Terbit online pada laman web jurnal: **http://journal.isas.or.id/index.php/JACOST**

JOURNAL OF APPLIED COMPUTER SCIENCE AND TECHNOLOGY (JACOST)

Vol. 6 No. 2 (2025) 92 – 100 | ISSN: 2723-1453 (Media Online)

Klasifikasi Pemohon Pinjaman dengan *Hyperparameter Tuning* dan Teknik Penyeimbangan Data

Donata Yulvida[1], Stefanie Quinevera[2], Ricky Mardianto[3], Steven Joses[4]
[1,2]Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Universitas Widya Dharma Pontianak
[3,4]Jurusan Informatika, Fakultas Teknologi Informasi, Universitas Widya Dharma Pontianak
[1]donata_yulvida@widyadharma.ac.id, [2]stefanie_quinevera@widyadharma.ac.id, [3]ricky_mardianto@widyadharma.ac.id, [4]stevenjoses@widyadharma.ac.id

Abstract

Loan classification is a critical component of credit risk management, as it categorizes loans based on risk levels and supports the financial stability of banks, where loan-related income represents a substantial share of assets. Effective classification aims to ensure secure asset allocation, minimize credit risk, and prevent potential repayment issues. This study enhances loan classification performance through two strategies: hyperparameter optimization of Decision Tree and Random Forest algorithms, and data balancing techniques to address class imbalance. Experimental results show that the Decision Tree achieves 89.21% accuracy with an F1-Score of 70.17%, while the Random Forest demonstrates higher performance, reaching 94.04% accuracy and an F1-Score of 79.75%. Random Oversampling reduces bias toward majority classes by improving model sensitivity, while hyperparameter tuning with GridSearchCV identifies optimal parameter settings, thereby strengthening predictive performance. The findings highlight that combining data balancing with hyperparameter optimization effectively improves accuracy and F1-Scores. These approaches are not limited to the algorithms tested but can also be applied to other classification methods, offering broader potential for enhancing credit risk prediction in banking.

Keywords: Decision Tree, GridSearchCV, Loan Classification, Random Forest, Random Oversampling

Abstrak

Klasifikasi pinjaman merupakan komponen penting dalam manajemen risiko kredit, karena berfungsi untuk mengategorikan pinjaman berdasarkan tingkat risiko serta mendukung stabilitas keuangan bank, di mana pendapatan yang terkait dengan pinjaman mewakili porsi signifikan dari aset. Klasifikasi yang efektif bertujuan untuk memastikan alokasi aset yang aman, meminimalkan risiko kredit, serta mencegah potensi masalah pembayaran kembali. Penelitian ini meningkatkan kinerja klasifikasi pinjaman melalui dua strategi, yaitu optimasi *hyperparameter* pada algoritma *Decision Tree* dan Random Forest, serta teknik penyeimbangan data untuk mengatasi ketidakseimbangan kelas. Hasil eksperimen menunjukkan bahwa *Decision Tree* mencapai akurasi sebesar 89,21% dengan *F1-Score* sebesar 70,17%, sedangkan *Random Forest* menunjukkan kinerja yang lebih tinggi dengan akurasi mencapai 94,04% dan *F1-Score* sebesar 79,75%. *Random Oversampling* mengurangi bias terhadap kelas mayoritas dengan meningkatkan sensitivitas model, sementara penyetelan *hyperparameter* menggunakan *GridSearchCV* mampu mengidentifikasi konfigurasi parameter optimal sehingga memperkuat kinerja prediksi. Temuan ini menegaskan bahwa kombinasi antara penyeimbangan data dan optimasi *hyperparameter* secara efektif dapat meningkatkan akurasi dan *F1-Score*. Pendekatan tersebut tidak terbatas pada algoritma yang diuji, melainkan juga dapat diterapkan pada metode klasifikasi lainnya, sehingga menawarkan potensi lebih luas untuk meningkatkan prediksi risiko kredit dalam perbankan.

Kata kunci: *Decision Tree*, *GridSearchCV*, Klasifikasi Pinjaman, *Random Forest*, *Random Oversampling*

1. Pendahuluan

Klasifikasi pinjaman merupakan langkah penting dalam manajemen risiko kredit keuangan yang digunakan untuk mengelompokkan pinjaman berdasarkan tingkat risiko yang terkait. Proses ini sangat krusial bagi bank dan lembaga keuangan dalam upaya mengelola risiko kredit secara efektif. Dalam industri perbankan, penyaluran pinjaman merupakan transaksi keuangan yang memiliki peran signifikan terhadap keberhasilan bank. Sebagian besar aset bank secara langsung diperoleh dari keuntungan yang dihasilkan melalui pinjaman yang disetujui [1]. Oleh karena itu, bank berupaya meminimalkan risiko kredit dengan melakukan evaluasi menyeluruh terhadap status pinjaman melalui proses penilaian yang cermat. Langkah ini dilakukan untuk menghindari kemungkinan terjadinya praktik ilegal maupun peristiwa tak terduga yang dapat menghambat pelunasan pinjaman [2].

Tujuan pemberian pinjaman dapat bervariasi sesuai dengan kebutuhan nasabah. Secara umum, pinjaman dibedakan menjadi dua jenis, yaitu pinjaman terbuka dan pinjaman tertutup. Contoh pinjaman terbuka yaitu kartu kredit. Sementara itu, pinjaman tertutup akan berkurang nilainya setiap kali dilakukan pembayaran, sehingga jumlah pinjaman menurun seiring cicilan yang dibayarkan. Dalam konteks hukum, jenis pinjaman tertutup merupakan ketentuan yang tidak dapat diubah oleh peminjam. Beberapa contoh umum pinjaman tertutup adalah pinjaman pribadi, hipotek, kredit kendaraan bermotor, cicilan atau angsuran, serta pinjaman pendidikan. Adapun pinjaman dengan jaminan merupakan jenis pinjaman yang dilindungi oleh aset tertentu, seperti rumah, kendaraan, maupun tabungan [3].

Keputusan pemberian pinjaman bergantung pada berbagai karakteristik peminjam yang mencerminkan kemampuan mereka untuk melunasi utang. Salah satu karakteristik penting yang menjadi pertimbangan adalah riwayat kredit peminjam. Namun, informasi tersebut tidak selalu tersedia. Sebagai contoh, imigran, mahasiswa, dan profesional muda memerlukan waktu untuk membangun riwayat kredit mereka [4].

