Penerapan Strategi Greedy dalam Pembentukan Pohon Keputusan pada Algoritma Random Forest

Raudhah Yahya Kuddah - 13122003
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: raudhahyk@gmail.com , 13122003@mahasiswa.itb.ac.id

***Abstract*— Algoritma *Random Forest* merupakan metode *ensemble learning* yang terdiri dari sejumlah pohon keputusan (*decision tree*) yang dibangun secara independen dan digabungkan untuk menghasilkan prediksi yang lebih akurat dan stabil. Proses pembentukan setiap pohon keputusan dalam Random Forest memanfaatkan strategi *greedy*, yaitu pendekatan heuristik yang memilih atribut terbaik secara lokal pada setiap node berdasarkan kriteria tertentu seperti *information gain* atau *Gini index*. Pendekatan ini memprioritaskan efisiensi dalam membagi data pada setiap cabang tanpa mempertimbangkan dampak global dari pemilihan tersebut. Makalah ini membahas bagaimana strategi greedy diimplementasikan dalam pembentukan pohon keputusan, mengevaluasi kelebihan dan keterbatasannya dalam konteks Random Forest, serta pengaruhnya terhadap performa model secara keseluruhan. Dengan meninjau proses algoritmik dan struktur internal Random Forest, makalah ini bertujuan memberikan pemahaman lebih mendalam mengenai kontribusi strategi greedy dalam menghasilkan model klasifikasi yang efektif dan efisien.**

Kata Kunci*— *Random Forest; Pohon Keputusan; Strategi Greedy; Information Gain; Gini Index; Ensemble Learning; Algoritma Klasifikasi.

I. PENDAHULIUAN

A. Latar Belakang

Dalam era digital saat ini, data telah menjadi aset yang sangat penting bagi berbagai bidang, mulai dari bisnis, kesehatan, hingga teknologi. Untuk menggali nilai dari data tersebut, dibutuhkan metode yang efektif dalam melakukan analisis dan pengambilan keputusan. Salah satu pendekatan yang banyak digunakan dalam bidang machine learning adalah algoritma klasifikasi, yang bertujuan memetakan input data ke dalam kelas-kelas tertentu.

Algoritma Random Forest merupakan salah satu metode klasifikasi yang populer dan banyak digunakan karena kemampuannya menghasilkan model yang akurat dan tahan terhadap overfitting. Random Forest bekerja dengan membangun sekumpulan pohon keputusan (decision trees) dan menggabungkan hasil prediksinya melalui voting atau rata-rata. Setiap pohon dalam Random Forest dibentuk melalui proses pelatihan pada subset data yang diambil secara acak, baik dari segi data maupun atribut. Hal ini menjadikan Random Forest sebagai algoritma berbasis ensemble learning yang kuat.

Dalam proses pembentukan masing-masing pohon keputusan, digunakan suatu strategi pencarian yang bersifat greedy. Strategi greedy berarti bahwa pada setiap langkah pembentukan pohon, algoritma akan memilih atribut terbaik untuk memisahkan data pada titik tersebut, tanpa mempertimbangkan konsekuensi dari pemilihan tersebut terhadap langkah-langkah selanjutnya. Keputusan tersebut didasarkan pada metrik tertentu seperti Information Gain atau Gini Index, yang mengukur seberapa baik suatu atribut dapat memisahkan data berdasarkan label kelasnya.

Meskipun pendekatan greedy tidak menjamin solusi global yang optimal, pendekatan ini terbukti sangat efisien dan efektif dalam konteks pembentukan pohon keputusan. Oleh karena itu, pemahaman tentang bagaimana strategi greedy diterapkan dalam pembentukan pohon keputusan sangat penting, khususnya dalam algoritma Random Forest. Penelitian ini bertujuan untuk mengeksplorasi lebih lanjut peran strategi greedy dalam membentuk struktur internal pohon keputusan serta dampaknya terhadap performa model secara keseluruhan.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam makalah ini dapat dirumuskan sebagai berikut:

- Bagaimana strategi greedy diterapkan dalam proses pembentukan pohon keputusan pada algoritma Random Forest?
- Apa peran metrik seperti Gini Index dan Information Gain dalam pengambilan keputusan secara greedy?
- Apa dampak penerapan strategi greedy terhadap struktur pohon keputusan dan performa keseluruhan model Random Forest?
- Apakah terdapat keterbatasan atau potensi perbaikan dari penggunaan strategi greedy dalam algoritma Random Forest?

C. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk:

- Menjelaskan konsep strategi greedy dan bagaimana penerapannya dalam proses pembentukan pohon keputusan.
- Menganalisis peran metrik pemilihan atribut seperti Gini Index dan Information Gain dalam strategi greedy pada Random Forest.
- Mengkaji pengaruh penerapan strategi greedy terhadap struktur pohon dan performa klasifikasi model Random Forest.
- Mengidentifikasi kelebihan dan keterbatasan pendekatan greedy dalam konteks pembelajaran mesin berbasis pohon keputusan.
- Memberikan pemahaman yang lebih mendalam mengenai proses internal pembentukan model Random Forest yang efisien dan akurat.

