

# Implementasi *Extremely Randomized Trees* dengan Optimasi Hyperparameter *Accelerated Particle Swarm Optimization* untuk Klasifikasi Subtipe Anemia

Adelia Yuandhika<sup>1</sup>, Trimono<sup>2</sup>, Mohammad Idhom<sup>3</sup>

<sup>1,2,3</sup>Prodi Sains Data, Fakultas Ilmu Komputer, UPN "Veteran" Jawa Timur

<sup>1</sup>22083010066@student.upnjatim.ac.id\*, <sup>2</sup> trimono.stat@upnjatim.ac.id, <sup>3</sup> idhom@upnjatim.ac.id

## Abstract

Anemia is a health problem that negatively affects both medical outcomes and social well-being, highlighting the need for accurate early detection. This study applies a machine learning approach to classify anemia subtypes to support clinical intervention and further examination. The Extra Trees method employs a hierarchical decision-tree structure with extreme randomization, making it robust to overfitting and capable of good generalization on small to medium datasets. Accelerated Particle Swarm Optimization (APSO) is utilized as an efficient optimization technique to improve classification performance. The novelty of this study lies in integrating Extra Trees with APSO to optimize anemia subtype classification. The dataset consists of 385 records collected from a regional hospital in East Java, Indonesia, covering four classes: thalassemia, iron deficiency anemia, anemia of chronic disease, and non-anemia. The features include patient initials, gender, age, and hematological parameters (Hb, HCT, RBC, MCV, MCH, MCHC, RDW). The optimized model achieved 85% accuracy, 87% precision, 85% recall, 85% F1-score, 95% specificity, and 94% AUC, outperforming the non-optimized model. These results indicate that the proposed approach is effective for anemia subtype classification.

**Keywords:** accelerated particle swarm optimization (APSO), extremely randomized trees (extra trees), classification, subtypes of anemia

## Abstrak

Anemia merupakan masalah kesehatan yang menyebabkan dampak negatif bagi kesehatan maupun kehidupan sosial. Oleh karena itu, diperlukan upaya deteksi dini yang akurat untuk menekan angka kejadian. Penelitian ini melakukan pendekatan *machine learning* untuk deteksi dini subtipe anemia guna membantu tindakan intervensi maupun menentukan pemeriksaan lanjutan. Metode *Extremely Randomized Trees* atau *Extra Trees* menggunakan struktur hierarkis pohon keputusan untuk mengenali pola data. Metode ini berbasis pengacakan ekstrim sehingga tahan *overfitting* dan memiliki kemampuan generalisasi yang baik pada dataset skala kecil sampai menengah. *Accelerated Particle Swarm Optimization* (APSO) digunakan sebagai metode optimasi yang efisien namun dapat memberikan hasil yang baik untuk meningkatkan performa model klasifikasi. Kebaruan penelitian ini terletak pada penerapan kombinasi metode *Extra Trees* dan APSO dalam mengoptimasi klasifikasi subtipe anemia. Data dikumpulkan dari RSUD Haji Provinsi Jawa Timur berjumlah 385 data dengan empat kategori, yaitu thalassemia, anemia defisiensi besi, anemia penyakit kronis, dan non-anemia. Fitur yang digunakan terdiri dari inisial pasien, jenis kelamin, usia, Hb, HCT, RBC, MCV, MCH, MCHC, dan RDW. Hasil penelitian menunjukkan bahwa performa model yang dioptimasi lebih baik dengan akurasi, *recall*, dan *F1-score* sebesar 85%, presisi 87%, spesifisitas 95%, dan AUC 94% dibandingkan model tanpa optimasi dengan akurasi 83%, presisi dan *F1-score* 66%, *recall* 68%, spesifisitas 94%, serta AUC 94%. Dengan demikian, model yang diusulkan terbukti efektif dalam mengklasifikasikan subtipe anemia.

**Kata kunci:** accelerated particle swarm optimization (APSO), extremely randomized trees (extra trees), klasifikasi, subtipe anemia

©This work is licensed under a Creative Commons Attribution -ShareAlike 4.0 International License

## 1. Pendahuluan

Anemia didefinisikan sebagai keadaan penurunan sel darah merah dibawah ambang batas normal [1]. Dampak negatif anemia diantaranya adalah menurunkan kecerdasan, menghambat pertumbuhan dan perkembangan pada usia produktif, mengganggu kesehatan reproduksi, dan gagal jantung [2]. Dalam kehidupan sosial, anemia dapat menurunkan kualitas hidup, tingkat produktivitas, hingga kesejahteraan sosial [3].

Berdasarkan Survei Kesehatan Indonesia 2023, Indonesia masih memiliki angka prevalensi anemia yang tinggi. prevalensi anemia pada balita mencapai 23,8%, ibu hamil sebesar 27%, lansia sebesar 63,2%,

perempuan sebesar 18%, dan 14,4% pada laki-laki [4]. Menurut Profil Kesehatan Jawa Timur 2024 [5], anemia termasuk dalam tiga masalah kesehatan terbanyak pada jenjang SMP-SMA. Tercatat sebagai 10 besar penyakit pasien rawat inap, sebesar 65.333 penderita. Serta 10 penyakit dengan fatalitas terbesar pada pasien rawat inap, yaitu sebesar 4.781. Hal ini menunjukkan bahwa permasalahan terkait anemia perlu mendapatkan perhatian khusus. Sementara itu, berdasarkan Profil Kesehatan Kota Surabaya 2024 [6], tercatat penderita anemia sejumlah 3.835 pada ibu hamil, 3.809 calon pengantin wanita, serta 22.241 pasien rawat inap.

Anemia memiliki beberapa subtipe, seperti anemia defisiensi besi, *thalassemia*, dan anemia penyakit kronis. Masing-masing subtipe memiliki bentuk penanganan yang berbeda. Sementara itu, untuk menentukan subtipe anemia dibutuhkan pemeriksaan laboratorium menggunakan alat khusus. Biaya untuk menjalani pemeriksaan yang tergolong tinggi tentunya membebani masyarakat. Garcia-Casal dkk [7] menyebutkan bahwa keterbatasan sumber daya dan kapasitas layanan kesehatan masih menjadi hambatan dalam pelayanan kesehatan untuk kasus anemia. Selain itu, proses diagnosis yang tidak tepat berpotensi menimbulkan salah klasifikasi subtipe anemia, terutama pada kondisi di mana gejala antar subtipe memiliki kemiripan. Kesalahan klasifikasi ini dapat berdampak pada pemberian terapi yang tidak sesuai, misalnya pemberian suplemen zat besi pada pasien *thalassemia* yang justru dapat memperburuk kondisi akibat penumpukan zat besi dalam tubuh. Oleh karena itu, diperlukan suatu pendekatan yang mampu membedakan subtipe anemia secara lebih efisien. Berdasarkan kondisi ini, diusulkan model klasifikasi *machine learning* untuk deteksi dini subtipe anemia guna membantu menentukan tindakan intervensi maupun pemeriksaan lanjutan.

Penelitian mengenai klasifikasi anemia pernah dilakukan oleh peneliti lain, seperti Priandika dan Isnain [8] yang menggunakan metode *ensemble learning* seperti *Random Forest*, *KNN*, dan *Gradient Boosting*. Secara keseluruhan menghasilkan akurasi hingga 98%. Namun, data yang digunakan masih merupakan data dari situs *open source* sehingga belum sepenuhnya merepresentasikan kondisi nyata di lapangan.

Algoritma berbasis pohon menggunakan struktur hierarkis untuk memodelkan data. Model yang dihasilkan oleh algoritma ini bersifat fleksibel karena mampu membuat batas keputusan dalam pola data yang rumit. Ainurrohman dan Wiyanti [9] menggunakan algoritma berbasis pohon, yaitu *Decision Tree* untuk klasifikasi zona daerah risiko covid-19 di Indonesia dengan akurasi 87%, presisi 87%, dan *recall* 88%. Meskipun menunjukkan kinerja yang baik, metode tersebut rentan mengalami *overfitting*. *Decision Tree* cenderung menghasilkan pohon yang terlalu dalam sehingga model kehilangan kemampuan generalisasi dan tidak stabil.

