Analisis Perbandingan Algoritma *Support Vector Machine* dan *Random Forest* dalam Klasifikasi Jenis Penyakit Malaria pada Citra Darah Manusia Berbasis YOLO 11

Tugas Akhir

Oleh:
Muhammad Rezky Fatya (4212201032)
Muhammad Fikri (4212201091)

Program Studi Teknik Mekatronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2026

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : "Analisis Perbandingan Algoritma *Support Vector Machine* dan *Random Forest* dalam Klasifikasi Jenis Penyakit Malaria pada Citra Darah Manusia Berbasis YOLO 11 " adalah **hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan,** dan **bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.** Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 19 Januari 2026

Nama: Muhammad Rezky Fatya
NIM : 4212201032

Nama: Muhammad Fikri
NIM : 4212201091

Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Terapan Teknik (S.Tr.T)
di
Politeknik Negeri Batam

Disusun oleh:
Muhammad Rezky Fatya (4212201032)
Muhammad Fikri (4212201091)

Tanggal Sidang: 19 Januari 2026

Disetujui oleh:

1. Daniel Sutopo Pamungkas, S.T.,M.T., Ph.D
NIK: 100006

1. Adlian Jefiza, S.Pd., M.T.
NIK: 119220

2. Diono, S.Tr. T., M.Sc
NIK: 120243

2. Budiana, S.Si., M.Si
NIK: 0001019201

Analisis Perbandingan Algoritma *Support Vector Machine* dan *Random Forest* dalam Klasifikasi Jenis Penyakit Malaria pada Citra Darah Manusia Berbasis YOLO 11

Abstrak

Malaria merupakan penyakit menular yang masih menjadi permasalahan kesehatan global, terutama di daerah endemis. Deteksi dini dan klasifikasi jenis penyakit malaria sangat penting untuk memastikan penanganan yang tepat. Namun, metode konvensional berbasis analisis mikroskopis oleh tenaga medis memiliki keterbatasan dalam hal akurasi dan efisiensi. Oleh karena itu, penelitian ini bertujuan untuk membandingkan kinerja algoritma *Support Vector Machine* (SVM) dan *Random Forest* dalam mengklasifikasikan jenis penyakit malaria berdasarkan citra darah manusia yang telah dideteksi menggunakan *YOLO 11*. Metode yang digunakan terdiri dari beberapa tahap, yaitu pengunduhan dataset berlabel, *preprocessing* data melalui augmentasi dan normalisasi citra, serta pelatihan *YOLO 11* untuk mendeteksi parasit malaria. Citra hasil deteksi kemudian diproses lebih lanjut dengan teknik *cropping* dan ekstraksi fitur menggunakan ResNet-50 sebagai model *transfer learning*. Fitur yang diekstrak digunakan sebagai input untuk model klasifikasi SVM dan *Random Forest*, yang kemudian dilatih dan dievaluasi menggunakan metrik akurasi, presisi, *Recall*, dan *F1-Score*. Penelitian ini diharapkan dapat menentukan algoritma klasifikasi yang lebih optimal dalam mendeteksi jenis malaria dengan tingkat akurasi yang lebih tinggi. Untuk meningkatkan keterjangkauan pengguna, sistem dikembangkan dalam antarmuka berbasis GUI menggunakan Pyqt5, yang memungkinkan input gambar, deteksi otomatis, serta visualisasi hasil klasifikasi dan evaluasi model. Dengan pendekatan ini, penelitian ini berkontribusi dalam menyediakan sistem pendukung diagnosis yang lebih cepat dan akurat, sehingga membantu efisiensi dalam identifikasi malaria.

Kata kunci: Malaria, YOLO 11, *Support Vector Machine*, *Random Forest*, *Transfer Learning*, *ResNet-50*, Klasifikasi Citra

Comparative Analysis of *Support Vector Machine* and *Random Forest* Algorithms in the Classification of Malaria Types Based on Human Blood Microscopy Using YOLO 11

Abstract

Malaria is an infectious disease that remains a global health problem, especially in endemic areas. Early detection and classification of malaria disease types is crucial to ensure proper treatment. However, conventional methods based on microscopic analysis by medical personnel have limitations in terms of accuracy and efficiency. Therefore, this study aims to compare the performance of Support Vector Machine (SVM) and Random Forest algorithms in classifying malaria disease types based on human blood images that have been detected using YOLO 11. The method used consists of several stages, namely downloading labeled Datasets, preprocessing data through image augmentation and normalization, and training YOLO 11 to detect malaria parasites. The detected image is then further processed with cropping techniques and feature extraction using ResNet-50 *as a transfer learning model. The extracted features are used as input for SVM and Random Forest classification models, which are then trained and evaluated using accuracy, Precision, Recall, and F1-Score metrics. This research is expected to determine a more optimal classification algorithm in detecting malaria types with a higher level of accuracy. To improve user affordability, the system was developed in a GUI-based interface using Pyqt5, which allows image input, automatic detection, as well as visualization of classification results and model evaluation. With this approach, this research contributes to providing a Faster and more accurate diagnosis support system, thus helping efficiency in malaria identification.*

Keywords: Malaria, YOLO 11, Support Vector Machine, Random Forest, Transfer Learning, ResNet-50, Image Classification

Kata Pengantar

Dengan penuh rasa syukur, penulis mengucapkan terima kasih kepada Allah SWT yang telah memberikan rahmat, hidayah, dan kekuatan-Nya, sehingga Penulis dapat menyelesaikan Tugas Akhir yang berjudul "Analisis Perbandingan Algoritma *Support Vector Machine* dan *Random Forest* dalam Klasifikasi Jenis Penyakit Malaria pada Citra Darah Manusia Berbasis YOLO 11" tepat waktu.

Penulis juga ingin menyampaikan rasa terima kasih yang mendalam kepada semua pihak yang telah memberikan dukungan, ilmu, dan waktu yang berharga selama proses penyusunan Tugas Akhir ini. Tugas Akhir ini merupakan bagian dari syarat untuk meraih gelar S.Tr.T di Program Studi Teknik Mekatronika, Jurusan Teknik Elektro, Politeknik Negeri Batam. Penulis menyadari bahwa penyelesaian Tugas Akhir ini tidak akan terwujud tanpa bimbingan dan bantuan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis mengucapkan terima kasih kepada:

1. Bapak Adlian Jefiza, S.Pd., M.T., selaku dosen pembimbing 1, atas bimbingan, arahan, dan motivasinya dalam menyelesaikan tugas akhir ini.
2. Bapak Budiana, S.Si., M.Si. selaku dosen pembimbing 2, atas ilmu dan diskusi yang sangat membantu penyusunan penelitian ini.
3. Bapak Daniel Sutopo Pamungkas, S.T., M.T., Ph.D., selaku dosen penguji 1, atas saran, masukan, dan koreksi yang membangun.
4. Bapak Diono, S.Tr.T., M.Sc., selaku dosen penguji 2, atas saran, masukan, dan koreksi yang diberikan.
5. Orang tua, keluarga, dan teman-teman atas dukungan moral, material, dan doa untuk menyelesaikan tugas akhir ini.
6. Semua pihak yang telah membantu secara langsung maupun tidak langsung dalam penyelesaian tugas akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna karena adanya keterbatasan ilmu dan pengalaman yang dimiliki. Oleh karena itu, semua kritik dan saran yang bersifat membangun akan penulis terima dengan senang hati. Penulis berharap, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang memerlukan.

Batam, 19 Januari 2026

Muhammad Rezky Fatya & Muhammad Fikri

Daftar Isi

Daftar Gambar

Daftar Tabel

Bab 1. Pendahuluan

1.1. Latar Belakang

Malaria adalah penyakit yang disebabkan oleh gigitan nyamuk *Anopheles* betina yang terinfeksi oleh parasit bersel tunggal yang dikenal sebagai *Plasmodium parasite*. Umumnya, sekitar 8-12 hari setelah terkena sengatan nyamuk ini, akan tampak gejala-gejala seperti demam tinggi, flu, sakit kepala, kelelahan, menggigil, muntah-muntah, hingga kematian [1]. Selain itu, spesies *Plasmodium* yang menginfeksi tubuh manusia mempengaruhi tingkat keparahan penyakit, dimana *P. Falciparum* dan *P. Vivax* adalah yang paling berbahaya karena dapat menyebabkan parasitemia yang tinggi dan masalah besar [2].

Penyebaran yang luas dan dampak yang signifikan dari malaria, penyakit parasit yang sebagian besar menyerang populasi yang rentan di negara-negara tropis seperti Afrika, Asia Tenggara, dan Amerika Selatan, membuatnya menjadi bahaya yang terus mengancam kesehatan masyarakat global [3]. Berdasarkan laporan pada tahun 2021, jumlah kasus malaria di indonesia terus meningkat dari tahun sebelumnya. Kasus terbesar terjadi di Provinsi Papua dengan tingkat 89%. Selain dari itu, jumlah kasus malaria di indonesia menempati posisi tertinggi kedua di Asia Tenggara [4].

Seiring dengan perkembangan teknologi medis, diagnosis konvensional yang telah dilakukan dengan pemeriksaan mikroskop cahaya, *Polymerase Chain Reaction* (PCR), dan *Rapid Diagnostic Test* (RDT), telah berkontribusi dalam upaya global untuk menekan angka malaria. Dengan metode tersebut, indikasi penyakit malaria dapat diidentifikasi secara *real-time*, sehingga memungkinkan untuk dilakukan intervensi dini pada pasien. Namun, pemeriksaan tersebut sering berujung dengan penanganan yang tidak tepat waktu dan berpotensi memperburuk kondisi klinis pasien. Pemeriksaan menggunakan mikroskop cahaya membutuhkan waktu 15 hingga 30 menit per sampel dan ketergantungan pada tenaga ahli yang terlatih sehingga rentan terhadap *Human Error.* Sementara RDT memberikan hasil yang cepat dan resiko negatif palsu, namun tidak dapat membedakan jenis malaria terutama pada kepadatan parasit rendah. Sebaliknya, PCR memiliki sensitivitas tinggi dan mampu membedakan spesies malaria tetapi membutuhkan waktu, biaya, serta fasilitas laboratorium yang lebih kompleks [5].

Malaria masih menjadi ancaman global dalam keterbatasan metode deteksi yang cepat dan akurat. Sebagai respon terhadap tantangan dalam mendeteksi dan mengendalikan penyebaran malaria, sistem deteksi dan perhitungan parasit malaria telah dikembangkan oleh peneliti sebelumnya menggunakan kombinasi *Support Vector Machine* (SVM) dan *Convolutional Neural Network* (CNN). Namun, kombinasi tersebut memiliki kesulitan dalam beradaptasi di lingkungan nyata sehingga mempengaruhi hasil secara signifikan [6]. Selain itu, penggabungan

metode *You Only Look Once* (YOLO) dengan CNN telah diterapkan untuk meningkatkan akurasi deteksi serta klasifikasi malaria, namun masih menghadapi tantangan dalam optimalisasi kinerja pada perangkat dengan spesifikasi rendah [7]. Analisis perbandingan algoritma *Machine Learning* pada malaria telah dilakukan oleh peneliti sebelumnya dengan menyoroti tantangan dalam mencapai klasifikasi dan deteksi yang akurat dan optimal. Meskipun berbagai algoritma telah diterapkan, masih menghadapi tantangan dalam mencapai akurasi klasifikasi yang tinggi karena tidak semua algoritma mampu memberikan hasil yang memuaskan. Selain itu, sebagian besar fokus penelitian hanya terbatas pada analisis algoritma tanpa mempertimbangkan aspek keparahan kasus malaria yang terinfeksi pada pasien [8].

1.2. Rumusan Masalah

1. Bagaimana proses deteksi parasit malaria pada gambar mikroskopik darah manusia dapat dilakukan menggunakan YOLO 11?
2. Bagaimana hasil deteksi YOLO 11 dapat dimanfaatkan sebagai sumber ekstraksi fitur untuk mendukung proses klasifikasi jenis penyakit malaria?
3. Bagaimana perbandingan kinerja algoritma *Support Vector Machine* (SVM) dan *Random Forest* (RF) dalam mengklasifikasikan jenis penyakit malaria (*P. Falciparum*, *P. Malariae*, *P. Ovale*, dan *P. Vivax*) berdasarkan fitur hasil deteksi YOLO 11?
4. Bagaimana hasil evaluasi kinerja algoritma SVM dan RF menggunakan metode K-Fold Cross *Validation* dalam hal akurasi, presisi, recall, dan *F1-Score*?
5. Algoritma manakah yang memberikan kinerja paling optimal dan efisien dalam mengklasifikasikan jenis penyakit malaria berdasarkan fitur hasil deteksi YOLO 11?

1.3. Tujuan

1. Menerapkan algoritma YOLO 11 untuk mendeteksi keberadaan parasit malaria pada citra mikroskopik darah manusia secara otomatis dan efisien.
2. Menghasilkan ekstraksi fitur dari hasil deteksi YOLO 11 sebagai representasi karakteristik visual yang digunakan untuk proses klasifikasi jenis penyakit malaria.
3. Menganalisis dan membandingkan kinerja algoritma *Support Vector Machine* (SVM) dan *Random Forest* (RF) dalam mengklasifikasikan jenis penyakit malaria (P. *Falciparum*, P. *Malariae*, P. *Ovale*, dan P. *Vivax*) berdasarkan fitur hasil deteksi YOLO 11.
4. Mengevaluasi performa algoritma SVM dan RF menggunakan metode *K-Fold Cross Validation* berdasarkan metrik akurasi, presisi, recall, dan F1-*Score* untuk memperoleh hasil yang terukur secara kuantitatif.

5. Menentukan algoritma yang paling optimal dan efisien dalam mengklasifikasikan jenis penyakit malaria berbasis hasil deteksi YOLO 11 sehingga dapat menjadi referensi dalam pengembangan sistem diagnosis malaria otomatis berbasis kecerdasan buatan.

1.4. Manfaat

1. Menambah wawasan dan referensi ilmiah terkait penerapan YOLO 11 dalam ekstraksi fitur citra medis untuk klasifikasi penyakit.
2. Memberikan kontribusi terhadap pengembangan sistem deteksi dan klasifikasi malaria berbasis kecerdasan buatan yang cepat dan akurat.
3. Membantu tenaga medis dalam melakukan diagnosis awal malaria secara otomatis dengan tingkat kesalahan yang lebih rendah.
4. Menjadi acuan dalam memilih algoritma *machine learning* yang paling optimal untuk klasifikasi penyakit berbasis citra mikroskopik.
5. Menjadi dasar bagi penelitian selanjutnya dalam optimalisasi model AI untuk sistem diagnosis penyakit menular lainnya.

1.5. Batasan

Adapun batasan masalah pada klasifikasi malaria ini supaya pembahasannya lebih terarah dan terkonsep adalah sebagai berikut:

1. Penelitian ini hanya membandingkan performa algoritma *Support Vector Machine* (SVM) dan *Random Forest* dalam mengklasifikasikan citra darah malaria.
2. Citra darah yang digunakan berasal dari dataset publik, yang terdiri atas:
   - Citra darah positif malaria (mengandung parasit *Plasmodium*).
   - Citra darah negatif malaria (mengandung *Trypomastigote cells* dan *Sickle-shaped red blood cells*).
3. Anotasi untuk deteksi objek diperoleh dari dataset publik dan literatur terbuka, sedangkan label klasifikasi jenis *Plasmodium* diperoleh dari metadata yang tersedia dalam dataset, tanpa validasi medis langsung oleh tenaga professional kesehatan.
4. Penelitian ini tidak melibatkan proses pengambilan sampel darah secara langsung, tidak melakukan segmentasi morfologi sel darah secara klinis, dan tidak bertujuan untuk aplikasi diagnosis medis.
5. Hasil klasifikasi hanya digunakan untuk menilai kinerja model secara komputasional, bukan untuk pengambilan keputusan medis atau penerapan klinis secara langsung.
6. Dataset yang digunakan memiliki jumlah data yang terbatas dan tidak diketahui asal geografisnya, sehingga model belum diuji terhadap variasi global parasit atau sel darah.

7. Penelitian ini tidak mengklasifikasikan stadium perkembangan parasit malaria, seperti bentuk *Ring*, *Trophozoite*, *Schizont*, atau *Gametocytes*, melainkan hanya berdasarkan jenis *Plasmodium* seperti *Falciparum*, *Malariae*, *Vivax* , dan *Ovale*.
8. Penelitian ini menggunakan 2 tahap pemrosesan, yaitu:
    - Deteksi objek menggunakan YOLO 11 untuk mengidentifikasi apakah citra mengandung sel darah malaria, *Trypomastigote cells*, atau *Sickle-shaped red blood cells*.
    - Klasifikasi jenis malaria menggunakan *Support Vector Machine* (SVM) dan *Random Forest* hanya dilakukan pada citra yang terdeteksi sebagai malaria, untuk menentukan jenis parasit malaria tersebut.

1.6. *Work Breakdown Structure* (Opsional)

Tabel 1. *Work Breakdown Structure*

No	Nama	Tugas dan Tanggung Jawab dalam Tim
1	Muhammad Rezky Fatya	Analisis Algoritma YOLO dan *Random Forest*
2	Muhammad Fikri	Analisis Algoritma SVM dan Pembuatan GUI

Bab 2. Tinjauan Pustaka

2.1. Penelitian Terkait

Pengembangan riset malaria dari tahun 2021 hingga 2025 telah menunjukan perkembangan yang sangat pesat dalam mengidentifikasi malaria. Masing-masing penelitian mempunyai metode yang berbeda. Salah satu metode yang paling banyak digunakan yaitu *Convolutional Neural Network* (CNN) sebagai metode utama dalam mengidentifikasi malaria. Fokus penelitian yang telah dikembangkan yaitu berkaitan dengan ekstraksi fitur otomatis dari citra darah.

Selain itu, terdapat penelitian dengan mengkombinasikan *Support Vector Machine* (SVM) dengan *Principal Component Analysis* (PCA), *t-distributed Stochastic Neighbor Embedding* (t-SNE) dan CNN. Tujuan pengembangan penelitian untuk meningkatkan efisiensi dalam pengolahan data melalui PCA dan t-SNE. Hasil penelitian yang dilakukan dapat digunakan untuk membantu dalam reduksi dimensi dan untuk mengoptimalkan kinerja SVM. Sementara integrasi SVM dengan CNN memungkinkan klasifikasi yang lebih akurat berdasarkan fitur yang telah diekstraksi oleh CNN

Tabel 2. Penelitian mengenai Malaria

No	Nama Penulis	Topik Penelitian	Hasil yang Diperoleh
1	Lekeufack Foulefack	Malaria Parasite Detection in Microscopic Blood Smear Images Using *Deep learning* Techniques	Penelitian ini menunjukkan kinerja unggul dalam deteksi parasit malaria menggunakan model CNN dengan akurasi 96.66%, presisi 97.09%, *F1-Score* 96.56%, dan sensitivity 96.03%, serta efisiensi lebih tinggi dibanding DenseNet121 dan VGG16 [9].
2	Shashikiran	Malaria Cell Detection using Advanced SVM Techniques	Penelitian ini menunjukkan bahwa kombinasi SVM dengan PCA, t-SNE, atau CNN mencapai akurasi 97.2% dalam deteksi malaria, jauh lebih tinggi dari *rapid antigen testing* (50%) [6].

3	Abhik Paul	Malaria Parasite Classification using Deep Convolutional Neural Network	Penelitian ini menunjukkan model CNN dengan akurasi 96% lebih efektif dalam membedakan sel darah merah terinfeksi malaria dibandingkan *autoencoder* (87,5%) [10].
4	Gunawan	Malaria Parasite Detection and Classification using CNN and YOLOv5 Architectures	Penelitian ini menunjukkan bahwa model *Convolutional Neural Network* (CNN) berhasil mendeteksi apakah citra darah terinfeksi malaria atau tidak dengan akurasi keseluruhan 96,21% pada *testing set*, dengan tingkat presisi 95,42% dan *recall* 97,05%. sementara YOLOv5 terbukti mampu mengklasifikasikan parasit berdasarkan stadiumnya, meskipun menghadapi tantangan pada citra yang buram.
5	Suraksha	Classification of Malaria cell images using *Deep learning* Approach	Penelitian ini menunjukkan bahwa model *deep learning* yang menggabungkan *Convolutional Neural Network* (CNN) dengan arsitektur VGG-19 berhasil dalam klasifikasi citra sel darah parasit dan tidak terinfeksi, mencapai akurasi 96,02%. Akurasi model ini lebih tinggi dibandingkan dengan metode yang diuji lainnya, seperti CNN standar (93,71%), SVM (94,05%), dan RFC (91,68%). Model yang diusulkan ini menggunakan VGG-19, sebuah CNN pra-terlatih dengan 19 lapisan, untuk ekstraksi fitur yang efektif, yang bertujuan untuk membangun sistem diagnosis malaria otomatis, efisien, dan presisi.

6	Aluvala	A Web-Based Approach for Malaria Parasite Detection Using *Deep learning* in Blood Smears	Penelitian ini menunjukkan model *Convolutional Neural Network* (CNN) yang menggunakan arsitektur VGG-19, mencapai akurasi 97% dalam mendeteksi dan membedakan antara sampel darah yang terinfeksi dan yang tidak terinfeksi. Proyek ini menyimpulkan bahwa penggabungan teknologi web dengan VGG-19 dapat menghasilkan sistem diagnosis malaria yang otomatis, efisien, dan presisi, yang sangat penting untuk diagnosis dini dan pengobatan yang efektif, terutama di daerah endemik malaria.
7	Nascimento	Detection of Malaria Parasites in Thick Blood Smear Images using Shallow Neural Networks and Digital Image Processing Techniques	Penelitian ini menyimpulkan bahwa model Multilayer Perceptron (*shallow neural network*), yang dikombinasikan dengan pemrosesan citra digital, adalah metode yang sangat efisien dan sederhana untuk deteksi parasit malaria pada apusan darah tebal. Model ini mencapai F1-*Score* 97,57%—hasil terbaik dibandingkan studi sebelumnya pada *Dataset* yang sama—dan hanya membutuhkan waktu 3,29 detik untuk menganalisis satu *slide*.
8	Achmad Fauzi Saksenata	Klasifikasi Citra Sel Darah Untuk Penyakit Malaria Dengan Metode CNN	Penelitian ini menyimpulkan bahwa metode Convolutional Neural Network (CNN) yang diusulkan berhasil digunakan untuk klasifikasi citra sel darah dalam mendeteksi penyakit malaria. Model tersebut mencapai kinerja yang baik dengan akurasi terbaik 96%, serta mencatatkan hasil yang sama pada metrik Presisi 96%, Recall 96%, dan F1-*Score* 96%.

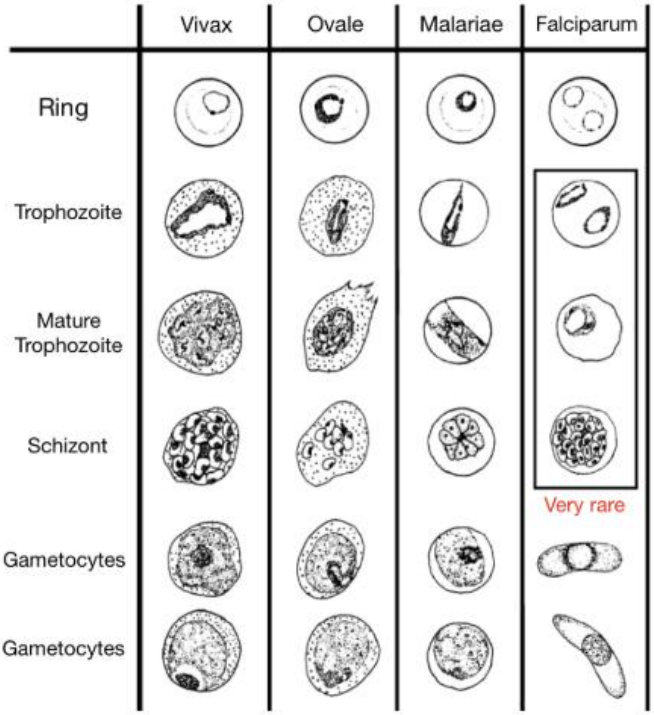
9	S Mathupriya	Malaria Disease Prediction using Faster RCNN	Penelitian ini menunjukkan Model *deep learning* Faster Region-based Convolutional Neural Network (Faster R-CNN) terbukti menjadi metode yang paling efektif dan presisi untuk memprediksi penyakit malaria pada apusan darah. Model ini mencapai kinerja luar biasa dengan akurasi 0.99, presisi 0.996, dan F1-*Score* 0.994, melampaui akurasi metode tradisional SVM (0.92). Keberhasilan ini menunjukkan potensi Faster R-CNN sebagai alat diagnosis otomatis yang cepat dan akurat.
10	Kalyan Kumar Jena	Classification of Malaria Parasitized and Uninfected Images Using Machine Learning Approach	Penelitian ini menyimpulkan bahwa metode Logistic Regression (LR) adalah yang paling efektif di antara delapan metode *Machine Learning* yang diuji (termasuk k-NN, Neural Network, dan SVM) untuk klasifikasi citra sel darah terinfeksi malaria dan tidak terinfeksi. Model LR mencapai akurasi tertinggi 0.981 (98,1%) dan secara konsisten memberikan hasil klasifikasi yang lebih baik pada nilai *cross-validation* (*folds*) yang lebih tinggi.

Jadi, meskipun penelitian ketiga unggul dalam akurasi, penelitian kedua lebih efisien dalam arsitektur CNN, sementara penelitian pertama menyoroti efektivitas CNN dibanding *Autoencoder*. Untuk implementasi di dunia nyata, perlu penelitian lebih lanjut untuk memastikan model dapat bekerja secara konsisten dalam berbagai kondisi klinis dan dataset yang lebih luas

2.1.1. Studi Epidemiologi dan Karakteristik Penyakit Malaria

Malaria merupakan penyakit infeksi yang disebabkan oleh parasit *Plasmodium* dan ditularkan melalui gigitan nyamuk *Anopheles* betina, yang hingga kini masih menjadi masalah kesehatan masyarakat di Indonesia, terutama di wilayah endemis, karena penyebarannya dipengaruhi oleh berbagai faktor seperti karakteristik epidemiologi, kondisi lingkungan, serta aspek sosial ekonomi masyarakat. Studi epidemiologi menunjukkan bahwa di beberapa wilayah, seperti

Kabupaten Purbalingga, mayoritas penderita malaria berusia 15–54 tahun dan berjenis kelamin laki-laki, dengan jenis infeksi terbanyak adalah *Plasmodium Falciparum* serta sebagian besar merupakan kasus indigenous [11], sedangkan di Papua, prevalensi malaria mencapai lebih dari 75% di beberapa area, yang disebabkan oleh aktivitas pekerjaan luar ruangan seperti pertanian dan penebangan hutan [12]. Selain itu, perubahan iklim seperti peningkatan suhu dan curah hujan dapat menyebabkan pertumbuhan populasi nyamuk *Anopheles* dan meningkatkan risiko penularan [13], apalagi bila ditambah dengan kondisi lingkungan seperti adanya genangan air serta sanitasi yang buruk [14].



Gambar 1. Representasi visual tahap intraeritrositik dari empat spesies *Plasmodium* penyebab malaria

Sumber: [15]

Secara klinis, penyakit ini ditandai dengan gejala seperti demam, menggigil, keringat berlebih, nyeri kepala, mual, dan muntah, yang muncul dalam siklus berulang sesuai dengan siklus hidup parasit dalam tubuh manusia, dimulai dari fase prepatent yang berlangsung antara 8 hingga 25 hari tergantung pada spesies

Plasmodium dan faktor individu lainnya, lalu dilanjutkan dengan fase akut yang ditandai dengan demam periodik dan perkembangan parasit dalam sel darah merah, dan jika tidak ditangani dengan baik, dapat berkembang menjadi fase komplikasi yang serius seperti anemia berat, gagal ginjal, edema paru, atau bahkan kematian. Untuk mencegah dan mengendalikan malaria, strategi yang diterapkan meliputi penggunaan kelambu berinsektisida, eliminasi tempat perindukan nyamuk, edukasi kesehatan guna meningkatkan kesadaran masyarakat, serta pengobatan yang cepat dan tepat bagi penderita [13].

2.1.2. Pengolahan Citra dalam Diagnosis Medis

Pengolahan citra medis merupakan proses penerapan teknik dan algoritma untuk menganalisis gambar medis, seperti sinar-X, CT *scan*, MRI, dan ultrasonografi, guna memperoleh informasi diagnostik yang akurat, sehingga teknologi ini menjadi komponen vital dalam praktik medis modern karena mampu membantu dokter dalam mendeteksi, mendiagnosis, dan memantau berbagai kondisi kesehatan. Teknik pengolahan citra berperan penting dalam deteksi serta klasifikasi penyakit, karena citra medis dianalisis untuk mengidentifikasi pola atau fitur tertentu yang mengindikasikan adanya kelainan, seperti pada diagnosis penyakit paru-paru menggunakan citra sinar-X, di mana algoritma pengolahan citra dapat mendeteksi area dengan tekstur atau kontras berbeda dari jaringan sehat yang mengindikasikan infeksi atau gangguan lainnya [16].

Selain itu, penerapan teknologi *Deep learning* telah meningkatkan akurasi diagnosis secara signifikan karena algoritma ini mampu mempelajari ciri-ciri penting dari citra dan memberikan prediksi yang lebih tepat, bahkan pengolahan citra juga digunakan untuk membantu perencanaan prosedur bedah atau terapi lainnya secara lebih akurat, serta mengevaluasi efektivitas pengobatan dengan memantau perubahan kondisi pasien dari waktu ke waktu. Untuk mendukung proses ini, digunakan berbagai teknologi dan alat seperti perangkat lunak MATLAB dan *OpenCV*, yang memungkinkan penerapan algoritma pengolahan citra secara efisien; khususnya kombinasi *OpenCV* dan *Python* banyak dimanfaatkan untuk membangun aplikasi medis, sedangkan teknik *machine learning* dan *Deep learning* telah banyak digunakan untuk meningkatkan akurasi dalam segmentasi maupun klasifikasi citra. Namun, dalam penerapannya, terdapat tantangan seperti kualitas citra yang bervariasi akibat perbedaan perangkat atau kondisi pengambilan gambar, yang dapat memengaruhi hasil analisis, sehingga dibutuhkan teknik prapemrosesan seperti peningkatan kontras, pengurangan *noise*, dan penajaman gambar guna memastikan kualitas citra yang optimal sebelum dilakukan analisis lebih lanjut [16].

Kompleksitas biologis, seperti variasi anatomi antar individu dan kondisi medis yang kompleks, juga menuntut penggunaan algoritma yang adaptif dan canggih agar hasilnya tetap akurat, dan di samping itu, aspek privasi serta keamanan data

menjadi pertimbangan penting karena penggunaan citra medis harus mematuhi regulasi untuk melindungi informasi pasien. Dengan terus berkembangnya teknologi, pengolahan citra medis menawarkan potensi besar dalam meningkatkan akurasi diagnosis serta efisiensi perawatan kesehatan di masa depan [16].

2.2. *Deep learning* dalam Analisis Citra Medis

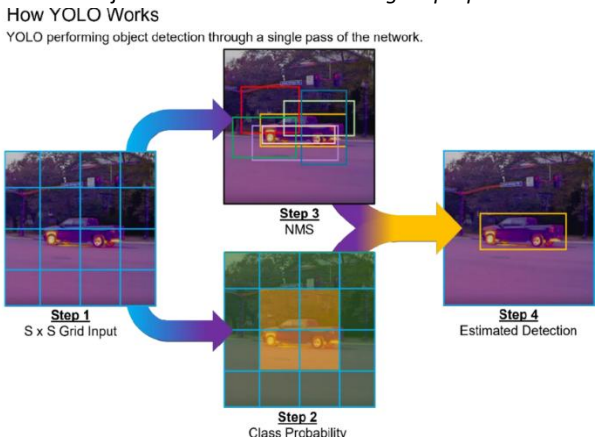
Deep learning (DL), sebagai cabang dari kecerdasan buatan, telah menunjukkan kemampuan luar biasa dalam mengidentifikasi pola dan mengolah data citra medis, sehingga penerapannya dalam analisis citra medis mencakup berbagai aspek penting seperti deteksi otomatis penyakit, segmentasi organ, dan peningkatan kualitas citra. Teknologi DL telah digunakan secara luas untuk mengidentifikasi berbagai jenis penyakit melalui pemrosesan citra medis, sebagaimana ditunjukkan oleh Khairi dan rekan-rekan yang menggunakan metode seperti *Fourier Adaptive Recognition System* (FARS) dan ResNet-50 untuk meningkatkan akurasi diagnosis penyakit di Indonesia [17]. Selain itu, DL juga dimanfaatkan dalam segmentasi organ dan jaringan, di mana Bintang dan Imaduddin mengembangkan model berbasis transfer learning untuk mendeteksi retinopati diabetik dan berhasil meningkatkan akurasi dalam segmentasi serta deteksi penyakit mata [18]. Tak hanya berperan dalam deteksi dan segmentasi, pendekatan ini juga berkontribusi terhadap peningkatan kualitas citra medis melalui pengurangan noise, peningkatan resolusi, dan koreksi artefak, yang sangat membantu dalam interpretasi klinis, sebagaimana disimpulkan oleh Gunawan dan rekan-rekan bahwa model *Convolutional Neural Networks* (CNN) memiliki keunggulan dalam akurasi dan interpretabilitas analisis citra medis [19].

Implementasi DL dalam pengolahan citra medis mencakup tahapan seperti pengumpulan data, pelabelan, pelatihan model, dan validasi, namun proses ini menghadapi tantangan berupa kebutuhan terhadap dataset besar yang teranotasi dengan baik, keterbatasan interpretabilitas model, serta kesulitan dalam integrasi ke dalam alur kerja klinis. Dalam hal ini, Anwar dan rekan menekankan pentingnya pemilihan model yang tepat sesuai jenis data dan kebutuhan klinis dengan membandingkan performa model *Long Short-Term Memory* (LSTM) dan CNN dalam mendeteksi penyakit paru-paru berdasarkan data pencitraan medis [20], sehingga strategi yang diterapkan harus disesuaikan secara kontekstual agar pemanfaatan DL dapat memberikan manfaat maksimal dalam praktik medis [20].

2.3. Deteksi Objek dengan *You Only Look Once* (YOLO 11)

Pemahaman mengenai mekanisme deteksi objek pada algoritma YOLO merupakan landasan penting dalam penelitian ini, sehingga Gambar 2 mengilustrasikan cara kerja YOLO dalam melakukan deteksi dan klasifikasi objek

secara simultan melalui pendekatan *single-shot detection* yang membedakannya dari metode deteksi objek konvensional berbasis *region proposal*.



Gambar 2. Cara kerja YOLO

Sumber: A simplified figure on how YOLO conducts object detection and... | Download Scientific Diagram

You Only Look Once (YOLO) adalah algoritma deteksi objek real-time yang bekerja dengan membagi citra menjadi grid dan melakukan prediksi *bounding box* serta kelas objek dalam satu tahap proses inferensi, menjadikannya cepat dan efisien. Secara fundamental, desain arsitektur YOLO mengintegrasikan tiga komponen utama, yakni *input layer* (lapisan masukan), *feature extraction layer* (lapisan ekstraksi fitur), dan *output layer* (lapisan keluaran). Lapisan ekstraksi fitur, yang merupakan inti dari arsitektur ini, umumnya dibangun menggunakan prinsip-prinsip *Convolutional Neural Network* (CNN). Lapisan ini terdiri dari beberapa jenis *layer* spesifik:

1. Backbone

   Bagian ini berfungsi sebagai ekstraktor fitur utama. Arsitektur *backbone* YOLO 11 menggunakan serangkaian blok CBS (terdiri dari Konvolusi, *Batch Normalization*, dan aktivasi SiLU) untuk *downsampling* citra secara progresif. Inovasi utama pada *backbone* ini:

- Blok C3k2: Menggantikan blok C2f yang digunakan pada YOLOv8. C3k2 adalah implementasi *CSP Bottleneck* kustom yang menggunakan dua konvolusi dengan *kernel* lebih kecil, yang terbukti mampu mempertahankan akurasi sekaligus meningkatkan efisiensi dan kecepatan inferensi [21].
- Modul C2PSA: Ditempatkan di akhir *backbone*, modul Cross-Stage Partial with Self-Attention (C2PSA) ini mengkombinasikan keunggulan CSP dengan mekanisme *self-attention*. Hal ini memungkinkan model untuk menangkap

informasi kontekstual secara lebih efektif, yang sangat bermanfaat untuk meningkatkan akurasi deteksi pada objek-objek kecil atau yang tumpang tindih (*occluded*) [21].

- SPPF (Spatial Pyramid Pooling Fast): Modul ini dipertahankan dari versi sebelumnya untuk menyempurnakan representasi fitur multi-skala. *Backbone* ini menghasilkan tiga *feature map* dari skala yang berbeda (dari blok C3k2(False), C3k2(True), dan C2PSA) yang kemudian disalurkan ke bagian Neck [21].

2. Neck

Bagian ini bertanggung jawab untuk fusi fitur multi-skala. Mengadopsi arsitektur Path Aggregation Network (PAN), *neck* mengambil tiga *feature map* dari *backbone* dan menggabungkannya. Seperti yang terlihat pada diagram, proses ini melibatkan serangkaian operasi Upsample (untuk memperbesar *feature map* dari *layer* yang lebih dalam) dan Concat (untuk menggabungkan *feature map* kaya semantik dengan *feature map* kaya spasial). Blok C3k2 juga digunakan di dalam *neck* untuk memproses fitur-fitur yang telah digabungkan ini sebelum meneruskannya ke Head [21].

3. Head

Ini adalah bagian prediksi akhir dari model. *Head* menerima tiga *feature map* yang telah diproses oleh *neck*, yang masing-masing bertanggung jawab untuk mendeteksi objek berukuran besar, sedang, dan kecil. Seperti yang diilustrasikan pada gambar, setiap jalur di *head* terdiri dari dua blok CBS (dengan k=3, s=1, p=1) untuk menyempurnakan fitur, yang diikuti oleh *layer* Conv2d (konvolusi 2D) akhir. *Layer* Conv2d inilah yang menghasilkan prediksi *tensor* akhir, yang berisi informasi *bounding box*, skor keyakinan (*confidence*), dan probabilitas kelas [21].

Untuk mengukur keyakinan (*confidence*) dari setiap *bounding box* yang diprediksi, YOLO mengkalkulasi *Confidence score* (C). Skor ini didefinisikan sebagai hasil perkalian antara probabilitas sel yang mengandung objek ($Pr(obj)$) dengan nilai *Intersection over Union* (IoU) antara *bounding box* terprediksi ($pred$) dan *bounding box* data latih ($truth$). Hubungan ini diformulasikan dalam Persamaan:

$$C = Pr(obj) \times IoU_{truth}^{pred} \quad (1)$$

Setelah prediksi dihasilkan, sebuah mekanisme pasca-pemrosesan bernama *Non-Maximum Suppression* (NMS) sangat krusial untuk mengeliminasi deteksi ganda pada objek yang sama. NMS bekerja dengan cara mengidentifikasi *bounding box* dengan *confidence score* tertinggi, kemudian menghapus semua *bounding box* lain yang memiliki tumpang tindih (IoU) signifikan (melebihi ambang batas tertentu) dengan *box* terpilih tersebut [22].

Selama fase pelatihan, model YOLO 11 (yang berbasis pada arsitektur YOLOv8) dioptimalkan menggunakan *loss function* komposit yang menangani tiga aspek kesalahan secara terpisah, tidak seperti pendekatan gabungan tunggal pada YOLOv1 [21]. Tiga komponen *loss function* utama ini adalah:

1. Classification Loss

Komponen ini menghitung kesalahan dalam memprediksi kelas objek yang benar (misalnya, "parasit" vs "sel darah merah"). YOLO 11 menggunakan *Binary Cross-Entropy (BCE) Loss* untuk tugas ini, yang diaplikasikan pada probabilitas kelas yang diprediksi [22]. Rumus dasarnya adalah:

$$L_{cls} = L_{BCE} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (2)$$

(dimana y adalah label *ground truth* dan $\hat{y}$ adalah prediksi probabilitas kelas).

2. Objectness Loss

Komponen ini menghitung kesalahan dalam menentukan apakah sebuah *bounding box* mengandung objek (keyakinan objek) atau hanya *background*. Ini juga menggunakan *Binary Cross-Entropy (BCE) Loss*, yang diterapkan pada skor *objectness* yang diprediksi oleh model [22]. Rumus dasarnya adalah:

$$L_{obj} = L_{BCE} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (3)$$

(dimana y adalah 1 jika ada objek, 0 jika *background*, dan y adalah prediksi skor *objectness*).

3. Localization (Regression) Loss

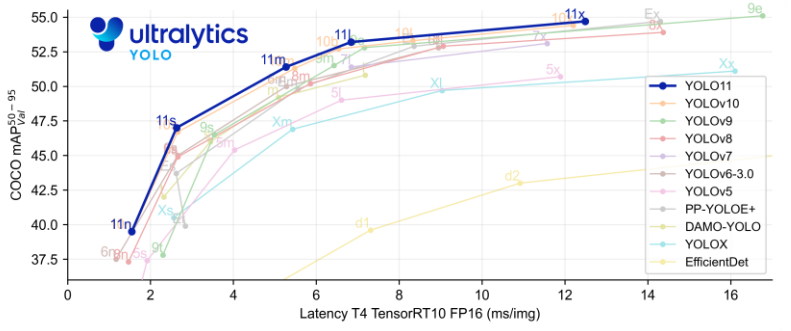
Komponen ini menghitung keakuratan posisi dan ukuran *bounding box*. Seiring perkembangan algoritma, *loss function* untuk regresi telah berevolusi. YOLO 11 mengadopsi *Complete-IoU (CIoU) Loss* (diperkenalkan sejak YOLOv4). CIoU menyempurnakan *Distance-IoU (DIoU)* dengan menambahkan parameter konsistensi rasio aspek (v). CIoU secara holistik mengevaluasi tiga faktor geometris: tumpang tindih area (IoU), jarak antar titik pusat (ρ^2), dan konsistensi rasio aspek [22]. Rumus CIoU Loss disajikan pada Persamaan:

$$L_{reg} = L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (4)$$

Dalam Persamaan (4), $\rho^2(b, b^{gt})$ merepresentasikan jarak Euklides antara titik pusat *box* prediksi (b) dan *ground truth* (b^{gt}). Nilai c adalah panjang diagonal dari kotak terkecil yang melingkupi kedua *box* tersebut, yang berfungsi sebagai normalisasi. Parameter v mengukur kesamaan rasio aspek , dan α adalah faktor bobot positif yang menyeimbangkan prioritas antara jarak dan rasio aspek [22].

Algoritma ini telah banyak diterapkan dalam berbagai penelitian kampus, seperti oleh Santiago dan rekan yang menggunakan YOLOv8 untuk mendeteksi manusia di ruang dosen dengan akurasi mencapai 93,33% dalam berbagai kondisi pencahayaan [23]. Triyanto dan rekan juga menerapkan YOLOv8 untuk mendeteksi botol plastik dan kaleng daur ulang dengan hasil mAP 99,5%, *Precision* 99,7%, dan *Recall* 99,5%, menunjukkan akurasi tinggi dalam klasifikasi objek daur ulang [24]. Sementara itu, penelitian oleh Honnainah dan Pawening memanfaatkan YOLOv5 dalam sistem pendeteksi pelanggaran pembuangan sampah secara otomatis, sebagai bentuk penerapan teknologi pada pengawasan lingkungan kampus [25]. Berbagai studi ini menunjukkan bahwa YOLO merupakan metode yang andal dan adaptif dalam berbagai konteks implementasi deteksi objek.

Dalam pengembangan sistem deteksi penyakit berbasis citra, kemampuan model untuk mengidentifikasi objek target secara presisi menjadi aspek krusial. YOLO 11 menunjukkan performa unggul dibandingkan algoritma lainnya, dengan capaian *mean Average Precision* (mAP) sebesar 96.8% dan Intersection over Union (IoU) sebesar 92.5% . Sebagai pembanding, *Faster* R-CNN hanya memperoleh mAP sebesar 87.5% dan IoU sebesar 85.23%, sedangkan SSD mencapai mAP 82.9% dan IoU 80.12% [26]. Capaian tersebut mencerminkan keandalan YOLO 11 dalam mengenali objek secara akurat, bahkan pada citra dengan kontras rendah atau kompleksitas visual tinggi. Oleh karena itu, pemilihan YOLO 11 dalam penelitian ini didasarkan pada superioritas performanya dalam mendeteksi parasit malaria secara efektif [26].



Gambar 3. Perbandingan Performa setiap versi pada YOLO

Sumber: Ultralytics YOLO11 - Ultralytics YOLO Docs

Pemilihan model YOLO 11 didasarkan pada berbagai pertimbangan teknis yang matang dan terukur, yang tidak hanya mengacu pada reputasi performa dari literatur sebelumnya, namun juga diperkuat oleh hasil pengujian langsung dalam konteks penelitian ini. YOLO 11 telah menunjukkan kinerja yang sangat kompetitif bahkan ketika dibandingkan dengan versi terbarunya, YOLO 12. Kinerja tersebut mencakup berbagai aspek kritis dalam sistem deteksi objek, seperti akurasi prediksi, kecepatan inferensi, efisiensi penggunaan sumber daya komputasi, serta kestabilan performa dalam kondisi operasional yang bervariasi dan menantang. Model ini dinilai sangat cocok untuk diadopsi dalam skenario aplikasi nyata yang menuntut kombinasi antara ketepatan deteksi, efisiensi proses, dan ketahanan terhadap fluktuasi lingkungan, terutama dalam sistem yang bergantung pada kemampuan real-time dan memiliki keterbatasan dalam hal daya komputasi atau konsumsi energi [21].

Keunggulan model YOLO 11 juga tercermin dalam hasil pengujian terhadap sejumlah metrik evaluasi utama yang digunakan dalam penilaian kinerja model deteksi objek, seperti nilai *Precision*, *Recall*, dan *mean Average Precision* (mAP).

Meskipun dalam beberapa pengujian, YOLO 12 sedikit lebih unggul dari segi akurasi secara numerik, namun keunggulan YOLO 11 dari sisi kecepatan inferensi dan efisiensi sumber daya jauh lebih signifikan. Hal ini menjadikannya sebagai solusi yang lebih stabil dan praktis, terutama dalam sistem yang dirancang untuk *deployment* jangka panjang dengan minimnya intervensi teknis secara berkala. Performa YOLO 11 yang relatif konsisten di berbagai lingkungan pengujian juga memperkuat persepsi bahwa model ini telah mencapai tingkat kematangan yang tinggi dari segi stabilitas operasional dan kesiapan integrasi industri, yang pada gilirannya meningkatkan kepercayaan baik dari kalangan peneliti maupun pelaku industri terhadap kapabilitasnya [21].

Sebaliknya, meskipun YOLO 12 merupakan versi paling mutakhir dalam keluarga algoritma YOLO, dengan menghadirkan sejumlah pembaruan arsitektural seperti *Area Attention Module* (A2) dan *Residual Efficient Layer Aggregation Networks* (R-ELAN) yang dirancang untuk meningkatkan akurasi dan efisiensi, model ini masih berada pada tahap penyempurnaan. Kompleksitas tambahan yang dibawa oleh pembaruan tersebut belum sepenuhnya memberikan dampak positif secara menyeluruh, terutama dalam hal kecepatan dan efisiensi inferensi. Evaluasi awal justru menunjukkan bahwa adanya peningkatan kompleksitas jaringan mengakibatkan beban komputasi yang lebih tinggi, serta waktu respon yang lebih lambat dibandingkan generasi sebelumnya. Dengan kata lain, meskipun secara konseptual YOLO 12 menjanjikan peningkatan performa, pada praktiknya model ini belum mampu mengungguli YOLO 11 secara menyeluruh dalam konteks aplikasi yang memerlukan kecepatan tinggi dan penggunaan sumber daya yang efisien [21].

