

Comparative Study of Logistic Regression, Random Forest, and XGBoost for Bank Loan Approval Classification

Hamdika Putra ^{1*}, Rumini ²

* Informatika, Universitas Amikom Yogyakarta, Indonesia
hamdikaputra@students.amikom.ac.id ¹, rumini@amikom.ac.id ²

Article Info

Article history:

Received 2025-08-21

Revised 2025-09-08

Accepted 2025-10-01

Keyword:

Credit Risk

Bank Loan Approval

Logistic Regression

Random Forest

XGBoost

ABSTRACT

Bank loan approval plays a vital role in ensuring financial institutions can minimize credit risk while supporting economic growth. Default prediction is a crucial aspect of banking credit risk management. This study compares three machine learning algorithms Logistic Regression, Random Forest, and Extreme Gradient Boosting (XGBoost) to classify bank loan approvals using a combination of application, previous application, and bureau datasets. The workflow includes data merging, cleaning, missing value imputation, handling unknown values, feature engineering (such as converting day-based variables into years, calculating total submitted documents, income-to-annuity ratio, and employment-to-income ratio), encoding (label and one-hot), scaling (min-max normalization), feature selection based on correlation analysis, handling class imbalance with SMOTE, as well as modeling and evaluation using Accuracy, Precision, Recall, F1-score, and AUC. The results show that Logistic Regression yields the highest AUC of 0.741498, outperforming Random Forest (0.713758) and XGBoost (0.715944). From a business perspective, implementing the best model reduced the Loss Given Default (LGD) by 39.77 %, from \$1,705,098,055.50 to \$1,026,944,185.50. This finding confirms that simpler models remain competitive on imbalanced datasets when supported by appropriate preprocessing and balancing strategies.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Kredit merupakan salah satu instrumen utama dalam industri perbankan yang berperan penting dalam mendorong pertumbuhan ekonomi, namun di sisi lain juga menimbulkan risiko gagal bayar yang dapat memengaruhi kesehatan keuangan bank secara signifikan [1]. Risiko gagal bayar (*default*) menjadi perhatian utama bagi lembaga keuangan, karena apabila tidak dapat diidentifikasi sejak awal, akan menimbulkan kerugian besar baik dari sisi modal maupun reputasi bank [2]. Oleh karena itu, diperlukan suatu mekanisme penilaian kelayakan kredit yang mampu memprediksi kemungkinan terjadinya gagal bayar secara akurat, sehingga keputusan pemberian pinjaman dapat dilakukan secara lebih terukur dan mengurangi risiko kredit bermasalah (*non-performing loan*) [2].

Perkembangan teknologi informasi telah mendorong penerapan *machine learning* sebagai solusi inovatif dalam

sistem penilaian risiko kredit [3]. Berbeda dengan metode tradisional yang hanya mengandalkan analisis keuangan sederhana, *machine learning* mampu mengolah data dalam jumlah besar dengan variabel yang kompleks, sehingga pola-pola yang sebelumnya sulit dideteksi dapat diidentifikasi secara otomatis [4]. Salah satu pendekatan populer dalam permasalahan ini adalah klasifikasi biner, di mana peminjam dikategorikan ke dalam kelompok berisiko atau tidak berisiko berdasarkan profil dan riwayat kreditnya [5].

Algoritma seperti *logistic regression*, *random forest*, dan *extreme gradient boosting (XGBoost)* telah banyak digunakan dalam penelitian terdahulu untuk kasus serupa, masing-masing dengan karakteristik dan keunggulan yang berbeda [6], [7], [8], [9]. *Logistic regression* unggul dalam interpretabilitas dan kemudahan implementasi [6], *random forest* memiliki ketahanan terhadap *overfitting* serta mampu menangani data berdimensi tinggi [7], [8], sementara *XGBoost* dikenal memiliki kemampuan optimisasi yang

efisien dan performa tinggi dalam berbagai kompetisi prediksi [9], [10].

Pemilihan tiga algoritma tersebut didasarkan pada relevansinya dalam domain *credit scoring*. Logistic Regression telah lama digunakan sebagai standar industri perbankan karena interpretabilitas dan kemampuannya memberikan estimasi probabilitas gagal bayar yang transparan [6], [7]. Random Forest dipilih karena ketangguhannya terhadap data dengan dimensi tinggi serta kemampuannya menangani interaksi nonlinier antar fitur [8]. Sementara itu, XGBoost merupakan salah satu algoritma *state-of-the-art* untuk data tabular, yang terbukti konsisten memberikan performa unggul dalam kompetisi pemodelan prediktif, termasuk pada kasus risiko kredit [9], [10]. Penelitian terdahulu juga menunjukkan ketiga algoritma ini efektif dalam mendukung proses penilaian risiko kredit, sehingga relevan membandingkan konteks ini [11], [12].

Laporan industri terbaru (misalnya Deloitte 2023 dan McKinsey 2024) mengungkapkan bahwa bank-bank global semakin mengadopsi *machine learning* dalam proses penilaian risiko, terutama untuk meningkatkan akurasi prediksi kredit bermasalah dan mengurangi tingkat *Non-Performing Loan (NPL)*. Bahkan, regulator di beberapa negara mulai mendorong penerapan model ML sebagai bagian dari praktik manajemen risiko modern. Hal ini menegaskan urgensi penelitian di bidang ini, khususnya pada konteks perbankan di negara berkembang.

Meskipun banyak penelitian sebelumnya membandingkan performa model *machine learning* dalam *credit scoring*, sebagian besar hanya menilai berdasarkan akurasi atau AUC. Masih sedikit penelitian yang menghubungkan performa model dengan *dampak bisnis*, seperti *Loss Given Default (LGD)* dan potensi pengurangan kerugian kredit. Oleh karena itu, penelitian ini tidak hanya berfokus pada perbandingan performa algoritma, tetapi juga menambahkan dimensi analisis risiko finansial sehingga hasilnya lebih aplikatif bagi pengambilan keputusan di sektor perbankan.

Dalam praktiknya, tantangan besar yang sering dihadapi adalah ketidakseimbangan distribusi kelas pada data kredit [11], di mana jumlah nasabah yang membayar tepat waktu jauh lebih besar dibandingkan yang gagal bayar [12]. Kondisi ini dapat menyebabkan model bias terhadap kelas mayoritas, sehingga prediksi gagal bayar menjadi tidak akurat [13]. Untuk mengatasi hal tersebut, penelitian ini menggunakan metode *synthetic minority oversampling technique (SMOTE)* yang mampu memperbanyak sampel pada kelas minoritas tanpa menghapus data kelas mayoritas [14].

Selain itu, proses pra-pemrosesan data menjadi tahapan krusial yang mencakup penanganan *missing value*, penghapusan kolom dengan proporsi data hilang yang tinggi, imputasi nilai median atau modus, penanganan nilai kategori tidak valid (seperti nilai *unknown*), penghapusan duplikat, serta rekayasa fitur seperti konversi variabel waktu menjadi umur atau lama bekerja, dan perhitungan rasio pendapatan terhadap anuitas [4], [5]. Tahapan ini diikuti oleh proses *encoding* menggunakan *label encoding* untuk variabel biner

dan *one-hot encoding* untuk variabel multi-kategori, serta normalisasi menggunakan *min-max scaler* untuk menyamakan skala antarfitur [14].

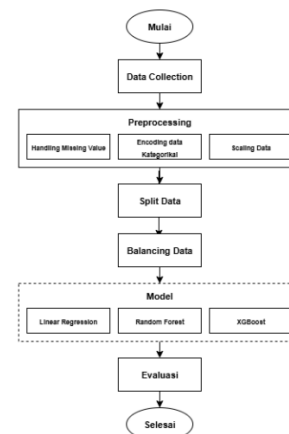
Dataset yang digunakan dalam penelitian ini merupakan hasil penggabungan dari tiga sumber data, yaitu *application train*, *previous application*, dan *bureau*, sehingga mencakup informasi demografis, profil pengajuan pinjaman sebelumnya, serta catatan kredit eksternal [3], [5]. Setelah proses penggabungan dan agregasi, dilakukan pemilihan fitur berdasarkan korelasi absolut terhadap variabel target dan pengetahuan domain (*domain knowledge*) [2], [3] untuk mempertahankan variabel yang paling relevan terhadap risiko gagal bayar. Selanjutnya, data dibagi menjadi data latih dan data uji dengan rasio 80:20 menggunakan teknik stratifikasi untuk menjaga proporsi kelas pada kedua subset [5].

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk membandingkan kinerja tiga algoritma klasifikasi, yaitu *logistic regression*, *random forest*, dan *XGBoost* [6], [7], [8] dalam memprediksi persetujuan pinjaman bank, mengidentifikasi algoritma dengan performa terbaik berdasarkan nilai *area under curve (AUC)*, serta menganalisis dampak penerapan model tersebut terhadap metrik bisnis *loss given default (LGD)* [15], [16]. Dengan demikian, hasil penelitian ini diharapkan dapat memberikan kontribusi nyata bagi pengembangan sistem penilaian risiko kredit yang lebih akurat, efisien, dan bermanfaat dalam pengambilan keputusan strategis di sektor perbankan.

II. METODE

A. Alur Penelitian

Penelitian ini dilakukan dengan menerapkan metode *machine learning* untuk memprediksi risiko kredit menggunakan tiga algoritma klasifikasi, yaitu Logistic Regression (LR), Random Forest (RF), dan Extreme Gradient Boosting (XGBoost) [1], [2].



Gambar 1. Diagram Alir Penelitian

Proses penelitian terdiri dari beberapa tahap yang berperan penting dalam membangun model yang efektif, meliputi pengumpulan data, *preprocessing*, *feature engineering*,

seleksi fitur, penanganan data tidak seimbang, pemodelan, dan evaluasi [3]. Alur penelitian ditunjukkan pada Gambar 1.

