

KLASIFIKASI PERSETUJUAN PINJAMAN MENGGUNAKAN ENSEMBLE SMOTEENN DAN BALANCED BAGGING BERBASIS DECISION TREE PADA LOAN RISK PREDICTION DATASET

Rosa Nur Aliana Sawafi^{*1}, Ika Wida Nuragusti²

^{1,2}Program Studi Sains Data, Universitas Telkom, Indonesia

Email: ¹alinxyz@student.telkomuniversity.ac.id, ²ikawidanuragustin@student.telkomuniversity.ac.id

Abstrak

Klasifikasi persetujuan pinjaman pada sistem credit scoring merupakan permasalahan penting dalam industri keuangan, terutama pada kondisi data tidak seimbang (*imbalanced dataset*) yang menyebabkan model cenderung bias terhadap kelas mayoritas sehingga menurunkan kemampuan mendeteksi calon peminjam berisiko. Penelitian terdahulu umumnya menerapkan teknik *resampling* atau *ensemble learning* secara terpisah, sedangkan integrasi keduanya pada *loan risk prediction dataset* masih jarang dikaji. Penelitian ini mengusulkan kombinasi *hybrid resampling* SMOTEENN dan *Balanced Bagging Classifier* berbasis *Decision Tree* yang menggabungkan penyeimbangan pada tingkat data dan algoritma dalam satu kerangka kerja. Kontribusi utama penelitian dibuktikan melalui tujuh skenario *ablation* yang menunjukkan bahwa kedua pendekatan saling melengkapi dan menghasilkan performa lebih stabil dibandingkan penerapan parsial maupun model tunggal. Evaluasi menggunakan metrik *Accuracy*, *Precision*, *Recall*, *Specificity*, *F1-score*, *G-Mean*, dan *AUC-ROC*, dengan *F1-score* sebagai indikator utama karena lebih representatif untuk data tidak seimbang. Hasil eksperimen menunjukkan bahwa kombinasi SMOTEENN dan *Balanced Bagging Classifier* mencapai *F1-score* tertinggi sebesar 0,9054 dengan *precision* 0,9393, serta kemampuan paling stabil dalam menekan kesalahan klasifikasi pada kedua kelas. Temuan ini menunjukkan bahwa integrasi teknik *resampling* dan *ensemble learning* efektif meningkatkan performa klasifikasi pada data persetujuan pinjaman yang tidak seimbang.

Kata kunci: *Balanced Bagging, credit scoring, ensemble learning, imbalanced data, SMOTEENN.*

Classification of Loan Approval Using the SMOTEENN Ensemble and Balanced Bagging Based on Decision Trees on the Loan Risk Prediction Dataset

Abstract

Loan approval classification in credit scoring is a critical task in the financial industry, particularly when dealing with imbalanced datasets, where predictive models tend to be biased toward the majority class and perform poorly in identifying high-risk borrowers. Previous studies have generally applied resampling or ensemble learning techniques separately, while their integration for loan risk prediction remains relatively underexplored. This study proposes a combination of SMOTEENN hybrid resampling and a Decision Tree-based Balanced Bagging Classifier, integrating data-level balancing and algorithm-level variance reduction within a unified framework. The main contribution is demonstrated through seven ablation scenarios, showing that both approaches complement each other and provide more stable performance than partial implementations or single models. The proposed approach was evaluated using Accuracy, Precision, Recall, Specificity, F1-score, G-Mean, and AUC-ROC, with the F1-score selected as the primary evaluation metric because it better reflects the balance between precision and recall in imbalanced classification problems. Experimental results show that the proposed combination achieved the highest F1-score of 0.9054 with a precision of 0.9393, while consistently reducing misclassification errors across both classes. These findings demonstrate that integrating resampling and ensemble learning effectively improves classification performance for imbalanced loan approval data.

Keywords: *Balanced Bagging, credit scoring, ensemble learning, imbalanced data, SMOTEENN.*

1. PENDAHULUAN

Klasifikasi persetujuan pinjaman merupakan komponen penting dalam sistem *credit scoring* karena berkaitan langsung dengan stabilitas dan profitabilitas lembaga keuangan [1]. Kesalahan keputusan memiliki dua sisi risiko yaitu menyetujui debitur berisiko dapat meningkatkan kredit macet, sedangkan menolak debitur yang layak mengurangi peluang keuntungan. Tantangan ini diperberat oleh kondisi data yang tidak seimbang atau *imbalanced dataset*, di mana jumlah debitur lancar umumnya jauh lebih besar daripada debitur gagal bayar [2]. Distribusi kelas yang timpang menyebabkan model cenderung bias terhadap kelas mayoritas sehingga kemampuannya mendeteksi nasabah berisiko menurun, padahal kesalahan pada kelas minoritas justru menimbulkan kerugian finansial yang paling besar [3]. Secara umum, teknik penanganan data tidak seimbang dapat dikelompokkan menjadi pendekatan tingkat data, tingkat algoritma, dan *hybrid* [4]. Oleh karena itu, dibutuhkan pendekatan klasifikasi yang tidak hanya akurat secara agregat, tetapi juga seimbang dalam mengenali kedua kelas.

Secara metodologis, pendekatan *credit scoring* telah bergeser dari model statistik tradisional seperti regresi, yang unggul dalam interpretabilitas namun terbatas dalam menangkap hubungan non-linear [5], menuju metode *machine learning* berbasis pohon keputusan dan *ensemble* yang lebih fleksibel [6]. Model *ensemble* berbasis pohon terbukti memiliki performa baik pada prediksi *default* berdimensi tinggi [6], sehingga pendekatan *ensemble* menjadi arah yang menjanjikan, bahkan berkembang menuju pemanfaatan pendekatan mutakhir seperti *language model* untuk penilaian risiko kredit secara otomatis [7]. Pada konteks SVM, metode yang memadukan *resampling* dan pendekatan algoritmik umumnya memberikan performa terbaik pada data tidak seimbang [8]. Namun, keunggulan pada tingkat algoritma saja belum cukup apabila ketidakseimbangan kelas tidak ditangani secara eksplisit. Salah satu teknik yang banyak digunakan adalah *Synthetic Minority Oversampling Technique* (SMOTE), yang terbukti meningkatkan kemampuan deteksi kasus *default* pada masalah pinjaman tidak seimbang [9]. Meskipun demikian, SMOTE secara tunggal berpotensi menimbulkan *noise* dan tumpang tindih antar kelas akibat data sintesis yang dihasilkan, sehingga peningkatan performanya tidak selalu konsisten pada seluruh metrik.

