Department of Digital Business
**Journal of Artificial Intelligence and Digital Business (RIGGS)**
Homepage: https://journal.ilmudata.co.id/index.php/RIGGS
Vol. 5 No. 2 (2026) pp: 17111-17124
P-ISSN: 2963-9298, e-ISSN: 2963-914X

# Komparasi Algoritma *Random Forest* dan *XGBoost* dalam Klasifikasi Risiko Perizinan Usaha pada UMKM

Rinda Fitriana Azmi[1]*, Galih Hendro Martono[2], Neny Suliastianingsih[3]
[1]Program Studi S1 Ilmu Komputer, Fakultas Teknik, Universitas Bumigora,
[2,3]Program Studi S2 Ilmu Komputer, Program Pasca Sarjana, Universitas Bumigora
[1]rrindafitrianaazmi@gmail.com*, [2]galih.hendro@universitasbumigora.ac.id,
[3]neny.sulistianingsih@universitasbumigora.ac.id

***Abstrak***

*Tingkat risiko perizinan usaha merupakan aspek penting dalam menentukan legalitas dan kewajiban perizinan yang harus dipenuhi oleh pelaku UMKM. Penelitian ini bertujuan membandingkan kinerja algoritma Random Forest dan XGBoost dalam mengklasifikasikan tingkat risiko perizinan usaha UMKM. Dataset yang digunakan berjumlah 4.081 data dengan variabel investasi, omset, jumlah tenaga kerja, skala usaha, luas tanah, dan risiko usaha sebagai label klasifikasi. Tahapan penelitian meliputi pengumpulan data, eksplorasi data, preprocessing, pembagian data, pemodelan menggunakan Random Forest dan XGBoost , evaluasi model, serta pengujian teknik resampling untuk menangani ketidakseimbangan kelas. Pengujian dilakukan pada model dengan parameter default dan model yang telah dioptimasi melalui hyperparameter tuning. Hasil evaluasi menunjukkan bahwa Random Forest dengan hyperparameter tuning memperoleh performa terbaik dengan accuracy sebesar 94%, precision macro sebesar 87%, recall macro sebesar 78%, dan F1-score macro sebesar 82%. Sementara itu, XGBoost dengan hyperparameter tuning memperoleh accuracy sebesar 91%, precision macro sebesar 81%, recall macro sebesar 79%, dan F1-score macro sebesar 79%. Hasil tersebut menunjukkan bahwa Random Forest dengan hyperparameter tuning tanpa resampling menjadi model paling stabil dalam mengklasifikasikan tingkat risiko perizinan usaha UMKM. Penelitian ini diharapkan dapat mendukung proses identifikasi risiko usaha secara lebih sistematis, objektif, dan berbasis data.*

*Kata kunci: UMKM, Risiko Usaha, Random Forest, XGBoost, Klasifikasi*

## 1. Latar Belakang

Tingkat risiko usaha menjadi salah satu dasar penting dalam menentukan jenis legalitas dan kewajiban perizinan yang harus dipenuhi oleh pelaku usaha [1]. Namun, masih banyak pelaku UMKM yang belum memahami cara menentukan tingkat risiko usahanya, sehingga berpotensi mengalami kendala dalam proses perizinan, bahkan dapat menghadapi sanksi seperti pencabutan izin hingga larangan menjalankan usaha [2]. Dalam kondisi tersebut, *machine learning* menjadi pendekatan yang efektif untuk membantu proses penentuan tingkat risiko usaha UMKM. Melalui teknik klasifikasi, data usaha dapat dikelompokkan secara otomatis berdasarkan pola dari atribut atau variabel yang tersedia, sehingga proses klasifikasi risiko usaha dapat dilakukan secara lebih konsisten dan dapat membantu UMKM menentukan jalur legalitas yang sesuai dengan karakteristik usahanya.

*Machine learning* merupakan bagian dari kecerdasan buatan yang memanfaatkan data dan algoritma agar sistem mampu belajar dari pola data serta menghasilkan keputusan menyerupai cara manusia [3]. Semakin baik algoritma yang digunakan, maka semakin akurat pula keputusan yang dihasilkan. Oleh karena itu, *machine learning* relevan diterapkan dalam membangun model klasifikasi yang mampu memetakan UMKM ke dalam kategori risiko usaha tertentu berdasarkan variabel dari *dataset* UMKM. Dalam kasus klasifikasi, algoritma *machine learning* yang banyak digunakan dalam tugas klasifikasi adalah *Random Forest* dan *XGBoost* [4].

Dalam permasalahan klasifikasi, distribusi kelas yang tidak seimbang dapat menyebabkan model lebih cenderung mempelajari kelas mayoritas, sehingga kelas minoritas kurang terwakili dalam proses prediksi [5]. Oleh karena itu, diperlukan teknik penyeimbangan data agar representasi kelas minoritas dapat ditingkatkan. Teknik penyeimbangan data tersebut dapat dilakukan melalui pendekatan *resampling*, seperti *oversampling*, *undersampling*, maupun metode *hybrid* untuk mengatasi permasalahan data tidak seimbang serta meningkatkan metrik evaluasi seperti *accuracy*, *precision*, dan *recall* [6].

*Random Forest* merupakan algoritma klasifikasi berbasis *ensemble learning* yang dibangun dari beberapa pohon keputusan atau *decision tree*. Algoritma ini bekerja dengan membentuk sejumlah pohon prediksi melalui proses *bootstrap sampling*, kemudian hasil prediksi dari setiap pohon digabungkan menggunakan mekanisme *majority vote* untuk menentukan kelas akhir pada kasus klasifikasi[7]. *Random Forest* banyak digunakan karena mampu menangani data dalam jumlah besar, menghasilkan akurasi yang baik, serta dapat mengurangi risiko *overfitting* [8]. Sementara itu, *XGBoost* merupakan algoritma *boosting* yang bekerja dengan membangun model secara bertahap untuk memperbaiki kesalahan prediksi sebelumnya, sehingga sering menghasilkan performa klasifikasi yang baik pada berbagai jenis data[9]. Kedua algoritma tersebut memiliki karakteristik yang berbeda, sehingga perlu dilakukan pengujian untuk mengetahui algoritma mana yang memberikan hasil klasifikasi paling optimal.

Beberapa penelitian sebelumnya juga melakukan klasifikasi data UMKM seperti penelitian oleh Castaka Agus Sugianto et al. [10] menggunakan algoritma Naïve Bayes untuk mengklasifikasikan kelayakan penerima Bantuan Langsung Tunai bagi UMKM di Kota Cimahi. Penelitian tersebut menggunakan data pelaku UMKM dari Dinas Sosial Kota Cimahi dan membagi kelas menjadi Layak dan Tidak Layak. Hasil pengujian menunjukkan bahwa algoritma Naïve Bayes memperoleh akurasi 100%, presisi 100%, *recal*l 100%, dan AUC 1,00. Meskipun demikian, Naïve Bayes memiliki keterbatasan karena menggunakan asumsi independensi antaratribut, sehingga performanya berpotensi menurun apabila terdapat hubungan atau korelasi antarvariabel. Selain itu, hasil akurasi yang sangat tinggi mengindikasikan terjadinya *overfitting*.

Penelitian oleh Royani et al. [11] melakukan klasifikasi tingkat digitalisasi UMKM di Kabupaten Tegal menggunakan algoritma C4.5 dan *Random Forest*. Hasil penelitian menunjukkan bahwa *Random Forest* memperoleh performa lebih baik dengan nilai akurasi sebesar 91,43% dan F1-*score* sebesar 89,63%, sedangkan C4.5 memperoleh akurasi sebesar 90% dan F1-*score* sebesar 86,24%. Hasil tersebut menunjukkan bahwa algoritma berbasis pohon keputusan mampu digunakan untuk memetakan kategori digitalisasi UMKM secara efektif. Namun, penelitian tersebut memiliki keterbatasan dengan jumlah data yang digunakan hanya 100 responden, sehingga kemampuan generalisasi model terhadap data UMKM yang lebih besar dan beragam masih perlu diuji lebih lanjut.