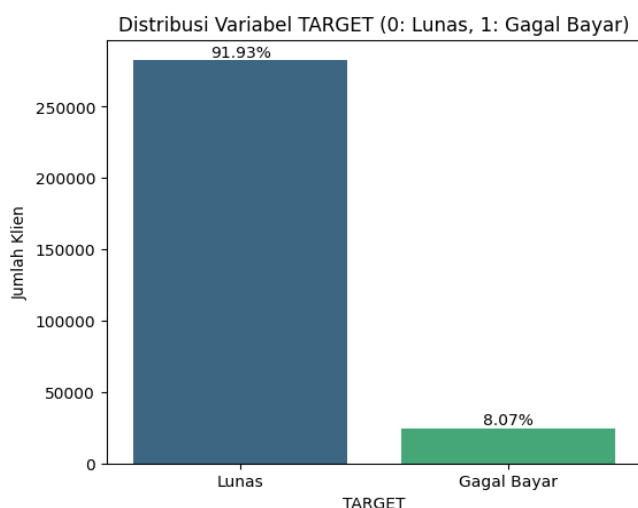
B. Data Collection

Jenis penelitian ini adalah penelitian kuantitatif dengan menggunakan data numerik yang sebagian berbentuk biner untuk menguji performa model machine learning dalam memprediksi risiko default kredit [3]. Dataset yang digunakan bersumber dari kompetisi *Home Credit Default Risk* di situs Kaggle [4], [5]. Dataset terdiri dari beberapa file, yaitu *application_train.csv*, *previous_application.csv*, dan *bureau.csv*. Penggabungan dataset dilakukan dengan menggunakan primary key berupa identitas unik nasabah untuk menyatukan informasi dari berbagai sumber [4]. Untuk identifikasi lebih jelasnya dapat dilihat pada Tabel 1.

TABEL 1
IDENTIFIKASI DATA

Judul	Home Credit Indonesia
Dataset	Application_train.csv, previous_application.csv, bureau.csv
Total Data	3.694.153
Duplicated	0
Imbalance Degree	0,67 (Dataset tidak seimbang, dengan kelas minoritas jauh lebih kecil daripada kelas mayoritas)
Tareget	sk_id_curr, target

Pada Tabel 1 dapat dilihat bahwa dataset memiliki total 3.694.153 baris data hasil penggabungan dari tiga file utama. Variabel target bersifat biner, dengan label 0 menunjukkan nasabah yang tidak *default* dan label 1 menunjukkan nasabah yang *default* [5]. Distribusi kelas target ditunjukkan pada Gambar 2, di mana kelas 0 memiliki 282.686 data (91,93%) dan kelas 1 memiliki 24.825 data (8,07%) [6].



Gambar 2. Bar Chart Distribusi Kelas Variabel TARGET

Berdasarkan distribusi tersebut, terlihat bahwa dataset memiliki ketidakseimbangan kelas yang cukup signifikan, dengan proporsi kelas minoritas (label 1) yang jauh lebih kecil

dibandingkan kelas mayoritas (label 0). Kondisi ini berpotensi mempengaruhi kinerja model, sehingga diperlukan teknik penanganan ketidakseimbangan data seperti *oversampling* atau *SMOTE* [7] pada tahap pra-pemrosesan.

C. Preprocessing Data

Pada tahap pra-pemrosesan, data mentah dibersihkan dari nilai yang tidak valid, seperti mengganti nilai *unknown* pada kolom jenis kelamin (12 entri) dan jenis organisasi (55.374 entri) menjadi *missing value* (NaN) [8]. Analisis *missing value* kemudian dilakukan untuk mengidentifikasi kolom dengan persentase kekosongan data tinggi. Sebanyak 18 kolom dengan proporsi *missing value* di atas 60%—misalnya luas area umum rata-rata (69,87%), apartemen non-huni rata-rata (69,43%), tahun bangunan rata-rata (66,50%), dan usia mobil milik pribadi (65,99%)—dihapus, sehingga jumlah fitur berkurang dari 122 menjadi 107 kolom [8], [9].

Selanjutnya, imputasi dilakukan untuk mengisi nilai hilang: fitur numerik seperti *person_income* dan *loan_amnt* diisi menggunakan nilai **median** untuk menghindari distorsi distribusi data akibat *outlier*, sedangkan fitur kategorikal seperti *person_home_ownership* dan *loan_intent* diisi menggunakan **modus (most frequent)** agar konsistensi kategori tetap terjaga. Setelah semua *missing value* teratasi (jumlah *missing value* menjadi 0), dilakukan *scaling* pada fitur numerik dengan *min-max scaler* untuk mengubah rentang nilai menjadi 0–1 sehingga setiap fitur memiliki bobot yang setara dalam proses pelatihan model [9], [10].

Mengingat distribusi kelas target tidak seimbang (kelas 1 sekitar 8,07% dan kelas 0 sekitar 91,93%), diterapkan metode **SMOTE (Synthetic Minority Over-sampling Technique)** untuk memperbesar proporsi kelas minoritas dan mengurangi bias model terhadap kelas mayoritas [7], [10].

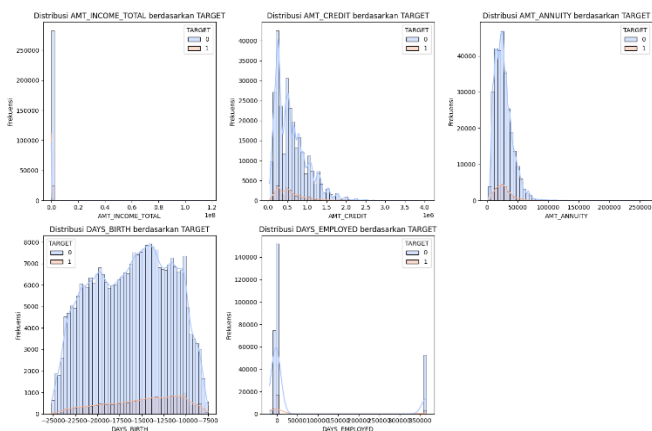
Terakhir, *feature encoding* dilakukan untuk mengubah data kategorikal menjadi bentuk numerik: **label encoding** digunakan pada kolom dengan dua kategori unik seperti *gender*, *car_ownership*, dan *previous_loan_defaults_on_file*, sementara **one-hot encoding** digunakan pada variabel nominal dengan lebih dari dua kategori unik seperti *person_home_ownership*, *loan_intent*, dan *person_emp_type*. Setelah proses ini, jumlah fitur meningkat menjadi 204 kolom [10].

Seluruh tahapan ini digunakan untuk memastikan bahwa data pada proyek klasifikasi kredit ini berada dalam format yang bersih, terstandarisasi, seimbang, dan optimal sehingga model dapat belajar dengan efektif serta menghasilkan prediksi yang lebih akurat dan tidak bias.

D. Exploratory Data Analysis (EDA)

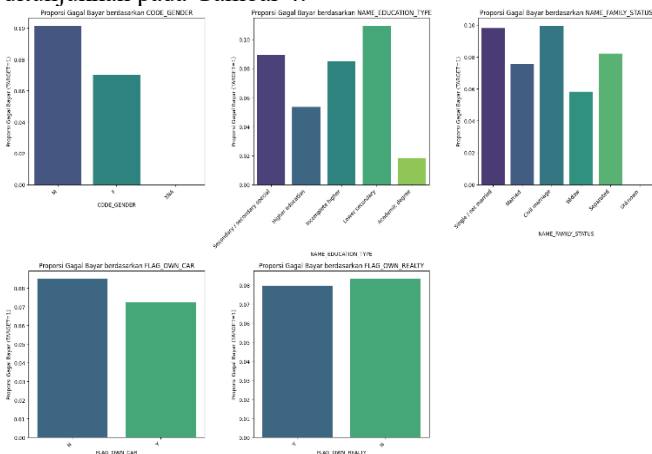
Tahap eksplorasi data dilakukan untuk memahami karakteristik dan pola pada dataset *Home Credit*, khususnya pada fitur yang berpotensi memengaruhi risiko gagal bayar [11]. Analisis diawali dengan pemeriksaan distribusi beberapa fitur numerik kunci terhadap variabel target, seperti jumlah kredit, total pendapatan, dan usia nasabah. Visualisasi distribusi ini digunakan untuk melihat perbedaan pola antara

nasabah yang tidak gagal bayar ($target = 0$) dan yang gagal bayar ($target = 1$), sehingga dapat memberikan indikasi awal relevansi fitur terhadap prediksi risiko kredit [12], [15], [16].sebagaimana ditunjukkan pada Gambar 3.



Gambar 3. Distribusi Fitur Numerik Kunci vs TARGET

Tahap selanjutnya adalah pemeriksaan fitur kategorikal kunci seperti *name_contract_type*, *code_gender*, dan *flag_own_car* dengan menampilkan proporsi gagal bayar pada setiap kategori [11], [12]. Analisis ini bertujuan untuk mengidentifikasi kategori tertentu yang memiliki tingkat risiko gagal bayar lebih tinggi dibandingkan kategori lainnya, sehingga dapat menjadi pertimbangan dalam proses pemodelan. Hasil analisis distribusi kategori tersebut ditunjukkan pada Gambar 4.



Gambar 4. Distribusi Fitur Kategorikal Kunci vs TARGET

Hasil eksplorasi ini memberikan gambaran awal yang penting sebelum analisis lanjutan pada tahap berikutnya, seperti pemetaan hubungan antar fitur dengan variabel target menggunakan korelasi serta pemilihan fitur yang relevan.

E. Feature Engineering

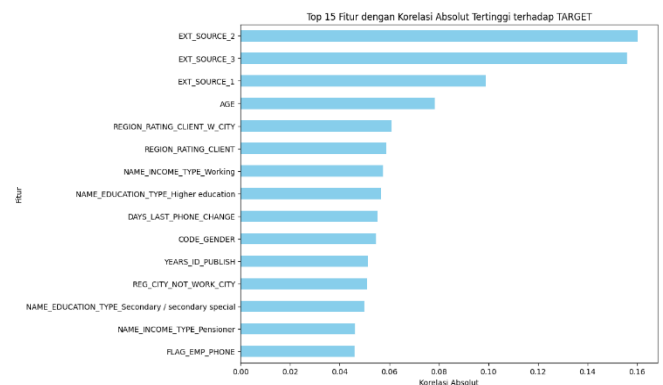
Tahap *feature engineering* dimulai dengan eksplorasi data untuk mengidentifikasi *missing value*, yang kemudian ditangani melalui imputasi: nilai pada fitur numerik diisi dengan median, sedangkan fitur kategorikal diisi dengan

mode, sehingga seluruh data bebas dari nilai hilang. Selanjutnya dilakukan *scaling* pada seluruh fitur numerik menggunakan *MinMaxScaler*, setelah kolom *target* dan *sk_id_curr* dikeluarkan dari proses ini. Mengingat distribusi kelas pada variabel *target* tidak seimbang, digunakan teknik penanganan data tidak seimbang (misalnya *SMOTE*) untuk menyeimbangkan proporsi kelas.