Berdasarkan hasil pengukuran, diketahui bahwa YOLO 12 memiliki latency inferensi yang mencapai 10.9 milidetik, yang berarti membutuhkan waktu lebih lama dalam memproses data dibandingkan YOLO 11, yang rata-rata menunjukkan waktu inferensi yang lebih singkat dan stabil. Selain itu, jumlah GFLOPs yang diperlukan oleh YOLO 12 juga jauh lebih besar, sehingga membutuhkan perangkat keras dengan spesifikasi lebih tinggi untuk dapat dijalankan secara optimal. Dalam situasi tertentu, kebutuhan ini menjadi tantangan serius karena tidak semua perangkat target memiliki kapabilitas komputasi yang mumpuni. Akibatnya, penerapan YOLO 12 berpotensi menjadi tidak efisien, terutama untuk tugas-tugas deteksi objek berskala kecil atau aplikasi yang dijalankan pada perangkat edge, embedded systems, atau sistem dengan kebutuhan konsumsi daya rendah. Kompleksitas arsitektural ini juga dapat menimbulkan penurunan kestabilan dalam konteks pengoperasian jangka panjang, yang secara langsung bertentangan dengan kebutuhan banyak sistem industri saat ini yang mengutamakan reliabilitas dan efisiensi [21].

Version	Acc.	Speed	GFLOPs	Size
YOLOv5un	**6**	2	3	5
YOLOv8n	4	2	5	**6**
YOLOv9t	**1**	**6**	4	**1**
YOLOv10n	5	**1**	**6**	4
YOLO11n	3	4	**1**	3
YOLOv12n	2	5	**1**	2
YOLOv3u tiny	**7**	**1**	**1**	**7**
YOLOv5us	5	3	4	5
YOLOv8s	5	4	**7**	6
YOLOv9s	2	**7**	6	**1**
YOLOv10s	3	2	5	2
YOLO11s	3	4	2	4
YOLOv12s	**1**	6	2	3
YOLOv5um	**6**	2	2	6
YOLOv8m	5	4	**6**	**7**
YOLOv9m	2	5	5	5
YOLOv10m	4	**1**	**1**	**1**
YOLO11m	**1**	3	4	2
YOLOv12m	2	**6**	3	3
YOLOv5ul	4	3	6	**7**
YOLOv8l	**6**	6	**7**	6
YOLOv9c	3	5	4	5
YOLOv10b	**6**	**1**	3	**1**
YOLOv10l	4	4	5	3
YOLO11l	**1**	2	**1**	2
YOLOv12l	2	**7**	2	4
YOLOv3u	**5**	2	**7**	**7**
YOLOv5ux	**5**	4	5	6
YOLOv8x	4	4	6	5
YOLOv9e	**5**	**6**	2	4
YOLOv10x	**1**	3	**1**	**1**
YOLO11x	2	**1**	3	2
YOLOv12x	2	**6**	4	3

Gambar 4. Ranking versi YOLO berdasarkan akurasi, kecepatan, dan ukuran model.

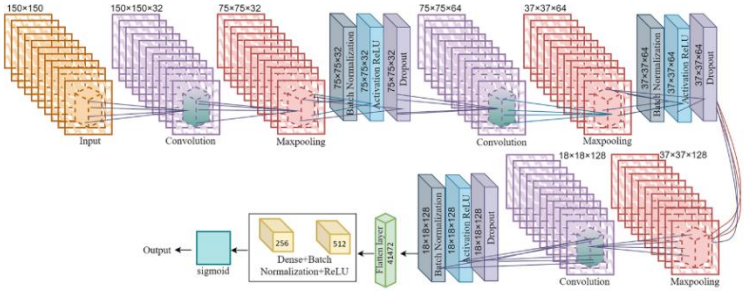
Sumber: [21]

Selain akurat, YOLO 11 juga efisien secara komputasi. Waktu rata-rata inferensi per citra hanya sebesar 13.5 milidetik, lebih cepat dibandingkan YOLOv10 (19.3 ms) maupun YOLOv8 (23 ms) [27]. Efisiensi ini menjadikannya sangat cocok untuk aplikasi real-time, seperti pemantauan medis langsung atau diagnosis otomatis di laboratorium. Kecepatan tersebut didukung oleh optimasi arsitektur dan teknik kompresi yang memungkinkan pemrosesan cepat tanpa mengorbankan akurasi. Dalam konteks aplikasi malaria, hal ini memungkinkan sistem memberikan hasil diagnosis dalam waktu singkat, sehingga mendukung pengambilan keputusan medis secara tepat waktu dan meningkatkan keselamatan pasien [27].

2.4. *Convolutional Neural Network* (CNN) dalam Ekstraksi Fitur Citra

Arsitektur *Convolutional Neural Network* (CNN) memiliki peran fundamental dalam ekstraksi fitur citra secara hierarkis, oleh karena itu Gambar 5

mendeskripsikan proses CNN secara komprehensif yang meliputi tahapan konvolusi, aktivasi non-*Linear*, dan pooling untuk menghasilkan representasi fitur yang diskriminatif dari citra input.



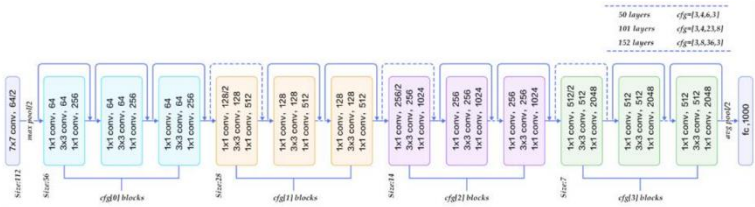
Gambar 5. Proses CNN

Sumber: [28]

Convolutional Neural Network (CNN) adalah arsitektur *deep learning* yang dirancang untuk mengolah data citra dengan cara mengekstraksi fitur melalui lapisan konvolusi dan pooling secara bertahap, sehingga dapat mengenali pola visual seperti tepi, warna, dan tekstur. Proses ini dilakukan dengan menggeser filter (kernel) pada citra untuk menghasilkan *feature* map, kemudian diikuti dengan pooling untuk mereduksi dimensi dan mempertahankan informasi penting. CNN telah digunakan secara luas dalam berbagai penelitian, seperti oleh Adenia dan rekan dari Universitas Muhammadiyah Malang yang memanfaatkan CNN untuk mendeteksi penyakit pada daun mangga, dengan akurasi tinggi saat dikombinasikan dengan *Random Forest* [29]. Permana dan rekan dari Universitas Udayana juga menggunakan CNN untuk ekstraksi fitur pada sistem rekomendasi fashion berbasis citra [30], sedangkan Peryanto dan rekan dari Universitas Ahmad Dahlan menunjukkan efektivitas CNN dalam klasifikasi citra menggunakan validasi silang K-Fold [31]. Dalam penelitian ini, proses ekstraksi fitur citra dilakukan secara otomatis menggunakan arsitektur Convolutional Neural Network (CNN). CNN memiliki kemampuan untuk mengekstraksi berbagai jenis fitur penting dari citra darah manusia, seperti bentuk sel darah, keberadaan parasit malaria, tekstur, dan distribusi warna. Output dari proses ekstraksi fitur berupa vektor berdimensi tinggi yang berisi angka-angka representasi karakteristik dari citra tersebut. Angka-angka ini berasal dari proses konvolusi, aktivasi non-*Linear* (ReLU), dan pooling yang dilakukan pada tiap lapisan CNN. Fitur-fitur yang diekstrak meliputi tepi objek, pola bentuk, dan tekstur sel, yang kemudian digunakan sebagai input bagi algoritma klasifikasi seperti *Support Vector Machine* (SVM) dan *Random Forest* (RF) dalam menentukan jenis penyakit malaria [32].

2.5. Ekstraksi Fitur Berbasis Arsitektur ResNet-50

Untuk memahami struktur jaringan yang digunakan dalam proses ekstraksi fitur pada penelitian ini, Gambar 6 menyajikan arsitektur ResNet-50 secara lengkap yang menampilkan 50 lapisan konvolusi beserta mekanisme *residual learning* melalui *skip connections* yang menjadi keunggulan utama arsitektur ini dalam mengatasi permasalahan *vanishing gradient* pada jaringan yang sangat dalam.



Gambar 6. Arsitektur ResNet-50

Sumber: [33]

Ekstraksi fitur merupakan tahapan fundamental dalam alur kerja klasifikasi citra, berfungsi untuk mentransformasi data piksel mentah menjadi representasi numerik yang informatif, ringkas, dan diskriminatif. Dalam konteks medis seperti identifikasi spesies Plasmodium penyebab malaria, proses ini bertujuan mengekstraksi karakteristik morfologis sel darah yang relevan untuk membedakan antara *P. Falciparum*, *P. Vivax* , *P. Malariae*, dan *P. Ovale*. Pada penelitian ini, arsitektur ResNet-50 digunakan sebagai pengekstrak fitur canggih. Masukan untuk ResNet-50 adalah citra sel darah individual yang telah dilokalisasi dan diisolasi oleh model deteksi objek YOLO 11, sehingga memastikan bahwa fitur yang diekstraksi benar-benar berasal dari area relevan (*Region of Interest*) [34].

ResNet-50 adalah sebuah *Convolutional Neural Network* (CNN) dengan kedalaman 50 lapisan yang diperkenalkan oleh He dkk. pada tahun 2015. Arsitektur ini dirancang secara spesifik untuk mengatasi masalah vanishing gradient, yaitu pelemahan sinyal gradien pada jaringan yang sangat dalam. Berbeda dengan arsitektur sekuensial konvensional seperti *VGGNet*, ResNet-50 memperkenalkan konsep revolusioner pembelajaran residual (*residual learning*) melalui implementasi blok residual dengan koneksi pintas (*skip connection*). Mekanisme ini memungkinkan keluaran dari lapisan sebelumnya ditambahkan secara langsung ke lapisan yang lebih dalam, menciptakan "jalan tol" bagi gradien selama proses *backpropagation*. Secara matematis, output blok dapat direpresentasikan sebagai:

$$y = F(x, \{W_i\}) + x \tag{5}$$

Struktur jaringan ini secara spesifik terdiri atas lapisan-lapisan konvolusi, *batch normalization* yang menstabilkan distribusi aktivasi, dan *Rectified Linear Unit* (ReLU) sebagai fungsi aktivasi *non-Linear*. Pendekatan ini memungkinkan ResNet-50 mencapai kedalaman jaringan yang signifikan tanpa mengalami penurunan kinerja, sehingga lebih stabil dalam pelatihan dibandingkan arsitektur yang lebih dangkal [35].

Dalam penelitian ini, ResNet-50 tidak dilatih dari awal, melainkan memanfaatkan pendekatan *transfer learning*. Konsep *transfer learning* memungkinkan sebuah model yang telah dilatih pada dataset besar dan umum untuk mentransfer pengetahuannya ke tugas baru, sehingga mempercepat waktu pelatihan dan meningkatkan performa [33]. Model ResNet-50 yang telah dilatih pada dataset raksasa *ImageNet* digunakan kembali. Bobot (*weights*) yang sudah matang dari pelatihan ini mengandung pengetahuan fitur visual dari tingkat rendah (seperti deteksi tepi dan tekstur) hingga tingkat tinggi yang dapat diadaptasi untuk domain citra mikroskopik darah [36]. Sesuai dengan metodologi ekstraksi fitur, yakni sebagai salah satu pendekatan dalam *transfer learning*, di mana lapisan klasifikasi akhir dari ResNet-50 tidak digunakan [33]. Sebaliknya, output diambil dari lapisan sebelumnya, yaitu *global average pooling layer* [37]. Lapisan ini menghasilkan vektor fitur berdimensi 2048 untuk setiap citra sel darah yang masuk. Vektor fitur inilah yang menjadi representasi numerik kaya informasi yang kemudian akan menjadi masukan bagi algoritma klasifikasi *Support Vector Machine* (SVM) dan *Random Forest*. Proses ini dapat dirumuskan sebagai:

$$f = \phi_{\text{ResNet-50}}(I) \tag{6}$$

Di mana I adalah citra sel darah hasil deteksi YOLO 11 dan $\phi_{\text{ResNet-50}}$ adalah fungsi pemetaan non-*Linear* yang diimplementasikan oleh arsitektur ResNet-50

Integrasi ResNet-50 dalam *pipeline* penelitian ini menciptakan alur kerja yang efisien dan sinergis. Model YOLO 11 berperan sebagai pendeteksi cerdas yang secara presisi mengidentifikasi lokasi sel darah, sementara ResNet-50 berfungsi sebagai analis mendalam yang mengekstraksi esensi visual dari sel darah tersebut. Penggunaan ResNet-50 sebagai *feature extractor* dalam konteks ini menawarkan beberapa keunggulan utama. Kekuatan diskriminatifnya yang tinggi, berkat kedalaman 50 lapisan, memungkinkan ekstraksi fitur hierarkis yang sangat kompleks dan abstrak, sehingga mampu menangkap nuansa morfologi halus antar spesies *Plasmodium*. Selain itu, mekanisme *residual learning* yang menjadi inti arsitekturnya memastikan stabilitas pelatihan yang andal selama proses ekstraksi fitur. Pemanfaatan model *pre-trained* juga memberikan efisiensi *transfer learning* yang signifikan dengan mengurangi kebutuhan akan data latih dalam jumlah besar serta waktu komputasi. Terakhir, fitur yang dihasilkan menunjukkan ketahanan (*robustness*) yang tinggi, terbukti tangguh terhadap variasi seperti pencahayaan, pewarnaan, dan artefak minor pada citra mikroskopik, bahkan pada citra berkualitas rendah sekalipun [34].

Berbagai studi komparatif telah menunjukkan superioritas arsitektur berbasis ResNet-50 dibandingkan arsitektur lain seperti VGG16 dalam tugas klasifikasi citra medis. Penelitian oleh Rajaraman dkk. (2018) menunjukkan bahwa saat digunakan sebagai *feature extractor* untuk klasifikasi malaria, ResNet-50 secara konsisten lebih unggul dalam hal akurasi (95,9%), sensitivitas, dan F1-*Score* dibandingkan VGG-16 dan AlexNet [38]. Studi lain oleh Kohsasih dkk. (2022) juga menemukan bahwa meskipun waktu pelatihannya sedikit lebih lama, peningkatan akurasi dan stabilitas ResNet50 menjadikannya pilihan optimal, dengan akurasi mencapai 99% dibandingkan VGG16 (95%) dan VGG19 (94%) [35]. Serupa dengan itu, Muhandisin & Azhar (2022) mengonfirmasi bahwa model *fine-tuned* ResNet50 mampu mencapai akurasi hingga 98%, melampaui VGG16 yang hanya mencapai 96% [36]. Dengan demikian, pemilihan ResNet-50 sebagai *feature extractor* menyediakan landasan fitur yang kuat dan optimal untuk dianalisis lebih lanjut oleh algoritma SVM dan *Random Forest*.

2.6. Penerapan *Machine Learning* dalam Klasifikasi Citra Darah

Machine Learning merupakan teknologi kecerdasan buatan yang digunakan untuk mengolah suatu data dengan lebih baik dan efisien sesuai dengan jumlah dataset yang tersedia dengan menerapkan mesin untuk belajar secara mandiri [12]. Penerapan teknik *Machine Learning* memungkinkan identifikasi dan klasifikasi dalam diagnosis citra medis, termasuk klasifikasi citra darah. Dalam proses klasifikasi citra darah, tahap awal melibatkan pengolahan citra untuk meningkatkan kualitas gambar, seperti normalisasi warna, segmentasi, dan reduksi noise. Setelah itu fitur utama di ekstraksi menggunakan teknik seperti histogram warna, transformasi gelombang pendek, atau menggunakan metode berbasis *Deep learning* seperti *Convolutional Neural Networks* (CNN).

2.7. Model *Support Vector Machine* (SVM) untuk Klasifikasi Jenis Malaria

Support Vector Machine (SVM) merupakan algoritma teratur *Machine Learning* yang menawarkan pendekatan efektif dalam klasifikasi suatu data, yaitu dengan cara mencari *hyperplane* terbaik yang dapat memsisahkan data. *Hyperplane* ini dibangun berdasarkan vektor-vektor bobot dan bias, yang ditentukan melalui proses optimasi [39]. Secara matematis, fungsi Keputusan pada SVM untuk klasifikasi *Linear* didefinisikan sebagai:

$$f(x) = w^T x + b \tag{7}$$

Dimana:

- w^T merupakan transpose dari bobot w
- x adalah fitur input

- b adalah bias yang menggeser posisi *hyperplane*.

Dalam kasus klasifikasi *Linear*, SVM memaksimalkan margin pemisah antar kelas dengan mensyaratkan bahwa semua titik data memenuhi ketentuan:

$$[(w^T x_i) + b] \geq 1 \quad \text{jika } y_i = 1 \qquad (8)$$
$$[(w^T x_i) + b] \leq -1 \quad \text{jika } y_i = -1$$

Pada persamaan (8), setiap komponen dapat dijelaskan sebagai berikut:

- x_i merepresentasikan vektor fitur untuk data ke-i dari dataset training.
- $w^T x_i$ menunjukkan hasil perkalian dot product antara transpose vektor bobot dengan vektor fitur.
- y_i merupakan label kelas dari data tersebut, dengan dua kemungkinan nilai:
- $y_i = 1$ menunjukkan data termasuk kelas positif, di mana data harus memenuhi syarat.
- $y_i = -1$ menunjukkan data termasuk kelas negatif, di mana data harus memenuhi syarat.

Nilai ambang batas 1 dan -1 menentukan jarak minimal yang harus dipenuhi oleh setiap titik data dari *hyperplane*. Ketentuan ini memastikan bahwa setiap titik data tidak hanya berada pada sisi yang benar dari hyperplane, tetapi juga memiliki jarak minimal tertentu dari hyperplane tersebut. Dengan demikian, tercipta zona pemisah yang jelas dan lebar antara kedua kelas yang disebut sebagai margin.

Berdasarkan ketentuan klasifikasi tersebut, tujuan utama SVM adalah memaksimalkan margin antar kelas, yang dihitung sebagai:

$$\text{Margin} = \frac{|w|}{2} \qquad (9)$$

Untuk memahami persamaan (9) secara lebih rinci, berikut adalah penjelasan dari setiap komponennya:

- Margin merupakan lebar zona pemisah antara kedua kelas yang diukur dari hyperplane ke support vector terdekat
- $|w|$ merupakan norma Euclidean dari vektor bobot w.
- Angka 2 pada penyebut menunjukkan bahwa margin diukur dari hyperplane ke salah satu sisi (setengah dari total lebar pemisah antar kelas)

Margin ini merepresentasikan jarak antara *hyperplane* pemisah dengan titik data terdekat dari masing-masing kelas, yang dikenal sebagai *support vectors*. Semakin besar nilai margin, semakin *robust* model dalam melakukan klasifikasi terhadap data baru.

Optimasi dilakukan menggunakan metode *Quadratic Programming*, yang meminimalkan $\frac{1}{2}|w|^2$ dengan syarat bahwa semua data diklasifikasikan dengan benar. Untuk data yang tidak dapat dipisahkan secara *Linear*, SVM menggunakan teknik *kernel trick*, yang memetakan data ke ruang fitur berdimensi lebih tinggi agar pemisahan *Linear* menjadi mungkin. Fungsi keputusan dengan kernel ditulis sebagai:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \quad (10)$$

Untuk memahami persamaan (10) secara lebih rinci, berikut adalah penjelasan dari setiap komponennya:

- $f(x)$ merupakan fungsi keputusan yang menghasilkan nilai prediksi untuk data input x.
- Σ (sigma) menunjukkan operasi penjumlahan dari semua komponen.
- i adalah indeks yang berjalan dari 1 hingga n, menunjukkan urutan data training.
- n merupakan jumlah total data training yang digunakan dalam model.
- α_i adalah Lagrange multipliers yang menentukan seberapa besar kontribusi setiap data training terhadap keputusan klasifikasi.
- y_i adalah label kelas untuk data training ke-i, dengan nilai {-1, 1}.
- $K(x_i, x)$ adalah fungsi kernel yang mengukur tingkat kemiripan antara data training x_i dan data input x.
- x_i merepresentasikan vektor fitur dari data training ke-i.
- x adalah vektor fitur dari data input yang akan diklasifikasi.
- b tetap berfungsi sebagai bias yang menggeser posisi hyperplane.

Bentuk fungsi $K(x_i, x)$ bergantung pada jenis kernel yang digunakan, seperti Linear, Polynomial, Radial Basis Function (RBF), atau Sigmoid. Pemilihan kernel yang tepat memungkinkan SVM untuk menangani data yang memiliki pola non-Linear dengan cara mentransformasikan data ke ruang dimensi yang lebih tinggi tanpa perlu menghitung transformasi secara eksplisit [39].

Dalam kasus deteksi malaria, citra hapusan darah dapat mengandung banyak fitur, sehingga SVM dapat secara efektif menangani kompleksitas dan variabilitas data [40]. Dalam salah satu studi, SVM menunjukkan akurasi sebesar 96,3% dalam klasifikasi terhadap vaksinasi *COVID-19* di media, mengungguli algoritma lain seperti *Naïve Bayes* dan *K-Nearest Neighbor* (KNN) yang masing-masing mencatatkan akurasi sebesar 94% dan 91% [41]. Studi lain yang membandingkan SVM dan KNN dalam klasifikasi data medis, khususnya penyakit diabetes, juga menunjukkan bahwa SVM mampu memberikan hasil akurasi yang kompetitif sebesar 81,67%, hanya sedikit di bawah KNN yang mencatatkan 83,33% [39]. Hasil ini menunjukkan bahwa SVM tetap menjadi algoritma yang layak dipertimbangkan, terutama efisiensi komputasi dan generalisasi terhadap data baru menjadi prioritas utama dalam klasifikasi.

2.8. Pendekatan *Random Forest* untuk Klasifikasi jenis Malaria

Salah satu pendekatan yang banyak digunakan dalam klasifikasi citra medis, khususnya untuk identifikasi jenis malaria, adalah algoritma *Random Forest*.

Metode ini merupakan teknik pembelajaran ansambel yang menggabungkan sejumlah pohon indicator (*decision tree*) untuk meningkatkan akurasi klasifikasi dan mengurangi risiko overfitting. Proses pembentukan pohon indicator dalam RF pada dasarnya mengikuti prosedur yang sama seperti pada *Classification and Regression Tree* (CART), namun perbedaannya terletak pada tidak dilakukannya proses *pruning* (pemangkasan) pada pohon-pohon yang dihasilkan [42].

Pada setiap simpul internal dari pohon indicator, pemilihan fitur dilakukan berdasarkan perhitungan *Gini Index*, yang berfungsi sebagai indicator ketidakmurnian suatu node. Semakin rendah nilai indeks Gini, maka semakin tinggi kemurnian data dalam node tersebut. Nilai Gini untuk suatu subset data S_i dapat dihitung menggunakan rumus sebagai berikut:

$$Gini(S_i) = 1 - \sum_{i=0}^{c-1} p_i^2 \tag{11}$$

Untuk memahami persamaan (11) secara lebih rinci, berikut adalah penjelasan dari setiap komponennya:

- $Gini(S_i)$: Nilai Gini Index untuk subset data SiS_{i} Si, mengukur tingkat ketidakmurnian (impurity) node. Nilai berkisar 0-0.5 untuk klasifikasi biner (0 = murni sempurna, 0.5 = paling tidak murni).
- S_i : Subset data ke-i yang berada pada node tertentu dalam decision tree. Subset ini berisi sampel-sampel yang telah dikelompokkan berdasarkan kriteria split sebelumnya.
- $\sum_{i=0}^{c-1} p_i^2$: Simbol sigma (Σ) menunjukkan penjumlahan (summation) dari indeks i=0 hingga c-1, artinya menjumlahkan untuk semua kelas yang ada.
- c : Jumlah total kelas dalam dataset (class count). Untuk klasifikasi malaria biner: c=2 (Parasitized dan Uninfected). Untuk multiclass bisa lebih dari 2.
- p_i : Proporsi atau frekuensi relatif sampel dari kelas ke-i dalam subset SiS_{i} Si. Dihitung sebagai: $p_i = \frac{n_{kelas\_i}}{n_{total\_dalam\_S_i}}$ Nilai berkisar 0-1.
- p_i^2: Kuadrat dari proporsi kelas ke-i. Pengkuadratan ini memberikan bobot lebih tinggi pada kelas dominan dan digunakan dalam perhitungan probabilitas dua sampel berbeda kelas.
- i : Indeks iterator untuk kelas, dimulai dari 0 hingga c-1 (mengikuti konvensi 0-based indexing).

Dalam rumus tersebut, $p_i$merupakan frekuensi relatif dari kelas $C_i$dalam subset S_i, dan *c*adalah jumlah total kelas. Nilai Gini ini mencerminkan probabilitas bahwa dua elemen yang dipilih secara acak dari subset tersebut masuk ke dalam kelas yang berbeda. Jika seluruh elemen dalam node termasuk ke dalam satu kelas yang sama, maka nilai Gini menjadi nol, yang menunjukkan tingkat kemurnian tertinggi.

Setelah itu, kualitas *split* (pemisahan data) untuk fitur tertentu diukur berdasarkan nilai rata-rata tertimbang dari indeks Gini setiap subset hasil *split*. Rumus untuk menghitung *Gini Split* adalah sebagai berikut:

$$Gini_{split} = \sum_{i=0}^{k-1} \left(\frac{n_i}{n}\right) Gini(S_i) \quad (12)$$

Pada persamaan (12), setiap komponen dapat dijelaskan sebagai berikut:

- $Gini_{split}$: Nilai Gini setelah melakukan split (pemisahan) data berdasarkan fitur tertentu. Ini adalah rata-rata tertimbang (weighted average) dari Gini Index semua subset hasil split.
- $\sum_{i=0}^{k-1}$: Penjumlahan untuk semua subset yang dihasilkan dari split, dari indeks 0 hingga k-1.
- k: Jumlah subset yang dihasilkan setelah split. Untuk split biner: k=2 (left child dan right child). Untuk split multiway bisa lebih dari 2.
- n_i: Jumlah sampel dalam subset S_i setelah split. Ini adalah ukuran (size) dari child node ke-i.
- n: Jumlah total sampel pada node parent (sebelum split). Nilai ini tetap konstan untuk semua subset hasil split dari parent yang sama.
- $\frac{n_i}{n}$: Proporsi atau bobot (weight) dari subset ke-i terhadap total sampel parent node. Nilai ini berkisar 0-1 dan semua proporsi berjumlah 1: $\sum_{i=0}^{k-1} \frac{n_i}{n} = 1$
- $Gini(S_i)$: Nilai Gini Index untuk subset ke-i (dihitung menggunakan rumus 11).

Dengan menggunakan pendekatan ini, algoritma RF memilih fitur terbaik untuk melakukan pembelahan data dengan mempertimbangkan nilai *Gini Split* terendah, sehingga menghasilkan pemisahan yang paling optimal.

Secara umum, RF bekerja dengan membangun banyak pohon keputusan $h(X, \Theta_k)$, di mana $\Theta_k$adalah vektor parameter acak yang bersifat independent and identically distributed (i.i.d). Setiap pohon memberikan hasil klasifikasi, dan hasil akhir diperoleh melalui mekanisme majority voting, yaitu kelas yang paling banyak dipilih oleh semua pohon.

Evaluasi performa teoritis dari metode Random Forest dilakukan melalui pendekatan fungsi margin (*margin function*) dan kesalahan generalisasi (*generalization error*), yang dalam beberapa studi terdahulu digunakan sebagai kerangka analisis matematis untuk mengukur kekuatan klasifikasi dan tingkat ketergantungan antar pohon keputusan dalam model. Fungsi margin untuk Random Forest didefinisikan sebagai:

$$mr(X, Y) = P_{\Theta}(h(X, \Theta) = Y) - \max_{j \neq Y} P_{\Theta}\left(h(X, \Theta) = j\right) \quad (13)$$

Untuk memahami persamaan (13) secara lebih rinci, berikut adalah penjelasan dari setiap komponennya:

- $mr(X,Y)$: Margin function atau fungsi margin untuk sampel dengan fitur X dan label benar Y. Mengukur confidence atau kepercayaan prediksi model.
- X: Vektor fitur input untuk satu sampel. Dalam konteks malaria dengan ResNet50: X adalah vektor 2048 dimensi.
- Y: Label kelas yang benar (true class) untuk sampel X. Untuk malaria: Y ∈ {0, 1} atau {Parasitized, Uninfected}.
- Θ: Vektor parameter acak yang mendefinisikan satu decision tree dalam Random Forest. Parameter ini mencakup fitur yang dipilih untuk split, threshold values, dan struktur tree. Setiap tree memiliki Θ yang berbeda.
- P_Θ: Probabilitas atau proporsi trees dalam Random Forest yang memenuhi kondisi dalam tanda kurung. Dihitung sebagai: $P_\Theta(\dots) = \frac{\text{jumlah trees yang memenuhi kondisi}}{\text{total trees}}$
- $h(X,\Theta)$: Fungsi hipotesis atau classifier function untuk satu tree dengan parameter Θ, yang memetakan input X ke prediksi kelas.
- $h(X,\Theta) = Y$: Kondisi bahwa prediksi tree sama dengan label benar. Bernilai TRUE atau FALSE (1 atau 0).
- $P_\Theta(h(X,\Theta) = Y)$: Probabilitas bahwa tree memprediksi kelas yang benar Y. Ini adalah accuracy rate atau voting proportion untuk kelas benar.
- j: Indeks kelas alternatif (bukan kelas benar Y). Untuk semua kelas kecuali Y.
- $j \neq Y$: Constraint yang menyatakan j adalah kelas selain kelas benar. Untuk klasifikasi biner dengan Y=0, maka j=1, dan sebaliknya.
- $\max_{j \neq Y}$: Operator maximum yang mencari nilai maksimum dari ekspresi untuk semua j yang bukan Y. Ini adalah probabilitas tertinggi untuk kelas yang salah.
- $P_\Theta(h(X,\Theta) = j)$**)**: Probabilitas bahwa tree memprediksi kelas j yang salah. Untuk klasifikasi biner, ini adalah voting proportion untuk kelas alternatif

Fungsi ini mengukur seberapa jauh prediksi mayoritas mendukung label yang benar Y dibandingkan dengan label lain. Nilai margin positif menunjukkan bahwa klasifikasi mayoritas benar, sedangkan nilai negative menunjukkan kemungkinan kesalahan klasifikasi. Dari fungsi margin ini, kekuatan (*strength*) dari himpunan pengklasifikasi dapat didefinisikan sebagai ekspektasi fungsi margin terhadap pasangan data (X,Y):

$$s = E_{X,Y}[mr(X,Y)] \quad (14)$$

Dimana:

- s : Strength atau kekuatan classifier ensemble Random Forest. Ini adalah rata-rata margin function across seluruh distribusi data. Nilai tinggi menunjukkan model kuat dan confident.
- $E_{X,Y}$: Expected value atau ekspektasi matematis terhadap distribusi bersama (joint distribution) dari fitur X dan label Y. Operator ini menghitung rata-rata weighted berdasarkan probabilitas kemunculan setiap pasangan (X,Y).

- $E_{X,Y}[\ldots]$: Notasi ekspektasi dengan tanda kurung siku menunjukkan operasi averaging.
- $mr(X,Y)$: Margin function (dari rumus 13) yang dievaluasi untuk pasangan (X,Y)

Dengan asumsi bahwa $s \geq 0$, serta menggunakan ketaksamaan *Chebyshev* dan penurunan variansi dari fungsi margin, diperoleh batas atas kesalahan generalisasi sebagai berikut:

$$\bar{E} \leq \bar{\rho}\left(\frac{1-s^2}{s^2}\right) \tag{15}$$

Dimana:

- $\overline{E}$: Generalization error atau kesalahan generalisasi, yaitu expected error rate model pada data baru yang belum pernah dilihat. Nilai berkisar 0-1, dimana 0 = perfect, 1 = worst.
- $\leq$: Simbol "kurang dari atau sama dengan", menunjukkan bahwa sisi kiri adalah batas atas (upper bound). Actual error bisa lebih rendah dari nilai ini.
- $\overline{\rho}$: Rho-bar, rata-rata korelasi (average correlation) antar decision trees dalam Random Forest. Nilai berkisar -1 hingga 1, dimana nilai mendekati 0 ideal (trees independent).
- s^2 : Kuadrat dari strength (dari rumus 14). Pengkuadratan memberikan nilai non-negatif dan mempenalti low strength secara eksponensial.
- $1-s^2$: Komplemen dari squared strength, mengukur "kelemahan" model.
- $\frac{1-s^2}{s^2}$: Rasio antara kelemahan dan kekuatan model. Nilai tinggi menunjukkan model lemah, nilai rendah menunjukkan model kuat.
- $\overline{\rho}\left(\frac{1-s^2}{s^2}\right)$: Produk dari correlation dan weakness ratio, memberikan upper bound untuk generalization error.

Dalam rumus tersebut, $\bar{E}$merupakan kesalahan generalisasi dan $\bar{\rho}$adalah rata-rata korelasi antar-pohon, yang dihitung sebagai::

$$\bar{\rho} = \frac{E_{\Theta,\Theta'}[\rho(\Theta,\Theta') \cdot sd(\Theta) \cdot sd(\Theta')]}{E_{\Theta,\Theta'}[sd(\Theta) \cdot sd(\Theta')]} \tag{16}$$

Dimana:

- $\overline{\rho}$: Rho-bar, rata-rata korelasi antar decision trees dalam ensemble, weighted berdasarkan variabilitas masing-masing tree.
- Θ : Parameter untuk tree pertama (seperti dijelaskan di rumus 13).
- Θ' : Theta-prime, parameter untuk tree kedua yang berbeda dari Θ. Digunakan untuk menghitung correlation antara pasangan trees.
- $E_{\Theta,\Theta'}$: Expected value terhadap distribusi bersama dari dua parameter tree yang berbeda. Operator ini mengaverage over semua possible pairs of trees.
- $\rho(\Theta,\Theta')$: Correlation coefficient antara dua trees dengan parameter Θ dan Θ'. Mengukur seberapa similar prediksi kedua trees. Nilai berkisar -1 hingga 1.

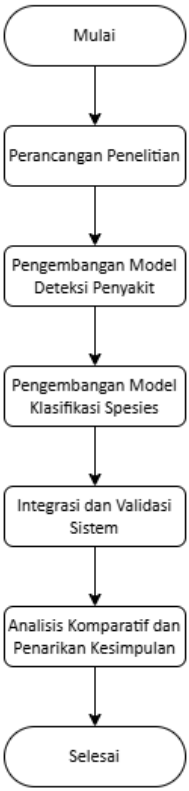
- $sd(\Theta)$: Standard deviation dari margin function untuk tree dengan parameter Θ across distribusi data.
- $sd(\Theta')$: Standard deviation untuk tree dengan parameter Θ', analog dengan sd(Θ).
- $sd(\Theta) \cdot sd(\Theta')$: Produk dari dua standard deviations, digunakan sebagai normalization factor dalam perhitungan correlation.
- Pembilang (numerator) $E_{\Theta,\Theta'}[\rho(\Theta,\Theta') \cdot sd(\Theta) \cdot sd(\Theta')]$: Pembilang ini menghitung rata-rata korelasi antar pohon, di mana setiap korelasi diberi bobot oleh produk standar deviasi masing-masing pohon. Dengan kata lain, ini menunjukkan seberapa besar hubungan (covariance) antar pohon dalam model.
- Penyebut (denominator) $E_{\Theta,\Theta'}[sd(\Theta) \cdot sd(\Theta')]$: Penyebut ini adalah rata-rata dari produk standar deviasi antar pohon, berfungsi sebagai faktor normalisasi agar korelasi yang dihitung berada pada skala yang tepat (−1 sampai 1).

Rumus ini menjelaskan bahwa kesalahan generalisasi dapat diminimalkan jika kekuatan klasifikasi tinggi (s besar) dan korelasi antar pohon rendah (ρ kecil), yang menunjukkan pentingnya diversifikasi antar pohon dalam *Random Forest* [42].

Dalam konteks diagnosis malaria, RF terbukti efektif dalam mengklasifikasikan berbagai jenis parasit *Plasmodium* berdasarkan fitur citra darah mikroskopis. Penelitian oleh Suryani dan rekan-rekan menunjukkan bahwa penggunaan RF dalam klasifikasi citra sel darah berhasil membedakan antara Plasmodium *Falciparum*, Plasmodium *Vivax* , serta jenis malaria lainnya dengan tingkat akurasi yang tinggi [19]. Keunggulan utama dari metode ini terletak pada kemampuannya dalam menangani data berdimensi tinggi dan menangkap kompleksitas pola non-*Linear* antar fitur. Selain itu, RF juga memberikan interpretasi yang lebih baik terhadap pentingnya setiap fitur dalam proses klasifikasi, yang dapat mendukung pengambilan keputusan klinis secara lebih akurat dan efisien. Dalam klasifikasi data medis, khususnya pada kasus diabetes, *Random Forest* menunjukkan performa yang baik dengan akurasi mencapai 86,4%, sensitivitas 88,2%, presisi 82,3%, dan F1-*Score* 85,1% setelah penerapan teknik penyeimbangan data menggunakan Smote-Tomek Link [43]. Selain itu, dalam studi lain, *Random Forest* mencapai AUC sebesar 0,8612 dalam klasifikasi diabetes setelah penerapan seleksi fitur berbasis Genetic Algorithm, menunjukkan peningkatan performa yang signifikan dibandingkan dengan model tanpa seleksi fitur [44]. Hasil-hasil ini menegaskan bahwa *Random Forest* merupakan algoritma yang andal dan efisien untuk diterapkan dalam berbagai sistem klasifikasi, baik dalam domain kesehatan maupun analisis opini publik.

Bab 3. Metodologi Penelitian

Guna memberikan gambaran sistematis mengenai alur kerja penelitian secara keseluruhan, Gambar 7 menampilkan *flowchart* perancangan penelitian yang mendeskripsikan tahapan-tahapan utama penelitian mulai dari studi literatur, pengumpulan dataset, pengembangan model deteksi dan klasifikasi, hingga analisis komparatif dan penarikan kesimpulan.



Gambar 7. *Flowchart* perancangan penelitian

Bab ini menguraikan secara sistematis kerangka kerja dan tahapan teknis yang diimplementasikan dalam penelitian. Metodologi ini dirancang untuk mencapai tujuan penelitian, yaitu menganalisis dan membandingkan performa algoritma *Support Vector Machine* (SVM) dan *Random Forest* (RF) dalam klasifikasi jenis malaria dari citra darah.

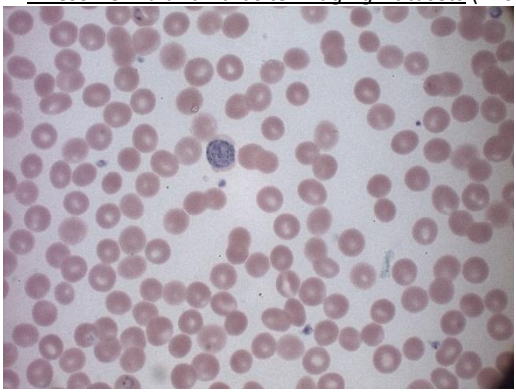
3.1. Perancangan Penelitian

1. Studi Literatur

Penelitian diawali dengan melakukan kajian literatur yang mendalam untuk memahami metode-metode terkini dalam deteksi penyakit darah menggunakan *deep learning*, khususnya arsitektur YOLO, serta untuk mengidentifikasi dataset publik yang relevan.

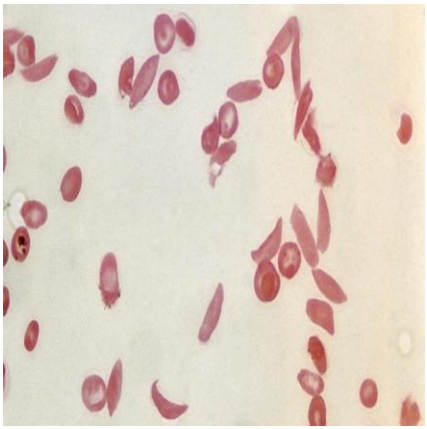
2. Pengumpulan Dataset

Penelitian ini menggunakan dataset berupa kumpulan citra darah manusia yang diambil melalui mikroskop. Citra mikroskopis darah yang terdiri dari tiga kategori utama yaitu citra yang mengandung parasit *Trypomastigote* dari repositori publik *Dataset Trypomastigotes* (61 citra), citra yang mengandung sel darah merah dari penderita anemia sel sabit dari repositori *Dataset Sickle-Shaped Blood Cells* (73 citra), dan citra sel darah yang terinfeksi parasit malaria dari repositori publik *Awesome Malaria Parasite Imaging Datasets* (210 citra).



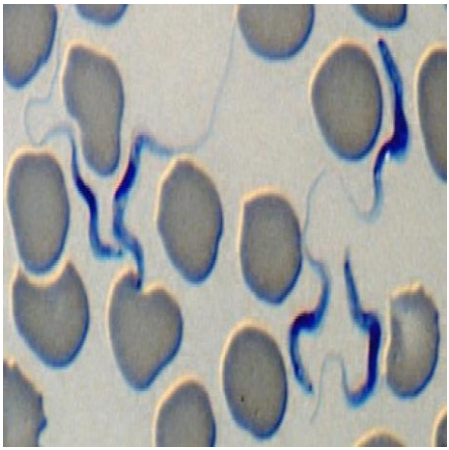
Gambar 8. Dataset Malaria

Dataset malaria pada Gambar 8 menunjukkan variabilitas morfologi parasit pada berbagai stadium perkembangan, menyediakan representasi komprehensif yang diperlukan untuk melatih model YOLO 11 dalam mendeteksi parasit malaria dengan akurasi tinggi pada kondisi visual yang beragam.



Gambar 9. Dataset Anemia

Gambar 9 mengilustrasikan contoh representatif dataset anemia dengan sel darah merah berbentuk sabit yang karakteristik, menyediakan ground truth visual yang esensial untuk pembelajaran model dalam mengidentifikasi pola morfologi spesifik anemia sel sabit.

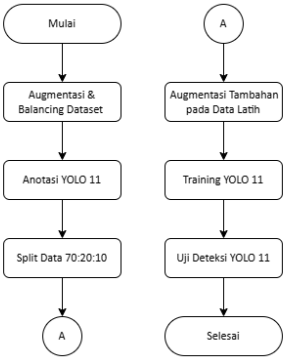


Gambar 10. Dataset *Trypomastigote*

Dataset *Trypomastigote* pada Gambar 10 menampilkan karakteristik morfologi khas sel yang terinfeksi parasit ini, memberikan variabilitas visual yang diperlukan untuk melatih model dalam membedakan *Trypomastigote* dari sel malaria dan anemia dengan akurat.

3.2. Pengembangan Model Deteksi Penyakit

Proses pengembangan model deteksi YOLO 11 melibatkan serangkaian tahapan yang sistematis dan terstruktur, sehingga Gambar 11 menyajikan *flowchart* pengembangan model deteksi penyakit yang mencakup tahapan augmentasi dataset, anotasi objek, *split* data, *training* YOLO 11, hingga evaluasi performa deteksi pada data uji.



Gambar 11. *Flowchart* Pengembangan Model Deteksi Penyakit

Fase kedua penelitian ini berfokus pada persiapan dataset dan pengembangan model deteksi primer yang mampu mengidentifikasi keberadaan penyakit darah secara umum. Tujuan utama dari tahap ini adalah untuk menghasilkan sebuah model *object detection* berbasis YOLO 11. Alur kerja pada fase ini, yang ditampilkan pada Gambar 11, mencakup serangkaian proses mulai dari studi literatur hingga pengujian awal model. Langkah-langkah yang dieksekusi pada fase ini adalah sebagai berikut:

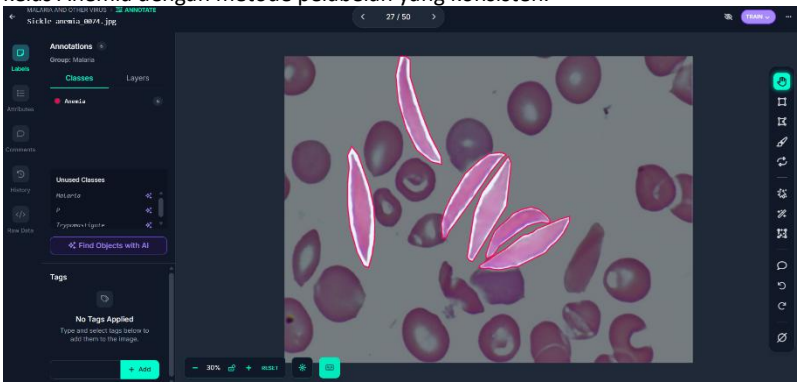
1. Augmentasi dan Balancing Dataset

   Untuk mengatasi ketidakseimbangan jumlah data antar kelas, diterapkan teknik augmentasi pada kelas minoritas. Proses ini menghasilkan distribusi data yang lebih seimbang, yaitu 210 citra untuk Malaria *cells*, 212 citra untuk *Sickle-shaped red blood cells*, dan 212 citra untuk *Trypomastigote cells*.
2. Anotasi YOLO 11

   Setiap objek (sel darah yang terinfeksi atau abnormal) pada keseluruhan dataset dianotasi menggunakan format YOLO. Proses ini melibatkan pembuatan Poligon di sekitar objek relevan dan pemberian label kelas yang sesuai untuk melatih model deteksi.

Gambar 12. Anotasi Data Malaria

Visualisasi anotasi pada Gambar 12 menunjukkan proses pelabelan parasit malaria yang sistematis dan presisi, memastikan bahwa dataset pelatihan memiliki ground truth berkualitas tinggi yang esensial untuk convergence model YOLO 11 yang efektif. Proses anotasi yang sistematis tidak hanya diterapkan pada dataset malaria, tetapi juga diperluas untuk mencakup sel darah abnormal lainnya, sebagaimana ditunjukkan pada Gambar 13 yang menampilkan hasil anotasi untuk kelas Anemia dengan metode pelabelan yang konsisten.



Gambar 13. Anotasi Data Anemia

Gambar 13 memperlihatkan hasil anotasi sel darah anemia dengan bounding box yang akurat mengelilingi sel berbentuk sabit karakteristik, mengonfirmasi bahwa proses pelabelan data dilakukan secara cermat untuk mendukung pembelajaran model yang optimal. Melengkapi proses anotasi untuk ketiga kelas penyakit darah yang menjadi fokus penelitian, Gambar 14 mengilustrasikan hasil

pelabelan untuk kelas *Trypomastigote* dengan tingkat presisi yang setara, memastikan konsistensi kualitas ground truth di seluruh dataset multi-kelas.

Gambar 14. Anotasi Data *Trypomastigote*

Proses anotasi pada Gambar 14 mendemonstrasikan pemberian label polygon yang presisi untuk sel *Trypomastigote*, memastikan bahwa ground truth yang digunakan dalam pelatihan model memiliki kualitas tinggi dan representatif terhadap karakteristik morfologi yang sesungguhnya.

3. Split Data

Dataset yang telah di-anotasi kemudian dibagi menjadi tiga himpunan data terpisah dengan rasio 70:20:10, yaitu data latih (*training*), data validasi (*validation*), dan data uji (*testing*).

