

**KLASIFIKASI PENYAKIT MALARIA DISEBABKAN
PARASIT *PLASMODIUM VIVAX* BERDASARKAN CITRA SEL
DARAH MENGGUNAKAN *CONVOLUTIONAL NEURAL
NETWORK***

TUGAS AKHIR

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1)
Program Studi Teknik Informatika Fakultas Teknologi Industri,
Institut Teknologi Sumatera

Oleh:
MOHAMAD MEAZZA APRILIANDA
121140216



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2025**

RINGKASAN

KLASIFIKASI PENYAKIT MALARIA DISEBABKAN PARASIT *PLASMODIUM VIVAX* BERDASARKAN CITRA SEL DARAH MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

Mohamad Meazza Aprilianda

Malaria merupakan salah satu penyakit menular yang masih menjadi permasalahan kesehatan global, khususnya di negara-negara tropis seperti Indonesia. Penyakit ini disebabkan oleh parasit *Plasmodium* dan ditularkan melalui gigitan nyamuk *Anopheles* betina. Di antara beberapa spesies penyebab malaria, *Plasmodium vivax* merupakan salah satu yang paling banyak ditemukan di Indonesia. Parasit ini memiliki kemampuan unik membentuk hipnozoit di hati manusia yang dapat menyebabkan kekambuhan infeksi dalam jangka waktu lama. Diagnosis malaria secara konvensional umumnya dilakukan melalui pemeriksaan mikroskopis, Rapid Diagnostic Test (RDT), atau Polymerase Chain Reaction (PCR). Namun, metode-metode tersebut masih memiliki keterbatasan dalam hal akurasi dan ketergantungan terhadap tenaga medis ahli. Oleh karena itu, diperlukan inovasi berbasis teknologi kecerdasan buatan untuk meningkatkan kecepatan dan ketepatan diagnosis malaria, salah satunya melalui penerapan *deep learning* dengan arsitektur *Convolutional Neural Network* (CNN).

Penelitian ini bertujuan untuk mengklasifikasikan penyakit malaria yang disebabkan oleh *Plasmodium vivax* berdasarkan citra sel darah menggunakan dua arsitektur CNN, yaitu MobileNetV2 dan EfficientNetB0. Penelitian ini difokuskan untuk membandingkan performa kedua model tersebut berdasarkan metrik evaluasi *accuracy*,

precision, recall, dan f1-score untuk menentukan arsitektur yang paling optimal untuk klasifikasi citra sel darah.

Rumusan masalah penelitian ini adalah bagaimana membandingkan performa MobileNetV2 dan EfficientNetB0 dalam melakukan klasifikasi penyakit malaria yang disebabkan oleh *Plasmodium vivax*. Penelitian dibatasi pada pengklasifikasian lima kelas citra sel darah, yaitu *uninfected, ring, trophozoite, schizont, dan gametocyte*. Dataset yang digunakan berasal dari Broad Bioimage Benchmark Collection (BBBC041 Version 1) dengan total 780 citra berformat PNG dan JPG.

Metodologi penelitian terdiri atas beberapa tahap, dimulai dengan identifikasi masalah untuk memahami potensi CNN dalam mendeteksi malaria secara otomatis. Tahap berikutnya adalah studi literatur untuk mengkaji penelitian-penelitian terdahulu yang menggunakan CNN dalam klasifikasi citra medis. Setelah itu dilakukan pengumpulan data dari dataset BBBC041 yang telah dianotasi sesuai stadium perkembangan parasit *Plasmodium vivax*. Tahap prapemrosesan data meliputi *cropping* untuk memotong bagian citra berdasarkan bounding box setiap kelas, *resize* untuk menyamakan ukuran citra menjadi 224x224 piksel, serta normalisasi dengan menggunakan nilai rata-rata dan standar deviasi dari dataset ImageNet agar distribusi piksel lebih stabil. Selanjutnya dilakukan pengembangan model dengan menerapkan metode *k-fold cross validation* untuk memastikan hasil evaluasi lebih objektif. Hyperparameter yang digunakan mencakup *optimizer* (Adam dan RMSProp), *learning rate* (1e-4 dan 1e-3), jumlah *epoch* (50 dan 100), serta *batch size* sebesar 32. Evaluasi model dilakukan menggunakan *confusion matrix* metrik *accuracy, precision, recall, dan f1-score*.

Hasil pelatihan menunjukkan bahwa MobileNetV2 dengan kombinasi *optimizer* RMSProp, *learning rate* 0,001, *epoch* 100, dan *batch size* 32 menghasilkan rata-rata *accuracy* 0,799, *precision* 0,807, *recall* 0,799, dan *f1-score* 0,795. Sementara itu, EfficientNetB0 dengan kombinasi parameter yang sama menghasilkan *accuracy* 0,832, *precision* 0,835, *recall* 0,832, dan *f1-score* 0,832. Berdasarkan hasil ini, EfficientNetB0 memiliki performa yang lebih unggul dibandingkan MobileNetV2.

Hasil *confusion matrix* memperlihatkan bahwa EfficientNetB0 mampu memprediksi dengan benar 134 dari 156 data pada lipatan terbaik, sedangkan MobileNetV2 memprediksi dengan benar 129 dari 156 data. Pada kelas *gametocyte*, nilai *precision*, *recall*, dan *f1-score* untuk EfficientNetB0 masing-masing sebesar 0,826, 0,903, dan 0,862, yang menunjukkan kemampuan model dalam mendekripsi pada tahap tersebut.

Meskipun EfficientNetB0 unggul dalam akurasi, hasil analisis metrik per parameter menunjukkan bahwa MobileNetV2 lebih efisien secara komputasi. Nilai metrik per parameter pada MobileNetV2 adalah 3,58e-7 untuk *accuracy* dan *recall*, 3,62e-7 untuk *precision*, serta 3,56e-7 untuk *f1-score*. Sedangkan EfficientNetB0 memiliki nilai 2,07e-7 untuk *accuracy*, *recall*, dan *f1-score*, serta 2,08e-7 untuk *precision*. Hal ini menunjukkan bahwa meskipun performa EfficientNetB0 lebih tinggi, MobileNetV2 memerlukan sumber daya komputasi yang lebih sedikit.

Hasil pengujian pada beberapa citra menunjukkan bahwa kedua model mampu mengklasifikasikan kelas *schizont*, *uninfected*, dan *ring* dengan benar. Namun, MobileNetV2 masih menunjukkan kesalahan klasifikasi pada kelas *trophozoite* dan *gametocyte*, sedangkan

EfficientNetB0 menunjukkan kesalahan pada kelas *trophozoite*. Penelitian ini berhasil menunjukkan bahwa kedua arsitektur CNN memiliki potensi yang baik dalam mengklasifikasikan penyakit malaria berbasis citra sel darah, dengan EfficientNetB0 sebagai model yang paling optimal dari sisi performa dan MobileNetV2 sebagai model yang unggul dalam efisiensi komputasi.

Sebagai kesimpulan, EfficientNetB0 menghasilkan kinerja terbaik dengan nilai rata-rata *accuracy* 0,832, *precision* 0,835, *recall* 0,832, dan *f1-score* 0,832, sementara MobileNetV2 memperoleh nilai rata-rata *accuracy* 0,799, *precision* 0,807, *recall* 0,799, dan *f1-score* 0,795. Hasil ini menunjukkan bahwa EfficientNetB0 memiliki kemampuan performa lebih baik, sedangkan MobileNetV2 lebih sedikit dalam komputasi.

Penelitian ini diharapkan dapat berkontribusi pada pengembangan sistem deteksi otomatis penyakit malaria berbasis citra mikroskopis. Untuk penelitian selanjutnya disarankan untuk menerapkan teknik prapemrosesan lanjutan seperti penyesuaian kontras dan pencahayaan, serta menggunakan *data augmentation* untuk meningkatkan keragaman data pelatihan. Selain itu, model yang dikembangkan dapat diimplementasikan dalam bentuk aplikasi berbasis web atau mobile.

ABSTRAK

Malaria merupakan penyakit menular yang masih menjadi permasalahan kesehatan global, khususnya di wilayah tropis seperti Indonesia. Penyakit ini disebabkan oleh parasit *Plasmodium* yang ditularkan melalui gigitan nyamuk *Anopheles* betina, dengan *Plasmodium vivax* sebagai salah satu spesies yang paling umum ditemukan. Penelitian ini bertujuan untuk membandingkan kinerja dua arsitektur *Convolutional Neural Network* (CNN), yaitu MobileNetV2 dan EfficientNetB0 dalam mengklasifikasikan penyakit malaria yang disebabkan oleh *Plasmodium vivax* berdasarkan citra sel darah. Dataset yang digunakan berasal dari Broad Bioimage Benchmark Collection (BBBC041 Version 1) dengan total 780 citra sel darah yang terbagi menjadi lima kelas: *uninfected*, *ring*, *trophozoite*, *schizont*, dan *gametocyte*. Proses penelitian meliputi *cropping*, *resize*, normalisasi, dan pelatihan model menggunakan metode *k-Fold Cross Validation* dengan variasi *hyperparameter* berupa *optimizer* (Adam, RMSProp), *learning rate* (1e-4, 1e-3), *epoch* (50, 100), dan *batch size* (32). Hasil penelitian menunjukkan bahwa EfficientNetB0 memiliki performa terbaik dengan rata-rata *accuracy* 0,832, *precision* 0,835, *recall* 0,832, dan *f1-score* 0,832, sedangkan MobileNetV2 memperoleh *accuracy* 0,799 dan *f1-score* 0,795. Meskipun demikian, MobileNetV2 lebih efisien secara komputasi. Penelitian ini menunjukkan bahwa penerapan CNN dalam klasifikasi citra sel darah malaria dengan EfficientNetB0 sebagai model dengan performa optimal dan MobileNetV2 sebagai model lebih baik pada komputasi.

Kata Kunci: Malaria, *Plasmodium vivax*, Klasifikasi, MobileNetV2, EfficientNetB0

ABSTRACT

Malaria remains a major global health concern, particularly in tropical regions such as Indonesia. The disease is caused by the Plasmodium parasite transmitted through the bite of female Anopheles mosquitoes, with Plasmodium vivax being one of the most prevalent species. This study aims to compare the performance of two Convolutional Neural Network (CNN) architectures, MobileNetV2 and EfficientNetB0, in classifying malaria infection caused by Plasmodium vivax based on blood cell images. The dataset used in this research was obtained from the Broad Bioimage Benchmark Collection (BBBC041 Version 1), consisting of 780 blood cell images divided into five classes: uninfected, ring, trophozoite, schizont, and gametocyte. The research process included image cropping, resizing, normalization, and model training using the k-Fold Cross Validation method with various hyperparameters, including optimizers (Adam, RMSProp), learning rates (1e-4, 1e-3), epochs (50, 100), and batch size (32). The experimental results show that EfficientNetB0 achieved the best performance with an average accuracy of 0.832, precision of 0.835, recall of 0.832, and F1-score of 0.832, while MobileNetV2 achieved an accuracy of 0.799 and F1-score of 0.795. However, MobileNetV2 demonstrated greater computational efficiency due to its smaller number of parameters. This study demonstrates the effectiveness of CNN-based models for classifying malaria blood cell images, with EfficientNetB0 providing optimal performance and MobileNetV2 offering better computational efficiency.

Keywords: Malaria, Plasmodium vivax, Classification, MobileNetV2, EfficientNetB0

DAFTAR ISI

RINGKASAN	ii
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
DAFTAR RUMUS	xv
DAFTAR KODE	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah	5
1.5 Manfaat Penelitian.....	5
1.6 Sistematika Penulisan	5
1.6.1 Bab I Pendahuluan.....	6
1.6.2 Bab II Tinjauan Pustaka	6
1.6.3 Bab III Metodologi Penelitian	6
1.6.4 Bab IV Hasil dan Pembahasan	6
1.6.5 Bab V Kesimpulan dan Saran.....	6
BAB II TINJAUAN PUSTAKA.....	7

2.1	Tinjauan Pustaka	7
2.2	Dasar Teori	11
2.2.1	Malaria.....	12
2.2.2	<i>Plasmodium Vivax</i>	13
2.2.3	Stadium <i>Plasmodium Vivax</i>	14
2.2.3.1	Stadium <i>Ring</i>	15
2.2.3.2	Stadium <i>Trophozoite</i>	15
2.2.3.3	Stadium <i>Schizont</i>	16
2.2.3.4	Stadium <i>Gametocyte</i>	17
2.2.4	Citra	18
2.2.5	Sel Darah	19
2.2.6	<i>Deep Learning</i>	20
2.2.7	<i>Convolutional Neural Network (CNN)</i>	21
2.2.7.1	<i>Convolutional Layer</i>	22
2.2.7.2	<i>Rectified Linear Unit</i>	23
2.2.7.3	<i>Pooling Layer</i>	24
2.2.7.4	<i>Fully Connected Layer</i>	25
2.2.7.5	<i>Softmax</i>	26
2.2.8	MobileNetV2.....	26
2.2.9	EfficientNetB0.....	28
2.2.10	<i>K-Fold Cross Validation</i>	29
2.2.11	<i>Confusion Matrix</i>	30

BAB III METODE PENELITIAN	34
3.1 Alur Penelitian.....	34
3.2 Penjabaran Langkah Penelitian.....	35
3.2.1 Identifikasi Masalah	35
3.2.2 Studi Literatur.....	35
3.2.3 Pengumpulan Data.....	36
3.2.4 Prapemrosesan Data	37
3.2.4.1 <i>Cropping</i>	37
3.2.4.2 <i>Resize</i>	38
3.2.4.3 Normalisasi.....	38
3.2.5 Pengembangan Model	39
3.2.5.1 Pembagian Data.....	39
3.2.5.2 Pelatihan Model.....	40
3.2.6 Evaluasi Model.....	42
3.2.7 Hasil dan Pembahasan	42
3.3 Alat dan Bahan Tugas Akhir	42
3.3.1 Alat	43
3.3.2 Bahan.....	43
3.4 Metode Pengembangan.....	43
3.5 Ilustrasi Perhitungan Metode.....	43
3.5.1 <i>Convolutional Layer</i>	44
3.5.2 <i>Pooling Layer</i>	48

3.5.3	<i>Fully Connected Layer</i>	49
3.5.4	<i>Softmax</i>	51
3.6	Ilustrasi Rancangan Pengujian	52
BAB IV HASIL DAN PEMBAHASAN		56
4.1	Hasil Penelitian	56
4.1.1	Pengumpulan Data	56
4.1.2	Prapemrosesan Data	56
4.1.2.1	<i>Cropping</i>	57
4.1.2.2	<i>Resize</i>	58
4.1.2.3	Normalisasi	59
4.1.3	Pengembangan Model	60
4.1.3.1	Pembagian Data	60
4.1.3.2	Pelatihan Model	63
4.2	Hasil Evaluasi	67
4.2.1	Model MobileNetV2	68
4.2.2	Model EfficientNetB0	72
4.3	Perbandingan Evaluasi Model	76
4.4	Pengujian Model	77
4.5	Analisis Hasil Keseluruhan	79
BAB V KESIMPULAN DAN SARAN		81
5.1	Kesimpulan	81
5.2	Saran	81

DAFTAR PUSTAKA	83
LAMPIRAN.....	91

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka	9
Tabel 2.2 <i>K-Fold Cross Validation</i>	30
Tabel 2.3 <i>Confusion Matrix</i> Klasifikasi Biner	31
Tabel 2.4 <i>Confusion Matrix</i> Klasifikasi <i>Multiclass</i>	31
Tabel 3.1 Variasi Hyperparameter	41
Tabel 3.2 Contoh Tabel Hasil <i>Confusion Matrix</i>	53
Tabel 4.1 Pembagian Data	62
Tabel 4.2 Hasil Evaluasi Model MobileNetV2	68
Tabel 4.3 Hasil Evaluasi EfficientNetB0	72
Tabel 4.4 Hasil Perbandingan Evaluasi Model Terbaik.....	76

DAFTAR GAMBAR

Gambar 2.1 Nyamuk Anopheles	12
Gambar 2.2 Siklus Hidup Plasmodium	14
Gambar 2.3 Stadium <i>Ring</i>	15
Gambar 2.4 Stadium <i>Trophozoite</i>	16
Gambar 2.5 Stadium <i>Schizont</i>	17
Gambar 2.6 Stadium <i>Gametocyte</i>	18
Gambar 2.7 Arsitektur <i>Convolutional Neural Network</i>	21
Gambar 2.8 Proses Konvolusi	22
Gambar 2.9 <i>Pooling Layer</i>	24
Gambar 2.10 <i>Fully Connected Layer</i>	25
Gambar 2.11 Arsitektur MobileNetV2.....	28
Gambar 2.12 Arsitektur EfficientNetB0.....	29
Gambar 3.1 Alur Penelitian.....	34
Gambar 3.2 Contoh Gambar <i>Dataset</i>	36
Gambar 3.3 Contoh Gambar <i>Cropping</i>	37
Gambar 3.4 Alur Pengembangan Model.....	39
Gambar 3.5 Nilai Piksel	44
Gambar 3.6 Filter Matriks	44
Gambar 3.7 Hasil Perhitungan <i>Convolutional Layer</i>	48
Gambar 3.8 Ilustrasi Perhitungan <i>Pooling Layer</i>	48
Gambar 4.1 Hasil <i>Cropping</i>	58
Gambar 4.2 Hasil <i>Resize</i>	58
Gambar 4.3 Visualisasi Hasil Evaluasi MobileNetV2	70
Gambar 4.4 Confusion Matrix MobileNetV2	71
Gambar 4.5 Visualisasi Hasil Evaluasi EfficientNetB0	74
Gambar 4.6 Confusion Matrix EfficientNetB0	75

DAFTAR RUMUS

Rumus 2.1 <i>Convolutional Layer</i>	23
Rumus 2.2 <i>Rectified Linear Unit</i>	23
Rumus 2.3 <i>Rectified Linear Unit</i>	23
Rumus 2.4 <i>Pooling Layer</i>	24
Rumus 2.5 <i>Fully Connected Layer</i>	25
Rumus 2.6 <i>Softmax</i>	26
Rumus 2.7 <i>Accuracy</i>	32
Rumus 2.8 <i>Precision</i>	32
Rumus 2.9 <i>Recall</i>	33
Rumus 2.10 <i>FI-Score</i>	33

DAFTAR KODE

Kode 4.1 Proses <i>Cropping</i>	57
Kode 4.2 Proses <i>Resize</i> dan Normalisasi.....	60
Kode 4.3 Proses Pembagian Data.....	61
Kode 4.4 Mendefinisikan MobileNetV2	63
Kode 4.5 Mendefinisikan EfficientNetB0.....	64
Kode 4.6 Pelatihan Model	66
Kode 4.7 Evaluasi Model	67

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Malaria adalah penyakit infeksi yang disebabkan oleh parasit *Plasmodium* dan ditularkan melalui gigitan nyamuk *Anopheles* betina [1]. Selain melalui gigitan nyamuk, penularan juga dapat terjadi melalui transfusi darah, transplantasi organ, serta penggunaan jarum suntik yang terkontaminasi [2]. Penyakit ini menjadi masalah kesehatan global, terutama di wilayah tropis dan subtropis, dengan angka kejadian yang tinggi di Afrika, Asia Tenggara, dan Amerika Selatan [3]. Gejala utama malaria meliputi demam, kelelahan, muntah, sakit kepala, serta anemia akibat penghancuran sel darah merah [4]. Pada kasus yang lebih parah, malaria dapat menyebabkan komplikasi serius seperti gangguan kesadaran, gagal organ, hingga kematian [5].

Diagnosis malaria dapat dilakukan dengan beberapa pendekatan, yaitu pemeriksaan klinis berdasarkan gejala pasien dan konfirmasi laboratorium menggunakan mikroskopi darah, *Rapid Diagnostic Test* (RDT), atau metode molekuler seperti *Polymerase Chain Reaction* (PCR) [6]. Meskipun mikroskopi darah merupakan metode standar untuk mendiagnosis malaria, metode ini memiliki keterbatasan seperti memerlukan tenaga medis yang ahli dan dapat memiliki keterbatasan dalam akurasi interpretasi hasil [7].

Plasmodium adalah genus parasit yang menyebabkan penyakit malaria. Terdapat lima spesies *Plasmodium* yang diketahui dapat menginfeksi manusia, yaitu *Plasmodium falciparum*, *Plasmodium vivax*, *Plasmodium malariae*, *Plasmodium ovale*, dan *Plasmodium knowlesi* [1]. Setiap spesies memiliki karakteristik yang berbeda dalam

hal distribusi geografis, tingkat keparahan infeksi, serta respons terhadap pengobatan. Berdasarkan spesies *Plasmodium* tersebut, *Plasmodium vivax* merupakan salah satu penyebab utama malaria di Asia dan Amerika Selatan, termasuk Indonesia [8]. *Plasmodium vivax* dapat dikategorikan menjadi empat stadium, yaitu *ring*, *trophozoite*, *schizont*, dan *gametocyte*. Parasit ini memiliki kemampuan unik untuk membentuk *hipnozoit*, yaitu bentuk dorman yang dapat bertahan di hati manusia dan menyebabkan kekambuhan infeksi di kemudian hari. Karakteristik ini menjadikan *Plasmodium vivax* lebih sulit dikendalikan dan membutuhkan metode diagnosis yang lebih sensitif serta pengobatan yang tepat [9].

Di Indonesia, malaria masih menjadi masalah kesehatan masyarakat yang signifikan, terutama di wilayah timur seperti Nusa Tenggara Timur, Maluku, dan Papua. Meskipun upaya pengendalian telah dilakukan seperti ACT, tantangan dalam deteksi dan diagnosis tetap ada. Terutama untuk *Plasmodium vivax* metode mikroskopi sering kali kurang sensitif dalam mendeteksi infeksi dengan tingkat parasitemia rendah [10]. Oleh sebab itu, diperlukan metode yang lebih akurat dan efisien guna mendukung upaya pemberantasan malaria di Indonesia.

Solusi dalam mendeteksi malaria yang disebabkan oleh *Plasmodium vivax* yaitu dapat dilakukan dengan memanfaatkan citra sel darah. Metode berbasis *deep learning* telah menunjukkan potensi besar dalam bidang medis, termasuk dalam diagnosis malaria. Salah satu metode yang digunakan adalah *Convolutional Neural Network* (CNN). CNN adalah salah satu jenis arsitektur jaringan saraf tiruan dalam bidang kecerdasan buatan yang dirancang untuk menangani data berbentuk gambar atau visual. CNN dapat mengekstrak fitur dari input yang berupa gambar lalu mengubah dimensi gambar tersebut menjadi lebih

kecil tanpa merubah karakteristik gambar tersebut [11]. CNN sering digunakan dalam berbagai aplikasi seperti pengenalan objek, klasifikasi citra, deteksi wajah, dan analisis gambar medis.

Berbagai penelitian telah dilakukan dengan menggunakan CNN pada klasifikasi citra. Penelitian yang dilakukan oleh Saksenata, Minarno, Azhar pada tahun 2022 mengenai klasifikasi citra sel darah mendapatkan hasil *accuracy*, *precision*, *recall*, dan *f1-score* sebesar 96% [12]. Penelitian yang dilakukan oleh Hermanto, Aziz, Sudianto pada tahun 2024 mengenai klasifikasi jenis buah kurma dengan membandingkan MobileNetV2 dan ResNet50. Pada hasil MobileNetV2 mendapatkan akurasi 95% dan ResNet50 mendapatkan akurasi 85% [13]. Penelitian yang dilakukan oleh Fiojati, Mintarsih, Arianti pada tahun 2023 mengenai klasifikasi citra jenis anjing dengan membandingkan EfficientNetB0 dan InceptionV3. Pada hasil pelatihan InceptionV3 mendapatkan akurasi 97,30% dan EfficientNetB0 mendapatkan akurasi 90,89% [14]. Berdasarkan penelitian tersebut MobileNetV2 dan EfficientNetB0 dapat memperoleh akurasi yang tinggi dengan arsitektur yang lebih ringan dengan parameter yang lebih sedikit dibandingkan dengan arsitektur CNN lain.

CNN memiliki beberapa arsitektur yang digunakan dalam berbagai pemrosesan gambar. MobileNetV2 merupakan arsitektur CNN yang dirancang untuk meningkatkan performa model pada berbagai tugas dan ukuran model. Arsitektur ini menggunakan unit dasar yang efisien secara memori yang memungkinkan inferensi yang sangat efisien dengan memisahkan kapasitas dan ekspresivitas jaringan melalui blok bottleneck terbalik dan bottleneck linear [15]. EfficientNetB0 adalah model dasar dari EfficientNet yang dirancang dengan pendekatan skala model yang efisien untuk CNN yang menggunakan metode *compound*

scaling untuk mengoptimalkan kedalaman, lebar, dan resolusi secara bersamaan sehingga menghasilkan model yang lebih kecil dan lebih cepat namun tetap memiliki akurasi tinggi [16].

Penelitian ini bertujuan untuk membandingkan kinerja arsitektur CNN yaitu MobileNetV2 dan EfficientNetB0 dalam mengklasifikasikan penyakit malaria yang disebabkan oleh *Plasmodium vivax* berdasarkan citra sel darah. Evaluasi performa model dilakukan menggunakan *confusion matrix* untuk menganalisis *accuracy*, *precision*, *recall*, dan *f1-score*. Melalui penelitian ini diharapkan dapat berkontribusi dalam pengembangan metode deteksi malaria sehingga dapat membantu upaya penanganan penyakit ini terutama di wilayah dengan keterbatasan tenaga medis dan fasilitas diagnosis.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah yang akan dibahas pada penelitian ini adalah membandingkan performa dari arsitektur MobileNetV2 dan EfficientNetB0 dalam melakukan klasifikasi penyakit malaria yang disebabkan oleh *Plasmodium vivax* berdasarkan citra sel darah.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dijelaskan, penelitian ini bertujuan untuk membandingkan hasil performa arsitektur MobileNetV2 dan EfficientNetB0 dalam melakukan klasifikasi penyakit malaria yang disebabkan oleh *Plasmodium vivax* berdasarkan citra sel darah. Evaluasi performa model ini dilakukan menggunakan

confusion matrix dengan metrik evaluasi yaitu, *accuracy*, *precision*, *recall*, dan *f1-score*.

1.4 Batasan Masalah

Batasan masalah yang akan dijelaskan pada penelitian ini adalah sebagai berikut.

1. Penelitian dilakukan pada deteksi malaria yang disebabkan oleh *Plasmodium vivax*.
2. *Dataset* yang digunakan berasal dari *Broad Bioimage Benchmark Collection* (BBBC), yaitu BBBC041 Version 1.
3. Citra yang digunakan menggunakan format PNG atau JPG.
4. Klasifikasi citra sel darah yang akan dilakukan berjumlah 5 kelas yaitu *uninfected*, *ring*, *trophozoite*, *schizont*, dan *gametocyte*.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian yang dilakukan adalah sebagai berikut:

1. Memberikan kontribusi dalam pengembangan model *deep learning* untuk citra sel medis.
2. Menjadi referensi bagi penelitian serupa di bidang analisis citra medis dan penyakit malaria.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan pada penelitian ini adalah sebagai berikut.

1.6.1 Bab I Pendahuluan

Pada Bab ini berisikan latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah penelitian, manfaat penelitian, dan sistematika penulisan yang digunakan.

1.6.2 Bab II Tinjauan Pustaka

Pada Bab ini berisikan tinjauan pustaka dari penelitian yang telah dilakukan sebelumnya dan dasar teori topik penelitian terkait untuk menjadi referensi penelitian.

1.6.3 Bab III Metodologi Penelitian

Pada Bab ini berisikan metode dan langkah-langkah yang digunakan pada penelitian ini.

1.6.4 Bab IV Hasil dan Pembahasan

Pada Bab ini berisikan tentang hasil dan pembahasan dari metode dan langkah-langkah yang digunakan pada penelitian ini.

1.6.5 Bab V Kesimpulan dan Saran

Pada Bab ini berisikan kesimpulan dari penelitian dan saran untuk pengembangan penelitian lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Tinjauan pustaka berisi berbagai penelitian terdahulu yang memiliki kesamaan topik dengan penelitian ini dan digunakan sebagai referensi utama. Penelitian ini merujuk pada 5 studi literatur yang membahas metode yang akan diterapkan. Berikut adalah beberapa sumber literatur yang dijadikan acuan.

Penelitian yang dilakukan oleh Sakseneta, Minarno, Azhar pada tahun 2022 yaitu penggunaan *Convolutional Neural Network* (CNN) untuk klasifikasi gambar sel darah dalam mendekripsi malaria. CNN yang diusulkan mencapai *accuracy*, *precision*, *recall*, dan *f1-score* sebesar 96% menggunakan dataset yang terdiri dari 27,558 gambar sel darah. Metodologi yang digunakan melibatkan berbagai lapisan CNN, seperti *convolutional layer*, *pooling*, dan *fully connected*, serta menggunakan metrik evaluasi *accuracy*, *precision*, *recall*, dan *f1-score* untuk mengevaluasi efektivitas model [12].

Penelitian yang dilakukan oleh Nurhaeni, Prasty, Hidayat *et al.* pada tahun 2024 yaitu mengembangkan sistem deteksi parasit malaria menggunakan *Convolutional Neural Network* (CNN) untuk meningkatkan akurasi dan kecepatan deteksi malaria pada citra sel darah. Menggunakan dataset yang terdiri dari 11.312 citra sel darah dari Kaggle, model ini berhasil mencapai akurasi 97,17% setelah 10 epoch, menunjukkan kemampuan yang baik dalam mengidentifikasi parasit. Penelitian ini juga menyoroti potensi penerapan model ini pada perangkat edge computing untuk diagnosis mandiri di daerah endemik malaria [17].

Penelitian yang dilakukan oleh Hermanto, Aziz, Sudianto pada tahun 2024 yaitu bertujuan untuk membandingkan performa model deep learning MobileNetV2 dan ResNet50 dalam klasifikasi tiga jenis buah kurma (Ajwa, Alqassim, dan Khenaizi). Kedua model dilatih menggunakan dataset berisi 300 gambar kurma yang diambil dengan *smartphone* dan diproses melalui augmentasi data. Hasil penelitian menunjukkan bahwa MobileNetV2 lebih unggul dengan akurasi tertinggi mencapai 95%, sedangkan ResNet50 memperoleh akurasi 85% [13].

Penelitian yang dilakukan oleh Fiojati, Mintarsih, Arianti pada tahun 2023 yaitu membandingkan performa dua algoritma *deep learning*, yaitu EfficientNetB0 dan InceptionV dalam melakukan klasifikasi citra jenis ras anjing. Penelitian ini dilakukan untuk menentukan model mana yang lebih akurat dalam mengklasifikasikan 15 kelas ras anjing menggunakan dataset berisi 2806 gambar yang dibagi menjadi data pelatihan dan validasi. Hasil penelitian menunjukkan bahwa InceptionV3 mencapai akurasi tertinggi sebesar 97,07% dengan loss 0,1077, sedangkan EfficientNetB0 memperoleh akurasi 88,09% dengan loss 0,0531 [14].

Penelitian yang dilakukan oleh Fauzi, Haqdu pada tahun 2025 yaitu membandingkan kinerja tiga model CNN, yaitu EfficientNetB0, MobileNetV2, dan ResNet50 dalam klasifikasi kepadatan lalu lintas. Penelitian ini dilakukan untuk mengidentifikasi model yang paling efektif dalam mengklasifikasikan kepadatan lalu lintas menjadi lima kelas (*Empty, Low, Medium, High, dan Traffic Jam*) dengan menggunakan dataset Singapore Traffic Density yang telah diperbesar dari 4.038 menjadi 6.850 gambar dan diubah ukurannya menjadi 224x224 piksel. Model dilatih selama 20 epoch dengan pembagian data

80% untuk pelatihan, 10% validasi, dan 10% pengujian. Hasil penelitian menunjukkan bahwa ResNet50 memberikan performa terbaik dengan akurasi validasi sekitar 85% serta nilai precision, recall, dan F1-score yang lebih tinggi dibandingkan EfficientNetB0 dan MobileNetV2 [18]. Tinjauan pustaka dapat dilihat pada Tabel 2.1 Tinjauan Pustaka.