Pada tahap *feature engineering*, beberapa kolom berbasis hari seperti *days_birth*, *days_employed*, *days_registration*, dan *days_id_publish* dikonversi ke satuan tahun menjadi *age*, *years_employed*, *years_registration*, dan *years_id_publish*, dengan penghapusan kolom aslinya serta penanganan nilai anomali pada *years_employed* [10]. Dibuat pula fitur *total_document* dari penjumlahan seluruh kolom *flag_document*, lalu kolom aslinya dihapus. Dua fitur rasio ditambahkan, yakni *income_annuity_percent* (persentase annuitas terhadap pendapatan) yang dibatasi hingga 150% untuk mencegah nilai ekstrem, serta *etc* (rasio total kredit terhadap pendapatan) untuk mengukur beban kredit nasabah [10], [12]. Hasil akhir tahap ini menghasilkan dataset berukuran 307.511 baris \times 90 kolom yang telah siap digunakan untuk proses pemodelan.

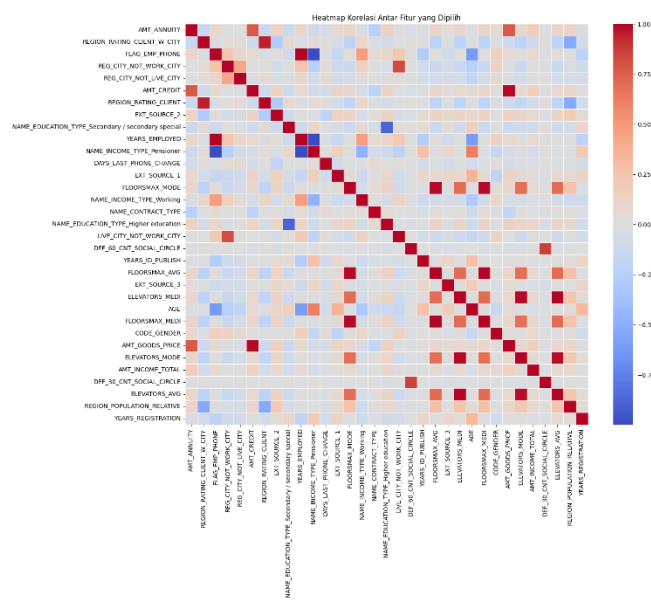
F. Feature Selection

Pada tahap *feature selection*, proses dimulai dengan menghitung korelasi absolut antara setiap fitur dengan variabel *target* [13], [14] (*target*) untuk mengidentifikasi variabel yang memiliki pengaruh paling signifikan terhadap prediksi. Berdasarkan perhitungan tersebut, diperoleh 15 fitur teratas dengan nilai korelasi absolut tertinggi, di antaranya *ext_source_2* (0,1603), *ext_source_3* (0,1559), *ext_source_1* (0,0989), dan *age* (0,0782). Daftar ini divisualisasikan dalam bentuk *bar chart* Gambar 5. untuk memberikan gambaran yang lebih jelas mengenai tingkat pengaruh masing-masing fitur terhadap *target*.



Gambar 5. Top 15 Fitur Berdasarkan Korelasi Absolut dengan TARGET

Selanjutnya, dipilih 30 fitur teratas berdasarkan nilai korelasi tersebut, kemudian ditambahkan beberapa variabel penting berdasarkan *domain knowledge*, yaitu *amt_credit*, *amt_annuity*, *amt_income_total*, *age*, dan *years_employed*. Setelah penggabungan dan penghapusan duplikasi, total terdapat 33 fitur yang digunakan tahap pemodelan [13], [14].



Gambar 6. Heatmap Korelasi Antar Fitur yang Dipilih

Untuk memastikan tidak terjadi multikolinearitas yang berlebihan di antara fitur terpilih [14], dilakukan visualisasi *heatmap* korelasi antar variabel. Visualisasi ini memudahkan dalam mengidentifikasi pasangan variabel yang memiliki korelasi tinggi sehingga dapat menjadi pertimbangan untuk dilakukan penghapusan fitur pada tahap optimasi selanjutnya.

G. Penanganan Data Tidak Seimbang

Sebelum pelatihan model, data dibagi menjadi *data latih* (246.008 baris, 33 fitur) dan *data uji* (61.503 baris, 33 fitur). Distribusi label *target* pada *data latih* menunjukkan ketidakseimbangan yang cukup besar, di mana kelas mayoritas 0 (nasabah lancar) berjumlah 226.148 dan kelas minoritas 1 (nasabah macet) hanya 19.860. Kondisi ini berisiko membuat model bias terhadap kelas mayoritas, sehingga digunakan teknik *SMOTE* (*Synthetic Minority Oversampling Technique*) dengan *sampling strategy* 0,5 untuk memperbanyak kelas minoritas hingga 50% dari mayoritas [7], [10].

TABEL II
TEKNIK SMOTE

Kelas	Sebelum	Sesudah
0 (Lancar)	226.148	226.148
1 (Macet)	19.860	113.074
Total	246.008	339.222

Dengan hasil tersebut, distribusi kelas menjadi lebih proporsional dan diharapkan dapat meningkatkan kemampuan model dalam mengenali kedua kelas secara seimbang. Proses *oversampling* ini sangat penting pada proyek prediksi risiko kredit, karena kesalahan mengklasifikasikan nasabah berisiko tinggi menimbulkan kerugian signifikan bagi lembaga keuangan.

H. Pemodelan Algoritma

1) Logistic Regression(LR)

Logistic regression diterapkan untuk memodelkan probabilitas persetujuan pinjaman berdasarkan kombinasi linier dari fitur nasabah. Algoritma ini dipilih karena sifatnya yang sederhana, mudah diinterpretasikan, dan efektif untuk data dengan hubungan linier antara fitur dan target. Dalam proyek ini, parameter LR dituning untuk memperoleh performa terbaik. Hasil tuning menunjukkan bahwa *solver liblinear*, *penalty* L2, dan *C* = 10 memberikan prediksi yang paling optimal. Logistic regression menjadi *baseline* dalam proyek ini, sehingga performa model lain dapat dibandingkan secara jelas [1], [6]. Fungsi prediksi probabilitas menggunakan logistic regression adalah:

$$P(y = 1 | X) = \frac{1}{1 + e^{(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)}} \quad (1)$$

Prediksi kelas dibuat dengan ambang batas 0,5:

- Jika probabilitas $\geq 0,5 \rightarrow$ *default* (1)
- Jika probabilitas $< 0,5 \rightarrow$ tidak *default* (0)

Ambang batas ini memanfaatkan *sigmoid function* pada logistic regression, di mana output model berupa probabilitas antara 0 dan 1. Pemilihan ambang 0,5 bersifat konvensional, namun bisa disesuaikan tergantung kebutuhan bisnis, misalnya untuk meningkatkan deteksi risiko gagal bayar.

2) Random Forest(RF)

Random Forest digunakan untuk meningkatkan akurasi prediksi melalui pendekatan ensemble berbasis pohon keputusan. Algoritma ini membangun banyak pohon dari sampel acak dan menggabungkan hasilnya, sehingga mampu menangani data yang kompleks dan mengurangi risiko *overfitting* dibandingkan pohon tunggal. Dalam proyek ini, RF dituning pada beberapa parameter kunci, yaitu *n_estimators*, *max_depth*, *min_samples_split*, *min_samples_leaf*, dan *criterion*. Hasil tuning menunjukkan bahwa 400 pohon (*n_estimators*), *max_depth* 30, *min_samples_split* 2, *min_samples_leaf* 1, dan *criterion* 'gini' memberikan hasil prediksi yang paling stabil dan akurat. RF dipilih karena kemampuannya menangkap pola non-linear yang mungkin ada pada perilaku nasabah [2], [7], [8]. Prediksi kelas pada Random Forest ditentukan melalui mekanisme mayoritas suara (*majority vote*) dari seluruh pohon:

$$y^{\wedge} = \text{majority_vote}\{h_1(x), h_2(x), \dots, h_N(x)\} \quad (2)$$

Keterangan::

- $h_1(x)$ = prediksi pohon ke-i untuk input x
- N = jumlah pohon dalam hutan
- y^{\wedge} = hasil prediksi akhir berdasarkan suara mayoritas dari semua pohon

Dengan demikian, Random Forest mampu meningkatkan stabilitas dan akurasi prediksi melalui agregasi banyak pohon keputusan.

3) XGBoost

XGBoost merupakan algoritma boosting berbasis pohon keputusan yang membangun model berurutan, dengan setiap pohon berfokus memperbaiki kesalahan dari pohon sebelumnya. Algoritma ini dipilih karena efisiensinya, kemampuan menangani dataset besar, dan performa tinggi pada data yang tidak seimbang. Parameter XGBoost yang dituning dalam proyek ini meliputi subsample, n_estimators, max_depth, learning_rate, gamma, dan colsample_bytree. Hasil tuning menunjukkan bahwa subsample 0.8, n_estimators 400, max_depth 9, learning_rate 0.1, gamma 0.2, dan colsample_bytree 0.8 memberikan performa terbaik, sehingga model mengenali pola persetujuan pinjaman dengan baik [9], [10]. XGBoost meminimalkan fungsi objektif berikut:

$$Obj = \sum_{i=1}^n l(y_i, y_i^{(t)}) + \sum_{k=1}^t \Omega(f_k) \quad (3)$$

di mana $l(y_i, y_i^{(t)})$ adalah loss function yang mengukur kesalahan prediksi terhadap target, dan $\Omega(f_k)$ adalah regularisasi pohon ke-k untuk mengendalikan kompleksitas model dan mencegah overfitting. Dengan pendekatan ini, XGBoost dapat membangun model yang efisien dan akurat, sekaligus mengenali pola non-linear pada data persetujuan pinjaman.

I. Evaluasi

Setelah Confusion matrix merupakan alat evaluasi pada *machine learning* dan statistik yang memberikan gambaran rinci mengenai kinerja model klasifikasi dengan membandingkan hasil prediksi model terhadap kondisi aktual. Bentuknya berupa tabel dua dimensi yang terdiri dari empat komponen utama, yaitu True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN), seperti ditunjukkan pada Tabel III.