Penelitian oleh Saepuloh et al. [12] melakukan prediksi keberhasilan Usaha Kecil Menengah menggunakan algoritma Credal C4.5 dan Credal *Decision tree*. Penelitian tersebut menggunakan *dataset* sebanyak 1.200 UKM pada sektor kuliner dan ritel di beberapa wilayah Indonesia dengan variabel modal awal, pengalaman wirausaha, strategi pemasaran, dan lokasi usaha. Hasil penelitian menunjukkan bahwa algoritma Credal C4.5 memperoleh performa lebih baik dengan akurasi sebesar 88,6%, presisi 0,89, *recal*l 0,87, dan F1-*score* 0,88 dibandingkan Credal *Decision tree* yang memperoleh akurasi sebesar 84,2%, presisi 0,85, *recal*l 0,83, dan F1-*score* 0,84. Temuan tersebut menunjukkan bahwa algoritma berbasis pohon keputusan mampu digunakan untuk memprediksi keberhasilan UKM berdasarkan faktor internal dan strategi usaha. Penelitian mendatang disarankan untuk memperluas *dataset* ke sektor lain dan menambahkan serta membandingkan dengan algoritma modern seperti *Random Forest* atau *Gradient boosting*.

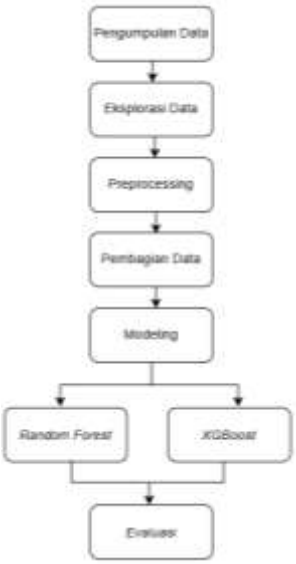
Berdasarkan keterbatasan pada penelitian terdahulu, penelitian ini menggunakan algoritma *Random Forest* dan *XGBoost* (*Extreme Gradient Boosting*) sebagai metode klasifikasi. *Random Forest* digunakan karena memiliki kemampuan yang stabil dalam menangani data tabular, relatif tahan terhadap noise, serta mampu mengenali hubungan non-linear antar variabel [8]. Sementara itu, *XGBoost* digunakan karena memiliki mekanisme *boosting* yang dapat memperbaiki kesalahan prediksi secara bertahap, sehingga efektif dalam mempelajari pola data yang kompleks dan sering menghasilkan performa tinggi pada kasus klasifikasi data tabular [13].

Oleh karena itu, penelitian ini berupaya membandingkan kinerja algoritma *Random Forest* dan *XGBoost* dalam mengklasifikasikan tingkat risiko usaha UMKM berdasarkan data yang diperoleh dari salah satu dinas PLUT KUMKM. Kedua algoritma tersebut diuji dan dibandingkan menggunakan metrik *accuracy*, *precision*, *recal*l, dan F1-*score* untuk menentukan model dengan performa terbaik. Selain itu, penelitian ini juga menerapkan teknik penyeimbangan data untuk menangani ketidakseimbangan kelas, sehingga model tidak hanya dominan mempelajari kelas mayoritas, tetapi juga mampu mengenali kelas minoritas secara lebih baik. Hasil klasifikasi ini diharapkan dapat membantu pihak PLUT KUMKM dalam memberikan gambaran awal mengenai kategori risiko usaha, sehingga pelaku UMKM dapat memahami jalur legalitas dan kewajiban perizinan yang sesuai dengan karakteristik usahanya.

DOI: https://doi.org/10.31004/riggs.v5i2.12016

## 2. Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan data sekunder UMKM dari salah satu dinas PLUT KUMKM periode 2024. Tahapan penelitian meliputi pengumpulan data, eksplorasi data, *preprocessing*, pembagian data, pemodelan *Random Forest* dan *XGBoost* , evaluasi model, serta pengujian teknik *resampling*. Berikut metode penelitian yang dijabarkan melalui diagram alir ditampilkan pada Gambar 1.




Gambar 1. Metode penelitian

### 2.1. Pengumpulan data

Data penelitian merupakan data sekunder UMKM dari salah satu dinas PLUT KUMKM periode 2024. *Dataset* yang digunakan berjumlah 4.081 entri data UMKM dan memuat delapan variabel utama, yaitu nama perusahaan, kecamatan usaha, investasi, omset, jumlah tenaga kerja, skala usaha, luas tanah, dan risiko usaha. Dalam penelitian ini, variabel risiko usaha digunakan sebagai label atau target klasifikasi, sedangkan atribut usaha digunakan sebagai variabel input dalam proses pemodelan.

Tabel 1. Deskripsi Variabel *Dataset* Penelitian

| Nama Variabel | Keterangan |
|---|---|
| Nama Perusahaan | Identitas usaha UMKM. |
| Kecamatan Usaha | Lokasi kecamatan tempat usaha beroperasi. |
| Investasi | Total nilai investasi yang dimiliki usaha dalam satuan rupiah. |
| Omset | Rata-rata pendapatan usaha dalam satuan rupiah. |
| TKI | Jumlah tenaga kerja yang dipekerjakan oleh usaha. |
| Skala Usaha | Klasifikasi skala usaha, yaitu mikro, kecil, atau menengah. |
| Luas Tanah | Luas area atau lokasi usaha dalam satuan meter persegi. |
| Risiko Usaha | Tingkat risiko perizinan usaha, yaitu rendah, menengah rendah, dan menengah tinggi. |

### 2.2. Eksplorasi data

Eksplorasi data dilakukan untuk memahami struktur dan karakteristik *dataset* sebelum memasuki tahap *preprocessing* dan pemodelan. Tahap ini digunakan untuk mengetahui jumlah data, jenis variabel, distribusi kelas target, persebaran data berdasarkan kecamatan, serta hubungan antarvariabel. Hasil eksplorasi data menjadi dasar

DOI: https://doi.org/10.31004/riggs.v5i2.12016

dalam menentukan langkah *preprocessing* yang diperlukan agar data siap digunakan dalam proses pelatihan model [14].

### 2.3. *Preprocessing*

Tahap *preprocessing* merupakan proses pengolahan awal data yang bertujuan untuk mempersiapkan *dataset* agar dapat digunakan secara optimal dalam proses pemodelan [15]. Pada penelitian ini, tahapan *preprocessing* dilakukan melalui beberapa proses, yaitu penghapusan atribut yang tidak digunakan, pemisahan fitur dan target, pemeriksaan data duplikat, *encoding* data kategorikal, deteksi *outlier*, serta normalisasi data. Atribut Nama Perusahaan dan Kecamatan Usaha dihapus karena tidak digunakan dalam proses pemodelan. Selanjutnya, variabel Risiko Usaha dipisahkan sebagai target klasifikasi, sedangkan variabel lainnya digunakan sebagai fitur. Data kategorikal seperti Risiko Usaha dan Skala Usaha diubah ke dalam bentuk numerik melalui proses *encoding*. Setelah itu, deteksi *outlier* dilakukan menggunakan metode Interquartile Range (IQR), kemudian fitur yang memiliki nilai ekstrem dinormalisasi agar pengaruh *outlier* terhadap proses pelatihan model dapat dikurangi.

### 2.4. Pembagian data

*Dataset* pada penelitian ini dibagi menjadi data latih dan data uji menggunakan rasio 80:20. Dari total 4.081 data, sebanyak 3.264 data atau 80% digunakan sebagai data latih untuk membangun model, sedangkan 817 data atau 20% digunakan sebagai data uji untuk mengevaluasi kemampuan model terhadap data yang tidak digunakan selama proses pelatihan. Pembagian ini dilakukan agar model dapat dilatih pada sebagian besar data, kemudian diuji menggunakan data berbeda untuk mengetahui kemampuan model dalam melakukan klasifikasi.

