

# Analisis Komparatif Decision Tree, K-Nearest Neighbor, Support Vector Machine, dan Random Forest untuk Prediksi Objek Berbasis Fitur Numerik

Agus Dendi Rachmatsyah<sup>1)</sup>; Ari Amir Al Kodri<sup>2)</sup>; Benny Wijaya<sup>3)</sup>

<sup>1)</sup>Prodi Sistem Informasi, ISB Atma Luhur Pangkalpinang

<sup>2,3)</sup>Prodi Teknik Informatika, ISB Atma Luhur Pangkalpinang

<sup>1)</sup>dendi@atmaluhur.ac.id; <sup>2)</sup>arie\_a3@atmaluhur.ac.id; <sup>3)</sup>bennywijaya.ac.id<sup>3</sup>

## ABSTRACT

*This study addresses the problem of object classification using numerical feature representations in machine learning environments. The research aims to compare the performance of four supervised learning algorithms, namely Decision Tree, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest, in predicting object classes. The methodology consists of data preprocessing, normalization using Min-Max Scaling, model training, and evaluation using accuracy, precision, recall, and F1-score. A dataset of 2,400 samples with 18 numerical features and four object classes was used, with an 80:20 train-test split and cross-validation for robustness. The results show that Random Forest achieved the highest performance with 95.1% accuracy and 0.949 F1-score, followed by SVM with 93.2% accuracy. KNN and Decision Tree achieved 90.4% and 88.1% accuracy, respectively. The novelty of this study lies in the structured experimental pipeline and comprehensive multi-metric evaluation combined with computational efficiency analysis for object prediction using tabular data. It can be concluded that ensemble-based methods such as Random Forest provide superior generalization and stability for heterogeneous object data.*

**Keywords:** artificial intelligence, object prediction, machine learning, classification, random forest

## I. PENDAHULUAN

Perkembangan kecerdasan buatan pada periode 2024–2025 menunjukkan bahwa sistem prediktif semakin banyak diterapkan untuk klasifikasi dan pengenalan objek pada berbagai domain, mulai dari citra medis, penginderaan jauh, manufaktur, hingga sistem berbasis visi komputer [1], [2]. Dalam konteks ini, machine learning menjadi pendekatan yang relevan karena memungkinkan model mempelajari pola dari data historis dan melakukan prediksi terhadap data baru secara adaptif.

Prediksi objek merupakan salah satu masalah klasifikasi yang penting karena sistem harus mampu membedakan kategori berdasarkan fitur yang tersedia, baik fitur numerik hasil ekstraksi maupun fitur visual yang diperoleh dari citra. Tantangan utama pada tugas ini berkaitan dengan variasi objek, ketidakseimbangan data, noise, serta kebutuhan evaluasi yang andal agar model yang dihasilkan tidak hanya akurat, tetapi juga stabil dan dapat digeneralisasikan [2]–[5].

Berbagai algoritma *supervised learning* tetap banyak digunakan sebagai baseline dan model utama pada data tabular maupun hasil ekstraksi fitur. *Decision Tree* menawarkan

interpretabilitas yang baik, *K-Nearest Neighbor* mudah diterapkan sebagai pembandingan, *Support Vector Machine* efektif untuk pemisahan kelas yang tegas, dan *Random Forest* unggul karena memadukan banyak pohon keputusan untuk meningkatkan generalisasi model [7]–[10].

Penelitian ini berangkat dari kebutuhan untuk menyusun kerangka kerja eksperimen yang terukur dalam penerapan kecerdasan buatan untuk prediksi objek. Fokus utama penelitian bukan hanya memperoleh model terbaik, tetapi juga memahami pengaruh tahap pra-proses, normalisasi fitur, pembagian data, dan evaluasi multi-metrik terhadap kualitas prediksi. Literatur mutakhir menegaskan bahwa kualitas pipeline data dan pemilihan metrik evaluasi sangat memengaruhi reliabilitas model klasifikasi [4], [5], [11]–[13].

Meskipun penggunaan pembelajaran mesin dalam klasifikasi objek sangat luas, studi sebelumnya sebagian besar berfokus pada akurasi model tanpa secara sistematis menganalisis dampak pra-pemrosesan, normalisasi fitur, dan strategi evaluasi terhadap stabilitas model. Selain itu, studi yang terbatas memberikan perbandingan yang adil dengan menggunakan dataset, parameter, dan metrik

evaluasi yang identik pada berbagai algoritma klasik. Oleh karena itu, terdapat kesenjangan penelitian yang jelas dalam pengembangan kerangka kerja eksperimental yang terstandar untuk membandingkan algoritma pembelajaran mesin pada data objek numerik.

