(2025), 3 (6): 161-170 **Scientica** 3021-8209

Jurnal Ilmiah Sain dan Teknologi

PERBANDINGAN KINERJA NAIVE BAYES DAN RANDOM FOREST DENGAN PENANGANAN *IMBALANCE DATA*

Fortunatus Adhiethera Tuah Putra [*1], Arif Bijaksana Putra Negara [2], Helen Sastypratiwi [3]

[1,2,3] Informatika, Teknik, Universitas Tanjungpura

[1*]d1041211048@student.untan.ac.id

Abstrak (Indonesia)

Data mining merupakan proses penting untuk mengekstraksi informasi berharga dari kumpulan data besar dan kompleks. Salah satu teknik utamanya adalah klasifikasi, yang digunakan untuk memprediksi kategori data berdasarkan fitur tertentu. Penelitian ini membandingkan performa algoritma Naïve Bayes dan Random Forest dalam mengatasi klasifikasi pada data tidak seimbang. Dataset yang digunakan adalah *Bank Marketing* dari UCI *Machine Learning Repository* yang memiliki distribusi kelas tidak seimbang, dengan perbandingan signifikan antara label "yes" dan "no". Penelitian ini mengevaluasi pengaruh dua teknik penyeimbangan data, yaitu *Synthetic Minority Oversampling Technique* (SMOTE) dan *undersampling*, terhadap kinerja kedua algoritma dengan metrik akurasi, presisi, *recall*, dan *F1-score*. Pada Naïve Bayes, model *default* memberikan hasil terbaik (akurasi 91,78%, presisi 90,59, *recall* 91,78, *F1-Score* 90,93), sedangkan penggunaan SMOTE atau *undersampling* justru menurunkan seluruh metrik, dengan penurunan terbesar pada SMOTE (−9,25%). Pada Random Forest, SMOTE meningkatkan akurasi, *recall*, dan *F1-Score* secara signifikan, yaitu 5,18% pada akurasi, menghasilkan kombinasi terbaik (akurasi 93,08%, presisi 93,35, *recall* 93,08, *F1-Score* 93,07). Hal ini menunjukkan bahwa SMOTE efektif untuk algoritma berbasis pohon, sementara Naïve Bayes lebih optimal tanpa penyeimbangan data tambahan.

Sejarah Artikel

Submitted: 11 September 2025

Accepted: 14 September 2025

Published: 15 September 2025

Kata Kunci

data mining, klasifikasi, naive bayes, random forest, SMOTE, undersampling.

1. Pendahuluan

Data mining digunakan untuk mengekstrak informasi berharga dari kumpulan data yang besar dan kompleks. Proses ini melibatkan berbagai teknik dan metode dari beberapa disiplin ilmu, termasuk statistik, *machine learning*, dan kecerdasan buatan (Saputra et al., 2021). Teknik *data mining* dapat mengidentifikasi korelasi, pola, dan penemuan pengetahuan dari dataset, dan telah berhasil diterapkan di berbagai sektor seperti ritel, pemasaran, kesehatan, dan lain sebagainya (Singh & Singh, 2024).

Salah satu teknik utama dalam *data mining* adalah klasifikasi, yaitu metode untuk memprediksi kategori atau label suatu data berdasarkan kombinasi fitur tertentu. Di antara berbagai algoritma klasifikasi yang umum digunakan, Naïve Bayes dan Random Forest menjadi dua algoritma yang sering diterapkan baik dalam penelitian maupun praktik industri karena karakteristik dan keunggulan masing-masing. Naïve Bayes dikenal karena kesederhanaannya serta kemampuannya menangani fitur kategorial secara efisien (Meidina & Abidin, 2023; Wibowo et al., 2023; Halasz et al., 2021), sementara Random Forest unggul dalam menangani data kompleks, termasuk data yang tidak seimbang serta adanya *noise* dan *outlier* (Puspa et al., 2023).

