 ISSN:2302-8149
e-ISSN:2540-9719

Komparasi Penerapan Metode *Bagging* dan *Adaboost* pada Algoritma C4.5 untuk Prediksi Penyakit Stroke

Comparison of Bagging and Adaboost Methods on C4.5 Algorithm for Stroke Prediction

Nur Diana Saputri*, Khalid Khalid, Dwi Rolliawati
Prodi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Ampel
Jl. Veteran 5D No.25 Kecamatan Kebomas, Kabupaten Gresik, Jawa Timur, Indonesia
*e-mail: nurdianasaputrii@gmail.com

(***received*:** 27 Oktober 2021, ***revised*:** 16 November 2021, ***accepted*:** 5 Juli 2022)

Abstrak

Stroke merupakan penyakit tidak menular dan sangat berbahaya karena disebabkan oleh gangguan fungsional otak yang disebabkan oleh tersumbatnya aliran peredaran darah. Penyakit ini tergolong ke dalam *cerebrovascular disease* karena membutuhkan penanganan selama 24 jam, jika tidak ditangani secara cepat dapat menyebabkan kematian. Tujuan dari penelitian ini adalah untuk mengatasi masalah tersebut adalah membuat model prediksi berbasis machine learning untuk membantu ahli medis dalam menangani penyakit stroke untuk mengurangi risiko kematian. Metode yang diterapkan untuk penelitian ini adalah menerapkan metode klasifikasi algoritma C4.5 serta metode bagging dan Adaboost dari *Ensemble Learning*. Data stroke diolah menggunakan 2 tahap proses pengolahan data yaitu tahap data cleaning dan tahap transformasi data. Penelitian klasifikasi ini dilakukan perbandingan antara algoritma C4.5, metode bagging + algoritma C4.5 dan metode Adaboost + algortima C4.5 menggunakan metode confusion matrix, k-fold cross validation dan uji validasi berdasarkan nilai TP, TN, FP, FN, recall, precision, F1-Score dan akurasi. Hasil pengujian klasifikasi menggunakan *Confusion Matrix* dan *k-fold cross validation* untuk algoritma C4.5 menghasilkan akurasi sebesar 92.87%. Kemudian hasil akurasi dari algoritma C4.5 dengan metode bagging meningkat menjadi 95.02% dan ketika dikombinasikan dengan metode Adaboost nilai akurasinya juga meningkat menjadi 94.63%. Dari hasil akurasi tersebut dapat disimpulkan bahwa penggabungan algoritma pengklasifikasi tunggal yaitu algoritma C4.5 dengan metode bagging dan Adaboost terbukti dapat meningkatkan performa klasifikasi.

Kata kunci: Klasifikasi, Dataset Stroke Disease, *Decision Tree*, Bagging, Adaboost

Abstract

Stroke is a non-communicable disease and is very dangerous because of functional disorders of the brain caused by blockage of blood circulation. This disease is classified as a cerebrovascular disease because it requires treatment for 24 hours, if not treated quickly it can cause death. The purpose of this research is to overcome this problem is to create a machine learning-based prediction model for medical experts in dealing with diseases to help reduce the risk of death. The method applied for this research is to apply the C4.5 algorithm classification method as well as the bagging and Adaboost methods from Ensemble Learning. Stroke data is processed using 2 stages of data processing, namely the data cleaning stage and the data transformation stage. In this study, a comparison will be made between the C4.5 algorithm, the bagging method + the C4.5 algorithm and the Adaboost method + the C4.5 algorithm using the confusion matrix, k-fold cross validation and validation test based on the values of TP, TN, FP, FN, recall, precision, F1-Score and accuracy. The results of the classification test using the Confusion Matrix and k-fold cross validation for the C4.5 algorithm resulted in an accuracy of 92.87%. Then the accuracy of the C4.5 algorithm with the bagging method increased to 95.02% and when combined with the Adaboost method the accuracy value also increased to 94.63%. From these results, it can be said that a single classifier algorithm, namely the C4.5 algorithm with the bagging and Adaboost methods, has been proven to improve classification performance.

Keywords: *Classification, Stroke Disease Dataset, Decision Tree, Bagging, Adaboost*

1 Pendahuluan

Stroke adalah penyakit otak yang disebabkan oleh tekanan darah tinggi. Stroke dapat terjadi kapan saja bila terjadi perdarahan akibat tekanan intrakranial yang tinggi dan keluarnya embolus dari pembuluh darah non serebral dan jika terjadi peningkatan dapat menyebabkan hipertensi. Stroke juga dapat disebabkan oleh perubahan tekanan darah kronis yang menghalangi aliran darah, sehingga fungsi otak dapat dengan mudah terserang dan terjadi kematian sel-sel saraf di otak [1]. Penyakit ini tergolong penyakit serebrovaskular (CVD) karena terjadi secara tiba-tiba dan memerlukan penanganan yang sangat cepat [2]. Semua negara masih berjuang untuk mengatasi masalah stroke dengan keterbatasan sumber daya dan ahli saraf stroke [3]. Untuk mengatasi masalah tersebut, terdapat beberapa penelitian telah menunjukkan bahwa sistem klasifikasi dapat mengatasi masalah di bidang kesehatan seperti stroke dengan menggabungkan metode atau menggunakan metode klasifikasi tunggal [4].

