e-ISSN: 2460-870X | p-ISSN: 2085-1367

Perbandingan Logistic Regression, Random Forest, dan Perceptron pada Klasifikasi Pasien Gagal Jantung

Comparison of Logistic Regression, Random Forest, and Perceptron on Classification of Heart Failure Patients

Taufik Zulhaq Jasman*[a,1], Erfan Hasmin[b,2], Sunardi[c,3], Cucut Susanto [d,4], Wilem Musu[e,5]
[a,b,c,d]Teknik Informatika, Universitas Dipa Makassar; Jl. Perintis Kemerdekaan Km. 9 Makassar, (0411)587194
taufikzulhak42@gmail.com*[1], erfan.hasmin@undipa.ac.id[2], sunardi@undipa.ac.id[3], cucut@undipa.ac.id[4], wilem.musu@undipa.ac.id[5]

ABSTRAK

Menurut data WHO (World Health Organization), sepertiga dari kematian secara global disebabkan oleh penyakit jantung. Gagal Jantung menyebabkan kematian kira - kira 17,9 juta penduduk pada tiap tahun di penjuru dunia ini serta memiliki prevalensi yang lebih tinggi di Asia. Dengan bantuan teknologi, data dari analisis biostatistik dapat diolah dengan teknik data mining untuk mencari pola korelasi tiap data dari data historis yang ada sehingga dapat membuat prediksi penyakit jantung jika diimplementasikan dari pola tersebut. Algoritma dengan akurasi yang tinggi setidaknya dapat membantu ahli medis dalam pencegahan kematian akibat penyakit gagal jantung. Tujuan akhir dari penelitian ini dalah untuk membandingkan algoritma dan metode Standard Scaler scaler pada dataset yang telah diimplementasikan metode SMOTE dan tanpa SMOTE, untuk mencari algoritma dan metode yang memiliki akurasi tertinggi dan performa dari algoritma yang terbaik menggunakan nilai kurva ROC dan nilai AUC. Pada penelitian ini, peneliti akan menggunakan algoritma Logistic Regression, Random Forest dan Perceptron. akurasi dari Logistic Regression dengan SMOTE yaitu sebesar 89% kemudian tanpa SMOTE sebesar 87%. Random Forest dengan SMOTE memiliki akurasi sebesar 94% dan tanpa SMOTE sebesar 87%. Kemudian untuk Perceptron dengan SMOTE mendapatkan akurasi sebesar 89% dan tanpa SMOTE memiliki akurasi sebesar 75%. Algoritma Random Forest dengan teknik SMOTE cocok klasifikasi dan prediksi pada penyakit gagal jantung ini. Sehingga dapat membantu para dokter maupun pihak terkait untuk mendiagnosa dan menghindari penyakit gagal jantung ini. Saran untuk penelitian selanjutnya yaitu dapat diterapkan teknik Hyper Parameter Tuning untuk mencari parameter terbaik dari masing – masing algoritma dan menggunakan algoritma yang lain demi menemukan akurasi yang lebih tinggi lagi.

Kata Kunci : Akurasi, Gagal Jantung, Logistic Regression, Perceptron, Random Forest

ABSTRACT

According to WHO (World Health Organization) data, one-third of deaths globally are caused by heart disease. Heart failure causes the death of approximately 17.9 million people worldwide and has a higher prevalence in Asia. With the help of technology, data from the biostatistical analysis can be processed with data mining techniques to find correlation patterns for each data from existing historical data so that it can make predictions of heart disease if implemented from these patterns. Algorithms with high accuracy can at least help medical experts prevent heart failure deaths. The ultimate goal of this research is to compare the algorithms and methods of the Standard Scaler scaler on datasets that have implemented the SMOTE method and without SMOTE, to find the algorithm and method with the highest accuracy and performance from the best algorithm using ROC curve values and AUC values. In this study, researchers will use Logistic Regression, Random Forest and Perceptron algorithms. The accuracy of Logistic Regression with SMOTE is 89% then without SMOTE is 87%. Random Forest with SMOTE has an accuracy of 94%, and without SMOTE it is 87%. Then for Perceptron with SMOTE it gets an accuracy of 89% and without SMOTE it has an accuracy of 75%. Random Forest algorithm with SMOTE technique is suitable for classifying and predicting heart failure. So that it can help doctors and related parties to diagnose and avoid heart failure, suggestions for further research are Hyper Parameter Tuning techniques can be applied to find the best parameters for each algorithm and use other algorithms to find higher accuracy.

Keywords : Accuracy, Heart Failure, Logistic Regression, Perceptron, Random Forest

DOI: https://www.doi.org/10.22303/csrid.14.3.2022.271-286

Info Artikel :
Disubmit: 15 September 2022 *Direview: 16 Oktober 2022* *Diterima: 08 November 2022*

1. PENDAHULUAN

Sekitar 26 juta pemuda - pemudi di seluruh dunia menderita penyakit gagal jantung [1]. Menurut data WHO (World Health Organization), sepertiga dari kematian secara global disebabkan oleh penyakit jantung. Gagal Jantung menyebabkan kematian kira - kira 17,9 juta penduduk pada tiap tahun di penjuru dunia ini serta memiliki prevalensi yang lebih tinggi di Asia [2]. Penyakit jantung adalah satu dari sekian penyakit yang lumayan beresiko kala melanda seseorang, dimana pemicu inti dari penyakit jantung ini berpokok dari pola kehidupan hidup seseorang yang tidak memperhatikan kesehatannya, memakan santapan yang memiliki kolesterol tinggi, pemakaian minuman beralkohol, tembakau, diet yang berlebihan dan berbagai pemicu yang lain. Penyakit jantung ini seringkali dialami oleh mereka yang laki–laki, dimana komparasinya mulai dari satu hingga 3 mungkin menderita penyakit jantung di bawah umur 60 tahunan. Sedangkan untuk kaum perempuan perbandingannya kira – kira satu dari 10 probabilitas yang menderita penyakit jantung ini. Skala yang lumayan besar itu terpaut mengenai jantung, menjadikan penyakit jantung ini sebagai salah satu penyakit yang hendak menciptakan sejumlah besar data penderita penyakit jantung [3]. Dengan bantuan teknologi, data dari analisis biostatistik dapat diolah dengan teknik data mining untuk mencari pola korelasi tiap data dari data historis yang ada sehingga dapat membuat prediksi penyakit jantung jika diimplementasikan dari pola tersebut [4].

Ada beberapa penelitian yang dilakukan sebelumnya yaitu: di penelitian [4], Sri Rahayu dkk. Melakukan komparasi algoritma dengan data yang sama dengan yang penulis akan gunakan di penelitian ini, dengan menggunakan algoritma SVM, Random Forest, ANN, Decision Tree, Naïve Bayes dan KNN. Di penelitian tersebut, mereka menggunakan metode SMOTE untuk mengatasi data yang tidak seimbang, yang mana penulis juga akan menggunakannya di penelitian ini. Serta menggunakan Cross Validation sebagai validasi data training serta data testing. Menghasilkan akurasi Random Forest memiliki akurasi tertinggi sebesar 85.82% dan algoritma yang paling rendah yaitu Naïve Bayes dengan akurasi sebesar 77.21%. Di penelitian kedua [5], juga dengan data yang sama, menggunakn algoritma Naïve Bayes dengan implementasi metode Particle Swam Optimization dengan menghsilkan akurasi 75% tanpa menggunakan PSO dan 91.67% dengan menggunakan PSO.

