

Prediksi Penyakit Hipertensi Menggunakan metode *Decision Tree* dan *Random Forest*

M. Fahrul Rizki Aditya, Nuril Lutvi Azizah dan Uce Indahyanti

Informatika, Fakultas Sains dan Teknologi Universitas Muhammdiyah Sidoarjo Kampus 2
JL. Raya Gelam No.250, Pagerwaja, Gelam, Kec.Candi, Kabupaten Sidoarjo, Jawa Timur 61271
E-mail : mochfahrul476@gmail.com, nurillutviazizah@umsida.ac.id, uceindahyanti@umsida.ac.id.

Abstrak

Hipertensi merupakan isu kesehatan masyarakat yang penting dimana jarang menyebabkan gejala nyata pada kesehatan fungsional pasien. Hipertensi merupakan faktor resiko utama pada penyakit jantung coroner, gagal jantung serta stroke. Faktor penyebab terjadinya hipertensi adalah umur, jenis kelamin, riwayat keluarga, genetic, kebiasaan merokok, obesitas, kurang aktivitas fisik, stress, dan penggunaan esterogen. Saat ini teknologi semakin berkembang. Dalam konteks perkembangan teknologi, komunitas medis diberikan kemudahan dengan adanya solusi berbasis teknologi. Salah satu di antaranya adalah program yang memanfaatkan kecerdasan buatan untuk deteksi penyakit hipertensi, dengan implementasi dalam bidang kecerdasan mesin (*machine learning*). Penelitian ini bertujuan memberikan kontribusi terhadap pengembangan aplikasi yang dapat membantu para profesional medis, terutama dokter dan rumah sakit, dalam mendiagnosis penyakit hipertensi dengan tingkat akurasi yang optimal. Metode yang digunakan dalam penelitian ini yaitu *decision tree* dan *random forest*. Tahapannya meliputi pemrosesan ulang data hingga evaluasi. Metode *decision tree* menghasilkan akurasi mencapai 100%, dan hasil yang sama juga diperoleh dari metode *random forest*. Dengan keduanya meraih akurasi maksimal, penelitian ini memperkuat kemampuan kedua metode tersebut dalam mendiagnosis serta mengelola penyakit hipertensi secara efektif.

Kata kunci : klasifikasi, prediksi, penyakit hipertensi, *decision tree*, *random forest*.

Pendahuluan

Hipertensi merupakan isu kesehatan masyarakat yang penting dimana jarang menyebabkan gejala nyata pada kesehatan fungsional pasien. Hipertensi merupakan faktor resiko utama pada penyakit jantung coroner, gagal jantung serta stroke. Faktor penyebab terjadinya hipertensi adalah umur, jenis kelamin, riwayat keluarga, genetic, kebiasaan merokok, obesitas, kurang aktivitas fisik, stress, dan penggunaan esterogen [1]. Hipertensi atau tekanan darah tinggi adalah kondisi ketika tekanan darah mencapai 140/90 mmHg atau lebih. Hipertensi merupakan gangguan pada pembuluh darah yang dapat mengakibatkan suplai oksigen dan nutrisi yang dibawa oleh darah terhambat sampai ke jaringan tubuh yang membutuhkan. Penyakit hipertensi ini dapat menjadi ancaman karena dengan tiba-tiba seseorang dapat divonis menderita darah tinggi[2]. Gejala utama yang dirasakan oleh penderita penyakit hipertensi dibagi menjadi 2 tipe, yaitu penderita hipertensi ringan dan hipertensi berat. Penderita hipertensi ringan akan merasakan sakit kepala, mimisan, nyeri dada,

dan sesak nafas. Sedangkan, penderita hipertensi berat akan merasakan nyeri pada dada, tremor otot, dan mual [3].