Metode klasifikasi adalah metode data mining yang berfungsi sebagai pengelompokan data berdasarkan jumlah dan nama kelompoknya [5]. Metode klasifikasi merupakan metode sederhana dan populer karena sudah banyak digunakan oleh para peneliti di dunia. Metode klasifikasi memiliki beberapa algoritma yang sering digunakan seperti *Decission Tree, Naive Bayes, Support Vector Machine*, Jaringan Syaraf Tiruan atau *Neural Network* dan *kNN* [6]. Beberapa metode klasifikasi yang digunakan untuk prediksi stroke adalah *Support Vector Machine*[7], *Fuzzy Tsukamoto* [8], *Decision Tree, kNN* [9][10], *Naïve Bayes* dan *Neural Network* [11]. Berdasarkan penelitian sebelumnya, penelitian ini akan menggunakan metode klasifikasi pohon keputusan karena memiliki akurasi yang lebih unggul dibandingkan dengan metode lainnya. Oleh karena itu *Decision Tree* dipilih dalam penelitian ini sebagai metode utama.

Pohon keputusan atau *Decision Tree* adalah salah satu metode klasifikasi yang menghasilkan suatu keputusan dalam bentuk pohon. Algoritma ini membentuk model keputusan yang terdiri dari root node sebagai akar pohon yang diprioritaskan dan tidak memiliki input, selanjutnya internal node sebagai akar pohon yang memiliki input dan output sedangkan leaf node sebagai akar pohon yang menjadi output. Setiap node berisi data yang sudah dikelompokkan dengan memperhatikan variabel tujuannya [12]. Algoritma *Decision Tree* memiliki kelebihan seperti dapat memecahkan masalah overfitting, menangani nilai atribut yang hilang dan dapat meningkatkan efisiensi komputasi [13]. Tetapi dalam penelitian kansadub, hasil prediksi stroke dari metode pohon keputusan masih menghasilkan nilai yang tinggi sebesar 70 pasien diprediksi dapat terkena stroke. Oleh karena itu, hasil prediksinya tersebut kurang optimal, tetapi akurasi yang dihasilkan sangat tinggi [14]. Untuk mengatasi masalah tersebut terdapat penelitian yang menerapkan teknik *ensemble* dalam memperbaiki ketidaktepatan nilai prediksi pada algoritma C4.5. Oleh karena itu pada penelitian ini akan menerapkan teknik *ensemble* agar dapat menghasilkan hasil prediksi yang baik [15].

Teknik ensemble adalah metode yang dapat membantu meningkatkan kinerja algoritma klasifikasi tunggal dengan cara membentuk beberapa pengklasifikasi dari data latih pada saat melakukan klasifikasi dan dapat menangani ketidakseimbangan data lebih baik dari resampling [16]. Teknik *ensemble* memiliki dua metode populer yaitu bagging dan boosting. Bagging mampu membantu algoritma klasifikasi yang tidak stabil seperti *Decision Tree* yang mengalami perubahan yang besar dalam prediksi ketika dikombinasikan dengan bagging. Dengan demikian algoritma C4.5 dapat mencapai akurasi yang lebih baik daripada pengklasifikasian tunggal [17]. Sedangkan Adaboost berperan sebagai penyeimbang serta kombinasi algoritma klasifikasi yang dapat meningkatkan kinerja klasifikasi dan meminimalkan *function error* pada klasifikasi [12].

Dilihat dari permasalahan tersebut, tujuan dari penelitian ini adalah menerapkan sistem klasifikasi data mining untuk membantu ahli saraf dalam melakukan prediksi stroke dan untuk mengetahui hasil dari komparasi algoritma C4.5 dengan metode bagging dan Adaboost untuk prediksi stroke. Penelitian ini diharapkan dapat mengatasi masalah ketidaktepatan prediksi dan memastikan kinerja dari metode penelitian yang diajukan dapat memprediksi secara akurat.

2 Tinjauan Literatur

Pada bab ini membahas mengenai beberapa penelitian terdahulu yang relevan dengan penelitian ini. Beberapa penelitian tersebut dapat dijadikan sebagai bahan referensi atau dapat dijadikan sebagai perbandingan karena telah menerapkan metode atau topik penelitian yang sama dengan penelitian ini. Penelitian pertama yaitu penelitian yang berjudul mendefinisikan program perawatan rehabilitasi pasien stroke dengan menerapkan model *Neural Network* dan *Decision Tree*. Dalam penelitian tersebut menjelaskan bahwa algoritma *Decision Tree* dan *Neural Network* dapat menentukan program perawatan yang tepat dan nilai akurasi yang dihasilkan oleh algoritma *Decision Tree* lebih unggul dari algoritma *Neural Network*, Selain itu nilai sensitivitas kedua algoritma tersebut adalah sama, tetapi nilai spesifisitas dan akurasinya tidak sama [18].

Penelitian kedua yaitu prediksi stroke menggunakan data mining yang dianalisis menggunakan WEKA. Metode yang digunakan adalah algoritma C4.5 dan kNN. Berdasarkan perhitungan WEKA, algoritma C4.5 menghasilkan prediksi lebih tepat dan akurasi yang tinggi. Iterasi dari *K'* terdekat' menunjukkan bahwa nilai spesifisitas, presisi dan akurasi yang dihasilkan menurun berturut - turut setelah kenaikan *K*. Sedangkan nilai akurasi, presisi dan spesifisitas pada algoritma C4.5 telah melampaui nilai yang dihasilkan oleh algoritma kNN meskipun perbandingannya kecil. Akurasi yang dihasilkan oleh algoritma C4.5 adalah sebesar 95.42%, sedangkan algoritma k-NN sebesar 94,18%. Sehingga kedua algoritma tersebut telah terbukti dapat memprediksi stroke dengan baik [10].