Untuk mengatasi keterbatasan tersebut, berbagai penelitian mengembangkan pendekatan *hybrid* yang menggabungkan *resampling* dan *ensemble learning*. Song et al. (2023) menunjukkan bahwa permasalahan utama klasifikasi risiko kredit bukan sekadar pencapaian akurasi tinggi, melainkan kemampuan mendeteksi kelas minoritas seperti pendekatan *rating-specific multi-objective ensemble learning* yang mereka usulkan menghasilkan *sensitivity*, *G-mean*, dan *F1-score* lebih baik dibandingkan sejumlah metode pembandingan [10]. Komponen-komponen penanganan *imbalance*, baik pada tingkat data maupun algoritma, juga telah terbukti efektif secara terpisah pada domain kredit [2][11][12]. Meskipun demikian, integrasi spesifik antara SMOTEENN yang menggabungkan *oversampling* SMOTE dengan pembersihan *noise* melalui *Edited Nearest Neighbors* dan *Balanced Bagging Classifier* berbasis *Decision Tree*, yang menyeimbangkan kelas pada setiap *bootstrap sample* sekaligus menstabilkan variansi model, masih jarang dikaji secara khusus pada *loan risk prediction dataset*. Penelitian terdahulu cenderung menerapkan teknik *resampling* atau *ensemble* secara parsial, sehingga potensi keterpaduan antara pembersihan data dan stabilisasi *ensemble* dalam satu kerangka kerja yang utuh belum banyak dieksplorasi. Kesenjangan inilah yang menjadi fokus penelitian ini.

Penelitian ini menggunakan *Loan Risk Prediction Dataset* dari platform Kaggle karena karakteristiknya representatif terhadap permasalahan yang dikaji. Dataset ini memuat ketidakseimbangan kelas yang nyata dengan rasio 3,34:1 serta kombinasi fitur numerik dan kategorikal yang mencerminkan variabel penilaian kredit di dunia nyata, sehingga sesuai untuk menguji efektivitas penanganan data tidak seimbang. Berdasarkan uraian tersebut, tujuan penelitian ini adalah yaitu menerapkan kombinasi SMOTEENN dan *Balanced Bagging Classifier* berbasis *Decision Tree* untuk klasifikasi persetujuan pinjaman pada data tidak seimbang, kemudian menganalisis kontribusi tiap komponen, yaitu *resampling*, *ensemble*, dan penyeimbangan kelas, melalui rangkaian skenario eksperimen dan mengevaluasi performa model secara komprehensif menggunakan metrik yang sensitif terhadap ketidakseimbangan kelas, yaitu *Accuracy*, *Precision*, *Recall*, *Specificity*, *F1-score*, *G-Mean*, dan *AUC-ROC*, guna memperoleh model yang seimbang dalam menekan kesalahan pada kedua kelas.

2. METODE PENELITIAN

2.1. Dataset

Data yang digunakan adalah dataset *Loan Risk Prediction* dari platform Kaggle untuk klasifikasi biner status persetujuan pinjaman, yaitu disetujui (*Loan Approved* = 1) atau tidak disetujui = 0. Dataset terdiri dari 5.000 observasi dengan 10 variabel, mencakup 9 variabel independen dan 1 variabel dependen. Variabel fitur meliputi *Age*, *Income*, *LoanAmount*, *CreditScore*, *YearsExperience*, *Gender*, *Education*, *City*, dan *EmploymentType*. Variabel target adalah *LoanApproved* dengan dua kelas (0 = *Not Approved*; 1 = *Approved*).

Berdasarkan analisis awal, distribusi kelas menunjukkan ketidakseimbangan dengan rasio 3,34:1, di mana kelas 0 lebih dominan.

2.2. Eksplorasi dan Persiapan Data

Tahap awal dilakukan melalui *Exploratory Data Analysis* (EDA) meliputi pemeriksaan struktur data, statistik deskriptif, distribusi kelas, distribusi fitur numerik, serta deteksi *missing value* dan *outlier*. Penanganan *missing value* dilakukan dengan imputasi nilai median pada variabel numerik (*Income* dan *CreditScore*) dan modus pada variabel kategorikal (*Education*). Seluruh variabel kategorikal dikonversi ke bentuk numerik menggunakan *label encoding* manual. Dataset dibagi menjadi data latih (60%), validasi (20%), dan uji (20%) menggunakan *stratified split*. Seluruh fitur numerik distandarisasi menggunakan *StandardScaler* dengan *fitting* hanya pada data latih untuk menghindari *data leakage*, sebagaimana Persamaan (1).

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

2.3. Penanganan Data Tidak Seimbang dengan SMOTEENN

Penanganan ketidakseimbangan kelas menggunakan metode *hybrid sampling* SMOTEENN yang hanya diterapkan pada data latih untuk menghindari *data leakage*. SMOTE membangkitkan sampel sintetis pada kelas minoritas melalui interpolasi linier antar tetangga terdekat (Persamaan 2) [13], dengan penentuan tetangga terdekat menggunakan jarak Euclidean (Persamaan 3). Selanjutnya, *Edited Nearest Neighbors* (ENN) membersihkan data berdasarkan aturan mayoritas tetangga (Persamaan 4) [14]. Efektivitas SMOTEENN dalam menyeimbangkan data sekaligus membersihkan *noise* telah dibuktikan pada berbagai domain klasifikasi [15], dan kombinasi *hybrid sampling* seperti SMOTE-ENN terbukti meningkatkan *recall* dan *F1-score* dibandingkan tanpa penyeimbangan

$$x_{new} = x_i + \lambda(x_j - x_i), \quad \lambda \in [0,1] \quad (2)$$

Penentuan tetangga terdekat menggunakan jarak Euclidean:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (3)$$

Selanjutnya, *Edited Nearest Neighbors* (ENN) melakukan pembersihan data dengan aturan penghapusan berdasarkan mayoritas tetangga :

$$\hat{y}_i \neq \text{mode}(y_{kNN}(x_i)) \Rightarrow x_i \text{ is removed} \quad (4)$$

dengan x_i = sampel kelas minoritas; x_j = tetangga terdekat; λ = bilangan acak $[0,1]$; x_{new} = sampel sintetis; $d(x_i, x_j)$ = jarak Euclidean; m = jumlah fitur; $kNN(x_i)$ = himpunan tetangga terdekat; dan mode = label mayoritas.