### 2.5. Pemodelan

Tahap pemodelan dilakukan untuk membangun model klasifikasi tingkat risiko perizinan usaha UMKM. Fitur yang digunakan dalam proses pemodelan meliputi investasi, omset, jumlah tenaga kerja, skala usaha, dan luas tanah, sedangkan variabel target yang digunakan adalah risiko usaha. Penelitian ini menggunakan dua algoritma *machine learning*, yaitu *Random Forest* dan *Extreme Gradient boosting* (*XGBoost* ). Pada masing-masing algoritma, pemodelan dilakukan dengan membandingkan model parameter *default* dan model yang telah dioptimasi melalui *hyperparameter tuning*. Model dengan parameter *default* digunakan untuk mengetahui performa awal algoritma sebelum dilakukan optimasi. Selanjutnya, *hyperparameter tuning* dilakukan untuk mencari kombinasi parameter yang lebih optimal agar model dapat menghasilkan performa klasifikasi yang lebih baik. Proses optimasi parameter dilakukan menggunakan *RandomizedSearchCV* dengan evaluasi berbasis *cross-validation*.

*Random Forest* merupakan algoritma *ensemble learning* yang dikembangkan dari *Decision tree* dengan membangun sejumlah pohon keputusan secara bersamaan. Setiap pohon dilatih menggunakan sampel data yang berbeda, sedangkan pemilihan atribut pada proses pemisahan cabang dilakukan secara acak. Pada proses klasifikasi, hasil prediksi tidak hanya bergantung pada satu pohon, tetapi diperoleh dari penggabungan prediksi seluruh pohon yang terbentuk [16].

Sejalan dengan hal tersebut, Ishakar et al. [17] menjelaskan bahwa *Random Forest* dapat digunakan dalam permasalahan klasifikasi dengan menggabungkan beberapa pohon keputusan untuk menghasilkan prediksi yang lebih stabil. Dalam proses pembentukan pohon keputusan, pemilihan fitur dan pemisahan *node* menjadi bagian penting untuk menghasilkan model yang baik. Salah satu ukuran yang dapat digunakan untuk menilai kualitas pemisahan data adalah Gini Index. Nilai Gini Index menunjukkan tingkat ketidakmurnian data pada suatu *node*. Semakin kecil nilai Gini, maka semakin homogen data pada *node* tersebut.

Persamaan Gini Index disajikan sebagai berikut.

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

S menggambarkan himpunan data pada suatu *node*, m menunjukkan jumlah kelas yang terdapat dalam data, i merupakan indeks kelas ke-i, $p_i$ menunjukkan proporsi data kelas ke-i terhadap seluruh data pada *node* S,

DOI: https://doi.org/10.31004/riggs.v5i2.12016

sedangkan $\sum_{i=1}^{m}$ menunjukkan proses penjumlahan proporsi seluruh kelas mulai dari kelas pertama sampai kelas ke-m.

*XGBoost* atau *Extreme Gradient boosting* merupakan algoritma *machine learning* berbasis ensemble tree yang menggunakan pendekatan *gradient boosting*. Algoritma ini bekerja dengan membangun model secara bertahap, di mana setiap pohon keputusan baru ditambahkan untuk memperbaiki kesalahan prediksi dari model sebelumnya. *XGBoost* memiliki keunggulan berupa kemampuan regularisasi untuk mengontrol kompleksitas model, sehingga dapat meningkatkan kemampuan generalisasi model [18]. Sejalan dengan hal tersebut, penelitian Karfindo et al. [19] menunjukkan bahwa *XGBoost* dapat digunakan sebagai salah satu model dalam pendekatan *ensemble machine learning* untuk menyelesaikan permasalahan klasifikasi.

Secara umum, model *XGBoost* dibangun secara aditif dengan menambahkan fungsi pohon baru pada setiap iterasi. Prediksi pada iterasi ke-t dapat dituliskan pada Persamaan 2.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \quad (2)$$

$\hat{y}_i^{(t)}$ menunjukkan hasil prediksi data ke-i pada iterasi ke-t, $\hat{y}_i^{(t-1)}$ menggambarkan hasil prediksi pada iterasi sebelumnya, sedangkan $f_i(x_i)$ merupakan fungsi pohon keputusan baru yang ditambahkan pada iterasi ke-t untuk memperbaiki hasil prediksi sebelumnya.

Fungsi objektif *XGBoost* terdiri atas fungsi kerugian dan komponen regularisasi. Fungsi kerugian digunakan untuk mengukur perbedaan antara nilai aktual dan nilai prediksi, sedangkan regularisasi digunakan untuk mengontrol kompleksitas model agar tidak mudah mengalami *overfitting*. Fungsi objektif *XGBoost* disajikan pada Persamaan 3.

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

$l(y_i, \hat{y}_i^{(t)})$ menunjukkan fungsi kerugian antara nilai aktual $y_i$ dan hasil prediksi $\hat{y}_i^{(t)}$, sedangkan $\Omega(f_k)$ menggambarkan fungsi regularisasi yang diberikan pada pohon ke-k untuk mengontrol kompleksitas model.

Komponen regularisasi pada *XGBoost* dapat dituliskan pada Persamaan 4

$$\Omega(f) = \gamma T + \tfrac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \quad (4)$$

T menunjukkan jumlah *leaf* pada pohon, $w_j$ menggambarkan bobot pada *leaf* ke-j, $\lambda$ merupakan parameter regularisasi L2 yang digunakan untuk mengontrol besar bobot *leaf*, sedangkan $\gamma$ merupakan parameter penalti terhadap jumlah *leaf* agar struktur pohon tidak terlalu kompleks. Melalui fungsi objektif dan regularisasi tersebut, *XGBoost* dapat membangun model yang lebih terkontrol, efisien, dan tidak mudah mengalami *overfitting* [18].

### 2.6. Teknik *Resampling*

Teknik *resampling* merupakan salah satu pendekatan yang digunakan untuk menangani ketidakseimbangan kelas pada data klasifikasi. Ketidakseimbangan kelas dapat menyebabkan model lebih dominan mempelajari kelas mayoritas, sehingga kelas minoritas kurang dikenali dengan baik dalam proses prediksi. Oleh karena itu, teknik *resampling* digunakan untuk menyeimbangkan distribusi data agar model dapat mempelajari pola pada setiap kelas secara lebih proporsional. SMOTE merupakan teknik *oversampling* yang membentuk data sintetis baru pada kelas minoritas berdasarkan kedekatan antar sampel, sehingga tidak hanya menggandakan data yang sudah ada[20]. ADASYN merupakan teknik *oversampling* sintetis, namun lebih menekankan pembentukan sampel baru pada data minoritas yang sulit dipelajari oleh model. Selain itu, SMOTE-ENN dan SMOTE-Tomek Links digunakan sebagai teknik *hybrid* karena menggabungkan SMOTE dengan metode pembersihan data untuk mengurangi data yang tumpang tindih antar kelas [21]. Penggunaan beberapa teknik *resampling* tersebut bertujuan untuk mengetahui pengaruh penyeimbangan distribusi kelas terhadap kemampuan model dalam mengenali seluruh kelas risiko usaha, khususnya kelas minoritas.

DOI: https://doi.org/10.31004/riggs.v5i2.12016

### 2.7. Evaluasi

Evaluasi model dilakukan untuk mengetahui kemampuan algoritma dalam mengklasifikasikan data uji. Kinerja model klasifikasi dapat diukur menggunakan *confusion matrix* serta beberapa metrik evaluasi, yaitu *accuracy*, *precision*, *recal*l, dan F1-*score*. *Confusion matrix* digunakan untuk menunjukkan jumlah prediksi yang benar dan salah pada setiap kelas. Komponen utama dalam *confusion matrix* terdiri atas True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). TP dan TN menunjukkan prediksi yang benar, sedangkan FP dan FN menunjukkan kesalahan prediksi model [22]. Penggunaan metrik tersebut banyak diterapkan pada penelitian klasifikasi berbasis *machine learning* untuk menilai performa model secara kuantitatif.

