1st National Conference on Electrical, Informatics and Industrial Technology (NEIIT-2024)
July 20, 2024

Implementasi Algoritma *Random Forest* Untuk Mendeteksi Penyakit *Multiple Sclerosis*

Ahmad Hanafi[1], Abd. Charis Fauzan[2], Fatra Nonggala Putra[3]
[123]Ilmu Komputer, Fakultas Ilmu Eksakta, Universitas Nahdlatul Ulama Blitar, Blitar
E-mail: [*1]ahmadhanafi.go.id@gmail.com, [2]abdcharis@unublitar.ac.id , [3]fnputra@unublitar.ac.id

ARTICLE INFO

Article history:

Submited:
July 10, 2024

Accepted:
July 18, 2024

Published:
August 1, 2024

ABSTRACT

Multiple sclerosis (MS) is a neurodegenerative autoimmune disease that can cause neurological effects and disorders. Based on the latest data from the 3rd edition of Atlas MS in 2020 issued by the Multiple Sclerosis International Federation, the number of sufferers of the disease (MS) globally continues to increase from 2.3 million sufferers in 2013 to 2.8 million sufferers in 2020. In addition Therefore, the increase in new cases is increasingly varied, even children under 18 years of age have increased compared to previous case findings, making (MS) considered the main representative example of an autoimmune disease. The development of increasingly sophisticated machine learning technology is often used to detect disease. In this research, the random forest algorithm was implemented to detect (MS) through several stages, such as the preprocessing stage, then the random forest algorithm classification stage by building a model using a proportion of 80%:20% comparison of training data and test data to build 100 classification models. tree using bootstrap aggregating, random feature selection, and entropy methods. The results of implementing the random forest algorithm to detect (MS) were accumulated according to the majority vote and evaluation of the confusion matrix test resulted in an accuracy value of 81%, precision of 80%, recall of 85%, specificity of 77%, and f1-score of 82%. The results of this research show that the classification model using the random forest algorithm that was developed has good performance.

Keywords:

Random Forest Algorithm, Bagging, Classification, Multiple Sclerosis

Kata Kunci:

Algoritma Random Forest, Bagging, Klasifikasi, Multiple Sclerosis

ABSTRAK

Multiple sclerosis (MS) merupakan penyakit autoimun neurodegeneratif yang dapat memberikan pengaruh dan gangguan neurologis. Berdasarkan data terbaru Atlas MS edisi ke-3 tahun 2020 yang dikeluarkan oleh Multiple Sclerosis International Federation, jumlah penderita penyakit (MS) secara global terus mengalami kenaikan dari 2,3 juta penderita pada tahun 2013 menjadi 2,8 juta penderita pada tahun 2020. Selain itu, pertambahan kasus baru semakin bervariasi, bahkan anak dibawah umur 18 tahun meningkat dibandingkan penemuan kasus sebelumya, menjadikan (MS) dianggap sebagai contoh utama yang representatif dari penyakit autoimun. Perkembangan teknologi mechine learning yang semakin canggih seringkali dimanfaatkan dalam mendeteksi suatu penyakit. Pada penelitian ini dilakukan implementasi algoritma random forest untuk mendeteksi (MS) dengan melalui beberapa tahapan, seperti tahap preprocessing, kemudian tahap klasifikasi algoritma random forest dengan pembentukan model menggunakan proporsi perbandingan data latih dan data uji sebesar 80%:20% untuk membangun 100 model klasifikasi pohon menggunakan metode bootsrap aggregating, random feature selection, dan entropy. Hasil implementasi algoritma random forest untuk mendeteksi (MS) diakumulasikan sesuai majority vote dan dilakukan evaluasi pengujian confusion matrix menghasilkan nilai accuracy sebesar 81%, precision sebesar 80%, recall sebesar 85%, specificity sebesar 77%, dan f1-score sebesar 82%. Hasil penelitian ini menunjukkan bahwa model klasifikasi menggunakan algoritma random forest yang dikembangkan memiliki peforma yang baik

Corresponding Author:
Author,
Ilmu Komputer, Fakultas Ilmu Eksakta, Universitas Nahdlatul Ulama Blitar,
Blitar
Jl. Masjid No.22, Kauman, Kec. Kepanjenkidul, Kota Blitar, Jawa Timur
66117
Email: ahmadhanafi.go.id@gmail.com

1. PENDAHULUAN

Multiple sclerosis merupakan penyakit autoimun neurodegeneratif yang dapat memberikan pengaruh dan gangguan terhadap alat gerak serta pengelihatan manusia karena berhubungan dengan sistem saraf pusat (SSP), bersifat progresif dan relaps yang menyebabkan kerusakan mielin sehingga menimbulkan lesi demielinisasi multipel [1]. Laporan tertua kasus penderita penyakit multiple sclerosis tercatat oleh seorang wanita bernama St Lidwana berasal dari sebuah kota di Belanda bernama Sciedam pada akhir abad ke-14. Pada awal abad ke-19 *kasus multiple sclerosis* baru pertama kali dapat diidentifikasi pada seorang cucu George III bernama Augustus d'Este. Pada tahun 1838 Robert Carswell pertama kali mencatat dalam sebuah buku mengenai kerusakan otak akibat penyakit *multiple sclerosis*. Jean Cruveilhier mencatat secara individu didalam sebuah buku tentang adanya *scarring* pada otak dan tulang belakang dan terbit pada tahun 1841, Jean Cruveilhier menjadi orang pertama yang mengasosiasikan bekas luka scarring dengan gejala sulit berjalan dan gemetar, meskipun tidak menyebutkan identifikasi penyebabnya secara pasti. Pada tahun 1868 Jean-Martin Charcot ahli saraf dari Prancis memperluas karya dan catatan pendahulunya dengan menyatukan dan mendeskripsikan karakeristik penyakit *multiple sclerosis* dengan mengaitkan gejala dan gangguan konduksi pada sistem saraf pusat sebagai indikator penyakit *multiple sclerosis*. Meskipun Jean-Martin Charcot menganggap penyakit *multiple sclerosis* cukup langka, setidaknya terdapat kurang lebih 40 laporan kasus yang berhasil diterbitkan. Jean-Martin Charcot melaporkan peradangan dan *scarring* pada jaringan saraf serta menggambarkan lesi *multiple sclerosis* secara rinci, deskripsinya mengenai struktur lesi masih berlaku dan karyanya dianggap sebagai tolak ukur studi awal tentang penyakit *multiple sclerosis* [2]