2.4. Pemodelan

Penelitian ini menggunakan tujuh skenario eksperimen (*ablation study*): (S1) SVM *baseline* tanpa *balancing*; (S2) SVM + SMOTE; (S3) *Decision Tree* tanpa *ensemble* dan *balancing*; (S4) *Bagging* + *Decision Tree*; (S5) SMOTEENN + *Decision Tree*; (S6) SMOTEENN + *Balanced Bagging Classifier* + *Decision Tree* sebagai model utama; dan (S7) *Random Forest* + SMOTEENN sebagai model pembanding.

Untuk menjamin reproduisibilitas, seluruh eksperimen menggunakan *random seed* konsisten (*random_state* = 42) pada setiap tahap yang melibatkan proses acak, meliputi pembagian data, *resampling*, dan pelatihan model. Konfigurasi *hyperparameter* tiap algoritma dirangkum pada Tabel 1.

Tabel 1. Konfigurasi *Hyperparameter* Model

Skenario	Algoritma	Konfigurasi Parameter
S1, S2	SVM	kernel = RBF; C = 1; gamma = 'scale'; probability = True
S3	Decision Tree	kriteria = gini; max_depth = None; class_weight = None
S4	Bagging + DT	base estimator = Decision Tree; n_estimators = 20; bootstrap = True
S5	SMOTEENN + DT	Decision Tree dengan class_weight = 'balanced'

S6	Balanced Bagging + DT	base = Decision Tree (class_weight= 'balanced'); n_estimators = 20; sampling_strategy = 'auto'; replacement = False
S7	Random Forest	n_estimators = 100; class_weight = 'balanced'; kriteria = gini
-	SMOTE / SMOTEENN	k_neighbors = 5 (baku imbalanced-learn); random_state = 42

Decision Tree melakukan pemisahan data secara hierarkis hingga mencapai *leaf node*. *Bagging (Bootstrap Aggregating)* menggabungkan beberapa model dasar yang dilatih pada subset data berbeda melalui *majority voting* (Persamaan 5) [17]. *Balanced Bagging Classifier* menambahkan tahap penyeimbangan kelas pada setiap *bootstrap sample* sehingga lebih sensitif terhadap kelas minoritas tanpa mengorbankan stabilitas *ensemble*.

$$\hat{y}(x) = \arg \max_c \sum_{n=1}^N I(h_n(x) = c) \quad (5)$$

dengan $\hat{y}(x)$ = hasil prediksi akhir *ensemble* untuk masukan x ; $\arg \max_c$ = kelas c yang memaksimalkan jumlah suara; c = label kelas; N = jumlah *base estimator* (pada penelitian ini $N = 20$); n = indeks *base estimator*; $h_n(x)$ = prediksi *base estimator* ke- n untuk masukan x ; dan I = fungsi indikator, bernilai 1 jika kondisi terpenuhi dan 0 jika tidak.

2.5. Evaluasi Model

Evaluasi model menggunakan *confusion matrix* [18] dan metrik *Accuracy*, *Precision*, *Recall (Sensitivity)*, *Specificity*, *F1-score*, *G-Mean*, dan *AUC-ROC*. Bentuk *confusion matrix* ditunjukkan pada Tabel 2, dan perhitungan metrik pada Persamaan (6)-(12).

Tabel 2. Confusion Matrix

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Perhitungan masing-masing metrik evaluasi dilakukan menggunakan persamaan berikut:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (9)$$

$$F_1 - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

$$G\text{-Mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \quad (11)$$

$$\text{TPR} = \frac{TP}{TP+FN} \quad (12)$$

$$\text{FPR} = \frac{FP}{FP+TN} \quad (13)$$

dengan TP = *True Positive*; TN = *True Negative*; FP = *False Positive*; FN = *False Negative*; TPR = *True Positive Rate*; FPR = *False Positive Rate*; dan Sensitivity = Recall = TPR.

Nilai AUC-ROC merepresentasikan luas area di bawah kurva ROC dengan rentang 0-1. Semakin mendekati 1, semakin baik kemampuan diskriminasi model. Di antara ketujuh metrik tersebut, *F1-score* dijadikan indikator utama dalam pemilihan model karena merepresentasikan keseimbangan antara *precision* dan *recall*, sehingga lebih sesuai daripada *accuracy* untuk menilai performa pada kelas minoritas dalam konteks data tidak seimbang. Selain itu, digunakan pendekatan *adaptive threshold optimization* untuk mencari ambang keputusan terbaik dengan memaksimalkan *G-Mean*.

2.6. Adaptive Threshold Optimization

Pada klasifikasi biner, keputusan akhir umumnya menggunakan ambang baku 0,5 terhadap probabilitas keluaran model. Pada data tidak seimbang, ambang tersebut cenderung menguntungkan kelas mayoritas sehingga menurunkan kemampuan deteksi kelas minoritas. Untuk mengatasinya, penelitian ini menerapkan pencarian ambang optimal t^* yang memaksimalkan *G-Mean* melalui *grid search* pada 200 titik di rentang [0,01; 0,99]. *G-Mean* dipilih sebagai fungsi objektif karena secara simultan mempertimbangkan *sensitivity* dan *specificity*, sehingga ambang yang dihasilkan menjaga keseimbangan performa kedua kelas, berbeda dengan optimasi berbasis *accuracy* yang rentan bias terhadap kelas mayoritas. Prinsip *adaptive boundary* ini diadaptasi dari kerangka evaluasi Song et al. (2023) [10]. Ambang optimal ditentukan melalui Persamaan (13).

t^* = nilai ambang (*threshold*) optimal; t = kandidat nilai ambang pada rentang [0,1]; $\arg \max_t$ = nilai t yang memaksimalkan fungsi objektif; $Sensitivity(t)$ = nilai *sensitivity* (*recall*) pada ambang t ; dan $Specificity(t)$ = nilai *specificity* pada ambang t .