Proses persetujuan pinjaman masih sangat bergantung pada prosedur manual. Hal ini menjadi tantangan yang cukup besar, karena berkaitan dengan efisiensi dan akurasi. Proses ini menugaskan manajer bank secara individu untuk menilai kelayakan serta risiko gagal bayar dari calon peminjam. Dampak dari proses manual ini cukup signifikan, mulai dari potensi kerugian finansial bagi bank hingga skenario ekstrem berupa gangguan sistemik yang dapat merugikan perekonomian secara lebih luas [5]. Dalam konteks tersebut, model prediksi pinjaman berbasis *machine learning* dapat sangat bermanfaat bagi bank karena mampu memproses pengajuan pinjaman sekaligus menentukan apakah aman untuk memberikan pinjaman kepada pemohon atau tidak [6]. Salah satu metode yang populer adalah penggunaan algoritma *Random Forest*, yang memungkinkan pengambilan keputusan secara akurat berdasarkan himpunan data yang kompleks.

Meskipun klasifikasi pinjaman merupakan langkah penting dalam manajemen risiko kredit keuangan untuk mengelompokkan pinjaman berdasarkan tingkat risiko, masih terdapat beberapa celah penelitian yang perlu dieksplorasi lebih lanjut. Salah satu celah utama adalah kurangnya metodologi yang jelas dan terperinci terkait *hyperparameter tuning* serta optimasi pada algoritma *Random Forest* dan *Decision Tree* untuk mencapai kinerja yang optimal. Selain itu, permasalahan ketidakseimbangan data antara kategori kredit aman dan beresiko tetap menjadi tantangan signifikan yang dapat memengaruhi kemampuan model dalam menghasilkan prediksi yang akurat. Model prediksi pinjaman juga perlu ditingkatkan untuk mencegah bias terhadap kelompok tertentu. Terakhir, keterbatasan studi kasus praktis dan penerapan nyata menunjukkan perlunya penelitian lebih lanjut guna memberikan bukti empiris mengenai efektivitas model klasifikasi. Upaya untuk menutup celah penelitian ini sangat penting dalam mengembangkan model klasifikasi pinjaman yang lebih andal dan efektif, sehingga dapat meningkatkan manajemen risiko kredit di sektor perbankan.

Untuk mencapai performa optimal, penerapan model *Random Forest* sering kali melibatkan proses penyesuaian parameter atau *hyperparameter tuning*. Selain itu, dalam bidang keuangan, ketidakseimbangan data antara kategori kredit aman dan beresiko dapat memengaruhi kemampuan model dalam menghasilkan prediksi yang akurat.

Tahun 2021, [7] melakukan penelitian efektivitas berbagai teknik penanganan data tidak seimbang dalam konteks prediksi kanker paru-paru menggunakan dua dataset medis besar, yaitu PLCO dan NLST. Kedua dataset memiliki rasio ketidakseimbangan sekitar 25:1, di mana jumlah kasus negatif jauh lebih banyak daripada kasus positif. Hasilnya menunjukkan bahwa teknik over-sampling, khususnya *Random Over-Sampling* (ROS), memberikan performa paling stabil dan akurat, terutama saat dikombinasikan dengan algoritma *Random Forest*.

Tahun 2021, [6] melakukan penelitian penerapan algoritma klasifikasi *Random Forest* dalam memprediksi kelayakan pemberian pinjaman oleh lembaga keuangan. Meningkatnya jumlah pemohon pinjaman, proses verifikasi manual menjadi tidak efisien dan rentan terhadap kesalahan. Oleh karena itu, penulis mengusulkan penggunaan model pembelajaran mesin untuk mengotomatisasi proses evaluasi kelayakan pemohon berdasarkan data historis. Hasil penelitian menunjukkan bahwa algoritma *Random Forest* memiliki performa terbaik dibandingkan algoritma lain seperti *Support Vector Machine*, *Decision Tree*, dan *Logistic Regression*. *Random Forest* mampu menangani data berdimensi tinggi, mengurangi risiko *overfitting*, dan memberikan akurasi prediksi yang tinggi. Model ini efektif dalam membantu bank mengambil keputusan yang lebih cepat dan tepat dalam proses persetujuan pinjaman, serta dapat meningkatkan efisiensi dan akurasi dalam manajemen risiko kredit.

Tahun 2024, penelitian yang dilakukan oleh [8] membuktikan penggunaan *Grid Search Cross Validation* yang secara signifikan meningkatkan performa algoritma *machine learning* dalam memprediksi gagal bayar pinjaman. Teknik ini mengoptimalkan *hyperparameter* setiap algoritma melalui pencarian kombinasi terbaik secara sistematis. Hasil penelitian menunjukkan bahwa akurasi meningkat pada *Logistic Regression* (5%), KNN (4%), *Random Forest* (3%), *Decision Tree* (3%), dan XGBoost (2%). Melalui optimasi dan validasi silang ini, model menjadi lebih akurat sehingga mampu menggeneralisasi data

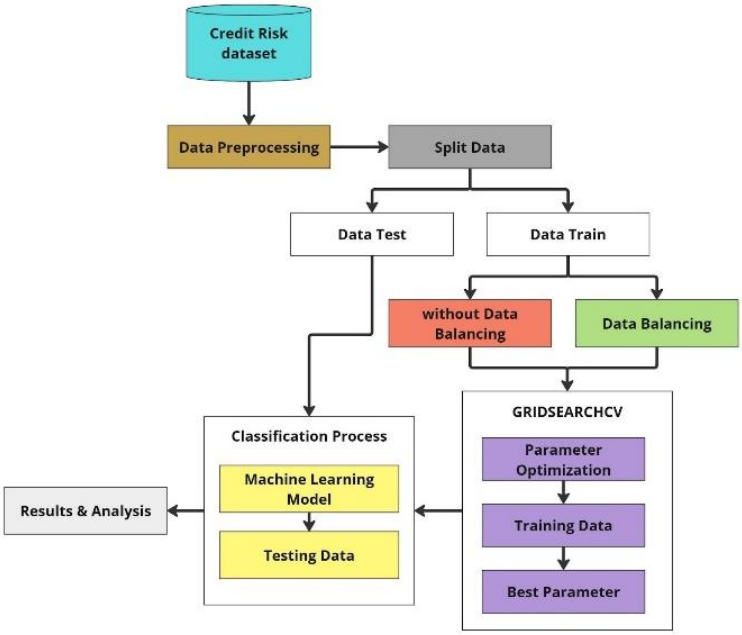
DOI : https://doi.org/10.52158/krjtrh05

dengan baik, serta meningkatkan keandalan sistem penilaian risiko lembaga pembiayaan.