Penelitian ini berfokus pada analisis komparatif terhadap performa kedua algoritma tersebut dalam mengklasifikasikan data Bank Marketing yang diperoleh dari UCI *Machine Learning Repository*. Dataset ini memiliki karakteristik ketidakseimbangan kelas yang signifikan, sehingga sesuai untuk diuji dengan berbagai teknik praproses data. Penanganan imbalanced data dilakukan dengan tiga pendekatan: tanpa penanganan khusus, menggunakan

teknik SMOTE (Synthetic Minority Over-sampling Technique), dan *undersampling*. *Imbalanced data* adalah kondisi distribusi kelas yang tidak seimbang, di mana jumlah data pada satu kelas jauh lebih banyak dibandingkan kelas lainnya, sehingga model cenderung bias terhadap kelas mayoritas dan kurang akurat dalam mengenali kelas minoritas(Asassfeh et al., 2023).

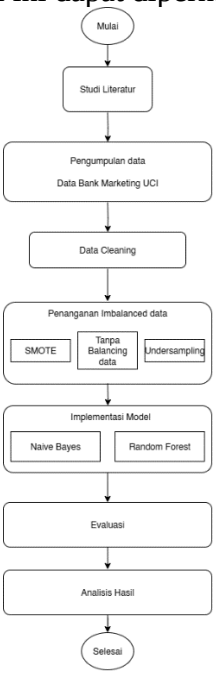
SMOTE adalah metode untuk mengatasi ketidakseimbangan data pada *machine learning* dengan menghasilkan sampel baru pada kelas minoritas, sehingga distribusi kelas menjadi lebih seimbang. Teknik ini juga mencegah risiko duplikasi data yang sering terjadi pada *random oversampling* (Byeon, 2021). Metode *undersampling* adalah teknik penanganan ketidakseimbangan data dengan mengurangi jumlah sampel pada kelas mayoritas agar seimbang dengan kelas minoritas, sehingga dapat mengurangi bias model akibat dominasi kelas mayoritas dan meningkatkan kinerja prediksi (Liu et al., 2008).

Pada penelitian yang dilakukan oleh Prakoso Indaryono, yang menunjukkan bahwa akurasi Naive Bayes mencapai 36,61% dan Random Forest 86,55% (Indaryono, 2024), pada penelitian oleh (Momole, 2022) dalam klasifikasi bahasa daerah algoritma Naive Bayes mendapat akurasi 99% dan Random Forest 65%. Penelitian lain oleh (Leonardo et al., 2020) menunjukkan skor akurasi 85% untuk Naive Bayes dan 90% untuk Random Forest dengan data yang hampir sama. Pada penelitian (Fitriani & Febrianto, 2021) algoritma Random Forest dengan metode SMOTE mencapai hasil 92,61% yang mana meningkat dari sebelumnya hanya 90, 38%, sedangkan untuk algoritma Naive Bayes dengan SMOTE mencapai hasil 82,17% yang mana hasil ini turun dibandingkan hasil tanpa SMOTE, yaitu 88%.

Dengan pendekatan tersebut, penelitian ini bertujuan untuk mengevaluasi dan membandingkan akurasi, presisi, *recall*, dan *F1-Score* dari kedua algoritma secara menyeluruh. Evaluasi dilakukan menggunakan *confusion matrix* sebagai alat ukur performa klasifikasi. Hasil penelitian ini diharapkan dapat memberikan wawasan mengenai efektivitas masing-masing algoritma dalam merespons tantangan umum dalam klasifikasi, khususnya dalam konteks ketidakseimbangan data.

2. Metodologi Penelitian

Langkah pengerjaan pada penelitian ini dapat diperhatikan pada gambar 1.



Gambar 1. Metodologi penelitian

Alur penelitian dimulai dengan tahap studi literatur untuk memahami teori, metode, dan penelitian terdahulu yang relevan dengan topik. Selanjutnya dilakukan pengumpulan data menggunakan dataset *Bank Marketing* dari UCI *Machine Learning Repository*. Dataset kemudian melalui tahap *data cleaning* untuk memastikan kualitas data dengan menghapus atau memperbaiki data yang hilang, duplikat, atau tidak konsisten. Setelah itu, dilakukan penanganan *imbalanced data* menggunakan tiga skenario, yaitu tanpa penyeimbangan data (tanpa balancing data), *Synthetic Minority Oversampling Technique* (SMOTE), dan *undersampling*. Data yang telah melalui tahap tersebut kemudian digunakan pada tahap implementasi model dengan menerapkan dua algoritma klasifikasi, yaitu Naïve Bayes dan Random Forest. Model yang dihasilkan dievaluasi menggunakan metrik kinerja yang sesuai untuk menilai performa masing-masing kombinasi metode. Penelitian diakhiri dengan tahap analisis hasil dan kesimpulan, yang menganalisis dan merangkum temuan utama berdasarkan hasil evaluasi.