2.7. Validasi Model

Penelitian ini menerapkan strategi *hold-out validation* dengan pembagian data secara *stratified* menjadi data latih (60%), validasi (20%), dan uji (20%). *Stratified split* dipilih untuk mempertahankan proporsi kelas pada setiap subset ($IR \approx 3,34$ pada seluruh subset), sehingga karakteristik ketidakseimbangan tetap terjaga dan evaluasi bersifat representatif. Data uji yang sepenuhnya terpisah hanya digunakan pada tahap evaluasi akhir untuk memperoleh estimasi performa yang objektif.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Exploratory Data Analysis (EDA)

Pemeriksaan struktur data menunjukkan dataset terdiri dari 5.000 observasi dengan 10 variabel yang mencakup 3 bertipe *integer*, 3 bertipe *float*, dan 4 bertipe *object*. Hasil pengecekan *missing value* ditunjukkan pada Tabel 3.

Tabel 3. Hasil Pengecekan Missing Value

Variabel	Jumlah Missing Value	Persentase
Income	196	3,92%
CreditScore	194	3,88%
Education	198	3,96%
Variabel lainnya	0	0%
Total	588	1,18% dari seluruh sel

Berdasarkan Tabel 3, total 588 sel hilang setara 1,18% dari seluruh sel data (5.000×10) sehingga tergolong rendah dan tidak memerlukan penghapusan data. Dilakukan imputasi median untuk variabel numerik yaitu pada variabel *Income* dan *CreditScore* serta modus untuk variabel kategorikal yaitu *Education*.

Analisis statistik deskriptif menunjukkan *Age* berkisar 18-69 tahun dengan rata-rata 43,58. *Income* memiliki variasi besar dengan nilai minimum negatif sebesar -3.731, demikian pula pada *LoanAmount* sebesar -10.059, yang mengindikasikan data tidak wajar atau *noise*. Nilai negatif tersebut diperlakukan sebagai *invalid* dan diimputasi menggunakan median agar tidak mendistorsi proses pelatihan. *CreditScore* berkisar 300-849 dengan rata-rata 575,49, dan *YearsExperience* berkisar 0-39 tahun. Distribusi kelas target ditunjukkan pada

Tabel 4.

Tabel 4. Distribusi Kelas Target LoanApproved			
Kelas	Label	Jumlah Data	Persentase
0	Not Approved	3.849	76,98%
1	Approved	1.151	23,02%
Total		5.000	100%
Imbalance Ratio		3,34 : 1	

Berdasarkan Tabel 4, kelas *Not Approved* (0) jauh lebih dominan dengan *imbalance ratio* 3,34:1. Kondisi ini menyebabkan model tanpa penanganan khusus berpotensi menghasilkan akurasi tinggi secara statistik namun lemah dalam mendeteksi pemohon yang layak. Analisis *histogram* dan *boxplot* menunjukkan *Age* dan *YearsExperience* relatif merata, sedangkan *Income* dan *LoanAmount* memuat beberapa *outlier* ekstrem.

3.2. Hasil Data Preprocessing

Proses *preprocessing* berhasil menangani seluruh *missing value* dan mengonversi variabel kategorikal ke bentuk numerik melalui *label encoding* manual (Tabel 5). Dataset dibagi menggunakan *stratified sampling* dengan rasio 60:20:20 (Tabel 6), dengan proporsi kelas yang konsisten antar subset.

Tabel 5. Skema Encoding Variabel Kategorikal

Variabel	Skema Encoding
Gender	Female = 1, Male = 2
Education	High School = 1, Bachelors = 2, Masters = 3, PhD = 4
City	Chicago = 1, Houston = 2, New York = 3, San Francisco = 4
EmploymentType	Salaried = 1, Self-Employed = 2, Unemployed = 3

Tabel 6. Hasil Pembagian Data

Subset	Jumlah Data	Imbalance Ratio
Training	3.000	3,342
Validasi	1.000	3,348
Testing	1.000	3,348

3.3. Penanganan Data Tidak Seimbang (SMOTEENN)

SMOTEENN diterapkan hanya pada data latih. Hasil *resampling* (Tabel 7) menunjukkan *imbalance ratio* turun dari 3,342 menjadi 0,755. SMOTE meningkatkan jumlah kelas minoritas melalui data sintetis, sedangkan ENN menghapus data *noise* pada batas antar kelas, menghasilkan data latih yang lebih seimbang dan bersih.

Tabel 7. Hasil SMOTEENN Resampling pada Data Training

Kondisi	Jumlah	Not Approved (0)	Approved (1)	IR
Sebelum	3.000	2.310	690	3,342
Sesudah	3.782	1.627	2.155	0,755

3.4. Hasil Pemodelan

Sebelum pemodelan, digunakan mekanisme *adaptive threshold optimization* untuk mencari ambang keputusan terbaik dengan memaksimalkan *G-Mean*. Seluruh skenario dievaluasi pada *test set* menggunakan kerangka evaluasi yang konsisten. Hasil tiap skenario diuraikan berikut ini:

1. Skenario S1: SVM Baseline

Skenario S1 menggunakan SVM kernel RBF sebagai *baseline* tanpa penanganan ketidakseimbangan; konfigurasi kernel RBF dipilih karena umum digunakan sebagai acuan pada studi klasifikasi berbasis SVM [19]. Hasil evaluasi ditunjukkan pada Tabel 8.

Tabel 8. Hasil Evaluasi Model S1 (SVM Baseline)

Metrik	Nilai
Accuracy	0,9260
Sensitivity	0,8565
Specificity	0,9468
Precision	0,8277
G-Mean	0,9005
F1-score	0,8419
AUC-ROC	0,9239

Model S1 memperoleh G-Mean 0,9005 dan AUC-ROC 0,9239, menandakan kemampuan diskriminasi yang cukup baik meskipun belum ada penanganan *imbalance*. Namun, nilai F1-score 0,8419 relatif lebih rendah daripada G-Mean, yang mengindikasikan masih adanya ketimpangan antara *precision* (0,8277) dan *recall* (0,8565). Hal ini terjadi karena SVM dengan ambang baku cenderung membentuk *decision boundary* yang condong ke kelas mayoritas, sehingga sebagian kelas minoritas terklasifikasi sebagai kelas mayoritas. Model S1 menjadi acuan pembandingan bagi skenario S2 hingga S7.