## 3. Hasil dan Diskusi

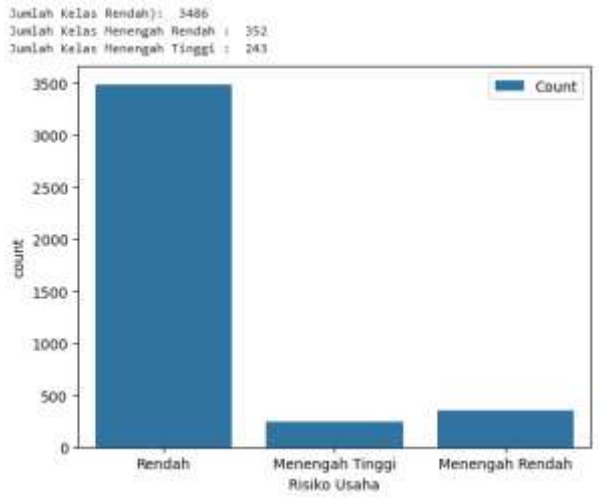
### 3.1. Data Penelitian

Data penelitian disimpan dalam format Microsoft Excel dan memuat 4.081 entri data UMKM dengan delapan variabel, yaitu Nama Perusahaan, Kecamatan Usaha, Investasi, TKI, Skala Usaha, Omset, Luas Tanah, dan Risiko Usaha. Struktur awal *dataset* diperiksa menggunakan fungsi df.info() untuk mengetahui jumlah data, nama variabel, tipe data, nilai terisi, dan kapasitas memori yang digunakan. Hasil pemeriksaan menunjukkan bahwa *dataset* terdiri atas 4.081 data UMKM dengan delapan variabel. Variabel Risiko Usaha digunakan sebagai target klasifikasi, sedangkan variabel lainnya dianalisis untuk menentukan fitur yang digunakan dalam pemodelan.

### 3.2. Eksplorasi Data

Pada tahap ini, *dataset* divisualisasikan untuk memberikan pemahaman yang lebih jelas mengenai kondisi data.

1. Distribusi Kelas Target

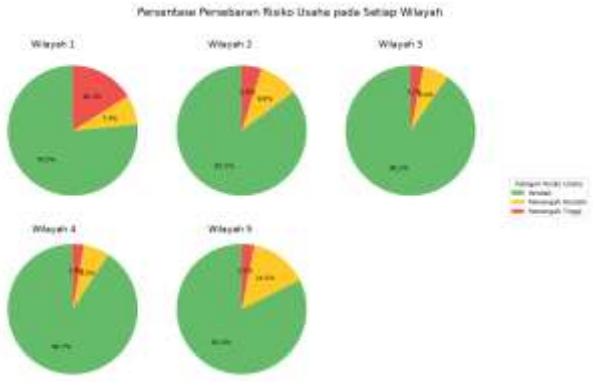



Gambar 2. Distribusi kelas target

Distribusi kelas target menunjukkan bahwa data penelitian memiliki ketidakseimbangan kelas. Kelas Rendah menjadi kelas mayoritas dengan 3.486 data, sedangkan kelas Menengah Rendah berjumlah 352 data dan kelas Menengah Tinggi berjumlah 243 data.

DOI: https://doi.org/10.31004/riggs.v5i2.12016

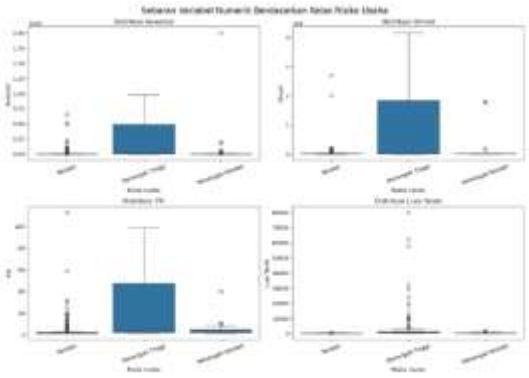
2. Distribusi persebaran kelas target per wilayah




Gambar 3. Distribusi persebaran kelas target per wilayah

Persebaran kelas target pada beberapa wilayah menunjukkan bahwa kelas Rendah mendominasi hampir seluruh wilayah, sedangkan kelas Menengah Rendah dan Menengah Tinggi memiliki jumlah yang lebih kecil. Kondisi ini memperkuat adanya ketidakseimbangan kelas pada *dataset* .

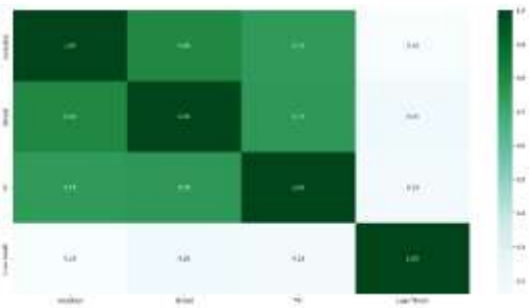
3. Distribusi variabel numerik berdasarkan kelas risiko usaha



Gambar 4. Sebaran variabel numerik berdasarkan kelas risiko usaha

Berdasarkan Gambar 4, sebaran variabel Investasi, Omset, TKI, dan Luas Tanah menunjukkan adanya perbedaan pola pada setiap kelas Risiko Usaha. Kelas Menengah Tinggi cenderung memiliki rentang nilai yang lebih luas dibandingkan kelas lainnya, sedangkan kelas Rendah dan Menengah Rendah lebih terkonsentrasi pada nilai yang lebih rendah.

4. Korelasi



Gambar 5. Korelasi antarvariabel numerik

DOI: https://doi.org/10.31004/riggs.v5i2.12016

Gambar 5 menunjukkan bahwa Investasi, Omset, dan TKI memiliki korelasi yang kuat satu sama lain, dengan nilai berkisar antara 0,75 hingga 0,80, yang mengindikasikan bahwa ketiga variabel ini saling berkaitan erat. Sementara itu, Luas Tanah menunjukkan korelasi yang lemah terhadap ketiga variabel tersebut (0,16–0,20), sehingga dapat diketahui bahwa luas tanah tidak banyak memengaruhi besarnya investasi, omset, maupun jumlah tenaga kerja.

### 3.3. *Preprocessing*

Tahap *preprocessing* dilakukan guna mempersiapkan data agar siap digunakan dalam proses pelatihan model.

**Penghapusan fitur yang tidak digunakan dalam model**

Variabel Nama Perusahaan dan Kecamatan Usaha dihapus dari fitur pemodelan menggunakan fungsi drop(). Nama Perusahaan dihapus karena hanya berupa identitas usaha yang tidak berkontribusi terhadap klasifikasi, sedangkan Kecamatan Usaha dihapus karena hanya digunakan untuk analisis persebaran wilayah pada tahap EDA, bukan untuk menentukan risiko usaha. Fitur yang digunakan pada penelitian ini adalah investasi, TKI, skala usaha, omset, luas tanah dan risiko usaha sebagai variabel target.

**Encoding**

Tahap encoding dilakukan untuk mengubah variabel kategorikal menjadi bentuk numerik agar dapat diproses oleh algoritma *Random Forest* dan *XGBoost* . Pada penelitian ini, variabel Risiko Usaha sebagai target klasifikasi dikodekan menjadi tiga kelas, yaitu Rendah menjadi 0, Menengah Rendah menjadi 1, dan Menengah Tinggi menjadi 2. Sementara itu, variabel Skala Usaha sebagai salah satu fitur input dikodekan menjadi Usaha Mikro bernilai 1, Usaha Kecil bernilai 2, dan Usaha Menengah bernilai 3. Proses ini dilakukan karena algoritma *machine learning* membutuhkan data dalam bentuk numerik untuk melakukan proses pelatihan dan prediksi model.