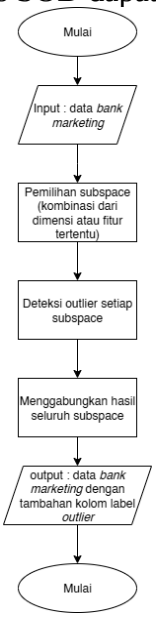
2.1 Pengumpulan Data

Penelitian ini menggunakan dataset *Bank Marketing* versi *additional full* yang diperoleh dari UCI *Machine Learning Repository*. Dataset ini berisi 41.188 baris dan 21 kolom, yang merekam aktivitas pemasaran bank di Portugal melalui sambungan telepon. Terdapat 20 fitur dan 1 label klasifikasi (yes/no) yang menunjukkan apakah nasabah tertarik dengan produk yang ditawarkan. Fitur dalam data terdiri dari 10 fitur numerik dan 10 fitur kategorial. Sebagian besar nasabah dalam data berprofesi sebagai admin atau *blue-collar*, berstatus menikah, dan merupakan lulusan sekolah menengah. Distribusi label menunjukkan ketidakseimbangan data, dengan 36.548 data berlabel ‘no’ (88,73%) dan hanya 4.640 berlabel ‘yes’ (11,27%). Oleh karena itu, dataset ini dikategorikan sebagai data tidak seimbang dan relevan untuk dieksplorasi dengan berbagai teknik penyeimbangan data.

2.2 *Exploratory Data Analysis* (EDA)

Tahap EDA dilakukan untuk memahami struktur dan karakteristik awal data, mencakup identifikasi tipe data, nilai yang hilang, duplikasi data, serta distribusi kelas target. Analisis ini dilakukan menggunakan pustaka pandas untuk memeriksa atribut-atribut dasar dan kualitas data. Hasil EDA menjadi dasar dalam menentukan langkah praproses selanjutnya.

Selain itu, deteksi *outlier* dilakukan menggunakan metode Subspace Outlier Detection (SOD) yang diimplementasikan melalui pustaka pyod. SOD digunakan untuk mengidentifikasi data yang menyimpang dalam ruang berdimensi tinggi, dengan memberikan label “1” untuk *outlier* dan “0” untuk data normal. Proses SOD dapat diperhatikan pada gambar 2.



Gambar 2. Proses SOD

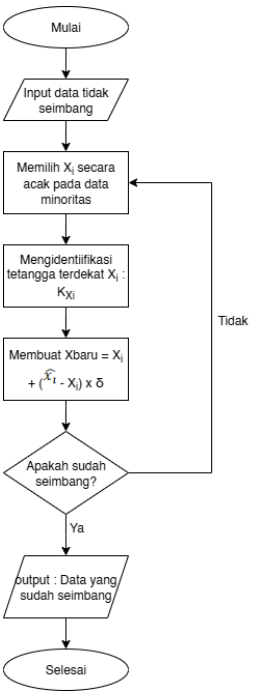
Proses Subspace Outlier Detection (SOD) dimulai dengan memasukkan data *bank marketing* sebagai input. Langkah pertama adalah melakukan pemilihan *subspace*, yaitu kombinasi dari dimensi atau fitur tertentu yang relevan untuk analisis. Selanjutnya, pada setiap *subspace* dilakukan deteksi *outlier* secara terpisah untuk mengidentifikasi data yang menyimpang dari pola umum. Hasil deteksi dari seluruh *subspace* kemudian digabungkan untuk mendapatkan gambaran komprehensif mengenai *outlier* pada data. Tahap akhir adalah menghasilkan *output* berupa data *bank marketing* yang telah dilengkapi dengan kolom tambahan berisi label *outlier*, sehingga data siap digunakan untuk analisis lanjutan atau pemodelan.

2.3 *Data Cleaning*

Proses *data cleaning* dilakukan untuk menghapus nilai-nilai yang tidak relevan guna meningkatkan akurasi model. Tahapan ini mencakup penghapusan data duplikat dan *outlier* menggunakan pustaka pandas. Pembersihan dilakukan sebelum penanganan data tidak seimbang agar distribusi data yang dihasilkan lebih representatif dan tidak dipengaruhi oleh nilai-nilai yang tidak valid.

