

# Implementasi *You Only Look Once* (YOLO) Menggunakan Metode *Deep Learning* dalam Mendeteksi Telur Menetas pada Reptil

Isa Mahfudi<sup>a,1,\*</sup>, Ahmad Rozak Setia Nugraha<sup>a,2</sup>, Azam Muzakhim Imammuddin<sup>a,3</sup>

<sup>a</sup>Politeknik Negeri Malang, Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Indonesia

<sup>1</sup> isa\_mahfudi@polinema.ac.id\*; <sup>2</sup>2041160152@student.polinema.ac.id; <sup>3</sup>azam@polinema.ac.id

\* Penulis Koresponden

## INFO ARTIKEL

### Histori Artikel

Pengajuan  
Diperbaiki  
Diterima

### Kata Kunci

YOLO  
Reptil  
*Leopard Gecko*  
*Epoch*  
Telur

## ABSTRAK

Saat ini reptil dijadikan hewan peliharaan karena perawatan yang mudah dan warna motif yang beragam salah satunya adalah *Leopard Gecko*. Perawatan *Leopard Gecko* yang baru menetas berbeda dari yang dewasa karena *Leopard Gecko* yang baru menetas harus segera ditempatkan di kandang yang diberi alas tisu dan disemprot air untuk menghindari kehilangan air. penelitian ini bertujuan untuk mengembangkan sistem deteksi telur menetas menggunakan YOLO (You Only Look Once). Hasil penelitian menunjukkan bahwa algoritma YOLO dapat digunakan untuk mendeteksi telur *Leopard Gecko* menetas secara *real-time*. persentase keberhasilan deteksi YOLO mencapai 94,73% pada jarak kamera 25 cm, pencahayaan 512-895 lux, dan 200 *epoch* pelatihan atau dapat mendeteksi 54 dari 57 objek. Jadi, model YOLO yang telah dilatih sudah memiliki keandalan yang baik dalam mendeteksi telur menetas pada reptil, sehingga pengguna dapat mengetahui jika ada telur yang menetas di dalam inkubator dan dapat melakukan tindakan lebih lanjut terhadap reptil yang baru menetas.

Ini adalah artikel akses terbuka di bawah lisensi [CC-BY-SA](#).



## 1. Pendahuluan

Reptil merupakan salah satu jenis hewan yang dijadikan peliharaan. Alasan orang memelihara reptil adalah karena eksotisme, perawatan yang cukup mudah, sifat yang tenang, dan tidak mudah sakit [1]. Salah satu contoh reptil yang mudah untuk dipelihara adalah *Leopard Gecko* [2]. *Leopard Gecko* memiliki keunggulan pada segi warna dan motif yang beragam, perawatan yang mudah dan harga yang berkisar ratusan ribu hingga jutaan rupiah [3].

Perawatan pada *Leopard Gecko* dewasa dengan *Leopard Gecko* yang baru menetas memiliki perbedaan karena saat *Leopard Gecko* baru menetas maka harus segera dipindahkan dari inkubator ke kandang yang diberi alas menggunakan tisu dan disemprotkan dengan air untuk mencegah *Leopard Gecko* dehidrasi [4]. Untuk mengetahui telur yang telah menetas diperlukan sistem yang dapat mendeteksi telur menetas secara *real-time*. YOLO merupakan algoritma yang dapat digunakan mendeteksi suatu objek secara *real-time* [5]. Algoritma YOLO membutuhkan beberapa data gambar untuk dipelajari agar saat diimplementasikan, YOLO

dapat mendeteksi beberapa objek dan mengklasifikasikannya berdasarkan gambar yang dipelajari sebelumnya oleh algoritma YOLO [6].

