

ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK KLASIFIKASI PENYAKIT MALARIA

Winda Widya Ariestya¹, Diny Wahyuni², Bia Irawan³

^{1,2}Fakultas Ilmu Komputer dan Teknologi Informasi, ³Fakultas Teknologi Industri,
^{1,2,3}Universitas Gunadarma

^{1,2,3}Jl. Margonda Raya No. 100 Depok

¹winda_widya@staff.gunadarma.ac.id, ²dwahyuni@staff.gunadarma.ac.id,

³biairawan00@gmail.com

Abstrak : Malaria adalah penyakit yang disebabkan oleh parasit plasmodium, yang berada di aliran darah. Penularan penyakit ini terjadi ketika seseorang digigit nyamuk *Anopheles*. Selain itu, Malaria juga dapat ditularkan melalui penggunaan jarum suntik atau transfusi darah. Proses klasifikasi melibatkan kategorisasi data mentah menjadi data diskrit. Tujuan dari penelitian ini adalah untuk memanfaatkan metode *Convolutional Neural Network* (CNN) untuk memprediksi keakuratan klasifikasi citra sel darah penyakit Malaria, khususnya yang termasuk dalam kategori terinfeksi (*Infected*) dan tidak terinfeksi (*Uninfected*). Data yang digunakan dalam penelitian ini terdiri dari citra sel darah merah yang diperoleh dari situs web <https://www.kaggle.com/>. Model dievaluasi menggunakan metode CNN, menggunakan pemisahan kumpulan data 80:20, pengoptimalan Adam, dan epoch 10 iterasi. Hasilnya menghasilkan tingkat akurasi 89,70% dan loss 25,47%. Selanjutnya, pengujian dilakukan pada 5512 data uji, menghasilkan tingkat akurasi 92%, nilai presisi, recall, dan skor F1 rata-rata 92%.

Kata Kunci : *Machine Learning*, Klasifikasi, Penyakit Malaria, CNN.

Abstract: *Malaria is an illness caused by the plasmodium parasite, which resides within the bloodstream. The transmission of this illness occurs when an individual is bitten by an Anopheles mosquito. Furthermore, Malaria can also be transmitted through the utilization of injection needles or blood transfusions. The process of classification entails the categorization of raw data into discrete data. The objective of this study is to employ the Convolutional Neural Network (CNN) technique in order to forecast the accuracy of blood cell image classification for Malaria illness, particularly for those that fall under the infected and uninfected categories. The dataset utilized in this investigation comprises images of red blood cells that have been acquired from the website <https://www.kaggle.com/>. The CNN approach was used to test the model, with an 80:20 split in the data set, Adam optimization, and epoch 10 iterations. The statistics show an 89.70% accuracy rate and a loss of 25.47%. Furthermore, testing was performed on 5512 test data sets, yielding 92% accuracy, precision, recall, and F1 score averages.*

Keywords: *Machine Learning, Classification, Malaria, CNN.*

PENDAHULUAN

Malaria merupakan penyakit yang disebabkan oleh parasit darah *plasmodium*. Penyakit ini ditularkan

melalui gigitan nyamuk *Anopheles*. Selain ditularkan oleh nyamuk, Malaria juga dapat ditularkan melalui transfusi darah maupun jarum suntik yang digunakan secara

bergantian (Chandley et al., 2023). Kementerian Kesehatan (Kemenkes) mencatat pada Tahun 2022 terdapat 443.530 kasus penyakit Malaria, Tahun 2021 terdapat 304.607 kasus, sementara pada 2020 terdapat sekitar 235.700 kasus di Indonesia (Kemenkes, 2023). Indonesia merupakan salah satu negara endemis Malaria, sehingga penyakit Malaria harus ditangani secara cepat dan tepat untuk menghindari penularan Malaria lebih lanjut dan mencegah komplikasi. Parasit Malaria (*plasmodium*) memiliki siklus hidup yang kompleks, dan membutuhkan inang atau tempat untuk hidup, baik manusia maupun nyamuk *Anopheles*. Terdapat 5 jenis parasit Malaria yang dapat menyerang manusia, yaitu *Plasmodium Falciparum*, *Plasmodium Vivax*, *Plasmodium Malariae*, *Plasmodium Ovale*, dan *Plasmodium Knowlesi* (Yohannes et al., 2020).

