Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722

Analisis Kinerja Algoritma Decision Tree Dan Random Forest Dalam Klasifikasi Penyakit Kardiovaskular

Nisa Utami*, Kiki Ahmad Baihaqi, Elsa Elvira Awal, Deden Wahiddin

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Buana Perjuangan, Karawang, Indonesia
Email: if20.nisautami@mhs.ubpkarawang.ac.id, kikiahmad@ubpkarawang.ac.id, elsaelvira@ubpkarawang.ac.id. deden.wahiddin@ubpkarawang.ac.id
Email Penulis Korespondensi: [1]if20.nisautami@mhs.ubpkarawang.ac.id
Submitted: **01/08/2024**; Accepted: **10/09/2024**; Published**: 11/09/2024**

Abstrak−Kardiovaskular menjadi penyakit dengan jumlah kematian yang cukup tinggi di Indonesia, istilah kardiovaskular lebih populer dengan penyakit jantung yaitu suatu kondisi yang mana dapat menyebabkan penyumbatan dan penyempitan pembuluh darah. Pada penyakit kardiovaskular ini mempunyai dua resiko, yang pertama resiko dapat diubah seperti stres, peningkatan tekanan darah, diet tidak sehat, peningkatan kadar glukosa, kolesterol yang tidak normal dan kurangnya aktifitas fisik. Sedangkan resiko yang tidak dapat diubah meliputi penyakit keluarga, jenis kelamin, usia, dan obesitas. Pada penelitian ini dapat mengkaji dan menganalisis dari dua kinerja algoritma terbaik klasifikasi dengan algoritma *decision tree* dan algoritma *random forest*, dalam pengklasifikasian penyakit kardiovaskular berdasarkan penyebab terjadinya penyakit tersebut. Maka dari itu proses penting dalam penelitian ini salah satunya yaitu mendiagnosis suatu penyakit agar data yang didapat lebih akurat atau tidak terjadi kesalahan dalam diagnosis. Algoritma yang diterapkan dalam penelitian adalah *Decision Tree* dan *Random Forest* dikarnakan pada penelitian sebelumnya dengan menggunakan algoritma *Random Forest* menjadi algoritma terbaik. Alasan algoritma *Decision Tree* digunakan pada penelitian ini dikarenakan algoritma yang serumpun dengan algoritma *Random Forest*. Adapun kinerja dari masing-masing algoritma dievaluasi menggunakan *Area Under the Curve (AUC), classification report*, *k-Fold Cross Validation* dan *Confusion matrix*. Dataset yang digunakan diambil dari website Kaggle dengan data yang digunakan yaitu data *Cardiovascular Disease* yang terdiri dari 68.205 baris (data pasien) dan 17 atribut. Berdasarkan hasil evaluasi dengan menggunakan nilai *Area Under The Curve* (AUC), didapatkan hasil tertinggi sebesar 0.761 oleh algoritma *Random Forest* dengan kondisi data seimbang dengan *Random oversampling*. Sedangkan nilai AUC terendah didapatkan oleh algoritma *Decision Tree* dengan data yang tidak seimbang sebesar 0.592. Berdasarkan hasil ini diketahui bahwa algoritma *Random Forest* dengan skema data yang seimbang menjadi algoritma yang lebih baik, dengan skenario data yang seimbang menggunakan teknik SMOTE dan *Random Oversampling*.

Kata Kunci: *Decision Tree*, Kardiovaskular, Klasifikasi, Penyakit, *Random Forest.*

Abstract−Cardiovascular disease is a disease with a fairly high number of deaths in Indonesia. The term cardiovascular is more popular with heart disease, which is a condition which can cause blockage and narrowing of blood vessels. Cardiovascular disease has two risks, the first is a risk that can be changed, such as stress, increased blood pressure, unhealthy diet, increased glucose levels, abnormal cholesterol and lack of physical activity. Meanwhile, risks that cannot be changed include family disease, gender, age and obesity. In this research, we can examine and analyze the performance of the two best classification algorithms, namely the decision tree algorithm and the random forest algorithm, in classifying cardiovascular disease based on the cause of the disease. Therefore, one of the important processes in this research is diagnosing a disease so that the data obtained is more accurate or errors in diagnosis do not occur. The algorithms applied in the research are Decision Tree and Random Forest because in previous research the Random Forest algorithm was the best algorithm. The reason the Decision Tree algorithm is used in this research is because the algorithm is related to the Random Forest algorithm. The performance of each algorithm is evaluated using Area Under the Curve (AUC), classification report, k-Fold Cross Validation and Confusion matrix. The dataset used was taken from the Kaggle website with the data used being Cardiovascular Disease data which consists of 68,205 rows (patient data) and 17 attributes. Based on the evaluation results using the Area Under The Curve (AUC) value, the highest result was obtained at 0.761 by the Random Forest algorithm with balanced data conditions with Random oversampling. Meanwhile, the lowest AUC value was obtained by the Decision Tree algorithm with unbalanced data of 0.592. Based on these results, it is known that the Random Forest algorithm with a balanced data scheme is a better algorithm, with a balanced data scenario using SMOTE and Random Oversampling techniques.

Keywords: Cardiovascular, Classification, Decision Tree, Disease, Random Forest

1. PENDAHULUAN

Di Indonesia, istilah kardiovaskular lebih populer dengan penyakit jantung, salah satu perihal yang dapat menyebabkan terjadinya penyempitan dan penyumbatan pembuluh darah[1]. Hal ini dapat memicu serangan jantung, sakit dada atau angina, serangan stroke, gagal ginjal dan hipertensi [2]. Jumlah penyakit kardiovaskular yang semakin meningkat dengan jumlah kematian yang tinggi menjadi penyebab beban yang cukup berat untuk sistem perawatan kesehatan [3]. Maka dari itu proses penting dalam penelitian ini salah satunya yaitu mendiagnosis suatu penyakit agar data yang didapat lebih akurat atau tidak terjadi kesalahan diagnosis, untuk meningkatkan akurasi pada proses diagnosis dapat dilakukan dengan tahapan data *mining* [4]. Dua faktor terjadinya penyakit kardiovaskular yang pertama yaitu penyakit keluarga, jenis kelamin, usia, dan obesitas termasuk kedalam faktor resiko yang tidak dapat diubah. Sedangkan stres, peningkatan tekanan darah, diet tidak sehat, peningkatan kadar glukosa, kolesterol yang tidak normal dan kurangnya aktifitas fisik termasuk faktor resiko yang dapat diubah [5].