Algoritma berbasis pohon lain, yaitu *Extremely Randomized Trees* mengatasi kelemahan tersebut melalui teknik *ensemble* yang membangun banyak pohon keputusan secara bersamaan. Pada setiap cabang pohon atau *node*, fitur dipilih secara acak. Selain itu, titik pemisahan atau titik *split* juga ditentukan secara acak. Hasil klasifikasi diperoleh melalui teknik *majority voting* dari seluruh pohon yang terbentuk [10]. Dengan demikian, *Extremely Randomized Trees* dapat menangani data dengan karakteristik kompleks dan non-linear, seperti pada data anemia yang memiliki keterkaitan antar variabel laboratorium. *Extremely*

*Randomized Trees* bekerja dengan pemilihan atribut dan titik *split* secara acak, sehingga dapat mengurangi variansi model dan meningkatkan robustnes terhadap noise pada data.

Dibandingkan dengan metode lain seperti *Decision Tree*, dan *Random Forest*, *Extremely Randomized Trees* dipilih karena memiliki keunggulan dalam menangani data dengan hubungan *non-linear* dan kompleks, seperti pada kasus klasifikasi subtipe anemia. *Decision Tree* sebagai model tunggal memiliki kecenderungan *overfitting* dan sensitif terhadap variasi data, sehingga kurang stabil dalam melakukan generalisasi. Sementara itu, *Random Forest* menggunakan pemilihan *split* terbaik pada setiap node yang masih berpotensi menimbulkan korelasi antar pohon. Berbeda dengan itu, *Extremely Randomized Trees* melakukan pemilihan atribut dan titik *split* secara acak, sehingga mampu mengurangi variansi model secara lebih signifikan, serta menghasilkan model yang lebih stabil. Karakteristik ini menjadikan *Extremely Randomized Trees* lebih sesuai untuk membedakan subtipe anemia yang memiliki kemiripan pola antar kelas dan hubungan antar variabel yang tidak sederhana.

Moglia dan Almufata [11] menggunakan beberapa algoritma termasuk *Extremely Randomized Trees* untuk prediksi kanker payudara *benign* dan *malignant*. Hasilnya menunjukkan performa yang baik dengan akurasi 88,71%. Sementara itu, Fern Soon dkk [12] menggunakan model berbasis pohon untuk mengklasifikasikan beberapa jenis serangan jaringan atau *network attacks*. Dalam hal ini, model menghasilkan performa akurasi sebesar 82,52%. Berdasarkan temuan ini, diketahui bahwa terjadi penurunan performa pada klasifikasi multikelas dibandingkan klasifikasi biner. Oleh sebab itu, penelitian ini menggunakan tuning hyperparameter untuk meningkatkan kemampuan model dalam klasifikasi multikelas.

Tuning hyperparameter merupakan upaya untuk mengoptimalkan kinerja model, menjaga stabilitas model, dan memastikan model memiliki generalisasi yang baik. Tuning *hyperparameter* pada *Extremely Randomized Trees* membutuhkan eksplorasi pada ruang pencarian yang luas untuk menemukan kombinasi *hyperparameter* yang optimal. Metode tuning konvensional sering kali menghadapi tantangan signifikan dalam ruang pencarian yang luas dan rumit. Sementara itu, terdapat metode metaheuristik seperti *Accelerated Particle Swarm Optimization* (APSO) yaitu variasi dari *Particle Swarm Optimization* (PSO). Tidak seperti PSO yang menggunakan dua komponen utama, yaitu *global best* dan *personal best*, APSO hanya menggunakan *global best* untuk meningkatkan efektivitas pencarian. APSO bekerja dengan mempercepat perpindahan partikel (solusi) menuju posisi optimal sehingga dapat memperkuat kemampuan eksplorasi global.

Yang dkk [13] menggunakan *Accelerated Particle Swarm Optimization* (APSO) sebagai metode optimasi hyperparameter *Support Vector Machine* (SVM) untuk optimasi bisnis. Terdapat dua parameter yang dioptimalkan yakni  $C$  dan  $\sigma^2$  (gamma) dengan hasil performa deviasi yang sangat kecil, sebesar 0,19. Sementara itu, Emara dkk [14] membahas mengenai penggunaan APSO pada model *deep learning*, yaitu CNN atau *Convolutional Neural Network*. Hasil penelitian menunjukkan bahwa APSO dapat meningkatkan kinerja CNN, terutama pada arsitektur GoogleNet dan ResNet yang masing-masing mencapai akurasi hingga 83% dan 84%. Temuan ini menegaskan bahwa *Accelerated Particle Swarm Optimization* (APSO) adalah metode yang efektif untuk mengoptimalkan kinerja algoritma melalui tuning *hyperparameter*.

Kebaruan dalam penelitian terletak pada pemanfaatan *Accelerated Particle Swarm Optimization* (APSO) dalam proses optimasi hyperparameter pada metode *Extremely Randomized Trees* secara lebih efisien. Berbeda dengan metode konvensional yang melakukan pencarian parameter secara ekstensif dan cenderung membutuhkan waktu komputasi yang besar, APSO mengandalkan mekanisme percepatan konvergensi berbasis *global best* untuk mempercepat penemuan solusi optimal. Pendekatan ini memungkinkan proses pencarian *hyperparameter* menjadi lebih terarah dengan jumlah iterasi yang lebih sedikit, sehingga mampu mengurangi beban komputasi tanpa menurunkan kinerja model. Integrasi APSO dengan *Extremely Randomized Trees* diharapkan tidak hanya meningkatkan akurasi dan stabilitas dalam klasifikasi multikelas, tetapi juga menghasilkan proses optimasi yang lebih efisien dan efektif dalam memperoleh kombinasi *hyperparameter* terbaik pada klasifikasi sub tipe anemia.

Dataset yang digunakan dalam penelitian ini memiliki distribusi kelas yang tidak seimbang dimana kelas non anemia memiliki jumlah yang paling mendominasi sedangkan kelas *thalassemia* memiliki jumlah yang paling sedikit. Jika dibiarkan hal ini dapat menyebabkan model bias terhadap kelas minoritas sehingga perlu dilakukan penyeimbangan data. Teknik penyeimbangan data yang digunakan adalah *synthetic minority oversampling technique* (SMOTE).

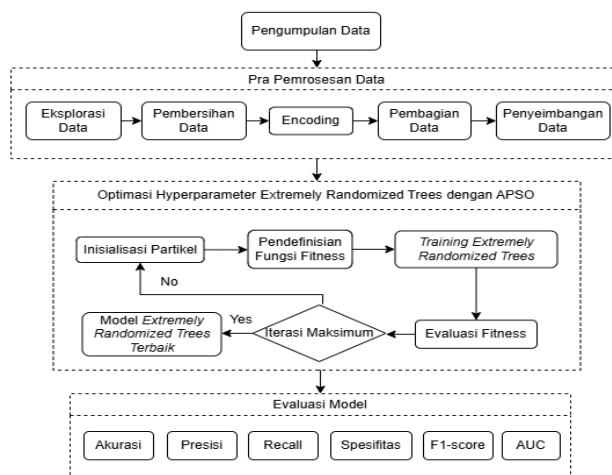
Fitur-fitur yang digunakan dalam penelitian ini antara lain, jenis kelamin, usia, Hb, HCT, RBC, MCV, MCH, MCHC, dan RDW. Fitur laboratorium seperti Hb, HCT, dan RBC diketahui memiliki korelasi yang kuat satu sama lain, sedangkan MCV, MCH, dan RDW memiliki korelasi positif dan negatif yang bervariasi. Ini menunjukkan adanya keterkaitan yang kompleks dan tidak seragam. Selain itu, data cenderung memiliki karakteristik non-linier dan heterogen, sehingga membutuhkan metode yang dapat menangkap pola data yang kompleks sekaligus didukung oleh proses optimasi yang efektif. Dalam hal ini, *Accelerated Particle Swarm Optimization* (APSO) dipilih karena

memiliki keunggulan dibandingkan *Particle Swarm Optimization* (PSO) standar, yaitu mampu mempercepat konvergensi dengan memanfaatkan informasi *global best* secara lebih dominan sehingga proses pencarian solusi optimal menjadi lebih cepat dan efisien. Dengan jumlah iterasi minimal, APSO dapat menekan biaya komputasi tanpa mengurangi kualitas sehingga lebih sesuai digunakan untuk mengoptimalkan kinerja *Extremely Randomized Trees* dalam klasifikasi sub tipe anemia. Dengan demikian, penelitian ini ingin menggunakan tuning hyperparameter *Accelerated Particle Swarm Optimization* (APSO) pada metode *Extremely Randomized Trees* untuk model klasifikasi multikelas sub tipe anemia guna meningkatkan performanya.