Penelitian ini bertujuan untuk meningkatkan akurasi klasifikasi pinjaman dalam manajemen risiko kredit. Fokus utama diarahkan pada dua pendekatan, yaitu penerapan *hyperparameter tuning* pada algoritma *Random Forest* serta penggunaan teknik penyeimbangan data untuk mengatasi ketidakseimbangan kelas. Dengan memanfaatkan kombinasi kedua metode ini, diharapkan dapat memberikan prediksi risiko kredit yang lebih andal, sehingga dapat membantu lembaga keuangan dalam mengambil keputusan yang lebih tepat dan responsif di lingkungan bisnis yang dinamis.

2. Metode Penelitian

Metode penelitian merupakan prosedur atau pendekatan ilmiah yang digunakan untuk mengumpulkan data yang diperlukan dalam suatu studi. Metode penelitian dalam studi ini digambarkan pada Gambar 1.



Gambar 1. Metode Penelitian

Pada penelitian ini, metode *Random Forest* dan *Decision Tree* yang telah dioptimalkan digunakan untuk klasifikasi pinjaman. Penelitian ini melibatkan beberapa tahapan, dimulai dari pengumpulan data, pra pemprosesan data, penyeimbangan data, pelatihan model, prediksi data, hingga analisis hasil.

2.1. Sumber Data

Penelitian ini menggunakan *dataset* yang diperoleh dari *Kaggle.com* dengan nama *Credit Risk Dataset*. *Dataset* tersebut terdiri atas 12 fitur dan mencakup variabel keluaran. Jumlah data yang digunakan dalam penelitian ini adalah sebanyak 32.581 data. Fitur-fitur yang terdapat dalam *dataset* ditunjukkan pada Tabel 1.

2.2. Pra pemrosesan data

Pra pemrosesan data merupakan langkah awal dalam menyiapkan data masukan agar sesuai dan siap digunakan pada tahap selanjutnya. Hal ini bertujuan untuk mencapai kinerja sistem yang optimal. Analisis secara mendalam melalui teknik pra pemrosesan data dapat meningkatkan akurasi prediksi dan klasifikasi yang dihasilkan oleh model yang digunakan. Pra pemrosesan data dianggap sangat penting dan memerlukan upaya teknis yang cukup besar dalam pengembangan model. Oleh karena itu, persiapan dan penanganan data sangat mempengaruhi hasil akhir dari model yang dikembangkan [9].

Tabel 1. Fitur-Fitur Dalam *Dataset*

No.	Nama Fitur	Deskripsi
1	*person_age*	Usia pemohon pinjaman
2	*person_income*	Jumlah pendapatan tahunan pemohon
3	*person_home_ ownership*	Status kepemilikan rumah (misalnya milik sendiri, sewa, atau cicilan)
4	*person_emp_length*	Lama masa kerja pemohon (dalam tahun)
5	*loan_intent*	Tujuan atau keperluan pengajuan pinjaman
6	*loan_grade*	Kategori/grade pinjaman berdasarkan tingkat risiko
7	*loan_amnt*	Jumlah pinjaman yang diajukan
8	*loan_int_rate*	Tingkat suku bunga pinjaman
9	*loan_percent_ income*	Persentase pinjaman terhadap pendapatan pemohon
10	*cb_person_default_ on_file*	Riwayat gagal bayar sebelumnya (ya/tidak)
11	*cb_preson_cred_hist _length*	Lama riwayat kredit yang dimiliki pemohon (dalam tahun)
12	*loan_status*	Status pinjaman (0 = tidak gagal bayar, 1 = gagal bayar)

DOI : https://doi.org/10.52158/krjtrh05

Pra pemrosesan data melibatkan berbagai langkah untuk mengubah data mentah menjadi bentuk yang lebih konsisten, lengkap, dan bersih dengan mengoreksi kesalahan maupun inkonsistensi [10]. Langkah awal pra pemrosesan data adalah penanganan *missing values*. Penanganan missing values merupakan proses penting dalam analisis data karena analisis tidak dapat dilakukan secara optimal dengan dataset yang tidak lengkap [11]. Terdapat beberapa pendekatan yang dapat digunakan, seperti penghapusan data atau imputasi. Jika data yang hilang cukup besar namun tidak mempengaruhi atribut penting, maka data tersebut dapat dihapus. Jika proporsinya kecil, teknik imputasi lebih disarankan. Imputasi dapat dilakukan dengan mengganti nilai yang hilang menggunakan mode, median, atau menggunakan model statistik. Dalam praktiknya, *median* dan *mode* lebih dipilih dibandingkan *mean* karena lebih tahan terhadap nilai ekstrim. Untuk variabel diskrit, imputasi biasanya dilakukan dengan *mode*, sedangkan untuk variabel kontinu, *median* digunakan [12]. Pemilihan metode penanganan *missing values* sangat bergantung pada konteks analisis, jumlah data yang hilang, dan akurasi yang diharapkan. Tahapan ini dilakukan untuk menjaga integritas dan validitas data sehingga hasil analisis yang dihasilkan tetap dapat diandalkan.

Langkah berikutnya adalah pengecekan data duplikat. Data duplikat terjadi ketika satu entitas direkam lebih dari satu kali. Kondisi ini dapat menyebabkan bias dalam analisis dan menurunkan kualitas model yang dikembangkan. Setiap data dalam dataset harus bersifat unik untuk menjaga kualitas data secara keseluruhan [13]. Penanganan data duplikat meliputi proses identifikasi dan penghapusan data yang tercatat lebih dari sekali. Identifikasi dilakukan menggunakan teknik data matching, yaitu membandingkan kesamaan antar variabel untuk mendeteksi duplikasi. Setelah data duplikat ditemukan, penghapusan dilakukan untuk memastikan hanya satu data yang valid yang tersisa. Proses ini penting untuk memastikan konsistensi data dan menghasilkan interpretasi yang lebih akurat dan tidak bias.

Langkah lain yang dapat dilakukan pada pra pemrosesan data adalah seleksi atribut atau seleksi fitur. Atribut yang tidak memiliki relevansi signifikan dapat berdampak negatif terhadap kinerja model klasifikasi. Penghapusan atribut dilakukan karena atribut tersebut dianggap tidak relevan dan berpotensi mengganggu proses klasifikasi [14].

2.5. Pengkodean Fitur Kategorikal

Pengkodean fitur kategorikal merupakan proses penting dalam analisis data yang melibatkan transformasi variabel kategorikal menjadi bentuk numerik agar dapat diproses oleh algoritma *machine learning*. Variabel kategorikal tidak dapat diproses secara langsung oleh algoritma *machine learning*. Oleh karena itu, diperlukan langkah transformasi untuk mengubah data kategorikal menjadi data numerik [15].