Selanjutnya penelitian ketiga yaitu penerapan metode Adaboost untuk mengoptimasi penyakit stroke dengan menggunakan metode naive bayes.Metode yang digunakan adalah algoritma naive bayes yang dikombinasikan dengan metode Adaboost. Hasil akhir dari penelitian menunjukkan bahwa algoritma naive bayes tanpa Adaboost dapat menghasilkan akurasi yang tinggi sebesar 97% dan setelah dikombinasikan dengan Adaboost akurasi yang dihasilkan meningkat 1% yaitu sebesar 98%. Berdasarkan nilai akurasi tersebut, metode Adaboost telah terbukti dapat meningkatkan kinerja dari algoritma naïve bayes karena telah menghasilkan nilai akurasi yang lebih baik dan memiliki margin error yang kecil [15].

Penelitian yang keempat dengan judul increasing accuracy of C4.5 algorithm using Adaboost for classification of chronic kidney disease merupakan penelitian yang menerapkan information gain ratio dan Adaboost *ensemble* ke dataset penyakit ginjal kronis menggunakan algoritma C4.5. Penerapan information gain digunakan sebagai pemilihan atribut yang berpengaruh dalam penelitian. Terdapat 12 atribut yang dipilih dari 24 atribut di dataset penyakit ginjal kronis. Hasil akurasi dari algoritma C4.5 diperoleh sebesar 96,66%. Sedangkan nilai akurasi untuk algoritma C4.5 yang dikombinasikan dengan information gain diperoleh sebesar 97,5%. Namun ketika digabungkan dengan metode Adaboost diperoleh akurasi sebesar 98,33%. Sehingga penelitian ini telah menghasilkan nilai akurasi terbaik secara berturut – turut pada saat algoritma C4.5 dikombinasikan dengan information gain dan Adaboost [19].

Penelitian yang keenam dengan judul penerapan teknik Bagging dan reduksi fitur split pada algoritma C4.5 dalam mendiagnosis penyakit diabetes. Reduksi fitur split yang diterapkan pada penelitian ini merupakan proses preprocessing yang membagi dataset ke jumlah n. Kumpulan data dibagi menjadi 4 bagian. Untuk split 1 terdiri dari 16 fitur, split 2 terdiri dari 12 fitur, split 3 terdiri dari 8 fitur, dan split 4 terdiri dari 4 fitur. Kemudian, algoritma C4.5 diterapkan pada setiap split yang sudah dibagi sebelumnya. Hasil akurasi terbaik diperoleh dari split 3 sebesar 73,1%. dan setelah menerapkan fitur split model dan metode bagging pada algoritma C4.5 menghasilkan akurasi terbaik pada split 3 sebesar 75,1%. Dibandingkan dengan akurasi algoritma C4.5 saja, penerapan model fitur split dan bagging pada algoritma C4.5 diperoleh peningkatan akurasi sebesar 4,6% dari akurasi pertama [16].

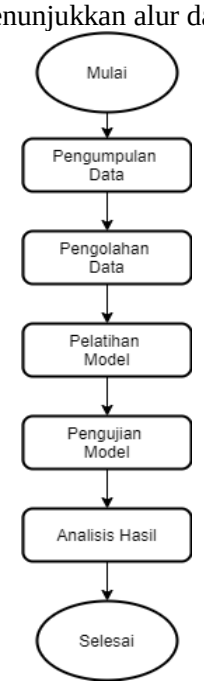
Penelitian yang terakhir berjudul optimasi algoritma C4.5 menggunakan seleksi fitur PSO dan Bagging untuk diagnosis penyakit kanker payudara. Penerapan PSO (Particle Swarm Optimization) pada diagnosa kanker payudara digunakan sebagai seleksi atribut pada dataset kanker payudara. Setelah dilakukan seleksi atribut, dataset yang semula berjumlah 9 atribut dan 1 kelas berkurang menjadi 8 atribut dan 1 kelas. Dataset kanker payudara merupakan dataset yang tidak seimbang, sehingga pada penelitian ini menerapkan metode bagging untuk mengatasi masalah tersebut. Hasil akurasi yang diperoleh ketika menerapkan PSO dan bagging pada algoritma C4.5 adalah sebesar 98,54%. Sedangkan jika hanya menerapkan algortima C4.5 tanpa PSO dan bagging menghasilkan

 ISSN:2302-8149
e-ISSN:2540-9719

akurasi sebesar 93,43%. Sehingga dapat disimpulkan bahwa adanya peningkatan akurasi sebesar 5,11% [20].

3 Metode Penelitian

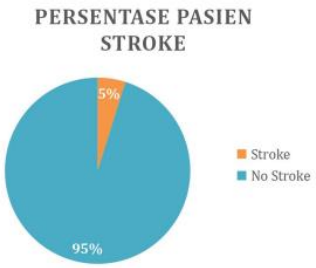
Penelitian dapat bekerja dengan baik jika langkah - langkah yang diperlukan harus konsisten dan mengikuti alur yang telah ditetapkan, tahapan penelitian ini dimulai dari perumusan masalah hingga pengujian model. Pada Gambar 1 berikut menunjukkan alur dari penelitian ini.