Perkembangan Teknologi sangat membantu Dalam upaya untuk mengatasi masalah ini, klasifikasi penyakit Malaria menjadi salah satu aspek penting dalam diagnosis dan pengobatan yang efektif. Identifikasi parasit Malaria dalam sampel darah membutuhkan waktu dan tenaga yang cukup, terutama jika dilakukan secara manual oleh ahli mikroskopis. Oleh karena itu, penggunaan teknologi pengolahan citra dan pembelajaran mesin dapat mempercepat proses ini serta meningkatkan akurasi diagnosa.

Teknologi yang dapat mengenali dan klasifikasi objek adalah *Deep Learning*. *Deep Learning* merupakan bagian dari *Machine Learning* dimana algoritmanya mampu memahami pola dengan akurasi yang tinggi berdasarkan data yang sangat besar melalui berbagai macam variabel kompleks (Ross, 2023). Salah satu metode *Deep Learning* yang dapat digunakan yaitu metode *Convolutional Neural Network* (CNN).

CNN adalah jenis jaringan saraf yang mendominasi berbagai tugas visi komputer dan telah menarik perhatian di berbagai bidang. CNN bertujuan untuk secara otomatis dan adaptif mempelajari struktur

hierarki spasial elemen dengan menggunakan propagasi mundur dari beberapa blok penyusun (seperti lapisan konvolusional, lapisan komposit, dan lapisan yang sepenuhnya terhubung). CNN adalah konstruksi matematis, biasanya terdiri dari tiga jenis lapisan (atau blok penyusun): lapisan konvolusional, lapisan sambungan, dan lapisan yang sepenuhnya terhubung (Utari & Zulfikar, 2023).

Beberapa penelitian yang telah dilakukan sebelumnya digunakan sebagai rujukan dalam penelitian ini. Pada suatu penelitian, metode CNN berhasil melakukan Klasifikasi Penyakit Mata berarsitektur *Alexnet* dengan pembaruan berupa menggunakan 4 kelas yang. Hasil akurasi dari penelitian klasifikasi penyakit mata menggunakan metode CNN adalah 98.37% (Cahya et al., 2021).

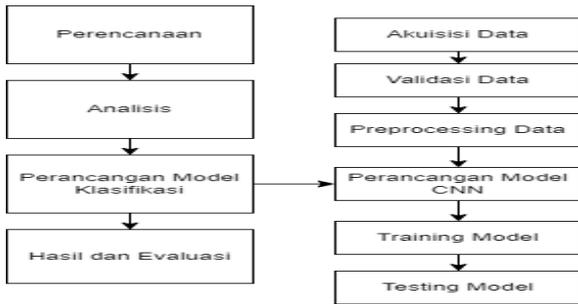
Penelitian lain, CNN digunakan sebagai metode klasifikasi atas kelas kulit normal, penyakit kulit seperti cacar air, campak, jerawat dan skabies. Hasil performansi sistem seperti akurasi, *precision*, *recall*, *f1-score* dan *loss* berturut-turut sebesar 96,53%, 95%, 95%, 95% dan 0,2486 (Hanin et al., 2021).

Berdasarkan uraian penelitian, metode CNN cukup baik dalam klasifikasi pada penyakit dengan menghasilkan akurasi yang baik, sehingga pada penelitian ini akan dilakukan klasifikasi deteksi objek sel darah untuk mendiagnosa penyakit Malaria dengan mengimplementasikan metode CNN.

METODE

Tahapan penelitian yang dilakukan dalam melakukan klasifikasi Malaria tampak pada Gambar 1.

Tahapan penelitian yang dilakukan terdiri dari 4 tahap utama, yaitu perencanaan, analisis, perancangan model klasifikasi dan hasil evaluasi.



Gambar 1. Tahapan Penelitian

Tahap pertama yang dilakukan adalah tahap perencanaan. Dalam penelitian ini, model klasifikasi dibuat untuk membedakan jenis sel darah terinfeksi Malaria (*parasite*) dari tidak terinfeksi (*uninfected*). Data yang diperlukan terdiri dari gambar sel darah merah yang dikumpulkan melalui www.kaggle.com.

Tahap kedua yaitu tahap analisis. Analisis kebutuhan adalah suatu metode untuk menggambarkan kebutuhan yang diperlukan dalam membangun sebuah program, khususnya dalam perancangan sistem klasifikasi penyakit Malaria.