II. DASAR TEORI

A. Pohon Keputusan (Decision Tree)

Pohon keputusan merupakan salah satu algoritma pembelajaran mesin yang paling dasar namun sangat penting, khususnya dalam tugas klasifikasi dan regresi. Algoritma ini bekerja dengan cara membuat struktur pohon yang terdiri dari serangkaian keputusan berbasis atribut dalam data, hingga mencapai hasil akhir berupa prediksi kelas atau nilai numerik. Dalam struktur pohon keputusan, terdapat beberapa komponen utama. Node pertama dalam pohon disebut root node, yang menjadi titik awal pemisahan data. Selanjutnya, terdapat internal node yang mewakili kondisi keputusan berdasarkan atribut tertentu, serta leaf node (node daun) yang menunjukkan hasil akhir prediksi. Hubungan antara node-node tersebut dibentuk melalui cabang (branch), yang masing-masing merepresentasikan hasil dari pengujian kondisi pada atribut terkait.

Proses pembentukan pohon keputusan dimulai dengan memilih atribut terbaik sebagai pemisah pada root node. Pemilihan atribut ini dilakukan berdasarkan kriteria pemisahan tertentu seperti Information Gain, Gini Index, atau Gain Ratio, yang mengukur seberapa baik suatu atribut dalam memisahkan data menjadi kelompok yang homogen. Setelah atribut dipilih, data dibagi ke dalam subset berdasarkan nilai dari atribut tersebut. Setiap subset kemudian diproses secara rekursif untuk membentuk sub-pohon dengan mekanisme yang sama hingga tercapai kondisi berhenti, seperti ketika semua data dalam sebuah node berasal dari kelas yang sama, tidak ada atribut yang tersisa, atau ukuran data dalam node terlalu kecil untuk dilanjutkan. Agar tidak menghasilkan pohon yang terlalu kompleks dan overfitting terhadap data pelatihan, teknik pruning (pemangkasan) sering digunakan setelah pohon terbentuk. Pruning bertujuan menghilangkan cabang-cabang yang kurang memberikan kontribusi signifikan terhadap akurasi model.

Pohon keputusan memiliki sejumlah keunggulan. Model ini mudah dipahami dan divisualisasikan, bahkan oleh pengguna non-teknis. Selain itu, algoritma ini dapat menangani data kategorikal maupun numerik, serta tidak memerlukan banyak pra-pemrosesan seperti normalisasi atau standarisasi data. Namun demikian, pohon keputusan juga memiliki beberapa kelemahan. Model ini cenderung mudah mengalami overfitting, terutama jika pohon tumbuh terlalu dalam tanpa kontrol. Selain itu, pohon keputusan bersifat sangat sensitif terhadap perubahan kecil pada data pelatihan dan bergantung pada pemilihan atribut yang dilakukan secara lokal atau greedy, sehingga hasil akhirnya tidak selalu optimal secara global [4].

Sebagai ilustrasi sederhana, misalkan kita ingin memprediksi apakah seseorang akan membeli laptop berdasarkan dua atribut: usia dan pendapatan. Pohon keputusan dapat bekerja dengan pertama-tama membagi data berdasarkan kriteria seperti "usia < 30", kemudian membagi lagi berdasarkan tingkat pendapatan, hingga akhirnya menghasilkan kesimpulan seperti "ya" atau "tidak" di node daun. Pendekatan logis dan berlapis inilah yang membuat pohon keputusan menjadi alat yang sangat kuat dalam pengambilan keputusan berbasis data.

B. Machine Learning

Pembelajaran mesin (machine learning) merupakan cabang dari kecerdasan buatan (artificial intelligence) yang berfokus pada pengembangan algoritma dan model statistik yang memungkinkan sistem komputer untuk belajar dari data dan membuat prediksi atau keputusan tanpa perlu diprogram secara eksplisit. Dalam pembelajaran mesin, sebuah sistem dilatih menggunakan data historis agar mampu mengenali pola, membuat generalisasi, dan menerapkan hasil pelatihan tersebut pada data baru yang belum pernah dilihat sebelumnya. Proses pembelajaran ini melibatkan tiga komponen utama: data masukan (fitur), algoritma pembelajaran, dan target output yang ingin diprediksi.

Machine learning secara umum dibagi menjadi tiga kategori utama: supervised learning, unsupervised learning, dan reinforcement learning. Pada supervised learning, algoritma dilatih menggunakan dataset berlabel, yaitu data yang sudah diketahui hasilnya, seperti pada tugas klasifikasi atau regresi. Contoh penggunaannya termasuk pengenalan wajah, deteksi email spam, dan prediksi harga rumah. Dalam unsupervised learning, data yang digunakan tidak memiliki label, dan algoritma bertugas untuk menemukan pola atau struktur tersembunyi dalam data, seperti pada klasterisasi atau reduksi dimensi. Sementara itu, reinforcement learning melibatkan agen yang belajar melalui interaksi dengan lingkungan untuk mencapai tujuan tertentu, seperti dalam permainan atau pengendalian robot.