Pada penelitian terkait menggunakan klasifikasi` penyakit jantung dengan perbandingan 5 algoritma *Nave Baye*, *K-Nearest Neighbor* (KNN), *Random Forest*, *Decision Tree* dan *Support Vector Maechine* (SVM). Pada penelitian ini *preprocessing* yang memanfaatkan suatu aplikasi rapidminer 9.0.1. lalu memberikan pelabelan dalam

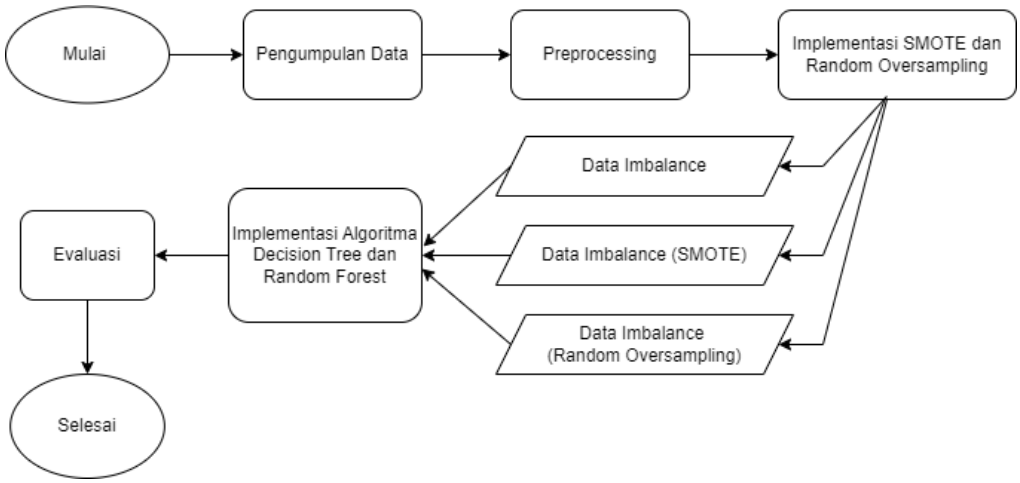
Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722

proses pengujian dengan menggunakan algoritma sebanyak 5 algoritma. Penelitian ini juga menggunakan *X-Fold Validation* untuk membagi data sebagai data berulang. Setelah dilakukan penelitan, ditemukan yakni algoritma *Random Fores*t merupakan algoritma terbaik dengan hasil nilai 82.93% dan AUC 0,896 [6]. Pada penelitia [7] dengan menggunakan data penyakit jantung dan metode yang digunakan pengklasifikasian algoritma *Naïve Bayes*, *Random forest Classifier* dan *Decision tree*. Proses klasifikasi model diawali dengan *training* data, *cross validation* untuk menegetahui data tersebut mengalami *overfitting* atau tidak, dan melakukan pengujian data *dummy*. Hasil dari pengujian rata-rata, *decision tree* mendapatkan nilai dengan pencarian secara acak 0,844 dan pencarian dengan jaringan 0,84. Sedangkan *navie bayes* tidak mempunyai perbedaan antara pencarian acak dengan pencarian secara jaringan yaitu 0,85. Namun pada algiritma *random forest classifier* menggunakan pencarian acak 0,852 dan pencarian jaringan 0,868. Penelitian selanjutnya memprediksi penyakit jantung dengan metode yang digunakan yaitu algoritma C4.5, *Navie Bayes* dan *Random Forest*. Penelitian ini pengujian yang di lakukan menggunakan *accuracy* untuk menentukan nilai akhir atau nilai dari algoritma terbaik. Hasil dari penelitian ini algoritma terbaik yaitu algoritma random forest dibanding dengan algoritma C4.5 dan *Navie Bayes*. Hasil pengujian nilai tertinggi pada penelitian ini yaitu algoritma *Random Forest* yang memiliki akurasi 98,60%, *recal* 99,44% *precision* 98,60% [8].

Mengacu pada penelitian terkait yang telah di lakukan, perbedaan antara penelitian ini dengan penelitian sebelumnya yaitu akan melakukan perbandingan algoritma *Decision Tree* dan *Random Forest.* Algoritma yang diterapkan dalam penelitian adalah *Decision Tree* dan *Random Forest* dikarnakan pada penelitian sebelumnya dengan menggunakan algoritma *Random Forest* menjadi algoritma terbaik. Alasan algoritma *Decision Tree* digunakan pada penelitian ini dikarenakan algoritma yang serumpun dengan algoritma *Random Forest*. Selain itu pada prosesnya juga akan menambahkan metode *Sinthetic Minority Over-sampling technique* (SMOTE) dan *Random Oversampling* untuk menyelesaikan data yang tidak seimbang dalam pembuatan model Klasifikasi penyakit yang menyerang sistem kardiovaskular. Penggunaan metode SMOTE dan Random Oversampling diterapkan pada penelitian ini dikarenakan data yang digunakan memiliki ketidakseimbangan data tidak terlalu besar, tetapi diperlukan untuk meningkatkan akurasi model. Adapun hasil yang akan diuji nantinya berupa data tidak seimbang dengan data yang seimbang dengan menggunakan metode SMOTE dan *Random Oversampling*. Adapun hasil evaluasi yang akan digunakan dalam penelitian ini meliputi *Confusion Matrix, Classification Report, Cross Validation*, dan *ROC* serta *AUC*. Dalam konteks pengolahan data yang digunakan menunjukkan hasil 1 atau 0, hasil tersebut menjelaskan bahwa 1 adalah terkena penyakit kardiovaskular dan 0 tidak terkena penyakit kardiovaskular.

2. METODOLOGI PENELITIAN

Bagian metode penelitian menjelaskan alur penelitian, dimana cara akumulasi data dan algoritma yang diterapkan untuk membangun suatu model. Berikut ini adalah tahapan metode penelitian analisis perbandingan algoritma klasifikasi penyakit kardiovaskular terdapat di Gambar 1.



Gambar 1. Alur Penelitian

Pada Gambar 1 terdapat alur penelitian atau proses yang dilakukan pada penelitian, diawali dengan proses pengumpulan data, *preprocessing*, dialanjut dengan implementasi SMOTE dan *Random Oversampling*. Setelah tahapan implementasi SMOTE dan *Random Oversmpling*, menghasilkan 3 data yang berbeda yaitu data *original* atau data *imbalance*, data *balance* (SMOTE) dan data *balance* (*Random Oversampling*). Selanjutnya akan dilakukan penerapan algoritma dengan menggunakan algoritma *Decision Tree* dan *Random Forest*. Tahapan terakhir yang akan dilakukan yaitu evaluasi model dengan menggunakan *Confusion matrix, classification report*, *k-Fold Cross Validation* dan *Area Under the Curve (AUC)*.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722

2.1. Pengumpulan Data

Pada tahap ini akan dilakukan pengumpulan data dengan mengkaji beberapa sumber ilmiah yang berkaitan dengan penelitian, serta melakukan pencarian data di website seperti Kaggle, UCI Machine Learning, serta beberapa website lain yang menyediakan data Terkait penyakit kardiovaskular.