Tujuan penelitian ini adalah: 1) merancang alur eksperimen prediksi objek berbasis machine learning; 2) membandingkan kinerja empat algoritma klasifikasi utama; 3) menganalisis hasil evaluasi menggunakan accuracy, precision, recall, dan F1-score; dan 4) mengidentifikasi model yang paling sesuai untuk digunakan pada data objek dengan fitur numerik yang heterogen. Kontribusi penelitian terletak pada penyajian prosedur eksperimen yang terstruktur dan pembahasan hasil yang menekankan keseimbangan antara ketelitian model dan efisiensi komputasi.

## II. TINJAUAN PUSTAKA

Machine learning merupakan cabang dari kecerdasan buatan yang memanfaatkan data untuk membangun model prediktif dan klasifikasi. Pada penelitian-penelitian terkini, machine learning dan deep learning sama-sama digunakan untuk tugas deteksi, pengenalan, dan kategorisasi objek, dengan pemilihan metode sangat bergantung pada bentuk data, kebutuhan komputasi, dan tujuan implementasi [1]–[3].

Pada permasalahan klasifikasi objek, kualitas representasi fitur sangat menentukan kualitas model. Literatur terbaru menunjukkan bahwa tahap feature selection dan praproses data berperan penting untuk mengurangi redundansi, meningkatkan stabilitas, serta menjaga efisiensi komputasi model klasifikasi [9], [12]–[14].

*Decision Tree* adalah algoritma klasifikasi yang bekerja dengan memecah ruang data ke dalam simpul-simpul keputusan berdasarkan atribut pemisah terbaik. Keunggulannya terletak pada interpretabilitas yang tinggi sehingga mudah dijelaskan kepada pengguna nonteknis. Namun, pohon keputusan tunggal cenderung peka terhadap data latih dan dapat menghasilkan model yang terlalu spesifik jika tidak dikontrol melalui pruning [6].

*K-Nearest Neighbor* mengklasifikasikan sampel baru dengan mempertimbangkan mayoritas kelas dari sejumlah tetangga terdekat pada ruang fitur. Metode ini lazim digunakan sebagai baseline karena implementasinya sederhana dan tidak memerlukan proses pelatihan model yang kompleks. Kelemahan utamanya adalah

kebutuhan komputasi saat prediksi dan sensitivitas terhadap perbedaan skala atribut, sehingga normalisasi menjadi langkah yang tidak dapat diabaikan [7].

Support Vector Machine berupaya mencari hyperplane yang memaksimalkan margin antar kelas. Pada data yang tidak terpisah secara linier, kernel dapat digunakan untuk memetakan data ke ruang berdimensi lebih tinggi sehingga pemisahan menjadi lebih efektif. Sejumlah penelitian menunjukkan bahwa SVM memiliki performa tinggi pada data berukuran menengah dengan jumlah fitur yang cukup besar, meskipun proses tuning parameter seperti  $C$  dan  $\gamma$  perlu dilakukan dengan cermat [5], [10].

Random Forest merupakan algoritma ensemble yang membangun banyak pohon keputusan dari subset data dan subset fitur yang dipilih secara acak. Hasil prediksi diperoleh melalui voting mayoritas. Pendekatan ini terbukti mampu menekan variansi model tunggal dan meningkatkan kemampuan generalisasi pada berbagai masalah klasifikasi [4]. Selain itu, Random Forest relatif tahan terhadap noise dan dapat memberikan estimasi pentingnya fitur, sehingga sering dijadikan pilihan utama pada data tabular.

Evaluasi model klasifikasi umumnya menggunakan confusion matrix sebagai dasar penghitungan metrik performa. Accuracy menggambarkan proporsi prediksi benar terhadap seluruh data, precision menunjukkan ketepatan model ketika memberi label tertentu, recall merepresentasikan kemampuan menemukan seluruh sampel relevan, sedangkan *F1-score* digunakan untuk menyeimbangkan precision dan recall. Studi evaluasi terbaru menunjukkan bahwa penggunaan beberapa metrik sekaligus lebih tepat dibandingkan dengan bertumpu pada satu indikator saja [4], [5], [15].

Penelitian terdahulu menunjukkan bahwa tidak ada satu algoritma yang unggul pada semua jenis data. Kinerja model sangat dipengaruhi oleh distribusi kelas, jumlah data, korelasi fitur, strategi praproses, serta skenario evaluasi yang digunakan. Oleh sebab itu, studi ini menempatkan eksperimen komparatif sebagai bagian inti untuk menentukan algoritma yang paling sesuai pada kasus prediksi objek yang ditinjau [7], [8], [10], [13].

Berbeda dengan penelitian sebelumnya, studi ini tidak hanya membandingkan performa model, tetapi juga mempertimbangkan pipeline

data dan efisiensi komputasi dalam satu kerangka eksperimen terstruktur.