Saat ini teknologi semakin berkembang, komunitas medis sangat terbantu dengan perkembangan teknologi. Salah satunya adalah program yang dapat digunakan untuk mendeteksi penyakit hipertensi menggunakan kecerdasan buatan[4]. Adapun bidang kecerdasan yang dapat digunakan adalah *machine learning*. *Machine learning* merupakan cabang ilmu bagian dari kecerdasan buatan (*artificial intelligence*), dengan pemrograman untuk memungkinkan komputer menjadi cerdas berperilaku seperti manusia, dan dapat meningkatkan pemahamannya melalui pengalaman secara otomatis[5]. *Machine learning* memiliki fokus pada pengembangan sistem yang mampu belajar sendiri untuk memutuskan sesuatu tanpa harus berulang kali diprogram oleh manusia. Hal ini menjadikan mesin tidak hanya mampu berperilaku mengambil keputusan, namun juga dapat beradaptasi dengan perubahan yang terjadi. *Machine learning* menggunakan teknik untuk menangani data be-

sar (big data) dengan cara yang cerdas untuk memberikan hasil yang tepat[6]. Berdasarkan teknik pembelajarannya, tipe-tipe machine learning dapat dibedakan menjadi supervised learning, unsupervised learning, semi supervised learning dan reinforcement learning.

Decision tree adalah metode klasifikasi dan prediksi yang sangat efektif dan terkenal. Metode decision tree mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturannya mudah dimengerti dalam bahasa alami[7]. Mereka juga dapat diekspresikan dalam format basis data seperti Structure Query Language (SQL) untuk menemukan catatan dalam data tertentu. Decision tree adalah struktur yang dapat digunakan untuk membagi kumpulan data besar menjadi kumpulan data yang lebih kecil menggunakan beberapa aturan keputusan. Pada decision tree, setiap simpul daun merepresentasikan pengidentifikasi kelas. Simpul yang bukan merupakan simpul terminal terdiri dari akar dan simpul internal yang terdiri dari kondisi uji atribut untuk sejumlah record dengan atribut yang berbeda[8].

Random forest merupakan salah satu algoritma machine learning yang kuat yang menggabungkan keunggulan dari beberapa pohon keputusan untuk meningkatkan akurasi prediksi. Dalam pendekatan ini, kumpulan pohon keputusan dibuat, masing-masing dilatih pada subset acak dari dataset dan menggunakan sebagian kecil fitur. Setiap pohon membuat prediksi secara independen, dan klasifikasi akhir ditentukan dengan menggabungkan hasil keluaran mereka melalui mekanisme voting atau perataan rata-rata[9]. Metode ensemble ini berperan dalam mengurangi masalah overfitting dan meningkatkan kemampuan umum model, menjadikan random forest sebagai opsi yang kuat untuk tugas klasifikasi. Hal ini menghasilkan hasil yang lebih andal dan tepat jika dibandingkan dengan hanya menggunakan satu pohon keputusan.

Beberapa penelitian mengenai prediksi penyakit hipertensi melalui berbagai metode yang pernah dilakukan sebelumnya. Diantaranya yaitu penelitian yang dilakukan oleh Mayanda Mega Santoni, Nurul Chamidah, Nurhaffah Matondang pada tahun 2020 menyampaikan nilai presisi 94,7%, nilai recall 91,5 dan akurasi sebesar 97,7 % [10].

Penelitian yang dilakukan oleh Purwono Purwono, Pramesti Dewi, Sony Kartika Wibowo, Bala Putra Dewa pada tahun 2022 dengan nilai akurasi ketepatan prediksi yang diperoleh sebesar 85%[11].

Penelitian yang dilakukan oleh Agus Wantoro, Admi Syarif, Khairun Nisa Berawi, Kurnia Muludi, Sri Ratna Sulistiyanti, Sutyarso pada tahun 2021 dengan memperoleh nilai akurasi sebesar 96.67%[12].

Penelitian yang dilakukan oleh Salsabilla C Nurzanah, Syariful Alam, Teguh I Hermanto pada tahun 2022 dengan nilai presisi 98,9%[13].