4. Augmentasi Tambahan pada Data Latih

Pada tahap ini, dilakukan proses augmentasi lanjutan khusus pada subset data latih (*train set*) dengan tujuan untuk meningkatkan keragaman citra serta memperkuat kemampuan generalisasi model YOLO 11 terhadap variasi bentuk dan orientasi objek. Proses augmentasi dilakukan setelah tahap pembagian data (*split*) sehingga hanya diterapkan pada data latih, guna mencegah kebocoran informasi (*data leakage*) ke data validasi maupun data uji. Setelah augmentasi tahap kedua, jumlah citra pada masing-masing kelas meningkat secara signifikan, yaitu dari 447 menjadi 1.000 citra untuk kelas Malaria, dari 452 menjadi 1.000 citra untuk kelas *Sickle* Anemia, serta dari 435 menjadi 1.000 citra untuk kelas *Trypomastigote*. Penambahan jumlah data ini diharapkan dapat memperbaiki keseimbangan distribusi antar kelas serta meningkatkan stabilitas proses pelatihan model. Teknik augmentasi yang diterapkan melibatkan kombinasi transformasi geometrik dan rotasional, yang dirancang untuk mensimulasikan berbagai kemungkinan posisi dan orientasi sel pada citra mikroskopis. Jenis augmentasi yang digunakan meliputi Horizontal Flip (*mirror*), Vertical Flip, Rotasi 90°, Rotasi 180°, Rotasi 270°, serta Rotasi Kecil dengan rentang −15° hingga 15°.

Kombinasi transformasi tersebut memungkinkan model untuk mempelajari fitur visual secara lebih menyeluruh dari berbagai arah pandang, sehingga hasil deteksi menjadi lebih robust terhadap variasi orientasi dan posisi sel darah pada citra input.

5. *Training* YOLO 11

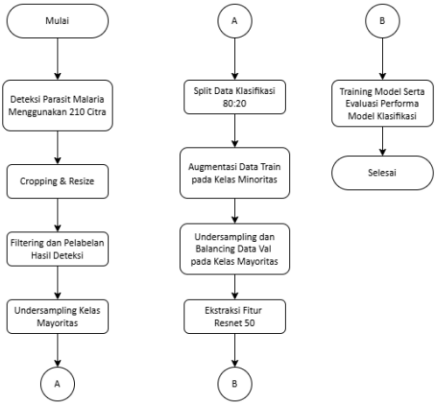
Model YOLO 11 dilatih menggunakan data latih dan divalidasi performanya secara periodik menggunakan data validasi. Proses pelatihan ini bertujuan untuk mengoptimalkan bobot model agar dapat mengenali pola-pola visual dari ketiga kategori penyakit.

6. Uji Deteksi YOLO 11

Model yang telah selesai dilatih kemudian diuji kinerjanya menggunakan data uji. Pengujian ini dilakukan untuk mengevaluasi kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya dan memastikan bahwa model siap digunakan untuk fase berikutnya.

3.3. Pengembangan Model Klasifikasi Spesies

Untuk memberikan visualisasi yang jelas mengenai alur kerja klasifikasi spesies malaria, Gambar 15 mendeskripsikan *flowchart* pengembangan model klasifikasi spesies yang dimulai dari deteksi parasit malaria menggunakan YOLO 11, dilanjutkan dengan *cropping* dan ekstraksi fitur menggunakan ResNet-50, hingga tahap klasifikasi menggunakan algoritma SVM dan *Random Forest*.



Gambar 15. Flowchart Pengembangan Model Klasifikasi Spesies

Setelah model deteksi primer berhasil dikembangkan, fase kedua penelitian berfokus pada pengembangan model klasifikasi sekunder. Tujuan utama fase ini adalah untuk membangun dan membandingkan dua model klasifikasi, SVM dan

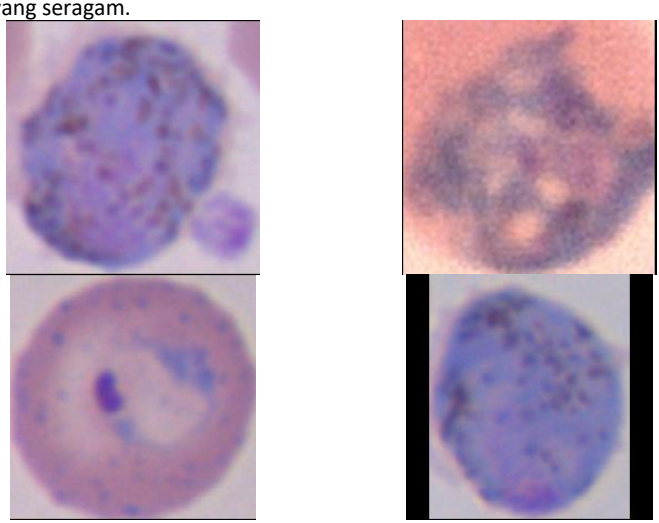
RF, yang mampu mengidentifikasi spesies spesifik parasit malaria (*Plasmodium Falciparum, Plasmodium Malariae, Plasmodium Ovale, Plasmodium Vivax*) dari citra yang telah terdeteksi positif malaria oleh model YOLO 11. Alur kerja yang sistematis pada fase ini disajikan pada Gambar 15 dengan proses-proses yang terperinci pada fase ini ialah sebagai berikut:

1. Deteksi Parasit Malaria Mengunakan 210 Citra

   Pada tahap ini, model YOLO 11 yang telah dilatih digunakan untuk memproses 210 citra malaria.

2. Cropping & Resize

   Setiap parasit yang terdeteksi oleh *bounding box* diekstraksi (*cropping*) dan diubah ukurannya menjadi 224x224 piksel untuk menciptakan *Region of Interest* (ROI) yang seragam.



Gambar 16. Hasil *Cropping* & *Resize* Malaria

Gambar 16 mengilustrasikan hasil tahap preprocessing yang menghasilkan Region of Interest (ROI) berukuran seragam 224×224 piksel dengan karakteristik visual parasit malaria yang terpelihara, memastikan input yang konsisten dan optimal untuk tahap ekstraksi fitur menggunakan arsitektur ResNet50.

3. Filtering dan Pelabelan Hasil Deteksi

   Hasil ekstraksi ROI kemudian difilter secara manual untuk memisahkan deteksi yang akurat dari deteksi yang salah. ROI yang akurat selanjutnya dilabeli sesuai dengan empat spesies malaria, menghasilkan dataset baru dengan distribusi: *Falciparum* (1803), *Malariae* (59), *Ovale* (39), dan *Vivax* (99).

4. Undersampling Kelas Mayoritas

Untuk menangani ketidakseimbangan data, strategi penyeimbangan hibrida diterapkan. Proses ini diawali dengan penerapan teknik *undersampling* pada kelas mayoritas (*Falciparum*) untuk mengurangi jumlah sampelnya menjadi 500.

5. Split Data Klasifikasi

Dataset yang telah diseimbangkan ini kemudian dibagi lagi dengan rasio 80:20 untuk data latih dan data validasi.

6. Augmentasi Data Train pada Kelas Minoritas

Untuk mengatasi kembali ketidakseimbangan pada data latih, peneliti menerapkan teknik augmentasi secara spesifik pada kelas-kelas minoritas (*Malariae*, *Ovale*, *Vivax*) hingga masing-masing mencapai jumlah yang sebanding dengan kelas mayoritas (400 sampel).

7. Undersampling dan Balancing Data Val pada Kelas Mayoritas

Data validasi juga diseimbangkan dengan teknik undersampling untuk memastikan bahwa setiap kelas memiliki jumlah sampel yang sama (8 sampel per kelas), sehingga evaluasi model menjadi lebih objektif.

8. Ekstraksi Fitur

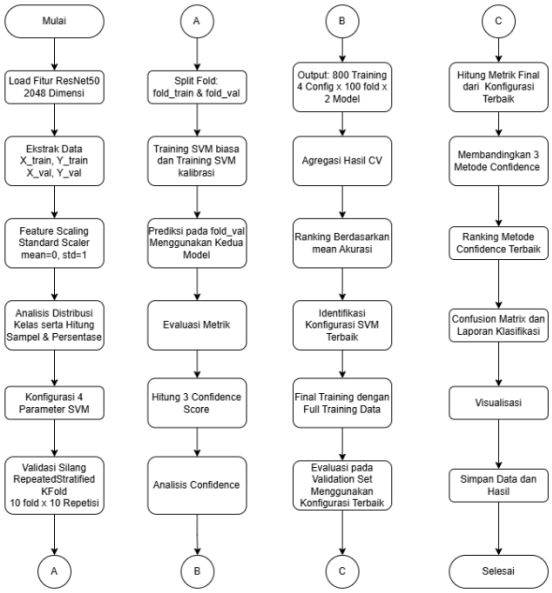
Dalam penelitian ini, arsitektur *Convolutional Neural Network* (CNN) ResNet-50 digunakan sebagai ekstraktor fitur berbasis *deep learning*. Setiap citra *Region of Interest* (ROI) dari data latih dan validasi diproses melalui model ini untuk menghasilkan representasi fitur yang informatif. Pemilihan ResNet-50 didasarkan pada kemampuannya mengekstraksi karakteristik visual secara mendalam dengan stabilitas tinggi. Melalui mekanisme *residual learning*, model ini mampu mengatasi masalah *vanishing gradient* sehingga informasi penting dari citra tetap terpelihara meskipun memiliki struktur jaringan yang kompleks. Hasil ekstraksi kemudian direpresentasikan sebagai vektor numerik berdimensi 2048.

9. *Training* Model Serta Evaluasi Performa Model Klasifikasi

Vektor fitur yang dihasilkan kemudian digunakan untuk melatih dua model *machine learning* klasik: *Support Vector Machine* (SVM) dan *Random Forest* (RF).

a. SVM

Proses pelatihan data menggunakan model SVM divisualisasikan secara komprehensif dalam Gambar 17 yang menampilkan *workflow* lengkap dari persiapan data hingga evaluasi model final.



Gambar 17. *Workflow Training* SVM

Gambar 17 menunjukkan *workflow* lengkap *training* SVM untuk klasifikasi gambar menggunakan fitur ResNet50 dengan fokus pada analisis tiga metode *confidence score*. Alur kerja ini terbagi menjadi empat tahapan utama yang saling berkesinambungan: persiapan data dan konfigurasi, *cross-validation* dengan *training* 800 model, final *training* dan evaluasi, serta visualisasi dan penyimpanan hasil. Setiap tahapan dirancang untuk menghasilkan model SVM terbaik dengan *confidence score* yang optimal melalui eksperimen komprehensif sebanyak 1,200 kali. Untuk memahami workflow secara menyeluruh, berikut penjelasan detail setiap blok proses yang digambarkan dalam flowchart:

- Mulai

  Program dimulai sebagai titik awal eksekusi sistem. Tahap ini menandakan inisialisasi seluruh proses eksperimen SVM untuk klasifikasi gambar menggunakan fitur ResNet50 yang akan menghasilkan total 802 model (800 model untuk *cross-validation* evaluasi + 2 model untuk *production*).

- Load Fitur ResNet50 2048 Dimensi

  Pada tahap ini, program memuat fitur hasil ekstraksi ResNet50 yang telah disimpan dalam file .npy dengan dimensi 2048. Fitur ini merupakan representasi vektor dari gambar yang telah diekstrak oleh arsitektur *deep learning* ResNet50,

yang kemudian akan digunakan sebagai input untuk model SVM. Fitur 2048 dimensi ini menjadi input utama untuk seluruh eksperimen yang akan dilakukan.

- Ekstrak Data X_train, Y_train, X_val, Y_val

Setelah fitur dimuat, program melakukan ekstraksi data menjadi empat komponen utama yaitu X_train (fitur *training*), Y_train (label *training*), X_val (fitur *validation*), dan Y_val (label *validation*). Proses ini juga mengekstrak informasi nama kelas untuk keperluan evaluasi dan visualisasi hasil klasifikasi nantinya. Data *validation* disimpan terpisah dan tidak akan digunakan dalam proses *cross-validation*, melainkan hanya untuk evaluasi final model terbaik.

- *Feature Scaling Standard Scaler mean*=0, std=1

Tahap feature scaling dilakukan menggunakan *StandardScaler* untuk mentransformasi seluruh fitur ke skala standar dengan *mean*=0 dan *standard deviation*=1. *Scaler* di-fit terlebih dahulu pada data *training* untuk mempelajari parameter *mean* dan std, kemudian parameter yang sama ditransformasi pada data *training* dan *validation* untuk memastikan konsistensi skala dan mencegah kebocoran data. *Scaling* ini penting untuk SVM agar semua fitur memiliki kontribusi yang seimbang dalam proses klasifikasi.

- Analisis Distribusi Kelas serta Hitung Sampel dan Persentasi

Program melakukan analisis distribusi kelas untuk memahami karakteristik dataset dengan menghitung jumlah sampel per kelas dan persentase masing-masing kelas. Analisis ini penting untuk mengidentifikasi potensi *class imbalance* yang dapat mempengaruhi performa model dan strategi validasi yang akan digunakan. Informasi distribusi ini juga membantu dalam interpretasi hasil evaluasi nantinya.

- Konfigurasi 4 Parameter SVM

Pada tahap ini, program mengatur empat konfigurasi SVM yang akan diuji dalam eksperimen yaitu: (1) *Linear* kernel, (2) RBF kernel dengan gamma='scale', (3) *Polynomial* kernel dengan degree=2, dan (4) RBF kernel dengan parameter C=10. Setiap konfigurasi merepresentasikan karakteristik *decision boundary* yang berbeda untuk dievaluasi performanya. Keempat konfigurasi ini dipilih untuk membandingkan berbagai pendekatan pemisahan kelas dan menemukan yang paling optimal untuk fitur ResNet50.

- Validasi Silang *RepeatedStratified*KFold 10 fold x 10 Repetisi

Program mengimplementasikan strategi *cross-validation* menggunakan *RepeatedStratified*KFold dengan 10 fold dan 10 repetisi, menghasilkan total 100 eksperimen per konfigurasi. *Stratified folding* memastikan distribusi kelas tetap proporsional di setiap fold, sementara repetisi meningkatkan reliabilitas hasil evaluasi dengan mengurangi variasi akibat *random splitting*. Strategi ini menghasilkan evaluasi yang sangat robust dan reliabel untuk pemilihan konfigurasi terbaik.

- *Split* Fold fold_train & fold_val

Pada setiap iterasi fold, program melakukan *split* data menjadi fold_train (data *training* untuk fold tersebut) dan fold_val (data *validation* untuk fold tersebut). Proses *splitting* ini dilakukan secara *stratified* untuk mempertahankan proporsi kelas yang seimbang antara *training* dan *validation* di setiap fold. Split ini berbeda dengan *validation* set final yang telah disisihkan di awal, karena fold_val hanya digunakan untuk evaluasi cross-*validation*.

- *Training* SVM biasa dan *Training* SVM kalibrasi

Untuk setiap fold, program melatih dua jenis model secara paralel: (1) SVM biasa yang menggunakan konfigurasi standar untuk prediksi utama dan menghasilkan *decision function*, dan (2) *CalibratedClassifier*CV yang membungkus SVM dengan *Platt Scaling* untuk menghasilkan probabilitas terkalibrasi. Kedua model ini dilatih pada fold_train yang sama untuk memastikan perbandingan yang fair. *Training* kedua model di setiap fold ini penting untuk membandingkan efektivitas *confidence score* yang berbeda.

- Prediksi pada fold_val Menggunakan Kedua Model

Setelah *training* selesai, kedua model (SVM biasa dan SVM terkalibrasi) digunakan untuk melakukan prediksi pada fold_val. SVM biasa menghasilkan prediksi kelas dan *decision function* yang akan digunakan untuk *confidence Normalized* dan Raw, sementara *CalibratedClassifier*CV menghasilkan prediksi kelas dan probabilitas terkalibrasi untuk setiap kelas yang akan digunakan untuk *confidence Calibrated*. Hasil prediksi dari kedua model ini menjadi dasar untuk analisis *confidence score*.

- Evaluasi Metrik

Program menghitung berbagai metrik evaluasi dari hasil prediksi kedua model terhadap fold_val, meliputi akurasi, presisi, *recall*, F1-*score*, dan metrik klasifikasi lainnya. Metrik-metrik ini digunakan untuk mengukur performa setiap konfigurasi SVM pada fold tersebut. Evaluasi metrik dilakukan untuk setiap fold sehingga memberikan gambaran performa yang komprehensif dan tidak bias terhadap *split* data tertentu.

- Hitung 3 *Confidence Score*

Pada tahap ini, program menghitung tiga jenis *confidence score* untuk setiap prediksi: (1) *Confidence* Normalized menggunakan *Sigmoid transformation* pada *decision function* untuk menghasilkan nilai [0-1] sebagai novelty untuk mencapai kesetaraan dengan RF (Random Forest) probabilitas *margins*, (2) *Confidence* Raw yang merupakan nilai *decision function* tanpa normalisasi sebagai pembanding, dan (3) *Confidence Calibrated* dari model *CalibratedClassifier*CV yang menghasilkan probabilitas terkalibrasi menggunakan *Platt Scaling*. Ketiga metode ini dihitung untuk setiap prediksi di setiap fold, menghasilkan total 1,200 eksperimen *confidence* (4 konfigurasi × 100 fold × 3 metode *confidence*).

- Analisis *Confidence*

Program melakukan analisis mendalam terhadap ketiga metode *confidence score* dengan menghitung: *mean confidence* untuk prediksi benar vs salah, *confidence* gap (selisih *confidence* antara prediksi benar dan salah), *correlation* antara *confidence* dengan *accuracy*, distribusi statistik *confidence*, dan analisis *threshold* optimal. Analisis ini mengungkap efektivitas masing-masing metode dalam mengukur kepercayaan prediksi model dan kemampuannya membedakan prediksi yang benar dan salah. Hasil analisis ini penting untuk menentukan metode *confidence* terbaik yang akan direkomendasikan untuk *deployment*.

- Output: 800 *Training* 4 Config x 100 fold x 2 Model

Proses cross-*validation* menghasilkan total 800 kali *training* model yang berasal dari 4 konfigurasi SVM dikalikan 100 fold (10 fold × 10 repetisi) dikalikan 2 jenis model (SVM biasa dan *CalibratedClassifier*CV). Setiap *training* menghasilkan metrik dan *confidence score* yang kemudian diagregasi untuk analisis lebih lanjut. Total 800 model ini merupakan bagian dari evaluasi *cross-validation*, dan belum termasuk 2 model final yang akan dilatih untuk *production*, sehingga keseluruhan program melatih 802 model.

- Agregasi Hasil CV

Hasil dari 100 fold untuk setiap konfigurasi diagregasi dengan menghitung rata-rata (*mean*) dan standar deviasi (std) dari semua metrik performa dan *confidence score*. Agregasi ini memberikan gambaran statistik yang *robust* tentang performa setiap konfigurasi SVM dengan memperhitungkan variabilitas antar fold. *Mean* dan std ini menjadi dasar untuk membandingkan konfigurasi secara objektif dan reliabel.

- Ranking Berdasarkan *mean* Akurasi

Program melakukan ranking terhadap keempat konfigurasi SVM berdasarkan *mean accuracy* dari model SVM biasa (bukan model terkalibrasi). Ranking ini menentukan konfigurasi mana yang memiliki performa klasifikasi terbaik secara konsisten di seluruh fold *cross-validation*. Penggunaan *mean accuracy* sebagai kriteria utama memastikan pemilihan konfigurasi yang paling akurat dalam mengklasifikasikan gambar.

- Identifikasi Konfigurasi SVM Terbaik

Berdasarkan hasil ranking, program mengidentifikasi dan memilih konfigurasi SVM dengan *mean accuracy* tertinggi sebagai konfigurasi terbaik (misalnya SVM *Linear*). Konfigurasi ini akan digunakan untuk melatih model final yang akan di-deploy untuk *production* dan dievaluasi pada *validation* set terpisah. Identifikasi ini menandai berakhirnya fase eksplorasi konfigurasi dan dimulainya fase *production* model *training*.

- Final *Training* dengan Full *Training* Data

Setelah konfigurasi terbaik teridentifikasi, program melatih dua model final menggunakan seluruh data *training* (tanpa split fold): *best*_svm sebagai model utama untuk prediksi *production* dan *best_Calibrated*_svm sebagai model

pendukung untuk *confidence Calibrated*. *Training* menggunakan full data memaksimalkan informasi yang dipelajari model untuk performa optimal di *deployment*. *Best*_svm dipilih sebagai model utama karena memiliki *accuracy* tertinggi, kecepatan prediksi lebih cepat tanpa overhead kalibrasi, dan mampu menghasilkan *confidence score* Normalized dan Raw yang lebih efisien. *Best_Calibrated*_svm hanya digunakan sebagai pembanding untuk menghasilkan *confidence score Calibrated* dan probabilitas yang dapat diinterpretasikan untuk aplikasi yang memerlukannya. Kedua model final ini merupakan model ke-801 dan 802 dari total 802 model yang dilatih dalam seluruh program.

- Evaluasi pada *Validation* Set Menggunakan Konfigurasi Terbaik

  Model final (*best*_svm dan *best_Calibrated*_svm) dievaluasi pada *validation* set terpisah yang tidak pernah dilihat selama *training* maupun cross-*validation*. Evaluasi ini menghitung semua metrik performa dari *best*_svm, membandingkan ketiga metode *confidence score*, dan menghasilkan *confusion matrix* serta laporan klasifikasi untuk analisis detail performa model pada data yang belum pernah dilihat. Evaluasi pada *validation* set ini memberikan estimasi performa yang realistis ketika model di-deploy pada data baru di *production*.

- Hitung Metrik Final dari Konfigurasi Terbaik

  Program menghitung metrik evaluasi final secara komprehensif dari *best*_svm pada *validation* set, meliputi *accuracy*, precision, recall, F1-*score* per kelas, *macro* dan *weighted averages*, serta metrik tambahan lainnya. Metrik final ini merepresentasikan performa aktual model yang akan dialami saat *deployment production* dan menjadi acuan untuk menilai apakah model sudah cukup baik untuk digunakan atau perlu perbaikan lebih lanjut.

- Membandingkan 3 Metode *Confidence*

  Pada tahap ini, dilakukan perbandingan mendalam antara ketiga metode *confidence score* (Normalized, Raw, dan *Calibrated*) berdasarkan hasil prediksi pada *validation* set. Perbandingan mencakup *mean confidence* untuk prediksi benar vs salah, *confidence* gap, *correlation* dengan *accuracy*, dan karakteristik distribusi masing-masing metode. Analisis ini menunjukkan bahwa Normalized *Sigmoid* menghasilkan *confidence score* yang lebih comparable dengan Random Forest probability margins, membuktikan efektivitas novelty yang diusulkan dalam program ini.

- Ranking Metode *Confidence* Terbaik

  Program melakukan ranking terhadap ketiga metode *confidence* berdasarkan skor gabungan (*correlation* × gap) yang mengukur seberapa baik metode tersebut dalam membedakan prediksi benar dan salah serta berkorelasi dengan *accuracy*. Metode dengan skor tertinggi (biasanya Normalized *Sigmoid* unggul dalam *correlation* dan gap) dianggap paling efektif dalam mengukur kepercayaan prediksi dan direkomendasikan untuk digunakan. Ranking ini memberikan rekomendasi yang berbasis data untuk memilih metode *confidence* yang optimal.

- *Confusion matrix* dan Laporan Klasifikasi

Program menghasilkan *confusion matrix* yang memvisualisasikan pola prediksi benar dan salah per kelas, serta laporan klasifikasi yang menampilkan presisi, *recall*, dan F1-*score* untuk setiap kelas. Visualisasi ini membantu mengidentifikasi kelas mana yang sulit dibedakan dan pola kesalahan klasifikasi yang terjadi. *Confusion matrix* memberikan insight mendalam tentang bagaimana model melakukan kesalahan dan kelas-kelas mana yang sering tertukar satu sama lain.

- Visualisasi

Tahap visualisasi menghasilkan 9 plot utama yang mencakup: distribusi *accuracy* per konfigurasi SVM, perbandingan *confidence* gap (*Normalized* vs Raw), *correlation* perbandingan antar metode *confidence*, *confusion matrix* dari *best*_svm, distribusi *confidence* untuk ketiga metode, *scatter plots* (*accuracy* vs *confidence* gap), perbandingan raw vs *normalized confidence*, dan ranking metode *confidence*. Visualisasi ini memberikan visual yang mudah dipahami tentang hasil eksperimen dan membantu dalam interpretasi serta presentasi hasil penelitian.

- Simpan Data dan Hasil

Program menyimpan seluruh hasil eksperimen dalam berbagai format untuk dokumentasi dan penggunaan kedepannya: model *best*_svm dan *best_Calibrated*_svm beserta scaler dalam format .pkl untuk *deployment*, hasil detail per fold dalam CSV, ringkasan perbandingan per konfigurasi, perbandingan final *validation*, analisis prediksi dengan semua *confidence score*s, dan hasil lengkap dalam format NPZ untuk analisis lanjutan. Penyimpanan komprehensif ini memastikan hasil eksperimen dapat direproduksi dan model dapat langsung di-deploy ke *production*.

- Selesai

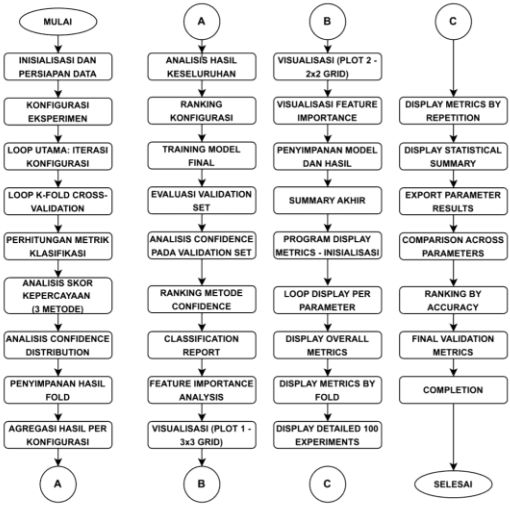
Program berakhir setelah menampilkan ringkasan dan *insights* yang mencakup: konfigurasi SVM terbaik (misalnya SVM *Linear*) dan performanya, analisis efek normalisasi pada *confidence score* yang menunjukkan *Normalized Sigmoid* lebih *comparable* dengan *Random Forest*, perbandingan *best*_svm vs *best_Calibrated*_svm yang menunjukkan akurasi sama tetapi *best*_svm lebih efisien untuk *deployment*, ranking metode *confidence* terbaik dengan *Normalized Sigmoid* yang biasanya unggul dalam *correlation* dan gap, rekomendasi penggunaan *best*_svm untuk prediksi *production* standar dan *best_Calibrated*_svm hanya jika membutuhkan probabilitas yang dapat diinterpretasikan, serta total eksekusi waktu eksperimen.

Poin utama akhir mencakup: input berupa fitur ResNet50 2048 dimensi, proses *training* 802 model (800 untuk *cross-validation* evaluasi + 2 untuk *production*), eksperimen sebanyak 1,200 kali (4 konfigurasi × 100 fold × 3 metode *confidence*), output berupa *best*_svm sebagai model utama dengan *confidence score* ternormalisasi [0-1] menggunakan *Sigmoid normalization* sebagai unsur kebaruan untuk mencapai kesetaraan dengan RF probabilitas *margins*, dan

*best_Calibrated*_svm sebagai model pendukung untuk analisis *confidence Calibrated* dan aplikasi yang memerlukan probabilitas. Semua hasil telah tersimpan dan siap untuk *deployment production* atau analisis lebih lanjut.

b. *Random Forest*

Proses pelatihan data menggunakan model *Random Forest* divisualisasikan secara komprehensif dalam Gambar 18 yang menampilkan *workflow* lengkap dari persiapan data hingga evaluasi model final.



Gambar 18. *Workflow Training Random Forest*

Gambar 18 menunjukkan *workflow* lengkap *training* SVM untuk klasifikasi gambar menggunakan fitur ResNet50. Untuk memahami workflow secara menyeluruh, berikut penjelasan detail setiap blok proses yang digambarkan dalam flowchart:

- Inisialisasi dan Persiapan Data

Proses dimulai dengan mengimpor pustaka utama seperti *NumPy, Pandas, Scikit-Learn, Matplotlib, dan Seaborn*, serta mengatur filter peringatan untuk menjaga kebersihan *output*. Data fitur malaria hasil ekstraksi ResNet50 dimuat dari file .npy, kemudian diekstraksi ke dalam variabel fitur dan label untuk set pelatihan dan validasi. Setelah memverifikasi keberhasilan pemuatan data, program menampilkan informasi dataset seperti jumlah sampel, fitur, dan kelas. Fitur-fitur tersebut kemudian dinormalisasi menggunakan *StandardScaler* (proses *fit* pada data latih dan *transform* pada kedua set) untuk memastikan skala data

konsisten, diikuti dengan analisis distribusi kelas untuk memahami proporsi setiap kategori.

- Konfigurasi Eksperimen

Pada tahap ini, program mendefinisikan empat konfigurasi parameter *Random Forest* yang berbeda, mencakup variasi pada *bootstrap*, kedalaman maksimal pohon (max_depth=15), jumlah estimator (n_estimators=100), dan kriteria *Gini*. Strategi validasi silang ditetapkan menggunakan *RepeatedStratifiedKFold* dengan 10 *Fold* dan 10 repetisi, menghasilkan total 400 unit eksperimen. Seluruh struktur penyimpanan hasil disiapkan, termasuk defaultdict untuk menyimpan metrik dan statistik skor kepercayaan, diikuti dengan dimulainya pencatat waktu eksperimen.

- *Loop* Utama: Iterasi Konfigurasi

Program melakukan iterasi melalui setiap konfigurasi *Random Forest* yang telah didefinisikan. Untuk setiap konfigurasi, sistem mencetak nama pengujian dan menyiapkan daftar kosong khusus untuk menampung metrik performa (akurasi, presisi, *recall*, *F1-score*) serta statistik kepercayaan dari tiga metode yang berbeda (Improved, Legacy, dan *Calibrated*). Penghitung *Fold* diatur ulang ke nol pada setiap awal iterasi konfigurasi untuk memastikan pelacakan data yang akurat.

- *Loop* K-*Fold Cross-Validation*

Data yang telah distandarisasi dibagi menggunakan pembagian indeks dari *RepeatedStratifiedKFold*. Pada setiap *Fold*, model RandomForestClassifier standar dan *Calibrated*ClassifierCV (versi terkalibrasi) dilatih menggunakan data latih *Fold* tersebut. Setelah pelatihan, model digunakan untuk melakukan prediksi pada data validasi *Fold*, yang kemudian menjadi dasar untuk seluruh perhitungan metrik dan analisis kepercayaan pada tahap berikutnya.

- Perhitungan Metrik Klasifikasi

Segera setelah prediksi diperoleh, program menghitung skor akurasi serta metrik klasifikasi lainnya seperti presisi, *recall*, dan *F1-score* menggunakan metode rata-rata terbobot (*weighted average*). Penggunaan rata-rata terbobot sangat penting untuk memberikan gambaran performa yang akurat meskipun terdapat potensi ketidakseimbangan jumlah sampel antar kelas. Seluruh metrik yang dihitung kemudian dimasukkan ke dalam daftar hasil masing-masing konfigurasi.

- Analisis Skor Kepercayaan (3 Metode)

Evaluasi tingkat keyakinan prediksi dilakukan melalui tiga pendekatan: Metode *Improved* menggunakan selisih probabilitas antara kelas teratas (margin), metode *Legacy* menggunakan probabilitas maksimal, dan metode *Calibrated* menggunakan probabilitas dari model yang telah dikalibrasi. Program menganalisis distribusi skor kepercayaan ini untuk melihat bagaimana perilaku model saat memberikan prediksi yang benar dibandingkan dengan prediksi yang salah.

- Analisis *Confidence Distribution*

Data kepercayaan dipisahkan berdasarkan kebenaran prediksinya untuk menghitung statistik deskriptif seperti rata-rata, standar deviasi, dan median. Fokus utama di sini adalah menghitung *Confidence gap* (selisih antara kepercayaan benar dan salah) serta korelasi kepercayaan dengan akurasi aktual. Selain itu, dilakukan analisis ambang batas (*threshold*) pada berbagai persentil untuk mengukur peningkatan akurasi ketika hanya prediksi dengan kepercayaan tinggi yang diambil.

- Penyimpanan Hasil *Fold*

Seluruh metrik performa dan statistik kepercayaan dari setiap *Fold* (*Improved, Legacy, dan Calibrated*) dimasukkan ke dalam penampung *detailed_results*. Program juga mencetak laporan kemajuan setiap 20 *Fold* untuk memberikan informasi status eksekusi kepada pengguna. Proses ini terus berulang hingga seluruh *Fold* dan seluruh konfigurasi selesai diproses.

- Agregasi Hasil Per Konfigurasi

Setelah semua repetisi *Fold* untuk satu konfigurasi selesai, program melakukan agregasi data dengan menghitung nilai rata-rata dan standar deviasi untuk semua metrik. Rata-rata *Confidence gap* dan korelasi dari ketiga metode kepercayaan dihitung dan disimpan ke dalam struktur data utama. Ringkasan konfigurasi tersebut kemudian dicetak sebelum sistem berpindah ke konfigurasi parameter berikutnya.

- Analisis Hasil Keseluruhan

Setelah semua iterasi konfigurasi tuntas, pencatat waktu dihentikan untuk menghitung total durasi eksperimen. Program membuat *DataFrame* rangkuman yang menggabungkan seluruh hasil konfigurasi, menampilkan tabel perbandingan yang mencakup nilai akurasi, *F1-score*, *Confidence gap*, dan korelasi dari semua metode yang diuji secara berdampingan untuk analisis komparatif yang mudah.

- *Ranking* Konfigurasi

Sistem melakukan pengurutan otomatis pada seluruh konfigurasi berdasarkan rata-rata akurasi tertinggi. Hasil peringkat ini menampilkan posisi, akurasi, serta perbaikan pada aspek korelasi dan *Confidence gap* (membandingkan metode *Improved* vs Legacy). Konfigurasi di posisi teratas secara otomatis ditetapkan sebagai best_config untuk digunakan dalam tahap pelatihan model akhir.

- *Training* Model Final

Menggunakan parameter dari konfigurasi terbaik, program melatih ulang model *Random Forest* dan model terkalibrasi menggunakan seluruh data latih yang telah distandarisasi (X_train_scaled). Pelatihan ulang pada set data lengkap ini bertujuan untuk memaksimalkan pengetahuan model sebelum dilakukan evaluasi akhir pada data validasi independen yang terpisah sejak awal.

- Evaluasi *Validation Set*

Model terbaik yang sudah dilatih penuh kemudian diuji pada set validasi akhir. Program menghitung metrik performa final yang meliputi akurasi, presisi, *recall*,

dan *F1-score*. Hasil ini mencerminkan kinerja model sesungguhnya pada data "dunia nyata" yang tidak pernah dilihat model selama proses pelatihan atau validasi silang.

- Analisis *Confidence* pada *Validation Set*

Pada set validasi, ketiga metode kepercayaan dibandingkan kembali secara mendetail melalui fungsi compare_confidence_methods_rf. Analisis ini mencakup perbandingan distribusi skor, perhitungan *Confidence gap*, korelasi, dan efektivitas ambang batas kepercayaan tinggi terhadap akurasi final. Hal ini memberikan validasi tambahan mengenai keandalan estimasi kepercayaan model.

- *Ranking* Metode Confidence

Program memberikan peringkat pada metode kepercayaan dengan menghitung skor gabungan yang memberi bobot 60% pada korelasi dan 40% pada *Confidence gap*. Hasil peringkat dicetak untuk menentukan metode kepercayaan mana yang paling efektif dalam membedakan prediksi benar dan salah, yang kemudian ditetapkan sebagai metode kepercayaan terbaik untuk sistem ini.

- *Classification Report*

Untuk analisis mendalam per kelas, program menghasilkan laporan klasifikasi dan matriks kebingungan (*confusion matrix*) pada set validasi. Laporan ini merinci presisi, *recall*, dan *F1-score* serta dukungan jumlah data untuk setiap kelas malaria, memungkinkan identifikasi kekuatan dan kelemahan model pada kategori spesifik.

- *Feature Importance* Analysis

Nilai kepentingan fitur (*Feature Importance*) diekstraksi dari model *Random Forest* terbaik untuk mengetahui fitur visual mana yang paling berkontribusi dalam klasifikasi malaria. Program mengurutkan fitur ini dari yang paling berpengaruh, menampilkan 20 fitur teratas, dan menyimpannya ke dalam file CSV untuk analisis interpretabilitas model lebih lanjut.

- Visualisasi (*Grid Plots* & *Feature Importance*)

Seluruh hasil eksperimen divisualisasikan melalui serangkaian grafik terorganisir dalam format grid (3x3 dan 2x2). Visualisasi mencakup distribusi akurasi, perbandingan korelasi, matriks kebingungan, hingga perbandingan distribusi metode kepercayaan *Improved* vs *Legacy* vs *Calibrated*. Selain itu, dibuat grafik batang khusus untuk 20 fitur terpenting untuk memvisualisasikan kontribusi fitur secara jelas.

- Penyimpanan Model dan Hasil

Sistem membuat direktori *rf_results* dan menyimpan semua aset secara permanen. Ini mencakup model *Random Forest* terbaik, model terkalibrasi, pemroses skala (*scaler*), serta berbagai *DataFrame* hasil eksperimen dalam format CSV. Selain itu, file .npz dibuat sebagai arsip lengkap yang menyimpan seluruh hasil, prediksi, dan metadata eksperimen untuk penggunaan di masa mendatang.

- *Summary* Akhir

Alur kerja ditutup dengan ringkasan perbaikan yang dicapai, detail metode kepercayaan terbaik (skor, korelasi, *gap*), dan hasil validasi akhir. Program menampilkan perbaikan performa pada tiap konfigurasi dan waktu eksekusi total sebelum mencetak pesan penyelesaian. Ini memberikan gambaran menyeluruh tentang keberhasilan eksperimen peningkatan kepercayaan model *Random Forest*.

- *Program Display Metrics* - Inisialisasi

Program kedua dimulai dengan mencetak *header* informasi dan menetapkan direktori hasil. Sistem memuat file CSV dan file NPZ yang disimpan sebelumnya untuk mengekstraksi metrik detail, ringkasan perbandingan, hasil validasi, prediksi, serta nama kelas. Validasi pemuatan dilakukan untuk memastikan seluruh data tersedia sebelum masuk ke proses penampilan metrik.

- *Loop Display* Metrik (Parameter, *Fold*, *Repetition*)

Program melakukan iterasi melalui empat konfigurasi parameter untuk menampilkan hasil secara mendalam. Metrik ditampilkan dalam beberapa tingkatan: rata-rata keseluruhan dari 100 eksperimen, statistik performa berdasarkan nomor *Fold* (1-10) lintas repetisi, rincian per eksperimen individu, serta statistik berdasarkan nomor repetisi. Hal ini memungkinkan analisis stabilitas model dari berbagai sudut pandang pembagian data.

- Ringkasan Statistik dan Ekspor Parameter

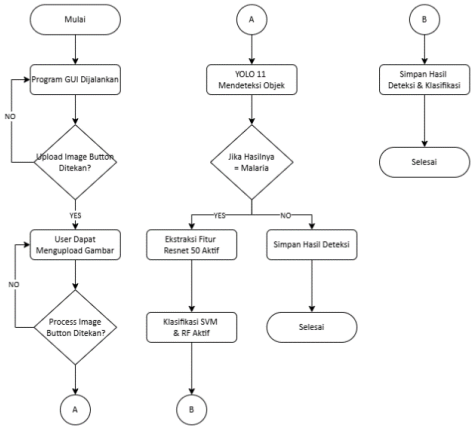
Sistem menyajikan tabel ringkasan statistik yang mencakup nilai *Mean, Std, Min, Max*, dan *Median* untuk seluruh metrik utama. Hasil detail untuk parameter yang sedang diproses kemudian diekspor ke dalam file CSV tersendiri. Proses ini diulang untuk setiap konfigurasi hingga semua data parameter ditampilkan dan disimpan kembali secara terorganisir.

- *Ranking*, Validasi Final, dan Penyelesaian

Setelah semua parameter ditampilkan, program menyajikan perbandingan lintas parameter dan memberikan peringkat berdasarkan akurasi rata-rata. Metrik validasi final dari parameter terbaik ditampilkan dengan detail per kelas melalui laporan klasifikasi lengkap. Program diakhiri dengan konfirmasi penyelesaian penampilan metrik dan daftar file yang telah berhasil diekspor.

3.4. Integrasi dan Validasi Sistem

Tahap akhir penelitian melibatkan integrasi seluruh komponen model ke dalam sistem fungsional yang koheren, sehingga Gambar 19 menyajikan *flowchart* integrasi dan validasi sistem yang menggambarkan alur kerja sistem dari input citra, deteksi objek menggunakan YOLO 11, percabangan logika berdasarkan hasil deteksi, hingga klasifikasi spesies malaria dan penyimpanan hasil analisis.



Gambar 19. *Flowchart* Integrasi dan Validasi Sistem

Fase terakhir penelitian ini adalah mengintegrasikan model-model yang telah dikembangkan ke dalam satu sistem fungsional dan melakukan validasi menyeluruh. Tujuan utama fase ini adalah untuk menguji alur kerja sistem dari awal hingga akhir dan menganalisis performa komparatif antara SVM dan RF dalam skenario penggunaan nyata. Alur implementasi dan pengujian sistem ini digambarkan pada Gambar 19. Sistem dirancang dengan sebuah antarmuka grafis (GUI) untuk memfasilitasi interaksi pengguna. Sebagaimana diilustrasikan pada flowchart, alur kerja sistem beroperasi melalui tahapan-tahapan berikut:

1. Inisiasi oleh Pengguna

   Proses dimulai ketika pengguna menjalankan program GUI, memilih sebuah citra darah untuk dianalisis, dan menekan tombol proses.

2. Deteksi Objek

   Sistem secara otomatis mengaktifkan model YOLO 11 untuk melakukan deteksi objek pada citra yang diunggah. Tahap ini bertujuan untuk mengidentifikasi kategori penyakit secara umum (Malaria, *Sickle-shaped red blood*, atau *Trypomastigote*)

3. Percabangan Logika

   Sistem kemudian mengevaluasi hasil deteksi dari YOLO 11. Terdapat sebuah proses keputusan yang akan menentukan alur kerja selanjutnya.

4. Alur Kerja Non-Malaria

   Jika hasil deteksi bukan Malaria, maka alur kerja klasifikasi tidak akan dijalankan. Sistem akan langsung menyimpan hasil deteksi awal dan proses dianggap selesai.

5. Alur Kerja Malaria

Jika hasil deteksi adalah Malaria, sistem akan secara otomatis mengaktifkan alur kerja sekunder. Proses ini diawali dengan ekstraksi fitur menggunakan ResNet50, yang kemudian dilanjutkan dengan klasifikasi spesies malaria menggunakan model SVM dan RF.

6. Penyimpanan Hasil Lengkap

Setelah proses klasifikasi selesai, sistem menyimpan hasil analisis yang komprehensif, mencakup hasil deteksi Malaria dan hasil klasifikasi spesiesnya. Proses kemudian berakhir.

3.5. Analisis Komparatif dan Penarikan Kesimpulan

Untuk memvalidasi kinerja dari setiap komponen dalam alur kerja tersebut, dilakukan pengujian kuantitatif menggunakan metrik evaluasi yang spesifik. Pada tahap deteksi objek, kinerja model YOLO 11 diukur menggunakan *Intersection over Union* (IoU) dan *mean Average Precision* (mAP).

Intersection over Union (IoU) digunakan untuk mengukur tingkat tumpang tindih antara kotak prediksi dan kotak *ground truth*, yang dihitung dengan persamaan sebagai berikut:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} = \frac{A \cap B}{A \cup B} \tag{17}$$

di mana:

- $A \cap B$ adalah luas irisan antara *bounding box* prediksi dan *ground truth*,
- $A \cup B$ adalah luas gabungan antara *bounding box* prediksi dan *ground truth* secara keseluruhan.

Sementara itu, *mean Average Precision* (mAP) digunakan untuk mengevaluasi keseluruhan kinerja model deteksi dalam berbagai ambang batas atau *Threshold*, dengan rumus sebagai berikut:

$$\text{mAP} = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{18}$$

di mana:

- n adalah jumlah kelas,
- AP_i adalah nilai average *Precision* untuk kelas ke i.

Selanjutnya, pada tahap klasifikasi jenis penyakit malaria, algoritma *Support Vector Machine* (SVM) dan *Random Forest* diterapkan pada citra yang telah melalui proses deteksi, dan untuk mengevaluasi performa klasifikasinya digunakan beberapa metrik evaluasi, yaitu akurasi, presisi, *Recall*, dan *F1-Score*, yang masing-masing dihitung menggunakan rumus sebagai berikut:

1. Akurasi

Mengukur seberapa banyak prediksi yang benar dibandingkan dengan keseluruhan prediksi, dihitung dengan:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

2. Presisi

Mengukur proporsi prediksi positif yang benar-benar positif, dihitung dengan:

$$Precision = \frac{TP}{TP + FP} \quad (20)$$

3. *Recall*

Mengukur seberapa banyak data positif yang berhasil dikenali dengan benar oleh model, dihitung dengan:

$$Recall = \frac{TP}{TP + FN} \quad (21)$$

4. *F1-Score*

Merupakan rata-rata harmonis dari presisi dan *Recall*, memberikan gambaran seimbang antara keduanya, dihitung dengan:

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (22)$$

3.5.1. Pengujian Sistem dengan Model YOLO 11

Pengujian model dilakukan untuk mengevaluasi pengaruh variasi *Learning rate* dan tipe arsitektur YOLO terhadap performa deteksi citra darah manusia yang mengandung parasit malaria. Model yang digunakan dalam penelitian ini adalah YOLO versi 11, yang dikenal memiliki peningkatan efisiensi pada tahap ekstraksi fitur dan deteksi objek melalui mekanisme *Cross-Stage Partial with Self-Attention (C2PSA)*. Tahapan pengujian dilakukan sebanyak empat kali eksperimen dengan parameter yang berbeda, sebagaimana ditunjukkan pada Tabel 3. Tujuan utama dari pengujian ini adalah untuk menentukan kombinasi *Learning rate* dan tipe model yang memberikan hasil deteksi paling optimal berdasarkan metrik evaluasi seperti *precision*, *recall*, *mAP50*, dan *mAP50–95*.

Pada pengujian pertama, digunakan model dasar yolo11l.pt dengan ukuran citra (*image size*) sebesar 1280 piksel, *batch size* sebesar 8, dan *initial Learning rate (lr0)* sebesar 0.001. Konfigurasi ini berfungsi sebagai acuan utama (baseline)

untuk membandingkan pengaruh penurunan maupun peningkatan nilai *Learning rate* terhadap stabilitas pelatihan dan akurasi deteksi.

Selanjutnya, pada pengujian kedua, nilai *Learning rate* diturunkan menjadi 0.0001 dengan parameter lainnya tetap sama (yolo11l.pt, *imgsz* = 1280, *batch* = 8). Penurunan nilai *Learning rate* ini bertujuan untuk mengamati efek pembelajaran yang lebih lambat namun stabil, sehingga dapat mengurangi risiko *overshooting* pada proses konvergensi model.

Pengujian ketiga menggunakan konfigurasi yang identik dengan dua pengujian sebelumnya, tetapi dengan peningkatan *Learning rate* menjadi 0.0015. Variasi ini dilakukan untuk menilai sejauh mana peningkatan *Learning rate* dapat mempercepat proses konvergensi tanpa menurunkan kualitas hasil deteksi akibat *overfitting* atau ketidakstabilan gradien.

Terakhir, pada pengujian keempat, digunakan arsitektur yang lebih kompleks, yaitu yolo11x.pt, dengan ukuran citra (*imgsz*) 1280 piksel, *batch size* 4, dan *Learning rate* 0.001. Model ini memiliki kapasitas parameter lebih besar dibandingkan yolo11l.pt, sehingga diharapkan dapat menangkap informasi spasial dan tekstural dengan lebih baik, meskipun membutuhkan sumber daya komputasi yang lebih tinggi.