Salah satu pendekatan yang sangat umum dan kuat dalam supervised learning adalah penggunaan model pohon keputusan (decision tree), baik sebagai model tunggal maupun dalam bentuk gabungan seperti Random Forest. Algoritma seperti

Random Forest memanfaatkan kekuatan pembelajaran mesin untuk menghasilkan prediksi yang akurat dengan tetap mempertahankan interpretabilitas dan efisiensi. Oleh karena itu, memahami konsep dasar pembelajaran mesin menjadi landasan penting sebelum mempelajari algoritma-algoritma klasifikasi yang lebih kompleks, termasuk strategi greedy dalam pembentukan pohon keputusan.

C. *Strategi Greedy*

Strategi greedy merupakan pendekatan algoritmik yang banyak digunakan dalam pemecahan masalah optimisasi, termasuk dalam pembentukan pohon keputusan. Secara umum, strategi greedy bekerja dengan cara membuat keputusan terbaik pada setiap langkah berdasarkan informasi lokal yang tersedia, tanpa mempertimbangkan konsekuensi jangka panjang atau keseluruhan solusi global. Keputusan yang diambil bersifat "rakus" karena selalu memilih solusi yang tampak paling optimal saat itu. Meskipun strategi ini tidak menjamin bahwa solusi akhir adalah yang terbaik secara keseluruhan, dalam banyak kasus strategi greedy menghasilkan solusi yang cukup baik dengan efisiensi komputasi yang tinggi.

Dalam konteks pembentukan pohon keputusan, strategi greedy digunakan untuk memilih atribut terbaik yang akan digunakan sebagai pemisah (split) pada setiap node pohon. Pada setiap langkah pembentukan, algoritma mengevaluasi semua atribut yang tersedia dan memilih atribut yang menghasilkan pemisahan data paling baik menurut kriteria tertentu, seperti Information Gain atau Gini Index. Pemilihan ini dilakukan tanpa mempertimbangkan dampaknya terhadap pembagian data di tingkat-tingkat node selanjutnya. Dengan kata lain, algoritma tidak melakukan pencarian ke seluruh kemungkinan struktur pohon, melainkan hanya fokus pada keputusan terbaik secara lokal untuk saat itu.

Keunggulan utama dari strategi greedy adalah kesederhanaan dan kecepatannya, yang membuatnya sangat cocok untuk digunakan dalam algoritma pohon keputusan seperti ID3, C4.5 [2], dan CART. Dalam algoritma Random Forest, strategi greedy tetap digunakan dalam proses pembentukan masing-masing pohon secara individual, meskipun data dan atribut yang digunakan dibatasi secara acak. Pendekatan ini memungkinkan setiap pohon untuk dibangun secara efisien, meskipun dengan ruang pencarian yang terbatas. Kombinasi antara strategi greedy dan ensembel acak inilah yang memberikan kekuatan utama dari Random Forest: menghasilkan model yang cepat, stabil, dan memiliki generalisasi yang baik terhadap data yang belum pernah dilihat.

Meskipun begitu, pendekatan greedy juga memiliki keterbatasan. Karena hanya mempertimbangkan solusi lokal, strategi ini berpotensi melewatkan pemisahan yang mungkin lebih baik jika dilihat dari perspektif struktur pohon secara keseluruhan. Hal ini membuka peluang untuk penelitian lanjutan, seperti pengembangan pendekatan yang mampu mengevaluasi dampak keputusan saat ini terhadap hasil akhir pohon atau penggunaan strategi alternatif seperti lookahead, beam search, atau algoritma berbasis optimisasi global.

D. *Gini Index dan Information Gain*

Dalam pembentukan pohon keputusan, proses pemilihan atribut terbaik untuk membagi data di setiap node sangat krusial. Untuk menentukan atribut mana yang paling baik digunakan sebagai pemisah, digunakan ukuran atau metrik tertentu yang mengukur kualitas pemisahan tersebut. Dua metrik yang paling umum digunakan dalam algoritma pohon keputusan adalah Gini Index dan Information Gain. Keduanya digunakan sebagai dasar dalam strategi greedy untuk membuat keputusan optimal secara lokal pada setiap langkah pembentukan pohon.