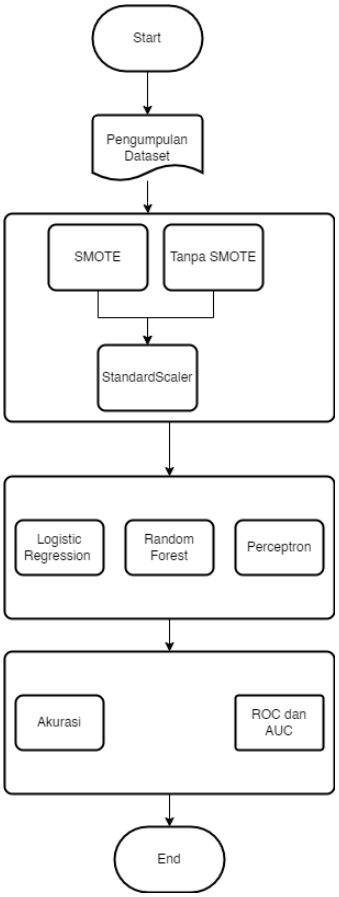
Pada penelitian ketiga dilakukan komparasi algoritma dengan menggunakan model Naïve Bayes, SVM, Decision Tree, Logistic Regression dan Backpropgation dengan Cross Validation sebagai validasi data training juga data testing. Menghasilkan akurasi Naive Bayes dengan akurasi tertinggi sebesar 84.07% disusul Logistic Regerssion 82.52%, Backpropagation dan SVM sebesar 81.85%, dan Decision Tree sebesar 74.81%. Di penelitian ke empat, Kleyko dkk. Melakukan prediksi akurasi Neural Network menggunakan Perceptron dengan kesimpulan bahwa perceptron secara akurat dapat memprediksi kinerja dari Echo State Networks [6]. Di penelitian kelima dilakukan oleh Jing Wang [7]. Dengan mengkomparasikan akurasi dan F1 score dari Min-Max Normalization dan Z-score Normalization dengan menggunakan SMOTE dan tidak menggunakan SMOTE. Ada 18 algoritma atau model yang dikomparasikan di penelitian tersebut. Kesimpulan dari penelitian tersebut adalah z-score lebih baik daripada min-max normalization dengan menggunakan SMOTE untuk kelas yang tidak seimbang.

Berdasarkan penelitian di atas, peneliti akan menggunakan algoritma Logistic Regression, Random Forest dan Perceptron. Untuk standarisasi data menggunakan teknik StandardScaler. Serta menggunakan teknik SMOTE dan tidak menggunakan SMOTE untuk mengatasi kelas yang tidak seimbang.

e-ISSN: 2460-870X | p-ISSN: 2085-1367

Tujuan akhir dari penelitian ini dalah untuk membandingkan ketiga algoritma pada dataset yang telah diimplementasikan metode SMOTE dan tanpa SMOTE, untuk mencari algoritma dan metode yang memiliki akurasi tertinggi dan performa dari algoritma yang terbaik menggunakan nilai kurva ROC dan nilai AUC.

2. METODE



Gambar 1. Alur Penelitian

Gambar 1 merupakan tahapan - tahapan dari penelitian ini. Tahapan penelitian sangat penting agar penelitian memlilki struktur dan arah yang jelas.

Jenis penelitian yang akan dilaksanakan pada penelitian ini adalah jenis penelitian kuantitatif. Karena data yang akan diolah berbentuk angka dan hasilnya akan dianalisis. Penelitian kuantitatif merupakan penelitian yang berisi dengan data berformat angka dalam metode pengumpulan data - data di lapangan [8].

Pada penelitian ini, peneliti akan menggunakan perangkat lunak VScode dan diinstall Jupyter Notebook di dalamnya dengan bahasa pemrogramannya adalah bahasa Python.Terakhir, selesaikan konten dan pengeditan organisasi sebelum memformat. Harap perhatikan hal-hal berikut saat mengoreksi ejaan dan tata bahasa:

A. Pengumpulan Dataset

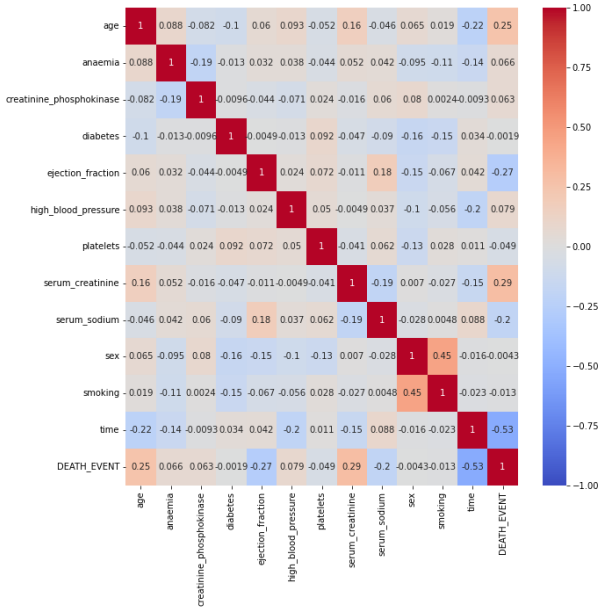
DOI: https://www.doi.org/10.22303/csrid.14.3.2022.271-286

Dataset merupakan sekumpulan dari berbagai jenis data yang disimpan dalam format digital. Data adalah komponen kunci dari setiap proyek Machine Learning. Kumpulan data tersebut terdiri dari gambar, teks, audio, video, titik data numerik, dll [9].

Dataset yang akan digunakan pada penelitian ini merupakan data *Heart Failure* yang didownload berasal dari *Kaggle*. Di mana data ini terdiri dari 299 pasien penderita gagal jantung. Jumlah fitur dari data ini sebanyak 13 fitur, di mana kelas dari dataset ini berupa bilangan *boolean* yaitu 0 mewakilkan (*Alive)* dan 1 *(Death)*. Tabel 2 merupakan tabel deskripsi dari seluruh fitur pada dataset ini.