Gambar 1. Kerangka Penelitian

3.1 Pengumpulan Data

Dataset Stroke Disease memiliki 12 atribut dan 1 kelas dengan jumlah keseluruhan 5110 yang terdiri dari id, gender, age, hypertension, heart disease, ever married, work type, residence, avg glucose level, body massa index, smoking status dan stroke. Jumlah data pasien stroke pada dataset hanya 249 dan jumlah pasien tidak stroke sebanyak 4861. Berdasarkan jumlah data pasien dapat disimpulkan bahwa atribut stroke merupakan kelas data yang tidak seimbang. Gambar 2 menunjukkan ilustrasi perbandingan jumlah data pasien stroke. Persentase pasien yang tidak terkena stroke sebanyak 95% dan yang terkena stroke hanya 5%.



Gambar 2. Perbandingan Jumlah Stroke

3.2 Pengolahan Data

Tahap ketiga yaitu pengolahan data. Tugas utama dari tahap ini yaitu melakukan proses preprocessing data untuk mengubah data mentah menjadi data yang dapat dengan mudah diolah menggunakan proses data mining. Beberapa data masih ada yang tidak konsisten seperti “N/A” dan terdapat data outlier serta tipe datanya juga memiliki perbedaan seperti numerik dan kategorik. Pada tipe data kategorik akan diubah menjadi angka yang dikodekan menjadi 0 atau 1 agar mudah

melakukan pengklasifikasi. Proses preprocessing yang akan dijelaskan pada bab 4 yaitu data cleaning dan data transformation .

3.3 Pelatihan Model

Tahap keempat ini akan melakukan klasifikasi menggunakan algoritma C4.5 dengan metode bagging dan metode Adaboost. Hal ini dilakukan dengan membagi proses klasifikasi menjadi tiga skenario. Tabel 1 menunjukkan proses skenario klasifikasi.

Tabel 1. Proses Skenario Klasifikasi

Algoritma C4.5	Algoritma C4.5 + Bagging	Algoritma C4.5 + Adaboost
Pada skenario pertama akan dilakukan klasifikasi menggunakan algoritma C4.5 saja tanpa menerapkan metode Bagging atau Adaboost.	Pada skenario kedua akan dilakukan klasifikasi pada kombinasi algoritma C4.5 dan metode Bagging untuk mendapatkan hasil klasifikasi yang akurat.	Pada skenario terakhir akan dilakukan klasifikasi dengan menggabungkan algoritma C4.5 dan metode Adaboost untuk mendapatkan hasil klasifikasi yang akurat.

a. Algoritma C4.5

Skenario pertama adalah klasifikasi menggunakan algoritma C4.5. Pada skenario ini melakukan pelatihan model klasifikasi menggunakan algoritma C4.5 saja tanpa menerapkan metode Bagging atau Adaboost. Proses klasifikasi dilakukan di Jupyter Notebook menggunakan bahasa pemrogramman Python. Langkah - langkah yang harus dilakukan ketika membentuk pohon keputusan berdasarkan tahapan algoritma C4.5 adalah [21]:

1. Mengelompokkan atribut ke dalam setiap kelas tertentu untuk dijadikan sebagai akar/node
2. Menentukan akar yang memiliki nilai tertinggi pada nilai gain atau nilai terendah pada *index entropy* menggunakan persamaan 1 dan 2

$$Entropy\ (S) = \sum_{i=1}^{n} - pi\ log_2 pi \quad (1)$$

$$Gain\ (S, A) = entropy\ (S) - \sum_{i=1}^{n} \frac{|Si|}{S}\ x\ Entropy\ (S) \quad (2)$$

3. Ulangi proses tahap ke 2 hingga semua node terisi secara merata
4. Proses akan berhenti jika semua node telah mendapatkan kelas yang sama

b. Algoritma C4.5 + Bagging

Skenario kedua adalah klasifikasi menggunakan algoritma C4.5 dan bagging. Pada skenario ini melakukan pelatihan model klasifikasi menggunakan algoritma C4.5 dan metode Bagging dengan bahasa pemrogramman Python. Langkah - langkah dari algoritma bagging yang digunakan untuk proses klasifikasi menggunakan persamaan sebagai berikut [22]:

Input :

Data Pelatihan = D
Pengklasifikasi Dasar = L
Jumlah iterasi = T

Proses :

1. Mengambil sampel pelatihan secara acak dari data pelatihan.
2. Latih pengklassifikasi dasar h_t dari sampel pelatihan menggunakan distribusi

$$D_t h_t = (L) D_t \quad (3)$$

3. Mengambil nilai rata – rata dari semua sampel dan hitung menggunakan persamaan agregat untuk menentukan output

Output :

Hasil dari pengklasifikasi akhir dihtung menggunakan persamaan :

$$H(x) = \arg\max_y \sum_{t=1}^{T} 1\ \ (y = h_t(x)) \quad (4)$$

c. Algoritma C4.5 + Adaboost

Skenario kedua adalah klasifikasi menggunakan algoritma C4.5 dan Adaboost. Pada skenario ini melakukan pelatihan model klasifikasi menggunakan algoritma C4.5 dan metode Adaboost dengan bahasa pemrogramman Python. Cara kerja algoritma Adaboost untuk melakukan proses klasifikasi dan proses pembobotan pada setiap atribut menggunakan persamaan berikut [13]:

Input :

Data pelatihan D = {(x_1, y_1),…, (x_m, y_m)

Pengklasifikasi Dasar = L

Jumlah iterasi = T

Proses :

Inisiasi nilai bobot pada sampel pelatihan dengan persamaan :

$$D_t(i) = 1/m, \quad (5)$$

Untuk i = 1, 2, …., m

Untuk t = 1, 2, …., *T*;

Latih Pembelajar Dasar (L), ht dari sampel pelatihan menggunakan distribusi

$$D_t\, h_t = L(D_t) \quad (6)$$

Hitung error sampel pelatihan menggunakan persamaan :

$$€t = \sum D_t(i) \quad (7)$$

Kemudian hitung nilai bobot sampel pelatihan menggunakan persamaan :

$$\alpha t = \frac{1}{2} \text{In} \left(\frac{1 - €t}{€t}\right) \quad (8)$$

Setelah itu update nilai bobot sampel pelatihan menggunakan persamaan :

$$D_{t+1}(i) = \frac{Dt(i)}{Zt} \text{ x\{}^{-\alpha t} \text{ atau } ^{\alpha t} \quad (9)$$

Output :

Hasil dari pengklasifikasi akhir dihtung menggunakan persamaan :

$$H(x) = sign\,(\, h\sum_{t=1}^{T} \alpha t\ h_t(x)) \quad (10)$$

3.4 Pengujian Model

Sebelum melakukan pengujian, data dipisah terlebih dahulu menjadi data training atau latih dan data uji atau data testing. Metode untuk pengujian model menggunakan metode *Confusion Matrix* yang terdiri dari *true positive, true negative, false positive dan false negative*. Untuk mengukur kinerja suatu model pada *Confusion Matrix* dapat diketahui melalui nilai *precision, sensitivity, specificity,* dan *F-score*. Namun, untuk mengetahui nilai *accuracy* pada model dapat diukur menggunakan *k-fold cross-validation* dengan pengambilan 10 sampel data secara acak untuk mengevaluasi model. Dari beberapa sampel data tersebut akan dipilih model yang memiliki nilai optimal dengan cara mencari rata – rata dari 10 sampel tersebut untuk menghasilkan nilai akurasi.

3.5 Analisis Hasil

Tahapan analisis membahas mengenai perbandingan hasil dari setiap skenario yang sudah dijelaskan pada sub bab pelatihan model sebelumnya untuk mengetahui hasil pengujian manakah yang terbaik. Hasil pengujian yang dibandingkan adalah nilai presisi, spesifisitas, sensivisitas, f1score dan akurasi dari setiap skenario.

4 Hasil dan Pembahasan

4.1 Pengolahan Data

a. Data Cleaning

Pada tahap ini melakukan pembersihan data pada atribut BMI dan transformasi data pada atribut yang memiliki tipe data kategorikal. Atribut BMI memiliki nilai yang kosong atau tidak konsisten sebanyak 201 data. Data tersebut dikatakan sebagai data noise yaitu data yang tidak konsisten, data hilang, atau data yang tidak beraturan sehingga harus dihapus atau diganti agar tidak mempengaruhi nilai akurasi pada proses klasifikasi. Data yang tidak konsisten berupa nilai N/A yang

 ISSN:2302-8149
e-ISSN:2540-9719

akan diubah menjadi nilai rata - rata dari atribut BMI. Pada Tabel 2 dan 3 menunjukkan perbedaan detail data pada atribut BMI sebelum dan setelah dilakukan data cleaning.

Tabel 2. Atribut BMI sebelum Dilakukan Tahap Data Cleaning

Gender	Age	Hyper-tension	Heart_ Disease	…	Avg_ glucose _level	BMI	Smoking Status	Stroke
Male	67.0	0	1	…	228.69	36.6	formerly smoked	1
Female	61.0	0	0	…	202.21	N/A	never smoked	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Male	51.0	0	0	…	166.29	25.6	formerly smoked	0
Female	44.0	0	0	…	85.28	26.2	Unknown	0

Tabel 3. Atribut BMI setelah Dilakukan Tahap Data Cleaning

Gender	Age	Hyper-tension	Heart_ Disease	…	Avg_ glucose _level	BMI	Smoking Status	Stroke
Male	67.0	0	1	…	228.69	36.6	formerly smoked	1
Female	61.0	0	0	…	202.21	28.8	never smoked	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Male	51.0	0	0	…	166.29	25.6	formerly smoked	0
Female	44.0	0	0	…	85.28	26.2	Unknown	0

b. Transformation data

Sedangkan transformasi data yaitu mengubah format tipe data pada suatu atribut untuk memudahkan proses klasifikasi. Seperti pada atribut gender, ever married, work type, residence type, dan smoking status yang memiliki tipe data kategorikal. Karena python tidak dapat mengkonversi data string menjadi float, maka dilakukan konversi data terlebih dahulu dengan mengubah tipe data menjadi numerik menggunakan *LabelEncoder()* dari sklearn.preprocessing yang ada di *library* Python.Tabel 4 menunjukkan detail dari hasil transformation data.