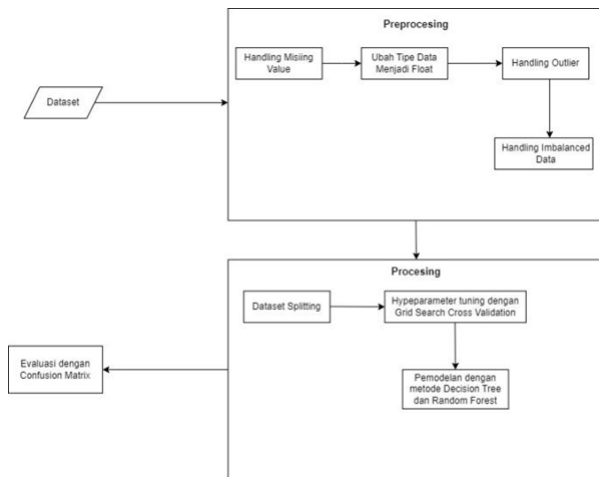
Adapun penelitian yang dilakukan oleh Supriyono, Erida, Fadila dengan pada tahun 2022 dengan

nilai akurasi sebesar 85% pada proses pelatihan dan 91,89 % pada proses pengujian[14].

Penelitian-penelitian ini mengisi kesenjangan dalam literatur sebelumnya dengan membandingkan kinerja model decision tree dan random forest terhadap penelitian sebelumnya dalam prediksi hipertensi. dengan menerapkan grid search cross validation untuk penyetelan hyperparameter, penelitian ini bertujuan untuk mengoptimalkan kinerja model dan memberikan wawasan mengenai keunggulan kedua metode dalam hal akurasi dan kemampuan prediksi. Penelitian ini berkontribusi dalam pemahaman lebih mendalam mengenai kelebihan dan kelemahan masing-masing pendekatan serta menyoroti potensi keunggulan random forest dibandingkan decision tree, yang memperlihatkan bagaimana kedua teknik ini dapat disesuaikan dengan karakteristik data tertentu guna meningkatkan akurasi prediksi penyakit hipertensi.

Metode Penelitian

Tahapan penelitian merupakan gambaran umum yang mencakup rangkaian alur penelitian yang akan dilakukan dalam pengerjaan penelitian ini dari awal hingga akhir menggunakan bahasa pemrograman python dan menggunakan software jupyter notebook. Tahapan penelitian ini dapat dipaparkan melalui diagram alir seperti yang ditunjukkan dalam Gambar 1.



Gambar 1: Tahapan Penelitian

Data

Data yang digunakan oleh peneliti terdiri dari 26083 record. Dataset ini terdiri dari 14 indeks hipertensi. Di bawah ini adalah rincian indikator atau atribut kumpulan data yang akan digunakan, lihat Tabel 1.

Tabel 1: Atribut Data Set

Atribut	Keterangan
Sex	Jenis Kelamin Pasien
Age	Umur Pasien
Cp	Tipe nyeri dada: 0: asimtomatik 1: angina tipikal 2: angina atipikal 3: nyeri non-angina
Trestbps	Tekanan darah istirahat (dalam mm Hg)
Chol	Kolesterol serum dalam mg/dl
Fbs	jika gula darah puasa pasien 120 mg/dl (1:ya; 0:tidak)
Restecg	Hasil EKG istirahat: 0: normal 1: kelainan gelombang ST-T (inversi gelombang T dan/atau ST elevasi atau depresi)
Thalach	Detak jantung maksimal tercapai
Exang	Latihan menginduksi angina (1: ya; 0: tidak ada)
Oldpeak	Depresi ST yang diinduksi oleh olahraga relatif terhadap istirahat.
Slope	Kemiringan segmen ST latihan puncak: 0: upsloping 1: flat 2: downsloping
Ca	Jumlah bejana utama (0-3) diwarnai oleh flourosopy
Thal	3: Biasa; 6: Memperbaiki cacat; 7: Cacat yang dapat dibalik
Target	Apakah pasien menderita hipertensi (1) atau tidak (0)

Tabel 1 merupakan atribut data set yang terdiri 14 atribut data yang meliputi sex, age, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, target.

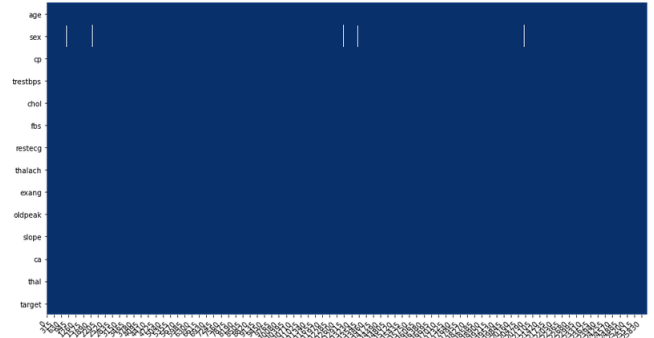
Propocising

Data mentah yang ada pada bab sebelumnya tidak dapat diproses oleh mesin secara langsung. Perlu dilakukan modifikasi pada data mentah agar menjadi data yang siap diproses oleh mesin. Adapun beberapa tahapan dari preprocessing adalah sebagai berikut.

