menghitung *recall* adalah:

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

Dimana: *TP* adalah jumlah prediksi benar untuk kelas positif dan *FN* adalah jumlah *prediksi* yang salah dimana model gagal mendekteksi kelas positif [28].

F1-Score adalah rata-rata dari *precision* dan *recall* untuk menghitungnya menggunakan rumus:

$$F1\ Score\ =\ 2\ x\ \frac{Precision.Recall}{Precision + Recall} \quad (16)$$

Dimana: *Precision* adalah *proporsi prediksi positif* yang benar dan *recall* adalah kemampuan model mendeteksi semua sampel positif [29].

AUC-ROC adalah menilai sejauh mana mode mampu untuk membedakan dari kelas negatif dan positif. tidak memiliki rumus langsung seperti *precision* atau *recall*, karena merupakan hasil dari *integrasi numerik* yang mengukur area di bawah kurva *ROC*. Rumus *ROC* adalah:

$$TPR\ =\ \frac{TP}{TP + FN} \text{ dan } FPR = \frac{FP}{FP + TN} \quad (17)$$

Nilai *AUC* berada antara 0 dan 1, dimana 1adalah model sempurna, 0.5: Model tidak lebih baik dari tebakan acak, < 0.5 performa lebih buruk dari tebakan acak. Metrik ini penting untuk memastikan model yang digunakan dapat memberikan prediksi yang andal dalam konteks kelayakan kredit.

4. HASIL DAN PEMBAHASAN

Dengan menggunakan data pada Bank BRI Tegal, sebanyak 200 data yang berhasil dikumpulkan dan setelah melalui proses *preprocessing* data yang meliputi data *missing*, *normalisasi* dan pengkategorian, diperoleh tabel seperti pada Tabel 2

Tabel 2. Data Pelanggan Setelah Preprocessing Data

ID	Pendapatan (Rp)	Jumlah Kredit (Rp)	Durasi (Bulan)	Skor Kredit	Status Pembayaran
1	0.68	0.54	0.88	0.66	1
2	0.72	0.57	0.88	0.68	2
3	0.85	0.49	0.15	0.77	1
4	0.75	0.12	0.88	0.88	2
5	0.72	0.72	0.44	0.44	2
…	…	…	…	…	…
200	0.23	0.31	0.15	0.55	2

Dari Tabel 2 diatas merupakan data hasil proses *preprocessing* data, pada kolom Pendapatan dan Jumlah Kredit dinormalisasi menggunakan rumus *Min-Max Scaling* untuk merubah rentang nilai ke dalam i*nterval* [0, 1]. Pada kolom Status Pembayaran dikonversi menjadi nilai *numerik* dengan lancar = 0, menunggak = 1 dan gagal bayar = 2.