2.4 Penanganan *Imbalanced Data*

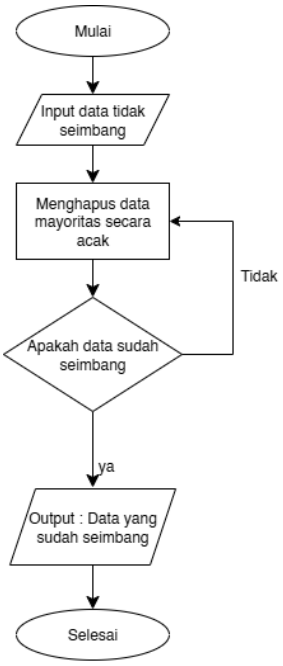
Untuk mengatasi ketidakseimbangan data, penelitian ini menerapkan dua skenario penyeimbangan. Pertama, menggunakan metode Synthetic Minority Over-sampling Technique (SMOTE) yang menghasilkan sampel sintetis pada kelas minoritas berdasarkan kedekatan antar tetangga terdekat. Kedua, menggunakan metode *undersampling* dengan mengurangi jumlah sampel dari kelas mayoritas secara acak agar distribusi kelas menjadi seimbang. Proses penanganan *imbalanced data* menggunakan SMOTE dapat dilihat pada gambar 3, dan proses penanganan *inbalanced data* menggunakan *undersampling* dapat diperhatikan pada gambar 4.



Gambar 3. Proses SMOTE

Proses SMOTE (Synthetic Minority Over-sampling Technique) dimulai dengan memasukkan data yang tidak seimbang, di mana jumlah data pada kelas minoritas lebih sedikit daripada kelas mayoritas. Langkah pertama adalah memilih secara acak satu sampel dari kelas minoritas. Kemudian, algoritma mencari tetangga terdekat dari sampel tersebut. Setelah itu,

dibuat data sintetis baru dengan cara mengambil selisih nilai antara sampel terpilih dan tetangganya, lalu mengalikannya dengan bilangan acak antara nol hingga satu, dan hasilnya ditambahkan kembali ke nilai sampel awal. Proses ini menghasilkan titik baru yang berada di antara dua sampel kelas minoritas. Langkah ini diulang hingga jumlah data pada kelas minoritas bertambah dan distribusi kelas menjadi seimbang. Setelah seimbang, data siap digunakan untuk pelatihan model.



Gambar 4. Proses *undersampling*

Dapat diperhatikan pada gambar 4, proses *undersampling* dimulai dengan memasukkan data yang tidak seimbang, di mana jumlah sampel pada kelas mayoritas jauh lebih banyak dibandingkan kelas minoritas. Langkah pertama adalah menghapus sebagian data dari kelas mayoritas secara acak untuk mengurangi jumlahnya. Setelah itu, sistem memeriksa apakah jumlah sampel di kedua kelas sudah seimbang. Jika belum, proses penghapusan data mayoritas diulangi hingga distribusi kelas menjadi seimbang. Setelah kondisi seimbang tercapai, keluaran yang dihasilkan adalah data dengan jumlah kelas yang seimbang, sehingga siap digunakan untuk pelatihan model tanpa bias akibat ketidakseimbangan kelas.

2.5 Implementasi Model

Tahap selanjutnya adalah implementasi model menggunakan dua algoritma klasifikasi, yaitu Naive Bayes dan Random Forest. Model dibangun menggunakan data latih, sedangkan data uji digunakan untuk mengukur performa model terhadap data yang belum pernah dilihat.

Naive Bayes merupakan algoritma klasifikasi berbasis teorema Bayes yang mengasumsikan bahwa seluruh fitur bersifat independen secara bersyarat terhadap label. Algoritma ini menghitung probabilitas posterior setiap kelas dan memilih kelas dengan probabilitas tertinggi sebagai hasil prediksi (Saepudin et al., 2023).

Sementara itu, Random Forest merupakan algoritma *ensemble* berbasis pohon keputusan yang dibangun secara paralel menggunakan metode *bootstrap sampling* dan hasil prediksi ditentukan melalui voting mayoritas. Setiap pohon dilatih pada subset data acak sehingga meningkatkan generalisasi dan mengurangi *overfitting* (Yang et al., 2023).