Penelitian ini bertujuan untuk mengetahui bagaimana kombinasi hyperparameter yang optimal pada metode *Extremely Randomized Trees* dalam mengklasifikasikan beberapa sub tipe anemia, antara lain anemia defisiensi besi, *thalassemia*, anemia penyakit kronis, dan non-anemia. Selain itu, juga untuk mengetahui seberapa besar pengaruh APSO dalam meningkatkan performa *Extremely Randomized Trees*.

## 2. Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif menggunakan algoritma pembelajaran mesin (*machine learning*) *extremely randomized trees* atau dikenal dengan *extra trees* dengan optimasi hyperparameter *accelerated particle swarm optimization* (APSO). Data yang digunakan merupakan data sekunder. Penelitian ini dilakukan menggunakan bahasa pemrograman python dengan tools *Google Colaboratory*. Pustaka utama yang digunakan dalam pemodelan adalah *scikit learn*. Terdapat beberapa pustaka pendukung yang juga digunakan antara lain *pandas*, *numpy*, *imbalanced learn*, *matplotlib*, dan *seaborn*. Alur penelitian dimulai dari pengumpulan data, pra pemrosesan data, optimasi hyperparameter *extremely randomized trees* dengan APSO hingga evaluasi model. Alur secara lebih lengkap ditampilkan dalam gambar 1.



Gambar 1. Alur Penelitian

Gambar 1 adalah diagram alur penelitian ini. Penelitian ini memanfaatkan algoritma *machine learning* untuk membedakan subtype anemia. Dataset yang digunakan berbentuk tabular dengan format csv. Dimana setiap baris data merepresentasikan satu orang pasien. Sementara itu, berikut merupakan rentang nilai *hyperparameter* pada metode *Extremely Randomized Trees* yang akan dioptimasi untuk memperoleh kombinasi parameter terbaik menggunakan APSO.

Tabel 1. Rentang Nilai Hyperparameter *Extremely Randomized*

<i>Trees</i>	
Nama Hyperparameter	Rentang Nilai
<i>n_estimators</i>	1 – 500
<i>max_depth</i>	1 – 100
<i>min_samples_split</i>	1 – 100
<i>min_samples_leaf</i>	1 – 100
<i>max_features</i>	0,2 – 1,0

### 2.1. Pengumpulan Data

Data yang digunakan adalah data sekunder yang didapatkan di RSUD Haji Provinsi Jawa Timur yang beralamat di Jl. Manyar Kertoadi, Kecamatan Sukolilo, Kota Surabaya. Data dikumpulkan mulai periode 1 Januari 2025 hingga 6 Meret 2026. Jumlah dataset yang berhasil dikumpulkan terdiri dari 385 baris data.

### 2.2. Pra Pemrosesan Data

Tahap ini terdiri dari 5 proses utama, yaitu eksplorasi data, pembersihan data, encoding, pembagian data, dan penyeimbangan data. Berikut adalah uraiannya.

#### a. Eksplorasi Data

Pra pemrosesan data diawali dengan melakukan eksplorasi data. Proses ini dilakukan untuk mengetahui karakteristik data sehingga dapat diidentifikasi pola distribusi, hubungan antar variabel, serta potensi permasalahan seperti ketidakseimbangan kelas, yang selanjutnya menjadi dasar dalam menentukan teknik pra pemrosesan lanjutan.

#### b. Pembersihan Data

Pembersihan data dilakukan melalui serangkaian pemeriksaan data meliputi duplikasi data dan nilai kosong atau *missing value*. Apabila ditemukan bahwa data mengalami salah satu atau beberapa permasalahan tersebut, maka akan dilakukan penanganan terlebih dahulu.

#### c. Encoding Variabel Kategorikal

Setelah permasalahan data diselesaikan, tahapan dilanjutkan dengan *encoding* variabel kategorikal, yaitu jenis kelamin dan subtype anemia. Proses ini dilakukan karena algoritma *machine learning* memerlukan input dalam bentuk numerik agar dapat diproses secara komputasional. Pada penelitian ini digunakan *label encoding*, dimana jenis kelamin Perempuan (P) dirubah menjadi angka 0. Sedangkan Laki-Laki (L) menjadi angka 1. Sementara untuk subtype anemia, thalassemia dikodekan dengan angka 0, anemia defisiensi besi dengan angka 1, non anemia angka 2, dan anemia penyakit kronis angka 3.

Pengkodean ini tidak merepresentasikan urutan atau tingkat tertentu, melainkan hanya sebagai identitas kelas agar model dapat membedakan setiap kategori secara komputasional.

#### d. Pembagian Data

Selanjutnya adalah pembagian data menjadi data latih dan data uji sebagai bagian dari tahap prapemrosesan data untuk mempersiapkan proses pemodelan. Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk mengevaluasi kinerja model pada data yang belum pernah dilihat sebelumnya. Pembagian dilakukan dengan beberapa rasio, yaitu 60:40, 70:30, dan 80:20, di mana 60, 70, dan 80 merupakan data latih serta 40, 30, dan 20 merepresentasikan data uji. Variasi rasio ini bertujuan untuk menganalisis pengaruh jumlah data latih terhadap performa model. Selain itu digunakan pula skema pengujian pembagian data menggunakan *k-fold cross validation* dengan variasi nilai k dari 2 hingga 10 sebagai skenario alternatif dalam pembagian data. Kedua pendekatan ini dibandingkan untuk mengetahui skema pembagian data yang memberikan performa model paling optimal dan stabil.

#### e. Penyeimbangan Data

Dataset penelitian ini terdiri dari beberapa kelas data yang tidak seimbang. Jika kondisi ini tidak ditangani, model klasifikasi akan cenderung lebih fokus mengenali pola pada kelas mayoritas sehingga kemampuan dalam memprediksi kelas minoritas menjadi menurun. Upaya yang dilakukan untuk mengatasi hal tersebut adalah dengan menerapkan metode *Synthetic Minority Oversampling Technique* (SMOTE). Teknik ini menambahkan jumlah sampel pada kelas minoritas melalui pembuatan data sintesis baru.

Secara teoritis, SMOTE bekerja dengan memilih sampel dari kelas minoritas, kemudian mencari sejumlah tetangga terdekatnya (*K-Nearest Neighbors*) [15]. Selanjutnya, data sintesis dibangkitkan dengan melakukan interpolasi antara sampel asli dan salah satu tetangga terdekatnya pada ruang fitur. Proses ini menghasilkan data baru yang berada di antara dua titik sampel minoritas, sehingga dapat memperluas representasi distribusi kelas minoritas tanpa sekadar melakukan duplikasi data. Penentuan tetangga dilakukan menggunakan jarak Euclidean yang dihitung menggunakan persamaan 1.

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2} \quad (1)$$

Dengan  $x$  dan  $y$  merupakan nilai fitur ke- $n$  dari data, serta  $n$  adalah jumlah dimensi data. Selanjutnya, data sintesis dibentuk menggunakan persamaan 2.

$$x_{syn} = x_i + (x_{knn} + x_i) \times \delta \quad (2)$$

Dengan  $x_i$  merupakan data asli kelas minoritas,  $x_{knn}$  adalah tetangga terdekat dari data asli, dan  $\delta$  adalah nilai acak dalam rentang 0 hingga 1. Data sintesis dihasilkan dari kombinasi linier antara data minoritas dan tetangga terdekat dapat memperluas variasi pola

data yang dipelajari oleh model sehingga karakteristik kelas minoritas dapat lebih dikenali oleh model.