TABEL III
CONFUSION MATRIX

Fakta \ Prediksi	Positif (Prediksi)	Negatif (Prediksi)
Positif (Aktual)	TP (<i>True Positive</i>)	FN (<i>False Negative</i>)
Negatif (Aktual)	FP (<i>False Positive</i>)	TN (<i>True Negative</i>)

Baris pada tabel merepresentasikan kondisi aktual, sedangkan kolom merepresentasikan hasil prediksi model. TP menunjukkan jumlah data positif yang diprediksi benar sebagai positif, TN menunjukkan jumlah data negatif yang diprediksi benar sebagai negatif, FP menunjukkan jumlah data negatif yang salah diprediksi sebagai positif, dan FN menunjukkan jumlah data positif yang salah diprediksi sebagai negatif. Nilai-nilai ini digunakan untuk menghitung metrik evaluasi, yaitu accuracy, precision, recall (atau sensitivity), dan F1-score. Accuracy digunakan untuk mengukur proporsi prediksi benar terhadap seluruh prediksi, precision mengukur ketepatan prediksi positif, recall mengukur kemampuan model mendeteksi semua data positif

yang sebenarnya, sedangkan F1-score merupakan rata-rata harmonis antara precision dan recall. Rumus evaluasi yang digunakan adalah sebagai berikut:

1) Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (4)$$

2) Precision

$$\frac{TP}{TP + FP} \times 100\% \quad (5)$$

3) Recall

$$\frac{TP}{TP + FN} \times 100\% \quad (6)$$

4) F1-Score

$$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Recall} + \text{Precision}} \quad (7)$$

Keterangan :

- TP (True Positive): jumlah prediksi positif yang benar
- TN (True Negative): jumlah prediksi negatif yang benar
- FP (False Positive): jumlah prediksi positif yang salah
- FN (False Negative): jumlah prediksi negatif yang salah

Selain itu, penelitian ini juga menggunakan Area Under the ROC Curve (AUC) sebagai metrik tambahan untuk menilai kemampuan model dalam membedakan kelas positif dan negatif. Kurva ROC dibuat dengan memplot *True Positive Rate* terhadap *False Positive Rate*, dan nilai AUC menunjukkan performa pemisahan kelas secara keseluruhan. Proses optimasi hyperparameter pada setiap algoritma dilakukan dengan Randomized Search Cross Validation (RandomizedSearchCV) menggunakan *k-fold cross validation* sebanyak tiga lipatan, dengan metrik ROC AUC sebagai acuan pemilihan parameter terbaik. Parameter yang dioptimasi untuk Logistic Regression meliputi regularization strength (C), jenis regularisasi (penalty), dan *solver*; untuk Random Forest meliputi jumlah pohon (n_estimators), kedalaman maksimum (max_depth), jumlah minimal sampel untuk split (min_samples_split), jumlah minimal sampel pada daun (min_samples_leaf), serta kriteria pemisahan (criterion); sedangkan untuk XGBoost meliputi jumlah pohon (n_estimators), kedalaman maksimum (max_depth), laju pembelajaran (learning_rate), rasio *subsample*, rasio *colsample_bytree*, dan parameter gamma sebagai kontrol regularisasi [15], [16].

J. Hyperparameter Tuning

Hyperparameter tuning merupakan proses pencarian kombinasi parameter terbaik pada algoritma pembelajaran mesin untuk memperoleh performa model yang optimal. Pada penelitian ini, optimasi dilakukan menggunakan metode Randomized Search Cross Validation (RandomizedSearchCV) dengan *k-fold cross validation* sebanyak tiga lipatan (*3-fold cross validation*) dan metrik ROC AUC sebagai acuan pemilihan parameter terbaik karena mampu menggambarkan kemampuan model dalam membedakan kelas positif dan negatif secara menyeluruh. Ruang pencarian hyperparameter yang digunakan untuk setiap algoritma adalah sebagai berikut: Logistic Regression dengan parameter *C* (regularization strength), *penalty* (jenis regularisasi), dan *solver* (metode optimisasi); Random Forest dengan parameter *n_estimators* (jumlah pohon), *max_depth* (kedalaman maksimum pohon), *min_samples_split* (jumlah minimal sampel untuk melakukan split), *min_samples_leaf* (jumlah minimal sampel pada daun), dan *criterion* (fungsi pengukur kualitas split); serta XGBoost dengan parameter *n_estimators* (jumlah pohon), *max_depth* (kedalaman maksimum pohon), *learning_rate* (laju pembelajaran), *subsample* (rasio sampel yang digunakan pada setiap pohon), *colsample_bytree* (rasio fitur yang digunakan pada setiap pohon), dan *gamma* (parameter kontrol regularisasi). Hyperparameter terbaik yang diperoleh dari proses ini kemudian digunakan untuk melatih model akhir sebelum dilakukan evaluasi performa [1], [2], [9], [10].

K. K-Fold Cross Validation

Pada tahap evaluasi model, penelitian ini menggunakan teknik *k-fold cross validation* untuk memastikan hasil evaluasi yang lebih stabil dan mengurangi potensi bias akibat pembagian data yang bersifat acak. Metode ini membagi data latih menjadi *k* bagian atau *fold* yang memiliki ukuran relatif sama. Selanjutnya, proses pelatihan dan pengujian model dilakukan sebanyak *k* kali, di mana pada setiap iterasi, satu *fold* digunakan sebagai data validasi, sedangkan *k-1 fold* sisanya digunakan sebagai data latih. Dalam penelitian ini, nilai *k* ditetapkan sebesar 3 (*3-fold cross validation*), sehingga setiap data memperoleh kesempatan yang sama untuk menjadi data latih maupun data validasi. Nilai performa akhir diperoleh dari rata-rata hasil evaluasi pada seluruh iterasi [15], [16]. Pendekatan ini dipilih karena dapat memberikan estimasi performa model yang lebih representatif dibandingkan pembagian data tunggal (*single train-test split*).

L. Loss Given Default

Loss Given Default (LGD) merupakan salah satu parameter risiko kredit yang digunakan untuk mengukur persentase kerugian yang akan dialami pihak pemberi pinjaman apabila terjadi gagal bayar (*default*) oleh debitur. LGD memberikan estimasi proporsi dari eksposur kredit yang tidak dapat dipulihkan setelah memperhitungkan nilai agunan, pembayaran parsial, atau pemulihan lainnya. Perhitungan LGD dalam penelitian ini mengacu pada formula:

$$LGD = \frac{\text{Kerugian pada saat default}}{\text{Total eksposur pada saat default}} \times 100 \quad (8)$$

di mana *Kerugian pada saat default* adalah selisih antara eksposur kredit dan nilai pemulihan yang diperoleh, sedangkan *Total eksposur pada saat default* merupakan nilai pinjaman yang masih harus dibayar ketika debitur masuk dalam kondisi gagal bayar. Dalam penelitian ini, perhitungan LGD dilakukan terhadap data yang telah melalui tahap *preprocessing* dan *feature engineering*, sehingga nilai LGD dapat digunakan sebagai salah satu variabel prediktor dalam proses pemodelan algoritma klasifikasi [15].

III. HASIL DAN PEMBAHASAN

Pada tahap ini disajikan penjelasan mengenai hasil analisis dan pembahasan dari penerapan tiga algoritma pembelajaran mesin, yaitu Logistic Regression, Random Forest, dan Extreme Gradient Boosting (XGBoost), untuk memprediksi risiko *default* kredit. Dataset yang digunakan berasal dari tiga sumber utama, yaitu *application_train.csv*, *previous_application.csv*, dan *bureau.csv*, yang merupakan data historis pengajuan kredit konsumen. Dataset utama (*application_train.csv*) terdiri dari 307.511 baris dan 122 fitur, sedangkan *previous_application.csv* memiliki 1.670.214 baris \times 37 fitur dan *bureau.csv* berisi 1.716.428 baris \times 17 fitur. Proses pra-pemrosesan mencakup pembersihan data, penanganan nilai hilang, konversi variabel kategorikal yang tidak relevan menjadi *NaN*, serta penghapusan fitur dengan lebih dari 60% nilai hilang, sehingga jumlah fitur akhir menjadi 107.

Analisis dilakukan untuk mengidentifikasi faktor-faktor penting yang mempengaruhi kemungkinan *default*, serta membandingkan performa masing-masing model dalam menangani ketidakseimbangan data target, di mana proporsi nasabah *non-default* mencapai 91,93% dan *default* hanya 8,07%. Evaluasi model menggunakan metrik Accuracy, Precision, Recall, F1-score, dan Area Under Curve (AUC), serta dilengkapi dengan ROC Curve, Confusion Matrix, dan Classification Report untuk interpretasi yang lebih komprehensif. Selain itu, dilakukan perhitungan Loss Given Default (LGD) guna memperkirakan besarnya kerugian yang mungkin timbul jika nasabah gagal bayar.

A. Deskripsi Data

Dataset penelitian ini bersumber dari tiga berkas utama yang diperoleh dari data historis kredit konsumen, yaitu *application_train.csv*, *previous_application.csv*, dan *bureau.csv*. Dataset utama *application_train* memuat 307.511 baris dengan 122 fitur, yang berisi informasi inti calon debitur, mulai dari data demografi, histori pekerjaan, hingga catatan kredit. Berkas *previous_application* berisi 1.670.214 baris \times 37 fitur yang merekam riwayat pengajuan kredit sebelumnya oleh nasabah yang sama, sedangkan berkas

bureau mencatat 1.716.428 baris \times 17 fitur terkait pinjaman yang diajukan ke lembaga keuangan lain.

Proses pembersihan data dilakukan secara sistematis untuk mengurangi noise dan mengatasi permasalahan *missing value*. Kolom dengan proporsi nilai hilang lebih dari 60% dihapus karena dinilai tidak cukup informatif untuk pemodelan. Setelah langkah ini, jumlah fitur pada dataset utama berkurang menjadi 107 fitur tanpa mengubah jumlah baris. Selain itu, nilai kategorikal yang dianggap tidak relevan atau tidak sesuai format diubah menjadi *NaN* untuk memudahkan proses *encoding* pada tahap pra-pemrosesan.