Berikut adalah implementasi Python untuk *Decision Tree, Support Vector Machine, Naïve Bayes dan Random Forest* yang mencakup *preprocessing*, *pembagian dataset*, *pelatihan model,* dan *evaluasi* menggunakan *akurasi, presisi, recall, F1-score, serta AUC-ROC*, seperti pada Gambar 3. Dari hasil implemenasi menggunakan Phyton untuk metode *SVM, NB, RF dan DT* pada gambar 3

```
# Load the newly uploaded dataset
file_path = "/mnt/data/ kredit_bri_tegal_dengan_nama.xlsx"
df = pd. read_excel(file_path)
# Re-importing necessary libraries
from sklearn. model_selection import train_test_split.
from sklearn. tree import DecisionTreeClassifier.
from sklearn. ensemble import RandomForestClassifier.
from sklearn. naive_bayes import GaussianNB.
from sklearn. metrics import (
    accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, roc_curve
)
from sklearn. preprocessing import MinMaxScaler, LabelEncoder
import matplotlib. pyplot as plt
import pandas as pd.
# Preprocessing
# Normalize numerical columns
scaler = MinMaxScaler ()
df [['Pendapatan (Rp)', 'Jumlah Kredit (Rp)']] = scaler.fit_transform (
    df [['Pendapatan (Rp)', 'Jumlah Kredit (Rp)']]
)
# Encode categorical column
```



```
label_encoder = LabelEncoder ()
df ['Status Pembayaran'] = label_encoder.fit_transform (df ['Status Pembayaran'])
# Features and target
X = df [['Pendapatan (Rp)', 'Jumlah Kredit (Rp)', 'Durasi Kredit (Bulan)', 'Skor Kredit']]
y = df ['Status Pembayaran']
# Split dataset into train and test sets
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_state=42)
# Models
models = {
   "Decision Tree": DecisionTreeClassifier (),
   "Random Forest": RandomForestClassifier (),
   "Naive Bayes": GaussianNB ()
}
# Initialize results
results = []
# Evaluate each model
for model_name, model in models. items ():
   # Train the model
   model.fit (X_train, y_train)

   # Predictions
   y_pred = model. predict(X_test)
   y_pred_prob = model. predict_proba(X_test) [: 1] if hasattr (model, "predict_proba") else None.
   # Metrics
   accuracy = accuracy_score (y_test, y_pred)
   precision = precision_score (y_test, y_pred, average='weighted', zero_division=0)
   recall = recall_score (y_test, y_pred, average='weighted', zero_division=0)
   f1 = f1_score (y_test, y_pred, average='weighted', zero_division=0)
   auc = roc_auc_score (y_test, y_pred_prob) if y_pred_prob is not No one else None
   # Append results
   results. append ({
      "Model": model_name,
      "Accuracy": accuracy,
      "Precision": precision,
      "Recall": recall,
      "F1 Score": f1,
      "AUC": auc
   })
   # Plot ROC Curve
   if y_pred_prob is not None:
      fpr, tpr, _ = roc_curve (y_test, y_pred_prob)
      plt. plot (fpr, tpr, label=f"{model_name} (AUC = {auc:.2f})")
# Plot ROC Curve
plt. plot ([0, 1], [0, 1], 'k--') # Diagonal line.
plt. title ('ROC Curve')
plt. xlabel ('False Positive Rate')
plt. ylabel ('True Positive Rate')
plt. legend ()
plt. show ()
# Convert results to a DataFrame for display
results_df = pd. DataFrame(results)
import ace_tools as tools, tools. display_dataframe_to_user (name="Model Evaluation Metrics",
dataframe=results_df)
```

Gambar 3. Implementasi Python untuk *Decision Tree*, *Random Forest*, *Naive Bayes*, dan *SVM*

Dari eksperimen tersebut akhirnya diperoleh hasil evaluasi berupa *accuracy, Precision, Recall, F1 Score dan AUC* seperti pada Tabel 2

Tabel 2. Hasil Evaluasi dari DT, RF, NB dan SVM

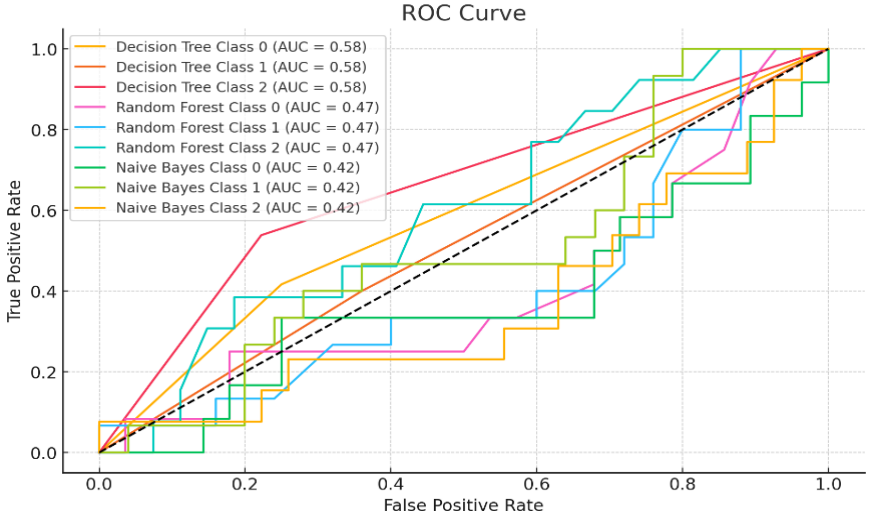
Model	Accuracy	Precision	Recall	F1 Score	AUC
Decision Tree	0,425	0.483	0.475	0.475	0.60
Random Forest	0.375	0.362	0.375	0.367	0.493
Naïve Bayes	0.275	0.252	0.275	0.252	0.425
SVM	0.325	0.325	0.325	0.293	

Tabel 2 yang berisi hasil evaluasi model *SVM, NB, RF dan DT*. Pada metode *Decision Tree*, menghasilkan *accuracy* sebesar 42.5% (Model memiliki kemampuan moderat dalam memprediksi dengan benar), *Precision* sebesar 48.3% (Hampir separuh prediksi positif model benar-benar benar), *Recall* sebesar47.5% (Model berhasil mendeteksi hampir separuh data positif). *F1 Score* sebesar 47.5% (Keseimbangan antara *precision dan recall* cukup baik). *AUC* sebesar 0.60 (Keberhasilan untuk membedakan kelas positif dan negatif cukup baik).