### 2.3. Extremely Randomized Trees (Extra Trees)

*Extremely Randomized Trees* atau dikenal dengan *Extra Trees* merupakan varian *ensemble* dari algoritma pohon keputusan. Proses pembentukan pohon melibatkan randomisasi atau pengacakan tingkat tinggi, baik pada pemilihan fitur ataupun titik pemisahan (titik *split*) [16]. Randomisasi tingkat tinggi mengakibatkan pembentukan pohon yang sangat acak dan strukturnya tidak saling terhubung. *Extra Trees* sangat efisien, tahan *overfitting*, dan mampu menghasilkan generalisasi yang baik pada dataset berukuran kecil hingga menengah [17]. *Extra Trees* memiliki empat tahapan utama, antara lain sebagai berikut.

#### a. Pembentukan Sampel Data

Data yang digunakan untuk membangun pohon merupakan seluruh data pelatihan asli, bukan replika *bootstrap* dan tidak menggunakan *resampling*.

#### b. Pemilihan Fitur Acak

Pemilihan subset fitur secara acak dari total fitur yang ada. Hal ini bertujuan untuk meningkatkan keberagaman antar pohon sehingga dapat mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi.

#### c. Penentuan *Split* Secara Acak dan Pembangunan Pohon

Setelah fitur dipilih secara acak, ditentukan nilai batas pemisahan (*threshold*) secara acak pula. Selanjutnya *threshold* acak dievaluasi menggunakan Gini Indeks. Persamaan Gini Indeks disajikan dalam persamaan 3 [18].

$$Gini(D) = 1 - \sum_{i=1}^m (p_i)^2 \quad (3)$$

Dengan  $D$  adalah dataset,  $m$  merupakan jumlah kelas variabel atribut,  $i$  sebagai kelas atribut, dan  $p_i$  merupakan perbandingan jumlah kelas pada atribut ke- $i$  terhadap total kelas atribut. Setelah menghitung Gini Indeks, selanjutnya adalah menentukan atribut dalam *node* menggunakan persamaan 4 [18].

$$Gini_A(D) = \frac{|D_1|}{D} gini(D_1) + \frac{|D_2|}{D} gini(D_2) \quad (4)$$

Dengan  $A$  merupakan total nilai bobot dari indeks Gini pada suatu atribut, dan  $D_1, D_2$  adalah jumlah data pada dataset  $D$  yang telah dibagi berdasarkan atribut  $A$  menjadi dua subset yaitu  $D_1$  dan  $D_2$ . Melalui persamaan tersebut, diperoleh nilai total bobot indeks Gini untuk setiap fitur, termasuk fitur  $A$  dan fitur lainnya. Setelah seluruh fitur dihitung nilai indeks Gini, langkah selanjutnya adalah memilih fitur dengan nilai terkecil sebagai dasar pengambilan keputusan pertama dalam pembentukan pohon.

#### d. *Agregating*

Tahapan ini merupakan proses menggabungkan hasil prediksi yang didapatkan oleh seluruh pohon untuk

memperoleh prediksi akhir. Hasil prediksi akhir didapatkan melalui *majority voting*, yaitu kelas yang paling sering dipilih oleh semua pohon sehingga model akhir yang didapatkan dapat lebih stabil dan akurat. *Majority voting* dapat dituliskan seperti persamaan 5.

$$\hat{y}_{final} = Mode(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N) \quad (5)$$

Dengan  $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N$  merupakan prediksi yang dihasilkan masing-masing pohon keputusan,  $N$  adalah jumlah total pohon, dan *Mode* adalah nilai yang paling sering muncul.

### 2.4. Accelerated Particle Swarm Optimization (APSO)

*Accelerated Partikel Swarm Optimization* (APSO) merupakan variasi dari *Particle Swarm Optimization* (PSO). PSO sendiri memiliki dua komponen utama yang digunakan sebagai acuan, yaitu *global best* dan *personal best*. Sedangkan APSO memiliki komponen yang lebih sederhana yaitu hanya menggunakan *global best* yang ditambahkan dengan elemen acak [13].

Solusi kombinasi *hyperparameter* dalam APSO diibaratkan sebuah partikel. Partikel tersebut berada dalam ruang pencarian yang luas. Terdapat dua konsep utama yang digunakan untuk menemukan partikel yang paling optimal yaitu kecepatan dan posisi partikel. Perubahan kecepatan partikel dirumuskan oleh persamaan 6 [13].

$$v_j^{(i+1)} = v_j^{(i)} + \alpha \epsilon_n + \beta (g^* - x_j^{(i)}) \quad (6)$$

Dengan  $v_j^{(i)}$  menyatakan kecepatan partikel  $j$  pada iterasi ke- $i$ ,  $x_j^{(i)}$  menunjukkan posisi partikel  $j$  pada iterasi yang sama,  $\alpha$  merupakan parameter untuk mengontrol besar kecilnya *noise*,  $\beta$  merupakan parameter untuk mengontrol kekuatan tarikan ke *global best*. Setelah nilai kecepatan partikel diperbarui, pembaruan posisi partikel selanjutnya ditentukan berdasarkan persamaan 7 [13].

$$x_j^{(i+1)} = x_j^{(i)} + v_j^{(i+1)} \quad (7)$$

Untuk meningkatkan konvergensi, kedua persamaan tersebut dilebur menjadi satu sehingga menghasilkan persamaan 8 [13], [19].

$$x_j^{(i+1)} = (1 - \beta)x_j^{(i)} + \beta g^* + \alpha \epsilon_n \quad (8)$$

Dalam persamaan tersebut, posisi baru ditentukan menggunakan kombinasi antara posisi lama dan *global best* yang ditambahkan dengan elemen acak. Semakin besar nilai  $\beta$  partikel akan lebih cepat tertarik ke dalam *global best*. Sedangkan jika  $\alpha$  semakin besar, partikel akan bergerak secara lebih acak.

### 2.5. Evaluasi Model

Evaluasi model klasifikasi dilakukan menggunakan konsep dasar *confusion matrix*, yaitu representasi yang menggambarkan perbandingan antara hasil klasifikasi dengan label aktual. Menggunakan konsep dasar tersebut, pengukuran kinerja model dilakukan

menggunakan akurasi, presisi, *recall*, spesifisitas, *F1-score*, dan *area under curve* (AUC) [20].

a. Akurasi

Akurasi mengukur presentase klasifikasi benar dari seluruh data uji. Persamaannya ditunjukkan oleh persamaan 9.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

b. Presisi

Presisi merepresentasikan kemampuan model dalam memberikan prediksi positif yang benar. Persamaannya ditunjukkan oleh persamaan 10.

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

c. Recall

*Recall* merepresentasikan tingkat keberhasilan model dalam mendeteksi data positif dari total data positif. Persamaannya ditunjukkan oleh persamaan 11.

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

d. Spesifisitas

Metrik ini digunakan untuk mengukur ketepatan model dalam mengidentifikasi kelas negatif. Persamaannya ditunjukkan oleh persamaan 12.

$$Spesifisitas = \frac{TN}{TN+FP} \quad (12)$$

e. *F1-Score*

*F1-score* digunakan untuk mengevaluasi performa model dengan mempertimbangkan keseimbangan antara presisi dan *recall* dalam mengidentifikasi data positif. Persamaannya ditunjukkan oleh persamaan 13.

$$F1 \text{ score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (13)$$

f. *Area Under Curve* (AUC)

AUC merepresentasikan hubungan antara *True Positive Rate* dan *False Positive Rate*. Nilai ini merepresentasikan peluang model membedakan secara acak satu data positif dan satu data negatif. Persamaannya ditunjukkan oleh persamaan 14.

$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN} \quad (14)$$

### 3. Hasil dan Pembahasan

Bagian ini memaparkan hasil penelitian dan analisis pada klasifikasi sub tipe anemia menggunakan metode *extremely randomized trees* atau *extra trees* dengan optimasi *accelerated particle swarm optimization* (APSO).

#### 3.1. Pengumpulan Data

Pengumpulan data dilakukan di Laboratorium Patologi Klinik RSUD Haji Provinsi Jawa Timur. Proses pengambilan data dilakukan selama satu bulan mulai tanggal 6 hingga 31 maret 2026. Pengambilan data berada di bawah pengawasan pembimbing lapangan

Laboratorium Patologi Klinik sehingga menjamin data yang dikumpulkan benar dan valid.