Tabel 1. Deskripsi Fitur Dataset

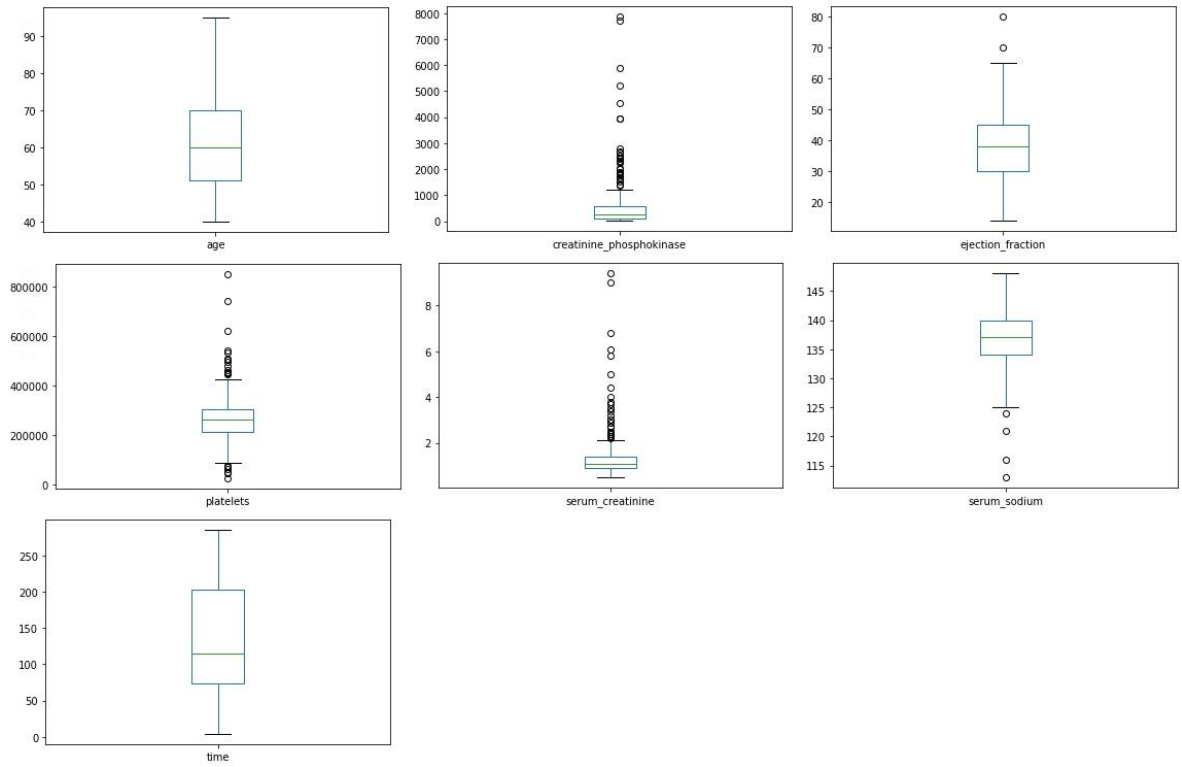
Fitur	Keterangan	Satuan
Age	Umur dari pasien	*Years*
Anaemia	Mengurangnya sel hemoglobin	*Boolean*
High Blood Pressure	Apabila pasien memiliki tekanan darah tinggi	*Boolean*
Creatinine Phospokinase	Konsentrasi CPK pada darah	mcg/L
Diabetes	Apabila pasien atau penderita memiliki riwayat Glukosaria	*Boolean*
Ejection Fraction	Tingkat persen darah yang hilang tiap kontraksi jantung	*Percentage*
Platelets	Trombosit pada darah	Kiloplatelets/mL
Sex	Jenis kelamin	*Binary*
Serum Cretinine	Konsentrasi kreatinin pada darah	mg/dL
Serum Sodium	Konsentrasi serum sodium pada darah	mEq/L
Smoking	Jika pasien adalah perokok	*Boolean*
Time	Durasi follow up	*Days*
Death Event	Jika pasien meninggal selama follow up	*Boolean*



Gambar 2 Heatmap Correlation

e-ISSN: 2460-870X | p-ISSN: 2085-1367

Gambar 2 merupakan korelasi antara atribut satu sama lain. Dari gambar tersebut didapati bahwa terdapat korelasi postif antara *age* dan *DEATH_EVENT*, *serum_creatinine* dan *DEATH_EVENT* dan korelasi negatif antara *time* dan *DEATH_EVENT*.



Gambar 3. Outlier dari data kolom numerik

Pada Gambar 3 terlihat pada kolom *creatinine_phosphokinase*, *ejection_fraction* dan *serum_creatinine* outliernya berada di atas nilai maksimum sedangkan kolom *platelets* memiliki outlier di atas nilai maksimum dan di bawah nilai minimum. Untuk *serum_sodium* hanya memiliki 4 outlier di bawah nilai minimum. Sedangkan kolom *age* dan *time* tidak memiliki outlier sama sekali.

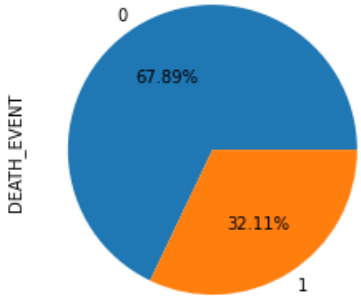
B. Preprocessing

Ada beberapa atribut yang memliki tipe data boolean yang berarti 0 sebagai False dan 1 sebagai True. Di mana pada kasus dataset ini, angka 0 sebagai "tidak menderita" dan 1 sebagai "penderita". Untuk kelas dari data ini yaitu *Death Event*, nilai boolean 0 sebagai "pasien yang masih hidup" sedangkan angka 1 sebagai "pasien yang telah wafat". Mengubah data boolean menjadi numerik dapat dilakukan dengan menggunakan *library* pandas. Di mana data kolom di *replace* yang awalnya boolean menjadi numerik.

Pada tahap ini akan dilakukan proses menyeimbangkan data. Di mana di dataset ini terdapat ketidakseimbangan data. Jumlah data yang tidak seimbang yaitu pasien dengan kelas 0 berjumlah 203 pasien dan pasien dengan kelas 1 ada 96 pasien. Permasalahan ketidakseimbngan ini merupakan permasalahan yang muncul, yaitu nilai kinerja model atau model menghasilkan akurasi yang cukup tinggi dikarenakan total kelas mayoritas yang banyak. Namun, persoalan tersebut sebenarnya menghasilkan performa klasifikasi yang buruk di saat melakukan klasifikasi label pada kelas minoritas [10]. Setelah mengatasi tidak seimbangnya data, tahapan selanjtnya yaitu standarisasi data dengan menggunakan Standard Scaler. Akan dilakukan komparasi terhadap data yang menggunkan SMOTE dan tanpa SMOTE serta menggunakan Standard Scaler. Setelah tahapan imbalance data dan standarisasi.

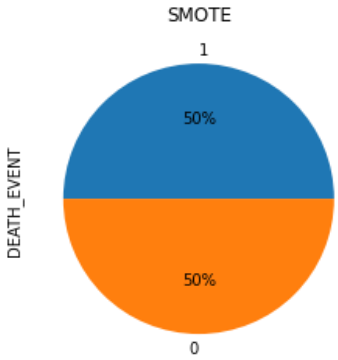
Dilakukan proses Split Validation di mana data latih dan data uji dibagi menjadi 80% data latih dan 20% data uji.