2.2. *Preprocessing*

Pada tahapan ini dapat dilakukan pengolahan data dengan tahapan *Data cleaning* (Pembersihan data) untuk mengidentifikasi atribut yang memiliki *duplicate data* dan *missing value*. Adapun bila data yang diolah memiliki *duplicate data* dan *missing value*, akan dilakukan pembersihan data dengan cara menghapus nilai duplikat dan *missing value* dengan menggunakan *function "drop_duplicates"* dan *"dropna"* pada *python*. Hal ini dilakukan untuk memastikan data yang akan di proses sudah bersih dari *duplicate* data dan *missing value* untuk menjaga keakuratan dan kualitas dari model yang dibuat. Selanjutnya juga dilakukan transformasi data, di mana pada tahap transformasi data dilakukan untuk mengubah beberapa variabel *numeric* menjadi *categorical*, hal ini dilakukan untuk membuat keseluruhan data menjadi seragam.

2.3. Implementasi SMOTE dan *Random Oversampling*

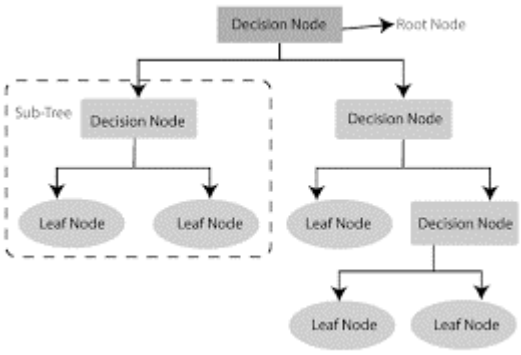
Pada penelitian ini setelah tahapan preprocesing dilakukan implementasi SMOTE dan *Random Oversampling*, yang mana metode ini dilakukan untuk melihat keseimbangan data pada kelas target (cardio). Berikut tinjauan teori metode SMOTE dan *Random Oversampling*.

a. SMOTE
   Dalam mengatasi masalah ketidakseimbangan data, Teknik metode *Synthetic Minority Over Sampling Technique* (SMOTE) digunakan untuk meningkatkan jumlah sampel pada kelas minoritas dengan cara menghasilkan sampel sintesis [9]
b. *Randem Oversampling*
   Random Oversamling menjadi suatu peroses pengambilan sampel ulang dengan melakukan memilih secara acak kelas minoritas pada dataset, lalu data terpilih akan diduplikasi atau ditambahkan pada set pelatihan yang baru[10].

2.4. Implementasi Algoritma

Pada tahapan implementasi algoritma menggunakan metode klasifikasi dengan algoritma *Decision Tree* dan *Random Forest*. Pada penelitian ini menggunakan perbandingan algoritma dengan hasil algoritma terbaik, berikut cara kerja algoritma diantaranya.

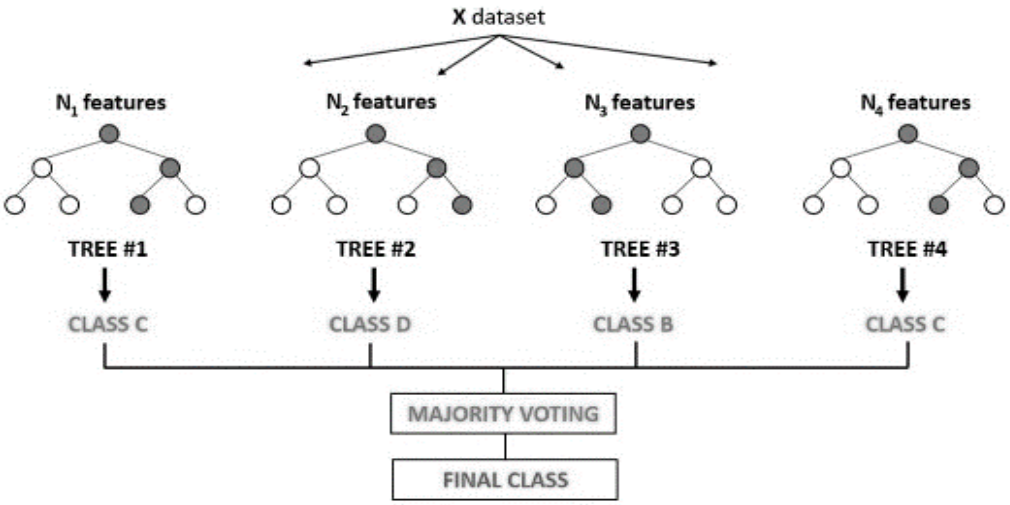
a. *Decision Tree*
   Salah satu algoritma klasifikasi yang berbentuk struktur pohon yaitu algotitma Decision Tree. Pada algoritma ini, proses dimulai dengan membandingkan nilai atribut root terhadap atribut record. Hasil dari perbandingan ini digunakan untuk memilih cabang yang tepat dan meneruskan navigasi ke simpul berikutnya. Pohon keputusan, yang terdiri dari node-node internal, dapat melakukan tes pada variabel individu atau membagi dataset menjadi subset yang lebih kecil berdasarkan atribut tertentu. [11]. Segmen yang dihasilkan Sekumpulan node dapat menentukan kelas untuk setiap pengamatan. Dalam penelitian konsep yang digunakan yaitu entropy informasi [12]. Adapun Gambar 2 merupakan contoh ilustrasi dari hasil pohon keputusan.



