Faktor Exacta

Vol. 18, No. 2, June 2025, pp. 161~171
eISSN: 2502-339X, pISSN: 1979-276X, DOI: https://doi.org/10.30998/faktorexacta.v18i2.28959

Perbandingan Kinerja Algoritma *Random Forest, AdaBoost,* dan *Gradient Boosting* dalam Memprediksi Risiko Penyakit Hipertensi

Hafidz Muftisany[1] , Tino Feri Efendi[2], Nendy Akbar Rozaq Rais[3]
[1,2,3]Program Studi Informatika, ITB AAS Indonesia

Article Info

Article history:

Received Apr 29, 2025
Revised
Accepted

Keywords:

Hypertension
Random Forest
AdaBoost
Gradient Boosting
Confussion Matrix

ABSTRACT

Early detection of hypertension is crucial to prevent severe complications such as heart disease and stroke. Conventional diagnostic methods often require extensive medical examinations and are not easily accessible, making them less efficient for large-scale screening. This study develops and compares machine learning models—Random Forest, AdaBoost, and Gradient Boosting—to predict hypertension risk based on medical and demographic data, including blood pressure, body mass index, age, gender, and family history. A supervised learning approach was applied, and model performance was evaluated using accuracy, precision, recall, and F1-score metrics. The research process involved data preprocessing, feature selection, model training, and evaluation to determine the most effective algorithm. Furthermore, an artificial intelligence–based application was designed to support early hypertension detection and provide personalized recommendations. The results are expected to identify the most accurate and computationally efficient algorithm for predicting hypertension risk, offering an alternative solution to improve accessibility and efficiency in preventive health care.

Corresponding Author:

Hafidz Muftisany,
Program Studi Informatika,
ITB AAS Indonesia,
Jl. Slamet Riyadi No.361, Makamhaji, Kartasura, Sukoharjo, Jawa Tengah
Email: muftisany@gmail.com

1. PENDAHULUAN

Penyakit hipertensi merupakan salah satu penyebab utama morbiditas dan mortalitas di seluruh dunia. Menurut laporan WHO, sekitar 1,28 miliar orang di seluruh dunia menderita hipertensi, dengan prevalensi yang terus meningkat setiap tahunnya. [1]

Hipertensi sering terjadi pada masyarakat pada umur 31 sampai 44 tahun (31,6 %), untuk umur 45 sampai 54 tahun (45,3%), dan umur 55 sampai 64 tahun (55,2%). Dari prevalensi hipertensi sebanyak 34,1% diketahui bahwa sebesar 8,8% terdiagnosa penyakit hipertensi dan 13,3% orang yang terdiagnosis hipertensi tidak minum obat dan 32,3% tidak rutin minum obat. Hal tersbut menunjukkan sebagian besar penduduk yang menderita hipertensi tidak mengetahui bahwa dirinya hipertensi sehingga tidak mendapatkan pengobatan. [2]

Kesadaraan rendah terhadap risiko hipertensi ini yang seharusnya diperbaiki. Orang kerap memandang sepele penyakit hipertensi. Padahal jika dibiarkan akan berdampak ke kasus kesehatan yang lebih serius. Faktor risiko berperan penting terhadap kejadian hipertensi. Apabila faktor risiko diketahui maka akan lebih mudah dilakukan pencegahan. [3]

Maka menjadi penting untuk bisa membuat prediksi atas penyakit hipertensi ini di masyarakat. Salah satu metode yang bisa digunakan untuk memprediksi sebuah penyakit adalah penggunaan algoritma. Penggunaan algoritma sebagai sistem pendukung keputusan medis yang cerdas dan efektif diharapkan bisa membantu prediksi penyakit hipertensi dengan tujuan utama pencegahan. Pemilihan fitur dan *classifier* yang tepat adalah hal terpenting dalam peningkatan akurasi dan komputasi dalam prediksi penyakit bisa menggunakan beberapa algoritma. [4]

Penelitian ini dilakukan untuk membandingkan beberapa metode algoritma dalam memprediksi penyakit hipertensi. Pemilihan tiga algoritma yakni *Random Forest, AdaBoost,* dan *Gradient Boosting* didasarkan pada pertimbangan bahwa ketiganya merupakan algoritma ensemble learning yang terbukti memiliki kinerja tinggi dalam klasifikasi data medis. Random Forest dikenal mampu mengurangi risiko overfitting melalui proses pembentukan banyak pohon keputusan, sementara *AdaBoost* efektif meningkatkan akurasi dengan memberikan bobot lebih besar pada data yang sulit diklasifikasikan. Adapun *Gradient Boosting* dipilih karena kemampuannya dalam melakukan optimasi bertahap untuk memperbaiki kesalahan model sebelumnya. Dengan membandingkan ketiga algoritma ini, penelitian diharapkan dapat menentukan metode yang paling efektif dan akurat dalam memprediksi risiko hipertensi, sehingga hasilnya dapat menjadi dasar bagi pengembangan sistem deteksi dini berbasis kecerdasan buatan di bidang kesehatan.

