Dampak Smpte Terhadap Kinerja Random Forest Classifier Berdasarkan Data Tidak Seimbang

by Nurliana Nasution

Submission date: 03-Jan-2023 03:31PM (UTC+0700)
Submission ID: 1988159403
File name: 2022_-_1726-Article_Text-11081-2-10-20220802.pdf (862.04K)
Word count: 6333
Character count: 40257

Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 21, No. 3, Juli 2022, pp. 677∼690
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v21i3.1726

Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak seimbang

Impact of SMOTE on Random Forest Classifier Performance based on Imbalanced Data

Erlin[1], Yenny Desnelita[2], Nurliana Nasution[3], Laili Suryati[4], Fransiskus Zoromi[5]
[1,2]Institut Bisnis dan Teknologi Pelita, Indonesia
[3]Universitas Lancang Kuning, Indonesia
[4]Universitas Persada, Indonesia
[5]STMIK Amik Riau, Indonesia

Informasi Artikel

Genesis Artikel:

Diterima, 27 Januari 2022
Direvisi, 25 Maret 2022
Disetujui, 09 Juni 2022

Kata Kunci:

Data Tidak seimbang
Machine Learning
Overfitting
Random Forest
SMOTE

ABSTRAK

Dalam aplikasi *machine learning* sangat umum ditemukan kumpulan data dalam berbagai tingkat ketidakseimbangan mulai dari ketidakseimbangan kecil, sedang sampai ekstrim. Sebagian besar model *machine learning* yang dilatih pada data tidak seimbang akan memiliki bias dengan memberikan tingkat akurasi yang tinggi pada kelas mayoritas dan sebaliknya rendah pada kelas minoritas. Tujuan penelitian ini adalah untuk mengevaluasi dampak dari SMOTE (*Synthetic Minority Oversampling Technique*) pada pengklasifikasi *Random Forest* untuk memprediksi penyakit jantung. Data berjumlah 299 berasal dari UCI *Machine learning Repository* digunakan untuk membangun model prediksi berdasarkan 12 variabel independen dan 1 variabel dependen. Kelas minoritas dalam dataset pelatihan di *oversampling* menggunakan teknik SMOTE (*Synthetic Minority Oversampling Technique*). Model dievaluasi tidak hanya menggunakan ukuran kinerja *Accuracy* dan *Precision* saja, namun juga menggunakan alternatif ukuran kinerja lainnya seperti *Sensitivity*, F1-*score*, *Specificity*, *G-Mean* dan *Youdens Index* yang lebih baik digunakan untuk data yang tidak seimbang. Hasil penelitian menunjukkan bahwa teknik SMOTE (*Synthetic Minority Oversampling Technique*) mampu mengurangi *overfitting* sekaligus meningkatkan kinerja model Random *Forest* pada semua indikator. Peningkatan skor *Accuracy* sebesar 3.45%, *Precision* 4.8%, *Sensitivity* 7.1%, F1-*score* 4.8%, *Specificity* 2.1%, *G-Mean* 4.4%, dan *Youdens Index* 6.3%. Penelitian ini membuktikan bahwa dalam menentukan pengklasifikasi dengan algoritma *machine learning* seperti *Random Forest*, kemiringan kelas dalam data perlu diperhitungkan dan diseimbangkan untuk hasil kinerja yang lebih baik.

Keywords:

Imbalanced Data
Machine Learning
Overfitting
Random Forest
SMOTE

ABSTRACT

In machine learning applications, it is prevalent to find datasets in various levels of imbalance ranging from small, moderate to extreme imbalances. Most machine learning models that are trained on imbalanced data will have a bias by providing a high level of Accuracy for the majority class and low on the minority class. This study aimed to evaluate the impact of the SMOTE (Synthetic Minority Oversampling Technique) on the Random Forest classifier for predicting heart disease. A total of 299 data from the UCI Machine learning Repository was used to build a prediction model based on 12 independent variables and 1 dependent variable. The minority class in the training dataset was oversampled using the SMOTE (Synthetic Minority Oversampling Technique) technique. The model is evaluated not only using performance measures of Accuracy and Precision but also using alternative performance measures such as Sensitivity, F1-score, Specificity, G-Mean, and Youden's Index which are better used for imbalanced data. The results showed that the SMOTE (Synthetic Minority Oversampling Technique) technique was able to reduce overfitting while increasing the performance of the Random Forest model on all indicators. Improved Accuracy scores of 3.45%, Precision 4.8%, Sensitivity 7.1%, F1-score 4.8%, Specificity 2.1%, G-Mean 4.4%, and Youden's Index 6.3%. This study proves that in determining classifiers with machine learning algorithms such as Random Forest, the skew of the class in the data needs to be taken into account and balanced for better performance.

Penulis Korespondensi:

Erlin,
Program Studi Teknik Informatika,
Institut Bisnis dan Teknologi Pelita, Indonesia
Email: erlin@lecturer.pelitaindonesia.ac.id

Journal homepage: *https://journal.universitasbumigora.ac.id/index.php/matrik*

1. PENDAHULUAN

Penerapan *machine learning* sudah sangat populer dalam berbagai bidang termasuk bidang studi medis. *Machine learning* menyediakan berbagai layanan manipulasi seperti mengeksplorasi pola yang tidak diketahui, melakukan proses klasifikasi, *clustering*, deteksi anomali data, meningkatkan model prediksi klinis serta membantu dalam pengambilan keputusan medis [1–3]. Pemanfaatan teknik ini dalam berbagai disiplin ilmu telah berkembang dan menunjukkan kontribusi pada ilmu pengetahuan termasuk dalam bidang kesehatan dan kedokteran. Beberapa algoritma *machine learning* yang paling umum digunakan dalam pemodelan prediksi medis diantaranya *Deep Learning* [4], algoritma C4.5 [5], *Naïve Bayes* [6], *Support Vector Machine* [7], *Artificial Neural Network* [8], *Logistic Regression* [9], dan *Random Forest* [10, 11]. Sebagian besar teknik dan algoritma pemodelan ini bekerja dengan sangat baik ketika distribusi kelas dalam dataset terdistribusi secara merata. Namun kenyataannya, sebagian besar kelas *dataset* tidak seimbang. Ini biasanya terjadi ketika kelas mayoritas lebih banyak dibandingkan kelas minoritas. Dalam aplikasi *machine learning*, sangat umum ditemukan kumpulan data (*dataset*) dengan berbagai tingkat ketidak seimbangan kelas, mulai dari ketidak seimbangan sedang seperti diagnosis medis dimana 10% didiagnosis menderita penyakit dan 90% sebaliknya, sampai ketidak seimbangan ekstrim, misal dari deteksi anomali transaksi perbankan, dimana ditemukan 1 transaksi curang atau palsu dari 10.000 transaksi yang terjadi. Sebagian besar model yang dilatih pada data yang tidak seimbang akan memiliki bias dalam memprediksi kelas yang besar dan mengabaikan kelas yang lebih kecil.

Ketika ada ketidak seimbangan kelas dalam data pelatihan, model *machine learning* biasanya akan mengklasifikasikan kelas yang lebih besar secara berlebihan karena probabilitas sebelumnya yang meningkat. Akibatnya, algoritma *machine learning* cenderung salah mengklasifikasikan kelas minoritas. Dampak lebih lanjut, model *machine learning* akan menghasilkan tingkat akurasi prediksi yang rendah pada kelas minoritas dan tinggi pada kelas mayoritas [12, 13]. Dalam banyak kasus penggunaan, seperti diagnosis medis, ini justru kebalikan dari apa yang ingin dicapai, karena sangat umum diketahui bahwa kelas minoritas (misalnya penderita penyakit) adalah kelas yang paling penting untuk diprediksi dengan benar karena merupakan kelas kritis dan sangat menentukan terhadap keberhasilan dan kinerja keseluruhan dari suatu model. Untuk mengatasi masalah ini, perlu menangani ketidak seimbangan kelas saat melatih model dalam beberapa cara. Terdapat sejumlah teknik untuk menangani kumpulan kelas yang tidak seimbang, baik pada tingkat data maupun algoritma. Pada tingkat data, teknik yang diadopsi secara luas adalah *resampling* seperti *oversampling* [14, 15] dan *undersampling* [16]. Teknik-teknik ini memodifikasi probabilitas sebelumnya dari kelas mayoritas dan minoritas dalam kumpulan data pelatihan untuk mendapatkan jumlah kasus yang lebih seimbang di setiap kelas. Pada *undersampling*, sejumlah data pada kelas mayoritas akan dihapus sedangkan pada *oversampling*, justru sebaliknya, sejumlah data pada kelas minoritas akan ditambahkan sehingga pada kedua teknik tersebut akan menghasilkan data yang seimbang. Namun teknik ini memiliki kelemahan. Implementasi paling sederhana dari *oversampling* adalah dengan menduplikasi data secara acak dari kelas minoritas, yang dapat menyebabkan resiko *overfitting* terhadap data yang langka dan tidak memberikan informasi tambahan apapun ke model. Pada *undersampling*, teknik paling sederhana melibatkan penghapusan atau membuang data secara acak dari kelas mayoritas, yang dapat menyebabkan hilangnya informasi. Untuk mengatasi masalah ini, dapat dilakukan dengan membuat data sintesis menggunakan teknik SMOTE (*Synthetic Minority Oversampling Technique*) untuk menyeimbangkan distribusi kelas dengan meningkatkan jumlah kelas minoritas untuk tujuan *oversampling*.