Hasil dari keempat konfigurasi pengujian ini akan dibandingkan secara kuantitatif berdasarkan performa deteksi terhadap citra uji, dengan fokus pada metrik *precision*, *recall*, serta *mean Average Precision (mAP)*. Analisis hasil pengujian tersebut disajikan pada Bab IV untuk menentukan konfigurasi terbaik yang digunakan sebagai model deteksi utama dalam sistem klasifikasi malaria berbasis YOLO 11.

Tabel 3. Pengujian Model YOLO 11

NO	Model	Image Size (px)	Batch Size	*Learning rate* (lr0)	Keterangan
1	yolo11l.pt	1280	8	0.001	Konfigurasi awal untuk acuan perbandingan
2	yolo11l.pt	1280	8	0.0001	Mengamati efek *Learning rate* kecil terhadap stabilitas pelatihan
3	yolo11l.pt	1280	8	0.0015	Menilai dampak *Learning rate* besar terhadap konvergensi model

3.5.2. Pengujian Sistem dengan Algoritma SVM

Guna memperoleh gambaran performa dari metode klasifikasi berbasis pembelajaran terawasi, algoritma *Support Vector Machine* (SVM) digunakan untuk memproses data citra darah manusia hasil deteksi objek menggunakan model YOLO 11. SVM merupakan algoritma klasifikasi yang bekerja dengan membentuk sebuah *hyperplane* optimal pada ruang fitur berdimensi tinggi untuk memisahkan data ke dalam dua atau lebih kelas berdasarkan karakteristik tertentu. Dalam praktiknya, SVM tidak hanya membentuk garis atau bidang pemisah, tetapi juga memaksimalkan margin antara *hyperplane* dan titik-titik data terdekat dari masing-masing kelas, yang dikenal sebagai *support vectors*.

Apabila data tidak dapat dipisahkan secara *Linear*, SVM memanfaatkan teknik *kernel trick*, seperti *Radial Basis Function* (RBF) dan *Polynomial kernel*, untuk memetakan data ke dalam ruang fitur berdimensi lebih tinggi sehingga pemisahan *Linear* tetap dapat dilakukan. Selain itu, parameter regulasi (C) digunakan untuk mengontrol *trade-off* antara margin maksimum dan kesalahan klasifikasi, sehingga model tetap memiliki kemampuan generalisasi yang baik terhadap data baru.

Dalam penelitian ini, algoritma SVM diterapkan untuk mengklasifikasikan jenis penyakit malaria berdasarkan morfologi parasit yang telah terdeteksi pada citra darah. Fitur-fitur yang digunakan sebagai masukan model diperoleh melalui proses ekstraksi karakteristik dari hasil segmentasi objek oleh model YOLO 11, yang selanjutnya digunakan pada tahap pelatihan dan pengujian model SVM.

Pengujian dilakukan menggunakan beberapa konfigurasi kernel dan parameter SVM sebagaimana ditunjukkan pada Tabel 4. Seluruh konfigurasi dievaluasi menggunakan skema *k-fold cross-validation* sebanyak 10 fold dengan 10 kali repetisi (*10×10 cross-validation*) guna memperoleh estimasi performa yang stabil serta meminimalkan bias akibat pembagian data.

Tabel 4. Konfigurasi dan Skema Pengujian Algoritma SVM

NO	Model	Konfigurasi Kernel & Parameter	Skema & Validasi
1	SVM	Kernel *Linear*	10-Fold × 10 Repetisi
2	SVM	Kernel RBF, γ = scale	10-Fold × 10 Repetisi
3	SVM	Kernel *Polynomial*, Degree = 2	10-Fold × 10 Repetisi
4	SVM	Kernel RBF, C = 10	10-Fold × 10 Repetisi

Evaluasi performa model dilakukan menggunakan beberapa metrik, yaitu akurasi, presisi, *recall*, dan F1-*score*, yang secara kolektif digunakan untuk menilai efektivitas serta ketepatan model dalam mengklasifikasikan masing-masing jenis parasit malaria. Hasil evaluasi kuantitatif dari setiap konfigurasi SVM serta perbandingannya dengan algoritma klasifikasi lain, seperti *Random Forest*, disajikan dan dibahas secara rinci pada Bab 4.

3.5.3. Pengujian Sistem dengan Algoritma *Random Forest*

Pertimbangan terhadap kestabilan dan akurasi dalam proses klasifikasi menjadi dasar pemilihan algoritma *Random Forest* sebagai salah satu metode yang diuji dalam penelitian ini. *Random Forest* merupakan algoritma *ensemble learning* yang mengombinasikan sejumlah *decision tree* yang dibangun secara independen dan acak, baik dari sisi pemilihan data latih maupun subset fitur yang digunakan. Pendekatan ini bertujuan untuk meningkatkan akurasi prediksi serta mengurangi risiko *overfitting* yang umumnya terjadi pada penggunaan satu *decision tree* tunggal.

Setiap *decision tree* dalam *Random Forest* dibangun menggunakan metode *bootstrap sampling*, yaitu proses pengambilan sampel data latih secara acak dengan pengembalian. Dengan demikian, setiap pohon dilatih menggunakan subset data yang berbeda sehingga menghasilkan model-model yang beragam dan tidak saling berkorelasi secara kuat. Pada setiap node pohon, pemilihan atribut pemisah dilakukan berdasarkan ukuran *impurity* tertentu, seperti *Gini impurity* atau *information gain*, dengan mempertimbangkan hanya subset acak dari seluruh fitur yang tersedia. Mekanisme ini memungkinkan *Random Forest* untuk menangkap pola kompleks dalam data sekaligus menjaga kestabilan model.

Proses klasifikasi pada *Random Forest* dilakukan melalui mekanisme *majority voting*, di mana setiap *decision tree* memberikan satu prediksi kelas terhadap data uji. Kelas dengan jumlah suara terbanyak kemudian dipilih sebagai hasil klasifikasi akhir. Pendekatan ini membuat *Random Forest* lebih robust terhadap noise dan variasi data dibandingkan metode klasifikasi tunggal.

Dalam konteks penelitian ini, algoritma *Random Forest* diterapkan untuk mengklasifikasikan jenis penyakit malaria berdasarkan morfologi parasit yang terdeteksi pada citra darah manusia yang telah diproses menggunakan model YOLO 11. Fitur-fitur hasil ekstraksi dari citra darah digunakan sebagai masukan bagi setiap pohon dalam *Random Forest* untuk menentukan kelas jenis parasit malaria.

Pengujian *Random Forest* dilakukan dengan memodifikasi beberapa parameter utama secara terpisah, meliputi mekanisme *bootstrap*, jumlah pohon (*number of trees*), kedalaman maksimum pohon (*maximum depth*), serta kriteria pemisahan (*criterion*), sebagaimana ditunjukkan pada Tabel 5. Seluruh konfigurasi dievaluasi menggunakan skema *k-fold cross-validation* sebanyak 10 fold dengan 10 kali repetisi (*10×10 cross-validation*) guna memperoleh estimasi performa yang stabil serta mengurangi bias akibat pembagian data.

Tabel 5. Konfigurasi dan Skema Pengujian Algoritma *Random Forest*

NO	Model	Konfigurasi Kernel & Parameter	Skema & Validasi
1	*Random Forest*	bootstrap= True	10-Fold × 10 Repetisi

2	*Random Forest*	Max_depth= 15	10-Fold × 10 Repetisi
3	*Random Forest*	n_estimators= 100	10-Fold × 10 Repetisi
4	*Random Forest*	Criterion= Gini	10-Fold × 10 Repetisi

Evaluasi performa model *Random Forest* dilakukan menggunakan sejumlah metrik, yaitu akurasi, presisi, recall, dan F1-*score*, yang secara kolektif digunakan untuk menilai efektivitas dan ketepatan model dalam mengklasifikasikan masing-masing jenis parasit malaria. Hasil evaluasi kuantitatif dari setiap konfigurasi *Random Forest* serta perbandingannya dengan algoritma *Support Vector Machine* (SVM) disajikan dan dibahas secara rinci pada Bab 4.

3.6. Alat dan Bahan

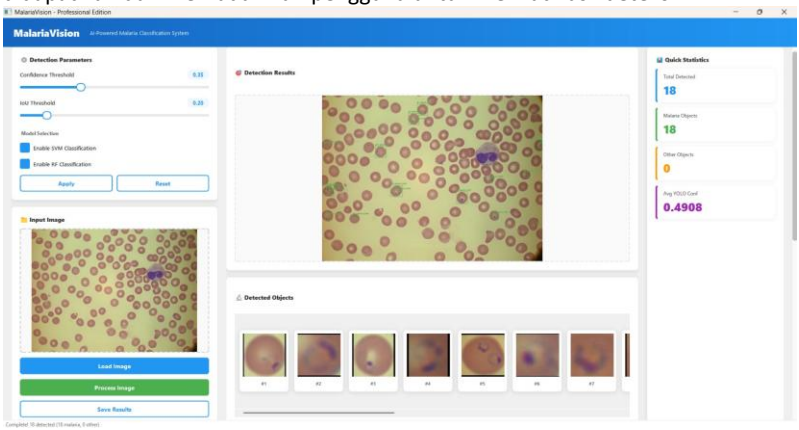
Tabel 6. Estimasi biaya

No.	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Keterangan[1]
1	Laptop	-	2	-	Milik Pribadi
2	Google Colab	142.500	1	142.500	Milik Pribadi
	Total			142.500	Milik Pribadi

Bab 4. Hasil dan Pembahasan

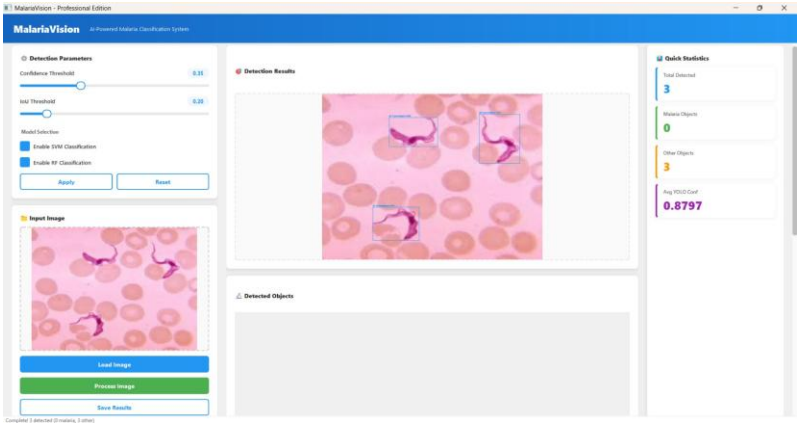
4.1. Tampilan GUI

Pada tahap ini, kami telah merancang GUI untuk menampilkan hasil yang didapat kan dan memudah kan pengguna untuk melihat hasil deteksi.



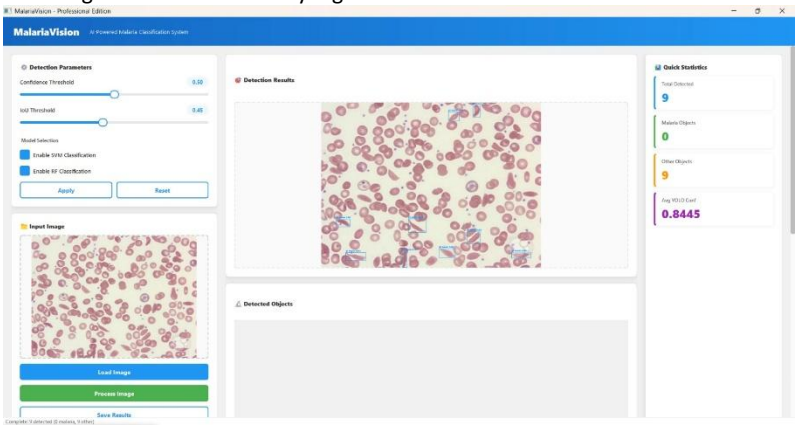
Gambar 20. Tampilan Hasil Deteksi Malaria

Antarmuka GUI pada Gambar 20 menampilkan hasil deteksi malaria secara komprehensif dengan visualisasi bounding box, klasifikasi spesies, serta metrik evaluasi yang terintegrasi, memfasilitasi interpretasi hasil diagnosis secara intuitif bagi pengguna klinis. Untuk mendemonstrasikan kemampuan sistem dalam membedakan berbagai jenis penyakit darah, Gambar 20 menyajikan hasil deteksi pada kelas yang berbeda, yaitu *Trypomastigote*, dengan antarmuka GUI yang mempertahankan konsistensi visualisasi dan format output diagnosis.



Gambar 21. Tampilan Hasil Deteksi *Trypomastigote*

Tampilan GUI pada Gambar 21 menunjukkan kemampuan sistem dalam mendeteksi sel *Trypomastigote* dengan visualisasi yang jelas, mengonfirmasi bahwa sistem tidak hanya spesifik untuk malaria tetapi juga dapat membedakan penyakit darah lainnya dengan akurat. Melengkapi evaluasi komprehensif terhadap kapabilitas sistem dalam mengklasifikasikan seluruh spektrum penyakit darah yang ditargetkan, Gambar 21 menampilkan hasil deteksi untuk kasus Anemia, mengonfirmasi kemampuan generalisasi model dalam menangani variasi morfologi sel darah abnormal yang berbeda.



Gambar 22. Tampilan Hasil Deteksi Anemia

Gambar 22 memperlihatkan hasil deteksi sel anemia melalui antarmuka GUI yang *user-friendly*, mendemonstrasikan kemampuan sistem dalam mengidentifikasi dan mengklasifikasikan sel darah abnormal selain parasit malaria, memperluas utilitas diagnostik sistem secara keseluruhan.

4.1.1. Fitur Utama Deteksi dan Klasifikasi

1. *Pipeline* AI Multi-Model

Pemanfaatan alur kerja Kecerdasan Buatan (*AI Pipeline*) Multi-Model ini dirancang untuk diagnosis komprehensif penyakit darah, dimulai dari tahap YOLO Detection yang bertugas mendeteksi sel darah yang relevan ke dalam tiga kategori utama: Malaria, Anemia, dan *Trypomastigote*. Setelah deteksi, objek yang terdeteksi kemudian diproses melalui dua metode klasifikasi untuk mengidentifikasi spesies parasit malaria: SVM *Classification* yang mengklasifikasikan ke dalam empat spesies (*Falciparum*, *Malariae*, *Ovale*, *Vivax*) dan *Random Forest Classification* yang berfungsi sebagai klasifikasi alternatif untuk validasi silang dan peningkatan reliabilitas sistem, di mana kedua model klasifikasi tersebut memanfaatkan fitur *deep learning* yang diekstraksi dari model ResNet50 *Feature Extraction* yang telah dilatih sebelumnya (*pre-trained*) pada ImageNet.

2. Normalisasi Skor Kepercayaan

Sistem ini menerapkan Normalisasi *Confidence score* dengan menggunakan metode Min-Max *Normalization* untuk menyesuaikan *confidence score* dari klasifikasi SVM ke rentang 0-1, demi mencapai konsistensi dengan hasil yang dihasilkan oleh model *Random Forest* (RF). Proses ini didukung oleh *Tracking Decision Scores*, di mana sistem mengumpulkan semua *decision Scores* yang dihasilkan pada alur awal (*first pass*) untuk memastikan perhitungan normalisasi yang dilakukan dapat mencapai akurasi maksimal.

4.1.2. Fitur Input dan Processing

1. Input Gambar Fleksibel

Antarmuka pengguna dirancang untuk menyediakan Input Gambar Fleksibel, memungkinkan pengguna untuk memasukkan citra yang akan didiagnosis melalui dua cara mudah: secara langsung menggunakan *Drag-and-Drop Interface* atau dengan memilih file melalui File Browser yang diakses melalui tombol "*Load Image*" pada dialog sistem. Sistem ini memastikan kompatibilitas luas karena mendukung berbagai format gambar populer, termasuk PNG, JPG, JPEG, BMP, dan TIFF.

2. Penyesuaian Parameter

Seluruh penyesuaian parameter sistem dikelompokkan dalam Parameter *Adjustment* yang menyediakan kontrol penuh kepada pengguna atas proses

deteksi dan klasifikasi. Pengguna dapat secara dinamis mengatur ambang batas *confidence* YOLO menggunakan *Confidence Threshold Slider* (antara 0.10 hingga 0.90) dan menyesuaikan IoU *Threshold Slider* untuk *Non-Maximum Suppression* (NMS) dengan rentang yang sama (0.10 hingga 0.90). Selain itu, terdapat Model *Selection Checkboxes* yang memungkinkan pengguna untuk mengaktifkan atau menonaktifkan model klasifikasi SVM dan *Random Forest* sesuai kebutuhan analisis. Semua perubahan pada pengaturan ini didukung oleh *Value Display*, yang menampilkan nilai parameter secara dinamis, serta tombol *Reset to Default* untuk mengembalikan semua konfigurasi ke pengaturan awal dengan cepat.

4.1.3. Fitur Visualisasi

1. Tampilan Hasil Deteksi

Tampilan hasil deteksi (*Detection Results Display*) memberikan feedback visual yang jelas kepada pengguna, dimulai dengan *Annotated Image Output* yang menyajikan gambar input yang diperkaya dengan *bounding box* dan *label* di sekitar setiap objek yang terdeteksi. Untuk memudahkan identifikasi, *Color-Coded Boxes* digunakan, di mana *bounding box* berwarna hijau dikhususkan untuk objek *Malaria* dan warna oranye digunakan untuk objek lain (*Anemia* dan *Trypomastigote*). Selain itu, setiap deteksi dilengkapi dengan *Confidence Labels* yang menampilkan secara rinci nomor objek, kelas prediksi, dan *confidence score* model.

2. Galeri Hasil Potongan Objek

Sistem ini menyediakan Cropped Objects Gallery yang berfungsi sebagai galeri horizontal yang dapat digulir (Horizontal *Scrollable Gallery*), menampilkan semua objek yang berhasil dideteksi oleh YOLO. Setiap objek ditampilkan dalam format *Object Cards* yang terstruktur, di mana setiap *card* memuat gambar objek yang telah dipotong (*cropped*) dengan *padding*, nomor identifikasi objek, dan distandarisasi dalam Ukuran standar 224 x 224 piksel untuk memudahkan proses analisis lebih lanjut.

3. Panel Statistik Cepat

Panel Quick Statistics Panel dirancang untuk memberikan ringkasan kuantitatif instan mengenai hasil deteksi. Panel ini menampilkan *Total Detected* (jumlah keseluruhan objek yang berhasil diidentifikasi), rincian spesifik mengenai Malaria *Objects* (jumlah objek yang diklasifikasikan sebagai malaria) dan *Other Objects* (jumlah objek non-malaria, yaitu Anemia dan *Trypomastigote*). Selain itu, panel ini juga menyajikan *Average* YOLO *Confidence* yang merupakan rata-rata *confidence score* dari semua deteksi yang dilakukan oleh model YOLO.

4.1.4 Fitur Analitik Multi-Tab

1. Tab Statistik Terperinci

Panel Tab *Detailed Statistics* berfungsi sebagai pusat analisis mendalam yang menyajikan tiga komponen utama. Pertama, *SVM vs RF Comparison Table* menampilkan perbandingan statistik antara kedua model klasifikasi, meliputi *Average Confidence*, *Standard Deviation*, dan *Min/Max Confidence*, dengan nilai yang dikodekan warna (biru untuk SVM dan ungu untuk RF). Kedua, *Agreement Analysis* memberikan metrik validasi silang dengan menyajikan jumlah *agreement* (kesesuaian) dan jumlah *disagreement* (ketidaksesuaian) antara prediksi SVM dan RF, dihitung sebagai *Agreement Rate* (persentase), serta menampilkan *Average Confidence Gap* (rata-rata selisih *confidence*). Ketiga, *Species Distribution* memberikan rincian kuantitatif prediksi dengan menyajikan *breakdown* prediksi per spesies untuk kedua model, baik untuk SVM maupun RF.

2. Tab Grafik Kepercayaan

Panel Tab *Confidence Charts* berfungsi untuk memvisualisasikan performa dan distribusi *confidence score* dari model klasifikasi. Bagian ini menampilkan *Bar Chart Comparison*, yang menyajikan perbandingan *confidence score* antara SVM vs RF per objek menggunakan visualisasi *dual-bar*. Setiap nilai *confidence* dilabeli di atas *bar* yang bersangkutan, dan latar belakang ber-*grid* ditambahkan untuk mempermudah pembacaan dan perbandingan. Selain itu, panel ini memuat *Distribution Histogram*, yang menunjukkan distribusi *confidence scores* dalam *bins* (interval), menggunakan visualisasi *overlay* untuk membandingkan kedua model. Histogram ini juga dilengkapi dengan *mean lines* untuk SVM dan RF, serta ringkasan *statistical summary* untuk memberikan pemahaman statistik yang lebih mendalam mengenai sebaran *confidence*.

3. Tab Hasil Per-Objek

Panel Tab *Per-Object Results* menyediakan hasil analisis mendalam dan individual untuk setiap objek yang terdeteksi dalam format *Detailed Result Cards*. Setiap *card* menampilkan serangkaian data yang komprehensif, meliputi Object ID dan YOLO class awal, YOLO *confidence score*, hasil prediksi SVM beserta *confidence score* yang telah dinormalisasi (0-1), hasil prediksi dan *confidence* dari RF, serta status Agreement antar kedua model (ditandai dengan ✓ *Agree* atau ✗ *Disagree*). Seluruh kartu hasil ini disajikan dalam *Scrollable List* yang mendukung penjelajahan dan analisis sejumlah besar objek terdeteksi dengan guliran vertikal.

4. Tab Analisis Fitur

Panel Tab *Feature Analysis* memberikan pandangan mendalam terhadap proses *ResNet50 Feature Extraction Visualization*, menampilkan seluruh proses secara komprehensif dari awal hingga akhir (*Comprehensive Pipeline View*). Visualisasi dimulai dari *Original Image* (input 224 × 224 × 3 piksel), dilanjutkan ke output rinci *Conv1 Layer* dalam format *grayscale*, *heatmap*, *overlay*, dan *channel grid*. Kemudian, panel menyajikan *Layer 1–4 Visualizations* yang menunjukkan evolusi *feature maps* di setiap blok residual ResNet. Puncaknya, *Final Feature Vector* (2048 dimensi) direpresentasikan dalam format *2D heatmap* (32 × 64), *line*

plot dengan *fill*, dan ringkasan *statistical summary* (*mean*, *std*, *min*, *max*, *sparsity*). Proses ini didukung oleh fitur *Auto-Generation* yang secara otomatis menampilkan visualisasi fitur untuk objek malaria yang dipilih secara acak segera setelah pemrosesan selesai.

4.1.5. Fitur Pengalaman Pengguna

1. Status Pemrosesan

Sistem ini dilengkapi dengan *Loading States* yang informatif untuk memberikan *feedback* kepada pengguna selama pemrosesan data. Ketika sistem sedang memproses, sebuah Loading *Overlay full-screen* akan muncul, disertai dengan *Progress Bar* berupa indikator progres animasi untuk menunjukkan aktivitas yang sedang berlangsung. Selain itu, Status *Messages* ditampilkan secara dinamis untuk memberikan pesan informatif yang menjelaskan tahapan proses yang sedang dijalankan oleh sistem.

2. Bilah Status

Status Bar pada sistem dirancang untuk memberikan *feedback* berkelanjutan dan transparan kepada pengguna. Fitur ini menyediakan Updates yang menampilkan informasi status terkini, memberikan *Processing Messages* sebagai *feedback* spesifik untuk setiap tahap *processing* yang sedang berlangsung, dan menampilkan *Error Notifications* berupa pesan *error* yang jelas dan informatif saat terjadi masalah, sehingga memudahkan pengguna untuk memahami kondisi sistem.

3. Desain Responsif

Antarmuka pengguna dikembangkan dengan fokus pada *Responsive Design* untuk memastikan pengalaman yang optimal bagi pengguna. Sistem mendukung *Scrollable Content* dengan *scroll* vertikal untuk menampung konten yang panjang, sambil memastikan tata letak tetap utuh melalui *Minimum Size Protection* dengan ukuran jendela minimum 1500 x 850 piksel. Seluruh elemen ditampilkan menggunakan *Dynamic Resizing*, di mana konten menyesuaikan diri secara mulus dengan perubahan ukuran jendela. Terakhir, semua informasi diorganisir dalam *Clean Card-Based Layout* yang rapi dan terstruktur, menggunakan *shadow* untuk memisahkan setiap bagian konten secara visual.

4.1.6. Fitur Ekspor dan Pelaporan

1. Simpan Hasil

Fungsi Save Results memungkinkan pengguna untuk mengekspor semua hasil analisis dalam format yang beragam dan komprehensif. Pengguna dapat melakukan Annotated Image Export untuk menyimpan gambar hasil deteksi yang telah dianotasi (*bounding box* dan label) dalam format JPG. Selain itu, sistem

menyediakan Text Report Generation berupa laporan komprehensif dalam format .txt, yang mencakup Metadata (*timestamp* dan nama *file*), Detection parameters yang digunakan, Summary statistics, hasil klasifikasi rinci Per-object classification results, dan Informasi normalisasi *confidence*. Terakhir, untuk kemudahan analisis data lanjutan, disediakan CSV Data Export yang menghasilkan data terstruktur (.csv) dengan kolom rinci: Object_ID, YOLO_Class dan YOLO_Conf, prediksi SVM_Pred dan SVM_Conf_*Normalized*, prediksi RF_Pred dan RF_Conf, serta Agreement status antar kedua model.

2. Output Bertanda Waktu

Sistem ini menjamin konsistensi dan kemudahan pelacakan data melalui Timestamped Outputs, di mana semua *output file* yang dihasilkan secara otomatis diberi timestamp yang spesifik. Hal ini diimplementasikan dengan menggunakan Naming convention yang konsisten untuk semua *file* yang diekspor, dan semua *file* tersebut disimpan dalam Organized output directory (direktori keluaran yang terorganisir), sehingga memudahkan pengguna untuk mengelola riwayat analisis.

4.1.7. Fitur Teknis

1. Pemrosesan Citra

Proses awal pemrosesan citra (Image Processing) dilakukan secara sistematis, dimulai dengan Resize with Padding yang bertujuan mempertahankan rasio aspek citra asli dengan menambahkan *padding* berwarna hitam pada tepi citra. Setelah itu, dilakukan Color Space Conversion untuk mengubah ruang warna dari BGR ke RGB, yang merupakan format yang diperlukan untuk pemrosesan lebih lanjut. Terakhir, dilakukan Image Normalization menggunakan standar ImageNet, yang merupakan langkah krusial sebelum citra dimasukkan ke dalam model ekstraksi fitur ResNet50.

2. Penanganan Error

Sistem ini memastikan keandalan operasional melalui manajemen kesalahan (Error Handling) yang kuat. Implementasi Try-Catch Blocks pada setiap tahap *processing* memungkinkan sistem untuk menangani kesalahan secara terstruktur. Jika terjadi kegagalan, sistem akan menampilkan User-Friendly Messages berupa dialog *error* yang informatif dan mudah dipahami pengguna. Selain itu, sistem mendukung Graceful Degradation, di mana sistem utama tetap berfungsi dan terus berjalan meskipun salah satu model klasifikasi gagal dimuat (*load*), sehingga meminimalkan gangguan pada alur kerja pengguna.

3. Optimasi Performa

Untuk memastikan kecepatan dan efisiensi analisis, sistem menerapkan Performance Optimization yang mencakup berbagai aspek teknis. Akselerasi pemrosesan diaktifkan melalui GPU Acceleration, dengan deteksi otomatis

dukungan CUDA untuk *framework* PyTorch yang digunakan. Efisiensi ekstraksi fitur ditingkatkan melalui Batch Processing, yang memungkinkan pemrosesan data dalam jumlah besar secara efisien. Selain itu, Memory Management diterapkan melalui prosedur *cleanup* yang tepat untuk objek-objek Qt, sehingga mengurangi *overhead* memori dan mencegah kebocoran memori (*memory leaks*).

4.1.8. Fitur Desain UI/UX

1. Desain Modern dan Bersih

Antarmuka sistem ini dirancang menggunakan prinsip Modern Clean Design untuk memastikan pengalaman pengguna yang intuitif dan profesional. Estetika yang diterapkan bersifat Minimalist Aesthetic, menekankan desain yang bersih dan fungsional tanpa *over-styling*. Efek visual didukung oleh penggunaan Subtle Shadows (*drop shadow*) pada elemen-elemen kunci untuk memberikan persepsi kedalaman (*depth perception*). Skema warna (Color Scheme) yang digunakan telah ditetapkan secara spesifik untuk memberikan konsistensi visual dan fungsional, meliputi Primary (Blue: #2196F3), Success (Green: #4CAF50), Warning (Orange: #FF9800), dan Secondary (Purple: #9C27B0).

2. Tipografi

Elemen visual pada sistem didukung oleh implementasi Typography yang berfokus pada kejelasan dan kemudahan membaca. Sistem menggunakan Segoe UI Font sebagai *font* utama karena dikenal sangat *readable* sebagai *font* sistem. Penerapan Hierarchical Sizing memastikan adanya perbedaan ukuran *font* yang jelas untuk membangun hierarki visual, membedakan antara judul, subjudul, dan teks utama. Selain itu, Weight Variations (variasi ketebalan *font*) dimanfaatkan, khususnya penggunaan huruf Bold, untuk memberikan penekanan yang tepat pada informasi kunci (*emphasis*).

3. Elemen Interaktif

Antarmuka sistem ini ditingkatkan dengan berbagai Interactive Elements untuk memberikan *feedback* visual dan meningkatkan pengalaman pengguna. Elemen-elemen ini mencakup penggunaan Hover Effects yang memberikan umpan balik visual saat kursor mengarah pada elemen interaktif. Kemudian, Cursor Changes memastikan bahwa pengguna mendapatkan petunjuk visual yang jelas (mengubah kursor menjadi *pointer*) untuk elemen-elemen yang dapat diklik (*clickable elements*). Selain itu, semua tombol (Button) dikonfigurasi dengan Button States yang jelas dan informatif, meliputi status *disabled*, *normal*, *hover*, dan *pressed*.

4.2. Hasil Deteksi Objek pada Citra Darah (YOLO 11)

Tahap awal penelitian ini menerapkan algoritma YOLO 11 untuk mendeteksi sel darah yang terinfeksi *Plasmodium* pada citra mikroskopis. Citra masukan dilabeli menjadi tiga kelas utama, yaitu Anemia, Malaria, dan *Trypomastigote*.

Model dilatih dengan pengaturan maksimum 200 *epoch*, resolusi citra 1280 x 1280 piksel, dan *batch size* 8. Namun, berkat penerapan *early stopping*, *epoch* pelatihan terbaik yang didapatkan bervariasi di setiap eksperimen, menjamin model tidak mengalami *overfitting* pada data latih. Selanjutnya, untuk mengevaluasi pengaruh variasi *Learning rate* terhadap performa deteksi dan meninjau perbedaan *epoch* terbaik, dilakukan tiga eksperimen pelatihan model dengan nilai *Learning rate* yang berbeda. Hasil eksperimen tersebut akan dibahas lebih rinci pada subbab-subbab berikut.

4.2.1 YOLO11l dengan *Learning rate* (0.001)

Model YOLO 11l dilatih menggunakan 127 citra dengan total 1.071 instance dari kelas Anemia, Malaria, dan *Trypomastigote*, dengan *Learning rate* awal 0.001. Arsitektur model terdiri dari 190 lapisan dengan 25,28 juta parameter yang dioptimasi untuk mendeteksi objek berukuran kecil pada citra mikroskopik darah. Performa model berdasarkan metrik precision, recall, dan mAP disajikan pada Tabel 7.

Tabel 7. Metrik Kinerja Deteksi YOLO 11 (LR = 0.001)

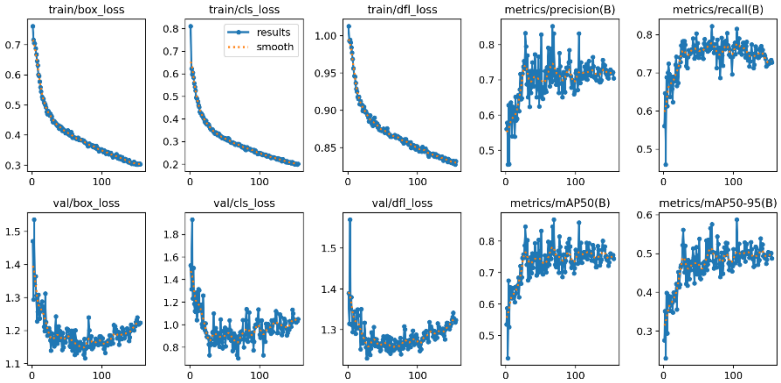
Class	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
Anemia	0.949	0.914	0.968	0.665
Malaria	0.739	0.830	0.835	0.651
Trypomastigote	0.814	0.709	0.776	0.449
Rata-Rata Makro	0.834	0.818	0.859	0.588

Hasil evaluasi menunjukkan performa deteksi yang sangat baik dengan mAP@0.5 mencapai 0.859 dan mAP@0.5:0.95 sebesar 0.588, mengindikasikan bahwa model mampu mendeteksi objek dengan akurasi tinggi. Rata-rata presisi sebesar 0.834 menunjukkan tingkat *false positive* yang rendah, artinya sebagian besar prediksi model adalah benar. Sementara itu, recall sebesar 0.818 mengindikasikan bahwa model berhasil mendeteksi sekitar 82% dari seluruh objek yang sebenarnya ada dalam citra.

Analisis per kelas menunjukkan variasi performa yang signifikan. Kelas Anemia mencapai presisi tertinggi (0.949), *recall* (0.914), dan mAP@0.5 (0.968), diikuti oleh kelas Malaria dengan presisi 0.739, recall 0.830, dan mAP@0.5 sebesar 0.835. Sebaliknya, kelas *Trypomastigote* menunjukkan performa terendah dengan presisi 0.814, *recall* 0.709, mAP@0.5 sebesar 0.776, dan mAP@0.5:0.95 hanya 0.449,

mengindikasikan tantangan deteksi yang lebih besar pada kelas ini, terutama pada *threshold* IoU yang lebih ketat. Dari aspek efisiensi komputasi, model menunjukkan waktu inferensi 27.2 ms dengan total waktu pemrosesan (termasuk preprocessing 0.5 ms dan postprocessing 3.5 ms) sebesar 31.4 ms per citra, setara dengan throughput sekitar 32 frame per detik (fps). Performa ini menunjukkan efisiensi komputasi yang memadai untuk implementasi sistem deteksi otomatis dalam setting laboratorium klinis, di mana kecepatan analisis menjadi faktor penting dalam diagnosis cepat penyakit infeksi darah.

Proses konvergensi model selama pelatihan disajikan pada Gambar 23, yang memberikan wawasan mendalam tentang dinamika pembelajaran dan validasi efektivitas strategi early stopping yang diterapkan.



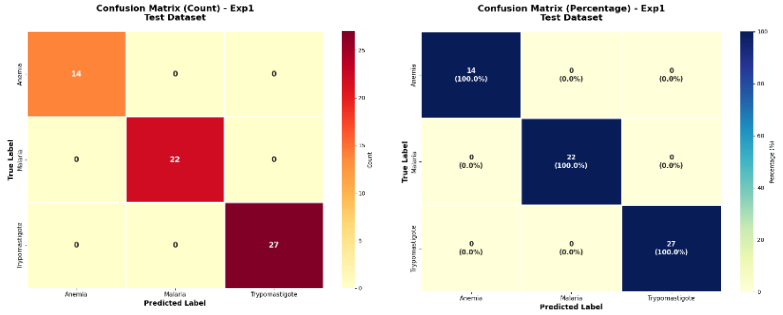
Gambar 23. Grafik Hasil Pelatihan YOLO 11l (LR = 0.001)

Dinamika pelatihan model pada Gambar 23 menunjukkan stabilitas pembelajaran yang optimal. Komponen *loss function* pada data latih (box_loss, cls_loss, dan dfl_loss) mengalami penurunan monoton yang mengindikasikan konvergensi fitur berjalan lancar tanpa kendala *gradien*. *Loss function* turun secara konsisten dari nilai awal box_loss=0.76, cls_loss=0.81, dfl_loss=1.01 pada *epoch* 1 hingga mencapai box_loss≈0.30, cls_loss≈0.20, dfl_loss≈0.83 pada *epoch* akhir. Sementara itu, kurva loss validasi memperlihatkan penurunan signifikan pada fase awal, dari nilai box_loss≈1.55 turun menjadi stabil di rentang 1.15-1.25 setelah *epoch* 50, di mana ketiadaan tren kenaikan error memastikan bahwa model tidak mengalami *overfitting* selama *proses fine-tuning*.

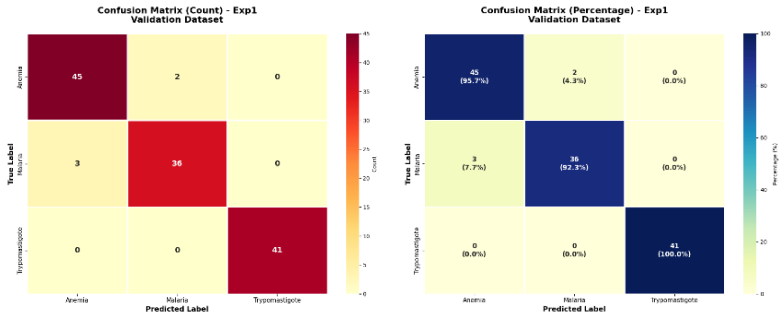
Selanjutnya, metrik presisi dan *recall* menunjukkan tren peningkatan yang jelas dari 0.56 dan 0.56 pada epoch 1 menjadi stabil di rentang 0.70-0.85 pada fase pertengahan hingga akhir pelatihan. Performa puncak dicapai pada *epoch* 105 dengan mAP@0.5 sebesar 0.859 dan mAP@0.5:0.95 sebesar 0.588. Setelah *epoch* 105, tidak terdapat peningkatan signifikan pada metrik evaluasi, dengan nilai

mAP@0.5 berfluktuasi di rentang 0.72-0.80 dan mAP@0.5:0.95 di rentang 0.45-0.54. Stabilitas ini membuktikan bahwa model berhasil menjaga keseimbangan antara kemampuan deteksi objek dan ketepatan prediksi tanpa mengalami degradasi performa, sebagaimana dikonfirmasi oleh mekanisme *early stopping* yang menghentikan *training* pada *epoch* 155 setelah tidak ada peningkatan selama 50 *epoch* sejak *epoch* 105 dengan total durasi pelatihan adalah 16.027 jam.

Untuk meninjau akurasi klasifikasi dan potensi kesalahan antar kelas (misklasifikasi), gambar 24 menyajikan *Confusion matrix* untuk *Dataset Uji* dan gambar 25 menyajikan *Confusion matrix Dataset Validasi*.



Gambar 24. *Confusion matrix* pada Dataset Uji



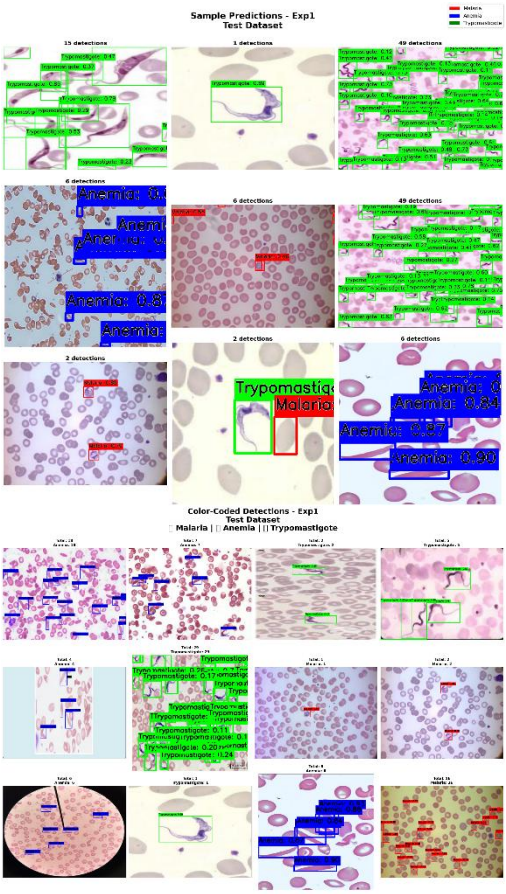
Gambar 25. *Confusion matrix* pada Dataset Validasi

Pada dataset uji (63 citra), model menunjukkan performa klasifikasi yang sempurna dengan akurasi 100% untuk semua kelas. Sebanyak 14 instance Anemia, 22 instance Malaria, dan 27 instance *Trypomastigote* berhasil diklasifikasikan dengan benar tanpa ada satupun kesalahan prediksi (*false positive* maupun false negative). Hal ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang sangat baik pada data yang belum pernah dilihat sebelumnya.

Pada dataset validasi (127 citra dengan 126 instance), model menunjukkan performa yang sedikit berbeda dengan akurasi keseluruhan sebesar 96,8%. Dari total 47 instance Anemia, 45 (95,7%) diklasifikasikan dengan benar, sementara 2 instance (4,3%) salah diprediksi sebagai Malaria. Untuk kelas Malaria, dari 39 instance, 36 (92,3%) diprediksi dengan tepat, namun 3 instance (7,7%) salah diklasifikasikan sebagai Anemia. Sebaliknya, kelas *Trypomastigote* menunjukkan performa sempurna dengan semua 41 instance (100%) berhasil diidentifikasi dengan benar tanpa kesalahan.

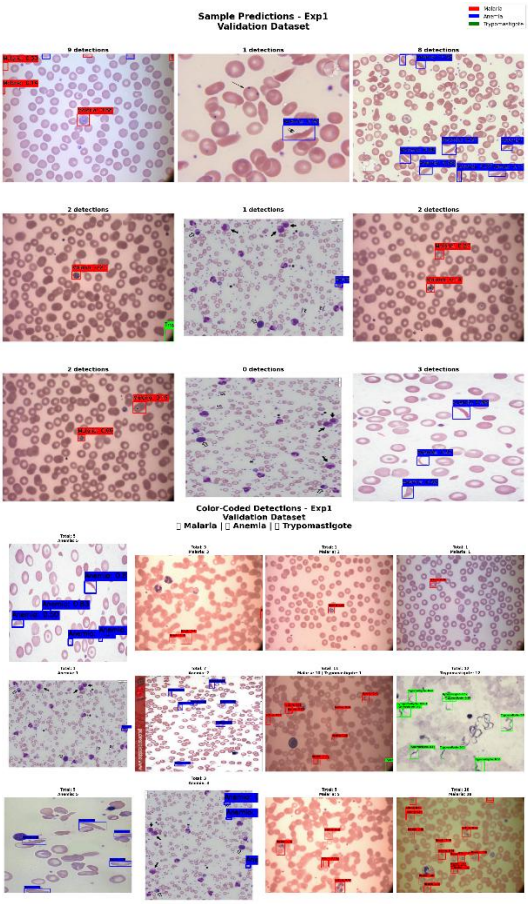
Pola misklasifikasi yang terjadi, yaitu kebingungan antara kelas Anemia dan Malaria, mengindikasikan adanya tantangan model dalam membedakan kedua kelas tersebut dalam beberapa kasus tertentu. Hal ini menarik mengingat secara visual terdapat perbedaan yang jelas antara keduanya: sel darah merah dengan anemia memiliki bentuk karakteristik seperti bulan sabit, sedangkan sel yang terinfeksi Malaria menunjukkan adanya bercak-bercak atau titik-titik berwarna gelap di dalam sel yang merupakan parasit. Misklasifikasi yang terjadi kemungkinan disebabkan oleh faktor-faktor seperti variasi kualitas citra, sudut pandang sel, atau overlap antar sel yang mengaburkan karakteristik morfologi khas. Meskipun demikian, tingkat misklasifikasi yang rendah (di bawah 8%) menunjukkan bahwa model memiliki kemampuan diskriminatif yang kuat dalam mengenali perbedaan morfologi kedua kelas tersebut.

Hasil *Confusion matrix* mengonfirmasi konsistensi dengan metrik evaluasi sebelumnya dan menunjukkan bahwa model tidak memiliki bias sistematis terhadap kelas tertentu. Performa yang sempurna pada Test Dataset dan sangat tinggi pada *Validation* Dataset (96,8%) memvalidasi bahwa model YOLO 11l mampu melakukan deteksi dan klasifikasi multi-kelas dengan andal pada citra mikroskopik darah. Secara visual, performa deteksi ditunjukkan pada Gambar 26 dan 27.



Gambar 26. Kolase Visual Deteksi pada Data Uji YOLO 11 (LR = 0.001)

Kolase visual pada Gambar 26 mengilustrasikan performa superior model YOLO 11 dengan learning rate 0.001 pada dataset uji, di mana setiap objek parasit terdeteksi dengan akurasi lokalisasi tinggi dan *confidence score* yang optimal, mengonfirmasi efektivitas konfigurasi ini sebagai setting terbaik. Untuk memvalidasi kemampuan generalisasi model pada data yang tidak digunakan dalam proses pelatihan, Gambar 27 menyajikan visualisasi performa deteksi pada dataset validasi dengan learning rate yang sama (0.001), memberikan perspektif komplementer terhadap hasil pada dataset uji.



Gambar 27. Kolase Visual Deteksi pada Data Validasi YOLO 11 (LR = 0.001)

Gambar 27 mendemonstrasikan konsistensi performa deteksi pada dataset validasi yang sejalan dengan hasil pada dataset uji, memvalidasi bahwa learning rate 0.001 menghasilkan model dengan keseimbangan optimal antara akurasi deteksi dan kemampuan generalisasi untuk implementasi klinis.

4.2.2 YOLO11l dengan *Learning rate* (0.0001)

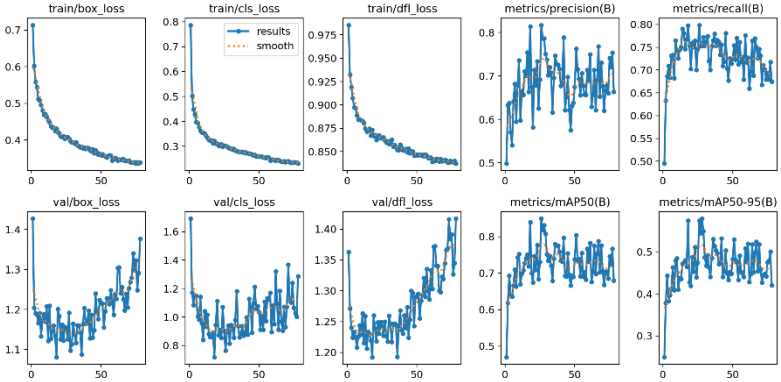
Eksperimen kedua dilakukan dengan menurunkan *Learning rate* awal menjadi 0.0001, bertujuan untuk menguji dampak pembelajaran yang lebih lambat dan

hati-hati terhadap stabilitas pelatihan dan performa deteksi. Metrik kinerja model dari eksperimen ini disajikan pada Tabel 8.