1) SMOTE
   Synthetic Minority Oversampling Technique atau sering disingkat dengan SMOTE adalah turunan dari teknik Oversampling. Teknik SMOTE beroperasi dengan menggandakan data kelas minor. Penggandaan itu disebut dengan nama data sintetis (syntetic data). Teknik ini bekerja dengan mencari KNN pada setiap data di dalam kelas minor, sesudah proses tersebut dibuatlah data sintetis sebanyak persentase data yang diduplikasi yang dibutuhkan diantara data minoritas dan KNN yang dipilih dengan acak [11]. Gambar 4 merupakan persentase kelas pada dataset ini sebelum diimplementasikan metode SMOTE.



Gambar 4. Grafik Perbandingan Kelas Sebelum SMOTE

Pada Gambar 4 terlihat perbedaan kelas yang sangat jauh. Untuk kelas 0 yaitu pasien yang masih hidup digambarkan pada area berwarna biru sedangkan yang kuning untuk pasien yang telah wafat. Persentase kelas alive sebanyak 67.89%, sedangkan kelas death sebanyak 32.11%. Persentase selisih dari kelas tersebut dalah 30.78%. di mana selisih tersebut lumayan banyak. Di mana jumlah pasien dari yang masih hdup terlihat mendominasi.



Gambar 5. Grafik Perbandingan Kelas Setelah SMOTE

Terlihat pada Gambar 5, dataset telah diseimbangkan dengan menggunakan teknik SMOTE yang mana terjadi perbedaan yang signifikan dari Gambar 2. Bahwa persentase jumlah kedua kelas telah seimbang.

e-ISSN: 2460-870X | p-ISSN: 2085-1367

2) Standard Scaler

StandardScaler merupakan salah satu teknik penting terutama dilakukan dalam langkah - langkah *preprocessing* sebelum proses pada *machine learning*, digunakan untuk menstandarisasi rentang fungsionalitas input pada dataset [12].

StandardScaler menstandardisasi atribut dengan mengurangi rata - rata selanjutnya menskalakan ke varians unit. Varians unit yang berarti membagi seluruh value menggunakan standar deviasi. StandardScaler tidak memenuhi definisi skala yang diperkenalkan sebelumnya. StandardScaler menghasilkan distribusi yang mana standar deviasi adalah 1. Varians sama dengan 1 juga, karena varians = standar deviasi kuadrat. Dan 1 kuadrat = 1. StandardScaler membuat rata-rata distribusi mendekati 0 [13].

Untuk tipe data boolean, dikarenakan StandardScaler hanya untuk data numerik. Maka data booelan tersebut harus dikonversi dulu ke dalam bentuk data numerik.

C. Modelling

Pada tahap ini dilakukan permodelan pada algoritma. Model yang akan dipakai di penelitian ini ialah model Logistic Regression, Random Forest dam Perceptron.

1) Logistic Regression

Model Logistic Regression atau regresi logistik yaitu algoritma untuk suatu variabel prediksi X serta variabel Y di mana saling berkontradiksi. Variabel Y = 1 menyatakan bahwa terdapat karakter Y = 0 menyatakan tidak terdapat karakter. Algoritma Logistic Regression yang variabel responnya memiliki dua kelas disebut algortma Linear Regression biner [14]. Logistic Regression memiliki beberapa kelebihan yaitu: a. Logistic regression adalah salah satu algoritma yang paling sederhana dan mudah diterapkan serta memberikan efisiensi training yang baik dalam beberapa kasus. Juga karena alasan ini, melatih model dengan algoritma ini tidak memerlukan daya komputasi yang tinggi. b. Parameter yang diprediksi (bobot terlatih) memberikan inferensi tentang pentingnya setiap fitur. Arah asosiasi yaitu positif atau negatif juga diberikan. Sehingga kita dapat menggunakan algoritma ini untuk mengetahui hubungan antara fitur yang satu dengan yang lain. c. Algoritma ini memungkinkan model untuk diperbarui dengan mudah untuk memproses data baru, tidak seperti random forest atau svm. Pembaruan dapat dilakukan menggunakan penurunan gradien stokastik [15]. Selain kelebihan algoritma ini memiliki beberapa kekurangan: a. Jika fitur independen dikorelasikan, dapat memengaruhi kinerja algoritma ini. b. Cukup rentan terhadap data noise data dan overfitting data. c. Masalah non-linier tidak dapat diselesaikan dengan algoritma ini karena memiliki permukaan keputusan linier. Data yang dapat dipisahkan secara linier jarang ditemukan dalam skenario dunia nyata [16].

Rumus sederhana dari Logistic Regression yaitu [17]:

$$Y = \beta 0 + \beta 1X + \varepsilon \quad (1)$$

Diketahui:

Y: variabel dependen
β0: konstatnta
β1: koefisien regresi
X: variabel bebas
ε: kesalahan acak

2) Random Forest

Random Forest merupakan model atau algoritma yang memakai teknik pembagian kelas biner berulang demi mendapat titik final di dalam struktur pohon keputusan berdasar dari pohon klasifikasi serta regresi. Model ini memiliki kelebihan – kelebihan diantaranya dapat menciptakan galat yang lumayan rendah, kinerja yang bagus dalam pengklasifikasian, mampu menghindari data latih dalam total yang besar secara efektif, juga teknik yang lumayan efektif

untuk memprediksi data yang hilang. Diamping memiliki kelebihan, algoritma ini juga memiliki beberapa kekurangan di antaranya meskipun random forest dapat menjadi perbaikan pada decision tree, teknik yang lebih canggih telah tersedia. Akurasi prediksi pada masalah kompleks biasanya lebih rendah daripada pohon *gradient - boosted*, *forest* kurang dapat diinterpretasikan daripada sebuah decision tree. Pohon tunggal dapat divisualisasikan sebagai urutan keputusan, *forest* yang telah dilatih mungkin memerlukan memori yang signifikan untuk penyimpanan, karena kebutuhan untuk menyimpan informasi dari beberapa ratus pohon individu[18]. Random Forest ini menghasilkan beraneka ragam pohon bebas dengan subset yang dipilih dengan acak melalui teknik bootstrap dari sampel pelatihan serta dari variable input pada setiap node [19].

3) Perceptron

Perceptron adalah salah satu model algoritma *supervised learning*. Model perceptron mendeteksi apakah suatu fungsi merupakan input atau tidak dan mengklasifikasikannya ke dalam satu kelas label [20]. Perceptron adalah algoritma dari ANN yang dipakai untuk melakukan proses klasifikasi suatu pola yang lebih dikenali sebagai pola pembagian secara linier [21]. Perceptron adalah algoritma klasifikasi linear. Hal ini berarti bahwa algoritma mempelajari batas keputusan yang memisahkan dua kelas menggunakan garis (*hyperplane*) pada area fitur. Oleh karena itu, algoritma ini cocok untuk masalah di mana kelas - kelas dapat dipisahkan dengan baik oleh garis ataupun linear model, yang yang artinya dapat dipisahkan secara linier[22].