Berdasarkan Atlas MS edisi ke-3 tahun 2020 yang dikeluarkan oleh *Multiple Sclerosis International Federation* (MSIF), jumlah penderita penyakit ini diseluruh dunia telah meningkat dari 2,3 juta penderita pada tahun 2013 mengalami kenaikan menjadi 2,8 juta penderita pada tahun 2020. Penyakit ini umumnya diderita oleh dewasa muda atau usia produktif kisaran antara umur 18 sampai umur 50 tahun, dengan jumlah perempuan yang mengidap penyakit *multiple sclerosis* dua kali lebih banyak daripada laki-laki. Selain itu, jumlah prevalensi global penderita penyakit *multiple sclerosis* juga mengalami peningkatan dari 33 penderita per 100.000 populasi pada tahun 2013 menjadi 36 penderita per 100.000 populasi pada tahun 2020, artinya satu dari 3000 populasi orang didunia mengidap penyakit ini dan dalam setiap lima menit satu orang didiagnosis mengidap penyakit *multiple sclerosis* [3]. Berdasarkan publikasi peta persebaran penyakit *multiple sclerosis*, prevalensi Indonesia yang menepati wilayah Asia Tenggara berkisar diantara nol sampai lima penderita per 100.000 populasi, posisi ini menepati peringkat tiga dari enam kelompok negara-negara yang dikelompokkan oleh *World Health Organization* (WHO). Menurut data MSIF pada tahun 2020 terdapat 160 kasus penyakit *multiple sclerosis* di Indonesia. Sedangkan menurut catatan [4] jumlah kasus penyakit *multiple sclerosis* di Indonesia lebih besar, yakni sekitar 7.056 kasus dan diperkirakan meningkat 37,1%, menjadi peringkat satu yang tertinggi di wilayah Asia Tenggara. Di RSUPN dr. Cipto Mangunkusumo, rumah sakit rujukan nasional yang berlokasi di Jakarta, [5] mencatat data terbaru antara tahun 2015 sampai tahun 2020, terdapat 56 kasus penderita penyakit *multiple sclerosis* baru dengan jumlah penambahan setiap tahunnya berkisar satu sampai dua kasus yang dilaporkan per tahun. Sesuai laporan demografi global, pasien penyakit multiple sclerosis di Indonesia didominasi oleh perempuan, dengan rasio 4:1 dibandingkan dengan laki-laki dengan proporsi terbesar kelompok etnis jawa sebesar 36,4% diikuti keturunan Tionghoa sebesar 24,2% [6].

Dengan perkembangan zaman yang semakin maju dan perkembangan teknologi yang semakin canggih, *mechine learning* sebagai salah satu cabang artifical intellegence seringkali dimanfaatkan dalam mendeteksi, mendiagnosa dan mengidentifikasikan suatu penyakit. Prediksi penyakit *multiple sclerosis* dapat dilakukan dengan menggunakan algoritma klasifikasi untuk melakukan pengolahan dan pembelajaran dataset yang ada sebagai penentu hasil prediksi. Algoritma *random forest* merupakan salah satu *ensemble learning* istimewa yang menggunakan teknik *bootsrap aggregating* atau *bagging* dan *random feature selection* untuk meningkatkan kinerja dan kestabilan prediksi. Penggabungan beberapa model secara independen dengan teknik *bootstrap aggregating* atau yang biasa disinggkat dengan *bagging* merupakan teknik *ensemble* yang menggunakan subset data untuk membuat set pelatihan model pembelajaran dengan menggabungkan hasil dari beberapa model yang dilatih pada subset berbeda, dengan meminimalkan variasi variabel independen untuk meningkatkan prediksi klasifikasi pohon dari pohon klasifikasi tunggal sehingga dapat meningkatkan akurasi dan stabilitas model, serta mengurangi risiko *overfitting* [7]. Selain itu pada algoritma *random forest* juga diterapkan sebuah teknik *random feature selection* dalam pembelajaran mesin dan

analisis data di mana sejumlah fitur dipilih secara acak dari himpunan fitur yang lebih besar untuk digunakan dalam model pelatihan. Tujuan dari penggunaan teknik *random feature selection* ini adalah untuk mengurangi dimensi data, mengurangi risiko model yang terlalu dominan, meningkatkan variasi antara model-model individual dalam ensemble. Teknik ini sangat membantu membuat model pembelajaran mesin lebih kuat, efisien, dan mampu menghasilkan prediksi yang lebih baik [8].

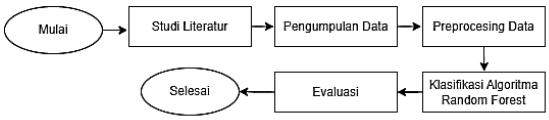
Beberapa penelitian telah dilakukan dengan menggunakan algoritma *random forest*, seperti penelitian [9] tentang klasifikasi data kesehatan mental di industri teknologi menggunakan algoritma *random forest* dengan dilakukan pembagian 3 skenario proporsi model data latih dan uji yakni 60%:40, 70%:30%, dan 80%:20% menghasilkan hasil akurasi tertinggi pada proporsi data 80%:20%, yaitu nilai akurasi sebesar 0,8412. Kemudian penelitian [10] tentang identifikasi penyakit pada daun kopi menggunakan metode *local binary pattern* dan *random forest*, pada penelitian tersebut disebutkan sitem yang dirancang memiliki performa yang baik dalam mengidentifikasi penyakit pada daun kopi berdasarkan presentasi nilai rata-rata dari parameter *Precision*, *Recall*, dan *F1-Score* yang didapatkan sebesar 96%, dengan *n-estimators* yang paling baik digunakan adalah 100 pohon dengan nilai akurasi sebesar 95,83%. Kemudian pelitian [11] tentang klasifikasi balita stunting menggunakan *random forest classifier* di kabupaten blitar memberikan hasil evaluasi terbaik, bahwa skenario terbaik adalah menggunakan proporsi 8:2 atau 80% data untuk pelatihan dan 20% data untuk pengujian, dengan akurasi tertinggi sebesar 90.1%, presisi 71.4%, dan recall 62.5% dan semua skenario memiliki performa yang baik dalam memprediksi status stunting pada balita dengan menerapkan evaluasi *confution matrix*.