Tahap ketiga yaitu tahap perancangan model klasifikasi. Pada tahap ini dilakukan proses akuisisi data, validasi data, data preprocessing, perancangan model CNN, training model, dan testing model.

Akuisisi Data

Akuisisi data adalah proses untuk mengambil dan mengumpulkan data (Astuti et al., 2022), pada penelitian ini data yang digunakan berupa citra yang bersumber dari www.kaggle.com berupa dataset citra sel darah. Sebanyak 27.560 data citra sel darah yang digunakan sebagai sample penelitian ini dengan jumlah citra gambar per kelas yang diambil dijabarkan pada Tabel 1. Beberapa contoh citra sel darah tergambar pada Gambar 2 (a) dan (b).

Tabel 1 Dataset Citra Sel Darah

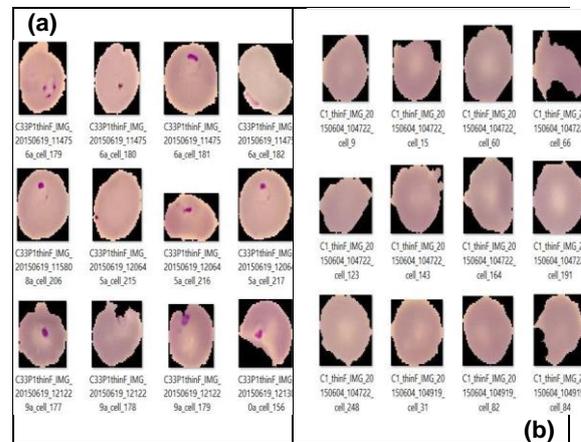
No.	Kelas	Jumlah Citra Gambar
1	Terinfeksi (<i>Parasite</i>)	13.780
2	Tidak Terinfeksi (<i>Uninfected</i>)	13.780

Validasi Data

Validasi Data dilakukan untuk memastikan dataset yang diambil dari Kaggle dinyatakan benar, dengan menanyakan ke Dosen kedokteran Gunadarma yaitu, Dr. Sri Mukti Suhartini.

Data Preprocessing

Sebelum data dapat digunakan pada model *Convolutional Neural Network*, data akan dilakukan proses *Preprocessing* agar sesuai dengan keperluan arsitektur model. *Preprocessing* yang dilakukan yaitu pembagian data, augmentasi data dan persiapan data generator.



Gambar 2 (a) Sel darah terinfeksi (*parasite*) dan (b) Sel darah tidak terinfeksi (*uninfected*)

1. Pembagian Data

Dataset yang digunakan dalam penelitian ini dikelompokkan berdasarkan rasio pembagian dimana kelas dibagi secara acak, seperti yang ditunjukkan dalam tabel 2.

Tabel 2 Pembagian Rasio Data

Pembagian Rasio	Data	Presentase	Jumlah Data
80:20	Latih	80%	22048
	Uji	20%	5512

2. Augmentasi Data

Proses augmentasi citra dilakukan karena karakteristik dari metode CNN yang tidak invarian terhadap augmentasi gambar dilakukan agar program dapat mengenali lebih banyak bentuk citra. Proses ini diawali dengan mendefinisikan objek image data generator untuk *training datagen* dan *test datagen*. Citra dilakukan proses rescale untuk

normalisasi atau perubahan skala pada nilai piksel dalam gambar (Supiyani et al., 2022). *Rescale* pada *train_datagen* dan *test_datagen* menggunakan *rescale* sebesar 1/255. Dengan parameter *rotation_range* 10, gambar-gambar dalam dataset pelatihan dapat mengalami rotasi acak antara -10 derajat dan +10 derajat. *Zoom_range* digunakan untuk memperbesar gambar dan melakukan rotasi horizontal dan vertical, dan parameter *rescale* digunakan untuk proses meningkatkan *test_datagen*.

- Persiapan data generator, meliputi *Target Size*, *Class mode*, *Batch Size*, dan *Shuffle*.

Target size merupakan dimensi gambar yang akan digunakan dalam proses *training* (Suwitono & Kaunang, 2022). *Target size* yang digunakan pada penelitian ini adalah 224x224 *pixel*.