Gini Index, juga dikenal sebagai Gini Impurity, adalah metrik yang mengukur ketidakhomogenan suatu himpunan data. Nilai Gini akan bernilai nol jika seluruh elemen dalam himpunan tersebut berasal dari satu kelas yang sama, artinya node tersebut bersifat "murni". Sebaliknya, semakin tinggi nilai Gini, semakin campuran data dalam node tersebut. Rumus dari Gini Index adalah:

$$Gini(t) = 1 - \sum_{i=1}^{c} p_i^2$$

Yang mana p_i merupakan proporsi data dari kelas ke-i di node tersebut, dan C adalah jumlah total kelas. Saat pohon dibangun, algoritma akan memilih atribut yang meminimalkan nilai Gini setelah pemisahan, karena semakin kecil nilai Gini, semakin baik kemurnian hasil pemisahan tersebut.

Sementara itu, Information Gain merupakan metrik yang berasal dari teori informasi dan digunakan dalam algoritma seperti ID3 dan C4.5. Metrik ini mengukur pengurangan ketidakpastian (entropi) setelah data dibagi menggunakan atribut tertentu. Entropi menggambarkan tingkat ketidakpastian dalam kumpulan data, dan Information Gain mengukur seberapa besar penurunan entropi yang terjadi akibat pembagian data tersebut. Rumus Information Gain adalah:

$$IG(S, A) = Entropy\ (S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|}\ .\ Entropy\ (S_v)$$

Yang mana S adalah himpunan data awal, A adalah atribut yang diuji, dan S_v adalah subset data untuk nilai atribut v. Semakin besar nilai Information Gain, semakin efektif atribut tersebut dalam mengurangi ketidakpastian, dan semakin besar kemungkinannya dipilih sebagai pemisah pada node.

Meskipun keduanya memiliki tujuan yang sama yaitu menemukan atribut yang paling baik untuk memisahkan data pemilihan antara Gini Index dan Information Gain sering bergantung pada algoritma yang digunakan. CART (Classification and Regression Tree) secara default menggunakan Gini Index, sementara ID3 dan C4.5 menggunakan Information Gain atau *Gain Ratio*. Dalam algoritma Random Forest, pemilihan atribut untuk setiap pohon umumnya menggunakan Gini Index karena efisiensinya yang

lebih tinggi dalam perhitungan dan sifatnya yang cenderung menghasilkan pohon yang seimbang.

E. Random Forest

Random Forest adalah algoritma pembelajaran mesin berbasis ensemble learning [1] yang menggabungkan banyak pohon keputusan (decision trees) untuk membentuk model prediktif yang kuat, stabil, dan akurat. Metode ini dikembangkan oleh Leo Breiman sebagai pengembangan dari teknik bagging (bootstrap aggregating) [1], dengan penambahan proses pemilihan atribut secara acak pada setiap pemisahan node. Tujuan utama dari Random Forest adalah untuk mengurangi kelemahan yang dimiliki oleh pohon keputusan tunggal, yaitu kecenderungan terhadap overfitting dan ketidakstabilan terhadap perubahan data.

Pada prinsipnya, Random Forest bekerja dengan membuat banyak pohon keputusan secara independen. Setiap pohon dilatih menggunakan subset data yang diambil secara acak dengan pengembalian dari dataset pelatihan asli (bootstrap sampling). Selain itu, pada setiap node dalam proses pembentukan pohon, hanya sebagian kecil atribut yang dipilih secara acak [3] untuk dipertimbangkan sebagai kandidat pemisah. Hal ini bertujuan untuk meningkatkan keberagaman antar pohon dalam hutan, sehingga ketika hasil dari semua pohon digabungkan (melalui voting mayoritas untuk klasifikasi atau rata-rata untuk regresi), model akhir menjadi lebih general dan tidak mudah terjebak pada noise dari data pelatihan.

Pembentukan masing-masing pohon dalam Random Forest tetap menggunakan strategi greedy, di mana atribut terbaik dipilih secara lokal berdasarkan metrik seperti Gini Index atau Information Gain. Meski demikian, karena setiap pohon dibangun dari kombinasi data dan atribut yang berbeda, keseluruhan model tidak terlalu bergantung pada satu pohon atau satu keputusan lokal tertentu. Hal inilah yang membuat Random Forest sangat tahan terhadap overfitting dibandingkan dengan pohon keputusan tunggal.

Keunggulan utama dari Random Forest antara lain adalah akurasi yang tinggi, kemampuan menangani data berukuran besar dan berdimensi tinggi, serta ketahanannya terhadap data hilang atau outlier. Di samping itu, Random Forest juga menyediakan informasi penting seperti estimasi pentingnya setiap fitur (feature importance) terhadap prediksi akhir. Namun, kekurangannya adalah model yang dihasilkan sulit untuk diinterpretasikan secara langsung karena terdiri dari ratusan bahkan ribuan pohon, sehingga transparansi dan penelusuran keputusan menjadi rendah.

Secara keseluruhan, Random Forest merupakan salah satu algoritma yang sangat efektif dalam berbagai aplikasi klasifikasi dan regresi, serta menjadi salah satu metode yang paling sering digunakan dalam praktik pembelajaran mesin karena keseimbangan antara akurasi dan ketangguhannya terhadap variabilitas data.

