

Prediksi Malaria Menggunakan Metode Pre-Trained Model Algoritma *EfficientNet-B0* dan *MobileNet-V2*

Asep Eryana, Sriyanto dan Firmansyah

Teknik Informatika, IIB Darmajaya, Bandar Lampung

Jl. ZA. Pagar Alam No.93 Rajabasa Bandar Lampung Lampung 35141

E-mail: aer.2121211015@darmajaya.ac.id, sriyanto@darmajaya.ac.id*, firmansyahyunialfi@darmajaya.ac.id

Abstrak

Malaria merupakan penyakit mematikan yang penularannya terjadi melalui gigitan nyamuk *Anopheles*. Berdasarkan laporan WHO tahun 2021 di Asia Tenggara angka kematian diakibatkan Malaria menurun sebesar 40% kecuali di Indonesia kurang dari 25%. Tingginya tingkat kematian akibat Malaria di Indonesia salah satunya disebabkan karena faktor kompetensi tenaga medis; keterlambatan penanganan, kesalahan diagnosis dan kesalahan terapi. Selain observasi langsung, pengujian menggunakan Mikroskop dan *Rapid Diagnostic Test* (RDT) dapat digunakan untuk mendeteksi Malaria. Namun kedua metode ini membutuhkan kompetensi yang baik untuk menyimpulkan hasil pengujian. Alternatif solusi yang diajukan pada penelitian ini yaitu mendeteksi Malaria menggunakan bantuan algoritma. Penelitian ini dilakukan untuk menguji prediksi Malaria menggunakan *pre-trained models* dari algoritma *EfficientNet-B0* dan *MobileNet-V2*. Hasil penelitian menyimpulkan bahwa *pre-trained models*, algoritma *EfficientNet-B0* dan *MobileNet-V2* dapat digunakan untuk prediksi Malaria. Algoritma terbaik pada penelitian ini yaitu *MobileNet-V2* dengan rata-rata skor akurasi sebesar 97,27%, rata-rata *F1-score* sebesar 97,29% dan skor *sanity check* sebesar 100%, skor kesalahan tipe 1 sebesar 1,74% dan kesalahan tipe 2 sebesar 0,89% dari 5512 sel darah yang diuji.

Kata kunci : Algoritma, *EfficientNet-B0*, Malaria, *MobileNet-V2*, *Pre-Trained*.

Pendahuluan

Malaria merupakan penyakit mematikan yang penularannya terjadi melalui gigitan Nyamuk Malaria (*Anopheles*) yang telah terinfeksi. Nyamuk ini membawa parasit *Plasmodium* yang ketika menggigit manusia, akan masuk ke dalam aliran darah[1]. Manusia yang telah digigit nyamuk pembawa parasit *Plasmodium* akan menyebabkan parasit masuk ke dalam tubuh manusia, kemudian menetap di organ hati sebelum menyerang sel darah merah. Gejala muncul dalam tiga tahap selama 6 sampai 12 jam, seperti menggigil, demam, sakit kepala, dan berkeringat sampai dengan suhu tubuh kembali normal[2], [3].

Laporan WHO tahun 2021 menyebutkan bahwa pada tahun 2020 terdapat 627.000 orang meninggal akibat terjangkit Malaria. Di Asia Tenggara meskipun trend menunjukkan penurunan angka kematian yang diakibatkan Malaria rata-rata menurun sebesar 40%, namun angka penurunan kematian akibat Malaria di Indonesia kurang dari 25%[4]. Tingginya tingkat kematian akibat Malaria di Indonesia disebabkan oleh banyak faktor. Keterlambatan penanganan medis pasien dan salah diagnosis

dapat menyebabkan pasien terinfeksi Malaria berat memungkinkan mengalami komplikasi[5].

Penelitian yang dilakukan oleh Oktavianus dkk, menemukan bahwa kesalahan medis pada penatalaksanaan Malaria sebesar 1,87 kali per pasien. Dalam kajian ini ditemukan 98 kesalahan diagnosis yaitu; 16 kejadian *error of omission* (17,39%) dan 82 kejadian *error of commission* (89,13%). Selain kesalahan diagnosis, juga ditemukan 92 kejadian kesalahan terapi, terdiri dari 19 kejadian *error of omission* (20,65%) dan 73 kejadian *error of commission* (79,35%) [6]. Kesalahan diagnosis pada pasien terjangkit Malaria menunjukkan adanya inkonsistensi kompetensi atau pengetahuan tenaga medis terkait penyakit Malaria.

Selain observasi langsung, pengujian menggunakan mikroskop dan *Rapid Diagnostic Test* (RDT) dapat digunakan untuk mendeteksi Malaria. Namun metode RDT ini memiliki kelemahan tidak dapat mendeteksi kepadatan parasit yang rendah, RDT hanya bisa membaca kehadiran parasit dengan rentang 40 -100 parasit per mikroliter darah. Sedangkan metode mikroskop hanya dapat membaca kehadiran parasit dengan rentang 5-10 para-

sit per microliter darah [7]. Dengan demikian dapat dipahami bahwa kekurangan dari metode observasi langsung, mikroskop, dan RDT yaitu membutuhkan kompetensi atau pengetahuan yang baik untuk dapat menyimpulkan apakah pasien terinfeksi Malaria atau tidak. Sehingga hal ini masih dimungkinkan terjadinya kesalahan diagnosis, baik itu terkait dengan faktor pengetahuan atau kompetensi tenaga medis maupun kesalahan dalam proses pengambilan kesimpulan atas diagnosis yang telah dilakukan.

Sebagai salah satu alternatif solusi atas permasalahan yang telah diuraikan tersebut, beberapa penelitian di bidang *deep learning* telah dilakukan. Menggunakan algoritma untuk membantu mengambil keputusan apakah pasien terinfeksi Malaria atau tidak. Dengan menggunakan metode klasifikasi terhadap sampel sel darah, dapat dilakukan prediksi terhadap sel darah baru apakah terinfeksi malaraiia atau tidak. Metode ini tidak memerlukan kompetensi tenaga medis untuk menentukan kesimpulan atau keputusan akhir, karena proses sepenuhnya dilakukan oleh algoritma komputer.