Tabel 2.1 Tinjauan Pustaka

No	Peneliti	Judul	Permasalahan	Metode	Hasil
1.	Saksenata, Minarno, Azhar	Klasifikasi Citra Sel Darah Untuk Penyakit Malaria Dengan Metode CNN (2022)	Mengklasifikasi citra sel darah merah untuk penyakit malaria	CNN	Akurasi, presisi, recall, dan f1-score sebesar 96% dalam mengklasifik asikan citra sel darah merah penyakit malaria.
2.	Nurhaeni, Prasty, Hidayat, Anisa	Pemodelan Sistem Deteksi Parasit Malaria pada Citra Mikroskopis Sel Darah Menggunakan Metode Deep	Mendeteksi parasit malaria pada citra mikroskopis sel darah merah	CNN	Akurasi model sebesar 97,17% dalam mendeteksi malaria dengan citra mikroskopis

No	Peneliti	Judul	Permasalahan	Metode	Hasil
		Learning (2024)			sel darah merah.
3.	Hermanto, Aziz, Sudianto	Perbandingan Arsitektur MobileNetV2 dan RestNet50 untuk Klasifikasi Jenis Buah Kurma (2024)	Mengklasifikasi citra jenis buah kurma	CNN: MobileNet V2 dan ResNet50	MobileNetV2 mendapatkan akurasi 95% dan ResNet50 mendapatkan akurasi 85% dalam klasifikasi citra jenis buah kurma.
4.	Fiojati, Mintarsih, Arianti	Algoritma EfficientNetB0 dan InceptionV3 dalam Klasifikasi Citra Jenis Anjing (2023)	Mengklasifikasi citra jenis anjing	CNN: EfficientNet tB0 dan InceptionV3	EfficientNet B0 mendapatkan akurasi latih 90,89% dan InceptionV3 mendapatkan akurasi latih 97,30%.
5.	Fauzi, Haqdu	Comparison of CNN Models Using EfficientNetB0, MobileNetV2,	Mengklasifikasi citra tingkat kepadatan lalu lintas	CNN: EfficientNet tB0, MobileNet V2, dan ResNet50	EfficientNet B0 mendapatkan akurasi 81%, MobileNetV2 mendapatkan akurasi 2

No	Peneliti	Judul	Permasalahan	Metode	Hasil
		and ResNet50 for Traffic Density with Transfer Learning (2025)			mendapatkan akurasi 79%, dan ResNet50 mendapatkan akurasi 85% dalam klasifikasi tingkat kepadatan lalu lintas.

Penelitian ini memiliki perbedaan dibandingkan dengan penelitian sebelumnya yaitu dengan membandingkan arsitektur MobileNetV2 dan EfficientNetB0. Pada penelitian ini akan dilakukan klasifikasi citra sel darah yang tidak terinfeksi (*uninfected*) dan terinfeksi malaria akibat parasit *Plasmodium vivax* berdasarkan empat stadium perkembangan, yaitu *ring*, *trophozoite*, *schizont*, dan *gametocyte*. Evaluasi kinerja model dilakukan dengan mengukur *accuracy*, *precision*, *recall*, dan *f1-score* menggunakan *confusion matrix*.

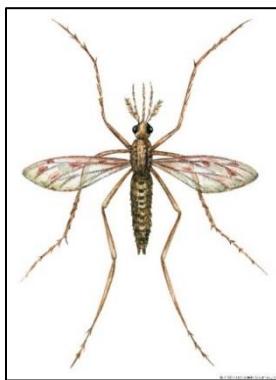
2.2 Dasar Teori

Dasar teori merupakan landasan intelektual yang menyajikan pemahaman umum terhadap konsep-konsep yang relevan dengan ruang lingkup penelitian, menyediakan kerangka konseptual yang diperlukan untuk memahami fenomena yang sedang dipelajari, dan menguraikan

prinsip-prinsip teoretis yang menjadi dasar bagi pengembangan pengetahuan dalam konteks penelitian yang dilakukan.

2.2.1 Malaria

Malaria merupakan penyakit yang disebabkan oleh parasit *Plasmodium* dan ditularkan melalui gigitan nyamuk *Anopheles* betina [1], nyamuk *Anopheles* dapat dilihat pada Gambar 2.1 berikut. Selain itu, penularan juga dapat terjadi melalui transfusi darah, transplantasi organ, serta penggunaan jarum suntik yang telah terkontaminasi [2]. Penyakit ini ditandai dengan gejala utama seperti demam, kelelahan, muntah, sakit kepala, serta anemia [4]. Dalam kondisi yang lebih parah, malaria dapat menimbulkan komplikasi serius, termasuk gangguan kesadaran, kegagalan organ, hingga kematian [5].



Gambar 2.1 Nyamuk Anopheles [19]

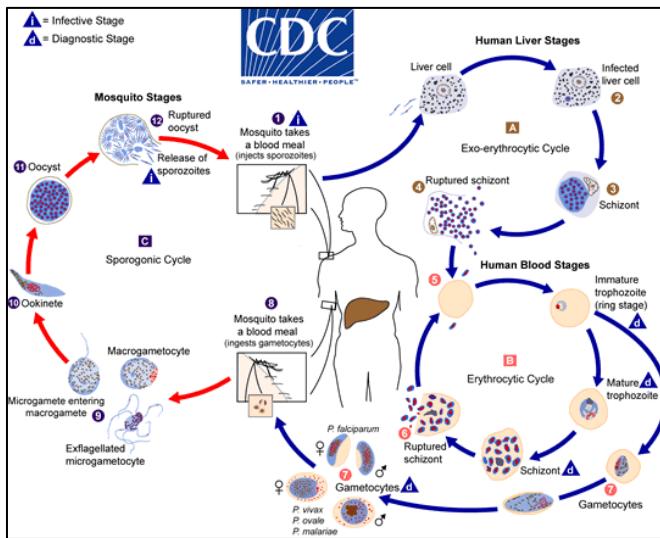
. Diagnosis malaria dapat dilakukan dengan beberapa pendekatan, yaitu pemeriksaan klinis berdasarkan gejala pasien dan konfirmasi laboratorium menggunakan mikroskopi darah, *Rapid Diagnostic Test*

(RDT), atau metode molekuler seperti *Polymerase Chain Reaction* (PCR) [6].

2.2.2 *Plasmodium Vivax*

Plasmodium vivax merupakan salah satu penyebab utama malaria di Asia dan Amerika Selatan, termasuk Indonesia [8]. Parasit ini memiliki siklus hidup kompleks yang mencakup fase seksual dalam tubuh nyamuk *Anopheles* dan fase aseksual dalam tubuh manusia. Salah satu karakteristik unik *Plasmodium vivax* adalah kemampuannya membentuk *hipnozoit* di hati yang dapat menyebabkan kekambuhan infeksi dalam jangka waktu berbulan-bulan hingga bertahun-tahun setelah infeksi primer [20]. Siklus hidupnya dimulai ketika *sporozoit* yang dibawa oleh nyamuk memasuki aliran darah manusia dan menginfeksi *hepatosit*.

Setelah berkembang menjadi *merozoit*, parasit ini dilepaskan ke dalam darah dan menginvasi *eritrosit* menyebabkan gejala malaria. Siklus hidup *Plasmodium* dapat dilihat pada Gambar 2.2 berikut.



Gambar 2.2 Siklus Hidup Plasmodium [21]

Selain itu, *Plasmodium vivax* memiliki karakteristik morfologi yang dapat menyerupai *Plasmodium falciparum* yang berpotensi menyebabkan kesalahan diagnosis klinis [22].

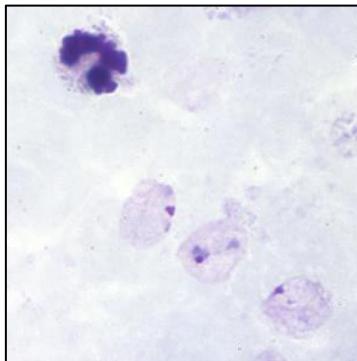
2.2.3 Stadium *Plasmodium Vivax*

Siklus hidup *Plasmodium vivax* dalam sel darah melibatkan beberapa stadium perkembangan yang penting untuk patogenesis malaria. Stadium-stadium ini mencakup *ring* (cincin), *trophozoite* (trofozoit), *schizont* (skizon), dan *gametocyte* (gametosit) yang dapat diamati melalui pemeriksaan mikroskopis [23]. Setiap stadium memiliki morfologi dan fungsi yang berbeda dalam siklus replikasi

parasit yang berkontribusi terhadap penyebaran infeksi dan timbulnya gejala malaria [24].

2.2.3.1 Stadium *Ring*

Stadium cincin merupakan tahap awal setelah merozoit memasuki eritrosit. Parasit berbentuk seperti cincin dengan vakuola besar di tengah dan inti kecil di pinggir. Pada tahap ini, aktivitas utama yang terjadi adalah adaptasi parasit di dalam eritrosit dan mulai mengkonsumsi hemoglobin sebagai sumber nutrisi. Parasit juga mulai mensintesis protein yang diperlukan untuk pertumbuhan lebih lanjut [25]. Gambar sel darah stadium cincin dapat dilihat pada Gambar 2.3 berikut.

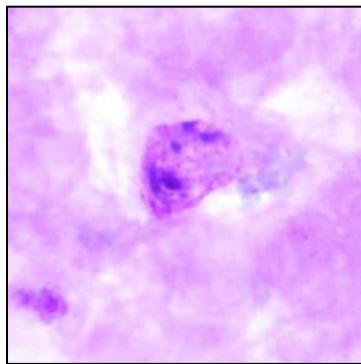


Gambar 2.3 Stadium *Ring* [21]

2.2.3.2 Stadium *Trophozoite*

Pada stadium trofozoit, parasit mulai aktif tumbuh dan memperbanyak organel selnya. Hemoglobin dari eritrosit mulai dicerna dan diubah menjadi pigmen hemozoin yang dapat terlihat di dalam sitoplasma parasit. Selain itu, metabolisme energi meningkat secara

signifikan untuk mendukung pertumbuhan. Parasit juga memperkuat sistem transportasi membran untuk meningkatkan asupan nutrisi dari sel inang. Aktivitas ini memungkinkan parasit untuk berkembang menjadi skizon dalam waktu 24 hingga 48 jam setelah memasuki eritrosit [26]. Gambar sel darah stadium trofozoit dapat dilihat pada Gambar 2.4 berikut.

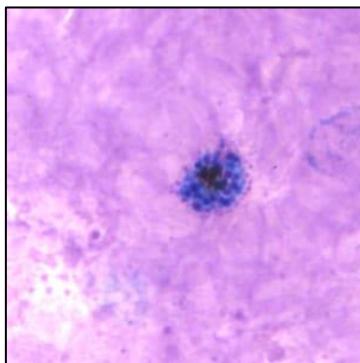


Gambar 2.4 Stadium *Trophozoite* [21]

2.2.3.3 Stadium *Schizont*

Stadium skizon ditandai dengan peningkatan aktivitas replikasi parasit secara aseksual. Parasit mengalami pembelahan mitosis berulang untuk membentuk merozoit baru. Dalam satu eritrosit, skizon dapat menghasilkan sekitar 12 hingga 24 merozoit yang siap dilepaskan ke dalam darah. Pada tahap ini, sintesis protein, RNA, dan DNA terjadi secara masif, mendukung produksi parasit baru yang akan menginfeksi eritrosit lainnya.

Ketika skizon matang eritrosit akan mengalami lisis, melepaskan merozoit yang dapat menyebabkan siklus infeksi baru dan memicu gejala klinis malaria seperti demam dan anemia [23]. Gambar sel darah stadium skizon dapat dilihat pada Gambar 2.5 berikut.



Gambar 2.5 Stadium *Schizont* [21]

2.2.3.4 Stadium *Gametocyte*

Pada stadium ini sebagian parasit berdiferensiasi menjadi gametosit jantan dan betina, yang merupakan bentuk seksual *Plasmodium vivax*. Gametosit memiliki aktivitas metabolismik yang lebih rendah dibandingkan stadium aseksual karena fungsinya lebih berfokus pada persiapan untuk transmisi ke vektor nyamuk. Gametosit matang akan bersirkulasi di dalam darah hingga dihisap oleh nyamuk *Anopheles*.

Setelah berada dalam tubuh nyamuk, gametosit akan mengalami fertilisasi dan berkembang menjadi ookinet sebelum akhirnya membentuk sporozoit yang siap ditularkan ke inang baru [27]. Gambar sel darah stadium gametosit dapat dilihat pada Gambar 2.6 berikut.



Gambar 2.6 Stadium *Gametocyte* [21]

2.2.4 Citra

Citra merupakan representasi atau gambaran dari suatu objek yang dapat dikategorikan menjadi dua jenis, yaitu citra analog dan citra digital. Citra analog tidak dapat diproses langsung oleh komputer, sehingga perlu dikonversi terlebih dahulu menjadi citra digital. Contohnya, foto dapat diubah menjadi bentuk digital melalui pemindaian dengan *scanner*, persebaran panas tubuh dapat direkam menggunakan kamera inframerah dan dikonversi menjadi data numerik, sedangkan informasi mengenai densitas serta komposisi bagian dalam tubuh manusia dapat diperoleh melalui sinar-X dan sistem deteksi radiasi yang menghasilkan data digital [28]. Sementara itu, citra digital adalah jenis citra yang dapat langsung diproses oleh komputer dan disimpan dalam bentuk matriks yang berisi nilai warna *Red*, *Green*,

Blue (RGB). Secara matematis, citra digital dapat dinyatakan sebagai fungsi dua variabel x dan y , di mana x dan y merupakan koordinat spasial, sedangkan nilai $f(x,y)$ merepresentasikan intensitas citra pada titik koordinat tersebut [28].

2.2.5 Sel Darah

Sel darah merupakan komponen utama dalam sistem peredaran darah yang memiliki peran penting dalam transportasi oksigen, pertahanan imun, dan pembekuan darah. Darah terdiri dari plasma dan elemen seluler, yaitu sel darah merah (*erythrocytes*), sel darah putih (*leukocytes*), dan trombosit (*platelets*). Plasma, yang membentuk sekitar 55% dari volume darah, adalah cairan yang mengandung air, protein, hormon, dan nutrisi penting bagi tubuh. Sementara itu, sekitar 45% dari darah terdiri dari sel-sel darah yang menjalankan fungsi-fungsi spesifiknya [29].

Jenis-jenis sel darah dapat dibedakan menjadi tiga kelompok utama. Sel darah merah (*erythrocytes*) adalah komponen yang paling banyak dalam darah dan berperan utama dalam transportasi oksigen dari paru-paru ke jaringan serta membawa karbon dioksida kembali ke paru-paru untuk dikeluarkan. Fungsi ini dimungkinkan berkat hemoglobin, protein yang mampu mengikat dan melepaskan oksigen secara efisien. Sel darah merah memiliki bentuk bikonkaf yang fleksibel, memungkinkan mereka untuk melewati kapiler kecil dalam sistem peredaran darah [30].

Sel darah putih (*leukocytes*) bertanggung jawab atas pertahanan tubuh terhadap infeksi dan patogen. Leukosit terdiri dari beberapa jenis, yaitu neutrofil, eosinofil, basofil, limfosit, dan monosit. Neutrofil adalah jenis leukosit yang paling banyak dan berperan dalam melawan

infeksi bakteri melalui proses fagositosis. Eosinofil dan basofil berperan dalam respons alergi dan melawan parasit, sementara limfosit (termasuk sel B dan T) memiliki peran utama dalam imunitas spesifik terhadap patogen tertentu. Monosit, yang merupakan leukosit terbesar, berkembang menjadi makrofag setelah memasuki jaringan tubuh dan berperan dalam fagositosis serta regulasi respons imun [31].

Jenis sel darah lainnya adalah trombosit (platelets), yaitu fragmen sel yang berperan penting dalam proses pembekuan darah. Ketika terjadi luka atau cedera pembuluh darah, trombosit akan berkumpul di area tersebut dan membentuk sumbatan sementara untuk menghentikan perdarahan. Selain itu, trombosit juga melepaskan faktor-faktor pembekuan yang membantu dalam proses koagulasi darah untuk mencegah kehilangan darah yang berlebihan [32].

2.2.6 Deep Learning

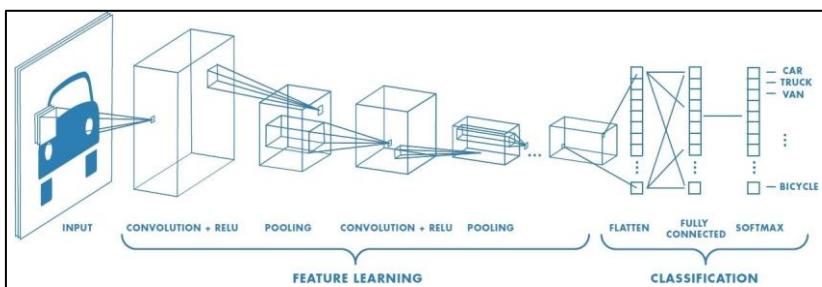
Deep learning merupakan cabang ilmu pembelajaran mesin dalam Jaringan Saraf Tiruan (JST) [33]. *Deep learning* mampu mengenali pola-pola kompleks serta merepresentasikan data dalam bentuk yang lebih abstrak karena mempunyai struktur jaringan yang lebih dalam. Metode ini telah mengubah cara analisis informasi dan pengolahan data, memberikan dampak signifikan di berbagai bidang, seperti pengenalan gambar dan pemrosesan bahasa alami [34].

Deep learning menggunakan algoritma yang dapat membentuk abstraksi tingkat tinggi dari data melalui serangkaian fungsi transformasi non-linear yang tersusun dalam beberapa lapisan. Salah satu contoh algoritma *deep learning* adalah *Convolutional Neural Network* (CNN). *Deep learning* mempunyai proses tahapan pengenalan

fitur secara otomatis yang terintegrasi di dalam model CNN yang berfungsi sebagai *classifier* [35].

2.2.7 Convolutional Neural Network (CNN)

Salah satu jenis Jaringan Saraf Tiruan (ANN) yang paling sering digunakan adalah *Convolutional Neural Network* (CNN) yang dapat digunakan untuk analisis dan klasifikasi gambar [36]. CNN dikembangkan untuk menangani data dalam bentuk dua dimensi dan sering diterapkan dalam berbagai tugas pengenalan gambar, seperti klasifikasi, deteksi objek, segmentasi, dan lain-lain [37]. CNN banyak digunakan dalam berbagai bidang, termasuk segmentasi gambar medis, deteksi objek, serta pemrosesan bahasa alami, seperti analisis teks [38]. Arsitektur CNN memiliki kemampuan untuk secara otomatis dan hierarkis mempelajari fitur-fitur dari data gambar . Pemrosesan citra dengan metode ini dilakukan secara bertahap dan berulang melalui serangkaian lapisan konvolusi (*convolutional layer*), lapisan penggabungan (*pooling layer*), dan lapisan yang terhubung sepenuhnya (*fully connected layer*) [39]. Arsitektur *Convolutional Neural Network* dapat dilihat pada Gambar 2.7 berikut.

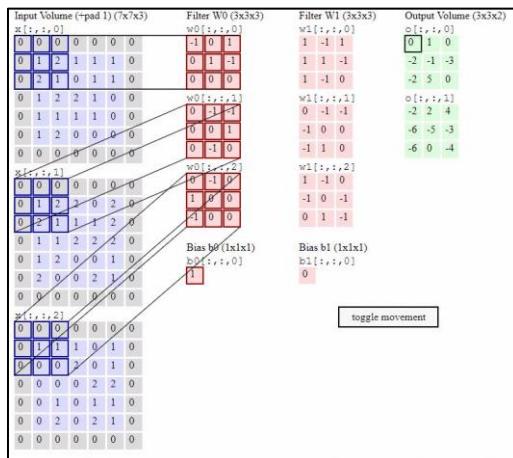


Gambar 2.7 Arsitektur *Convolutional Neural Network* [40]

2.2.7.1 Convolutional Layer

Lapisan konvolusi dalam *Convolutional Neural Network* (CNN) berfungsi untuk mengekstraksi fitur dari citra dengan menggunakan filter atau kernel melalui operasi konvolusi. Filter ini umumnya berbentuk matriks dua dimensi dengan ukuran tertentu, seperti 5×5 , 3×3 , atau 1×1 yang digeser dari bagian kiri atas ke kanan bawah citra. Proses ini menghasilkan *feature map*, yang kemudian diproses lebih lanjut oleh lapisan aktivasi. Fungsi aktivasi seperti *Rectified Linear Unit* (ReLU) diterapkan untuk mengubah nilai *output* neuron menjadi nol jika input bernilai negatif, sedangkan jika bernilai positif, input tetap dipertahankan.

Selain itu, *stride* menentukan jumlah langkah perpindahan filter selama proses konvolusi, sedangkan *padding* digunakan untuk mempertahankan dimensi keluaran agar tetap sesuai dengan dimensi input awal. Proses konvolusi pada *convolutional layer* dapat dilihat pada Gambar 2.8 berikut.



Gambar 2.8 Proses Konvolusi

Perhitungan pada *convolutional layer* yaitu menggunakan Rumus 2.1 sebagai berikut [41].

$$n_{(w,h)} = \left\lceil \frac{n_{in} + 2p - k}{s} \right\rceil + 1 \quad (2.1)$$

Keterangan:

- $n_{(w,h)}$ = Hasil input ukuran citra
- n_{in} = Nilai ukuran citra input
- p = Ukuran padding
- k = Ukuran kernel yang digunakan
- s = Ukuran stride

2.2.7.2 Rectified Linear Unit

Rectified Linear Unit (ReLU) adalah fungsi aktivasi nonlinear yang digunakan dalam multilayer *neural network* atau *deep neural network*. *Rectified Linear Unit* (ReLU) biasanya digunakan sebagai fungsi aktivasi di CNN untuk menghindari hilangnya gradien dan meningkatkan kecepatan pelatihan. Fungsi ReLU didefinisikan sebagai:

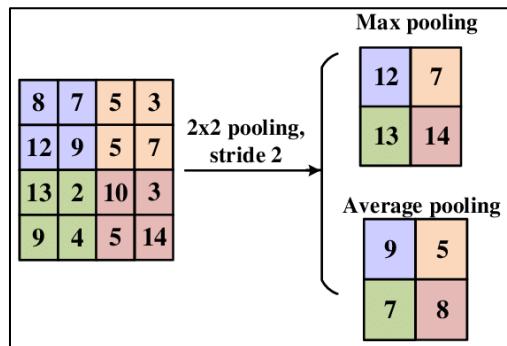
$$f(x) = \max (0, x) \quad (2.2)$$

Menurut Rumus 2.2, *output* ReLU adalah nilai maksimum antara nol dan nilai input. *Output* sama dengan nol ketika nilai input negatif dan nilai input ketika input positif. Oleh karena itu, Rumus 2.2 dapat diubah menjadi Rumus 2.3 sebagai berikut [42].

$$f(x) = \begin{cases} 0, & \text{jika } x < 0 \\ x, & \text{jika } x \geq 0 \end{cases} \quad (2.3)$$

2.2.7.3 Pooling Layer

Lapisan pooling berfungsi untuk mengurangi dimensi piksel dari *feature map* yang dihasilkan oleh lapisan konvolusi sebelumnya. Tujuan dari proses ini adalah untuk menyederhanakan representasi fitur, mengurangi kompleksitas dimensi, serta meningkatkan efisiensi komputasi dengan mengurangi jumlah parameter yang harus diperbarui dan mengatasi *overfitting*. Pada proses pooling, ukuran downsampling yang digunakan harus ditentukan terlebih dahulu. Terdapat beberapa jenis pooling yang digunakan, seperti *Max Pooling* dan *Average Pooling*. Hasil dari *pooling layer* akan menghasilkan *features* dengan ukuran $w \times h \times d$, dari ukuran ini dapat dihitung menggunakan rumus perhitungan. *Pooling layer* dapat dilihat pada Gambar 2.9 Berikut.



Gambar 2.9 *Pooling Layer*

Perhitungan yang dilakukan pada *pooling layer* dapat menggunakan Rumus 2.4 sebagai berikut [41].

$$n_{(w,h)} = \frac{(n_{(w,h)-1} - f)}{s} + 1 \quad (2.4)$$

Keterangan:

$n_{(w,h)}$ = Hasil ukuran width dan height

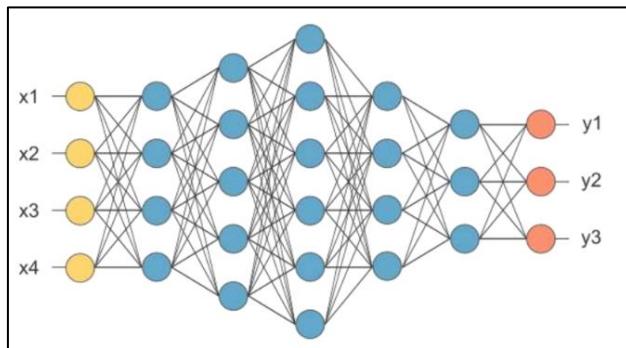
$n_{(w,h)-1}$ = Ukuran *width* dan *height* sebelumnya

f = Ukuran *kernel*

s = Ukuran *stride*

2.2.7.4 Fully Connected Layer

Fully connected layer merupakan salah satu komponen dalam jaringan saraf tiruan yang terdiri dari *input layer*, *hidden layer*, dan *output layer*. Pada lapisan ini, setiap neuron terhubung secara penuh dengan seluruh neuron pada lapisan sebelumnya maupun lapisan berikutnya. Koneksi penuh antar neuron ini memungkinkan pertukaran informasi yang lebih luas di dalam jaringan, sehingga meningkatkan kemampuan model dalam mempelajari pola dari data yang diberikan. *Fully connected layer* dapat dilihat pada Gambar 2.10 berikut.



Gambar 2.10 *Fully Connected Layer*

Fully connected layer mengklasifikasikan citra gambar sesuai dengan citra fitur yang telah terekstraksi, seperti pada Rumus 2.5 berikut [43].

$$y = wx + b \quad (2.5)$$

Keterangan:

y = output fully connected layer

w = matriks bobot (*weights*)

x = vektor input

b = vektor bias

2.2.7.5 Softmax

Tujuan dari penggunaan aktivasi *softmax* ini adalah menghitung probabilitas dari setiap kelas target dari semua kelas target yang ada, Kelebihan menggunakan aktivasi ini adalah rentang probabilitas keluarannya dengan nilai 0 hingga 1. Untuk mendapatkan nilai luaran probabilitas satu objek dapat menggunakan Rumus 2.6 berikut [44].

$$f_i(x) = \frac{e^{x_i}}{\sum_{j=1}^n \sum e^{x_j}} \quad (2.6)$$

Keterangan:

$f_i(x)$ = Probabilitas objek ke- i dari vektor output

x = Vektor nilai dari *fully connected layer* terakhir

e^{x_i} = Eksponensial dari nilai input x_i

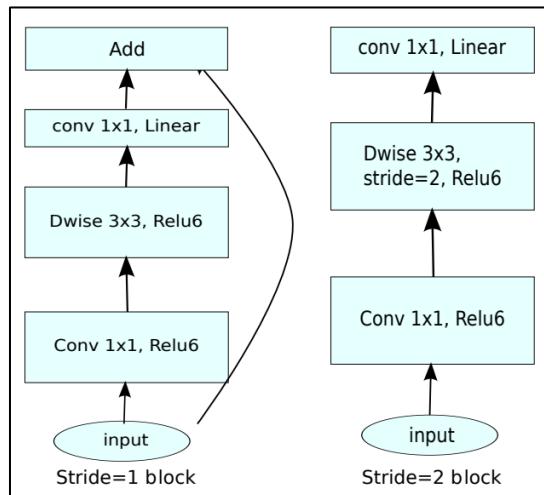
2.2.8 MobileNetV2

MobileNetV2 merupakan arsitektur jaringan saraf yang dirancang khusus untuk aplikasi mobile dengan tujuan meningkatkan performa dan efisiensi model pada berbagai tugas seperti klasifikasi gambar, deteksi objek, dan segmentasi semantik. Arsitektur ini merupakan pengembangan dari MobileNetV1 yang mempertahankan kesederhanaan desain namun berhasil meningkatkan akurasi dan efisiensi komputasi. MobileNetV2 memperkenalkan konsep blok *inverted residuals* dan *bottleneck linear* yang memungkinkan

pemisahan kapasitas jaringan secara efisien sehingga menghasilkan inferensi yang hemat memori dan cepat pada perangkat dengan sumber daya terbatas. Pendekatan ini juga mengurangi kebutuhan akses memori utama dengan tidak mematerialisasi tensor intermediate yang besar sehingga sangat cocok untuk perangkat dengan cache memori terbatas.

Arsitektur MobileNetV2 bekerja dengan memproses input citra melalui serangkaian lapisan yang menggunakan modul utama berupa *inverted residual* dengan *linear bottleneck*. Pada awalnya, input citra yang berukuran rendah-dimensi diperluas ke dimensi yang lebih tinggi menggunakan konvolusi 1x1, kemudian fitur tersebut difilter menggunakan *depthwise convolution* yang ringan secara komputasi. Setelah itu, fitur diproyeksikan kembali ke representasi berdimensi rendah melalui konvolusi linear tanpa aktivasi non-linear, yang bertujuan menghindari hilangnya informasi penting akibat fungsi aktivasi. Struktur *inverted residual* ini memungkinkan jaringan mempertahankan informasi penting sambil mengurangi kebutuhan memori dan operasi komputasi, sehingga sangat efisien untuk perangkat mobile.

Proses ini diulang dalam beberapa blok hingga menghasilkan fitur akhir yang dapat digunakan untuk tugas seperti klasifikasi, deteksi, atau segmentasi [15]. Arsitektur MobileNetV2 dapat dilihat pada Gambar 2.11 berikut.



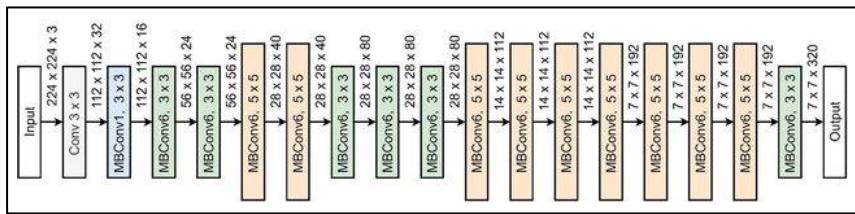
Gambar 2.11 Arsitektur MobileNetV2 [15]

2.2.9 EfficientNetB0

EfficientNet merupakan *Convolutional Neural Networks* yang dirancang dengan pendekatan skala model yang efisien, yaitu dengan mengoptimalkan kedalaman, lebar, dan resolusi input secara bersamaan menggunakan metode *compound scaling*. Pendekatan ini berbeda dari metode skala tradisional yang hanya mengubah satu dimensi saja, sehingga menghasilkan model yang lebih seimbang dan efisien dalam penggunaan sumber daya. EfficientNet menggunakan *neural architecture search* untuk menemukan arsitektur dasar yang optimal sebelum melakukan *scaling*, sehingga menghasilkan model yang tidak

hanya lebih kecil dan lebih cepat, tetapi juga memiliki akurasi yang lebih tinggi dibandingkan model ConvNet sebelumnya.

EfficientNetB0 menggunakan arsitektur berbasis blok *mobile inverted bottleneck convolution* (MBConv) yang dilengkapi dengan optimasi *squeeze and excitation* untuk meningkatkan efisiensi dan akurasi. Proses dimulai dari input citra berukuran 224x224 piksel yang melewati lapisan konvolusi awal Conv3x3 dengan 32 filter, kemudian dilanjutkan ke beberapa blok MBConv dengan variasi ukuran kernel (3x3 dan 5x5) dan jumlah filter yang bertingkat, mulai dari 16 hingga 320 filter, dengan pengurangan resolusi spasial secara bertahap melalui stride 2 pada beberapa blok. Setelah rangkaian MBConv, fitur diekstraksi melalui lapisan Conv1x1, pooling, dan fully connected layer untuk menghasilkan output klasifikasi [16]. Arsitektur EfficientNetB0 dapat dilihat pada Gambar 2.12 berikut.



Gambar 2.12 Arsitektur EfficientNetB0 [16]

2.2.10 K-Fold Cross Validation

K-fold cross validation merupakan metode yang digunakan untuk mengukur performa model atau algoritma secara lebih menyeluruh. Pada proses ini, dataset dibagi menjadi dua bagian, yaitu data latih dan data validasi. Setelah itu, model dilatih menggunakan data latih dan dievaluasi menggunakan data validasi secara berulang sebanyak k kali.

Metode ini juga dapat mengurangi ketergantungan hasil evaluasi pada pembagian data tertentu serta meningkatkan kemampuan model dalam melakukan generalisasi terhadap data yang belum pernah dilihat sebelumnya [45]. Ilustrasi dari *k-fold cross validation* ditampilkan pada Tabel 2.2 berikut.