Berdasarkan pemaparan diatas, maka tujuan dilakukan penelitian ini adalah untuk mengetahui proses implementasi algoritma *random forest* dalam melakukan deteksi pada penyakit *multiple sclerosis* dan diharapkan dengan dilakukan penelitian ini dapat memberikan gambaran yang lebih jelas tentang proses implementasi algoritma random forest dalam melakukan deteksi penyakit *multiple sclerosis* berdasarkan klasifikasi dataset yang digunakan. Selain itu, penelitian ini diharapkan dapat memberikan kontribusi dalam bidang *data mining* khususnya dalam mengimplementasikan algoritma *random forest* serta dapat dijadikan referensi pada penelitian selanjutnya.

2. METODE PENELITIAN

2.1. Alur Penelitian

Dalam membangun sistem klasifikasi penyakit *multiple sclerosis* dibutuhkan alur atau tahapan yang berguna sebagai landasan dalam melakukan penelitian. Alur penelitian diawali dengan dengan studi literatur, pengumpulan data, *preprocessing data*, klasifikasi dan evaluasi. *Flowchart* alur penelitian ditunjukkan pada Gambar 1 sebagai berikut.



Gambar 1. Flowchart Penelitian

2.2 Studi Literatur

Studi literatur merupakan proses yang sistematis yang melibatkan pengumpulan, analisis, dan integrasi informasi dari berbagai sumber pustaka yang berkaitan dengan penelitian yang dilakukan yang berguna sebagai pendukung dan sumber literasi dalam penelitian ilmiah, studi literatur dapat diperoleh dari berbagai sumber, termasuk jurnal, buku, dokumentasi, internet, dan pustaka yang relevan dengan topik atau variabel yang dibahas [10].

2.3 Pengumpulan Data

Pengumpulan data penelitian dilakukan melalui data sekunder yang bersifat publik, berisi data gejala yang muncul serta hasil tes pendukung kesehatan yang terkait dengan penyakit *multiple sclerosis*, dataset berjumlah 20 variabel dan terdiri dari 273 *record*, dataset diperoleh dari *Kaggle Repository* dengan link URL https://s.id/datasetmultipleslerosis. Data yang dipakai merupakan dataset pasien mestizo Meksiko yang didiagnosis dengan CIS yang dipresentasikan di Institut Nasional Neurologi dan Bedah Saraf (NINN) di Mexico City, Meksiko.

2.4 Teknik Analisis Data

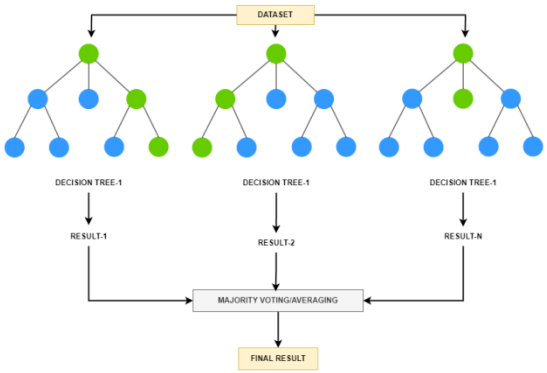
2.4.1. *Preprocessing*

Tahap *preprocessing* merupakan tahap awal pengolahan data. tahap *preprocessing* dilakukan sebelum memulai proses analisis data lebih lanjut dengan tujuan untuk memastikan bahwa data berada dalam kondisi optimal untuk

dilakukan analisis. *Preprocessing data* yang dilakukan adalah *data cleaning* dengan menghapus atribut pada kolom yang terlalu banyak nilai yang hilang, kemudian dilanjutkan pembobotan variabel dengan mendiskritisasi variabel kontinu menjadi variabel kategorik, yang semula memiliki rentang nilai yang luas dikelompokkan menjadi rentang nilai yang lebih sempit.

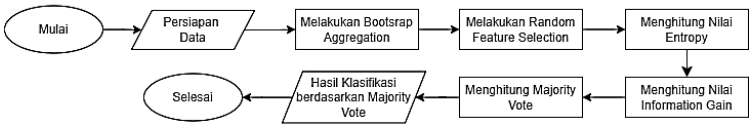
2.4.2. Klasifikasi Algoritma *Random Forest*

Setelah dilakukan preprocessing terhadap data berisi data gejala yang muncul serta hasil tes pendukung kesehatan yang terkait dengan penyakit *multiple sclerosis*, dataset siap digunakan untuk perhitungan menggunakan algoritma *random forest*. Algoritma *random forest* merupakan pengembangan dari metode *classification and regression tree* (CART), dengan menerapkan *bagging* atau metode *bootstrap aggregating* dimana setiap pohon keputusan dalam hutan dilatih pada subset acak melalui pengambilan sampel dengan *bootsrap sampling*, sehingga setiap pohon pada hutan memiliki data yang berbeda [8]. Algoritma *random forest* merupakan algoritma hasil pengembangan dari *decision tree* yang memperbaiki kekurangan pada metode sebelumya seperti *overfitting* dengan menggabungkan beberapa pohon keputusan atau *decision tree* untuk menghasilkan prediksi yang lebih stabil dan akurat [12]. *Random forest* merupakan salah satu algoritma klsasifikasi yang di implementasikan dengan membangun banyak pohon klasifikasi dengan cara membangkitkan simpul anak untuk setiap *node* atau simpul diatasnya dan dilakukan secara acak [13]. Kemudian hasil klasifikasi dari beberapa pohon keputusan atau *decision tree* yang membentuk hutan atau *forest* secara acak atau *random* akan diakumulasikan dan dipilih sesuai *majority vote* atau berdasarkan hasil klasifikasi yang paling banyak muncul, sehingga kesalahan yang dibuat oleh beberapa pohon individual dapat dikompensasi oleh pohon-pohon lain yang memberikan prediksi benar. Berikut adalah Gambar 2 merupakan ilustrasi metode *random forest* [14].