Penelitian dilakukan Sasongko, melakukan pengujian arsitektur model *CNN*; *ResNet-101*, *AlexNet*, dan *VGG-16-19* untuk mendeteksi parasit penyakit Malaria. Hasil penelitian menginformasikan bahwa arsitektur model *ResNet-101* memiliki akurasi yang jauh lebih baik dibandingkan model *AlexNet* dan *VGG-16-19* [8]. Penelitian lain dilakukan oleh Bibin dkk, menguji metode *deep belief network* (DBN) untuk mengidentifikasi parasit Malaria. Pengujian dilaksanakan dengan menggunakan 4100 sampel gambar dan menghasilkan informasi bahwa metode DBN memiliki kemampuan yang lebih baik daripada metode lainnya dengan *F-score* 89,66%, sensitivitas 97,60%, dan akurasi sebesar 95,92% [9].

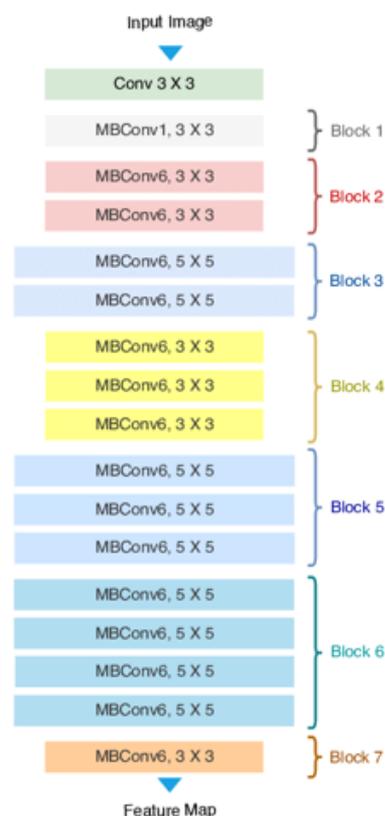
Dalam bidang *deep learning*, sejumlah besar arsitektur telah diperkenalkan yang memungkinkan untuk dapat diterapkan pada banyak kasus, di antaranya adalah *Convolutional Neural Networks* (*CNN*). Pemikiran tentang *CNN* terinspirasi dari penelitian yang dilakukan pada korteks visual mamalia, yaitu bagaimana mereka memandang dunia menggunakan arsitektur neuron di otak [10]. *CNN* juga merupakan salah satu algoritma yang dikembangkan untuk tujuan klasifikasi dan pengelolaan gambar. Terdapat tiga layer utama pada *CNN* yang digunakan untuk memproses dan mengekstrak fitur dari gambar, utama yaitu *convolutional layer*, *pooling layer* dan *Fully Connected* (FC)[11][12][13].

Deep learning dan *Convolutional Neural Networks* merupakan seperangkat teknik yang sangat powerful dan memberikan solusi terbaik untuk menyelesaikan masalah dalam pengenalan dan pemrosesan gambar. Meskipun demikian, kelemahan terbesar dari model algoritma ini yaitu memerlukan sumber daya (GPU dan Memori) yang besar[14]. *Transfer Learning*, adalah solusi yang tidak hanya

mengatasi masalah sumber daya yang besar tetapi juga dapat mempersingkat waktu yang diperlukan dalam proses analisis.

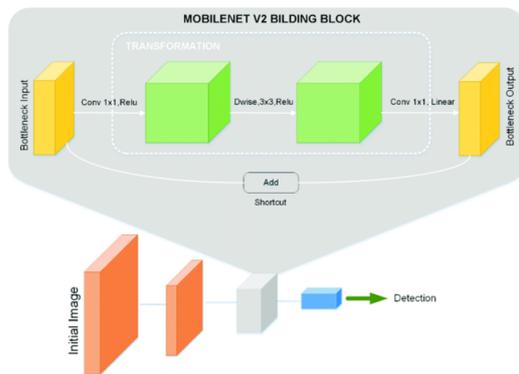
Transfer Learning merupakan teknik dimana menggunakan model yang sudah dilatih (*pre-trained models*) untuk mengklasifikasikan dataset baru. Sehingga tidak perlu untuk melakukan training data dari awal dan hanya perlu melakukan penyesuaian pada bagian akhir dari model[15]. *Pre-trained models* merupakan model yang telah dilatih pada sekumpulan data yang besar untuk memecahkan masalah yang serupa dengan yang ingin kita selesaikan [14]. Terdapat 20 *pre-trained models* pada *library pytorch* [15], dan pada penelitian ini digunakan dua *pre-trained models* yaitu; *EfficientNet-B0* dan *MobileNet-V2*. Pemilihan *pre-trained models* tersebut didasarkan pada ukuran parameter kecil (sebesar 3,5 M) dan akurasi cukup besar (93,53 dan 90,82) [16].

EfficientNet merupakan salah satu *pre-trained models* *CNN* yang diusulkan oleh Mingxing Tan dan Quoc V. Le dari Google Research pada tahun 2019[17], [18]. *EfficientNet* menggunakan teknik yang disebut koefisien majemuk untuk memperbesar model dengan cara yang sederhana namun efektif[19]. Model *EfficientNet* saat ini memiliki 8 model, yaitu *EfficientNet-B0* sampai *EfficientNet-B7* yang memiliki jumlah parameter yang semakin banyak dari 5,3 juta sampai 66 juta parameter diikuti dengan nilai akurasi yang semakin tinggi, lihat Gambar 1.



Gambar 1: Arsitektur EfficientNet-B0[20]

MobileNets, merupakan salah satu arsitektur *convolutional neural network* (CNN) yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* berlebih[21]. *MobileNet* telah dilatih pada kumpulan data *ImageNet* untuk keperluan deteksi dan klasifikasi dengan menggunakan *library TensorFlow*[22]. Perbedaan mendasar antara arsitektur *MobileNet* dan arsitektur CNN pada umumnya adalah penggunaan layer convolution dengan ketebalan filter yang sesuai dengan ketebalan dari input image. Gambar 2, *MobileNet* membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution*[21], [23].



