**Jurnal Riset Rumpun Matematika dan Ilmu Pengetahuan Alam (JURRIMIPA)**
**Vol.2, No.2 Oktober 2023**
p-ISSN: 2828-9382; e-**ISSN:** 2828-9390-0143, Hal 87-103
DOI: https://doi.org/ 10.55606/jurrimipa.v2i2.1336

# Analisis Perbandingan Algoritma *XGBoost* dan Algoritma *Random Forest Ensemble Learning* pada Klasifikasi Keputusan Kredit

**Jan Melvin Ayu Soraya Dachi**
Universitas Negeri Medan
*Email: jan.melvin.ayu@mhs.unimed.ac.id*
**Pardomuan Sitompul**
Universitas Negeri Medan
*Email: ptmath.@unimed.co.id*

***Abstract***. *Providing credit always has risks such as bad credit, so creditors (banks) are required to be more objective and accurate in evaluating each credit application. This research is conducted to find which algorithm is most accurate in providing a credit decision, by comparing the XGBoost algorithm and the Random Forest algorithm. Both algorithms used data of size 10,000 and 100,000 with 19 variables that are relevant in making credit card decisions. The research process involves data pre-processing, data splitting, data training, parameter tuning with Random Search, data testing, and model evaluation with confusion matrix. The experimental results show that both algorithms produce quite competitive model performance, where XGBoost is able to achieve 1.0 for all evaluation metrics both on data size 10,000 and data size 100,000. Random Forest alone has an accuracy of 0.998 for data size 10,000 and 0.999 for data size 100,000. However, Random Forest is only able to achieve an F1-score of 0.700 for data of size 10,000. Based on the results obtained in this study, it can be concluded that both algorithms have excellent and accurate performance in classifying decisions on credit card data. However, Random Forest is less accurate when used on small-sized unbalanced data.*
***Keywords***: *XGBoost, Random Forest, Credit Decision Classification, Parameterized Tuning, Ensemble Learning*.

**Abstrak**. Pemberian kredit selalu memiliki risiko seperti kredit macet, sehingga pihak kreditur (bank) dituntut untuk lebih objektif dan akurat dalam mengevaluasi setiap permohonan kredit. Penelitian ini dilakukan guna menemukan algoritma mana yang paling akurat dalam memberikan suatu keputusan kredit, dengan melakukan perbandingan terhadap algoritma *XGBoost* dan algoritma *Random Forest*. Pada kedua algoritma digunakan data berukuran 10.000 dan 100.000 dengan 19 variabel yang relevan dalam pengambilan keputusan kartu kredit. Proses penelitian ini melibatkan *pre-processing data*, *splitting data*, *training data*, *parameter tuning* dengan *Random Search*, *testing data*, serta evaluasi model dengan *confusion matrix*. Hasil eksperimen menunjukkan bahwa kedua algoritma menghasilkan kinerja model yang cukup kompetitif, dimana *XGBoost* mampu mencapai 1.0 untuk semua metrik evaluasi baik pada data berukuran 10.000 maupun data berukuran 100.000. *Random Forest* sendiri berakurasi 0.998 untuk data berukuran 10.000 dan 0.999 untuk data berukuran 100.000. Akan tetapi, *Random Forest* hanya mampu mencapai *F1-score* sebesar 0.700 untuk data berukuran 10.000. Berdasarkan hasil yang diperoleh dalam penelitian ini, dapat disimpulkan bahwa kedua algoritma memiliki performa yang sangat baik dan akurat dalam mengklasifikasikan keputusan pada data kartu kredit. Namun, *Random Forest* kurang akurat bila digunakan pada data berukuran kecil yang tidak seimbang.

**Kata kunci**: *XGBoost*, *Random Forest*, Klasifikasi Keputusan Kredit, *Parameter Tuning*, *Ensemble Learning*.

*Received April 30, 2023; Revised Mei 30, 2023; Accepted Juni 15, 2023*
*Jan Melvin Ayu Soraya Dachi*, jan.melvin.ayu@mhs.unimed.ac.id

## LATAR BELAKANG

Dalam beberapa tahun terakhir, integrasi teknik Statistik dan *Machine Learning* terhadap bidang keuangan semakin mendalam sehingga telah menyebabkan banyak perubahan terhadap permasalahan industri tersebut khususnya dalam hal pendugaan (Y. Wang et al., 2020). *Machine Learning* sendiri adalah cabang ilmu komputer yang memanfaatkan data masa lalu untuk dipelajari dan menggunakan pengetahuannya tersebut untuk membuat keputusan dimasa depan (Dangeti, 2017).

*Ensemble Learning* adalah teknik kombinasi algoritma *Machine Learning* yang bertujuan untuk memperoleh hasil akurasi prediksi yang lebih tinggi dibandingkan dengan algoritma *Machine Learning* yang digunakan secara tunggal (Steinki & Mohammad, 2015). Selain meningkatkan akurasi dari sistem pengambilan keputusan, *Ensemble Learning* juga telah berhasil digunakan untuk mengatasi berbagai masalah *Machine Learning* seperti pemilihan fitur, estimasi kepercayaan, fitur yang hilang, pembelajaran tambahan, koreksi kesalahan, kelas dengan data yang tidak seimbang, dan lain-lain (Poliker, 2012). Dengan demikian *Ensemble Learning* telah terbukti sangat efektif dan sangat serbaguna dalam spektrum yang luas dari domain masalah dan aplikasi dunia nyata.