Beberapa penelitian sebelumnya telah membahas penerapan algoritma machine learning dalam mendeteksi dan memprediksi penyakit hipertensi. Penelitian oleh Emma Rosinta Simarmata menerapkan metode Forward Chaining dan Teori Probabilitas untuk membangun sistem pakar diagnosis penyakit hipertensi. Studi ini lebih menitikberatkan pada pendekatan berbasis pengetahuan (*rule-based system*), bukan pada machine learning, sehingga kemampuan model untuk belajar dari data secara dinamis masih terbatas. [5]

Selain itu, penelitian yang dilakukan oleh Cici Emilia Sukmawati, Anis Fitri, Masruriyah, dan Ayu Ratna Juwita membandingkan efektivitas algoritma *AdaBoost* dan *XGBoost* dalam memprediksi penyakit obesitas pada populasi dewasa. Meskipun penelitian ini tidak secara langsung meneliti penyakit hipertensi, hasilnya menunjukkan bahwa metode boosting memiliki kemampuan yang baik dalam meningkatkan akurasi model prediksi terhadap data kesehatan. [6]

Berdasarkan kajian tersebut, dapat diidentifikasi adanya gap penelitian, yaitu belum adanya studi yang secara komprehensif membandingkan kinerja tiga algoritma *ensemble* populer *Random Forest, AdaBoost,* dan *Gradient Boosting* dalam konteks prediksi risiko penyakit hipertensi. Penelitian ini hadir untuk mengisi kekosongan tersebut dengan menganalisis performa ketiga algoritma pada dataset medis yang sama, sehingga dapat diketahui algoritma mana yang paling efektif dan efisien untuk deteksi dini hipertensi berbasis machine learning.

Pemilihan algoritma *Random Forest* karena Algoritma ini merupakan kumpulan pohon keputusan (*decision trees*) yang bekerja secara kolektif untuk menghasilkan prediksi yang lebih akurat. *Random Forest* memiliki keunggulan mampu menangani berbagai jenis fitur, baik numerik maupun kategorikal, yang relevan dalam data medis seperti usia, tekanan darah, pola makan, dan gaya hidup.[7]

Sementara, pemilihan algoritma *AdaBoost* karena bisa meningkatkan kinerja dengan memberikan bobot lebih tinggi pada data yang sulit diklasifikasikan. Alasan algoritma ini cocok karena fokus pada sampel yang lebih sulit (misalnya, pasien dengan gejala hipertensi tidak khas), meningkatkan akurasi deteksi pada berbagai populasi pasien.[8]

Perlu ada satu lagi penilaian dengan algoritma *Gradient Boosting* karena pada beberapa penelitian sebelumnya, prediksi sebuah penyakit hanya menggunakan satu dan dua algoritma saja. Gradient Boosting dikenal menghasilkan akurasi prediksi yang sangat tinggi, khususnya dalam tugas klasifikasi dan regresi.[9] Ini karena setiap iterasi berfokus memperbaiki kesalahan sebelumnya. Gradient Boosting efektif untuk menangani hubungan non-linear dan interaksi antar fitur yang kompleks, sehingga cocok untuk berbagai jenis masalah dunia nyata, termasuk prediksi risiko penyakit seperti hipertensi.

Penelitian ini bertujuan untuk mengevaluasi kinerja algoritma-algoritma tersebut dalam memprediksi risiko hipertensi, sehingga dapat memberikan panduan bagi pengembangan sistem pendukung keputusan di sektor kesehatan. Penelitian ini memiliki keunikan dengan menggunakan dataset yang beragam, meliputi faktor medis dan demografis, yang belum banyak diterapkan dalam studi sebelumnya.

2. METODE

2.1 Dataset

Dataset adalah sekumpulan data yang tersusun secara terstruktur di dalam memori, menyerupai sistem penyimpanan dalam basis data. Dataset terdiri dari koleksi tabel data yang saling berhubungan serta mengandung atribut yang merepresentasikan karakteristik dari setiap objek di dalamnya. [10] Dalam statistik, dataset umumnya diperoleh melalui pengamatan langsung atau pengambilan sampel dari populasi tertentu. Selain itu, dataset juga dapat digunakan dalam pengujian perangkat lunak, terutama dalam pengembangan dan

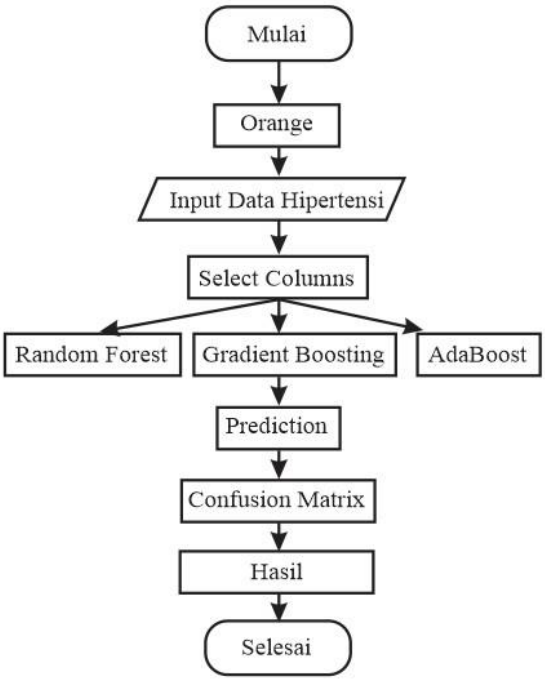
evaluasi algoritma. Dataset berperan penting dalam berbagai bidang, termasuk kecerdasan buatan, analisis data, dan pembelajaran mesin, karena menjadi sumber informasi yang dapat diolah untuk menghasilkan wawasan atau prediksi yang lebih akurat.