Gambar 2 Ilustrasi Pohon Keputusan

b. *Random Forest*
   Random forest bagian dari algoritma klasifikasi dengan pengklasifikasian data yang berjumlah besar[13]. Pada pemrosesan klasifikasi algoritma *random forest* diawali dengan pemecahan data berikutnya sampel dimasukan kedalam pohon keputusan secara acak. Setelah terbentuk pohon keputusan, selanjutnya pencocokan tiap-tiap kelas dengan data sampel dan melakukan *voting*, lalu vot terbanyak yang akan diambil. Berikut dapat dilihat pada Gambar 2 diagram alur model *random forest* [14]

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722



Gambar 3 Diagram alur Random Forest

2.5. Evaluasi Model

Dalam mengukur suatu kinerja pada model, langkah terakhir akan dilakukannya pengukuran nilai *accuracy, recall, f1-score, cross validation,* dan ROC serta AUC. Adapun pengujian akan dilakukan menggunakan library dari *sklearn metrics* seperti *Confusion matrix, classification report, k-Fold Cross Validation* dan *Area Under the Curve (AUC)*. Adapun tinjauan teori dari setiap metode yang digunakan dalam proses evaluasi seperti berikut.

a. Confusion matrix adalah suatu table dengan kolom sebagai prediksi dan baris sebagai klasifikasi benar yang dapat digunakan untuk menghasilkan kinerja dalam masalah klasifikasi[15]. Adapun pada metode ini berfungsi untuk menggambarkan keakuratan klasifikasi nilai yang sebenarnya dibandingkan dengan nilai prediksi *[16]*. Berikut tabel Confusion matrix pada Tabel 2.

Tabel 1. *Confusion Matrix*

		Prediksi	
		Positif	*Negatif*
Aktual	*Positif*	*TP*	*FP*
	Negatif	*FN*	*TN*

Keterangan:
TP (*True Positive*): data yang positif namun diprediksi benar
FP (*False Positive*): data yang negatif namun diprediksi positif atau benar
FN (*False Negative*): data yang positif namun diprediksi salah atau negatif
TN (*True Negative*): data yang negatif namun diprediksi negatif

b. *Classification Report* menyediakan informasi sebuah rincian laporan tentang kinerja model klasifikasi pada data uji. Nilai *precision, recall, f1-score,* dan *accuracy* dapat diperlihatkan dengan report yang bermanfaat untuk suatu evaluasi performa klasifikasi model [17]. Berikut rumus perhitungan *precision, recall, f1-score,* dan *accuracy* dapat dilihat pada persamaan (2), (3), (4) dan (5) :

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$recall = \frac{TP}{TP+FP} \quad (3)$$

$$f1score = 2 \times \frac{recall \times precision}{recall+precision} \quad (4)$$

$$acuracy \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

c. *K-fold Cross Validation* optimasi parameter yang dapat menentukan nilai k terbaik, untuk mengetahui nilai rata-rata dapat menggunakan metode ini [18]. Denagan cara mengacak atribut dan melakukan perulangan dapat menguji eror pada dataset sehingga sistem dapat teruji dari beberapa inputan atribut yang acak [19].

d. *Area Under the Curve* (AUC) ini memiliki kinerja yang baik dapat mengukur dan mendapatkan nilai dari kinerja klasifikasi secara umum terhadap ketidakseimbangan kelas [20]

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722

3. HASIL DAN PEMBAHASAN

Akumulasi data yang dilakukan menjadi tahap awal penelitian, data yang digunakan dalam penelitian ini dikumpulkan dari *website Kaggle* pada tahun 2023. Data yang digunakan yaitu data *Cardiovascular Disease* yang mencakup 68.205 baris (data pasien) dan 17 atribut. Berikut variabel-variabel dalam dataset penelitian ini dapat dilihat pada Tabel 2.

Tabel 2. Rincian Data

Atribut	Keterangan	Tipe Data
Id	Identifikasi	Int (Integer)
Age	Umur perhari	Int (Integer)
gender	Jenis kelamin	Categorical
height	Tinggi tubuh	Int (Integer)
weight	Masa tubuh	Float
ap-hi	Tekanan sistolik	Int (Integer)
ap-lo	Tekanan diastolik	Int (Integer)
cholesterol	Riwayat kolesterol	Categorical
gluc	Riwayat glukosa	Categorical
smoke	Status peroko	Int (Integer)
alco	Asupan alkohol	Int (Integer)
Active	Aktifitas fisik	Int (Integer)
cardio	Target variable	Int (Integer)
age-years	Umur pertahun	Int (Integer)
bmi	Gabungan variable berat badan dan tinggi badan	Float
bp_category	Ap-hi dan ap-lo dengan kategori	Object
bp_category_ encoded	Encoded dari bp_category	Object

Data yang telah didapat terlebih dahulu akan langsung dilakukan *preprocessing*, di mana pada tahap ini data akan disiapkan sebelum masuk dalam tahapan selanjutnya.

3.1 Hasil *Preprosessing*

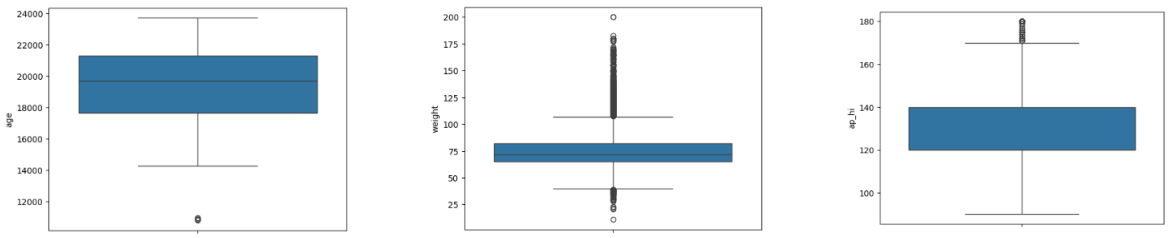
Tahap ini diawali dengan pengecekan *duplicated data*, *missing value* dan *Outliers* jika memiliki *duplicated data*, *missing value* dan *Outliers* akan dilakukan pembersihan data dengan cara menghapus nilai duplikat data, *missing value* dan *Outliers*. Cara untuk menghapus *duplicated data* dan *missing value* dapat menggunakan *function "drop_duplicates"* dan *"dropna"* pada *python*. Adapun hasil dari pengecekan data dapat dilihat pada Tabel 3.