2. Skenario S2: SVM + SMOTE

Skenario S2 menambahkan SMOTE pada data latih sebelum melatih SVM. Hasil evaluasi ditunjukkan pada Tabel 9.

Tabel 9. Hasil Evaluasi Model S2 (SVM + SMOTE)

Metrik	Nilai	Perubahan dari S1
Accuracy	0,9370	+0,011
Sensitivity	0,8652	+0,009
Specificity	0,9584	+0,012
Precision	0,8615	+0,034
G-Mean	0,9106	+0,010
F1-score	0,8633	+0,021
AUC-ROC	0,9213	-0,003

Penerapan SMOTE meningkatkan G-Mean dari 0,9005 ke 0,9106 dan F1-score dari 0,8419 ke 0,8633, tetapi AUC-ROC justru turun tipis dari 0,9239 menjadi 0,9213. Fenomena ini penting untuk dicermati, SMOTE memperbaiki operating point pada ambang tertentu dengan menyeimbangkan distribusi kelas, sehingga metrik yang bergantung pada G-Mean serta F1 membaik. Namun AUC-ROC mengukur kemampuan ranking probabilitas secara keseluruhan, dan penurunannya mengindikasikan bahwa sebagian sampel sintesis yang dihasilkan SMOTE menambah tumpang tindih antar kelas di ruang fitur, sehingga urutan probabilitas menjadi kurang tajam. Temuan ini konsisten dengan karakteristik SMOTE yang dapat menimbulkan noise pada daerah perbatasan kelas [9][20].

3. Skenario S3: Decision Tree

Skenario S3 menggunakan Decision Tree tunggal tanpa ensemble maupun balancing, model ini digunakan untuk melihat performa dasar satu pohon keputusan dalam menangani data tidak seimbang [21]. Hasil evaluasi ditunjukkan pada Tabel 10.

Tabel 10. Hasil Evaluasi Model S3 (Decision Tree)

Metrik	Nilai
Accuracy	0,9180
Sensitivity	0,8609
Specificity	0,9351
Precision	0,7984
G-Mean	0,8972
F1-score	0,8285
AUC-ROC	0,8980

Model S3 menunjukkan performa terendah di antara seluruh skenario dengan F1-score 0,8285 kemudian AUC-ROC 0,8980. Penyebab utamanya adalah sifat Decision Tree tunggal yang memiliki variansi tinggi dan mudah overfitting hal ini dikarenakan pohon tunggal membentuk decision boundary yang kaku dan mengikuti pola data latih, termasuk noise, sehingga generalisasinya terhadap data uji lemah. Tidak adanya mekanisme ensemble membuat model tidak memiliki mekanisme reduksi variansi, sehingga diperlukan pendekatan ensemble untuk meningkatkan stabilitas [17].

4. Skenario S4: Bagging + Decision Tree

Skenario S4 menggunakan ensemble Bagging dengan 20 Decision Tree sebagai base learner. Hasil evaluasi ditunjukkan pada Tabel 11.

Tabel 11. Hasil Evaluasi Model S4 (Bagging + Decision Tree)

Metrik	Nilai	Perubahan dari S3
Accuracy	0,9530	+0,035
Sensitivity	0,8826	+0,022
Specificity	0,9740	+0,039
Precision	0,9103	+0,112
G-Mean	0,9272	+0,030
F1-score	0,8962	+0,068
AUC-ROC	0,9417	+0,044

Bagging meningkatkan seluruh metrik secara signifikan dibandingkan S3, dengan lonjakan terbesar pada Precision (+0,112) dan AUC-ROC (+0,044). Peningkatan ini terjadi karena Bagging mengagregasi 20 pohon yang dilatih pada bootstrap sample berbeda, sehingga variansi antar pohon menjadi lebih rendah dan menghasilkan decision boundary yang lebih halus serta stabil. Lonjakan Precision yang tinggi menunjukkan berkurangnya false positive secara nyata. Meskipun demikian, belum adanya penanganan imbalance secara eksplisit menyisakan ruang perbaikan melalui kombinasi dengan teknik balancing.

5. Skenario S5: SMOTEENN + Decision Tree

Skenario S5 menerapkan SMOTEENN sebelum melatih *Decision Tree* tunggal. Hasil evaluasi ditunjukkan pada Tabel 12.

Tabel 12. Hasil Evaluasi Model S5 (SMOTEENN + Decision Tree)

Metrik	Nilai	Perubahan dari S4
Accuracy	0,8970	-0,056
Sensitivity	0,8826	0,000
Specificity	0,9013	-0,073
Precision	0,7276	-0,183
G-Mean	0,8919	-0,035
F1-score	0,7976	-0,099
AUC-ROC	0,8920	-0,050

Meskipun data telah diseimbangkan, performa S5 justru menurun tajam dibandingkan S4, terutama pada Precision (-0,183). Penurunan ini menurunkan keterbatasan penting yaitu SMOTEENN menggeser decision boundary ke arah kelas minoritas secara agresif, sementara Decision Tree tunggal tidak memiliki mekanisme stabilisasi variansi. Akibatnya, model menjadi terlalu longgar dalam memprediksi kelas Approved sehingga jumlah false positive meningkat drastis. Hasil ini membuktikan bahwa balancing pada tingkat data saja tidak cukup, dan bahkan dapat menurunkan performa, apabila tidak dipadukan dengan stabilisasi pada tingkat algoritma [15][22].

6. Skenario S6: SMOTEENN + Balanced Bagging Classifier + Decision Tree (Model Utama)

Skenario S6 merupakan model utama yang menggabungkan *data-level balancing* (SMOTEENN) dan *algorithm-level stabilization* (*Balanced Bagging Classifier* dengan *base estimator Decision Tree*). Hasil evaluasi ditunjukkan pada Tabel 13.