Data klinikal yang digunakan sejumlah 1008 diambil dari Kaggle (https://www.kaggle.com/datasets/aanya08/hypertension-data) yang terdiri dari enam atribut yakni pendidikan, umur, BMI, status merokok, prevalensi hipertensi dan heart rate.

Berdasarkan wawancara dengan Direktur Klinik Pratama BSMI Klaten dr Fath Arina Fahma, ada beberapa faktor risiko penyakit hipertensi. Pertama, karena faktor genetis. Menurutnya ada risiko dua kali lebih besar seseorang bisa memiliki penyakit hipertensi jika ada keluarga terdekat yang punya riwayat darah tinggi.

Kedua, kelebihan berat badan atau obesitas. Obesitas bisa diukur dengan *Body Mass Index* (BMI). Jika seseorang memiliki BMI di atas 25, maka risiko menderita hipertensinya juga turut menjadi besar. Ketiga kebiasan merokok. Seseorang yang punya kebiasaan merokok bisa memicu terjadinya penyakit hipertensi. Bahkan menurut dr Arin, kebiasaan merokok adalah faktor risiko terbesar terjadinya hipertensi dibandingkan penyebab yang lain.

Selanjutnya adalah konsumsi garam yang berlebihan. Menurut dr Arin, kandungan natrium di garam bisa menyebabkan masuknya cairan ke pembuluh darah sehingga meningkatkan volume darah dan akhirnya menimbulkan tekanan darah yang lebih tinggi. Batas ambang konsumsi garam adalah tidak lebih dari 10 gram garam setiap harinya.Terakhir, jika seseorang mengonsumsi obat-obatan tertentu maka bisa meningkatkan risiko hipertensi lebih tinggi. Seperti Pil KB dan beberapa obat antidepresan.



Gambar 1. Diagram Alir penelitian

2.2 Software Orange

Pada penelitian ini, perangkat lunak **Orange** digunakan untuk memproses data dan membangun model prediksi penyakit hipertensi. Langkah-langkah penggunaannya meliputi: (1) mengimpor dataset dalam format CSV ke dalam Orange, (2) melakukan *data preprocessing* seperti pembersihan data, penghapusan nilai kosong, dan normalisasi atribut, (3) melakukan *feature selection* untuk menentukan variabel yang paling relevan terhadap hipertensi, (4) memilih algoritma pembelajaran seperti Random Forest, AdaBoost, dan Gradient Boosting dari *widget* yang tersedia, (5) melatih model menggunakan data yang telah disiapkan, dan (6) membandingkan hasil performa antar model menggunakan *Test & Score* serta *Confusion Matrix* untuk menentukan algoritma terbaik berdasarkan akurasi, presisi, dan recall.[11]

2.3 Random Forest

Dalam penelitian ini, algoritma Random Forest diterapkan melalui beberapa tahapan. Pertama, dataset dibagi menjadi data latih dan data uji. Kedua, Orange secara otomatis membentuk sejumlah pohon keputusan *(decision trees)* berdasarkan sampel acak dari data latih dan subset fitur yang berbeda di setiap node. [12]

Ketiga, setiap pohon memberikan hasil prediksi independen terhadap risiko hipertensi. [13] Keempat, hasil akhir ditentukan melalui mekanisme *majority voting*, di mana kategori dengan suara terbanyak menjadi prediksi akhir. Proses ini memungkinkan evaluasi terhadap kestabilan model dan mencegah *overfitting* pada data pelatihan.

2.4 Gradient Boosting

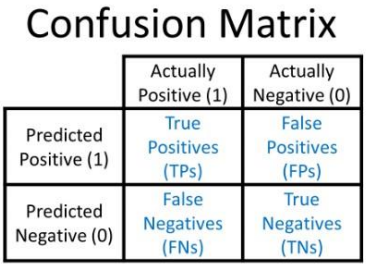
Penerapan *Gradient Boosting* dilakukan dengan melatih model secara bertahap. Model pertama dibangun untuk memprediksi data awal, kemudian model berikutnya dilatih untuk memperbaiki kesalahan dari model sebelumnya dengan menyesuaikan bobot pada data yang salah diprediksi. Proses pembelajaran ini terus berulang hingga jumlah iterasi yang ditentukan tercapai. Parameter seperti *learning rate* dan jumlah *trees* diatur untuk mencapai keseimbangan antara akurasi dan waktu komputasi. Setelah semua model terbentuk, hasil prediksi akhir diperoleh dari kombinasi berbobot seluruh model yang telah dilatih. [14]

2.5 AdaBoost

Langkah-langkah penerapan *AdaBoost* dalam penelitian ini dimulai dengan memberikan bobot awal yang sama pada seluruh sampel data. Setelah model pertama dilatih, Orange menyesuaikan bobot sampel: data yang salah diklasifikasikan diberi bobot lebih besar agar menjadi fokus pada iterasi berikutnya. [15] Proses ini berulang hingga jumlah iterasi maksimal tercapai atau ketika peningkatan akurasi sudah tidak signifikan. Model akhir dibentuk melalui kombinasi *weighted voting* dari seluruh *weak classifiers* yang dihasilkan. Evaluasi dilakukan untuk melihat peningkatan performa dibandingkan metode lain dalam memprediksi risiko hipertensi.[16]