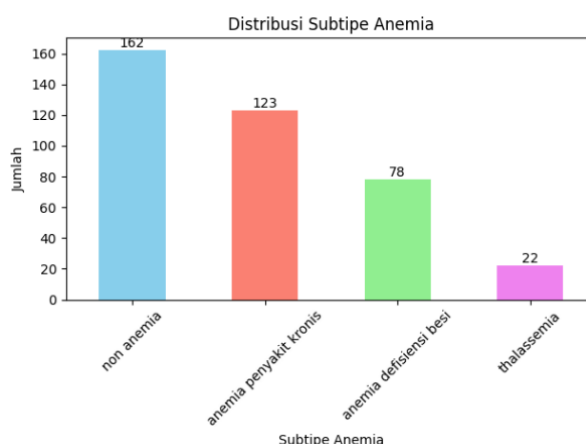
Dataset merupakan data sekunder periode tahun 2025 hingga awal maret 2026. Hal ini dikarenakan sistem manajemen rumah sakit hanya menyimpan data yang dibutuhkan penelitian ini mulai tahun 2025. Jumlah data yang berhasil dikumpulkan adalah sebanyak 385 data. Kategori sub tipe anemia yang diperoleh antara lain, anemia defisiensi besi, *thalassemia*, anemia penyakit kronis, dan non anemia. Dengan demikian klasifikasi yang dilakukan merupakan klasifikasi multikelas atau multilabel. Tabel 2 di bawah ini menunjukkan daftar fitur dataset.

Tabel 2. Variabel Penelitian

No	Nama Variabel	Keterangan
1.	Inisial Pasien	Inisial nama pasien
2.	Jenis Kelamin	Jenis kelamin pasien (P/L)
3.	Usia	Usia pasien dalam tahun
4.	Hb	Kadar hemoglobin
5.	HCT	Persentase volume eritrosit terhadap total volume darah
6.	RBC	Jumlah sel darah merah
7.	MCV	Volume rata-rata eritrosit
8.	MCH	Jumlah hemoglobin per eritrosit
9.	MCHC	Konsentrasi hemoglobin dalam eritrosit
10.	RDW	Variasi ukuran sel darah merah
11.	Sub tipe Anemia	Jenis sub tipe anemia yang diderita pasien.

#### 3.2. Praproses Data

Praproses data diawali dengan pemeriksaan distribusi data. Distribusi data berdasarkan sub tipe anemia ditunjukkan oleh gambar 2.



Gambar 2. Distribusi Sub tipe Anemia

Melalui visualisasi diagram batang gambar 2, diketahui bahwa jumlah data terbanyak adalah kategori non anemia yang diikuti anemia penyakit kronis kemudian anemia defisiensi besi serta yang paling sedikit adalah *thalassemia*. Jumlah data kategori *thalassemia* menunjukkan ketimpangan besar dengan kategori lainnya. Hal ini dapat menyebabkan model cenderung

mempelajari kelas mayoritas. Oleh sebab itu, diperlukan langkah untuk menyeimbangkan data menggunakan metode seperti SMOTE. Penyesuaian distribusi data hanya dilakukan pada data *training* sebagai upaya mencegah *data leakage*.

Selanjutnya merupakan tahap pembersihan data meliputi pengecekan dan penanganan *missing value*, serta data duplikat. Berdasarkan hasil pengecekan, tidak ditemukan nilai yang hilang dan duplikasi pada dataset seperti yang ditunjukkan oleh gambar 2 dan 3.

Total missing value: 0

Gambar 3. Jumlah Missing Value

Jumlah data duplikat: 0

Gambar 3. Jumlah Duplikasi Data

Dengan demikian data yang telah melewati tahap pembersihan dinyatakan siap untuk memasuki tahapan selanjutnya, yaitu *encoding* variabel kategorikal, yakni variabel jenis kelamin dan sub tipe anemia. *Encoding* bertujuan mengubah data kategori menjadi representasi angka sehingga dapat diproses oleh model *machine learning*. Tabel 3 berikut menunjukkan beberapa data sebelum *encoding* dilakukan.

Tabel 3. Data Sebelum *Encoding*

Inisial Pasien	Jenis Kelamin	...	Subtipe Anemia
Ny. S	Perempuan	...	thalassemia
Ny. AF	Perempuan	...	anemia defisiensi besi
Sdr. AA	Laki-laki	...	non anemia
Ny. NA	Perempuan	...	anemia defisiensi besi
Tn. HS	Laki-laki	...	anemia penyakit kronis

Sementara itu, dibawah ini adalah tabel 4 yang menunjukkan beberapa data setelah *encoding* dilakukan terhadap kedua variabel tersebut. Terlihat bahwa variabel yang sebelumnya bertipe teks kategorikal berubah menjadi representasi numerik.

Tabel 4. Data Sesudah *Encoding*

Inisial Pasien	Jenis Kelamin	...	Subtipe Anemia
Ny. S	0	...	0
Ny. AF	0	...	1
Sdr. AA	1	...	3
Ny. NA	0	...	1
Tn. HS	1	...	2

Setelah proses *encoding* dilakukan, data akan masuk ke tahap pembagian data sesuai skenario yang telah ditentukan sebelumnya, yaitu 60:40, 70:30, 80:20, serta 2 – 10 *fold cross validation*. Skenario ini bertujuan untuk mengevaluasi performa model secara lebih komprehensif dengan berbagai proporsi data latih dan data uji, sehingga dapat diketahui tingkat konsistensi dan kemampuan generalisasi model. Tabel 5 menyajikan beberapa skema pembagian data yang dilakukan.

Tabel 5. Distribusi Data Latih dan Data Uji

Skema Pembagian Data	Jumlah Data Latih	Jumlah Data uji
60 : 40	231	154

Skema Pembagian Data	Jumlah Data Latih	Jumlah Data uji
70 : 30	269	116
80 : 20	308	77
2-fold	193	192
3-fold	257	128
...	...	...
10-fold	347	38

Data *training* yang telah dipisahkan selanjutnya dilakukan penyeimbangan jumlah menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*) dengan memanfaatkan pustaka bawaan python yakni *imbalence-learn*. Tabel 6 berikut menampilkan contoh perbedaan jumlah distribusi kelas data latih antara sebelum dan sesudah SMOTE pada skema pembagian data 80:20.

Tabel 6. Perbedaan Jumlah Data Latih Sebelum dan Sesudah SMOTE pada skema 80:20

Kategori Kelas	Sebelum SMOTE	Sesudah SMOTE
Thalassemia	18	130
Anemia Defisiensi Besi	62	130
Non Anemia	130	130
Anemia Penyakit Kronis	98	130

### 3.3. Pemodelan

Penerapan metode *Extremely Randomized Trees* atau *Extra trees* dilakukan dengan memanfaatkan pustaka bawaan python yaitu *scikit-learn*. Penggunaan pustaka ini memungkinkan proses pelatihan model dilakukan dengan lebih sistematis serta memudahkan dalam pengaturan berbagai parameter. Sedangkan metode optimasi *Accelerated Particle Swarm Optimization* (APSO) dilakukan secara *scratch* dengan mendefinisikan algoritma secara langsung dalam kode program.

Beberapa hyperparameter yang dioptimalkan antara lain, *n\_estimators* yang berfungsi untuk menentukan banyaknya pohon keputusan yang dibangun dalam model. Selanjutnya, *min\_samples\_split* mengatur jumlah minimum sampel yang diperlukan agar suatu node dapat dilakukan proses pemisahan. Parameter *min\_samples\_leaf* digunakan untuk menetapkan jumlah minimal sampel yang harus berada pada setiap node daun (*leaf*). Sementara itu, *max\_features* berperan dalam menentukan jumlah fitur yang dipertimbangkan ketika mencari pemisahan terbaik pada setiap *node*. Dengan mengoptimasi *hyperparameter* tersebut, diharapkan model dapat menemukan kombinasi yang mampu meningkatkan performa klasifikasi secara keseluruhan.