Tabel 4. Detail Data Transformation

Atribut	Data Dikonversi
Gender	Female = 0 Male = 1
Ever married	Tidak = 0 Ya = 1
Work type	Govt_Jov = 0 Never worked = 1 Private = 2 Self employed = 3 Children = 4
Residence type	Urban = 0 Rural = 1
Smooking status	Unknown = 0 Formely_smoked = 1 Never_smoked = 2 Smokes = 3

 ISSN:2302-8149
e-ISSN:2540-9719

4.2 Pengujian Model

4.2.1 Algoritma C4.5

Pengujian model ini menerapkan metode *Confusion Matrix*, untuk mengetahui nilai dari kelas yang sebenarnya dan nilai kelas prediksi dari Dataset Stroke Disease. Pasien yang dinyatakan terkena stroke sebanyak 10, pasien dinyatakan tidak terkena stroke atau non stroke sebanyak 1430, pasien stroke yang diidentifikasi menjadi non stroke sebanyak 27 pasien, sedangkan pasien non stroke yang diidentifikasi menjadi stroke sebanyak 66 pasien. Tabel 5 menunjukkan detail dari hasil pengujian *Confusion Matrix* menggunakan algoritma C4.5

Tabel 5. Detail Hasil Pengujian Algoritma C4.5

Confusion Matrix	True Stroke (TP)	True No Stroke (TN)	Class Precision
Pred. Stroke (FP)	10	66	0.27
Pred. No Stroke (FN)	27	1430	0.96
Class Recall	0.13	0.98	
Class F1-score	0.18	0.97	

Selain melakukan perhitungan pada nilai - nilai *Confusion Matrix*. Adapun nilai akurasi yang dihitung menggunakan *k-fold cross validation* untuk menentukan akurasi mana yang memiliki nilai terbaik. Nilai akurasi yang diperoleh dari perhitungan rata–rata *10-fold cross validation* adalah sebesar 92.87%. Berikut hasil perhitungan dari keseluruhan pengujian yang ditunjukkan pada Tabel 6

Tabel 6. Hasil Perhitungan Uji Validasi Algoritma C4.5

UJI VALIDASI	Nilai
Precision	27%
Specificity	98%
Sensitivity	13%
F-score	18%
Accuracy	92.87%

4.2.2 Algoritma C4.5 + Bagging

Pengujian model kedua yaitu kombinasi algoritma C4.5 dan Bagging yang menerapkan metode *Confusion Matrix* dan cross validation untuk mengetahui nilai dari kelas yang sebenarnya dan nilai kelas prediksi dari Dataset Stroke Disease. Pasien yang dinyatakan terkena stroke sebanyak 2, pasien yang dinyatakan tidak terkena stroke atau non stroke sebanyak 1455, pasien stroke yang diidentifikasi menjadi non stroke sebanyak 2 pasien, sedangkan pasien non stroke yang diidentifikasi menjadi stroke sebanyak 74 pasien. Tabel 7 menunjukkan detail dari hasil pengujian *Confusion Matrix* menggunakan algoritma C4.5 dan Bagging.

Tabel 7. Detail Hasil *Confusion Matrix* dari Algoritma C4.5 + Bagging

Confusion Matrix	True Stroke (TP)	True No Stroke (TN)	Class Precision
Pred. Stroke (FP)	2	74	0.50
Pred. No Stroke (FN)	2	1455	0.95
Class Recall	0.03	1.00	
Class F1-score	0.05	0.97	

Selain melakukan perhitungan pada nilai - nilai *Confusion Matrix*. Adapun nilai akurasi yang dihitung menggunakan *k-fold cross validation* untuk menentukan akurasi mana yang memiliki nilai terbaik. Nilai akurasi yang diperoleh dari perhitungan rata – rata *10-fold cross validation* adalah sebesar 95.02%. Hasil perhitungan dari keseluruhan pengujian ditunjukkan pada Tabel 8

 ISSN:2302-8149
e-ISSN:2540-9719

Tabel 8. Hasil Perhitungan Uji Validasi Algoritma C4.5 + Bagging

UJI VALIDASI	Nilai
Precision	50%
Specificity	10%
Sensitivity	3%
F-score	5%
Accuracy	95,02%

4.2.3 Algoritma C4.5 + Adaboost

Pengujian model ini menerapkan metode *Confusion Matrix*, sehingga dapat diketahui nilai dari kelas yang sebenarnya dan nilai kelas prediksi dari Dataset Stroke Disease. Pasien yang dinyatakan terkena stroke sebanyak 6, pasien dinyatakan tidak terkena stroke atau non stroke sebanyak 1450, pasien stroke yang diidentifikasi menjadi non stroke sebanyak 7 pasien, sedangkan pasien non stroke yang diidentifikasi menjadi stroke sebanyak 70 pasien. Tabel 9 menunjukkan detail dari hasil pengujian *Confusion Matrix* menggunakan algoritma C4.5 dan Adaboost.