Pada metode *Random Forest* menghasikan *accuracy* sebesar 37.5% (lebih rendah dari *Decision Tree*), *Precision* sebesar 36.2% (lebih rendah dari *prediksi* positif yang benar), *Recall* sebesar 37.5% (deteksi data positif cukup rendah), *F1 Score* sebesar 36.7% (*precision* dan *recall* masih seimbang), *AUC* sebesar 0.493.

Untuk metode *Naïve Bayes* hasilnya *accuracy*: 27.5%, *Precision*: 25.2%, *Recall* sebesar 27.5% (Kemampuan mendeteksi data positif sangat rendah), *F1 Score* sebesar 25.2% (Keseimbangan *precision* dan *recal*l kecil). *AUC* sebesar 0.425

Pada metode SVM menghasilkan *accuracy* sebesar 32.5% lebih baik dari *Naïve Bayes*, lebih rendah dari *Decision Tree* dan *Random Forest*, *Precision* sebesar 32.5% *Recall* sebesar 32.5% (setara dengan *precision*), *F1 Score* sebesar 29.3% (Keseimbangan *precision* dan *recall* lebih kecil dibandingkan *Random Forest*), sedangkan untuk *AUC* pada SVM tidak dihitung dalam tabel ini karena SVM dengan *kernel linear* tidak mendukung *probabilitas* secara langsung. Gambar *kurva* pada *ROC* untuk ke empat metode bisa dilihat pada gambar 4.



Gambar 4. Kurva *ROC* dari *NB, DT, dan RF*

Dari gambar 4 tampak bahwa sumbu *X (FPR*), sumbu *Y (TPR),* warna metode *Decision Tree* (Merah): *ROC* untuk kelas 0 dengan AUC 0.58, metode *Naïve Bayes* (Kuning): *ROC* untuk kelas 2 dengan *AUC 0.42*. *Decision Tree*, *AUC* Class 0, 1, 2 = 0.58, *performa*nya sedikit lebih baik dibandingkan *Random Forest*, *Random Forest: AUC* Class 0, 1, 2 = 0.47, tidak mampu membedakan kelas positif dan negatif. Begitu juga dengan *Naïve Bayes*, performanya lebih buruk dibandingkan *Random Forest*. *AUC Class* 0, 1, 2 = 0.42, begitu juga kurva *ROC* nya, *Decision Tree* adalah model terbaik pada grafik ini dengan *AUC* sekitar 0.58 untuk semua kelas. *Random Forest dan Naïve Bayes* memiliki *AUC* mendekati atau di bawah 0.5, menunjukkan kinerja yang rendah dalam membedakan kelas.

5. SIMPULAN DAN SARAN

Dari hasil penelitian yang dilakukan, disimpulkan bahwa Decision Tree menunjukkan kinerja terbaik dibandingkan dengan model lain dengan nilai akurasi 42.5%, *precision* 48.3%, *recall* 47.5%, *F1 score* 47.5%, dan *AUC* 0.60, mampu membedakan kelas positif dan negatif, menjadikannya pilihan yang lebih andal untuk prediksi kelayakan kredit, sedangkan *Naïve Bayes* memiliki performa terendah dengan akurasi hanya 27.5% dan *AUC* 0.425. Metode *Random Forest* menunjukkan performa moderat dengan akurasi 37.5% dan *AUC* mendekati tebakan acak (0.493) dan *SVM* lebih baik dari *Naïve Bayes* tetapi masih lebih

rendah dari *Decision Tree*, dengan akurasi 32.5%. Pada penelitian berikutnya disarankan untuk kualitas dataset dengan menambahkan fitur relevan, seperti riwayat kredit dan rasio cicilan terhadap pendapatan, yang dapat membantu meningkatkan akurasi model dan mengintegrasikan model *Decision Tree* ke dalam sistem pendukung keputusan perbankan untuk mempermudah proses evaluasi kelayakan kredit secara *real-time*.