Terdapat beberapa teknik pengkodean yang umum digunakan, seperti *one-hot encoding*, *label encoding*, dan *ordinal encoding*. *One-hot encoding* mengubah nilai kategorikal menjadi angka biner sebagai langkah penting bagi algoritma *machine learning*. Karena algoritma *machine learning* tidak dapat secara langsung memproses data kategorikal, maka proses *one-hot encoding* menjadi sangat penting untuk mengkonversi nilai tersebut ke dalam format yang sesuai untuk analisis numerik [16]. *Label encoding* menggantikan nilai kategori dengan nilai bilangan bulat unik. Sementara itu, *ordinal encoding* memberikan nilai bilangan bulat berurutan sesuai dengan urutan tingkat atau hierarki pada variabel kategorikal [17]. Pemilihan teknik pengkodean yang tepat sangat penting untuk memastikan bahwa data kategorikal dapat diproses dengan benar sehingga menghasilkan analisis dan model *machine learning* yang akurat.

2.6. Penanganan Data Tidak Seimbang

Banyak solusi telah diajukan untuk mengatasi masalah ketidakseimbangan kelas, baik pada algoritma klasifikasi standar maupun algoritma *ensemble*. Salah satu pendekatan adalah pada tingkat data, yaitu dengan menyesuaikan sampel pelatihan agar distribusi kelas lebih seimbang sehingga algoritma dapat bekerja lebih optimal [7]. Data yang tidak seimbang dapat menyebabkan model bias terhadap kelas mayoritas dan menurunkan kinerja dalam memprediksi kelas minoritas.

Banyak solusi telah diajukan untuk mengatasi masalah ketidakseimbangan kelas, baik pada algoritma klasifikasi standar maupun algoritma *ensemble*. Pendekatan pada tingkat data dilakukan dengan menyesuaikan sampel data pelatihan agar distribusi kelas menjadi lebih seimbang, sehingga algoritma dapat bekerja secara lebih optimal. Strategi ini umumnya melibatkan dua metode, yaitu *oversampling* dan *undersampling*. Teknik *oversampling* mengatasi ketidakseimbangan kelas dengan membuat sampel data baru dari kelas minoritas. Sebaliknya, prosedur *undersampling* menangani ketidakseimbangan kelas dengan mengurangi jumlah sampel data dari kelas mayoritas [18]. Penanganan data yang tidak seimbang secara tepat sangat penting dalam membangun model prediktif yang kuat dan akurat di berbagai bidang.

2.7. *Decision Tree*

Decision Tree merupakan salah satu jenis algoritma *supervised learning*, di mana pengguna dapat menentukan *input* serta *output* yang sesuai pada *dataset* pelatihan. *Decision Tree* menawarkan teknik yang kuat untuk klasifikasi dan prediksi, serta dapat bekerja dengan baik pada variabel *input* maupun *output* yang bersifat kategorikal maupun kontinu [19]. Algoritma ini

DOI : https://doi.org/10.52158/krjtrh05

beroperasi dengan membagi ruang fitur ke dalam struktur pohon yang terdiri dari *decision node* dan *leaf node*, di mana setiap *internal node* merepresentasikan keputusan berdasarkan nilai fitur, sedangkan setiap *leaf node* menunjukkan hasil prediksi atau label kelas [20].

Decision Tree merupakan sebuah diagram alir yang terdiri atas dua entitas, yaitu *decision node* dan *leaf*. *Decision node* mencakup simpul *non-leaf* yang melakukan pengujian kondisi pada parameter tertentu, sedangkan *leaf* merepresentasikan label kelas sebagai hasil prediksi atau klasifikasi. Setiap jalur dari *root* menuju *leaf* menunjukkan aturan keputusan yang digunakan untuk memperoleh hasil akhir [19]. Proses pengambilan keputusan dilakukan melalui serangkaian keputusan biner pada cabang-cabang pohon hingga mencapai *leaf node*, tempat prediksi akhir ditentukan. *Decision Tree* dikenal memiliki tingkat interpretabilitas yang tinggi, karena mampu memvisualisasikan proses pengambilan keputusan serta mengidentifikasi fitur paling penting untuk klasifikasi atau prediksi [21].

2.8. *Random Forest*

Random Forest merupakan salah satu teknik dalam *machine learning* yang banyak digunakan untuk klasifikasi dan regresi [6]. Model *Random Forest* dibangun menggunakan *Decision Tree* yang dibentuk dari berbagai subsampel data. *Random Forest* merupakan algoritma *ensemble* yang menggabungkan banyak *Decision Tree* dan menerapkan teknik bagging untuk mengurangi terjadinya *overfitting* [22]. Teknik ini melibatkan pelatihan setiap *Decision Tree* dengan menggunakan subsampel data yang berbeda, di mana proses pengambilan sampel dilakukan dengan pengembalian [23]. Pendekatan ini memungkinkan *Random Forest* untuk mengatasi masalah *overfitting* serta menghasilkan model yang lebih stabil dan umumnya memiliki kinerja baik pada berbagai jenis data. Dengan demikian, proses ini menghasilkan model *Decision Tree* yang secara umum lebih baik [19].

2.9. *GridSearchCV*

Model *machine learning* memerlukan proses penyesuaian *hyperparameter* [24]. Hal ini dapat dilakukan secara efektif menggunakan *grid search*, yaitu metode yang secara sistematis mengeksplorasi berbagai nilai *hyperparameter* untuk menemukan kombinasi yang optimal. Metode optimasi *GridSearchCV* merupakan salah satu pendekatan yang umum digunakan dalam menentukan parameter terbaik pada model *machine learning*. *GridSearchCV* menjadi alat yang penting dalam pengembangan model karena mempertimbangkan semua kemungkinan kombinasi parameter ketika melakukan pelatihan data. *GridSearchCV* menyediakan fungsi bawaan yang melakukan *cross-validation* pada data uji, sehingga meningkatkan akurasi dan presisi dalam mengevaluasi kinerja model. Dengan demikian, *GridSearchCV* menawarkan pendekatan yang lebih sistematis dan andal dalam pemilihan parameter serta evaluasi model [25].

3. Hasil dan Pembahasan

Penelitian ini dimulai dengan tahap pra pemrosesan data. Pada tahap pra pemrosesan data dilakukan beberapa langkah, antara lain penanganan *missing values*, penanganan duplikasi data, serta *encoding* fitur kategorikal. Pada tahap penanganan *missing values*, dilakukan pemeriksaan terhadap nilai NaN atau data kosong. Dari hasil pemeriksaan, ditemukan beberapa nilai yang hilang pada fitur tertentu sebagaimana ditunjukkan pada Tabel 2. Data pada fitur yang memiliki *missing values* tersebut kemudian dihapus, sehingga jumlah total data yang tersisa adalah 28.638 *record*.