III. METODOLOGI

Penelitian ini dilakukan untuk menganalisis dan memahami penerapan strategi greedy dalam proses pembentukan pohon keputusan pada algoritma Random Forest. Metodologi yang digunakan mencakup pendekatan deskriptif dan eksperimental. Pendekatan deskriptif digunakan untuk menguraikan mekanisme internal algoritma Random Forest serta bagaimana strategi greedy bekerja dalam proses pemilihan atribut. Sementara itu, pendekatan eksperimental digunakan untuk menguji pengaruh strategi greedy terhadap performa model dalam klasifikasi dataset tertentu.

Langkah pertama dalam metodologi ini adalah melakukan studi literatur terhadap algoritma Random Forest, strategi greedy, serta metrik evaluasi pemisahan seperti Gini Index dan Information Gain. Studi literatur ini bertujuan untuk membangun pemahaman teoretis mengenai peran masing-masing komponen dalam proses pembentukan pohon.

Selanjutnya, dilakukan implementasi simulasi proses pemilihan atribut terbaik menggunakan bahasa pemrograman C++. Dalam implementasi ini, dibuat dua pendekatan, yaitu strategi greedy dan exhaustive, untuk memilih split terbaik berdasarkan nilai Gini Index. Dataset yang digunakan bersifat sintetis, dibangkitkan secara acak, sehingga hasil dapat dikendalikan dan diuji dalam skenario yang terstandarisasi.

Selama proses pelatihan, dilakukan logging terhadap atribut yang terpilih di tiap node dan nilai Gini atau Information Gain yang digunakan untuk pemilihan tersebut. Informasi ini kemudian digunakan untuk melakukan analisis lebih lanjut mengenai bagaimana keputusan lokal (greedy) memengaruhi struktur pohon secara keseluruhan, termasuk kedalaman pohon dan distribusi label pada setiap node daun.

Evaluasi dilakukan dengan membandingkan performa dua strategi berdasarkan waktu eksekusi dan jumlah evaluasi split yang dibutuhkan dalam menentukan pemisahan terbaik pada satu node. Dengan cara ini, efisiensi dan efektivitas dari pendekatan greedy dapat diukur secara kuantitatif.

Metodologi ini diharapkan dapat memberikan pemahaman menyeluruh mengenai penerapan strategi greedy dalam algoritma Random Forest, baik dari segi teoritis maupun praktis, serta dampaknya terhadap performa dan struktur model.

IV. IMPLEMENTASI DAN ANALISIS ALGORITMA

A. Penjelasan Algoritma Random Forest

Random Forest merupakan algoritma pembelajaran mesin berbasis ensemble yang menggabungkan sejumlah pohon keputusan (decision trees) untuk meningkatkan akurasi prediksi dan ketahanan terhadap overfitting. Berbeda dengan model pohon tunggal yang rentan terhadap fluktuasi data dan bias lokal, Random Forest membentuk banyak pohon secara independen dan menggabungkan hasilnya melalui proses agregasi. Dengan pendekatan ini, kesalahan yang mungkin muncul dari satu pohon dapat dikompensasi oleh pohon-pohon lainnya, sehingga menghasilkan model yang lebih stabil dan andal.

Proses kerja Random Forest terdiri dari tiga tahap utama. Pertama adalah bootstrap sampling, di mana untuk setiap pohon,

algoritma mengambil subset data pelatihan secara acak dengan pengembalian (sampling with replacement). Hal ini menciptakan variasi antar pohon dalam hal data pelatihan, yang penting untuk mengurangi korelasi antar pohon. Tahap kedua adalah pembentukan pohon, di mana setiap pohon dilatih menggunakan subset data yang telah diambil. Pada tahap ini, di setiap node pohon, hanya sebagian kecil fitur yang dipilih secara acak untuk dipertimbangkan sebagai kandidat pemisah (biasanya √d fitur dari total d fitur), dengan tujuan menambah keragaman model. Untuk memilih atribut dan threshold terbaik sebagai pemisah, digunakan pendekatan strategi greedy, yaitu memilih opsi terbaik secara lokal pada setiap node berdasarkan metrik seperti Gini Index atau Information Gain. Strategi greedy tidak menjelajahi seluruh kemungkinan split, melainkan hanya mengevaluasi sebagian kandidat dan langsung memilih yang memberi hasil terbaik saat itu, sehingga prosesnya menjadi jauh lebih cepat dan efisien.

Tahap ketiga adalah voting agregasi, di mana hasil prediksi dari seluruh pohon digabungkan untuk membentuk hasil akhir. Pada kasus klasifikasi, digunakan metode majority voting, yaitu kelas yang paling banyak dipilih oleh pohon-pohon akan menjadi hasil prediksi akhir. Sementara pada regresi, digunakan metode averaging terhadap seluruh output numerik dari setiap pohon. Kombinasi dari ketiga tahap ini menjadikan Random Forest sebagai algoritma yang kuat, cepat, dan efektif dalam menangani masalah klasifikasi maupun regresi, terutama pada data berskala besar dan berdimensi tinggi.