Pada penelitian yang berjudul "Implementasi Pengolahan Citra pada *Quadcopter* untuk Deteksi Manusia Menggunakan Algoritma YOLO model yang digunakan adalah YOLOv5" dengan jumlah data latih 700 gambar dan *epoch* sebanyak 100 menghasilkan model dengan akurasi 86,8% [6]. Pada penelitian ini jumlah data latih yang digunakan adalah sebesar 683 gambar dan *epoch* sebanyak 200 menghasilkan akurasi sebesar 94,73%.

Tujuan dari penelitian ini adalah membuat sistem deteksi telur menetas pada reptil agar pengguna dapat melakukan tindakan pada telur yang baru menetas ke tempat yang sesuai dengan kebutuhan masing-masing reptil.

## 2. Metode penelitian

### 2.1 Deep Learning

*Deep Learning* adalah salah satu cabang pembelajaran dari *Machine Learning* yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang disusun secara berlapis-lapis. Pada dasarnya *Deep Learning* dibangun menggunakan jaringan syaraf tiruan (*Neural Network*) [7][8]. Pada arsitektur pada *Deep Learning*, terdapat tiga jenis layer yang terdapat pada *Deep Learning*, yaitu *input layer*, *hidden layer*, dan *output layer*. Layer adalah komponen dari *Neural Network* yang berfungsi untuk mengumpulkan masukkan, memproses informasi, dan menghasilkan keluaran [7][9].

*Input layer* adalah layer pertama dari jaringan saraf tiruan (*Neural Network*) yang menerima input atau masukkan dari luar [10]. Setelah menerima *input*, layer ini akan meneruskan data ke *hidden layer* pada neuron yang merupakan representasi dari *input* tersebut [11].

*Hidden layer* adalah layer menengah dalam jaringan saraf tiruan (*Neural Network*) yang terletak antara *input layer* dan *output layer* [7]. Bagian ini disebut sebagai *hidden layer* karena proses yang dilakukan tidak dapat dilihat oleh pengguna secara langsung [12]. *Hidden layer* berfungsi untuk melakukan pengecekan pola yang ada dalam data dan mengekstraksi fitur-fitur yang berguna [13].

*Output layer* adalah layer terakhir dalam jaringan saraf tiruan (*Neural Network*) yang menghasilkan *output* dari proses yang telah dilakukan oleh *hidden layer* [7]. Pada *output layer*, hasil yang dihasilkan akan diolah untuk membuat keputusan yang tepat [14].

### 2.2 Dataset

*Dataset* adalah kumpulan data yang disusun dengan cara yang terstruktur [15]. Pada penelitian ini *dataset* yang digunakan berupa gambar *dataset* diperoleh dengan mengumpulkan secara manual di internet dan sosial media sebanyak 275 gambar yang terdiri dari dua macam gambar yaitu telur normal dan menetas untuk tiap reptil yaitu *Leopard Gecko*, *Ball Python*, dan *Bearded Dragon*. Selanjutnya, gambar yang telah diperoleh dilakukan anotasi menggunakan Roboflow dengan dua kelas yaitu telur normal dan telur menetas. Roboflow adalah *platform web* yang digunakan untuk pemrosesan pra-data dan teknik pelatihan model *computer vision* [16]. Setelah gambar selesai dianotasi, gambar akan dibagi secara otomatis menjadi tiga yaitu gambar latih, gambar validasi, dan gambar uji [17]. Setelah gambar dianotasi, dilakukan augmentasi dan duplikasi pada data gambar sehingga menjadi 615 gambar untuk pelatihan, 54 gambar untuk validasi, dan 14 gambar untuk pengujian.

