JSiI | Jurnal Sistem Informasi
Vol. 11 No. 2 September 2024, Hal.87-96

DOI 10.30656/jsii.v11i2.9161
p-ISSN: 2406-7768
e-ISSN: 2581-2181

OPTIMASI KLASIFIKASI DECISION TREE DENGAN TEKNIK PRUNING UNTUK MENGURANGI OVERFITTING

Cindy Novi Syahputri[1*], Muhammad Siddik Hasibuan[2]

[1,2] Fakultas Sains dan Teknologi, Ilmu Komputer, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia
Jl.Lap.Golf No.120, Kp.Tengah, Kec. Pancur Batu, Kabupaten Deli Serdang, Sumatera Utara,20353
[1]cindyputri2018@gmail.com
[2]muhammadsiddik@uinsu.ac.id

Abstract

Penelitian ini bertujuan untuk mengoptimalkan klasifikasi *Decision Tree* menggunakan teknik *pruning* untuk mengurangi *overfitting* pada dataset penyakit jantung Kaggle. *Overfitting* adalah masalah umum dalam pembelajaran mesin, ketika model terlalu cocok dengan data pelatihan dan kehilangan kemampuannya untuk menggeneralisasi data baru dengan baik. Teknik *pruning*, termasuk *prepruning* dan *postpruning*, diterapkan untuk membatasi kompleksitas model dan meningkatkan kemampuannya dalam mengklasifikasikan data baru. Hasilnya menunjukkan bahwa model dengan postpruning memiliki performa terbaik, dengan akurasi 0,8841, recall 0,8571, presisi 0,8571, dan skor F1 0,8571. Sebagai perbandingan, model dengan prepruning memiliki akurasi sebesar 0,8333, recall sebesar 0,8304, presisi sebesar 0,8304, dan skor F1 sebesar 0,7434. Peningkatan metrik ini menegaskan bahwa postpruning lebih efektif dalam mengurangi overfitting dan meningkatkan kemampuan generalisasi model. Dengan demikian, teknik postpruning dapat dianggap sebagai metode unggulan dalam mengoptimalkan kinerja *Decision Tree Classifier* untuk klasifikasi penyakit jantung. Penelitian ini diharapkan dapat berkontribusi pada pengembangan model prediksi yang lebih akurat dalam diagnosis penyakit jantung, sehingga membantu upaya pencegahan dan pengobatan yang lebih baik.

Kata Kunci: Decision Tree, Pruning, Prepruning, Postpruning, Overfitting, Heart Disease Dataset, Kaggle, Machine Learning, Classification, Model Optimization.

I. PENDAHULUAN

Dalam dunia kesehatan, diagnosis dini dan akurat dari penyakit jantung sangat penting untuk mengurangi angka kematian dan meningkatkan kualitas hidup pasien. Penyakit jantung merupakan salah satu penyebab utama kematian di seluruh dunia, sehingga metode analisis data yang efektif sangat diperlukan untuk membantu dalam diagnosis penyakit ini [1] .

Metode analisis dalam diagnosis penyakit telah mengalami perkembangan signifikan seiring dengan kemajuan teknologi dan pemanfaatan data besar (big data) [2]. Salah satu pendekatan yang banyak digunakan adalah algoritma machine learning, yang memungkinkan komputer untuk belajar dari data historis dan membuat prediksi atau keputusan tanpa diprogram secara eksplisit. Contohnya, algoritma Decision Tree [3], Random Forest, dan Support Vector Machine (SVM) sering digunakan untuk menganalisis data medis dan mengidentifikasi pola yang relevan dengan kondisi kesehatan pasien [4]. Selain itu, teknik deep learning seperti Convolutional Neural Networks (CNN) [5] dan Recurrent Neural Networks (RNN) [6] telah terbukti efektif dalam menangani data yang kompleks dan tidak terstruktur, seperti citra medis dan rekam medis elektronik. Metode-metode ini dapat digunakan untuk mendeteksi berbagai penyakit, termasuk kanker, diabetes, dan penyakit jantung, dengan tingkat akurasi yang tinggi. Dengan menggabungkan kekuatan komputasi dan analisis data, metode-metode ini membantu dalam diagnosis dini, memberikan rekomendasi pengobatan yang lebih tepat, dan meningkatkan hasil kesehatan pasien secara keseluruhan. Salah satu teknik yang sering digunakan dalam analisis data medis adalah algoritma Decision Tree [7].

Pada penelitian terdahulu yang dilakukan oleh [8] Artikel ini menggunakan percobaan menggunakan dataset ILPD Kinerja sistem dengan menggunakan metode pruning mempunyai akurasi sebesar 73,76%. Namun, masih terdapat kekurangan pada penelitian ini yang terletak pada nilai akurasi masih sedikit bervariasi antara metode lain menggunakan pruning. Diperlukan penelitian lebih lanjut untuk mengoptimalkan metode decision tree dalam klasifikasi penyakit. Artikel ini juga mencatat perlunya pengembangan lebih lanjut untuk meningkatkan akurasi dan optimalisasi metode pruning pada decision tree dalam klasifikasi. Sedangkan pada penelitian berikutnya yang dilakukan oleh [9] menawarkan model yang lebih kecil dengan skala yang ringan, yang melibatkan sedikit parameter dan memiliki waktu asumsisi cepat serta kompleksitas komputasi floating-point yang rendah. Ini sangat penting terutama ketika sumber daya komputasi terbatas. namun Artikel ini tidak memberikan detail yang cukup tentang implementasi teknis dari WP-UNet. Informasi seperti arsitektur jaringan yang spesifik, pemilihan hiperparameter, dan langkah-langkah pengurangan bobot tidak dijelaskan secara mendalam. Ini membuat sulit bagi penelit lain untuk mengulangi atau mengembangkan lebih lanjut dari hasil yang diperoleh.

Received: July 15th, 2024| Published: September 15th, 2024

DOI 10.30656/jsii.v11i2.9161
p-ISSN: 2406-7768
e-ISSN: 2581-2181

Algoritma Decision Tree memiliki beberapa keunggulan, termasuk interpretabilitas yang tinggi dan kemampuan untuk menangani data dengan berbagai tipe. Namun, salah satu kelemahan utama dari algoritma ini adalah kecenderungannya untuk mengalami overfitting, terutama ketika diterapkan pada dataset yang kompleks. Overfitting terjadi ketika model terlalu menyesuaikan dengan data pelatihan, sehingga kehilangan kemampuan untuk menggeneralisasi pada data baru [10].