Tabel 2.2 *K-Fold Cross Validation*

K-Fold	Cross Validation				
	Validation	Train	Train	Train	Train
1	Train	Validation	Train	Train	Train
2	Train	Train	Validation	Train	Train
3	Train	Train	Train	Validation	Train
4	Train	Train	Train	Train	Validation
5	Train	Train	Train	Train	Validation

2.2.11 Confusion Matrix

Confusion matrix adalah matriks yang digunakan untuk menggambarkan hasil klasifikasi pada suatu dataset. Matriks ini disusun dalam bentuk tabel dan berfungsi sebagai alat evaluasi kinerja model klasifikasi [46]. Tabel *confusion matrix* terdiri dari baris dan kolom yang jumlahnya sesuai dengan jumlah kelas yang diklasifikasikan. Terdapat empat komponen yaitu *false positives* (FP) yang menunjukkan jumlah data negatif yang salah diklasifikasikan sebagai positif, *false negatives* (FN) yang menggambarkan jumlah data positif yang salah diklasifikasikan sebagai negatif, *true positives* (TP) yang merupakan jumlah data positif yang diklasifikasikan dengan benar, dan *true negatives* (TN) yang menunjukkan jumlah data negatif yang diklasifikasikan dengan benar [47]. Berdasarkan hasil *confusion matrix*

akan dilakukan perhitungan mengeluarkan *output* berbagai metrik evaluasi model seperti *accuracy*, *precision*, *recall*, dan *f1-score*.

Tabel 2.3 *Confusion Matrix* Klasifikasi Biner

		Predicted Positive	Predicted Negative
Actual Positive		True Positive	False Negative
Actual Negative		False Positive	True Negative

Tabel 2.3 di atas merupakan *confusion matrix* untuk klasifikasi biner. Namun, karena penelitian ini dilakukan klasifikasi untuk multiclass, maka confusion matrix yang digunakan yaitu seperti Tabel 2.4 di bawah.

Tabel 2.4 *Confusion Matrix* Klasifikasi Multiclass

	Predicted					
	Kelas	A	B	C	D	E
A	TN	TN	FP	TN	TN	
B	TN	TN	FP	TN	TN	
C	FN	FN	TP	FN	FN	
D	TN	TN	FP	TN	TN	
E	TN	TN	FP	TN	TN	

Pada Tabel 2.3 di atas adalah permasalahan untuk salah satu kelas pada *confusion matrix*, yaitu Kelas C.

Penjelasan untuk Tabel 2.2 dan Tabel 2.3 adalah sebagai berikut.

- True Positive (TP)* : Ketika label aktual bernilai positif, dan model juga memprediksi positif.
- True Negative (TN)* : Ketika label aktual bernilai negatif, dan model juga memprediksi negatif

- c. *False Positive* (FP) : Ketika aktual label bernilai negatif, tetapi model memprediksi positif.
- d. *False Negative* (FN) : Ketika aktual label bernilai positif, tetapi model memprediksi negatif.

Berdasarkan hasil dari *confusion matrix* akan dilakukan perhitungan untuk metrik evaluasi. Metrik tersebut berupa *accuracy*, *precision*, *recall*, dan *f1-score*. Berikut adalah rumus yang digunakan pada *confusion matrix* [48].

1. Accuracy

Accuracy merupakan metrik yang digunakan untuk mengukur sejauh mana suatu model mampu mengklasifikasikan data dengan benar. Rumus untuk menghitung *accuracy* dapat dilihat pada Rumus 2.7 berikut.

$$\text{Accuracy} = \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N \sum_{j=1}^N C_{i,j}} \quad (2.7)$$

Keterangan:

N = Jumlah kelas dalam klasifikasi

TP_i = Jumlah data dari kelas i yang diprediksi benar sebagai kelas tersebut

$C_{i,j}$ = Jumlah data aktual kelas i yang diprediksi sebagai kelas j

2. Precision

Precision menggambarkan tingkat ketepatan dalam mengklasifikasikan data yang digunakan dengan hasil prediksi yang dihasilkan oleh model. Rumus untuk menghitung *precision* dapat dilihat pada Rumus 2.8 berikut.

$$\text{Precision} = \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N (TP_i + FP_i)} \quad (2.8)$$

Keterangan:

N = Jumlah kelas dalam klasifikasi

TP_i = Jumlah data dari kelas ke- i yang diprediksi benar sebagai kelas tersebut

FP_i = Jumlah data dari kelas ke- i yang diprediksi salah sebagai kelas tersebut

3. Recall

Recall mengukur kemampuan model dalam menemukan kembali seluruh data positif yang sebenarnya ada dalam dataset. Rumus untuk menghitung recall dapat dilihat pada Rumus 2.9 berikut.

$$Recall = \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N (TP_i + FN_i)} \quad (2.9)$$

Keterangan:

N = Jumlah kelas dalam klasifikasi

TP_i = Jumlah data dari kelas ke- i yang diprediksi benar sebagai kelas tersebut

FN_i = Jumlah data dari kelas ke- i yang diprediksi sebagai kelas lain

4. F1-score

F1-score digunakan untuk menilai keseimbangan antara precision dan recall dalam satu nilai dengan menggunakan rata-rata dari keduanya. Rumus untuk menghitung *f1-score* dapat dilihat pada Rumus 2.10 berikut.

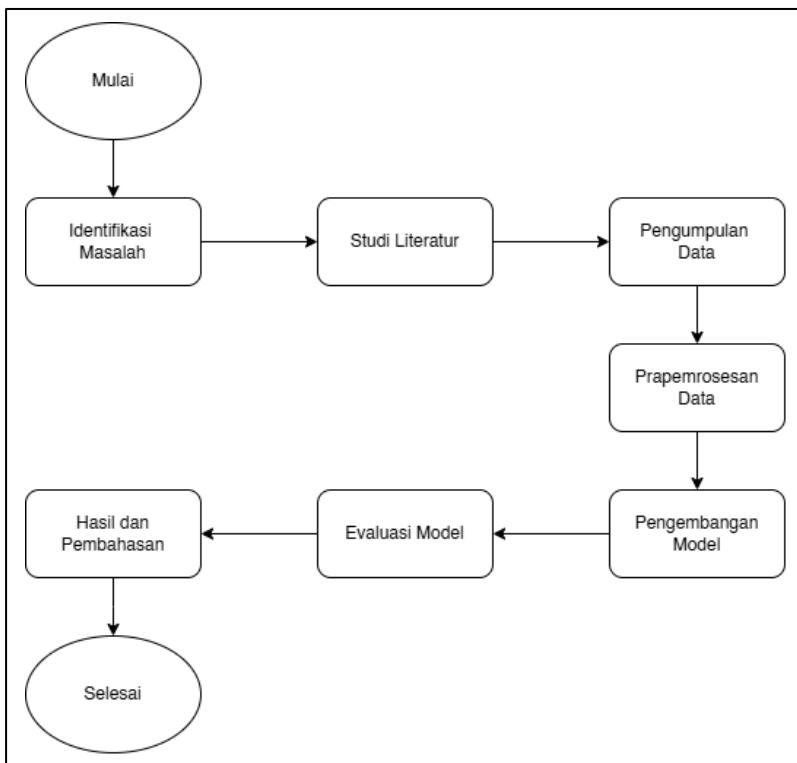
$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (2.10)$$

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Pada penelitian ini alur yang digunakan dalam tugas akhir klasifikasi penyakit malaria yang disebabkan oleh parasit *Plasmodium vivax* berdasarkan citra sel darah menggunakan *Convolutional Neural Network* (CNN) dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

3.2 Penjabaran Langkah Penelitian

Pada alur penelitian yang dapat dilihat pada Gambar 3.1 diatas, berikut merupakan penjabaran mengenai langkah penelitian yang akan dilakukan.

3.2.1 Identifikasi Masalah

Pada tahap identifikasi masalah ditemukan bahwa diagnosis malaria dapat dilakukan dengan beberapa pendekatan, yaitu pemeriksaan klinis berdasarkan gejala pasien dan konfirmasi laboratorium menggunakan mikroskopi darah, *Rapid Diagnostic Test* (RDT), atau *Polymerase Chain Reaction* (PCR). Meskipun mikroskopi darah merupakan metode standar untuk mendiagnosis malaria, metode ini memiliki keterbatasan seperti memerlukan tenaga medis yang ahli, dapat memiliki keterbatasan dalam akurasi interpretasi hasil, dan menghabiskan banyak waktu. Oleh karena itu, penelitian ini akan berfokus untuk mengklasifikasi penyakit malaria dengan menggunakan metode CNN yang berpotensi dapat menyelesaikan masalah untuk melakukan diagnosis penyakit malaria.

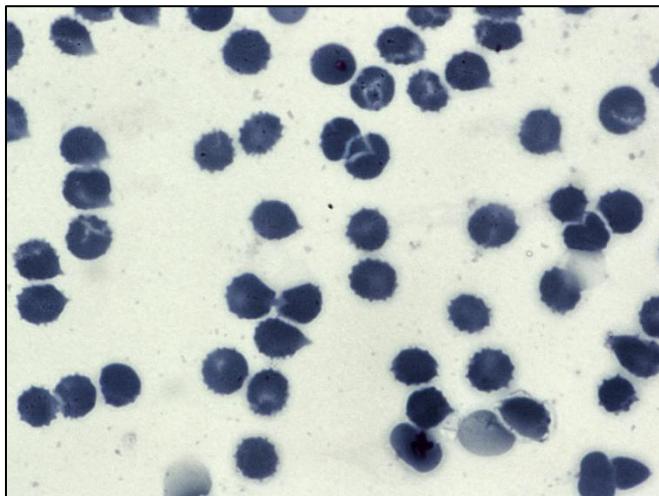
3.2.2 Studi Literatur

Pada tahap ini dilakukan pengumpulan berbagai macam informasi, referensi, konsep dasar, serta ilmu yang menjadi landasan dasar dilakukannya penelitian ini. Proses ini dilakukan dengan mencari referensi berupa buku, jurnal, penelitian yang pernah dilakukan, publikasi, seminar, maupun referensi lainnya mengenai metode CNN yaitu arsitektur MobileNetV2 dan EfficientNetB0 yang digunakan untuk melakukan klasifikasi

penyakit malaria yang disebabkan *Plasmodium vivax* berdasarkan citra sel darah.

3.2.3 Pengumpulan Data

Data yang digunakan pada penelitian ini merupakan dataset citra sel darah merah penyakit malaria yang disebabkan oleh *Plasmodium vivax*. Data tersebut didapatkan dari web *Broad Bioimage Benchmark Collection* (BBBC¹) dengan *accession number* BBBC041 Version 1. Data yang didapatkan terdiri dari citra sel darah yang dibagi menjadi tidak terinfeksi (*uninfected*), *ring*, *trophozoite*, *schizont*, *gametocyte*. Contoh gambar pada dataset dapat dilihat pada Gambar 3.2 berikut.



Gambar 3.2 Contoh Gambar Dataset

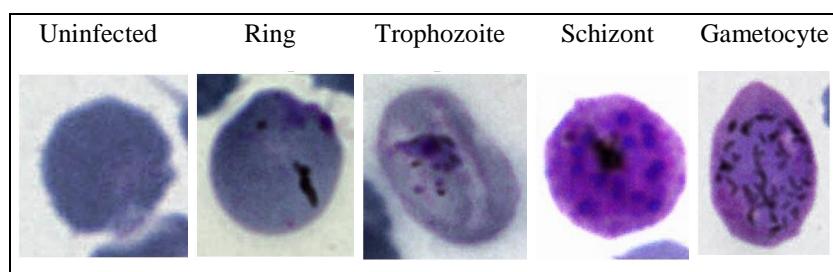
¹ <https://bbbc.broadinstitute.org/BBBC041>

3.2.4 Prapemrosesan Data

Pada penelitian ini tahap prapemrosesan data dilakukan untuk mempersiapkan citra agar sesuai dengan kebutuhan model. Proses prapemrosesan meliputi *cropping*, yaitu memotong bagian citra, *resize* yang berfungsi untuk menyamakan ukuran citra agar sesuai dengan input yang dibutuhkan, dan normalisasi dilakukan untuk memudahkan model belajar dengan memastikan input yang diterima memiliki distribusi konsisten dan stabil.

3.2.4.1 *Cropping*

Tahap awal yang dilakukan pada prapemrosesan data yaitu melakukan *cropping* pada *dataset*. Cropping yang dilakukan yaitu memotong gambar pada *dataset* berdasarkan kelas yang sudah dianotasi oleh Stefanie Lopes, yaitu peneliti malaria di rumah sakit Dr. Heitor Vieira Dourado Tropical Medicine Foundation yang menunjukkan tahap perkembangan sel darah. *Cropping* akan menghasilkan 5 kelas yaitu *uninfected*, *ring*, *trophozoite*, *schizont*, dan *gametocyte*. Contoh gambar setelah *cropping* dapat dilihat pada Gambar 3.3 berikut.



Gambar 3.3 Contoh Gambar *Cropping*

3.2.4.2 *Resize*

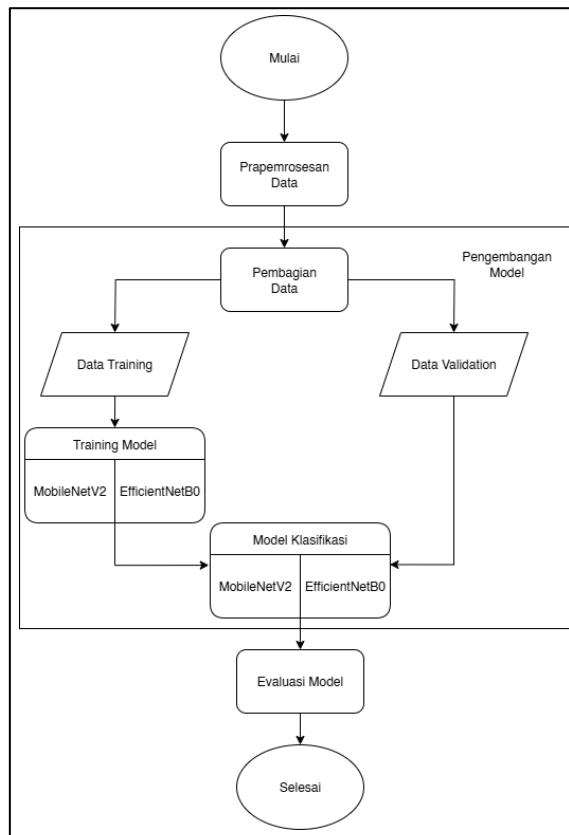
Setelah dilakukan *cropping*, selanjutnya akan dilakukan *resize*. *Resize* dilakukan karena gambar hasil *cropping* akan menghasilkan ukuran piksel yang berbeda-beda. Gambar setelah *cropping* akan dilakukan *resize* menjadi piksel dengan ukuran 224×224 seperti penelitian yang dilakukan oleh Nurhaeni, Prastyia, Hidayat *et al.* [17]

3.2.4.3 Normalisasi

Proses normalisasi dilakukan dengan menggunakan nilai mean dan standar deviasi (std) yang diperoleh dari dataset ImageNet, yaitu mean sebesar [0,485, 0,456, 0,406] dan standar deviasi sebesar [0,229, 0,224, 0,225]. Nilai-nilai ini diperoleh dari perhitungan statistik piksel per channel (RGB) pada dataset ImageNet, setelah gambar diskalakan ke rentang. Tujuan penggunaan mean dan standar deviasi ini adalah untuk melakukan penskalaan channel-wise, sehingga data input memiliki distribusi dengan mean nol dan standar deviasi satu.

3.2.5 Pengembangan Model

Alur pengembangan model penelitian ini dapat dilihat pada Gambar 3.4 berikut.



Gambar 3.4 Alur Pengembangan Model

3.2.5.1 Pembagian Data

Pada penelitian ini akan digunakan metode *k-fold cross validation* dengan k bernilai 5 untuk membagi data menjadi lima bagian yang seimbang. Bagian data yang digunakan untuk validasi akan bergantian pada setiap fold, sehingga seluruh data memperoleh kesempatan untuk

menjadi data validasi setidaknya satu kali. Sehingga rasio data train:validasi tetap pada setiap fold, namun pembagiannya dilakukan secara bergantian sebanyak lima kali.

3.2.5.2 Pelatihan Model

Pada tahap ini, arsitektur MobileNetV2 dan EfficientNetB0 akan dilakukan pelatihan dengan menggunakan data latih yang sudah disiapkan sebelumnya. Penelitian terdahulu yang memiliki topik relevan dengan penelitian ini, meskipun menggunakan dataset yang berbeda, telah berhasil menentukan *hyperparameter* yang optimal [49][50]. *Hyperparameter* yang akan digunakan pada penelitian ini yaitu *optimizer* Adam dan RMSProp, *learning rate* 1e-4 dan 1e-3, *epoch* 50 dan 100, dan *batch size* 32.

Hyperparameter tersebut selanjutnya akan digunakan dan divariasikan untuk pelatihan model pada penelitian ini. Variasi *hyperparameter* dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Variasi Hyperparameter

Arsitektur	Optimizer	Learning Rate	Epoch	Batch Size
MobileNetV2	Adam	1e-4	50	32
	Adam	1e-3	50	
	RMSProp	1e-4	50	
	RMSProp	1e-3	50	
	Adam	1e-4	100	
	Adam	1e-3	100	
	RMSProp	1e-4	100	
	RMSProp	1e-3	100	
EfficientNetB0	Adam	1e-4	50	32
	Adam	1e-3	50	
	RMSProp	1e-4	50	
	RMSProp	1e-3	50	
	Adam	1e-4	100	
	Adam	1e-3	100	
	RMSProp	1e-4	100	
	RMSProp	1e-3	100	

3.2.6 Evaluasi Model

Pada evaluasi model akan dilakukan pada model masing-masing *hyperparameter* yang sudah ditentukan. Model akan dievaluasi dengan data *validasi* yang sebelumnya dipisah agar mendapatkan hasil evaluasi yang objektif serta tidak terpengaruh oleh bias dari data yang digunakan dalam data latih. Setelah itu akan dilakukan perbandingan performa arsitektur MobileNetV2 dan EfficientNetB0 dengan menghitung nilai *accuracy*, *precision*, *recall*, dan *f1-score*.

3.2.7 Hasil dan Pembahasan

Pada tahap ini setelah model selesai dilakukan evaluasi akan mengeluarkan hasil perbandingan antara arsitektur MobileNetV2 dan EfficientNetB0. Hasil performa dari kedua arsitektur tersebut akan dibandingkan untuk mengetahui model mana yang paling optimal dalam melakukan klasifikasi penyakit malaria yang disebabkan *Plasmodium vivax* berdasarkan citra sel darah berdasarkan metrik evaluasi. Berdasarkan hasil analisis, akan diperoleh kesimpulan mengenai efektivitas masing masing arsitektur CNN dalam melakukan klasifikasi citra sel darah. Selain itu, kesimpulan juga dapat menjadi saran untuk penelitian atau pengembangan lebih lanjut.

3.3 Alat dan Bahan Tugas Akhir

Alat dan Bahan yang digunakan dalam penelitian tugas akhir adalah sebagai berikut.

3.3.1 Alat

Alat yang digunakan untuk melakukan penelitian, adalah sebagai berikut

1. Laptop Lenovo Ideapad 310 dengan spesifikasi sistem operasi Windows 10 Pro 64-bit, RAM 12 GB, Processor Intel(R) Core(TM) i5-7200U.
2. Visual Studio Code Versi 1.99.3 untuk Prapemrosesan Data.
3. Kaggle Notebook P100 GPU untuk Pelatihan Model.

3.3.2 Bahan

Bahan yang digunakan pada penelitian ini yaitu dataset “P. vivax (malaria) infected human blood smears” BBBC041 Version 1 yang didapat dari Broad Bioimage Benchmark Collection [51].

3.4 Metode Pengembangan

Pada penelitian ini metode pengembangan yang digunakan adalah CNN dengan membandingkan arsitektur MobileNetV2 dan EfficientNetB0. Penelitian ini akan mengklasifikasikan penyakit malaria yang disebabkan oleh *Plasmodium vivax* dengan menggunakan citra sel darah.

3.5 Ilustrasi Perhitungan Metode

Berikut ini adalah ilustrasi perhitungan yang diterapkan pada metode yang digunakan.

3.5.1 Convolutional Layer

Untuk memperjelas proses perhitungan pada convolution layer, berikut ini merupakan contoh proses konvolusi dengan menggunakan filter 3×3 , *padding same* (0), *activation ReLU*, dan *stride* 1, seperti yang terlihat pada Gambar 3.5 berikut.

0	0	0	0	0	0
0	7	2	3	3	0
0	4	5	3	8	0
0	3	3	2	8	0
0	2	8	7	2	0
0	0	0	0	0	0

Gambar 3.5 Nilai Piksel

Selanjutnya diterapkan filter matriks pada citra untuk mengekstraksi fitur-fitur citra seperti tepi dan pola tertentu. Filter matriks yang dapat dilihat pada Gambar 3.6 berikut.

1	0	-1
1	0	-1
1	0	-1

Gambar 3.6 Filter Matriks

Langkah berikutnya dilakukan proses perhitungan pada Gambar 3.5 yang dikaitkan dengan kernel 3×3 pada Gambar 3.6. Proses *convolutional layer* akan menggunakan perhitungan dari Rumus 2.1 diulang dengan menggeser kernel sebanyak 1 stride, sehingga diperoleh hasil perhitungan dengan nilai sebagai berikut.

Posisi 1

0	0	0	0	0	0
0	7	2	3	3	0
0	4	5	3	8	0
0	3	3	2	8	0
0	2	8	7	2	0
0	0	0	0	0	0

Perhitungan Posisi 1:

$$(0 \times 1 + 0 \times 0 + 0 \times -1) + \\ (0 \times 1 + 7 \times 0 + 2 \times -1) + \\ (0 \times 1 + 4 \times 0 + 5 \times -1 = 0)$$

Perhitungan dilakukan dengan cara mengkalikan nilai piksel (0,0) dengan filter matriks (0,0), dilanjutkan dengan nilai piksel (0,1) dengan filter matriks (0,1) dan seterusnya sampai nilai piksel (2,2) dengan filter matriks (2,2) didapatkan hasil -7, tetapi karena fungsi aktivasi ReLu sehingga hasil akhir perhitungan adalah 0. Perhitungan dilanjutkan sampai dengan posisi selanjutnya.

Posisi 2

0	0	0	0	0	0
0	7	2	3	3	0
0	4	5	3	8	0
0	3	3	2	8	0

0	2	8	7	2	0
0	0	0	0	0	0

Perhitungan Posisi 2:

$$(0 \times 1 + 0 \times 0 + 0 \times -1) + \\ (7 \times 1 + 2 \times 0 + 3 \times -1) + \\ (4 \times 1 + 5 \times 0 + 3 \times -1) = 5$$

Posisi 3

0	0	0	0	0	0
0	7	2	3	3	0
0	4	5	3	8	0
0	3	3	2	8	0
0	2	8	7	2	0
0	0	0	0	0	0

Perhitungan Posisi 3:

$$(0 \times 1 + 0 \times 0 + 0 \times -1) + \\ (2 \times 1 + 3 \times 0 + 3 \times -1) + \\ (5 \times 1 + 3 \times 0 + 8 \times -1) = 0$$

Posisi 4

0	0	0	0	0	0
0	7	2	3	3	0
0	4	5	3	8	0
0	3	3	2	8	0

0	2	8	7	2	0
0	0	0	0	0	0

Perhitungan Posisi 4:

$$(0 \times 1 + 0 \times 0 + 0 \times -1) + \\ (3 \times 1 + 3 \times 0 + 0 \times -1) + \\ (3 \times 1 + 8 \times 0 + 0 \times -1) = 6$$

Setelah perhitungan selesai sampai dengan baris dan kolom terakhir matriks nilai piksel maka berpindah kepada baris ke 2 kolom pertama pada nilai piksel. Perhitungan dilakukan sampai baris terakhir kolom terakhir pada nilai piksel

Setelah dilakukan perhitungan dengan cara yang sama untuk semua nilai piksel yang ada, sehingga dapat diketahui:

$$\begin{array}{lll} W & = \text{input size} & = 4 \times 4 \times 1 \quad (W = 4) \\ F & = \text{size filters} & = 3 \times 3 \times 1 \quad (F = 3) \\ P & = \text{input padding} & = 1 \quad (P = 1) \\ K & = \text{kernel} & = 1 \quad (K = 1) \\ S & = \text{jumlah pergeseran} & = 1 \quad (S = 1) \end{array}$$

Sehingga, dimensi output dari proses konvolusi dapat dihitung sebagai berikut:

$$V = \frac{W - F + 2P}{S} + 1 = \frac{4 - 3 + 2(1)}{1} + 1 = \frac{3}{1} + 1 = 4$$

Maka, didapatkan hasil untuk matriks outputnya adalah $V \times V \times K = 4 \times 4 \times 1$. Berikut ini adalah hasil matriks dari layer konvolusi.

0	5	0	6
x	x	x	x
x	x	x	x
x	x	x	x

Gambar 3.7 Hasil Perhitungan *Convolutional Layer*

3.5.2 *Pooling Layer*

Setelah melalui proses konvolusi, hasil output akan digunakan sebagai inputan untuk *pooling layer*. Pada tahap ini diterapkan metode *max pooling* dengan ukuran 2×2 dan stride 2. *Max pooling* adalah teknik yang berfungsi untuk mereduksi dimensi data dengan cara memilih nilai maksimum dari setiap area 2×2 pada feature map. Proses *max pooling* dilihat pada Gambar 3.8.

6	3	8	19
2	1	12	18
4	1	9	10
5	6	7	2

6	19
6	10

Gambar 3.8 Ilustrasi Perhitungan *Pooling Layer*

Gambar 3.8 merupakan tahap *max pooling* yang dilakukan dengan ukuran kernel 2×2 . Dari nilai kernel yang ada, hanya nilai tertinggi saja yang dipilih. Setiap kernel akan bergeser 2 langkah (strides 2), dan proses ini berjalan hingga seluruh area di proses.

3.5.3 Fully Connected Layer

Hasil dari tahap flatten layer berupa vektor satu dimensi yang berisi sejumlah nilai. Vektor ini kemudian digunakan sebagai input pada fully connected layer. Pada tahap ini, setiap nilai dalam vektor akan dikalikan dengan bobot tertentu yang telah dipelajari oleh model selama proses pelatihan. Setelah itu, hasil perkalian tersebut akan dijumlahkan dengan bias, yaitu nilai tambahan yang membantu model untuk lebih fleksibel dalam melakukan prediksi.

Misalkan layer fully connected layer memiliki 5 output, yang artinya mengklasifikasikan citra sel darah sebagai lima kelas berbeda (*uninfected*, *ring*, *trophozoite*, *schizont*, dan *gametocyte*). Berikut adalah contoh perhitungan *fully connected layer*.

Misalkan bobot setiap neuron adalah sebagai berikut:

Bobot neuron 1 (*uninfected*) = $[w_{11}, w_{12}, w_{13}, w_{14}] = [0.2, -0.1, 0.4, 0.3]$

Bobot neuron 2 (*ring*) = $[w_{21}, w_{22}, w_{23}, w_{24}] = [-0.3, 0.5, -0.2, 0.1]$

Bobot neuron 3 (*trophozoite*) = $[w_{31}, w_{32}, w_{33}, w_{34}] = [0.6, 0.2, 0.3, -0.5]$

Bobot neuron 4 (*schizont*) = $[w_{41}, w_{42}, w_{43}, w_{44}] = [0.1, 0.1, 0.1, 0.1]$

Bobot neuron 5 (*gametocyte*) = $[w_{51}, w_{52}, w_{53}, w_{54}] = [-0.2, 0.3, 0.4, 0.5]$

Nilai bias sebagai berikut:

$[b_1, b_2, b_3, b_4, b_5] = [0.5, -0.4, 0.3, 0.0, 0.2]$

Maka dilakukan perhitungan untuk setiap neuron nya.

Perhitungan untuk neuron 1

$$\begin{aligned}
 &= (6 \times 0.2 + 19 \times (-0.1) + 6 \times 0.4 + 10 \times 0.3) + 0.5 \\
 &= (1.2 - 1.9 + 2.4 + 3.0) + 0.5 \\
 &= 4.7 + 0.5 \\
 &= 5.2
 \end{aligned}$$

Perhitungan untuk neuron 2

$$\begin{aligned}
 &= (6 \times (-0.3) + 19 \times 0.5 + 6 \times (-0.2) + 10 \times 0.1) + 0.4 \\
 &= (-1.8 + 9.5 - 1.2 + 1.0) - 0.4 \\
 &= 7.5 - 0.4 \\
 &= 7.1
 \end{aligned}$$

Perhitungan untuk neuron 3

$$\begin{aligned}
 &= (6 \times 0.6 + 19 \times 0.2 + 6 \times 0.3 + 10 \times (-0.5)) + 0.3 \\
 &= (3.6 + 3.8 + 1.8 - 5.0) + 0.3 \\
 &= 4.2 + 0.3 \\
 &= 4.5
 \end{aligned}$$

Perhitungan untuk neuron 4

$$\begin{aligned}
 &= (6 \times 0.1 + 19 \times 0.1 + 6 \times 0.1 + 10 \times 0.1) + 0.0 \\
 &= (0.6 + 1.9 + 0.6 + 1.0) \\
 &= 4.1
 \end{aligned}$$

Perhitungan untuk neuron 5

$$\begin{aligned}
 &= (6 \times (-0.2) + 19 \times 0.3 + 6 \times 0.4 + 10 \times 0.5) + 0.2 \\
 &= (-1.2 + 5.7 + 2.4 + 5.0) + 0.2 \\
 &= 11.9 + 0.2 \\
 &= 12.1
 \end{aligned}$$

3.5.4 Softmax

Tahap selanjutnya dalam proses perhitungan adalah menentukan nilai softmax menggunakan rumus yang membandingkan nilai eksponensial dari setiap output terhadap total eksponensial dari seluruh output dengan menggunakan Rumus 2.4.

Selanjutnya menghitung jumlah semua nilai eksponensial dari masing masing nilai output pada *fully connected layer*.

$$\sum_{j=1}^5 \sum e^{x_j} = 181.27 + 1210.94 + 90.02 + 60.34 + 179747.53 \\ \approx 181290.10$$

Langkah berikutnya adalah menghitung nilai softmax untuk setiap output.

$$\text{Softmax}(o_1) = \frac{e^{5.2}}{\sum_{j=1}^5 \sum e^{x_j}} = \frac{181.27}{181290.10} \approx 0.0010$$

$$\text{Softmax}(o_2) = \frac{e^{7.1}}{\sum_{j=1}^5 \sum e^{x_j}} = \frac{1210.94}{181290.10} \approx 0.0067$$

$$\text{Softmax}(o_3) = \frac{e^{4.5}}{\sum_{j=1}^5 \sum e^{x_j}} = \frac{90.02}{181290.10} \approx 0.0005$$

$$\text{Softmax}(o_4) = \frac{e^{4.1}}{\sum_{j=1}^5 \sum e^{x_j}} = \frac{60.34}{181290.10} \approx 0.0003$$

$$\text{Softmax}(o_5) = \frac{e^{12.1}}{\sum_{j=1}^5 \sum e^{x_j}} = \frac{179747.53}{181290.10} \approx 0.9915$$

Maka didapatkan hasil nilai akhir untuk masing-masing probabilitas adalah sebagai berikut:

Kelas uninfected ≈ 0.0010 (0.10%)

Kelas ring ≈ 0.0067 (0.67%)

Kelas trophozoite ≈ 0.0005 (0.05%)

Kelas schizont ≈ 0.0003 (0.03%)

Kelas gametocyte ≈ 0.9915 (99.15%)

Sehingga didapatkan hasil output ke-5 menunjukkan probabilitas tertinggi yaitu 99,15%, model memprediksi bahwa objek tersebut masuk ke dalam kelas gametocyte.

3.6 Ilustrasi Rancangan Pengujian

Setelah model selesai melakukan pelatihan akan dilakukan evaluasi dengan menggunakan data validasi yang akan digunakan untuk mengevaluasi kinerja model. Data ini terdiri dari data-data yang tidak digunakan selama pelatihan, sehingga model dapat dievaluasi berdasarkan kemampuannya untuk menggeneralisasi pada data baru. Input data validasi memiliki format dan struktur yang sama dengan data pelatihan agar model dapat melakukan klasifikasi dengan benar.

Setelah itu model akan dievaluasi dengan menggunakan *confusion matrix* dengan menggunakan empat metrik yaitu *accuracy*, *precision*, *recall* dan *f1-score*. *Accuracy* mengukur seberapa banyak prediksi klasifikasi sel darah yang benar dibandingkan dengan total prediksi yang dilakukan oleh model. *Precision* menunjukkan seberapa akurat model dalam mengidentifikasi sel darah tertentu. *Recall* mengukur seberapa banyak dari total positif yang berhasil diidentifikasi oleh model. *F1-score* adalah metrik yang menggabungkan presisi dan recall menjadi satu nilai yang memberikan keseimbangan antara keduanya.