### 2.3 Perancangan

YOLO menggunakan teknologi *deep learning* dan menggabungkan berbagai teknik seperti *Convolutional Neural Network* (CNN) dan *Non-Maximum Suppression* (NMS) [7][18]. Kepepresian dari model YOLO dapat ditentukan dengan menggunakan matriks evaluasi [19]. Matriks evaluasi terdiri dari *Precision*, *Recall*, mAP50, mAP50-95 [20]. *Precision* adalah kepresisian atau kecocokan permintaan informasi dengan jawaban dari permintaan tersebut [21]. *Recall* adalah evaluasi klasifikasi yang mengukur kemampuan model dalam mengidentifikasi sampel positif yang sebenarnya [6]. Sedangkan, mAP50 adalah nilai yang dihitung dari *Average Precision* (AP) dengan nilai *Intersection of Union* (IoU) sebesar 50% [22]. Nilai kepresisian pada matriks ini dipengaruhi oleh jumlah *epoch*, semakin banyak *epoch* maka model akan semakin presisi [23]. Namun, jika *epoch* terlalu banyak akan mengakibatkan model *over fitting* [24].

Dalam menentukan matriks evaluasi terdapat empat istilah yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), *False Negative* (FN). Rumus untuk menghitung matriks evaluasi dapat dilihat pada persamaan 1, 2, dan 3 [25].

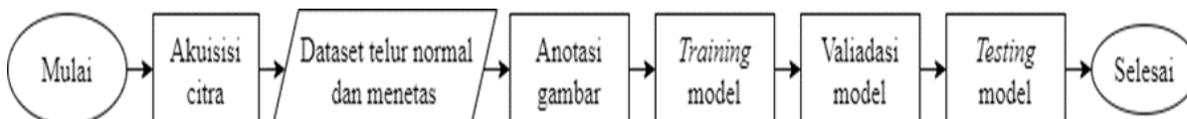
$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$IoU \frac{Truth}{Pred} = \frac{Area\ of\ Overlap}{Area\ Union} \quad (3)$$

### 2.4 Pelatihan

Tahapan pelatihan YOLO ditunjukkan pada Gambar 1



Gambar 1. Alur Pelatihan YOLO

Proses akuisisi citra dilakukan dengan cara melakukan pengumpulan citra telur normal dan menetas secara manual di internet dan media sosial. Setelah data citra terkumpul dilakukan anotasi menggunakan Roboflow yang dilakukan secara manual. Selanjutnya, melakukan augmentasi pada gambar yang telah dianotasi untuk memberikan varian gambar untuk data pelatihan.

Data gambar yang telah dianotasi dan augmentasi digunakan untuk melakukan pelatihan YOLO menggunakan Google Colab. Pada Google Colab jumlah *epoch* dapat ditentukan secara manual. Jika data menunjukkan tingkat kepresisian model tidak meningkat saat *epoch* ditingkatkan maka pelatihan dapat dihentikan.

Untuk menguji kepresisian model YOLO dapat dilakukan validasi dan pengujian model menggunakan Google Colab. Data yang digunakan untuk melakukan validasi dan pengujian adalah data yang tidak digunakan dalam pelatihan.

### 3. Hasil dan Analisis

#### 3.1. Pengujian Jarak Optimal Deteksi YOLO

Pengujian menggunakan gambar telur normal dan menetas yang dicetak pada kertas. Total jumlah objek yang digunakan untuk pengujian adalah sebanyak 57 objek. Gambar 2 merupakan hasil pengujian jarak terhadap YOLO dalam mendeteksi telur. Pada jarak 25 cm YOLO dapat mendeteksi 51 objek dan semakin bertambahnya jarak maka jumlah objek yang mampu dideteksi YOLO semakin menurun. Hal ini terjadi karena semakin jauh objek maka fitur pada objek akan semakin tidak terlihat. Sehingga, YOLO akan sulit mendeteksi objek.