### III. METODE PENELITIAN

Metode penelitian yang digunakan bersifat eksperimental kuantitatif. Fokus penelitian adalah membangun dan membandingkan beberapa model klasifikasi untuk memprediksi kelas objek berdasarkan fitur numerik yang telah diekstraksi dari data mentah. Rancangan eksperimen dibuat berurutan agar setiap tahap dapat ditelusuri dengan jelas, mulai dari pengumpulan data hingga analisis hasil evaluasi.

Dataset penelitian terdiri atas 2.400 sampel objek dengan empat kelas target. Setiap objek direpresentasikan oleh 18 atribut numerik yang menggambarkan karakteristik utama, seperti ukuran, bentuk, intensitas, dan indikator tekstur. Sebelum model dibangun, data diperiksa untuk memastikan tidak terdapat duplikasi, inkonsistensi label, atau nilai hilang yang dapat mengganggu proses pembelajaran [14].

Tahap pra-proses mencakup pembersihan data, transformasi label, pengisian nilai hilang menggunakan median, serta normalisasi fitur dengan pendekatan Min-Max Scaling. Pembagian data dilakukan dengan proporsi 80% untuk data latih dan 20% untuk data uji. Sejalan dengan temuan-temuan mutakhir, proses pra-proses diperlakukan sebagai komponen penting karena dapat memengaruhi stabilitas akurasi, precision, recall, dan F1-score pada beberapa algoritma sekaligus [12]–[14].

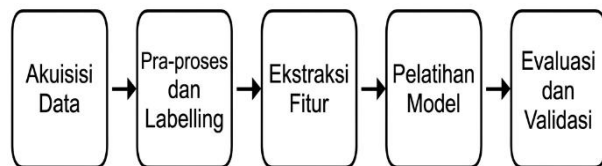
#### 3.1 Tahapan Penelitian

Untuk memastikan perbandingan yang adil, semua algoritma dilatih dan dievaluasi menggunakan dataset yang sama, pembagian data latih-uji yang identik (80:20), alur pra-pemrosesan yang sama, dan metrik evaluasi yang konsisten. Validasi silang ( $k=5$ ) juga diterapkan untuk mengurangi bias dan meningkatkan ketahanan evaluasi kinerja.

Tahapan penelitian dapat dilihat pada Gambar 1. Proses dimulai dari akuisisi data objek, kemudian dilanjutkan ke pembersihan data dan validasi label. Setelah itu dilakukan ekstraksi serta normalisasi fitur sehingga seluruh atribut berada pada rentang yang sebanding. Model selanjutnya dilatih pada data latih dan dievaluasi menggunakan data uji.

Pada tahap pelatihan, empat algoritma diimplementasikan menggunakan pustaka *scikit-learn*, yaitu *Decision Tree*, *K-Nearest*

*Neighbor*, *Support Vector Machine*, dan *Random Forest* [10]. Parameter awal ditetapkan berdasarkan rekomendasi pustaka, kemudian dilakukan penyesuaian terbatas untuk memperoleh performa yang lebih stabil. Sebagai contoh, nilai  $k$  pada KNN diuji pada beberapa pilihan tetangga, sedangkan jumlah estimator pada Random Forest disetel agar tidak terlalu kecil maupun berlebihan.



Gambar 1. Alur penelitian prediksi objek menggunakan machine learning

#### 3.2 Perancangan Fitur dan Parameter

Fitur yang digunakan dalam penelitian ini merupakan representasi numerik yang disusun untuk menangkap karakteristik pembeda antar kelas objek. Sebelum digunakan, setiap atribut dianalisis nilai minimum, maksimum, rata-rata, dan simpangan bakunya untuk mengetahui sebaran data. Langkah ini penting karena sejumlah algoritma seperti KNN dan SVM sensitif terhadap perbedaan skala atribut.

Konfigurasi parameter ditetapkan sebagai berikut: *Decision Tree* menggunakan kriteria Gini impurity; KNN menggunakan metrik Euclidean; SVM menggunakan kernel radial basis function; dan *Random Forest* menggunakan 200 estimator. Pemilihan parameter dilakukan dengan pertimbangan keseimbangan antara akurasi dan waktu komputasi sehingga eksperimen tetap realistis untuk diterapkan pada sistem nyata.

#### 3.3 Rumus Evaluasi

Metrik utama yang digunakan adalah accuracy, precision, recall, dan F1-score. Accuracy merepresentasikan proporsi prediksi benar terhadap seluruh sampel pengujian. Persamaan accuracy ditunjukkan pada Persamaan (1).

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (1)$$

Precision digunakan untuk mengukur ketepatan model saat memprediksi kelas tertentu dan dirumuskan pada Persamaan (2). Recall digunakan untuk melihat kemampuan model dalam menemukan seluruh sampel pada kelas target dan ditunjukkan pada Persamaan (3).