Tabel 3.2 Contoh Tabel Hasil *Confusion Matrix*

	Predicted					
	Kelas	Uninfected	Ring	Trophozoite	Schizont	Gametocyte
Actual	Uninfected	90	5	3	1	1
	Ring	4	85	6	3	2
	Trophozoite	3	6	80	7	4
	Schizont	2	3	5	88	2
	Gametocyte	1	2	4	6	87

Berdasarkan nilai hasil *confusion matrix* pada Tabel 3.2, maka perhitungan metode evaluasi adalah sebagai berikut. Keterangan dan rumus dapat dilihat pada subbab 2.2.10.

Diketahui:

$$\text{Total TP} = 90 + 85 + 80 + 88 + 87 = 430$$

$$\text{Total FP} = 10 + 16 + 18 + 17 + 9 = 70$$

$$\text{Total FN} = 10 + 15 + 20 + 12 + 13 = 70$$

a. Accuracy

Accuracy dihitung dengan menggunakan Rumus 2.7.

$$\text{Accuracy} = \frac{90 + 85 + 80 + 88 + 87}{500}$$

$$\text{Accuracy} = \frac{430}{500} = 0,86$$

b. Precision

Precision dihitung dengan menggunakan Rumus 2.8.

1. Uninfected

$$\text{Precision} = \frac{90}{90 + 10} = \frac{90}{100} = 0.9000 \text{ atau } 90,00\%$$

2. Ring

$$Precision = \frac{85}{85 + 16} = \frac{85}{101} = 0.8416 \text{ atau } 84,16\%$$

3. Trophozoite

$$Precision = \frac{80}{80 + 18} = \frac{80}{98} = 0.8163 \text{ atau } 81,63\%$$

4. Schizont

$$Precision = \frac{88}{88 + 17} = \frac{88}{105} = 0.8381 \text{ atau } 83,81\%$$

5. Gametocyte

$$Precision = \frac{87}{87 + 9} = \frac{87}{96} = 0.9063 \text{ atau } 90,63\%$$

c. Recall

Recall dihitung dengan menggunakan Rumus 2.9.

1. Uninfected

$$Recall = \frac{90}{90 + 10} = \frac{90}{100} = 0.9000 \text{ atau } 90,00\%$$

2. Ring

$$Recall = \frac{85}{85 + 15} = \frac{85}{100} = 0.8500 \text{ atau } 85,00\%$$

3. Trophozoite

$$Recall = \frac{80}{80 + 28} = \frac{80}{100} = 0.8000 \text{ atau } 80,00\%$$

4. Schizont

$$Recall = \frac{88}{88 + 12} = \frac{88}{100} = 0.8800 \text{ atau } 88,00\%$$

5. Gametocyte

$$Recall = \frac{87}{87 + 13} = \frac{87}{100} = 0.8700 \text{ atau } 87,00\%$$

d. F1-Score

F1-Score dihitung dengan menggunakan Rumus 2.10.

1. Uninfected

$$F1 - Score = 2 \times \frac{(0,9000 \times 0,9000)}{(0,9000 + 0,9000)} = \frac{1,62}{1,80} \\ = 0,9000 \text{ atau } 90,00\%$$

2. Ring

$$F1 - Score = 2 \times \frac{(0,8416 \times 0,8500)}{(0,8416 + 0,8500)} = \frac{1,43}{1,69} \\ = 0,8462 \text{ atau } 84,62\%$$

3. Trophozoite

$$F1 - Score = 2 \times \frac{(0,8163 \times 0,8000)}{(0,8163 + 0,8000)} = \frac{1,31}{1,62} \\ = 0,8086 \text{ atau } 80,86\%$$

4. Schizont

$$F1 - Score = 2 \times \frac{(0,8381 \times 0,8800)}{(0,8381 + 0,8800)} = \frac{1,48}{1,72} \\ = 0,8605 \text{ atau } 86,05\%$$

5. Gametocyte

$$F1 - Score = 2 \times \frac{(0,9063 \times 0,8700)}{(0,9063 + 0,8700)} = \frac{1,58}{1,78} \\ = 0,8876 \text{ atau } 88,76\%$$

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Bagian ini menyajikan hasil penelitian yang diperoleh melalui tahapan yang telah dilaksanakan. Proses penelitian mencakup kegiatan pengumpulan data, prapemrosesan data, pengembangan model, serta evaluasi model menggunakan *confusion matrix*. Hasil yang diperoleh berasal dari percobaan terhadap dua arsitektur model, yaitu MobileNetV2 dan EfficientNetB0, dengan penerapan variasi *hyperparameter* yang serupa pada masing-masing model.

4.1.1 Pengumpulan Data

Dataset yang digunakan terdiri atas lima kelas citra sel darah merah malaria yang disebabkan oleh *Plasmodium vivax*, yaitu tidak terinfeksi (*uninfected*), *ring*, *trophozoite*, *schizont*, dan *gametocyte*. Dataset ini merupakan data sekunder yang diperoleh dari situs *Broad Bioimage Benchmark Collection* (BBBC) dengan nomor akses BBBC041 Versi 1. Contoh dataset ditampilkan pada Gambar 3.2. Dataset tersebut kemudian dibagi menjadi data pelatihan dan data validasi.

4.1.2 Prapemrosesan Data

Pada tahap selanjutnya yaitu dilakukan prapemrosesan data. Tahap-tahap pada prapemrosesan data adalah sebagai berikut.

4.1.2.1 Cropping

Cropping yang dilakukan yaitu memotong gambar pada *dataset* berdasarkan kelas yang sudah dianotasi. Tahap yang dilakukan yaitu mendeteksi *bounding box* pada gambar yang sudah dianotasi, setelah itu dilakukan pemotongan pada gambar berdasarkan *bounding box* tersebut dan dikelompokkan berdasarkan kelas. Kode yang digunakan pada tahap cropping dapat dilihat pada Kode 4.1.

```
# Extract bounding box coordinates
bbox = extract_bounding_box(obj)

if bbox is None:
    print(f"Object {i} in {image_path} does not have
valid bounding box coordinates")
    continue

x, y, w, h = bbox

# Ensure coordinates are within image bounds
x = max(0, int(x))
y = max(0, int(y))
w = min(image.shape[1] - x, int(w))
h = min(image.shape[0] - y, int(h))

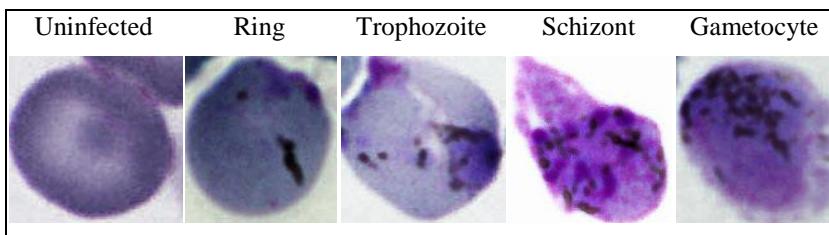
if w > 0 and h > 0:
    # Crop the object directly without any
    # preprocessing
    cropped = image[y:y+h, x:x+w]

    # Generate unique filename with dataset type
    category_counters[category] += 1
    filename =
f"{dataset_type}_{image_name_prefix}_{category}_{category_coun
ters[category]:03d}.png"
    output_path = os.path.join(category_dir, filename)

    cv2.imwrite(output_path, cropped)
    print(f"Cropped {category} saved to:
{output_path}")
    cropped_count += 1
else:
    print(f"Invalid crop dimensions for object {i}:
w={w}, h={h}")
```

Kode 4.1 Proses *Cropping*

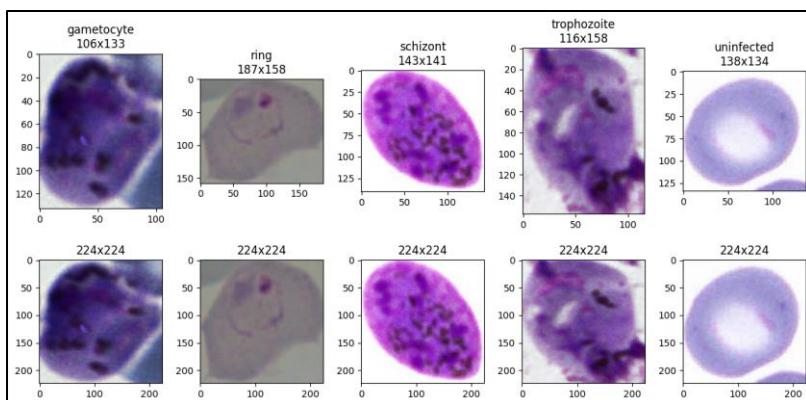
Proses cropping diawali dengan mengambil nilai koordinat bounding box. Selanjutnya, nilai koordinat disesuaikan agar tetap berada dalam batas ukuran gambar. Setelah itu dilakukan pengecekan agar lebar dan tinggi hasil crop bernilai positif, sehingga area yang akan dipotong valid. Jika valid, maka bagian citra sesuai bounding box akan dipotong.. Hasil cropping pada citra dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 Hasil *Cropping*

4.1.2.2 *Resize*

Resize dilakukan dengan mengubah gambar agar sesuai. Pada tahap ini citra dari masing masing kelas diubah menjadi ukuran 224×224 piksel. Hasil *resize* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Hasil *Resize*

4.1.2.3 Normalisasi

Normalisasi dilakukan dengan menggunakan nilai rata-rata (*mean*) dan standar deviasi (*std*), yaitu [0,485, 0,456, 0,406] untuk *mean* dan [0,229, 0,224, 0,225] untuk standar deviasi. Nilai-nilai tersebut digunakan untuk menormalkan setiap kanal warna (merah, hijau, dan biru) dengan cara mengurangkan *mean* kemudian membaginya dengan standar deviasi, sehingga distribusi intensitas piksel pada citra menjadi lebih merata dan piksel citra memiliki rata-rata mendekati nol. Kode untuk tahap resize dan normalisasi dapat dilihat pada Kode 4.2.

```
def get_dataloaders_from_json(json_path, dataset_path,
fold_name, batch_size=32, img_size=(224, 224)):
    # Ambil semua nama kelas dari folder dataset
    class_names = sorted([d for d in os.listdir(dataset_path)
if os.path.isdir(os.path.join(dataset_path, d))])
    class_to_idx = {cls_name: i for i, cls_name in
enumerate(class_names)}

    # Transformasi gambar
    transform = transforms.Compose([
        transforms.Resize(img_size),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406],
std=[0.229, 0.224, 0.225])
    ])

    # Load JSON
    with open(json_path, "r") as f:
        folds_data = json.load(f)

    train_files = folds_data[fold_name]["train"]
    val_files = folds_data[fold_name]["val"]

    # Dataset
    train_dataset = MalariaDataset(dataset_path, train_files,
class_to_idx, transform=transform)
    val_dataset = MalariaDataset(dataset_path, val_files,
class_to_idx, transform=transform)
```

```

# DataLoader
train_loader = DataLoader(train_dataset,
batch_size=batch_size, shuffle=True, num_workers=4,
pin_memory=True, persistent_workers=True)
val_loader = DataLoader(val_dataset,
batch_size=batch_size, shuffle=False, num_workers=4,
pin_memory=True, persistent_workers=True)

return train_loader, val_loader, class_to_idx

```

Kode 4.2 Proses *Resize* dan Normalisasi

Tahap pemrosesan dapat dilihat pada bagian *transform* dengan menggunakan fungsi `transforms.Compose()`. Fungsi `transforms.Resize(img_size)` digunakan untuk mengubah ukuran setiap citra menjadi 224×224 piksel. Fungsi `transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])` digunakan untuk normalisasi nilai setiap kanal warna (merah, hijau, biru) berdasarkan rata-rata (mean) dan standar deviasi (std).

4.1.3 Pengembangan Model

Pada tahap pengembangan model dilakukan pembagian data dengan menggunakan *k-fold cross validation*, setelah itu dilakukan pelatihan dan evaluasi pada masing masing model arsitektur.

4.1.3.1 Pembagian Data

Proses pembagian data dilakukan untuk memisahkan dataset menjadi data pelatihan dan data validasi. Jumlah data yang digunakan berjumlah 780 citra, yang terbagi ke dalam lima kelas dengan rasio pembagian 80:20 pada setiap iterasi. Kode yang digunakan untuk proses pembagian data tersebut dapat dilihat pada Kode 4.3.

```

image_paths = []
labels = []

for class_name in os.listdir(dataset_path):
    class_dir = os.path.join(dataset_path, class_name)
    if os.path.isdir(class_dir):
        for fname in os.listdir(class_dir):
            if fname.lower().endswith('.jpg', '.png',
'.jpeg')):
                image_paths.append(os.path.join(class_name,
fname))
                labels.append(class_name)

# Konversi list ke array
image_paths = list(image_paths)
labels = list(labels)

print(f"Total data: {len(image_paths)}")
# print("Contoh data:", image_paths[:5], labels[:5])

# Buat K-Fold
skf = StratifiedKFold(n_splits=5, shuffle=True,
random_state=2025)

folds_data = {}

for fold, (train_idx, val_idx) in
enumerate(skf.split(image_paths, labels), start=1):
    folds_data[f"fold_{fold}"] = {
        "train": [image_paths[i] for i in train_idx],
        "val": [image_paths[i] for i in val_idx]
    }

# Simpan ke file JSON
with open("skfold_TA_156.json", "w") as f:
    json.dump(folds_data, f, indent=4)

```

Kode 4.3 Proses Pembagian Data

Proses kfold diawali dengan membuat daftar berkas gambar (image_paths) dan labelnya (labels) dengan menelusuri struktur folder, setiap nama folder diperlakukan sebagai label kelas, sedangkan setiap berkas bertipe .jpg/.png/.jpeg dicatat sebagai sampel. StratifiedKFold(shuffle=True, random_state=2025) digunakan untuk

menghasilkan indeks pelatihan dan validasi pada tiap lipatan secara acak, namun tetap mempertahankan distribusi kelas yang seimbang di train dan val. Setiap lipatan, kode menyusun pasangan subset “train” dan “val” dalam sebuah kamus folds_data, lalu menyimpannya ke berkas JSON. Setelah dilakukan pembagian data menggunakan kfold, akan menghasilkan pembagian data yang dapat dilihat pada Tabel 4.1.

Tabel 4.1 Pembagian Data

Iterasi	K-Fold					Data Train	Data Validation
	1	2	3	4	5		
1	156	156	156	156	156	624	156
2	156	156	156	156	156	624	156
3	156	156	156	156	156	624	156
4	156	156	156	156	156	624	156
5	156	156	156	156	156	624	156
Total Keseluruhan Citra						780	

Pada tabel pembagian data terlihat bahwa dataset dibagi menjadi data pelatihan (*train*) dan data validasi (*validation*). Pada proses *k-fold cross validation* dengan lima *fold*, setiap blok berwarna biru menunjukkan *fold* yang berfungsi sebagai data validasi pada iterasi ke-n. Sebagai contoh, pada iterasi pertama *Fold 1* digunakan sebagai data validasi, sedangkan *Fold 2* hingga *Fold 5* berperan sebagai data pelatihan. Proses ini berlanjut hingga setiap *fold* mendapatkan giliran menjadi data validasi satu kali. Pada setiap iterasi, jumlah data pelatihan sebanyak 624 data dan data validasi sebanyak 156 data.

4.1.3.2 Pelatihan Model

Pelatihan model akan dilakukan dengan menggunakan arsitektur MobileNetV2 dan EfficientNetB0 yang sudah ada.

1. Pendefinisian Arsitektur MobileNetV2

```
# Mendefinisikan jumlah kelas
num_classes = len(os.listdir(dataset_path))

# Model MobileNetV2 dari torchvision
def mobnetv2_model(num_classes):
    model = models.mobilenet_v2(weights=None)
    model.classifier[1] = nn.Linear(model.last_channel,
num_classes)
    return model

# Menampilkan ringkasan model
summary(mobnetv2_model(num_classes), input_size=(32, 3, 224,
224))
```

Kode 4.4 Mendefinisikan MobileNetV2

Kode untuk mendefinisikan arsitektur MobileNetV2 dapat dilihat pada Kode 4.4. Fungsi mobnetv2_model memanggil arsitektur models.mobilenet_v2 dari torchvision dengan weights=None yaitu tanpa bobot pralatin, kemudian mengganti lapisan klasifier terakhir model.classifier[1] dengan nn.Linear(model.last_channel, num_classes) agar dimensi keluaran sesuai jumlah kelas. Pemanggilan summary(...) digunakan untuk meninjau ringkasan arsitektur pada citra berukuran 224×224 piksel. Jumlah parameter yang dihasilkan dari MobileNetV2 adalah 2.230.277 parameter.

2. Pendefinisian Arsitektur EfficientNetB0

```
# Mendefinisikan jumlah kelas
num_classes = len(os.listdir(dataset_path))

# Model EfficientNetB0 dari torchvision
def effb0_model(num_classes):
```

```

model = models.efficientnet_b0(weights=None)
model.classifier[1] =
nn.Linear(model.classifier[1].in_features, num_classes)
return model

# Menampilkan ringkasan model
summary(effb0_model(num_classes), input_size=(32, 3, 224,
224))

```

Kode 4.5 Mendefinisikan EfficientNetB0

Kode untuk mendefinisikan arsitektur EfficientNetB0 dapat dilihat pada Kode 4.5. Fungsi effb0_model memanggil arsitektur models.efficientnet_b0 dari torchvision dengan weights=None yaitu tanpa bobot pralatih, kemudian mengganti lapisan klasifier terakhir model.classifier[1] dengan nn.Linear(model.classifier[1].in_features, num_classes) agar dimensi keluaran sesuai jumlah kelas. Pemanggilan summary(...) digunakan untuk meninjau ringkasan arsitektur pada citra berukuran 224×224 piksel. Jumlah parameter yang dihasilkan dari MobileNetV2 adalah 4.013.953 parameter.

3. Pelatihan Model untuk Setiap Arsitektur

Setelah kedua arsitektur sudah didefinisikan, selanjutnya dilakukan pelatihan dan evaluasi pada masing-masing arsitektur. Hyperparameter yang digunakan untuk pelatihan model dapat dilihat pada Tabel 3.1. Kode untuk melakukan pelatihan model dapat dilihat pada Kode 4.6.

```

num_epochs = 50 #100
learning_rate = 0.0001 #0.001

# Dictionary untuk menyimpan hasil semua fold
all_fold_results = {}

# Fungsi training per fold dengan tracking yang lebih detail
def train_fold(fold_name, train_loader, val_loader,
num_classes):

```

```

print(f"Training {fold_name.replace('_', ' ').title()}")


model = mobnetv2_model(num_classes).to(device)
#effb0_model
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(),
lr=learning_rate) #RMSprop

# Track metrics
train_losses, train_accs = [], []
val_losses, val_accs = [], []
best_val_acc = 0.0
best_epoch = 0
best_model_path = f'{fold_name}_best_model.pth'

for epoch in range(num_epochs):
    # Training phase
    model.train()
    running_loss, running_corrects = 0.0, 0

        for inputs, labels in tqdm(train_loader,
desc=f"Epoch {epoch+1}/{num_epochs} - Train"):
            inputs, labels = inputs.to(device),
labels.to(device)

            optimizer.zero_grad()
            outputs = model(inputs)
            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()

            running_loss += loss.item() * inputs.size(0)
            running_corrects += (outputs.argmax(1) ==
labels.sum()).item()

        train_loss = running_loss /
len(train_loader.dataset)
        train_acc = running_corrects /
len(train_loader.dataset)
        train_losses.append(train_loss)
        train_accs.append(train_acc)

    # Validation phase
    model.eval()
    val_loss, val_corrects = 0.0, 0
    with torch.no_grad():
        for inputs, labels in tqdm(val_loader,
desc=f"Epoch {epoch+1}/{num_epochs} - Val"):
            inputs, labels = inputs.to(device),
labels.to(device)

```

```

outputs = model(inputs)
loss = criterion(outputs, labels)

val_loss += loss.item() * inputs.size(0)
val_corrects += (outputs.argmax(1) ==
labels).sum().item()

val_loss /= len(val_loader.dataset)
val_acc = val_corrects / len(val_loader.dataset)
val_losses.append(val_loss)
val_accs.append(val_acc)

# Track best validation accuracy dan simpan model
terbaik
if val_acc > best_val_acc:
    best_val_acc = val_acc
    best_epoch = epoch + 1
    torch.save(model.state_dict(), best_model_path)

print(f"Epoch [{epoch+1}/{num_epochs}] "
      f"Train Loss: {train_loss:.4f} Acc:
{train_acc:.4f} | "
      f"Val Loss: {val_loss:.4f} Acc:
{val_acc:.4f}")

```

Kode 4.6 Pelatihan Model

Kode pelatihan training validation per lipatan (fold) menggunakan arsitektur MobileNetV2 atau EfficientNetB0. Hyperparameter didefinisikan dahulu (epoch 50 atau 100 dan *learning rate* 1e-4 atau 1e-3), fungsi *loss* CrossEntropyLoss digunakan untuk multikelas, serta optimizer Adam atau RMSProp pada model. Metrik setiap epoch dicatat dalam daftar (loss dan akurasi untuk train/val) dan ditampilkan, sementara model terbaik dipilih berdasarkan akurasi validasi tertinggi dan disimpan per fold sebagai berkas bobot (state_dict) dengan nama {fold_name}_best_model.pth. Pemakaian tqdm menyediakan progres bar untuk waktu komputasi, dan pemisahan antara mode pelatihan dan evaluasi.

4. Evaluasi Model

Setelah dilakukan pelatihan pada model, selanjutnya model dievaluasi. Kode untuk evaluasi model dapat dilihat pada Kode 4.7.

```
best_model = mobnetv2_model(num_classes).to(device)
#effb0_model
    best_model.load_state_dict(torch.load(best_model_path,
map_location=device))
    best_model.eval()

    all_preds, all_labels = [], []
    with torch.no_grad():
        for inputs, labels in val_loader:
            inputs = inputs.to(device)
            outputs = best_model(inputs)
            preds = outputs.argmax(1).detach().cpu().numpy()
            all_preds.append(preds)
            all_labels.append(labels.cpu().numpy())
    all_preds = np.concatenate(all_preds)
    all_labels = np.concatenate(all_labels)
```

Kode 4.7 Evaluasi Model

Arsitektur MobileNetV2 atau EfficientNetB0 didefinisikan ulang dan best_model dimuat ke perangkat komputasi yang tersedia. Model kemudian dievaluasi (eval()) untuk menonaktifkan perilaku pelatihan seperti dropout dan batch normalization yang memperbarui statistik. Seluruh batch digabung (np.concatenate) menjadi all_preds dan all_labels, yang selanjutnya dipakai untuk menghitung metrik evaluasi (misalnya akurasi, presisi, recall, F1-score) serta menyusun *confusion matrix* secara menyeluruh pada set validasi.

4.2 Hasil Evaluasi

Berdasarkan penelitian yang telah dilakukan, diperoleh hasil evaluasi sebagai berikut. Hasil evaluasi ditampilkan berdasarkan

arsitektur model yang digunakan, yakni MobileNetV2 dan EfficientNetB0. Setiap tabel menampilkan konfigurasi *hyperparameter* yang meliputi *optimizer*, *learning rate* (LR), *epoch*, *batch size* (BS), *folding*, serta *confusion matrix* yang terdiri atas *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Selain itu, disertakan pula metrik evaluasi berupa *accuracy* (acc), *precision* (pre), *recall* (rec), dan *f1-score* (f1).

4.2.1 Model MobileNetV2

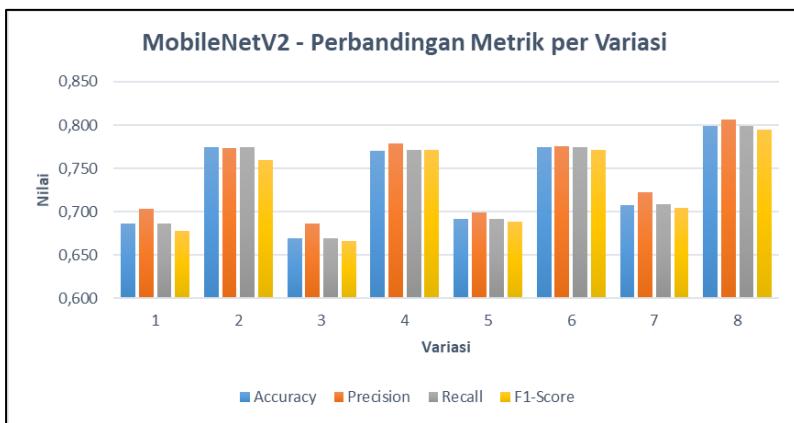
Hasil evaluasi Model MobileNetV2 dengan kombinasi *hyperparameter* dan dievaluasi dengan menggunakan *confusion matrix* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil Evaluasi Model MobileNetV2

No	Optim	LR	Epoch	Fold	Confusion Matrix				Metrik Evaluasi			
					TP	TN	FP	FN	Acc	Prec	Rec	F1-Score
1	Adam	1E-04	50	1	100	568	56	56	0,641	0,647	0,643	0,634
				2	108	576	48	48	0,692	0,709	0,691	0,671
				3	101	569	55	55	0,647	0,711	0,649	0,644
				4	114	582	42	42	0,731	0,733	0,731	0,724
				5	112	580	44	44	0,718	0,717	0,719	0,717
				Rata-Rata				0,686	0,703	0,686	0,678	
2	Adam	1E-03	50	1	113	581	43	43	0,724	0,733	0,725	0,720
				2	125	593	31	31	0,801	0,809	0,800	0,798
				3	118	586	38	38	0,756	0,739	0,756	0,703
				4	129	597	27	27	0,827	0,829	0,827	0,824
				5	119	587	37	37	0,763	0,759	0,765	0,755
				Rata-Rata				0,774	0,774	0,775	0,760	
3	RMSprop	1E-04	50	1	100	568	56	56	0,641	0,668	0,644	0,628
				2	105	573	51	51	0,673	0,684	0,672	0,669
				3	106	574	50	50	0,679	0,700	0,680	0,684

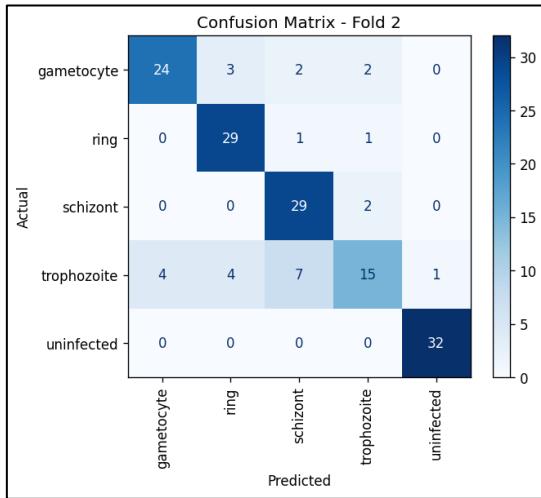
No	Optim	LR	Epoch	Fold	Confusion Matrix				Metrik Evaluasi			
					TP	TN	FP	FN	Acc	Prec	Rec	F1-Score
4	RMSprop	1E-03	50	4	106	574	50	50	0,679	0,690	0,678	0,677
				5	105	573	51	51	0,673	0,687	0,674	0,670
				Rata-Rata				0,669	0,686	0,669	0,666	
				1	116	584	40	40	0,744	0,757	0,744	0,746
				2	131	599	25	25	0,840	0,837	0,839	0,837
				3	118	586	38	38	0,756	0,763	0,757	0,755
5	Adam	1E-04	100	4	122	590	34	34	0,782	0,805	0,783	0,790
				5	114	582	42	42	0,731	0,732	0,732	0,731
				Rata-Rata				0,771	0,779	0,771	0,772	
				1	104	572	52	52	0,667	0,680	0,667	0,658
				2	110	578	46	46	0,705	0,710	0,704	0,706
				3	106	574	50	50	0,679	0,687	0,680	0,681
6	Adam	1E-03	100	4	109	577	47	47	0,699	0,705	0,699	0,697
				5	110	578	46	46	0,705	0,716	0,707	0,700
				Rata-Rata				0,691	0,699	0,691	0,688	
				1	121	589	35	35	0,776	0,777	0,776	0,769
				2	117	585	39	39	0,750	0,757	0,749	0,746
				3	123	591	33	33	0,788	0,794	0,789	0,790
7	RMSprop	1E-04	100	4	120	588	36	36	0,769	0,765	0,769	0,763
				5	123	591	33	33	0,788	0,785	0,790	0,786
				Rata-Rata				0,774	0,776	0,774	0,771	
				1	110	578	46	46	0,705	0,743	0,707	0,698
				2	108	576	48	48	0,692	0,707	0,692	0,690
				3	107	575	49	49	0,686	0,697	0,687	0,686
8	RMSprop	1E-03	100	4	120	588	36	36	0,769	0,777	0,769	0,770
				5	107	575	49	49	0,686	0,687	0,687	0,679
				Rata-Rata				0,708	0,722	0,708	0,705	
				1	123	591	33	33	0,788	0,820	0,788	0,795
				2	129	597	27	27	0,827	0,825	0,826	0,816
				3	126	594	30	30	0,808	0,808	0,808	0,805
				4	121	589	35	35	0,776	0,782	0,775	0,764
				5	124	592	32	32	0,795	0,799	0,796	0,795
				Rata-Rata				0,799	0,807	0,799	0,795	

Visualisasi dari hasil evaluasi MobileNetV2 dapat dilihat pada Gambar 4.3 di bawah.



Gambar 4.3 Visualisasi Hasil Evaluasi MobileNetV2

Berdasarkan hasil evaluasi model dengan *kfold*, dapat dilihat bahwa kombinasi ke-4 yaitu dengan *optimizer* RMSprop, *learning rate* 0,001, *epoch* 50, dan *batch size* 32 dengan model terbaiknya terdapat pada *fold* 2 dengan mendapatkan nilai *accuracy* 0,840, *precision* 0,837, *recall* 0,839, dan *f1-score* 0,839. Namun, kombinasi yang mendapatkan nilai rata rata terbaik didapatkan pada kombinasi ke-8 yaitu dengan *optimizer* RMSprop, *learning rate* 0,001, *epoch* 100, dan *batch size* 32 dengan mendapatkan nilai *accuracy* 0,799, *precision* 0,807, *recall* 0,799, dan *f1-score* 0,795.



Gambar 4.4 Confusion Matrix MobileNetV2

Confusion matrix pada Gambar 4.4 di atas merupakan *confusion matrix* dari kombinasi variasi terbaik yaitu variasi ke-8 dengan fold ke-2 sebagai fold terbaiknya. *Confusion matrix* tersebut menunjukkan bahwa model dapat benar memprediksi sebanyak 129 dari 156 data. Contoh perhitungan pada kelas Gametocyte adalah sebagai berikut.

1. Precision

Precision dihitung dengan menggunakan Rumus 2.8.

$$\frac{24}{24 + 4} = \frac{24}{28} = 0,857$$

2. Recall

Recall dihitung dengan menggunakan Rumus 2.9.

$$\frac{24}{24 + 7} = \frac{24}{31} = 0,774$$

3. F1-Score

F1-Score dihitung dengan menggunakan Rumus 2.8.

$$2 \times \frac{(0,857 \times 0,774)}{(0,857 + 0,774)} = \frac{1,327}{1,631} = 0,814$$

Berdasarkan perhitungan tersebut kelas Gametocyte mendapatkan nilai metrik evaluasi sebesar *precision* 0,857, *recall* sebesar 0,774, dan *f1-score* sebesar 0,814.