Tabel 8. Metrik Kinerja Deteksi YOLO 11 (LR = 0.0001)

Kelas	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Anemia	0.756	0.944	0.952	0.644
Malaria	0.810	0.830	0.882	0.719
Trypomastigote	0.771	0.490	0.660	0.375
Rata-Rata Makro	0.779	0.755	0.831	0.579

Hasil evaluasi menunjukkan pola performa yang berbeda dibandingkan eksperimen pertama. Rata-rata mAP@0.5 mencapai 0.831, sedikit lebih rendah dibandingkan eksperimen pertama (0.859), namun dengan distribusi performa yang berbeda antar kelas. Rata-rata precision sebesar 0.779 dan recall 0.755 mengindikasikan *Trade-off* antara akurasi prediksi dan kemampuan deteksi yang seimbang. Analisis per kelas menunjukkan variasi performa yang signifikan. Kelas Anemia mempertahankan performa terbaik dengan recall tertinggi (0.944) dan mAP@0.5 sebesar 0.952, menunjukkan kemampuan deteksi yang sangat baik. Kelas Malaria menunjukkan peningkatan yang menarik dengan mAP@0.5:0.95 tertinggi (0.719), mengindikasikan akurasi lokalisasi yang lebih baik pada berbagai *threshold* IoU. Sebaliknya, kelas *Trypomastigote* mengalami penurunan performa dengan recall terendah (0.490) dan mAP@0.5:0.95 sebesar 0.375, menunjukkan tantangan signifikan dalam mendeteksi kelas ini dengan *Learning rate* yang lebih rendah. Proses konvergensi model selama pelatihan disajikan pada Gambar 28, yang memberikan wawasan tentang dinamika pembelajaran dengan *Learning rate* yang lebih rendah.



Gambar 28. Grafik Hasil Pelatihan YOLO 11 (LR = 0.0001)

Dinamika pembelajaran pada eksperimen ini menunjukkan karakteristik konvergensi yang bertahap akibat penggunaan *learning rate* yang lebih kecil (0,0001). Ketiga komponen *loss* pada data latih (*box_loss*, *cls_loss*, dan *dfl_loss*) mengalami penurunan yang lambat namun konsisten dari awal hingga akhir pelatihan. Pola ini mengindikasikan bahwa model mampu mempelajari fitur pada data latih secara stabil tanpa gangguan gradien yang signifikan, meskipun membutuhkan waktu yang lebih lama dibandingkan eksperimen sebelumnya.

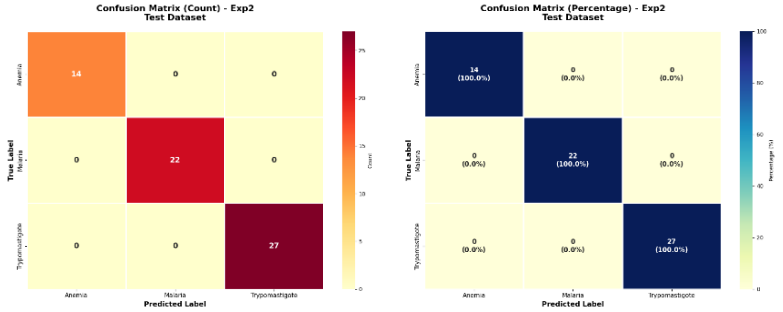
Kurva loss validasi menunjukkan karakteristik yang berbeda dengan data latih. Penurunan tajam terjadi pada *epoch* 0-10, di mana model dengan cepat mempelajari representasi fitur dasar. Namun, setelah *epoch* 20-30, kurva loss validasi mulai menunjukkan fluktuasi dengan rentang yang lebih lebar. Nilai val/box_loss berfluktuasi di kisaran 1.1-1.4, val/cls_loss di rentang 0.8-1.3, dan val/dfl_loss di kisaran 1.2-1.4, tanpa penurunan substansial lebih lanjut. Pada *epoch-epoch* akhir (60-78), *validation* loss bahkan menunjukkan kecenderungan meningkat, mengindikasikan potensi overfitting.

Metrik precision dan recall pada data validasi menunjukkan pola yang konsisten namun dengan fluktuasi yang lebih besar. Precision berfluktuasi di kisaran 0.57-0.81 setelah *epoch* 20, dengan variasi sekitar ±0.10 dari nilai rata-rata, sementara recall relatif lebih stabil di rentang 0.68-0.79. Metrik mAP50 dan mAP50-95 menunjukkan peningkatan pesat pada *epoch* awal (0-20), mencapai puncak performa di sekitar *epoch* 20-30, kemudian berfluktuasi dengan variasi yang cukup besar di sepanjang sisa proses pelatihan tanpa menunjukkan peningkatan yang konsisten.

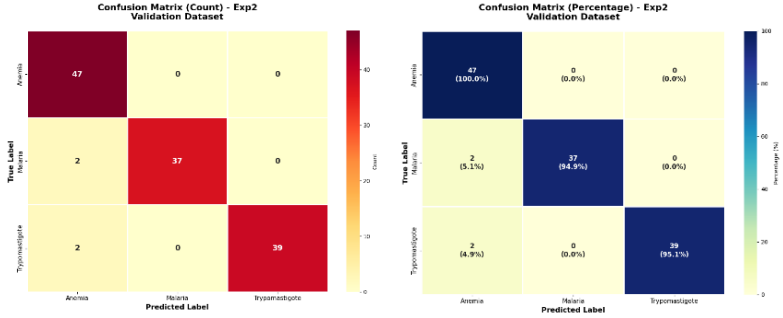
Performa terbaik dicapai pada *epoch* 28 dengan mAP@0.5 sebesar 0.831 dan mAP@0.5:0.95 sebesar 0.579. Setelah *epoch* tersebut, metrik evaluasi pada data validasi menunjukkan fluktuasi tanpa peningkatan yang konsisten, dengan nilai terendah dan tertinggi yang bervariasi signifikan, mengindikasikan ketidakstabilan dalam proses konvergensi.

Mekanisme early stopping dengan patience 50 *epoch* mendeteksi bahwa tidak ada peningkatan signifikan setelah *epoch* 28 dan menghentikan pelatihan pada *epoch* 78 setelah total durasi 8.053 jam. Pola pembelajaran ini mengindikasikan bahwa *Learning rate* yang lebih rendah (0.0001), meskipun memberikan konvergensi yang lebih stabil pada data latih, menciptakan tantangan dalam generalisasi pada data validasi dengan fluktuasi performa yang lebih besar. Model terbaik (*best*.pt) yang disimpan pada *epoch* 28 menjadi checkpoint final untuk evaluasi lebih lanjut.

Untuk menganalisis pengaruh *Learning rate* yang lebih rendah terhadap diferensiasi kelas, gambar 29 menyajikan *Confusion matrix* untuk *Dataset Uji* dan gambar 30 menyajikan *Confusion matrix* untuk *Dataset Validasi*.



Gambar 29. *Confusion matrix* pada Dataset Uji



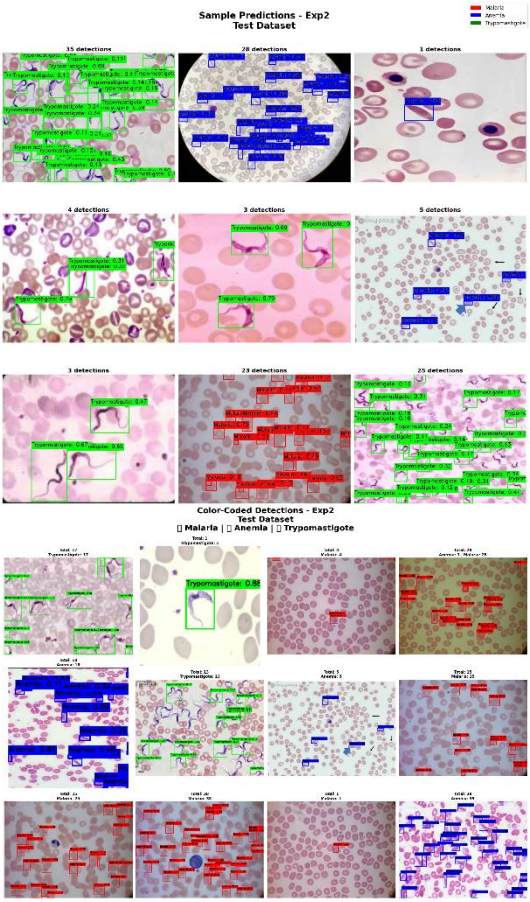
Gambar 30. *Confusion matrix* pada Dataset Validasi

Pada Test Dataset, model kembali menunjukkan performa sempurna dengan akurasi 100% untuk semua kelas, identik dengan eksperimen pertama. Sebanyak 14 instance Anemia, 22 instance Malaria, dan 27 instance *Trypomastigote* berhasil diklasifikasikan dengan benar tanpa kesalahan.

Pada *Validation* Dataset, model menunjukkan peningkatan performa signifikan dengan akurasi keseluruhan 97.6%. Dari 47 instance Anemia, seluruhnya (100%) diklasifikasikan dengan benar tanpa kesalahan, menunjukkan peningkatan dibandingkan eksperimen pertama (95.7%). Untuk kelas Malaria, dari 39 instance, 37 (94.9%) diprediksi dengan tepat, sementara 2 instance (5.1%) salah diklasifikasikan sebagai Anemia, menunjukkan perbaikan dari eksperimen pertama (92.3%). Kelas *Trypomastigote* menunjukkan hasil 39 dari 41 instance (95.1%) teridentifikasi dengan benar, dengan 2 instance (4.9%) salah diprediksi sebagai Anemia.

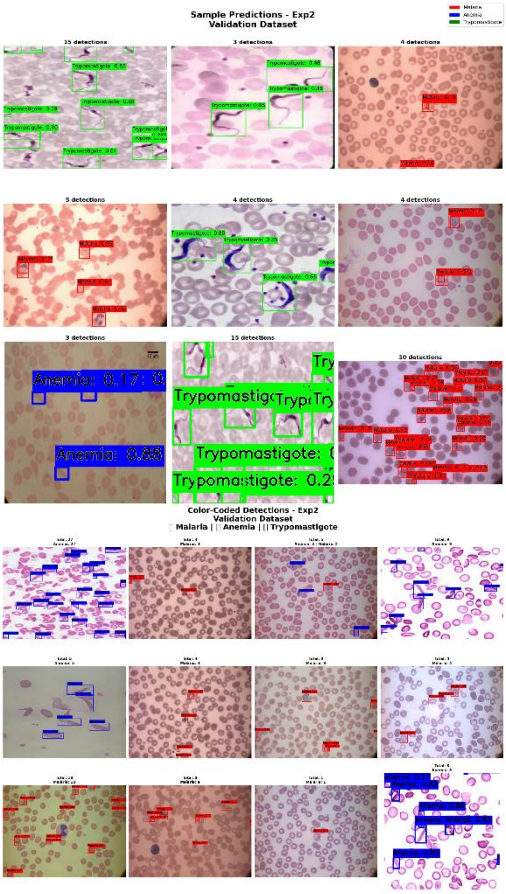
Pola misklasifikasi pada eksperimen ini menunjukkan karakteristik berbeda: tidak ada lagi kesalahan Anemia yang diprediksi sebagai Malaria, namun muncul kesalahan baru berupa *Trypomastigote* yang diprediksi sebagai Anemia. Hal ini

mengindikasikan bahwa *Learning rate* yang lebih rendah membuat model lebih konservatif dalam membedakan kelas dengan karakteristik yang kompleks. Secara keseluruhan, eksperimen kedua menunjukkan peningkatan performa klasifikasi pada *validation* dataset (97.6% vs 96.8%), mengonfirmasi bahwa *Learning rate* yang lebih rendah dapat meningkatkan kemampuan diskriminatif model meskipun dengan *Trade-off* berupa proses konvergensi yang lebih lambat dan risiko overfitting yang lebih tinggi pada fase akhir pelatihan. Secara visual, performa deteksi ditunjukkan pada Gambar 31 dan 32.



Gambar 31. Kolase Visual Deteksi pada Data Uji YOLO 11 (LR = 0.0001)

Visualisasi pada Gambar 31 memperlihatkan performa deteksi yang sangat baik pada dataset uji, mengonfirmasi bahwa learning rate 0.0001 mampu menghasilkan model dengan kemampuan lokalisasi parasit yang presisi meskipun memerlukan waktu konvergensi yang lebih lama. Sejalan dengan evaluasi pada eksperimen learning rate 0.001, performa model dengan learning rate 0.0001 juga dievaluasi pada dataset yang berbeda, di mana Gambar 32 menampilkan hasil deteksi pada dataset validasi untuk mengonfirmasi konsistensi dan robustness model terhadap variasi data.



Gambar 32. Kolase Visual Deteksi pada Data Validasi YOLO 11 (LR = 0.0001)

Gambar 32 menunjukkan hasil deteksi pada dataset validasi yang konsisten dengan performa pada dataset uji, memvalidasi bahwa learning rate 0.0001 menghasilkan model dengan kemampuan generalisasi yang memadai untuk aplikasi deteksi malaria secara praktis.

4.2.3 YOLO11l dengan *Learning rate* (0.0015)

Eksperimen ketiga dilakukan dengan menggunakan *Learning rate* 0.0015, nilai yang merupakan kompromi antara kecepatan konvergensi (0.001) dan kehati-hatian fine-tuning (0.0001). Tujuan dari eksperimen ini adalah untuk mengidentifikasi apakah terdapat *Learning rate* optimal di antara kedua batas tersebut yang dapat meningkatkan stabilitas dan akurasi model. Metrik kinerja model dari eksperimen ini disajikan pada Tabel 9.

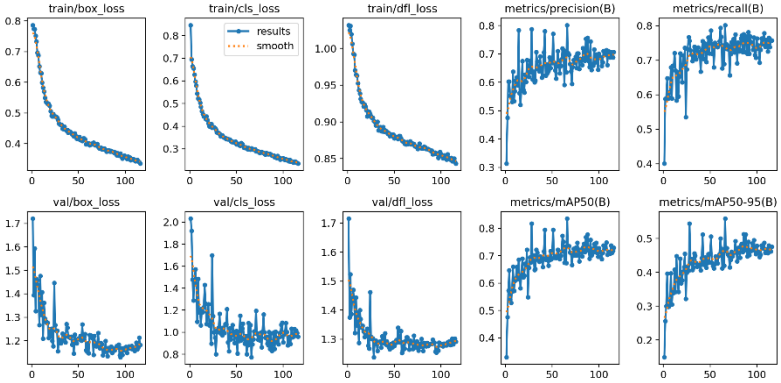
Tabel 9. Matriks Kinerja Deteksi YOLO 11 (LR = 0.0015)

Kelas	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
Anemia	0.863	0.915	0.958	0.635
Malaria	0.722	0.787	0.788	0.600
Trypomastigote	0.820	0.695	0.765	0.443
Rata-Rata Makro	0.802	0.799	0.837	0.559

Hasil evaluasi menunjukkan peningkatan performa yang konsisten dibandingkan eksperimen sebelumnya. Rata-rata precision sebesar 0.802 dan recall 0.799 mengindikasikan keseimbangan yang sangat baik antara akurasi prediksi dan kemampuan deteksi objek. Nilai mAP@0.5 sebesar 0.837 berada di antara hasil eksperimen pertama (0.859) dan kedua (0.831), menunjukkan bahwa *Learning rate* 0.0015 mampu mencapai performa deteksi yang kompetitif dengan stabilitas yang lebih baik.

Analisis per kelas menunjukkan distribusi performa yang seimbang. Kelas Anemia mempertahankan performa terbaik dengan precision 0.863, recall tertinggi (0.915), dan mAP@0.5 mencapai 0.958, mengonfirmasi konsistensi deteksi yang sangat baik untuk kelas ini. Kelas Malaria menunjukkan peningkatan signifikan pada mAP@0.5:0.95 (0.600), tertinggi di antara semua eksperimen, mengindikasikan akurasi lokalisasi bounding box yang superior pada berbagai *threshold* IoU. Kelas *Trypomastigote*, meskipun masih menunjukkan tantangan dengan recall 0.695, mencapai precision tertinggi (0.820) dibandingkan eksperimen sebelumnya, menunjukkan bahwa *Learning rate* 0.0015 efektif dalam mengurangi *false positive* pada kelas yang paling menantang ini. Kecepatan inferensi model mencapai 27,1 ms per citra, konsisten dengan eksperimen sebelumnya dan memadai untuk implementasi sistem deteksi real-time dalam

setting laboratorium klinis. Proses konvergensi model selama pelatihan disajikan pada Gambar 33, yang memberikan wawasan tentang dinamika pembelajaran dengan *Learning rate* 0.0015.



Gambar 33.Grafik Hasil Pelatihan YOLO 11 (LR = 0.0015)

Pada Dinamika pembelajaran pada eksperimen ini menunjukkan karakteristik yang lebih unggul dibandingkan dengan eksperimen sebelumnya. Ketiga komponen loss pada data latih, yaitu *train/box_loss*, *train/cls_loss*, dan *train/dfl_loss*, mengalami penurunan secara konsisten dan halus sepanjang 116 *epoch*. Pola penurunan yang bersifat gradual namun stabil tersebut mencerminkan proses pembelajaran yang efisien. Kondisi ini menunjukkan bahwa nilai *Learning rate* sebesar 0,0015 berhasil mencapai keseimbangan optimal antara kecepatan konvergensi dan stabilitas proses pelatihan.

Kurva *loss* pada data validasi juga menunjukkan pola yang sangat baik. Penurunan tajam terjadi pada *epoch* 0 hingga 30, yang mengindikasikan bahwa model mampu mempelajari representasi fitur dasar dari ketiga kelas objek secara cepat. Setelah *epoch* 30, kurva *val/box_loss*, *val/cls_loss*, dan *val/dfl_loss* bergerak menuju kondisi stabil dengan fluktuasi yang sangat terkontrol. Tidak terdapat tren peningkatan nilai loss yang signifikan sebagaimana terjadi pada eksperimen kedua, sehingga dapat disimpulkan bahwa model tidak mengalami *overfitting* meskipun proses pelatihan dilanjutkan hingga *epoch* ke-116.

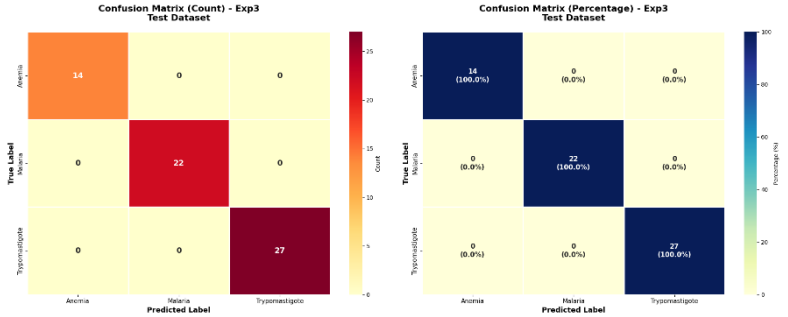
Metrik *precision* dan *recall* pada data validasi menunjukkan peningkatan secara konsisten dan stabil. Nilai *precision* yang semula berada pada kisaran 0,50 hingga 0,60 pada *epoch* awal meningkat dan menjadi stabil pada kisaran 0,65 hingga 0,80 setelah *epoch* ke-50. Sementara itu, nilai *recall* yang semula berada pada rentang 0,55 hingga 0,65 menunjukkan peningkatan serupa dan mencapai

kestabilan pada kisaran 0,70 hingga 0,80. Pola tersebut mengindikasikan adanya proses pembelajaran yang progresif dan berkelanjutan tanpa fluktuasi ekstrem.

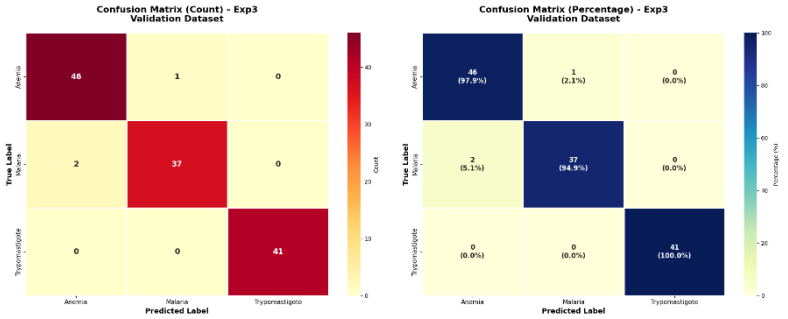
Metrik *mAP50* dan *mAP50-95* menunjukkan karakteristik konvergensi yang optimal. Kedua metrik mengalami peningkatan signifikan pada tahap awal pelatihan, yaitu pada *epoch* 0 hingga 30. Selanjutnya, metrik terus meningkat secara bertahap dan mencapai performa terbaik pada *epoch* ke-66 dengan nilai mAP@0,5 sebesar 0,837 dan mAP@0,5:0,95 sebesar 0,559. Setelah *epoch* tersebut, nilai kedua metrik berfluktuasi di sekitar titik optimal dengan variasi yang minimal, sehingga menunjukkan tingkat stabilitas model yang sangat baik.

Performa terbaik dicapai pada *epoch* ke-66, kemudian mekanisme *early stopping* dengan *patience* selama 50 *epoch* menghentikan pelatihan pada *epoch* ke-116 setelah tidak ditemukan peningkatan signifikan dalam jangka waktu tersebut. Total durasi pelatihan adalah 11.943 jam. Waktu tersebut lebih efisien dibandingkan eksperimen pertama yang berlangsung selama 16.027 jam, tetapi lebih lama dibandingkan eksperimen kedua yang membutuhkan 8.053 jam. Model terbaik (*best.pt*) yang disimpan pada *epoch* ke-66 dijadikan sebagai *checkpoint* akhir dengan performa optimal.

Untuk menganalisis pengaruh *Learning rate* yang lebih rendah terhadap diferensiasi kelas, gambar 34 menyajikan *Confusion matrix* untuk Dataset Uji dan gambar 35 menyajikan *Confusion matrix* untuk Dataset Validasi.



Gambar 34. *Confusion matrix* pada Dataset Uji

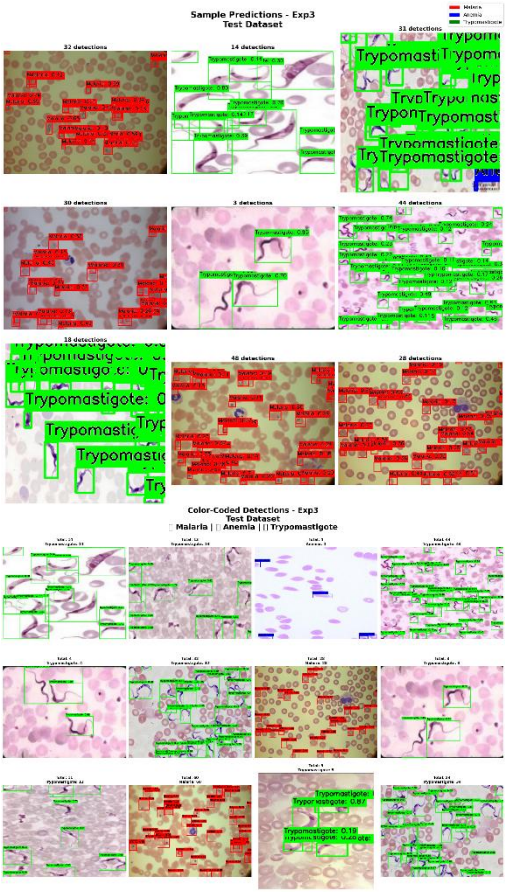


Gambar 35. *Confusion matrix* pada Dataset Validasi

Confusion matrix pada Dataset Uji menunjukkan performa klasifikasi yang sempurna dengan akurasi 100% untuk semua kelas. Sebanyak 14 objek Anemia, 22 objek Malaria, dan 27 objek *Trypomastigote* diklasifikasikan dengan benar tanpa ada kesalahan klasifikasi sama sekali. Hasil ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang sangat baik pada data yang tidak pernah dilihat sebelumnya.

Confusion matrix pada Dataset Validasi menunjukkan performa yang sangat baik dengan tingkat misklasifikasi yang minimal. Dari 47 objek Anemia, 46 (97.9%) diklasifikasikan dengan benar dan hanya 1 (2.1%) salah diklasifikasikan sebagai Malaria. Untuk kelas Malaria, 37 dari 39 objek (94.9%) terdeteksi dengan benar, sementara 2 objek (5.1%) salah diklasifikasikan sebagai Anemia. Kelas *Trypomastigote* mencapai akurasi sempurna 100% dengan 41 dari 41 objek terdeteksi dengan benar.

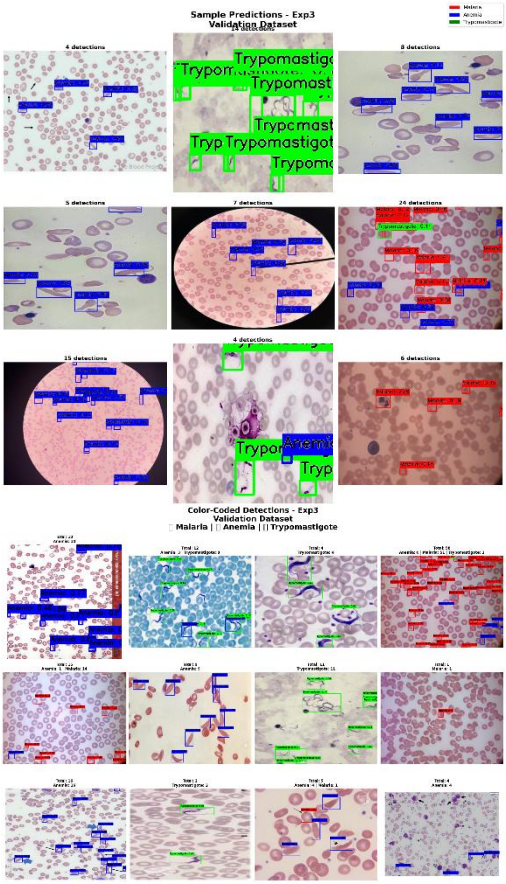
Hasil ini menunjukkan bahwa *Learning rate* 0.0015 efektif dalam mengurangi kesalahan klasifikasi, terutama pada kelas-kelas yang memiliki karakteristik visual yang mirip seperti Anemia dan Malaria. Dibandingkan dengan eksperimen sebelumnya, konfigurasi ini menghasilkan model dengan kemampuan diferensiasi kelas yang superior dan tingkat kesalahan yang sangat rendah. Secara visual, performa deteksi ditunjukkan pada Gambar 36 dan 37.



Gambar 36. Kolase Visual Deteksi pada Data Uji YOLO 11 (LR = 0.0015)

Kolase visual pada Gambar 36 mendemonstrasikan kemampuan konsisten model YOLO 11 dengan learning rate 0.0015 dalam mendeteksi dan melokalisasi parasit malaria pada dataset uji, dengan bounding box yang akurat dan *confidence score* yang tinggi untuk setiap objek terdeteksi. Mengikuti protokol evaluasi yang konsisten dengan eksperimen sebelumnya, Gambar 37 mengilustrasikan hasil

deteksi model dengan learning rate 0.0015 pada dataset validasi, melengkapi analisis komparatif mengenai pengaruh learning rate terhadap kemampuan generalisasi model dalam kondisi data yang berbeda.



Gambar 37. Kolase Visual Deteksi pada Data Validasi YOLO 11 (LR = 0.0015)

Gambar 37 mengonfirmasi generalisasi model yang baik pada dataset validasi, di mana visualisasi deteksi menunjukkan konsistensi performa dengan tingkat

kesalahan minimal, memvalidasi efektivitas learning rate 0.0015 dalam mencapai keseimbangan optimal antara kecepatan konvergensi dan akurasi deteksi.

4.2.4. Perbandingan Perbandingan Kinerja Tiga Eksperimen *Learning rate* YOLO 11l

Untuk mengidentifikasi konfigurasi optimal dalam deteksi penyakit infeksi darah, dilakukan analisis komparatif terhadap tiga eksperimen dengan variasi *Learning rate* yang berbeda: 0.001, 0.0001, dan 0.0015. Perbandingan komprehensif ini mencakup metrik performa, dinamika konvergensi, efisiensi komputasi, dan kemampuan klasifikasi model.

1. Perbandingan Metrik Performa Deteksi

Tabel 10 menyajikan perbandingan metrik kinerja deteksi untuk ketiga eksperimen *Learning rate* pada model YOLO 11l.

Tabel 10. Perbandingan Metrik Kinerja Deteksi Antar *Learning Rate*

Learning rate (LR)	Macro Precision (P)	Macro Recall (R)	mAP@0.5	mAP@0.5:0.95	Inference Time (ms)
0.001	0.834	0.818	0.859	0.588	27.2
0.0001	0.779	0.755	0.831	0.579	26.9
0.0015	0.802	0.799	0.837	0.559	27.1

Analisis metrik global menunjukkan bahwa eksperimen pertama dengan *Learning rate* 0.001 mencapai performa terbaik secara keseluruhan. Model ini memperoleh nilai mAP@0.5 tertinggi sebesar 0.859, mengungguli *Learning rate* 0.0001 (0.831) dan 0.0015 (0.837) dengan margin 2.8% dan 2.2% secara berturut-turut. Keunggulan ini mengindikasikan bahwa *Learning rate* 0.001 memberikan keseimbangan optimal antara kecepatan pembelajaran dan stabilitas konvergensi untuk dataset yang digunakan.

Metrik precision menunjukkan pola serupa, di mana *Learning rate* 0.001 mencapai nilai tertinggi 0.834, diikuti oleh 0.0015 (0.802) dan 0.0001 (0.779). Precision yang lebih tinggi pada *Learning rate* 0.001 mengindikasikan tingkat *false positive* yang lebih rendah, yang sangat penting dalam aplikasi diagnostik medis untuk menghindari kesalahan deteksi yang dapat mempengaruhi keputusan klinis.

Dari sisi recall, *Learning rate* 0.001 kembali menunjukkan performa superior dengan nilai 0.818, mengungguli 0.0015 (0.799) dan 0.0001 (0.755). Recall yang lebih tinggi menunjukkan kemampuan model untuk mendeteksi lebih banyak objek yang sebenarnya ada, mengurangi risiko *false negative* yang dapat menyebabkan kasus penyakit tidak terdeteksi.

Menariknya, meskipun *Learning rate* 0.001 memiliki performa terbaik pada mAP@0.5, pada metrik mAP@0.5:0.95 yang lebih ketat, perbedaan antar

eksperimen menjadi lebih kecil. *Learning rate* 0.001 mencapai 0.588, hanya sedikit lebih tinggi dibanding 0.0001 (0.579) dan 0.0015 (0.559). Hal ini mengindikasikan bahwa ketiga *Learning rate* menghasilkan akurasi lokalisasi bounding box yang relatif sebanding pada berbagai *threshold* IoU yang lebih ketat.

Kecepatan inferensi menunjukkan konsistensi yang sangat baik di antara ketiga eksperimen, berkisar antara 26.9-27.2 ms per citra. Perbedaan kecepatan yang minimal ini (maksimal 0.3 ms) menunjukkan bahwa variasi *Learning rate* tidak mempengaruhi efisiensi komputasi pada fase inferensi, sehingga pemilihan *Learning rate* dapat difokuskan pada akurasi deteksi tanpa *Trade-off* terhadap kecepatan.

2. Perbandingan Performa Per Kelas

Tabel 11. Perbandingan Metrik Kinerja Per Kelas Antar *Learning Rate*

Kelas	*Learning Rate*	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Anemia	0.001	0.949	0.914	0.968	0.665
	0.0001	0.756	0.944	0.952	0.644
	0.0015	0.863	0.915	0.958	0.635
Malaria	0.001	0.739	0.830	0.835	0.651
	0.0001	0.810	0.830	0.882	0.719
	0.0015	0.722	0.787	0.788	0.600
Trypomastigote	0.001	0.814	0.709	0.776	0.449
	0.0001	0.771	0.490	0.660	0.375
	0.0015	0.820	0.695	0.765	0.443

Tabel 11 menyajikan perbandingan metrik performa deteksi untuk masing-masing kelas penyakit pada ketiga eksperimen *Learning Rate*. Kelas Anemia menunjukkan performa terbaik secara konsisten pada *Learning rate* 0.001 dengan precision tertinggi 0.949 dan mAP@0.5 sebesar 0.968. Namun, *Learning rate* 0.0001 mencapai recall tertinggi (0.944), mengindikasikan kemampuan superior dalam mendeteksi semua instance Anemia yang ada. *Learning rate* 0.0015 memberikan performa seimbang dengan precision 0.863 dan recall 0.915. Perbedaan performa antar *Learning rate* pada kelas Anemia relatif kecil, dengan semua konfigurasi mencapai mAP@0.5 di atas 0.95, menunjukkan bahwa karakteristik morfologi Anemia (bentuk bulan sabit) mudah dipelajari oleh model dengan berbagai *Learning Rate*.

Kelas Malaria menunjukkan variasi performa yang lebih signifikan. *Learning rate* 0.0001 mencapai performa terbaik dengan mAP@0.5 sebesar 0.882 dan mAP@0.5:0.95 tertinggi 0.719, mengungguli *Learning rate* 0.001 (0.835 dan 0.651) serta 0.0015 (0.788 dan 0.600). Temuan ini menunjukkan bahwa pembelajaran yang lebih lambat dan hati-hati (*Learning rate* 0.0001) lebih efektif dalam mempelajari fitur-fitur halus dari parasit Malaria yang muncul sebagai

bercak gelap dalam sel darah merah. Recall yang konsisten (0.830) pada *Learning rate* 0.001 dan 0.0001 menunjukkan kemampuan deteksi yang sebanding, sementara precision yang lebih tinggi pada 0.0001 (0.810 vs 0.739) mengindikasikan pengurangan *false positive* yang signifikan.

Kelas *Trypomastigote* menjadi kelas paling menantang untuk semua *Learning Rate*, dengan mAP@0.5 berkisar 0.660-0.776 dan mAP@0.5:0.95 hanya 0.375-0.449. *Learning rate* 0.001 memberikan performa terbaik dengan mAP@0.5 sebesar 0.776 dan recall 0.709, sementara *Learning rate* 0.0015 mencapai precision tertinggi (0.820). *Learning rate* 0.0001 menunjukkan performa terendah dengan recall hanya 0.490, mengindikasikan bahwa pembelajaran yang terlalu lambat kurang efektif untuk kelas yang kompleks ini. Tantangan deteksi *Trypomastigote* kemungkinan disebabkan oleh variasi morfologi yang lebih besar dan ukuran objek yang lebih kecil dibanding dua kelas lainnya.

Secara keseluruhan, tidak ada satu *Learning rate* yang secara konsisten unggul untuk semua kelas. *Learning rate* 0.001 optimal untuk Anemia dan *Trypomastigote*, sementara 0.0001 lebih baik untuk Malaria. Hal ini mengindikasikan bahwa karakteristik visual yang berbeda dari setiap penyakit memerlukan kecepatan pembelajaran yang berbeda untuk mencapai performa optimal.

3. Perbandingan Efisiensi Pelatihan

Tabel 12 menyajikan perbandingan efisiensi komputasi dan konvergensi untuk ketiga eksperimen *Learning Rate*.

Tabel 12. Perbandingan Efisiensi Pelatihan Antar Learning Rate

Learning Rate	Total *Epochs*	*Best Epoch*	Durasi Pelatihan (Jam)	mAP@0.5 Terbaik
0.001	155	105	16.027	0.859
0.0001	78	28	8.053	0.831
0.0015	116	66	11.943	0.837

Analisis efisiensi pelatihan mengungkap *Trade-off* yang menarik antara kecepatan konvergensi dan performa akhir. *Learning rate* 0.0001 menunjukkan konvergensi tercepat, mencapai performa terbaik pada *epoch* 28 dan selesai pelatihan pada *epoch* 78 dengan total durasi 8.053 jam. Efisiensi ini menjadikannya pilihan yang menarik untuk eksperimen cepat atau ketika sumber daya komputasi terbatas. Namun, kecepatan konvergensi ini diiringi dengan performa akhir yang sedikit lebih rendah (mAP@0.5 = 0.831).

Learning rate 0.001 memerlukan waktu pelatihan terlama (16.027 jam) dan mencapai performa terbaik pada *epoch* 105 dari total 155 *epoch*. Meskipun membutuhkan waktu hampir dua kali lipat dibanding *learning rate* 0.0001, model ini berhasil mencapai performa tertinggi dengan mAP@0.5 sebesar 0.859. Pola pembelajaran yang lebih panjang ini mengindikasikan bahwa model terus

melakukan fine-tuning dan peningkatan performa secara bertahap hingga *epoch* yang lebih tinggi.

Learning rate 0.0015 menunjukkan karakteristik yang berada di antara kedua *learning rate* lainnya, dengan durasi pelatihan 11.943 jam dan performa terbaik pada *epoch* 66 dari total 116 *epoch*. Model ini mencapai keseimbangan yang baik antara waktu pelatihan dan performa akhir (mAP@0.5 = 0.837), menjadikannya kompromi yang menarik antara efisiensi dan akurasi.

Mekanisme early stopping berfungsi efektif pada semua eksperimen dengan patience 50 *epoch*. Pada *learning rate* 0.001, early stopping menghentikan pelatihan pada *epoch* 155 setelah tidak ada peningkatan sejak *epoch* 105. Pada *learning rate* 0.0001, pelatihan dihentikan pada *epoch* 78 setelah stagnasi sejak *epoch* 28. Sedangkan pada *learning rate* 0.0015, pelatihan berhenti pada *epoch* 116 setelah tidak ada peningkatan sejak *epoch* 66. Pola ini menunjukkan bahwa *learning rate* yang lebih tinggi memerlukan lebih banyak *epoch* untuk mencapai konvergensi penuh, namun juga mampu mencapai performa yang lebih tinggi.

Efisiensi waktu per *epoch* relatif konsisten di antara ketiga eksperimen, dengan rata-rata sekitar 6.2 menit per *epoch*. Konsistensi ini menunjukkan bahwa perbedaan durasi total pelatihan terutama disebabkan oleh jumlah *epoch* yang diperlukan untuk konvergensi, bukan oleh perbedaan kecepatan komputasi per *epoch*.

4. Perbandingan Dinamika Konvergensi

Analisis kurva learning menunjukkan pola konvergensi yang berbeda untuk setiap learning rate. *Learning rate* 0.001 menunjukkan penurunan loss yang smooth dan konsisten pada data latih, dengan kurva validasi yang stabil setelah *epoch* 50. Fluktuasi minimal pada loss validasi mengindikasikan pembelajaran yang stabil tanpa overfitting yang signifikan hingga akhir pelatihan.

Learning rate 0.0001 menunjukkan konvergensi yang sangat cepat pada *epoch* awal (0-20), namun diikuti dengan fluktuasi yang lebih besar pada loss validasi di *epoch* selanjutnya. Kurva loss validasi menunjukkan kecenderungan meningkat pada *epoch-epoch* akhir, mengindikasikan potensi awal overfitting. Meskipun demikian, model tetap mencapai performa yang kompetitif pada *epoch* 28 sebelum terjadi degradasi.

Learning rate 0.0015 menunjukkan karakteristik konvergensi yang optimal dengan penurunan loss yang bertahap namun konsisten. Kurva loss validasi menunjukkan stabilitas yang sangat baik dengan fluktuasi minimal setelah *epoch* 30. Tidak ada tren peningkatan loss yang signifikan hingga akhir pelatihan, mengindikasikan bahwa model tidak mengalami overfitting meskipun dilatih hingga *epoch* 116.

Metrik mAP50 dan mAP50-95 menunjukkan pola yang konsisten dengan loss function. Pada *learning rate* 0.001, kedua metrik mengalami peningkatan pesat hingga *epoch* 50, kemudian meningkat secara bertahap hingga mencapai puncak pada *epoch* 105. *Learning rate* 0.0001 menunjukkan peningkatan yang sangat

cepat pada *epoch* 0-20, mencapai puncak di *epoch* 28, namun kemudian berfluktuasi tanpa peningkatan konsisten. *Learning rate* 0.0015 menunjukkan peningkatan yang stabil dan berkelanjutan hingga *epoch* 66, dengan fluktuasi minimal setelahnya.

5. Perbandingan Performa Klasifikasi

Tabel 13. Perbandingan Akurasi Klasifikasi pada *Validation* Dataset

Learning Rate	Akurasi Keseluruhan (%)	Anemia (%)	Malaria (%)	*Trypomastigote* (%)
0.001	96.8	95.7	92.3	100
0.0001	97.6	100	94.9	95.1
0.0015	98.4	97.9	94.9	100

Analisis *confusion matrix* menunjukkan perbedaan nyata dalam kemampuan diskriminatif setiap model. *Learning rate* 0.0015 mencapai akurasi keseluruhan tertinggi sebesar 98.4%, melampaui 0.0001 (97.6%) dan 0.001 (96.8%). Peningkatan ini terutama disebabkan oleh berkurangnya kesalahan klasifikasi antara kelas Anemia dan Malaria.

Pada *learning rate* 0.001, terdapat 2 instance Anemia yang salah diklasifikasikan sebagai Malaria dan 3 instance Malaria yang salah diklasifikasikan sebagai Anemia, sehingga total terjadi 5 kesalahan dari 86 instance. Pola misklasifikasi dua arah ini mengindikasikan adanya tumpang tindih pada feature space antara kedua kelas.

Learning rate 0.0001 menunjukkan performa sempurna untuk kelas Anemia (100%), namun terdapat 2 kesalahan Malaria yang diklasifikasikan sebagai Anemia dan muncul kesalahan baru berupa 2 instance *Trypomastigote* yang diklasifikasikan sebagai Anemia. Pergeseran pola kesalahan ini menunjukkan bahwa *learning rate* yang lebih rendah membuat model lebih konservatif dan cenderung memilih kelas yang paling familiar, yaitu Anemia, ketika menghadapi ketidakpastian.

Learning rate 0.0015 memberikan keseimbangan terbaik dengan hanya 1 kesalahan Anemia yang diklasifikasikan sebagai Malaria dan 2 kesalahan Malaria yang diklasifikasikan sebagai Anemia, serta performa sempurna 100% pada kelas *Trypomastigote*. Distribusi kesalahan yang minimal dan hanya terjadi pada pasangan Anemia–Malaria mengindikasikan bahwa *learning rate* ini menghasilkan pemisahan feature space yang paling optimal.

Seluruh model mencapai akurasi sempurna 100% pada test dataset, menunjukkan kemampuan generalisasi yang sangat baik terhadap data yang belum pernah dilihat sebelumnya. Konsistensi performa ini menegaskan bahwa perbedaan pada *validation* set bukan disebabkan oleh overfitting, melainkan mencerminkan sensitivitas model terhadap variasi halus pada karakteristik data.

6. Rekomendasi Konfigurasi Optimal

Pemilihan *learning rate* 0.001 didasarkan pada evaluasi komprehensif terhadap performa deteksi keseluruhan, keseimbangan performa antar kelas, dan mitigasi risiko klinis. Konfigurasi ini memberikan kombinasi terbaik antara akurasi deteksi, reliabilitas klasifikasi, dan keamanan diagnostik.

Learning rate 0.001 mencapai mAP@0.5 tertinggi sebesar 85.9%, unggul 2.8% dari *learning rate* 0.0001 (83.1%) dan 2.2% dari *learning rate* 0.0015 (83.7%). Presisi rata-rata (0.834), *recall* rata-rata (0.818), dan mAP@0.5:0.95 (0.588) juga menunjukkan nilai tertinggi, mengindikasikan konsistensi performa pada berbagai tingkat presisi lokalisasi.

Meskipun *learning rate* 0.0015 memiliki akurasi klasifikasi tertinggi (98.4%), diikuti 0.0001 (97.6%) dan 0.001 (96.8%), *learning rate* 0.001 memberikan keseimbangan superior antara akurasi klasifikasi dan metrik deteksi objek. *Learning rate* 0.001 mencapai akurasi 100% pada kelas Trypomastigote, sementara *learning rate* 0.0001 meskipun mencapai 100% pada kelas Anemia, mengalami penurunan drastis pada *recall* Trypomastigote (49.0%), mengindikasikan kegagalan mendeteksi lebih dari separuh kasus.

Learning rate 0.001 menunjukkan distribusi performa paling seimbang dengan keunggulan signifikan pada Anemia (mAP@0.5: 96.8%, presisi: 94.9%) dan Trypomastigote (mAP@0.5: 77.6%, *recall*: 70.9%). Penurunan *recall* Trypomastigote pada *learning rate* 0.0001 (49.0%) mengindikasikan 51% kasus tidak terdeteksi, berpotensi salah diklasifikasikan sebagai Malaria atau Anemia, yang merupakan risiko tidak dapat diterima dalam sistem diagnostik.

Presisi Anemia pada *learning rate* 0.0001 hanya 75.6%, turun 19.3% dari *learning rate* 0.001 (94.9%). Dari setiap 100 prediksi "Anemia", terdapat 24 kasus salah klasifikasi pada 0.0001 dibanding hanya 5 kasus pada 0.001. Kesalahan ini berpotensi menghasilkan informasi diagnostik yang menyesatkan dan mengurangi reliabilitas sistem.

Learning rate 0.0015 mencapai mAP@0.5 sebesar 83.7%, lebih rendah 2.2% dari 0.001, dengan performa lebih rendah pada Malaria (78.8%). Nilai mAP@0.5:0.95 (0.559) menunjukkan konsistensi lokalisasi lebih rendah dibanding 0.001 (0.588), membuktikan akurasi klasifikasi tinggi tidak selalu berkorelasi dengan kemampuan deteksi superior.

Waktu pelatihan *learning rate* 0.001 (16.027 jam) lebih lama dari 0.0001 (8.053 jam), namun ini merupakan biaya satu kali yang dapat diterima. Kecepatan inferensi 27.2 ms per citra memungkinkan pemrosesan 36 citra per detik, sangat memadai untuk aplikasi diagnostik waktu nyata tanpa hambatan.

Learning rate 0.001 mencapai konvergensi stabil dengan kurva *loss* halus, menunjukkan pembelajaran bertahap dan konsisten. *Learning rate* terlalu besar (0.0015) dapat melewati titik optimal, sementara terlalu kecil (0.0001) dapat terjebak dalam titik minimal lokal dan gagal mengeksplorasi ruang solusi secara optimal.

Meskipun *learning rate* 0.0001 unggul pada Malaria dengan selisih 4.7% (mAP@0.5: 88.2% vs 83.5%), keunggulan ini tidak dapat membenarkan penurunan 21.9% pada *recall* Trypomastigote dan 19.3% pada presisi Anemia. Kemampuan membedakan Malaria dari kondisi darah lainnya sama pentingnya dengan akurasi deteksi Malaria itu sendiri.

Learning rate 0.0001, dengan tingkat hasil negatif palsu 51% pada Trypomastigote dan hasil positif palsu 24% pada Anemia, menciptakan banyak titik kegagalan. Kasus Trypomastigote tidak terdeteksi berpotensi salah diklasifikasikan sebagai Malaria atau Anemia, sementara Anemia salah diprediksi sebagai Malaria (18% kasus) menghasilkan label diagnostik tidak sesuai kondisi sebenarnya.

Learning rate 0.001, dengan presisi dan *recall* tinggi pada semua kelas, menunjukkan model telah mempelajari fitur pembeda asli untuk setiap kelas. Sebaliknya, *learning rate* 0.0001 yang terlalu dioptimalkan untuk Malaria mengindikasikan potensi *overfitting*, dapat menyebabkan penurunan performa pada data dunia nyata dengan variasi lebih luas.

Learning rate 0.001, dengan tingkat kesalahan rendah di semua kelas (rata-rata 5.1%), memberikan konsistensi untuk membangun kepercayaan klinis. *Learning rate* 0.0001 dengan tingkat kesalahan 24.4% pada Anemia akan menyebabkan ketidaksesuaian antara hasil AI dan penilaian klinis, mendorong klinisi mengabaikan rekomendasi sistem.