Gambar 2. Algoritma *Random Forest*

Adapun *flowchart* atau alur perhitungan menggunakan algoritma random forest dapat dilihat pada Gambar 3 sebagai berikut:



Gambar 3. *Flowchart* Algoritma *Random Forest*

Alur perhitungan Algoritma random forest sebagaimana pada Gambar 3 diatas, terdiri dari beberapa tahapan diantaranya:

a. Melakukan persiapan data atau *preprocessing data*, kemudian mengambil sampel acak dengan *bootstrap aggregating*, yakni sampel yang ukurannya sama dengan kumpulan data asli dapat dipilih lebih dari satu kali secara acak dengan penggantian.
b. Memilih variabel *independent* yang diambil secara acak m_{try} tanpa pengembalian dari semua variabel *(p)* dengan *random feature selection*. Jumlah m_{try} ditentukan melalui perhitungan $\sqrt{p}$ dengan ukuran simpul terkecil 1.
c. Menentukan *stopping criteria default,* yaitu jika didalam subnode/simpul anak hanya terdapat satu sampel makan subnode tersebut akan berhenti, sehingga menjadi terminal *node/leaf node* dan akan dihasilkan simpul terminal sebagai hasil prediksi satu pohon CART.
d. Langkah selanjutnya yakni membuat pohon menggunakan *random forest*, sebagai berikut :
   Setelah menentukan sampel berdasarkan hasil *bootsrap agrregating* kemudian menentukan variabel *root node* yaitu variabel *independent* yang terletak paling atas sebagai variabel pemisah. Kriteria pemisah untuk variabel

respon numerik pada klasifikasi menggunakan nilai *entropy* dan *information gain.* Menghitung nilai *entropy* untuk satu himpunan dataset menggunakan Persamaan 1, menghitung nilai *entropy* untuk satu atribut menggunakan Persamaan 2, sedangkan untuk menentukan nilai *information gain* dapat menggunakan Persamaan 3 berikut adalah persamaannya:

$Entropy\ (S) = \sum_{l=1}^{k} -p_l\ log_2 p_l$ ..... (1)

Dimana :

S = Himpunan dataset
k = Jumlah Kelas
p_l = Probabiltas frekuensi kelas ke-l dalam dataset

$Entropy\ (A,B) = \sum_{k \in X} P(k)E(k)$ ..... (2)

Dimana :

(A,B) = Atribut A dan atribut B
$P(k)$ = Probabilitas kelas atribut
$E(k)$ = Nilai entropy kelas atribut

$Gain\ (A) = Entropy\ (S) - \sum_{l=1}^{k} \frac{|Sl|}{|S|} \times\ Entropy\ S_l$ ..... (3)

Dimana :

S = Himpunan dataset
A = Atribut
$|\ Sl\ |$ = Jumlah sampel untuk nilai l
$|\ S\ |$ = Jumlah seluruh sampel data
$Entropy\ S_l$ = Entropy untuk sampel yang memiliki nilai l

e. Mengulangi langkah 1 sampai 3 hingga mendapatkan sejumlah pohon yang diinginkan. Sehingga mendapatkan *k* buah pohon acak.
f. Menghitung hasil akhir dengan cara menggabungkan atau diaggregasi (*aggregate)*. Untuk klasifikasi menggunakan *majority vote* atau dengan mengakumulasikan hasil klasifikasi pohon terbanyak.
g. Menetukan nilai akurasi algoritma *random forest* menggunakan nilai prediksi.

2.4.3. Evaluasi

Evaluasi sistem dalam pengujian kinerja model perhitungan algoritma random forest dalam klasifikasi penyakit *multiple sclerosis* yang digunakan adalah *confusion matrix,* yang dilakukan berdasarkan perhitungan perbandingan jumlah obyek penelitian yang diprediksi dengan benar dan salah [15]. Representasi metode evaluasi dengan menggunakan *confusion matrix* dapat dilihat pada Tabel 1 sebagai berikut.

Tabel 1. Evaluasi Confution Matrix

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

Berdasarkan Tabel 1, pengukuran kinerja menggunakan *confusion matrix* terdapat empat bagian untuk mengidentifikasi suatu prediksi, diantaranya sebagai berikut :

a. TP (*True Positive)* adalah jumlah data dengan nilai aktual positif dan nilai prediksi positif.
b. TN (*True Negative)* adalah jumlah data dengan nilai aktual positif dan nilai prediksi negatif.
c. FP (*False Positive)* adalah jumlah data dengan nilai aktual negatif dan nilai prediksi positif.
d. FN (*False Negative)* adalah jumlah data dengan nilai aktual negatif dan nilai prediksi negatif.

Selanjutnya berdasarkan nilai *confusion matrix.* Terdapat beberapa nilai evaluasi yang diperoleh melalui 5 persamaan sebagai berikut :

a. *Accuracy* adalah efektifitas keseluruhan dari hasil klasifikasi. Perhitungan nilai accuracy dengan mengambil proporsi prediksi yang benar terhadap total jumlah prediksi, dimana semakin tinggi nilai akurasi, semakin baik kinerja model dalam mengklasifikasikan semua kelas dengan benar. Menurut [16] semakin tinggi nilai akurasi dalam sebuat sistem, maka hasil prediksi model yang dilakukan akan semakin bagus. Nilai *accuracy* dihitung sesuai dengan Persamaan 4.

$$accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \text{ x } 100\% \quad \text{.....(4)}$$

b. *Precision* adalah presentase dari label data dengan label positif yang diberikan oleh klasifikasi. Perhitungan nilai

precision dengan mengambil proporsi prediksi positif yang benar terhadap total prediksi positif, dimana semakin tinggi nilai *precision*, semakin sedikit *false positives* (FP), yang berarti model lebih akurat dalam mengidentifikasi kelas positif. Nilai *precision* dihitung sesuai dengan Persamaan 5.

$$precision = \frac{TP}{(TP+FP)} \text{ x } 100\% \quad (5)$$

c. *Recall* atau *sensitivity* adalah efektifitas dari pengklasifikasi dalam mengidentifikasi label positif. Perhitungan nilai *recall* dengan mengambil proporsi prediksi positif yang benar terhadap total jumlah sebenarnya positif, dimana semakin tinggi nilai recall, semakin sedikit *false negatives* (FN), yang berarti model lebih baik dalam menangkap semua kasus positif. Nilai *recall* dihitung sesuai dengan Persamaan 6.