Tabel 3. Proses Pengecekan Data

Dataset	
Missing Value	0
Duplicated Data	0

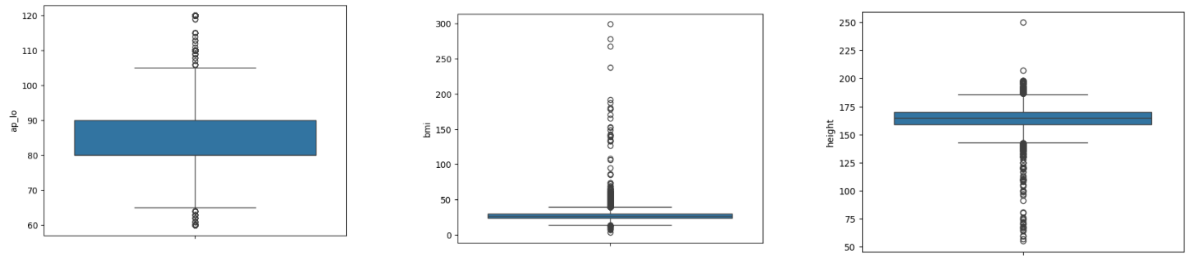
Pada Tabel 3 menjelaskan bahwa pengecekan *duplicated data* dan *missing value*, dimana data yang digunakan setelah di cek diketahui 0 atau tidak memiliki *duplicated data* dan *missing value*, maka tidak dilakukannya pembersihan data.

Adapun dengan tahapan pengecekan *outliers*, jika memiliki *outliers* pada penelitian dapat melakukan tahap penghapusan *outliers* dengan cara menggunakan *boxplot* dan perhitungan *Z-score* unruk menentukan suatu titik dari niali mean. Adapun hasil dari pengecekan *outliers* dapat dilihat pada Gambar 4.



(a) (b) (c)

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722



(d) (e) (f)

Gambar 4. Outliers Pada Variabel : (a). Age, (b).Weight, (c).Ap-Hi, (d). Ap-Lo, (e). Bmi dan (f). Height

Pada Gambar 4 menghasilkan hasil dari pengecekan *outliersi* yang terdapat pada variabel *age*, *height*, *weight*, *ap_hi*, *ap_lo* dan *bmi*, dengan memiliki *outliers* yang beragam. Setelah mengetahui dari variabel tersebut memiliki *outliers* selanjutnya mebuat variabel *Z-score* untuk menghitung nilai Z dengan cara menghitun seluruh jumlah data dikurangi dengan nilai rata-rata dari setiap variabel serta membagi nilai dari setiap data dengan standard deviation . Adapun hasil dari variabel *Z-score* dapat dilihat pada tabel 4.

Tabel 4. Sampel Hasil Penerapan Perhitungan Z-Score

Zs_age	Zs_height	Zs_weight	Zs_ap_hi	Zs_ap_lo	Zs_bmi
-0.433348	0.443591	-0.846862	-1.029648	-0.138225	-0.919837
0.310054	-1.023983	0.762784	0.849852	0.955390	1.230759
-0.245370	0.076698	-0.706892	0.223352	-1.231840	-0.664185
-0.745293	0.565889	0.552830	1.476353	2.049005	0.199115
-0.805656	-1.023983	-1.266769	-1.656149	-2.325455	-0.746592

Pada Tabel 4 menjelaskan hasil dari perhitungan menggunakan metode *Z-score*, beberapa variabel yang tedapat pada Tabel 4 akan di identifikasikan satu persatu untuk melihat apakah *outliers* berada pada limit atas dan limit bawah. Nilai yang dianggap sebagai *outliers* merupakan nilai yang melebihi batas atas dan batas bawah.

Setelah diketahui posisi *outliers* berdasarkan nilai *Z-score* selanjutnya dibuat variabel baru dengan nama ‘Cardio’ untuk menyimpan data yang sudah dibersihkan atau sudah tidak memiliki *outliers*. Pada proses ini nilai yang ditetapkan untuk memilih data yang tidak memiliki *outliers* sebesar < 2 serta >-2. Selanjutnya dilakukan pengecekan kembali *outliers* dengan *boxplot* untuk memastikan variabel yang memiliki *outliers* sudah tidak memiliki *outliers*. Adapun Tabel 5 *outliers* menjabarkan hasil temuan serta jumlah data sebelum dan sesudah pembersihan *outliers*.

Tabel 5. Hasil Penghapusan Outliers

Penghapusan *Outliers*	
Jumlah data sebelum dihapus *outliers*	**Jumlah data sesudah dihapus *outliers***
68.205	40.794
Jumlah *outliers* = 27.411	

Pada Tabel 5, terdapat 27.411 outlier dari total 68.205 data sebelum dihapus. Setelah proses penghapusan outlier, jumlah data berkurang menjadi 40.794.

Selanjutnya pada proses transformasi data, penelitian ini melakukan pengubahan data pada variabel *bp_category* dan *bp_category_encode* yang berisikan data *object* menjadi *integer*. Berikut hasil transformasi data dari variabel *bp_category* dan *bp_category_encode* dapat dilihat pada Tabel 6.

Tabel 6. Prosses Transformasi Data Variabel Age

No	*bp_category*	*bp_Category* setelah diubah	*bp_category-encode*	*bp_category_encode* seletah diubah
1.	*Elevated*	0	*Elevated*	0
2.	*Hypertension Stage1*	1	*Hypertension Stage1*	1
3.	*Hypertension Stage2*	2	*Hypertension Stage2*	2
4.	*Normal*	3	*Normal*	3

Pada Tabel 6 menjelaskan bahwa variabel *bp_category* dan *bp_category_encode* yang memiliki 4 kategori telah dilakukan transformasi data dari yang awalnya data berentuk *object* menjadi *integer* seperti *Elevated* menjasi 0, *Hypertension stage* 1 menjadi 1, *Hypertension stage* 2 menjadi 2 dan Normal menjadi 3.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722

Setelah tahap transformasi data, langkah berikutnya dalam penelitian ini adalah melakukan normalisasi data dengan proses yang dilakukan menggunakan fungsi "*MinMaxScaler*" yang menghasilkan nilai dalam bentuk *array*. Selanjutnya, data yang telah dinormalisasi ini akan diubah berdasarkan kolom menggunakan fungsi "*DataFrame*". Hasil dari data yang telah dinormalisasi dapat dilihat pada Tabel 7.

Tabel 7. Sampel Hasil Normalisasi Data

Id	0.00000	0.00001	0.00007	0.00011	0.00014
Age	0.620660	0.471427	0.804180	0.877109	0.871231
Gender	0.0	0.0	0.0	1.0	0.0
Height	0.200000	0.457143	0.057143	0.828571	0.571429
Weight	0.696429	0.321429	0.375000	0.875000	0.607143
Ap-hi	0.935484	0.612903	0.290323	0.612903	0.290323
Ap_lo	0.750000	0.194444	0.472222	0.750000	0.472222
Colesterol	1.0	1.0	0.5	1.0	0.0
Gluc	0.0	0.0	0.5	1.0	0.0
Smoke	0.0	0.0	0.0	0.0	0.0
Alco	0.0	0.0	0.0	0.0	0.0
Active	1.0	0.0	0.0	1.0	1.0
Age-years	0.64	0.48	0.84	0.88	0.88
Bmi	0.914805	0.300958	0.616854	0.649047	0.542974
Bp_category	0.666667	0.333333	0.333333	0.333333	0.333333
Bp_category_encoded	0.666667	0.333333	0.333333	0.333333	0.333333

Pada Tabel 7 terdapat hasil normalisasi data dari 16 variabel, yang bertujuan untuk mengubah fitur-fitur data ke dalam skala umum dengan rentang nilai 0 sampai dengan 1.