Tabel 13. Hasil Evaluasi Model S6 (SMOTEENN + Balanced Bagging + Decision Tree)

Metrik	Nilai	Perubahan dari S1
Accuracy	0,9580	+0,032
Sensitivity	0,8739	+0,017
Specificity	0,9831	+0,036
Precision	0,9393	+0,112
G-Mean	0,9269	+0,026
F1-score	0,9054	+0,064
AUC-ROC	0,9271	+0,003

Model S6 memberikan peningkatan pada seluruh metrik utama dibandingkan *baseline* S1, dengan peningkatan paling menonjol pada *Precision* (+0,112) dan *F1-score* (+0,064). Keunggulan ini bersumber dari kombinasi dua mekanisme yang saling melengkapi yaitu SMOTEENN membersihkan dan menyeimbangkan data pada tingkat data, sedangkan *Balanced Bagging* menyeimbangkan kelas pada setiap *bag* sekaligus mereduksi variansi pada tingkat algoritma. Perpaduan ini menghasilkan keseimbangan *bias variance* yang optimal, yaitu *decision boundary* yang sensitif terhadap kelas minoritas namun tetap stabil, sebagaimana tercermin dari *Precision* tinggi (0,9393) dan *false positive* yang rendah. Temuan ini selaras dengan penelitian mengenai keunggulan pendekatan hibrida SMOTEENN *ensemble* pada data kredit [23][24].

7. Skenario S7: Random Forest + SMOTEENN (Model Pembanding)

Skenario S7 menggunakan *Random Forest* (100 estimator, *class_weight=balanced*) dengan data hasil SMOTEENN sebagai model pembanding yaitu kombinasi SMOTE dan *Random Forest* diketahui efektif meningkatkan kinerja klasifikasi pada data tidak seimbang [25]. Hasil evaluasi ditunjukkan pada Tabel 14.

Tabel 14. Hasil Evaluasi Model S7 (Random Forest + SMOTEENN)

Metrik	Nilai
Accuracy	0,9580
Sensitivity	0,8739
Specificity	0,9831
Precision	0,9393
G-Mean	0,9269
F1-score	0,9054
AUC-ROC	0,9287

Model S7 menghasilkan performa yang hampir identik dengan S6 pada seluruh metrik berbasis *confusion matrix*, dengan AUC-ROC sedikit lebih tinggi (0,9287), selisih 0,0016 ini tidak signifikan. Analisis pemilihan model utama antara S6 dan S7 diuraikan pada Sub-bab 3.6.

3.5. Perbandingan Kinerja Model

Perbandingan kinerja seluruh model pada *test set* (N = 1000, IR = 3,35) ditunjukkan pada Tabel 15, dan ringkasan model terbaik per metrik pada Tabel 16.

Tabel 15. Hasil Evaluasi Kinerja Seluruh Model pada Test Set (N = 1000, IR = 3,35)

Model	Acc	Sens	Spec	Prec	G-Mean	F1	AUC
S1: SVM Baseline	0,926	0,857	0,947	0,828	0,901	0,842	0,924
S2: SVM + SMOTE	0,937	0,865	0,958	0,862	0,911	0,863	0,921
S3: Decision Tree	0,918	0,861	0,935	0,798	0,897	0,829	0,898
S4: Bagging + DT	0,953	0,883	0,974	0,910	0,927	0,896	0,942
S5: SMOTEENN + DT	0,897	0,883	0,901	0,728	0,892	0,798	0,892
S6: SMOTEENN+BBC+DT	0,958	0,874	0,983	0,939	0,927	0,905	0,927
S7: RF + SMOTEENN	0,958	0,874	0,983	0,939	0,927	0,905	0,929

Berdasarkan Tabel 15, model berbasis *ensemble learning* yang dikombinasikan dengan penanganan data tidak seimbang secara konsisten lebih unggul dibandingkan model dasar. S3 menunjukkan performa terendah, S4 memberikan peningkatan signifikan, S5 menurunkan *precision* secara drastis, sedangkan S6 dan S7 menghasilkan performa terbaik dan paling stabil dengan F1-score tertinggi (0,9054). Perbandingan *confusion matrix* tiga model kunci ditunjukkan pada Tabel 16.

Tabel 16. Perbandingan Confusion Matrix Model Kunci (N = 1000)

Model	TP	FN	FP	TN
S2: SVM + SMOTE	199	31	32	738
S5: SMOTEENN + DT	203	27	76	694
S6: SMOTEENN + BBC + DT	201	29	13	757

Model S5 menekan *false negative* menjadi 27, tetapi *false positive* melonjak menjadi 76, menunjukkan kecenderungan model yang terlalu agresif. Sebaliknya, S6 paling seimbang dengan hanya 13 *false positive* dan 29 *false negative*. Ringkasan model terbaik per metrik ditunjukkan pada Tabel 17.

Tabel 17. Model Terbaik Berdasarkan Setiap Metrik Evaluasi

Metrik	Model Terbaik	Nilai
Sensitivity	S4: Bagging + DT	0,8826
G-Mean	S4: Bagging + DT	0,9272
F1-Score	S6: SMOTEENN + BBC + DT	0,9054
AUC-ROC	S4: Bagging + DT	0,9417

3.6. Analisis Pemilihan Model Utama (S6 vs S7)

Model S6 dan S7 menghasilkan metrik berbasis *confusion matrix* yang identik, dengan perbedaan hanya pada AUC-ROC S6 (0,9271) dan S7 (0,9287). Selisih AUC sebesar 0,0016 terlalu kecil untuk berpengaruh terhadap performa keseluruhan. Pemilihan S6 sebagai model utama didasarkan pada tiga pertimbangan. Pertama, kompleksitas dan interpretabilitas S6 (*Balanced Bagging*) menggunakan 20 *base estimator*, sedangkan S7 (*Random Forest*) menggunakan 100 estimator dengan performa setara, S6 lebih ringkas, hemat komputasi, dan mudah diinterpretasi sesuai prinsip *parsimony*. Kedua, kesesuaian kerangka S6 menangani *imbalance* secara eksplisit pada dua tingkat, yaitu SMOTEENN dan *algorithm-level* melalui penyeimbangan kelas pada setiap *bootstrap sample*, sedangkan *Random Forest* hanya mengandalkan *class_weight*. Ketiga, prioritas metrik yaitu pada konteks kredit, *F1-score* merepresentasikan keseimbangan antara *precision* dan *recall* yang paling relevan, dan pada metrik ini S6 setara dengan S7 serta unggul atas seluruh skenario lain.