2.6 Confusion Matrix

Tahapan evaluasi menggunakan *Confusion Matrix* dilakukan setelah model dari masing-masing algoritma selesai dilatih. Orange secara otomatis menghasilkan matriks yang memperlihatkan jumlah *true positive, true negative, false positive,* dan *false negative.* Dari matriks ini dihitung nilai akurasi, presisi, recall, dan F1-score. Analisis *Confusion Matrix* membantu mengidentifikasi sejauh mana model mampu memprediksi pasien dengan hipertensi secara tepat dan seberapa sering terjadi kesalahan klasifikasi, sehingga model terbaik dapat ditentukan berdasarkan keseimbangan antara akurasi dan keandalan prediksi.[17]



Gambar 2. *Confusion Matrix*

- *True Positive (TP):* Jumlah data yang benar-benar positif dan diprediksi positif oleh model.
- *True Negative (TN):* Jumlah data yang benar-benar negatif dan diprediksi negatif.
- *False Positive (FP):* Jumlah data yang sebenarnya negatif tetapi diprediksi positif (juga disebut *Type I Error*).
- *False Negative (FN):* Jumlah data yang sebenarnya positif tetapi diprediksi negatif (juga disebut *Type II Error*).

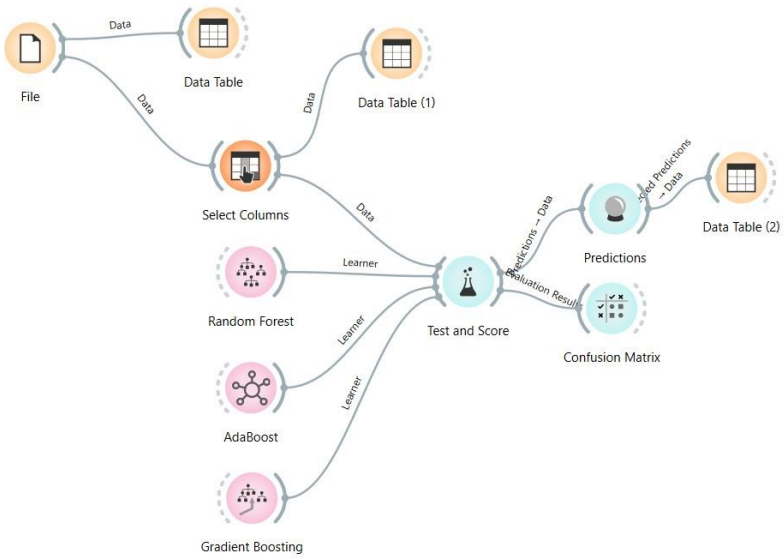
Rumus *Confusion Matrix* untuk menghitung *Accuracy, Precision, Recall* adalah sebagai berikut:

Accurany: $\frac{TP + TN}{TP + TN + FP + FN}$

Precision: $\frac{TP}{TP + FP}$

Recall: $\frac{TP}{TP + FN}$

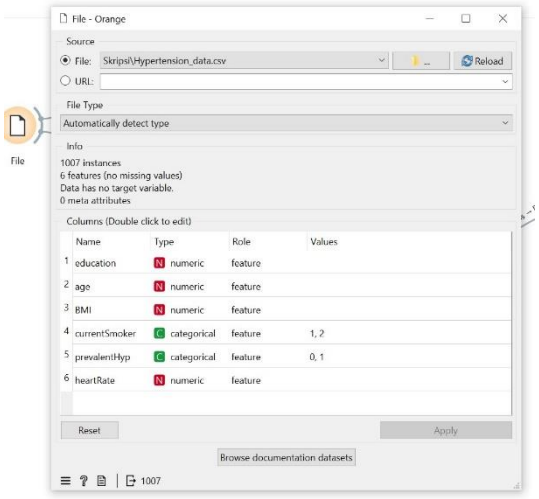
3. HASIL DAN PEMBAHASAN (10 PT)



Gambar 3. Model Penelitian

Sementara langkah-langkah penelitian akan dipaparkan satu per satu tahap dari awal hingga akhir sesuai dengan gambar model penelitian.

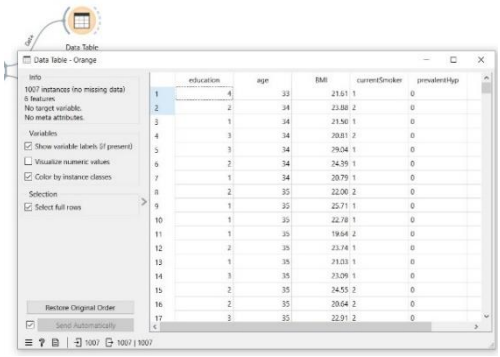
3.1. *Widget File*



Gambar 4. *Widget File*

Pada Gambar 4, melakukan input data hipertensi dari Kaggle yang sudah kita unduh dalam bentuk csv. Data yang kita unggah terdiri dari 1007 *instance* dan 6 *feature*. Dari 6 *feature* yang didapat, empat berupa data numerik dan dua berupa data *categorical.*