Class Mode merupakan metode pemilihan klasifikasi. *Class Mode* dapat berupa *binary* atau *categorical*. *Class_mode* yang dipilih adalah *categorical*.

Batch size merupakan banyaknya gambar yang dimasukkan dalam setiap steps dari *training* (Saputra & Imran, 2023). *Batch_size* yang dipakai sebesar 32.

Shuffle merupakan perintah agar urutan gambar didalam folder diacak sehingga tidak sesuai dengan urutan yang ada. *Shuffle* yang dipakai pada data *train* adalah *true* sedangkan *subset* dipakai pada data *test* adalah *none*.

Perancangan model CNN

Perancangan model CNN digunakan untuk melakukan klasifikasi jenis citra sel darah Malaria. Perancangan model menggunakan *library keras* dan *Tensorflow*.

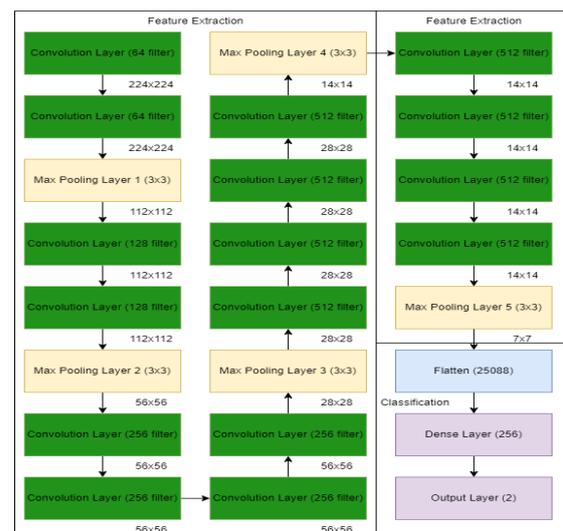
Tahap pertama pada perancangan CNN adalah *feature extraction*. Proses ini akan melakukan encoding dari gambar menjadi *feature* berupa angka yang merepresentasikan gambar tersebut. Proses *feature extraction* terdiri dari dua

bagian yaitu *Convolutional Layer* dan *Pooling Layer* (Hardi & Sundari, 2023).

Setelah gambar melalui tahap *feature extraction*, proses dilanjutkan ke tahap *classification*. Proses ini akan mempelajari hasil dari *feature extraction* sebelumnya. Bagian dari proses ini adalah *flatten layer* dan *Fully-Connected Layer*.

Bagian *flatten layer* akan menerima masukan dari hasil *maxpooling* terakhir berupa array 2 dimensi. Dikarenakan tahap selanjutnya yaitu *fully-connected layer* memerlukan masukan dalam bentuk array 1 dimensi, maka perlu dilakukan fungsi *flatten* agar gambar dapat diklasifikasikan.

Bagian *fully-connected layer* akan menerima masukan hasil *flatten layer* berupa array 1 dimensi. Layer pertama dari *fully-connected layer* memiliki 256 *neuron* dengan fungsi aktivasi yang digunakan adalah ReLu, kemudian layer terakhir memiliki 2 *neuron* sesuai dengan kelas jenis Malaria yang akan diklasifikasikan. Layer terakhir menggunakan fungsi aktivasi softmax. Fungsi softmax akan menentukan probabilitas pada tiap kelas yang ada.



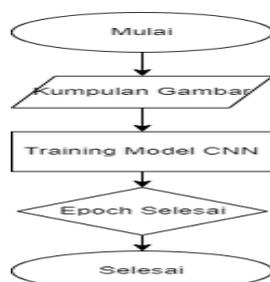
Gambar 3 Proses *Training* pada Arsitektur VGG19 dan Sequential

Gambar 3 merupakan proses *training* yang dilakukan pada penelitian ini. Setelah perancangan model CNN selesai

dibuat, model selanjutnya akan di-*compile* agar siap dilakukan proses *training*. Terdapat beberapa pengaturan proses *compile* diantaranya; *Optimizer* merupakan pengaturan optimasi model yang digunakan. Penelitian ini menggunakan 1 skenario optimasi untuk mencari optimasi terbaik. Optimasi yang dipakai adalah Adam. Optimasi Adam digunakan karena optimasi Adam merupakan optimasi yang menggabungkan properti terbaik dari optimasi AdaGrad dan RMSProp. *Loss* merupakan pengaturan pengukuran nilai loss yang dipakai. *Metrics* merupakan pengaturan pengukuran yang diukur pada model. *Learning Rate* merupakan pengaturan untuk menghitung nilai koreksi bobot pada proses *training*. Nilai learning rate berapa pada range 0 sampai 1. Semakin besar nilai *learning rate*, maka proses *training* berjalan semakin besar namun ketelitian berkurang (Nurfita, 2018).