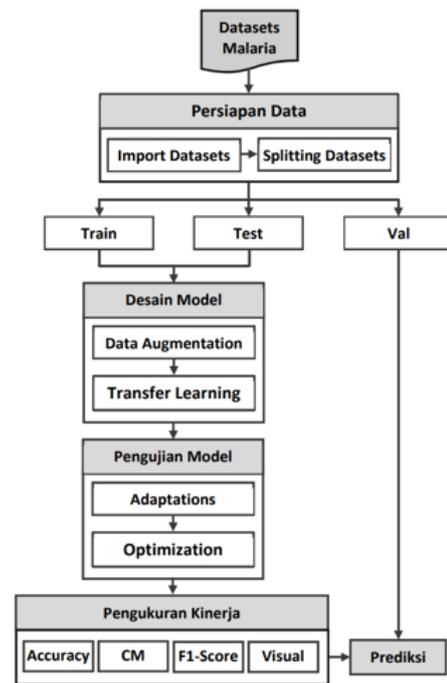
Gambar 2: Arsitektur MobileNet V2[24]

Metode Penelitian

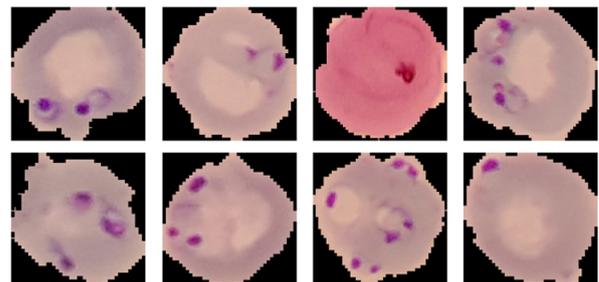
Metode yang digunakan pada penelitian ini merupakan penelitian eksperimen. Pada percobaan ini peneliti akan melakukan pengujian pada pre-trained model *EfficientNet-B0* dan *MobileNet-V2* dalam memprediksi citra dari sel darah. Proses pengujian dilakukan menggunakan aplikasi *google colab* dengan pengaturan runtime menggunakan GPU. Pengujian dilakukan sesuai dengan dokumentasi yang terdapat pada *library pytorch*[25]. Gambar 3 merupakan diagram alir prosedur pengujian yang dilakukan pada penelitian ini.

Dataset

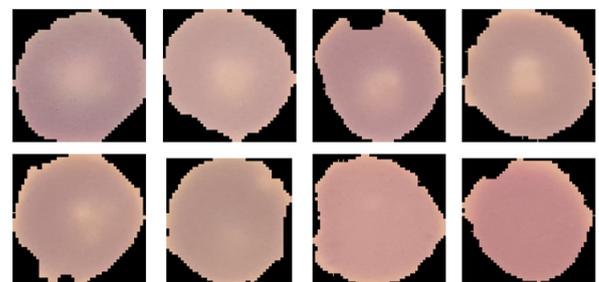
Dataset yang digunakan pada penelitian ini merupakan dataset publik berupa citra sel darah yang diunduh dari halaman <https://www.kaggle.com/datasets/iarunava/cell-images-for-detecting-Malaria>. Dataset yang digunakan sebanyak 27.558 file gambar, terdiri dari kelompok terinfeksi (*infected*) sebanyak 13.779 file gambar dan kelompok tidak terinfeksi (*uninfected*) juga sebanyak 13.779 file gambar, contoh ada di Gambar 4 dan 5.



Gambar 3: Prosedur pengujian algoritma



Gambar 4: Sel darah terinfeksi Malaria



Gambar 5: Sel darah tidak terinfeksi Malaria

Persiapan Data

Tahap persiapan data merupakan langkah pengaturan data sehingga siap digunakan pada proses

pengujian, pada tahap ini penulis membagi menjadi dua proses yaitu; *import datasets* dan *splitting datasets*.

1. Proses Import datasets

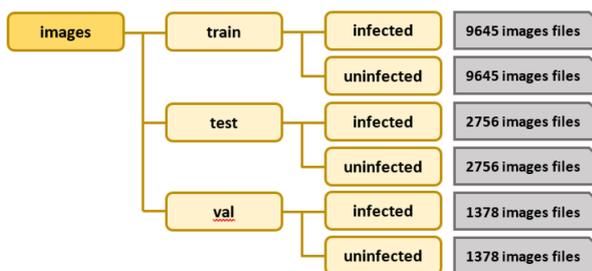
Pada tahap ini dataset berupa file gambar sel darah dibagi menjadi dua kelompok (*infected, uninfected*). Selanjutnya data di kompres dalam bentuk file zip dan diupload ke *google drive*. Data dalam bentuk zip file diimport dan diekstrak ke *Google Colaboratory* untuk dilakukan proses lebih lanjut, lihat Gambar 6.



Gambar 6: Struktur folder dataset

2. Proses Splitting Datasets

Pada tahap ini penulis melakukan pembagian dataset menjadi tiga kategori; train (data *training*), test (data *testing*), dan val (data *validation*). Pembagian dataset dilakukan menggunakan *feature splitfolders* dengan proporsi 70% untuk data train, 20% untuk data test, dan 10% untuk data val. Hasil dari proses *splitting datasets* seperti terlihat pada Gambar 7.



Gambar 7: Pembagian data pada dataset

Desain Model

Desain model merupakan proses dimana algoritma yang digunakan ditentukan sebagai dasar pemilihan parameter pada pengujian. Pada tahap ini penulis membagi menjadi dua bagian proses yang dilakukan yaitu *data augmentation* dan *transfer learning*.

1. Proses Data Augmentation

Data augmentation merupakan proses pengaturan parameter yang akan digunakan pada

proses transformasi gambar sesuai dengan algoritma yang digunakan. Terdapat enam parameter pengujian yang digunakan pada tahap ini yaitu; *batch size, crop_size, Center-Crop, resize_size, normalized mean, dan normalized std*. Pengaturan data augmentation dilakukan pada data train dan data *test* dengan mengacu pada data *augmentation* yang terdapat pada *library Pytorch*.