**Deteksi dan Penanganan Outlier**

Outlier adalah nilai ekstrem yang berbeda secara signifikan dari pola data secara umum, yang apabila tidak ditangani dapat memengaruhi akurasi dan kestabilan model dalam melakukan klasifikasi maupun prediksi. Pada penelitian ini, deteksi outlier dilakukan terhadap fitur-fitur untuk mengidentifikasi nilai-nilai yang menyimpang jauh dari rentang normal data. Identifikasi ini menjadi dasar dalam menentukan fitur mana yang memerlukan penanganan lebih lanjut. Hasil deteksi outlier disajikan dalam tabel 2.

Tabel 2. Fitur yang teridentifikasi Outlier

| No. | Nama fitur | Jumlah *Outlier* | Persentase (%) |
|---|---|---|---|
| 1 | Investasi | 400 | 9,802 |
| 2 | Omset | 157 | 3,847 |
| 3 | TKI | 554 | 13,575 |
| 4 | Luas Tanah | 224 | 5,489 |
| 5 | Skala Usaha | 202 | 4,950 |

Berdasarkan Tabel 2, terdapat tiga fitur dengan persentase outlier di atas 5%, yaitu Investasi, TKI, dan Luas Tanah. Oleh karena itu, ketiga fitur tersebut perlu diperhatikan pada tahap *preprocessing* dan dilakukan normalisasi untuk mengurangi pengaruh nilai ekstrem terhadap proses pelatihan model.

### 3.4. Pembagian Data

Setelah melalui tahap pra-pemrosesan, *dataset* dibagi menjadi data latih dan data uji menggunakan metode train-test split dengan proporsi 80:20. Pembagian ini dilakukan agar sebagian besar data dapat digunakan untuk melatih model, sedangkan sebagian lainnya digunakan untuk mengevaluasi kemampuan model terhadap data yang tidak digunakan selama proses pelatihan.

DOI: https://doi.org/10.31004/riggs.v5i2.12016

### 3.5. Pemodelan

Pemodelan dilakukan menggunakan algoritma *Random Forest* dan *XGBoost* , masing-masing diuji dalam dua konfigurasi yaitu parameter *default* dan *hyperparameter tuning*. Ketidakseimbangan kelas ditangani melalui parameter *class_weight* pada *Random Forest*, sedangkan pada *XGBoost* menggunakan *sample_weight* yang dihitung berdasarkan distribusi kelas pada data latih. Proses tuning dilakukan menggunakan *RandomizedSearchCV* dengan 50 iterasi dan validasi *StratifiedKFold* sebanyak 5 fold. Nilai parameter terbaik yang diperoleh dari proses *hyperparameter tuning* pada masing-masing algoritma disajikan pada Tabel 3.

Tabel 3. Hasil *hyperparameter tuning*

| Algoritma | Parameter | Nilai Terbaik |
|---|---|---|
| *Random Forest* | n_estimators | 140 |
| | max_features | sqrt |
| | max_depth | None |
| | min_samples_split | 20 |
| | min_samples_*leaf* | 2 |
| | *class_weight* | {0:1, 1:3, 2:6} |
| *XGBoost* | criterion | gini |
| | learning_rate / eta | 0,1 |
| | max_depth | 8 |
| | subsample | 0,5 |
| | n_estimators | 300 |
| | min_child_weight | 1 |
| | gamma | 0,1 |
| | colsample_bylevel | 1,0 |

Berdasarkan Tabel 3, hasil *hyperparameter tuning* menunjukkan bahwa pada algoritma *Random Forest* diperoleh konfigurasi terbaik dengan jumlah pohon sebanyak 140, pemilihan fitur menggunakan metode sqrt, kedalaman pohon tanpa batas (None), serta penanganan ketidakseimbangan kelas melalui *class_weight* dengan bobot {0:1, 1:3, 2:6}. Sementara itu, pada algoritma *XGBoost* diperoleh konfigurasi terbaik dengan *learning rate* sebesar 0,1, kedalaman pohon maksimum 8, jumlah estimator sebanyak 300, dan subsample sebesar 0,5. Konfigurasi parameter terbaik ini selanjutnya digunakan sebagai dasar pelatihan model yang telah dioptimasi untuk kemudian dievaluasi dan dibandingkan performanya dengan model parameter *default*.

### 3.6. Evaluasi

Evaluasi model dilakukan untuk mengetahui kemampuan setiap algoritma dalam mengklasifikasikan data uji. Model yang dievaluasi terdiri atas *Random Forest default*, *Random Forest* hasil *hyperparameter tuning*, *XGBoost default*, dan *XGBoost* hasil *hyperparameter tuning*. Penilaian dilakukan menggunakan *accuracy*, *precision*, *recal*l, F1-*score*, dan *support*.

Model pertama yang dievaluasi adalah *Random Forest* dengan parameter *default*. Evaluasi ini digunakan sebagai dasar untuk melihat performa awal algoritma sebelum dilakukan optimasi parameter.

Tabel 4. Hasil evaluasi *Random Forest* dengan parameter *default*

| Kelas | *Precision* | *Recal*l | F1-*Score* | *Support* |
|---|---|---|---|---|
| 0 (Rendah) | 0.95 | 0.99 | 0.97 | 698 |
| 1 (Menengah Rendah) | 0.69 | 0.49 | 0.57 | 70 |
| 2 (Menengah Tinggi) | 0.93 | 0.76 | 0.83 | 49 |
| *Accuracy* | - | - | 0.92 | 817 |
| *Macro* Avg | 0.86 | 0.74 | 0.79 | 817 |
| Weighted Avg | 0.92 | 0.93 | 0.92 | 817 |

DOI: https://doi.org/10.31004/riggs.v5i2.12016

Berdasarkan Tabel 4, *Random Forest default* memperoleh *accuracy* 92%, *precision macro* 86%, *recall macro* 74%, dan F1-*score macro* 79%. Hasil ini menunjukkan bahwa model memiliki akurasi tinggi, tetapi *recall macro* masih lebih rendah karena kemampuan model dalam mengenali kelas minoritas belum sepenuhnya seimbang.

Model kedua adalah *Random Forest* yang telah dioptimasi melalui *hyperparameter tuning*. Pada model ini, penanganan ketidakseimbangan kelas dilakukan menggunakan *class_weight*.

Tabel 5. Hasil evaluasi *Random Forest* dengan *hyperparameter tuning*

| **Kelas** | ***Precision*** | ***Recall*** | **F1-*Score*** | ***Support*** |
|---|---|---|---|---|
| 0 (Rendah) | 0.96 | 0.98 | 0.97 | 698 |
| 1 (Menengah Rendah) | 0.69 | 0.61 | 0.65 | 70 |
| 2 (Menengah Tinggi) | 0.95 | 0.76 | 0.84 | 49 |
| *Accuracy* | - | - | 0.94 | 817 |
| *Macro* Avg | 0.87 | 0.78 | 0.82 | 817 |
| Weighted Avg | 0.93 | 0.94 | 0.94 | 817 |

Berdasarkan Tabel 5, *Random Forest* hasil *hyperparameter tuning* memperoleh *accuracy* 94%, *precision macro* 87%, *recall macro* 78%, dan F1-*score macro* 82%. Dibandingkan model *default*, terjadi peningkatan pada *accuracy*, *recall macro* , dan F1-*score macro* . Hal ini menunjukkan bahwa optimasi parameter dan penggunaan *class_weight* mampu membantu model mengenali kelas risiko secara lebih seimbang.