Feng et al., [17] melakukan penelitian untuk menguji stabilitas teknik oversampling berbasis SMOTE (*Synthetic Minority Oversampling Technique*) untuk memprediksi cacat perangkat lunak. Hasil penelitian menunjukkan bahwa teknik SMOTE (*Synthetic Minority Oversampling Technique*) mampu membuat model lebih stabil dan memiliki kinerja lebih baik untuk pengukuran dari sisi AUC, *balance*, dan MCC. Penelitian selaras dilakukan oleh Mishra dan Singh [18] meneliti ketidakseimbangan kelas pada multilabel data menggunakan metode *Feature Construction and SMOTE (Synthetic Minority Oversampling Technique)-based Imbalance handling* (FCSMI). Hasil eksperimen menunjukkan efektivitas metode FCSMI sangat baik untuk menangani *imbalanced class* pada *dataset* yang digunakan. Peneliti lain fokus pada pendekatan tingkat data untuk menangani ketidak seimbangan kelas pada data beasiswa PPA menggunakan algoritma C4.5 yang disisipkan teknik SMOTE (*Synthetic Minority Oversampling Technique*). Hasil pengujian membuktikan bahwa teknik SMOTE (*Synthetic Minority Oversampling Technique*) mampu meningkatkan kinerja algoritma C4.5 untuk skor akurasi, sensitivitas, dan spesifisitas [19]. Meskipun terdapat beberapa publikasi dan penelitian mengenai teknik SMOTE (*Synthetic Minority Oversampling Technique*) dalam menangani ketidakseimbangan data pada berbagai kasus dengan metode dan algoritma yang berbeda-beda, namun sejauh pengetahuan penulis, belum ditemukan publikasi berkaitan dengan pengujian kehandalan teknik SMOTE (*Synthetic Minority Oversampling Technique*) menggunakan *Random Forest Classifier* untuk mendeteksi penyakit jantung menggunakan bahasa pemrograman *Python*. Selain itu, sebagian besar model pada penelitian terdahulu, dievaluasi menggunakan *Confusion Matrix* dengan pengukuran skor *Accuracy*, *Precision*, *Recall*, dan F1-*score*. Penelitian ini selain menggunakan *Confusion Matrix* untuk mengevaluasi model, juga menggunakan alternatif ukuran kinerja lainnya seperti *Sensitivity*, *Specificity*, *G-Mean*, dan *Youdens Index* yang lebih baik digunakan untuk data yang tidak seimbang.

Tujuan penelitian ini adalah untuk mengevaluasi dan menganalisis dampak dari SMOTE (*Synthetic Minority Oversampling Technique*) pada pengklasifikasi *Random Forest* untuk prediksi penyakit jantung. Teknik SMOTE (*Synthetic Minority Oversampling Technique*) akan diujicoba pada algoritma *Random Forest* sebagai salah satu algoritma yang populer untuk klasifikasi. Dampak yang akan dievaluasi tidak hanya berfokus kepada kinerja model tetapi juga terhadap kestabilan model yang dihasilkan. Bagian selanjutnya dari artikel ini disusun sebagai berikut: Bagian 2 membahas mengenai material dan metode penelitian. Hasil dan pembahasan didiskusikan pada bagian 3. Kesimpulan akan dijelaskan pada bagian 4.

2. METODE PENELITIAN

2.1. *Dataset* Penyakit Jantung

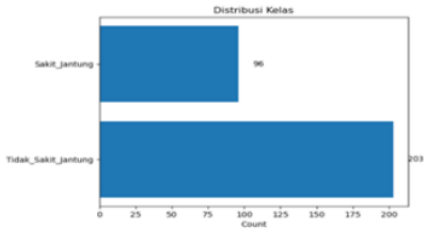
Dataset penderita penyakit jantung yang digunakan untuk eksperimen dalam penelitian ini menggunakan *dataset* umum yang terbuka untuk publik berasal dari UCI *machine learning repository* yang bisa diakses melalui link https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records. Versi original *dataset* dikumpulkan oleh Ahmad [20]. Versi yang digunakan dalam penelitian ini adalah versi *dataset* yang dielaborasi oleh Chicco [21]. *Dataset* berjumlah 299 data yang dibagi menjadi 2 bagian. 239 (80%) data digunakan untuk data latih (*training*) dan sisanya 60 (20%) data digunakan untuk data uji (*testing*). *Dataset* tersebut terdiri dari 13 variabel yang merupakan karakteristik input

seperti diperlihatkan pada Tabel 2 yang terdiri dari 12 variabel independen dan 1 variabel dependen. Variabel independen terdiri dari *age*, *anaemia*, *creatinine_phosphokinase*, *diabetes*, *ejection_fraction*, *high_blood_pressure*, *platelets*, *serum_creatinine*, *serum_sodium*, *sex*, *smoking* dan *time*, sedangkan variabel dependen merupakan target yang ingin diprediksi yaitu *death event*.

Tabel 1. Deskripsi *Dataset* yang digunakan

Fitur/Variabel	Deskripsi	Rentang
Age	Umur (tahun)	[40, . . . , 95]
Anaemia	Penurunan sel darah merah atau hemoglobin	0, 1
creatinine_phosphokinase (CPK)	Tingkat enzim CPK dalam darah (mcg/L)	[23, . . . , 7861]
diabetes	Apakah seorang pasien menderita diabetes	0, 1
ejection_fraction	Persentase darah yang dipompa keluar dari jantung selama satu kontraksi	[14, . . . , 80]
high_blood_pressure	Apakah pasien memiliki tekanan darah tinggi/hipertensi	0, 1
platelets	Trombosit dalam darah (kiloplatelet/mL)	[25.01, . . . , 850.00]
serum_creatinine	Tingkat kreatinin dalam darah (mg/dL)	[0.5, . . . , 9.40]
serum_sodium	Tingkat sodium dalam darah (mEq/L)	[114, . . . , 148]
sex	Jenis kelamin (1 = pria; 0 = wanita)	0, 1
smoking	Apakah pasien merokok atau tidak	0, 1
time	Periode tindak lanjut (Hari)	[4, . . . , 285]
death_event	[Target] peristiwa kematian: Jika pasien meninggal selama masa tindak lanjut.	0, 1

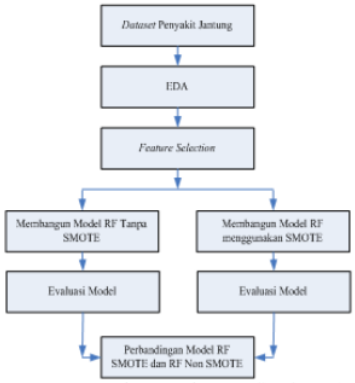
Dataset memiliki 2 (dua) kelas yaitu 96 sampel adalah kelas pasien penderita penyakit jantung dan 203 sampel adalah kelas yang tidak menderita penyakit jantung (non-penyakit jantung). Oleh karena itu, *dataset* ini merupakan *dataset* yang tidak seimbang. Gambar 1 menunjukkan distribusi kelas *dataset* penyakit jantung dimana kelas 0 (nol) atau kelas mayoritas lebih unggul dibandingkan dengan kelas 1 (satu) atau kelas minoritas.