$$recall = \frac{TP}{(TP+FN)} \text{ x } 100\% \quad (6)$$

d. *Specificity* merupakan kemampuan model untuk mengidentifikasi kasus negatif dengan benar. Perhitungan nilai *specificity* dengan mengambil proporsi prediksi negatif yang benar terhadap total jumlah sebenarnya negatif, dimana semakin tinggi nilai *specificity*, semakin sedikit *false positives* (FP), yang berarti model lebih baik dalam mengidentifikasi kelas negatif. Nilai *specificity* dihitung sesuai dengan Persamaan 7.

$$specificity = \frac{TN}{(TN+FP)} \text{ x } 100\% \quad (7)$$

e. *F1-score* digunakan untuk menghitung rata-rata harmonik dari presisi dan *recall*, yang memberikan keseimbangan antara keduanya. Semakin tinggi nilai *F1-score*, semakin baik keseimbangan antara *precision* dan *recall*. Ini sangat berguna ketika kita memiliki ketidakseimbangan antara kelas positif dan negatif. Nilai *f1-score* dihitung sesuai dengan Persamaan 8.

$$F1\text{-}score = 2 * \frac{Precision * Recall}{Precision * Recall} \text{ x } 100\% \quad (8)$$

3. HASIL DAN PEMBAHASAN

Berdasarkan alur penelitian yang telah dijelaskan pada bab sebelumnya, maka dapat diuraikan hasil dan pembahasan penelitian seperti dibawah ini:

3.1. Studi literatur

Peneliti melakukan studi literatur dengan mengumpulkan dan mengutip literatur dari berbagai sumber pustaka yang berkaitan dengan penelitian yang dilakukan yang berguna sebagai pendukung dan sumber literasi dalam penelitian ilmiah, pada penelitian ini studi literatur diperoleh dari berbagai sumber, termasuk jurnal ilmiah, laporan ilmiah, buku, skripsi, internet, dan pustaka yang relevan dengan topik atau variabel yang terkait penelitian ini. Konsep dan teori yang dimaksud seperti penyakit *multiple sclerosis*, *data mining*, klasifikasi, algoritma *random forest*, *random feature selections, dan bootsrap aggregating* yang diperoleh melalui jurnal ilmiah maupun artikel dari internet.

3.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini diperoleh dari *Kaggle Repository* berupa dataset data gejala yang muncul serta hasil tes pendukung kesehatan yang terkait dengan penyakit multiple sclerosis, dataset berjumlah 20 variabel dan terdiri dari 273 *record*. Data yang dipakai merupakan dataset pasien mestizo Meksiko yang didiagnosis dengan CIS yang dipresentasikan di Institut Nasional Neurologi dan Bedah Saraf (NINN) di Mexico City, Meksiko. Dengan atribut label pada kolom group terbagi menjadi 2 kelas klasifikasi yakni CDMS dan non-CDMS, dimana kelas CDMS dapat diartikan sebagai kelompok data yang dikategorikan sebagai *multiple sclerosis* yang telah didiagnosisi secara klinis, sedangkan kelas non-CDMS adalah kelompok data yang dikategorikan *multiple sclerosis* yang belum dikonfirmasi diagnosis secara klinis.

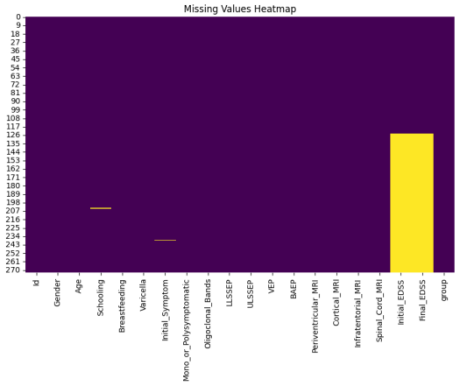
3.3. Klasifikasi Algoritma *Random Forest*

Dalam melakukan implementasi algoritma *random forest* untuk mendeteksi penyakit *multiple sclerosis* digunakan bahasa pemrograman *Pyhton* dengan menggunakan *tool* berupa *Google Collaboratory*. Dengan langkah –langkah sebagai berikut :

3.3.1. *Preprocessing Data*

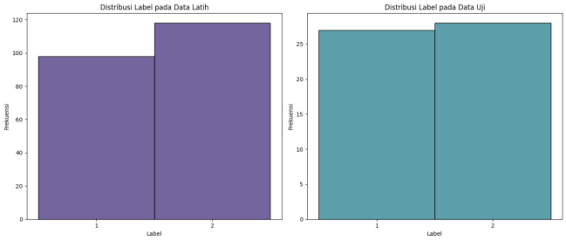
Tahap pertama *preprocessing* data dilakukan dengan melakukan *data cleaning* dengan menghapus atribut pada kolom atau baris yang terlalu banyak nilai yang hilang. Pada proses ini dilakukan *data cleaning* dari dataset asli yang semula berjumlah 20 *columns* dan 273 *rows* dilakukan pembersihan data pada kolom "Initial_EDSS" dan kolom "Final_EDSS" karena pada kedua kolom tersebut ditemukan data hilang yang diidentifikasi 148 *missing value* dan jumlah data yang hilang tersebut lebih besar dari nilai *threshold* yang ditentukan sehingga kolom tersebut dihilangkan

atau tidak dipakai dalam analisi data nantinya. Kemudian atribut “Id” juga dihilangkan karena tidak dipakai dalam klasifikasi model yang akan dibuat. Selain itu, terdapat 2 *record* data yang dilakukan *cleaning data* karena memiliki satu nilai yang hilang yakni pada kolom “Schooling” dan kolom “Initial_symptom”, dengan demikian dataset setelah dilakukan *data cleaning* menghasilkan dataset dengan jumlah 271 *rows* dan 17 *columns* yang yang siap digunakan untuk klasifikasi dan analisis data berikutnya. Tahap *preprocessing* selanjutnya yaitu dengan melakukan pembobotan dua variabbel yaitu variabel “Age” dan variabel “Schooling”, dengan mendiskritisasi variabel kontinu yang memiliki rentang nilai yang luas menjadi variabel kategorik dengan pengelompokkan rentang nilai yang lebih sempit. Untuk mengetahui *visualisasi missing value* dapat ditampilkan dengan *heatmap* agar tampilan lebih informatif dan menarik secara visual seperti dilihat pada Gambar 4.