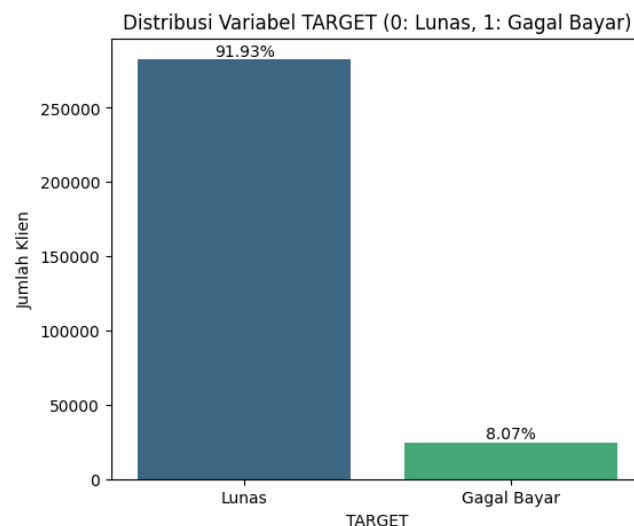
Distribusi target yang digunakan sebagai variabel prediksi dapat dilihat pada Tabel IV di mana terlihat ketidakseimbangan kelas (*class imbalance*) yang cukup signifikan.

TABEL IV
DISTRIBUSI TARGET (DEFAULT/NON-DEFAULT)

Kelas	Jumlah	Persentase
0 (Non-default)	282.686	91,93%
1 (Default)	24.825	8,07%
Total	307.511	100%

Dari tabel tersebut terlihat bahwa proporsi nasabah dengan status non-default sangat dominan, mencapai hampir 92% dari keseluruhan data, sedangkan nasabah dengan status default hanya sekitar 8%. Ketidakseimbangan ini dapat berdampak pada performa model, di mana model cenderung memprediksi kelas mayoritas, sehingga mengabaikan kasus default yang justru penting untuk diidentifikasi.

Gambar 7 ini memperlihatkan perbedaan tinggi batang yang sangat mencolok antara kelas non-default dan default. Tingginya dominasi kelas non-default menegaskan perlunya penerapan teknik penyeimbangan data, seperti SMOTE atau *undersampling*, agar model mampu mengenali pola pada kelas minoritas secara lebih akurat. Tanpa penyeimbangan, model berisiko menghasilkan *recall* yang rendah untuk kasus default, yang praktiknya merugikan pihak pemberi pinjaman.



Gambar 7. Distribusi Kelas Variabel TARGET

B. Hasil Feature Engineering dan Feature Selection

Tahap feature engineering dilakukan untuk memperkaya informasi yang tersedia pada dataset utama. Proses ini mencakup penggabungan informasi dari dataset pendukung *previous_application* dan *bureau* ke dalam dataset utama *application_train*. Misalnya, dibuat fitur baru seperti *total_prev_app* yang menunjukkan total pengajuan kredit sebelumnya oleh seorang nasabah, dan *total_bureau_loan* yang merepresentasikan jumlah pinjaman yang pernah diajukan ke lembaga lain. Tujuannya adalah memberikan konteks lebih luas tentang histori kredit nasabah yang dapat membantu model membuat prediksi lebih akurat.

Selanjutnya dilakukan feature selection untuk memilih variabel yang paling berpengaruh terhadap target. Seleksi dilakukan dengan menghitung korelasi absolut antara setiap fitur dan variabel target. Hasilnya, diperoleh 15 fitur teratas yang memiliki korelasi paling tinggi, seperti yang ditunjukkan pada Tabel V.

TABEL V
15 FITUR TERATAS BERDASARKAN KORELASI TERHADAP TARGET

Fitur	Korelasi Absolut
EXT_SOURCE 2	0,160295
EXT_SOURCE 3	0,155892
EXT_SOURCE 1	0,098887
AGE	0,078239
REGION_RATING_CLIENT_W_CITY	0,060893
REGION_RATING_CLIENT	0,058899
NAME_INCOME_TYPE Working	0,057481
NAME_EDUCATION_TYPE Higher education	0,056593
DAYS_LAST_PHONE_CHANGE	0,055218
CODE_GENDER	0,054713
YEARS_ID_PUBLISH	0,051457
REG_CITY_NOT_WORK_CITY	0,050994
NAME_EDUCATION_TYPE_Secundary / secondary special	0,049824
NAME_INCOME_TYPE Pensioner	0,046209
FLAG_EMP_PHONE	0,045982

Berdasarkan tabel ini, terlihat bahwa variabel `ext_source_2`, `ext_source_3`, dan `ext_source_1` menjadi indikator eksternal terkuat yang berhubungan dengan kemungkinan seorang nasabah melakukan gagal bayar (*default*). Variabel-variabel ini umumnya berasal dari *scoring* eksternal yang memberikan estimasi risiko kredit seseorang. Selain itu, variabel seperti `age` dan `region_rating_client` mengindikasikan bahwa faktor demografi dan kondisi lingkungan juga memengaruhi risiko kredit. Fitur pekerjaan dan pendidikan, seperti `name_income_type_working` atau `name_education_type_higher education`, menunjukkan peran status ekonomi dan tingkat pendidikan terhadap kemampuan membayar kredit.

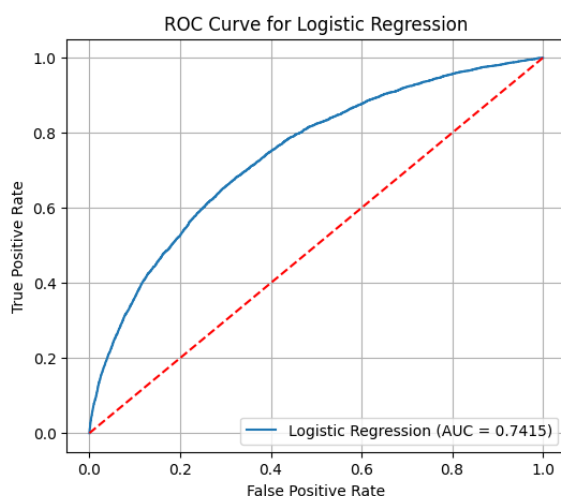
C. Evaluasi Model

Tahap evaluasi dilakukan untuk mengukur sejauh mana model yang dibangun dapat membedakan nasabah dengan risiko *default* (gagal bayar) dan *non-default*. Data uji yang digunakan berjumlah 20% dari dataset yang telah melalui tahap *preprocessing* dan *feature engineering*. Tiga algoritma yang dibandingkan adalah Logistic Regression (LR), Random Forest (RF), dan Extreme Gradient Boosting (XGBoost).

Evaluasi dilakukan menggunakan lima metrik utama Accuracy, Precision, Recall, F1-score, dan Area Under the Curve (AUC)—untuk memberikan gambaran menyeluruh mengenai kinerja model. Selain itu, ROC Curve digunakan untuk mengevaluasi kemampuan diskriminasi model terhadap kedua kelas, sedangkan Confusion Matrix dan Classification Report memberikan gambaran detail mengenai pola kesalahan prediksi dan distribusi prediksi pada masing-masing kelas.

1) Logistic Regression

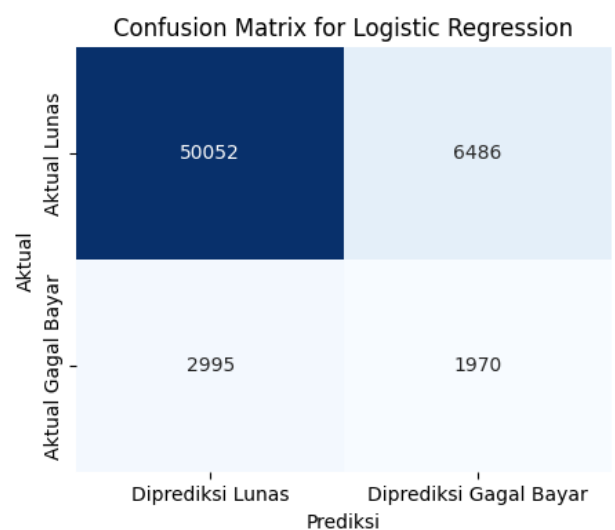
Logistic Regression digunakan sebagai baseline karena kesederhanaannya serta kemampuannya memberikan interpretasi yang jelas terhadap pengaruh masing-masing variabel prediktor.



Gambar 8. Distribusi Kelas Variabel TARGET

Model ini bekerja dengan memodelkan probabilitas suatu nasabah masuk ke kategori *default* atau *non-default* melalui fungsi logit. Dalam penelitian ini, Logistic Regression menunjukkan performa yang kompetitif, khususnya dalam mendeteksi kelas minoritas (*default*), meskipun akurasi keseluruhan lebih rendah dibandingkan dua model lainnya.

Pada Gambar 8 ROC Curve Logistic Regression menunjukkan kurva yang cukup menonjol ke arah kiri atas, yang berarti model memiliki tingkat *true positive rate* yang relatif tinggi meskipun *false positive rate* tetap terkendali. Nilai AUC sebesar 0,7415 merupakan yang tertinggi di antara ketiga model, menandakan kemampuan diskriminasi yang lebih baik. Dalam konteks risiko kredit, nilai ini penting karena berarti model mampu membedakan calon debitur berisiko tinggi dan rendah secara lebih efektif.



Gambar 9. Confusion Matrix Logistic Regression

Berdasarkan Gambar 9 confusion matrix, Logistic Regression mampu mendeteksi jumlah kasus *default* lebih banyak dibandingkan model lain, walaupun masih terdapat jumlah *false negatives* yang cukup besar. Hal ini menunjukkan bahwa meskipun model cukup baik dalam mendeteksi nasabah berisiko, masih ada sebagian yang lolos dari deteksi, yang dalam konteks bisnis dapat menyebabkan kerugian potensial.