4.2.2 Model EfficientNetB0

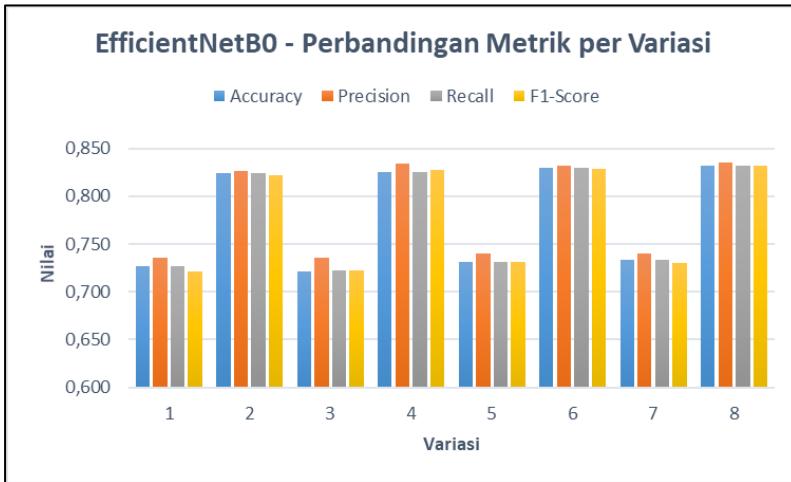
Hasil evaluasi Model EfficientNetB0 dengan kombinasi *hyperparameter* dan dievaluasi dengan menggunakan *confusion matrix* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil Evaluasi EfficientNetB0

No	Optim	LR	Epoch	Fold	Confusion Matrix				Metrik Evaluasi			
					TP	TN	FP	FN	Acc	Prec	Rec	F1-Score
1	Adam	1E-04	50	1	111	579	45	45	0,712	0,725	0,711	0,704
				2	116	584	40	40	0,744	0,747	0,742	0,738
				3	110	578	46	46	0,705	0,712	0,706	0,706
				4	117	585	39	39	0,750	0,748	0,749	0,742
				5	113	581	43	43	0,724	0,745	0,726	0,716
				Rata-Rata				0,727	0,736	0,727	0,721	
2	Adam	1E-03	50	1	131	599	25	25	0,840	0,852	0,840	0,843
				2	127	595	29	29	0,814	0,816	0,813	0,809
				3	132	600	24	24	0,846	0,846	0,846	0,843
				4	128	596	28	28	0,821	0,818	0,820	0,814
				5	125	593	31	31	0,801	0,799	0,803	0,799
				Rata-Rata				0,824	0,826	0,824	0,822	
3	RMSprop	1E-04	50	1	103	571	53	53	0,660	0,690	0,661	0,664
				2	115	583	41	41	0,737	0,737	0,736	0,735
				3	115	583	41	41	0,737	0,749	0,737	0,740
				4	119	587	37	37	0,763	0,765	0,765	0,763
				5	111	579	45	45	0,712	0,738	0,712	0,713
				Rata-Rata				0,722	0,736	0,722	0,723	
4	RMSprop	1E-03	50	1	129	597	27	27	0,827	0,830	0,827	0,828
				2	128	596	28	28	0,821	0,831	0,819	0,820

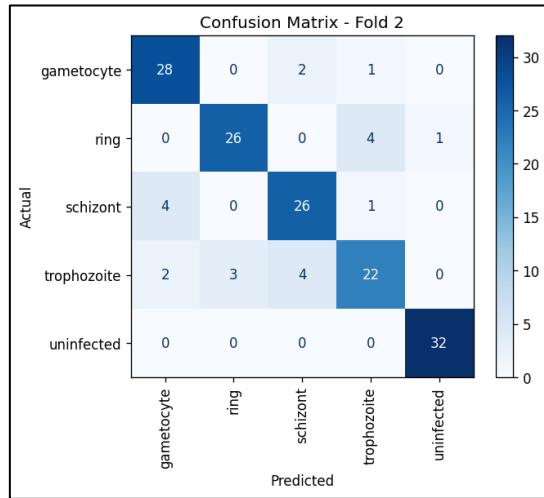
No	Optim	LR	Epoch	Fold	Confusion Matrix				Metrik Evaluasi				
					TP	TN	FP	FN	Acc	Prec	Rec	F1-Score	
					3	128	596	28	28	0,821	0,840	0,821	0,826
5	Adam	1E-04	100		4	129	597	27	27	0,827	0,829	0,827	0,826
					5	130	598	26	26	0,833	0,841	0,834	0,835
					Rata-Rata				0,826	0,834	0,826	0,827	
					1	112	580	44	44	0,718	0,720	0,719	0,717
					2	113	581	43	43	0,724	0,725	0,723	0,723
					3	115	583	41	41	0,737	0,763	0,738	0,743
6	Adam	1E-03	100		4	118	586	38	38	0,756	0,769	0,757	0,757
					5	112	580	44	44	0,718	0,720	0,719	0,718
					Rata-Rata				0,731	0,740	0,731	0,731	
					1	126	594	30	30	0,808	0,813	0,807	0,808
					2	130	598	26	26	0,833	0,833	0,833	0,833
					3	130	598	26	26	0,833	0,832	0,833	0,829
7	RMSprop	1E-04	100		4	130	598	26	26	0,833	0,834	0,833	0,831
					5	131	599	25	25	0,840	0,848	0,841	0,843
					Rata-Rata				0,829	0,832	0,829	0,829	
					1	103	571	53	53	0,660	0,659	0,660	0,654
					2	124	592	32	32	0,795	0,797	0,794	0,793
					3	115	583	41	41	0,737	0,745	0,738	0,735
8	RMSprop	1E-03	100		4	120	588	36	36	0,769	0,787	0,769	0,768
					5	110	578	46	46	0,705	0,714	0,705	0,702
					Rata-Rata				0,733	0,740	0,733	0,731	
					1	128	596	28	28	0,821	0,829	0,821	0,823
					2	134	602	22	22	0,859	0,858	0,858	0,857
					3	129	597	27	27	0,827	0,826	0,827	0,826
					4	129	597	27	27	0,827	0,830	0,826	0,827
					5	129	597	27	27	0,827	0,832	0,828	0,829
					Rata-Rata				0,832	0,835	0,832	0,832	

Visualisasi dari hasil evaluasi EfficientNetB0 dapat dilihat pada Gambar 4.5 di bawah.



Gambar 4.5 Visualisasi Hasil Evaluasi EfficientNetB0

Berdasarkan hasil evaluasi model dengan *kfold*, dapat dilihat bahwa kombinasi terbaik didapatkan pada kombinasi ke-8 yaitu dengan *optimizer* RMSprop, *learning rate* 0,001, *epoch* 100, dan *batch size* 32 dengan model terbaiknya terdapat pada *fold* 2. Kombinasi ini mendapatkan nilai rata-rata tertinggi, yaitu *accuracy* 0,832, *precision* 0,835, *recall* 0,832, dan *f1-score* 0,832.



Gambar 4.6 Confusion Matrix EfficientNetB0

Confusion matrix pada Gambar 4.6 di atas merupakan *confusion matrix* dari kombinasi variasi terbaik yaitu variasi ke-8 dan fold ke-2. *Confusion matrix* tersebut menunjukkan bahwa model dapat benar memprediksi sebanyak 134 dari 156 data. Contoh perhitungan pada kelas Gametocyte adalah sebagai berikut.

1. Precision

Precision dihitung dengan menggunakan Rumus 2.8.

$$\frac{28}{28 + 6} = \frac{28}{34} = 0,826$$

2. Recall

Recall dihitung dengan menggunakan Rumus 2.9.

$$\frac{28}{28 + 3} = \frac{28}{31} = 0,903$$

3. F1-Score

F1-Score dihitung dengan menggunakan Rumus 2.8.

$$2 \times \frac{(0,826 \times 0,903)}{(0,826 + 0,903)} = \frac{1,492}{1,729} = 0,862$$

Berdasarkan perhitungan tersebut kelas Gametocyte mendapatkan nilai metrik evaluasi sebesar *precision* 0,826, *recall* sebesar 0,903, dan *f1-score* sebesar 0,862.

4.3 Perbandingan Evaluasi Model

Hasil perbandingan dari kedua model berdasarkan kombinasi *hyperparameter* dengan performa terbaik berdasarkan rata-rata metrik evaluasi *accuracy*, *precision*, *recall*, dan *f1-score* dari hasil kfold.

Tabel 4.4 Hasil Perbandingan Evaluasi Model Terbaik

Model	Optimizer	LR	Epoch	BS	Metrik Evaluasi			
					Acc	Pre	Rec	F1
MobileNetV2	RMSprop	1e-3	100	32	0,799	0,807	0,799	0,795
EfficientNetB0	RMSprop	1e-3	100	32	0,832	0,835	0,832	0,832

Pada Tabel 4.4 dapat dilihat bahwa hasil analisis perbandingan dua model terbaik didapatkan bahwa EfficientNetB0 dengan kombinasi *hyperparameter optimizer RMSprop, learning rate 0,001, epoch 100, dan batch size 32* memperoleh rata rata *accuracy* 0,832, *precision* 0,835, *recall* 0,832, dan *f1-score* 0,832. MobileNetV2 dengan model terbaik memiliki kombinasi *hyperparameter* yang sama memperoleh rata rata *accuracy* 0,799, *precision* 0,807, *recall* 0,799, dan *f1-score* 0,795. Berdasarkan hal tersebut menjadikan EfficientNetB0 sebagai model dengan performa terbaik.

Tabel 4.5 Hasil Perbandingan Evaluasi Model Terbaik per Parameter

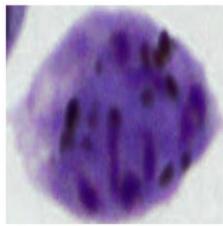
Model	Parameter	Perbandingan (Metrik/Parameter)			
		Acc/Parameter	Pre/Parameter	Rec/Parameter	F1/Parameter
MobileNetV2	2.230.277	3,58e-7	3,62e-7	3,58e-7	3,56e-7
EfficientNetB0	4.013.953	2,07e-7	2,08e-7	2,07e-7	2,07e-7

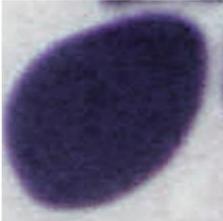
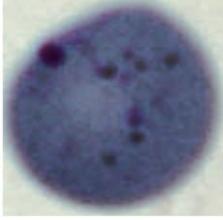
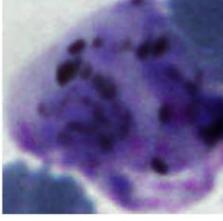
Pada Tabel 4.5 dapat dilihat perbandingan untuk nilai metrik per parameter, pada EfficientNetB0 yaitu $2,07\text{e-}7$ untuk *accuracy*, *recall*, dan *f1-score*, serta $2,08\text{e-}7$ untuk *precision*. Sementara itu, MobileNetV2 memperoleh nilai metrik per parameter sebesar $3,58\text{e-}7$ pada *accuracy* dan *recall*, $3,62\text{e-}7$ pada *precision*, serta $3,56\text{e-}7$ pada *f1-score*. Berdasarkan hasil tersebut, dapat diketahui bahwa MobileNetV2 lebih unggul dalam hal komputasi dibandingkan EfficientNetB0.

4.4 Pengujian Model

Pengujian model akan dilakukan perbandingan antara label aktual dan hasil prediksi dari kedua arsitektur model, yaitu MobileNetV2 dan EfficientNetB0, terhadap data yang digunakan. Tujuan dari perbandingan ini adalah untuk menilai sejauh mana model mampu mengenali citra dengan akurat. Tabel 4.6 menampilkan contoh hasil klasifikasi model terhadap beberapa data beserta label aktualnya.

Tabel 4.6 Perbandingan Hasil Klasifikasi

No	Citra	Aktual	Prediksi MobileNetV2	Prediksi EfficientNetB0
1		Schizont	Schizont	Schizont

No	Citra	Aktual	Prediksi MobileNetV2	Prediksi EfficientNetB0
2		Uninfected	Uninfected	Uninfected
3		Gametocyte	Trophozoite	Gametocyte
4		Ring	Ring	Ring
5		Trophozoite	Gametocyte	Gametocyte

Berdasarkan tabel di atas kedua model memprediksi citra label *Schizont*, *Uninfected*, dan *Ring* dengan benar. Pada citra aktual *Gametocyte* model MobileNetV2 memprediksi citra tersebut sebagai *Trophozoite* dan pada citra aktual *Trophozoite* kedua model memprediksinya sebagai *Gametocyte*.

4.5 Analisis Hasil Keseluruhan

Berdasarkan seluruh tahapan penelitian yang sudah dilakukan mulai dari pengumpulan data hingga evaluasi yang dilakukan selama proses pengembangan model, dapat dirangkum bahwa penelitian ini berhasil membandingkan performa arsitektur MobileNetV2 dan EfficientNetB0 dalam melakukan klasifikasi penyakit malaria yang disebabkan oleh parasit Plasmodium Vivax berdasarkan citra sel darah. Dataset yang digunakan merupakan data sekunder yang diperoleh dari situs *Broad Bioimage Benchmark Collection* (BBBC) dengan nomor akses BBBC041 Versi 1 dan terbagi menjadi lima kelas stadium yaitu *uninfected, ring, trophozoite, schizont, gametocyte*. Pada penelitian ini menerapkan teknik *k-fold cross validation* sebanyak lima fold untuk menguji konsistensi data.

Hasil percobaan dari kombinasi *hyperparameter* untuk setiap arsitektur didapatkan bahwa model dengan variasi *optimizer* RMSprop, *learning rate* 1e-3, *epoch* 100, dan *batch size* 32 merupakan yang paling optimal untuk setiap arsitektur. Hasil evaluasi menunjukkan bahwa arsitektur EfficientNetB0 memperoleh nilai *accuracy*, *recall*, dan *f1-score* sebesar 0,832 dan nilai *precision* sebesar 0,835. Nilai tersebut memiliki selisih 0,033 pada *accuracy* dan *recall*, 0,028 pada *precision*, dan 0,037 pada *f1-score* yang lebih tinggi dibandingkan MobileNetV2, menunjukkan bahwa EfficientNetB0 memiliki performa terbaik.

Pada perbandingan untuk nilai metrik per parameter, MobileNetV2 memperoleh nilai metrik per parameter sebesar 3,58e-7 pada *accuracy* dan *recall*, 3,62e-7 pada *precision*, serta 3,56e-7 pada *f1-score*. Dibandingkan dengan EfficientNetB0 yang memiliki nilai 2,07e-7

untuk *accuracy*, *recall*, dan *f1-score*, serta 2,08e-7 untuk *precision*. Hal tersebut menunjukkan bahwa MobileNetV2 lebih baik pada komputasi.

Pada pengujian citra pada beberapa gambar MobileNetV2 dan EfficientNetB0 dapat memprediksi citra label *Schizont*, *Uninfected*, dan *Ring* dengan benar. Pada citra aktual *Gametocyte* model MobileNetV2 memprediksi citra tersebut sebagai *Trophozoite*. Sementara itu pada citra aktual *Trophozoite*, kedua model memprediksi citra tersebut sebagai *Gametocyte*

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, diperoleh kesimpulan sebagai berikut. Penelitian ini berhasil membandingkan kinerja arsitektur MobileNetV2 dan EfficientNetB0 dalam mengklasifikasikan penyakit malaria yang disebabkan oleh *Plasmodium vivax* berdasarkan citra sel darah yang terbagi ke dalam lima kelas. Hasil evaluasi menunjukkan bahwa EfficientNetB0 memperoleh nilai *accuracy*, *recall*, dan *f1-score* sebesar 0,832 dan nilai *precision* sebesar 0,835. Nilai tersebut memiliki selisih 0,033 pada *accuracy* dan *recall*, 0,028 pada *precision*, dan 0,037 pada *f1-score* yang lebih tinggi dibandingkan MobileNetV2. Pada perbandingan nilai metric per parameter MobileNetV2 memperoleh nilai metrik per parameter sebesar 3,58e-7 pada *accuracy* dan *recall*, 3,62e-7 pada *precision*, serta 3,56e-7 pada *f1-score*. Dibandingkan dengan EfficientNetB0 yang memiliki nilai 2,07e-7 untuk *accuracy*, *recall*, dan *f1-score*, serta 2,08e-7 untuk *precision*. Hal tersebut menunjukkan bahwa EfficientNetB0 memiliki performa paling optimal, tetapi MobileNetV2 lebih baik pada hal komputasi.

5.2 Saran

Berdasarkan hasil penelitian, terdapat beberapa saran yang dapat diberikan adalah sebagai berikut.

1. Menerapkan berbagai teknik prapemrosesan citra, seperti penyesuaian warna atau pencahayaan, guna memperoleh distribusi

warna dan kontras yang lebih seragam sehingga citra lebih mudah dikenali.

2. Penggunaan teknik augmentasi perlu dipertimbangkan untuk meningkatkan keragaman data citra tanpa mengurangi detail penting pada gambar.
3. Model yang telah dikembangkan dapat diimplementasikan dalam bentuk aplikasi agar dapat dimanfaatkan untuk membantu proses deteksi malaria yang disebabkan oleh *Plasmodium vivax*.

DAFTAR PUSTAKA

- [1] A. Rajagopalan and C. M. Ilboudo, “Malaria,” *Pediatr. Rev.*, vol. 40, no. 3, pp. 151–153, Mar. 2019, doi: 10.1542/pir.2017-0225.
- [2] E. Ahmadpour *et al.*, “Transfusion-transmitted malaria: A systematic review and meta-analysis,” *Open Forum Infectious Diseases*, vol. 6, no. 7. 2019. doi: 10.1093/ofid/ofz283.
- [3] F. E. Siagian, “Pulmonary Complication in Severe Malaria,” *Int. J. Pathog. Res.*, 2021, doi: 10.9734/ijpr/2021/v8i130194.
- [4] F. Elhag, “A Survey of Malaria Transmission, Knowledge about Malaria Disease, Symptoms and Treatment,” vol. 11, 2020.
- [5] D. D. Brown, S. Solomon, D. Lerner, and M. Del Rio, “Malaria and acute kidney injury,” *Pediatr. Nephrol.*, vol. 35, no. 4, 2020, doi: 10.1007/s00467-018-4191-0.
- [6] A. M. Gimenez, R. F. Marques, M. Regiart, and D. Y. Bargieri, “Diagnostic Methods for Non-Falciparum Malaria,” *Frontiers in Cellular and Infection Microbiology*, vol. 11. 2021. doi: 10.3389/fcimb.2021.681063.
- [7] D. Das *et al.*, “Review Article: A systematic literature review of microscopy methods reported in malaria clinical trials,” *American Journal of Tropical Medicine and Hygiene*, vol. 104, no. 3. 2021. doi: 10.4269/ajtmh.20-1219.
- [8] K. Habtamu, B. Petros, and G. Yan, “Plasmodium vivax: the potential obstacles it presents to malaria elimination and eradication,” *Tropical Diseases, Travel Medicine and Vaccines*, vol. 8, no. 1. 2022. doi: 10.1186/s40794-022-00185-3.
- [9] R. E. Howes *et al.*, “Global Epidemiology of Plasmodium vivax,” *Am. J. Trop. Med. Hyg.*, vol. 95, no. 6 Suppl, pp. 15–34, Dec.

- 2016, doi: 10.4269/ajtmh.16-0141.
- [10] C. Surjadjaja, A. Surya, and J. K. Baird, “Epidemiology of Plasmodium vivax in Indonesia,” *Am. J. Trop. Med. Hyg.*, vol. 95, 2016, doi: 10.4269/ajtmh.16-0093.
 - [11] Y. Omori, “Image Augmentation for Eye Contact Detection Based on Combination of Pre-trained Alex-Net CNN and SVM,” *J. Comput.*, vol. 15, no. 3, 2020, doi: 10.17706/jcp.15.3.85-97.
 - [12] 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, 2022, doi: 10.22219/repositor.v4i2.1283.
 - [13] A. R. Hermanto, A. Aziz, and S. Sudianto, “Perbandingan Arsitektur MobileNetV2 dan ResNet50 untuk Klasifikasi Jenis Buah Kurma,” *J. Sist. dan Teknol. Inf.*, vol. 12, no. 4, 2024.
 - [14] Oriza Sativa Fiojati, Nani Mintarsih, and Yuli Maharetta Arianti, “PERBANDINGAN ALGORITMA EFFICIENTNETB0 DAN INCEPTIONV3 DALAM KLASIFIKASI CITRA JENIS ANJING,” *J. Ilm. Tek.*, vol. 2, no. 2, pp. 12–16, May 2023, doi: 10.56127/juit.v2i2.677.
 - [15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00474.
 - [16] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *36th International Conference on Machine Learning, ICML 2019*, 2019.
 - [17] N. Nurhaeni, S. E. Prastyo, A. Hidayat, and F. N. Anisa, “Pemodelan Sistem Deteksi Parasit Malaria pada Citra

- Mikroskopis Sel Darah Menggunakan Metode Deep Learning,” *SMATIKA J.*, vol. 14, no. 02, pp. 409–416, Dec. 2024, doi: 10.32664/smatika.v14i02.1475.
- [18] D. R. Fauzi and G. A. Haqdu D, “Comparison of CNN Models Using EfficientNetB0, MobileNetV2, and ResNet50for Traffic Density with Transfer Learning,” *J. Intell. Syst. Technol. Informatics*, vol. 1, no. 1, pp. 22–30, 2025.
- [19] S. P. Raikar, “Anopheles,” Encyclopedia Britannica. [Online]. Available: <https://www.britannica.com/animal/Anopheles>
- [20] N. Khan and J. P. Daily, “Update on pathogenesis, management, and control of Plasmodium vivax,” *Curr. Opin. Infect. Dis.*, vol. 35, no. 5, pp. 404–409, Oct. 2022, doi: 10.1097/QCO.0000000000000867.
- [21] “CDC - DPDx - Malaria.” Accessed: Mar. 12, 2025. [Online]. Available: <https://www.cdc.gov/dpdx/malaria/>
- [22] N. Ghanchi, A. A. Khan, A. Raheem, and M. A. Beg, “Plasmodium vivax Mimicking Morphologic Features of Plasmodium falciparum,” *Cureus*, Nov. 2020, doi: 10.7759/cureus.11406.
- [23] H. A. Nugroho, I. M. D. Maysanjaya, N. A. Setiawan, E. E. H. Murhandarwati, and W. K. Oktoeberza, “Feature analysis for stage identification of Plasmodium vivax based on digital microscopic image,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 13, no. 2, p. 721, Feb. 2019, doi: 10.11591/ijeeecs.v13.i2.pp721-728.
- [24] T. Xu, N. Theera-Umpon, and S. Auephanwiriyakul, “Staining-Independent Malaria Parasite Detection and Life Stage Classification in Blood Smear Images,” *Appl. Sci.*, vol. 14, no.

- 18, p. 8402, Sep. 2024, doi: 10.3390/app14188402.
- [25] F. A. S. Araujo, N. D. Colares, U. P. Carvalho, C. F. F. Costa Filho, and M. G. F. Costa, “Plasmodium Life Cycle-Stage Classification on Thick Blood Smear Microscopy Images using Deep Learning: A Contribution to Malaria Diagnosis,” in *2023 19th International Symposium on Medical Information Processing and Analysis (SIPAIM)*, IEEE, Nov. 2023, pp. 1–4. doi: 10.1109/SIPAIM56729.2023.10373424.
- [26] L. Piel, C. Hernandez, Y. Jaszczyzyn, D. Naquin, and J. M. Santos, “Single cell RNA-sequencing of malaria merozoites reveals that the parasite prepares for life within the red blood cell by activating a gene network.” Sep. 26, 2024. doi: 10.1101/2024.09.26.614664.
- [27] D. Kepple *et al.*, “From Genes to Biomarkers: Understanding the Biology of Malaria Gametocytes and Their Detection,” in *Genetic Polymorphisms - New Insights*, IntechOpen, 2022. doi: 10.5772/intechopen.99364.
- [28] N. Z. Munantri, H. Sofyan, and M. Y. Florestiyanto, “APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK IDENTIFIKASI UMUR POHON,” *Telematika*, vol. 16, no. 2, p. 97, Jan. 2020, doi: 10.31315/telematika.v16i2.3183.
- [29] A. Glenn and C. E. Armstrong, “Physiology of red and white blood cells,” *Anaesth. Intensive Care Med.*, vol. 20, no. 3, pp. 170–174, Mar. 2019, doi: 10.1016/j.mpaic.2019.01.001.
- [30] V. Pretini *et al.*, “Red Blood Cells: Chasing Interactions,” *Front. Physiol.*, vol. 10, Jul. 2019, doi: 10.3389/fphys.2019.00945.
- [31] I. M. I. Alkafrawi and Z. A. Dakhell, “Blood Cells Classification Using Deep Learning Technique,” in *2022 International*

- Conference on Engineering & MIS (ICEMIS)*, IEEE, Jul. 2022, pp. 1–6. doi: 10.1109/ICEMIS56295.2022.9914281.
- [32] A. K. Mishra, V. K. Jatav, and K. Bairwa, “A Study of Blood Cells, Work and Function,” *Int. J. Multidiscip. Res.*, vol. 5, no. 1, Feb. 2023, doi: 10.36948/ijfmri.2023.v05i01.1523.
 - [33] S. Ilahiyah and A. Nilogiri, “Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network,” *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.)*, vol. 3, no. 2, 2018.
 - [34] P. A. Nugroho, I. Fenriana, and R. Arijanto, “Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia,” *Algor*, vol. 2, no. 1, 2020.
 - [35] A. R. Bohari, M. K. Hidayat, and D. A. Andianingsari, “Review: Identifikasi Batubara dan Gangue Menggunakan Machine Learning dan Deep Learning,” *IMTechno J. Ind. Manag. Technol.*, vol. 5, no. 1, 2024, doi: 10.31294/imtechno.v5i1.3013.
 - [36] P. Oza, P. Sharma, S. Patel, F. Adedoyin, and A. Bruno, “Image Augmentation Techniques for Mammogram Analysis,” *J. Imaging*, vol. 8, no. 5, p. 141, May 2022, doi: 10.3390/jimaging8050141.
 - [37] E. Suherman, B. Rahman, D. Hindarto, and H. Santoso, “Implementation of ResNet-50 on End-to-End Object Detection (DETR) on Objects,” *SinkrOn*, vol. 8, no. 2, pp. 1085–1096, Apr. 2023, doi: 10.33395/sinkron.v8i2.12378.
 - [38] M. Masud, A. E. Eldin Rashed, and M. S. Hossain, “Convolutional neural network-based models for diagnosis of breast cancer,” *Neural Comput. Appl.*, vol. 34, no. 14, pp. 11383–11394, Jul. 2022, doi: 10.1007/s00521-020-05394-5.

- [39] S. A. Agrawal, V. D. Rewaskar, R. A. Agrawal, S. S. Chaudhari, Y. Patil, and N. S. Agrawal, “Advancements in NSFW Content Detection: A Comprehensive Review of ResNet-50 Based Approaches,” *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 4, 2023.
- [40] M. J. Nordin, O. W. Xin, and N. Aziz, “Food Image Recognition for Price Calculation using Convolutional Neural Network,” in *Proceedings of the 2019 3rd International Conference on Digital Signal Processing*, New York, NY, USA: ACM, Feb. 2019, pp. 80–85. doi: 10.1145/3316551.3316557.
- [41] N. S. Putra, B. F. Hutabarat, and U. Khaira, “Implementasi Algoritma Convolutional Neural Network Untuk Identifikasi Jenis Kelamin Dan Ras,” *Decod. J. Pendidik. Teknol. Inf.*, vol. 3, no. 1, pp. 82–93, Feb. 2023, doi: 10.51454/decode.v3i1.123.
- [42] J. Sanjaya and M. Ayub, “Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup,” *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, Aug. 2020, doi: 10.28932/jutisi.v6i2.2688.
- [43] F. A. A. Harahap, R. Mardianson Sinaga, K. Arifin, and K. Saputra, “Implementasi Algoritma Convolutional Neural Network Untuk Mendeteksi Penyakit Ginjal,” *J. Teknol. Informasi, Komput. dan Apl.*, vol. 4, no. 2, 2022.
- [44] K. R. Wardani and L. Leonardi, “Klasifikasi Penyakit pada Daun Anggur menggunakan Metode Convolutional Neural Network,” *J. Tekno Insentif*, vol. 17, no. 2, pp. 112–126, Oct. 2023, doi: 10.36787/jti.v17i2.1130.
- [45] Y. N. FUADAH, I. D. UBAIDULLAH, N. IBRAHIM, F. F. TALININGSING, N. K. SY, and M. A. PRAMUDITHO, “Optimasi Convolutional Neural Network dan K-Fold Cross

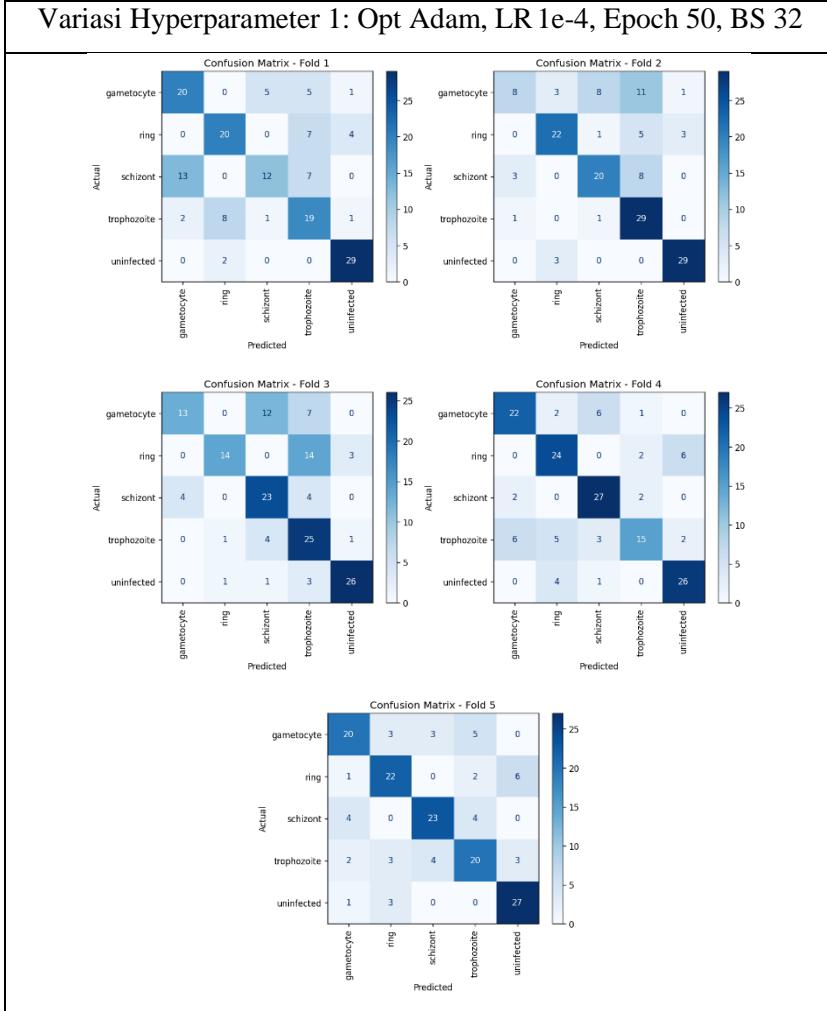
- Validation pada Sistem Klasifikasi Glaukoma,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 3, p. 728, Jul. 2022, doi: 10.26760/elkomika.v10i3.728.
- [46] F. Paraijun, R. N. Aziza, and D. Kuswardani, “Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Kesegaran Buah Berdasarkan Citra Buah,” *KILAT*, vol. 11, no. 1, pp. 1–9, Apr. 2022, doi: 10.33322/kilat.v10i2.1458.
- [47] D. Marcella, Y. Yohannes, and S. Devella, “Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Dengan Arsitektur VGG-19,” *J. Algoritm.*, vol. 3, no. 1, pp. 60–70, Oct. 2022, doi: 10.35957/algoritme.v3i1.3331.
- [48] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, “Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem,” *Technologies*, vol. 9, no. 4, 2021, doi: 10.3390/technologies9040081.
- [49] Afis Julianto, Andi Sunyoto, and Ferry Wahyu Wibowo, “OPTIMASI HYPERPARAMETER CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI PENYAKIT TANAMAN PADI,” *Tek. Teknol. Inf. dan Multimed.*, vol. 3, no. 2, 2022, doi: 10.46764/teknimedia.v3i2.77.
- [50] Dede Kurniadi, Rifky Muhammad Shidiq, and Asri Mulyani, “Perbandingan Penggunaan Optimizer dalam Klasifikasi Sel Darah Putih Menggunakan Convolutional Neural Network,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 14, no. 1, pp. 77–86, Mar. 2025, doi: 10.22146/jnteti.v14i1.17162.
- [51] “P. vivax (malaria) infected human blood smears | Broad