1. Handling Missing Value

Pada tahapan awal dari preprocessing adalah dengan melakukan pengecekan missing value atau data kosong yang ada pada data mentah. Data kosong ini biasanya terjadi karena adanya kesalahan dalam input data atau memang data tersebut sengaja tidak diisi. Mesin tidak dapat memproses data yang tidak bernilai. Oleh sebab itu perlu dilakukan impute missing value atau pengisian data kosong. Terdapat beberapa cara untuk melakukan pengisian data kosong diantaranya adalah mengambil nilai rata-rata suatu kolom atau melakukan penghapusan pada kolom tersebut. Namun penghapusan merupakan opsi

terakhir apabila kolom yang terdapat data kosong tersebut adalah kolom target atau terlalu banyak data kosong yang ada pada kolom tersebut, lihat Gambar 2.



Gambar 2: Visualisasi Missing Value

2. Ubah Data Menjadi Float

Data yang digunakan pada penelitian ini memiliki tipe data float dan integer. Agar data memiliki satu tipe data yang sama, maka peneliti mengkonversi tipe data semua kolom menjadi float, lihat Gambar 3.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26083 entries, 0 to 26082
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   age         26083 non-null  float64
1   sex         26058 non-null  float64
2   cp          26083 non-null  int64
3   trestbps    26083 non-null  int64
4   chol        26083 non-null  int64
5   fbs         26083 non-null  int64
6   restecg     26083 non-null  int64
7   thalach     26083 non-null  int64
8   exang       26083 non-null  int64
9   oldpeak     26083 non-null  float64
10  slope       26083 non-null  int64
11  ca          26083 non-null  int64
12  thal        26083 non-null  int64
13  target      26083 non-null  int64
dtypes: float64(3), int64(11)
memory usage: 2.8 MB
```

Gambar 3: Deskripsi Data

3. Handling Outlier

Dalam analisis data, istilah outlier merujuk pada nilai yang secara signifikan berbeda dari nilai-nilai lain dalam dataset. Outlier dapat muncul karena kesalahan pengukuran, variasi yang ekstrem secara alami, atau situasi-situasi yang tidak biasa dalam data. Outlier dapat memberikan dampak negatif pada analisis statistik dan kinerja model prediksi karena bisa mempengaruhi perkiraan parameter dan merusak kinerja model. Salah satu teknik untuk mendeteksi outlier adalah dengan menggunakan metode Z-Score. Rumus untuk menghitung Z-Score dari suatu nilai adalah:

$$Z = \frac{x - u}{\sigma} \quad (1)$$

Di mana:

- Z adalah Z-Score dari suatu nilai x.
- x adalah nilai individu dalam dataset.
- μ adalah rata-rata dari seluruh data.
- σ adalah simpangan baku (deviasi standar) dari seluruh data.

4. Handling Imbalanced Data

Tahap berikutnya yaitu melakukan proses *handling imbalanced data* yang bertujuan untuk menyeimbangkan Jumlah Kelas dengan menggunakan pendekatan *oversampling*. Teknik oversampling digunakan untuk memperluas ukuran sampel yang kurang umum dengan menerapkan metode SMOTE (*Sintetis Minoritas Sampling Teknik*).

Processing

Setelah melalui tahap preprocessing, selanjutnya adalah melakukan tahap processing atau pemrosesan data dengan menerapkan metode decision tree dan random forest. Adapun tahapan processing pada penelitian ini adalah sebagai berikut.

1. *Dataset Splitting* Pada tahap awal pemrosesan, dataset dibagi menjadi dua bagian yaitu data latih dan data uji, dengan perbandingan sekitar 90% untuk data latih dan 10% untuk data uji. Data latih akan digunakan dalam proses pembelajaran mesin, sementara data uji akan dimanfaatkan untuk menguji model yang telah dihasilkan.
2. *Hyperparameter Tuning dengan Grid Search Cross Validation* *Grid search* merupakan salah satu metode yang sering kali digunakan untuk melakukan pencarian nilai paling optimal dari sebuah parameter dalam proses pembentukan model. Proses ini biasa disebut dengan *hyper parameter tuning*. Tujuan proses grid search beserta cross-validation adalah untuk mengidentifikasi kombinasi hyper-parameter terbaik sehingga model dapat memprediksi data dengan akurat dan optimal[15].