*Ensemble Learning* memiliki beberapa algoritma dalam masalah pengklasifikasian yang dapat digunakan dan disesuaikan dengan permasalahan yang dihadapi. Penelitian perbandingan algoritma yang diimplementasikan pada klasifikasi keputusan kredit ini, akan menggunakan algoritma *XGBoost* dan *Random Forest*. Hal ini dikarenakan dalam penelitian yang dilakukan oleh (Y. Li & Chen, 2020) yang berjudul "*A comparative performance assessment of ensemble learning for credit scoring*", *XGBoost* dan *Random Forest* telah diakui sebagai sebuah estimator canggih dengan kinerja yang sangat tinggi baik dalam klasifikasi maupun regresi. Mampu mencegah *overfitting*, hasil prediksi yang relatif tinggi terhadap data yang hilang dan data yang tidak seimbang merupakan kemampuan kedua algoritma ini.

Penelitian yang lainnya adalah milik Kui Wang, Meixuan Li, Jingyi Cheng, Xiaomeng Zhou dan Gang Li pada tahun 2021 tentang evaluasi risiko kredit pribadi berdasarkan *XGBoost* dengan 10.000 data kredit bank X serta membandingkan indeks evaluasi kinerja *XGBoost*, *Decision Tree* dan *K-Nearest Neighbor*. Penelitian tersebut menunjukkan *XGBoost* menghasilkan nilai akurasi yang tinggi kemudian disusul oleh *Decision Tree* dan terakhir oleh KNN (K. Wang et al., 2021). Sri Elina Herni Yulianti, Oni Soesanto dan Yuana Sukmawaty juga melakukan Penelitian dengan judul Penerapan Metode *Extreme Gradiant Boosting*

(*XGBoost*) pada Klasifikasi Nasabah Kartu Kredit. Sebanyak 30.000 data nasabah dengan 24 variabel dan 2 kelas keputusan, *XGBoost* menghasilkan akurasi sebesar 80,039% sehingga dinyatakan cukup baik dalam klasifikasi nasabah kartu kredit (Herni Yulianti et al., 2022).

Algoritma *Random Forest* sendiri juga telah sering digunakan sebelumnya dalam berbagai masalah klasifikasi kredit diantaranya penelitian yang dilakukan oleh Lingxiao Tang, Fei Cai dan Yao Ouyang pada tahun 2018 dengan judul "*Applying a nonparametric random forest algorithm to assess the credit risk of the energy industry in China*". Penelitian ini mengukur risiko kredit secara ilmiah dari kartu kredit yang digunakan dalam industri energi China berdasarkan analisis beberapa faktor yang memengaruhi risiko kredit tersebut. Penelitian tersebut menghasilkan akurasi prediksi sebesar 91,5% dan juga stabilitasnya memuaskan (Tang et al., 2018). Penelitian selanjutnya pada tahun 2020 yang dilakukan oleh Yuelin Wang, Yihan Zhang, Yan Lu dan Xinran Yu, *Random Forest* menghasilkan nilai akurasi sebesar 94,57% dibandingkan algoritma KNN, *Decision Tree*, *Nave Bayes* dan Regresi Logistik (Y. Wang et al., 2020).

Mengenai keputusan kredit, tentunya kehidupan manusia tidak pernah lepas dalam memberikan sebuah keputusan mulai dari permasalahan kecil hingga permasalahan yang lebih besar. Demikian halnya dalam dunia keuangan dan perbankan yang selalu dihadapkan pada pilihan untuk menerima permohonan kredit nasabah atau justru sebaliknya menolak permohonan tersebut. Kata kredit dalam artikel (Deppalallo et al., 2020) dinyatakan berasal dari kata *Credere* yang berarti percaya (*to trust*), maknanya bank memiliki kepercayaan terhadap nasabah untuk menggunakan kredit sebaik mungkin. Namun pada praktiknya, kredit yang diberikan kepada debitur selalu memiliki sebuah risiko seperti kredit macet (Arora & Kaur, 2020). Kredit macet dapat terjadi salah satunya dikarenakan kekurangmampuan pihak bank dalam menilai risiko calon debitur secara tepat. Selain pada dasarnya dalam klasifikasi keputusan kredit jumlah kelas pengguna (debitur) yang 'berisiko' dan pengguna (debitur) yang 'tidak berisiko' sering tidak seimbang, kekurangmampuan pihak bank dalam menilai risiko calon debitur juga dipengaruhi oleh faktor-faktor lainnya seperti data yang tidak lengkap dan data yang salah atau tidak akurat.

Untuk itu, kreditor dalam hal ini adalah pihak bank dituntut harus mampu mengevaluasi permohonan kredit secara objektif dan lebih akurat lagi agar tidak salah dalam memberikan keputusan pada setiap permohonan kredit nasabah. Selain itu, sebagian besar kekayaan bank adalah dalam bentuk kredit yang merupakan sumber pendapatan bank (Yu et al., 2018) sehingga bila kredit macet tidak dihindari dengan baik, bank akan mengalami kerugian bahkan

mungkin kebangkrutan seperti penelitian yang dilakukan oleh bank dunia pada tahun 1992. Berdasarakan uraian latar belakang masalah dan alasan pemilihan algoritma diatas penulis melakukan penelitian ini dengan judul ”Analisis Perbandingan Algoritma *XGBoost* dan Algoritma *Random Forest Ensemble Learning* pada Klasifikasi Keputusan Kredit” untuk membandingkan dua algoritma tersebut agar pihak kreditur (bank) dapat mengevaluasi permohonan kredit secara cepat dan lebih akurat.

## KAJIAN TEORITIS

Saat ini penilaian risiko kredit memainkan peran penting dan berguna dalam pengembangan industri keuangan dan perbankan, terlebih guna menghindari terjadinya kredit macet dan kerugian yang besar bagi institusi pemberi kredit seperti bank komersial dan retailer terkait (Yu et al., 2018). Kredit adalah pinjaman dana yang diberikan oleh bank atau lembaga keuangan lainnya kepada individu atau perusahaan yang membutuhkan dana, dan kartu kredit adalah suatu produk yang dikeluarkan oleh pihak bank atau lembaga keuangan lainnya sebagai alat pembayaran (Deppalallo et al., 2020). *Machine Learning* sendiri ada di mana-mana dalam kehidupan sehari-hari. Misalnya, penggunaan berbagai produk Google atau sistem transportasi umum tertentu (Wuest, 2015).