B. Analisis Kompleksitas Waktu dan Ruang

Algoritma Random Forest terdiri dari banyak pohon keputusan yang dibangun secara independen, sehingga kompleksitas waktu dan ruang dari algoritma ini sangat bergantung pada proses pembentukan masing-masing pohon serta jumlah total pohon yang digunakan. Dalam pembentukan pohon keputusan, strategi greedy digunakan untuk memilih pemisahan terbaik pada setiap node berdasarkan metrik impuritas seperti Gini Index [4]. Analisis kompleksitas berikut ini mempertimbangkan peran strategi greedy dan dampaknya terhadap efisiensi waktu dan penggunaan memori secara keseluruhan.

1) Kompleksitas Waktu

Secara umum, kompleksitas waktu dari algoritma Random Forest dapat dinyatakan sebagai:

$$O(T \,.\, n \,.\, m \,.\, \log n).$$

Yang setiap notasinya dapat didefinisikan sebagai:

- T adalah jumlah pohon dalam hutan
- n adalah jumlah sampel pada setiap pohon yang berasal dari bootstrap sampling.
- m adalah jumlah fitur yang dipilih secara acak di setiap node
- $\log n$ menggambarkan kedalaman rata-rata pohon

Pada setiap node, strategi greedy hanya mengevaluasi m fitur, masing-masing dengan sejumlah kandidat threshold yang terbatas. Ini menjadikan proses pemilihan split jauh lebih cepat dibandingkan pendekatan exhaustive yang dapat memiliki kompleksitas waktu hingga O(T·n·d·k), dengan d adalah jumlah total fitur dan k adalah banyaknya threshold potensial per fitur. Dalam praktiknya, greedy memungkinkan waktu pelatihan berkurang secara signifikan tanpa penurunan akurasi yang besar.

2) Kompleksitas Ruang

Kompleksitas ruang Random Forest ditentukan oleh penyimpanan semua pohon yang dibentuk. Jika setiap pohon memiliki l node, maka ruang yang dibutuhkan oleh satu pohon adalah $O\ (l)$. Karena ada T pohon, maka total kompleksitas ruang menjadi:

$$O(T.l)$$

Faktor l bergantung pada kedalaman pohon, struktur data, serta tingkat impuritas data. Random Forest cenderung menghasilkan pohon yang relatif dalam karena tidak dilakukan pemangkasan (pruning), namun ukuran setiap pohon tetap dibatasi oleh parameter seperti max_depth, min_samples_split, atau max_leaf_nodes. Selain pohon itu sendiri, memori tambahan diperlukan untuk menyimpan indeks data hasil bootstrap, subset fitur, serta struktur split di tiap node.

Strategi greedy membantu menjaga kompleksitas ruang tetap terkendali karena hanya sebagian kecil fitur yang perlu disimpan dan diproses di setiap node, dibandingkan dengan strategi exhaustive yang membutuhkan alokasi memori untuk evaluasi banyak kandidat split secara serentak.

V.

A. Analisis Performa dan Kompleksitas

Untuk mengevaluasi efisiensi strategi greedy dalam pemilihan atribut pada pembentukan pohon keputusan, dilakukan sebuah eksperimen sederhana menggunakan bahasa pemrograman C++ [5]. Program ini dirancang untuk membandingkan dua metode pemilihan split: greedy dan exhaustive search. Dataset yang digunakan dihasilkan secara acak, terdiri dari 1000 data dan 100 fitur, dengan nilai fitur berupa bilangan bulat antara 0 hingga 100 dan label biner (0 atau 1). Dua fungsi pemilihan split diuji: findBestSplitGreedy dan findBestSplitExhaustive. Metrik evaluasi yang digunakan adalah Gini Index, yang umum digunakan dalam algoritma Random Forest sebagai ukuran impuritas.

Pada pendekatan greedy, algoritma memilih satu threshold acak untuk setiap fitur dan mengevaluasi performanya, kemudian memilih split terbaik di antara hasil tersebut. Dengan hanya melakukan satu evaluasi per fitur, jumlah total evaluasi terbatas pada jumlah fitur yang tersedia, yaitu 100. Sebaliknya, pendekatan exhaustive mencoba semua kemungkinan threshold

dari setiap baris data pada semua kolom, menghasilkan 100.000 evaluasi split. Hasil eksekusi program menunjukkan perbedaan performa yang sangat signifikan. Strategi greedy hanya membutuhkan sekitar 89.97 milidetik, sedangkan exhaustive search membutuhkan lebih dari 61.3 detik untuk menyelesaikan proses yang sama.