Klasifikasi dengan Metode Decision Tree dan Random Forest

Langkah pertama dalam tahap klasifikasi dengan metode decision tree dan random forest adalah membuka data yang diekstraksi di notebook Jupyter menggunakan library pandas. Setelah data berhasil dimuat, dilakukan pemisahan data antara data X dan data Y, dimana data X adalah kolom atribut dan data Y adalah kolom target.

Setelah dilakukan pemisahan data X dan Y, kemudian dilakukan pembagian data train dan data test pada data X menggunakan modul scikit-learn

yaitu `train_test_split` dengan prosentase 90% untuk data train dan 10% untuk data test. Untuk mendapatkan parameter yang optimal dalam hal ini, peneliti menggunakan teknik grid search cross validation. Teknik ini dapat digunakan untuk menemukan parameter optimal dari algoritma yang digunakan untuk kasus yang dianalisis.

Tahap Evaluasi

Tahapan ini digunakan untuk mengukur performa dari model machine learning yang telah dibuat. terdapat tiga metrik evaluasi yang dapat digunakan yaitu precision, recall dan confusion matrix dengan rumus sebagai berikut.

1. Precision

$$\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

Di mana:

- True Positives (TP) adalah jumlah data yang secara benar diprediksi sebagai positif oleh model.
- False Positives (FP) adalah jumlah data yang salah diprediksi sebagai positif oleh model.

2. Recall

$$\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (3)$$

Di mana:

- True Positives (TP) adalah jumlah data yang secara benar diprediksi sebagai positif oleh model.
- False Negatives (FN) adalah jumlah data yang salah diprediksi sebagai negatif oleh model.

3. Confusion Matrix

$$\begin{bmatrix} \text{TrueNegatives}(TN) & \text{FalsePositives}(FP) \\ \text{FalseNegatives}(FN) & \text{TruePositives}(TP) \end{bmatrix} \quad (4)$$

Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada confusion matrix. Dengan adanya perhitungan dari confusion matrix maka dapat diperoleh accuracy, sensitivity, dan specificity. Agar mendapatkan hasil terbaik, beberapa perbandingan data train dan data test yang ada pada tabel di atas nantinya akan dilakukan percobaan pada masing-masing perbandingan.

Untuk mendapatkan hasil terbaik, beberapa percobaan akan dilakukan dengan membandingkan data train dan data test yang tercantum dalam

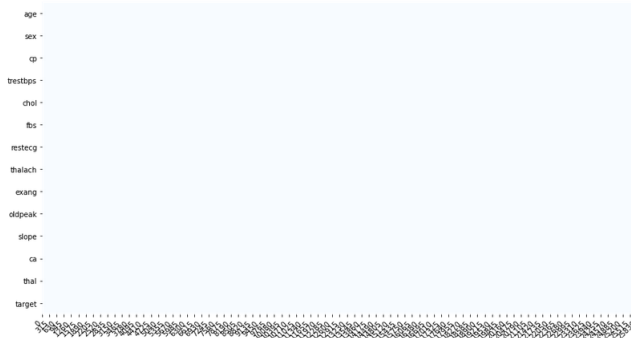
tabel di atas. Dalam penelitian ini, peneliti menggunakan randomized cross-validation dengan cross-validation 3. Jadi, kumpulan data dibagi menjadi tiga data yang sama. Jika bagian 1 menjadi data uji, bagian 2-3 menjadi data latih. Jika bagian 2 menjadi data uji, bagian 1 dan 3 menjadi data latih. Begitu seterusnya hingga bagian 3 menjadi data uji.

Hasil dan Pembahasan

Preprocessing

1. Handling Missing Value

Hasil dari tahapan ini adalah data yang telah dibersihkan dan dimodifikasi. Pada Gambar 4 terlihat bahwa data telah terisi secara keseluruhan (tidak ada missing value). Visualisasi dalam Gambar 4 menunjukkan pola warna yang seragam, menandakan bahwa missing value telah ter-impute.



Gambar 4: Deskripsi data setelah di handling missing value

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26083 entries, 0 to 26082
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   age         26083 non-null  float64
1   sex         26083 non-null  float64
2   cp          26083 non-null  float64
3   trestbps    26083 non-null  float64
4   chol        26083 non-null  float64
5   fbs         26083 non-null  float64
6   restecg     26083 non-null  float64
7   thalach     26083 non-null  float64
8   exang       26083 non-null  float64
9   oldpeak     26083 non-null  float64
10  slope       26083 non-null  float64
11  ca          26083 non-null  float64
12  thal        26083 non-null  float64
13  target      26083 non-null  float64
dtypes: float64(14)
memory usage: 2.8 MB
```