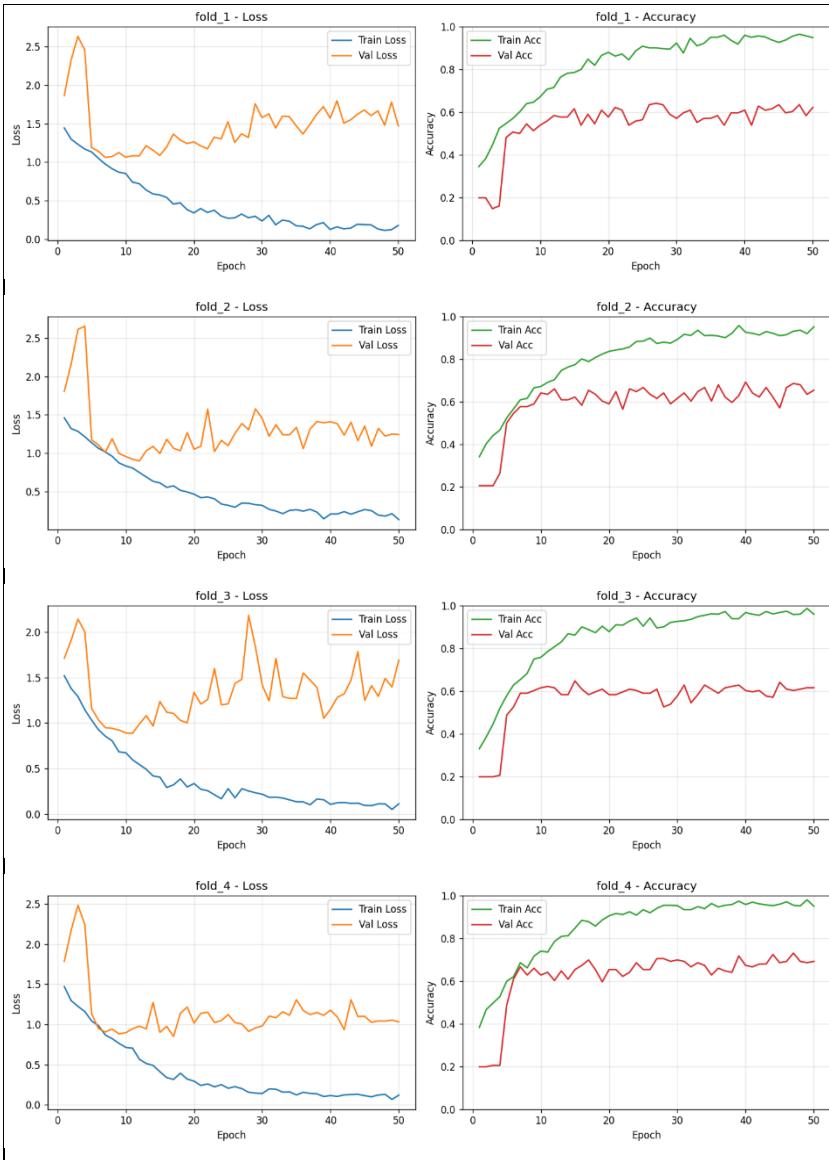
Bioimage Benchmark Collection.” Accessed: Apr. 16, 2025.
[Online]. Available: <https://bbbc.broadinstitute.org/BBBC041>

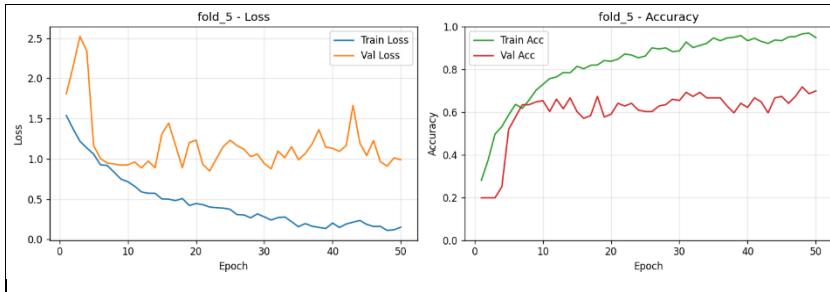
LAMPIRAN

Lampiran I Hasil Evaluasi Model MobileNetV2

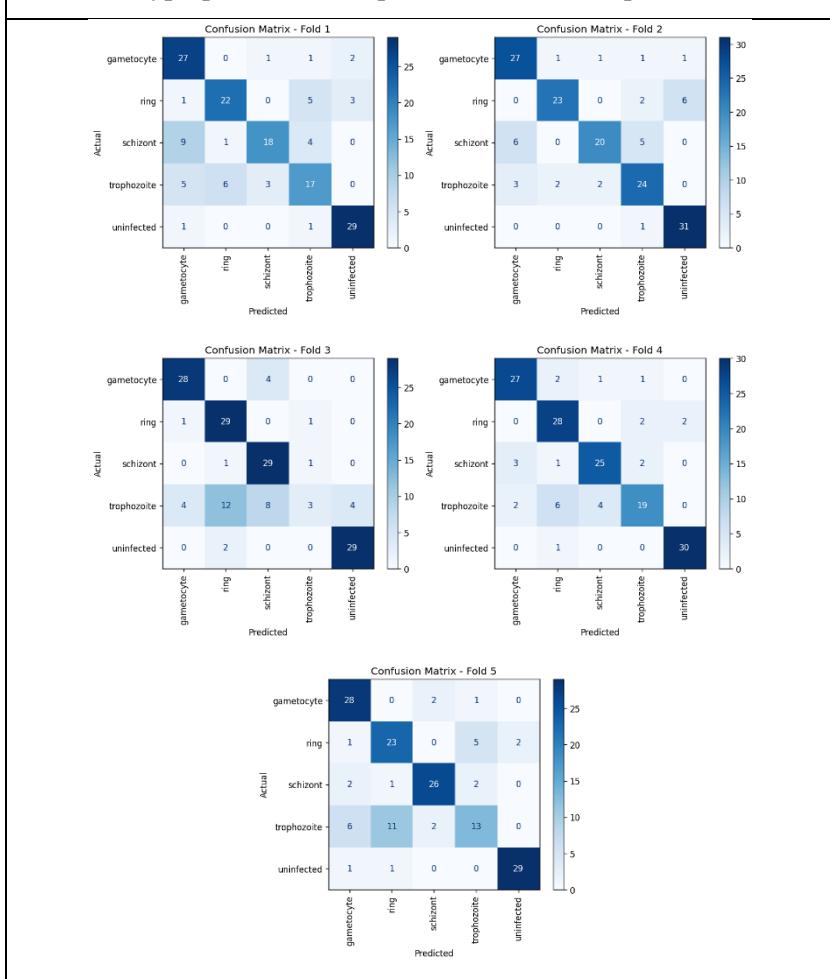
Variasi Hyperparameter 1: Opt Adam, LR 1e-4, Epoch 50, BS 32

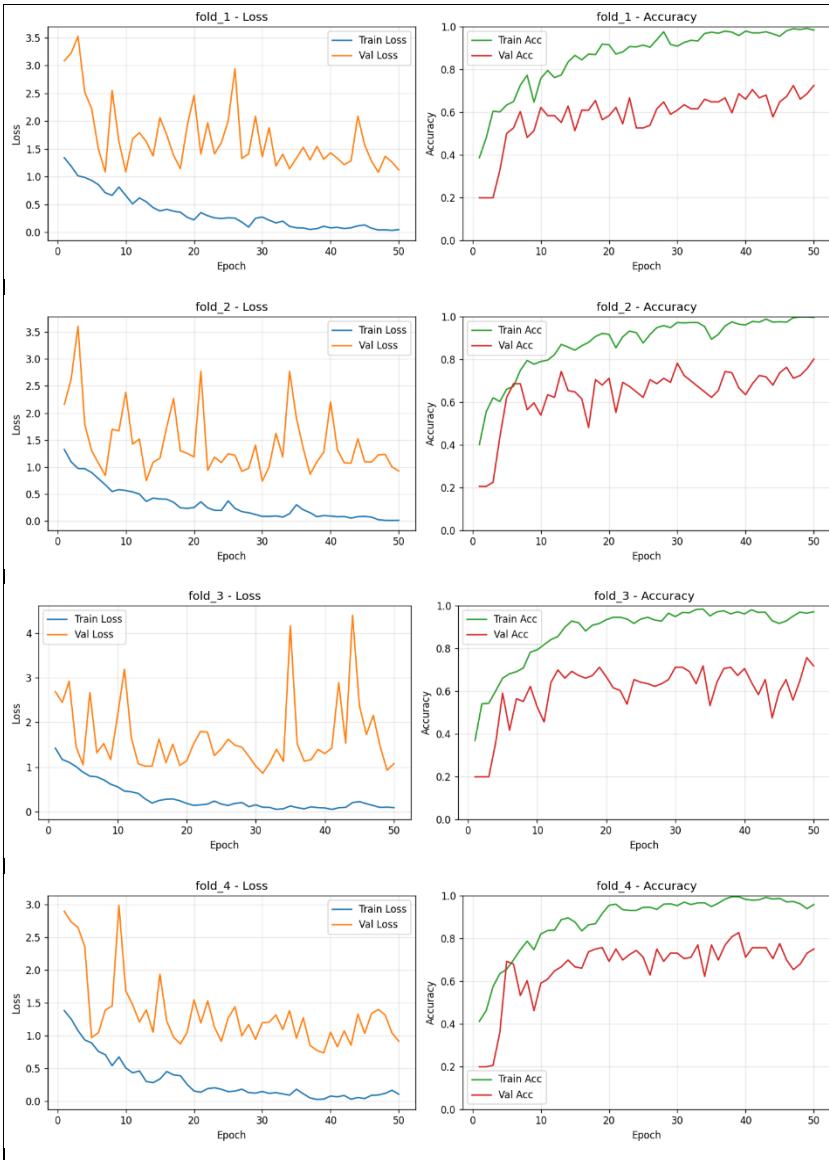


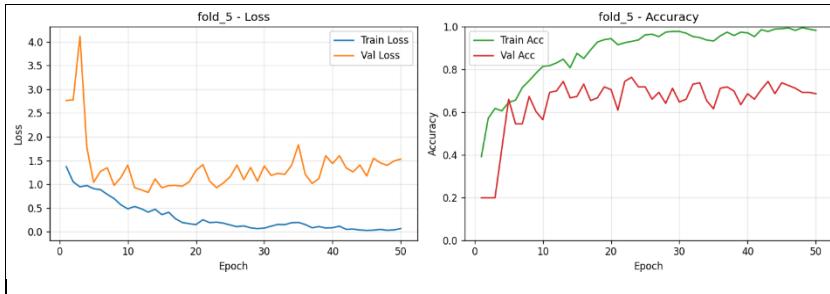




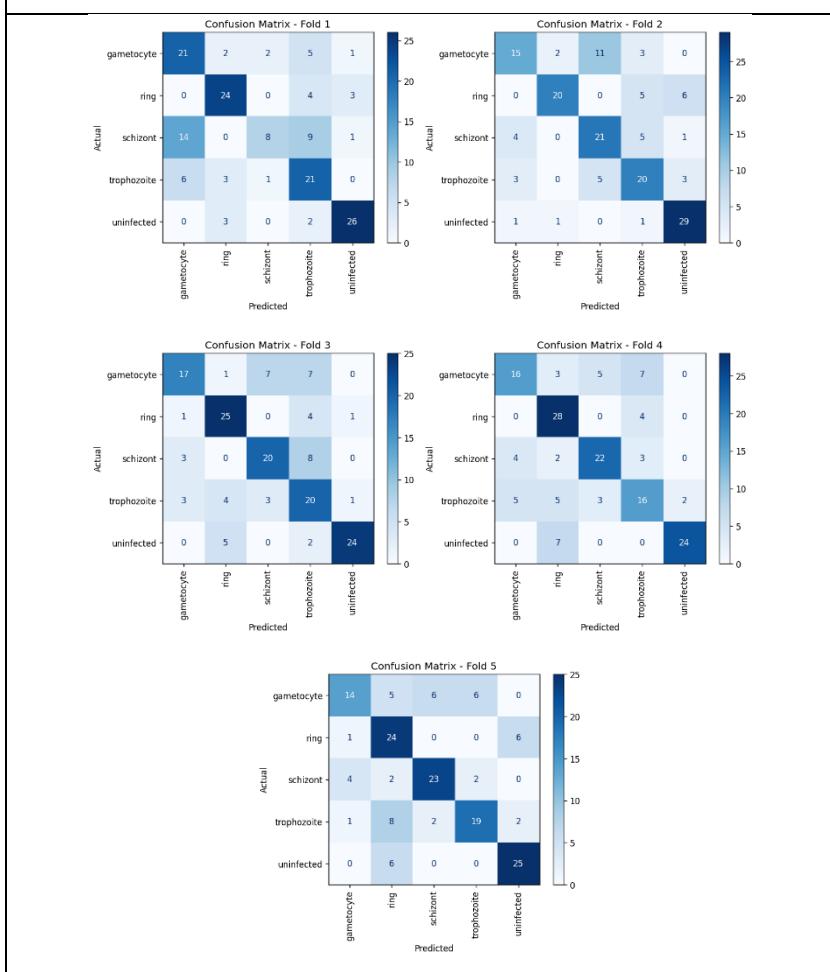
Variasi Hyperparameter 2: Opt Adam, LR 1e-3, Epoch 50, BS 32

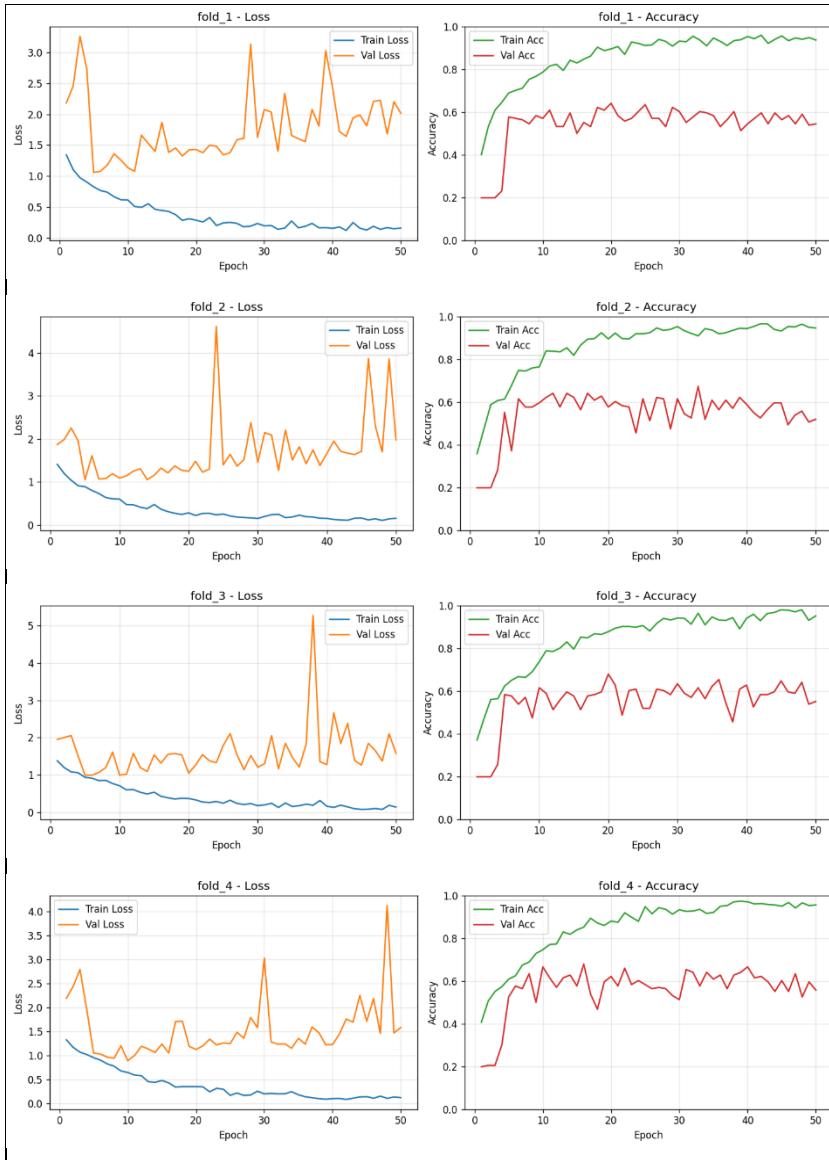


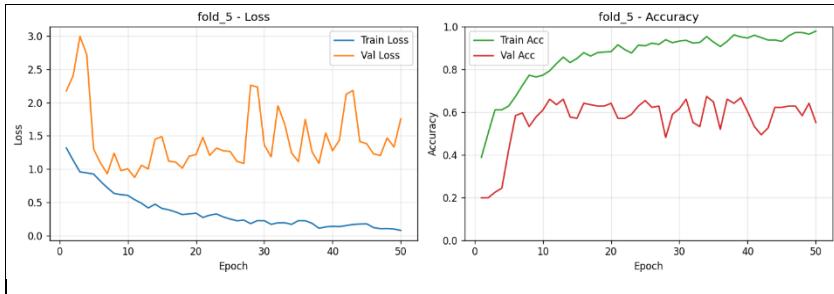




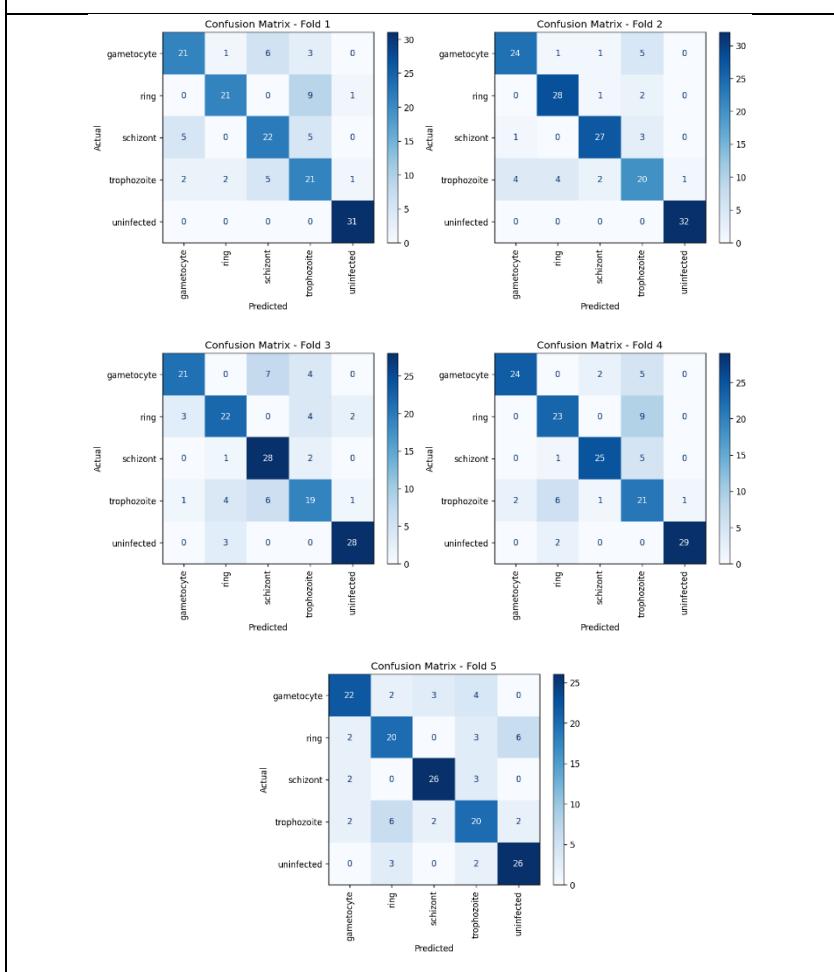
Variasi Hyperparameter 3: Opt RMSprop, LR 1e-4, Epoch 50, BS 32

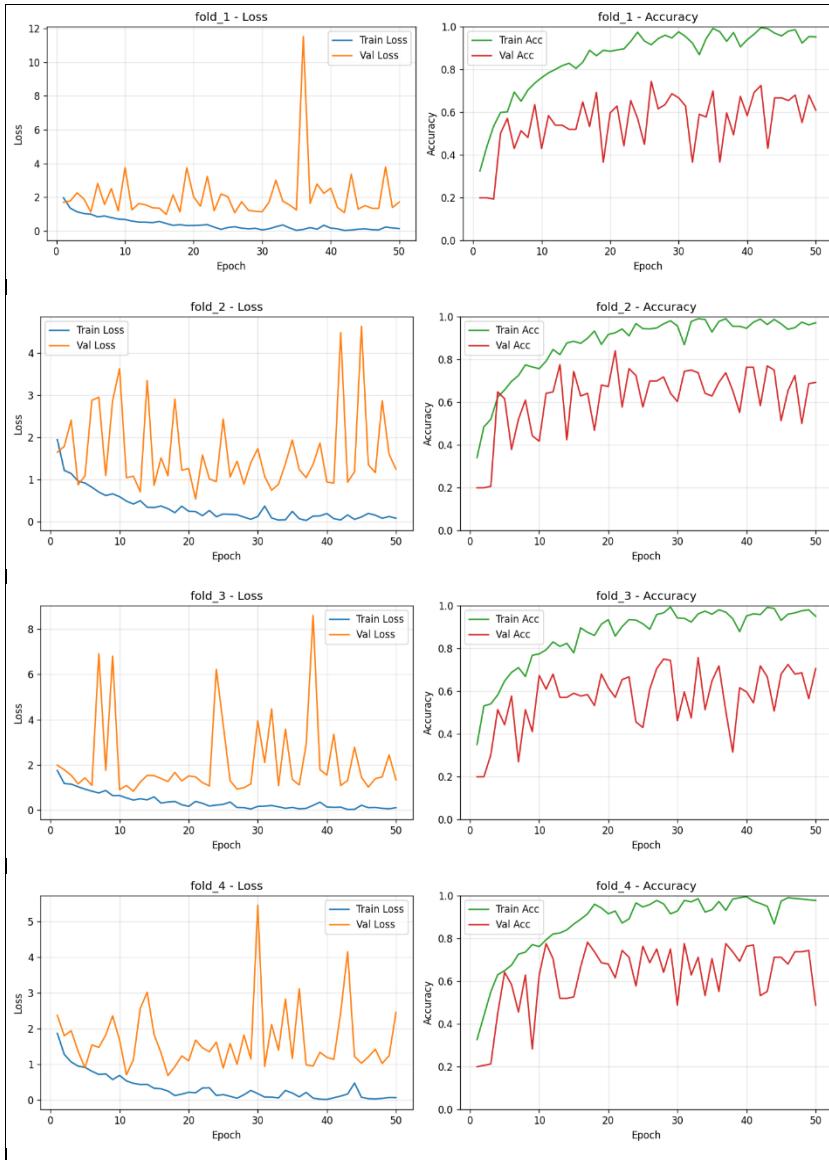


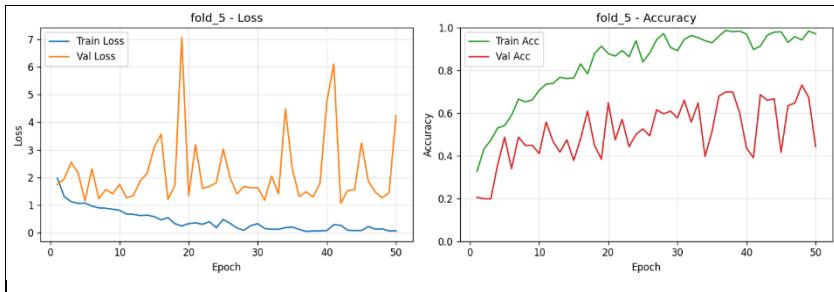




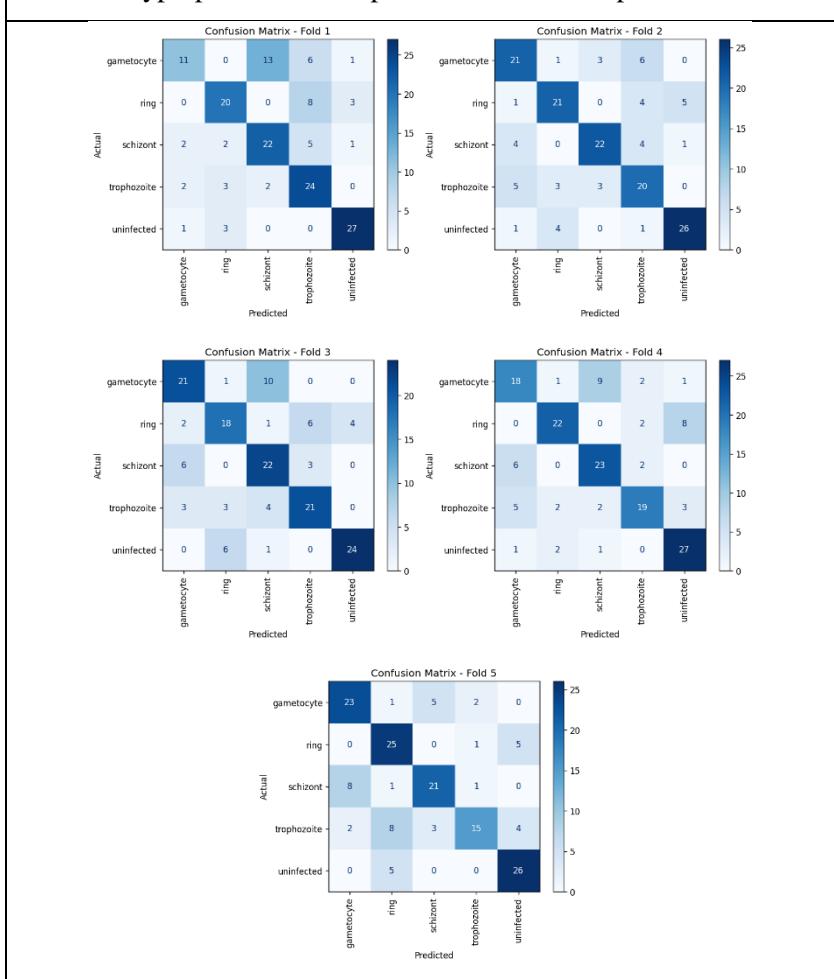
Variasi Hyperparameter 4: Opt RMSprop, LR 1e-3, Epoch 50, BS 32

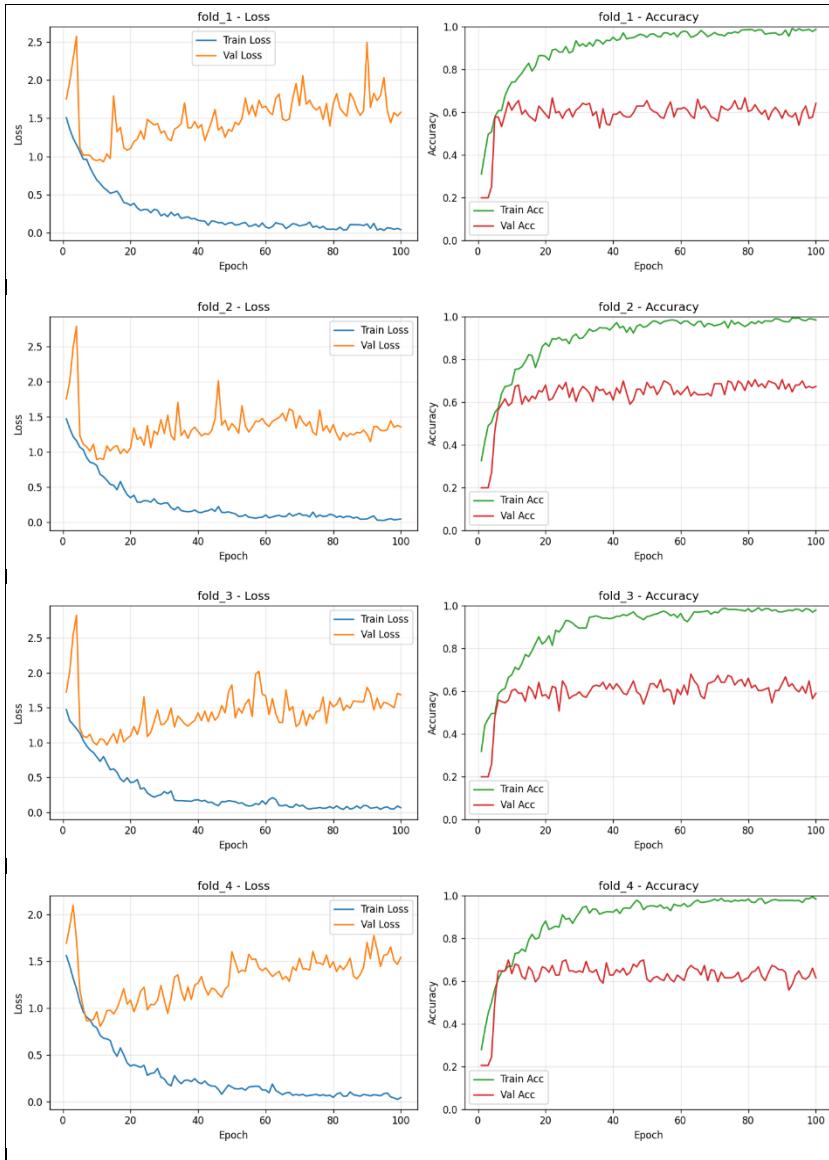


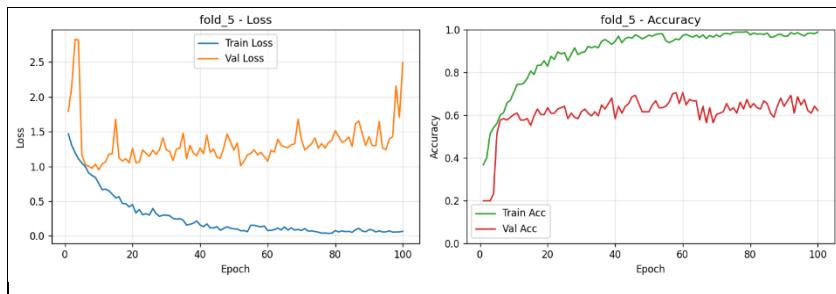




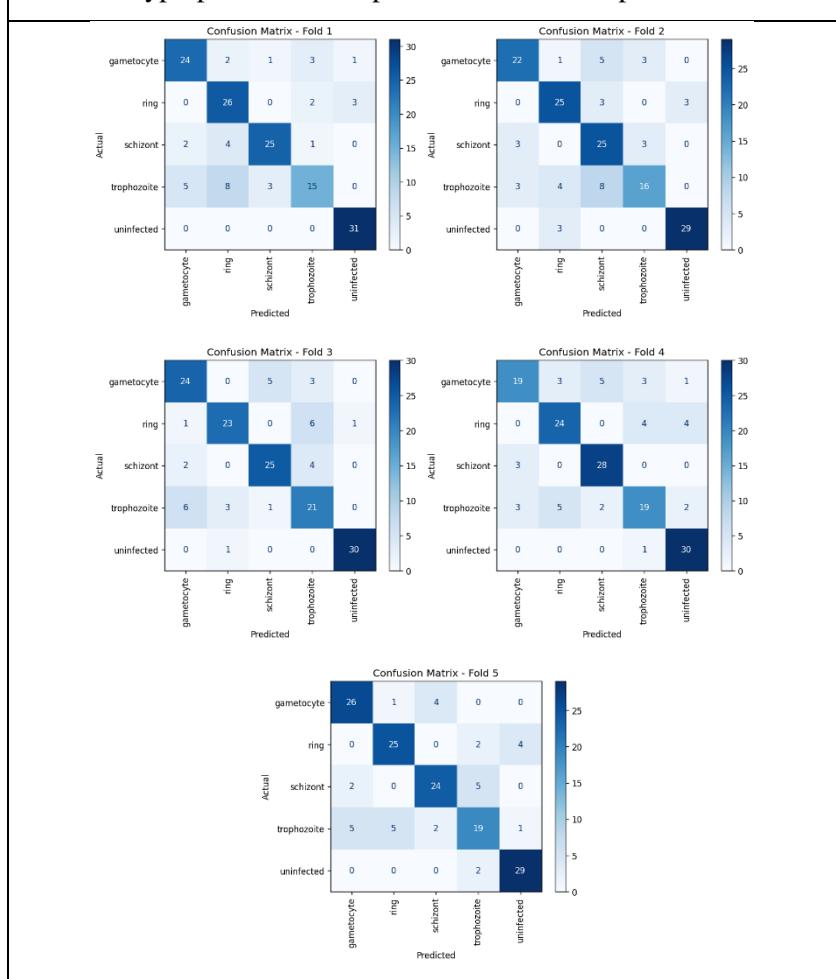
Variasi Hyperparameter 5: Opt Adam, LR 1e-4, Epoch 100, BS 32