Dari literatur yang telah dibahas, Overfitting merupakan salah satu masalah umum yang dihadapi dalam proses pembelajaran mesin, terutama dalam klasifikasi data. Overfitting terjadi ketika model terlalu kompleks dan menyesuaikan terlalu banyak dengan data pelatihan, sehingga kehilangan kemampuan untuk menggeneralisasi data baru dengan baik. Salah satu metode klasifikasi yang sering digunakan dalam pembelajaran mesin adalah Decision Tree. Decision Tree memiliki kelebihan dalam interpretabilitas dan kemudahan implementasi, namun rentan terhadap overfitting

Dalam upaya untuk mengatasi masalah overfitting, berbagai teknik telah dikembangkan, termasuk pruning, penggunaan ensemble methods seperti Random Forests dan Gradient Boosting, serta optimasi hyperparameter. Namun, masih diperlukan penelitian lebih lanjut untuk mengeksplorasi efektivitas metode-metode ini dalam konteks dataset spesifik, seperti dataset penyakit jantung dari Kaggle. Untuk mengatasi masalah overfitting pada Decision Tree, teknik pruning sering digunakan. Pruning adalah proses pemangkasan cabang-cabang pada pohon keputusan yang tidak memberikan kontribusi signifikan terhadap akurasi model. Dengan melakukan pruning, kompleksitas model dapat dikurangi, sehingga meningkatkan kemampuan generalisasi dan mengurangi risiko overfitting [11].

Dataset Heart Disease Kaggle adalah salah satu dataset yang sering digunakan dalam penelitian untuk mengembangkan model prediksi penyakit jantung. Dataset ini terdiri dari berbagai fitur yang berkaitan dengan kondisi medis dan riwayat kesehatan pasien, yang dapat digunakan untuk membangun model klasifikasi. Dengan mengoptimalkan algoritma Decision Tree pada dataset ini, diharapkan dapat diperoleh model yang tidak hanya akurat, tetapi juga mampu mengurangi overfitting dan memberikan hasil yang dapat diandalkan dalam diagnosis penyakit jantung[12].

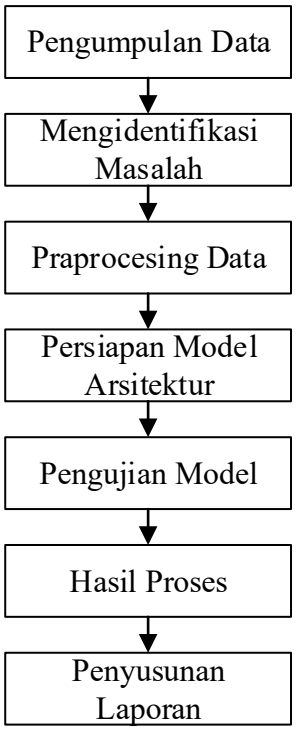
Dalam penelitian ini, optimasi klasifikasi Decision Tree dengan teknik pruning akan diterapkan pada dataset Heart Disease yang diambil dari Kaggle. Penyakit jantung adalah salah satu penyebab utama kematian di seluruh dunia, sehingga penting untuk mengembangkan model prediktif yang akurat dalam mendiagnosis penyakit ini. Dataset Heart Disease Kaggle menyediakan data yang relevan untuk melakukan analisis dan pengembangan model prediktif.

Penelitian ini bertujuan untuk mengevaluasi efektivitas teknik pruning dalam mengurangi overfitting pada Decision Tree menggunakan dataset Heart Disease Kaggle. Dengan demikian, diharapkan dapat menghasilkan model yang lebih akurat dan andal dalam memprediksi risiko penyakit jantung, yang pada akhirnya dapat berkontribusi pada upaya pencegahan dan pengobatan yang lebih baik.

II. METODOLOGI PENELITIAN

A. Kerangka Penelitian

Kerangka penelitian atau sering juga disebut sebagai framework penelitian, adalah kerangka konseptual yang digunakan untuk merancang dan juga menyusun suatu penelitian yang akan diteliti. Kerangka penelitian membantu dalam menjelaskan dan menjabarkan elemen- elemen utama yang ada pada penelitian, sebagai pembibing penyusun penelitian, hipotesis, dan juga sebagai desain sebuah penelitian. Kerangka yang dibuat harus kokoh untuk memudahkan pemahaman atau penyelesaian suatu topik penelitian .Rancangan penelitian digunakan untuk menguraikan dan menyelesaikan masalah dalam penelitian. Berikut adalah kerangka tahapan penelitian yang nantinya dijalankan dalam penyelesaian permasalahan yang akan dibahas:



Gambar 1. Rancangan Penelitian

Dari gambar 3 dapat diuraikan bahwa tahap pertama adalah mengumpulkan data yang diperoleh dari website Kaggle.com. Selanjutnya, dilakukan studi pustaka untuk mengumpulkan informasi yang relevan dengan topik masalah dan melengkapi pengetahuan teori yang digunakan dalam penelitian, diambil dari artikel ilmiah, prosiding, dan buku. Setelah semua data terkumpul, dilakukan identifikasi masalah untuk mendapatkan dataset yang sesuai dengan bobot yang ditentukan. Pada tahap praproses data, dilakukan perubahan terhadap beberapa tipe data atribut dataset untuk mempermudah pemahaman, serta melakukan seleksi dengan memperhatikan konsistensi data, missing value, dan redundancy agar data cocok dengan metode yang digunakan. Proses persiapan model arsitektur melibatkan pengujian klasifikasi standar dan klasifikasi menggunakan optimasi REP pruning melalui tahap pelatihan dan pengujian

Received: July 15th, 2024| Published: September 15th, 2024

DOI 10.30656/jsii.v11i2.9161
p-ISSN: 2406-7768
e-ISSN: 2581-2181

untuk menentukan model arsitektur terbaik yang akan menjadi pedoman dalam melakukan prediksi data. Pengujian model dilakukan dengan proses yang telah dibuat menggunakan bahasa pemrograman Python menggunakan google colab. Tahap hasil proses mengambil hasil akhir dari seluruh proses yang dilakukan. Terakhir, penyusunan laporan mencakup latar belakang, rumusan masalah, tujuan penelitian, metodologi penelitian, hasil dan analisis, kesimpulan, saran, dan daftar pustaka.

B. Identifikasi Masalah

Pada penelitian ini, fokus utama akan difokuskan pada analisis pruning pada model algoritma decision tree yang diuji menggunakan phyton dan menggunakan google colab dengan objek pengujianya adalah dataset pasien gagal jantung. Decision tree merupakan salah satu algoritma machine learning yang sering digunakan dalam pengambilan keputusan. Pruning merupakan teknik yang digunakan untuk mengoptimalkan performa decision tree dengan menghapus cabang-cabang yang tidak signifikan atau redundant. Rencana pembahasan akan mencakup pemahaman dasar mengenai algoritma decision tree, metode pruning, dan relevansi analisis pruning terhadap peningkatan akurasi dan efisiensi model. Selain itu, penelitian ini juga akan membahas berbagai teknik pruning yang ada dan dampaknya terhadap trade-off antara kompleksitas model dan kinerja prediktif. Diharapkan pembahasan ini dapat memberikan wawasan mendalam mengenai pengaruh analisis pruning terhadap decision tree, serta memberikan kontribusi pada pengembangan model machine learning yang lebih efisien dan efektif.

C. Dataset Penelitian

Untuk penelitian ini, kumpulan data yang terdiri dari lima kumpulan data berbeda dari berbagai sumber diperoleh dari Kaggle (https://www.kaggle.com). Kumpulan data ini meliputi Cleveland (303 observasi), Hongaria (294 observasi), Swiss (123 observasi), Long Beach VA (200 observasi), dan kumpulan data Stalog (hati) (270 observasi). Gabungan dataset terdiri dari total 918 observasi dan mencakup 12 variabel, dengan 11 variabel berfungsi sebagai masukan dan 1 variabel berfungsi sebagai keluaran (label). Setiap subset variabel disesuaikan menurut kebutuhan spesifik penelitian. Bagian selanjutnya memberikan penjelasan komprehensif tentang variabel-variabel yang digunakan dalam penelitian HF.