Perlu diperhatikan bahwa *G-Mean* dan *F1-score* berperan pada tahap yang berbeda. *G-Mean* digunakan sebagai fungsi objektif dalam penentuan ambang optimal setiap model, sehingga seluruh skenario dibandingkan pada *operating point* yang setara. Sementara itu, *F1-score* digunakan sebagai kriteria pemilihan model utama

karena paling merepresentasikan keseimbangan antara *precision* dan *recall* pada masalah persetujuan pinjaman. Dengan demikian, kedua metrik tidak saling bertentangan, melainkan bekerja pada tahap penyetelan ambang dan tahap seleksi model secara berurutan.

Perlu diakui secara objektif bahwa pada metrik *ranking* murni AUC-ROC, skenario S4 (*Bagging* + DT) justru mencapai nilai tertinggi (0,9417). Keunggulan S6 terletak pada *operating point* hasil penentuan ambang optimal berbasis *G-Mean*, bukan pada kemampuan *ranking* keseluruhan. Pemilihan S6 karenanya dilandasi keseimbangan antara *precision* dan *recall* dan kemampuan menekan *false positive* yang lebih relevan secara bisnis pada konteks persetujuan pinjaman.

3.7. Analisis dan Pembahasan

Perbedaan performa antar model tidak hanya disebabkan oleh jenis algoritma, tetapi terutama oleh cara model membentuk *decision boundary* pada data tidak seimbang. Distribusi kelas yang timpang menyebabkan model cenderung mempelajari pola kelas mayoritas lebih dominan, sehingga tanpa penanganan khusus *boundary* yang terbentuk akan menjadi bias. Penerapan SMOTEENN mengubah distribusi ruang fitur sehingga membantu model mengenali pola kelas minoritas, namun perubahan ini juga dapat meningkatkan tumpang tindih antar kelas yang berdampak pada naiknya *false positive*, sebagaimana terlihat pada S5 [22].

Metode *ensemble learning* seperti *Bagging* dan *Random Forest* bekerja dengan mereduksi variansi melalui agregasi beberapa model berbasis subset data berbeda, sehingga menghasilkan *decision boundary* yang lebih halus dan stabil [17]. Hal ini tercermin pada peningkatan *G-Mean* dan AUC-ROC pada S4 dibandingkan S3. Model S6 menunjukkan performa paling seimbang karena menggabungkan *data-level balancing* dan *algorithm-level stabilization*, menghasilkan keseimbangan *bias* dan variansi yang optimal. Temuan ini selaras dengan hasil penelitian yang menunjukkan pendekatan hibrida SMOTEENN dan *ensemble* menghasilkan performa terbaik pada data kredit [23][24]. Pada kasus persetujuan pinjaman, *false negative* menyebabkan kehilangan peluang bisnis, sedangkan *false positive* berisiko menimbulkan kerugian finansial, sehingga model yang mampu menyeimbangkan kedua jenis kesalahan seperti S6 menjadi pilihan paling tepat untuk penerapan nyata.

3.8. Keterbatasan Penelitian

Penelitian ini memiliki sejumlah keterbatasan yang perlu diperhatikan dalam menafsirkan hasil. Pertama, evaluasi hanya dilakukan pada satu dataset yaitu *Loan Risk Prediction Dataset*, sehingga generalisasi temuan ke domain kredit lain belum dapat dipastikan. Kedua, belum dilakukan pengujian pada dataset eksternal untuk memverifikasi konsistensi performa model pada distribusi data yang berbeda. Ketiga, evaluasi didasarkan pada satu skema *hold-out* tunggal tanpa uji signifikansi statistik antar model misalnya uji McNemar untuk perbedaan klasifikasi atau uji DeLong untuk perbedaan AUC, sehingga perbedaan performa yang kecil antar skenario, khususnya antara S4, S6, dan S7, belum dapat dinyatakan signifikan secara statistik.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa permasalahan klasifikasi persetujuan pinjaman pada data tidak seimbang dapat ditangani secara lebih efektif melalui integrasi teknik *resampling* pada tingkat data dan *ensemble learning* pada tingkat algoritma. Kontribusi utama penelitian terletak pada pembuktian empiris, melalui rangkaian skenario *ablation*, bahwa penyeimbangan data dan stabilisasi *ensemble* bersifat komplementer. Penerapan *resampling* secara tunggal cenderung memperlebar tumpang tindih antar kelas dan meningkatkan *false positive*, sedangkan *ensemble* tanpa penanganan *imbalance* belum optimal dalam mendeteksi kelas minoritas. Kombinasi keduanya, sebagaimana diwujudkan pada model SMOTEENN + *Balanced Bagging Classifier* berbasis *Decision Tree*, menghasilkan keseimbangan antara *precision* dan *recall* yang paling stabil serta kemampuan terbaik dalam menekan kesalahan klasifikasi pada kedua kelas secara proporsional.

Secara praktis, temuan ini memberikan implikasi bagi lembaga keuangan dalam merancang sistem *credit scoring* yang lebih adil terhadap kelas minoritas. Model yang mampu menyeimbangkan *false positive* dan *false negative* secara proporsional membantu institusi menekan risiko kredit macet sekaligus mengurangi hilangnya peluang bisnis, sehingga mendukung pengambilan keputusan kredit yang lebih objektif pada kondisi data yang timpang.

Untuk penelitian selanjutnya, disarankan pengujian model pada dataset nyata dari industri keuangan serta validasi eksternal lintas dataset. Selain itu, penerapan uji signifikansi statistik antar model dan validasi silang atau *cross-validation* perlu dilakukan untuk memperkuat keandalan perbandingan. Pengembangan teknik *balancing* alternatif seperti ADASYN atau *cost-sensitive learning* dan *ensemble* lanjutan seperti XGBoost atau

stacking juga layak dieksplorasi. Terakhir, integrasi *Explainable Artificial Intelligence* (XAI) akan meningkatkan transparansi dan interpretabilitas prediksi, sehingga hasil model lebih dapat dipercaya dalam pengambilan keputusan di domain keuangan.