D. Komparasi Performa

Di penelitian ini dilakukan komparasi dari akurasi dari ketiga model yang digunakan. Yaitu menghitung serta membandingkan nilai akurasi dan juga membandingkan dario kurva ROC dan nilai AUC.

1) Akurasi

Akurasi merupakan rasio prediksi Benar (positif dan negatif) dengan keseluruhan data [23]. Prediksi positif dan negatif diperoleh dari tabel confusion matrix. Selain akurasi juga ada precision, recall, dan F1-score:

Akurasi= $(TP+TN)/(TP+TN+FP+FN)$ (2)

Presisi= $TP/(TP+FP)$ (3)

Recall= $TP/(TP+FN)$ (4)

F1-score= $2*(Precision*Recall)/(Precision+Recall)$ (5)

Di mana untuk TP, TN, FP, FN didapat dari tabel confusion matrix:

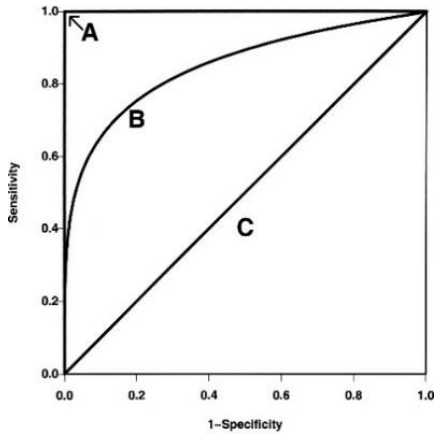
Tabel 2. Confusion Matrix

		PREDICTION	
		NEGATIVE	**POSITIVE**
TRUE	**NEGATIVE**	True Negative	False Positive
	POSITIVE	False Negative	True Positive

Confusion matrix berukuran n x n yang berhubungan dengan algoritma pengklasifikasian di mana menunjukkan klasifikasi yang diprediksi dan yang aktual, di mana n adalah jumlah kelas yang berbeda [24]. Confusion matrix adalah ringkasan hasil prediksi pada kasus klasifikasi. *Confusion matrix* menunjukkan kebingungan dari model klasifikasi ketika membuat prediksi [25].

e-ISSN: 2460-870X | p-ISSN: 2085-1367

2) Kurva ROC dan Nilai AUC
Setelah akurasi, selanjutnya menampilkan performa model dengan kurva ROC (*Receiver Operating Characteristic*) serta membandingkan nilai AUC (*Area Under Curve*). Kurva ROC sering digunakan untuk menggambarkan grafis hubungan atau pertukaran antara sensitivitas (*sensitivity*) dan spesifisitas (*specificity*) untuk setiap kemungkinan batas untuk suatu tes atau kombinasi dari tes [26]. Selama 40 tahun terakhir, analisis menggunakan teknik ROC telah menjadi metode yang terkenal untuk menilai keakuratan sistem diagnosis pada dunia medis. Properti yang paling diinginkan dari analisis ROC ini adalah bahwa indeks akurasi yang diperoleh dari teknik ini tidak terdistorsi oleh fluktuasi yang disebabkan oleh penggunaan berlebihan dari kriteria keputusan yang dipilih. Yang artinya, indeks dari akurasi tidak dipengaruhi oleh kriteria dari suatu keputusan (yaitu kecenderungan dari pembaca atau peneliti untuk memilih batas (*threshold*) tertentu pada variabel pemisah tersebut) dan untuk mempertimbangkan kemungkinan sebelumnya. [27]. Grafik kurva ROC dihasilkan dari mengilustrasikan sensitivitas (*True Positive Rate*) terhadap sumbu y terhadap 1-spesifisitas (*False Positive Rate*) di sumbu x pada berbagai nilai yang didiagramkan [28].



Gambar 6. Contoh Kurva ROC

Pada Gambar 6. Tiga kurva hipotetis ROC yang mewakili akurasi diagnosa pada *gold standard* (garis A; AUC=1) pada sumbu atas dan di kiri persegi, tipe kurva ROC (kurva B; AUC=0,85), dan garis diagonal yang sesuai dengan peluang acak (garis C; AUC=0,5). Saat akurasi diagnosa meningkat, kurva ROC bergerak menonjol ke titik A, dan AUC mendekati 1 [29].
Area Under the Curve (AUC) merupakan ukuran kemampuan algoritma klasifikasi untuk membedakan antara kelas dan digunakan sebagai ringkasan dari kurva ROC. Semakin tinggi AUC, semakin baik kinerja model dalam membedakan antara kelas positif dan negatif. Semakin tinggi nilai AUC untuk suatu classifier, semakin baik kemampuannya untuk membedakan antara kelas positif dan negatif [30].

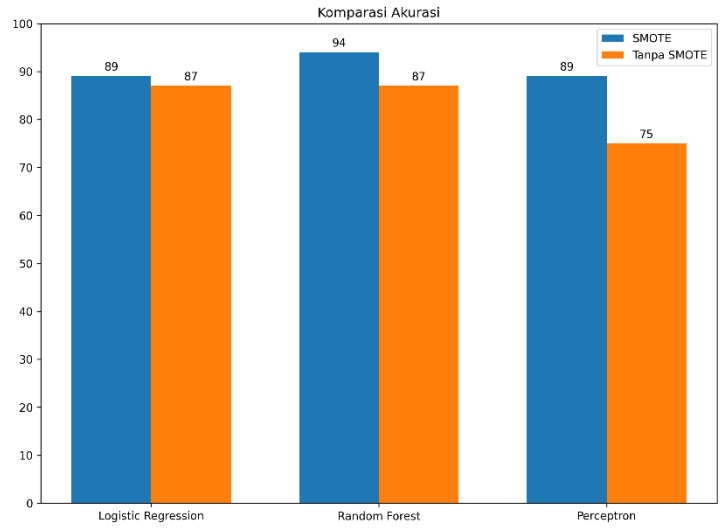
3. HASIL DAN PEMBAHASAN

Pada tahapan hasil dan pembahasan, peneliti mengkomparasikan Standard Scaler dengan SMOTE, Standard Scaler tanpa SMOTE.

A. Akurasi

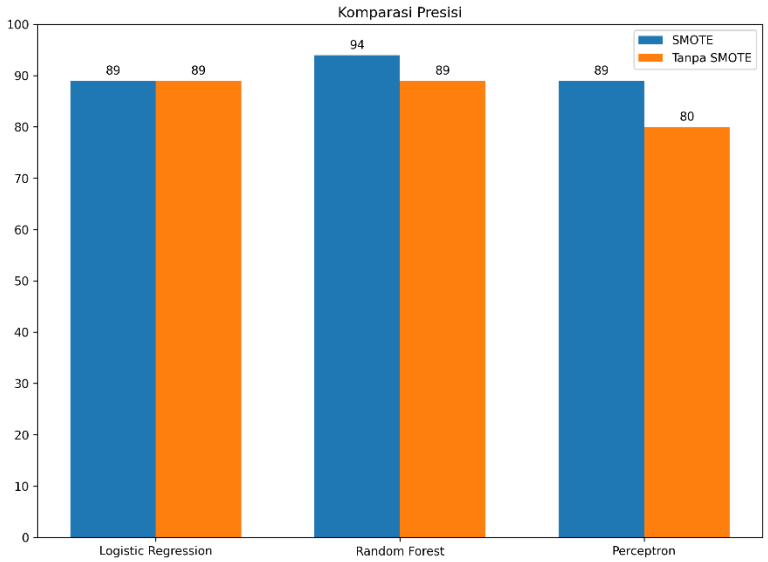
Hasil akurasi dari ketiga algoritma ditunjukkan pada Gambar 7.