Penelitian ini menggunakan dataset sampel (data lengkap dapat diakses di https://shorturl.at/klvS2), seperti disajikan pada Tabel 3.1, yang terdiri dari berbagai parameter terkait pasien. Parameter tersebut meliputi data medis dari pasien yang mencakup usia, jenis kelamin, tipe nyeri dada, tekanan darah istirahat, tingkat kolesterol, gula darah puasa, hasil elektrokardiogram istirahat, denyut jantung maksimal, angina yang diinduksi oleh latihan, depresi ST, kemiringan segmen ST saat latihan puncak, dan diagnosis penyakit jantung. Pasien berusia antara 37 hingga 68 tahun dengan berbagai tipe nyeri dada seperti Angina Tipikal (ATA), Angina Non-tipikal (NAP), Asimptomatik (ASY), dan Angina Tipikal (TA). Tekanan darah istirahat berkisar antara 110 hingga 164 mm Hg, dan tingkat kolesterol antara 131 hingga 339 mg/dl. Beberapa pasien memiliki gula darah puasa >120 mg/dl dan hasil elektrokardiogram menunjukkan normal, abnormalitas segmen ST-T, atau hipertrofi ventrikel kiri. Denyut jantung maksimal berkisar antara 90 hingga 182, dengan beberapa pasien mengalami angina saat latihan. Depresi ST berkisar antara 0 hingga 3,4 dengan kemiringan segmen ST bervariasi antara naik, datar, dan turun. Diagnosis akhir menunjukkan adanya atau tidak adanya penyakit jantung, dengan beberapa pasien dinyatakan normal dan lainnya menderita penyakit jantung dan kelas keluaran yang menunjukkan adanya HeartDisease penyakit jantung (1 untuk HF dan 0 untuk normal). Dataset bisa dilhat pada tabel 1.

Tabel 1. Sampel Dataset Heart disease

No	Age	Sex	ChestPain Type	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
1	40	M	ATA	140	289	0	Normal	172	N	0	Up	Normal
2	49	F	NAP	160	180	0	Normal	156	N	1	Flat	HF
3	37	M	ATA	130	283	0	ST	98	N	0	Up	Normal
4	48	F	ASY	138	214	0	Normal	108	Y	1,5	Flat	HF
5	54	M	NAP	150	195	0	Normal	122	N	0	Up	Normal
6	39	M	NAP	120	339	0	Normal	170	N	0	Up	Normal
7	45	F	ATA	130	237	0	Normal	170	N	0	Up	Normal
8	54	M	ATA	110	208	0	Normal	142	N	0	Up	Normal
9	37	M	ASY	140	207	0	Normal	130	Y	1,5	Flat	HF
10	48	F	ATA	120	284	0	Normal	120	N	0	Up	Normal
…	…	…	….	…	…	…	….	…	….	….	….	...
913	57	F	ASY	140	241	0	Normal	123	Y	0,2	Flat	HF
914	45	M	TA	110	264	0	Normal	132	N	1,2	Flat	HF
915	68	M	ASY	144	193	1	Normal	141	N	3,4	Flat	HF
916	57	M	ASY	130	131	0	Normal	115	Y	1,2	Flat	HF
917	57	F	ATA	130	236	0	LVH	174	N	0	Flat	HF
918	38	M	NAP	138	175	0	Normal	173	N	0	Up	Normal

D. Model Usulan Decesion Tree

Penelitian melibatkan serangkaian langkah yang komprehensif dalam pemrosesan dan pengembangan model. Tahap awal dimulai dengan pra-pemrosesan data, di mana data mentah disaring, dibersihkan, dan diubah menjadi format yang

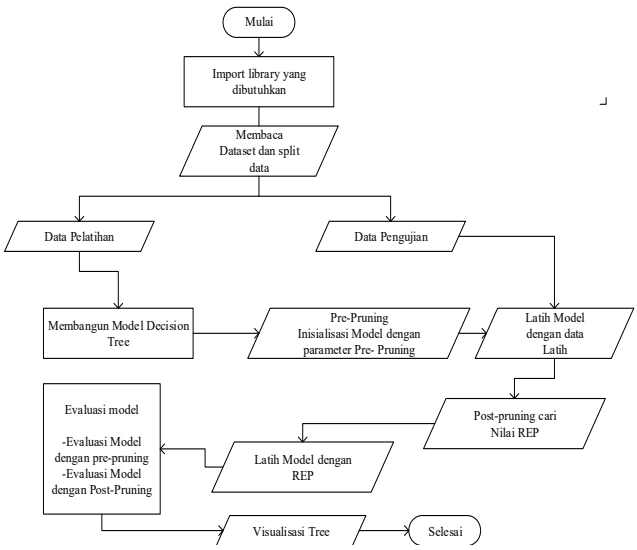
Received: July 15th, 2024| Published: September 15th, 2024

DOI 10.30656/jsii.v11i2.9161
p-ISSN: 2406-7768
e-ISSN: 2581-2181

sesuai untuk analisis. Langkah selanjutnya adalah pembentukan model decision tree menggunakan algoritma seperti CART, atau C4.5, yang didasarkan pada data pelatihan. Model ini kemudian menjalani tahap pelatihan, di mana parameter-parameter internalnya disesuaikan untuk mengoptimalkan kinerja pada data pelatihan tersebut.

Proses penting berikutnya adalah pruning, yang merupakan fokus utama dari penelitian ini. Pruning adalah teknik yang digunakan untuk mengurangi kompleksitas model dengan menghilangkan cabang-cabang dari decision tree yang tidak memberikan kontribusi signifikan atau dapat menyebabkan overfitting pada data pelatihan. Terdapat beberapa teknik pruning yang dapat diterapkan, seperti Reduced-Error Pruning(REP), Cost Complexity Pruning (CCP), dan Minimal Cost-Complexity Pruning (MCCP), yang masing-masing memiliki pendekatan dan keunggulan tersendiri.

Penerapan Teknik Pruning untuk Klasifikasi Data Mining dalam Prediksi HF: Langkah-langkah dan Evaluasi Model" menggambarkan proses utama dalam mengembangkan model prediksi gagal jantung (HF) dengan menggunakan teknik Pruning. Langkah-langkahnya mencakup persiapan data awal, pemilihan model, optimasi hyperparameter, pelatihan, evaluasi, dan pelaporan. Semua tahapan ini sangat penting dalam pembangunan model klasifikasi yang tepat dan handal untuk prediksi HF. Representasi visual dari langkah-langkah ini dapat dilihat dalam Gambar 2 yang menggarisbawahi signifikansinya dalam keseluruhan proses. Gambar tersebut mencerminkan teknik optimasi Pruning yang digunakan dalam penelitian ini. Model yang dihasilkan kemudian akan dibandingkan dengan model klasifikasi standar yang terdiri dari tempat pengklasifikasi untuk mengevaluasi performa dan efektivitas model yang diusulkan. Model usulan bisa dilihat pada gambar 2.