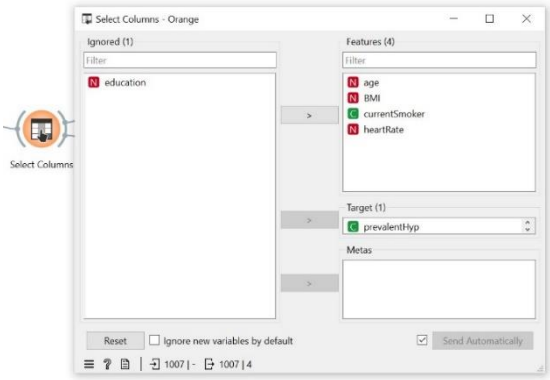
3.2. *Widget Data Table*



Gambar 5. *Widget Data Table*

Pada Gambar 5 adalah tampilan dalam bentuk tabel dari data hipertensi yang diunggah. Data ini menampilkan kolom masing-masing *feature* yakni *education, age, Body Mass Index (BMI), currentSmoker, prevalent Hypertention* dan *heart rate.*

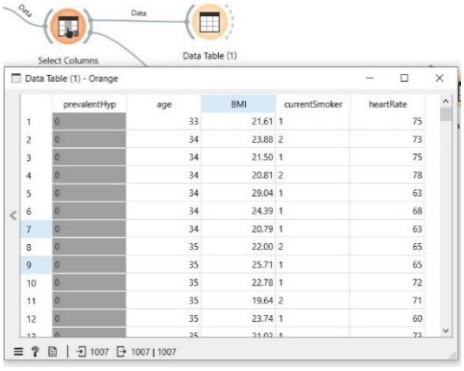
3.3. *Widget Select Column*



Gambar 6. *Widget Select Column*

Pada Gambar 6, dilakukan pemilahan data dengan *widget select column*. Pertama kita menentukan target penelitian adalah mencari prevalensi hipertensi sehingga *feature prevalenyHyp* dijadikan target. Selanjutnya kita memilah mana *feature* yang tidak terkait secara langsung dengan tingkat prevalensi hipertensi seseorang. Feature *education* adalah *feature* yang tidak terkait secara langsung dengan tingkat prediksi hipertensi seseorang. Sehingga *feature education* dipisahkan dari data (*ignored*).

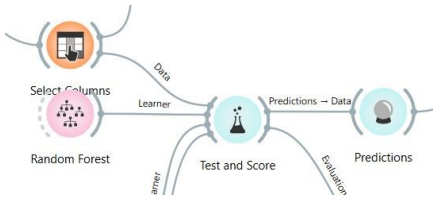
3.4. *Widget Data Table*



Gambar 7. *Widget data Table* setelah proses *Select Column*

Pada Gambar 7 dapat dilihat tampilan data terbaru dengan *widget Data Table* setelah dilakukan pengolahan data dengan *widget Select Column*. Terlihat kolom *prevalentHyp* dijadikan target dan *feature education* sudah tidak ada. Sehingga prevalensi hipertensi dalam data ini akan ditentukan oleh umur (*age*), BMI, perokok atau tidak (*current smoker*) serta detak jantung (*heart rate*).

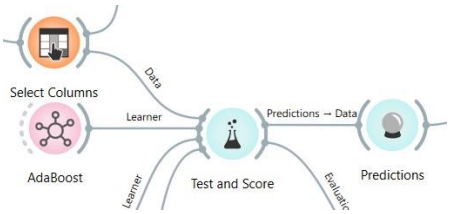
3.5. *Widget Random Forest*



Gambar 8. *Widget Random Forest*

Pada Gambar 8, dimulai proses perhitungan dengan algoritma *Random Forest*. Data yang sudah diseleksi dengan *widget Select Column* lalu dihubungkan dengan *widget Random Forest* yang sudah terhubung dengan *widget Test and Score* untuk dilakukan perhitungan lalu dihubungkan lagi dengan *widget Predictions* untuk mendapatkan hasil perhitungan dengan algoritma *Random Forest.*

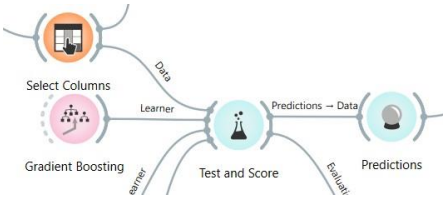
3.6. *Widget AdaBoost*



Gambar 9. *Widget AdaBoost*

Pada Gambar 9, dilakukan proses perhitungan dengan algoritma *AdaBoost*. Data hasil selesi dengan *widget Select Column* dihubungkan dengan *widget AdaBoost* yang langsung terhubung dengan *widget Test and Score* dan *widget Predictions* guna mendapatkan hasil prediksi data prevalensi hipertensi dengan algoritma *AdaBoost.*

3.6. *Widget Gradient Boosting*



Gambar 10. *Widget Gradient Boosting*

Pada gambar 10, data hasil pengolahan dengan *widget select column* kemudian dihubungkan *widget Gradient Boosting* untuk perhitungan algoritma *Gradient Boosting*. Perhitungan juga langsung dilakukan dengan menghubungkan dengan *widget Test and Score* dan *widget Predictions* untuk mendapatkan hasil prediksi prevalensi hipertensi.