Dalam penerapan APSO diperlukan penentuan nilai untuk jumlah partikel (*particle size*) dan jumlah iterasi (*maximum iteration*) yang berfungsi mengatur proses pencarian solusi optimal. Pada penelitian ini, jumlah partikel yang digunakan sebanyak 10, sementara jumlah iterasi yang ditetapkan adalah 20 iterasi. Penetapan tersebut didasarkan pada pertimbangan

efisiensi komputasi, kemampuan konvergensi yang cepat dari metode APSO, serta untuk menghindari *overfitting* pada proses optimasi. Jumlah partikel yang tidak terlalu besar dipilih untuk menjaga waktu komputasi tetap efisien, namun masih mampu merepresentasikan ruang pencarian solusi secara memadai. Sementara itu, jumlah iterasi yang terbatas dinilai cukup untuk mencapai kondisi konvergen.

Sebagai bagian dari proses optimasi, penentuan konfigurasi parameter APSO menjadi aspek penting yang memengaruhi kinerja algoritma dalam menemukan solusi optimal. Adapun konfigurasi parameter APSO dalam penelitian ini ditunjukkan dalam tabel 7.

Tabel 7. Parameter APSO

Nama Parameter	Nilai
Alpha	0.3
Beta	0.7

APSO memiliki komponen berupa fungsi *fitness* yang berperan dalam mengevaluasi setiap partikel yang dieksplorasi sehingga APSO dapat menemukan partikel yang merupakan kombinasi *hyperparameter* paling optimal. Fungsi *fitness* yang digunakan adalah  $1 - accuracy$ . Oleh karena itu, nilai *fitness* yang rendah mengindikasikan performa model yang lebih optimal. Berikut ini merupakan potongan kode skrip untuk metode optimasi APSO.

**Kode Program APSO**

```
for iter in range(n_iter):
    # Evaluasi fitness
    for i in range(n_particles):
        score = fitness_function(particles[i],
                                model_type, X, y)
        if score < gbest_score:
            gbest_score = score
            gbest = particles[i].copy()
    # Update posisi partikel
    for i in range(n_particles):
        epsilon = np.random.rand(dim)
        particles[i] = (
            (1 - beta) * particles[i] +
            beta * gbest +
            alpha * epsilon)
```

Dengan menggunakan metode APSO, kombinasi *hyperparameter Extremely Randomized Trees* berhasil ditemukan. Tabel 8 berikut ini menunjukkan kombinasi *hyperparameter* bawaan dan optimal yang berhasil ditemukan.

Tabel 8. Perbandingan Kombinasi *Hyperparameter*

Nama Hyperparameter	Nilai Default	Nilai Optimal
<i>n_estimators</i>	100	446,09
<i>max_depth</i>	None	53,36
<i>min_samples_split</i>	2	88,08
<i>min_samples_leaf</i>	1	9,24
<i>max_features</i>	sqrt	0,69

Tabel tersebut menunjukkan bahwa optimasi menggunakan APSO menghasilkan konfigurasi *hyperparameter* yang lebih optimal pada model *Extra Trees* dibandingkan nilai *default*. Dengan menggunakan kombinasi *hyperparameter* optimal

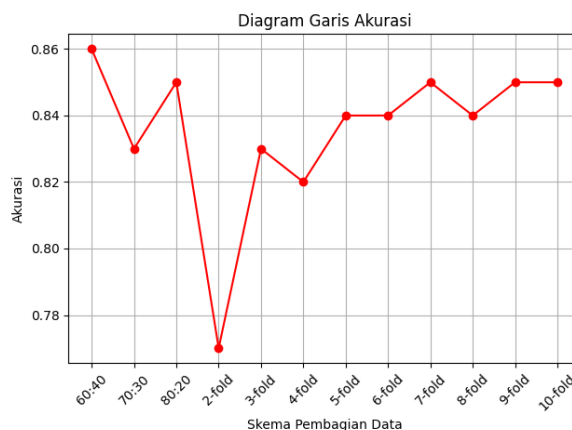
algoritma *Extremely Randomized Trees* dilatih untuk dapat melakukan klasifikasi sub tipe anemia. Tabel 9 berikut ini menunjukkan beberapa hasil klasifikasi.

Tabel 9. Hasil Klasifikasi

Inisial Pasien	Jenis Kelamin	Usia	Hb	...	Subtipe Anemia	Hasil Klasifikasi
Ny. S	0	49	10,3	...	0	0
Ny. U	0	47	9,7	...	0	0
Tn. P	1	60	14,5	...	2	2
Ny. N	0	34	9,4	...	1	1
Tn. H	1	54	6,4	...	3	3

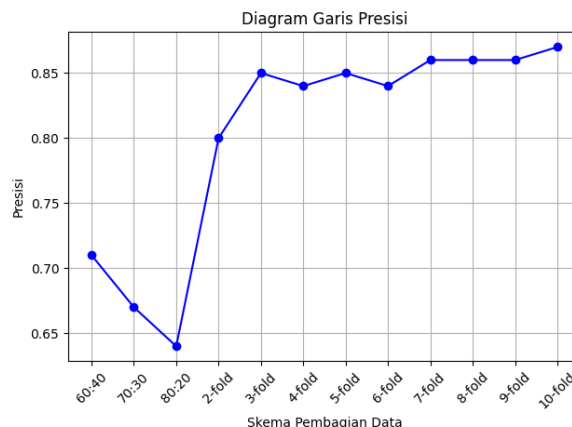
3.4. Evaluasi Model

Berdasarkan beberapa skenario pembagian data yang digunakan pada klasifikasi sub tipe anemia menggunakan *Extremely Randomized Trees* dengan *Accelerated Particle Swarm Optimization*. Beberapa gambar dibawah ini menunjukkan nilai evaluasi model klasifikasi sub tipe anemia.



Gambar 4. Diagram Akurasi

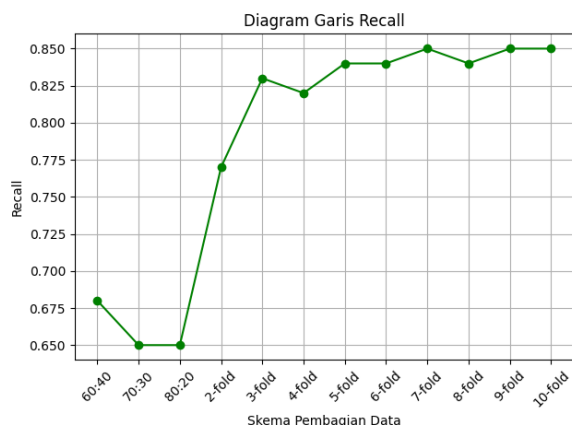
Grafik akurasi pada gambar 4 menunjukkan bahwa performa model cenderung stabil pada berbagai skenario pengujian. Akurasi tertinggi dicapai pada skenario pembagian data 60:40 dengan nilai 0,86, sedangkan pada k-fold cross validation akurasi berada pada rentang 0,82–0,85.



Gambar 5. Diagram Presisi

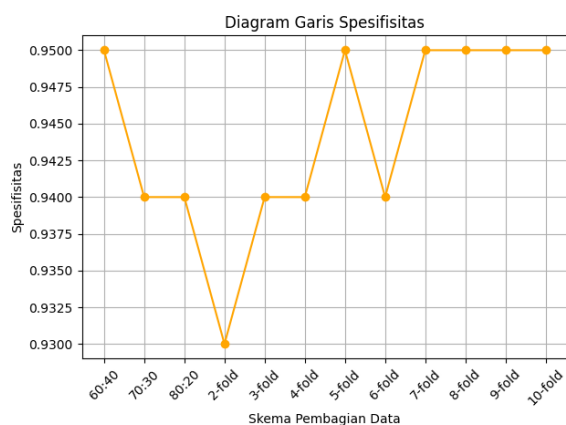
Grafik presisi pada gambar 5 menunjukkan adanya peningkatan performa model pada skenario *k-fold cross*

validation dibandingkan pembagian data sederhana. Nilai presisi tertinggi mencapai 0,87 pada 10-fold.