Gambar 2. Model proses

Secara sederhana, pada gambar 2 menjelaskan *pruning* pada *decision tree* dimulai dengan langkah pertama dalam proses di mana semua kegiatan untuk membangun dan mengevaluasi model *decision tree* dimulai. Langkah berikutnya adalah mengimpor pustaka yang diperlukan seperti numpy untuk operasi numerik, pandas untuk manipulasi data, sklearn (scikit-learn) untuk algoritma machine learning, dan matplotlib untuk visualisasi data dan hasil model. Selanjutnya, dataset yang akan digunakan untuk melatih dan menguji model dimuat, yang bisa berasal dari berbagai sumber seperti file CSV, database, atau dataset yang tersedia di pustaka seperti scikit-learn. Dataset kemudian dibagi menjadi dua bagian, yaitu set latih (70%) untuk melatih model dan set uji (30%) untuk menguji performa model yang telah dilatih. Setelah itu, model decision tree dibuat menggunakan set latih, diikuti dengan penerapan teknik pre-pruning untuk mencegah overfitting dengan membatasi kompleksitas model pada tahap awal pelatihan. Parameter pre-pruning seperti max_depth, min_samples_split, dan min_samples_leaf ditentukan sebelum melatih model, yang kemudian dilatih dengan data latih. Teknik post-pruning kemudian diterapkan setelah model dilatih, dengan memangkas cabang-cabang dari decision tree yang tidak signifikan untuk meningkatkan generalisasi model.

Post-pruning dengan Reduced Error Pruning (REP) adalah teknik dimana pohon keputusan dibangun hingga mencapai kedalaman maksimal tanpa batasan, dan kemudian secara bertahap dilakukan penghapusan node yang tidak signifikan. Proses ini melibatkan pengujian dampak penghapusan setiap node terhadap kinerja model pada set validasi terpisah. Jika penghapusan suatu node tidak menurunkan akurasi atau bahkan meningkatkan kinerja pada set validasi, maka node tersebut dihapus, sehingga model yang dihasilkan menjadi lebih sederhana, mengurangi overfitting, dan lebih baik dalam generalisasiSelain mengembangkan model yang akurat dan kuat, penting juga untuk mengevaluasi keakuratan model dalam memprediksi HF. Hal ini dilakukan melalui matriks konfusi dan kurva karakteristik operasi penerima (ROC)/area under cover (AUC). Kurva ROC dibuat berdasarkan nilai yang dihitung dari matriks konfusi, yang membandingkan tingkat positif palsu (FPR) dan tingkat positif sebenarnya.

tarif (TPR). Di mana:

a) $FPR = False\ Positive/(False\ Positiv\ e + True\ Negative)$;

b) $TPR = True\ Positive/(True\ Positive + F\ alse\ Negative)$;

Selanjutnya BURUK jika kurva yang dihasilkan mendekati garis pangkal atau garis yang memotong titik 0,0. dan BAIK, jika kurvanya mendekati 0,1 poin.

III. HASIL DAN PEMBAHASAN

A. Hasil

a. Permrosesan Data

Pembagian data untuk melatih model decision tree dengan teknik pruning menjadi data pelatihan dan validasi, langkah ini menjadi langkah krusial dalam proses pembangunan model. Setelah persiapan data, pada saat masuk ketahapan prmosesan data, terdapat ***Error***, muncul karena ada nilai string dalam data yang tidak dapat diubah menjadi float. Dalam dataset heartdisease, beberapa kolom berisi data kategorikal yang perlu

Received: July 15th, 2024| Published: September 15th, 2024

DOI 10.30656/jsii.v11i2.9161
p-ISSN: 2406-7768
e-ISSN: 2581-2181

dikonversi menjadi nilai numerik sebelum digunakan untuk melatih model.

Untuk mengatasi masalah ini, maka dilakukan beberapa langkah preprocessing data, seperti encoding data kategorikal menjadi numerik. Peneliti menggunakan OneHotEncoder atau LabelEncoder dari scikit-learn untuk tujuan ini.

Berikut *pseudocode* nya:

```
  # Handle categorical data if any
  categorical_columns = data.select_dtypes(include=['object']).columns
  label_encoders = {}
  # Encode categorical columns
  for column in categorical_columns:
      le = LabelEncoder()
      data[column] = le.fit_transform(data[column])
      label_encoders[column] = le
```

Penjelasan dari *pseudocode*: Handle categorical data: Kode ini pertama-tama memeriksa kolom mana dalam dataset yang bersifat kategorikal menggunakan select_dtypes. Kemudian, kita menggunakan LabelEncoder untuk mengubah nilai kategorikal menjadi numerik. Label encoding: Setiap kolom kategorikal diubah menjadi nilai numerik menggunakan LabelEncoder.

Pastikan untuk mengganti path menuju ke data yaitu heartdisease.csv dengan path yang sebenarnya dari file datase. Setelah preprocessing data, Anda seharusnya dapat melatih model pohon keputusan tanpa mengalami error konversi data. Jika dataset mengandung lebih banyak jenis data kategorikal kompleks, mungkin perlu menggunakan OneHotEncoder atau teknik encoding lain yang lebih sesuai dengan kebutuhan dataset.

Dalam praktik ini, data dibagi menjadi dua set: data pelatihan dan data pengujian, dan pada pengujian model menggunakan tools jupiter nootbook dengan bahasa python, dalam pembagian data pelatihan dan pengujian dilakukan secara ***random/acak.*** Data pelatihan, yang terdiri dari sekitar 70% dari total data, digunakan untuk membangun dan melatih model, memungkinkan model untuk belajar pola dan hubungan dalam data. Sementara itu, data pengujian, yang terdiri dari sisa 30% data, digunakan untuk mengevaluasi kinerja model setelah pelatihan. Pembagian ini memastikan bahwa model dapat diuji pada data yang belum pernah dilihatnya selama pelatihan, sehingga memberikan gambaran yang lebih akurat tentang kemampuannya untuk menggeneralisasi pada data baru yang belum pernah ditemukan sebelumnya.

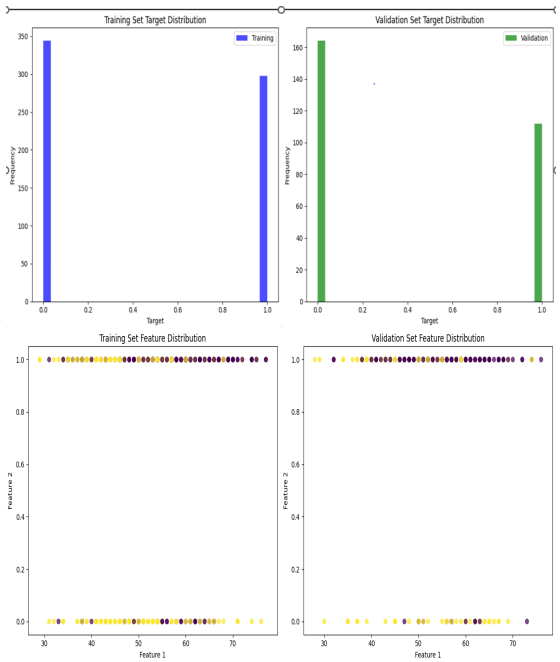
Berikut merupakan *pseudocode* pada proses pembagian data pada bahasa permrograman python yang dirancangan untuk membagi data menjadi dua, yaitu data training dan validation.

```
  # Split the data into training and validation sets
  X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.3, random_state=42)
```

Penjelasan Kode:

Data Preparation: Membaca dataset dan melakukan encoding kolom kategorikal jika diperlukan. Memisahkan dataset menjadi fitur (X) dan target (y). **Split Data:** Memisahkan dataset menjadi data pelatihan dan validasi menggunakan train_test_split. **Visualisasi Distribusi Target:** Membuat histogram untuk menunjukkan distribusi variabel target dalam set pelatihan dan validasi. **Visualisasi Distribusi Fitur:** Jika terdapat minimal dua fitur, membuat scatter plot untuk menunjukkan distribusi dua fitur pertama dalam set pelatihan dan validasi.