Tabel 2.Jumlah *Missing values* pada *Dataset*

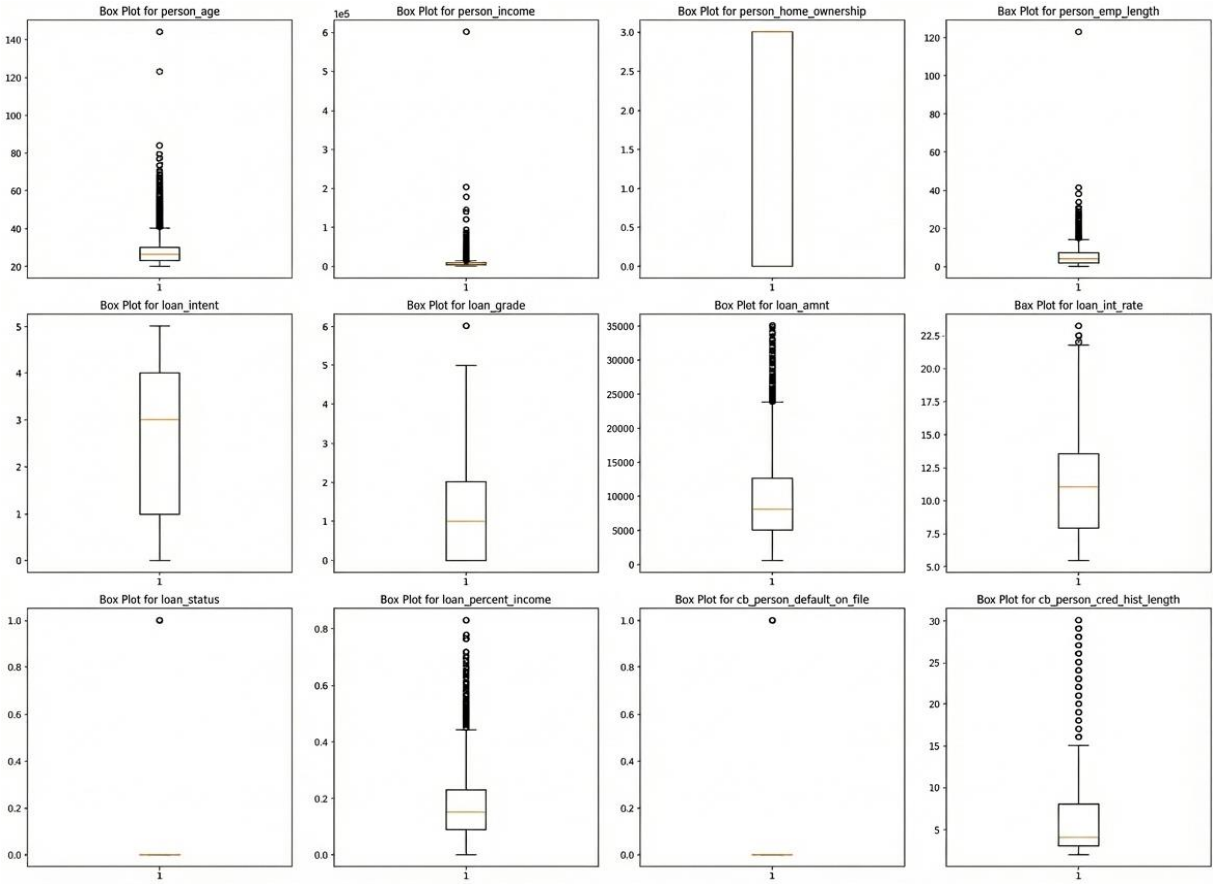
No.	Nama Fitur	Deskripsi
1	*person_age*	0
2	*person_income*	0
3	*person_home_ ownership*	0
4	*person_emp_length*	895
5	*loan_intent*	0
6	*loan_grade*	0
7	*loan_amnt*	0
8	*loan_int_rate*	3116
9	*loan_percent_ income*	0
10	*cb_person_default_ on_file*	0
11	*cb_preson_cred_hist _length*	0
12	*loan_status*	0

Pada tahap penanganan data duplikat, dilakukan pemeriksaan menyeluruh terhadap data untuk mengidentifikasi *record* yang sama. Proses ini melibatkan pengecekan setiap *record* guna menemukan entri yang benar-benar identik. Setelah proses identifikasi selesai, ditemukan sebanyak 137 *record* duplikat. *Record* tersebut kemudian dihapus untuk menjaga integritas data serta menghindari bias atau distorsi dalam analisis selanjutnya. Dengan dihapusnya data duplikat tersebut, jumlah data berkurang menjadi 28.501 *record* dari jumlah awal. Penanganan data duplikat merupakan langkah penting dalam proses pra pemrosesan data untuk memastikan bahwa analisis berikutnya didasarkan pada data yang akurat dan representatif. Tahap berikutnya dalam pra pemrosesan data adalah melakukan *label encoding* pada beberapa atribut yang masih bersifat kategorikal. *Label encoding* berfungsi untuk mengubah nilai kategorikal menjadi nilai numerik, sehingga dapat digunakan dalam model *machine learning* yang umumnya memerlukan input berbentuk numerik. Pada *dataset* ini, *label encoding* diterapkan pada beberapa fitur utama, yaitu *loan_intent*, *person_home_ownership*, *loan_grade*, dan *cb_person_default_on_file*.

Setelah proses *label encoding* selesai dilakukan, setiap kategori dalam fitur-fitur tersebut diubah menjadi

DOI : https://doi.org/10.52158/krjtrh05

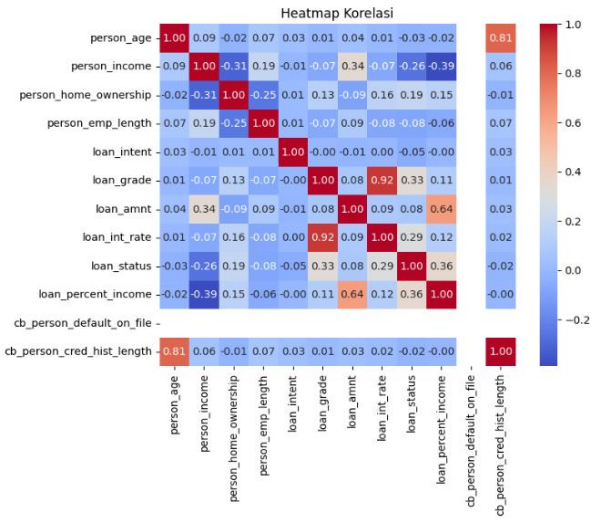
representasi numerik yang unik. Hal ini memungkinkan model *machine learning* untuk memproses dan memahami hubungan antar kategori secara lebih efektif. Setelah semua atribut kategorikal berhasil diubah menjadi nilai numerik, tahap selanjutnya adalah melakukan deteksi *outlier* pada *dataset*. Data yang teridentifikasi sebagai *outlier* kemudian dihapus untuk menjaga kualitas dan integritas *dataset*. Visualisasi *boxplot* dari hasil pemeriksaan *outlier* pada *dataset* ditampilkan pada Gambar 2.