Tabel 9. Detail Hasil *Confusion Matrix* dari Algoritma C4.5 + Adaboost

Confusion Matrix	True Stroke (TP)	True No Stroke (TN)	Class Precision
Pred. Stroke (FP)	6	70	0.46
Pred. No Stroke (FN)	7	1450	0.95
Class Recall	0.08	1.00	
Class F1-score	0.13	0.97	

Selain melakukan perhitungan pada nilai - nilai *Confusion Matrix*. Adapun nilai akurasi yang dihitung menggunakan *k-fold cross validation* untuk menentukan akurasi yang terbaik. Nilai akurasi yang diperoleh dari perhitungan rata – rata *10-fold cross* validation adalah sebesar 94.63%. Hasil perhitungan dari keseluruhan pengujian ditunjukkan pada Tabel 10

Tabel 10. Hasil Perhitungan Uji Validasi Algoritma C4.5 + Adaboost

UJI VALIDASI	Nilai
Precision	46%
Specificity	100%
Sensitivity	8%
F-score	13%
Accuracy	94.6%

4.3 Analisis Hasil

Klasifikasi Dataset Stroke Disease yang menggunakan algoritma C4.5 saja memberikan nilai yang kecil untuk nilai presisi yaitu sebesar 27%, sensivisitas dan f1score pada kelas stroke. Namun berbeda dengan nilai yang hasilkan oleh spesifisitas yang menghitung kelas no stroke dan akurasi keduanya memiliki nilai yang sangat tinggi. Kemudian jika dibandingkan dengan skenario kedua yaitu algortima C4.5 dan metode Bagging, nilai presisi yang dihasilkan mengalami peningkatan sebesar 23% karena nilai presisi yang dihasilkan oleh algoritma C4.5 dan Bagging adalah 50%. Selain itu, adapun skenario ketiga yaitu algoritma C4.5 yang dikombinasikan dengan metode Adaboost memiliki peningkatan nilai presisi untuk kelas stroke sebesar 46%, dimana nilai presisinya meningkat menjadi 19%.

Sedangkan untuk nilai akurasi setelah dikombinasikan dengan metode bagging mengalami peningkatan sebanyak 3% dan ketika dikombinasikan dengan metode Adaboost mengalami peningkatan sebanyak 2%. Karena akurasi awal dari algoritma C4.5 adalah 92% dan meningkat menjadi 95% oleh metode bagging dan 94% oleh metode Adaboost pada algoritma C4.5. Dari hasil tersebut dapat disimpulkan bahwa metode bagging dan Adaboost terbukti dapat meningkatkan dan memperbaiki kinerja dari algoritma C4.5 seperti pada nilai presisi, spesifisitas dan akurasi. Namun

karena peningkatan setiap nilai tidak terlalu tinggi sehingga dapat dikatakan bahwa nilai - nilai tersebut kurang signifikan. Dilihat dari ketiga nilai tersebut penerapan metode bagging pada algoritma C4.5 terbukti lebih unggul dibandingkan metode Adaboost.

5 Kesimpulan

Penerapan metode Bagging dan Adaboost pada algoritma C4.5 digunakan untuk mengatasi masalah ketidakseimbangan kelas dan meningkatkan performa klasifikasi. Metode Bagging dan Adaboost sangat berpengaruh dalam meningkatkan nilai akurasi, presisi, dan spesitifitas pada algoritma C4.5. Berdasarkan hasil evaluasi, algoritma C4.5 menghasilkan nilai akurasi sebesar 92,87%. Sedangkan akurasi dari algoritma C4.5 setelah menerapkan metode bagging adalah 95,02%, dan setelah menerapkan metode Adaboost adalah sebesar 94,63%. Komparasi dari metode Bagging dan Adaboost pada algoritma C4.5 terbukti dapat meningkatkan dan memperbaiki kinerja klasifikasi. Nilai akurasi algoritma C4.5 meningkat sebanyak 3% dan 2% setelah dikombinasikan dengan metode bagging dan Adaboost. Terlihat dari hasil evaluasi tersebut dapat disimpulkan bahwa kedua metode tersebut masih kurang signifikan dalam meningkatkan nilai akurasi dari algoritma C4.5. Selain itu, metode bagging dan Adaboost juga tidak dapat meningkatkan nilai sensivisitas dan *f1-score* dari algortima C4.5, karena nilai kelas *false negative* yang dihasilkan masih tinggi.

Referensi

[1] N. Permatasari, "Perbandingan Stroke Non Hemoragik dengan Gangguan Motorik Pasien Memiliki Faktor Resiko Diabetes Melitus dan Hipertensi," *J. Ilm. Kesehat. Sandi Husada*, vol. 11, no. 1, pp. 298–304, 2020, doi: 10.35816/jiskh.v11i1.273.

[2] N. R. Wardhani, S. Martini, and J. Timur, "Faktor yang Berhubungan dengan Pengetahuan tentang Stroke pada Pekerja Institusi Pendidikan Tinggi," *J. Berk. Epidemiol.*, vol. 2, pp. 13–23, 2014.

[3] N. C. Suwanwela and N. Poungvarin, "Stroke Burden and Stroke Care System in Asia," *Neurol India*, vol. 64, no. 7, pp. 46–51, 2016.

[4] D. W. Nugraha, A. Y. E. Dodu, and N. Chandra, "Klasifikasi Penyakit Stroke menggunakan Metode Naive Bayes Classifier (Studi Kasus Pada Rumah Sakit Umum Daerah Undata Palu)," *semanTIK*, vol. 3, no. 2. pp. 13–22, 2017.

[5] I. Setiawati, A. P. Wibowo, and A. Hermawan, "Implementasi Decision Tree untuk Mendiagonis Penyakit Liver," *J. Inf. Syst. Manag.*, vol. 1, no. 1, pp. 13–17, 2019.