*Machine Learning* bertujuan untuk memecahkan masalah dengan menerapkan pengetahuan yang diperoleh dari analisis masalah (data) sebelumnya yang serupa dengan masalah yang akan dipecahkan. Berlandaskan tujuan tersebut, definisi *Machine Learning* menurut (Roihan et al., 2020) adalah aplikasi komputer dan algoritma matematika yang memanfaatkan data di masa lampau untuk dipelajari dan menggunakan pengetahuannnya untuk membuat keputusan di masa yang akan datang. *Machine Learning* secara luas terdiri dari tiga area permasalahan yakni klasifikasi, regresi dan *clustering* (Jo, 2021). Pemberian keputusan terhadap permohonan kredit dalam penelitian ini termasuk kedalam jenis permasalahan klasifikasi, tepatnya klasifikasi biner. Klasifikasi biner adalah jenis masalah dalam *Machine Learning* yang memerlukan prediksi terhadap sampel data yang telah dipisahkan menjadi dua kelas yang berbeda berdasarkan sejumlah atribut yang ada pada data tersebut.

(Nguyen et al., 2021) mengartikan *Ensemble Machine Learning* sebagai teknik yang menggabungkan beberapa model dasar *Machine Learning* (baik homogen maupun heterogen) untuk membuat prediksi yang lebih baik dengan mengurangi *noise* atau kesalahan antara data yang diamati dan data yang diprediksi. Metode *ensemble* biasanya dikelompokkan ke dalam metode *bootstrap aggregating* (*bagging*), *boosting*, dan *stacking*. Ketiga kategori tersebut

melakukan penyesuaian prediksi dengan observasi berdasarkan pengurangan varians model, bias, atau keduanya secara bersamaan. Perbedaan utamanya adalah bahwa *bagging* dan *boosting* biasanya bekerja dengan model yang homogen, sedangkan *stacking* unggul dalam mengkombinasikan model yang heterogen.

*eXtreme Gradient Boosting* atau yang disebut *XGBoost* adalah algoritma *Ensemble Learning* dengan metode *boosting* yang dikembangkan pada tahun 2014 oleh Tianqi Chen berdasarkan prinsip *gradient boosting* (Dangeti, 2017) yaitu semakin memfokuskan pada contoh yang salah diklasifikasikan oleh pengklasifikasi sebelumnya. Di dalam artikel milik (Zheng et al., 2017), pohon yang ditingkatkan dalam *XGBoost* dibagi menjadi dua yaitu pohon regresi dan klasifikasi. Selain itu, *XGBoost* telah diakui secara luas dalam kompetisi *Machine Learning* Kaggle dikarenakan keunggulannya dalam hal efisiensi yang tinggi dan fleksibilitas yang memadai (W. Zhang et al., 2021). Hal ini terbukti pada kompetisi *Machine Learning* Kaggle di tahun 2015 silam, *XGBoost* ditemukan sebagai metode yang paling populer dengan 17 solusi dari 29 solusi pemenang (D. Zhang et al., 2018).

*XGBoost* bertujuan untuk mencegah *overfitting* dan juga untuk mengoptimalkan kemampuan komputasi. Hal ini diperoleh dengan menyederhanakan fungsi objektif yang memungkinkan penggabungan istilah prediktif dan regularisasi (untuk mengendalikan kompleksitas model dan mencegah *overfitting*), namun demikian tetap mempertahankan kecepatan komputasi yang optimal (Fan et al., 2018). Dalam kasus klasifikasi biner, fungsi objektif menentukan bagaimana *XGBoost* mengukur kesalahan prediksi dan memperbarui model untuk meningkatkan akurasinya. Fungsi objektif sendiri terdiri dari dua bagian yakni fungsi kerugian (*loss function*) dan istilah regularisasi. Berlandaskan artikel milik (H. Li et al., 2020), fungsi kerugian yang akan digunakan dalam kasus ini adalah :

$$\min L^{(t)}(y, \hat{y}^{(t)}) = \min(\sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)} + (\sum_{k=1}^{t} \Omega(f_k))$$

Dimana $l^{(t)}(y_i, \hat{y}_i^{(t)})$ adalah fungsi kerugian dengan $y_i$ adalah nilai riil dan $\hat{y}_i^{(t)}$ adalah nilai prediksi. Sedangkan $\sum_{k=1}^{t} \Omega(f_k)$ adalah istilah regularisasi (penalti) dari model yang digunakan untuk mengukur kompleksitas keseluruhan model.