2. Proses Transfer Learning

Transfer learning merupakan proses penerapan model yang telah dilatih (*pre-trained models*) pada proses pengujian sesuai dengan data *augmentation* yang telah dilakukan. Proses *transfer learning* ini dibagi menjadi empat sub proses yaitu; *create models, freeze layer, replacing sequential, save model*.

Create model; pada tahap ini model algoritma *EfficientNet-B0* dan *MobileNet-V2* diimport dari *library torchvision* dan diinput kedalam variabel *mnet* dengan menerapkan pendekatan *pre-trained models*.

Freeze layer; tahap *freeze layer* merupakan langkah yang harus dilakukan pada saat menggunakan pendekatan *pre-trained models*. Tujuan dari *freeze layer* yaitu untuk mencegah hilangnya/rusaknya informasi yang terkandung dalam *pre-trained models*. Parameter yang digunakan pada tahap ini yaitu *requires_grad = False*.

Replacing sequential; karena layer telah dibekukan maka pada tahap ini perlu dilakukan pengaturan terhadap *classifier sequential*. Parameter yang digunakan pada proses *replacing sequential* ini yaitu *mnet.classifier = nn.Sequential(nn.Linear(1280, 2, bias=True), nn.LogSoftmax(dim=1))*.

Save models; pada tahap ini model yang telah dibuat disimpan dengan nama file model.py dan dipanggil kembali pada proses implementasi models.

3. Pengujian Model

Proses pengujian model ini dilakukan dalam 2 tahapan pengujian yaitu; tahap pengujian model fase 1 (*adaptations*) dan model fase 2 (*optimization*).

Adaptations; pada tahap ini pengujian model dilakukan pada data train dan data *test*. Pada tahap *adaptations* model ini diterapkan dua parameter yaitu *Adam optimizer (lr=10e3)* dan *early_stop_patience=2*.

Optimization; pada tahap ini dilakukan pengujian model pada data train dan data *test* dengan menggunakan parameter *Adam optimizer (lr=10e5)* dan *early_stop_patience=5*. Memperkecil nilai parameter *Adam optimizer* pada tahap ini dimaksudkan agar proses

learning dilakukan dengan lebih teliti. Sementara itu nilai parameter *early_stop_patience* dinaikan, dengan tujuan memberikan tambahan 2 kali kali proses learning jika memperoleh nilai akurasi yang sama dengan proses *learning* sebelumnya.

4. Pengukuran Kinerja

Proses pengukuran kinerja dilakukan untuk mengetahui seberapa baik kinerja dari *pre-trained models EfficientNet-B0* dan *MobileNet-V2* pada prediksi Malaria. Pengukuran kinerja model dilakukan dengan menggunakan empat parameter pengujian yaitu; *accuracy*, *confusion matrix*, *F1-score*, dan *sanity check*.

Accuracy; menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. *Accuracy* juga dapat dimaknai sebagai rasio prediksi benar (TP dan TN) terhadap jumlah data secara keseluruhan[26]–[30]. terdapat dua parameter yang digunakan pada tahap ini yaitu *train_score* dan *test_score*. *Train_score* merupakan persentase perbandingan skor akurasi model terhadap data *train* yang digunakan *train_score* = *accuracy(model, trainloader)*. Sedangkan *test_score* merupakan persentase perbandingan skor akurasi model terhadap data *test* yang digunakan *test_score* = *accuracy(model, testloader)*.

Confusion matrix; merupakan salah parameter yang sering digunakan untuk mengukur kinerja metode klasifikasi pada *deep learning* [31], [32]. Pengujian ini digunakan untuk mengukur kinerja dari model klasifikasi maupun prediktif. *Confusion matrix* dapat menggambarkan lebih detail tentang jumlah data yang diklasifikasikan dengan benar maupun salah. Pada pengujian ini digunakan empat parameter yaitu; *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)* [31], [32].

F1-score; dapat didefinisikan sebagai rata-rata harmonik dari *precision* dan *recall* [33], [34]. *F1-score* pada umumnya digunakan untuk membandingkan 2 model dengan dengan kondisi dimana model A memiliki skor *recall* yang lebih tinggi daripada skor *precision*, sedangkan model B memiliki skor *recall* yang lebih rendah daripada skor *precision*. Parameter *F1-score* dapat digunakan sebagai acuan pengukuran kinerja algoritma jika data *False Negatif* dan *False Positif* memiliki nilai yang berbeda cukup signifikan (*asymmetric*)[34].

Sanity check; digunakan untuk menampilkan (visualisasi) hasil prediksi model yang telah dibuat terhadap data target (data sebenarnya). Banyaknya prediksi yang dilakukan

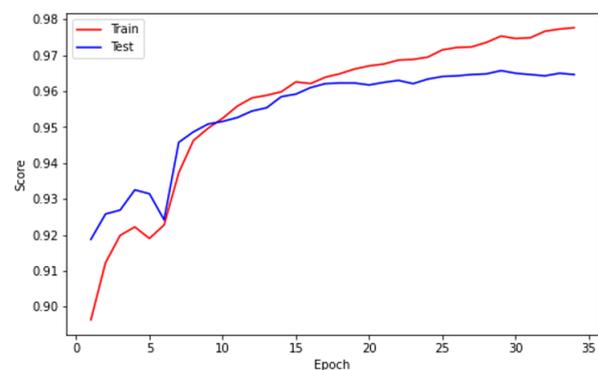
pada pengujian ini mengacu pada parameter *batch_size* (bs=32) yang telah ditentukan pada langkah awal pembuatan model. Hasil pengujian ditampilkan dalam bentuk gambar dan label sesuai hasil prediksi yang telah dilakukan.

Hasil dan Pembahasan

Penulis menyajikan hasil penelitian dan pembahasan pada bagian ini mengacu pada langkah pengukuran kinerja (*performance measure*) model. Dengan demikian proses penyajian data dan pembahasan fokus pada empat parameter yang telah ditentukan yaitu, *accuracy*, *confusion matrix*, *F1-score*, dan pengujian visual menggunakan metode *sanity check*.