DOI: https://www.doi.org/10.22303/csrid.14.3.2022.271-286



Gambar 7. Grafik Perbandingan Akurasi

Pada Gambar 7, akurasi dari Logistic Regression dengan SMOTE yaitu sebesar 89% kemudian tanpa SMOTE sebesar 87%. Random Forest dengan SMOTE memiliki akurasi sebesar 94% dan tanpa SMOTE sebesar 87%. Kemudian untuk Perceptron dengan SMOTE mendapatkan akurasi sebesar 89% dan tanpa SMOTE memiliki akurasi sebesar 75%.

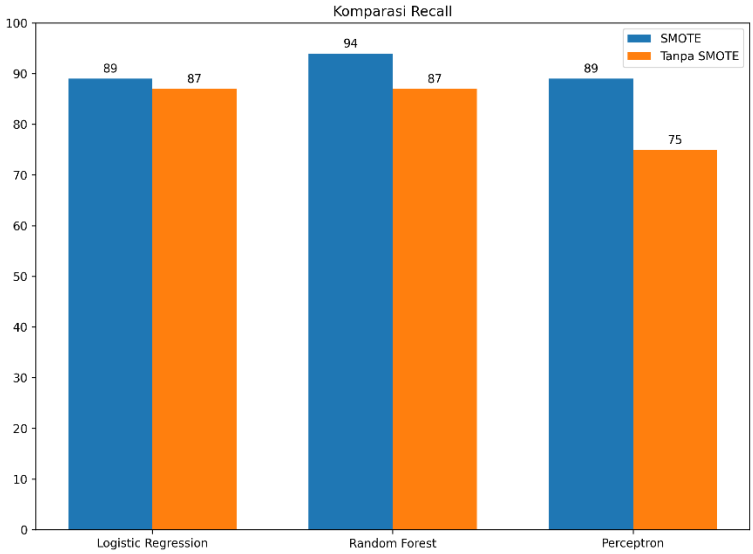


Gambar 8. Komparasi Presisi

Nilai presisi untuk Logistic Regression menggunakan SMOTE dan Tanpa SMOTE bernilai sama yaitu 89%. Sementara Random Forest terjadi selisih sebesar 5%. Kemudian untuk Perceptron selisih antara menggunakan MSOTE dan Tanpa SMOTE sebesar 9%.

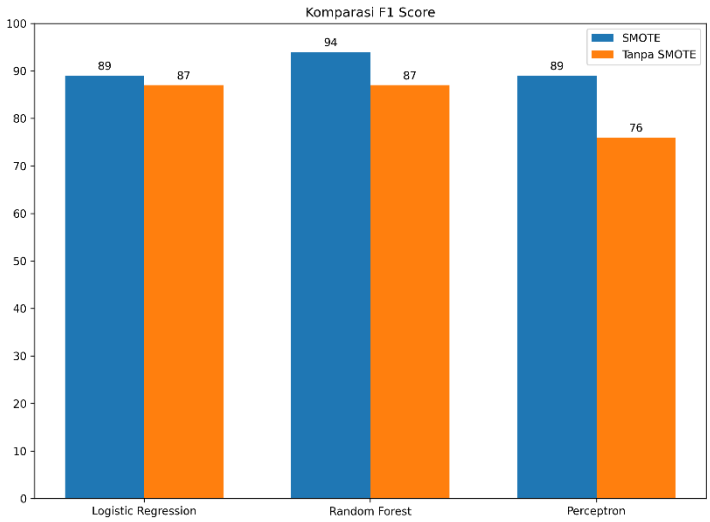
Dari hasil di atas terlihat bahwa akurasi prediksi model untuk Random Forest dengan SMOTE lah yang paling tinggi yaitu sebesar 94%. Di mana Random Forest memprediksi bahwa 94% pasien yang diprediksi meninggal dari jumlah pasien yang diprediksi meninggal.

e-ISSN: 2460-870X | p-ISSN: 2085-1367



Gambar 9. Komparasi Recall

Pada komparasi *recall* terlihat sama dengan akurasi. Di mana Random Forest dengan SMOTE memiliki akurasi tertinggi sedangkan Perceptron Tanpa SMOTE memiliki akurasi terendah.



Gambar 10 Komparasi F1 Score

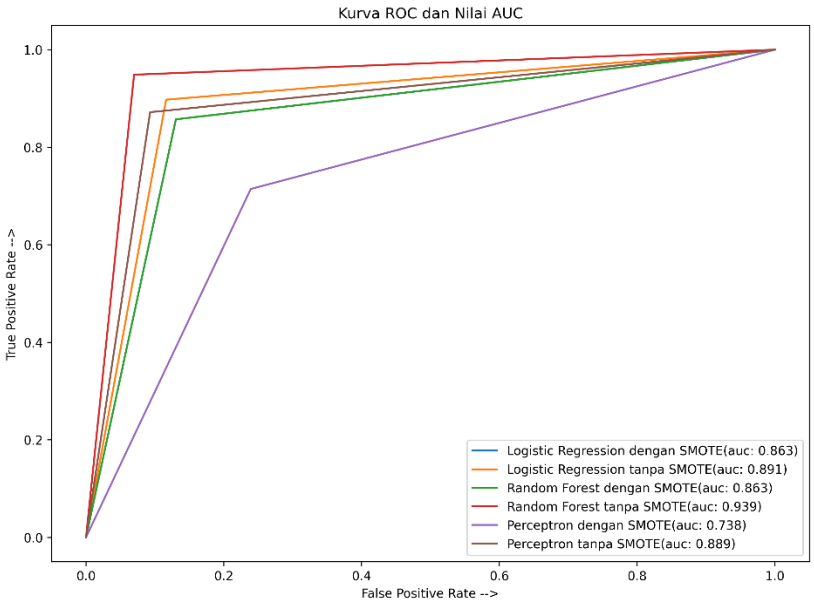
Nilai F1 Score untuk Logistic Regression untuk tanpa SMOTE dan menggunakan SMOTE memiliki selisih sebesar 2% yaitu 87% dan 89%. Kemudian untuk Random Forest F1 Score dengan menggunakan SMOTE sebesar 94% lebih tinggi dari Tanpa SMOTE sebesar 87%. Sedangkan untuk Perceptron untuk SMOTE sebesar 89% dan tanpa SMOTE 76%.