[6] Y. Pristyanto, "Penerapan Metode Ensemble untuk Meningkatkan Kinerja Algoritme Klasifikasi pada Imbalanced Dataset," *J. Teknoinfo*, vol. 13, no. 1, p. 11, 2019, doi: 10.33365/jti.v13i1.184.

[7] J. R. S and D. S. Kumar, "Stroke Prediction Using SVM," *Int. Conf. Control. Instrumentation, Commynication Comput. Technol.*, pp. 600–602, 2016.

[8] V. Adelina, D. E. Ratnawati, and M. A. Fauzi, "Klasifikasi Tingkat Risiko Penyakit Stroke menggunakan Metode GA-Fuzzy Tsukamoto," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. September, pp. 3015–3021, 2018.

[9] M. Zainuddin, K. Hidjah, and I. W. Tunjung, "Penerapan Case Based Reasoning (CBR) untuk Mendiagnosis Penyakit Stroke menggunakan Algoritma K-Nearest Neighbor," *Citesee*, pp. 21–26, 2016.

[10] L. Amini, R. Azarpazhouh, M. T. Farzadfar, S. A. Mousavi, and F. Jazaieri, "Prediction and Control of Stroke by Data Mining," *Int. J. Prev. Med.*, vol. 4, no. May 2013, pp. 245–249, 2014.

[11] I. Rohmana and R. Arifudin, "Perbandingan Jaringan Syaraf Tiruan dan Naive Bayes dalam Deteksi Seseorang Terkena Penyakit Stroke," *J. MIPA*, vol. 37, no. 2, pp. 105–114, 2014.

[12] Pareza Alam Jusia, "Analisis Komparasi Pemodelan Algoritma Decision Tree menggunakan Metode Particle Swarm Optimization dan Metode Adaboost untuk Prediksi Awal Penyakit Jantung," *Semin. Nas. Sist. Inf. 2018*, pp. 1048–1056, 2018.

[13] A. Bisri, "Penerapan Adaboost untuk Penyelesaian Ketidakseimbangan Kelas pada Penentuan Kelulusan Mahasiswa dengan Metode Decision Tree," *J. Intell. Syst.*, vol. 1, no. 1, pp. 27–32, 2015.

[14] T. Kansadub, S. Thammaboosadee, S. Kiattisin, and C. Jalayondeja, "Stroke Risk Prediction Model Based on Demographic Data," *BMEiCON 2015 - 8th Biomed. Eng. Int. Conf.*, pp. 3–5, 2016, doi: 10.1109/BMEiCON.2015.7399556.

[15] A. Byna and M. Basit, "Penerapan Metode Adaboost untuk Mengoptimasi Prediksi Penyakit Stroke dengan Algoritma Naïve Bayes," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, pp. 407–411, 2020, doi: 10.32736/sisfokom.v9i3.1023.

[16] M. Mirqotussa'adah, M. A. Muslim, E. Sugiharti, B. Prasetiyo, and S. Alimah, "Penerapan Dizcretization dan Teknik Bagging untuk Meningkatkan Akurasi Klasifikasi Berbasis Ensemble pada Algoritma C4.5 dalam Mendiagnosa Diabetes," *Lontar Komput. J. Ilm. Teknol. Inf.*, no. August, p. 135, 2017, doi: 10.24843/lkjiti.2017.v08.i02.p07.

[17] C. T. Tran, M. Zhang, P. Andreae, and B. Xue, "Bagging And Feature Selection for Classification with Incomplete Data," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10199 LNCS, pp. 471–486, 2017, doi: 10.1007/978-3-319-55849-3_31.

[18] T. Prasertsakul, P. Kaimuk, and W. Charoensuk, "Defining the Rehabilitation Treatment Programs for Stroke Patients by Applying Neural Network and Decision Trees Models," *Biomed. Eng. Int. Conf.*, 2014.

[19] A. A. Aprilia Lestari, "Increasing Accuracy of C4 . 5 Algorithm using Information Gain Ratio and Adaboost for Classification of Chronic Kidney Disease," *J. Soft Comput. Explor.*, vol. 1, no. 1, pp. 32–38, 2020.

[20] R. H. Saputra, "Optimasi Algoritma C4.5 menggunakan Seleksi Fitur Particle Swarm Optimization (PSO) dan Teknik Bagging pada Diagnosis Penyakit Kanker Payudara," *Skripsi*, 2020.

[21] W. D. Septiani, "Komparasi Metode Klasifikasi Data Mining Algoritma C4.5 Dan Naive Bayes untuk Prediksi Penyakit Hepatitis," *None*, vol. 13, no. 1, pp. 76–84, 2017, doi: 10.33480/pilar.v13i1.149.

[22] A. Saifudin, U. Pamulang, R. S. Wahono, U. Dian, and N. Semarang, "Penerapan Teknik Ensemble untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software," *J. Softw. Eng.*, vol. 1, no. 1, pp. 28–37, 2015.

[23] A. Ilham, "Komparasi Algoritma Kasifikasi dengan Pendekatan Level Data untuk menangani Data Kelas Tidak Seimbang," *J. Ilm. Ilmu Komput.*, vol. 3, no. May, 2017, doi: 10.35329/jiik.v3i1.60.