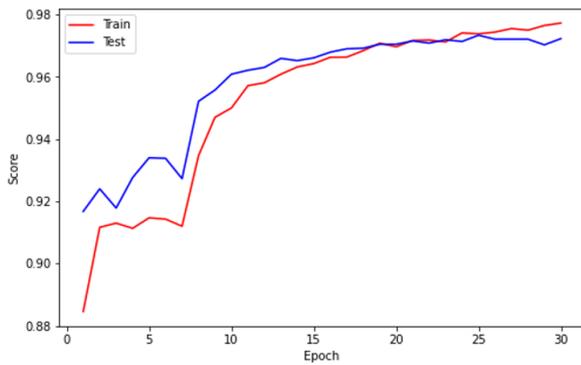
1. Kurva Skor Akurasi

Kurva skor akurasi merupakan kurva yang menampilkan progres perolahan skor akurasi pada setiap proses learning yang telah dilakukan. Gambar 8 menunjukkan kurva perbandingan antara jumlah *epoch* terhadap skor akurasi untuk model algoritma *EfficientNet-B0*, Gambar 9 menunjukkan kurva perbandingan antara jumlah *epoch* terhadap skor akurasi untuk model algoritma *MobileNet-V2* pada tahap adaptasi model khususnya pada fase 2.



Gambar 8: Kurva skor akurasi model *EfficientNet-B0*

Terdapat tiga informasi penting yang diperoleh dari grafik pada Gambar 8 tersebut yaitu: pertama, jumlah total epoch yang diperoleh sebanyak 34; kedua, setelah mencapai *epoch* ke-10, kurva data train menjadi lebih tinggi dari kurva data test; dan ketiga; setelah *epoch* ke-10 kurva data train masih menunjukkan trend naik dengan sampai akhir proses (*epoch* terakhir), sedangkan kurva data test naik dengan lambat dan cenderung stabil.



Gambar 9: Kurva skor akurasi model *MobileNet-V2*

Sedangkan berdasarkan data pada Gambar 9 diketahui bahwa: pertama, jumlah total *epoch* yang diperoleh sebanyak 30; kedua, setelah mencapai *epoch* ke-19, kurva data train menjadi lebih tinggi dari kurva data test; dan ketiga; setelah *epoch* ke-19 kurva data train masih menunjukkan trend naik dengan sampai akhir proses (*epoch* terakhir), sedangkan kurva data test naik dengan lambat dan cenderung stabil.

Dari kedua kurva tersebut diketahui bahwa *epoch* tertinggi pada model algoritma *EfficientNet-B0* dan *MobileNet-V2* merupakan batas akhir dari proses pengujian akurasi. Dimana 1 *Epoch* merupakan 1 putaran penuh algoritma berhasil menyelesaikan rangkaian proses *learning* (analisis). Penentuan batas akhir proses pengujian tersebut diatur dalam parameter *early_stop_patience*=5. Pemberian nilai 5 pada parameter *early_stop_patience* itu dimaksudkan bahwa rangkaian proses analisis yang berjalan akan dihentikan secara otomatis hanya jika algoritma telah memperoleh skor akurasi yang beturut-turut sama dengan skor akurasi pada *epoch* sebelumnya sebanyak 5 kali pengulangan.

Dengan demikian dapat dipahami bahwa pada pengujian model algoritma *EfficientNet-B0* dimulai dari *epoch* ke-30 sampai dengan *epoch* ke-34 memperoleh skor akurasi yang sama, yaitu sebesar 96,57%. Sedangkan pada pengujian model algoritma *MobileNet-V2* dimulai dari *epoch* ke-26 sampai dengan *epoch* ke-30 memperoleh skor akurasi yang sama, yaitu sebesar 97,33%.

2. Skor Akurasi

Data pada Tabel 1 di bawah ini merupakan data skor akurasi data test pada tahap adaptasi model (fase 1).

Tabel 1: Skor akurasi tahap adaptasi model

Test Score	EfficientNet-B0	MobileNet-V2
Pengujian 1	92,78 %	93,66 %
Pengujian 2	92,74 %	93,49 %
Pengujian 3	93,25 %	93,40 %

Berdasarkan data pada Tabel 1 di atas diketahui bahwa pada tahap adaptasi (fase 1) setelah melakukan tiga kali pengujian, model algoritma *EfficientNet-B0* dan *MobileNet-V2*. Skor akurasi model *MobileNet-V2* baik pengujian 1, pengujian 2, maupaun pengujian 3 memiliki nilai yang lebih besar jika dibandingkan dengan skor akurasi model *EfficientNet-B0*. Pada tahap adaptasi model, skor akurasi terkecil untuk model algoritma *EfficientNet-B0* yaitu pada pengujian 2 sebesar 92,74%. Sementara itu skor akurasi terkecil untuk model algoritma *MobileNet-V2* yaitu pada pengujian 3 sebesar 93,40%.

Tabel 2: Skor akurasi tahap optimasi model

Test Score	EfficientNet-B0	MobileNet-V2
Pengujian 1	96,30 %	97,37 %
Pengujian 2	96,39 %	97,12 %
Pengujian 3	96,57 %	97,33 %

Data pada Tabel 2 di atas merupakan data skor akurasi model algoritma *EfficientNet-B0* dan *MobileNet-V2* pada tahap optimasi (fase 2). Pada tahap optimasi skor akurasi model *MobileNet-V2* baik pengujian 1, pengujian 2, maupaun pengujian 3 juga memiliki skor yang lebih besar jika dibandingkan skor akurasi model *EfficientNet-B0*. Pada tahap optimasi model, skor akurasi terkecil untuk model algoritma *EfficientNet-B0* yaitu pada pengujian 1 sebesar 96,30%. Sementara itu skor akurasi terkecil untuk model algoritma *MobileNet-V2* yaitu pada pengujian 2 sebesar 97,12%.

Berdasarkan data pada tabel 1 dan tabel 2 tersebut juga diketahui bahwa terdapat peningkatan rata-rata skor akurasi dari fase 1 ke fase 2 untuk model algoritma *EfficientNet-B0* sebesar 3,5%. Sedangkan peningkatan skor akurasi untuk model algoritma *MobileNet-V2* dari fase 1 ke fase 2 yaitu sebesar 3,76%. Peningkatan skor akurasi baik untuk model algoritma *EfficientNet-B0* maupun *MobileNet-V2* dipengaruhi oleh parameter parameter learning rate dan *early_stop_patience* yang digunakan.