Untuk nilai F1 Score didapat dari perbandingan antara presisi dan recall. Random Forest memiliki akurasi tertinggi sedangkan Perceptron memiliki akurasi terendah.

B. Kurva ROC dan Nilai AUC

Untuk kurva ROC dan nilai AUC dari Standard Scaler dengan dan tanpa SMOTE terdapat pada Gambar 9.

DOI: https://www.doi.org/10.22303/csrid.14.3.2022.271-286



Gambar 11. Kurva ROC dan Nilai AUC Standard Scaler - SMOTE

Pada grafik kurva di Gambar 11. Random Forest dengan SMOTE (hijau), berada paling atas dan yang paling mendekati dari dari garis TPR, untuk nilai AUC sebesar 0.939 dan kurva tanpa SMOTE (merah) berada di bawah kurva biru, ungu dan kuning dan untuk nilai AUC sebesar 0.863. Ketika garis kurva mendekati garis TPR maka akan semakin tinggi nilai AUC tersebut, sebaliknya ketika garis kurva menjauhi garis TPR maka nilai AUC akan semakin menurun. Kurva dari Logistic Regression dengan SMOTE (biru) berada ketiga di bawah kurva hijau dan ungu sedangkan untuk nilai AUC dari Logistic Regression sebesar 0.891. Kemudian disusul kurva tanpa SMOTE (oranye), di mana berada di posisi yang sama dengan kurva merah dan untuk nilai AUC memiliki nilai yang sama dengan nilai AUC dari kurva merah yaitu sebesar 0.863, lebih rendah dari yang menggunakan SMOTE. Kemudian kurva untuk Perceptron dengan SMOTE (ungu) berada di posisi kedua di bawah kurva hijau yang berarti nilai AUC dari algoritma ini juga tinggi yaitu sebesar 0.889 dan kurva untuk Perceptron tanpa SMOTE (cokelat) merupakan yang paling jauh dari garis TPR yang mana mempengaruhi nilai AUC yaitu senilai 0.738.

Kemudian untuk pengukuran waktu dan penggunaan memori masing masing algoritma yaitu:

Tabel 3. Perbandingan penggunaan waktu dari masing – masing algoritma

	SMOTE	**Tanpa SMOTE**
Logistic Regression	0.033s	0.067s
Random Forest	0.482s	2.297s
Perceptron	0.077s	0.027s

Perbandingan penggunaan waktu eksekusi program menggunakan *library time*, terlihat pada Tabel 3. Proses penggunaan waktu eksekusi dengan metode SMOTE yaitu Random Forest menggunakan waktu yang paling banyak yaitu 0.482 detik kemudian disusul oleh Perceptron sebesar 0.077 detik dan Logistic Regression sebesar 0.033 detik. Dan untuk Tanpa SMOTE Random Forest berada di paling atas sebesar 2.297 detik, Logistic Regresson 0.067 detik dan Perceptron sebesar 0.027 detik. Random Forest menggunakan penggunaan waktu terbanyak dikarenakan algoritma ini terdiri dari sekumpulan *decision tree*.

e-ISSN: 2460-870X | p-ISSN: 2085-1367

Tabel 4. Perbandingan penggunaan memori dari masing – masing algoritma

	SMOTE	Tanpa SMOTE
Logistic Regression	55065, 84825	44992, 72715
Random Forest	79903, 148306	80283, 132778
Perceptron	48763, 107872	11381, 56561

Pada penelitian ini peneliti mengukur tingkat penggunaan memori mengunakan *library tracemalloc*. Di mana (*current*, *peak*) *current* merupakan jumlah memori yang digunakan saat ini dan *peak* adalah kapasitas maksimum memori yang program gunakan sewaktu dieksekusi. Pada Tabel 4 ditunjukkan bahwa penggunaan memori pada Logistic Regression dengan SMOTE sebesar (55065, 84825) dan Tanpa SMOTE sebesar (44992, 72715). Untuk random Forest sebesar dengan SMOTE (79903, 148306) sementara Tanpa SMOTE (80283, 132778). Sedangkan untuk Perceptron dengan SMOTE sebesar (48763, 107872) dan Tanpa SMOTE sebesar (11381, 56561). Dari ketiga algoritma tersebut Random Forest yang paling banyak menggunakan penyimpanan dari penyimpanan saat ini hingga penyimpanan maksimum. Hal ini selaras dengan konsep dari algoritma ini yang menggunakan sekumpulan Decision Tree untuk proses pengklasifikasiannya. Kemudian disusul oleh Perceptron dan Logistic Regression.

4. KESIMPULAN

1) Akurasi dan performa dari algoritma dengan menggunakan teknik SMOTE dan Standard Scaler lebih baik dibanding tidak menggunakan SMOTE dalam kasus dataset penyakit gagal jantung ini.
2) Dari segi performa dan akurasi dalam dataset ini, Random Forest memiliki kinerja yang sangat baik namun dari segi efisiensi waktu dan memori, algoritma ini tidak efisien.
3) Meskipun akurasi tidak sebaik Random Forest dan tidak seburuk Perceptron, Logistic Regression efisien dalam hal penggunaan memori dan waktu.
4) SMOTE sangat berpengaruh dalam peningkatan akurasi pada dataset ini dan cocok untuk mengatasi data yang tidak seimbang.
5) Algoritma Random Forest dengan teknik SMOTE cocok klasifikasi dan prediksi pada penyakit gagal jantung ini. Sehingga dapat membantu para dokter maupun pihak terkait untuk mendiagnosa dan menghindari penyakit gagal jantung ini.
6) Meskipun dapat meningkatkan akurasi, SMOTE memiliki kekurangan yaitu SMOTE melakukan oversampling pada data noise.
7) Penelitian ini hanya menggunakan satu dataset, kinerja untuk akurasi, waktu, dan memori dari masing – masing algoritma dapat saja berbeda pada dataset yang lain.
8) Pada penelitian ini penulis tidak menambahkan parameter apapun pada masing – masing algoritma, kecuali untuk Random Forest, penulis menambahkan satu parameter yaitu *random_state* berjumlah 1. Selain itu pada penelitian ini penulis mengabaikan adanya outlier. Hal tersebut dapat mempengaruhi akurasi dan performa dari masing – masing algoritma.
9) Di penelitian selanjutnya peneliti akan memakai algoritma klasifikasi yang lain, juga akan menggunakan beberapa metode utnuk meningkatkan akurasi dengan menggunakan *feature selection* atupun *hyper parameter tuning* dengan teknik *cross valodation*.

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada Universitas Dipa Makassar yang telah membantu seluruh kegiatan peneliti sehingga penelitian ini dapat diselesaikan dengan baik, tak lupa pula peneliti juga

mengucapkan pada semua pihak yang terlibat secara langsung atau tidak langsung yang tidak dapat peneliti ucapkan satu persatu.

REFERENSI

[1] P. Ponikowski *et al.*, "Heart failure: preventing disease and death worldwide," 2014, doi: 10.1002/ehf2.12005.