Berdasarkan evaluasi komprehensif, *learning rate* 0.001 merupakan pilihan paling optimal dengan metrik performa terbaik (mAP@0.5: 85.9%, presisi: 83.4%, *recall*: 81.8%) dan keseimbangan tepat antara sensitivitas dan spesifisitas. Meskipun akurasi klasifikasi sedikit lebih rendah (96.8% vs 98.4% dan 97.6%), perbedaan ini tidak signifikan dibanding keunggulan substansial pada metrik deteksi objek. Pertukaran minimal antara performa kelas, risiko kesalahan klasifikasi rendah, dan kecepatan inferensi memadai (27.2 ms) menjadikan *learning rate* 0.001 sebagai konfigurasi paling dapat dipertahankan, aman, dan praktis untuk implementasi sistem deteksi penyakit infeksi darah berbasis YOLO11l.

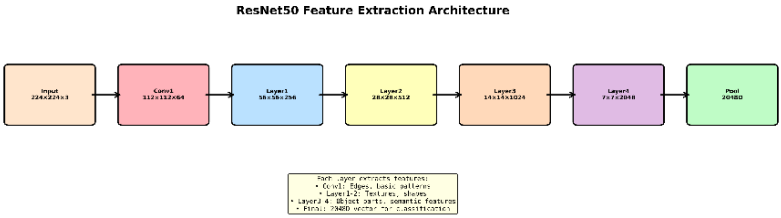
4.3. Hasil Ekstraksi Fitur

Proses ekstraksi fitur merupakan tahapan fundamental dalam penelitian ini yang bertujuan untuk mengubah citra sel darah malaria menjadi representasi numerik yang dapat diproses oleh algoritma klasifikasi. Ekstraksi fitur dilakukan menggunakan arsitektur ResNet50 yang telah dilatih sebelumnya (*pretrained*) pada dataset ImageNet1K_V2. Model ini dipilih karena kemampuannya dalam menangkap representasi fitur hierarkis dari tingkat rendah hingga tingkat tinggi, yang sangat relevan untuk analisis citra medis dengan karakteristik kompleks seperti sel darah malaria.

Implementasi ekstraksi fitur menggunakan dua konfigurasi model ResNet50. Konfigurasi pertama adalah model lengkap yang digunakan untuk keperluan visualisasi dan analisis proses ekstraksi fitur di setiap layer. Konfigurasi kedua adalah model tanpa lapisan *fully connected* (FC), yang khusus digunakan untuk menghasilkan vektor fitur final berukuran 2048 dimensi. Kedua model dijalankan dalam mode evaluasi dengan semua parameter dalam keadaan beku (*frozen*) untuk mempertahankan bobot yang telah dipelajari dari dataset ImageNet, sehingga mampu mengenali pola-pola visual yang general namun tetap aplikatif untuk domain citra medis.

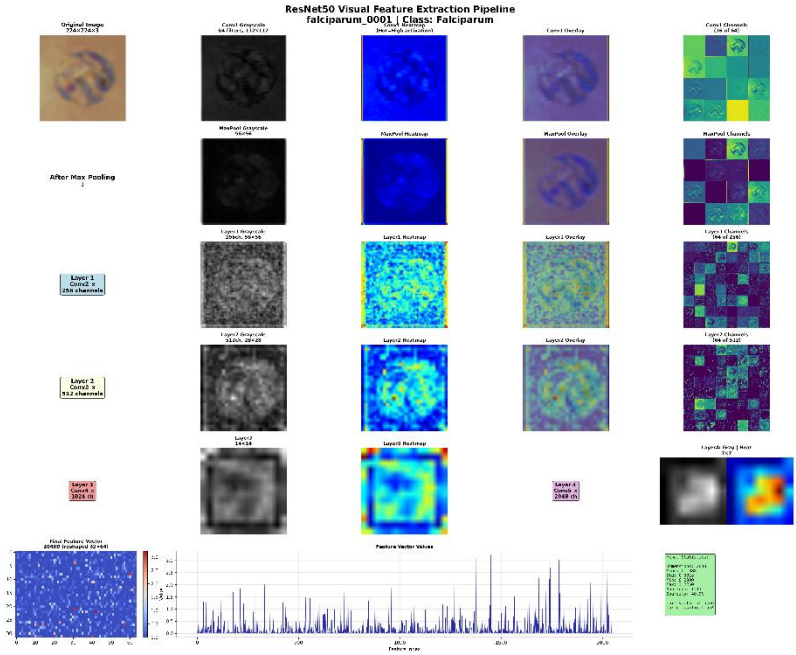
Sebelum diproses oleh model ResNet50, seluruh citra input mengalami preprocessing standar yang konsisten dengan normalisasi ImageNet. Setiap citra diresize ke dimensi 224×224 piksel, dikonversi ke tensor, dan dinormalisasi menggunakan nilai *mean* [0.485, 0.456, 0.406] dan standar deviasi [0.229, 0.224, 0.225] untuk masing-masing kanal RGB. Normalisasi ini penting untuk memastikan distribusi nilai piksel input sesuai dengan distribusi data yang digunakan saat pelatihan model ResNet50, sehingga model dapat menghasilkan fitur yang optimal dan konsisten.

Arsitektur ResNet50 yang digunakan dalam penelitian ini terdiri dari beberapa tahapan pemrosesan secara hierarkis seperti yang divisualisasikan pada Gambar 29. Arsitektur dimulai dengan input citra berukuran 224×224×3 (tinggi × lebar × kanal RGB), yang kemudian diproses melalui layer konvolusi pertama (Conv1) menghasilkan *feature maps* berukuran 112×112×64. Setelah melalui operasi max pooling, dimensi spasial berkurang namun tetap mempertahankan informasi penting. Proses ekstraksi fitur berlanjut melalui empat layer residual utama: Layer 1 (Conv2_x) menghasilkan *feature maps* berukuran 56×56×256 yang menangkap tekstur dan pola menengah, Layer 2 (Conv3_x) menghasilkan 28×28×512 *feature maps* yang mampu mengidentifikasi bentuk dan struktur yang lebih kompleks, Layer 3 (Conv4_x) menghasilkan 14×14×1024 *feature maps* yang menangkap komponen objek dan fitur semantik, dan Layer 4 (Conv5_x) menghasilkan 7×7×2048 *feature maps* yang merepresentasikan fitur semantik tingkat tinggi. Pada tahap akhir arsitektur, dilakukan global average pooling yang mengubah *feature maps* 7×7×2048 menjadi vektor fitur satu dimensi berukuran 2048 dimensi (Pool 2048D). Setiap layer dalam arsitektur ini memiliki peran spesifik: Conv1 mengekstrak edges dan pola dasar, Layer 1-2 menangkap textures dan shapes, Layer 3-4 mengidentifikasi object parts dan semantic features, dan hasil final berupa vektor 2048 dimensi siap untuk tahap klasifikasi.



Gambar 38. Arsitektur ResNet50 untuk Ekstraksi Fitur

Untuk memahami transformasi fitur secara visual dan detail, penelitian ini mengimplementasikan sistem visualisasi komprehensif yang menunjukkan bagaimana ResNet50 mentransformasi informasi visual di setiap layer. Visualisasi dilakukan pada sampel citra sel terinfeksi *Plasmodium Falciparum* yang terdiri dari empat kolom utama untuk setiap layer: *grayScale representation* yang menunjukkan rata-rata aktivasi absolut dari semua kanal, heatmap berwarna menggunakan colormap untuk melihat intensitas aktivasi dengan warna hangat (merah-kuning) menunjukkan *high activation* dan warna dingin (biru-ungu) menunjukkan *low activation*, *overlay* dengan citra asli untuk mengidentifikasi *region* yang paling berkontribusi terhadap ekstraksi fitur, dan individual channels yang menampilkan subset dari *feature maps* untuk melihat variasi aktivasi antar kanal.



Gambar 39. Visualisasi Proses Ekstraksi Fitur ResNet50 pada Sampel Citra *Plasmodium Falciparum*

Hasil visualisasi pada Gambar 39 menunjukkan bahwa pada layer Conv1, deteksi tepi dan kontur sel sangat jelas dengan heatmap mengindikasikan aktivasi tinggi pada batas membran sel. Setelah max pooling, dimensi spasial berkurang dari 112×112 menjadi 56×56 namun informasi struktural tetap terjaga. Visualisasi channels menampilkan 16 dari 64 kanal Conv1, menunjukkan bahwa setiap kanal mendeteksi pola yang berbeda seperti edge horizontal, vertikal, dan diagonal. Layer 1 (Conv2_x dengan 256 channels) menunjukkan aktivasi yang lebih kompleks dengan heatmap mengungkapkan fokus pada area parasit dalam sel, sementara individual channels menampilkan 64 dari 256 kanal yang menunjukkan diversifikasi deteksi pola. Layer 2 (Conv3_x dengan 512 channels) menghasilkan *feature maps* 28×28 yang menangkap informasi semantik menengah, dengan heatmap menunjukkan konsentrasi aktivasi pada region parasit. Layer 3 (Conv4_x dengan 1024 channels) menghasilkan representasi yang lebih abstrak dengan dimensi 14×14, dengan visualisasi menunjukkan aktivasi yang lebih general dan semantik pada keseluruhan morfologi sel terinfeksi. Layer 4 (Conv5_x dengan 2048 channels) menghasilkan *feature maps* 7×7 yang merepresentasikan fitur

semantik tingkat tinggi dengan aktivasi yang sangat abstrak dan tidak lagi berkorespondensi langsung dengan fitur visual spesifik dalam citra asli.

Hasil akhir dari proses ekstraksi ditampilkan dalam bentuk final feature vector berukuran 2048 dimensi yang telah di-*reshape* menjadi grid 32×64 untuk visualisasi. Pola *sparsity* terlihat jelas dimana hanya sebagian kecil dari 2048 dimensi memiliki nilai aktivasi yang signifikan, sementara sebagian besar mendekati nol. Grafik *feature vector values* menampilkan distribusi nilai pada 2048 dimensi fitur dengan statistik: dimensi fitur 2048, nilai minimum 0.0, maksimum 3.8228, *mean* 0.4836, standar deviasi 0.6211, dan *sparsity* 46.58%. Tingkat *sparsity* yang tinggi ini sebenarnya menguntungkan karena mengurangi kompleksitas komputasi pada tahap klasifikasi dan membantu model klasifikasi untuk fokus pada fitur-fitur yang benar-benar diskriminatif.

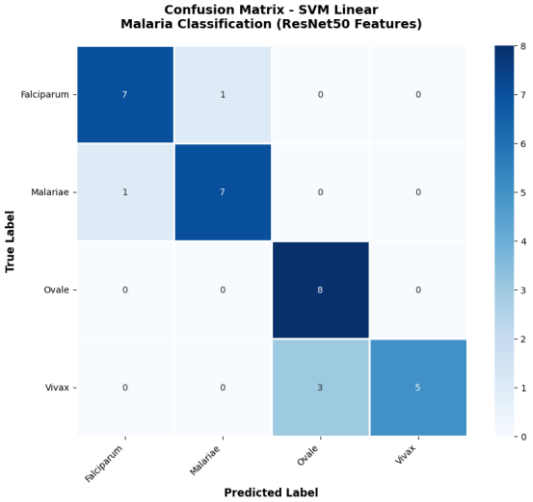
Ekstraksi fitur dilakukan pada dua dataset berbeda: dataset *training* yang telah mengalami augmentasi (Iterasi 80) dan dataset *validation* 2. Proses ekstraksi menggunakan batch size 32 dengan non-blocking transfer data ke GPU untuk optimalisasi kecepatan. Hasil ekstraksi menunjukkan bahwa setiap citra berhasil direpresentasikan sebagai vektor fitur 2048 dimensi dengan distribusi nilai yang bervariasi namun konsisten antara dataset *training* dan *validation*. Seluruh hasil ekstraksi fitur disimpan dalam format NumPy array (.npy) yang mencakup fitur *training*, label *training*, fitur *validation*, label *validation*, nama kelas, dimensi fitur, informasi model yang digunakan, parameter normalisasi, dan *metadata* lainnya. Struktur penyimpanan ini memudahkan proses loading data untuk tahap klasifikasi selanjutnya dan memastikan reprodusibilitas eksperimen. File terpisah juga dibuat untuk features dan labels dari masing-masing dataset, memberikan fleksibilitas dalam eksperimen dengan berbagai konfigurasi klasifikasi.

Hasil visualisasi arsitektur ResNet50 dan proses ekstraksi fitur per layer juga dihasilkan dalam bentuk diagram dan gambar beresolusi tinggi (300 DPI). Visualisasi ini tidak hanya berfungsi sebagai dokumentasi teknis, tetapi juga sebagai alat analisis untuk memahami bagaimana model ResNet50 mentransformasi informasi visual dari citra sel darah menjadi representasi fitur abstrak. Dengan memahami proses transformasi ini, dapat diidentifikasi layer mana yang paling berkontribusi dalam membedakan berbagai jenis infeksi malaria, yang kemudian dapat digunakan untuk optimalisasi lebih lanjut atau interpretasi hasil klasifikasi. Secara keseluruhan, proses ekstraksi fitur menggunakan ResNet50 berhasil menghasilkan representasi fitur yang kaya dan informatif dari citra sel darah malaria. Vektor fitur 2048 dimensi yang dihasilkan mengandung informasi hierarkis mulai dari karakteristik visual tingkat rendah hingga representasi semantik tingkat tinggi, yang sangat penting untuk membedakan berbagai jenis infeksi malaria (Plasmodium *Falciparum*, *Malariae*, *Ovale*, dan *Vivax*) serta sel normal. Fitur-fitur ini selanjutnya akan menjadi input untuk algoritma klasifikasi *Support Vector Machine* dan *Random Forest* pada tahap berikutnya dari penelitian.

4.4. Hasil Implementasi Algoritma Klasifikasi

4.4.1. Hasil Klasifikasi Menggunakan *Support Vector Machine* (SVM)

Implementasi algoritma Support Vector Machine (SVM) dilakukan dengan menggunakan fitur ResNet50 yang telah diekstraksi dari dataset citra malaria. Eksperimen dilakukan dengan 4 konfigurasi kernel SVM yang berbeda untuk menemukan model dengan performa terbaik dalam klasifikasi empat jenis parasit malaria: Plasmodium Falciparum, Plasmodium Malariae, Plasmodium *Ovale*, dan Plasmodium *Vivax* . Untuk memberikan gambaran mengenai performa model terbaik, Gambar 40 menyajikan *confusion matrix* dari model SVM *Linear* yang menunjukkan distribusi prediksi untuk setiap kelas parasit malaria pada data validasi. *Confusion matrix* tersebut memvisualisasikan tingkat akurasi klasifikasi per kelas, dimana diagonal utama menunjukkan prediksi yang benar, sementara elemen di luar diagonal menunjukkan kesalahan klasifikasi antar kelas.



Gambar 40. *Confusion matrix* Klasifikasi SVM *Linear*

Berdasarkan *confusion matrix* pada Gambar 40, dapat diamati bahwa model SVM *Linear* menunjukkan performa klasifikasi yang baik pada data validasi. Model berhasil mengklasifikasikan dengan sempurna untuk kelas *Plasmodium Ovale* (8 dari 8 sampel benar), sementara untuk kelas *Plasmodium Falciparum* dan *Plasmodium Malariae* masing-masing memiliki tingkat akurasi 87.5% (7 dari 8 sampel benar). Kesalahan klasifikasi utama terjadi pada kelas *Plasmodium Vivax* , dimana 3 sampel salah diprediksi sebagai *Plasmodium Ovale* dan 5 sampel diprediksi dengan benar, menghasilkan akurasi 62.5%. Secara keseluruhan, model

SVM *Linear* mencapai akurasi validasi sebesar 84.38% dengan total 27 prediksi benar dari 32 sampel validasi. Untuk memahami secara komprehensif bagaimana hasil klasifikasi tersebut diperoleh, berikut dijelaskan detail metodologi eksperimen yang mencakup konfigurasi dataset, parameter model, dan proses evaluasi yang dilakukan.

1. Dataset dan Konfigurasi Eksperimen

Eksperimen klasifikasi menggunakan SVM dilakukan dengan dataset yang terdiri dari 1.600 sampel data *training* dan 32 sampel data validasi. Data *training* tersebut terbagi secara seimbang ke dalam empat kelas parasit malaria dengan masing-masing kelas memiliki 400 sampel, sehingga proporsi distribusi kelas mencapai 25% untuk setiap kelas. Sementara itu, data validasi terdiri dari 8 sampel untuk setiap kelas parasit. Setiap sampel data direpresentasikan oleh 2.048 fitur yang diekstraksi menggunakan arsitektur *deep learning* ResNet50. Data *training* yang digunakan berasal dari iterasi ke-80 yang terletak pada direktori *train_iteration*/Iterasi 80, sedangkan data validasi diambil dari direktori validasi.

Metode validasi yang digunakan dalam eksperimen ini adalah K-Fold *Cross-Validation* dengan 10 fold yang diulang sebanyak 10 kali, sehingga total eksperimen yang dilakukan mencapai 400 eksperimen (4 konfigurasi × 10 fold × 10 repetisi). Pendekatan ini dipilih untuk memastikan evaluasi yang robust dan komprehensif terhadap performa model. Sebelum digunakan untuk pelatihan, seluruh fitur dilakukan *preprocessing* berupa standarisasi untuk memastikan bahwa setiap fitur memiliki *mean* sebesar 0,000000 dan *standard deviation* sebesar 1,000000. Standarisasi ini penting untuk memastikan bahwa semua fitur memiliki skala yang sama sehingga tidak ada fitur yang mendominasi proses pembelajaran karena perbedaan magnitudo nilai.

Untuk mengukur tingkat kepercayaan prediksi model, penelitian ini mengimplementasikan metode *confidence scoring* dengan pendekatan *Normalized confidence* menggunakan transformasi *Sigmoid* pada skala 0-1. Metode *confidence* yang digunakan adalah *margin-based*, yaitu menghitung gap atau selisih antara skor dua kelas teratas yang kemudian dinormalisasi menggunakan fungsi *Sigmoid*. Pendekatan ini memungkinkan perbandingan yang fair dengan algoritma lain seperti *Random Forest*. Untuk memastikan reproducibility hasil eksperimen, seluruh konfigurasi model menggunakan *random state* yang sama yaitu 42. Dengan konfigurasi eksperimen yang komprehensif dan sistematis ini, diharapkan hasil evaluasi yang diperoleh dapat memberikan gambaran yang akurat dan reliable mengenai performa algoritma SVM dalam mengklasifikasikan parasit malaria.

2. Konfigurasi Model SVM yang Diuji

Empat konfigurasi SVM yang berbeda diuji dalam penelitian ini untuk menemukan kombinasi parameter yang paling optimal. Konfigurasi pertama adalah SVM *Linear* yang menggunakan *kernel Linear* dengan parameter *kernel*='*Linear*' dan *random_state*=42. *Kernel Linear* dipilih karena kemampuannya

dalam menangani data berdimensi tinggi dengan kompleksitas komputasi yang lebih rendah. Konfigurasi kedua adalah SVM RBF (*gamma=Scale*) yang menggunakan *kernel Radial Basis Function* dengan *gamma* otomatis, dengan parameter *kernel*='rbf', *gamma*='*Scale*', dan *random_state*=42. Penggunaan *gamma*='*Scale*' memungkinkan model menyesuaikan parameter *gamma* secara otomatis berdasarkan jumlah fitur dan variansi data. Konfigurasi ketiga adalah SVM *Polynomial* (*degree*=2) yang menggunakan *kernel Polynomial* dengan derajat 2, dengan parameter *kernel*='poly', *degree*=2, dan *random_state*=42. *Kernel Polynomial* memberikan kemampuan untuk menangkap hubungan *non-Linear* dengan kompleksitas yang terkontrol melalui parameter *degree*. Konfigurasi keempat adalah SVM RBF (C=10) yang menggunakan *kernel* RBF dengan parameter regularisasi tinggi, dengan parameter *kernel*='rbf', C=10, *gamma*='*Scale*', dan *random_state*=42. Parameter C=10 yang lebih tinggi dari nilai *default* memungkinkan model lebih *fitting* terhadap data *training* dengan toleransi *error* yang lebih rendah. Semua konfigurasi menggunakan *random_state*=42 yang sama untuk memastikan *reproducibility* hasil eksperimen.

3. Evaluasi pada *Validation* Set

Model terbaik ditentukan berdasarkan hasil *K-Fold Cross-Validation* (10-fold dengan 10 repetisi) menggunakan dua kriteria utama, yaitu *Mean Accuracy* tertinggi sebagai indikator kemampuan klasifikasi secara keseluruhan, serta *Standard Deviation* terendah sebagai indikator konsistensi dan stabilitas performa model di seluruh *fold* pengujian.

Metrik evaluasi lain seperti *Mean F1-Score* menunjukkan pola yang identik dengan *Mean Accuracy* karena dataset yang digunakan bersifat *balanced*, dengan jumlah sampel yang sama pada setiap kelas. Oleh karena itu, metrik tersebut tidak memberikan informasi tambahan yang signifikan dalam proses pemilihan model. Sementara itu, metrik *Normalized Confidence Gap*, *Raw Confidence Gap*, *Normalized Correlation*, dan *Calibrated Confidence Gap* tidak digunakan sebagai dasar seleksi model karena metrik-metrik tersebut dirancang untuk menganalisis karakteristik *confidence scoring* model, yang dibahas secara khusus pada Subbab 4.5.2, bukan untuk mengevaluasi performa klasifikasi.

Penggunaan *confidence metrics* sebagai kriteria pemilihan model berpotensi menimbulkan interpretasi yang keliru, karena nilai *confidence* yang tinggi tidak selalu berkorelasi dengan tingkat akurasi klasifikasi yang optimal. Hal ini tercermin pada salah satu varian model SVM yang menunjukkan karakteristik *confidence* paling kuat, namun menghasilkan performa klasifikasi yang relatif lebih rendah dibandingkan model lainnya.

Berdasarkan kriteria tersebut, SVM *Linear* teridentifikasi sebagai model terbaik. Selanjutnya, model ini dilatih ulang menggunakan seluruh data *training* dan dievaluasi pada *validation set* yang terpisah untuk mengukur kemampuan generalisasi terhadap data yang tidak pernah terlibat dalam proses *K-Fold Cross-Validation*. Hasil evaluasi pada *validation set* disajikan secara rinci pada Tabel 14.

Tabel 14. Performa SVM *Linear* pada *Validation* Set

Metrik Evaluasi	Nilai	Persentase
Akurasi	0,8438	84,38%
Presisi	0,8693	86,93%
Recall	0,8438	84,38%
F1-*Score*	0,8403	84,03%

Hasil evaluasi pada Tabel 14 menunjukkan bahwa model SVM *Linear* mencapai akurasi validasi sebesar 84,38% dengan presisi 86,93%, *recall* 84,38%, dan *F1-Score* 84,03%, yang mengindikasikan kemampuan generalisasi yang baik dengan keseimbangan performa antara kemampuan identifikasi positif dan minimalisasi kesalahan prediksi. Meskipun hasil validasi model SVM *Linear* menunjukkan performa yang memuaskan, perlu dilakukan analisis komprehensif untuk memahami bagaimana konfigurasi tersebut dibandingkan dengan variasi *kernel* SVM lainnya dalam proses *cross-validation*.

Untuk memberikan gambaran menyeluruh mengenai performa berbagai konfigurasi SVM, dilakukan perbandingan sistematis terhadap empat konfigurasi *kernel* yang berbeda menggunakan metode *K-Fold Cross-Validation* dengan 10 *fold* yang diulang sebanyak 10 kali, dengan hasil rata-rata performa masing-masing konfigurasi disajikan pada Tabel 15:

Tabel 15. Pengujian Sistem SVM dengan K-Fold

No	Model	Parameter & Value	K-Fold x Repetisi	Akurasi (%)	Presisi (%)	*Recall* (%)	F1-*Score* (%)
1	SVM	Kernel= *Linear*	10-Fold × 10	99.25	99.27	99.25	99.25
2	SVM	Kernel= RBF, Gamma= *Scale*	10-Fold × 10	98.24	98.35	98.24	98.24
3	SVM	Kernel= poly, *Degree*= 2	10-Fold × 10	98.03	98.12	98.03	98.03
4	SVM	Kernel= rbf, C= 10	10-Fold × 10	98.64	98.72	98.64	98.64

Berdasarkan hasil pada Tabel 15, konfigurasi SVM dengan *kernel Linear* menunjukkan performa terbaik dengan akurasi rata-rata 99,25%, diikuti oleh *kernel* RBF dengan C=10 (98,64%), *kernel* RBF dengan gamma=scale (98,24%), dan *kernel Polynomial* degree=2 (98,03%), yang mengonfirmasi bahwa *kernel Linear* paling sesuai untuk klasifikasi parasit malaria dengan fitur ekstraksi ResNet50.

Setelah mengidentifikasi *kernel Linear* sebagai konfigurasi terbaik berdasarkan akurasi rata-rata, penting untuk menganalisis distribusi detail performa pada setiap *fold* dan repetisi guna memahami tingkat konsistensi dan stabilitas model secara mendalam.

Untuk menunjukkan konsistensi dan stabilitas performa model SVM dengan *kernel Linear* sebagai konfigurasi terbaik, dilakukan pencatatan detail hasil klasifikasi untuk setiap kombinasi *fold* dan repetisi pada proses *K-Fold Cross-Validation*, dengan distribusi lengkap performa ditampilkan pada Tabel 16:

Tabel 16. Pengujian Sistem SVM 10 Uji Parameter Kernel= *Linear*

Fold ke-	Repetisi ke-	Akurasi (%)	Presisi (%)	*Recall* (%)	F1-*Score* (%)
1	1	0,993	0,993	0,993	0,993
	2	1,000	1,000	1,000	1,000
	3	0,993	0,993	0,993	0,993
	4	0,987	0,988	0,987	0,987
	5	0,993	0,993	0,993	0,993
	6	0,993	0,993	0,993	0,993
	7	0,981	0,981	0,981	0,981
	8	0,993	0,993	0,993	0,993
	9	1,000	1,000	1,000	1,000
	10	0,981	0,981	0,981	0,981
2	1	0,993	0,993	0,993	0,993
	2	0,981	0,981	0,981	0,981
	3	0,993	0,993	0,993	0,993
	4	0,993	0,993	0,993	0,993
	5	0,987	0,987	0,987	0,987
	6	0,993	0,993	0,993	0,993
	7	0,993	0,993	0,993	0,993
	8	0,993	0,993	0,993	0,993
	9	0,993	0,993	0,993	0,993
	10	1,000	1,000	1,000	1,000
3	1	0,993	0,993	0,993	0,993
	2	0,981	0,981	0,981	0,981
	3	1,000	1,000	1,000	1,000
	4	0,993	0,993	0,993	0,993
	5	1,000	1,000	1,000	1,000
	6	0,993	0,993	0,993	0,993
	7	0,993	0,993	0,993	0,993
	8	0,981	0,981	0,981	0,981

	9	1,000	1,000	1,000	1,000
	10	1,000	1,000	1,000	1,000
4	1	0,993	0,993	0,993	0,993
	2	0,987	0,987	0,987	0,987
	3	0,993	0,993	0,993	0,993
	4	0,993	0,993	0,993	0,993
	5	0,981	0,982	0,981	0,981
	6	0,993	0,993	0,993	0,993
	7	0,993	0,993	0,993	0,993
	8	0,993	0,993	0,993	0,993
	9	0,993	0,993	0,993	0,993
	10	0,981	0,982	0,981	0,981
5	1	0,993	0,993	0,993	0,993
	2	0,993	0,993	0,993	0,993
	3	1,000	1,000	1,000	1,000
	4	0,993	0,993	0,993	0,993
	5	0,975	0,975	0,975	0,974
	6	0,975	0,975	0,975	0,974
	7	0,981	0,981	0,981	0,981
	8	0,981	0,981	0,981	0,981
	9	0,993	0,993	0,993	0,993
	10	0,993	0,993	0,993	0,993
6	1	1,000	1,000	1,000	1,000
	2	0,987	0,988	0,987	0,987
	3	1,000	1,000	1,000	1,000
	4	0,993	0,993	0,993	0,993
	5	1,000	1,000	1,000	1,000
	6	0,981	0,981	0,981	0,981
	7	1,000	1,000	1,000	1,000
	8	1,000	1,000	1,000	1,000
	9	0,993	0,993	0,993	0,993
	10	0,993	0,99	0,993	0,993
7	1	1,000	1,000	1,000	1,000
	2	1,000	1,000	1,000	1,000
	3	0,987	0,987	0,987	0,987
	4	0,993	0,993	0,993	0,993
	5	0,993	0,993	0,993	0,993
	6	1,000	1,000	1,000	1,000
	7	1,000	1,000	1,000	1,000
	8	0,993	0,993	0,993	0,993

	9	0,981	0,981	0,981	0,981
	10	1,000	1,000	1,000	1,000
8	1	0,993	0,993	0,993	0,993
	2	0,993	0,993	0,993	0,993
	3	0,993	0,993	0,993	0,993
	4	0,981	0,981	0,981	0,981
	5	0,993	0,993	0,993	0,993
	6	0,993	0,993	0,993	0,993
	7	1,000	1,000	1,000	1,000
	8	1,000	1,000	1,000	1,000
	9	0,993	0,993	0,993	0,993
	10	0,987	0,988	0,987	0,987
9	1	1,000	1,000	1,000	1,000
	2	0,993	0,993	0,993	0,993
	3	0,993	0,993	0,993	0,993
	4	0,987	0,987	0,987	0,987
	5	0,993	0,993	0,993	0,993
	6	1,000	1,000	1,000	1,000
	7	0,993	0,993	0,993	0,993
	8	0,987	0,987	0,987	0,987
	9	1,000	1,000	1,000	1,000
	10	0,981	0,981	0,981	0,981
10	1	1,000	1,000	1,000	1,000
	2	0,993	0,993	0,993	0,993
	3	1,000	1,000	1,000	1,000
	4	0,993	0,993	0,993	0,993
	5	0,987	0,987	0,987	0,987
	6	0,968	0,969	0,968	0,968
	7	1,000	1,000	1,000	1,000
	8	0,993	0,993	0,993	0,993
	9	0,987	0,987	0,987	0,987
	10	0,993	0,993	0,993	0,993

Tabel 16 memperlihatkan bahwa dari 100 eksperimen yang dilakukan (10 *fold* × 10 repetisi), model SVM *Linear* menunjukkan konsistensi tinggi dengan akurasi minimum 96,8% dan maksimum 100%, dimana 14 eksperimen mencapai akurasi sempurna, mendemonstrasikan stabilitas algoritma yang sangat baik dalam mengklasifikasikan keempat jenis parasit malaria. Untuk memberikan konteks perbandingan yang lebih komprehensif, perlu dilakukan analisis terhadap distribusi detail performa konfigurasi *kernel* lainnya, khususnya *kernel* RBF dengan gamma=scale sebagai konfigurasi non-*Linear* yang paling umum digunakan.

Sebagai pembanding terhadap konfigurasi *Linear*, performa detail model SVM dengan *kernel* RBF dan parameter gamma=scale dicatat untuk setiap kombinasi *fold* dan repetisi guna menganalisis pola variabilitas dan stabilitas algoritma, dengan hasil lengkap disajikan pada Tabel 17:

Tabel 17. Pengujian Sistem SVM 10 Uji Parameter Kernel= RBF, Gamma= *Scale*

Fold ke-	Repetisi ke-	Akurasi (%)	Presisi (%)	*Recall* (%)	F1-*Score* (%)
1	1	0,981	0,982	0,981	0,981
	2	0,993	0,993	0,993	0,993
	3	0,981	0,981	0,981	0,981
	4	0,993	0,993	0,993	0,993
	5	0,962	0,965	0,962	0,961
	6	0,987	0,988	0,987	0,987
	7	0,981	0,982	0,981	0,981
	8	0,981	0,982	0,981	0,981
	9	0,987	0,988	0,987	0,987
	10	0,981	0,981	0,981	0,981
2	1	0,981	0,982	0,981	0,981
	2	0,968	0,972	0,968	0,969
	3	0,975	0,976	0,975	0,974
	4	0,987	0,987	0,987	0,987
	5	0,987	0,988	0,987	0,987
	6	0,981	0,982	0,981	0,981
	7	0,987	0,987	0,987	0,987
	8	0,981	0,981	0,981	0,981
	9	0,993	0,993	0,993	0,993
	10	0,993	0,993	0,993	0,993
3	1	0,993	0,993	0,993	0,993
	2	0,975	0,977	0,975	0,974
	3	0,981	0,981	0,981	0,981
	4	1,000	1,000	1,000	1,000
	5	1,000	1,000	1,000	1,000
	6	0,981	0,981	0,981	0,981
	7	0,981	0,981	0,981	0,981
	8	0,968	0,972	0,968	0,969
	9	0,962	0,964	0,962	0,962
	10	0,987	0,987	0,987	0,987
4	1	1,000	1,000	1,000	1,000
	2	0,956	0,960	0,956	0,956

	3	0,981	0,981	0,981	0,981
	4	0,981	0,982	0,981	0,981
	5	0,975	0,977	0,975	0,975
	6	1,000	1,000	1,000	1,000
	7	0,981	0,981	0,981	0,981
	8	0,968	0,970	0,968	0,968
	9	0,993	0,993	0,993	0,993
	10	0,975	0,976	0,975	0,974
5	1	0,993	0,993	0,993	0,993
	2	0,975	0,977	0,975	0,975
	3	0,981	0,981	0,981	0,981
	4	1,000	1,000	1,000	1,000
	5	0,968	0,970	0,968	0,968
	6	0,950	0,955	0,950	0,949
	7	0,981	0,981	0,981	0,981
	8	0,981	0,981	0,981	0,981
	9	0,981	0,982	0,981	0,981
	10	0,993	0,993	0,993	0,993
6	1	1,000	1,000	1,000	1,000
	2	0,962	0,965	0,962	0,962
	3	0,987	0,988	0,987	0,987
	4	0,987	0,988	0,987	0,987
	5	1,000	1,000,	1,000	1,000
	6	0,993	0,993	0,993	0,993
	7	0,937	0,947	0,937	0,938
	8	0,993	0,993	0,993	0,993
	9	0,981	0,982	0,981	0,981
	10	0,975	0,976	0,975	0,974
7	1	0,981	0,981	0,981	0,981
	2	0,987	0,988	0,987	0,987
	3	0,987	0,988	0,987	0,987
	4	0,987	0,988	0,987	0,987
	5	0,993	0,993	0,993	0,993
	6	0,987	0,987	0,987	0,987
	7	0,987	0,987	0,987	0,987
	8	0,968	0,972	0,968	0,969
	9	0,981	0,981	0,981	0,981
	10	0,981	0,982	0,981	0,981
8	1	0,968	0,970	0,968	0,968
	2	0,975	0,976	0,975	0,974

	3	0,987	0,987	0,987	0,987
	4	0,987	0,988	0,987	0,987
	5	0,968	0,970	0,968	0,968
	6	0,993	0,993	0,993	0,993
	7	0,987	0,987	0,987	0,987
	8	1,000	1,000	1,000	1,000
	9	0,993	0,993	0,993	0,993
	10	0,950	0,954	0,950	0,949
9	1	0,962	0,967	0,962	0,963
	2	0,962	0,967	0,962	0,963
	3	0,993	0,993	0,993	0,993
	4	0,993	0,993	0,993	0,993
	5	0,993	0,993	0,993	0,993
	6	0,981	0,981	0,981	0,981
	7	0,987	0,988	0,987	0,987
	8	0,981	0,981	0,981	0,981
	9	0,975	0,976	0,975	0,974
	10	0,987	0,987	0,987	0,987
10	1	0,968	0,970	0,968	0,969
	2	0,975	0,977	0,975	0,974
	3	0,993	0,993	0,993	0,993
	4	0,987	0,988	0,987	0,987
	5	0,987	0,988	0,987	0,987
	6	0,975	0,976	0,975	0,974
	7	0,993	0,993	0,993	0,993
	8	0,987	0,987	0,987	0,987
	9	0,968	0,971	0,968	0,968
	10	0,987	0,988	0,987	0,987

Hasil pada Tabel 17 menunjukkan bahwa konfigurasi SVM RBF dengan gamma=scale menghasilkan rentang akurasi yang lebih lebar (93,7%-100%) dibandingkan konfigurasi *Linear*, dengan 6 eksperimen mencapai akurasi sempurna namun juga terdapat eksperimen dengan akurasi di bawah 95%, mengindikasikan sensitivitas *kernel* RBF yang lebih tinggi terhadap variasi distribusi data antar *fold*. Selain *kernel* RBF, eksplorasi terhadap *kernel Polynomial* juga penting dilakukan untuk memahami bagaimana pendekatan non-*Linear* alternatif dapat menangkap pola kompleks dalam data klasifikasi parasit malaria.

Untuk mengeksplorasi kemampuan *kernel* non-*Linear* dalam menangkap hubungan kompleks antar fitur, dilakukan evaluasi komprehensif terhadap konfigurasi SVM dengan *kernel Polynomial* degree=2 pada seluruh kombinasi *fold* dan repetisi, dengan distribusi performa ditampilkan pada Tabel 18:

Tabel 18. Pengujian Sistem SVM 50 Uji Parameter Kernel= Poly, Degree= 2

Fold ke-	Repetisi ke-	Akurasi (%)	Presisi (%)	*Recall* (%)	F1-*Score* (%)
1	1	0,993	0,993	0,993	0,993
	2	0,981	0,982	0,981	0,981
	3	0,962	0,963	0,962	0,962
	4	0,993	0,993	0,993	0,993
	5	0,981	0,981	0,981	0,981
	6	0,987	0,987	0,987	0,987
	7	0,987	0,988	0,987	0,987
	8	0,975	0,975	0,975	0,974
	9	0,981	0,982	0,981	0,981
	10	0,981	0,981	0,981	0,981
2	1	0,981	0,981	0,981	0,981
	2	0,943	0,949	0,943	0,944
	3	0,987	0,988	0,987	0,987
	4	0,987	0,987	0,987	0,987
	5	0,981	0,981	0,981	0,981
	6	0,993	0,993	0,993	0,993
	7	0,968	0,972	0,968	0,968
	8	0,975	0,975	0,975	0,974
	9	0,993	0,993	0,993	0,993
	10	0,968	0,969	0,968	0,968
3	1	0,981	0,981	0,981	0,981
	2	0,968	0,970	0,968	0,968
	3	0,993	0,993	0,993	0,993
	4	0,993	0,993	0,993	0,993
	5	0,975	0,977	0,975	0,974
	6	0,987	0,987	0,987	0,987
	7	0,975	0,975	0,975	0,974
	8	0,987	0,988	0,987	0,987
	9	0,950	0,953	0,950	0,949
	10	0,981	0,982	0,981	0,981
4	1	0,981	0,982	0,981	0,981
	2	0,975	0,975	0,975	0,974
	3	0,968	0,970	0,968	0,968
	4	0,981	0,981	0,981	0,981
	5	1,000	1,000	1,000	1,000
	6	1,000	1,000	1,000	1,000

	7	0,981	0,981	0,981	0,981
	8	0,981	0,981	0,981	0,981
	9	0,975	0,977	0,975	0,974
	10	0,956	0,958	0,956	0,956
5	1	0,981	0,981	0,981	0,981
	2	0,993	0,993	0,993	0,993
	3	0,968	0,970	0,968	0,968
	4	0,981	0,981	0,981	0,981
	5	0,987	0,987	0,987	0,987
	6	0,968	0,969	0,968	0,968
	7	0,968	0,970	0,968	0,968
	8	0,975	0,976	0,975	0,975
	9	0,981	0,981	0,981	0,981
	10	0,987	0,987	0,987	0,987
6	1	0,975	0,977	0,975	0,974
	2	0,975	0,975	0,975	0,974
	3	1,000	1,000	1,000	1,000
	4	0,987	0,987	0,987	0,987
	5	0,968	0,972	0,968	0,968
	6	0,975	0,976	0,975	0,974
	7	0,975	0,975	0,975	0,974
	8	0,981	0,981	0,981	0,981
	9	0,993	0,993	0,993	0,993
	10	0,950	0,951	0,950	0,950
7	1	0,968	0,970	0,968	0,968
	2	0,975	0,975	0,975	0,974
	3	0,987	0,987	0,987	0,987
	4	0,993	0,993	0,993	0,993
	5	0,993	0,993	0,993	0,993
	6	0,975	0,975	0,975	0,974
	7	0,993	0,993	0,993	0,993
	8	0,987	0,988	0,987	0,987
	9	0,975	0,976	0,975	0,974
	10	0,987	0,987	0,987	0,98
8	1	0,981	0,982	0,981	0,981
	2	0,993	0,993	0,993	0,993
	3	0,987	0,988	0,987	0,987
	4	0,962	0,962	0,962	0,962
	5	0,981	0,981	0,981	0,981
	6	0,975	0,975	0,975	0,974

	7	0,981	0,982	0,981	0,981
	8	0,993	0,993	0,993	0,993
	9	1,000	1,000	1,000	1,000
	10	0,950	0,950	0,950	0,949
9	1	0,987	0,987	0,987	0,987
	2	0,962	0,965	0,962	0,962
	3	0,987	0,987	0,987	0,987
	4	0,981	0,981	0,981	0,981
	5	1,000	1,000	1,000	1,000
	6	0,987	0,987	0,987	0,987
	7	0,975	0,976	0,975	0,974
	8	0,968	0,969	0,968	0,968
	9	0,987	0,987	0,987	0,987
	10	0,975	0,976	0,975	0,974
10	1	0,968	0,968	0,968	0,968
	2	0,956	0,958	0,956	0,956
	3	0,975	0,976	0,975	0,974
	4	0,975	0,976	0,975	0,974
	5	0,993	0,993	0,993	0,993
	6	0,981	0,981	0,981	0,981
	7	0,993	0,993	0,993	0,993
	8	0,987	0,988	0,987	0,987
	9	0,981	0,981	0,981	0,981
	10	0,987	0,988	0,987	0,987

Tabel 18 memperlihatkan bahwa konfigurasi SVM *Polynomial* degree=2 menghasilkan performa dengan variabilitas sedang, dimana akurasi berkisar antara 94,3% hingga 100% dengan 4 eksperimen mencapai akurasi sempurna, menunjukkan bahwa meskipun *kernel Polynomial* mampu menangkap pola non-*Linear*, stabilitasnya masih berada di bawah konfigurasi *Linear* untuk kasus klasifikasi parasit malaria. Mengingat variabilitas yang ditunjukkan oleh *kernel* RBF standar, dilakukan investigasi lebih lanjut terhadap pengaruh parameter regularisasi untuk mengevaluasi apakah peningkatan nilai C dapat meningkatkan stabilitas dan konsistensi prediksi.

Untuk menginvestigasi pengaruh parameter regularisasi terhadap performa klasifikasi, dilakukan pengujian model SVM dengan *kernel* RBF dan nilai C=10 yang lebih tinggi dari *default* pada seluruh kombinasi *fold* dan repetisi, dengan hasil detail performa disajikan pada Tabel 19:

Tabel 19. Pengujian Sistem SVM 10 Uji Parameter Kernel= RBF, C= 10

Fold ke-	Repetisi ke-	Akurasi (%)	Presisi (%)	*Recall* (%)	F1-*Score* (%)
1	1	0,981	0,982	0,981	0,981
	2	0,993	0,993	0,993	0,993
	3	1,000	1,000	1,000	1,000
	4	0,993	0,993	0,993	0,993
	5	0,968	0,971	0,968	0,968
	6	0,987	0,988	0,987	0,987
	7	0,987	0,988	0,987	0,987
	8	0,981	0,982	0,981	0,981
	9	0,987	0,988	0,987	0,987
	10	0,993	0,993	0,993	0,993
2	1	0,987	0,988	0,987	0,987
	2	0,968	0,972	0,968	0,969
	3	0,987	0,988	0,987	0,987
	4	0,987	0,987	0,987	0,987
	5	0,987	0,988	0,987	0,987
	6	0,981	0,982	0,981	0,981
	7	0,993	0,993	0,993	0,993
	8	0,981	0,981	0,981	0,981
	9	0,993	0,993	0,993	0,993
	10	0,993	0,993	0,993	0,993
3	1	0,993	0,993	0,993	0,993
	2	0,981	0,982	0,981	0,981
	3	0,987	0,988	0,987	0,987
	4	1,000	1,000	1,000	1,000
	5	1,000	1,000	1,000	1,000
	6	0,987	0,987	0,987	0,987
	7	0,981	0,981	0,981	0,981
	8	0,975	0,977	0,975	0,975
	9	0,975	0,977	0,975	0,975
	10	0,993	0,993	0,993	0,993
4	1	1,000	1,000	1,000	1,000
	2	0,975	0,977	0,975	0,975
	3	0,981	0,981	0,981	0,981
	4	0,981	0,982	0,981	0,981
	5	0,975	0,977	0,975	0,975
	6	1,000	1,000	1,000	1,000
	7	0,993	0,993	0,993	0,993
	8	0,975	0,976	0,975	0,974

	9	1,000	1,000	1,000	1,000
	10	0,987	0,988	0,987	0,987
5	1	1,000	1,000	1,000	1,000
	2	0,975	0,977	0,975	0,975
	3	0,993	0,993	0,993	0,993
	4	1,000	1,000	1,000	1,000
	5	0,975	0,976	0,975	0,975
	6	0,950	0,955	0,950	0,949
	7	0,987	0,987	0,987	0,987
	8	0,975	0,975	0,975	0,974
	9	0,981	0,982	0,981	0,981
	10	0,993	0,993	0,993	0,993
6	1	1,000	1,000	1,000	1,000
	2	0,962	0,965	0,962	0,962
	3	0,987	0,988	0,987	0,987
	4	0,987	0,988	0,987	0,987
	5	1,000	1,000	1,000	1,000
	6	0,993	0,993	0,993	0,993
	7	0,956	0,960	0,956	0,956
	8	0,993	0,993	0,993	0,993
	9	0,981	0,982	0,981	0,981
	10	0,993	0,993	0,993	0,993
7	1	0,987	0,987	0,987	0,987
	2	0,993	0,993	0,993	0,993
	3	0,987	0,988	0,987	0,987
	4	0,981	0,982	0,981	0,981
	5	0,993	0,993	0,993	0,993
	6	0,993	0,993	0,993	0,993
	7	0,993	0,993	0,993	0,993
	8	0,981	0,982	0,981	0,981
	9	0,981	0,981	0,981	0,981
	10	0,981	0,982	0,981	0,981
8	1	0,981	0,982	0,981	0,981
	2	0,981	0,982	0,981	0,981
	3	0,993	0,993	0,993	0,993
	4	0,987	0,988	0,987	0,987
	5	0,981	0,982	0,981	0,981
	6	0,993	0,993	0,993	0,993
	7	0,993	0,993	0,993	0,993
	8	1,000	1,000	1,000	1,000

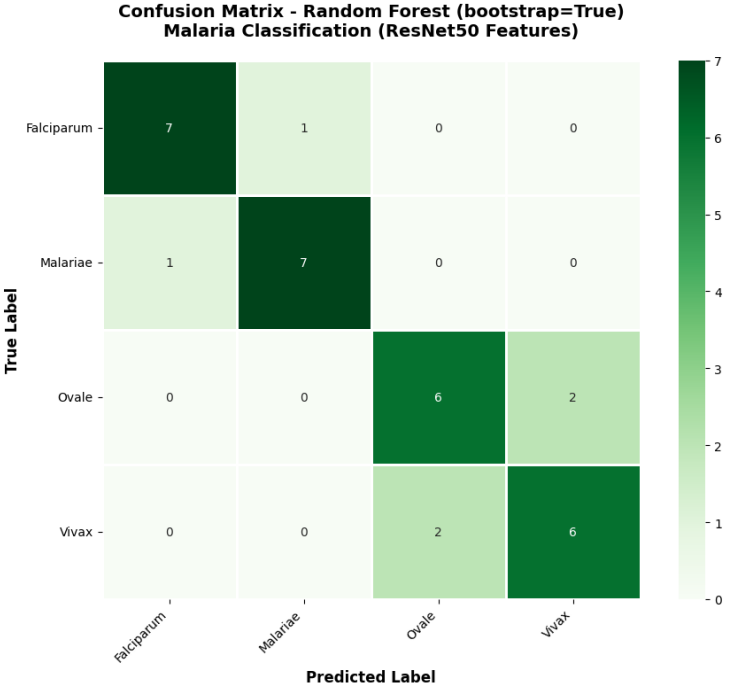
	9	0,987	0,987	0,987	0,987
	10	0,968	0,970	0,968	0,968
9	1	0,968	0,972	0,968	0,969
	2	0,962	0,967	0,962	0,963
	3	0,993	0,993	0,993	0,993
	4	0,993	0,993	0,993	0,993
	5	1,000	1,000	1,000	1,000
	6	0,993	0,993	0,993	0,993
	7	0,987	0,988	0,987	0,987
	8	0,987	0,987	0,987	0,987
	9	0,981	0,982	0,981	0,981
	10	0,987	0,987	0,987	0,987
10	1	0,981	0,982	0,981	0,981
	2	0,993	0,993	0,993	0,993
	3	0,993	0,993	0,993	0,993
	4	0,993	0,993	0,993	0,993
	5	0,987	0,988	0,987	0,987
	6	0,975	0,976	0,975	0,974
	7	0,993	0,993	0,993	0,993
	8	0,993	0,993	0,993	0,993
	9	0,968	0,971	0,968	0,968
	10	0,987	0,988	0,987	0,987

Berdasarkan Tabel 19, peningkatan parameter C menjadi 10 pada konfigurasi SVM RBF menghasilkan performa yang lebih stabil dibandingkan konfigurasi RBF standar dengan rentang akurasi 95,0%-100%, mendemonstrasikan bahwa toleransi error yang lebih rendah dapat meningkatkan konsistensi prediksi meskipun tetap menunjukkan variabilitas lebih tinggi dibandingkan kernel *Linear*.