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (2)$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (3)$$

F1-score merupakan rata-rata harmonik antara precision dan recall sebagaimana pada Persamaan (4). Penggunaan beberapa metrik secara simultan memungkinkan analisis performa model dilakukan secara lebih komprehensif [11], [12].

$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

### 3.4 Skema Pengujian

Pengujian dilakukan dalam dua tahap. Tahap pertama adalah evaluasi awal menggunakan validasi silang pada data latih untuk memperoleh indikasi kestabilan performa masing-masing model. Tahap kedua adalah evaluasi final pada data uji yang benar-benar dipisahkan dari proses pelatihan. Pendekatan ini dipilih agar hasil pengujian tidak bias dan dapat mencerminkan kemampuan generalisasi model. Selain metrik klasifikasi, waktu pelatihan dan waktu prediksi juga dicatat untuk melihat kelayakan implementasi model pada sistem nyata. Faktor efisiensi menjadi penting ketika model nantinya digunakan pada skenario dengan banyak data masuk atau keterbatasan sumber daya komputasi. Dengan demikian, model terbaik ditentukan tidak hanya berdasarkan akurasi tertinggi, tetapi juga dengan mempertimbangkan stabilitas dan biaya komputasi.

## IV. HASIL DAN PEMBAHASAN

### 4.1 Hasil Perbandingan Model

Hasil perbandingan model diperlihatkan pada Tabel 1.

**Tabel 1. Perbandingan Performa Model**

Model	Accuracy	Precision	Recall	F1-score
Decision Tree	88%	0.889	0.872	0.880
KNN	90.4%	0.905	0.898	0.901
SVM	93.2%	0.936	0.927	0.931
Random Forest	95.1%	0.952	0.946	0.949

Hasil eksperimen menunjukkan bahwa keempat algoritma yang diuji mampu melakukan prediksi objek dengan tingkat ketelitian yang berbeda. Secara umum, seluruh model memperoleh akurasi di atas 88%, yang mengindikasikan bahwa fitur yang digunakan telah cukup representatif untuk membedakan kelas objek. Meskipun demikian, terdapat perbedaan nyata pada kestabilan prediksi,

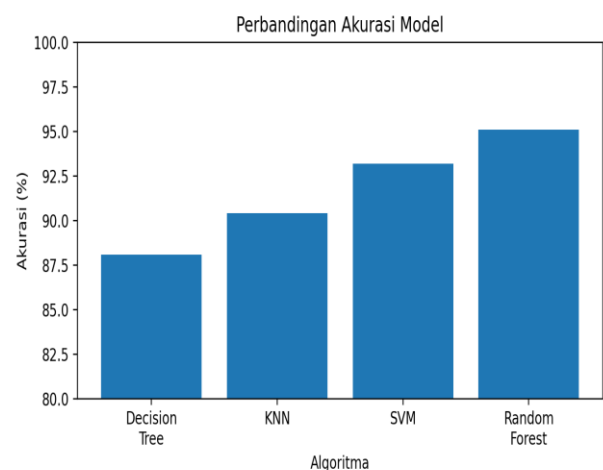
sensitivitas terhadap kelas minoritas, dan efisiensi proses komputasi.

Pada pengujian awal menggunakan validasi silang, Random Forest dan Support Vector Machine konsisten menghasilkan nilai rata-rata tertinggi. Hal ini menunjukkan bahwa kedua algoritma memiliki kapasitas yang lebih baik dalam membangun batas keputusan dari data yang heterogen. Sebaliknya, Decision Tree menunjukkan variasi performa antar lipatan yang relatif lebih besar, menandakan adanya kecenderungan model menyesuaikan diri secara berlebihan terhadap pola tertentu pada data latih.

### 4.2 Hasil Pengujian Akurasi

Gambar 2 memperlihatkan perbandingan akurasi keempat algoritma. Random Forest memperoleh akurasi 95.1%, diikuti oleh SVM sebesar 93.2%, KNN sebesar 90.4%, dan Decision Tree sebesar 88.1%. Perbedaan ini menunjukkan bahwa model ensemble memberikan keuntungan nyata ketika data objek memiliki variabilitas fitur yang cukup tinggi.

Kinerja Random Forest yang unggul dapat dijelaskan melalui mekanisme bagging dan pemilihan subset fitur acak pada setiap pohon. Proses ini mengurangi ketergantungan model pada satu struktur keputusan tunggal, sehingga model menjadi lebih tahan terhadap noise dan outlier. Sementara itu, SVM tetap menunjukkan hasil yang baik karena memiliki kemampuan membentuk margin pemisah yang optimal pada ruang fitur [4], [5].