Kode ini memberikan gambaran visual tentang bagaimana data dibagi antara set pelatihan dan validasi. Visualisasi bisa dilihat pada gambar 4.



Gambar 4. Gambaran visual tentang bagaimana data dibagi antara set pelatihan dan validasi

Pada gambar 4 tersebut menunjukkan distribusi fitur dari data yang dibagi antara set pelatihan (kiri) dan set validasi (kanan). Dari plot, terlihat bahwa distribusi fitur pada kedua set data ini cukup konsisten, di mana nilai Feature 1 berada di rentang yang sama pada kedua set, sekitar 30 hingga 75, sementara Feature 2 tampaknya memiliki dua nilai dominan, yaitu 0 dan 1. Warna yang berbeda pada titik-titik dalam grafik kemungkinan merepresentasikan kelas atau kelompok yang berbeda dalam data. Secara keseluruhan, distribusi fitur pada kedua set data menunjukkan bahwa pembagian data antara set pelatihan dan validasi dilakukan dengan cukup merata, yang ideal untuk menjaga generalisasi model yang dilatih pada set pelatihan.

Berikutnya untuk mendapatkan hasil dari penelitian untuk menganalisis teknik *pruning* pada model algoritma *Decision Tree* dapat meningkatkan kinerja dan efisiensi klasifikasi menggunakan Algoritma C4.5. Proses perhitungan untuk menentukan klasifikasi pada faktor penyakit jantung

Received: July 15th, 2024| Published: September 15th, 2024

DOI 10.30656/jsii.v11i2.9161
p-ISSN: 2406-7768
e-ISSN: 2581-2181

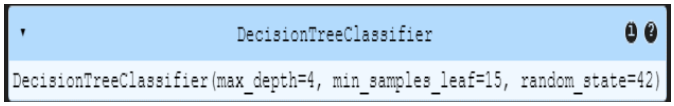
menggunakan Algoritma C4.5 adalah dengan melihat keadaan normal atau gagal jantung (HF) Heart Failure. Dan melihat hasil analisis *Gini* dari semua kasus dengan membandingkan decision tree ***prepruning*** dengan ***postpruning***. Setelah itu akan dilakukan perhitungan manual dan penyesuaian hasil melalui pengujian menggunakan *software Google colag menggunakan tensor.*

b. Membangun Model *Decesion Tree PrePruning*

Untuk melihat peforma kinerja dari decision tree prepruning dengan yang sudah dioptimasi dengan teknik pruning (postpruning), pada bagian ini akan mendapatkan hasil dari penelitian untuk menganalisis model algoritma *Decision Tree* prepruning. Untuk permosesan data sama semua pada setiap model, data data menggunakan data heartdisease, dengan membagi dua data menjadi training dan testing seklaigus validasi.

Berikut *pseudocode* membangun pengklasifikasi pohon keputusan tanpa pemangkasan.

```
  # Build the decision tree classifier with pre-pruning
  clf_pre_pruning                                =
DecisionTreeClassifier(max_depth=4,
min_samples_leaf=15, random_state=42)
  clf_pre_pruning.fit(X_train, y_train)
```



Gambar 5. Model Klasifikasi *DecesionTree*

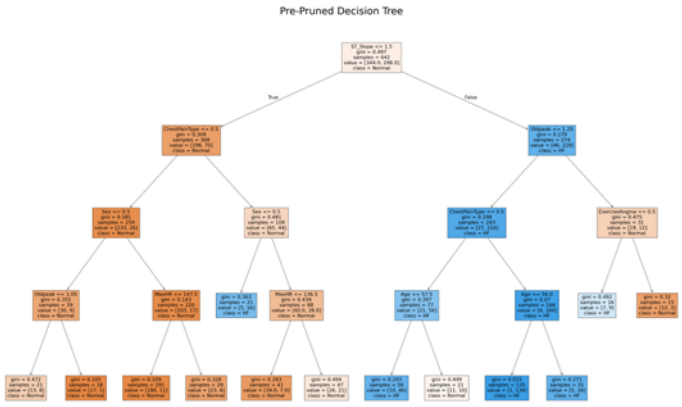
Pada gambar 5 merupakan Model DecisionTreeClassifier dimana algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi dengan struktur seperti diagram alir di mana setiap simpul internal mewakili tes pada suatu atribut, setiap cabang mewakili hasil tes, dan setiap simpul daun mewakili label kelas. Algoritma ini bekerja dengan cara membagi dataset menjadi subset berdasarkan fitur tertentu hingga semua data dalam subset memiliki label yang sama atau kedalaman maksimum pohon tercapai. Keunggulan model ini adalah kemudahan interpretasi dan kemampuannya menangkap hubungan non-linear antara fitur-fitur, namun memiliki kelemahan seperti cenderung overfitting dan sensitif terhadap perubahan kecil pada data pelatihan. Parameter utama yang dapat disesuaikan dalam scikit-learn termasuk criterion, splitter, max_depth, min_samples_split, min_samples_leaf, max_features, dan random_state. Model ini dapat diterapkan dalam berbagai bidang seperti medis untuk mendiagnosis penyakit, keuangan untuk mendeteksi penipuan, dan pemasaran untuk segmentasi pelanggan. Visualisasi pohon keputusan dapat dilakukan menggunakan pustaka seperti graphviz atau fungsi plot_tree dari scikit-learn untuk memahami keputusan model.

Algoritma Decesion Tree pre-pruning untuk memperoleh model aturan pohon keputusan menggunakan persamaan $Gini = 1 - \sum_{i=1}^{j} P(i)^2$.

Berikut ditampilkan *pseudocode* dan visualisasi Treenya.

```
  # Visualize the pre-pruned decision tree
  plt.figure(figsize=(50,30))
plot_tree(clf_pre_pruning,
feature_names=X.columns,
class_names=['Normal', 'HF'], filled=True)
plt.title('Pre-Pruned                Decision
Tree',fontsize=40)
plt.show()
```

Berikut tampilan visualisasi decision tree pre-pruned yang dihasilkan dari komputasi menggunakan python.