Gambar 4. *Heatmap Missing Value*

Langkah selanjutnya yaitu menerapkan fungsi *train_test_split* yang disediakan oleh *library scikit-learn* untuk membagi dataset menjadi dua subset *training set* dan *testing set.* Pada penelitian ini proporsi dari dataset yang akan digunakan sebagai *training set* adalah 0.8 atau setara dengan 80% dari seluruh data dan 20% untuk *testing set.* Dengan *Seed* untuk pengacakan *random_state* bernilai 42, nilai ini memastikan bahwa pemisahan data konsisten setiap kali kode dijalankan. Hasil output tampilan histogram distribusi label yang sebelumya telah ditentukan proporsi sebesar 8:2 menghasilkan data latih yang berjumlah 216 dengan kelas 1 dan 2 dan distribusi label pada data uji yang berjumlah 55 dengan kelas 1 dan 2. Hasil output tampilan histogram distribusi label data latih dan uji dapat dilihat pada Gambar 6.

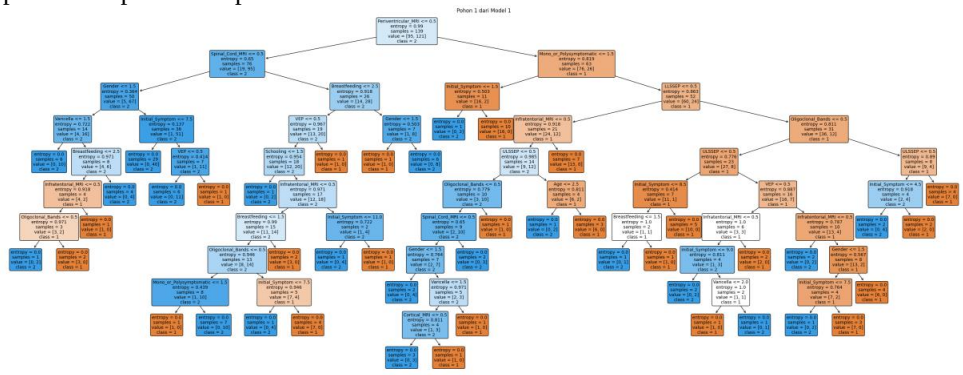


Gambar 5. Histogram Distribusi Label Data Latih Dan Uji

3.3.2. Membuat Model Klasifikasi Algoritma *Random Forest*

Pembuatan model klasifikasi dilakukan dengan membuat beberapa *instance* dari klasifikasi *random forest* dengan kondisi awal yang berbeda untuk setiap iterasinya pohon keputusan yang dibangun. Pembentukan model pohon dengan konsep perulangan sampai kondisi yang ditentukan terpenuhi. Pembuatan model klasifikasi dibentuk dengan menginisialisasi klasifikasi *random forest* dengan menggunakan fungsi pengklasifikasi *random forest* dari pustaka *scikit-learn*, kemudian mengatur jumlah pohon keputusan yang akan dibuat dalam model klasifikasi, menentukan kriteria untuk mengukur kualitas *split*, dalam hal ini menggunakan kriteria *entropy*, menentukan jumlah fitur yang dipertimbangkan untuk setiap split adalah akar kuadrat dari jumlah fitur, pengaturan untuk menggunakan teknik b*ootstrap* saat membangun pohon, dan mengatur seed atau kondisi awal yang digunakan untuk menghasilkan angka acak. Setelah dilakukan pembuatan model klasifikasi *random forest* langkah selanjutnya yaitu dengan pemilihan data subset dengan mengambil sampel acak dengan *np.random.choice:* dari *X_train* sejumlah *len(X_train)* (jumlah total data) dengan penggantian *(replace=True)*, atau data dapat dipilih lebih dari sekali. Kemudian memilih subset *X_train* dan *Y_train* yang selanjutnya akan digunakan untuk pelatihan model *random forest.* Setelah itu menambahkan model yang baru dilatih kedalam *list* model klasifikasi. Setelah dilakukan pembentukan model *random forest*, selanjutnya dilakukan visualisasi hanya untuk menampilkan 1 set pohon keputusan dari 100 pohon keputusan

yang dibentuk sesuai dengan *n-estimators yang ditentukan*. Dengan cara mengimpor modul *plot_tree* dari *sklearn.tree* dan *plt* dari *matplotlib.pyplot*, kemudian menggunakan *chosen_models* untuk mengambil list yang 1 sampel model pertama dari *ensambel random forest* yang sudah ada sebelumnya. Hasil output visualisasi yang menampilkan model pohon keputusan pertama dapat dilihat pada Gambar 7.



Gambar 6. Hasil Visualisasi Model Pohon Keputusan Pertama

Langkah selanjutnya menampilkan hasil output dari DataFrame yang berisi kelas asli, prediksi mayoritas, dan prediksi setiap model. Pada Gambar 8. Ditampilkan kelas asli, prediksi mayoritas, dan kelas hasil klasifikasi untuk sampel model pohon ke-1 sampai 24.