Gambar 1. Distribusi Kelas *Dataset* Penyakit Jantung

2.2. Metode Penelitian

Rancangan penelitian SMOTE (*Synthetic Minority Oversampling Technique*) pada *Random Forest Classifier* diperlihatkan pada Gambar 2. Menentukan *dataset* penyakit jantung merupakan langkah pertama yang dilakukan dalam penelitian ini sebagai dasar terhadap data yang akan diproses dan dimanipulasi. Data berasal dari UCI *machine learning repository* berjumlah 299 sampel yang terdiri dari 13 variabel. Selanjutnya dilakukan *Exploratory Data Analysis* menggunakan *library Python* untuk mendapatkan gambaran data secara utuh. Seleksi fitur menjadi langkah selanjutnya untuk menentukan variabel yang berpengaruh terhadap kinerja pembentukan model. Skenario ujicoba akan dilakukan dalam 2 tahap, pertama membangun model tanpa SMOTE (*Synthetic Minority Oversampling Technique*) dan kedua membangun model menggunakan SMOTE (*Synthetic Minority Oversampling Technique*). Hasil kedua skenario ini akan dievaluasi untuk melihat hasil perbandingan dan menentukan dampak dari penggunaan teknik SMOTE (*Synthetic Minority Oversampling Technique*) tersebut.

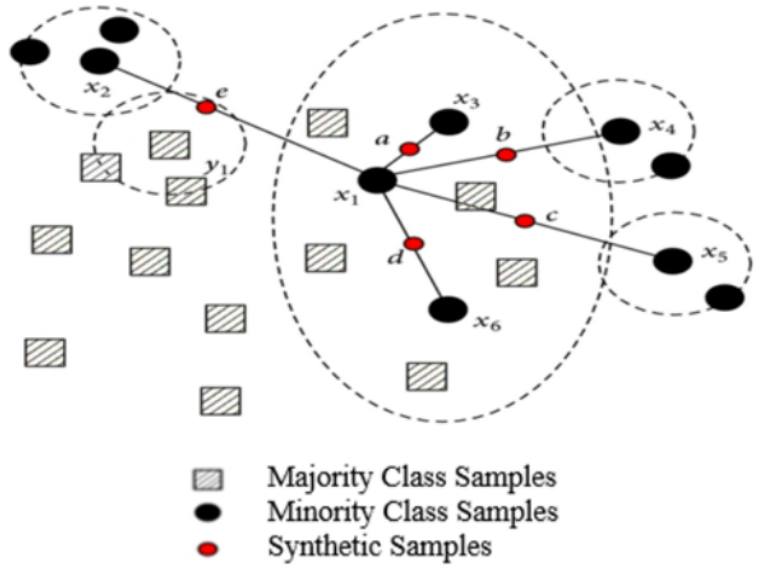


Gambar 2. Rancangan Penelitian *Synthetic Minority Oversampling Technique* SMOTE (*Synthetic Minority Oversampling Technique*) pada *Random Forest Classifier*

2.3. SMOTE (*Synthetic Minority Oversampling Technique*)

Data yang tidak seimbang menjadi masalah saat membuat model prediksi menggunakan *machine learning*. Salah satu cara untuk mengatasi masalah ini dengan melakukan *oversampling* pada data minoritas [22]. Teknik *oversampling* klasik memiliki kelemahan seperti *overfitting* dan hilangnya informasi. Implementasi paling sederhana dari *oversampling* adalah dengan menduplikasi data secara acak dari kelas minoritas, yang dapat menyebabkan resiko *overfitting* terhadap data yang langka. Dalam *undersampling*, teknik paling sederhana melibatkan penghapusan atau membuang data secara acak dari kelas mayoritas, yang dapat menyebabkan hilangnya informasi. Untuk mengatasi masalah ini, dapat dilakukan *oversampling* data dengan membuat data sintesis menggunakan teknik SMOTE (*Synthetic Minority Oversampling Technique*).

SMOTE (*Synthetic Minority Oversampling Technique*) adalah salah satu metode *oversampling* yang paling umum digunakan untuk menyelesaikan masalah ketidak seimbangan distribusi data pada pemodelan *machine learning*. SMOTE (*Synthetic Minority Oversampling Technique*) bertujuan untuk menyeimbangkan distribusi kelas dengan meningkatkan jumlah kelas minoritas secara acak dengan cara membuat data sintesis untuk tujuan *oversampling* [23]. SMOTE (*Synthetic Minority Oversampling Technique*) menghasilkan data pelatihan sintesis yang baru dengan menginterpolasi linier untuk kelas minoritas. Data pelatihan sintesis dihasilkan dengan memilih secara acak satu atau lebih dari *k-nearest neighbors* untuk setiap sampel pada kelas minoritas seperti diperlihatkan pada Gambar 3. Setelah proses *oversampling*, data direkonstruksi dan beberapa model klasifikasi dapat diterapkan untuk data yang sudah diproses.



Gambar 3. Representasi skema dari Algoritma SMOTE (*Synthetic Minority Oversampling Technique*) [24]

Cara kerja algoritma SMOTE (*Synthetic Minority Oversampling Technique*) diuraikan dalam langkah-langkah sebagai berikut:
Langkah 1 : Menetapkan kelas minoritas himpunan A untuk setiap $X \in A$, k-tetangga terdekat (*k-nearest neighbors*) dari X yang diperoleh dengan menghitung jarak *Euclidean* antara X dan setiap sampel lainnya dalam himpunan A.
Langkah 2 : Tingkat pengambilan sampel N diatur sesuai dengan proporsi yang tidak seimbang. Untuk setiap $X \in$, N sampel $(x1, x2, \ldots, xn)$ dipilih secara acak dari *k-nearest neighbors*, dan membangun himpunan A_1.
Langkah 3: Untuk setiap sampel $X\_k \in A\_1(k = 1, 2, 3, \ldots, N)$ - biasanya $k = 5$, persamaan berikut digunakan untuk menghasilkan sampel baru: $X' = X + rand(0, 1) * |X - X\_k|$, dimana rand (0, 1) mewakili angka acak antara 0 dan 1. Selanjutnya menarik garis antara tetangga dan menghasilkan titik acak pada garis.

2.4. *Random Forest Classifier*

Random Forest adalah algoritma untuk *supervised learning* yang bisa digunakan untuk klasifikasi maupun regresi. Algoritma ini paling fleksibel dan mudah digunakan. *Random Forest* (RF) terdiri dari beberapa *Decision Tree*. Semakin banyak *Decision Tree* yang dimiliki, semakin kuat algoritma *Random Forest* tersebut. Algoritma RF sudah banyak diaplikasikan pada berbagai bidang seperti memprediksi pergerakan terarah harga saham untuk perdagangan *intraday* [25], mengevaluasi efektivitas perangkat anti-burung [26], memprediksi penyebaran obligasi di bursa saham [27] sampai manajemen komentar media sosial [28].

Algoritma *Random Forest* menggunakan rata-rata untuk meningkatkan akurasi prediksi dan kontrol *overfitting*. Ukuran sub-sampel dikontrol dengan parameter *max_samples* jika *bootstrap=True* (*default*), jika tidak, seluruh *dataset* digunakan untuk membangun setiap pohon [29]. Saat menggunakan *Random Forest* untuk klasifikasi data, formula Indeks *Gini* seperti diperlihatkan pada persamaan (1) digunakan untuk memutuskan bagaimana node pada sebuah cabang pohon keputusan. Rumus ini menggunakan kelas dan probabilitas untuk menentukan *Gini* dari setiap cabang pada sebuah simpul, menentukan cabang mana yang lebih mungkin terjadi.

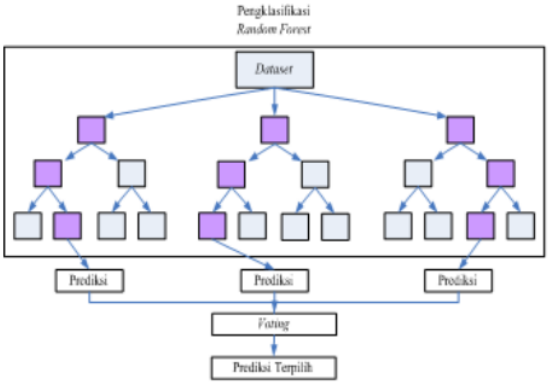
$$Gini = 1 - \Sigma_{i=1}^{c}(p_i)^2 \quad (1)$$

dimana p_i mewakili frekuensi relatif dari kelas yang diamati dalam kumpulan data dan c mewakili jumlah kelas. Selain *Gini*, *entropi* juga sering digunakan dalam menentukan bagaimana node bercabang di pohon keputusan. Formula untuk *entropi* terdapat pada persamaan (2).

$$Entropy = \Sigma_{i=1}^{c} - p_i * log_2(p_i) \quad (2)$$

Entropi menggunakan probabilitas hasil untuk membuat keputusan tentang bagaimana node harus bercabang. Berbeda dengan indeks *Gini*, indeks ini lebih intensif matematis karena fungsi logaritmik yang digunakan dalam menghitungnya.