Gambar 5: Deskripsi data

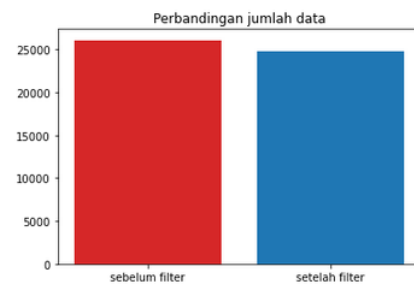
2. Ubah Data Menjadi Float

Hasil dari tahapan ini menunjukkan bahwa semua kolom memiliki jenis data yang seragam,

yaitu float. Ini terjadi karena data yang semula bersifat kategorikal telah diubah menggunakan metode label encoder, sebagaimana dijelaskan pada sub bab sebelumnya. Setelah proses encoding, seluruh kolom telah diubah menjadi tipe data float, lihat Gambar 5.

3. Handling Oulier

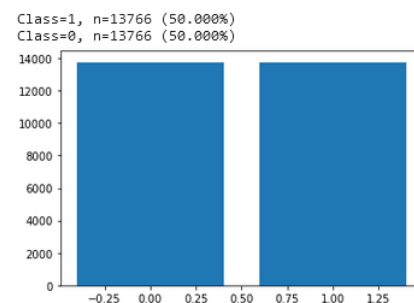
Tahapan berikutnya adalah menerapkan filter outlier menggunakan z-score seperti yang telah dijelaskan pada sub bab sebelumnya. hasil dari penerapan filter outlier ini adalah jumlah data berkurang sekitar 1300 baris. Gambar 6 menunjukkan hasil perbandingan jumlah data sebelum dan setelah filter outlier dilakukan.



Gambar 6: Perbandingan Data

4. Handling Imbalanced Data

Setelah melakukan tahap handling outlier maka tahap selanjutnya dilakukan handling imbalanced data. Tahapan ini dilakukan karena jumlah kelas data yang tidak seimbang. Peneliti menggunakan Teknik oversampling dengan SMOTE. Hasil dari tahap handling imbalanced data menggunakan Teknik oversampling dengan SMOTE dapat dilihat pada Gambar 7.



Gambar 7: hasil Handling Imbalanced Data

Procesing

Dalam bab sebelumnya telah diuraikan bahwa peneliti menggunakan metode grid search cross validation untuk mencari hyperparameter terbaik. Tabel 2 adalah hasil yang diperoleh dari penerapan metode decision tree dan random forest:

1. Decision Tree

Tabel 2: Hasil Hyperparameter Tuning Decision Tree

Hyperparameter	Value
Criterion	gini
Max_depth	none
Mean_sampels_leaf	1
Min_sampels_split	2

Dari nilai parameter terbaik yang dihasilkan melalui grid search cross validation seperti yang ditunjukkan dalam Tabel 2, tercatat bahwa skor pelatihan (train) mencapai 100%, sementara skor pengujian (test) juga berhasil mencapai 100%.