Model ketiga adalah *XGBoost* dengan parameter *default*. Model ini dilatih tanpa penyesuaian parameter manual dan tanpa penanganan ketidakseimbangan kelas secara eksplisit.

Tabel 6. Hasil evaluasi *XGBoost* dengan parameter *default*

| **Kelas** | ***Precision*** | ***Recall*** | **F1-*Score*** | ***Support*** |
|---|---|---|---|---|
| 0 (Rendah) | 0.95 | 0.98 | 0.97 | 698 |
| 1 (Menengah Rendah) | 0.65 | 0.56 | 0.60 | 70 |
| 2 (Menengah Tinggi) | 0.93 | 0.78 | 0.84 | 49 |
| *Accuracy* | - | - | 0.93 | 817 |
| *Macro* Avg | 0.84 | 0.77 | 0.80 | 817 |
| Weighted Avg | 0.93 | 0.93 | 0.93 | 817 |

Berdasarkan Tabel 6, *XGBoost default* memperoleh *accuracy* 93%, *precision macro* 84%, *recall macro* 77%, dan F1-*score macro* 80%. Nilai tersebut menunjukkan bahwa *XGBoost default* mampu menghasilkan performa yang cukup baik dan lebih seimbang dibandingkan *Random Forest default* berdasarkan F1-*score macro* .

Model keempat adalah *XGBoost* yang telah dioptimasi melalui *hyperparameter tuning*. Pada model ini, penanganan ketidakseimbangan kelas dilakukan menggunakan *sample_weight* berdasarkan distribusi kelas pada data latih.

Tabel 7. Hasil evaluasi *XGBoost* dengan *hyperparameter tuning*

| **Kelas** | ***Precision*** | ***Recall*** | **F1-*Score*** | ***Support*** |
|---|---|---|---|---|
| 0 (Rendah) | 0.96 | 0.95 | 0.95 | 698 |
| 1 (Menengah Rendah) | 0.52 | 0.66 | 0.58 | 70 |
| 2 (Menengah Tinggi) | 0.95 | 0.76 | 0.84 | 49 |
| *Accuracy* | - | - | 0.91 | 817 |
| *Macro* Avg | 0.81 | 0.79 | 0.79 | 817 |
| Weighted Avg | 0.92 | 0.91 | 0.92 | 817 |

DOI: https://doi.org/10.31004/riggs.v5i2.12016

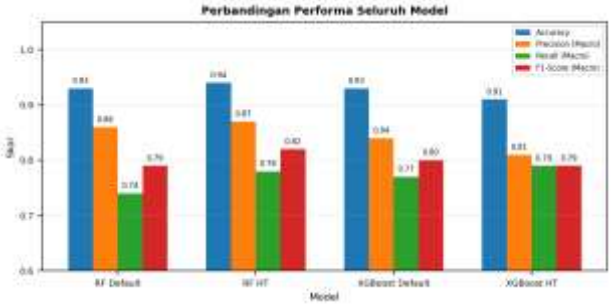
Berdasarkan Tabel 7, *XGBoost* hasil *hyperparameter tuning* memperoleh *accuracy* 91%, *precision macro* 81%, *recall macro* 79%, dan F1-*score macro* 79%. Nilai *recall macro* meningkat dibandingkan *XGBoost default*, tetapi *accuracy*, *precision macro* , dan F1-*score macro* menurun. Hal ini menunjukkan bahwa *sample_weight* membuat model lebih sensitif terhadap kelas minoritas, tetapi belum meningkatkan performa keseluruhan.

### 3.7. Komparasi Performa Model

Komparasi dilakukan untuk melihat model yang memiliki performa terbaik secara keseluruhan. Karena *dataset* memiliki ketidakseimbangan kelas, metrik *macro* digunakan agar setiap kelas memperoleh bobot penilaian yang sama. Hasil perbandingan seluruh model disajikan pada Tabel 8 dan divisualisasikan pada Gambar 6.

Tabel 8. Hasil perbandingan seluruh model

| **Model** | ***Accuracy*** | ***Precision* (*Macro* )** | ***Recall* (*Macro* )** | **F1-*Score* (*Macro* )** |
|---|---|---|---|---|
| *Random Forest* HT | 94% | 87% | 78% | 82% |
| *XGBoost Default* | 93% | 84% | 77% | 80% |
| *XGBoost* HT | 91% | 81% | 79% | 79% |
| *Random Forest Default* | 93% | 86% | 74% | 79% |




Gambar 6. Perbandingan performa seluruh model

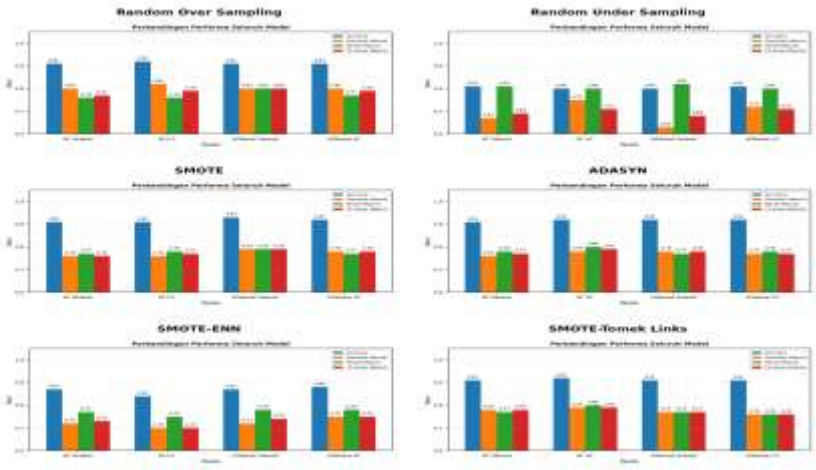
Berdasarkan Tabel 8 dan Gambar 6, *Random Forest* dengan *hyperparameter tuning* memperoleh performa terbaik dengan *accuracy* 94%, *precision macro* 87%, *recall macro* 78%, dan F1-*score macro* 82%. Model ini unggul dibandingkan *XGBoost default*, *XGBoost* dengan *hyperparameter tuning*, dan *Random Forest default*. Nilai F1-*score macro* tertinggi menunjukkan bahwa model tersebut memiliki keseimbangan performa yang lebih baik dalam mengenali ketiga kelas risiko usaha.

Pengaruh optimasi terlihat paling jelas pada *Random Forest*. Setelah *hyperparameter tuning* dan penggunaan *class_weight*, F1-*score macro* meningkat dari 79% menjadi 82%. Sebaliknya, *hyperparameter tuning* pada *XGBoost* meningkatkan *recall macro* dari 77% menjadi 79%, tetapi menurunkan *accuracy* dan F1-*score macro* . Oleh karena itu, model *Random Forest* dengan *hyperparameter tuning* dinilai sebagai model terbaik pada pengujian utama.

### 3.8. Penggunaan Teknik *Resampling*

Penelitian ini juga menguji beberapa teknik *resampling* untuk menangani ketidakseimbangan data. Teknik yang digunakan meliputi *Random Over Sampling*, *Random Under Sampling*, SMOTE, ADASYN, SMOTE-ENN, dan SMOTE-Tomek Links. Pengujian ini dilakukan karena distribusi kelas yang tidak seimbang dapat membuat model lebih memprioritaskan kelas mayoritas, sehingga kelas minoritas berisiko kurang dikenali. Oleh karena itu, *resampling* digunakan sebagai pembanding untuk mengetahui apakah penyeimbangan distribusi kelas mampu meningkatkan performa klasifikasi pada *dataset* UMKM.

DOI: https://doi.org/10.31004/riggs.v5i2.12016




Gambar 7. Hasil evaluasi seluruh teknik *resampling*

Setelah seluruh teknik *resampling* diuji, dilakukan rekapitulasi berdasarkan model terbaik pada setiap teknik. Rekapitulasi ini digunakan untuk mengetahui pendekatan yang paling stabil dalam menangani ketidakseimbangan kelas pada *dataset* UMKM.