Gambar 4 memperlihatkan cara kerja algoritma *Random Forest* dengan membuat sekumpulan pohon keputusan (*decision tree*) dari *subset* yang dipilih secara acak, mendapatkan prediksi dari setiap pohon keputusan, melakukan *voting* untuk setiap hasil yang diprediksi, dan memilih hasil prediksi terbaik berdasarkan *voting* terbanyak yang ditetapkan sebagai prediksi akhir.



Gambar 4. Pengklasifikasi *Random Forest*

2.5. Evaluasi Model

Memilih formula ukuran kinerja yang tepat untuk evaluasi algoritma adalah sebuah tahapan yang kritis, karena pengklasifikasi yang dilatih pada sekumpulan data yang tidak seimbang akan memberikan tingkat akurasi yang tinggi namun sebenarnya bias dikelas mayoritas. Ukuran kinerja yang tepat akan membantu dalam menilai kemampuan adaptasi algoritma secara efisien. Tujuan utamanya adalah untuk mendapatkan *True Positive* (TP) dan *True Negative* (TN) sebanyak mungkin dan selaras dengan mengurangi *False* Negatif sebanyak mungkin juga. Akurasi (*Accuracy*) mewakili kemampuan pengklasifikasi secara keseluruhan, namun ukuran akurasi dapat menyesatkan ketika data tidak seimbang karena lebih banyak bobot ditempatkan pada kelas mayoritas dibandingkan kelas minoritas sehingga sulit bagi *classifier* untuk berkinerja baik pada kelas minoritas. Ukuran kinerja lainnya adalah *Recall/Sensitivity* yaitu mengukur keakuratan kelas positif dan *Spesificity* untuk mengukur keakuratan kelas negatif. *Sensitivity* menilai efektivitas *classifier* pada kelas positif/mayoritas sedangkan *Spesificity* menilai efektivitas *classifier* pada kelas negatif/mayoritas. *Precision*, ukuran kinerja lainnya, adalah ukuran ketepatan model. Nilai presisi yang tinggi dari sebuah *classifier* merupakan indikasi *classifier* yang baik [30].

Selain ukuran kinerja diatas, terdapat ukuran kinerja kombinasi untuk menyeimbangkan antara tingkat *False Positive* (FP) dan *False Negative* (FN) diantaranya F1-*score*, *G-Mean*, dan *Youdens Index* yang dapat mengevaluasi kinerja dalam ketidak seimbangan data, karena jumlah sampel yang diprediksi dengan benar dari kelas positif dan atau negatif tersirat dalam parameter ini. F1-*score* mengukur keseimbangan antara presisi dan sensitivi8. Nilai F1-*score* yang tinggi menyiratkan akurasi yang lebih tinggi dikelas minoritas. Nilai F1-*score* 0 ketika presisi dan sensitifitasnya juga 0. *G-Mean* (*Geometric Mean*) fokus untuk mengukur keseimbangan antara kinerja klasifikasi pada kelas mayoritas dan minoritas. Nilai *G-Mean* yang rendah merupakan indikasi rendahnya kinerja dalam klasifikasi kasus positif bahkan jika kasus kelas negatif diklasifikasikan dengan benar, sebaliknya jika nilai *G-Mean* tinggi menunjukkan bahwa pengklasifikasi memiliki kinerja yang sama baiknya dengan sampel kelas minoritas dan mayoritas. Ukuran kinerja ini penting untuk menghindari *overfitting* kelas negatif dan *underfitting* kelas positif. Selanjutnya, *Youdens Index* mengevaluasi kemampuan *clasifier* untuk menghindari kesalahan klasifikasi. Indeks ini memberikan bobot yang sama pada kinerja pengklasifikasi baik pada kasus positif maupun negatif. Nilai indeks yang tinggi merupakan indikasi pengklasifikasi berkinerja dengan baik [31].

Penelitian ini akan menggunakan metrik evaluasi sebagaimana yang sudah digunakan oleh penelitian sebelumnya [32, 33] yaitu *Accuracy*, *Precision*, *Sensitivity*, F1-*score*, G-*Mean*, dan *Youdens Index* sebagaimana terdapat dalam persamaan (3), (4), (5), (6), (7), (8), (9).

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn} \quad (3)$$

$$Precision = \frac{tp}{tp + fp} \quad (4)$$

$$Recall/Sensitivity = \frac{tp}{tp + fn} \tag{5}$$

$$Specificity = \frac{tn}{tn + fp} \tag{6}$$

$$F1 - Score = \frac{2(recall.precision)}{recall + precision} \tag{7}$$

$$G - Mean = \sqrt{Sensitivity \times Specificity} \tag{8}$$

$$\text{Youden' s Index ()} = Sensitivity - (1 - Spesificity) \tag{9}$$

3. HASIL DAN ANALISIS

3.1. *Dataset*

Dataset penyakit jantung yang digunakan adalah *dataset* versi terakhir yang diakses pada UCI *machine learning repository*. Gambar 5 merupakan informasi umum *dataset* yang terdiri dari 299 sampel dan 13 kolom (variabel). Variasi *type* data adalah *float* sebanyak 3 variabel yaitu untuk variabel *age*, *platelets*, dan *serum_creatinine*, sedangkan sisanya sebanyak 10 variabel lainnya memiliki *type* data *integer* yaitu untuk variabel *anaemia*, *creatinine_phosphokinase*, diabetes, *ejection_fraction*, *high_blood_pressure*, *serum_sodium*, *sex*, *smoking*, *time*, dan *death_event*.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   age                       299 non-null    float64
 1   anaemia                   299 non-null    int64
 2   creatinine_phosphokinase  299 non-null    int64
 3   diabetes                  299 non-null    int64
 4   ejection_fraction         299 non-null    int64
 5   high_blood_pressure       299 non-null    int64
 6   platelets                 299 non-null    float64
 7   serum_creatinine          299 non-null    float64
 8   serum_sodium              299 non-null    int64
 9   sex                       299 non-null    int64
 10  smoking                   299 non-null    int64
 11  time                      299 non-null    int64
 12  DEATH_EVENT               299 non-null    int64
dtypes: float64(3), int64(10)
memory usage: 30.5 KB

None
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
0	75.0	0	582	0	20	1	265000.00	1.9	130	1	0	4	1
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1	0	6	1
2	65.0	0	146	0	20	0	162000.00	1.3	129	1	1	7	1
3	50.0	1	111	0	20	0	210000.00	1.9	137	1	0	7	1