4.4.2. Hasil Klasifikasi Menggunakan *Random Forest* (RF)

Implementasi algoritma Random Forest (RF) dilakukan dengan menggunakan fitur ResNet50 yang telah diekstraksi dari dataset citra malaria. Eksperimen dilakukan dengan 4 konfigurasi parameter Random Forest yang berbeda untuk menemukan model dengan performa terbaik dalam klasifikasi empat jenis parasit malaria: Plasmodium Falciparum, Plasmodium Malariae, Plasmodium *Ovale*, dan Plasmodium *Vivax* . Untuk memberikan gambaran mengenai performa model terbaik, Gambar 41 menyajikan *confusion matrix* dari model Random Forest dengan parameter bootstrap=True yang menunjukkan distribusi prediksi untuk setiap kelas parasit malaria pada data validasi. *Confusion matrix* tersebut memvisualisasikan tingkat akurasi klasifikasi per kelas, dimana diagonal utama

menunjukkan prediksi yang benar, sementara elemen di luar diagonal menunjukkan kesalahan klasifikasi antar kelas.



Gambar 41. *Confusion matrix* Klasifikasi *Random Forest* bootstrap=*True*

Berdasarkan *confusion matrix* pada Gambar 41, dapat diamati bahwa model Random Forest dengan parameter bootstrap=True menunjukkan performa klasifikasi yang baik pada data validasi. Model berhasil mengklasifikasikan dengan tingkat akurasi tinggi untuk kelas Plasmodium Falciparum dan Plasmodium Malariae, masing-masing dengan akurasi 87.5% (7 dari 8 sampel benar). Untuk kelas Plasmodium *Ovale*, model mencapai akurasi 75% dengan 6 sampel diprediksi benar dan 2 sampel salah diklasifikasikan sebagai Plasmodium *Vivax* . Kesalahan klasifikasi yang serupa juga terjadi pada kelas Plasmodium *Vivax* , dimana 2 sampel salah diprediksi sebagai Plasmodium *Ovale* dan 6 sampel diprediksi dengan benar, menghasilkan akurasi 75%. Secara keseluruhan, model Random Forest mencapai akurasi validasi sebesar 81.25% dengan total 26 prediksi benar dari 32 sampel validasi, dengan pola kesalahan utama terjadi pada confusion antara kelas *Ovale* dan *Vivax* yang mengindikasikan kemiripan karakteristik visual kedua spesies parasit tersebut. Untuk memahami secara komprehensif bagaimana hasil

klasifikasi tersebut diperoleh, berikut dijelaskan detail metodologi eksperimen Random Forest yang mencakup konfigurasi dataset, parameter model, dan proses evaluasi yang dilakukan.

1. Dataset dan Konfigurasi Eksperimen

Eksperimen klasifikasi menggunakan *Random Forest* dilakukan dengan dataset yang terdiri dari 1.600 sampel data *training* dan 32 sampel data validasi. Data *training* tersebut terbagi secara seimbang ke dalam empat kelas parasit malaria dengan masing-masing kelas memiliki 400 sampel, sehingga proporsi distribusi kelas mencapai 25% untuk setiap kelas. Sementara itu, data validasi terdiri dari 8 sampel untuk setiap kelas parasit. Setiap sampel data direpresentasikan oleh 2.048 fitur yang diekstraksi menggunakan arsitektur *deep learning* ResNet50. Data *training* yang digunakan berasal dari iterasi ke-80 yang terletak pada direktori *train_iteration*/Iterasi 80, sedangkan data validasi diambil dari direktori *validation*

Metode validasi yang digunakan dalam eksperimen ini adalah K-Fold *Cross-Validation* dengan 10 fold yang diulang sebanyak 10 kali, sehingga total eksperimen yang dilakukan mencapai 400 eksperimen (4 konfigurasi × 10 fold × 10 repetisi). Pendekatan ini dipilih untuk memastikan evaluasi yang robust dan komprehensif terhadap performa model. Sebelum digunakan untuk pelatihan, seluruh fitur dilakukan preprocessing berupa standarisasi untuk memastikan bahwa setiap fitur memiliki *mean* sebesar 0,000000 dan standard deviation sebesar 1,000000. Standarisasi ini penting untuk memastikan bahwa semua fitur memiliki skala yang sama sehingga tidak ada fitur yang mendominasi proses pembelajaran karena perbedaan magnitude nilai.

Untuk mengukur tingkat kepercayaan prediksi model, penelitian ini mengimplementasikan metode *confidence scoring* dengan pendekatan *improved confidence* menggunakan probability-based method dengan kalibrasi. Metode *confidence* yang digunakan adalah probability margin-based, yaitu menghitung gap atau selisih antara probabilitas dua kelas teratas. Pendekatan ini memanfaatkan distribusi probabilitas yang secara inheren sudah dalam skala [0-1] dari *Random Forest*. Untuk memastikan reproducibility hasil eksperimen, seluruh konfigurasi model menggunakan random state yang sama yaitu 42, dan parameter n_jobs diset ke -1 untuk memanfaatkan semua core processor yang tersedia. Dengan konfigurasi eksperimen yang komprehensif dan sistematis ini, diharapkan hasil evaluasi yang diperoleh dapat memberikan gambaran yang akurat dan reliable mengenai performa algoritma *Random Forest* dalam mengklasifikasikan parasit malaria.

2. Konfigurasi Model *Random Forest* yang Diuji

Empat konfigurasi *Random Forest* yang berbeda diuji dalam penelitian ini untuk mengidentifikasi parameter yang memberikan performa klasifikasi terbaik. Konfigurasi pertama menggunakan *bootstrap sampling* (*bootstrap*=True) dengan parameter *bootstrap*=True, *random_state*=42, dan *n_jobs*=-1, yang memungkinkan setiap *tree* dilatih dengan *subset* data berbeda melalui *random*

sampling with replacement untuk meningkatkan diversitas *ensemble*. Konfigurasi kedua membatasi kedalaman maksimal pohon pada level 15 (*max_depth*=15) dengan parameter *max_depth*=15, *random_state*=42, dan *n_jobs*=-1 untuk mengontrol kompleksitas model dan mencegah *overfitting*. Konfigurasi ketiga menetapkan jumlah *decision trees* sebanyak 100 (*n_estimators*=100) dengan parameter *n_estimators*=100, *random_state*=42, dan *n_jobs*=-1, di mana jumlah *tree* yang lebih banyak dapat meningkatkan akurasi melalui agregasi prediksi yang lebih *robust*. Konfigurasi keempat menggunakan *Gini impurity* sebagai kriteria pemisahan *node* (*criterion*=*gini*) dengan parameter *criterion*='gini', *random_state*=42, dan *n_jobs*=-1, yang mengukur tingkat ketidakmurnian dalam *split* untuk menentukan fitur terbaik. Seluruh konfigurasi menggunakan *random_state*=42 untuk *reproducibility* dan *n_jobs*=-1 untuk paralelisasi komputasi menggunakan semua *core processor* yang tersedia.

3. Evaluasi pada *Validation* Set

Model terbaik (*Random Forest* dengan *bootstrap*=*True*) kemudian dievaluasi pada *validation* set yang terpisah untuk mengukur kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya. Model Random Forest dengan konfigurasi bootstrap=True yang menunjukkan performa terbaik pada tahap *cross-validation* kemudian dievaluasi pada *validation set* yang terdiri dari 32 sampel untuk mengukur kemampuan generalisasi algoritma *ensemble learning* terhadap data baru, dengan hasil evaluasi ditunjukkan pada Tabel 20:

Tabel 20. Performa *Random Forest* (*bootstrap*=*True*) pada *Validation* Set

Metrik Evaluasi	Nilai	Persentase
Akurasi	0,8125	81,25%
Presisi	0,8125	81,25%
Recall	0,8125	81,25%
F1-*Score*	0,8125	81,25%

Hasil pada Tabel 20 menunjukkan bahwa model Random Forest dengan bootstrap=True mencapai akurasi validasi sebesar 81,25% dengan keseimbangan sempurna antara metrik presisi, *recall*, dan *F1-Score* (semua bernilai 81,25%), mengindikasikan performa yang tidak bias terhadap kelas tertentu meskipun akurasi validasinya lebih rendah 3,13% dibandingkan model SVM *Linear*. Untuk memahami secara menyeluruh bagaimana performa model Random Forest dengan konfigurasi bootstrap=True dibandingkan dengan variasi parameter lainnya, diperlukan analisis komparatif terhadap berbagai konfigurasi Random Forest yang diuji dalam penelitian ini.

Untuk mengidentifikasi konfigurasi parameter optimal algoritma Random Forest, dilakukan perbandingan sistematis terhadap empat konfigurasi berbeda menggunakan metode *K-Fold Cross-Validation* dengan 10 *fold* yang diulang

sebanyak 10 kali, dengan ringkasan performa rata-rata masing-masing konfigurasi disajikan pada Tabel 21:

Tabel 21. Pengujian *Random Forest*

No	Model	Parameter & Value	K-Fold x Repetisi	Akurasi (%)	Presisi (%)	*Recall* (%)	F1-*Score* (%)
1	*Random Forest*	bootstrap= True	10-Fold × 10	97.50	97.50	97.50	97.49
2	*Random Forest*	Max_depth= 15	10-Fold × 10	97.50	97.52	97.50	97.50
3	*Random Forest*	n_estimators= 100	10-Fold × 10	97.50	97.50	97.50	97.49
4	*Random Forest*	Criterion= Gini	10-Fold × 10	97.50	97.52	97.50	97.50

Tabel 21 memperlihatkan bahwa keempat konfigurasi Random Forest menghasilkan performa yang sangat seimbang dengan akurasi rata-rata identik sebesar 97,50%, dimana konfigurasi max_depth=15 dan criterion=gini menunjukkan sedikit keunggulan pada metrik presisi (97,52%), mendemonstrasikan karakteristik *ensemble learning* yang stabil namun dengan performa rata-rata 1,75% di bawah SVM *Linear*. Meskipun performa rata-rata keempat konfigurasi menunjukkan keseimbangan yang tinggi, analisis distribusi detail pada setiap *fold* dan repetisi diperlukan untuk memahami pola variabilitas dan konsistensi prediksi, khususnya untuk konfigurasi bootstrap=True sebagai konfigurasi standar Random Forest.

Untuk menunjukkan pola distribusi performa dan tingkat konsistensi algoritma Random Forest dengan konfigurasi bootstrap=True, dilakukan pencatatan detail hasil klasifikasi untuk setiap kombinasi *fold* dan repetisi pada proses *K-Fold Cross-Validation*, dengan hasil lengkap ditampilkan pada Tabel 22:

Tabel 22. Pengujian Sistem RF 10 Uji Parameter Boostrap= True

Fold ke-	Repetisi ke-	Akurasi (%)	Presisi (%)	*Recall* (%)	F1-*Score* (%)
1	1	1,000	1,000	1,000	1,000
	2	0,975	0,975	0,975	0,974
	3	0,968	0,969	0,968	0,968
	4	0,993	0,993	0,993	0,993
	5	0,975	0,974	0,975	0,974
	6	0,956	0,957	0,956	0,955
	7	1,000	1,000	1,000	1,000
	8	0,987	0,988	0,987	0,987

	9	0,981	0,981	0,981	0,981
	10	0,956	0,956	0,956	0,955
2	1	0,975	0,976	0,975	0,974
	2	0,981	0,981	0,981	0,981
	3	0,975	0,975	0,975	0,974
	4	0,975	0,975	0,975	0,974
	5	0,975	0,976	0,975	0,974
	6	0,987	0,987	0,987	0,987
	7	0,987	0,988	0,987	0,987
	8	0,981	0,981	0,981	0,981
	9	1,000	1,000	1,000	1,000
	10	0,981	0,981	0,981	0,981
3	1	0,981	0,981	0,981	0,981
	2	0,968	0,969	0,968	0,968
	3	0,987	0,988	0,987	0,987
	4	0,987	0,987	0,987	0,987
	5	0,975	0,976	0,975	0,974
	6	0,981	0,981	0,981	0,981
	7	0,975	0,974	0,975	0,974
	8	0,981	0,981	0,981	0,981
	9	0,987	0,987	0,987	0,987
	10	1,000	1,000	1,000	1,000
4	1	0,987	0,988	0,987	0,987
	2	0,962	0,962	0,962	0,962
	3	0,981	0,981	0,981	0,981
	4	0,981	0,981	0,981	0,981
	5	1,000	1,000	1,000	1,000
	6	0,993	0,993	0,993	0,993
	7	0,987	0,987	0,987	0,987
	8	0,987	0,987	0,987	0,987
	9	0,968	0,970	0,968	0,968
	10	0,987	0,987	0,987	0,987
5	1	0,987	0,987	0,987	0,987
	2	0,987	0,987	0,987	0,987
	3	0,975	0,975	0,975	0,974
	4	0,987	0,988	0,987	0,987
	5	0,993	0,993	0,993	0,993
	6	0,956	0,958	0,956	0,956
	7	0,950	0,950	0,950	0,949
	8	0,956	0,958	0,956	0,956

	9	0,981	0,981	0,981	0,981
	10	1,000	1,000	1,000	1,000
6	1	0,975	0,976	0,975	0,975
	2	0,968	0,972	0,968	0,968
	3	0,987	0,987	0,987	0,987
	4	0,993	0,993	0,993	0,993
	5	0,975	0,975	0,975	0,974
	6	0,956	0,956	0,956	0,956
	7	0,981	0,981	0,981	0,981
	8	0,987	0,987	0,987	0,987
	9	0,987	0,987	0,987	0,987
	10	0,975	0,975	0,975	0,974
7	1	0,962	0,962	0,962	0,962
	2	0,987	0,987	0,987	0,987
	3	0,981	0,981	0,981	0,981
	4	0,987	0,988	0,987	0,987
	5	0,981	0,982	0,981	0,981
	6	0,975	0,975	0,975	0,975
	7	0,987	0,987	0,987	0,987
	8	0,981	0,981	0,981	0,981
	9	0,975	0,975	0,975	0,974
	10	0,993	0,993	0,993	0,993
8	1	0,993	0,993	0,993	0,993
	2	0,968	0,971	0,968	0,968
	3	0,962	0,962	0,962	0,962
	4	0,993	0,993	0,993	0,993
	5	0,993	0,993	0,993	0,993
	6	0,987	0,987	0,987	0,987
	7	0,981	0,982	0,981	0,981
	8	1,000	1,000	1,000	1,000
	9	0,987	0,987	0,987	0,987
	10	0,956	0,956	0,956	0,956
9	1	0,993	0,993	0,993	0,993
	2	0,993	0,993	0,993	0,993
	3	0,981	0,981	0,981	0,981
	4	0,993	0,993	0,993	0,993
	5	0,981	0,981	0,981	0,981
	6	0,987	0,987	0,987	0,987
	7	0,981	0,981	0,981	0,981
	8	0,962	0,964	0,962	0,962

	9	0,987	0,987	0,987	0,987
	10	0,981	0,981	0,981	0,981
10	1	0,987	0,987	0,987	0,987
	2	0,975	0,975	0,975	0,974
	3	0,987	0,988	0,987	0,987
	4	0,981	0,981	0,981	0,981
	5	0,981	0,981	0,981	0,981
	6	0,975	0,975	0,975	0,975
	7	0,987	0,987	0,987	0,987
	8	0,981	0,981	0,981	0,981
	9	0,975	0,975	0,975	0,974
	10	0,981	0,981	0,981	0,981

Berdasarkan Tabel 22, dari 100 eksperimen yang dilakukan, model Random Forest dengan bootstrap=True menunjukkan rentang akurasi antara 95,0% hingga 100% dengan 8 eksperimen mencapai akurasi sempurna, dimana variabilitas performa yang lebih tinggi dibandingkan SVM *Linear* mencerminkan karakteristik *ensemble learning* yang bergantung pada *random sampling* dan agregasi prediksi dari *multiple decision trees*. Untuk mengevaluasi pendekatan alternatif dalam mengontrol kompleksitas model Random Forest, dilakukan analisis terhadap konfigurasi dengan pembatasan kedalaman pohon yang bertujuan untuk mengurangi risiko *overfitting* sambil mempertahankan akurasi prediksi.

Untuk menganalisis dampak pembatasan kedalaman pohon terhadap performa klasifikasi, dilakukan evaluasi komprehensif model Random Forest dengan parameter max_depth=15 pada seluruh kombinasi *fold* dan repetisi, dengan distribusi detail performa disajikan pada Tabel 23:

Tabel 23. Pengujian Sistem RF 50 Uji Parameter Max_depth= 15

Fold ke-	Repetisi ke-	Akurasi (%)	Presisi (%)	*Recall* (%)	F1-*Score* (%)
1	1	0,993	0,993	0,993	0,993
	2	0,981	0,981	0,981	0,981
	3	0,962	0,963	0,962	0,962
	4	0,987	0,988	0,987	0,987
	5	0,975	0,975	0,975	0,974
	6	0,975	0,977	0,975	0,974
	7	1,000	1,000	1,000	1,000
	8	0,987	0,988	0,987	0,987
	9	0,975	0,975	0,975	0,974
	10	0,968	0,969	0,968	0,968
2	1	0,987	0,987	0,987	0,987

	2	0,975	0,975	0,975	0,974
	3	0,968	0,969	0,968	0,968
	4	0,975	0,975	0,975	0,974
	5	0,987	0,988	0,987	0,987
	6	0,981	0,981	0,981	0,981
	7	0,981	0,981	0,981	0,981
	8	0,975	0,975	0,975	0,974
	9	1,000	1,000	1,000	1,000
	10	0,962	0,963	0,962	0,962
3	1	0,987	0,987	0,987	0,987
	2	0,962	0,963	0,962	0,962
	3	0,981	0,981	0,981	0,981
	4	0,987	0,987	0,987	0,987
	5	0,975	0,976	0,975	0,974
	6	0,987	0,987	0,987	0,987
	7	0,981	0,981	0,981	0,981
	8	0,981	0,981	0,981	0,981
	9	0,987	0,987	0,987	0,987
	10	0,993	0,993	0,993	0,993
4	1	0,968	0,970	0,968	0,968
	2	0,968	0,969	0,968	0,968
	3	0,975	0,976	0,975	0,974
	4	0,981	0,981	0,981	0,981
	5	0,993	0,993	0,993	0,993
	6	0,993	0,993	0,993	0,993
	7	0,981	0,981	0,981	0,981
	8	0,981	0,981	0,981	0,981
	9	0,981	0,982	0,981	0,981
	10	0,987	0,987	0,987	0,987
5	1	0,987	0,987	0,987	0,987
	2	0,987	0,987	0,987	0,987
	3	0,968	0,968	0,968	0,968
	4	0,987	0,988	0,987	0,987
	5	0,975	0,975	0,975	0,974
	6	0,968	0,970	0,968	0,968
	7	0,968	0,968	0,968	0,968
	8	0,962	0,963	0,962	0,962
	9	0,962	0,963	0,962	0,962
	10	1,000	1,000	1,000	1,000
6	1	0,981	0,982	0,981	0,981

	2	0,962	0,965	0,962	0,962
	3	1,000	1,000	1,000	1,000
	4	0,993	0,993	0,993	0,993
	5	0,981	0,981	0,981	0,981
	6	0,962	0,962	0,962	0,962
	7	0,981	0,981	0,981	0,981
	8	0,981	0,981	0,981	0,981
	9	0,987	0,987	0,987	0,987
	10	0,975	0,975	0,975	0,974
7	1	0,956	0,956	0,956	0,956
	2	0,987	0,987	0,987	0,987
	3	0,981	0,981	0,981	0,981
	4	0,987	0,987	0,987	0,987
	5	0,981	0,982	0,981	0,981
	6	0,981	0,981	0,981	0,981
	7	0,987	0,987	0,987	0,987
	8	0,981	0,981	0,981	0,981
	9	0,975	0,975	0,975	0,974
	10	0,981	0,981	0,981	0,981
8	1	0,993	0,993	0,993	0,993
	2	0,981	0,981	0,981	0,981
	3	0,962	0,962	0,962	0,962
	4	0,987	0,987	0,987	0,987
	5	0,993	0,993	0,993	0,993
	6	0,987	0,987	0,987	0,987
	7	0,975	0,976	0,975	0,974
	8	1,000	1,000	1,000	1,000
	9	0,981	0,981	0,981	0,981
	10	0,943	0,945	0,943	0,943
9	1	0,993	0,993	0,993	0,993
	2	0,993	0,993	0,993	0,993
	3	0,975	0,975	0,975	0,974
	4	0,993	0,993	0,993	0,993
	5	0,981	0,981	0,981	0,981
	6	0,987	0,987	0,987	0,987
	7	0,981	0,981	0,981	0,981
	8	0,968	0,969	0,968	0,968
	9	0,987	0,987	0,987	0,987
	10	0,975	0,976	0,975	0,975
10	1	0,987	0,987	0,987	0,987

	2	0,968	0,969	0,968	0,968
	3	0,993	0,993	0,993	0,993
	4	0,981	0,981	0,981	0,981
	5	0,987	0,988	0,987	0,987
	6	0,975	0,975	0,975	0,975
	7	0,981	0,981	0,981	0,981
	8	0,987	0,988	0,987	0,987
	9	0,981	0,981	0,981	0,981
	10	0,981	0,981	0,981	0,981

Hasil pada Tabel 23 menunjukkan bahwa pembatasan kedalaman maksimal pohon pada level 15 menghasilkan distribusi performa dengan rentang akurasi 94,3%-100%, dimana strategi pembatasan kedalaman berhasil mengontrol kompleksitas model untuk mencegah *overfitting* namun tetap menghasilkan variabilitas yang mencerminkan sensitivitas terhadap komposisi data pada *fold* tertentu. Selain pembatasan kedalaman pohon, faktor penting lainnya dalam algoritma Random Forest adalah jumlah *decision trees* dalam *ensemble*, dimana analisis terhadap parameter n_estimators diperlukan untuk memahami titik konvergensi optimal agregasi prediksi.

Untuk mengevaluasi pengaruh jumlah *decision trees* dalam *ensemble*, dilakukan pengujian model Random Forest dengan parameter n_estimators=100 pada seluruh kombinasi *fold* dan repetisi, dengan pencatatan detail performa ditampilkan pada Tabel 24:

Tabel 24. Pengujian Sistem RF 10 Uji Parameter n_estimators= 100

Fold ke-	Repetisi ke-	Akurasi (%)	Presisi (%)	*Recall* (%)	F1-*Score* (%)
1	1	1,000	1,000	1,000	1,000
	2	0,975	0,975	0,975	0,974
	3	0,968	0,969	0,968	0,968
	4	0,993	0,993	0,993	0,993
	5	0,975	0,974	0,975	0,974
	6	0,956	0,957	0,956	0,955
	7	1,000	1,000	1,000	1,000
	8	0,987	0,988	0,987	0,987
	9	0,981	0,981	0,981	0,981
	10	0,956	0,956	0,956	0,955
2	1	0,975	0,976	0,975	0,974
	2	0,981	0,981	0,981	0,981
	3	0,975	0,975	0,975	0,974
	4	0,975	0,975	0,975	0,974

	5	0,975	0,976	0,975	0,974
	6	0,987	0,987	0,987	0,987
	7	0,987	0,988	0,987	0,987
	8	0,981	0,981	0,981	0,981
	9	1,000	1,000	1,000	1,000
	10	0,981	0,981	0,981	0,981
3	1	0,981	0,981	0,981	0,981
	2	0,968	0,969	0,968	0,968
	3	0,987	0,988	0,987	0,987
	4	0,987	0,987	0,987	0,987
	5	0,975	0,976	0,975	0,974
	6	0,981	0,981	0,981	0,981
	7	0,975	0,974	0,975	0,974
	8	0,981	0,981	0,981	0,981
	9	0,987	0,987	0,987	0,987
	10	1,000	1,000	1,000	1,000
4	1	0,987	0,988	0,987	0,987
	2	0,962	0,962	0,962	0,962
	3	0,981	0,981	0,981	0,981
	4	0,981	0,981	0,981	0,981
	5	1,000	1,000	1,000	1,000
	6	0,993	0,993	0,993	0,993
	7	0,987	0,987	0,987	0,987
	8	0,987	0,987	0,987	0,987
	9	0,968	0,970	0,968	0,968
	10	0,987	0,987	0,987	0,987
5	1	0,987	0,987	0,987	0,987
	2	0,987	0,987	0,987	0,987
	3	0,975	0,975	0,975	0,974
	4	0,987	0,988	0,987	0,987
	5	0,993	0,993	0,993	0,993
	6	0,956	0,958	0,956	0,956
	7	0,950	0,950	0,950	0,949
	8	0,956	0,958	0,956	0,956
	9	0,981	0,981	0,981	0,981
	10	1,000	1,000	1,000	1,000
6	1	0,975	0,976	0,975	0,975
	2	0,968	0,972	0,968	0,968
	3	0,987	0,987	0,987	0,987
	4	0,993	0,993	0,993	0,993

	5	0,975	0,975	0,975	0,974
	6	0,956	0,956	0,956	0,956
	7	0,981	0,981	0,981	0,981
	8	0,987	0,987	0,987	0,987
	9	0,987	0,987	0,987	0,987
	10	0,975	0,975	0,975	0,974
7	1	0,962	0,962	0,962	0,962
	2	0,987	0,987	0,987	0,987
	3	0,981	0,981	0,981	0,981
	4	0,987	0,988	0,987	0,987
	5	0,981	0,982	0,981	0,981
	6	0,975	0,975	0,975	0,975
	7	0,987	0,987	0,987	0,987
	8	0,981	0,981	0,981	0,981
	9	0,975	0,975	0,975	0,974
	10	0,993	0,993	0,993	0,993
8	1	0,993	0,993	0,993	0,993
	2	0,968	0,971	0,968	0,968
	3	0,962	0,962	0,962	0,962
	4	0,993	0,993	0,993	0,993
	5	0,993	0,993	0,993	0,993
	6	0,987	0,987	0,987	0,987
	7	0,981	0,982	0,981	0,981
	8	1,000	1,000	1,000	1,000
	9	0,987	0,987	0,987	0,987
	10	0,956	0,956	0,956	0,956
9	1	0,993	0,993	0,993	0,993
	2	0,993	0,993	0,993	0,993
	3	0,981	0,981	0,981	0,981
	4	0,993	0,993	0,993	0,993
	5	0,981	0,981	0,981	0,981
	6	0,987	0,987	0,987	0,987
	7	0,981	0,98	0,981	0,981
	8	0,962	0,964	0,962	0,962
	9	0,987	0,987	0,987	0,987
	10	0,981	0,981	0,981	0,981
10	1	0,987	0,987	0,987	0,987
	2	0,975	0,975	0,975	0,974
	3	0,987	0,988	0,987	0,987
	4	0,981	0,981	0,981	0,981

	5	0,981	0,981	0,981	0,981
	6	0,975	0,975	0,975	0,975
	7	0,987	0,987	0,987	0,987
	8	0,981	0,981	0,981	0,981
	9	0,975	0,975	0,975	0,974
	10	0,981	0,981	0,981	0,981

Tabel 24 memperlihatkan bahwa konfigurasi dengan 100 *decision trees* menghasilkan pola performa yang identik dengan konfigurasi bootstrap=True (rentang akurasi 95,0%-100%), mengonfirmasi bahwa jumlah *trees* sebanyak 100 sudah mencapai titik konvergensi optimal dimana penambahan *trees* lebih lanjut tidak memberikan peningkatan signifikan pada akurasi *ensemble*. Aspek terakhir yang perlu dievaluasi adalah kriteria pemisahan *node* dalam konstruksi *decision trees*, dimana perbandingan antara metrik *Gini impurity* dan alternatifnya dapat memberikan wawasan tentang efektivitas strategi pemilihan *split* optimal dalam konteks klasifikasi multi-kelas.

Untuk mengevaluasi efektivitas kriteria pemisahan *node* berbasis *Gini impurity*, dilakukan pengujian model Random Forest dengan parameter criterion=gini pada seluruh kombinasi *fold* dan repetisi, dengan hasil performa lengkap disajikan pada Tabel 25:

Tabel 25. Pengujian Sistem RF 10 Uji Parameter Criterion= Gini

Fold ke-	Repetisi ke-	Akurasi (%)	Presisi (%)	*Recall* (%)	F1-*Score* (%)
1	1	1,000	1,000	1,000	1,000
	2	0,975	0,975	0,975	0,974
	3	0,968	0,969	0,968	0,968
	4	0,993	0,993	0,993	0,993
	5	0,975	0,974	0,975	0,974
	6	0,956	0,957	0,956	0,955
	7	1,000	1,000	1,000	1,000
	8	0,987	0,988	0,987	0,987
	9	0,981	0,981	0,981	0,981
	10	0,956	0,956	0,956	0,955
2	1	0,975	0,976	0,975	0,974
	2	0,981	0,981	0,981	0,981
	3	0,975	0,975	0,975	0,974
	4	0,975	0,975	0,975	0,974
	5	0,975	0,976	0,975	0,974
	6	0,987	0,987	0,987	0,987
	7	0,987	0,988	0,987	0,987

	8	0,981	0,981	0,981	0,981
	9	1,000	1,000	1,000	1,000
	10	0,981	0,981	0,981	0,981
3	1	0,981	0,981	0,981	0,981
	2	0,968	0,969	0,968	0,968
	3	0,987	0,988	0,987	0,987
	4	0,987	0,987	0,987	0,987
	5	0,975	0,976	0,975	0,974
	6	0,981	0,981	0,981	0,981
	7	0,975	0,974	0,975	0,974
	8	0,981	0,981	0,981	0,981
	9	0,987	0,987	0,987	0,987
	10	1,000	1,000	1,000	1,000
4	1	0,987	0,988	0,987	0,987
	2	0,962	0,962	0,962	0,962
	3	0,981	0,981	0,981	0,981
	4	0,981	0,981	0,981	0,981
	5	1,000	1,000	1,000	1,000
	6	0,993	0,993	0,993	0,993
	7	0,987	0,987	0,987	0,987
	8	0,987	0,987	0,987	0,987
	9	0,968	0,970	0,968	0,968
	10	0,987	0,987	0,987	0,987
5	1	0,987	0,987	0,987	0,987
	2	0,987	0,987	0,987	0,987
	3	0,975	0,975	0,975	0,974
	4	0,987	0,988	0,987	0,987
	5	0,993	0,993	0,993	0,993
	6	0,956	0,958	0,956	0,956
	7	0,950	0,950	0,950	0,949
	8	0,956	0,958	0,956	0,956
	9	0,981	0,981	0,981	0,981
	10	1,000	1,000	1,000	1,000
6	1	0,975	0,976	0,975	0,975
	2	0,968	0,972	0,968	0,968
	3	0,987	0,987	0,987	0,987
	4	0,993	0,993	0,993	0,993
	5	0,975	0,975	0,975	0,974
	6	0,956	0,956	0,956	0,956
	7	0,981	0,981	0,981	0,981

	8	0,987	0,987	0,987	0,987
	9	0,987	0,987	0,987	0,987
	10	0,975	0,975	0,975	0,974
7	1	0,962	0,962	0,962	0,962
	2	0,987	0,987	0,987	0,987
	3	0,981	0,981	0,981	0,981
	4	0,987	0,988	0,987	0,987
	5	0,981	0,982	0,981	0,981
	6	0,975	0,975	0,975	0,975
	7	0,987	0,987	0,987	0,987
	8	0,981	0,981	0,981	0,981
	9	0,975	0,975	0,975	0,974
	10	0,993	0,993	0,993	0,993
8	1	0,993	0,993	0,993	0,993
	2	0,968	0,971	0,968	0,968
	3	0,962	0,962	0,962	0,962
	4	0,993	0,993	0,993	0,993
	5	0,993	0,993	0,993	0,993
	6	0,987	0,987	0,987	0,987
	7	0,981	0,982	0,981	0,981
	8	1,000	1,000	1,000	1,000
	9	0,987	0,987	0,987	0,987
	10	0,956	0,956	0,956	0,956
9	1	0,993	0,993	0,993	0,993
	2	0,993	0,993	0,993	0,993
	3	0,981	0,981	0,981	0,981
	4	0,993	0,993	0,993	0,993
	5	0,981	0,981	0,981	0,981
	6	0,987	0,987	0,987	0,987
	7	0,981	0,981	0,981	0,981
	8	0,962	0,964	0,962	0,962
	9	0,987	0,987	0,987	0,987
	10	0,981	0,981	0,981	0,981
10	1	0,987	0,987	0,987	0,987
	2	0,975	0,975	0,975	0,974
	3	0,987	0,988	0,987	0,987
	4	0,981	0,981	0,981	0,981
	5	0,981	0,981	0,981	0,981
	6	0,975	0,975	0,975	0,975
	7	0,987	0,987	0,987	0,987

	8	0,981	0,981	0,981	0,981
	9	0,975	0,975	0,975	0,974
	10	0,981	0,981	0,981	0,981

Berdasarkan Tabel 25, penggunaan *Gini impurity* sebagai kriteria pemisahan menghasilkan distribusi performa yang konsisten dengan konfigurasi lainnya (rentang akurasi 95,0%-100%), mendemonstrasikan bahwa metrik *Gini* merupakan pilihan yang tepat untuk mengukur ketidakmurnian *node* dalam konteks klasifikasi multi-kelas dengan distribusi kelas seimbang, sebagaimana ditunjukkan oleh konsistensi performa pada 100 eksperimen yang dilakukan. Secara keseluruhan, hasil evaluasi komprehensif terhadap empat konfigurasi Random Forest menunjukkan bahwa algoritma *ensemble learning* ini menghasilkan performa yang stabil dan seimbang dengan akurasi rata-rata 97,50%, meskipun masih berada 1,75% di bawah performa SVM *Linear* yang mencapai 99,25%, mengindikasikan bahwa untuk kasus klasifikasi parasit malaria dengan fitur ekstraksi ResNet50 berdimensi tinggi, pendekatan *Linear classifier* memberikan hasil yang lebih optimal dibandingkan metode *ensemble* berbasis pohon keputusan.

4.5. Perbandingan Kinerja SVM dan *Random Forest*

Setelah melakukan eksperimen komprehensif terhadap kedua algoritma klasifikasi, dilakukan analisis perbandingan menyeluruh untuk mengevaluasi kinerja *Support Vector Machine* (SVM) dan *Random Forest* dalam mengklasifikasikan spesies malaria. Perbandingan ini mencakup beberapa aspek penting: akurasi klasifikasi, metrik evaluasi dan skor kepercayaan (*confidence score*).

4.5.1. Perbandingan Akurasi dan Metrik Evaluasi

Berdasarkan hasil eksperimen K-Fold *Cross-Validation* dengan 10 fold dan 10 repetisi (total 400 eksperimen per konfigurasi), diperoleh perbandingan kinerja sebagai berikut:

Tabel 26. Perbandingan Akurasi K-Fold *Cross-Validation*

Algoritma	**Konfigurasi Terbaik**	**Rata-rata Akurasi**	**Standar Deviasi**	**Rata-rata F1-*Score***
SVM	*Linear* Kernel	0.9925	±0.0068	0.9925
SVM	RBF (C=10)	0.9864	±0.0102	0.9864
SVM	RBF (gamma=*Scale*)	0.9824	±0.0119	0.9824
SVM	*Polynomial* (degree=2)	0.9803	±0.0118	0.9803

Random Forest	max_depth=15	0.9808	±0.0106	0.9808
Random Forest	bootstrap=True	0.9813	±0.0113	0.9813
Random Forest	n_estimators=100	0.9813	±0.0113	0.9813
Random Forest	criterion=gini	0.9813	±0.0113	0.9813

Dari tabel 26 di atas, terlihat bahwa SVM dengan kernel *Linear* menunjukkan kinerja terbaik dengan *mean accuracy* 99.25%, unggul tipis dibandingkan konfigurasi *Random Forest* terbaik yang mencapai 98.13%. Selain itu, SVM *Linear* juga memiliki standar deviasi terkecil (±0.0068), mengindikasikan konsistensi yang sangat baik dan stabilitas model yang tinggi.

Tabel 27. Perbandingan Kinerja pada *Validation Set*

Algoritma	Konfigurasi	Akurasi	Presisi	*Recall*	F1-*Score*
SVM	*Linear*	84.38%	86.93%	84.38%	84.03%
Random Forest	Bootstrap=True	81.25%	81.25%	81.25%	81.25%

Pada *validation* set yang terdiri dari 32 sampel (8 sampel per kelas), SVM *Linear* menunjukkan performa superior dengan akurasi 84.38%, lebih tinggi 3.13% dibandingkan *Random Forest* (81.25%). Presisi SVM juga lebih tinggi (86.93% vs 81.25%), menunjukkan kemampuan yang lebih baik dalam meminimalkan *false positive*.

Tabel 28. Perbandingan Performa per Kelas pada *Validation* Set

Model	Kelas	Presisi	*Recall*	F1-*Score*	*Support*
SVM *Linear*	***Falciparum***	0.88	0.88	0.88	8
	Malariae	0.88	0.88	0.88	8
	Ovale	0.73	1.00	0.84	8
	Vivax	1.00	0.62	0.77	8
Random Forest Bootsrap = True	***Falciparum***	0.88	0.88	0.88	8
	Malariae	0.88	0.88	0.88	8
	Ovale	0.75	0.75	0.75	8
	Vivax	0.75	0.75	0.75	8

Perbandingan per kelas menunjukkan karakteristik berbeda antara kedua algoritma:

1. *Falciparum* dan *Malariae*: Kedua algoritma menunjukkan performa identik (presisi, *recall*, dan F1-*Score* = 0.88)
2. *Ovale*: SVM menunjukkan *recall* sempurna (1.00) namun presisi lebih rendah (0.73), sementara *Random Forest* lebih seimbang (0.75 untuk keduanya)

3. *Vivax* : SVM memiliki presisi sempurna (1.00) tetapi *recall* rendah (0.62), sedangkan *Random Forest* menunjukkan performa seimbang (0.75 untuk keduanya)

Pola ini mengindikasikan bahwa SVM cenderung lebih "*confident*" dalam prediksinya, menghasilkan *Trade-off* antara presisi dan *recall* yang lebih ekstrem, sementara *Random Forest* menghasilkan prediksi yang lebih seimbang.

4.5.2. Perbandingan *Confidence score*

Salah satu aspek penting dalam sistem klasifikasi medis adalah kemampuan model untuk mengukur tingkat kepercayaan prediksinya. Penelitian ini mengimplementasikan metode *confidence scoring* yang comparable antara SVM dan *Random Forest*.

Tabel 29. Perbandingan Metode *Confidence Scoring*

Algoritma	Metode Terbaik	*Correlation*	*Confidence* Gap	Combined *Score*
SVM	Raw (*UnNormalized*)	0.3369	0.1023	0.2430
SVM	*Normalized Sigmoid*	0.3426	0.0323	0.2131
SVM	*Calibrated* (*Platt*)	0.0723	0.2431	0.0529
Random Forest	Improved (Margin)	0.3178	0.1974	0.2696
Random Forest	Legacy (Max Prob)	0.2342	0.0940	0.1781
Random Forest	*Calibrated*	0.1681	0.1330	0.1541

Confidence Gap merepresentasikan selisih rata-rata *confidence score* antara prediksi benar dan salah. Gap yang lebih besar mengindikasikan model dapat membedakan dengan lebih baik antara prediksi yang yakin dan tidak yakin.

Hasil menunjukkan bahwa:

1. *Random Forest* dengan *Improved Margin Method* menghasilkan *combined score* tertinggi (0.2696), dengan *confidence* gap terbesar (0.1974), menunjukkan kemampuan superior dalam mengkuantifikasi ketidakpastian prediksi.
2. SVM dengan Raw (*UnNormalized*) Method juga menunjukkan performa baik (*combined score* 0.2430) dengan korelasi tertinggi (0.3369), mengindikasikan *confidence score* yang reliabel.
3. *SVM Calibrated* (*Platt Scaling*) memiliki *confidence* gap yang sangat besar (0.2431), namun korelasi yang rendah (0.0723), menunjukkan bahwa

meskipun gap-nya tinggi, hubungannya dengan akurasi prediksi tidak konsisten.

4. Metode *Normalized Sigmoid* untuk SVM memiliki gap terkecil (0.0323), namun korelasi tertinggi (0.3426), menunjukkan *trade-off* antara *discriminative power* dan *reliability*.

Ini menunjukkan bahwa ketika *Random Forest* memberikan *confidence score* tinggi, prediksi tersebut sangat dapat diandalkan, yang sangat penting untuk aplikasi medis di mana *clinical decision* making memerlukan tingkat kepercayaan yang tinggi.

4.6. Analisis Kesalahan Deteksi Objek

Evaluasi komprehensif terhadap performa model YOLO 11l pada ketiga eksperimen *learning rate* mengungkapkan berbagai karakteristik kesalahan yang terjadi selama proses deteksi dan klasifikasi sel darah terinfeksi. Analisis error ini mencakup identifikasi pola misklasifikasi, faktor-faktor penyebab kesalahan, serta dampak variasi *learning rate* terhadap performa model. Pemahaman mendalam terhadap sumber dan karakteristik error menjadi krusial untuk optimalisasi sistem deteksi otomatis penyakit infeksi darah berbasis *deep learning*.

4.6.1. Karakteristik dan Pola Kesalahan Klasifikasi

Misklasifikasi terjadi konsisten antara kelas Anemia dan Malaria. LR 0.001 menghasilkan 5 kesalahan (3,9%): 2 Anemia ke Malaria dan 3 Malaria ke Anemia. LR 0.0001 menghasilkan 4 kesalahan: 2 Malaria ke Anemia dan 2 *Trypomastigote* ke Anemia, tanpa kesalahan pada Anemia (100%). LR 0.0015 mencapai *error rate* terendah (2,4%) dengan 3 kesalahan: 1 Anemia ke Malaria dan 2 Malaria ke Anemia. Pola kesalahan terbatas pada pasangan Anemia dan Malaria mengindikasikan tumpang tindih *feature space* antara kedua kelas.

Faktor penyebab meliputi variasi kualitas citra, sudut pandang sel, dan *overlap* antar sel yang mengaburkan karakteristik morfologi khas. Meskipun secara visual berbeda (Anemia: bentuk bulan sabit; Malaria: bercak gelap parasit), faktor teknis tersebut menyebabkan ambiguitas deteksi. Pada LR 0.0001, model bersifat konservatif dengan memilih Anemia sebagai *default* saat menghadapi ketidakpastian. Kolase visual menunjukkan *confidence score* optimal dan tinggi, namun tidak berkorelasi sempurna dengan akurasi klasifikasi. Ini mengindikasikan *overconfidence* model, terutama pada pasangan Anemia dan Malaria dengan *feature space* tumpang tindih.

4.6.2. Variasi Performa dan Pengaruh Learning Rate

Trypomastigote menunjukkan performa terendah (*recall* 0.490 sampai 0.709, mAP@0.5:0.95 = 0.375 sampai 0.449) akibat variasi morfologi yang lebih besar. LR 0.0001 menghasilkan *recall* terendah (0.490) dengan 51% *false negative*. Anemia

menunjukkan performa tertinggi dan stabil (mAP@0.5 >0.95) karena bentuk bulan sabit yang distinktif. Malaria menempati posisi menengah dengan performa terbaik pada LR 0.0001 (mAP@0.5:0.95 = 0.719), menunjukkan fitur halus parasit memerlukan pembelajaran lambat.

Variasi LR menghasilkan *trade-off* berbeda. LR 0.001 menghasilkan kesalahan bilateral Anemia dan Malaria (5 kesalahan, 3,9%) namun sempurna pada *Trypomastigote* (100%). LR 0.0001 mengeliminasi kesalahan Anemia (100%) tetapi memperkenalkan kesalahan *Trypomastigote* ke Anemia (4,9%). LR 0.0015 mencapai keseimbangan optimal (3 kesalahan, 2,4%), menghasilkan akurasi validasi tertinggi: 96,8% (LR 0.001), 97,6% (LR 0.0001), 98,4% (LR 0.0015).

Anemia pada LR 0.001 mencapai keseimbangan optimal (*precision* 0.949, *recall* 0.914, *false positive* 5,1%, *false negative* 8,6%). LR 0.0001 meningkatkan *recall* (0.944) namun menurunkan *precision* (0.756, *false positive* 24,4%). Malaria menunjukkan *trade-off* ekstrem pada LR 0.001 (*precision* 0.739, *recall* 0.830, *false positive* 26,1%), diperbaiki pada LR 0.0001 (*precision* 0.810, *recall* 0.830). *Trypomastigote* menunjukkan masalah kritis dengan *recall* sangat rendah (0.490 sampai 0.709) meski *precision* tinggi (0.771 sampai 0.820). Pada LR 0.0001, *false negative* mencapai 51%.

4.6.3. Masalah Lokalisasi dan Distribusi Data

Disparitas besar antara mAP@0.5 dan mAP@0.5:0.95 menunjukkan masalah lokalisasi *bounding box*: LR 0.001 (31,5%), LR 0.0001 (30,3%), LR 0.0015 (33,2%). *Trypomastigote* mengalami degradasi paling ekstrem (~42%), Anemia (31 sampai 34%), Malaria terendah (18 sampai 24%, LR 0.0001 = 18,5%).

Dataset validasi menunjukkan ketidakseimbangan moderat: Anemia 47 *instance* (37,3%), Malaria 39 *instance* (30,9%), *Trypomastigote* 41 *instance* (32,5%). Ketidakseimbangan 6,4% menyebabkan *bias* pembelajaran, dimana LR 0.0001 cenderung memprediksi kasus tidak pasti sebagai Anemia (*conservative prediction bias*). Dataset pelatihan berisi 127 citra dengan 1.071 *instance* (rata-rata 8,4 *instance*/citra). Performa sempurna pada *test dataset* (100%) kontras dengan variasi pada *validation dataset* (96,8% sampai 98,4%), mengindikasikan *validation set* memiliki variabilitas lebih tinggi dan lebih representatif terhadap kondisi klinis riil.