Untuk istilah regularisasi dapat ditentukan dengan rumus berikut:

$$\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda \sum_{j=1}^{T_k} \omega_{kj}^2$$

Keterangan:

$T_k$ : simpul daun pada pohon ke- $k$

$\gamma$ : koefisien pengurangan jumlah simpul daun $T$

$\omega_{kj}$ : nilai simpul daun ke- $j$ pada pohon ke- $k$

$\lambda$ : koefisien penalti dari nilai simpul daun $\omega$

*Random Forest* merupakan kumpulan dari beberapa pohon keputusan (*decision tree*) yang digunakan untuk membuat prediksi dengan memecahkan data menjadi beberapa kategori berdasarkan atribut tertentu dan membuat keputusan berdasarkan perbandingan nilai tertentu (Dangeti, 2017). Mengenai kategori dalam penelitian klasifikasi keputusan kredit ini kategori yang digunakan adalah pengguna yang 'berisiko' dan pengguna yang 'tidak berisiko'. (Breiman, 2001) mengatakan bahwa "*Random Forest* adalah sebuah pengklasifikasi yang terdiri dari kumpulan pohon pengklasifikasi yang terstruktur $\{h(x,\Theta_k), k = 1,\cdots\}$ dimana $\{\Theta_k\}$ adalah vektor acak independen yang terdistribusi secara identik dan setiap pohon akan memberikan satu suara untuk kelas yang paling populer pada input $x$". Dengan kata lain, *Random Forest* merupakan sekelompok pohon klasifikasi atau regresi yang belum dipangkas, yang dilatih pada *sampel bootstrap* dari data *training* menggunakan pemilihan fitur acak saat proses pembuatan pohon. Data *training* dalam algoritma ini dapat dirumuskan sebagai $S = \{(x_i, y_j)i = 1,2,\cdots,N; j = 1,2,\cdots,M\}$ dimana $x$ adalah sampel dan $y$ adalah variabel fitur $S$ (Religia et al., 2020).

Setelah sejumlah besar pohon dihasilkan, setiap pohon memilih kelas yang paling popular sebagai hasil prediksi kelas pada data *input* $x$ (Brown & Mues, 2012). Pemilihan kelas yang paling populer dapat menggunakan rumus berikut:

$$f(x) = Average(f_1(x), f_2(x), \cdots, f_n(x))$$

Keterangan:

$f(x)$ : hasil prediksi

$f_1(x), f_2(x), \cdots, f_n(x)$ : hasil prediksi dari setiap pohon keputusan

$x$ : *input*

Kekuatan dari masing-masing pohon dan korelasi antar pohon yang berbeda menentukan kapasitas dari algoritma *Random Forest*. Bila kekuatan pohon tunggal semakin besar dan korelasi antar pohon yang berbeda semakin kecil, maka semakin baik kinerja *Random Forest* (Xuan et al., 2018).

*Pre-processing data* adalah langkah penting untuk mencapai kinerja klasifikasi yang baik sebelum mengevaluasi data pada algoritma *Machine Learning*. Dalam penelitian ini, *pre-processing data* terbagi atas tiga bentuk yaitu *cleaning data*, *label encoding,* dan normalisasi data. *Cleaning data* adalah proses membersihkan dan mempersiapkan data yang akan digunakan dalam berbagai tugas *Machine Learning* yang bertujuan untuk meningkatkan kualitas data agar data yang akan digunakan dalam model *Machine Learning* sudah bersih, valid, dan relevan (Ganti & Sarma, 2013). *Cleaning data* dalam Penelitian ini merujuk pada *handling missing value*. Ada beberapa *handling missing value*, yaitu menghapus *missing value*, mengisi *missing value* dengan nilai rata-rata atau median, mengisi *missing value* dengan nilai modus, mengisi *missing value* dengan metode imputasi, dan mengisi *missing value* dengan informasi yang tersedia.

*Label encoding* merupakan suatu metode dalam *pre-processing data* untuk mengubah tipe data kategorik menjadi data numerik (Yustanti & Rochmawati, 2022). Singkatnya, metode ini mengonversi data teks secara langsung menjadi nilai integer yang memiliki makna nominal tanpa memperhatikan urutan atau tingkatan. Penggunaan *label encoding* dalam penelitian ini dikarenakan algoritma *XGBoost* dan algoritma *Random Forest* memerlukan data numerik sebagai input. Selain itu, kinerja model dari *XGBoost* dan *Random Forest* dapat ditingkatkan karena data kategorikal yang telah diubah menjadi data numerik efektif untuk digunakan dalam model.

Hal ini juga berkaitan dengan dimensi data yang berkurang sehingga mempermudah pelatihan model. Dalam penelitian ini terdapat 9 variabel kategorik yang akan diubah menjadi numerik pada tahap *label encoding*, yaitu Code Gender, Flag Own Car, Flag Own Realty, Cnt Children, Name Education Type, Name Family Status, Job, Name Housing Type, dan Status.

(Singh & Singh, 2020) menyatakan bahwa normalisasi data adalah salah satu pendekatan *pre-processing data* yang merupakan suatu tindakan pada data mentah yang mengubah ukuran atau mentransformasikan data tersebut sehingga setiap fitur memiliki kontribusi yang seragam. Dalam Bahasa sederhana, normalisasi data dapat dipahami sebagai tindakan yang mengubah variabel atau fitur pada data ke dalam skala tertentu agar variabel memiliki rentang nilai yang seimbang dan berskala sama sehingga dapat membantu dalam analisis dan pemodelan data. Banyak penulis telah memvalidasi dampak normalisasi data untuk meningkatkan kinerja klasifikasi diberbagai bidang, salah satunya untuk data klasifikasi persetujuan kredit oleh (Huang & Dun, 2008). Cheng-Lung menggunakan *min-max* untuk menskalakan fitur-fitur kedalam rentang [0, 1] dan ternyata normalisasi data sebelum menggunakan data sangat berdampak pada kinerja klasifikasi dan membantu dalam memprediksi model yang lebih akurat dari algoritma *Machine Learning*.

*Splitting data* adalah proses membagi dataset menjadi dua atau lebih bagian yang saling ekslusif (tidak dapat terjadi bersamaan) untuk melatih dan menguji model (Shmueli et al., 2020). Dari pengertian tersebut dapat diketahui bahwa splitting data dilakukan dengan tujuan untuk menghasilkan dataset yang dapat digunakan untuk menglatih model dan menguji kinerja model. Pada umumnya, dataset yang lebih besar akan dibagi menjadi dataset pelatihan (*training*) dan dataset pengujian (*testing*). Dalam *Machine Learning*, *splitting data* dilakukan sebelum melakukan *training* model untuk memastikan bahwa model yang dibangun memiliki kemampuan untuk melakukan generalisasi pada data yang belum dilihat sebelumnya. Dalam prakteknya, *train-test split* biasanya dilakukan dengan rasio 80:20 atau 70:30 antara data *training* dan data *testing*.