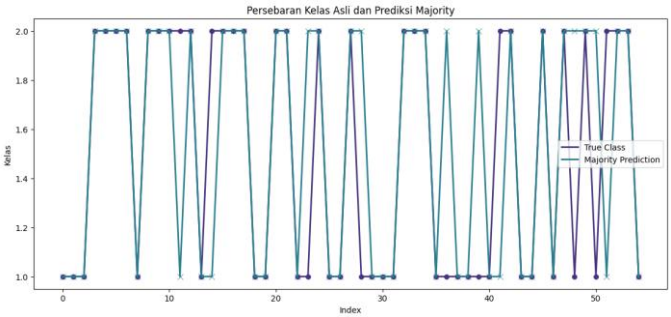
Tabel hasil Model Prediksi and Majority Predictions:

	True Class	Majority Prediction	1	2	3	4	5	6	7	8	9	10	\
30	1		1	2	2	2	1	2	2	1	1	2	
116	1		1	1	1	1	1	1	1	1	1	1	
79	1		1	1	1	1	1	1	1	1	1	1	
127	2		2	2	2	2	2	2	2	2	2	2	
196	2		2	2	2	2	2	2	2	2	2	2	
..	...		...	..	..	..	..	..	..	..	..	..	...
42	1		1	2	2	2	1	1	1	2	1	2	
185	2		2	2	2	2	2	1	1	1	2	2	
9	1		1	1	1	1	1	1	1	2	1	1	
22	1		1	1	1	1	2	1	1	1	1	1	
199	2		2	2	2	2	2	2	2	2	1	2	

	11	12	13	14	15	16	17	18	19	20	21	22	23	24	\
30	2	1	2	1	1	1	1	1	1	1	2	1	2	2	
116	1	1	1	1	1	2	2	1	1	1	1	1	1	1	
79	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
127	2	1	2	1	2	2	2	2	2	2	2	2	2	2	
196	2	2	2	2	2	2	2	2	2	2	2	2	2	1	
..	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
42	2	1	1	1	1	2	2	2	2	1	2	1	1	1	
185	2	2	2	2	2	2	2	2	2	2	2	1	2	1	
9	1	1	2	1	1	1	1	2	1	1	1	1	2	1	
22	1	2	1	1	1	1	1	1	1	1	1	1	1	1	
199	2	1	2	2	2	2	2	2	2	2	2	2	1	2	

Gambar 7. Hasil Output Dataframe Kelas Asli, Kelas Majority Dan Prediksi Pohon Ke Ke-1 Sampai 24

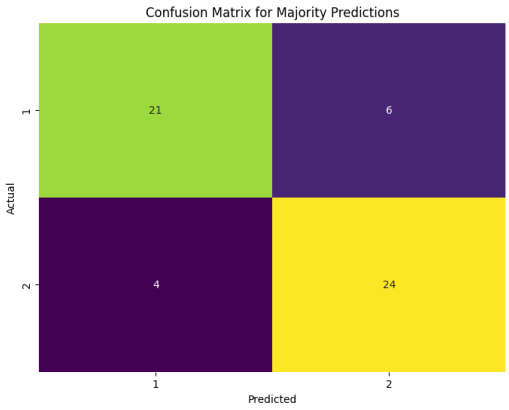
Langkah selanjutnya menampilkan *line plot* dan *scatter plot* visualisasi distribusi persebaran *plot* kelas asli ditandai dengan plot warna ungu dan kelas prediksi berdasarkan hasil *majority* dengan warna biru. Dari distribusi persebaran *plot* kelas asli dan kelas prediksi berdasarkan hasil *majority* di dihubungkan dengan *line plot* secara berurutan dengan warna yang sama dengan *scatter plot*. Dimana apabila *plot* dan *line* yang ditampilkan berhimpitan maka kelas prediksi benar dan sesuai dengan kelas asli, berbeda apabila *plot* dan *line* yang ditampilkan tidak berhimpitan maka kelas prediksi salah dan tidak sesuai dengan kelas asli.



Gambar 8. *Line Plot* Dan *Scatter Plot* Visualisasi Distribusi Persebaran Plot Kelas Asli Dan Kelas Majority

3.1.3. Evaluasi

Representasi metode evaluasi dengan menggunakan *confusion matrix* ditampilkan pada Gambar 10. Pada tampilan *heatmap* ini dapat dilihat keseluruhan data pada kolom *actual* dan *predicted* menampilkan jumlah obyek penelitian yang diprediksi berdasarkan *majority vote* dengan kelas 1 sebagai kategori CDMS dan kelas 2 sebagai kategori non-CDMS. Dalam Gambar 10 tersebut, terdapat 21 data yang di prediksi benar sesuai dengan nilai aktual CDMS bernilai *True Positif* (TP), kemudian 6 data dengan nilai aktual CDMS namun diprediksi salah sebagai non-CDMS, menunjukkan prediksi model salah bernilai *False Negatif* (FN), kemudian 4 data dengan nilai aktual non-CDMS namun diprediksi salah sebagai CDMS, menunjukkan prediksi model salah bernilai *False Positif* (FP), dan 24 data yang di prediksi benar sesuai dengan nilai aktual non-CDMS bernilai *True Negatif* (FN).



Gambar 9. Hasil Tampilan *Output Heatmap Confution Matrix*

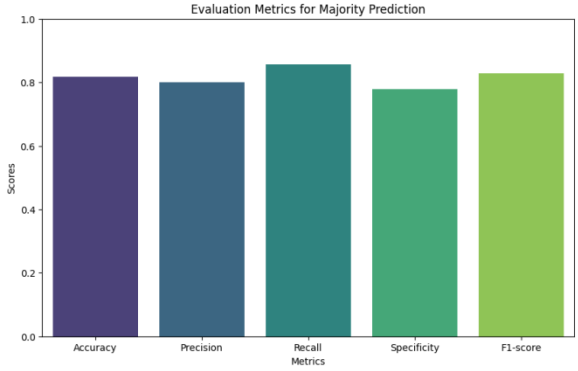
Output hasil evaluasi perhitungan menggunakan confution matrix meliputi *accuracy, precision, recall (sensitivity), specificity, dan f1-score* yang dihitung berdasarkan jumlah obyek penelitian yang diprediksi dengan benar dan salah dapat dilihat pada Gambar 11.

```
Perhitungan Nilai Evaluasi :
        Metric  Majority Prediction
0     Accuracy             0.818182
1    Precision             0.800000
2       Recall             0.857143
3  Specificity             0.777778
4     F1-score             0.827586

Confusion Matrix untuk Majority Predictions:
[[21  6]
 [ 4 24]]
```

Gambar 10. Hasil Perhitungan Nilai Evaluasi

Langkah selanjutnya adalah menampilkan *bar plot* hasil visualisasi dari evaluasi menggunakan metode *confusion matrix,* Secara umum, untuk evaluasi yang baik yakni dengan memperoleh hasil setinggi mungkin setiap komponen evaluasi *confusion matrix* meliputi *accuracy, precision, recall, specificity, f1-score.*