3.2 Hasil implementasi SMOTE dan *Random Oversampling*

Pada tahapan ini untuk permasalahan ketidak seimbangan data menggunakan SMOTE dan Random Oversampleing, dikarnakan data yang digunakan pada penelitian ini memiliki imbalance data. Meskipun jumlah imbalane data yang tidak terlalu banyak, tetapi tahapan ini tetap digunakan untuk menghasilkan data yang balance. Berikut hasil penggunaan SMOTE dan Random Oversampling pada Table 8.

Tabel 8. Hasil Penyeimbangan Data Menggunakan SMOTE Dan Random Oversampling

Kategori	Banyak data awal (%)	Banyak data setelah SMOTE(%)	Banyak data setelah Random Oversampler(%)
No Cardio	21.186(51,9%)	21.186 (50%)	21.186 (50%)
Yes Cardio	19.608(48,0%)	21.186 (50%)	21.186 (50%)
Jumlah	40.794 (100%)	42.372(100%)	42.372(100%)

Pada Tabel 8 menjelaskan hasil dari data *imbalance* menjadi *balance* dengan menggunakan metode SMOTE dan *Random Oversampling*. Proses *balancing* data sudah terselesaikan selanjutnya akan melakukan klasifikasi menggunakan algoritma *decision tree* dan *random forest*.

3.3. Hasil Implementasi Algoritma

Tahapan implementasi algoritma diawali dengan membuat suatu variabel X dan Y untuk menyimpan data *input* dan data *output*. Selajutnya dari variabel yang sudah dibuat dijadikan 4 variabel yang mana menjadi *x-train, x-test, y-train* dan *y-test*, Hal ini dilakukan untuk melatih dan menguji suatu dataset pada model. Pada penelitian ini data dibagi menjadi data latih 75% dan data uji sebanyak 25%, pada tahapan data latih dapat menggunakan *library* ".fit" data latih dilakukan untuk membuat suatu pola pada model. Sementara data uji atau *x-test* dan *y-tes* dilakukan untuk menguji hasil kinerja suatu model dengan mengguakan algoritma *decision tree* dan *random forest*. Pada implementasi algoritma ini menggunakan *Decision Tree* dan *Random Fores*, Hasil menunjukan algoritma terbaik adalah algoritma *Random Forest* dengan nilai *accuracy*, *precision*, *recall* dan *F1-score* 69%. Sementara nilai dari algoritma *Decision tree* memiliki *accuracy*, *precision*, *recall* dan *F1-score* memiliki nilai 59%.

3.4 Hasil Evaluasi

Proses evaluasi menghasilkan seberapa baik kinerja pada model, pada penelitian ini menggunakan empat evaluasi diantaranya yaitu *Confusion matrix*, *Classification Report*, *K-fold Cross Validation* dan *Area Under the Curve* (AUC). Berikut hasil evaluasi *Classification Report* pada model dapat dilihat pada Tabel 9.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722

Tabel 9. Hasil Evaluasi Menggunakan Classification Report

Metode	*Classification Report*				
	Precision	*Recall*	*F1-score*	*Support*	*Accuracy*
Decision Tree	0,59	0,59	0,59	10199	59%
Random Forest	0,69	0,69	0,69	10199	69%
SMOTE + Decision Tree	0,61	0,61	0,61	10593	61%
SMOTE + Random Forest	0,70	0,70	0,69	10593	70%
Random Oversampling + Decision Tree	0,62	0,62	0,62	10593	62%
Random Oversampling + Random Forest	0,70	0,70	0,70	10593	70%

Evaluasi *Classification report* pada Tabel 9 menghasilkan hasil *accuracy*, *precision*, *recall*, *f1-score* dan *support* dari setiap metode yang digunakan pada model penelitian ini yaitu, *decision tree*, *random forest*, SMOTE *+decision tree*, SMOTE + *random forest*, *random oversampling + decision tree* dan *random oversampling + random forest.* Dengan menghasilkan nilai *Accuracy* tertinggi pada SMOTE + *Random Forest* dan *Random Oversampling + Random Forest* dengan nilai 70%.

Hasil dari evaluasi menggunakan *Confucion Matrix* dari setiap pengujian algoritma *Decision Tree* dan *Random Forest* serta pengimplementasian SMOTE dan *Random Oversampling*. Hasil tersebut dapat dilihat pada Tabel 10-15.

Tabel 10. Hasil Evaluasi Menggunakan Confucion Matrix Algoritma Decision Tree

		Prediksi	
		1	*0*
Aktual	*1*	*3215*	*2064*
	0	*2089*	*2830*

Pada bagian Tabel 10 hasil dari evaluasi menggunakan *Confucion Matrix* dengan Algoritma *Decision Tree* menghasilkan nilai TP = 3215, FP = 2064, FN = 2089 dan TN = 2830.

Tabel 11. Hasil Evaluasi Menggunakan Confucion Matrix Algoritma Random Forest

		Prediksi	
		1	*0*
Aktual	*1*	*3979*	*1856*
	0	*1325*	*3039*

Tabel 11 menjelaskan hal yang sama seperti tabel sebelumnya yaitu menjelaskan hasil dari evaluasi dengan menggunakan evaluasi *Confucion Matrix* hanya saja pada tabel ini menggunakan Algoritma *Random Forest*. Hasil dari evaluasi *Confucion Matrix* dengan *Random Forest* menghasilkan nilai TP = 3979, FP = 1856, FN = 1325 dan TN = 3039.

Tabel 12. Hasil Evaluasi Menggunakan Confucion Matrix SMOTE + Decision Tree

		Prediksi	
		1	*0*
Aktual	*1*	*3238*	*2044*
	0	*2070*	*3241*

Pada Tabel 12 ini menjelaskan hasil dari evaluasi *Confucion Matrix* menggunakan metode SMOTE dan Algoritma Decision Tree, adapun hasil dari tabel ini menghasilkan nilai TP = 3238, FP =2044, FN = 2070 dan TN = 3241.