Gambar 5. Informasi Umum *Dataset*

3.2. EDA (*Exploratory Data Analysis*)

Exploratory Data Analysis (EDA) merupakan langkah penting sebelum melakukan pemodelan data. Melalui proses EDA, dapat dipahami secara utuh data yang ada. Gambar 6 memperlihatkan *dataset* secara statistik menggunakan perintah *describe*() bahasa pemograman *Python*. Pada *dataset* ini *age* (umur) terendah yang terdapat dalam *dataset* adalah 40 tahun dan tertinggi adalah 95 tahun. *Anaemia* memiliki nilai dengan rentang 0 sampai 1 dengan rata-rata nilai 0.43, sampai variabel input terakhir yaitu *time* dengan rentang nilai 4 285 dengan rata-rata nilai 130.36. Untuk variabel target yaitu *Death_Event* memiliki rentang nilai 0-1, dimana 0 berarti pasien tidak meninggal selama masa tindak lanjut sedangkan 1 berarti sebaliknya.

```
df.describe()
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.00000	299.000000	299.000000	299.00000	299.000000	299.00000
mean	60.833893	0.431438	581.839465	0.418060	38.083612	0.351171	263358.029264	1.39388	136.625418	0.648829	0.32107	130.260870	0.32107
std	11.894809	0.496107	970.287881	0.494067	11.834841	0.478136	97804.236869	1.03451	4.412477	0.478136	0.46767	77.614208	0.46767
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.000000	25100.000000	0.50000	113.000000	0.000000	0.00000	4.000000	0.00000
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.000000	212500.000000	0.90000	134.000000	0.000000	0.00000	73.000000	0.00000
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.000000	262000.000000	1.10000	137.000000	1.000000	0.00000	115.000000	0.00000
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.000000	303500.000000	1.40000	140.000000	1.000000	1.00000	203.000000	1.00000
max	95.000000	1.000000	7861.000000	1.000000	80.000000	1.000000	850000.000000	9.40000	148.000000	1.000000	1.00000	285.000000	1.00000

Gambar 6. Statistik *Dataset*

Visualisasi merupakan langkah terpenting dalam EDA yang menunjukkan keterkaitan antar variabel independent dengan target. Matrik korelasi dalam bentuk heatmap diperlihatkan pada Gambar 7 yang menunjukkan data tabular hubungan antara pasangan variabel dalam data yang saling berkaitan. Matrik ini penting untuk memperlihatkan statistik deskriptif dari data multi variabel. *Heatmap* telah memberikan wawasan yang bagus tentang data yang dimiliki, dimana memperlihatkan korelasi positif dari variabel *death_event* dengan variabel *age* dan *serum creatinine*, variabel *smoking* dengan variabel *sex*, dan variabel *serum sodium* dengan variabel *ejection_fraction*. Sebaliknya terdapat korelasi negatif antara variabel *time* dengan variabel *death_event*, *high_blood_pressure* dan *age*, variabel *serum sodium* dengan variabel *death_event*, dan serum *creatinine*, variabel *ejection_fraction* dengan variabel *death_event*, dan variabel *creatine_phosphokinase* dengan variabel *anaemia*.

Gambar 7. Korelasi Matrik dalam Bentuk *Heatmap*

Dataset memiliki 6 (enam) variabel kategorikal (*categorical features*) yaitu *anaemia*, *diabetes*, *high_blood_pressure*, *sex*, *smoking* dan *death_event*. Untuk melihat bagaimana variabel kategorikal didistribusikan satu sama lain digunakan tabel kontigensi. Uji *chi-square* adalah teknik statistik untuk menguji hubungan antara dua variabel kategori. Uji *chi- square* seperti diperlihatkan pada Gambar 8 menunjukkan tidak ada variabel kategori yang memiliki hubungan dengan variabel target.

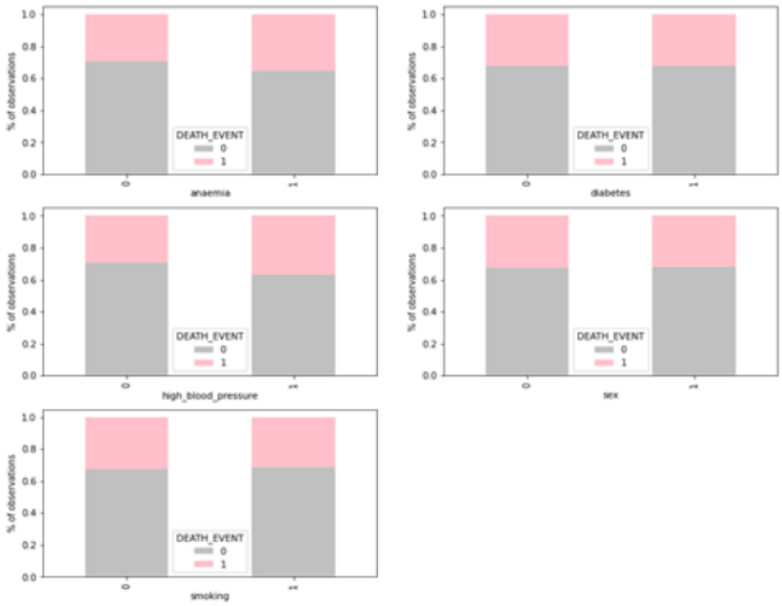
```
------------------------------------
Crosstab Antara ANAEMIA & DEATH_EVENT
------------------------------------
anaemia        0   1
DEATH_EVENT
0            120  83
1             50  46

H0: Tidak ada Hubungan antara DEATH_EVENT & ANAEMIA
H1: Terdapat Hubungan antara DEATH_EVENT & ANAEMIA

P-Value: 0.31
H0 Diterima

------------------------------------
Crosstab Antara DIABETES & DEATH_EVENT
------------------------------------
diabetes       0   1
DEATH_EVENT
0            118  85
1             56  40

H0: Tidak ada Hubungan antara DEATH_EVENT & DIABETES
H1: Terdapat Hubungan antara DEATH_EVENT & DIABETES

P-Value: 0.93
H0 Diterima
```

```
------------------------------------------------
Crosstab Antara HIGH_BLOOD_PRESSURE & DEATH_EVENT
------------------------------------------------
high_blood_pressure    0   1
DEATH_EVENT
0                    137  66
1                     57  39

H0: Tidak ada Hubungan antara DEATH_EVENT & HIGH_BLOOD_PRESSURE
H1: Terdapat Hubungan antara DEATH_EVENT & HIGH_BLOOD_PRESSURE

P-Value: 0.21
H0 Diterima

--------------------------------
Crosstab Antara SEX & DEATH_EVENT
--------------------------------
sex          0    1
DEATH_EVENT
0           71  132
1           34   62

H0: Tidak ada Hubungan antara DEATH_EVENT & SEX
H1: Terdapat Hubungan antara DEATH_EVENT & SEX

P-Value: 0.96
H0 Diterima
```

```
------------------------------------
Crosstab Antara SMOKING & DEATH_EVENT
------------------------------------
smoking        0   1
DEATH_EVENT
0            137  66
1             66  30

H0: Tidak ada Hubungan antara DEATH_EVENT & SMOKING
H1: Terdapat Hubungan antara DEATH_EVENT & SMOKING

P-Value: 0.93
H0 Diterima

----------------------------------------
Crosstab Antara DEATH_EVENT & DEATH_EVENT
----------------------------------------
DEATH_EVENT    0   1
DEATH_EVENT
0            203   0
1              0  96

H0: Tidak ada Hubungan antara DEATH_EVENT & DEATH_EVENT
H1: Terdapat Hubungan antara DEATH_EVENT & DEATH_EVENT

P-Value: 0.0
H0 Ditolak
```

Gambar 8. Uji *Chi-Square* Variabel Kategorikal

Selanjutnya *Barplot* dibawah (Gambar 9) memperlihatkan tidak ada hubungan yang kuat antara variabel kategorikal dengan variabel target seperti yang terlihat pada uji *Chi-Square* diatas.



Gambar 9. *Barplot* Hubungan Variabel Kategorikal dengan Target

3.3. *Feature Selection*

Menemukan variabel yang menentukan terhadap keberhasilan suatu model merupakan langkah krusial dalam *machine learning*. Hasil pengolahan data menunjukkan bahwa *Time*/Waktu merupakan variabel yang paling berpengaruh dalam kasus ini disusul oleh variabel *serum_creatin ejection_fraction*, *age*, *creatinine_phosphokinase*, *platelets*, dan *serum_sodium*. Semua fitur seperti yang diperlihatkan dari uji *chi-square* sebelumnya, fitur kategoris tidak begitu penting.

Setelah menentukan fitur penting, selanjutnya adalah mengidentifikasi ambang batas untuk fitur penting tersebut. Berdasarkan hasil pengolahan data diketahui ambang batas nilai untuk seleksi fitur adalah 7.22. Selanjutnya variabel/fitur yang memiliki nilai dibawah 7.22 akan dihapus dan tidak dipergunakan untuk proses selanjutnya. Karena data tidak seimbang maka penilaian terhadap kinerja model ini akan lebih cocok menggunakan F1*score* dibandingkan menggunakan akurasi.

3.4. Membangun Model Tanpa SMOTE (*Synthetic Minority Oversampling Technique*)

Untuk melihat dampak dari penggunaan teknik SMOTE (*Synthetic Minority Oversampling Technique*) pada penelitian ini, maka pertama membangun model tanpa teknik SMOTE (*Synthetic Minority Oversampling Technique*) untuk mendapatkan gambaran tentang kinerja model awal. Menemukan ambang batas dengan membangun model melalui penghapusan satu fitur yang paling tidak berpengaruh sesudah fitur *serum_sodium* yang telah ditentukan diatas merupakan langkah selanjutnya sebelum membangun model itu sendiri. Langkah ini akan menghasilkan satu set fitur dengan nilai F1-*score* terbaik seperti diperlihatkan pada Gambar 10 yang menunjukkan hasil pemilihan variabel tanpa SMOTE (*Synthetic Minority Oversampling Technique*) menggunakan bahasa pemograman *Python* untuk *Random Forest Classifier*. Pada gambar dapat dilihat bahwa model dengan 8 variabel teratas memiliki nilai F1-*score* terbaik yaitu sebesar 0.8435. Oleh karena itu ambang batas baru yang ditemukan adalah 1.375402 yaitu ambang batas variabel *anaemia*. Penelitian ini akan menyertakan semua fitur dengan kepentingan di atas ambang batas.