Pada fase 1 parameter learning rate yang digunakan yaitu $lr=1e-3$, sedangkan pada fase 2 parameter learning rate diturunkan menjadi

$lr=1e-5$. Penurunan nilai parameter *learning rate* berpengaruh terhadap kinerja algoritma. Semakin kecil nilai parameter *learning rate*, maka semakin teliti algoritma dalam melakukan analisis. Namun demikian, semakin teliti maka semakin lama waktu yang diperlukan dalam melakukan analisis dan semakin besar *resource* (kapasitas GPU) yang dibutuhkan.

Parameter *early_stop_patience* juga turut berpengaruh terhadap peningkatan skor akurasi model algoritma *EfficientNet-B0* dan *MobileNet-V2*. Proses *learning* fase 1 menggunakan parameter *early_stop_patience=3*, sementara itu pada fase 2 menggunakan parameter *early_stop_patience=5*. Meningkatkan nilai parameter *early_stop_patience* menjadi 5 berarti memberikan 5 kali kesempatan pengulangan mendapatkan skor akurasi yang sama dengan skor akurasi pada epoch sebelumnya. Setelah 5 kali pengulangan mendapatkan skor akurasi yang sama dengan epoch sebelumnya, maka algoritma baru berhenti melakukan analisis.

3. Confusion Matrix

Confusion matrix merupakan salah parameter yang sering digunakan untuk mengukur kinerja metode klasifikasi[11]. *Confusion matrix* pada dasarnya digunakan ketika data sampel yang diuji merupakan data yang bersifat penting dan membahayakan kesehatan atau keselamatan. Data pada Gambar 10 merupakan data hasil pengujian *confusion matrix* pada model algoritma *EfficientNet-B0*. Sedangkan data pada Gambar 11 merupakan data hasil pengujian *confusion matrix* hasil pada model algoritma *MobileNet-V2*.

		Label	
		Infected (1)	Uninfected (0)
Predicted	Infected (1)	2689 True Positive	137 False Positive
	Uninfected (0)	67 False Negative	2619 True Negative

Gambar 10: Confusion matrix model *EfficientNet-B0*

Berdasarkan data hasil pengujian *confusion matrix* untuk model *EfficientNet-B0* pada

Gambar 10, diketahui bahwa terdapat dua kesalahan dalam melakukan prediksi, yaitu kesalahan tipe 1 dan kesalahan tipe 2. Kesalahan tipe 1 pada model ini terjadi karena salah dalam melakukan prediksi terhadap 137 sel darah dinyatakan terinfeksi Malaria, sedangkan faktanya adalah tidak terinfeksi Malaria (FP). Sementara itu kesalahan tipe 2 terjadi karena salah dalam melakukan prediksi terhadap 67 sel darah dinyatakan tidak terinfeksi Malaria, sedangkan faktanya adalah terinfeksi Malaria (FN). Dengan demikian dapat dimaknai bahwa pada pengujian *confusion matrix* model *EfficientNet-B0* masih terdapat kesalahan tipe 1 sebesar 2,48% dan kesalahan tipe 2 sebesar 1,21% dari dari total 5512 sel darah yang digunakan sebagai sampel.

		Label	
		Infected (1)	Uninfected (0)
Predicted	Infected (1)	2707 True Positive	96 False Positive
	Uninfected (0)	49 False Negative	2660 True Negative

Gambar 11: Confusion matrix model *MobileNet-V2*

Selanjutnya berdasarkan data hasil pengujian *confusion matrix* untuk model *MobileNet-V2* pada gambar 11 diketahui bahwa terdapat kesalahan dalam melakukan prediksi yaitu kesalahan tipe 1 dan kesalahan tipe 2. Kesalahan tipe 1 pada model ini terjadi karena salah dalam melakukan prediksi terhadap 96 sel darah dinyatakan terinfeksi Malaria, sedangkan faktanya adalah tidak terinfeksi Malaria (FP). Sementara itu kesalahan tipe 2 terjadi karena salah dalam melakukan prediksi terhadap 49 sel darah dinyatakan tidak terinfeksi Malaria, sedangkan faktanya adalah terinfeksi Malaria (FN). Dengan demikian dapat dimaknai bahwa pada pengujian *confusion matrix* model *MobileNet-V2* masih terdapat kesalahan tipe 1 sebesar 1,74% dan kesalahan tipe 2 sebesar 0,89% dari 5512 sel darah yang digunakan sebagai sampel.

4. F1-score

Data pada Tabel 3 di bawah ini merupakan data data hasil pengujian parameter *F1-score* pada model algoritma *EfficientNet-B0* dan *MobileNet-V2*.

Tabel 3: F1-Score model *EfficientNet-B0* dan *MobileNet-V2*

F1-Score	EfficientNet-B0	MobileNet-V2
Pengujian 1	96,35 %	97,39 %
Pengujian 2	96,45 %	97,15 %
Pengujian 3	96,60 %	97,35 %

Nilai *F1-score* pada umumnya digunakan sebagai acuan untuk mengukur kinerja suatu model algoritma ketika kondisi data *False Negative* dan *False Positive* memiliki perbedaan nilai yang cukup signifikan (*asymmetric*)[10]. Jadi meskipun skor akurasi memiliki nilai yang sangat tinggi tetapi ketika kesalahan tipe 1 dan kesalahan tipe 2 memiliki resiko yang sangat berbahaya, maka nilai *F1-score* lebih tepat digunakan untuk mengukur kinerja algoritma.

Berdasarkan data pada tabel 3 diketahui bahwa nilai *F1-Score* terkecil untuk model algoritma *EfficientNet-B0* yaitu pada pengujian 1 sebesar 96,35%. Sementara itu nilai *F1-Score* terkecil untuk model algoritma *MobileNet-V2* yaitu pada pengujian 2 sebesar 97,15%. Berdasarkan data pada tabel 2 diketahui bahwa skor akurasi terkecil untuk model algoritma *EfficientNet-B0* yaitu pada pengujian 1 sebesar 96,30%. Sementara itu skor akurasi terkecil untuk model algoritma *MobileNet-V2* yaitu pada pengujian 2 sebesar 97,12%.