Gambar 11. Hasil Tampilan *Output Bar Plot* Nilai Hasil Evaluasi

3. KESIMPULAN

Berdasarkan permasalahan yang telah dipaparkan, maka dapat disimpulkan bahwa algoritma random forest dapat diimplementasikan untuk mendeteksi penyakit multiple sclerosis melalui beberapa proses, diantaranya preprocessing data, meliputi pencarian missing value, melakukan cleaning data yang mengandung missing value, dan melakukan pembobotan variabel yang menghasilkan 271 record dataset dengan 16 variabel fitur dan 1 variabel label yang siap digunakan untuk klasifikasi. Kemudian proses klasifikasi menggunakan proporsi perbandingan 80% data latih dan 20% data uji, dengan membangun 100 model klasifikasi algoritma random forest untuk menghasilkan pohon keputusan berbeda menggunakan kriteria bootsrap aggregating atau bagging, random feature selection, dan entropy. Hasil setiap klasifikasi penyakit multiple sclerosis diakumulasikan atau dipilih sesuai majority vote dan dilakukan evaluasi menggunakan pengujian kinerja confusion matrix menghasilkan nilai accuracy sebesar 81%, precision sebesar 80%, recall sebesar 85%, specificity sebesar 77%, dan f1-score sebesar 82%. Hasil penelitian yang dikembangkan berdasarkan beberapa studi literatur dan referensi penelitian terdahulu yang memiliki topik dan tema yang serupa, menunjukkan bahwa model klasifikasi menggunakan algoritma random forest memiliki peforma yang cukup baik dan dapat diimplementasikan dalam klasifikasi dataset penyakit multiple sclerosis.

DAFTAR PUSTAKA

[1] J. C. Suryo, "Sklerosis Multipel: Diagnosis dan Tatalaksana," *Cermin Dunia Kedokt.*, vol. 48, no. 8, pp. 296–303, 2021, doi: 10.55175/cdk.v48i8.111.

[2] I. M. Dighriri *et al.*, "An Overview of the History, Pathophysiology, and Pharmacological Interventions of Multiple Sclerosis," *Cureus*, vol. 15, no. 1, pp. 1–12, 2023, doi: 10.7759/cureus.33242.

[3] MSIF, "Atlas of MS 3 rd edition," 2020.

[4] M. T. Wallin *et al.*, "Global, regional, and national burden of multiple sclerosis 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016," *Lancet Neurol.*, vol. 18, no. 3, pp. 269–285, 2019, doi: 10.1016/S1474-4422(18)30443-5.

[5] I. P. E. Kusmadana, W. Margo, and G. Suputra, "Sklerosis multipel pada pria Bali: sebuah laporan kasus," *Intisari Sains Medis*, vol. 14, no. 1, pp. 407–411, 2023, doi: 10.15562/ism.v14i1.1625.

[6] H. Larassati, R. Estiasari, R. E. Yunus, and P. M. Parizel, "State-of-the-Art Review: Demyelinating Diseases in Indonesia," *Mult. Scler. Int.*, vol. 2021, pp. 1–13, 2021, doi: 10.1155/2021/1278503.

[7] S. Mutmainnah, G. Abdurrahman, and H. A. Al Faruq, "Optimasi Algoritma C4. 5 Menggunakan Teknik Bagging Pada Data Kadar Karat Emas," *Metode*, 2018, [Online]. Available: http://repository.unmuhjember.ac.id/4416/9/ARTIKEL.pdf

[8] F. S. Pamungkas, B. D. Prasetya, and I. Kharisudin, "Perbandingan Metode Klasifikasi Supervised Learning pada Data Bank Customers Menggunakan Python," *Prism. Pros. Semin. Nas. Mat.*, vol. 3, pp. 692–697, 2020, [Online]. Available: https://journal.unnes.ac.id/sju/index.php/prisma/article/view/37875

[9] E. R. B. Sebayang, Y. H. Chrisnanto, and Melina, "Klasifikasi Data Kesehatan Mental di Industri Teknologi Menggunakan Algoritma Random Forest," *IJESPG J.*, vol. 1, no. 3, pp. 237–253, 2023.

[10] B. Wahyuningtyas, I. I. Tritoasmoro, and N. Ibrahim, "Identifikasi Penyakit Pada Daun Kopi Menggunakan Metode Local Binary Pattern Dan Random Forest," *e-Proceeding Eng.*, vol. 8, no. 6, pp. 2972–2980, 2022.

[11] M. R. Akbar Ariyadi, S. Lestanti, and S. Kirom, "Klasifikasi Balita Stunting Menggunakan Random Forest Classifier Di Kabupaten Blitar," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 6, pp. 3846–3851, 2024, doi: 10.36040/jati.v7i6.7822.

[12] A. Ekawijana, A. Bakhrun, and M. . Kurniawan, "Deteksi Serangan DDOS Pada Jaringan SDN dengan Metode Random," *J. Media Inform. Budidarma*, vol. 8, pp. 685–694, 2024, doi: 10.30865/mib.v8i1.6928.

[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] A. Nugroho, I. Asror, and Y. F. A. Wibowo, "Klasifikasi Tingkat Kualitas Udara DKI Jakarta Berdasarkan Open Government Data Menggunakan Algoritma Random Forest," *eProceedings Eng.*, vol. 10, No. 2, no. 2, pp. 1824–1834, 2023, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/20030%0Ahttps://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/20030/19395

[15] A. H. Harahap, I. M. B. A. Malik, M. I. N. Imam, M. T. S. Bilhaq, A. A. Nur, and S. L. D. Agustini, "Klasifikasi Diagnosa Penyakit Jantung menggunakan Algoritma Random Forest," *Tek. Inform. UIN Sunan Gunung Djati Bandung*, vol. 3, pp. 1–51, 2021.

[16] E. Suryati, Styawati, and A. A. Aldino, "Analisis Sentimen Transportasi Online Menggunakan Ekstraksi Fitur Model Word2vec Text Embedding Dan Algoritma Support Vector Machine (SVM)," *J. Teknol. Dan Sist. Inf.*, vol. 4, no. 1, pp. 96–106, 2023, [Online]. Available: https://doi.org/10.33365/jtsi.v4i1.2445