Evaluasi model atau proses mengukur seberapa baik kinerja model dalam memprediksi nilai target dari data *testing* pada penelitian ini dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* adalah tabel dua dimensi yang menunjukkan jumlah prediksi yang benar dan salah yang dibuat oleh model klasifikasi. Dengan kata lain, *Confusion matrix* memberikan informasi tentang keseimbangan antara kelas yang sebenarnya dan kelas yang diprediksi (Lewis & Brown, 2001). Metrik evaluasi yang akan dihitung dalam penelitian ini adalah akurasi, presisi, *recall*, dan *F1-score*.

Namun sebelum itu, saat membangun model algoritma *XGBoost* dan *Random Forest* sebaiknya dilakukan *parameter tuning* guna membuat model yang lebih baik dan meningkatkan kinerja model tersebut pada data uji. Menggunakan metode *Random Search* dalam penelitian ini, parameter terbaik untuk model *XGBoost* dan *Random Forest* dapat ditemukan. Beberapa parameter yang dapat diatur dalam penelitian ini untuk memperoleh kinerja model yang lebih baik pada data uji, yakni:

*a) XGBoost*

- *Learning Rate*, yaitu menentukan seberapa besar kontribusi setiap model ke model berikutnya dalam setiap iterasi.
- *Maximum Depth*, yaitu menentukan kedalaman maksimum setiap pohon keputusan dalam model.
- *Number of Trees*, yaitu menentukan jumlah pohon keputusan dalam model.
- *Subsampling*, yaitu menentukan seberapa banyak sampel yang digunakan dalam setiap iterasi dalam proses boosting.
- *Minimum Child Weight*, yaitu menentukan jumlah minimum sampel yang dibutuhkan di setiap cabang pohon keputusan.

*b) Random Forest*

- *Number of Trees*, yaitu menentukan jumlah pohon keputusan dalam model.
- *Maximum Depth*, yaitu menentukan kedalaman maksimum setiap pohon keputusan dalam model.
- *Minimum Sample Split*, yaitu menentukan jumlah minimum sampel yang dibutuhkan untuk membagi suatu node dalam pohon keputusan.
- *Minimum Sample Leaf*, yaitu menentukan jumlah minimum sampel yang dibutuhkan dalam suatu leaf node dalam pohon keputusan.
- *Maximum Features*, yaitu menentukan jumlah maksimum fitur yang digunakan dalam membangun setiap pohon keputusan.

## METODE PENELITIAN

Variabel dalam penelitian ini terdiri dari X sebagai variabel independen dan y sebagai variabel dependen. Dengan populasi data yaitu seluruh entri pada dataset "*Credit Card Approval - With Target*" yang berjumlah 537.668 dengan 19 variabel termasuk variabel target, dan sampel yang digunakan adalah sebesar 10.000 dan 100.000 data yang diambil secara acak dari populasi tersebut. Data yang digunakan merupakan data sekunder yang diunduh dari www.kaggle.com, platform kompetisi *data science* menggunakan perintah pengunduhan melalui API Kaggle. Data yang telah diunduh kemudian akan dianalisis menggunakan *software Python* dengan library *Scikit-learn* dan *XGBoost*. Model yang digunakan adalah model klasifikasi biner, dimana model tersebut akan melatih algoritma *XGBoost* dan *Random Forest* pada data *training* yang memiliki label keputusan kredit. Selanjutnya model yang dihasilkan akan digunakan untuk melakukan prediksi keputusan kredit pada data *testing*. Hasil prediksi akan dievaluasi menggunakan metrik evaluasi yaitu akurasi, presisi, *recall*, dan *F1-score*.

## HASIL DAN PEMBAHASAN

### Pengumpulan Data

Data yang digunakan dalam Penelitian ini diambil dari platform kaggle. Dataset yang digunakan adalah dataset klasifikasi keputusan kredit dengan judul "*Credit Card Approval - With Target*". Proses pengunduhan data dan pengolahan data dilakukan dalam rentang waktu kurang lebih 2 bulan yang dilaksanakan di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Medan.

### Analisis Data

Sebelum menganalisis data dengan menggunakan algoritma *XGBoost* dan Algoritma *Random Forest*, ada beberapa proses yang perlu dilakukan yaitu *pre-processing data*, *splitting data*, *training data*, dan *parameter tuning* dengan *Random Search*. Setelah melakukan proses-proses tersebut maka dilakukan analisis kinerja model dari algoritma *XGBoost* dan *Random Forest* pada data *testing* menggunakan evaluasi model dengan *confusion matrix*.

## 1. *Pre-processing Data*

*Pre-processing data* dalam penelitian ini terdiri atas 3 proses yaitu pengecekkan *missing value* serta melakukan *handling missing value* bila *missing value* ditemukan pada data yang digunakan, kemudian ada *label encoding* dan normalisasi data.

### a) *Missing Value*

Saat melakukan analisis data, terkadang ada beberapa data yang hilang (*missing value*) atau tidak lengkap sehingga *cleaning data* menjadi solusi dalam mengatasi masalah ini. Hasil pengecekkan *missing value* pada setiap kolom dalam dataset yang digunakan dalam penelitian ini menunjukkan bahwa tidak terdapat *missing value* sehingga tidak diperlukan langkah *handling missing value* pada data.