[2] W. Nugraha, "Prediksi penyakit jantung cardiovascular menggunakan model algoritma klasifikasi," *Jurnal Sigmata*, vol. 9, no. 2, pp. 78–84, 2021.

[3] P. D. Putra and D. P. Rini, "Prediksi Penyakit Jantung dengan Algoritma Klasifikasi," in *Prosiding Annual Research Seminar*, 2019, vol. 5, no. 1. [Online]. Available: http://archive.ics.uci.edu/ml/machine-learning-databases/

[4] S. Rahayu, J. Jaya Purnama, A. Baroqah Pohan, F. Septia Nugraha, S. Nurdiani, and S. Hadianti, "Prediction Of Survival of Heart Failure Patients Using Random Forest," 2020. [Online]. Available: www.ubsi.ac.id

[5] F. Novaldy and A. Herliana, "Penerapan Pso Pada Naïve Bayes Untuk Prediksi Harapan Hidup Pasien Gagal Jantung," *Jurnal Responsif : Riset Sains dan Informatika*, vol. 3, no. 1, pp. 37–43, 2021, doi: 10.51977/jti.v3i1.396.

[6] D. Kleyko, A. Rosato, E. P. Frady, M. Panella, and F. T. Sommer, "Perceptron Theory for Predicting the Accuracy of Neural Networks," Dec. 2020, [Online]. Available: http://arxiv.org/abs/2012.07881

[7] J. Wang, "Heart failure prediction with machine learning: A comparative study," in *Journal of Physics: Conference Series*, Sep. 2021, vol. 2031, no. 1. doi: 10.1088/1742-6596/2031/1/012068.

[8] A. F. Djollong, "Tehnik Pelaksanaan Penelitian Kuantitatif," *Jurnal UM Parepare*, vol. 2, no. 1, pp. 86–100, 2014.

[9] R. Khan, "Importance of Datasets in Machine Learning and AI Research," May 13, 2020. https://www.datatobiz.com/blog/datasets-in-machine-learning/ (accessed Sep. 29, 2022).

[10] A. Y. Triyanto and R. Kusumaningrum, "Implementasi Teknik Sampling untuk Mengatasi Imbalanced Data pada Penentuan Status Gizi Balita dengan Menggunakan Learning Vector Quantization Implementation of Sampling Techniques for Solving Imbalanced Data Problem in Determination of Toddler Nutritional Status using Learning Vector Quantization," vol. 19, pp. 39–50, 2017.

[11] R. Siringoringo, "Klasifikasi Data Tidak Seimbang Menggunakan Algoritma Smote Dan K-nearest Neighbor," 2018.

[12] A. Kharwal, "StandardScaler in Machine Learning," Sep. 22, 2020. https://thecleverprogrammer.com/2020/09/22/standardscaler-in-machine-learning/ (accessed Sep. 28, 2022).

[13] J. Hale, "Scale, Standardize, or Normalize with Scikit-Learn," Mar. 04, 2019. https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02 (accessed Jul. 10, 2022).

e-ISSN: 2460-870X | p-ISSN: 2085-1367

[14] N. Lusty *et al.*, "Model Regresi Logistik Untuk Melihat Pengaruh Faktor Demografis, Self Efficacy, Terhadap Perilaku Mencontek," 2017.

[15] K. Grover, "Advantages and Disadvantages of Logistic Regression." https://iq.opengenus.org/advantages-and-disadvantages-of-logistic-regression/ (accessed Sep. 28, 2022).

[16] P. Pareek, "Logistic Regression: Essential Things to Know | by Praveen Pareek | DataDrivenInvestor," Sep. 02, 2021. https://medium.datadriveninvestor.com/logistic-regression-essential-things-to-know-a4fe0bb8d10a (accessed Sep. 28, 2022).

[17] Y. Adriani Tampil, H. Komalig, and Y. Langi, "Analisis Regresi Logistik Untuk Menentukan Faktor-Faktor Yang Mempengaruhi Indeks Prestasi Kumulatif (IPK) Mahasiswa FMIPA Universitas Sam Ratulangi Manado," 2017.

[18] J. Hoare, "What is a Random Forest? - Displayr." https://www.displayr.com/what-is-a-random-forest/ (accessed Sep. 29, 2022).

[19] F. Yulian Pamuji, V. Puspaning Ramadhan, and R. Artikel, "Jurnal Teknologi dan Manajemen Informatika Komparasi Algoritma Random Forest Dan Decision Tree Untuk Memprediksi Keberhasilan Immunotheraphy Info Artikel ABSTRAK," vol. 7, pp. 46–50, 2021, [Online]. Available: http://http://jurnal.unmer.ac.id/index.php/jtmi

[20] S. Rawat, "Introduction to Perceptron Model in Machine Learning," 2021. https://www.analyticssteps.com/blogs/introduction-perceptron-model-machine-learning (accessed May 23, 2022).

[21] M. Yanto, R. Sovia, and E. P. W. Mandala, "Jaringan Syaraf Tiruan Perceptron Untuk Penentuan Pola Sistem Irigasi Lahan Pertanian Di Kabupaten Pesisir Selatan Sumatra Barat".

[22] J. Brownlee, "Perceptron Algorithm for Classification in Python," Dec. 11, 2020. https://machinelearningmastery.com/perceptron-algorithm-for-classification-in-python/ (accessed Oct. 11, 2022).

[23] R. Arthana, "Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning ," Apr. 05, 2019. https://rey1024.medium.com/mengenal-accuracy-precission-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8 (accessed Jul. 10, 2022).

[24] B. Ramsay, A. Ralescu, E. van der Knaap, and S. Visa, "Confusion Matrix-based Feature Selection. Confusion Matrix-based Feature Selection," 2011. [Online]. Available: https://www.researchgate.net/publication/220833270

[25] J. Brownlee, "What is a Confusion Matrix in Machine Learning," Aug. 15, 2020. https://machinelearningmastery.com/confusion-matrix-machine-learning/ (accessed Sep. 29, 2022).

[26] S. Ekelund, "ROC curves – what are they and how are they used?," Jan. 2011. https://acutecaretesting.org/en/articles/roc-curves-what-are-they-and-how-are-they-used (accessed May 29, 2022).

[27] K. Hajian-Tilaki, "Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation," 2018. [Online]. Available: https://www.researchgate.net/publication/256453215

[28] Z. H. Hoo, J. Candlish, and D. Teare, "What is an ROC curve?," *Emergency Medicine Journal*, vol. 34, no. 6, pp. 357–359, Jun. 2017, doi: 10.1136/emermed-2017-206735.

[29] K. H. Zou, A. J. O'Malley, and L. Mauri, "Receiver-Operating Characteristic Analysis for Evaluating Diagnostic Tests and Predictive Models," *Circulation*, vol. 115, no. 5, pp. 654–657, Feb. 2007, doi: 10.1161/CIRCULATIONAHA.105.594929.

[30] A. Bhandari, "AUC-ROC Curve in Machine Learning Clearly Explained - Analytics Vidhya," Jun. 16, 2020. https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/ (accessed Sep. 29, 2022).