2.6 Evaluasi

Tahap evaluasi dilakukan menggunakan *confusion matrix* untuk membandingkan hasil prediksi model dengan label aktual. Evaluasi ini menghasilkan metrik kinerja seperti akurasi, presisi, *recall*, dan *F1-score*. Sebanyak 6 model dievaluasi, yang mana 3 model untuk masing-masing algoritma, yaitu Naive Bayes dan Random Forest.

Keenam model tersebut mencakup berbagai kombinasi antara kondisi data (tanpa penyeimbangan, SMOTE, dan *undersampling*) dan dua algoritma (Naive Bayes dan Random Forest). Evaluasi ini bertujuan untuk membandingkan kinerja model berdasarkan strategi praproses yang diterapkan pada setiap model.

2.7 Analisis Hasil

Analisis hasil dilakukan berdasarkan ecaluasi yang telah dilakukan. Hal yang dianalisis mencakup akurasi model, presisi, *recall*, dan skor F1. Hasil yang diperoleh dari model Naive Bayes dan Random Forest akan dibandingkan untuk melihat pengaruh dari seleksi fitur dan penanganan *imbalance data*. Akurasi model mencakup seberapa baik model dapat memprediksi kelas dengan benar pada data uji.

3. Hasil dan Pembahasan

Bab ini memaparkan hasil penelitian beserta pembahasannya, dimulai dari Exploratory Data Analysis (EDA), proses pembersihan data (*data cleaning*), penanganan *imbalanced data*, hingga evaluasi kinerja model.

3.1 Hasil Exploratory Data Analysis (EDA)

Hasil Exploratory Data Analysis (EDA) dapat diperhatikan pada gambar 5.

```
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             41188 non-null  int64
 1   job             41188 non-null  object
 2   marital         41188 non-null  object
 3   education       41188 non-null  object
 4   default         41188 non-null  object
 5   housing         41188 non-null  object
 6   loan            41188 non-null  object
 7   contact         41188 non-null  object
 8   month           41188 non-null  object
 9   day_of_week     41188 non-null  object
 10  duration        41188 non-null  int64
 11  campaign        41188 non-null  int64
 12  pdays           41188 non-null  int64
 13  previous        41188 non-null  int64
 14  poutcome        41188 non-null  object
 15  emp.var.rate    41188 non-null  float64
 16  cons.price.idx  41188 non-null  float64
 17  cons.conf.idx   41188 non-null  float64
 18  euribor3m       41188 non-null  float64
 19  nr.employed     41188 non-null  float64
 20  y               41188 non-null  object
dtypes: float64(5), int64(5), object(11)
```

Gambar 5. Karakteristik Data

Berdasarkan pada gambar 5, tidak ada data yang hilang dan memiliki 10 fitur dengan tipe data numerik (*age, duration, campaign, pdays, previous, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m,* dan *nr.employed*) serta 10 fitur dengan tipe data kategorial (*job, marital, education, default, housing, loan, contact, month, day_of_week,* dan *poutcome*). Selain mencari tahu karakteristik data dan *missing values*, juga dilakukan pengecekan apakah ada data duplikat dan data yang *outlier*. Pengecekan data *outlier* akan menggunakan *Subspace Outlier Detection*. *Subspace Outlier Detection* bekerja dengan mengukur sejauh mana suatu titik menyimpang dari subruang lokal yang didefinisikan tetangganya. SOD berguna untuk mendeteksi *outlier* yang tidak terlihat jelas di seluruh dimensi, tapi sangat menyimpang di subruang tertentu. Pada SOD, setiap baris data akan diberikan skor dan label *outlier*. Data *outlier* akan diberikan label 1.

Hasil pengeceken *outlier* dan duplikat dapat diperhatikan pada gambar 6.