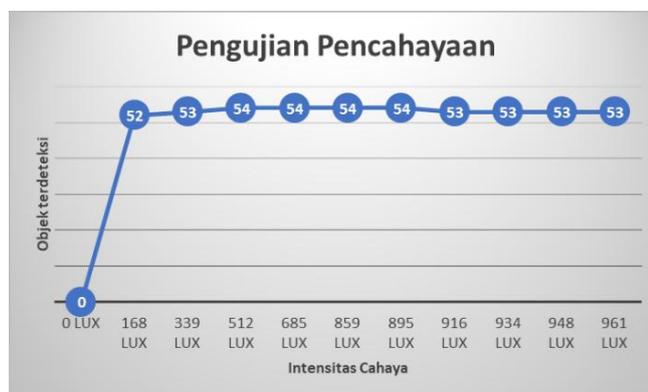


Gambar 2. Hasil Pengujian Jarak

#### 3.2. Pengujian Pengaruh Pencahayaan

Gambar 3 menunjukkan hasil pengujian pencahayaan terhadap deteksi objek pada YOLO. Jumlah objek pada citra adalah 11 dan YOLO hanya dapat mendeteksi 7 – 8 objek. Hal ini dikarenakan terdapat objek yang berdempetan dan kontras antara objek dengan latar belakang yang rendah sehingga menyebabkan fitur dan tepi objek sulit dideteksi YOLO.

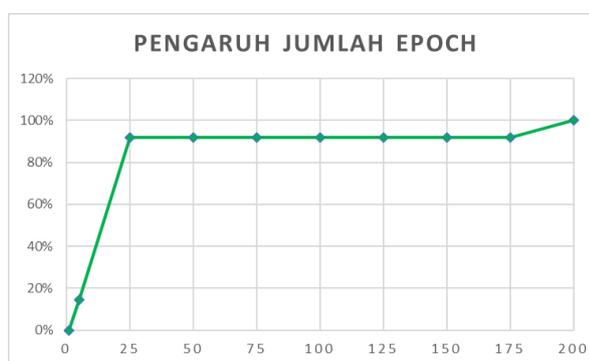
Saat tidak ada cahaya YOLO tidak dapat mendeteksi objek satu pun. Namun, saat intensitas cahaya pada 168 – 339 lux YOLO mendeteksi 7 objek. Saat intensitas pada rentang 512 – 895 lux jumlah objek yang dideteksi YOLO meningkat karena tepi dan fitur pada objek terlihat lebih jelas pada rentang tersebut dan menyebabkan jumlah objek yang terdeteksi meningkat. Pada rentang 916 – 961 lux jumlah menurun menjadi 7 objek karena cahaya yang berlebihan menyebabkan fitur dan tepi objek tidak terlihat. Selain itu, cahaya yang berlebihan menyebabkan penurunan kontras. Ini menyebabkan YOLO sulit membedakan objek dengan latar belakangnya.



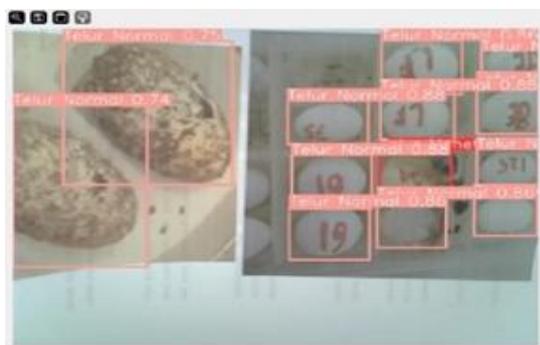
Gambar 3. Hasil Pengujian Pencahayaan

### 3.3 Pengujian Pengaruh Jumlah Epoch

Gambar 4 merupakan hasil pengujian pengaruh jumlah *epoch* pelatihan terhadap persentase keberhasilan deteksi objek YOLO. Pada saat YOLO dilatih pada *epoch* 1 persentase keberhasilan YOLO dalam mendeteksi adalah 0%. Pada *epoch* 5 objek yang persentase keberhasilan meningkat menjadi 14,26%. Saat *epoch* ditingkat menjadi 25 sampai 175 persentase keberhasilan meningkat menjadi 92% dan pada *epoch* 200 persentase keberhasilan deteksi mencapai 100%. Hal ini menandakan semakin banyak jumlah *epoch* pelatihan maka YOLO semakin memahami fitur khas dari sebuah objek dan ini dapat meningkatkan keberhasilan deteksi YOLO. Hasil deteksi YOLO terhadap gambar telur normal dan menetas ditunjukkan pada Gambar 5.