Training Model

Setelah perancangan model CNN selesai dilakukan, proses dilanjutkan dengan *training* model. Proses ini diharapkan mendapatkan model yang sudah terlatih untuk mengingat pola gambar dengan baik dari masing-masing kelas yang dilatih. Proses *training* yang dilakukan menggunakan data latih dan data validasi yang sudah dilakukan pra-pemrosesan sebelumnya. Alur proses *training* model dijabarkan pada Gambar 4.



Gambar 4 Alur *Training* model

Training dilakukan dengan menggunakan data latih dan data validasi yang akan dibagi menjadi sekumpulan gambar (*batch*). Sekumpulan gambar tersebut akan dilatih menggunakan model CNN yang telah dibuat sampai proses *training*

dilakukan hingga seluruh kumpulan gambar selesai dilatih dan jumlah *epoch* yang didefinisikan selesai dilakukan. Proses *training* memiliki beberapa parameter diantaranya: *Epoch* merupakan pengaturan banyaknya iterasi pada proses *training*, *Batch Size* merupakan jumlah gambar yang digunakan pada satu kali proses *training*, *Validation Data* merupakan pendefinisian validation data yang akan dievaluasi setiap *epoch* selesai dilakukan, *Verbose* adalah pengaturan bagaimana progress dari tiap *epoch* ditampilkan. Proses pengujian pada *training* dilakukan menggunakan 2 skenario *epoch* yaitu, dan 10 *epoch*. Dengan adanya penambahan skenario *epoch* dapat digunakan sebagai perbandingan untuk menentukan tingkat akurasi optimal.

Testing Model

Dari beberapa model pengujian pada tahap *training*, skenario model pengujian tahap *training* dengan hasil terbaik akan dilakukan proses *testing* model. Tahap *testing* ini menghasilkan prediksi dari setiap gambar pada data uji.

HASIL DAN PEMBAHASAN

Pembuatan model CNN menggunakan model VGG19 dimana model yang dibuat memiliki 16 *layer*. *Input_shape* yang digunakan yaitu 224x224x3. *Layer convolutional* pertama dan kedua menggunakan 64 *filter*. Angka 3 merupakan *input channel RGB*. *Layer* pertama dan kedua memiliki *pooling layer* yang menggunakan *maxpooling* dengan ukuran kernel 3x3.

Layer Convolutional ketiga dan keempat menggunakan 128 *filter* dengan ukuran 3x3 memiliki *pooling layer* yang menggunakan *maxpooling* dengan ukuran kernel 3x3. *Layer Convolutional* kelima, keenam, ketujuh, kedelapan menggunakan 256 *filter* dengan ukuran 3x3 memiliki *pooling layer* yang menggunakan *maxpooling* dengan ukuran kernel 3x3. *Layer Convolutional*

kesembilan, kesepuluh, kesebelas, kedua belas menggunakan 512 *filter* dengan ukuran 3x3 memiliki *pooling layer* yang menggunakan *maxpooling* dengan ukuran kernel 3x3. *Layer Convolutional* ketiga belas, keempat belas, kelima belas, keenam belas menggunakan 512 *filter* dengan ukuran 3x3 memiliki *pooling layer* yang menggunakan *maxpooling* dengan ukuran kernel 3x3.

Setelah keenam belas *layer* selesai dibuat, terdapat perintah untuk melakukan *flatten* agar dapat digunakan ke proses *fully-connected layer*. Setelah dilakukan proses *flatten*, proses akan memanggil *dropout* yang berfungsi untuk mencegah terjadinya *overfitting* dan mempercepat proses *training*. Setelah dilakukan proses *dropout*, proses berlanjut dengan perintah membuat *fully-connected layer*. *Fully-connected layer* memiliki 2 buah *hidden layer*. *Layer* pertama memiliki 256 *neuron* dengan fungsi aktivitas *ReLU*. *Layer* kedua merupakan *output layer* yang memiliki 2 *neuron* dengan fungsi aktivasi *softmax*.