Gambar 2. Visualisasi *boxplot outlier*

Setelah proses *label encoding* selesai dilakukan, setiap kategori dalam fitur-fitur tersebut diubah menjadi representasi numerik yang unik. Hal ini memungkinkan model *machine learning* untuk memproses dan memahami hubungan antar kategori secara lebih efektif. Setelah semua atribut kategorikal berhasil diubah menjadi nilai numerik, tahap selanjutnya adalah melakukan deteksi *outlier* pada *dataset*. Data yang teridentifikasi sebagai *outlier* kemudian dihapus untuk menjaga kualitas dan integritas *dataset*. Visualisasi *boxplot* dari hasil pemeriksaan *outlier* pada *dataset* ditampilkan pada Gambar 2.

Setelah tahap penghapusan *outlier* selesai dilakukan, tahap selanjutnya adalah melakukan pemeriksaan korelasi antar fitur. Pemeriksaan ini penting untuk mengidentifikasi hubungan linier di antara berbagai fitur dalam *dataset*. Dengan memahami korelasi antar fitur, peneliti dapat menentukan fitur-fitur yang memiliki keterkaitan erat, yang dapat mempengaruhi hasil analisis maupun model prediktif. Tahap ini merupakan bagian penting dari proses data *pra pemrosesan* untuk memastikan bahwa analisis dan model yang dibangun memiliki dasar yang kuat dan akurat. Matriks korelasi untuk setiap fitur ditunjukkan pada Gambar 3.



Gambar 3. Matriks Korelasi

DOI : https://doi.org/10.52158/krjtrh05

Berdasarkan Gambar 3, terlihat bahwa fitur *cb_person_default_on_file* tidak memiliki korelasi dengan fitur lainnya. Oleh karena itu, fitur *cb_person_default_on_file* dihapus dan tidak digunakan dalam proses pengolahan data. Selain itu, penghapusan fitur *loan_int_rate* dari analisis *dataset* juga dilakukan karena fitur ini memiliki korelasi yang sangat tinggi dengan fitur *loan_grade*. Korelasi yang tinggi antara dua fitur menunjukkan bahwa keduanya menyampaikan informasi yang hampir sama, dan menyertakan keduanya dalam model dapat menyebabkan redundansi data. Lebih jauh lagi, keberadaan dua fitur dengan korelasi tinggi dalam model prediktif dapat menimbulkan masalah multikolinearitas yang berdampak negatif terhadap stabilitas dan interpretabilitas model. Oleh karena itu, untuk menyederhanakan model dan meningkatkan kinerja analisis, fitur *loan_int_rate* dihapus, dengan hanya mempertahankan fitur *loan_grade* sebagai representasi informasi terkait tingkat bunga pinjaman.

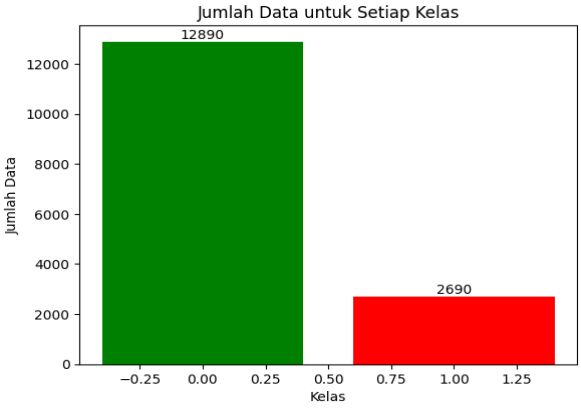
Setelah dilakukan analisis korelasi antar fitur, tahap selanjutnya adalah penanganan data yang tidak seimbang. Pertama, data dibagi menjadi data latih dan data uji dengan rasio 50:50, 60:40, 70:30, dan 80:20. Selanjutnya dilakukan pengukuran akurasi untuk masing-masing rasio pembagian data tersebut dengan menggunakan metode yang masih menerapkan parameter *default*. Hasil pengukuran akurasi ditunjukkan pada Tabel 3.

Tabel 3.Akurasi Berdasarkan Rasio Split Data

Rasio Data Latih : Data Uji	Metode	Akurasi (%)
50:50	*Decision Tree*	88,57
	Random Forest	93,32
60:40	*Decision Tree*	88,92
	Random Forest	93,40
70:30	*Decision Tree*	88,82
	Random Forest	93,75
80:20	*Decision Tree*	88,75
	Random Forest	93,99

Tabel 3 menunjukkan bahwa rasio terbaik adalah 80:20 karena akurasi tertinggi diperoleh dengan menggunakan metode *Random Forest*. Selanjutnya dilakukan pemeriksaan ketidakseimbangan data pada data latih. Ketidakseimbangan data terjadi ketika jumlah sampel pada setiap kelas atau kategori tidak merata. Hal ini dapat menjadi masalah serius dalam analisis data karena model yang dibangun cenderung bias terhadap kelas mayoritas dan mengabaikan kelas minoritas yang mungkin mengandung informasi penting. Rasio jumlah data untuk setiap kelas ditunjukkan pada Gambar 4.