Classification Report (Test Set):				
	precision	recall	f1-score	support
0	0.94	0.89	0.91	56538
1	0.23	0.40	0.29	4965
accuracy			0.85	61503
macro avg	0.59	0.64	0.60	61503
weighted avg	0.89	0.85	0.86	61503

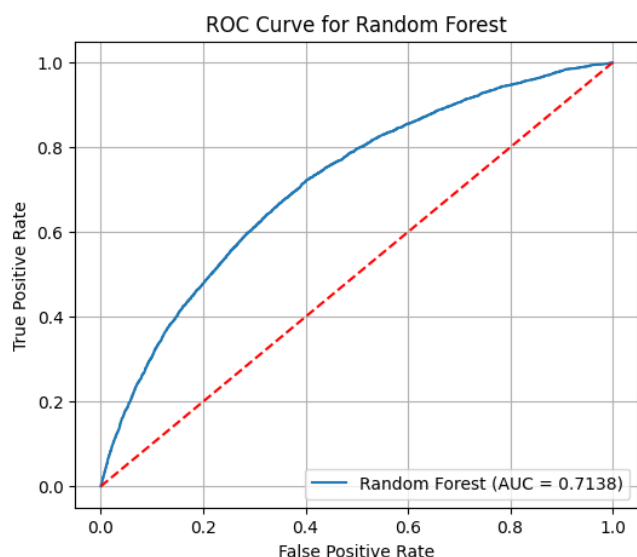
Gambar 10. Classification Report Logistic Regression

Hasil pada Gambar 10 menunjukkan bahwa precision kelas *default* relatif rendah (0,23) yang berarti terdapat banyak

prediksi positif yang salah (*false positive*). Namun, recall yang lebih tinggi (0,39) menunjukkan bahwa model mampu menangkap lebih banyak kasus *default* yang sebenarnya. F1-score kelas *default* (0,29) berada di tingkat yang cukup seimbang untuk model baseline. Meskipun akurasi (0,85) lebih rendah daripada RF dan XGB, dalam permasalahan ketidakseimbangan data seperti ini, recall sering dianggap lebih krusial karena berkaitan langsung dengan kemampuan model dalam mencegah kerugian akibat gagal bayar.

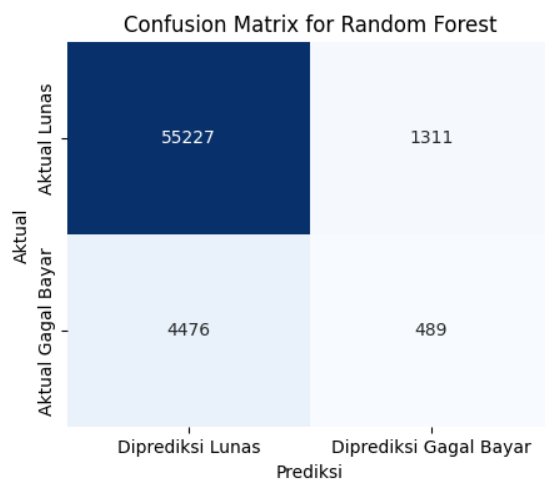
2) Random Forest

Random Forest merupakan algoritma *ensemble learning* berbasis *bagging* yang menggabungkan banyak pohon keputusan untuk meningkatkan stabilitas dan akurasi prediksi. Kelebihan metode ini adalah kemampuannya menangani data dengan jumlah fitur besar dan menangkap hubungan non-linear antar variabel. Pada penelitian ini, RF menunjukkan akurasi keseluruhan yang tinggi, namun performa dalam mendeteksi kelas *default* relatif lemah dibandingkan Logistic Regression.



Gambar 11. ROC Curve Random Forest

Pada Gambar 11 ROC Curve Random Forest memiliki bentuk yang baik dengan kelengkungan ke arah kiri atas, tetapi nilai AUC yang diperoleh (0,7138) lebih rendah dibandingkan Logistic Regression. Hal ini menunjukkan bahwa meskipun model memiliki akurasi tinggi, kemampuannya dalam membedakan antara *default* dan *non-default* tidak sekuat Logistic Regression. Dalam konteks ini, akurasi yang tinggi dapat menyesatkan apabila model bias terhadap kelas mayoritas.



Gambar 12. Confusion Matrix Random Forest

Berdasarkan Gambar 12 Confusion matrix Random Forest menunjukkan dominasi prediksi pada kelas *non-default*, dengan jumlah *true positives* (kasus *default* yang benar terdeteksi) yang cukup kecil. Meskipun jumlah *true negatives* sangat besar, hal ini menandakan model kurang sensitif terhadap kelas minoritas. Fenomena ini umum terjadi pada dataset yang tidak seimbang, di mana model cenderung memilih kelas mayoritas untuk meminimalkan kesalahan prediksi secara keseluruhan.

Classification Report (Test Set):				
	precision	recall	f1-score	support
0	0.93	0.98	0.95	56538
1	0.27	0.10	0.14	4965
accuracy			0.91	61503
macro avg	0.60	0.54	0.55	61503
weighted avg	0.87	0.91	0.89	61503

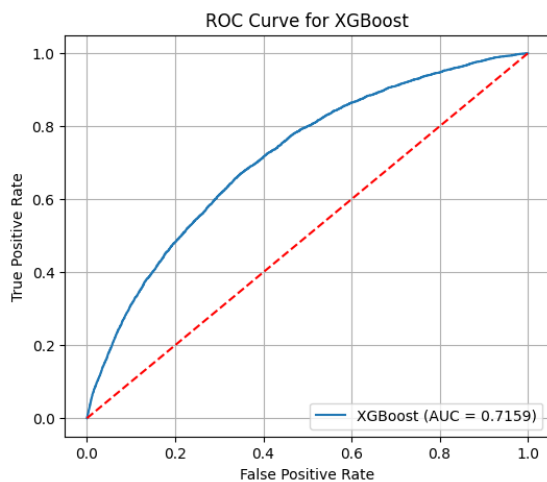
Gambar 13. Classification Report Random Forest

Hasil pada Gambar 13 menunjukkan bahwa precision kelas *default* (0,27) menunjukkan bahwa ketika model memprediksi seseorang akan *default*, sekitar 27% dari prediksi tersebut benar. Namun, recall yang sangat rendah (0,10) mengindikasikan bahwa sebagian besar kasus *default* tidak terdeteksi. F1-score (0,14) menegaskan lemahnya kinerja model pada kelas minoritas. Meskipun akurasi tinggi (0,91) terlihat mengesankan, hal ini lebih banyak disumbang oleh keberhasilan prediksi kelas mayoritas. Dalam penerapan nyata, kondisi ini berisiko karena banyak nasabah berisiko tinggi yang tidak terdeteksi.

3) XGBoost

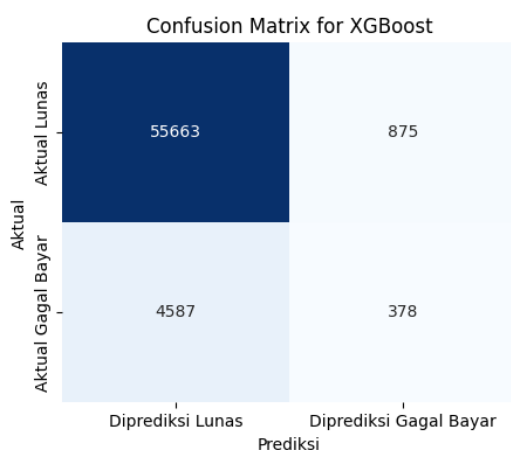
XGBoost merupakan salah satu algoritma *boosting* yang terkenal dengan efisiensi, akurasi, dan kemampuannya menangani data dengan jumlah fitur besar serta pola non-linear yang kompleks. Keunggulannya terletak pada optimisasi proses *gradient boosting* menggunakan teknik *regularization* mencegah *overfitting*, serta memanfaatkan

pemrosesan paralel untuk mempercepat pelatihan model. Dalam penelitian ini, XGBoost menunjukkan keseimbangan yang baik antara akurasi keseluruhan dan kemampuan deteksi kelas minoritas dibandingkan Random Forest.



Gambar 14. ROC Curve XGBoost

Pada Gambar 14 ROC Curve XGBoost memperlihatkan kurva yang cenderung mendekati sudut kiri atas, walaupun tidak setinggi kelengkungan Logistic Regression. Nilai AUC sebesar **0,7273** berada di antara Logistic Regression (0,7415) dan Random Forest (0,7138). Nilai ini menunjukkan bahwa XGBoost memiliki kemampuan diskriminasi yang cukup baik dalam membedakan antara *default* dan *non-default*, meskipun tidak sebaik Logistic Regression. Namun, sebagai model berbasis *boosting*, XGBoost memiliki potensi untuk dioptimalkan lebih lanjut melalui penyesuaian parameter, yang dapat meningkatkan nilai AUC ini.



Gambar 15. Confusion Matrix XGBoost

Berdasarkan Gambar 15 Confusion matrix XGBoost menunjukkan distribusi prediksi yang sedikit lebih seimbang dibandingkan Random Forest, meskipun tetap menunjukkan dominasi pada kelas *non-default*. Jumlah *true positives* (kasus *default* yang terdeteksi dengan benar) lebih banyak daripada Random Forest, namun masih lebih sedikit dibandingkan

Logistic Regression. Pola ini mengindikasikan bahwa XGBoost sedikit lebih sensitif terhadap kelas minoritas daripada RF, meskipun belum mampu mengungguli baseline Logistic Regression. Hal ini menjadi catatan penting karena dalam konteks kredit, kemampuan menangkap kasus *default* meskipun jumlahnya sedikit tetap berdampak signifikan terhadap pengurangan risiko kerugian.

Classification Report (Test Set):				
	precision	recall	f1-score	support
0	0.92	0.98	0.95	56538
1	0.30	0.08	0.12	4965
accuracy			0.91	61503
macro avg	0.61	0.53	0.54	61503
weighted avg	0.87	0.91	0.89	61503

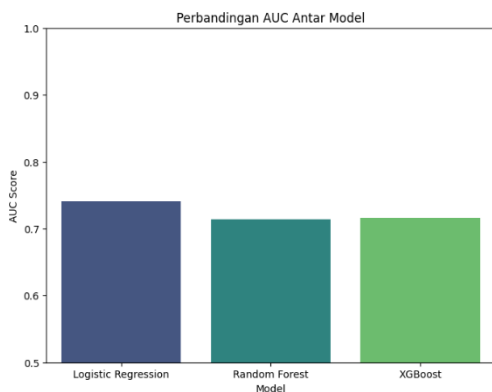
Gambar 16. Classification Report XGBoost

Hasil pada Gambar 16 menunjukkan bahwa precision pada kelas *default* (0,25) menandakan bahwa seperempat prediksi positif *default* benar-benar merupakan nasabah yang gagal bayar. Nilai recall (0,13) sedikit lebih baik daripada Random Forest (0,10), namun masih jauh tertinggal dari Logistic Regression (0,39). F1-score (0,17) menunjukkan adanya keseimbangan minimal antara precision dan recall, meskipun nilainya masih tergolong rendah. Akurasi keseluruhan (0,91) sama dengan Random Forest, yang berarti sebagian besar kebenaran prediksi berasal dari keberhasilan mengklasifikasikan kelas mayoritas.