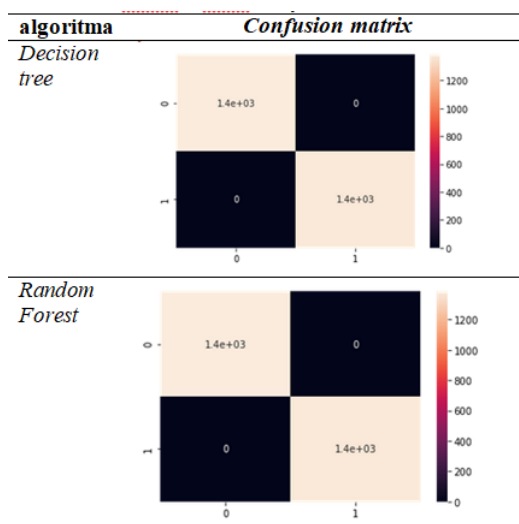
2. Random Forest

Tabel 3: Hyperparameter Tuning Random Forest

Hyperparameter	Value
Criterion	gini
Max_depth	none
Mean_sampels_leaf	1
Min_sampels_split	10
n_estimator	200

Dari nilai parameter terbaik yang dihasilkan melalui grid search cross validation seperti yang ditunjukkan dalam Tabel 3, tercatat bahwa skor pelatihan (train) mencapai 100%, sementara skor pengujian (test) juga berhasil mencapai 100%.

Tabel 4: Hasil Confusion Matrix



Evaluasi

Setelah menyelesaikan tahap-tahap procesing dan klasifikasi, tahap selanjutnya adalah evaluasi. Peneliti menjalankan tahap evaluasi ini dengan menggunakan confusion matrix.

Tabel 5: Hasil Clasification Report

algoritma	Clasification Report				
<i>Decision tree</i>	Classification Report:				
		precision	recall	f1-score	support
	0.0	1.00	1.00	1.00	1385
	1.0	1.00	1.00	1.00	1369
	accuracy			1.00	2754
	macro avg	1.00	1.00	1.00	2754
weighted avg	1.00	1.00	1.00	2754	
	Confusion Matrix:				
	[[1385 0]				
	[0 1369]]				
<i>Random Forest</i>	Classification Report:				
		precision	recall	f1-score	support
	0.0	1.00	1.00	1.00	1385
	1.0	1.00	1.00	1.00	1369
	accuracy			1.00	2754
	macro avg	1.00	1.00	1.00	2754
weighted avg	1.00	1.00	1.00	2754	
	Confusion Matrix:				
	[[1385 0]				
	[0 1369]]				

Dari Tabel 4 dan 5 yang tertera di atas, terlihat bahwa metode decision tree memberikan hasil akurasi, presisi, recall, dan F1-score sebesar 100%. Begitu juga, metode random forest juga menunjukkan hasil yang sama dengan semua nilai akurasi, presisi, recall, dan F1-score sebesar 100%.

Penutup

Melalui penggunaan model yang dibangun dalam penelitian ini, peneliti berhasil melakukan prediksi penyakit hipertensi. Dengan menerapkan metode decision tree menghasilkan nilai akurasi sebesar 100%, dan metode random forest juga menghasilkan akurasi yang sama, yaitu 100%. Penelitian ini menunjukkan bahwa menggunakan pendekatan ensemble seperti decision tree dan random forest memiliki potensi besar untuk meningkatkan akurasi prediksi dalam dunia medis. Dengan tingkat akurasi mencapai 100% pada kedua metode tersebut, hasil penelitian ini menguatkan kemampuan kedua metode tersebut dalam mendiagnosis dan mengelola penyakit hipertensi.

Daftar Pustaka

[1] B. L. Yudha, L. Muflikhah dan R. C. Wihandika, "Klasifikasi Risiko Hipertensi Menggunakan Metode Neighbor Weighted K- Nearest Neighbor (NWKNN)", J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 2, no. 2, pp. 897–904, 2018.