```
12 variables:  RandomForestClassifier(n_estimators=500, random_state=11)  F1 score: 0.8340649678089506
11 variables:  RandomForestClassifier(n_estimators=5000, random_state=11)  F1 score: 0.8263002529863771
10 variables:  RandomForestClassifier(n_estimators=500, random_state=11)  F1 score: 0.8306172323033565
9 variables:  RandomForestClassifier(random_state=11)  F1 score: 0.8410399170248493
8 variables:  RandomForestClassifier(n_estimators=1000, random_state=11)  F1 score: 0.8435036680809833
7 variables:  RandomForestClassifier(criterion='entropy', n_estimators=500, random_state=11)  F1 score: 0.8333208818231832
```

Gambar 10. Pemilihan Variabel dengan Nilai F1-*Score* Terbaik pada Model Tanpa SMOTE (*Synthetic Minority Oversampling Technique*)

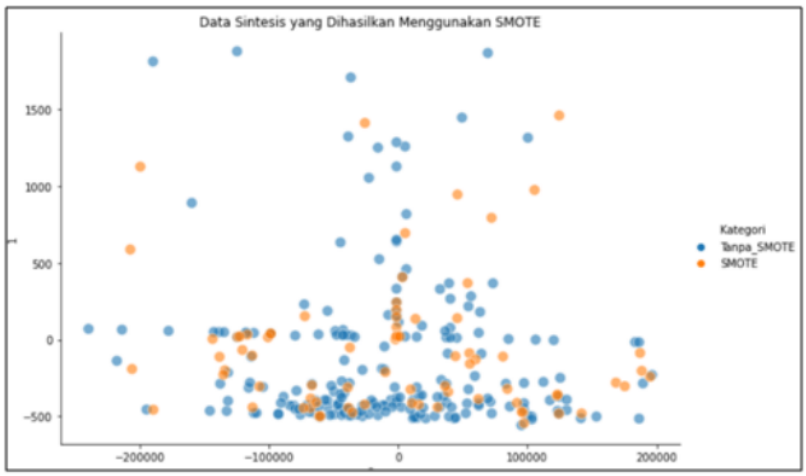
Selanjutnya membuat model RF Tanpa SMOTE (*Synthetic Minority Oversampling Technique*) dengan 8 variabel yang sudah terpilih. Kinerja model yang dibangun tanpa SMOTE (*Synthetic Minority Oversampling Technique*) memiliki skor akurasi 1 untuk data latih dan 0.87 untuk data uji seperti diperlihatkan pada Gambar 11. Berdasarkan skor akurasi tersebut model termasuk kategori *overfitting*. Tahap berikutnya dilakukan penyetelan *hyperparameter* menggunakan *max_depth* untuk mencegah *tree* dari *overfitting*.

```
ModelRF_Tanpa_SMOTE.score(X_train,y_train)

1.0

train_pred = ModelRF_Tanpa_SMOTE.predict(X_train)
print(classification_report(y_train,train_pred))

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       160
           1       1.00      1.00      1.00        79

    accuracy                           1.00       239
   macro avg       1.00      1.00      1.00       239
weighted avg       1.00      1.00      1.00       239

test_pred = ModelRF_Tanpa_SMOTE.predict(X_test)
print(classification_report(y_test,test_pred))

              precision    recall  f1-score   support

           0       0.93      0.88      0.90        43
           1       0.74      0.82      0.78        17

    accuracy                           0.87        60
   macro avg       0.83      0.85      0.84        60
weighted avg       0.87      0.87      0.87        60
```

Gambar 11. Hasil Skor Kinerja Model Random *Forest* Tanpa SMOTE (*Synthetic Minority Oversampling Technique*)

3.5. Membangun Model menggunakan SMOTE (*Synthetic Minority Oversampling Technique*)

SMOTE singkatan dari '*Synthetic Minority Oversampling Technique*' adalah teknik *oversampling* dari kategori minoritas. SMOTE (*Synthetic Minority Oversampling Technique*) menghasilkan sampel sintetis untuk membawa jumlah kategori dalam variabel target pelatihan ke jumlah yang sama. Jumlah data sesudah *resample* berjumlah 380 data yang terbagi menjadi 320 data latih dan 60 data uji. Gambar 12 memperlihatkan *plot* untuk menunjukkan sampel data sintetis yang dihasilkan menggunakan SMOTE (*Synthetic Minority Oversampling Technique*) dibandingkan dengan sampel data asli.



Gambar 12. Plot Data Sintesis Menggunakan SMOTE (*Synthetic Minority Oversampling Technique*)

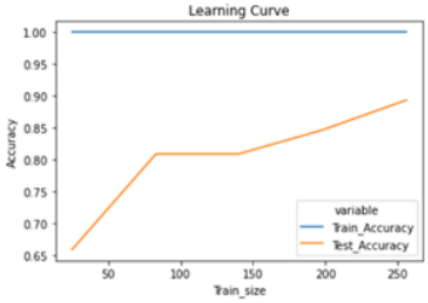
Gambar 13 memperlihatkan bahwa SMOTE (*Synthetic Minority Oversampling Technique*) telah mampu meningkatkan nilai F1-*score*. Model terbaik tetap menggunakan 8 variabel teratas dengan nilai F1-*score* sebesar 0.89. Tahap berikutnya membangun model menggunakan 8 variabel yang sudah dipilih menggunakan teknik SMOTE (*Synthetic Minority Oversampling Technique*). Skor untuk dataset latih dan skor CV model yang dilatih pada data dengan penerapan SMOTE (*Synthetic Minority Oversampling Technique*) lebih baik dibandingkan dengan model yang dilatih tanpa SMOTE (*Synthetic Minority Oversampling Technique*). Selanjutnya, penelitian ini akan menggunakan model yang dilatih pada data dengan penerapan SMOTE (*Synthetic Minority Oversampling Technique*) sebagai model akhir.