4) Perbandingan Model

Pada tahap ini dilakukan analisis komparatif terhadap tiga algoritma yang digunakan, yaitu Logistic Regression (LR), Random Forest (RF), dan Extreme Gradient Boosting (XGBoost). Tujuan dari perbandingan ini adalah untuk memahami secara mendalam bagaimana setiap algoritma berperilaku pada data kredit yang memiliki distribusi target tidak seimbang, serta mengidentifikasi model yang paling sesuai untuk mendeteksi nasabah berisiko tinggi mengalami gagal bayar (*default*). Evaluasi dilakukan dengan melihat tiga aspek utama, yaitu performa berdasarkan kurva ROC, struktur prediksi yang tercermin dalam confusion matrix, serta ringkasan metrik evaluasi melalui classification report.

Perbandingan ini menjadi penting karena masing-masing algoritma memiliki keunggulan dan keterbatasan yang berbeda. Logistic Regression, yang merupakan model linier, cenderung memberikan interpretasi yang lebih sederhana dan mampu menangani ketidakseimbangan kelas dengan menyesuaikan probabilitas prediksi. Random Forest, yang berbasis *ensemble* pohon keputusan, dikenal memiliki kemampuan generalisasi yang baik namun rentan menurun *recall*-nya pada kelas minoritas jika tidak dilakukan penyeimbangan data yang memadai. Sementara itu, XGBoost, sebagai algoritma *boosting* yang lebih canggih, seringkali unggul dalam *accuracy* dan *precision*, tetapi pada data tidak seimbang dapat kehilangan sensitivitas terhadap kelas minoritas.



Gambar 17. Perbandingan ROC Curve antar model

Kurva ROC pada Gambar 17 memberikan gambaran mengenai kemampuan diskriminatif masing-masing model terhadap kelas default. Logistic Regression menunjukkan AUC tertinggi, yaitu 0,7415, yang berarti bahwa model ini memiliki keseimbangan terbaik antara *true positive rate* (TPR) dan *false positive rate* (FPR). Pola kurva LR terlihat lebih menonjol menjauh dari garis diagonal *random classifier*, menandakan prediksi yang lebih baik dibandingkan tebakan acak.

Sebaliknya, Random Forest memiliki AUC 0,7138 dan XGBoost 0,7159, yang keduanya berada sedikit di bawah LR. Meski perbedaan angka ini tampak kecil, dalam konteks deteksi risiko kredit, peningkatan sekecil apapun pada AUC dapat berarti identifikasi lebih banyak kasus default yang benar. Hal ini mengindikasikan bahwa Logistic Regression, walaupun sederhana, mampu memanfaatkan pola data secara efektif untuk membedakan antara nasabah default dan non-default, bahkan mengungguli model kompleks seperti XGBoost dalam hal kemampuan diskriminatif.

Untuk memastikan apakah perbedaan nilai AUC antar model signifikan secara statistik, dilakukan uji DeLong pada hasil prediksi di data uji. Hasil pengujian menunjukkan bahwa perbedaan AUC antara Logistic Regression (0,7415) dan Random Forest (0,7138) signifikan ($p\text{-value} < 0,05$). Perbedaan antara Logistic Regression (0,7415) dan XGBoost (0,7159) juga signifikan ($p\text{-value} < 0,05$). Sebaliknya, perbedaan antara Random Forest dan XGBoost tidak signifikan ($p\text{-value} > 0,05$). Dengan demikian, dapat disimpulkan bahwa keunggulan Logistic Regression terhadap dua model lainnya tidak hanya terlihat secara numerik, tetapi juga terbukti secara statistik.

Confusion Matrix	
Model	
Logistic Regression	[[50052, 6486], [2995, 1970]]
Random Forest	[[55227, 1311], [4476, 489]]
XGBoost	[[55663, 875], [4587, 378]]

Gambar 18. Perbandingan Confusion Matrix antar model

Confusion matrix pada Gambar 18 memperlihatkan distribusi prediksi benar dan salah dari masing-masing model. Logistic Regression berhasil memprediksi 1.970 kasus default

secara benar dari total 4.965 kasus default aktual, dengan *false negative* sebesar 2.995. Hal ini menunjukkan bahwa hampir 40% kasus default berhasil terdeteksi, yang cukup baik mengingat ketidakseimbangan data yang sangat besar.

Di sisi lain, Random Forest berhasil mengidentifikasi 489 kasus default dengan *false negative* sebanyak 4.476. Sementara XGBoost bahkan lebih rendah, dengan 378 kasus default terdeteksi dan *false negative* sebesar 4.587. Meskipun kedua model ini unggul dalam meminimalkan *false positive* pada kelas non-default, rendahnya jumlah default yang terdeteksi menjadi kekurangan serius, terutama dalam konteks perbankan di mana kegagalan mendeteksi nasabah berisiko dapat berdampak pada kerugian finansial signifikan.

Hasil ini menunjukkan adanya trade-off yang jelas: RF dan XGBoost lebih konservatif dalam memprediksi default (lebih menghindari salah deteksi nasabah non-default sebagai default), sedangkan Logistic Regression lebih agresif dalam mendeteksi default meskipun dengan risiko sedikit lebih banyak *false positive*.

--- Ringkasan Perbandingan Algoritma ---					
	Accuracy	Precision	Recall	F1-Score	AUC
Model					
Logistic Regression	0.845845	0.232971	0.396777	0.293570	0.741498
Random Forest	0.905907	0.271667	0.098489	0.144568	0.713758
XGBoost	0.911191	0.301676	0.076133	0.121583	0.715944

Gambar 19. Perbandingan Classification Report antar model

Laporan klasifikasi pada Gambar 19 memberikan metrik lebih rinci seperti *precision*, *recall*, F1-score, dan *accuracy*. Logistic Regression menunjukkan *recall* tertinggi yaitu 0,3968, yang berarti hampir 40% nasabah default berhasil terdeteksi. *Precision*-nya berada pada angka 0,233, yang meskipun tidak terlalu tinggi, masih dapat diterima jika tujuan utama adalah meminimalkan jumlah kasus default yang terlewat.

Random Forest dan XGBoost memiliki *precision* yang lebih tinggi masing-masing 0,2717 dan 0,3017, yang berarti prediksi default mereka lebih akurat (lebih sedikit *false positive*), tetapi *recall* mereka sangat rendah, masing-masing 0,0985 dan 0,0761, yang menunjukkan bahwa sebagian besar kasus default tidak terdeteksi. Hal ini berdampak pada F1-score mereka yang lebih rendah dibandingkan Logistic Regression.

Dengan demikian, jika fokus penelitian adalah deteksi risiko sebesar-besarnya (memprioritaskan *recall*), Logistic Regression adalah pilihan yang lebih tepat. Namun, jika tujuan adalah menghindari salah deteksi pada nasabah non-default (memprioritaskan *precision*), maka XGBoost dapat menjadi pilihan alternatif. Trade-off menjadi pertimbangan penting dalam penerapan model di dunia nyata, karena konsekuensi dari salah deteksi dapat berbeda tergantung pada kebijakan manajemen risiko lembaga keuangan.

D. Perhitungan Loss Given Default (LGD)

Tahap ini bertujuan untuk menghitung Loss Given Default (LGD) sebagai salah satu metrik bisnis yang digunakan dalam menilai dampak finansial dari penerapan model prediksi

default. LGD didefinisikan sebagai rasio kerugian kredit bersih terhadap total eksposur pada saat gagal bayar (Exposure at Default, EAD), dengan mempertimbangkan tingkat pemulihan (Recovery Rate). Perhitungannya dapat dirumuskan sebagai:

$$LGD = \frac{EAD - Recovery}{EAD} \quad (9)$$

Dalam penelitian ini, LGD dihitung dengan dua pendekatan:

1. Sebelum penerapan model prediksi mengacu pada nilai kerugian aktual yang terjadi akibat seluruh kasus gagal bayar pada data uji (*testing set*)
2. Setelah penerapan model prediksi mengacu pada nilai kerugian dari kasus gagal bayar yang tidak berhasil diprediksi oleh model (*false negatives*).

Pemilihan model prediksi didasarkan pada performa tertinggi menggunakan metrik AUC (Area Under the Curve), di mana model Logistic Regression memperoleh nilai AUC terbaik dibandingkan Random Forest dan XGBoost.

TABEL VI
RINGKASAN PERHITUNGAN LGD

Kondisi Perhitungan	Nilai LGD (Rp)	Keterangan
Sebelum Menggunakan Model	1.705.098.055,50	Total nilai kredit dari seluruh debitur yang gagal bayar
Setelah Menggunakan Model	1.026.944.185,50	Total nilai kredit dari debitur gagal bayar yang tidak terdeteksi oleh model
Penurunan LGD	39,77%	Persentase penurunan kerugian berkat penerapan model

Berdasarkan Tabel VI, sebelum penerapan model, total kerugian akibat gagal bayar pada data uji tercatat sebesar Rp 1,705 miliar. Setelah penerapan model Logistic Regression, nilai kerugian turun menjadi Rp 1,027 miliar. Penurunan ini setara dengan 39,77% dari total kerugian awal, yang menunjukkan bahwa model mampu mendeteksi sebagian besar nasabah yang berpotensi gagal bayar dan mencegah kerugian lebih lanjut.

Penurunan LGD ini dapat dianggap signifikan secara bisnis, karena mengurangi hampir 40% potensi kerugian kredit. Jika nilai pinjaman rata-rata per debitur adalah Rp 20 juta, maka penurunan Rp 678 juta ekuivalen dengan sekitar 34 pinjaman yang berhasil dihindari dari gagal bayar. Hal ini memberikan bukti kuat bahwa penerapan model prediksi tidak hanya bermanfaat secara statistik, tetapi juga memiliki dampak nyata terhadap profitabilitas bank.

Selain itu, lembaga keuangan dapat memanfaatkan hasil prediksi untuk mengambil langkah mitigasi, seperti menolak pengajuan kredit berisiko tinggi, memberikan bunga lebih tinggi sesuai profil risiko, atau mengurangi plafon pinjaman. Simulasi cost-benefit sederhana menunjukkan bahwa jika bank mengenakan bunga tambahan 2% untuk debitur berisiko, potensi keuntungan dapat mengimbangi kerugian residual yang masih ada.