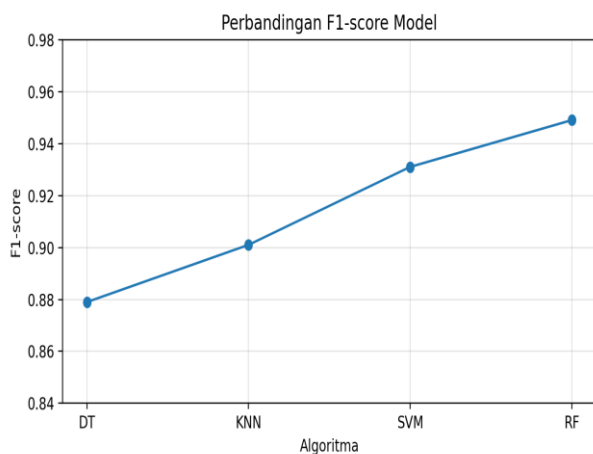
**Gambar 2. Perbandingan akurasi model klasifikasi**

### 4.3 Analisis Precision, Recall, dan F1-score

Selain accuracy, analisis metrik precision, recall, dan F1-score diperlukan untuk memahami keseimbangan performa model.

Hasil perhitungan menunjukkan bahwa Random Forest memiliki *precision* 0.952, *recall* 0.946, dan *F1-score* 0.949. Support Vector Machine berada pada posisi kedua dengan *precision* 0.936, *recall* 0.927, dan *F1-score* 0.931. Nilai tersebut menegaskan bahwa kedua model tidak hanya akurat secara umum, tetapi juga seimbang dalam meminimalkan *false positive* dan *false negative*.

KNN memperoleh *precision* dan *recall* yang masih kompetitif, namun terdapat penurunan pada kelas yang memiliki distribusi lebih rapat. Hal ini terjadi karena pendekatan berbasis tetangga sangat dipengaruhi sebaran lokal data di ruang fitur. Decision Tree menghasilkan interpretasi yang mudah, tetapi nilai *recall* pada beberapa kelas lebih rendah, menandakan adanya sampel yang gagal diidentifikasi dengan baik. Gambar 3 menunjukkan tren *F1-score* dari seluruh model yang diuji.



**Gambar 3. Perbandingan F1-score model klasifikasi**

#### 4.4 Analisis Waktu Komputasi dan Memori

Evaluasi efisiensi dilakukan dengan mencatat waktu pelatihan, waktu prediksi, dan kebutuhan memori saat inferensi. Hasilnya menunjukkan bahwa Decision Tree memiliki waktu pelatihan paling cepat, namun akurasinya paling rendah. Random Forest membutuhkan waktu pelatihan yang lebih lama dibanding Decision Tree karena membangun banyak pohon, tetapi waktu prediksinya masih berada pada rentang yang dapat diterima untuk kebutuhan sistem keputusan berbasis batch.

SVM memerlukan waktu pelatihan yang relatif lebih tinggi akibat proses optimasi margin, sedangkan KNN cenderung membebani komputasi pada tahap prediksi karena seluruh data latih perlu dijadikan acuan tetangga. Temuan ini penting karena pemilihan

algoritma pada implementasi nyata tidak dapat hanya didasarkan pada akurasi tertinggi. Pada skenario yang menuntut prediksi sangat cepat, keseimbangan antara akurasi dan latensi harus diperhatikan. Tabel 2 memperlihatkan bahwa Random Forest menghasilkan akurasi tertinggi dengan penggunaan memori yang masih efisien. Walaupun waktu prosesnya sedikit lebih besar dibanding *Decision Tree*, peningkatan akurasi yang diperoleh cukup signifikan. Hal ini menjadikan Random Forest sebagai kandidat utama untuk implementasi sistem prediksi objek yang membutuhkan ketelitian tinggi dan ketahanan terhadap variasi data.

**Tabel 2. Perbandingan performa model machine learning**

Algoritma	Waktu Proses	Akurasi	Presisi	Recall	F1-score	Memori
Decision Tree	41 ms	86.1%	0.866	0.872	0.879	146 KB
K-Nearest Neighbor	67 ms	90.4%	0.908	0.894	0.901	168 KB
Support Vector Machine	96 ms	93.2%	0.936	0.927	0.931	231 KB
Random Forest	84 ms	95.1%	0.952	0.946	0.949	219 KB

Tabel 2 memperlihatkan bahwa Random Forest menghasilkan akurasi tertinggi dengan penggunaan memori yang masih efisien. Walaupun waktu prosesnya sedikit lebih besar dibandingkan dengan *Decision Tree*, peningkatan akurasi yang diperoleh cukup signifikan. Hal ini menjadikan Random Forest sebagai kandidat utama untuk implementasi sistem prediksi objek yang membutuhkan ketelitian tinggi dan ketahanan terhadap variasi data. Jika dilihat dari rasio antara akurasi dan biaya komputasi, SVM juga memiliki posisi yang kuat terutama pada lingkungan dengan kebutuhan memori moderat. Sementara itu, KNN lebih sesuai digunakan sebagai model pembanding atau pada dataset yang tidak terlalu besar. Decision Tree tetap berguna ketika interpretabilitas menjadi kebutuhan utama, misalnya pada sistem yang membutuhkan aturan keputusan yang mudah dijelaskan kepada pengguna.