3.7. *Widget Test and Score*

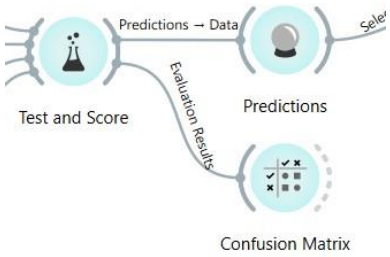
Evaluation results for target (None, show average over classes)

Model	AUC	CA	F1	Prec	Recall	MCC
Random Forest	0.687	0.708	0.691	0.688	0.708	0.257
Gradient Boosting	0.721	0.708	0.687	0.685	0.708	0.248
AdaBoost	0.589	0.646	0.648	0.650	0.646	0.176

Gambar 11. Widget Test and Score

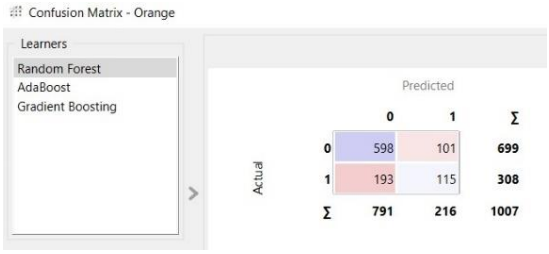
Pada gambar 11 terlihat masing-masing perhitungan dari algoritma yang digunakan yakni *Random Forest*, *Gradient Boosting* dan *AdaBoost*. Dari sumber data yang diolah, didapatkan hasil prediksi untuk algoritma *Random Forest* dengan hasil AUC (0,687), CA (0,708), F1 (0,691), *Precision* (0,688) dan *Recall* (0,708). Sementara untuk hasil prediksi dengan algoritma *Gradient Boosting* didapatkan hasil AUC (0,721), CA (0,708), F1 (0,687), *Precision* (0,685) dan *Recall* (0,708). Pada perhitungan prediksi dengan algoritma *AdaBoost* didapatkan hasil AUC (0,589), CA (0,646), F1 (0,648), *Precision* (0,650) dan *Recall* (0,646).

3.8. *Widget Confusion Matrix*



Gambar 12. *Widget Confusion Matrix*

Pada tahap selanjutnya, peneliti melakukan evaluasi dengan menggunakan *widget confusion matrix*. Hasil evaluasi dengan *confusion matrix* nanti akan dilakukan pada masing-masing perhitungan algoritma. Setelah itu hasil evaluasi akan dimasukkan ke dalam rumus perhitungan untuk mendapatkan *accurasy*, *precision* dan *recall* akhir. Hasil prediksi data hipertensi dengan *widget confusion matrix* pada masing-masing algoritma bisa dilihat pada gambar 13, 14 dan 15.



Gambar 13. Hasil evaluasi *confusion matrix* dengan algoritma *Random Forest*

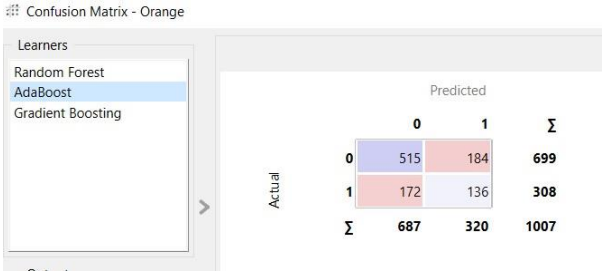
Pada gambar 13 angka 0 melambangkan *No* sementara angka 1 melambangkan *Yes*. Dalam baris *actual* 0 terdapat 699 data hipertensi dengan hasil 598 data dinyatakan tidak terkena penyakit dan 101 data dinyatakan terkena hipertensi. Sementara dari baris *actual* 1 terdapat 308 data hipertensi dengan hasil 193 data diprediksi tidak terkena hipertensi dan 115 data diprediksi terkena penyakit hipertensi.

Dari hasil ini didapatkan nilai benar positif: 598, benar negatif: 115, salah positif: 193 dan salah negatif:115 . Maka perhitungan algoritma *Random Forest* sebagai berikut:

$$Accuracy : \frac{(598+115)}{598+193+101+115} \text{ x } 100\ \% = 70{,}8\ \%$$

$Precision: \frac{598}{598+193}\ x\ 100\ \% = 75{,}44\ \%$

$Recall: \frac{598}{598+101}\ x\ 100\ \% = 85{,}55\ \%$



Gambar 14. Hasil evaluasi *confusion matrix* dengan algoritma *Ada Boost*

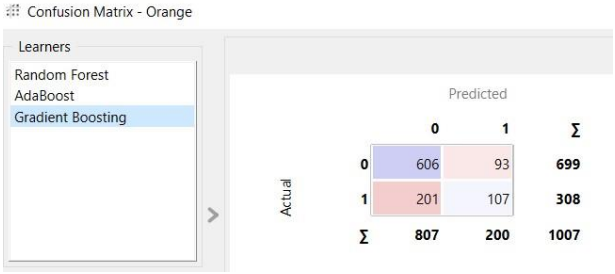
Pada gambar 14 angka 0 melambangkan *No* sementara angka 1 melambangkan *Yes*. Dalam baris *actual* 0 terdapat 699 data hipertensi dengan hasil 515 data dinyatakan tidak terkena penyakit dan 184 data dinyatakan terkena hipertensi. Sementara dari baris *actual* 1 terdapat 308 data hipertensi dengan hasil 172 data diprediksi tidak terkena hipertensi dan 136 data diprediksi terkena penyakit hipertensi.