Tabel 13. Hasil Evaluasi Menggunakan Confucion Matrix SMOTE + Random Forest

		Prediksi	
		1	*0*
Aktual	*1*	*3836*	*1757*

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722

	0	*1472*	*3528*

Dapat dilihat pada Tabel 13 menjelaskan bahwa evaluasi *confucion matrix* menggunakan metode SMOTE dan algotirma *Random Forest* dengan menghasilkan nilai dari TP = 3836, FP = 1757, FN = 1472 dan TN = 3528.

Tabel 14. Hasil Evaluasi Menggunakan Confucion Matrix Random oversampling + Decesion Tree

		Prediksi	
		1	*0*
Aktual	*1*	*3836*	*1757*
	0	*1472*	*3528*

Bagian Tabel 14 menjelaskan penggunaan evaluasi *confucion matrix* dengan metode Random Oversampling dan algoritma Decision tree, menghasilkan nilai yang sama dengan penerapan pada tabel 10 yaitu TP = 3836, FP = 1757, FN = 1472 dan TN = 3528.

Tabel 15. Hasil Evaluasi Menggunakan Confucion Matrix Random oversampling + Random Forest

		Prediksi	
		1	*0*
Aktual	*1*	*3836*	*1757*
	0	*1472*	*3528*

Bagian terakhir penerapan evaluasi *confucion matrix* yaitu pada Tabel 15, yang mana menjelaskan hasil dari metode *Random oversampling* dan algoritma *Random forest* menghasilkan nilai TP = 3836, FP = 1757, FN = 1472 dan TN = 3528.

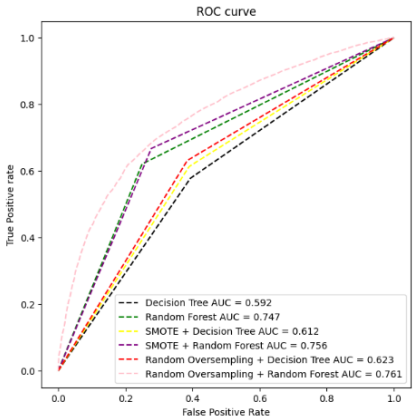
Selanjutnya akan menunjukan hasil dari evaluasi model *K-fold cross validation* menunjukan hasil dengan niali yang sama dari masing- masing metode yang di gunakan, berikut hasil dari masing-masing metode dapat dilihat pada Tabel 16.

Tabel 16. Hasil evaluasi menggunakan K-fold Cross Validation

Metode	Hasil Evaluasi *K-fold Cross Validation*
Decision Tree	0,585
Random Forest	0,693
SMOTE + Decision Tree	0,586
SMOTE + Random Forest	0,693
Random Oversampling + Decision Tree	0,585
Random Oversampling + Random Forest	0,673

Pada bagian Tabel 16 menghasilkan hasil dari evaluasi *K-Fold Cross Validation* dari masing-masing metode yaitu Decision Tree, random Forest, SMOTE + decision Tree, SMOTE + Random Forest, Random oversampling + Deecision tree dan Random oversampling + Randoom forest. Hasil dari metode tersebut dapat menghasilkan nilai yang beragam dengan nilai tertinggi 0,693.

Pada tahapan terakhir menggunakan evaluasi ROC dan AUC akan mentukan nilai dari algoritma terbaik memiliki nilai yang tinggi. Pada tahapan ROC dan AUC menunjukan bahwa algoritma *Random Forest* menggunakan metode *Random Oversampling* yang terbaik dengan nilai AUC 0.761. Sementara nilai terendah pada algoritma *Decision tree* memiliki nilai AUC 0.592. hasil nilai dari ROC dan AUC terdapat pada Gambar 10.



Gambar 5. Hasil Evaluasi menggunakan ROC dan AUC

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722

Pada Gambar 5 menjelaskan hasil dari penerapan evaluasi ROC dan AUC yang mana menjelaskan bahwa algoritma Random Forest dengan menggunakan metode Random Oversampling menjadi yang terbaik dengan nilai tinggi sebesar 0,761. Semntara nilai terendah didapat pada algoritma Decision Tree dengan data original atau data asli nilai sebesar 0,592.

4. KESIMPULAN

Penelitian ini menggunakan algoritma *Random Forest* dan *Decision Tree* dengan penerapan evaluasi *Confusion Matrix, Classification Report, k-Fold Cross Validation,* dan *Area Under the Curve (AUC)*. Berdasarkan hasil pengujian, algoritma terbaik dengan nilai tertinggi adalah *Random Forest* dengan nilai akurasi 69% berdasarkan perhitungan *Confusion Matrix*. Adapun dengan metode SMOTE dan *Random Oversampling* akurasi dapat ditingkatan, dengan SMOTE mendapatkan nilai akurasi sebesar 70% dan *Random Oversampling* sebesar 70%. Nilai AUC pada algoritma *Random Forest* adalah 0,747, pada SMOTE 0,756 dan Random Oversampling 0,761. Sementara algoritma *Decision Tree*, nilai akurasi lebih rendah dibandingkan dengan *Random Forest*, yaitu 59% berdasarkan perhitungan *Confusion Matrix*. Pengujian menggunakan metode SMOTE dan *Random Oversampling* dengan algoritma *Decision Tree* menghasilkan dengan nilai akurasi 61% untuk SMOTE dan 62% untuk *Random Oversampling*. Sementara itu, nilai AUC pada algoritma *Decision Tree* adalah 0,592, untuk SMOTE 0,612, dan *Random Oversampling* menunjukkan peningkatan dengan nilai AUC 0,623. Kesimpulan dari penelitian ini adalah bahwa penggunaan metode SMOTE dan *Random Oversampling* pada data yang tidak seimbang dengan ukuran kecil atau sedikit dapat digunakan. Hal ini dikarenakan penerapan metode tersebut menghasilkan peningkatan yang signifikan dalam akurasi model. Bahkan, dalam beberapa kasus, metode ini justru menyebabkan kenaikan akurasi. Oleh karena itu, perlu dipertimbangkan metode lain atau penyesuaian lebih lanjut ketika menangani data yang tidak seimbang dengan ukuran yang terbatas.

REFERENCES

[1] E. Fauziah and A. Fikri Zulfikar, "OKTAL : Jurnal Ilmu Komputer dan Science Penerapan Metode Decision Tree Menggunakan Algoritma Iterative Dichotomiser 3 (ID3) Untuk Klasifikasi Resiko Penyakit Jantung," *Jurnal Ilmu Komputer dan Science*, vol. 2, no. 4, pp. 1207–1219, 2023.