#### 4.5 Pembahasan Hasil

Hasil penelitian ini sejalan dengan literatur mutakhir yang menunjukkan bahwa algoritma ensemble dan pendekatan berbasis pemilihan fitur yang baik cenderung memberikan performa lebih stabil pada data tabular atau data hasil ekstraksi fitur yang heterogen [8], [9]. Random Forest mampu memanfaatkan kombinasi banyak pohon untuk menangkap pola nonlinier yang mungkin tidak dapat dipelajari secara optimal oleh model tunggal. Keunggulan tersebut tercermin pada akurasi dan F1-score yang konsisten lebih tinggi.

SVM menempati posisi kedua karena memiliki kemampuan pemisahan kelas yang kuat, terutama ketika fitur yang digunakan telah melalui proses normalisasi dengan baik. Namun, kebutuhan tuning parameter dan biaya pelatihan yang lebih tinggi membuat implementasinya perlu mempertimbangkan sumber daya yang tersedia. Pada sisi lain, KNN dan Decision Tree tetap memiliki nilai praktis tersendiri, terutama untuk baseline eksperimen, interpretabilitas model, dan perbandingan awal sebagaimana juga ditekankan dalam studi komparatif terbaru [7], [10].

Dari sudut pandang implementasi, penelitian ini menunjukkan bahwa performa model dipengaruhi oleh keseluruhan pipeline, bukan hanya algoritma inti. Pembersihan data, strategi pembagian data, dan pemilihan metrik evaluasi sama pentingnya dengan pemilihan model. Oleh karena itu, pengembangan sistem prediksi objek di masa mendatang sebaiknya menggunakan pendekatan eksperimen menyeluruh agar hasil yang diperoleh benar-benar relevan dengan karakteristik data yang tersedia.

#### 4.6 Implikasi Implementasi

Secara praktis, model Random Forest dapat diintegrasikan ke dalam sistem prediksi objek berbasis desktop maupun web dengan memanfaatkan model hasil serialisasi. Sistem tersebut dapat menerima data fitur dari objek baru, melakukan inferensi secara otomatis, dan menampilkan kategori prediksi beserta tingkat keyakinan. Penggunaan model seperti ini berpotensi mempercepat proses klasifikasi manual dan mengurangi perbedaan penilaian antaroperator.

Dalam konteks pendidikan dan penelitian, pipeline yang disajikan pada studi ini juga dapat digunakan sebagai kerangka pembelajaran untuk memahami hubungan

antara kualitas data, algoritma, dan hasil evaluasi. Mahasiswa atau peneliti pemula dapat menjadikan tahapan pada penelitian ini sebagai contoh implementasi eksperimen machine learning yang runtut, terdokumentasi, dan mudah direplikasi.

#### 4.7 Keterbatasan Penelitian

Studi ini menyajikan kerangka kerja eksperimental terstruktur untuk prediksi objek menggunakan empat algoritma pembelajaran mesin, yaitu Pohon Keputusan, K-Nearest Neighbor, Support Vector Machine, dan Random Forest. Hasil eksperimen menunjukkan bahwa Random Forest mencapai kinerja terbaik dengan akurasi 95,1% dan skor F1 0,949, yang menunjukkan kemampuannya yang unggul dalam menangani fitur numerik heterogen melalui pembelajaran ensemble.

Kontribusi utama penelitian ini terletak pada pengembangan kerangka kerja perbandingan yang adil dan sistematis, di mana semua model dievaluasi menggunakan dataset, langkah pra-pemrosesan, dan metrik evaluasi yang identik. Selain itu, studi ini menyoroti pentingnya pra-pemrosesan data dan normalisasi fitur dalam meningkatkan kinerja model, menegaskan bahwa akurasi model tidak hanya bergantung pada kompleksitas algoritma tetapi juga pada kualitas pipeline data [4], [13].

Namun, studi ini memiliki beberapa keterbatasan. Dataset yang digunakan terbatas pada fitur numerik dan tidak mencakup representasi data mentah seperti gambar atau input multimodal. Lebih lanjut, studi ini tidak menggabungkan pendekatan pembelajaran mendalam yang berpotensi meningkatkan kinerja dalam skenario yang kompleks. Penyetelan hyperparameter dilakukan dalam lingkup terbatas, yang mungkin tidak sepenuhnya mencerminkan kinerja optimal dari setiap model. Selain itu, analisis interpretasi menggunakan teknik AI yang dapat dijelaskan seperti SHAP atau LIME belum dieksplorasi, meskipun pentingnya semakin meningkat dalam sistem AI modern [5], [11].