### b) *Label Encoding*

Salah satu metode *pre-processing data* ini dilakukan terhadap 9 variabel yang merupakan data dengan tipe kategorik dan akan diubah menjadi data numerik menggunakan *label encoder*. Kesembilan variabel tersebut adalah Code Gender, Flag Own Car, Flag Own Realty, Cnt Children, Name Education Type, Name Family Status, Name Housing Type, Job, dan Status.

| | ID | CODE_GENDER | FLAG_OWN_CAR | FLAG_OWN_REALTY | CNT_CHILDREN | AMT_INCOME_TOTAL | NAME_EDUCATION_TYPE | NAME_FAMILY_STATUS | NAMI |
|---|---|---|---|---|---|---|---|---|---|
| **56036** | 5021430 | 0 | 0 | 1 | 2 | 126000.0 | 1 | 3 | |
| **208386** | 5116206 | 1 | 1 | 1 | 1 | 112500.0 | 4 | 1 | |
| **64400** | 5023933 | 0 | 0 | 1 | 1 | 135000.0 | 4 | 2 | |
| **346812** | 5035422 | 0 | 0 | 1 | 2 | 292500.0 | 1 | 1 | |
| **100258** | 5050503 | 0 | 0 | 0 | 1 | 135000.0 | 4 | 1 | |

**Gambar 1. Hasil *label encoding* pada 10.000 data**

| | ID | CODE_GENDER | FLAG_OWN_CAR | FLAG_OWN_REALTY | CNT_CHILDREN | AMT_INCOME_TOTAL | NAME_EDUCATION_TYPE | NAME_FAMILY_STATUS | NAMI |
|---|---|---|---|---|---|---|---|---|---|
| **231836** | 5090287 | 1 | 1 | 0 | 1 | 450000.0 | 4 | 1 | |
| **335207** | 5053446 | 0 | 0 | 1 | 0 | 135000.0 | 1 | 2 | |
| **498678** | 5148965 | 0 | 0 | 1 | 2 | 90000.0 | 4 | 1 | |
| **482719** | 5047668 | 0 | 0 | 0 | 0 | 180000.0 | 1 | 1 | |
| **257500** | 5114551 | 0 | 0 | 1 | 0 | 135000.0 | 4 | 4 | |

**Gambar 2. Hasil *label encoding* pada 100.000 data**

**c) Normalisasi Data**

Normalisasi data atau tindakan yang mengubah variabel (fitur) pada data ke dalam skala tertentu agar variabel memiliki rentang nilai yang seimbang dan berskala sama, dilakukan setelah variabel target telah dipisah dari variabel lainnya. Dimana variabel lainnya sebagai X dan variabel target sebagai y. Sesudah pemisahan variabel tersebut, maka dilakukan normalisasi data menggunakan *min-max scaler*.

```
[[0.08910157 0.         0.         ... 0.82352941 0.98333333 0.71428571]
 [0.75804095 1.         1.         ... 0.47058824 0.81666667 0.85714286]
 [0.10676802 0.         0.         ... 0.17647059 0.68333333 0.85714286]
 ...
 [0.37808881 0.         0.         ... 0.         1.         0.        ]
 [0.19543905 1.         0.         ... 0.58823529 0.91666667 0.85714286]
 [0.68374023 0.         0.         ... 0.88235294 0.48333333 0.85714286]]
```

**Gambar 3. Hasil normalisasi data pada 10.000 data**

```
[[0.57510181 1.         1.         ... 0.47058824 0.35       0.        ]
 [0.315074   0.         0.         ... 0.         0.83333333 0.        ]
 [0.98925756 0.         0.         ... 0.47058824 0.91666667 0.        ]
 ...
 [0.23362342 0.         0.         ... 0.82352941 0.76666667 1.        ]
 [0.6160812  0.         0.         ... 0.         0.53333333 0.14285714]
 [0.29960969 1.         1.         ... 0.23529412 0.66666667 1.        ]]
```

**Gambar 4. Hasil normalisasi data pada 100.000 data**

**2. *Splitting Data***

*Splitting data* atau pembagian dataset dalam penelitian ini dilakukan dengan rasio 70:30, yaitu 70% untuk data *training* dan 30% untuk data *testing*. Setelah melakukan pembagian data, untuk memastikan perlu dilakukan beberapa hal yaitu pemeriksaan jumlah data dimasing-masing set dan pengecekkan apakah pembagian data sudah dilakukan secara acak dengan melihat 5 baris pertama dari X_train dan y_train serta X_test dan y_test.

**3. *Training Data* dan *Parameter Tuning***

Kegunaan *training data* adalah untuk membentuk model dari algoritma *XGBoost* dan *Random Forest* yang akan digunakan dalam penelitian ini dengan set data *training* sebagai *input*. Sesudah *training* perlu dilakukan *parameter tuning* untuk meningkatkan kinerja model yang telah dibentuk. Berikut beberapa parameter pada *XGBoost* yang di-*tuning* dalam penelitian ini menggunakan *Random Search*:

```
param_xgb = {
    'learning_rate': [0.1, 0.01, 0.001],
    'max_depth': [3, 5, 7],
    'n_estimators': [100, 200, 300],
    'subsample': [0.6, 0.8, 1.0],
    'min_child_weight': [1, 3, 5]
}
```

*Fitting* terhadap data *training* kemudian dilakukan menggunakan kelima parameter tersebut. Setelah menemukan parameter terbaik untuk model *XGBoost*, maka dilakukan *training* ulang menggunakan parameter terbaik. Parameter terbaik untuk model *XGBoost* yang telah ditemukan pada 10.000 data adalah '*learning_rate*' = 0.1, '*max_depth*' = 3, '*n_estimators*' = 300, '*subsample*' = 0.6, dan '*min_child_weight*' = 1. Sedangkan untuk 100.000 data parameter terbaik untuk model *XGBoost* yang ditemukan adalah '*learning_rate*' = 0.1, '*max_depth*' = 3, '*n_estimators*' = 300, '*subsample*' = 0.6, dan '*min_child_weight*' = 5.