Tabel 9. Rekapitulasi model pada setiap teknik *resampling*

| **Teknik** | **Model** | ***Accuracy*** | ***Precision* (*Macro* )** | ***Recall* (*Macro* )** | **F1-*Score* (*Macro* )** |
|---|---|---|---|---|---|
| Tanpa *Resampling* | *Random Forest* HT | 94% | 87% | 78% | 82% |
| *Random Over Sampling* | *XGBoost Default* | 91% | 80% | 80% | 80% |
| *Random Under Sampling* | *Random Forest* HT | 80% | 75% | 80% | 71% |
| SMOTE | *XGBoost Default* | 93% | 79% | 79% | 79% |
| ADASYN | *Random Forest* HT | 92% | 78% | 80% | 79% |
| SMOTE-ENN | *XGBoost* HT | 88% | 75% | 78% | 75% |
| SMOTE-Tomek Links | *Random Forest* HT | 92% | 79% | 80% | 79% |

Berdasarkan Tabel 9, model terbaik secara keseluruhan tetap diperoleh pada skenario tanpa *resampling*, yaitu *Random Forest* dengan *hyperparameter tuning* yang menghasilkan *accuracy* 94% dan F1-*score macro* 82%. Pada skenario *resampling*, nilai F1-*score macro* tertinggi diperoleh oleh *XGBoost default* dengan *Random Over Sampling* sebesar 80%, tetapi hasil tersebut masih lebih rendah dibandingkan model tanpa *resampling*.

Hasil ini menunjukkan bahwa penerapan *resampling* dapat membantu model menyesuaikan pembelajaran terhadap kelas minoritas, tetapi belum mampu memberikan performa yang lebih tinggi dibandingkan pendekatan tanpa *resampling*. Model tanpa *resampling* dengan *Random Forest hyperparameter tuning* memperoleh F1-*score macro* tertinggi sebesar 82%, sedangkan teknik *resampling* terbaik diperoleh oleh *Random Over Sampling* pada *XGBoost default* dengan F1-*score macro* sebesar 80%. Teknik SMOTE, ADASYN, dan SMOTE-Tomek Links menghasilkan F1-*score macro* sebesar 79%, sedangkan *Random Under Sampling* dan SMOTE-ENN cenderung menurunkan performa. Temuan ini menunjukkan bahwa *resampling* tidak selalu meningkatkan performa model, terutama apabila data sintetis atau pengurangan data mayoritas tidak sepenuhnya merepresentasikan pola asli *dataset*.

DOI: https://doi.org/10.31004/riggs.v5i2.12016

### 3.9. Analisis Komparasi Model dan Penanganan Ketidak Seimbangan Kelas

Berdasarkan hasil evaluasi, *Random Forest* dengan *hyperparameter tuning* memperoleh performa terbaik dibandingkan model lainnya. Hal ini menunjukkan bahwa karakteristik *Random Forest* lebih sesuai dengan pola data UMKM yang digunakan dalam penelitian ini. *Random Forest* bekerja dengan membangun banyak pohon keputusan dan menggabungkan hasil prediksi melalui mekanisme *voting*, sehingga model menjadi lebih stabil dalam mengenali pola data. Selain itu, penggunaan *class_weight* pada *Random Forest* membantu model memberikan perhatian lebih besar terhadap kelas minoritas, sehingga performa model menjadi lebih seimbang. Hal ini terlihat dari peningkatan *F1-score macro* dari 79% pada Random Forest *default* menjadi 82% setelah dilakukan *hyperparameter tuning*. Sementara itu, *XGBoost* juga menerapkan penanganan ketidakseimbangan kelas melalui *sample_weight*, yaitu pemberian bobot pada setiap sampel berdasarkan distribusi kelas pada data latih. Penggunaan *sample_weight* bertujuan agar sampel dari kelas minoritas memperoleh perhatian lebih besar selama proses pelatihan. Namun, hasil evaluasi menunjukkan bahwa *XGBoost* dengan *hyperparameter tuning* mengalami penurunan performa dibandingkan *XGBoost default*. *XGBoost default* memperoleh *accuracy* sebesar 93% dan *F1-score macro* sebesar 80%, sedangkan *XGBoost* dengan *hyperparameter tuning* dan *sample_weight* memperoleh *accuracy* sebesar 91% dan *F1-score macro* sebesar 79%. Penurunan ini terjadi karena kombinasi parameter hasil *tuning* dan pemberian *sample_weight* membuat model menjadi lebih sensitif terhadap kelas minoritas, tetapi di sisi lain meningkatkan kesalahan prediksi pada kelas lain. Hal tersebut terlihat dari meningkatnya *recall macro* dari 77% menjadi 79%, namun diikuti dengan penurunan *accuracy*, *precision macro*, dan *F1-score macro*. Dengan demikian, penggunaan *sample_weight* pada *XGBoost* mampu meningkatkan sensitivitas model terhadap kelas minoritas, tetapi belum mampu meningkatkan performa keseluruhan model.

Penggunaan *class_weight* dilakukan untuk menangani ketidakseimbangan kelas pada *dataset* . Distribusi kelas yang tidak seimbang dapat menyebabkan model lebih dominan mempelajari kelas mayoritas, sehingga kelas minoritas kurang dikenali dengan baik. Dengan menggunakan *class_weight*, model memberikan bobot yang lebih besar pada kelas minoritas saat proses pelatihan. Pada penelitian ini, Random Forest dengan *hyperparameter tuning* menggunakan *class_weight* {0:1, 1:3, 2:6}. Bobot tersebut membuat model tidak hanya berfokus pada kelas Risiko Rendah sebagai kelas mayoritas, tetapi juga lebih memperhatikan kelas Menengah Rendah dan Menengah Tinggi. Hasil evaluasi menunjukkan bahwa penggunaan *class_weight* mampu meningkatkan performa Random Forest. Random Forest *default* memperoleh *F1-score macro* sebesar 79%, sedangkan Random Forest dengan *hyperparameter tuning* dan *class_weight* memperoleh *F1-score macro* sebesar 82%. Selain itu, *recall* pada kelas Menengah Rendah meningkat dari 49% menjadi 61%. Peningkatan ini menunjukkan bahwa *class_weight* membantu model mengenali kelas minoritas dengan lebih baik tanpa perlu mengubah jumlah data asli seperti pada teknik *resampling*.

Pada pengujian teknik *resampling*, *Random Over Sampling* (ROS) menghasilkan performa terbaik dibandingkan teknik *resampling* lainnya dengan F1-score macro sebesar 80%. Hasil ini lebih tinggi dibandingkan SMOTE yang memperoleh F1-score macro sebesar 79%, ADASYN sebesar 79%, SMOTE-*Tomek Links* sebesar 79%, SMOTE-ENN sebesar 75%, dan *Random Under Sampling* sebesar 71%. Perbedaan tersebut menunjukkan bahwa pada dataset penelitian ini, penambahan data minoritas melalui penggandaan sampel asli pada ROS lebih sesuai dibandingkan pembentukan data sintetis. ROS mempertahankan pola asli kelas minoritas karena data yang ditambahkan berasal dari sampel yang sudah ada, sedangkan SMOTE dan ADASYN membentuk data sintetis berdasarkan kedekatan antar sampel. Pada data UMKM yang memiliki tumpang tindih antar kelas dan beberapa nilai ekstrem, data sintetis yang dibentuk belum tentu sepenuhnya merepresentasikan pola asli kelas minoritas, sehingga performanya belum mampu mengungguli ROS. Sementara itu, SMOTE-Tomek Links dan SMOTE-ENN memperoleh performa lebih rendah karena proses pembersihan data setelah pembentukan sampel sintetis dapat menghapus sebagian data yang masih memiliki informasi penting. *Random Under Sampling* menghasilkan performa paling rendah karena teknik ini mengurangi jumlah data pada kelas mayoritas, sehingga sebagian informasi penting dari kelas mayoritas berpotensi hilang. Dengan demikian, ROS menjadi teknik resampling terbaik pada penelitian ini, tetapi performanya masih belum melampaui model utama Random Forest dengan hyperparameter tuning tanpa resampling.