```
12 variables:  RandomForestClassifier(criterion='entropy', random_state=11)  F1 score: 0.8873161575080999
11 variables:  RandomForestClassifier(criterion='entropy', n_estimators=500, random_state=11)  F1 score: 0.8842412152469079
10 variables:  RandomForestClassifier(random_state=11)  F1 score: 0.8839385437774332
9 variables:  RandomForestClassifier(random_state=11)  F1 score: 0.887216678453329
8 variables:  RandomForestClassifier(criterion='entropy', n_estimators=500, random_state=11)  F1 score: 0.8934649174455945
7 variables:  RandomForestClassifier(criterion='entropy', random_state=11)  F1 score: 0.8840295095155696
```

Gambar 13. Peningkatan Nilai F1*score* Setelah Menggunakan SMOTE (*Synthetic Minority Oversampling Technique*)

Mengidentifikasi *overfitting* dengan *Learning Curve* menjadi tahapan selanjutnya. *Learning Curve* membantu mengidentifikasi bias dan varian serta membantu menginformasikan apakah menambahkan lebih banyak data pelatihan akan meningkatkan kinerja pada data yang tidak terlihat atau justru sebaliknya. Seperti diperlihatkan pada Gambar 14, model RF *overfit* ke *dataset* latih meskipun sudah mengurangi jumlah fitur. Hal ini disebabkan jumlah *dataset* yang sangat kecil. Seperti diperlihatkan pada *Learning Curve*, menambahkan jumlah data sintetis ke *dataset* latih akan meningkatkan kinerja model pada data yang tidak terlihat. Telah terjadi peningkatan skor CV dan skor data uji setelah menambahkan data sintetis menggunakan SMOTE (*Synthetic Minority Oversampling Technique*). Selanjutnya, penyetelan *hyperparameter GridSearchCV* digunakan untuk mencegah *classifier* dari *overfitting*.

Mengatasi *overfitting* bisa dilakukan dengan menambah jumlah data sintesis. Selain itu dalam penelitian ini juga diperkuat dengan mengimplementasikan *hyperparameter tuning GridSearchCV*. Parameter yang digunakan dalam penelitian ini adalah *n_estimator* dengan nilai [5000, 7000], *criterion*: [*gini, entropy*], *max_depth* dengan nilai [3,5,7], *min_samples_split*: [80, 100] dan *min_samples_leaf*: [40,50]. Setelah mengimplementasikan *hyperparameter tuning* pada data yang sudah *diresample* oleh teknik SMOTE (*Synthetic Minority Oversampling Technique*), model tidak *overfitting* lagi ke *dataset* pelatihan.



Gambar 14. *Learning Curve* Akurasi Data Latih dan Data Uji

3.6. Evaluasi Model

Gambar 15 (a) dan 15 (b) memperlihatkan bahwa skor data latih sudah mendekati skor Cross_Validation dan skor data uji, menunjukkan bahwa model sudah tidak *overfit*. Hal ini disebabkan karena telah dipilih sejumlah besar *n_estimator* untuk menumbuhkan lebih banyak pohon, sehingga mencegah *overfitting*, memilih angka yang rendah untuk *max_depth* dan angka yang besar untuk *max_samples_split* dan *max_samples_leaf* yang memastikan *daun/leaf* memiliki jumlah sampel yang baik dan memadai.

	precision	recall	f1-score	support
0	0.87	0.85	0.86	160
1	0.85	0.87	0.86	160
accuracy			0.86	320
macro avg	0.86	0.86	0.86	320
weighted avg	0.86	0.86	0.86	320

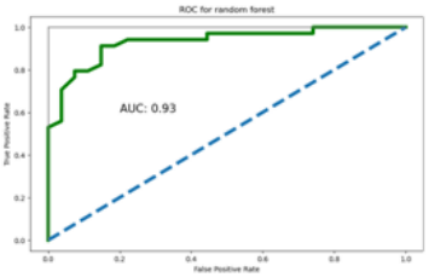
(a)

	precision	recall	f1-score	support
0	0.97	0.88	0.93	43
1	0.76	0.94	0.84	17
accuracy			0.90	60
macro avg	0.87	0.91	0.88	60
weighted avg	0.91	0.90	0.90	60

(b)

Gambar 15. (a) *Confusion Matrix* Data Uji, (b) *Confusion Matrix* Data Uji

Kurva ROC diperlihatkan pada Gambar 16 yang menunjukkan model RF SMOTE (*Synthetic Minority Oversampling Technique*) memiliki *True Positive Rate* (TPR) yang baik yang sangat penting untuk model yang digunakan untuk keperluan diagnosis medis.



Gambar 16. Kurva ROC untuk *Positive Rate*

3.7. Perbandingan evaluasi model tanpa SMOTE (*Synthetic Minority Oversampling Technique*) dan dengan SMOTE (*Synthetic Minority Oversampling Technique*)

Hasil perbandingan pengujian dua skenario terhadap kinerja model *Random Forest* diperlihatkan pada Tabel 1 yang menunjukkan bahwa kinerja model RF pada data latih tanpa menggunakan teknik SMOTE (*Synthetic Minority Oversampling Technique*) memberikan hasil skor 1 untuk semua kriteria penilaian. Namun berdasarkan hasil pengujian menggunakan *Learning Curve* terdeteksi bahwa model *overfitting*. Model bekerja sangat baik pada data latih namun akurasi menurun ketika diaplikasikan pada data uji sehingga terdapat perbedaan akurasi yang cukup signifikan antara data latih dan data uji.

Model *Random Forest* menggunakan SMOTE (*Synthetic Minority Oversampling Technique*) pada data latih memberikan hasil akurasi yang lebih rendah dibandingkan model awal, namun model lebih fit dan stabil. Nilai skor pengujian kinerja antara data latih dan data uji saling mendekati yang berarti model bekerja sangat baik pada latih maupun data uji. Selain itu, pada data uji model RF menggunakan SMOTE (*Synthetic Minority Oversampling Technique*), terjadi peningkatan pada semua indikator kinerja. Peningkatan skor akurasi 3.45%, Presisi 4.8%, *Sensitivity* 7.1%, F1*score* 4.8%, *Specificity* 2.1%, *G-Mean* 4.4%, dan *Youdens Index* 6.25%.

Nilai skor yang penting untuk bidang klinis dan untuk data tidak seimbang pada pengukuran kinerja kombinasi yang menyeimbangkan antara tingkat *False Positive* dengan *False Negative* menunjukkan nilai yang tinggi. Skor nilai untuk *Sensitivity* sebesar 0.91 yang menunjukkan bahwa sebanyak 91% diidentifikasikan menderita penyakit jantung dengan benar. *Spesivicity* memiliki nilai sebesar 0.97 yang berarti 97% diidentifikasi tidak menderita penyakit jantung dengan benar. Selanjutnya nilai F1-*score* sebesar 0.88, menunjukkan bahwa akurasi yang lebih tinggi diperoleh untuk kelas minoritas. G-Mean memiliki nilai skor tinggi sebesar 0.94, membuktikan bahwa model memiliki kinerja yang sama baiknya dengan sampel kelas minoritas dan mayoritas. Skor nilai *Youdens Index* sebesar 0.85 yang berarti memiliki indeks yang tinggi, dan menjadi indikasi pengklasifikasi berkinerja baik.

Berdasarkan hasil pengujian pada kedua model, terbukti bahwa teknik SMOTE (*Synthetic Minority Oversampling Technique*) mampu mengurangi *overfitting* pada model sekaligus dapat meningkatkan kinerja dari model yang dibangun. SMOTE (*Synthetic Minority Oversampling Technique*) tidak memerlukan duplikat data, namun menyeimbangkan distribusi kelas melalui penambahan data sintesis pada kelas minoritas dan membuat titik data sintetis berdasarkan titik data asli yang memberikan dampak terhadap peningkatan kinerja model secara keseluruhan. Model RF menggunakan SMOTE (*Synthetic Minority Oversampling Technique*) dan diperkuat dengan tambahan fungsi *hyperparameter tuning* akan menghasilkan model yang lebih ideal, fit dan stabil.

Tabel 2. Perbandingan Model RF-SMOTE (*Synthetic Minority Oversampling Technique*) dan Non-SMOTE (*Synthetic Minority Oversampling Technique*)

Kinerja	Model RF-Non SMOTE (*Synthetic Minority Oversampling Technique*)		Model RF-SMOTE (*Synthetic Minority Oversampling Technique*)	
	Latih	Uji	Latih	Uji
Accuracy	1	0.87	0.86	0.90
Precision	1	0.83	0.86	0.87
Sensitivity	1	0.85	0.86	0.91
F1score	1	0.84	0.86	0.88
Specificity	1	0.95	0.86	0.97
G-Mean	1	0.90	0.86	0.94
Youdens Index	1	0.80	0.86	0.85
Keterangan	Model Overfitting		Model FIT	

4. KESIMPULAN

Kinerja model *machine learning* akan bias apabila data tidak seimbang. Tingkat akurasi yang dihasilkan tinggi pada kelas mayoritas dan rendah pada kelas minoritas. Dalam penelitian ini, pengujian terhadap dampak penambahan data sintetis dengan teknik SMOTE (*Synthetic Minority Oversampling Technique*) telah dilakukan pada data latih terhadap dataset penyakit jantung. Hasil penelitian menunjukkan bahwa teknik SMOTE (*Synthetic Minority Oversampling Technique*) mampu mengatasi masalah *overfitting*, dengan menghasilkan model yang fit dan stabil. Ukuran kinerja *Random Forest classifier* juga menunjukkan peningkatan pada semua indikator penilaian, mulai peningkatan skor Akurasi sebesar 3.45%, Presisi 4.8%, *Sensitivity* 7.1%, F1-*score* 4.8%, *Specificity* 2.1%, *G-Mean* 4.4% dan *Youdens Index* 6.3%. Penelitian ini memperlihatkan bahwa dalam menentukan *classifier* menggunakan *machine learning* seperti *Random Forest*, kemiringan kelas perlu diseimbangkan terlebih dahulu sebelum membangun model untuk hasil kinerja yang lebih baik. Penelitian ini juga membuktikan bahwa *Random Forest* menjadi salah satu pengklasifikasi yang handal yang bisa digunakan untuk memprediksi penyakit khususnya dalam bidang klinis. Penelitian berikutnya menggunakan beberapa *classifier machine learning* untuk mengidentifikasi *classifier* terbaik, termasuk pengujian pada beberapa *dataset* klinis yang berbeda-beda. Selanjutnya melakukan ujicoba pada beberapa fungsi *hyperparameter tuning* untuk meningkatkan akurasi model *machine learning* terbaik.