DAFTAR PUSTAKA

- [1] R. Anand, H. Singh, K. Sardana, D. N. Gupta, N. Sindhwani, and M. Mittal, "Loan approval prediction using machine learning," in *Intelligent Systems Design and Applications (ISDA 2023)*, Lecture Notes in Networks and Systems, vol. 1052, Springer, 2024, doi: 10.1007/978-3-031-64776-5_34.
- [2] S. Hamori et al., "Credit card default prediction: An empirical analysis on predictive performance using statistical and machine learning methods," *Journal of Risk and Financial Management*, vol. 18, no. 1, art. 23, 2025, doi: 10.3390/jrfm18010023.
- [3] W. Yotsawat, P. Wattuya, and A. Srivihok, "A novel method for credit scoring based on cost-sensitive neural network ensemble," *IEEE Access*, vol. 9, pp. 78521-78537, 2021.
- [4] S. Rezvani and X. Wang, "A broad review on class imbalance learning techniques," *Applied Soft Computing*, vol. 143, art. 110415, 2023, doi: 10.1016/j.asoc.2023.110415.
- [5] E. Dumitrescu, S. Hué, C. Hurlin, and S. Tokpavi, "Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects," *European Journal of Operational Research*, vol. 297, no. 3, pp. 1178-1192, 2021.
- [6] R. K. Ch, K. Meenadevi, D. Kumar D, and R. Nagaraj, "Deep learning in credit risk assessment: A data-driven approach to transforming financial decision-making and risk analytics," *Journal of Risk and Financial Management*, vol. 19, no. 5, art. 361, 2026, doi: 10.3390/jrfm19050361.
- [7] K. Shamsi, D. Novokmet, J. Peters, M. L. Liu, P. K. Edwards, and V. Khoshdel, "LendNova: Towards automated credit risk assessment with language models," *arXiv:2601.02573*, 2026, doi: 10.48550/arXiv.2601.02573.
- [8] S. Rezvani, F. Pourpanah, C. P. Lim, and Q. M. J. Wu, "Methods for class-imbalanced learning with support vector machines: a review and an empirical evaluation," *Soft Computing*, vol. 28, pp. 11873-11894, 2024, doi: 10.1007/s00500-024-09931-5.
- [9] F. Z. El-Hassani, A. Amri, N.-E. Joudar, and K. Haddouch, "Performance evaluation of hybrid machine learning algorithms for online lending credit risk prediction," *Applied Artificial Intelligence*, vol. 38, no. 1, art. 2358661, 2024, doi: 10.1080/08839514.2024.2358661.
- [10] Y. Song, Y. Wang, X. Ye, R. Zaretzki, and C. Liu, "Loan default prediction using a credit rating-specific and multi-objective ensemble learning scheme," *Information Sciences*, vol. 629, pp. 599-617, 2023, doi: 10.1016/j.ins.2023.02.014.
- [11] M. Alamri and M. Ykhlef, "Hybrid undersampling and oversampling for handling imbalanced credit card data," *IEEE Access*, vol. 12, pp. 14050-14060, 2024, doi: 10.1109/ACCESS.2024.3355813.
- [12] C. He, L. Yu, X. Xi, et al., "An ensemble learning model with dynamic sampling and feature fusion network for class sparsity in credit risk classification," *Annals of Operations Research*, vol. 353, no. 2, pp. 761-791, 2025, doi: 10.1007/s10479-025-06528-5.
- [13] W. Priani, E. Sulistianingsih, and N. Imro'a, "Implementasi teknik oversampling SMOTENC pada algoritma KNN," *BIMASTER*, vol. 14, no. 3, 2025.
- [14] T. Yulian and E. R. Susanto, "Comparative analysis of oversampling and SMOTEENN techniques," *Sistemasi*, vol. 14, no. 3, pp. 1332-1343, 2025.
- [15] M. A. Latief, L. R. Nabila, W. Miftakhurrahman, S. Ma'rufatullah, and H. Tantyoko, "Handling imbalance data using hybrid sampling SMOTE-ENN in lung cancer classification," *Int. J. Eng. Comput. Sci. Appl. (IJECSA)*, vol. 3, no. 1, pp. 11-18, 2024, doi: 10.30812/ijecs.v3i1.3758.
- [16] A. K. Yadav, B. Alam, and O. Pal, "Hybrid resampling using SMOTE-ENN and SMOTE-Tomek to handle data imbalance on EEG signal for seizure detection and classification," *International Journal of Information Technology*, 2025, doi: 10.1007/s41870-025-02832-2.
- [17] G. Ngo, R. Beard, and R. Chandra, "Evolutionary bagging for ensemble learning," *Neurocomputing*, vol. 510, pp. 1-14, 2022.
- [18] S. Sathyanarayanan and B. R. Tantri, "Confusion matrix-based performance evaluation metrics," *African Journal of Biomedical Research*, vol. 27, no. 4S, pp. 4023-4031, 2024.

-
- [19] A. Setiawan et al., "Optimasi performa SVM dan Random Forest dengan hyperparameter tuning," JUKTISI, vol. 4, no. 3, pp. 2141-2149, 2026.
- [20] A. H. Maulia and A. Salam, "Benchmarking oversampling strategies," Journal of Applied Informatics and Computing, vol. 10, no. 1, pp. 747-761, 2026.
- [21] S. N. Sari et al., "Teknik data mining dengan menggunakan algoritma decision tree untuk mengetahui pola pemahaman mahasiswa," Bulletin of Information Technology, vol. 6, no. 4, pp. 417-431, 2025.
- [22] G. Husain, D. Nasef, R. Jose, et al., "SMOTE vs. SMOTEENN: A study on the performance of resampling algorithms for addressing class imbalance in regression models," Algorithms, vol. 18, no. 1, art. 37, 2025, doi: 10.3390/a18010037.
- [23] M. Milli, S. Aras, and İ. Kocakoç, "Investigating the effect of class balancing methods on classification performance," İzmir Yönetim Dergisi, vol. 5, no. 2, 2024.
- [24] A. Akinjole et al., "Ensemble-based machine learning algorithm for loan default risk prediction," Mathematics, vol. 12, no. 21, art. 3423, 2024.
- [25] E. Erlin et al., "Dampak SMOTE terhadap kinerja Random Forest classifier," MATRIK, vol. 21, no. 3, pp. 677-690, 2022.