Mengacu pada skor kedua parameter ini (skor akurasi pada tabel 2 dan *F1-score* pada tabel 3) diketahui bahwa kedua parameter memiliki nilai yang hampir sama besar. Dengan demikian baik skor akurasi maupun *F1-score* dapat digunakan sebagai parameter untuk mengukur kinerja model algoritma *EfficientNet-B0* maupun *MobileNet-V2*.

5. Sanity Check

Data pada Tabel 4 di bawah ini merupakan data persentase prediksi benar hasil pengujian *sanity check*.

Tabel 4: Prediksi benar pengukuran *sanity check*

Predicted True	EfficientNet-B0	MobileNet-V2
Pengujian 1	96,88 %	100 %
Pengujian 2	100 %	100 %
Pengujian 3	100 %	100 %

Pengujian *sanity check* digunakan untuk menampilkan (visualisasi) hasil prediksi model yang telah dibuat terhadap data target (data sebenarnya). Berdasarkan data

pada tabel di atas dapat diketahui bahwa persentase terkecil prediksi benar pada model *EfficientNet-B0* yaitu sebesar 96,88% dan persentase terbesar yaitu sebesar 100%. Sedangkan pada model *MobileNet-V2* persentase prediksi benar sudah mencapai sebesar 100%. Pengujian *sanity check* dilakukan terhadap sampel data test sebanyak 32 sel darah yang dipilih secara acak. Jumlah sampel yang diuji pada metode *sanity check* ini mengacu pada nilai *parameter batch_size = 32* yang ditentukan pada proses data *transformation*.

Hasil pengujian tersebut menampilkan data bahwa meskipun terdapat persentase *false* (salah prediksi) pada model algoritma *EfficientNet-B0*. Namun demikian dapat disimpulkan bahwa berdasarkan parameter pengujian visual (*sanity check*) model algoritma *EfficientNet-B0* dan *MobileNet-V2* dapat dengan akurat melakukan prediksi sampel sel darah.

Penutup

Berdasarkan penelitian yang sudah dilakukan dapat disimpulkan bahwa metode *pre-trained models* algoritma *EfficientNet-B0* dan *MobileNet-V2* dapat digunakan untuk prediksi Malaria. Hal ini dapat terlihat dari beberapa parameter yang digunakan untuk mengukur kinerja model menunjukkan hasil kinerja model yang sangat baik.

Model algoritma terbaik pada penelitian ini yaitu model algoritma *MobileNet-V2*. Model algoritma ini memperoleh rata-rata skor akurasi sebesar 97,27%, rata-rata *F1-score* sebesar 97,29% dan skor *sanity check* sebesar 100%. Berdasarkan parameter pengujian confusion matrix model algoritma *MobileNet-V2* memperoleh skor kesalahan tipe 1 sebesar 1,74% dan kesalahan tipe 2 sebesar 0,89% dari 5512 sel darah yang diuji. Sementara model algoritma *EfficientNet-B0* memperoleh rata-rata akurasi sebesar 96,42%, rata-rata *F1-score* sebesar 96,47% dan skor *sanity check* sebesar 98,96%. Sedangkan berdasarkan parameter pengujian *confusion matrix* model algoritma *EfficientNet-B0* memperoleh skor kesalahan tipe 1 sebesar 2,48% dan kesalahan tipe 2 sebesar 1,21% dari 5512 sel darah yang diuji.

Selanjutnya untuk memaksimalkan hasil penelitian lebih lanjut, terdapat tiga saran bagi peneliti yang tertarik pada bidang penelitian *image classification*, khususnya yang menggunakan pendekatan *pre-trained models*, yaitu: menggunakan parameter *batch size* lebih besar dari 32, parameter *early_stop_patience=8*; dan parameter *lr=10e7*.

Ucapan Terimakasih

Terima kasih penulis ucapkan kepada semua pihak yang telah membantu penelitian ini, terutama

rekan-rekan dan Bapak/Ibu Dosen di Program Studi Magister Teknik Informatika IIB Darmajaya Lampung.