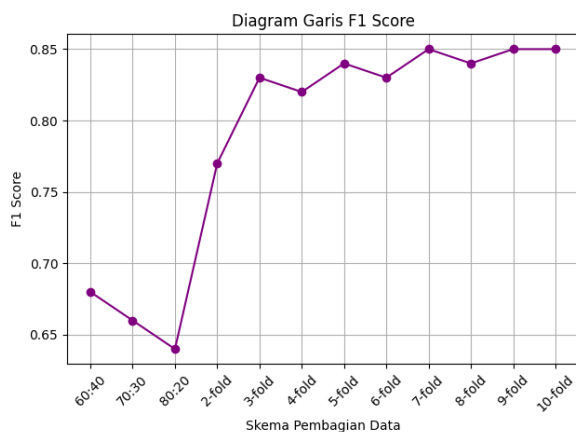
Gambar 6. Diagram Recall

Berdasarkan grafik recall pada Gambar 6, kemampuan model dalam mendeteksi data pada tiap kelas mengalami peningkatan pada skenario *k-fold cross validation*. Nilai *recall* meningkat dari 0,65 pada pembagian data awal menjadi sekitar 0,85 pada 10-fold.



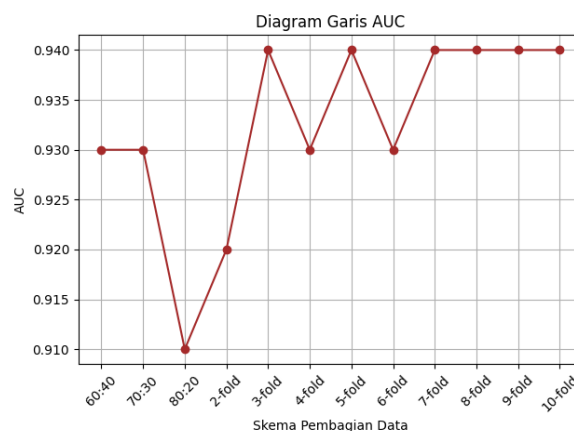
Gambar 7. Diagram Spesifisitas

Dari grafik spesifisitas pada Gambar 7 terlihat bahwa model memiliki kemampuan yang unggul dalam mengidentifikasi kelas negatif, dengan nilai spesifisitas yang tidak hanya tinggi tetapi juga stabil, yakni selalu berada di atas 0,90 pada setiap skenario pengujian.



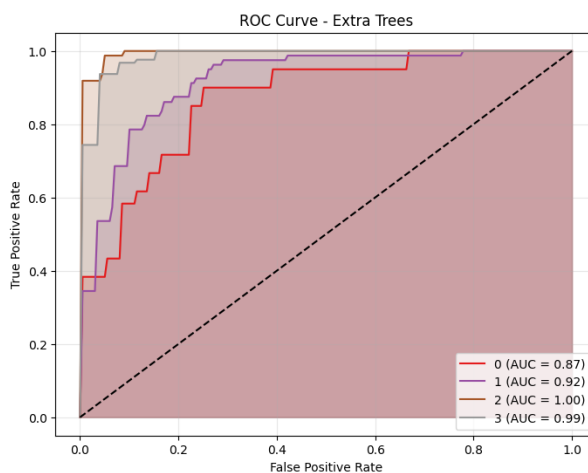
Gambar 8. Diagram F1-Score

Grafik *F1 Score* pada gambar 8 menunjukkan adanya peningkatan keseimbangan antara presisi dan *recall*, terutama pada skenario *k-fold cross validation*. Nilai *F1-score* meningkat dari sekitar 0,64 pada pembagian data awal menjadi 0,85 pada 10-fold.



Gambar 9. Diagram Perubahan AUC

Grafik AUC pada Gambar 9 menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam membedakan antar kelas, yang dapat dilihat dari nilai AUC yang tinggi sehingga menandakan performa model dalam melakukan klasifikasi sudah sangat optimal yang konsisten tinggi di kisaran 0,91–0,94.



Gambar 10. Grafik ROC Extra Trees

Berdasarkan hasil pengujian, diketahui bahwa model klasifikasi anemia menggunakan metode *Extra Trees* dengan optimasi APSO menunjukkan performa yang baik dan konsisten pada berbagai skenario pembagian data maupun *k-fold cross validation*. Akurasi tertinggi diperoleh pada skenario 60:40, tapi nilai metrik lainnya masih relatif rendah khususnya pada presisi, *recall*, dan *F1 Score*. Sedangkan pada skenario 10-fold, seluruh metrik evaluasi menunjukkan nilai yang tinggi dan stabil dengan akurasi, *recall*, dan *F1-score* 0,85, presisi 0,87, spesifisitas 0,95, serta AUC sebesar 0,94.

Penelitian ini membandingkan performa *Extra Trees* tanpa optimasi dengan *Extra Trees* yang dioptimasi menggunakan APSO untuk mengetahui sejauh mana proses optimasi parameter mampu meningkatkan

kinerja model dalam melakukan klasifikasi subtype anemia. Tabel 10 berikut ini merupakan perbandingan performa model sebelum dioptimasi dan sesudah dioptimasi.

Tabel 10. Perbandingan Performa Model

Metrik Evaluasi	<i>Extra Trees</i>	<i>Extra Trees + APSO</i>
Akurasi	0,83	0,85
Presisi	0,66	0,87
<i>Recall</i>	0,68	0,85
Spesifisitas	0,94	0,95
<i>F1-Score</i>	0,66	0,85
AUC	0,94	0,94

Berdasarkan hasil perbandingan performa model sebelum dan sesudah optimasi, terlihat bahwa penerapan metode *Accelerated Particle Swarm Optimization* (APSO) memberikan peningkatan kinerja yang cukup konsisten pada sebagian besar metrik evaluasi. Model *Extra Trees* yang telah dioptimasi menunjukkan peningkatan terutama pada akurasi, presisi, *recall*, dan *F1-score*, yang mengindikasikan bahwa optimasi parameter mampu meningkatkan kemampuan model dalam memahami pola data.

Peningkatan pada nilai *recall* menjadi salah satu indikator penting, karena menunjukkan bahwa model hasil optimasi lebih mampu mendeteksi data pada setiap kelas. Nilai presisi yang meningkat menunjukkan kemampuan model yang lebih baik dalam meminimalkan kesalahan prediksi positif. Selain itu, nilai *F1-score* yang meningkat mengindikasikan bahwa keseimbangan antara presisi dan *recall* semakin baik.

Di sisi lain, kestabilan nilai spesifisitas dan AUC yang tinggi setelah optimasi menandakan bahwa kemampuan model yang sangat baik dalam membedakan setiap kelas. Hal ini penting karena peningkatan performa pada satu metrik terkadang berpotensi menurunkan metrik lainnya, namun pada hasil ini justru terlihat bahwa optimasi mampu meningkatkan performa tanpa mengorbankan aspek metrik lainnya.

Tabel 11. Perbandingan Performa Algoritma

Metrik Evaluasi	<i>Decision Tree + APSO</i>	<i>Random Forest + APSO</i>	<i>Extra Trees + APSO</i>
Akurasi	0,84	0,85	0,85
Presisi	0,85	0,85	0,87
<i>Recall</i>	0,84	0,85	0,85
Spesifisitas	0,95	0,95	0,95
<i>F1-Score</i>	0,84	0,85	0,85
AUC	0,91	0,93	0,94

Berdasarkan tabel 11, model *extra trees* yang dioptimasi APSO menunjukkan performa paling unggul dibandingkan *decision tree* dan *random forest*. Meskipun memiliki nilai akurasi yang setara dengan *random forest*, model ini memiliki presisi tertinggi, yaitu 0,87 yang menunjukkan kemampuan lebih baik dalam meminimalkan kesalahan prediksi positif. Selain itu, *extra trees* juga unggul dalam nilai AUC yang menunjukkan kemampuan paling baik dalam membedakan kelas secara keseluruhan. Dengan

demikian, *extra trees* dengan optimalisasi *hyperparameter* APSO merupakan model paling optimal dalam penelitian ini.