DAFTAR PUSTAKA

[1] R. Oktapiani, D. Prayudi, A. Fajria, N. S. Z. Nufus, and R. N. Lestari, "Sistem Pendukung Keputusan Untuk Menentukan Managemen Kelayakan Pemberian Kredit Di Bank Mandiri Taspen Sukabumi Menggunakan Metode Analytic Hierarchy Process," *Indones. J. Softw. Eng*., vol. 8, no. 1, pp. 36–45, 2022, doi: 10.31294/ijse.v8i1.12054.

[2] A. Gupta, "Data Mining for Risk Assessment in Banking and Finance: A Review," *Turkish J. Comput. Math. Educ*., 2019, doi: 10.17762/turcomat.v10i1.13550.

[3] F. Martinelli, F. Mercaldo, D. Raucci, and A. Santone, "Bank Credit Risk Management based on Data Mining Techniques," *Jurnal,* pp. 837–843, 2020, doi: 10.5220/0009371808370843.

[4] SJIN PHEN and Suartini, "Analisa Kredit Macet Pada Bank Umum Milik Negara Termasuk Dalam Tindak Pidana Korupsi Kerugian Keuangan Negara," *J. Huk. to-ra Huk*., vol. 10, no. 2, pp. 200–219, 2024, doi: 10.55809/tora.v10i2.367.

[5] N. Arora and P. Kaur, "A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment," *Appl. Soft Comput*., vol. 86, 2020, doi: 10.1016/j.asoc.2019.105936.

[6] H. Hartatik and R. Rosyid, "Pengaruh User Profiling Pada Rekomendasi Sistem Menggunakan K Means Dan Knn," *J. Inf. Syst. Manag*., vol. 2, no. 1, pp. 13–18, 2020, doi: 10.24076/joism.2020v2i1.199.

[7] D. ARIYOGA, "Perbandingan Metode Seleksi Fitur Filter, Wrapper, Dan Embedded Pada Klasifikasi Data Nirs Mangga Menggunakan Random Forest Dan Support Vector Machine (Svm)," 2022.

[8] K. P. Sinaga and M.-S. Yang, "Unsupervised K-Means Clustering Algorithm," *IEEE Access,* vol. 8, pp. 80716–80727, 2020, doi: 10.1109/ACCESS.2020.2988796.

[9] D. Lestari, A. C. Fauzan, and H. Harliana, "PENERAPAN ALGORITMA PILLAR UNTUK OPTIMASI PENENTUAN TITIK AWAL CENTROID PADA ALGORITMA K-MEANS CLUSTERING," *JOISIE (Journal Inf. Syst. Informatics Eng*., vol. 6, no. 1, pp. 15–24, 2022.

[10] V. Vibrianti, E. Wahyudin, K. Kaslani, D. Pratama, and G. Dwilestari, "Klasifikasi Barang Produksi Pada Tnt. Guitar Workshop Dengan Metode Naive Bayes Menggunakan Rapid Miner," *JATI (Jurnal Mhs. Tek. Inform*., vol. 8, no. 2, pp. 1432–1438, 2024, doi: 10.36040/jati.v8i2.8966.

[11] D. Irawan, E. B. Perkasa, Y. Yurindra, D. Wahyuningsih, and E. Helmud, "Perbandingan Klassifikasi SMS Berbasis Support Vector Machine, Naive Bayes Classifier, Random Forest dan Bagging Classifier," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 10, no. 3, pp. 432–437, 2021, doi: 10.32736/sisfokom.v10i3.1302.

[12] J. Good, T. Kovach, K. Miller, and A. Dubrawski, "Feature Learning for Interpretable, Performant Decision Trees," 2023, [Online]. Available: https://consensus.app/papers/feature-learning-for-interpretable-performant-decision-good-kovach/157dbca6735c58cba287a21e9e0f60df/

[13] V. G. Costa and C. E. Pedreira, "Recent advances in decision trees: an updated survey," *Artif. Intell. Rev.,* vol. 56, no. 5, pp. 4765–4800, 2023, doi: 10.1007/s10462-022-10275-5.

[14] D. Vos, J. Vos, T. Li, Z. Erkin, and S. Verwer, "Differentially-Private Decision Trees and Provable Robustness to Data Poisoning," 2023, [Online]. Available: https://consensus.app/papers/differentiallyprivate-decision-trees-and-provable-vos-vos/d0a907e3edd259faa4e5cdabdc4bdf44/

[15] A. Kurani, P. Doshi, A. Vakharia, and M. Shah, "A Comprehensive Comparative Study of Artificial Neural Network (ANN) and Support Vector Machines (SVM) on Stock Forecasting," *Ann. Data Sci.*, vol. 10, no. 1, pp. 183–208, 2023, doi: 10.1007/s40745-021-00344-x.