Dari hasil ini didapatkan nilai benar positif: 515, benar negatif: 136, salah positif: 172 dan salah negatif:136 . Maka perhitungan algoritma *AdaBoost* sebagai berikut:

$Accuracy: \frac{(515+136)}{515+172+184+136}$ x 100 % = 64,64 %

$Precision: \frac{515}{515+172}\ x\ 100\ \% = 74{,}96\ \%$

$Recall: \frac{515}{515+184}\ x\ 100\ \% = 73{,}67\ \%$



Gambar 15. Hasil evaluasi *confusion matrix* dengan algoritma *Gradient Boost*

Pada gambar 15 angka 0 melambangkan *No* sementara angka 1 melambangkan *Yes*. Dalam baris *actual* 0 terdapat 699 data hipertensi dengan hasil 606 data dinyatakan tidak terkena penyakit dan 93 data dinyatakan terkena hipertensi. Sementara dari baris *actual* 1 terdapat 308 data hipertensi dengan hasil 201 data diprediksi tidak terkena hipertensi dan 107 data diprediksi terkena penyakit hipertensi.

Dari hasil ini didapatkan nilai benar positif: 606, benar negatif: 107, salah positif: 201 dan salah negatif: 93 . Maka perhitungan algoritma *Gradient Boosting* sebagai berikut:

$Accuracy: \frac{(606+107)}{606+201+93+107}$ x 100 % = 70,80 %

$Precision: \frac{606}{606+201}\ x\ 100\ \% = 75{,}09\ \%$

$Recall: \frac{606}{606+93}\ x\ 100\ \% = 86{,}69\ \%$

Tabel 1. Hasil Evaluasi

Algoritma	Accuracy	Precision	Recall
Random Forest	70,8 %	75,44 %	85,55 %
AdaBoost	64,64 %	74,96 %	73,67 %
Gradient Boosting	70,8 %	75,09 %	86,69 %

Dari pengolahan data yang sudah dilakukan dengan tiga prediksi algoritma didapatkan hasil sebagai berikut. Prediksi hipertensi menggunakan algoritma *Random Forest* didapatkan hasil benar positif: 598, benar negatif: 115, salah positif: 193 dan salah negatif:115 dengan hasil perhitungan *accuracy* 70,8 %, *precision* 75,44 % dan *recall* 85,55 %. Sementara prediksi prevalensi hipertensi dengan perhitungan algoritma *AdaBoost* didapatkan hasil benar positif: 515, benar negatif: 136, salah positif: 172 dan salah negatif:136 dengan nilai *accuracy* 64,64 %, *precision* 74,96 % dan *recall* 73,67 %. Sementara pada algoritma *Gradient Boosting* didapatkan hasil benar positif: 606, benar negatif: 107, salah positif: 201 dan salah negatif: 93 dengan *accuracy* 70,8 %, *precision* 75,09 % dan *recall* 86.69 %.

4. PENUTUP

Penelitian ini bertujuan untuk membandingkan kinerja tiga algoritma machine learning, yaitu *Random Forest, AdaBoost,* dan *Gradient Boosting,* dalam memprediksi risiko penyakit hipertensi berdasarkan data yang telah dianalisis.

Dari ketiga algoritma tersebut, *Gradient Boosting* menunjukkan performa yang paling unggul dalam memprediksi risiko hipertensi. Meskipun nilai *accuracy Gradient Boosting* sama dengan *Random Forest,* keunggulan *Gradient Boosting* terletak pada nilai *recall* yang lebih tinggi (86,69% dibandingkan 85,55% pada *Random Forest).* Dalam konteks prediksi penyakit seperti hipertensi, nilai *recall* yang tinggi sangat penting, karena recall mengukur kemampuan model untuk mengidentifikasi seluruh kasus positif (hipertensi) secara benar. Semakin tinggi *recall,* semakin sedikit kasus hipertensi yang tidak terdeteksi oleh model, sehingga secara medis lebih aman dalam mengurangi risiko salah diagnosis.

Selain itu, *precision* dari ketiga algoritma relatif seimbang, namun *Gradient Boosting* tetap memberikan precision yang cukup kompetitif dibandingkan *Random Forest* dan *AdaBoost*. Sementara itu, performa *AdaBoost* berada di bawah kedua algoritma lainnya baik dari sisi *accuracy*, *precision*, maupun *recall*, menunjukkan bahwa untuk dataset ini, *AdaBoost* kurang optimal dalam mendeteksi risiko hipertensi.

Penelitian ini menunjukkan bahwa dalam penerapan *machine learnin*g untuk bidang kesehatan, pemilihan algoritma yang tepat sangat krusial, bergantung pada karakteristik data dan tujuan aplikasi. *Gradient Boosting*, dengan keunggulan di *recall* dan *accuracy* yang solid, menjadi pilihan yang lebih direkomendasikan untuk prediksi penyakit hipertensi dalam studi ini.

Implikasi dari hasil penelitian ini adalah pentingnya menggunakan algoritma yang tidak hanya memiliki akurasi tinggi, tetapi juga mampu mendeteksi sebanyak mungkin kasus positif, terutama dalam konteks penyakit kronis yang memerlukan intervensi dini.