Untuk *Random Forest* juga menggunakan *Random Search*, parameter yang di-*tuning* adalah:

```
param_rf = {
    'n_estimators': [100, 200, 300],
    'max_features': ['sqrt', 'log2'],
    'max_depth': [3, 5, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

Seperti halnya pada *XGBoost*, kelima parameter tersebut kemudian di-*fitting* terhadap data *training* dan dilakukan *training* ulang menggunakan parameter terbaik yang telah ditemukan. Parameter terbaik untuk model *Random Forest* pada 10.000 data adalah '*n_estimators*' = 200, '*max_features*' = sqrt, '*max_depth*' = None, '*min_samples_split*' = 2, dan '*min_samples_leaf*' = 1. Sedangkan pada 100.000 data, parameter terbaik untuk model *Random Forest* yang ditemukan adalah '*n_estimators*' = 100, '*max_features*' = log2, '*max_depth*' = None, '*min_samples_split*' = 2, dan '*min_samples_leaf*' = 2.

**Analisis Kinerja Model**

Evaluasi model dilakukan untuk mengukur seberapa baik kinerja model *XGBoost* dan *Random Forest* dalam memprediksi nilai target dari data *testing*. Sehingga sebelum melakukan evaluasi model, tentu perlu dilakukan *testing data* terlebih dahulu menggunakan set *testing* sebagai input pada kedua model. Hasil evaluasi model menggunakan *confusion matrix* yang diperoleh dari penelitian dapat dilihat dalam tabel berikut:

**Tabel 1. Hasil Evaluasi Model**

| **Metrik Evaluasi** | | **10.000 Data** | **100.000 Data** |
|---|---|---|---|
| Akurasi | *XGBoost* | 1.0 | 1.0 |
| | *Random Forest* | 0.998 | 0.999 |
| Presisi | *XGBoost* | 1.0 | 1.0 |
| | *Random Forest* | 1.0 | 1.0 |
| *Recall* | *XGBoost* | 1.0 | 1.0 |
| | *Random Forest* | 0.538 | 0.990 |
| *F1-score* | *XGBoost* | 1.0 | 1.0 |
| | *Random Forest* | 0.700 | 0.995 |

Berdasarkan tabel tersebut dapat dilihat bahwa kinerja model *XGBoost* lebih unggul dari model *Random Forest*. Dimana *XGBoost* mampu memberikan hasil kinerja yang konsisten dalam keadaan data yang bagaimanapun, dan hasil konsisten tersebut bernilai 1.0 yang berarti model *XGBoost* sangat akurat dan baik untuk digunakan untuk mengklasifikasikan keputusan kredit berbeda dengan model *Random Forest* yang kurang berkinerja baik pada data berjumlah kecil sekaligus tidak seimbang. Bisa dilihat dari tabel bahwa model *Random Forest* hanya mampu mencapai nilai *F1-score* sebesar 0.700 pada data yang berjumlah 10.000 data. Sedangkan pada data yang berjumlah besar yaitu 100.000, model *Random Forest* mampu mencapai nilai *F1-score* sebesar 0.995.

Tentunya hasil yang diperoleh dalam penelitian ini tidak sejalan dengan hasil penelitian sebelumnya yang dilakukan oleh Yiheng Li dan Weidong Chen pada tahun 2020. Pada artikel mereka yang berjudul "*A comparative performance assessment of ensemble learning for credit scoring*", disebutkan bahwa algoritma *Random Forest* mencapai akurasi yang tinggi dibandingkan dengan algoritma *XGBoost* walaupun dengan perbedaan yang tidak terlalu jauh (Y. Li & Chen, 2020).

## KESIMPULAN DAN SARAN

Keakuratan hasil kinerja model dari algoritma *XGBoost* dan *Random Forest* berdasarkan evaluasi model menggunakan *confusion matrix* pada klasifikasi keputusan kredit khususnya pada dataset *Credit Card Approval-With Target* menunjukkan bahwa model dari algoritma *XGBoost* memiliki keakuratan yang lebih tinggi dibandingkan model dari algoritma *Random Forest*. Sebenarnya algoritma *Random Forest* juga cukup baik untuk digunakan dalam mengklasifikasikan keputusan kredit, akan tetapi sebaiknya perlu dilakukan langkah-langkah seperti *oversampling* atau *undersampling*. Untuk itu, mengingat pentingnya kelas tidak seimbang dalam dataset pemberian keputusan kredit, disarankan untuk melakukan ekplorasi lebih lanjut dalam pemrosesan data untuk mengatasi ketidakseimbangan tersebut.

## DAFTAR REFERENSI

Arora, N., & Kaur, P. D. (2020). A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. *Applied Soft Computing Journal*, *86*, 1–28. https://doi.org/10.1016/j.asoc.2019.105936

Breiman, L. (2001). *Random Forests*. *45*(1), 5–32.

Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, *39*(3), 3446–3453. https://doi.org/10.1016/j.eswa.2011.09.033

Dangeti, P. (2017). *Statistics for Machine Learning: Build supervised, unsupervised, and reinforcement learning models using both Python and R* (Safis Editing (ed.)). Packt Publishing Ltd.

Deppalallo, H., Titaley, J., & Hatidja, D. (2020). *Penerapan Algoritma Naïve Bayes Untuk Klasifikasi*. *IV*, 127–140.

Fan, J., Wang, X., Wu, L., Zhou, H., Zhang, F., Yu, X., Lu, X., & Xiang, Y. (2018). Comparison of Support Vector Machine and Extreme Gradient Boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in China. *Energy Conversion and Management*, *164*, 102–111. https://doi.org/10.1016/j.enconman.2018.02.087