Gambar 6. Visualisasi Klasifikasi *DecesionTree Prepruning*

Gambar 6 merupakan visualisasi decision tree pre-pruned yang ditampilkan menunjukkan struktur pohon keputusan yang digunakan untuk memprediksi kelas berdasarkan beberapa fitur, seperti ChestPainType, ST Slope, Oldpeak, Sex, MaxHR, dan Age. Pohon ini memiliki beberapa tingkat percabangan, dengan masing-masing node internal mewakili keputusan berdasarkan nilai suatu fitur, dan masing-masing daun mewakili prediksi kelas akhir. Gini index pada setiap node menunjukkan seberapa baik node tersebut memisahkan kelas yang berbeda; nilai yang lebih rendah menunjukkan pemisahan yang lebih baik. Dalam hal ini, ChestPainType <= 0.5 dan ST Slope <= 1.5 adalah keputusan awal yang paling penting, membagi data menjadi dua cabang utama. Dari visualisasi ini, dapat dilihat bahwa sebagian besar keputusan awal cenderung memisahkan kelas "Normal" dari kelas lainnya dengan cukup baik, namun beberapa node daun masih menunjukkan nilai Gini yang relatif tinggi, yang menunjukkan bahwa beberapa ketidakpastian masih ada dalam klasifikasi akhir, dan mungkin memerlukan penyesuaian lebih lanjut.

Dari gambar 6 berikut ditampilkan prepruning rules:

```
Decision Tree PrePruning in Text Format:
|--- ST_Slope <= 1.50
|   |--- ChestPainType <= 0.50
|   |   |--- MaxHR <= 175.50
|   |   |   |--- Sex <= 0.50
|   |   |   |   |--- Oldpeak <= 1.05
|   |   |   |   |   |--- RestingBP <= 140.00
|   |   |   |   |   |   |--- MaxHR <= 133.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- MaxHR >  133.50
|   |   |   |   |   |   |   |--- RestingBP <=
128.50
```


Received: July 15th, 2024| Published: September 15th, 2024

DOI 10.30656/jsii.v11i2.9161
p-ISSN: 2406-7768
e-ISSN: 2581-2181

```
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- RestingBP >
128.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- RestingBP >  140.00
|   |   |   |   |   |   |   |--- Cholesterol <=
327.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- Cholesterol >
327.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- Oldpeak >  1.05
|   |   |   |   |   |   |--- Age <= 67.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- Age >  67.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- Sex >  0.50
...
|   |   |   |--- class: 0

Number of nodes before pruning: 189
Maximum depth before pruning: 15
```

Berikut hasil hasil *Classification Report* dari pre-pruning dari komputasi python decision tree.

```
Prepruning Accuracy on Training Data:
0.8629283489096573
Pre-Pruned Tree Validation Accuracy:
0.8333333333333334
Pre-Pruned Tree Validation Classification
Report:
              precision    recall  f1-score
support
           0   0.87  0.85   0.86    164
           1   0.78  0.81   0.80    112
accuracy   0.83   276
macro avg  0.83  0.83  0.83  276
weighted avg  0.83  0.83   0.83  276
Prepruning    Accuracy    on    Test    Data:
0.8369565217391305
```

c. Membangun Model *Decesion Tree PostPruning* (*Pruning*)

Setelah itu masuk ketahap Teknik postpruning (pruning), pada subbab ini akan mendapatkan hasil dari penelitian untuk menganalisis teknik *pruning* pada model algoritma *Decision Tree* dapat meningkatkan kinerja dan efisiensi klasifikasi menggunakan Algoritma C4.5. Post-pruning dilakukan dengan menggunakan Reduced Error Pruning (REP) yang tersedia di scikit-learn. Kita akan pertama-tama membangun pohon keputusan tanpa pruning, lalu menerapkan Reduced Error Pruning (REP) pruning untuk memangkas pohon tersebut.

Reduced Error Pruning (REP) adalah teknik pruning yang dilakukan setelah pohon keputusan telah sepenuhnya tumbuh. REP melibatkan pemangkasan cabang pohon dengan memeriksa apakah memangkas cabang tersebut meningkatkan atau setidaknya tidak mengurangi akurasi model pada data validasi.

Cara kerja REP adalah mempertimbangkan setiap simpul atau cabang pohon untuk pemangkasan. Pemangkasan merupakan tahap menghapus subpohon dan menetapkan kelas yang paling dominan muncul di simpul itu. Sebuah simpul akan dihapus jika akurasi pohon yang dihasilkan tidak lebih buruk daripada hasil saat training serta dapat meminimalkan nilai estimasi error. *Node* atau cabang akan dihapus secara iteratif jika dapat meningkatkan akurasi pohon keputusan. Selain itu, proses pemangkasan dilakukan dari bagian bawah menuju bagian atas pohon. Nilai estimasi error merupakan penentu proses *pruning* perlu dilakukan atau tidak. Selama nilai estimasi error pada proses *pruning* lebih kecil atau sama dengan nilai estimasi error sebelumnya maka *proses pruning* dilakukan. Proses tersebut terus dilakukan selama ditemukan nilai estimasi error yang lebih baru dan tidak mengurangi nilai akurasi yang dihasilkan dibandingkan nilai error lama.

Untuk menghitung rata –rata error pada setiap kelas yang terdapat pada sebuah subtree menggunakan:

Rumus : Menghitung rata –rata error pada setiap kelas

$$\text{Total}Error = \sum_{i=1}^{n} \frac{E_i}{N} * e_i$$

Keterangan:
Ei = Jumlah data yang salah terklasifikasi pada leaf node ke-i.
N = Jumlah data sampel.
ei = estimasi kesalahan simpul daun ke-i.

Berikut *pseudocode* build model *Postpruning* :

```
# Function to prune the tree using Reduced
Error Pruning (REP)
def prune_tree(tree, X_val, y_val):
    """
    Prune the tree using Reduced Error Pruning
(REP).
    """
    # Extract the tree attributes
    n_nodes = tree.tree_.node_count
    children_left = tree.tree_.children_left
    children_right = tree.tree_.children_right
    feature = tree.tree_.feature
    threshold = tree.tree_.threshold

    # Recursive function to prune the tree
    def prune_index(node):
        if children_left[node] == -1 and
children_right[node] == -1:
            return
        prune_index(children_left[node])
        prune_index(children_right[node])

        left_child = children_left[node]
        right_child = children_right[node]
        if left_child != -1 and right_child !=
-1:
            original_prediction =
tree.predict(X_val)
            original_accuracy =
accuracy_score(y_val, original_prediction)

            # Backup the original children
            tree.tree_.children_left[node] = -
1
            tree.tree_.children_right[node] =
-1
```


Received: July 15th, 2024| Published: September 15th, 2024

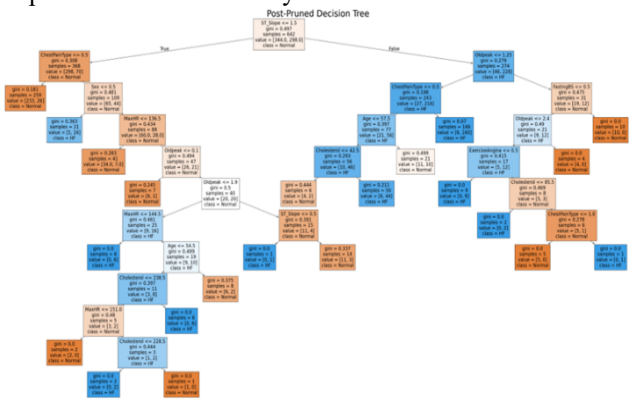
DOI 10.30656/jsii.v11i2.9161
p-ISSN: 2406-7768
e-ISSN: 2581-2181

```
            # Evaluate the pruned tree
            pruned_prediction =
tree.predict(X_val)
            pruned_accuracy =
accuracy_score(y_val, pruned_prediction)

            # Restore the original children if
pruning didn't improve accuracy
            if pruned_accuracy <
original_accuracy:
                tree.tree_.children_left[node]
= left_child
                tree.tree_.children_right[node
] = right_child

    prune_index(0)
    return tree
```

Berikut tampilan visualisasi decision tree postpruned yang dihasilkan dari komputasi menggunakan python. Berikut ditampilkan visualisasi Treenya.