Daftar Pustaka

- [1] N. N. Y. Artini dan E. Y. Tatontos, "Analisis Jenis Plasmodium Penyebab Malaria Terhadap Hitung Jumlah Trombosit", *Jurnal AnalisisMedika Bio Sains*, Feb. 2019.
- [2] A. H. P. Mawuntu, "Malaria Serebral", *Jurnal Neurologi Manado Sinaps*, vol. 1, no. 3, pp. 1–21, 2018.
- [3] J. Fitriany dan A. Sabiq, "Malaria", *Jurnal Averrous*, vol. 4, no. 2, 2018.
- [4] Anonym, "World Malaria Report 2021", diakses daring pada <https://www.who.int/publications/i/item/9789240040496>, 2021.
- [5] R. Irmanita, S. S. Prasetiyowati, and Y. Sibaroni, "Classification of Malaria Complication Using CART (Classification and Regression Tree) and Naïve Bayes", *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 10–16, doi: 10.29207/resti.v5i1.2770, Feb. 2021.
- [6] H. Oktavianus, C. Prihatin, F. Maruhawa, A. Probandari, A. Utarini dan S. Tjokrosonto, "Medical Error dan Perilaku Klinis Petugas Kesehatan dalam Penatalaksanaan Malaria di RSUD Gunung Sitoli Nias", *Jurnal Manajemen Pelayanan Kesehatan*, vol. 12, no. 1, pp. 12–19, Mar. 2009.
- [7] M. Ferdinand Wowor, O. Amelia Waworuntu, H. Polii dan J. Berty Bradly Bernadus, "Sensitivitas dan Spesifisitas Rapid Diagnostic Test Malaria sebagai Diagnostik Laboratorium Malaria di RSUD Noongan", *Jurnal Kedokteran Klinik*, vol. 3, no. 2, pp. 27–33, 2019.
- [8] T. B. Sasongko, "Identifikasi Citra Plasmodium Guna Mendeteksi Parasit Penyakit Malaria dengan Menggunakan Algorithm Convolutional Neural Network", *Science Tech: Jurnal Ilmu Pengetahuan dan Teknologi*, vol. 8, pp. 113–123, Aug. 2022.
- [9] D. Bibin, M. S. Nair and P. Punitha, "Malaria Parasite Detection from Peripheral Blood Smear Images Using Deep Belief Networks", *IEEE Access*, vol. 5, pp. 9099–9108, doi: 10.1109/ACCESS.2017.2705642, 2017.
- [10] A. Nehme, "Understanding Convolutional Neural Networks", *Towards Data Science*, diakses daring pada <https://medium.com/towards-data-science/understanding-convolutional>, May 25, 2018.
- [11] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental Concepts of Convolutional Neural Network", 2020. doi: DOI: 10.1007/978-3-030-32644-9_36.
- [12] Dharmaraj, "Convolutional Neural Networks (CNN) - Architecture Explained", diakses daring pada <https://medium.com/@draj0718/convolutional-neural-networks-cnn-arc>, 2022. [Online].
- [13] R. Awati, "convolutional neural network (CNN)", diakses daring pada <https://www.techtarget.com/searchenterpriseai/definition/convolutional->, 2022.
- [14] P. Marcelino, "Transfer learning from pre-trained models", *Towards Data Science*, diakses daring pada <https://medium.com/towards-data-science/transfer-learning-from-pre-tr>, Oct. 23, 2018.
- [15] M. Tan and Q. v. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", diakses daring pada <http://arxiv.org/abs/1905.11946>, May 2019.
- [16] "Models and pre-trained weights - Torchvision main documentation", diakses daring pada <https://pytorch.org/vision/main/models.html>,
- [17] D. V. Kumar, "Implementing EfficientNet: A Powerful Convolutional Neural Network", diakses daring pada <https://analyticsindiamag.com/implementing-efficientnet-a-powerful-co.>, Jun. 19, 2020.
- [18] A. Shahid, "EfficientNet: Scaling of Convolutional Neural Networks done right", diakses daring pada <https://towardsdatascience.com/efficientnet-scaling-of-convolutional-ne.>, Jun. 17, 2020.
- [19] A. Sarkar, "Understanding EfficientNet - The most powerful CNN architecture", diakses daring pada <https://medium.com/mllearning-ai/understanding-efficientnet-the-most-powerful-cnn-architecture-eaeb40386fad>, May 08, 2021.
- [20] T. Ahmed and N. H. N. Sabab, "Classification and understanding of cloud structures via satellite images with EfficientUNet", diakses daring pada <http://arxiv.org/abs/2009.12931>, Sep. 2020
- [21] R. Ekoputris, "MobileNet: Deteksi Objek pada Platform Mobile", <https://medium.com/nodeflux/mobilenet-deteksi-objek-pada-platform-mobile-bbbf3806e4b3>, May 09, 2018.
- [22] E. Elfatimi, R. Eryigit, and L. Elfatimi, "Beans Leaf Diseases Classification Using MobileNet Models", *IEEE Access*, vol. 10, pp. 9471–9482, doi: 10.1109/ACCESS.2022.3142817, 2022.

- [23] A. Pujara Follow, "Published in Analytics Vidhya", diakses daring pada <https://medium.com/analytics-vidhya/image-classification-with-mobilen>, Jul. 04, 2020.
- [24] J. Rodriguez, "The Evolution of Google's MobileNet Architectures to Improve Computer Vision Models", diakses daring pada <https://analyticsindiamag.com/why-googles-mobilenetv2-is-a-revolutionary-next-gen-on-device->, Feb. 16, 2021.
- [25] Anonym,, "Pytorch Tutorials for Beginner", diakses daring pada https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html, 2022.
- [26] Ridhovan A., Suharso A., and C. Rozikin, "Disease Detection in Banana Leaf Plants using DenseNet and Inception Method", *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 5, pp. 710–718, doi: 10.29207/resti.v6i5.4202, Oct. 2022 .
- [27] A. D. Huri, Suseno R. A., and Azhar Y., "Brain Tumor Classification for MR Images Using Transfer Learning and EfficientNetB3", *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 6, pp. 952–957, doi: 10.29207/resti.v6i6.4357, Dec. 2022 .
- [28] S. Rizal, N. Ibahim, N. K. C. Pratiwi, S. Saidah dan R. Y. N. Fu'adah, "Deep Learning untuk Klasifikasi Diabetic Retinopathy menggunakan Model EfficientNe", *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 8, no. 3, p. 693, doi: 10.26760/elkomika.v8i3.693, Aug. 2020.
- [29] R. Andre, B. Wahyu dan R. Purbaningtyas, "Klasifikasi Tumor Otak Menggunakan Convolutional Neural Network Dengan Arsitektur Efficientnet-B3", *JUST IT: Jurnal Sistem Informasi, Teknologi Informasi dan Komputer*, vol. 11, no. 3, pp. 55–59, 2021.
- [30] Y. Miftahuddin dan F. Zaelani, "Perbandingan Metode Efficientnet-B3 dan Mobilenet-V2 Untuk Identifikasi Jenis Buah-buahan Menggunakan Fitur Daun", *Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 9, no. 1, 2022.
- [31] I. A. Dewi dan F. Rafiqi, "Deteksi Manusia menggunakan Pre-Trained MobileNet untuk Segmentasi Citra Menentukan Bentuk Tubuh", *MIND Journal | ISSN*, vol. ISSN, no. 1, pp. 65–79, doi: 10.26760/mindjourna, 2019.
- [32] S. Aras and A. Setyanto, "Deep Learning Untuk Klasifikasi Motif Batik Papua Menggunakan EfficientNet dan Transfer Learning", *INSECT*, vol. 8, no. 1, 2022.
- [33] J. Korstanje, "The F1 score", diakses daring pada <https://towardsdatascience.com/the-f1-score-bec2bbc38aa6>, Aug. 31, 2021.
- [34] S. Ghoneim, "Accuracy, Recall, Precision, F-Score and Specificity, which to optimize on", diakses daring pada <https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124> Apr. 02, 2019.