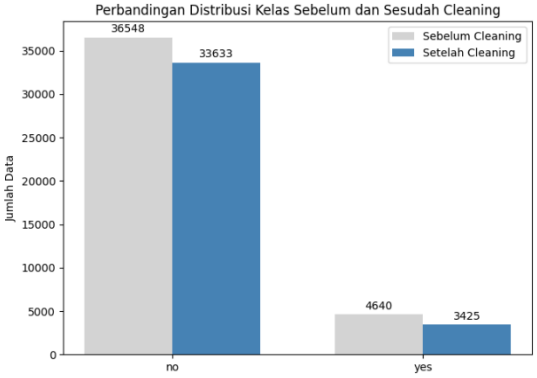
```
jumlah data duplikat :  12
     outlier_count  total_count  percentage
y
no            2892        36548    7.912882
yes           1227         4640   26.443966

(4119, 10.000485578323785)
```

Gambar 6. Hasil Pengecekan *Outlier* dan Duplikat

Dari hasil pada gambar 6 dapat dilihat bahwa data duplikat pada data *bank marketing* terdapat 12 dan data *outlier* pada label "no" terdapat 2829 data dan pada label "yes" terdapat 1227 data.

3.2 Hasil *Data Cleaning*

Setelah dilakukan pengumpulan data, data ini dibersihkan dari data yang duplikat serta dibersihkan dari data yang *outlier* (data yang nilainya jauh berbeda dari nilai lainnya dalam sekumpulan data). Hasil pembersihan data dapat diperhatikan pada gambar 7.

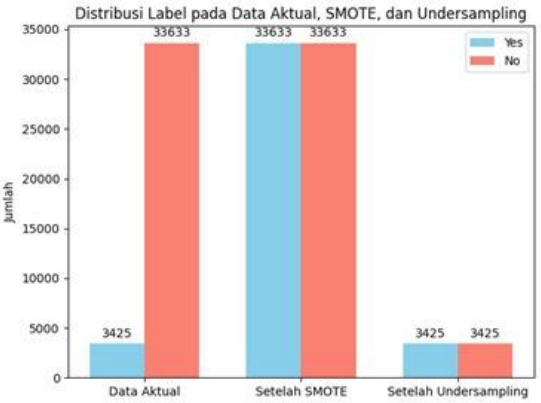


Gambar 7. Hasil Pembersihan Data

Setelah proses pembersihan data, jumlah data berlabel Yes berkurang dari 4.640 menjadi 3.425, sedangkan data berlabel No berkurang dari 36.548 menjadi 33.633. Berikut diagram perbandingan

3.3 Hasil Penanganan *Imbalanced Data*

Hasil penanganan *imbalance data* Dapat diperhatikan pada gambar 8.



Gambar 8. Hasil Penanganan *Imbalance data*

Setelah dilakukan penyeimbangan data menggunakan SMOTE, data dengan label “yes” mengalami peningkatan jumlah dari sebelumnya 3.245 menjadi 33.633 dan pada label “no” jumlah data tetap 33.633. Sedangkan pada penyeimbangan data menggunakan undersampling, data dengan label “yes” tetap 3.425 dan data dengan label “no” mengalami pengurangan jumlah dari 33.633 menjadi 3.425.

3.4 Hasil Evaluasi

Berdasarkan hasil evaluasi pada tabel 1, metode Random Forest dengan SMOTE menunjukkan kinerja terbaik di seluruh metrik evaluasi dengan akurasi sebesar 93,08%, presisi 93,35%, recall 93,08%, dan F1-score 93,07%. Pada algoritma Naïve Bayes, performa terbaik justru diperoleh dari model default tanpa penanganan ketidakseimbangan data, dengan akurasi 91,78%, presisi 90,59%, recall 91,78%, dan F1-score 90,93%. Penerapan SMOTE pada Naïve Bayes menghasilkan penurunan kinerja menjadi 82,53% di seluruh metrik, sedangkan metode undersampling memberikan hasil sedikit lebih baik dengan akurasi 83,36%. Sementara itu, pada Random Forest, penggunaan undersampling menghasilkan kinerja yang cukup baik dengan akurasi 89,34%, namun masih lebih rendah dibandingkan model default dan SMOTE. Temuan ini menunjukkan bahwa penanganan ketidakseimbangan data memberikan pengaruh yang bervariasi tergantung pada algoritma yang digunakan, di mana SMOTE cenderung lebih menguntungkan untuk Random Forest, tetapi tidak untuk Naïve Bayes.