Hasil yang didapatkan berdasarkan model yang telah dilatih sebanyak 10 epoch. Pada *training* data menghasilkan Nilai *Accuracy* sebesar 0.8951 dan Nilai *Loss* 0.2575. Sedangkan pada test data menghasilkan Nilai *Accuracy* sebesar 0.8988 dan Nilai *Loss* 0.2519. Sehingga hasil akhir **Accuracy** yaitu 89.70%. Hasil dari tahap pelatihan model dijabarkan pada tabel 3.

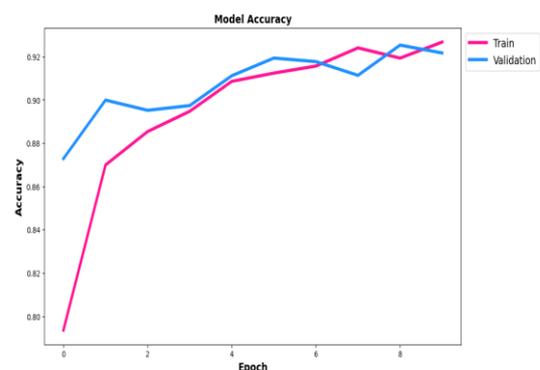
Berdasarkan hasil pelatihan, jika dilihat dari hasil akhir Nilai *Accuracy* dan Nilai *Val Accuracy* dapat disimpulkan bahwa model yang sudah berhasil dilatih tidak terjadi *overfitting*. *Overfitting* adalah kondisi ketika model tidak bisa melakukan prediksi yang tepat jika diberikan dataset serupa, menangkap *noise* yang seharusnya

diabaikan dan menyebabkan tingkat akurasi rendah dan nilai *loss* yang rendah (Tantyoko et al., 2023).

Tabel 3 Hasil Training Model

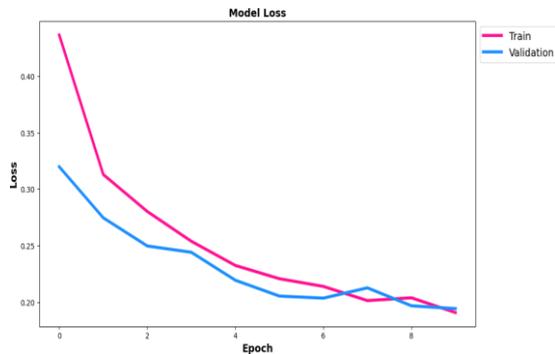
Epoch	Waktu Pelatihan (detik)	Nilai Accuracy	Nilai Loss	Val Accuracy	Val Loss
1	221	0.7947	0.4438	0.8721	0.3074
2	200	0.8695	0.3143	0.8859	0.2893
3	199	0.8868	0.2793	0.9086	0.2412
4	198	0.8973	0.2554	0.9019	0.2351
5	198	0.9039	0.2386	0.8543	0.3442
6	198	0.9115	0.2263	0.9138	0.2133
7	219	0.9177	0.2115	0.9224	0.2016
8	199	0.9215	0.2028	0.9013	0.2442
9	199	0.9239	0.2005	0.9149	0.2105
10	198	0.9263	0.1935	0.9213	0.1985

Hasil *training* model, dapat di visualisasikan kedalam bentuk grafik. Berikut adalah grafik akurasi dan kegagalan selama pelatihan model yang divisualisasikan pada Gambar 5 dan 6.



Gambar 5 Grafik Model Accuracy

Sebagai perbandingan untuk menentukan tingkat akurasi optimum berdasarkan epoch dapat dilihat pada tabel 4. Dalam penelitian ini, epoch yang paling ideal adalah 10 epoch, seperti yang ditunjukkan pada Tabel 4.