[16] O. Almomani, "A feature selection model for network intrusion detection system based on pso, gwo, ffa and ga algorithms," *Symmetry (Basel).*, vol. 12, no. 6, pp. 1–20, 2020, doi: 10.3390/sym12061046.

[17] H. Chen, S. Hu, R. Hua, and X. Zhao, "Improved naive Bayes classification algorithm for traffic risk management," *EURASIP J. Adv. Signal Process*., vol. 2021, no. 1, 2021, doi: 10.1186/s13634-021-00742-6.

[18] J. Bai, Y. Li, J. Li, X. Yang, Y. Jiang, and S.-T. Xia, "Multinomial random forest," *Pattern Recognit*., vol. 122, p. 108331, 2022, doi: https://doi.org/10.1016/j.patcog.2021.108331.

[19] U. Yani, "*Analisis Literasi Keuangan dan Promosi LPMUKP dalam Pengambilan Keputusan Kredit oleh Para Nelayan di Desa Pangalasiang Kecamatan Sojol Kabupaten Donggala*," 2023, Universitas Islam Negeri Datokarama Palu.

[20] R. A. Winita, "*PENGARUH PENILAIAN KELAYAKAN KREDIT TERHADAP KOLEKTIBILITAS KREDIT PADA BPR*," 2021, Program Studi Magister Akuntansi Sekolah Pasca Sarjana Universitas Widyatama.

[21] P. Mishra, A. Biancolillo, J. M. Roger, F. Marini, and D. N. Rutledge, "New data preprocessing trends based on ensemble of multiple preprocessing techniques," *TrAC Trends Anal. Chem*., vol. 132, p. 116045, 2020, doi: https://doi.org/10.1016/j.trac.2020.116045.

[22] J. R. van Ginkel, M. Linting, R. C. A. Rippe, and A. van der Voort, "Rebutting Existing Misconceptions About Multiple Imputation as a Method for Handling Missing Data," *J. Pers. Assess.*, vol. 102, no. 3, pp. 297–308, 2020, doi: 10.1080/00223891.2018.1530680.

[23] F. Adams, R. A. Dwi Anggoro, M. B. Satria, A. W. Oktavia, and N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma Naïve Bayes, Decision Tree, dan Support Vector Machine," *Semin. Nas. Mhs. Ilmu Komput. dan Apl.*, vol. 2, no. 2, pp. 260–268, 2021, [Online]. Available: https://conference.upnvj.ac.id/index.php/senamika/article/view/1744

[24] X. Zhang and C.-A. Liu, "Model averaging prediction by K-fold cross-validation," *J. Econom*., vol. 235, no. 1, pp. 280–301, 2023, doi: https://doi.org/10.1016/j.jeconom.2022.04.007.

[25] J. Saintikom, J. Sains, M. Informatika, F. Fiddin, M. Y. Syahbarna, and M. Ridwan, "Penggunaan Supervised Learning untuk Prediksi Validitas Ulasan Negatif Aplikasi Tokopedia Berdasarkan Pengalaman Pengguna Ahli," *Jurnal,* vol. 23, pp. 409–417, 2024.

[26] E. N. R. Khakim, A. Hermawan, and D. Avianto, "Implementasi Correlation Matrix Pada Klasifikasi Dataset Wine," *JIKO (Jurnal Inform. dan Komputer)*, vol. 7, no. 1, p. 158, 2023, doi: 10.26798/jiko.v7i1.771.

[27] D. Chrisinta and J. E. Simarmata, "Analisis Sentimen Penilaian Masyarakat Terhadap Pejabat Publik Menggunakan Algoritma Naïve Bayes Classifier," *Komputika J. Sist. Komput*., vol. 12, no. 1, pp. 93–101, 2023.

[28] D. G. Manurung et al., "Deteksi Dan Klasifikasi Hama Potato Beetle Pada Tanaman Kentang Menggunakan YOLOV8," *J. Teknol. Inf. dan Ilmu Komput*., vol. 11, no. 4, pp. 723–734, 2024.

[29] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics,* vol. 21, no. 1, pp. 1–13, 2020, doi: 10.1186/s12864-019-6413-7.