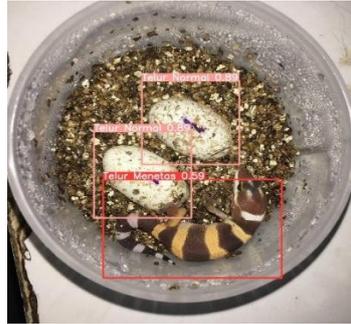


Gambar 4. Grafik Pengujian Jumlah Epoch



Gambar 5. Hasil Deteksi YOLO pada Gambar Telur

Gambar 6 merupakan hasil deteksi YOLO terhadap telur normal dan menetas dari *Leopard Gecko*. Persentase keberhasilan dalam mendeteksi objek mencapai 100% dengan nilai *confidence* di atas 0,5. Pada objek telur normal nilai *confidence* yang didapat yaitu 0,89 dan untuk telur menetas 0,59. Hal ini menunjukkan YOLO yang telah dilatih memiliki performansi yang baik karena nilai *confidence* pada objek yang dideteksi nilainya di atas 0,5 atau keyakinan YOLO di atas 50%.



**Gambar 6.** Hasil Deteksi YOLO pada Telur *Leopard Gecko*

#### 4. Conclusion

Pengujian menunjukkan bahwa jarak, pencahayaan, dan jumlah *epoch* pelatihan mempengaruhi keberhasilan deteksi objek YOLO. Dengan jarak kamera dengan objek 25 cm, pencahayaan 512–895 lux, dan 200 *epoch* pelatihan, model YOLO yang telah dilatih mencapai hasil terbaik. YOLO dapat menemukan 54 dari 57 objek telur normal dan menetas, dengan persentase keberhasilan 94,73%. Model ini hanya dapat digunakan untuk mendeteksi objek telur dari tiga reptil yaitu *Leopard Gecko*, *Ball Python*, dan *Bearded Dragon*.

Hasil dari pelatihan dan pengujian model ini lebih tinggi dari penelitian sebelumnya karena pengaruh dari jumlah *epoch* yang digunakan saat pelatihan. Dengan tingkat kepresisian model yang mencapai 94,73% maka model YOLO ini dapat bekerja dengan baik untuk mendeteksi telur reptil yang menetas, sehingga pengguna dapat mengetahui jika ada telur yang menetas di dalam inkubator dan dapat melakukan tindakan lebih lanjut terhadap reptil yang baru menetas.

#### References

- [1] A. I. Asmarany and P. P. Setiawan, "PERBEDAAN KEBAHAGIAAN PADA INDIVIDU YANG MEMELIHARA MAMALIA ATAU REPTIL," *UGJ.*, vol. 13, no. 12, pp. 46–57, 2022.
- [2] I. S. Aditya Pratama, Beauty Anggraheny Ikawanty, "Sistem Kontrol dan Monitoring Suhu Serta Kelembapan Untuk Kandang Bayi Leopard Gecko Menggunakan Metode PI Berbasis IOT," *Sci. J. Ilm. dan Teknol.*, vol. 15, no. 1, pp. 72–86, 2024, doi: 10.25130/sc.24.1.6.
- [3] Y. R. D. Wahana, "Sistem Pendukung Keputusan Pemilihan Morph Leopard Gecko Untuk Pemula Menggunakan Metode Simple Additive Weighting," Politeknik Negeri Jember, 2021.
- [4] P. Tani, "Cara Ternak Leopard Gecko," *anekabudidaya.com*. Accessed: Jun. 21, 2024. [Online]. Available: <https://anekabudidaya.com/cara-ternak-leopard-gecko/>
- [5] F. Rachmawati and D. Widhyaestoeti, "Deteksi Jumlah Kendaraan di Jalur SSA Kota Bogor Menggunakan Algoritma Deep Learning YOLO," *Pros. LPPM UIKA Bogor*, pp. 360–370, 2020.
- [6] A. N. Sugandi, B. Hartono, and K. Kunci, "Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO," *Pros. 13th Ind. Res. Work. Natl. Semin.*, pp. 13–14, 2022.
- [7] I. Cholissodin and A. A. Soebroto, *AI, MACHINE LEARNING & DEEP LEARNING (Teori & Implementasi)*, no. December. 2021.
- [8] S. R. Dewi, "Deep Learning Object Detection Pada Video," *Deep Learn. Object Detect. Pada*