Gambar 7. Visualisasi Klasifikasi DecesionTree Postpruning yang dihasilkan dari Komputasi Menggunakan Python.

Gambar 7 merupakan visualisasi decision tree post-pruned ini menunjukkan struktur pohon keputusan yang telah dioptimalkan dengan pemangkasan untuk mengurangi kompleksitas dan meningkatkan generalisasi model. Dibandingkan dengan versi pre-pruned, pohon ini memiliki lebih sedikit cabang dan node, yang mengindikasikan bahwa beberapa keputusan atau percabangan yang tidak terlalu signifikan telah dihilangkan. Keputusan awal utama tetap sama, yaitu berdasarkan fitur ChestPainType <= 0.5 dan ST Slope <= 1.5, namun cabang-cabang lebih lanjut yang mempertimbangkan fitur seperti MaxHR, Oldpeak, Age, dan Cholesterol telah mengalami pengurangan, dengan node-node yang tersisa menunjukkan nilai Gini yang lebih rendah, menandakan pemisahan kelas yang lebih bersih. Secara keseluruhan, pohon ini lebih sederhana dan lebih fokus, yang seharusnya meningkatkan performa model pada data yang tidak terlihat sebelumnya dengan mengurangi overfitting.

Dari gambar 7, berikut ditampilkan postpruning rules:

```
|--- ST_Slope <= 1.50
|   |--- ChestPainType <= 0.50
|   |   |--- Sex <= 0.50
|   |   |   |--- Oldpeak <= 1.05
|   |   |   |   |--- class: 0
|   |   |   |--- Oldpeak >  1.05
|   |   |   |   |--- class: 0
|   |   |--- Sex >  0.50
|   |   |   |--- MaxHR <= 147.50
|   |   |   |   |--- class: 0
|   |   |   |--- MaxHR >  147.50
|   |   |   |   |--- class: 0
|   |--- ChestPainType >  0.50
|   |   |--- Sex <= 0.50
|   |   |   |--- class: 1
|   |   |--- Sex >  0.50
|   |   |   |--- MaxHR <= 136.50
|   |   |   |   |--- class: 0
|   |   |   |--- MaxHR >  136.50
|   |   |   |   |--- class: 0
|--- ST_Slope >  1.50
|   |--- Oldpeak <= 1.25
|   |   |--- ChestPainType <= 0.50
|   |   |   |--- Age <= 57.50
|   |   |   |   |--- class: 1
...
|   |   |   |--- class: 1
|   |   |--- ExerciseAngina >  0.50
|   |   |   |--- class: 0
```

Kemudian berikut hasil hasil *Classification Report* dari postpruning (pruning) dari komputasi python decision tree.

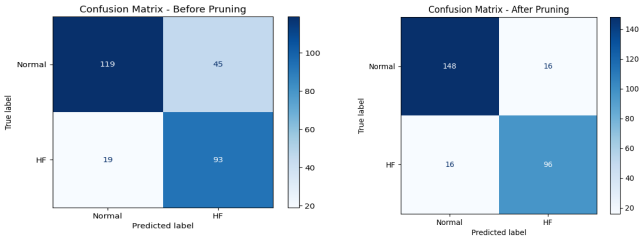
```
Postpruning Accuracy on Training Data:
0.8707165109034268
Post-Pruned Tree Validation Accuracy:
0.8840579710144928
Post-Pruned Tree Validation Classification Report:
              precision    recall  f1-score
support
           0       0.90      0.90      0.90
164
           1       0.86      0.86      0.86
112
    accuracy                           0.88
276
   macro avg       0.88      0.88      0.88
276
weighted avg       0.88      0.88      0.88
276
Postpruning Accuracy on Test Data:
0.8623188405797102
```

B. Pembahasan

Dalam penelitian ini menggunakan beberapa faktor penyebab gagal jantung. yaitu Age, Sex, ChestPainType, RestingBP, Cholesterol, FastingBS, RestingECG, MaxHR, ExerciseAngina, Oldpeak, ST_Slope, HeartDisease. Penelitian ini menggunakan data Kaggle yaitu pada situs website https://www.kaggle.com/datasets/arezaei81/heartcsv, semakin banyak data yang digunakan maka akan semakin bagus juga hasil yang akan diperoleh.

Received: July 15th, 2024| Published: September 15th, 2024

DOI 10.30656/jsii.v11i2.9161
p-ISSN: 2406-7768
e-ISSN: 2581-2181



a. Pre-pruning b. Postpruning

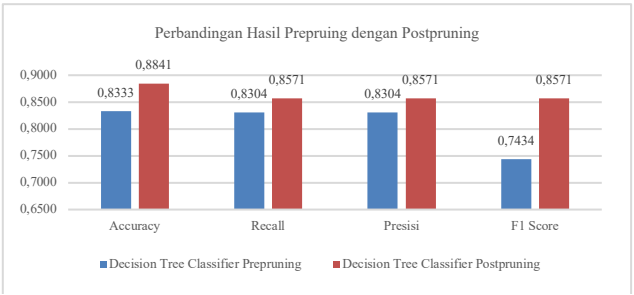
Gambar 8. Confusion matrix sebelum *pruning* dan setelah *pruning*

Hasil penelitian menunjukkan bahwa penggunaan postpruning(pruning) pada decision tree dengan split index Gini memberikan kinerja yang lebih baik dibandingkan dengan decision tree yang standart (prepruning). Secara spesifik, kinerja model dalam hal akurasi, recall, precision, dan F1 score menunjukkan peningkatan signifikan ketika menggunakan Gini.

Tabel 3. Hasil Perbandingan Decesion Tree menggunakan Pruning dan nonpruning

Metric	Decision Tree Classifier	
	Prepruning	*Postpruning*
Accuracy	0,8333	0,8841
Recall	0,8304	0,8571
Presisi	0,8304	0,8571
F1 Score	0,7434	0,8571

Pada table 3 merupakan hasil dari perbandingan perhitungan yang telah dilakukan **prepruning** dan **pruning(postpruning),** model Decision Tree Classifier menunjukkan peningkatan pada semua metrik evaluasi utama. Akurasi meningkat dari 0.8333 menjadi 0.8841, menunjukkan peningkatan proporsi prediksi yang benar. Recall meningkat dari 0.8304 menjadi 0.8571, mengindikasikan bahwa model lebih baik dalam mengidentifikasi kasus positif yang sebenarnya dan mengurangi false negatives. Presisi juga meningkat dari 0.8304 menjadi 0.8571, menandakan model lebih akurat dalam memprediksi kelas positif dengan benar dan mengurangi false positives. F1 Score meningkat signifikan dari 0.7434 menjadi 0.8571, mencerminkan keseimbangan yang lebih baik antara presisi dan recall setelah pruning. Secara keseluruhan, pruning berhasil meningkatkan kinerja model dengan mengurangi overfitting dan membuat prediksi yang lebih akurat pada data baru.