Untuk penelitian selanjutnya, disarankan melakukan pengujian dengan menggunakan dataset yang lebih besar, lebih beragam, dan mempertimbangkan fitur-fitur tambahan yang mungkin lebih spesifik terhadap faktor risiko hipertensi. Selain itu, eksplorasi terhadap metode balancing data (seperti SMOTE) atau tuning hyperparameter lebih lanjut pada masing-masing algoritma juga dapat dilakukan untuk meningkatkan performa model secara keseluruhan.

DAFTAR PUSTAKA

[1] A. Wulandari, S. A. Sari, and Ludiana, "Penerapan Relaksasi Benson Terhadap Tekanan Darah Pada Pasien Hipertensi Di Rsud Jendral Ahmad Yani Kota Metro Tahun 2022," *J. Cendikia Muda*, vol. 3, no. 2, pp. 163–171, 2023.

[2] K. Abdul Khalim, U. Hayati, and A. Bahtiar, "Perbandingan Prediksi Penyakit Hipertensi Menggunakan Metode Random Forest Dan Naïve Bayes," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 498–504, 2023, doi: 10.36040/jati.v7i1.6376.

[3] J. Ansar, I. Dwinata, and A. M, "Determinan Kejadian Hipertensi Pada Pengunjung Posbindu DiWilayah Kerja Puskesmas Ballaparang Kota Makassar," *J. Nas. Ilmu Kesehat.*, vol. 1, no. 3, pp. 28–35, 2019.

[4] M. Kesuma, "Prediksi Penyakit Liver Menggunakan Algoritma Random Forest," *J. Inf. dan Komput.*, vol. 11, no. 2, p. 2023, 2023.

[5] E. R. Simarmata, "Sistem Pakar Diagnosis Penyakit Hipertensi Dengan Mengunakan Metode Forward Chaining Dan Toeri Probabilitas," *Method. J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 56–64, 2021,

doi: 10.46880/mtk.v7i1.398.
[6] C. E. Sukmawati, A. Fitri, N. Masruriyah, and A. R. Juwita, "Efektivitas algoritma AdaBoost dan XGBoost pada dataset obesitas populasi dewasa," vol. 6, no. 2, pp. 101–111, 2024, doi: 10.37905/jji.
[7] Suci Amaliah, M. Nusrang, and A. Aswi, "Penerapan Metode Random Forest Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konijiwa Bantaeng," *VARIANSI J. Stat. Its Appl. Teach. Res.*, vol. 4, no. 3, pp. 121–127, 2022, doi: 10.35580/variansiunm31.
[8] T. Z. Jasman, M. A. Fadhlullah, A. L. Pratama, and R. Rismayani, "Analisis Algoritma Gradient Boosting, Adaboost dan Catboost dalam Klasifikasi Kualitas Air," *J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 2, pp. 392–402, 2022, doi: 10.28932/jutisi.v8i2.4906.
[9] S. E. Suryana, B. Warsito, and S. Suparti, "Penerapan Gradient Boosting Dengan Hyperopt Untuk Memprediksi Keberhasilan Telemarketing Bank," *J. Gaussian*, vol. 10, no. 4, pp. 617–623, 2021, doi: 10.14710/j.gauss.v10i4.31335.
[10] L. Wijaya and N. A. Pratiwi, "Penerapan Algoritma K-Means Untuk Pendataan Obat Berdasarkan Laporan Bulanan Pada Dinas Kesehatan Kabupaten Lombok Timur Pendahuluan dengan jenis dan jumlah yang mencukupi sehingga obat dapat diperoleh dengan cepat oleh rumah sakit , puskesmas , maupun ap," vol. 3, no. 2, pp. 147–156, 2020.
[11] L. Larasanti, "Pengelompokan Data Material Proyek Mv Doulos Phos Hotel Menggunakan Algoritma K-Means Clustering," 2020.
[12] C. W. Cahyana and A. Nurlayli, "Analisis Performa Logistic Regression, Naïve Bayes, dan Random Forest sebagai Algoritma Pendeteksi Kanker Payudara," *Inser. Inf. Syst. Emerg. Technol. J.*, vol. 4, no. 1, pp. 51–64, 2023.
[13] G. A. Sandag, "Prediksi Rating Aplikasi App Store Menggunakan Algoritma Random Forest," *CogITo Smart J.*, vol. 6, no. 2, pp. 167–178, 2020, doi: 10.31154/cogito.v6i2.270.167-178.
[14] I. Wardhana, Musi Ariawijaya, Vandri Ahmad Isnaini, and Rahmi Putri Wirman, "Gradient Boosting Machine, Random Forest dan Light GBM untuk Klasifikasi Kacang Kering," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 1, pp. 92–99, 2022, doi: 10.29207/resti.v6i1.3682.
[15] Y. Crismayella, N. Satyahadewi, and H. Perdana, "Algoritma Adaboost pada Metode Decision Tree untuk Klasifikasi Kelulusan Mahasiswa," *Jambura J. Math.*, vol. 5, no. 2, pp. 278–288, 2023, doi: 10.34312/jjom.v5i2.18790.
[16] 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.
[17] L. Farokhah, "Implementasi K-Nearest Neighbor Untuk Klasifikasi Bunga Dengan Ekstraksi Fitur Warna Rgb Implementation of K-Nearest Neighbor for Flower Classification With Extraction of Rgb Color Features," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 6, pp. 1129–1136, 2020, doi: 10.25126/jtiik.202072608.