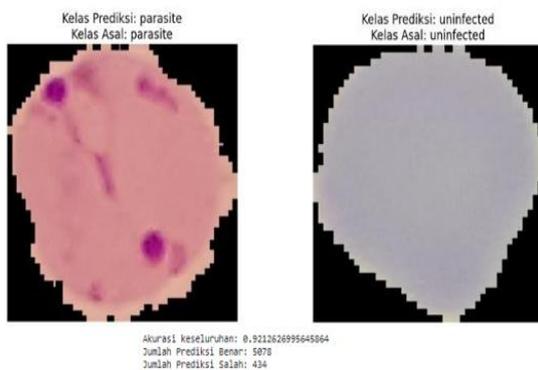
Gambar 6 Grafik Model Loss

Tabel 4 Perbandingan Epoch

Banyak Epoch	Waktu Pelatihan (detik)	Nilai Accuracy	Nilai Loss
5	198	0.8881	0.2778
10	206	0.8970	0.2547

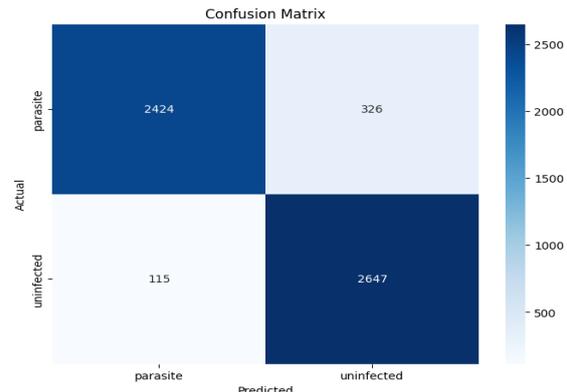
Pelatihan dengan 10 epoch membutuhkan waktu sekitar 206 detik dengan nilai akurasi yang tinggi dengan 0.8970 dan nilai kehilangan 0.2547, pelatihan dengan 5 epoch membutuhkan waktu sekitar 198 detik dengan nilai akurasi terkecil yaitu 0.881 dan nilai kehilangan 0.2778.

Setelah pelatihan selesai, citra sel darah dapat diklasifikasikan menjadi dua kategori penyakit Malaria, yaitu terinfeksi (*parasite*) dan tidak terinfeksi (*uninfected*). Tingkat akurasi hasil uji coba model pada uji coba gambar setelah dilatih ditunjukkan pada Gambar 7.



Gambar 7 Hasil Testing Model

Confusion matrix digunakan pada tahap evaluasi yang berfungsi untuk mengetahui apakah kinerja model dibuat sudah baik atau belum dengan menggunakan. Hasil dari visualisasi evaluasi ini digambarkan pada Gambar 8.



Gambar 8 Confusion Matrix Hasil Prediksi

Pada Gambar 8 dapat dilihat bahwa semua prediksi data uji Malaria *parasite* dan *uninfected* sesuai dengan labelnya. *Parasite* mendeteksi benar 2424 citra, sedangkan *uninfected* berhasil mendeteksi benar 2647 citra, dan *parasite* mendeteksi salah 326 citra, sedangkan *uninfected* mendeteksi salah 115 citra.

Setelah *Confusion Matrix* selesai dibuat, proses dilanjutkan dengan menghitung akurasi, *presisi*, *recall*, dan *f1-score*. Berdasarkan pengujian yang telah dilakukan maka didapatkan nilai TP (*True Positive*), FN (*False Negative*), TN (*True Negative*), dan FP (*False Positive*).

	precision	recall	f1-score	support
parasite	0.95	0.88	0.92	2750
uninfected	0.89	0.96	0.92	2762
accuracy			0.92	5512
macro avg	0.92	0.92	0.92	5512
weighted avg	0.92	0.92	0.92	5512

Gambar 9 Hasil akurasi, presisi, recall, dan f1-score

Gambar 9 merupakan hasil perhitungan *confusion matrix* menggunakan program. Diperoleh nilai akurasi sebesar 92%, rata-rata presisi 92%, rata-rata *recall* sebesar 92%, dan rata-rata *f1-score* sebesar 92%. Dari perhitungan akurasi, presisi, recall, dan *f1-score* tersebut dapat disimpulkan bahwa model klasifikasi yang dibuat dapat melakukan klasifikasi dengan cukup baik.

KESIMPULAN

Pembangunan model klasifikasi Malaria dengan CNN untuk

mengklasifikasikan dua jenis Malaria yaitu, Malaria terinfeksi (*parasite*) dan Malaria tidak terinfeksi (*uninfected*) telah berhasil dilakukan dengan menggunakan dataset citra sel darah sebanyak 27.560 citra. Model klasifikasi yang dibuat dapat melakukan klasifikasi dengan cukup baik, dilihat dari hasil pengujian diperoleh nilai akurasi sebesar 92%, nilai *presisi*, *recall*, dan *f1-score* rata-rata sebesar 92%.