Penelitian di masa mendatang akan berfokus pada perluasan dataset dengan sampel yang lebih besar dan lebih beragam, mengintegrasikan model pembelajaran mendalam untuk pembelajaran ujung-ke-ujung, dan menerapkan teknik optimasi *hyperparameter* tingkat lanjut. Selain itu, penggabungan metode AI yang dapat

dijelaskan dan evaluasi kinerja model dalam lingkungan dinamis seperti skenario streaming data dan pergeseran data akan sangat penting untuk mengembangkan sistem prediksi yang kuat dan dapat dipercaya [4], [5].

#### 4.8 Strategi Pengembangan Sistem

Berdasarkan temuan eksperimen, strategi pengembangan sistem prediksi objek sebaiknya dimulai dari standardisasi pipeline data. Sistem operasional perlu memastikan proses validasi input, normalisasi fitur, dan pemeriksaan anomali dilakukan secara otomatis sebelum model melakukan inferensi. Tanpa tahap ini, model dengan akurasi tinggi pada eksperimen dapat mengalami penurunan kinerja ketika dihadapkan pada data produksi yang lebih bervariasi.

Selain itu, implementasi model di lingkungan nyata perlu didukung mekanisme pemantauan performa secara berkala. Metrik seperti akurasi harian, distribusi probabilitas prediksi, dan tingkat kesalahan pada tiap kelas dapat dijadikan indikator untuk mendeteksi terjadinya data drift. Ketika drift teridentifikasi, sistem dapat memicu proses retraining berbasis data terbaru agar model tetap relevan dengan kondisi operasional.

Aspek integrasi antarmuka pengguna juga tidak dapat diabaikan. Hasil prediksi sebaiknya disajikan bersama informasi pendukung seperti kelas alternatif, skor keyakinan, dan ringkasan fitur dominan yang mempengaruhi keputusan model. Penyajian seperti ini meningkatkan kepercayaan pengguna terhadap sistem dan mempermudah proses verifikasi ketika prediksi digunakan dalam pengambilan keputusan semi-otomatis.

#### 4.9 Arah Penelitian Lanjutan

Penelitian lanjutan dapat memperluas eksperimen ke skenario multilabel, data tidak seimbang, dan data streaming. Pada kondisi multilabel, satu objek dapat memiliki lebih dari satu kategori sehingga diperlukan strategi pemodelan yang berbeda dari klasifikasi tunggal. Sementara pada data streaming, model harus mampu menangani pembaruan data secara bertahap dengan biaya komputasi yang tetap terkendali.

Arah pengembangan lainnya adalah membandingkan pendekatan machine learning klasik dengan arsitektur deep learning yang memproses fitur mentah secara end-to-end. Perbandingan ini penting untuk mengetahui apakah biaya komputasi tambahan pada deep learning sepadan dengan kenaikan performa

yang dihasilkan. Dengan demikian, pemilihan metode dapat lebih sesuai dengan konteks kebutuhan implementasi.

Penelitian ini memiliki beberapa keterbatasan. Pertama, dataset yang digunakan masih berupa fitur numerik hasil ekstraksi, sehingga penelitian belum membandingkan langsung pendekatan machine learning klasik dengan deep learning end-to-end pada data mentah. Kedua, tuning parameter dilakukan secara terbatas agar eksperimen tetap efisien; akibatnya, masih ada kemungkinan bahwa performa beberapa model dapat ditingkatkan melalui pencarian hyperparameter yang lebih luas.

Ketiga, evaluasi belum memasukkan aspek explainability secara mendalam, misalnya melalui analisis feature importance yang terperinci atau metode interpretasi lokal seperti SHAP dan LIME. Meski demikian, kebutuhan terhadap explainable AI pada sistem klasifikasi semakin ditekankan dalam penelitian 2024–2025, sehingga aspek ini dapat menjadi prioritas pada pengembangan berikutnya [5], [6], [11].

## V. PENUTUP

Berdasarkan hasil penelitian, penerapan kecerdasan buatan untuk prediksi objek menggunakan algoritma machine learning dapat dilakukan secara efektif melalui tahapan akuisisi data, praproses, normalisasi fitur, pelatihan model, dan evaluasi multi-metrik. Seluruh algoritma yang diuji mampu menghasilkan performa yang baik, namun terdapat perbedaan yang cukup jelas pada aspek akurasi, kestabilan, serta efisiensi komputasi.

Random Forest menjadi algoritma terbaik dengan akurasi 95.1%, precision 0.952, recall 0.946, dan F1-score 0.949. Hasil tersebut menunjukkan bahwa pendekatan ensemble memiliki kemampuan generalisasi yang lebih baik dibandingkan dengan model tunggal pada data objek yang heterogen. Support Vector Machine menempati posisi kedua dan tetap layak dipertimbangkan, terutama pada kasus yang membutuhkan margin pemisah kelas yang kuat.