Penelitian ini menerapkan proses penyeimbangan data dengan menggunakan beberapa teknik, antara lain *Random Oversampling*, *Random Under Sampling*, *Adaptive Synthetic Sampling*, dan *Synthetic Minority Over-sampling Technique* (SMOTE). Tujuan dari teknik penyeimbangan data adalah untuk mengatasi ketidakseimbangan kelas dalam *dataset*, di mana salah satu kelas mungkin memiliki jumlah *instance* yang jauh lebih sedikit dibandingkan dengan kelas lainnya. Dengan menerapkan teknik seperti *oversampling* atau *undersampling*, diharapkan model *machine learning* tidak bias terhadap kelas mayoritas serta mampu mengenali pola dari kelas minoritas dengan lebih baik. Hasil pengukuran akurasi menggunakan berbagai teknik penyeimbangan data tersebut disajikan pada Tabel 4



Gambar 4. Perbandingan Jumlah Data

Tabel 4. Akurasi Berdasarkan Teknik Penyeimbangan Data

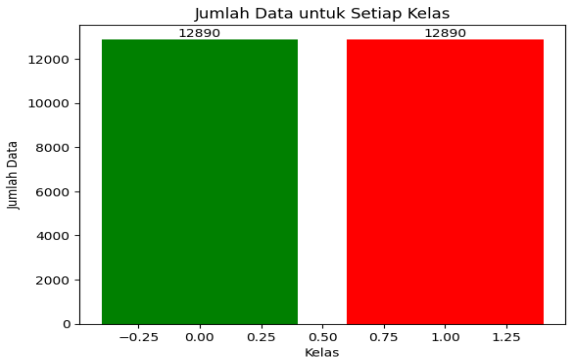
Teknik Penyeimbangan Data	*Decision Tree* (%)	*Random Forest* (%)
Random Under Sampling	77,90	87,37
Adaptive Synthetic Sampling	85,42	90,52
Synthetic Minority Over-sampling Technique	86,21	91,76
Random Oversampling	89,09	93,91

Berdasarkan Tabel 4, metode *Random Oversampling* digunakan untuk menangani ketidakseimbangan data. Pemilihan metode ini didasarkan pada capaian akurasi yang cukup tinggi, yaitu sebesar 89,09% pada algoritma *Decision Tree* dan 93,91% pada algoritma *Random Forest*. *Random Oversampling* merupakan teknik dengan menambahkan sejumlah sampel pada kelas minoritas melalui proses pengambilan secara acak dari kelas tersebut, kemudian mengembalikannya ke dalam *dataset* hingga jumlah sampel pada kelas minoritas seimbang dengan kelas mayoritas. Perbandingan jumlah target setelah penerapan *Random Oversampling* ditunjukkan pada Gambar 5.

Setelah melalui tahap penyeimbangan data guna mengatasi permasalahan ketidakseimbangan kelas pada *dataset*, langkah selanjutnya adalah melakukan klasifikasi dengan menggunakan dua metode yang umum digunakan, yaitu *Decision Tree* dan *Random Forest*. Pada tahap awal, parameter kedua metode ditetapkan dalam kondisi *default*, kemudian dilakukan pelatihan model dengan *dataset* yang telah disesuaikan.

DOI : https://doi.org/10.52158/krjtrh05

Setelah proses pelatihan selesai, hasil akurasi dari masing-masing metode dicatat dan disajikan pada Tabel 5. Tahap ini bertujuan untuk memberikan pemahaman yang lebih mendalam mengenai kinerja model klasifikasi setelah penyeimbangan data, sekaligus membandingkan efektivitas dua metode klasifikasi dalam menangani permasalahan ketidakseimbangan kelas pada *dataset*.



Gambar 5. Jumlah Data Setelah Menggunakan *Random Oversampling*

Tabel 5.Hasil Klasifikasi

Metode	Akurasi	*F1-Score*
Decision Tree	89,09 %	69,13 %
Random Forest	93,91 %	79,58 %

Tahap selanjutnya adalah melakukan penyetelan parameter pada masing-masing metode dengan menggunakan *GridSearchCV*. Hasil pengujian menunjukkan bahwa akurasi *Random Forest* lebih tinggi dibandingkan dengan *Decision Tree*. Peningkatan pada akurasi mempengaruhi peningkatan *F1-Score*. Nilai akurasi dan *F1-Score* dari masing-masing metode setelah dilakukan penyetelan parameter dengan *GridSearchCV* ditampilkan pada Tabel 6.

Tabel 6.Hasil Klasifikasi Menggunakan *GridSearchCV*

Metode	Akurasi	*F1-Score*
Decision Tree	89,21 %	70,17 %
Random Forest	94,04 %	79,75 %

Peningkatan akurasi dan *F1-Score* pada model *Random Forest* dibandingkan dengan *Decision Tree* dapat dijelaskan oleh beberapa faktor utama yang berkaitan dengan karakteristik serta optimasi model. *Random Forest* merupakan metode *ensemble* yang menggabungkan prediksi dari sejumlah *Decision Tree* yang dibangun secara independen. Dengan mengagregasi hasil dari banyak pohon, *Random Forest* mampu mengurangi varians dan bias, sehingga menghasilkan prediksi yang lebih akurat dibandingkan *Decision Tree* tunggal yang cenderung mengalami *overfitting* terhadap data latih. Selain itu, penggunaan *GridSearchCV* untuk penyetelan parameter memungkinkan optimasi *hyperparameter* seperti jumlah pohon (*n_estimators*) dan kedalaman maksimum pohon (*max_depth*), yang berkontribusi pada peningkatan kinerja model. *GridSearchCV* juga memanfaatkan *cross-validation* dalam mengevaluasi setiap kombinasi parameter, sehingga menghasilkan estimasi kinerja yang lebih andal serta mengurangi risiko *overfitting*. Lebih lanjut, *Random Forest* memiliki kemampuan yang lebih baik dalam menangani ketidakseimbangan data melalui penyesuaian bobot kelas secara otomatis, sehingga kelas minoritas tetap mendapatkan perhatian yang memadai pada saat pelatihan model. Dengan demikian, baik akurasi maupun *F1-Score* mengalami peningkatan, yang pada akhirnya menghasilkan prediksi yang lebih konsisten dan akurat.

4. Kesimpulan

Berdasarkan hasil penelitian, penerapan metode *Random Oversampling* untuk menyeimbangkan data serta penyetelan parameter menggunakan *GridSearchCV* terbukti secara signifikan meningkatkan kinerja model klasifikasi. Metode *Decision Tree* menunjukkan peningkatan akurasi hingga 89,21% dan *F1-Score* sebesar 70,17%, sedangkan metode *Random Forest* mencapai akurasi 94,04% dengan *F1-Score* sebesar 79,75%. Penerapan *Random Oversampling* efektif dalam mengatasi ketidakseimbangan data yang sering menyebabkan bias terhadap kelas mayoritas, sehingga mampu meningkatkan sensitivitas model terhadap seluruh kelas. Sementara itu, penyetelan parameter melalui *GridSearchCV* memungkinkan diperolehnya kombinasi parameter yang optimal berdasarkan kinerja pada data validasi, sehingga mendorong peningkatan performa model secara keseluruhan. Dengan demikian, kombinasi strategi penyeimbangan data dan optimasi parameter ini terbukti efektif serta dapat diaplikasikan pada berbagai metode klasifikasi untuk meningkatkan akurasi dan *F1-Score* prediksi.