Model YOLO 11l (190 lapisan, 25,28 juta parameter) dioptimasi untuk objek berukuran kecil pada resolusi 1280×1280 piksel. Perbedaan karakteristik morfologi berpengaruh signifikan: Anemia (bentuk bulan sabit konsisten, mAP@0.5 >0.95), Malaria (fitur halus bercak gelap parasit, performa terbaik LR 0.0001 dengan mAP@0.5:0.95 = 0.719), *Trypomastigote* (variasi morfologi besar, mAP@0.5 = 0.660 sampai 0.776). Kecepatan inferensi konsisten (26,9 sampai 27,2 ms/citra).

4.6.4. Dinamika Konvergensi dan Overfitting

LR 0.001 menunjukkan pembelajaran sehat dengan *train loss* menurun konsisten dan *validation loss* stabil setelah *epoch* 50, fluktuasi ringan *epoch* 100 sampai 120 (fenomena normal *fine-tuning*), tidak ada *overfitting* hingga *epoch* 155. LR 0.0001 menunjukkan indikasi *overfitting* dengan *validation loss* meningkat setelah *epoch* 28 dan fluktuasi *precision* besar (0.55 sampai 0.85) setelah *epoch* 20. LR 0.0015 menunjukkan karakteristik anti-*overfitting* superior dengan *validation loss* sangat stabil setelah *epoch* 30 tanpa tren peningkatan hingga *epoch* 116. Mekanisme *early stopping* (*patience* 50 *epoch*) efektif pada semua eksperimen.

LR 0.0001 mencapai konvergensi tercepat (*epoch* 28, 8.053 jam) namun performa suboptimal (mAP@0.5 = 0.831), setelahnya metrik berfluktuasi tanpa peningkatan konsisten. LR 0.001 memerlukan waktu terlama (*epoch* 105, 16.027 jam) namun performa tertinggi (mAP@0.5 = 0.859), pola *diminishing return* terlihat setelah *epoch* 100. LR 0.0015 menunjukkan konvergensi moderat (*epoch* 66, 11.943 jam) dengan keseimbangan efisiensi dan performa (mAP@0.5 = 0.837). Efisiensi waktu konsisten (~6,2 menit/*epoch*), mengkonfirmasi perbedaan durasi disebabkan jumlah *epoch* konvergensi, bukan overhead komputasi.

4.7. Analisis Kesalahan Klasifikasi

Untuk memahami secara mendalam karakteristik dan pola kesalahan yang dihasilkan oleh kedua algoritma klasifikasi, dilakukan analisis kesalahan (error analysis) yang komprehensif terhadap hasil prediksi pada *validation* set. Analisis ini bertujuan untuk mengidentifikasi kelas-kelas yang paling sering mengalami kesalahan klasifikasi, pola misklasifikasi antar kelas, serta faktor-faktor yang berkontribusi terhadap kesalahan tersebut.

4.7.1. *Confusion matrix* dan Pola Misklasifikasi

Analisis *confusion matrix* dari kedua algoritma pada *validation* set mengungkapkan pola misklasifikasi yang berbeda namun memiliki kemiripan dalam hal kelas-kelas yang sulit dibedakan.

Tabel 30. *Confusion matrix* SVM *Linear* pada *Validation* Set

	Predicted Falciparum	***Predicted Malariae***	***Predicted Ovale***	***Predicted Vivax***
Falciparum	7	1	0	0
Malariae	1	7	0	0
Ovale	0	0	8	0
Vivax	0	0	3	5

Tabel 31. *Confusion matrix Random Forest* pada *Validation* Set

	Predicted Falciparum	***Predicted Malariae***	***Predicted Ovale***	***Predicted Vivax***
Falciparum	7	1	0	0
Malariae	1	7	0	0
Ovale	0	0	6	2
Vivax	0	0	2	6

Dari *confusion matrix* di atas, dapat diidentifikasi beberapa pola misklasifikasi yang konsisten:

1. Konfusi *Falciparum-Malariae* (Konsisten pada Kedua Algoritma)

Kedua algoritma menunjukkan pola kesalahan yang identik: masing-masing 1 sampel *Falciparum* salah diprediksi sebagai *Malariae*, dan 1 sampel *Malariae* salah diprediksi sebagai *Falciparum*. Pola bilateral ini mengindikasikan adanya overlap fitur visual yang signifikan antara kedua spesies ini dalam *feature space* ResNet50. Kesalahan yang konsisten pada kedua algoritma menunjukkan bahwa ini adalah *inherent difficulty* dalam dataset, bukan limitasi spesifik dari algoritma tertentu.

2. Konfusi *Ovale-Vivax* (Pola Berbeda Antar Algoritma)

SVM *Linear* Menunjukkan pola *unidirectional* yang ekstrem - 3 dari 8 sampel *Vivax* (37.5%) salah diklasifikasikan sebagai *Ovale*, sementara *Ovale* tidak pernah salah diklasifikasikan (100% *recall*). Hal ini mengindikasikan bahwa *hyperplane Linear* SVM menempatkan *decision boundary* yang sangat konservatif, menyebabkan banyak sampel *Vivax* "melewati batas" ke *region Ovale. Random Forest* Menunjukkan pola *bidirectional* yang lebih seimbang - 2 dari 8 sampel *Ovale* salah ke *Vivax* , dan 2 dari 8 sampel *Vivax* salah ke *Ovale.* Distribusi error yang simetris ini mencerminkan karakteristik *ensemble learning* yang menghasilkan *decision boundaries* yang lebih "demokratis" melalui voting dari *multiple trees*.

3. Performa Kontras pada Kelas *Ovale*

Pada SVM, *Ovale* mencapai *recall* sempurna (100%) tanpa satupun *false negative*, menunjukkan fitur *Ovale* sangat *distinctive* untuk *Linear separation*. Namun, ini dibayar dengan 3 *false positive* dari *Vivax* . Pada *Random Forest*, *Ovale* mengalami 2 *false negative* (*recall* 75%), menunjukkan bahwa pendekatan *ensemble* dengan *multiple decision boundaries* dapat menyebabkan *inconsistency* dalam prediksi untuk kelas ini, namun juga mengurangi *false positive*.

4. Performa Kontras pada Kelas *Vivax*

Vivax merupakan kelas yang paling bermasalah untuk SVM (37.5% *error rate*) tetapi lebih *manageable* untuk Random Forest (25% *error rate*). Ini menunjukkan bahwa *non-Linear decision boundaries* dari *Random Forest* lebih cocok untuk memisahkan *Vivax* dari kelas lain, khususnya dari *Ovale*.

4.7.2. Analisis *Confidence score* pada Prediksi Benar dan Misklasifikasi

Untuk memahami tingkat keyakinan model pada prediksi yang salah, dilakukan analisis terhadap *confidence scores* dari sampel-sampel yang mengalami misklasifikasi:

Tabel 32. Analisis *Confidence* Gap pada Prediksi yang Benar vs Salah

Algoritma	Metode	*Mean Correct*	*Mean Incorrect*	Gap
SVM	*Raw* (*UnNormalized*)	1.1505	1.0482	0.1023
SVM	*Normalized Sigmoid*	0.7590	0.7403	0.0187
SVM	*Calibrated*	0.9215	0.8977	0.0238
Random Forest	*Improved* (*Margin*)	0.4408	0.2433	0.1974
Random Forest	*Legacy* (*Max Prob*)	0.6523	0.5583	0.0940
Random Forest	*Calibrated*	0.7930	0.6600	0.1330

Dari tabel di atas, terlihat beberapa pemahaman penting:

1. Gap *Confidence* antara Prediksi Benar dan Salah

Random Forest Improved memiliki gap terbesar (0.1974 atau 19.74%), menunjukkan model paling mampu membedakan prediksi yang *confident* dari yang tidak *confident*. SVM *Raw* memiliki gap *moderate* (0.1023 atau 10.23%), sementara SVM *Normalized* memiliki gap terkecil (0.0187 atau 1.87%), menunjukkan *trade-off* antara normalisasi dan *discriminative power*.

2. *Confidence* Level pada Misklasifikasi

Berdasarkan hasil evaluasi pada *validation set*, model *Random Forest* menunjukkan kinerja yang sangat konsisten pada tingkat *confidence* yang tinggi. Pada ambang kepercayaan (*threshold*) sebesar 0,40, model mencapai akurasi 100% pada 16 sampel, yang mengindikasikan bahwa seluruh prediksi dengan tingkat kepercayaan di atas nilai tersebut diklasifikasikan secara tepat. Peningkatan *threshold* menjadi 0,54 tetap mempertahankan akurasi sempurna (100%) pada 10 sampel, sementara pada *threshold* yang lebih ketat, yaitu 0,79, akurasi 100% masih tercapai pada 4 sampel.

Hasil ini menunjukkan bahwa skor *confidence* yang dihasilkan oleh model *Random Forest* bersifat *well-Calibrated*, di mana tingkat kepercayaan yang lebih tinggi secara langsung merefleksikan tingkat akurasi prediksi yang lebih tinggi. Karakteristik ini sangat penting dalam konteks *clinical decision making*, karena memungkinkan tenaga medis untuk mengandalkan prediksi dengan *confidence*

tinggi secara lebih aman dan terukur, sehingga dapat meminimalkan risiko kesalahan dalam sistem pendukung keputusan berbasis kecerdasan buatan.

3. Implikasi untuk Aplikasi Medis

Gap *confidence* yang besar pada *Random Forest* (0.1974) dengan *correlation* yang baik (0.3178) menunjukkan bahwa model ini dapat memberikan *early warning* ketika tidak yakin dengan prediksinya. Dalam konteks diagnosis malaria, ini memungkinkan sistem untuk menandai kasus-kasus yang memerlukan verifikasi manual oleh tenaga medis.

4.7.3. Analisis *Error Rate* per Kelas

Untuk mengidentifikasi kelas-kelas yang paling *challenging*, dilakukan perhitungan *Error Rate* untuk setiap kelas:

Tabel 33. *Error Rate* dan *False* Positif / Negatif per Kelas

Kelas	SVM *Error Rate*	RF *Error Rate*	SVM *False Positive*	SVM *False Negative*	RF *False Positive*	RF *False Negative*
Falciparum	12.5% (1/8)	12.5% (1/8)	1	1	1	1
Malariae	12.5% (1/8)	12.5% (1/8)	1	1	1	1
Ovale	0% (0/8)	25% (2/8)	3	0	2	2
Vivax	37.5% (3/8)	25% (2/8)	0	3	2	2

Analisis *Error Rate* mengungkapkan:

1. *Vivax* sebagai Kelas Paling *Challenging*

Kelas *Vivax* merupakan yang paling sulit diklasifikasi oleh SVM *Linear*, menunjukkan *Error Rate* tertinggi (37.5%). Semua kesalahan berupa *false negative* (3 kasus), artinya sampel *Vivax* sering keliru diprediksi sebagai *Ovale*. Ketiadaan *false positive* (0) menunjukkan SVM sangat konservatif dalam memprediksi *Vivax* - model cenderung *under-predict* kelas ini karena fitur *Vivax* memiliki overlap signifikan dengan *Ovale* dalam ruang fitur *Linear*. Pada Random Forest, *Error Rate Vivax* lebih rendah (25%) dengan distribusi *balanced* antara *false positive* (2) dan *false negative* (2), menunjukkan *decision boundary* yang lebih *robust*.

2. *Trade-off* Presisi *Recall* pada SVM untuk *Ovale*

Kelas *Ovale* mencapai performa sempurna pada SVM dengan error rate 0% dan *recall* sempurna (0 *false negative*). Namun, hal ini dibayar dengan *trade-off* berupa 3 *false positive* tertinggi, di mana semua sampel Vivax yang salah diklasifikasikan "jatuh" ke kategori *Ovale*. Pola ini menunjukkan SVM bersikap lebih liberal dalam memprediksi *Ovale* (*decision boundary* yang lebih luas),

memprioritaskan *high recall* (tidak melewatkan kasus *Ovale*) dengan konsekuensi *over-predicting* kelas ini. Dalam konteks medis, ini bisa dipandang sebagai *false alarm* - lebih baik salah mendeteksi *Ovale* daripada melewatkannya.

3. Konsistensi Error pada *Falciparum* dan *Malariae*

Kedua algoritma menunjukkan tingkat kesalahan (*error rate*) yang identik, yaitu 12,5%, pada kelas *Falciparum* dan *Malariae*, dengan pola misklasifikasi bilateral yang sama, di mana masing-masing algoritma saling menukar satu sampel antar kedua kelas tersebut. Konsistensi pola ini mengindikasikan bahwa tingkat kesulitan dalam membedakan kedua kelas relatif sebanding dan tidak dipengaruhi secara signifikan oleh perbedaan mekanisme algoritmik yang digunakan.

4. Konsistensi Error pada *Falciparum* dan *Malariae*

Perbedaan pola *error* antara SVM dan *Random Forest* (SVM cenderung *over-predict Ovale*, RF lebih *balanced*) menunjukkan potensi untuk *ensemble heterogen*. Kombinasi kedua algoritma melalui voting atau stacking dapat menghasilkan prediksi yang lebih *robust* dengan memanfaatkan kekuatan masing-masing algoritma.

4.7.4. Analisis Faktor Penyebab Kesalahan

Analisis Analisis kesalahan klasifikasi berikut menggabungkan temuan dari literatur (Loddo dan rekan-rekan, 2018, 2019) dengan hasil eksperimen yang dilakukan dalam penelitian ini. Data performa model (akurasi, *confidence scores*, *feature importance*, *Error Rate* per kelas) berasal dari eksperimen menggunakan dataset MP-IDB dengan konfigurasi: 400 sampel *training* (hasil augmentasi), 8 sampel *validation* per kelas, ekstraksi fitur ResNet50 (*pretrained* ImageNet), dan *classifier* SVM *Linear* serta *Random Forest* (*bootstrap*=True). Evaluasi dilakukan melalui 10-*fold cross-validation* dengan 10 repetisi pada *training set*, diikuti evaluasi final pada *independent validation set*. Berdasarkan evaluasi menyeluruh terhadap sampel yang mengalami misklasifikasi dalam eksperimen ini, dapat diidentifikasi beberapa faktor utama yang berkontribusi terhadap kesalahan klasifikasi. Faktor ini dapat dikategorikan menjadi tiga kelompok besar: faktor inheren data, faktor teknis ekstraksi fitur, dan faktor karakteristik algoritma.

1. Faktor Inheren Data: Kompleksitas Morfologi Parasit

Kesalahan klasifikasi yang konsisten pada kedua algoritma, khususnya konfusi bilateral antara *Plasmodium Falciparum* dan *Plasmodium Malariae* (masing-masing 1 sampel), serta konfusi dominan antara *Plasmodium Ovale* dan *Plasmodium Vivax* mengindikasikan adanya *overlap* fitur visual yang signifikan dalam *feature space* ResNet50. Loddo dan rekan-rekan menekankan bahwa "*the life-cycle-stage of the parasite is defined by its morphology, size and the presence or absence of malarial pigment*," mengonfirmasi bahwa variasi stadium menghasilkan konfusi dalam klasifikasi otomatis terlepas dari pemilihan algoritma klasifikasi [15]. Kompleksitas ini diperparah oleh fakta bahwa spesies berbeda

dapat menunjukkan kemiripan morfologi pada stadium yang sama, sebagaimana dijelaskan bahwa "*the species differ in the changes of infected cell's shape, presence of some characteristic dots and the morphology of the parasite in some of the life-cycle-stages*" [45].

Pola kesalahan klasifikasi yang terkonsentrasi pada pasangan spesies tertentu mencerminkan kemiripan morfologi intrinsik yang telah didokumentasikan dalam literatur klinis. Konfusi antara *Plasmodium Ovale* dan *Plasmodium Vivax* yang mencapai 37.5% pada SVM dan 25% pada Random Forest dapat dijelaskan melalui kesamaan karakteristik morfologi kedua spesies ini, khususnya pada stadium *trophozoite* dan *gametocyte* dimana kedua spesies menunjukkan bentuk sel terinfeksi yang memanjang (*elongated erythrocytes*) dan ukuran parasit yang relatif sebanding. Distribusi stadium hidup yang tidak seimbang dalam dataset MP-IDB memperparah kondisi ini, dimana *Plasmodium Falciparum* didominasi oleh stadium *ring* (695 dari 720 parasit atau 96.5%) sementara *Plasmodium Vivax* dan *Plasmodium Ovale* memiliki distribusi yang lebih merata antara stadium *ring*, *trophozoite*, dan *gametocyte* [15]. Ketidakseimbangan representasi stadium ini menyebabkan model lebih *familiar* dengan pola visual spesifik, sehingga kesulitan menggeneralisasi variasi morfologi pada stadium yang kurang terwakili.

2. Faktor Variasi Pencahayaan dan Artefak

Faktor teknis yang signifikan berasal dari variabilitas kondisi akuisisi citra mikroskopis yang melekat pada dataset MP-IDB. Loddo dan rekan-rekan mengakui bahwa dataset MP-IDB mengalami "*non-uniform background illuminations and overexposed borders due to microscope lamp illumination*" dan "*regions can have different colouration due to the age of smears*," yang mengakibatkan kondisi citra yang bervariasi [15]. Variabilitas pewarnaan Giemsa memperparah masalah ini, dimana Loddo dan rekan-rekan mencatat "*different illumination conditions generate different images because of the absence of a standardized acquisition procedure*." [45]. Kondisi ini memerlukan "*a strong pre-processing step to make the image conditions the most similar possible, in order to realize an automated procedure*" karena meskipun citra tetap *intelligible* secara visual, metode segmentasi klasik berbasis *thresholding* akan mengalami kesulitan signifikan [15].

Penelitian ini menerapkan normalisasi standar ImageNet (*mean* [0.485, 0.456, 0.406] dan standar deviasi [0.229, 0.224, 0.225]) pada tahap ekstraksi fitur ResNet50, namun pendekatan ini dirancang untuk *natural images* dengan karakteristik distribusi warna yang berbeda dari citra mikroskopis. Normalisasi *Linear* tersebut tidak mampu mengatasi variasi non-*Linear* dari intensitas pencahayaan mikroskop yang tidak seragam (*non-uniform background illuminations*) dan efek *overexposed borders* yang menciptakan gradien intensitas artifisial pada tepi *field of view*. Lebih lanjut, variabilitas kolorasi akibat usia sediaan darah (*age of the analysed smears*) mempengaruhi intensitas pewarnaan Giemsa, dimana parasit pada sediaan yang lebih tua dapat menunjukkan kontras

yang lebih rendah terhadap sitoplasma eritrosit, mengurangi *discriminative power* fitur warna yang diekstraksi oleh ResNet50.

Artefak pencahayaan ini memiliki dampak spesifik terhadap pola kesalahan klasifikasi yang diamati. Pada kasus konfusi *Plasmodium Vivax* ke *Plasmodium Ovale*, sebagaimana ditunjukkan oleh SVM *Linear* dengan 3 *false negative* dari 8 sampel *Vivax* (Tabel 33), artefak *overexposed borders* dapat menyebabkan parasit yang terletak di tepi *field of view* kehilangan detail morfologi kritis seperti *Schüffner's dots* yang merupakan penanda diagnostik penting untuk membedakan *Vivax* dari *Ovale*. Ketiadaan preprocessing morfologi eksplisit sebagaimana direkomendasikan oleh Loddo dan rekan-rekan yang menyatakan bahwa "*microscopic image analysis and particularly malaria detection and classification can greatly benefit from the use of morphological operators*" menyebabkan algoritma klasifikasi harus mengandalkan sepenuhnya pada representasi fitur ResNet50 yang tidak dioptimalkan untuk menangani artefak spesifik citra mikroskopis [45].

3. Faktor Algoritmik: Karakteristik *Decision Boundary*

Perbedaan pola kesalahan antara SVM dan Random Forest mengungkapkan pengaruh karakteristik *decision boundary* terhadap *error pattern*. SVM *Linear* menunjukkan *trade-off* ekstrem antara presisi dan *recall* pada kelas *Ovale* (presisi 0.73, *recall* 1.00) dan *Vivax* (presisi 1.00, *recall* 0.62), dengan total 3 *false positive* terhadap kelas *Ovale* yang berasal dari misklasifikasi sampel *Vivax*, namun 0 *false negative* untuk *Ovale* itu sendiri (Tabel 30 dan 33). Pola unidirectional ini mengindikasikan bahwa *hyperplane Linear* SVM menempatkan *decision boundary* yang sangat konservatif, menyebabkan region *Ovale* melebar secara berlebihan dalam *feature space* 2048 dimensi. Dalam konteks ruang fitur berdimensi tinggi yang diekstraksi dari citra dengan variasi pencahayaan, *hyperplane Linear* cenderung mengkompensasi *noise* dan variabilitas dengan memperluas margin keamanan, yang dalam kasus ini merugikan kelas *Vivax* yang memiliki representasi fitur lebih bervariasi akibat sensitivitas tinggi terhadap kondisi pencahayaan.

Sebaliknya, Random Forest menunjukkan distribusi kesalahan yang lebih simetris dengan 2 *false positive* dan 2 *false negative* untuk masing-masing kelas *Ovale* dan *Vivax*, menghasilkan presisi dan *recall* yang seimbang (0.75). Karakteristik ini mencerminkan sifat *ensemble learning* yang mengagregasi prediksi dari *multiple decision trees* dengan *random feature subsampling*, menghasilkan *decision boundaries* yang lebih "demokratis" namun dengan konsekuensi kehilangan *recall* sempurna pada kelas tertentu. Pendekatan ensemble ini lebih robust terhadap *noise* lokal yang disebabkan oleh artefak pencahayaan, karena setiap *tree* dalam ensemble mungkin menggunakan subset fitur yang berbeda dengan tingkat kontaminasi artefak yang bervariasi, sehingga agregasi voting dapat meredam pengaruh fitur yang terkontaminasi artefak.

4. Faktor Representasi: Keterbatasan Fitur ResNet50

Meskipun ResNet50 menghasilkan representasi fitur 2048 dimensi yang relatif *sparse* dengan sejumlah besar nilai mendekati nol, arsitektur ini pada dasarnya dilatih pada dataset ImageNet yang berisi objek *natural generic*. Loddo dan rekan-rekan menekankan pentingnya "*domain-specific knowledge gained by human experts*" dalam desain sistem ekstraksi fitur untuk analisis citra medis. *Transfer learning* dari ImageNet ke domain citra mikroskopis sel darah mungkin menghasilkan representasi fitur yang suboptimal, dimana fitur-fitur penting untuk membedakan spesies parasit (seperti *malarial pigment*, bentuk *infected cell*, atau *characteristic dots*) tidak terekstraksi secara optimal karena perbedaan fundamental antara karakteristik visual objek *natural* dan struktur seluler mikroskopis [45].

Analisis *feature importance* Random Forest menunjukkan bahwa fitur dengan kontribusi tertinggi (Feature 602: 0.0157) hanya berkontribusi 1.57%, dan tidak ada fitur tunggal yang mendominasi dengan kontribusi di atas 2%. Distribusi kontribusi yang merata ini mengindikasikan bahwa tidak ada fitur spesifik dari ResNet50 yang secara konsisten *capture* karakteristik diskriminatif antar spesies, melainkan klasifikasi bergantung pada kombinasi kompleks dari ratusan fitur dengan kontribusi marginal, yang meningkatkan sensitivitas terhadap *noise* dan variabilitas data. Dalam konteks citra dengan variasi pencahayaan, fitur-fitur ResNet50 yang berasal dari *convolutional layers* awal cenderung merespons terhadap gradien intensitas global yang disebabkan oleh *non-uniform illumination*, bukan terhadap struktur morfologi parasit yang sesungguhnya, sehingga mengurangi *discriminative power* representasi fitur secara keseluruhan.

5. Faktor Validitas Dataset: Keterbatasan Sampel dan Generalisasi

Loddo dan rekan-rekan menekankan bahwa "independence of sample cannot be overlooked... any system for diagnosis should be tested on different images, acquired by different sources, and different specimens," menggarisbawahi pentingnya validasi sistem pada kondisi yang bervariasi [45]. Penelitian ini menggunakan *validation* set yang relatif kecil (32 sampel dengan 8 sampel per kelas) yang berasal dari dataset MP-IDB dengan karakteristik akuisisi yang homogen (mikroskop Leica DM2000 dengan magnifikasi 100× dan pewarnaan Giemsa standar). Keterbatasan ini dapat menyebabkan overestimation performa model karena *validation* set tidak merepresentasikan variabilitas kondisi akuisisi yang akan ditemui dalam praktik klinis nyata, seperti perbedaan tipe mikroskop, protokol pewarnaan, atau tingkat keahlian teknisi laboratorium dalam preparasi sediaan darah.

6. Implikasi terhadap *Confidence Scoring*

Analisis *confidence score* mengungkapkan bahwa Random Forest dengan metode *Improved Probability Margin* menghasilkan *confidence gap* terbesar (0.1974) dengan korelasi yang baik terhadap akurasi (0.3178), namun nilai *correlation* yang relatif rendah (di bawah 0.35) pada kedua algoritma mengindikasikan bahwa *confidence scores* tidak sepenuhnya *reliable* sebagai

indikator akurasi prediksi. Hal ini berimplikasi bahwa sistem tidak dapat sepenuhnya diandalkan untuk memberikan *early warning* pada kasus-kasus yang memerlukan verifikasi manual, terutama untuk konfusi antara *Ovale* dan *Vivax* dimana kedua algoritma menunjukkan kesulitan konsisten. Keterbatasan ini kemungkinan diperparah oleh fakta bahwa *confidence scores* dihitung berdasarkan margin atau probabilitas dalam *feature space* yang telah terkontaminasi oleh variasi pencahayaan dan artefak, sehingga tingkat kepercayaan yang tinggi tidak selalu mencerminkan kebenaran prediksi ketika sampel mengalami degradasi kualitas akibat kondisi akuisisi yang suboptimal.

4.8. Pembahasan Umum

Penelitian ini berhasil mengembangkan sistem deteksi dan klasifikasi otomatis untuk penyakit infeksi darah menggunakan kombinasi YOLO 11l untuk deteksi objek dan algoritma klasifikasi (SVM dan *Random Forest*) dengan ekstraksi fitur ResNet50. Sistem diimplementasikan dalam GUI komprehensif yang mendukung diagnosis multi-kelas mencakup Malaria (empat spesies: *Falciparum*, *Malariae*, *Ovale*, *Vivax*), Anemia, dan *Trypomastigote*.

Pada tahap deteksi objek, YOLO 11l dengan *learning rate* 0.001 menunjukkan performa optimal dengan mAP@0.5 sebesar 85.9% dan akurasi klasifikasi set validasi 96.8%. Meskipun *learning rate* 0.0001 menawarkan konvergensi tercepat (8.053 jam), dan *learning rate* 0.0015 mencapai akurasi klasifikasi validasi tertinggi (98.4%), *trade-off* antara performa deteksi, stabilitas konvergensi, dan efisiensi waktu menjadikan *learning rate* 0.001 sebagai konfigurasi yang paling seimbang untuk implementasi praktis. Kesalahan deteksi terkonsentrasi pada pasangan Anemia dan Malaria akibat overlap morfologi dan variasi kualitas citra.

Tahap klasifikasi spesies malaria menggunakan fitur ResNet50 (2048 dimensi) menunjukkan bahwa SVM *Linear* unggul dengan akurasi *cross-validation* 99.25% dan akurasi validasi 84.38%, melampaui *Random Forest* (97.50% dan 81.25%). Analisis error mengungkapkan kesalahan klasifikasi konsisten pada pasangan *Ovale* dan *Vivax*, mencerminkan kemiripan morfologi intrinsik pada stadium *trophozoite* dan *gametocyte*. Perbedaan pola error antara kedua algoritma dimana SVM menunjukkan bias *unidirectional* (*over-predict Ovale*) sementara *Random Forest* lebih seimbang mencerminkan karakteristik *decision boundary* yang berbeda.

Faktor penyebab kesalahan mencakup tiga aspek utama: (1) kompleksitas morfologi parasit dengan distribusi stadium yang tidak seimbang dalam dataset, (2) variabilitas teknis citra berupa pencahayaan tidak seragam dan artefak yang tidak teratasi oleh normalisasi standar ImageNet, dan (3) keterbatasan representasi fitur ResNet50 yang dilatih pada objek *natural generic*, bukan citra mikroskopis medis. Analisis *feature importance* menunjukkan tidak ada fitur dominan (kontribusi tertinggi 1.57%), mengindikasikan klasifikasi bergantung

pada kombinasi kompleks ratusan fitur dengan kontribusi marginal yang sensitif terhadap *noise*.

Evaluasi *confidence scoring* menunjukkan *Random Forest Improved Margin Method* menghasilkan kombinasi skor tertinggi (0.2696) dengan *confidence* gap terbesar (0.1974), mendemonstrasikan kemampuan superior dalam mengkuantifikasi ketidakpastian prediksi. Namun, korelasi *confidence* terhadap akurasi yang relatif rendah (di bawah 0.35) pada kedua algoritma mengindikasikan keterbatasan reliabilitas untuk *early warning* sistem. Keterbatasan validasi set yang kecil (32 sampel) dengan karakteristik akuisisi homogen berpotensi menyebabkan overestimasi performa model terhadap variabilitas kondisi klinis riil.

Sistem yang dikembangkan mendemonstrasikan kelayakan teknis untuk deteksi dan klasifikasi multi-kelas penyakit infeksi darah, namun implementasi klinis memerlukan validasi lebih luas pada dataset heterogen dengan variasi kondisi akuisisi, pengembangan preprocessing khusus untuk mengatasi artefak citra mikroskopis, serta eksplorasi arsitektur ekstraksi fitur yang dioptimalkan untuk domain medis.

Bab 5. Kesimpulan dan Saran

5.1. Kesimpulan

Penelitian ini berhasil mengembangkan sistem deteksi dan klasifikasi malaria berbasis kecerdasan buatan yang mengintegrasikan YOLO11l untuk deteksi objek, ResNet50 untuk ekstraksi fitur, serta SVM dan *Random Forest* untuk klasifikasi spesies parasit. Evaluasi komprehensif menggunakan *K-Fold Cross-Validation* dan *validation set* independen menghasilkan temuan-temuan berikut:

1. Performa Deteksi YOLO11l

   Model YOLO11l dengan *learning rate* 0,001 mencapai *mAP*@0.5 sebesar 85,9% dengan kecepatan inferensi 27,2 ms per citra, memadai untuk implementasi. Model berhasil mendeteksi tiga kelas objek (Anemia, Malaria, *Trypomastigote*) dengan akurasi tinggi pada dataset uji maupun validasi.

2. Superioritas SVM *Linear*

   SVM dengan *kernel Linear* menunjukkan performa terbaik dengan akurasi 99,25%±0,68% pada *K-Fold Cross-Validation* dan 84,38% pada *validation set*, mengungguli *Random Forest* (97,50%±1,13% dan 81,25%). Konsistensi dan stabilitas SVM *Linear* lebih tinggi, ditunjukkan oleh standar deviasi terendah dan *precision* validasi tertinggi (86,93%).

3. Ekstraksi Fitur ResNet50

   Ekstraksi fitur menggunakan ResNet50 *pre-trained* menghasilkan vektor 2.048 dimensi yang informatif untuk membedakan empat spesies malaria. Namun, *training-validation gap* yang signifikan (SVM: 14,87%, RF: 16,88%) mengindikasikan keterbatasan fundamental akibat *domain mismatch* antara ImageNet dan citra mikroskopik malaria.

4. Kontribusi Sistem GUI

   Sistem menyediakan GUI komprehensif dengan 15 fitur utama mencakup *pipeline* AI multi-model, visualisasi deteksi interaktif, analitik multi-tab, ekspor hasil format *multiple*, dan *persistent storage*, menjembatani kompleksitas teknis dengan kebutuhan praktis diagnosis klinis.

5. Algoritma Optimal untuk Klasifikasi Malaria

   SVM *kernel Linear* ditetapkan sebagai algoritma optimal untuk klasifikasi malaria berbasis deteksi YOLO11l, dengan keunggulan pada akurasi, konsistensi, *precision*, efisiensi komputasi, dan interpretabilitas model.

5.2. Saran

Pengembangan sistem diagnosis malaria berbasis kecerdasan buatan ke depan perlu memprioritaskan penguatan basis data melalui penggunaan *dataset* yang lebih besar, beragam secara geografis, dan terstratifikasi berdasarkan stadium parasit, yang disertai dengan standarisasi protokol akuisisi citra. Hal ini harus didukung oleh optimalisasi metode ekstraksi fitur, baik melalui pengembangan arsitektur *deep learning* yang spesifik untuk domain citra mikroskopis, *fine-tuning* menyeluruh, maupun eksplorasi model adaptif seperti *Vision Transformers* (ViT) yang dikombinasikan dengan fitur morfologi klinis untuk mengatasi masalah *domain mismatch*.

Di sisi lain, untuk menjamin keandalan implementasi praktis, penelitian lanjutan harus melakukan uji klinis prospektif dengan melibatkan ahli mikroskopi serta melakukan analisis efektivitas biaya. Fokus pengembangan juga perlu diarahkan pada teknologi *edge computing* agar sistem dapat beroperasi pada perangkat berspesifikasi rendah di daerah dengan infrastruktur terbatas. Terakhir, integrasi riset fundamental seperti *Explainable* AI (XAI) dan *multi-task learning* sangat diperlukan untuk memastikan transparansi, efisiensi, dan akurasi diagnosis yang mampu bersaing dengan metode konvensional.

Daftar Pustaka

[1] S. Suraksha, C. Santhosh, and B. Vishwa, "Classification of Malaria cell images using Deep Learning Approach," *2023 3rd Int. Conf. Adv. Electr. Comput. Commun. Sustain. Technol. ICAECT 2023*, pp. 1–5, 2023, doi: 10.1109/ICAECT57570.2023.10117649.

[2] S. Mathupriya, V. Raj Ronald Shaw, and M. Nihaal Tharwat, "Malaria Disease Prediction using Faster RCNN," *2023 Intell. Comput. Control Eng. Bus. Syst. ICCEBS 2023*, pp. 1–6, 2023, doi: 10.1109/ICCEBS58601.2023.10448699.

[3] S. Aluvala, K. Bhargavi, J. Deekshitha, B. Suresh, G. N. Rao, and A. Sravani, "A Web-Based Approach for Malaria Parasite Detection Using Deep Learning in Blood Smears," *2024 2nd World Conf. Commun. Comput. WCONF 2024*, pp. 1–6, 2024, doi: 10.1109/WCONF61366.2024.10692067.

[4] A. Kurniawan, A. Z. Widyawati, F. A. Pratiwi, N. Faizun, and R. Meliana, "Pemodelan Prevalensi Angka Kesakitan Malaria Berdasarkan Persentase Sanitasi Layak Dengan Pendekatan Estimator Least Square Spline," vol. 23, no. 3, pp. 287–293, 2024.

[5] M. S. Nascimento, M. G. F. Costa, and C. F. F. Costa Filho, "Detection of Malaria Parasites in Thick Blood Smear Images using Shallow Neural Networks and Digital Image Processing Techniques," *Proc. 19th Int. Symp. Med. Inf. Process. Anal. SIPAIM 2023*, pp. 1–4, 2023, doi: 10.1109/SIPAIM56729.2023.10373464.

[6] S. Shashikiran and H. D. Sunitha, "Malaria Cell Detection using Advanced SVM Techniques," *2nd IEEE Int. Conf. Networks, Multimed. Inf. Technol. NMITCON 2024*, pp. 1–6, 2024, doi: 10.1109/NMITCON62075.2024.10699300.

[7] W. R. W. M. Razin, T. S. Gunawan, M. Kartiwi, and N. M. Yusoff, "Malaria Parasite Detection and Classification using CNN and YOLOv5 Architectures," *8th IEEE Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2022*, pp. 277–281, 2022, doi: 10.1109/ICSIMA55652.2022.9928992.

[8] K. K. Jena, S. Kumar Bhoi, C. Mallick, D. Mohapatra, and P. Swain, "Classification of Malaria Parasitized and Uninfected Images Using Machine Learning Approach," *Proc. 5th Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2021*, pp. 1274–1279, 2021, doi: 10.1109/I-SMAC52330.2021.9640905.

[9] R. Z. Lekeufack Foulefack, A. A. Akinyelu, D. Ranirina, and B. N. Mpinda, "Malaria Parasite Detection in Microscopic Blood Smear Images Using Deep Learning Techniques," *2024 Int. Jt. Conf. Neural Networks*, pp. 1–8,

2024, doi: 10.1109/ijcnn60899.2024.10651385.

[10] A. Paul and R. K. Bania, “Malaria Parasite Classification using Deep Convolutional Neural Network,” *2021 Int. Conf. Comput. Intell. Comput. Appl. ICCICA 2021*, pp. 1–6, 2021, doi: 10.1109/ICCICA52458.2021.9697307.

[11] G. E. Sukendar, D. S. S. Rejeki, and D. Anandari, “Studi Endemisitas dan Epidemiologi Deskriptif Malaria di Kabupaten Purbalingga Tahun 2010-2019,” *J. Epidemiol. Kesehat. Indones.*, vol. 5, no. 1, 2021, doi: 10.7454/epidkes.v5i1.4625.

[12] E. Setia Budi, A. Nofriyaldi Chan, P. Priscillia Alda, and M. Arif Fauzi Idris, “RESOLUSI : Rekayasa Teknik Informatika dan Informasi Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra Menggunakan Algoritma Convolutional Neural Network,” *Media Online*, vol. 4, no. 5, p. 509, 2024, [Online]. Available: https://djournals.com/resolusi

[13] E. Farmi, A. Gusti, and M. Masrizal, “Penyakit Malaria Berdasarkan Faktor Resiko Demografis Lingkungan Dengan Pendekatan Spasial (Systematic Review),” *Jik J. Ilmu Kesehat.*, vol. 7, no. 2, p. 489, 2023, doi: 10.33757/jik.v7i2.721.

[14] Hermayani and T. Novianty Mansyur, “Tinjauan Literatur Analisis Insidensi Faktor Resiko Kejadian Malaria pada Balita di Wilayah Endemik,” *ProHealth J.*, vol. 21, no. 1, pp. 12–20, 2024, doi: 10.59802/phj.2024211126.

[15] A. Loddo, C. Di Ruberto, M. Kocher, and G. Prod’hom, *Mp-idb: The malaria parasite image database for image processing and analysis*, vol. 11379. Springer International Publishing, 2019. doi: 10.1007/978-3-030-13835-6_7.

[16] I. Jawaz and Reni Rahmadewi, “Sistem Deteksi Pneumonia Paru-Paru dengan Pengolahan Citra Digital dan Machine Learning,” *ELECTRON J. Ilm. Tek. Elektro*, vol. 5, no. 1, 2024, doi: 10.33019/electron.v5i1.114.

[17] M. Y. Khairi, E. Acantha, M. Sampetoding, and Y. S. Pongtambing, “Studi Literatur Penerapan Deep Learning dalam Analisis Citra Medis di Indonesia,” vol. 01, no. 01, pp. 15–24, 2024.

[18] F. A. Shidik, K. Musthofa, and A. P. Kartiningtyas, “Analisis citra medis untuk identifikasi penyakit mata dengan teknologi convolutional neural networks,” no. November, pp. 68–80, 2024.

[19] S. A. Wibowo, W. Andriani, J. P. No, K. Tegal, and J. Tengah, “Evaluasi Model Deep Learning pada Pola Dataset Biomedis segmentasi multi-kelas berbasis deep lesi gigi dari gambar medis (Zhang et al .,” vol. 14, no. 2, pp. 195–207, 2024.

[20] K. Anwar, R. Maruf, F. Susanto, and M. B. Ryando, “Analisis Performa Deteksi Penyakit Paru-Paru dengan Model Klasifikasi Gambar Menggunakan LSTM Deep Learning,” vol. 25, no. 1, pp. 972–979, 2025,

doi: 10.33087/jiubj.v25i1.5697.
[21] C. Y. K. Nidhal Jegham, Marwan F. Abdelatti, "YOLO Evolution : A Comprehensive Benchmark and Architectural Review of YOLO Evolution : A Comprehensive Benchmark and Architectural Review of YOLOv12 , YOLO11 , and Their Previous Versions," no. March, 2025, doi: 10.13140/RG.2.2.15952.83201.
[22] Z. Hua, K. Aranganadin, and C. Yeh, "A Benchmark Review of YOLO Algorithm Developments for Object Detection," no. May, pp. 123515–123545, 2025.
[23] G. A. Santiago *dan rekan-rekan*, "Object Detection Based on You Look Only Once Version 8 for Real-Time Applications".
[24] D. Triyanto, M. Zidan, M. Wahyudi, and L. Pujiastuti, "Pengembangan Sistem Deteksi Objek Botol Real-Time dengan YOLOv8 untuk Aplikasi Vision," vol. 3, no. 1, 2024.
[25] H. Honnainah and R. Enggar Pawening, "Deteksi Otomatis Terhadap Pelanggaran Pembuang Sampah Menggunakan Metode You Only Look Once (YOLO)," *TRILOGI J. Ilmu Teknol. Kesehatan, dan Hum.*, vol. 4, no. 2, pp. 98–105, 2023, doi: 10.33650/trilogi.v4i2.6676.
[26] W. Wei, Y. Huang, J. Zheng, Y. Rao, Y. Wei, and X. Tan, "Journal of Radiation Research and Applied Sciences YOLOv11-based multi-task learning for enhanced bone fracture detection and classification in X-ray images," *J. Radiat. Res. Appl. Sci.*, vol. 18, no. 1, p. 101309, 2025, doi: 10.1016/j.jrras.2025.101309.
[27] A. Sharma, V. Kumar, and L. Longchamps, "Smart Agricultural Technology Faster R-CNN models for detection of multiple weed species," *Smart Agric. Technol.*, vol. 9, no. November, 2024.
[28] M. Mujahid *dan rekan-rekan*, "Efficient deep learning-based approach for malaria detection using red blood cell smears," *Sci. Rep.*, vol. 14, no. 1, pp. 1–16, 2024, doi: 10.1038/s41598-024-63831-0.
[29] R. Adenia, A. E. Minarno, and Y. Azhar, "Implementasi Convolutional Neural Network Untuk Ekstraksi Fitur Citra Daun Dalam Kasus Deteksi Penyakit Pada Tanaman Mangga Menggunakan Random Forest," *J. Repos.*, vol. 4, no. 4, pp. 473–482, 2024, doi: 10.22219/repositor.v4i4.32287.
[30] I. G. T. Permana, I. B. G. Dwidasmara, M. A. Raharja, and I. W. Santiyasa, "Ekstraksi Fitur Dengan Convolutional Neural Network Dan Rekomendasi Fashion Menggunakan Algoritma K-Nearest Neighbours," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 12, no. 4, p. 845, 2024, doi: 10.24843/jlk.2024.v12.i04.p10.
[31] A. Peryanto, A. Yudhana, and R. Umar, "Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross *Validation*," *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 45–51, 2020, doi:

10.30871/jaic.v4i1.2017.
[32] A. F. Saksenata, A. E. Minarno, and Y. Azhar, "Klasifikasi Citra Sel Darah Untuk Penyakit Malaria Dengan Metode CNN," *J. Repos.*, vol. 4, no. 2, pp. 185–194, 2022, doi: 10.22219/repositor.v4i2.1283.
[33] Fauzan Muhammad, Aniati Murni Arimurthy, and Dina Chahyati, "Transfer learning pada Network VGG16 dan ResNet50," *Indones. J. Comput. Sci.*, vol. 12, no. 1, pp. 361–374, 2023, doi: 10.33022/ijcs.v12i1.3130.
[34] K. Ohdar and A. Nigam, "A Robust Approach for Malaria Parasite Identification with CNN Based Feature Extraction and Classification using SVM," *2023 14th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2023*, pp. 1–6, 2023, doi: 10.1109/ICCCNT56998.2023.10306840.
[35] K. L. Kohsasih, M. Zarlis, and B. H. Hayadi, "Comparison of CNN Architecture for White Blood Cells Image Classification," *ICOSNIKOM 2022 - 2022 IEEE Int. Conf. Comput. Sci. Inf. Technol. Bound. Free Prep. Indones. Metaverse Soc.*, pp. 1–7, 2022, doi: 10.1109/ICOSNIKOM56551.2022.10034875.
[36] A. Muhandisin and Y. Azhar, "Malaria Blood Cell Image Classification using Transfer Learning with Fine-Tune ResNet50 and Data Augmentation," *J. RESTI*, vol. 6, no. 5, pp. 891–897, 2022, doi: 10.29207/resti.v6i5.4322.
[37] J. Amin, M. A. Anjum, A. Ahmad, M. I. Sharif, S. Kadry, and J. Kim, "Microscopic parasite malaria classification using *best* feature selection based on generalized normal distribution optimization," *PeerJ Comput. Sci.*, vol. 10, pp. 1–23, 2024, doi: 10.7717/peerj-cs.1744.
[38] S. Rajaraman *dan rekan-rekan*, "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images," *PeerJ*, vol. 2018, no. 4, pp. 1–17, 2018, doi: 10.7717/peerj.4568.
[39] A. Mulyani, S. Khoerunisa, and D. Kurniadi, "Comparison of KNN and SVM Algorithms Performance Using SMOTE to Classify Diabetes," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 14, no. 1, pp. 25–34, 2025.
[40] A. G. Taye *dan rekan-rekan*, "Automated Web-Based Malaria Detection System with Machine Learning and Deep Learning Techniques," *Commun. Comput. Inf. Sci.*, 2024, [Online]. Available: https://arxiv.org/abs/2407.00120v1
[41] K. Munawaroh and A. Alamsyah, "Performance Comparison of SVM, Naïve Bayes, and KNN Algorithms for Analysis of Public Opinion Sentiment Against COVID-19 Vaccination on Twitter," *J. Adv. Inf. Syst. Technol.*, vol. 4, no. 2, pp. 113–125, 2023, doi: 10.15294/jaist.v4i2.59493.
[42] Suci Amaliah, M. Nusrang, and A. Aswi, "Penerapan Metode Random Forest Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konijiwa Bantaeng," *VARIANSI J. Stat. Its Appl. Teach. Res.*, vol. 4, no. 3, pp. 121–

127, 2022, doi: 10.35580/variansiunm31.

[43] H. Hairani, A. Anggrawan, and D. Priyanto, “Improvement Performance of the Random Forest Method on Unbalanced Diabetes Data Classification Using Smote-Tomek Link,” *Int. J. Informatics Vis.*, vol. 7, no. 1, pp. 258–264, 2023, doi: 10.30630/joiv.7.1.1069.

[44] D. N. Jawza, M. I. Mazdadi, A. Farmadi, T. H. Saragih, D. Kartini, and V. Abdullayev, “Enhancing Diabetes Prediction *Accuracy* Using Random Forest and XGBoost with PSO and GA-Based Feature Selection,” *J. Electron. Electromed. Eng. Med. Informatics*, vol. 7, no. 2, pp. 295–306, 2025, doi: 10.35882/jeeemi.v7i2.626.

[45] A. Loddo, C. Di Ruberto, and M. Kocher, “Recent advances of malaria parasites detection systems based on mathematical morphology,” *Sensors (Switzerland)*, vol. 18, no. 2, pp. 1–21, 2018, doi: 10.3390/s18020513.

Biodata

Nama	: Muhammad Rezky Fatya
TTL	: Batam, 04 Juni 2003
Agama	: Islam
Alamat	: Perum GMP Blok F No.71
Email	: rr913611@gmail.com
Riwayat Pendidikan	SMA/SMK : SMKN 3 Batam
	SMP : SMPN 40 Batam

Nama	: Muhammad Fikri
TTL	: Batam, 27 November 2003
Agama	: Islam
Alamat	: Bida Ayu Blok T No.06
Email	: fikrialbadawi@gmail.com
Riwayat Pendidikan	SMA/SMK : SMKN 3 Batam
	SMP : MTs Raudlatul Quran Batam