## 4. Kesimpulan

Berdasarkan hasil pengujian, *Random Forest* dengan *hyperparameter tuning* memperoleh performa terbaik dalam mengklasifikasikan tingkat risiko perizinan usaha UMKM. Model ini memperoleh *accuracy* sebesar 94%, *precision macro* sebesar 87%, *recal*l *macro* sebesar 78%, dan F1-*score macro* sebesar 82%. Sementara itu, *XGBoost* dengan *hyperparameter tuning* memperoleh *accuracy* sebesar 91%, *precision macro* sebesar 81%,

DOI: https://doi.org/10.31004/riggs.v5i2.12016

*recall macro* sebesar 79%, dan *F1-score macro* sebesar 79%. Random Forest *default* memperoleh *accuracy* sebesar 93%, *precision macro* sebesar 86%, *recall macro* sebesar 74%, dan *F1-score macro* sebesar 79% dan *XGBoost default* memperoleh *accuracy* sebesar 93%, *precision macro* sebesar 84%, *recall macro* sebesar 77%, dan *F1-score macro* sebesar 80%. Hasil tersebut menunjukkan bahwa kombinasi optimasi parameter dan *class_weight* mampu menghasilkan performa yang lebih stabil dibandingkan Random Forest *default*, *XGBoost default*, dan *XGBoost* dengan *hyperparameter tuning*. Pengujian teknik *resampling* menunjukkan bahwa *Random Over Sampling*, *Random Under Sampling*, SMOTE, ADASYN, SMOTE-ENN, dan SMOTE-Tomek Links belum mampu mengungguli performa Random Forest dengan *hyperparameter tuning* tanpa *resampling*. Dengan demikian, pendekatan *Random Forest* dengan *hyperparameter tuning* dan pembobotan kelas menjadi model yang paling sesuai untuk mengklasifikasikan tingkat risiko usaha UMKM berdasarkan fitur investasi, omset, jumlah tenaga kerja, skala usaha, dan luas tanah.

## Referensi

[1] M. Purgianto, C. D. Massie, and R. M. S. Sarapun, "Konsekuensi Hukum Bagi Penyimpangan Terhadap Kewajiban Persetujuan Lingkungan Hidup Terkait Dengan Perizinan Berusaha," *Fak. Huk. Lex Priv.*, vol. XII, no. I, 2023.
[2] H. Amalia Febrianti, K. Iskandar, N. Aisyah, M. Dini Adita Pembuatan Nomor Induk Berusaha, and M. Dini Adita, "Pembuatan Nomor Induk Berusaha (NIB) dan Kesadaran Legalitas Usaha di Usaha Mikro, Kecil, dan Menengah (UMKM) Desa Tegalreja," *Era Abdimas J. Pengabdi. dan Pemberdaya. Masy. Multidisiplin*, vol. 1, no. 4, pp. 54–61, 2023.
[3] I. M. Faiza and W. Andriani, "Tinjauan Pustaka Sistematis : Penerapan Metode Machine Learning untuk Deteksi Bencana Banjir," vol. 11, no. September, pp. 59–63, 2022.
[4] M. Erkamim, S. Suswadi, M. Z. Subarkah, and E. Widarti, "Komparasi Algoritme Random Forest dan *XGBoost* ing dalam Klasifikasi Performa UMKM," *J. Sist. Inf. Bisnis*, vol. 13, no. 2, pp. 127–134, 2023.
[5] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[6] K. Afane and Y. Zhao, "Selecting Classifiers and Resampling Techniques for Imbalanced *Dataset* s: A New Perspective," *Procedia Comput. Sci.*, vol. 246, no. C, pp. 1150–1159, Jan. 2024.
[7] T. A. Hasibuan, K. N. Simatupang, Z. Arbiah, and M. K. Gibran, "Evaluasi Kinerja Algoritma Machine Learning pada Data Administratif Warga Binaan," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 9, no. 1, pp. 154–161, 2026.
[8] M. N. Musyafa, E. K. Silalahi, I. A. Berutu, K. R. Azis, and F. Ramadhani, "Prediksi Churn Nasabah Bank Menggunakan Algoritma Random Forest," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 8, no. 4, pp. 1801–1807, 2025.
[9] R. S. Hidayana *et al.*, "Sistem Prediksi Risiko Penyakit Jantung Berbasis Machine Learning dan Framework Streamlit," vol. 8, no. 6, pp. 3672–3680, 2025.
[10] C. A. Sugianto and P. N. Sari, "Klasifikasi Kelayakan Penerima Bantuan Langsung Tunai Bagi UMKM di Kota Cimahi Menggunakan Algoritma Naïve Bayes," *J. Informatics Electron. Eng.*, vol. 4, no. 2, p. 64, 2024.
[11] G. I. Royani, N. S. Amelia, Marlina, and F. N. Cahya, "Studi Komparatif Algoritma C4.5 Dan Random Forest Pada Digitalisasi UMKM Kabupaten Tegal," *LPPM Universitas Nusa Mandiri*, p. VOL.20, 2026.
[12] A. Saepuloh, P. A. Sanusi, E. Rilvani, U. P. Bangsa, and K. Bekasi, "Analisis Komparatif Prediksi Keberhasilan Usaha Kecil Menengah Menggunakan Credal C4.5 dan Credal Decision Tree," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 3, no. 7, 2025.
[13] E. Danuarta and D. Alita, "Perbandingan Algoritma Naïve Bayes dan Random Forest untuk Melakukan Analisis Sentimen Cyberbullying Generasi Z Pada Twitter," *Build. Informatics, Technol. Sci.*, vol. 6, no. 4, pp. 2448–2458, 2025.
[14] E. I. Muhammad Radhi, Amalia,Daniel Ryan Hamonangan Sitompul,Stiven Hamonangan Sinurat, "Analisis Big Data Dengan Metode (EDA) menggunakan jupyter notebook," vol. 4, no. 2, pp. 2–6, 2022.
[15] Stacyana Jesika, S. Ramadhani, and Yohanna Permata Putri, "Implementasi Model Machine Learning dalam Mengklasifikasi Kualitas Air," *J. Ilm. Dan Karya Mhs.*, vol. 1, no. 6, pp. 382–396, 2023.
[16] I. made budi Adnyana, "Prediksi Lama Studi Mahasiswa STIKOM ," pp. 201–208, Aug. 2015.
[17] P. Ishakar, Andika, and M. I. R. Ihsan, "Aplikasi Prediksi Churn Pelanggan Menggunakan Algoritma Random Forest pada Perusahaan Telekomunikasi," *J. Teknol. Inf.*, vol. 8, 2025.
[18] T. Chen and C. Guestrin, "*XGBoost* : A Scalable Tree Boosting System," *Assoc. Comput. Mach.*, vol. 42, no. 8, p. 665, 2016.
[19] Karfindo, R. Turaina, and R. Saputra, "Optimalisasi Deteksi Malware pada Platform Android dengan Pendekatan Ensemble Machine Learning," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 7, no. 3, pp. 394–403, 2024.
[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
[21] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *Proc. Int. Jt. Conf. Neural Networks*, pp. 1322–1328, 2008.
[22] A. F. Azmi and A. Voutama, "Prediksi Churn Nasabah Bank Menggunakan Klasifikasi Random Forest Dan Decision Tree Dengan Evaluasi Confusion Matrix," *Komputa J. Ilm. Komput. dan Inform.*, vol. 13, no. 1, pp. 111–119, 2024.

DOI: https://doi.org/10.31004/riggs.v5i2.12016