Gambar 9. Grafik Perbandingan Klasifikasi Decesion Tree Prepruning dengan Postpruning

Pada gambar 9 Secara keseluruhan, hasil penelitian ini menunjukkan bahwa penggunaan teknik pruning(postpruning) pada decision tree dengan split index Gini memberikan hasil yang lebih baik dibandingkan dengan prepruning (decision tree standart) dalam semua metrik kinerja yang diukur. Dengan demikian, pruning(postpruning) dapat dianggap sebagai metode yang lebih efektif dalam membangun model decision tree yang lebih akurat dan andal.

IV. KESIMPULAN

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa penerapan teknik pruning pada Decision Tree Classifier, baik prepruning maupun postpruning, secara signifikan meningkatkan performa model dalam mengklasifikasikan dataset penyakit jantung dari Kaggle. Dari hasil evaluasi, model dengan postpruning menunjukkan kinerja terbaik dengan nilai akurasi sebesar 0,8841, recall 0,8571, presisi 0,8571, dan F1 score 0,8571. Sebagai perbandingan, model dengan prepruning memiliki akurasi 0,8333, recall 0,8304, presisi 0,8304, dan F1 score 0,7434. Penggunaan pruning atau post pruning dalam penelitian artikel ini memiliki hasil meningkatan nilai metrik yang menunjukkan bahwa postpruning lebih efektif dalam mengurangi overfitting dan meningkatkan kemampuan generalisasi model. Dengan demikian, teknik postpruning dapat dianggap sebagai metode yang lebih unggul dalam mengoptimalkan performa Decision Tree Classifier untuk dataset ini. Yang menyebabkan peningkatan nilai metrik ini adalah penggunaan pruning atau post pruning dalam

V. SARAN

Berdasarkan hasil penelitian ini, disarankan untuk menerapkan teknik postpruning pada model Decision Tree dalam berbagai kasus klasifikasi untuk mengurangi overfitting dan meningkatkan kemampuan generalisasi model. Peneliti di masa depan sebaiknya mengeksplorasi parameter pruning yang lebih beragam dan menggabungkan teknik ini dengan metode optimasi lain seperti ensemble learning atau fitur seleksi untuk mencapai performa yang lebih baik. Selain itu, penelitian lebih lanjut perlu dilakukan pada berbagai jenis dataset untuk menguji keefektifan teknik pruning dalam konteks yang berbeda, serta mempertimbangkan implementasi pada aplikasi nyata untuk memberikan kontribusi yang lebih signifikan dalam bidang medis dan diagnosis penyakit.

REFERENSI

[1] M. Minarni, E. I. Sari, A. Syahrani, and P. Mandarani, "Klasterisasi Penyakit Menggunakan Algoritma K-Medoids pada Dinas Kesehatan Kabupaten Agam," *J. Nas. Pendidik. Tek. Inform.*, vol. 10, no. 3, p. 137, 2021, doi: 10.23887/janapati.v10i3.34904.

[2] L. Hao, "Research on parallel association rule mining of big data based on an improved K-means clustering algorithm," *Int. J. Auton. Adapt. Commun. Syst.*, vol. 16, no. 3, pp. 233–247, 2023, doi: 10.1504/IJAACS.2023.131622.

[3] R. R. Damanik and M. H. Poernomo, "Prediksi Pembelian Barang Pada Distributor Lampu

Received: July 15th, 2024| Published: September 15th, 2024

DOI 10.30656/jsii.v11i2.9161
p-ISSN: 2406-7768
e-ISSN: 2581-2181

Menggunakan Metode Apriori pada PT. XYZ," *JDMIS J. Data Min. ...*, 2023, [Online]. Available: https://journal.yp3a.org/index.php/jdmis/article/view/1500

[4] A. S. Ritonga and I. Muhandhis, "Teknik Data Mining Untuk Mengklasifikasikan Data Ulasan Destinasi Wisata Menggunakan Reduksi Data Principal Component Analysis (Pca)," *Edutic - Sci. J. Informatics Educ.*, vol. 7, no. 2, 2021, doi: 10.21107/edutic.v7i2.9247.

[5] S. Defit, A. P. Windarto, and P. Alkhairi, "Comparative Analysis of Classification Methods in Sentiment Analysis: The Impact of Feature Selection and Ensemble Techniques Optimization," *Telematika*, vol. 17, no. 1, pp. 52–67, 2024.

[6] A. P. Windarto, I. R. Rahadjeng, M. N. H. Siregar, and P. Alkhairi, "Deep Learning to Extract Animal Images With the U-Net Model on the Use of Pet Images," *J. MEDIA Inform. BUDIDARMA*, vol. 8, no. 1, pp. 468–476, 2024.

[7] A. Prasetio, "Simulasi Penerapan Metode Decision Tree (C4.5) Pada Penentuan Status Gizi Balita," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 4, no. 3, pp. 209–214, 2021, doi: 10.32672/jnkti.v4i3.2983.

[8] A. K. Wardhani, E. Nugraha, and ..., "Optimization of the Decision Tree Method using Pruning on Liver Disease Classification," *J. Appl. ...*, 2022, [Online]. Available: https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/4350

[9] P. Rao, "Weight pruning-UNet: Weight pruning UNet with depth-wise separable convolutions for semantic segmentation of kidney tumors," *J. Med. Signals Sens.*, vol. 12, no. 2, pp. 108–113, 2022, doi: 10.4103/jmss.jmss_108_21.

[10] M. S. Hasibuan and Suhardi, "Analisis Sentimen Kebijakan Vaksin Covid-19 Menggunakan SVM dan C4.5," *J. Tek. Elektro Dan Komput. TRIAC*, pp. 19–21, 2022.

[11] K. F. Irnanda, D. Hartama, and A. P. Windarto, "Analisa Klasifikasi C4.5 Terhadap Faktor Penyebab Menurunnya Prestasi Belajar Mahasiswa Pada Masa Pandemi," *J. Media Inform. Budidarma*, vol. 5, no. 1, p. 327, 2021, doi: 10.30865/mib.v5i1.2763.

[12] M. M. Mijwil and R. A. Abttan, "Utilizing the genetic algorithm to pruning the C4. 5 decision tree algorithm," *Asian J. Appl. Sci.*, 2021, [Online]. Available: https://www.researchgate.net/profile/Maad-Mijwil/publication/349634676_Utilizing_the_Genetic_Algorithm_to_Pruning_the_C45_Decision_Tree_Algorithm/links/6038f64ca6fdcc37a8544bff/Utilizing-the-Genetic-Algorithm-to-Pruning-the-C45-Decision-Tree-Algorithm.pdf

Received: July 15th, 2024| Published: September 15th, 2024