#### 4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa tujuan penelitian telah tercapai, yaitu diperolehnya kombinasi *hyperparameter* optimal pada metode *Extra Trees* dalam mengklasifikasikan subtype anemia (anemia defisiensi besi, thalassemia, anemia penyakit kronis, dan non-anemia). Selain itu, penggunaan *Accelerated Particle Swarm Optimization* terbukti memberikan pengaruh signifikan dalam meningkatkan performa model, yang ditunjukkan oleh peningkatan nilai akurasi, presisi, *recall*, spesifisitas, *F1-score*, dan AUC dibandingkan tanpa optimasi. Kombinasi *Extra Trees + APSO* menjadi kontribusi utama penelitian ini karena mampu menghasilkan model klasifikasi multikelas yang lebih optimal, stabil, dan memiliki kemampuan diskriminasi yang baik, terutama terlihat dari nilai AUC yang tinggi. Pengujian juga menunjukkan bahwa teknik *10 - fold cross validation* memberikan hasil evaluasi paling konsisten dan representatif terhadap distribusi data.

Meskipun demikian, penelitian ini masih terbatas pada penggunaan dataset dari satu sumber. Oleh karena itu penelitian selanjutnya disarankan untuk menggunakan dataset yang lebih beragam serta membandingkan dengan algoritma maupun metode optimasi lain.

#### Daftar Rujukan

- [1] H. Nasruddin, R. Faisal Syamsu, and D. Permatasari, "Angka Kejadian Anemia Pada Remaja di Indonesia," *Cerdika J. Ilm. Indones.*, vol. 1, no. 4, pp. 357–364, 2021, doi: 10.59141/cerdika.v1i4.66.
- [2] dr. F. R. Makarim, "Ini 5 Akibat Fatal yang Bisa Muncul Akibat Anemia," halodoc. [Online]. Available: [https://www.halodoc.com/artikel/ini-5-akibat-fatal-yang-bisa-muncul-akibat-anemia?srsltid=AfmBOopmz4D95EVs-SEs9a6hs0QFc6Wq1nL05fs2cWFWmZKOZ652u9\\_2](https://www.halodoc.com/artikel/ini-5-akibat-fatal-yang-bisa-muncul-akibat-anemia?srsltid=AfmBOopmz4D95EVs-SEs9a6hs0QFc6Wq1nL05fs2cWFWmZKOZ652u9_2)
- [3] Siti Naila Sya'bani, Andriyani Andriyani, and Nurmalia Lusida, "Tinjauan Anemia pada Remaja Putri : Analisis Faktor Resiko dan Implikasi Kesehatan Jangka Panjang," *OBAT J. Ris. Ilmu Farm. dan Kesehat.*, vol. 3, no. 3, pp. 255–269, 2025, doi: 10.61132/obat.v3i3.1353.
- [4] *Survei Kesehatan Indonesia 2023 (SKI)*. Kementerian Kesehatan RI, 2023. [Online]. Available: <https://kemkes.go.id/id/survei-kesehatan-indonesia-ski-2023>
- [5] *Profil Kesehatan Jawa Timur 2024*. Dinas Kesehatan Provinsi Jawa Timur, 2024. [Online]. Available: [https://dinkes.jatimprov.go.id/index.php/source/bankdata&statis&sekretariat&PPA&105\\_1776148400\\_profil-kesehatan-provinsi-jawa-timur-tahun-2024.pdf](https://dinkes.jatimprov.go.id/index.php/source/bankdata&statis&sekretariat&PPA&105_1776148400_profil-kesehatan-provinsi-jawa-timur-tahun-2024.pdf)
- [6] *Profil Kesehatan Kota Surabaya 2024*. Dinas Kesehatan Kota Surabaya, 2024. [Online]. Available: [https://dinkes.surabaya.go.id/portal\\_dinkes/d/dkk/dokumen?kategори=3](https://dinkes.surabaya.go.id/portal_dinkes/d/dkk/dokumen?kategори=3)
- [7] M. N. Garcia-Casal, O. Dary, M. E. Jefferds, and S. R. Pasricha, "Diagnosing anemia: Challenges selecting methods, addressing underlying causes, and implementing actions at the public health level," *Ann. N. Y. Acad. Sci.*, vol. 1524, no. 1, pp. 37–50, 2023, doi: 10.1111/nyas.14996.
- [8] A. Priandika and A. R. Isnain, "Penerapan Teknik Ensemble Learning untuk Klasifikasi Jenis-jenis Anemia," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 5, no. July, pp. 972–980, 2025.

- [9] D. T. Wiyanti and Ainurrohmah, "Analisis Performa Algoritma Decision Tree , Naïve Bayes , K- Nearest Neighbor untuk Klasifikasi Zona Daerah Risiko Covid-19 di Indonesia Performance Analysis Of Decision Tree , Naïve Bayes , K- Nearest Neighbor Algorithm For Covid-19 Risk Zone Classificati," *J. Teknoogi Inf. dan Ilmu Komput.*, vol. 10, no. 1, 2023, doi: 10.25126/jtiik.2023105935.
- [10] S. Baladram, "Extra Trees, Explained: A Visual Guide with Code Examples," Medium. Accessed: Nov. 14, 2025. [Online]. Available: <https://medium.com/@samyladram/extra-trees-explained-a-visual-guide-with-code-examples-4c2967cedc75>
- [11] A. La Moglia and K. M. Almustafa, "Breast cancer prediction using machine learning classification algorithms," *J. Intell. Med.*, vol. 11, no. 2025, 2024, doi: 10.1016/j.ibmed.2024.100193.
- [12] H. F. Soon, A. Amir, H. Nishizaki, N. A. H. Zahri, L. M. Kamarudin, and S. N. Azemi, "Evaluating Tree-based Ensemble Strategies for Imbalanced Network Attack Classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 1, 2024, doi: 10.14569/IJACSA.2024.01501111.
- [13] X. S. Yang, S. Deb, and S. Fong, "Accelerated particle swarm optimization and support vector machine for business optimization and applications," in *Communications in Computer and Information Science*, 2011, pp. 53–66. doi: 10.1007/978-3-642-22185-9\_6.
- [14] A. H. M. Emara, G. Atteia, and J. H. Alkhateeb, "Fine Tuning Hyperparameters of Deep Learning Models Using Metaheuristic Accelerated Particle Swarm Optimization Algorithm," *IEEE Access*, vol. 13, no. June, pp. 134506–134518, 2025, doi: 10.1109/ACCESS.2025.3591403.
- [15] M. Sulistiyono, Y. Pristyanto, S. Adi, and G. Gumelar, "Implementasi Algoritma Synthetic Minority Over-Sampling Technique untuk Menangani Ketidakseimbangan Kelas pada Dataset Klasifikasi," *Sistemasi*, vol. 10, no. 2, p. 445, 2021, doi: 10.32520/stmsi.v10i2.1303.
- [16] S. A. R. Manaf, A. Fitrianto, and A. M. Soleh, "Perbandingan Algoritma Pohon dengan Beberapa Skenario Pelabelan untuk Analisis Sentimen pada Aplikasi Milik Pemerintah/BUMN," *J. Edukasi dan Penelit. Inform.*, vol. 10, no. 1, pp. 24–32, 2024.
- [17] Yanuardi, F. F. Basri, and M. L. Aksani, "Klasifikasi Kepribadian Berdasarkan Dimensi Ekstraversi Berbasis Data Mining Menggunakan Extremely Randomized Trees," *J. Ilm. Tek. Inform.*, vol. 14, no. 2, pp. 229–237, 2025, [Online]. Available: <https://share.google/Pf5PRcHrIDGkny0Ki>
- [18] M. Walidin, "Analisis Sentimen Opini Pelanggan Berbasis Aspek Aplikasi Halodoc Menggunakan Metode Extra Tree Classifier," Universitas Komputer Indonesia, 2021. [Online]. Available: <https://elibrary.unikom.ac.id/id/eprint/5673/>
- [19] M. A. Tariq *et al.*, "Comparative assessment of differently randomized accelerated particle swarm optimization and squirrel search algorithms for selective harmonics elimination problem," *Sci. Rep.*, pp. 1–11, 2024, doi: 10.1038/s41598-024-62686-9.
- [20] "Evaluation Metrics in Machine Learning," *geeksforgeeks*. Accessed: Sep. 10, 2025. [Online]. Available: <https://www.geeksforgeeks.org/machine-learning/metrics-for-machine-learning-model/>