3.5 Analisis Hasil

Berdasarkan hasil evaluasi setiap model dan perbandingannya dengan model *default* (tanpa penanganan *imbalanced data)* pada tabel 1, terlihat bahwa performa setiap algoritma dipengaruhi secara signifikan oleh metode penanganan ketidakseimbangan data yang digunakan. Pada Naïve Bayes, model default memberikan performa terbaik dengan akurasi 91,78%, presisi 90,59%, recall 91,78%, dan F1-score 90,93. Nilai presisi dan recall yang seimbang menunjukkan bahwa model mampu mengenali kedua kelas dengan proporsi yang hampir setara. Penerapan SMOTE justru menurunkan performa secara drastis dengan penurunan akurasi sebesar 9,25%, presisi 8,06%, recall 9,25%, dan F1-score 8,4%. Hal ini mengindikasikan bahwa data sintetis yang dihasilkan SMOTE tidak selaras dengan distribusi asli data, sehingga mengganggu asumsi independensi fitur pada Naïve Bayes. Undersampling juga menurunkan performa, meskipun dampaknya sedikit lebih kecil dibanding SMOTE, dengan penurunan akurasi sebesar 8,42% dan F1-score 7,63%. Penurunan ini disebabkan oleh hilangnya sebagian informasi penting dari kelas mayoritas akibat penghapusan data.

Sementara itu, pada Random Forest, model default memiliki akurasi 87,9% dan presisi tertinggi di seluruh eksperimen (93,51%), namun recall yang lebih rendah (87,9%) menunjukkan model lebih konservatif dalam memprediksi kelas positif. Penerapan SMOTE pada Random Forest menghasilkan peningkatan signifikan pada akurasi dan recall sebesar 5,18% dengan penurunan presisi yang sangat kecil (0,16%), menghasilkan keseimbangan yang optimal antara semua metrik (akurasi 93,08%, presisi 93,35%, recall 93,08%, F1-score 93,07). Hal ini menunjukkan bahwa algoritma berbasis ensemble seperti Random Forest mampu memanfaatkan variasi data sintetis untuk memperkaya representasi kelas minoritas tanpa kehilangan kemampuan generalisasi. Sebaliknya, metode undersampling hanya memberikan peningkatan kecil pada akurasi dan recall (1,44%), namun mengorbankan presisi cukup besar (-3,7%), sehingga meningkatkan risiko false positive.

Secara keseluruhan, hasil ini menunjukkan bahwa pemilihan metode penanganan ketidakseimbangan data harus mempertimbangkan karakteristik algoritma. Naïve Bayes bekerja lebih baik tanpa modifikasi distribusi data, sedangkan Random Forest mendapat manfaat signifikan dari SMOTE. Dalam implementasi praktis, SMOTE dapat direkomendasikan untuk algoritma berbasis pohon keputusan yang memiliki toleransi tinggi terhadap variasi data, sedangkan untuk algoritma probabilistik seperti Naïve Bayes, menjaga distribusi data asli cenderung lebih menguntungkan.

4. Kesimpulan

Berdasarkan analisis hasil, metode penanganan ketidakseimbangan data memberikan dampak berbeda pada tiap algoritma karena perbedaan cara kerja masing-masing. Naïve Bayes bekerja optimal pada model default karena metode ini sangat bergantung pada distribusi probabilitas asli data. Penambahan data sintetis melalui SMOTE dapat mengubah distribusi fitur, sehingga mengganggu asumsi independensi antar fitur dan menurunkan akurasi hingga lebih dari 9%. Undersampling juga mengurangi performa karena menghilangkan sebagian informasi penting dari kelas mayoritas. Sebaliknya, Random Forest memperoleh peningkatan tertinggi melalui SMOTE, dengan kenaikan akurasi lebih dari 5%, karena algoritma berbasis pohon keputusan mampu memanfaatkan variasi data sintetis untuk memperluas representasi kelas minoritas dan mengurangi bias prediksi, meski presisi sedikit turun akibat meningkatnya peluang false positive. Hal ini menunjukkan bahwa efektivitas metode balancing sangat dipengaruhi oleh sifat dasar algoritma yang digunakan.

Daftar Pustaka:

Asassfeh, M. R. (s), Rasmi, M., Alqammaz, A., Doumi, A. B., Al-Qawasmi, K., & Al-Shaikh, A. (2023). ENHANCING IMBALANCED DATA CLASSIFICATION: A CASE STUDY OF PORTUGUESE BANK MARKETING. *Journal of Southwest Jiaotong University*, *58*(6). https://doi.org/10.35741/issn.0258-2724.58.6.21

Byeon, H. (2021). Predicting the depression of the South Korean elderly using SMOTE and an imbalanced binary dataset. *International Journal of Advanced Computer Science and Applications*, *12*(1).