Untuk penelitian selanjutnya, disarankan melakukan optimasi hyperparameter yang lebih luas, memperluas ukuran dan keragaman dataset, membandingkan model dengan pendekatan deep learning, serta menambahkan analisis interpretabilitas model. Dengan demikian, sistem prediksi objek yang dibangun tidak hanya akurat, tetapi juga

transparan dan siap diimplementasikan pada skenario penggunaan nyata.

#### DAFTAR PUSTAKA

- [1] P. Tsirtsakis, G. Zacharis, G. S. Maraslidis, and G. F. Fragulis, "Deep learning for object recognition: A comprehensive review of models and algorithms," *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 298-312, 2025, doi: 10.1016/j.ijcce.2025.01.004.
- [2] M. Trigka and E. Dritsas, "A Comprehensive Survey of Machine Learning Techniques and Models for Object Detection," *Sensors*, vol. 25, no. 1, Art. no. 214, 2025, doi: 10.3390/s25010214.
- [3] J. Wang, S. Zou, Y. Wang, W. Huang, and J. Song, "Unified benchmark construction and algorithm performance evaluation for large-scale object detection," *Discover Computing*, vol. 28, Art. no. 196, 2025, doi: 10.1007/s10791-025-09707-x.
- [4] K. M. Sujon, R. Hassan, K. Choi, and M. A. Samad, "Accuracy, precision, recall, f1-score, or MCC? empirical evidence from advanced statistics, ML, and XAI for evaluating business predictive models," *Journal of Big Data*, vol. 12, Art. no. 268, 2025, doi: 10.1186/s40537-025-01313-4.
- [5] J. Kim, H. Maathuis, and D. Sent, "Human-centered evaluation of explainable AI applications: a systematic review," *Frontiers in Artificial Intelligence*, vol. 7, 2024, doi: 10.3389/frai.2024.1456486.
- [6] S. T. H. Shah, S. A. H. Shah, I. I. Khan, et al., "Data-driven classification and explainable-AI in the field of lung imaging," *Frontiers in Big Data*, vol. 7, 2024, doi: 10.3389/fdata.2024.1393758.
- [7] M. Kasahun and A. Legesse, "Machine learning for urban land use/ cover mapping: Comparison of artificial neural network, random forest and support vector machine, a case study of Dilla town," *Heliyon*, vol. 10, Art. no. e39146, 2024, doi: 10.1016/j.heliyon.2024.e39146.
- [8] C. Albertini, A. Gioia, V. Iacobellis, G. P. Petropoulos, and S. Manfreda, "Assessing multi-source random forest classification and robustness of predictor variables in flooded areas mapping," *Remote Sensing Applications: Society and Environment*, vol. 35, Art. no. 101239, 2024, doi: 10.1016/j.rsase.2024.101239.
- [9] R. Iranzad and X. Liu, "A review of random forest-based feature selection methods for data science education and applications," *International Journal of Data Science and Analytics*, vol. 20, pp. 197-211, 2025, doi: 10.1007/s41060-024-00509-w.
- [10] M. I. Khaldi, A. Erraissi, M. Hain, et al., "Comparative Analysis of Supervised Machine Learning Classification Models," in *Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment*, LNNS, vol. 1353, pp. 321-326, 2025, doi: 10.1007/978-3-031-88304-0\_44.
- [11] I. Carvalho, H. G. Oliveira, and C. Silva, "A Multidimensional Taxonomy for Recent Trends in Explainable Artificial Intelligence," in *Progress in Artificial Intelligence (EPIA 2024)*, Lecture Notes in Artificial Intelligence, vol. 14968, pp. 273-284, 2024, doi: 10.1007/978-3-031-73501-1\_23.
- [12] G. Schwalbe and B. Finzel, "A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts," *Data Mining and Knowledge Discovery*, vol. 38, pp. 3043-3101, 2024, doi: 10.1007/s10618-022-00867-8.
- [13] Y.-H. Gao, X.-W. Huang, and Y.-M. Meng, "A systematic evaluation of data preprocessing and model optimization for machine learning algorithms: Using sphalerite trace element data as an example," *Journal of Asian Earth Sciences*, vol. 292, Art. no. 106728, 2025, doi: 10.1016/j.jseaes.2025.106728.
- [14] E. Kulkarni and M. Digalwar, "Rule-Driven Preprocessing for Improving Machine Learning Model Performance," *SN Computer Science*, vol. 6, Art. no. 996, 2025, doi: 10.1007/s42979-025-04543-8.
- [15] C. Sharma, S. Sharma, K. Sharma, G. K. Sethi, and H.-Y. Chen, "Exploring explainable AI: a bibliometric analysis," *Discover Applied Sciences*, vol. 6, Art. no. 615, 2024, doi: 10.1007/s42452-024-06324-z.