REFERENSI

- [1]. Astuti, I., Ariestya, W. W., & Solehudin, B. (2022). Deteksi Objek Daun Semanggi Secara Real Time Menggunakan CNN-Single Shot Multibox Detector (SSD). *Jurnal Ilmiah FIFO*, 14(1), 47. <https://doi.org/10.22441/fifo.2022.v14i1.005>
- [2]. Cahya, F. N., Hardi, N., Riana, D., & Hadiyanti, S. (2021). Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN). *Sistemasi*, 10(3), 618. <https://doi.org/10.32520/stmsi.v10i3.1248>
- [3]. Chandley, P., Ranjan, R., Kumar, S., & Rohatgi, S. (2023). Host-parasite interactions during Plasmodium infection: Implications for immunotherapies. *Frontiers in Immunology*, 13(January), 1–27. <https://doi.org/10.3389/fimmu.2022.1091961>
- [4]. Hanin, M. A., Patmasari, R., Fuâ, R. Y. N., & others. (2021). Sistem Klasifikasi Penyakit Kulit Menggunakan Convolutional Neural Network (cnn). *EProceedings of Engineering*, 8(1), 273–281.
- [5]. Hardi, N., & Sundari, J. (2023). Pengenalan Telapak Tangan Menggunakan Convolutional Neural Network (CNN). *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 4(1), 10–15. <https://doi.org/10.31294/reputasi.v4i1.1951>
- [6]. Kemenkes. (2023). *Peta Endemisitas Malaria Periode 2023*. <https://malaria.kemkes.go.id/case>
- [7]. Nurfitra, R. D. (2018). Implementasi Deep Learning Berbasis Tensorflow. *Jurnal Emitor*, 18(01), 22–27.
- [8]. Ross, O. H. M. (2023). Various Deep Learning Algorithms in Computational Intelligence. *Axioms*, 12(5), 10–14. <https://doi.org/10.3390/axioms12050495>
- [9]. Saputra, D. H., & Imran, B. (2023). Object Detection Untuk Mendeteksi Citra Buah-Buahan Menggunakan Metode Yolo. *Jurnal Kecerdasan Buatan Dan ...*, 2(2), 70–80. <http://ojs.ninetyjournal.com/index.php/JKBTI/article/view/18%0Ahttp://ojs.ninetyjournal.com/index.php/JKBTI/article/download/18/14>
- [10]. Supiyani, I., & Arifin, N. (2022). Identifikasi Nomor Rumah Pada Citra Digital Menggunakan Neural Network. *METHODIKA: Jurnal Teknik Informatika Dan Sistem Informasi*, 8(1), 18–21. <https://doi.org/10.46880/mtk.v8i1.921>
- [11]. Suwitono, Y. A., & Kaunang, F. J. (2022). Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Daun Dengan Metode Data Mining SEMMA Menggunakan Keras. *Jurnal Komtika (Komputasi Dan Informatika)*, 6(2), 109–121. <https://doi.org/10.31603/komtika.v6i2.8054>
- [12]. Tantyoko, H., Sari, D. K., & Wijaya, A. R. (2023). Prediksi Potensial Gempa Bumi Indonesia Menggunakan Metode Random Forest Dan Feature Selection. *IDEALIS: InDonEsiA Journal Information System*, 6(2), 83–89. <https://doi.org/10.36080/idealis.v6i2.3036>
- [13]. Utari, L., & Zulfikar, A. (2023). Penerapan Convolutional Neural

- Networks Menggunakan Edge Detection Untuk Identifikasi Motif Jenis Batik. *TeknoIS : Jurnal Ilmiah Teknologi Informasi Dan Sains*, 13(1), 110–123. <https://doi.org/10.36350/jbs.v13i1.184>
- [14]. Yohannes, Y., Devella, S., & Arianto, K. (2020). Deteksi Penyakit Malaria Menggunakan Convolutional Neural Network Berbasis Saliency. *JUITA: Jurnal Informatika*, 8(1), 37. <https://doi.org/10.30595/juita.v8i1.6671>