[2] R. E. Putri, "Implementasi Expert System Diagnosa Penyakit Hipertensi Menggunakan Metode Dempster Shafer", JATISI (Jurnal Tek. Inform. dan Sist. Informasi), vol. 9, no. 2,

- pp. 1557–1567, doi: 10.35957/jatisi.v9i2.2100, 2022.
- [3] A. C. Telaumbanua dan Y. Rahayu, “Penyuluhan Dan Edukasi Tentang Penyakit Hipertensi”, *J. Abdimas Saintika*, vol. 3, no. 1, p. 119, doi: 10.30633/jas.v3i1.1069, 2021.
- [4] A. N. Syahrudin dan T. Kurniawan, “Input dan Output pada Bahasa Pemrograman Python”, *J. Dasar Pemrograman Python STMIK, Teknik Informatika di STMIK Sumedang*. January, pp. 1–7, 2018.
- [5] M. F. Rahman, D. Alamsah, M. I. Darmawidjadja dan I. Nurma, “Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN)”, *J. Inform.*, vol. 11, no. 1, p. 36, doi: 10.26555/jifo.v11i1.a5452, 2017.
- [6] E. Retnoningsih dan R. Pramudita, “Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python”, *Bina Insa. Ict J.*, vol. 7, no. 2, p. 156, doi: 10.51211/biict.v7i2.1422, 2020.
- [7] B. Kriswantara dan R. Sadikin, “Used Car Price Prediction with Random Forest Regressor Mode”, *J. Inf. Syst. Informatics Comput. Issue Period*, vol. 6, no. 1, pp. 40–49, doi: 10.52362/jisicom.v6i1.752, 2022.
- [8] A. Muzakir dan R. A. Wulandari, “Model Data Mining sebagai Prediksi Penyakit Hipertensi Kehamilan dengan Teknik Decision Tree”, *Sci. J. Informatics*, vol. 3, no. 1, pp. 19–26, doi: 10.15294/sji.v3i1.4610, 2016.
- [9] G. A. Sandag, “Prediksi Rating Aplikasi App Store Menggunakan Algoritma Random Forest”, *CogITO Smart J.*, vol. 6, no. 2, pp. 167–178, doi: 10.31154/cogito.v6i2.270.167-178, 2020.
- [10] M. M. Santoni, N. Chamidah, and N. Matondang, “Prediksi Hipertensi menggunakan Decision Tree, Naïve Bayes dan Artificial Neural Network pada software KNIME”, *Techno.Com*, vol. 19, no. 4, pp. 353–363, doi: 10.33633/tc.v19i4.3872, 2020.
- [11] Purwono, Pramesti Dewi, Sony Kartika Wibisono dan Bala Putra Dewa, “Model Prediksi Otomatis Jenis Penyakit Hipertensi Dengan Pemanfaatan Algoritma Machine Learning Artificial Neural Network”, *Insect (Informatics Secur. J. Tek. Inform.)*, vol. 7, no. 2, pp. 82–90, 2022.
- [12] A. Wantoro, A. Syarif, K. N. Berawi, K. Muludi, S. R. Sulistiyanti dan S. Sutyarso, “Implementasi Metode Pembobotan Berbasis Aturan Dan Metode Profile Matching Pada Sistem Pakar Medis Untuk Prediksi Risiko Hipertensi”, *J. Teknoinfo*, vol. 15, no. 2, p. 134, doi: 10.33365/jti.v15i2.1523, 2021.
- [13] S. Choerunnisa Nurzanah, S. Alam dan T. Iman Hermanto, “Analisis Association Rule Untuk Identifikasi Pola Gejala Penyakit Hipertensi Menggunakan Algoritma Apriori (Studi Kasus: Klinik Rafina Medical Center)”, *JIKO (Jurnal Inform. dan Komputer)*, vol. 5, no. 2, pp. 132–141, doi: 10.33387/jiko.v5i2.4792, 2022.
- [14] F. Supriyono dan Fadila Erida, “Sistem Pakar Penegakan Diagnosa Penyakit Hipertensi dengan Inferensi Forward Chaining Menggunakan Metode Support Vector Machine (SVM)”, *Jurnal Kesehatan Medika Udayana*, vol. 08, no. 02, pp. 207–221, 2022.
- [15] Ahmad Toha, Purwono dan Windu Gata, “Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi Hyperparameter GridSearch CV”, *Buletin Ilmiah Sarjana Tek. Elektro*, vol. 4, no. 1, pp. 12–21, 2022, doi: 10.12928/biste.v4i1.6079.

Halaman ini sengaja dikosongkan.