Ganti, V., & Sarma, A. Das. (2013). Data Cleaning: A Practical Perspective. In *Synthesis Lectures on Data Management* (Vol. 5, Issue 3). Morgan \& Claypool. https://doi.org/10.2200/s00523ed1v01y201307dtm036

Herni Yulianti, S. E., Oni Soesanto, & Yuana Sukmawaty. (2022). Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit. *Journal of Mathematics Theory and Application*, *4*(1), 21–26. https://doi.org/10.31605/jomta.v4i1.1792

Huang, C. L., & Dun, J. F. (2008). A distributed PSO-SVM hybrid system with feature selection and parameter optimization. *Applied Soft Computing Journal*, *8*(4), 1381–1391. https://doi.org/10.1016/j.asoc.2007.10.007

Jo, T. (2021). Machine Learning Foundations. In *Machine Learning Foundations*. Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-030-65900-4

Lewis, H. G., & Brown, M. (2001). A generalized confusion matrix for assessing area estimates from remotely sensed data. *International Journal of Remote Sensing*, *22*(16), 3223–3235. https://doi.org/10.1080/01431160152558332

Li, H., Cao, Y., Li, S., Zhao, J., & Sun, Y. (2020). XGBoost Model and Its Application to Personal Credit Evaluation. *IEEE Intelligent Systems*, *35*(3), 1–8. https://doi.org/10.1109/MIS.2020.2972533

Li, Y., & Chen, W. (2020). A comparative performance assessment of ensemble learning for credit scoring. *Mathematics*, *8*(10), 1–19. https://doi.org/10.3390/math8101756

Nguyen, K. A., Chen, W., Lin, B. S., & Seeboonruang, U. (2021). Comparison of Ensemble Machine Learning Methods for Soil Erosion Pin Measurements. *ISPRS International Journal of Geo-Information*, *10*(1), 1–17. https://doi.org/10.3390/ijgi10010042

Poliker, R. (2012). *Ensemble Machine Learning: Methods and Applications* (C. Zhang & Y. Ma (eds.)). Springer Science+Business Media. https://doi.org/10.1007/978-1-4419-9326-7

Religia, Y., Pranoto, G. T., & Santosa, E. D. (2020). South German Credit Data Classification Using Random Forest Algorithm to Predict Bank Credit Receipts. *JISA(Jurnal Informatika Dan Sains)*, *3*(2), 62–66. https://doi.org/10.31326/jisa.v3i2.837

Roihan, A., Sunarya, P. A., & Rafika, A. S. (2020). Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*, *5*(1), 75–82. https://doi.org/10.31294/ijcit.v5i1.7951

Shmueli, G., Bruce, P. C., Yahav, I., Patel, N. R., & Lichtendahl, K. C. (2020). *Data mining for Business Analytics: Concepts, Techniques, and Applications in R* (3rd ed.). John Wiley \& Sons, Inc.

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 1–23. https://doi.org/10.1016/j.asoc.2019.105524

Steinki, O., & Mohammad, Z. (2015). Introduction to Ensemble Learning. *SSRN Electronic Journal*, *1*(1), 1–9. https://doi.org/10.2139/ssrn.2634092

Tang, L., Cai, F., & Ouyang, Y. (2018). Applying a nonparametric random forest algorithm to assess the credit risk of the energy industry in China. *Technological Forecasting and Social Change*, *144*, 1–10. https://doi.org/10.1016/j.techfore.2018.03.007

Wang, K., Li, M., Cheng, J., Zhou, X., & Li, G. (2021). Research on personal credit risk evaluation based on XGBoost. *Procedia Computer Science*, *199*, 1128–1135. https://doi.org/10.1016/j.procs.2022.01.143

Wang, Y., Zhang, Y., Lu, Y., & Yu, X. (2020). A Comparative Assessment of Credit Risk Model Based on Machine Learning ——a case study of bank loan data. *Procedia Computer Science*, *174*, 141–149. https://doi.org/10.1016/j.procs.2020.06.069

Wuest, T. (2015). *Identifying Product and Process State Drivers in Manufacturing Systems Using Supervised Machine Learning* (P. D.-I. Habil (ed.)). Springer Theses. https://doi.org/10.1007/978-3-319-17611-6

Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., & Jiang, C. (2018). Random Forest for Credit Card Fraud Detection Shiyang. *Procedia Computer Science*, *4*(1), 80–86.

Yu, L., Zhou, R., Tang, L., & Chen, R. (2018). A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data. *Applied Soft Computing Journal*, *69*, 192–202. https://doi.org/10.1016/j.asoc.2018.04.049

Yustanti, W., & Rochmawati, N. (2022). Analisis Algoritma Klasifikasi untuk Memprediksi Karakteristik Mahasiswa pada Pembelajaran Daring. *JEPIN (Jurnal Edukasi Dan Penelitian Informatika)*, *8*(1), 57–61.

Zhang, D., Qian, L., Mao, B., Huang, C., Huang, B., & Si, Y. (2018). A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost. *IEEE Access*, *6*, 21020–21031. https://doi.org/10.1109/ACCESS.2018.2818678

Zhang, W., Wu, C., Zhong, H., Li, Y., & Wang, L. (2021). Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization. *Geoscience Frontiers*, *12*(1), 469–477. https://doi.org/10.1016/j.gsf.2020.03.007

Zheng, H., Yuan, J., & Chen, L. (2017). Short-Term Load Forecasting Using EMD-LSTM neural networks with a xgboost algorithm for feature importance evaluation. *Energies*, *10*(8). https://doi.org/10.3390/en10081168