[2] A. H. Yusufi, A. Kharisma, A. D. Adinata, D. F. Ramzy, and M. M. Santoni, "Prediksi Resiko Kematian Pada Penderita Penyakit Kadiovaskular Menggunakan Metode Ensemble Learning," *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya*, pp. 531–541, 2022.

[3] W. Nugraha, "Prediksi Penyakit Jantung Cardiovascular Menggunakan Model Algoritma Klasifikasi," *Jurnal SIGMATA*, vol. 9, no. 2, pp. 78–84, 2021, [Online]. Available: https://www.kaggle.com/andrewmvd/heart-

[4] N. Fajriati, B. Prasetiyo, and P. Korespondensi, "Optimasi Algoritma Naïve Bayes Dengan Diskritisasi K-Means Pada Diagnosis Penyakit Jantung," *Jurnal Teknologi informasi dan Ilmu Komputer (JTIIK)*, vol. 10, no. 3, pp. 503–512, 2023, doi: 10.25126/jtiik.2023106510.

[5] A. E. Cahyono, "Hipertensi Artikel Review," *Jurnal Perkembangan Ilmu Dan Praktek Kesehatan*, vol. 2, no. 2, pp. 100–117, 2023.

[6] A. Khoeruddin, F. Andriansyah Sudrajat, G. Purnama, I. Kuwangid, and R. Firmansyah, "Optimasi Fitur Seleksi Random Forest Menggunakan GA Dalam Klasifikasi Data Penyakit Gagal Jantung," *JPTIS : Jurnal Penelitian Teknologi Informasi Dan Sains*, vol. 1, no. 2, pp. 1–09, 2023, doi: 10.54066/jptis.v1i2.323.

[7] J. Dwi Muthohhar and A. Prihanto, "Analisis Perbandingan Algoritma Klasifikasi untuk Penyakit Jantung," *Journal of Informatics and Computer Science*, vol. 04, no. 03, pp. 298–304, 2023.

[8] & sriyanto khodijah, "Teknika 17 (2): 419-426 Perbandingan Kinerja Algoritma C4.5. Naive Bayes Dan Random Forest Dalam Prediksi Penyakit Jantung," *IJCCS*, vol. 17, no. 2, pp. 419–426, 2023.

[9] D. V. Ramadhanti, R. Santoso, and T. Widiharih, "Perbandingan SMOTE Dan ADASYN Pada Data Imbalance Untuk Klasifikasi Rumah Tangga Miskin Di Kabupaten Temanggung Dengan Algoritma K-Nearest Neighbor," *Jurnal Gaussian*, vol. 11, no. 4, pp. 499–505, Feb. 2023, doi: 10.14710/j.gauss.11.4.499-505.

[10] R. Arisandi, "PERBANDINGAN MODEL KLASIFIKASI RANDOM FOREST DENGAN RESAMPLING DAN TANPA RESAMPLING PADA PASIEN PENDERITA GAGAL JANTUNG," *Jurnal Gaussian*, vol. 12, no. 1, pp. 136–145, May 2023, doi: 10.14710/j.gauss.12.1.136-145.

[11] B. H. Agtira, H. H. Handayani, and A. F. N. Masruriyah, "Perbandingan Algoritma NBC dan Decision Tree pada Sentimen Analisis Mengenai Vaksinasi Covid-19 Di Indonesia," *remik*, vol. 7, no. 1, pp. 704–712, Jan. 2023, doi: 10.33395/remik.v7i1.12151.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 970−981
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5722

[12] R. Annisa, "Analisis Komparasi Algoritma Klasifikasi Data Mining Untuk Prediksi Penderita Penyakit Jantung," *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 3, no. 1, 2019.

[13] M. Mia, A. F. N. Masruriyah, and A. R. Pratama, "The Utilization of Decision Tree Algorithm In Order to Predict Heart Disease," *JURNAL SISFOTEK GLOBAL*, vol. 12, no. 2, p. 138, Sep. 2022, doi: 10.38101/sisfotek.v12i2.551.

[14] D. J. Muthohhar and A. Prihanto, "Analisis Perbandingan Algoritma Klasifikasi untuk Penyakit Jantung," *Journal of Informatics and Computer Science*, vol. 04, no. 03, pp. 298–304, 2023.

[15] Indarto, Ema Utami, and Suanto Raharjo, "Predikso Resiko Kematian Pasien Stroke Perdarahan Dengan Menggunakan Teknik Klasifikasi Data Mining," *Jurnal Informasi interaktif*, vol. 5, no. 2, pp. 39–91, 2020.

[16] V. Khoirunnisa and S. Lestari, "Implementasi Klasifikasi Kehamilan Beresiko Dengan Metode Naive Bayes Pada Puskesmas Kelurahan Malaka Jaya," *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, vol. 4, no. 3, pp. 1680–1693, Sep. 2023, doi: 10.35870/jimik.v4i3.396.

[17] M. Rizki, M. Fikri Hidayattullah, and Dwi Intan Af'idah, "Klasifikasi Opini Publik di Twitter Terhadap Bakal Calon Presiden Indonesia Tahun 2024 Menggunakan LSTM Secara Realtime Berbasis Website," *Infotekmesin*, vol. 14, no. 2, pp. 285–295, Jul. 2023, doi: 10.35970/infotekmesin.v14i2.1908.

[18] C. Fanny, A. Waworuntu, J. Christian, and J. C. Young, "Implementation of Conditional Random Field for Named Entity Recognition in Indonesian Traditional Arts Digital Article," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, vol. 5, no. 2, pp. 51–55, 2022.

[19] Y. Umaidah, T. Informatika, F. Ilmu Komputer, and U. Singaperbangsa Karawang, "Penerapan Algoritma K-Nearest Neighbor (K-NN) Dengan Pencarian Optimal Untuk Prediksi Prestasi Siswa," *Jurnal Of Information System, Informatics and Computing*, vol. 3, no. 2, pp. 1–8, 2019, [Online]. Available: http://journal.stmikjayakarta.ac.id/index.php/jisicomTelp.+62-21-3905050,

[20] R. Ubaidillah, M. Muliadi, D. T. Nugrahadi, M. R. Faisal, and R. Herteno, "Implementasi XGBoost Pada Keseimbangan Liver Patient Dataset dengan SMOTE dan Hyperparameter Tuning Bayesian Search," *Jurnal Media Informatika Budidarma*, vol. 6, no. 3, pp. 1723–1729, Jul. 2022, doi: 10.30865/mib.v6i3.414