Selain itu, strategi greedy hanya mengevaluasi 100 kandidat split, sedangkan exhaustive search melakukan 100.000 evaluasi. Ini menunjukkan bahwa pendekatan greedy secara drastis mengurangi beban komputasi, terutama ketika jumlah fitur dan data sangat besar. Meskipun split terbaik yang ditemukan oleh greedy berada di threshold 5 pada kolom 82, sedangkan exhaustive menemukan threshold 4 di kolom yang sama, hasil akhir dari kedua pendekatan tersebut masih berada dalam kisaran yang sangat dekat. Hal ini mendukung argumen bahwa strategi greedy mampu menemukan solusi yang cukup baik (near-optimal) dengan biaya waktu yang jauh lebih rendah.

TABLE I. PERBANDINGAN STRATEGI GREEDY DAN EXHAUSTIVE

Metode	Kolom Terbaik	Treshold	Waktu Eksekusi (ms)	Jumlah Evaluasi Split
Greedy	82	5	89.97	100
Exhaustive	82	4	61316.7	100.000

Dari segi kompleksitas waktu, implementasi greedy memiliki kompleksitas O $(n \cdot m)$, dengan n adalah jumlah data dan m adalah jumlah fitur yang diperiksa. Karena hanya satu threshold diuji per fitur, kompleksitasnya tetap linier terhadap ukuran data dan jumlah fitur. Sebaliknya, pendekatan exhaustive memiliki kompleksitas $O(n^2 \cdot d)$, karena setiap threshold potensial dari setiap baris dievaluasi untuk setiap fitur. Dengan asumsi 1000 baris dan 100 fitur, pendekatan exhaustive memeriksa hingga 100.000 kombinasi, menjadikannya sangat mahal secara waktu dan tidak cocok untuk aplikasi berskala besar seperti Random Forest, yang memerlukan efisiensi tinggi untuk membangun ratusan pohon.

Kompleksitas ruang kedua pendekatan tidak berbeda secara signifikan, karena keduanya hanya menyimpan salinan dataset dan struktur pembantu untuk evaluasi split. Namun, jika dieksekusi secara paralel dalam skala besar, pendekatan exhaustive dapat menyebabkan bottleneck memori dan kelebihan beban komputasi, terutama jika tidak dioptimalkan. Di sisi lain, strategi greedy lebih mudah diparalelkan karena evaluasinya minimal dan independen antar fitur.

Secara keseluruhan, analisis ini menunjukkan bahwa strategi greedy sangat sesuai untuk digunakan dalam pembentukan pohon keputusan di dalam algoritma Random Forest. Meskipun pendekatan ini tidak menjamin bahwa split yang dipilih adalah yang terbaik secara global, eksperimen menunjukkan bahwa hasilnya cukup kompetitif dengan waktu eksekusi yang jauh lebih efisien. Dengan demikian, trade-off antara kualitas split dan waktu eksekusi yang ditawarkan oleh strategi greedy dapat dianggap sepadan, terutama ketika dipakai dalam model ensemble besar yang membutuhkan efisiensi tinggi dalam pelatihan.

VI. KESIMPULAN

Penelitian ini telah membahas dan mengevaluasi penerapan strategi greedy dalam proses pembentukan pohon keputusan pada algoritma Random Forest. Berdasarkan kajian teoritis dan eksperimen yang dilakukan, dapat disimpulkan bahwa strategi greedy menawarkan efisiensi komputasi yang sangat tinggi tanpa mengorbankan kualitas hasil secara signifikan. Strategi ini diterapkan dengan memilih pemisahan terbaik pada setiap node secara lokal, tanpa mempertimbangkan dampak global terhadap struktur pohon secara keseluruhan. Hal ini dilakukan dengan mengevaluasi sebagian kecil kandidat split secara cepat, sehingga sangat cocok digunakan dalam algoritma ensemble berskala besar seperti Random Forest yang membentuk ratusan hingga ribuan pohon.

Dalam proses pengambilan keputusan secara greedy, metrik evaluasi seperti Gini Index digunakan untuk mengukur impuritas dari dataset yang dihasilkan oleh setiap split. Pemisahan dengan nilai Gini terkecil akan dipilih sebagai pemisahan terbaik pada node tersebut. Meskipun pendekatan ini bersifat lokal dan tidak mengevaluasi semua kemungkinan split seperti dalam strategi exhaustive, hasil eksperimen menunjukkan bahwa pemisahan yang diperoleh tetap mendekati optimal dan memberikan hasil klasifikasi yang kompetitif dengan beban komputasi yang jauh lebih rendah.

Eksperimen yang dilakukan menggunakan dataset sintetis dengan 1000 baris dan 100 fitur menunjukkan bahwa strategi greedy hanya membutuhkan waktu kurang dari 0,1 detik dengan 100 evaluasi split, sedangkan metode exhaustive membutuhkan lebih dari 60 detik dan 100.000 evaluasi untuk mencapai hasil yang sedikit lebih baik. Perbedaan performa yang signifikan ini memperkuat keunggulan strategi greedy dalam efisiensi waktu dan sumber daya, serta menjadikannya pilihan yang praktis untuk membangun model dalam lingkungan yang memerlukan komputasi cepat dan skalabilitas tinggi.