REFERENSI

[1] T. P. Pushpavathi, S. Kumari, and N. K. Kubra, "Heart Failure Prediction by Feature Ranking Analysis in Machine Learning," *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, pp. 915–923, 2021.

[2] E. D. Adler, A. A. Voors, L. Klein, F. Macheret, O. O. Braun, M. A. Urey, W. Zhu, I. Sama, M. Tadel, C. Campagnari, B. Greenberg, and A. Yagil, "Improving Risk Prediction in Heart Failure Using Machine Learning," *European Journal of Heart Failure*, vol. 22, no. 1, pp. 139–147, 2020.

[3] K. V. V. Reddy, I. Elamvazuthi, A. A. Aziz, S. Paramasivam, H. N. Chua, and S. Pranavanand, "Heart Disease Risk Prediction Using Machine Learning Classifiers with Attribute Evaluators," *Applied Sciences*, vol. 11, no. 18, 2021.

[4] A. Özdemir, K. Polat, and A. Alhudhaif, "Classification of Imbalanced Hyperspectral Images Using SMOTE-Based Deep Learning Methods," *Expert Systems with Applications*, vol. 178, no. April, 2021.

[5] E. Prasetyo and B. Prasetiyo, "Increased Classification Accuracy C4 . 5 Algorithm Using Bagging Techniques in Diagnosing Heart Disease," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 5, pp. 1035–1040, 2020.

[6] A. Riani, Y. Susianto, and N. Rahman, "Implementasi Data Mining untuk Memprediksi Penyakit Jantung Mengunakan Metode Naive Bayes," *Journal of Innovation Information Technology and Application (JINITA)*, vol. 1, no. 01, pp. 25–34, 2019.

[7] D. S. Permana and A. Silvanie, "Prediksi Penyakit Jantung Menggunakan Support Vector Machine dan Python pada Basis Data Pasien," *Jurnal Nasional Informatia*, vol. 2, no. 1, pp. 29–34, 2021.

[8] M. M. Bukhari, B. F. Alkhamees, S. Hussain, A. Gumaei, A. Assiri, and S. S. Ullah, "An Improved Artificial Neural Network Model for Effective Diabetes Prediction," *Complexity*, vol. 2021, 2021.

[9] Erlin, Y. N. Marlim, Junadhi, L. Suryati, and N. Agustina, "Early Detection of Diabetes Using Machine Learning with Logistic Regression Algorithm," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 11, no. 2, 2022.

[10] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, and H. Tang, "Predicting Diabetes Mellitus with Machine Learning Techniques," *Frontiers in Genetics*, vol. 9, no. November, pp. 1–10, 2018.

[11] K. Polat, "A Hybrid Approach to Parkinson Disease Classification Using Speech Signal: The Combination of SMOTE and Random Forests," *2019 Scientific Meeting on Electrical-Electronics and Biomedical Engineering and Computer Science, EBBT 2019*, pp. 1–3, 2019.

[12] T. Pan, J. Zhao, W. Wu, and J. Yang, "Learning Imbalanced Datasets Based on SMOTE and Gaussian Distribution," *Information Sciences*, vol. 512, pp. 1214–1233, 2020.

[13] D. Elreedy and A. F. Atiya, "A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for Handling Class Imbalance," *Information Sciences*, vol. 505, pp. 32–64, 2019.

[14] J. Li, Q. Zhu, Q. Wu, and Z. Fan, "A Novel Oversampling Technique for Class-Imbalanced Learning Based on SMOTE and Natural Neighbors," *Information Sciences*, vol. 565, pp. 438–455, 2021.

[15] Z. Xu, D. Shen, T. Nie, Y. Kou, N. Yin, and X. Han, "A Cluster-Based Oversampling Algorithm Combining SMOTE and K-Means for Imbalanced Medical Data," *Information Sciences*, vol. 572, pp. 574–589, 2021.

[16] D. S. Sisodia and U. Verma, "The Impact of Data Re-Sampling on Learning Performance of Class Imbalanced Bankruptcy Prediction Models," *International Journal on Electrical Engineering and Informatics*, vol. 10, no. 3, pp. 433–446, 2018.

[17] S. Feng, J. Keung, X. Yu, Y. Xiao, and M. Zhang, "Investigation on The Stability of SMOTE-Based Oversampling Techniques in Software Defect Prediction," *Information and Software Technology*, vol. 139, no. June, p. 106662, 2021.

[18] N. K. Mishra and P. K. Singh, "Feature Construction and Smote-Based Imbalance Handling for Multi-Label Learning," *Information Sciences*, vol. 563, pp. 342–357, 2021.

[19] H. Hairani, A. S. Suweleh, and D. Susilowaty, "Penanganan Ketidak Seimbangan Kelas Menggunakan Pendekatan Level Data," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 1, pp. 109–116, 2020.

[20] T. Ahmad, A. Munir, S. H. Bhatti, M. Aftab, and M. A. Raza, "Survival Analysis of Heart Failure Patients : A Case Study," *PLOS ONE*, vol. 12, no. 7, pp. 1–8, 2017.

[21] D. Chicco and G. Jurman, "Machine Learning Can Predict Survival of Patients with Heart Failure from Serum Creatinine and Ejection Fraction Alone," *BMCMedical Informatics and DecisionMaking*, vol. 5, pp. 1–16, 2020.

[22] S. Wang, S. Liu, J. Zhang, X. Che, Y. Yuan, Z. Wang, and D. Kong, "A New method of Diesel Fuel Brands Identification: SMOTE Oversampling Combined with XGBoost Ensemble Learning," *Fuel*, vol. 282, no. July, p. 118848, 2020.

[23] EngEd Community, "Introduction to Random Forest in Machine Learning," *Section's Engineering Education Program*, 2020.

[24] F. Hu and H. Li, "A Novel Boundary Oversampling Algorithm Based on Neighborhood Rough Set Model : NRSBoundary-SMOTE," *Mathematical Problems in Engineering*, 2013.

[25] P. Ghosh, A. Neufeld, and J. K. Sahoo, "Forecasting Directional Movements of Stock Prices for Intraday Trading Using LSTM and Random Forests," *Finance Research Letters*, no. November 2015, p. 102280, 2021.

[26] Q. Zhou, W. Lan, Y. Zhou, and G. Mo, "Effectiveness Evaluation of Anti-bird Devices based on Random Forest Algorithm," *2020 7th International Conference on Information, Cybernetics, and Computational Social Systems, ICCSS 2020*, pp. 743–748, 2020.

[27] Z. Chai and C. Zhao, "Multiclass Oblique Random Forests with Dual-Incremental Learning Capacity," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5192–5203, 2020.

[28] N. Soonthornphisaj, T. Sira-Aksorn, and P. Suksankawanich, "Social Media Comment Management Using SMOTE and Random Forest Algorithms," *International Journal of Networked and Distributed Computing*, vol. 6, no. 4, pp. 204–209, 2018.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, and B. Thirion, "Scikit-Learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[30] V. P. K. Turlapati and M. R. Prusty, "Outlier-SMOTE: A Refined Oversampling Technique for Improved Detection of COVID-19," *Intelligence-Based Medicine*, vol. 3-4, no. July, p. 100023, 2020.

[31] J. Akosa, "Predictive Accuracy : A Misleading Performance Measure for Highly Imbalanced Data Classified Negative," *Oklahoma State University*, pp. 1–12, 2017.

[32] M. Aria, C. Cuccurullo, and A. Gnasso, "A Comparison Among Interpretative Proposals for Random Forests," *Machine Learning with Applications*, vol. 6, no. January, p. 100094, 2021.

[33] S. Fotouhi, S. Asadi, and M. W. Kattan, "A Comprehensive Data Level Analysis for Cancer Diagnosis on Imbalanced Data," *Journal of Biomedical Informatics*, vol. 90, no. January, p. 103089, 2019.

Dampak Smpte Terhadap Kinerja Random Forest Classifier Berdasarkan Data Tidak Seimbang

ORIGINALITY REPORT

12%	12%	11%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	journal.universitasbumigora.ac.id Internet Source	8%
2	Submitted to Federal University of Technology Student Paper	1%
3	Submitted to Universiti Teknologi Petronas Student Paper	1%
4	www.peertechz.com Internet Source	1%
5	www.geeksforgeeks.org Internet Source	<1%
6	www.researchgate.net Internet Source	<1%
7	Submitted to Liverpool John Moores University Student Paper	<1%
8	Asma Cherif, Arwa Badhib, Heyfa Ammar, Suhair Alshehri, Manal Kalkatawi, Abdessamad Imine. "Credit card fraud	<1%

detection in the era of disruptive technologies: A systematic review", Journal of King Saud University - Computer and Information Sciences, 2022
Publication

9	Submitted to De Montfort University Student Paper	<1 %
10	medium.com Internet Source	<1 %
11	www.mdpi.com Internet Source	<1 %
12	Submitted to University of Strathclyde Student Paper	<1 %
13	Submitted to University of Sydney Student Paper	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	Off		