Daftar Rujukan

[1] Y. Dasari, K. Rishitha, and O. Gandhi, "Prediction of Bank Loan Status Using Machine Learning Algorithms," *Int. J. Comput. Digit. Syst.*, vol. 14, no. 1, 2023, doi: 10.12785/ijcds/140113.

[2] S. M. Fati, "Machine Learning-Based Prediction Model for Loan Status Approval," *J. Hunan Univ. Nat. Sci.*, vol. 48, no. 10, 2021.

[3] K. Gautam, A. P. Singh, K. Tyagi, and M. Suresh Kumar, "Loan Prediction using Decision Tree and Random Forest," *Int. Res. J. Eng. Technol.*, 2020.

[4] A. C. B. Garcia, M. G. P. Garcia, and R. Rigobon, "Algorithmic discrimination in the credit domain: what do we know about it?," *AI Soc.*, 2023, doi: 10.1007/s00146-023-01676-3.

[5] N. Uddin, M. K. Uddin Ahamed, M. A. Uddin, M. M. Islam, M. A. Talukder, and S. Aryal, "An ensemble machine learning based bank loan approval predictions system with a smart application," *Int. J. Cogn. Comput. Eng.*, vol. 4, 2023, doi: 10.1016/j.ijcce.2023.09.001.

[6] L. Sathish kumar, V. Pandimurugan, D. Usha, M. Nageswara Guptha, and M. S. Hema, "Random forest tree classification algorithm for predicating loan," *Mater. Today Proc.*, vol. 57, pp. 2216–2222, Jan. 2022, doi: 10.1016/j.matpr.2021.12.322.

[7] M. Khushi *et al.*, "A Comparative Performance Analysis of Data Resampling Methods on Imbalance Medical Data," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3102399.

[8] D. Ismunandar, M. R. Firdaus, and Y. Alkhalifi, "Penerapan Hyperparameter Machine Learning Dalam Prediksi Gagal

DOI : https://doi.org/10.52158/krjtrh05

Pinjam," *INTI Nusa Mandiri*, vol. 19, no. 1, pp. 62–70, 2024, doi: 10.33480/inti.v19i1.5612.

[9] K. Mallikharjuna Rao, G. Saikrishna, and K. Supriya, "Data preprocessing techniques: emergence and selection towards machine learning models - a practical review using HPA dataset," *Multimed. Tools Appl.*, vol. 82, no. 24, 2023, doi: 10.1007/s11042-023-15087-5.

[10] A. Al-Qerem, G. Al-Naymat, M. Alhasan, and M. Al-Debei, "Default prediction model: The significant role of data engineering in the quality of outcomes," *Int. Arab J. Inf. Technol.*, vol. 17, no. 4 Special Issue, 2020, doi: 10.34028/iajit/17/4A/8.

[11] A. R. Ismail, N. Z. Abidin, and M. K. Maen, "Systematic Review on Missing Data Imputation Techniques with Machine Learning Algorithms for Healthcare," *Journal of Robotics and Control (JRC)*, vol. 3, no. 2. 2022. doi: 10.18196/jrc.v3i2.13133.

[12] Z. Wu, "Using Machine Learning Approach to Evaluate the Excessive Financialization Risks of Trading Enterprises," *Comput. Econ.*, vol. 59, no. 4, 2022, doi: 10.1007/s10614-020-10090-6.

[13] A. Perwitasari, R. Septiriana, and T. Tursina, "Data preparation Structure untuk Pemodelan Prediktif Jumlah Peserta Ajar Matakuliah," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 1, p. 7, 2023, doi: 10.26418/jp.v8i3.57321.

[14] J. C. Alejandrino, J. P. Bolacoy, and J. V. B. Murcia, "Supervised and unsupervised data mining approaches in loan default prediction," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 2, 2023, doi: 10.11591/ijece.v13i2.pp1837-1847.

[15] J. Jemai and A. Zarrad, "Feature Selection Engineering for Credit Risk Assessment in Retail Banking," *Inf.*, vol. 14, no. 3, 2023, doi: 10.3390/info14030200.

[16] A. Y. Hussein, P. Falcarin, and A. T. Sadiq, "Enhancement performance of random forest algorithm via one hot encoding for IoT IDS," *Period. Eng. Nat. Sci.*, vol. 9, no. 3, 2021, doi: 10.21533/pen.v9i3.2204.

[17] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3104357.

[18] M. Z. Abedin, C. Guotai, P. Hajek, and T. Zhang, "Combining weighted SMOTE with ensemble learning for the class-imbalanced prediction of small business credit risk," *Complex Intell. Syst.*, vol. 9, no. 4, 2023, doi: 10.1007/s40747-021-00614-4.

[19] S. Hou, Z. Cai, J. Wu, H. Du, and P. Xie, "Applying Machine Learning to the Development of Prediction Models for Bank Deposit Subscription," *Int. J. Bus. Anal.*, vol. 9, no. 1, 2021, doi: 10.4018/ijban.288514.

[20] X. Li, S. Yi, A. B. Cundy, and W. Chen, "Sustainable decision-making for contaminated site risk management: A decision tree model using machine learning algorithms," *J. Clean. Prod.*, vol. 371, 2022, doi: 10.1016/j.jclepro.2022.133612.

[21] I. I. Febriansyah, R. Sarno, and R. N. Anggraini, "Decision Tree and Fuzzy Logic in The Audit of Information System for Tax Letter Issuance," in *IES 2022 - 2022 International Electronics Symposium: Energy Development for Climate Change Solution and Clean Energy Transition, Proceeding*, 2022. doi: 10.1109/IES55876.2022.9888372.

[22] N. Darapaneni *et al.*, "Tree Based Models: A Comparative and Explainable Study for Credit Default Classification," in *9th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering, UPCON 2022*, 2022. doi: 10.1109/UPCON56432.2022.9986411.

[23] M. R. Machado and S. Karray, "Assessing credit risk of commercial customers using hybrid machine learning algorithms," *Expert Syst. Appl.*, vol. 200, 2022, doi: 10.1016/j.eswa.2022.116889.

[24] N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization," *J. Inf. Secur. Appl.*, vol. 55, 2020, doi: 10.1016/j.jisa.2020.102596.

[25] M. E. Lokanan and K. Sharma, "Fraud prediction using machine learning: The case of investment advisors in Canada," *Mach. Learn. with Appl.*, vol. 8, 2022, doi: 10.1016/j.mlwa.2022.100269.

DOI : https://doi.org/10.52158/krjtrh05