Fitriani, M. A., & Febrianto, D. C. (2021). Data mining for potential customer segmentation in the marketing bank dataset. *JUITA: Jurnal Informatika*, *9*(1), 25–32.

Halasz, G., Sperti, M., Villani, M., Michelucci, U., Agostoni, P., Biagi, A., Rossi, L., Botti, A., Mari, C., & Maccarini, M. (2021). Predicting clinical outcomes in the Machine Learning era: The Piacenza score a purely data driven approach for mortality prediction in COVID-19 Pneumonia. *MedRxiv*, 2021–2023.

Indaryono, N. A. P. (2024). Analisa Perbandingan Algoritma Random Forest Dan Naïve Bayes Untuk Klasifikasi Curah Hujan Berdasarkan Iklim Di Indonesia. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *9*(1), 158–167.

Leonardo, R., Pratama, J., & Chrisnatalis, C. (2020). Perbandingan Metode Random Forest Dan Naïve Bayes Dalam Prediksi Keberhasilan Klien Telemarketing. *Jurnal Teknologi Dan Ilmu Komputer Prima (Jutikomp)*, *3*(2), 455–459.

Liu, X.-Y., Wu, J., & Zhou, Z.-H. (2008). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(2), 539–550.

Meidina, A., & Abidin, Z. (2023). Diagnosis of heart disease using optimized naïve Bayes algorithm with particle swarm optimization and gain ratio. *Recursive Journal of Informatics*, *1*(2), 47–54.

Momole, G. M. (2022). Perbandingan Naïve Bayes dan Random Forest Dalam Klasifikasi Bahasa Daerah. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *9*(2), 855–863.

Puspa, S. D., Puspitasari, F., Riyono, J., Pujiastuti, C. E., Bijlsma, D. L., & Leo, J. A. (2023). Customer Segmentation Analysis Using Random Forest & Naïve Bayes Method In The Case of Multi-Class Classification at PT. XYZ. *Mathline: Jurnal Matematika Dan Pendidikan Matematika*, *8*(4), 1359–1372.

Saepudin, S., Widiastuti, S., & Irawan, C. (2023). Sentiment analysis of social media platform reviews using the Naïve Bayes classifier algorithm. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, *12*(2), 236–243.

Saputra, D., Irmayani, W., Purwaningtias, D., Sidauruk, J., & Gurbuz, B. (2021). A Comparative Analysis of C4. 5 Classification Algorithm, Naïve Bayes and Support Vector Machine Based on Particle Swarm Optimization (PSO) for Heart Disease Prediction. *International Journal of Advances in Data and Information Systems*, *2*(2), 84–95.

Singh, P., & Singh, N. (2024). Role of data mining techniques in bioinformatics. In *Research Anthology on Bioinformatics, Genomics, and Computational Biology* (pp. 1406–1417). IGI Global Scientific Publishing.

Wibowo, R., Soeleman, M. A., & Affandy, A. (2023). Hybrid Top-K feature selection to improve high-dimensional data classification using naïve bayes algorithm. *Scientific Journal of Informatics*, *10*(2), 113–120.

Yang, Z., Cui, X., & Song, Z. (2023). Predicting sepsis onset in ICU using machine learning models: a systematic review and meta-analysis. *BMC Infectious Diseases*, *23*(1), 635.

Tabel 1. Perbandingan hasil evaluasi terhadap skenario default

Algoritma	Akurasi	Presisi	*Recall*	*F1-Score*	Δ Akurasi	ΔPresisi	Δ*Recall*	Δ*F1-Score*
Naïve Bayes + SMOTE	82,53	82,53	82,53	82,53	-9,25	-8,06	-9,25	-8,4
Naïve Bayes (Default)	91,78	90,59	91,78	90,93	0	0	0	0
Naïve Bayes + Undersampling	83,36	83,82	83,36	83,3	-8,42	-6,77	-8,42	-7,63
Random Forest+ SMOTE	93,08	93,35	93,08	93,07	5,18	-0,16	5,18	3,4
Random Forest (Default)	87,9	93,51	87,9	89,67	0	0	0	0
Random Forest + Undersampling	89,34	89,81	89,34	89,31	1,44	-3,7	1,44	-0,36