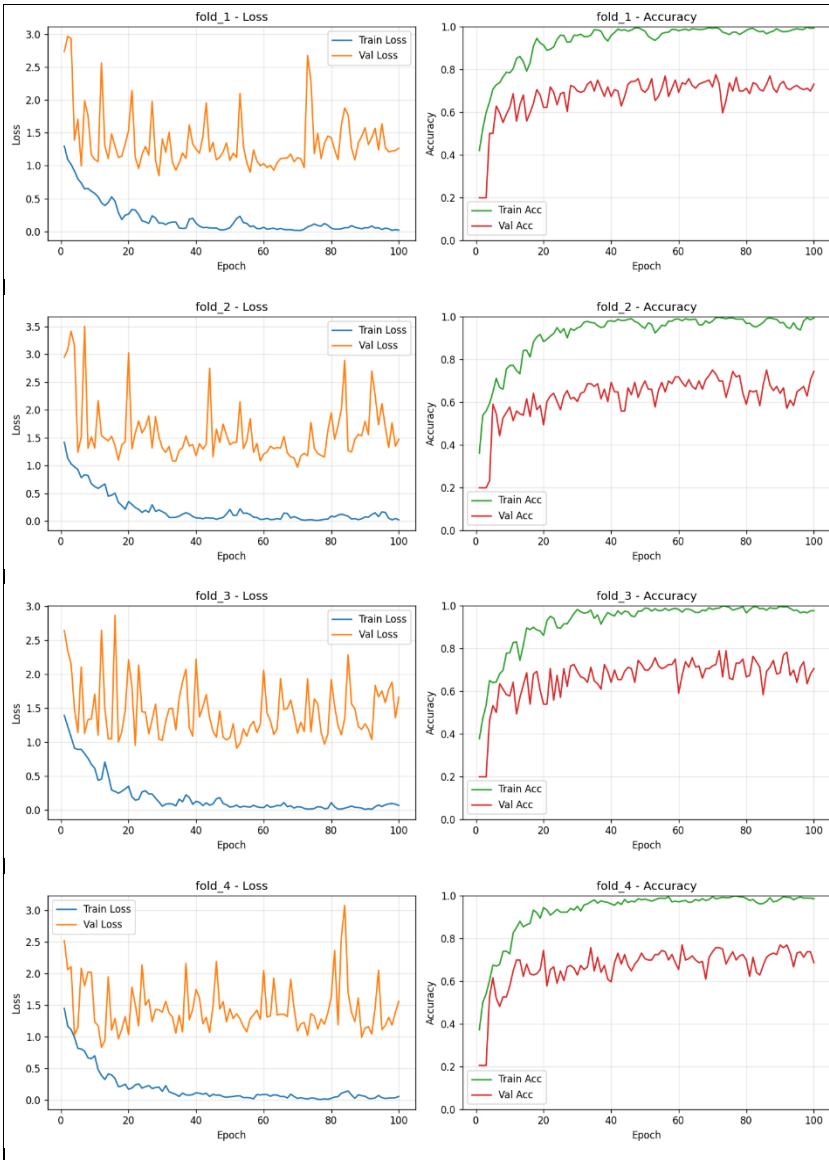


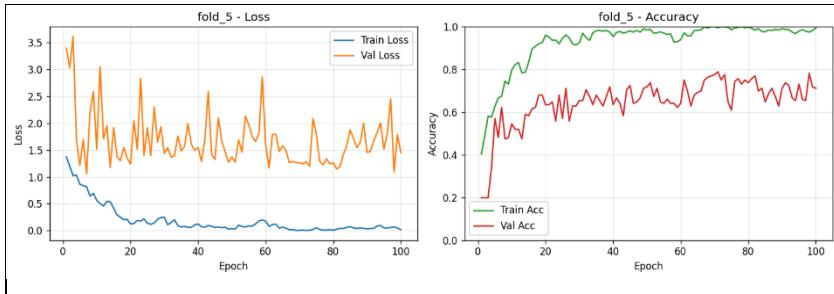




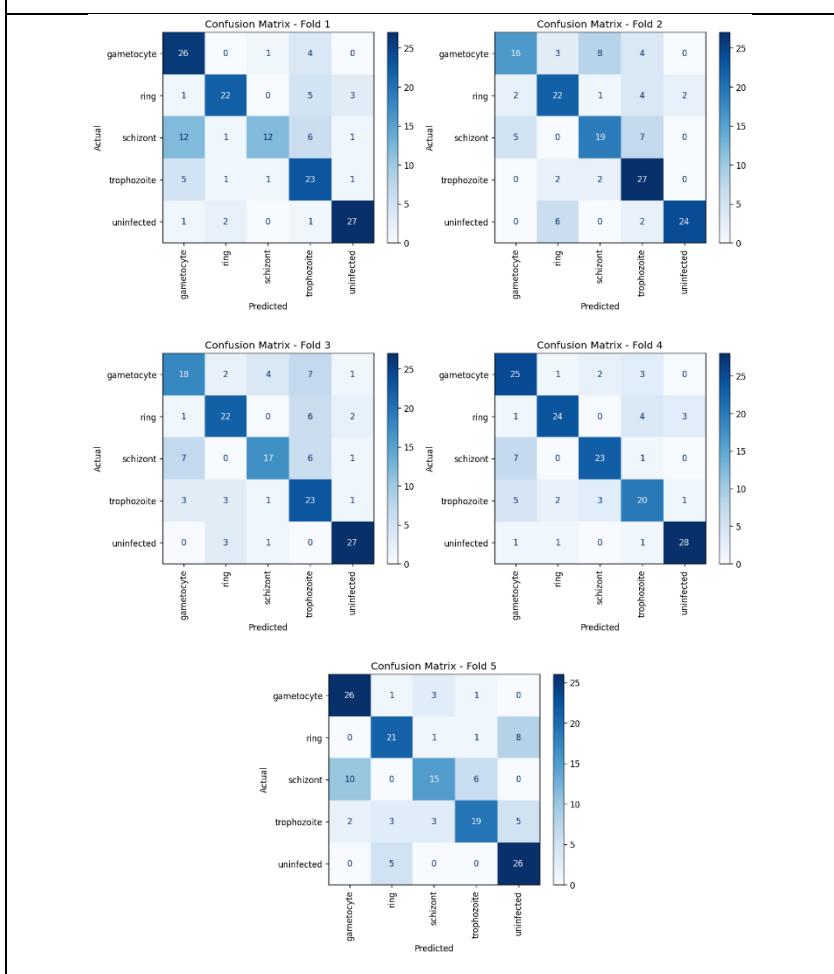
Variasi Hyperparameter 6: Opt Adam, LR 1e-3, Epoch 100, BS 32

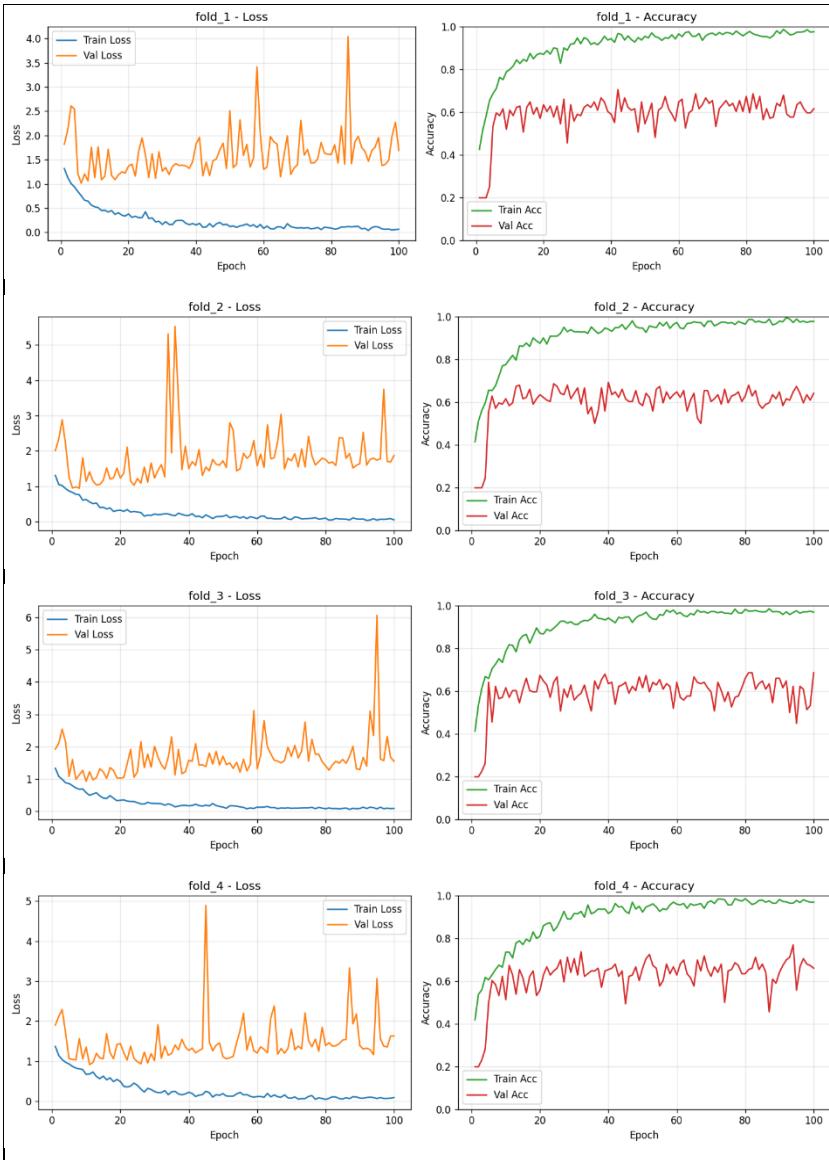


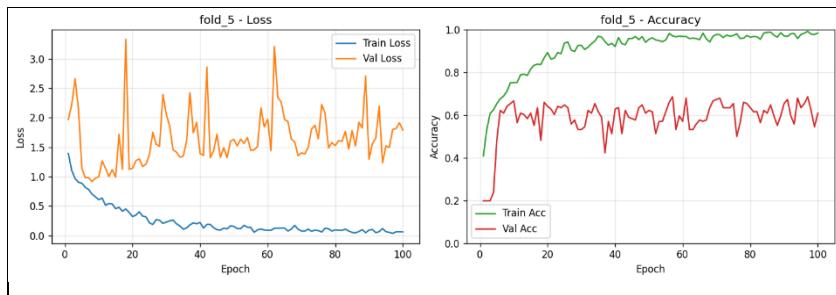




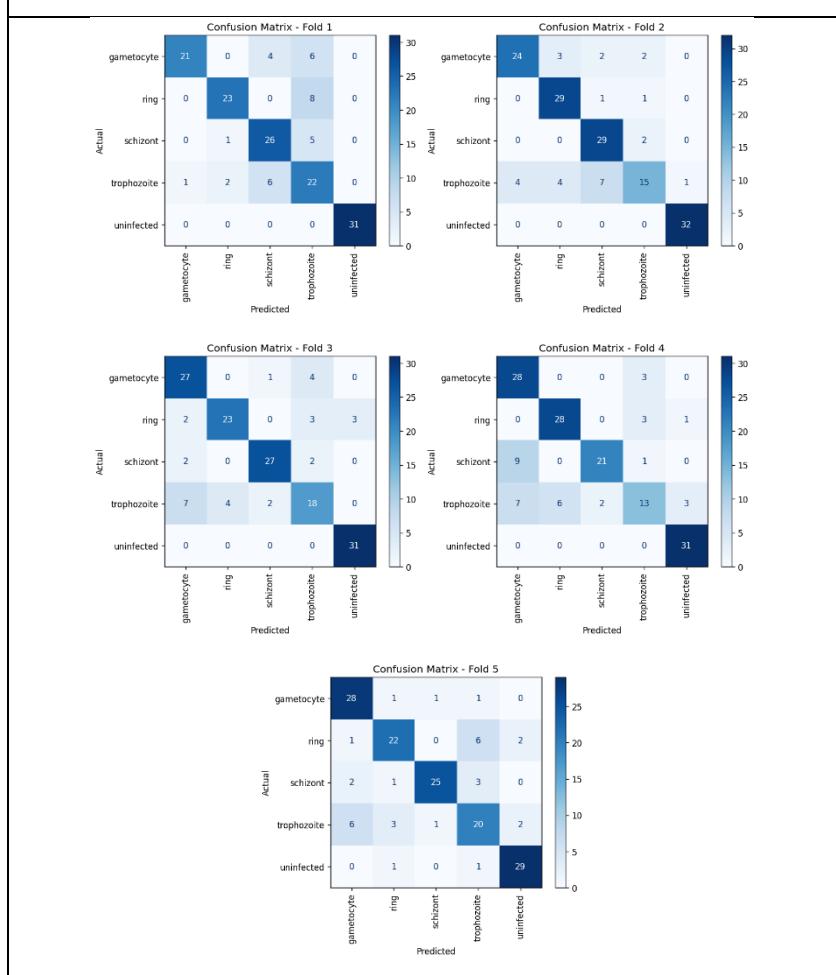
Variasi Hyperparameter 7: Opt RMSprop, LR 1e-4, Epoch 100, BS 32

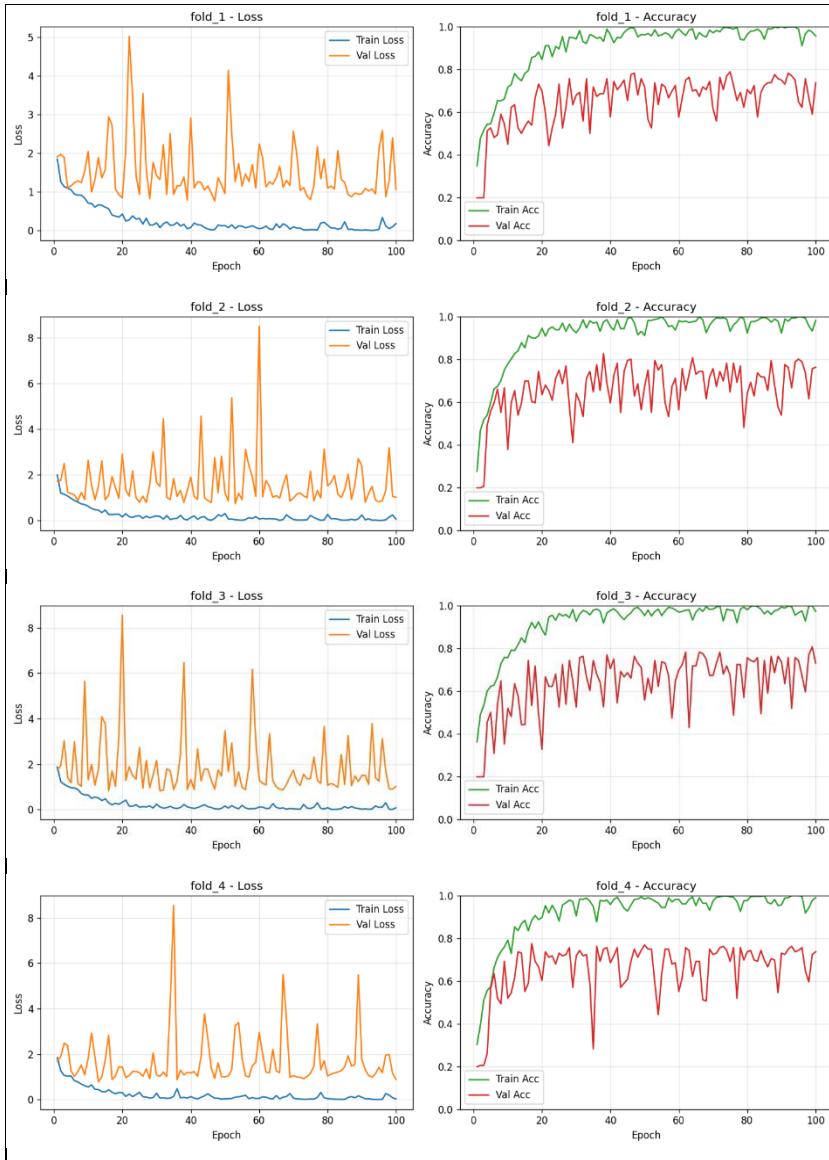


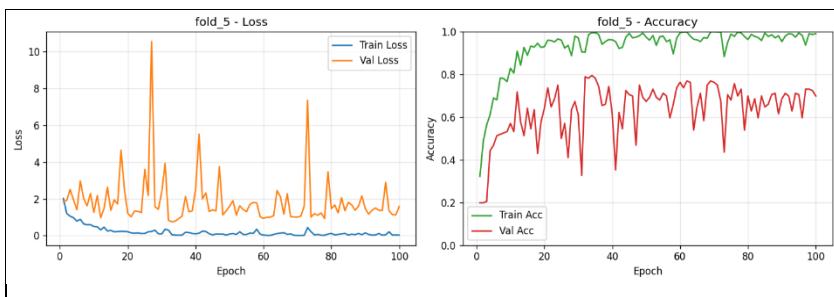




Variasi Hyperparameter 8: Opt RMSprop, LR 1e-3, Epoch 100, BS 32

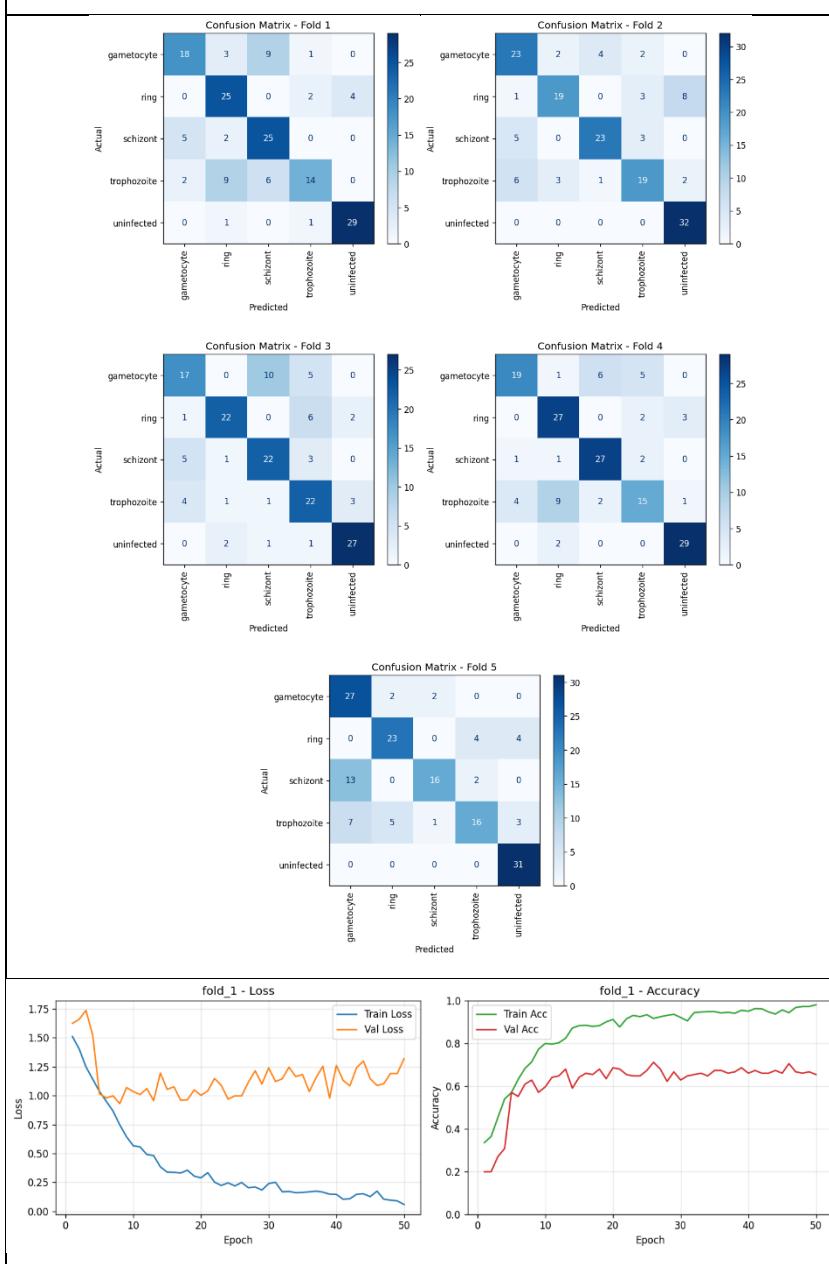


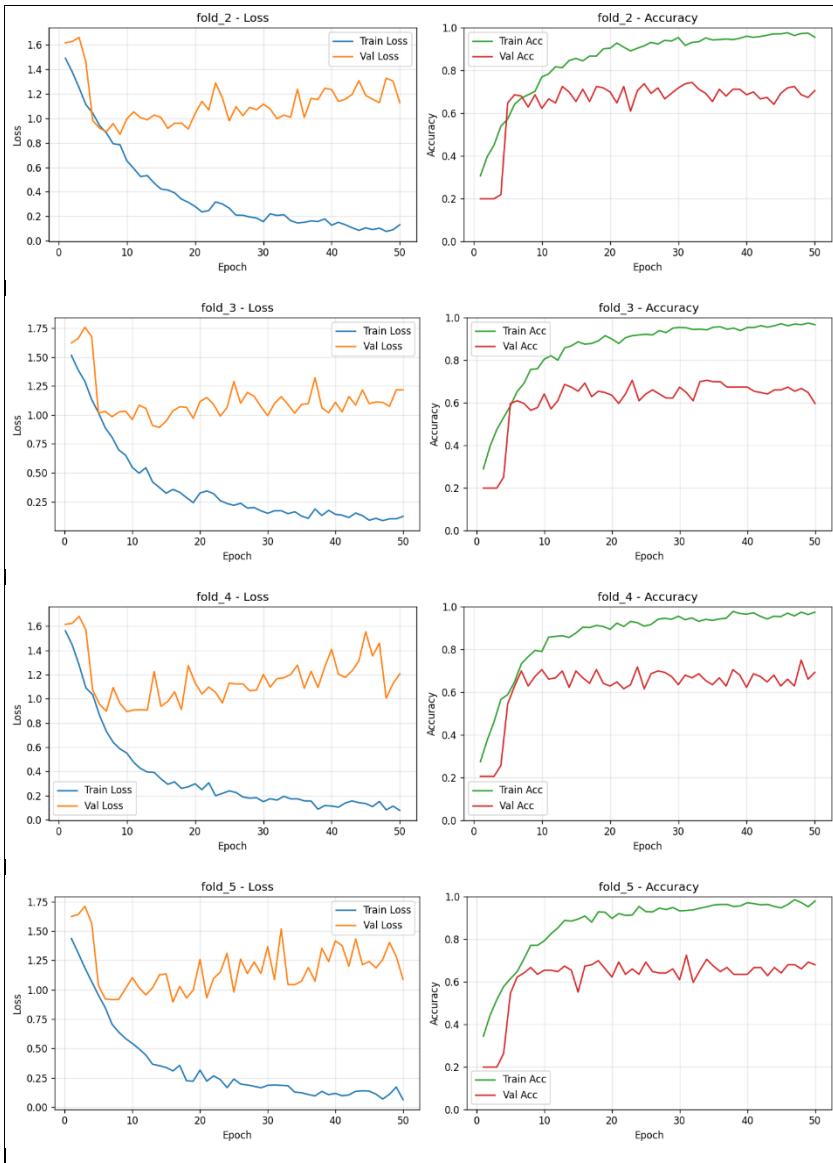




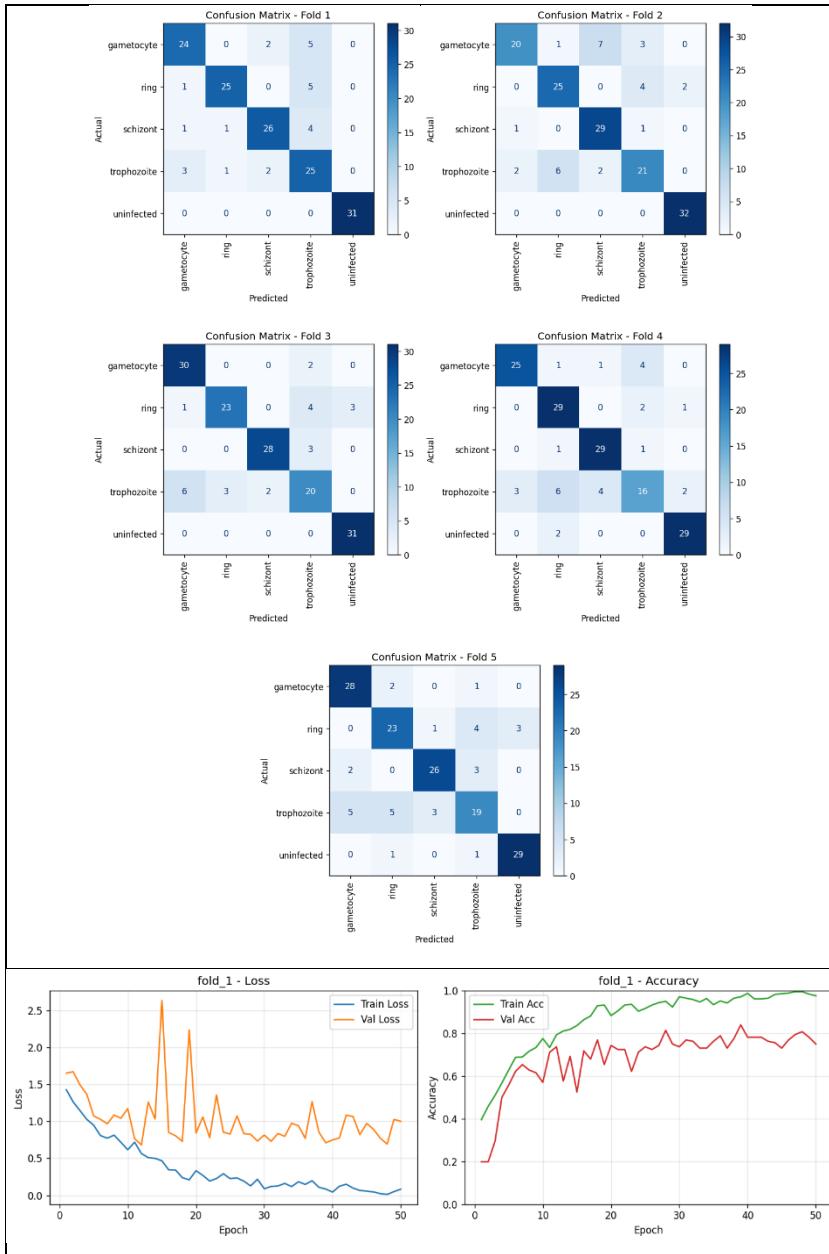
Lampiran II Hasil Training EfficientNetB0

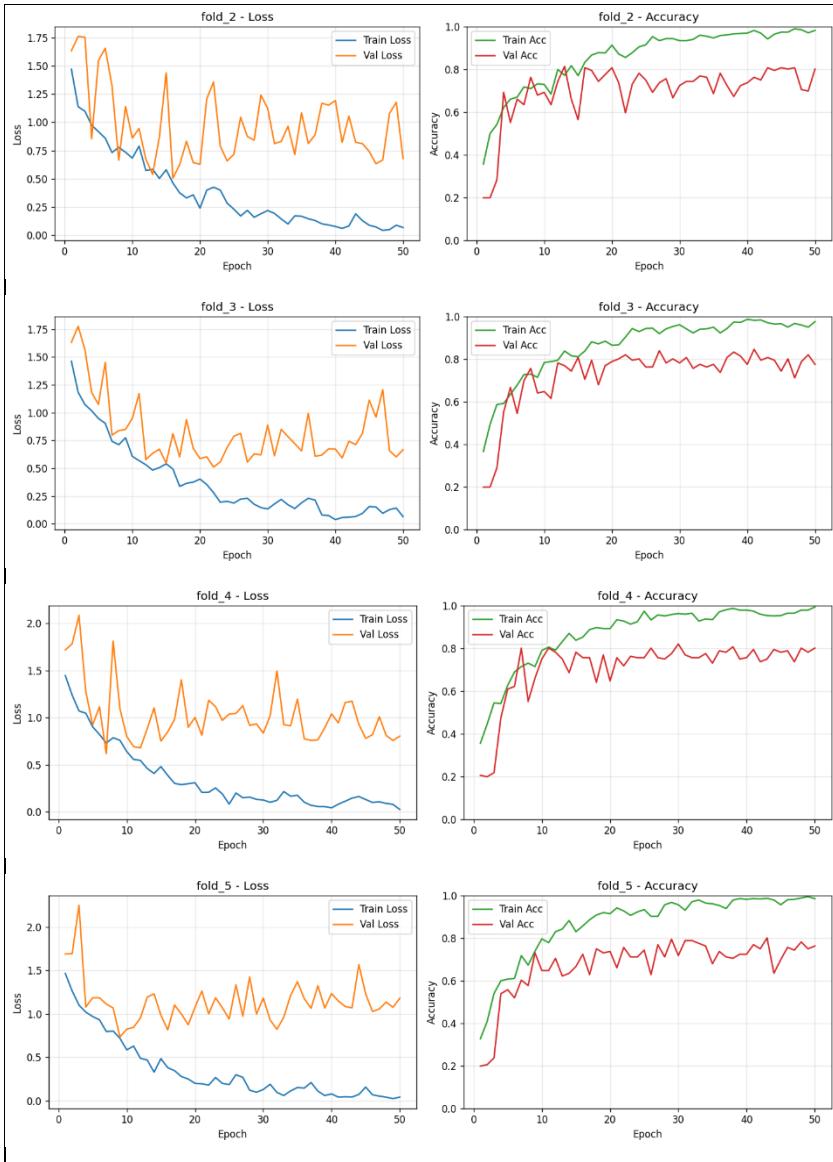
Variasi Hyperparameter 1: Opt Adam, LR 1e-4, Epoch 50, BS 32