Selain keunggulan dalam efisiensi, strategi greedy juga mendukung keragaman antar pohon dalam Random Forest. Karena setiap pohon dibentuk dari subset data dan subset fitur yang berbeda, keputusan lokal yang diambil oleh strategi greedy justru meningkatkan variasi struktur pohon, yang pada akhirnya memperkuat kemampuan generalisasi model secara keseluruhan. Ini menjelaskan mengapa Random Forest tetap mampu memberikan performa klasifikasi yang tinggi meskipun setiap pohon tidak dibentuk dengan cara yang optimal secara global.

Namun demikian, strategi greedy tetap memiliki keterbatasan. Karena hanya mempertimbangkan kondisi lokal pada saat pemisahan, ada kemungkinan model melewatkan pemisahan yang lebih baik jika dilihat dari konteks global. Oleh karena itu, penelitian selanjutnya dapat mengeksplorasi integrasi strategi greedy dengan pendekatan lain, seperti pencarian heuristik atau optimisasi berbasis populasi, untuk meningkatkan akurasi tanpa mengorbankan efisiensi.

Secara keseluruhan, strategi greedy dapat dianggap sebagai pendekatan yang sangat tepat dan seimbang dalam pembentukan pohon keputusan pada algoritma Random Forest. Ia mampu memberikan efisiensi komputasi yang signifikan, hasil pemisahan yang kompetitif, dan mendukung performa model secara keseluruhan, menjadikannya solusi praktis dalam penerapan pembelajaran mesin berskala besar.

VII. SARAN

Berdasarkan hasil analisis dan eksperimen yang telah dilakukan, terdapat beberapa saran yang dapat diajukan untuk pengembangan penelitian selanjutnya. Pertama, meskipun strategi greedy terbukti efisien dan cukup efektif dalam pembentukan pohon keputusan, pendekatan ini masih memiliki keterbatasan dalam hal optimalitas global. Oleh karena itu, saran pertama adalah melakukan eksplorasi terhadap variasi strategi heuristik lainnya, seperti beam search atau evolutionary algorithms, yang mungkin dapat menemukan pemisahan yang lebih baik tanpa mengorbankan efisiensi secara signifikan.

Kedua, penelitian ini masih menggunakan dataset sintetis dan uji coba terbatas dalam skala serta jenis data. Untuk meningkatkan validitas eksternal, pengujian terhadap berbagai dataset nyata dengan karakteristik berbeda (misalnya jumlah fitur yang sangat tinggi, distribusi tidak seimbang, atau data noisy) dapat dilakukan. Hal ini akan memberikan gambaran lebih luas mengenai sejauh mana strategi greedy tetap kompetitif dalam kondisi dunia nyata.

Ketiga, perlu dipertimbangkan integrasi evaluasi metrik tambahan selain waktu dan jumlah evaluasi split, seperti akurasi akhir model, ukuran pohon rata-rata, dan kompleksitas generalisasi, agar analisis performa menjadi lebih holistik. Evaluasi tersebut juga dapat diperluas dengan membandingkan hasil antara Random Forest dengan algoritma klasifikasi ensemble lain seperti Gradient Boosting atau Extra Trees.

Terakhir, dari sisi implementasi, penerapan kode dalam bahasa pemrograman seperti Python dengan pustaka pembelajaran mesin (misalnya scikit-learn) dapat memperkaya analisis dan memungkinkan visualisasi yang lebih baik terhadap proses pemilihan split. Dengan pengembangan-pengembangan ini, diharapkan penelitian sejenis dapat memberikan kontribusi yang lebih besar dalam penerapan algoritma pembelajaran mesin yang efisien dan adaptif.

REFERENCES

[1] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73. https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

[2] J. R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993. https://www.stat.berkeley.edu/~breiman/random-forests.pdf

[3] T. K. Ho, "The random subspace method for constructing decision forests," IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 8, pp. 832–844, Aug. 1998. https://pdfs.semanticscholar.org/b41d/0fa5fdaadd47fc882d3db04277d03fb21832.pdf

[4] G. Biau and E. Scornet, "A Random Forest Guided Tour," ArXiv preprint, Nov. 2015. https://arxiv.org/abs/1511.05741

[5] R. Kuddah. (2025). *Penerapan-Strategi-Greedy-dalam-Pembentukan-Pohon-Keputusan-pada-Algoritma-Random-Forest*. GitHub repository. [Online]. Available: https://github.com/raudhahkuddah/Penerapan-Strategi-Greedy-dalam-Pembentukan-Pohon-Keputusan-pada-Algoritma-Random-Forest

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Juni 2025

Raudhah Yahya Kuddah
13122003