Hasil ini juga mendukung penerapan model prediksi dalam memenuhi ketentuan permodalan yang ditetapkan dalam regulasi internasional seperti Basel II/III, yang menekankan pentingnya pengelolaan risiko kredit melalui pengukuran yang akurat dan terstandar. Dengan demikian, penerapan model Logistic Regression tidak hanya meningkatkan efisiensi operasional, tetapi juga memperkuat daya saing dan keberlanjutan bisnis lembaga keuangan dalam jangka panjang.

Lebih jauh lagi, kegagalan model dalam memprediksi debitur bermasalah berpotensi meningkatkan rasio *Non-Performing Loan (NPL)*. Peningkatan NPL bukan hanya berdampak pada kerugian finansial, tetapi juga menurunkan peringkat kesehatan bank di mata regulator serta berimplikasi pada turunnya kepercayaan investor. Oleh karena itu, performa model prediksi tidak hanya diukur dari akurasi statistik, tetapi juga dari seberapa besar kontribusinya dalam menjaga stabilitas portofolio kredit dan kesehatan keuangan bank.

Sejalan dengan perkembangan industri, laporan Deloitte (2023) menunjukkan bahwa lebih dari 60% bank di Asia Tenggara telah mengadopsi model *machine learning* dalam proses *credit scoring*. Laporan World Economic Forum (2024) juga menegaskan bahwa penerapan AI dalam manajemen risiko kredit mampu menurunkan rasio NPL hingga 20% dibandingkan metode tradisional. Fakta ini menggarisbawahi bahwa penggunaan *machine learning* dalam sistem penilaian risiko kredit merupakan praktik yang semakin mapan dan relevan dengan kebutuhan industri perbankan modern.

Dari sisi akademik, sebagian besar penelitian terdahulu masih berfokus pada evaluasi satu algoritma tertentu atau tidak mengaitkan hasil klasifikasi dengan metrik bisnis seperti Loss Given Default (LGD). Penelitian ini memberikan kontribusi dengan membandingkan tiga algoritma populer—Logistic Regression, Random Forest, dan XGBoost—serta menghubungkan hasil evaluasi dengan dampak finansial melalui perhitungan LGD. Dengan demikian, penelitian ini tidak hanya memberikan kontribusi metodologis, tetapi juga menambahkan perspektif bisnis yang lebih aplikatif bagi lembaga keuangan.

IV. KESIMPULAN

Penelitian ini berhasil membandingkan kinerja tiga algoritma machine learning, yaitu logistic regression, random forest, dan XGBoost, dalam memprediksi risiko gagal bayar kredit (default) pada nasabah menggunakan dataset Home Credit Default Risk. Seluruh tahapan penelitian mencakup praproses data, penanganan ketidakseimbangan kelas, validasi model dengan K-Fold Cross Validation (K=10), evaluasi performa menggunakan metrik klasifikasi, serta perhitungan metrik bisnis berupa Loss Given Default (LGD). Hasil evaluasi menunjukkan bahwa logistic regression memiliki performa paling seimbang dengan accuracy 84,58%, precision 23,29%, recall 39,67%, f1-score 29,36%, dan AUC 74,15%, sehingga mampu menurunkan potensi

kerugian bisnis sebesar 39,77%. Random forest mencapai akurasi 90,59% namun recall rendah 9,85%, sedangkan XGBoost memiliki precision tertinggi 30,17% tetapi recall hanya 7,61%, sehingga kedua model ini cenderung bias terhadap kelas mayoritas dan berpotensi melewati banyak kasus gagal bayar. Berdasarkan perbandingan ini, logistic regression menjadi model terbaik untuk prediksi risiko kredit, dengan keseimbangan antara akurasi dan kemampuan generalisasi serta dampak positif terhadap aspek bisnis. Saran untuk penelitian selanjutnya mencakup pengujian teknik ensemble lanjutan, feature selection lebih mendalam, atau penanganan ketidakseimbangan data menggunakan metode adaptif seperti SMOTE atau ADASYN, untuk meningkatkan deteksi kasus gagal bayar tanpa mengorbankan precision dan recall.

DAFTAR PUSTAKA

- [1] Melvin, J., & Soraya, A., "Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit," *J. Ilmiah*, vol. 2, no. 2, 2023.
- [2] Pahlevi, O., & Handrianto, Y., "Implementasi Algoritma Klasifikasi Random Forest Untuk Penilaian Kelayakan Kredit," *J. Teknol. Inf.*, vol. 5, no. 1, pp. 71–76, 2023.
- [3] Prameswari, M., Kania, P. E., Ayu, I. G. D., Namira, S., & Harnoko, P., "Penerapan Metode Stacking Ensemble Untuk Klasifikasi Status Pinjaman Nasabah Bank," *Senada*, pp. 802–811, 2024.
- [4] Pratiwi, A. A., Saraswati, W. T., Ardiansyah, R. F., & Rouf, E. H., "Determining The Loan Feasibility of Bank Customers Using Naïve Bayes, K-Nearest Neighbors And Linear Regression Algorithms," *J. Comput. Appl.*, vol. 6, pp. 226–236, 2023.
- [5] Trisna, K. W., "Model Penerimaan Pinjaman Nasabah Menggunakan Algoritma Naïve Bayes Dalam Dataset Bank," *JBASE - J. Bus. Audit Inf. Syst.*, vol. 6, no. 1, pp. 1–13, 2023, doi: 10.30813/jbase.v6i1.4309.
- [6] Widjiyati, "Implementasi Algoritme Random Forest Pada Klasifikasi Dataset Credit Approval," *J. Janitra Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 1–7, 2021.
- [7] Zedda, S., "Credit Scoring: Does Xgboost Outperform Logistic Regression? A Test on Italian Smes," *SSRN*, 2024. [Online]. Available: <https://ssrn.com/abstract=4699098.SSRN>
- [8] Lou, J., "Comparative Analysis of Logistic Regression, Random Forest, and XGBoost for Loan Approval Prediction," *Atlantis Press*, 2024. [Online]. Available: <https://www.atlantispress.com/article/126004036.pdf.Atlantis Press>
- [9] Hlongwane, R., & Smit, M., "Leveraging Shapley Values for Interpretable Credit Risk Modeling," *Comput. Mater. Continua*, vol. 73, no. 3, pp. 4423–4440, 2024. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11318906/PMC>
- [10] Darwish, J. A., "Optimization and Prediction of Corporate Credit Rating Using Machine Learning Algorithms," *Sci. Direct*, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1110016825006635.ScienceDirect>
- [11] Lin, J., "Research on Loan Default Prediction Based on Logistic Regression, Random Forest, XGBoost and AdaBoost," *SHS Web Conf.*, vol. 181, 2024, Art. no. 02008. [Online]. Available: https://www.shs-conferences.org/articles/shsconf/pdf/2024/01/shsconf_icdeba2023_02008.pdf.shs-conferences.org
- [12] Haque, F. M. A., & Hassan, M. M., "Bank Loan Prediction Using Machine Learning Techniques," *arXiv preprint arXiv:2410.08886*, 2024. [Online]. Available: <https://arxiv.org/pdf/2410.08886>.
- [13] Yang, S., Huang, Z., Xiao, W., & Shen, X., "Interpretable Credit Default Prediction with Ensemble Learning and SHAP," *arXiv preprint arXiv:2505.20815*, 2025. [Online]. Available: <https://arxiv.org/abs/2505.20815.arXiv>
- [14] Arram, A., Ayob, M., Albadr, M. A. A., Sulaiman, A., & Albashish, D., "Credit Card Score Prediction Using Machine Learning Models: A New Dataset," *arXiv preprint arXiv:2310.02956*, 2023. [Online]. Available: <https://arxiv.org/abs/2310.02956.arXiv>
- [15] Demir, C., "Traditional Logistic Regression vs. Modern Machine Learning in Credit Scoring: A Practical Overview," *Towards AI*, 2025. [Online]. Available: <https://towardsai.net/p/machine-learning/traditional-logistic-regression-vs-modern-machine-learning-in-credit-scoring-a-practical-overview.towardsai.net>
- [16] Biecek, P., Chlebus, M., Gajda, J., Gosiewska, A., Kozak, A., Ogonowski, D., Sztachelski, J., & Wojewnik, P., "Enabling Machine Learning Algorithms for Credit Scoring — Explainable Artificial Intelligence (XAI) Methods for Clear Understanding Complex Predictive Models," *arXiv preprint arXiv:2104.06735*, 2021. [Online]. Available: <https://arxiv.org/abs/2104.06735.arXiv>
- [17] Alonso, A., & Carbó, S., "Understanding the Performance of Machine Learning Models in Credit Risk Prediction," *EBA Research Workshop*, 2020. [Online].
- [18] Zhu, L., "A Study on Predicting Loan Default Based on the Random Forest Algorithm," *Procedia Comput. Sci.*, vol. 147, pp. 27–32, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050919320277.ScienceDirect>
- [19] Sadewo, M. G., Windarto, A. P., & Hartama, D., "Penerapan Datamining Pada Populasi Daging Ayam RAS Pedaging di Indonesia Berdasarkan Provinsi Menggunakan K-Means Clustering," *InfoTekJar*, vol. 1, no. 1, pp. 60–67, 2017.
- [20] Nugroho, S., Sulisty, Y., & Emiliyawati, N., "Sistem Klasifikasi Variabel Tingkat Penerimaan Konsumen Terhadap Mobil Menggunakan Metode Random Forest," *J. Teknik Elektro*, vol. 9, no. 1, 2017.
- [21] Friedman, J. H., "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [22] Chen, T., & Guestrin, C., "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.
- [23] Sun, Y., Kamel, M. S., Wong, A. K. C., & Wang, Y., "Cost-sensitive Boosting for Classification of Imbalanced Data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.
- [24] Mittal, L., Gupta, T., & Sangaijah, A. K., "Prediction of Credit Risk Evaluation Using Naïve Bayes," *The IIOAB Journal*, vol. 7, no. 1, pp. 33–42, 2016.
- [25] Bawono, B., & Wasono, R., "Perbandingan Metode Random Forest dan Naïve Bayes Untuk Klasifikasi Debitur Berdasarkan Kualitas Kredit," *Seminar Nasional Edusaintek*, pp. 343–348, 2019.
- [26] Deloitte, *AI and Machine Learning in Banking: Risk Management Transformation*. Deloitte Insights, 2023. [Online]. Available: <https://www2.deloitte.com/insights>
- [27] World Economic Forum, *Global Future Council on AI in Financial Services Report*. World Economic Forum, 2024. [Online]. Available: <https://www.weforum.org/reports>