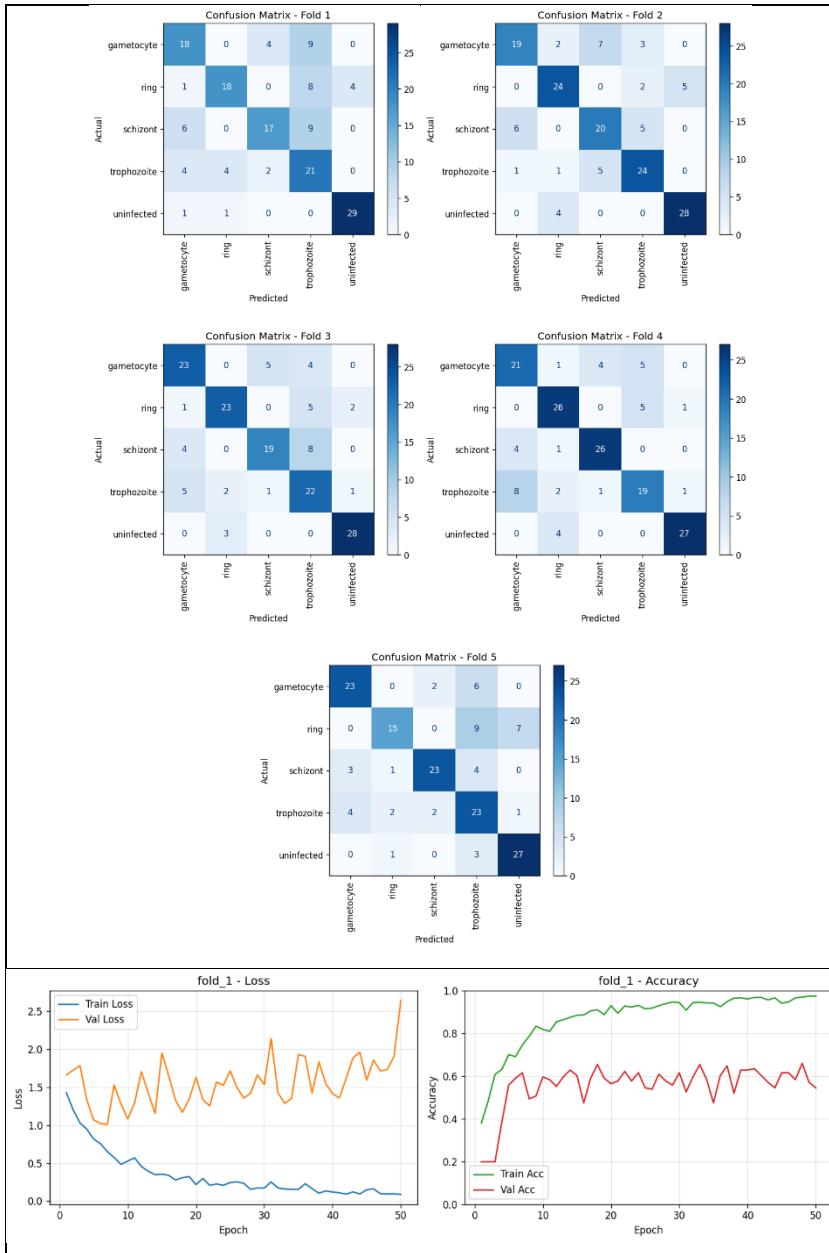


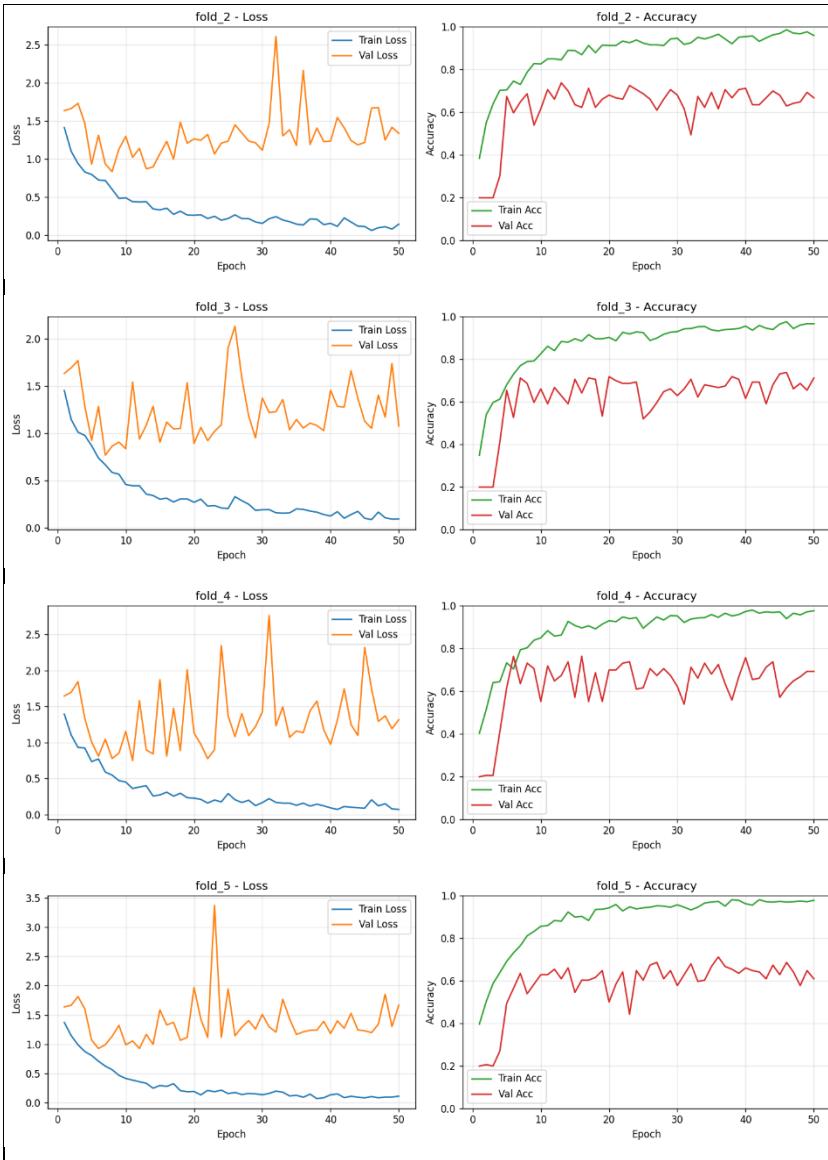
Variasi Hyperparameter 2: Opt Adam, LR 1e-3, Epoch 50, BS 32



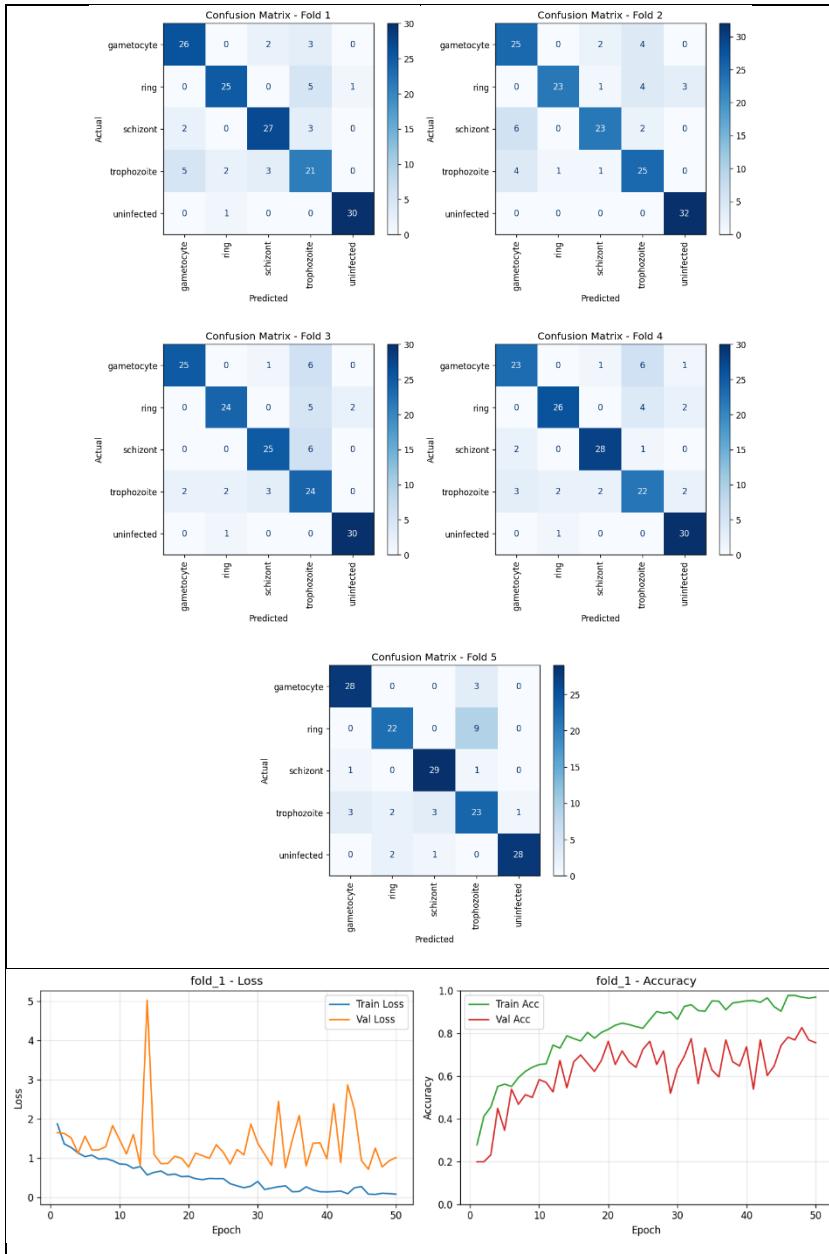


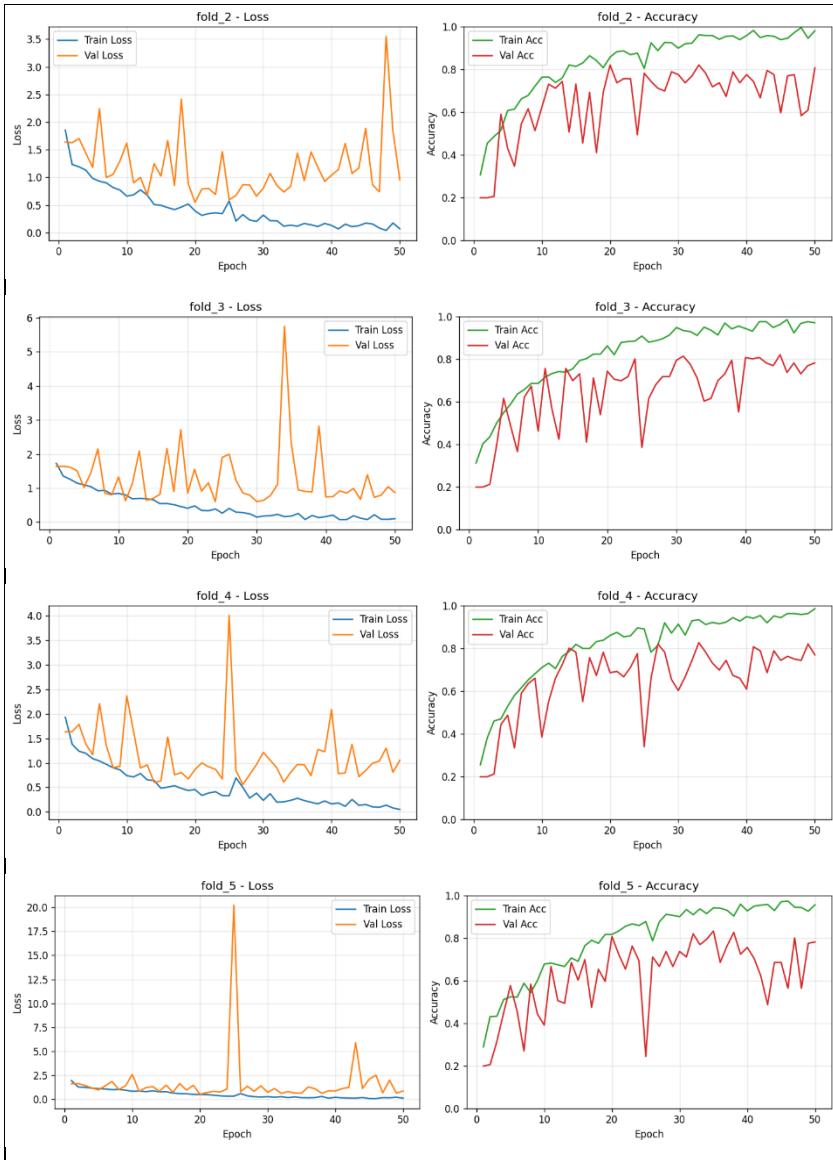
Variasi Hyperparameter 3: Opt RMSprop, LR 1e-4, Epoch 50, BS 32



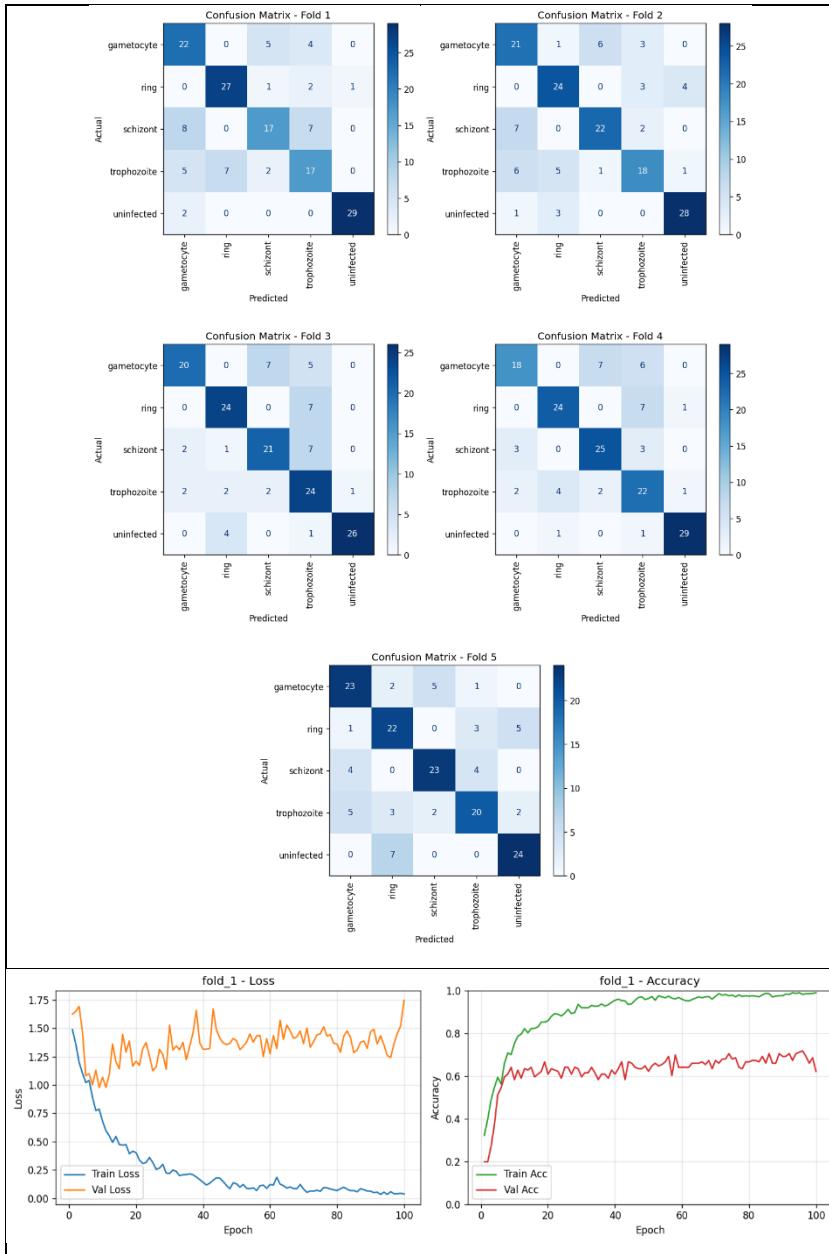


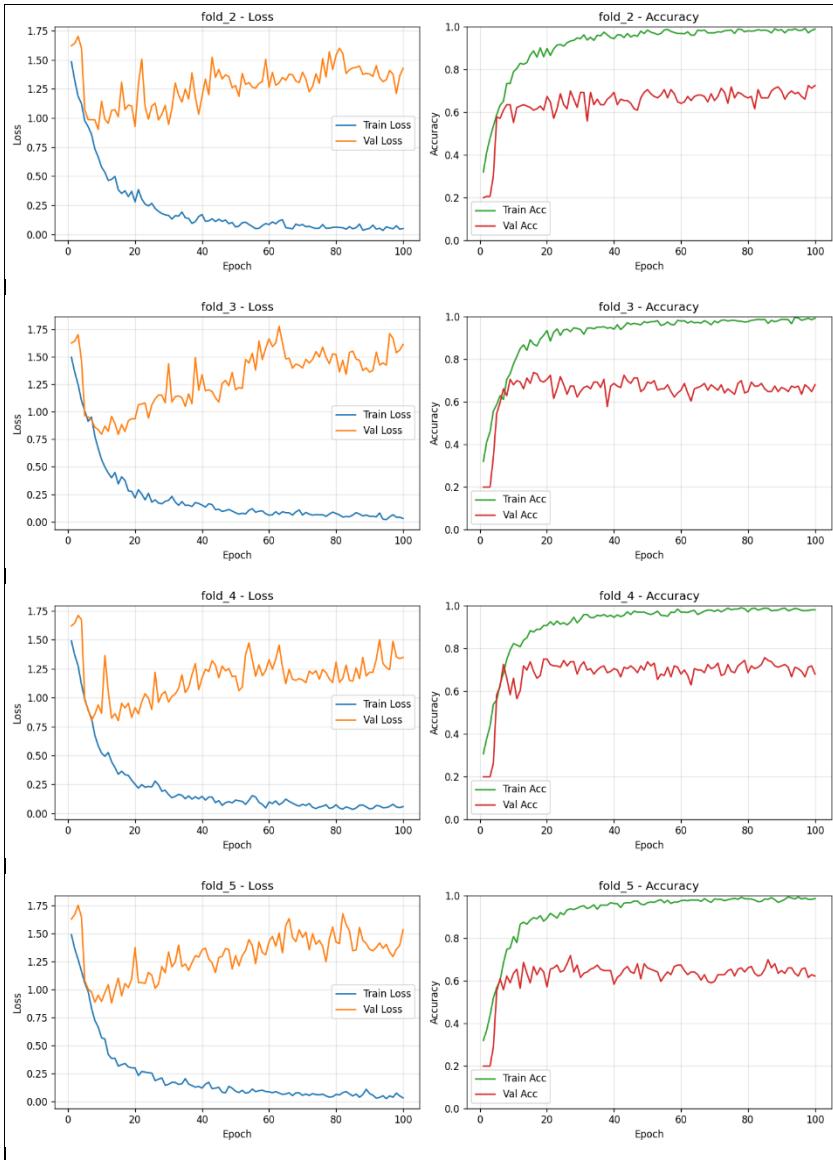
Variasi Hyperparameter 4: Opt RMSprop, LR 1e-3, Epoch 50, BS 32



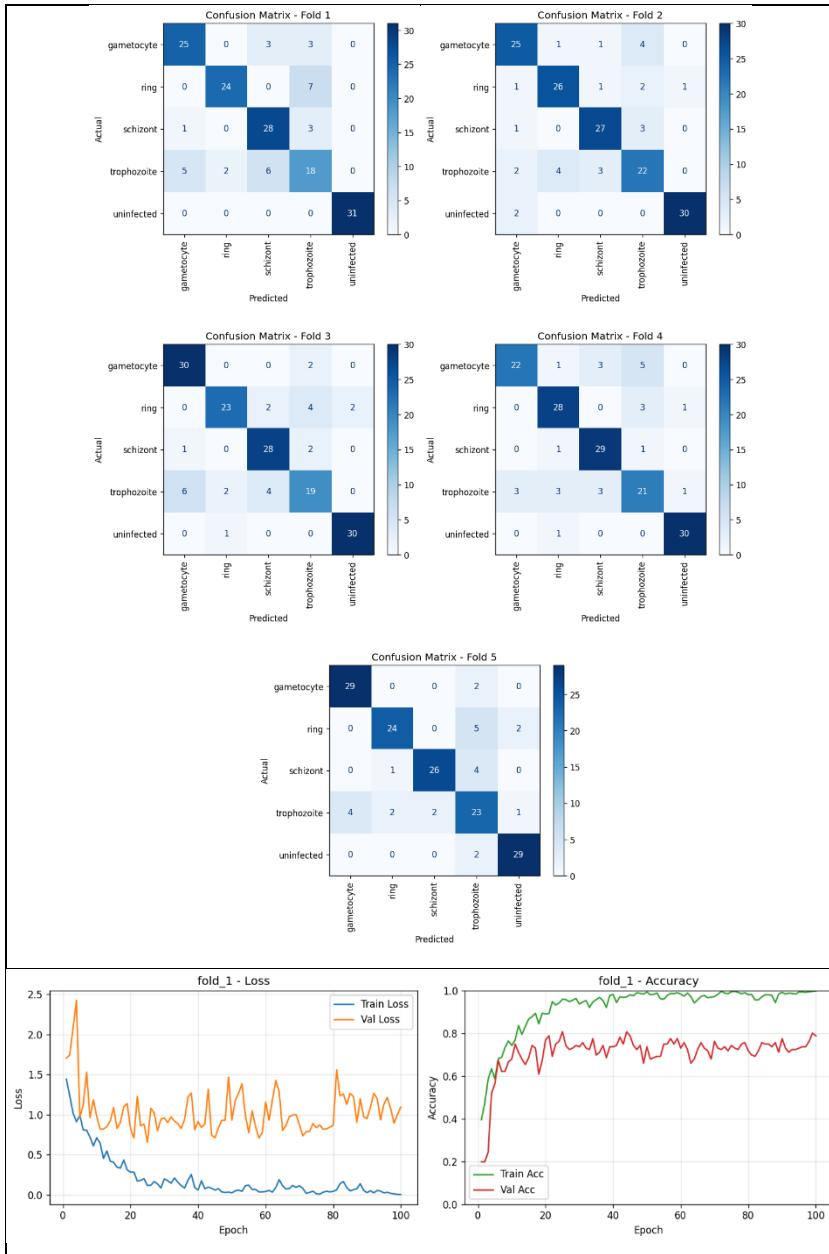


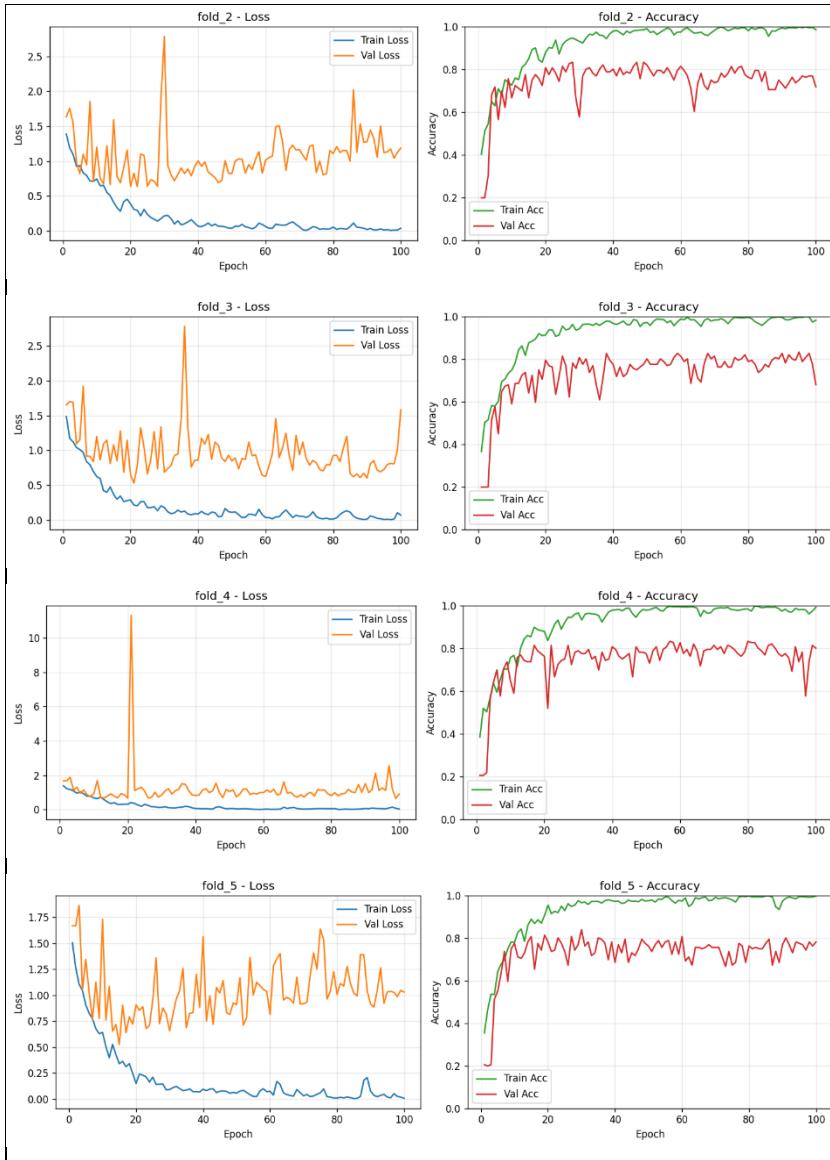
Variasi Hyperparameter 5: Opt Adam, LR 1e-4, Epoch 100, BS 32



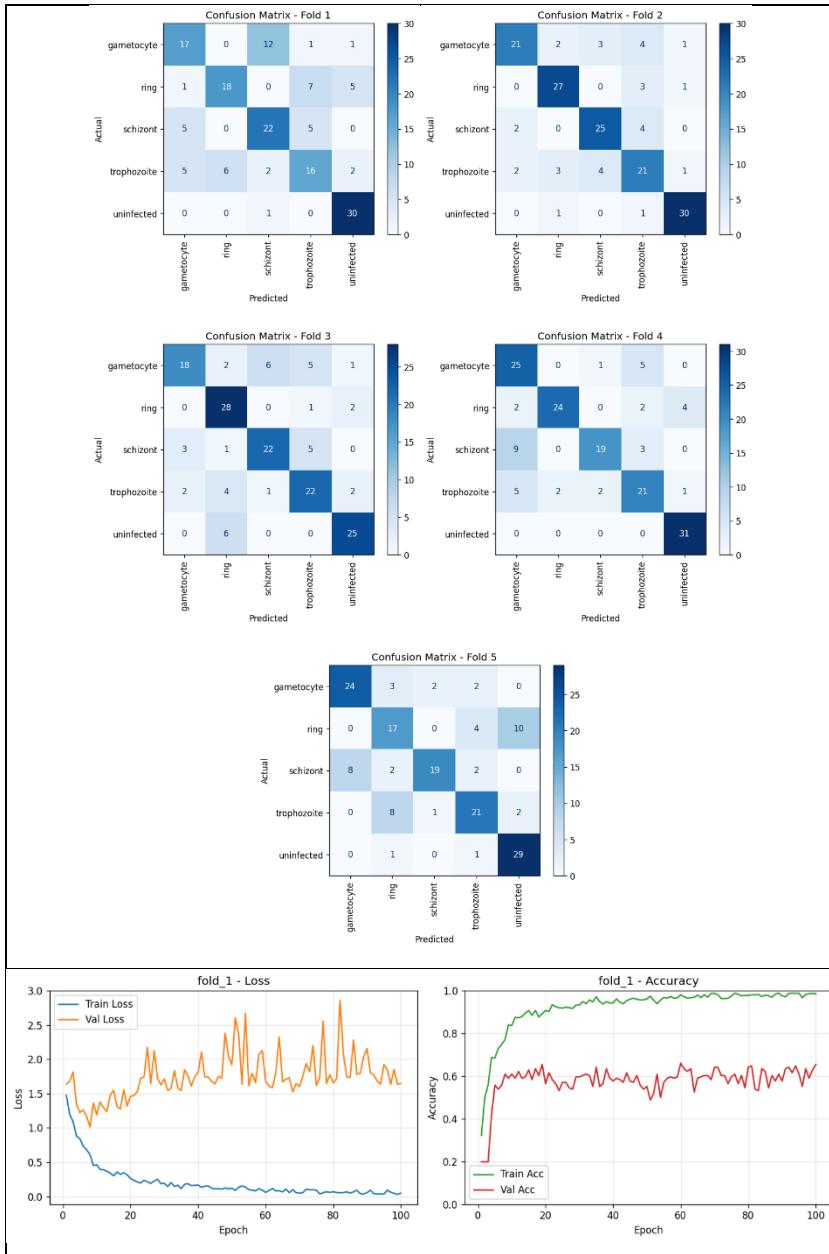


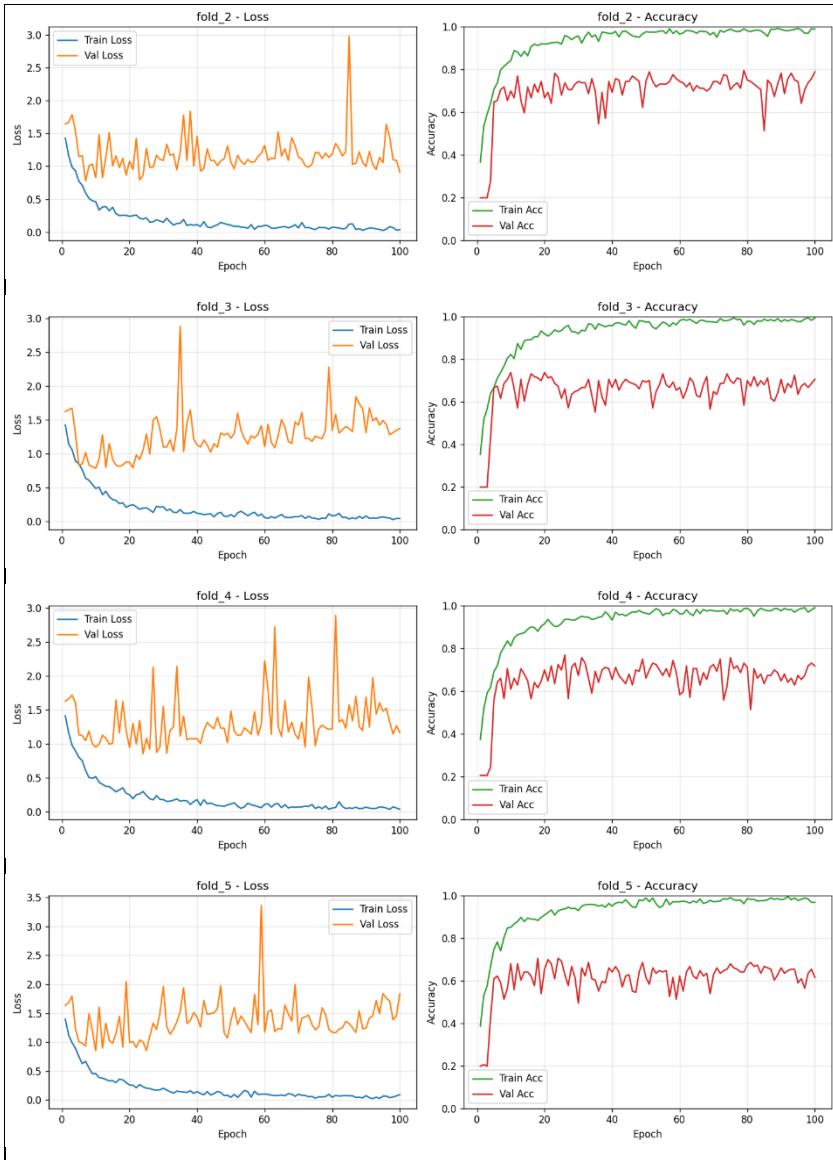
Variasi Hyperparameter 6: Opt Adam, LR 1e-3, Epoch 100, BS 32



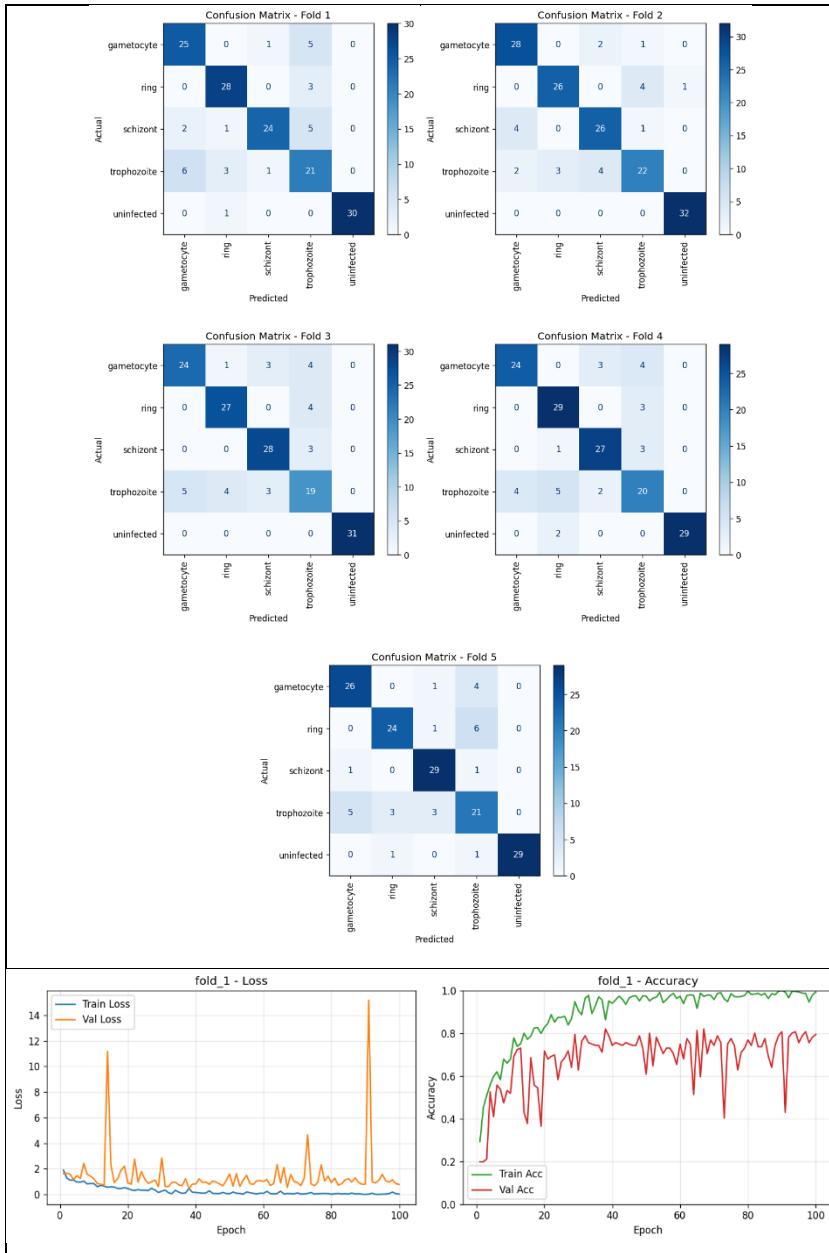


Variasi Hyperparameter 7: Opt RMSprop, LR 1e-4, Epoch 100, BS 32





Variasi Hyperparameter 8: Opt RMSprop, LR 1e-3, Epoch 100, BS 32



123