- Video Menggunakan Tensorflow Dan Convolutional Neural Netw.*, pp. 1–60, 2018, [Online]. Available: [https://dSPACE.uui.ac.id/bitstream/handle/123456789/7762/14611242\\_SyarifahRositaDewi\\_Statistika.pdf?sequence=1](https://dSPACE.uui.ac.id/bitstream/handle/123456789/7762/14611242_SyarifahRositaDewi_Statistika.pdf?sequence=1)
- [9] R. Windiawan and A. Suharso, "Identifikasi Penyakit pada Daun Kopi Menggunakan Metode Deep Learning VGG16," *Exploreit*, vol. 13, no. 2, pp. 9–16, 2019, [Online]. Available: <https://doi.org/10.35891/explorit>
- [10] R. Rikendry and A. Maharil, "Perbandingan Arsitektur Vgg16 Dan Resnet50 Untuk Rekognisi Tulisan Tangan Aksara Lampung," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 3, no. 2, pp. 236–243, 2022, doi: 10.33365/jatika.v3i2.2030.
- [11] E. Setyowati and S. Mariani, "Penerapan Jaringan Syaraf Tiruan dengan Metode Learning Vector Quantization ( LVQ ) untuk Klasifikasi Penyakit Infeksi Saluran Pernapasan Akut ( ISPA )," *Prism. Pros. Semin. Nas. Mat.*, vol. 4, pp. 514–523, 2021, [Online]. Available: <https://journal.unnes.ac.id/sju/index.php/prisma/article/view/44356>
- [12] A. Salim, "Estimasi Kecepatan Kendaraan Melalui Video Pengawas Lalu Lintas Menggunakan Parallel Line Model," 2020.
- [13] L. O. A. S. Sagala, "Klasifikasi Cats dan Dogs dengan Metode CNN dalam Fungsi Aktivasi Relu , Sigmoid , Softmax , Softplus , Softsign , dan Selu," *Pros. Semin. Nas. Inform. Bela Negara di UPN "Veteran" Jawa Timur*, vol. 2, no. 1, pp. 1–6, 2023.
- [14] Bambang Pulu Hartato, "Penerapan Convolutional Neural Network pada Citra Rontgen Paru-Paru untuk Deteksi SARS-CoV-2," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 747–759, 2021, doi: 10.29207/resti.v5i4.3153.
- [15] H. Tantyoko, D. K. Sari, and A. R. Wijaya, "Prediksi Potensial Gempa Bumi Indonesia Menggunakan Metode Random Forest Dan Feature Selection," *IDEALIS Indones. J. Inf. Syst.*, vol. 6, no. 2, pp. 83–89, 2023, doi: 10.36080/idealism.v6i2.3036.
- [16] Hadi Supriyanto, Sarosa Castrena Abadi, and Aliffa Shalsabilah, "Deteksi Helm Keselamatan Menggunakan Jetson Nano dan YOLOv7," *J. Appl. Comput. Sci. Technol.*, vol. 5, no. 1, pp. 1–8, 2024, doi: 10.52158/jacost.v5i1.637.
- [17] N. D. G. Drantantiyas *et al.*, "Performasi Deteksi Jumlah Manusia Menggunakan YOLOv8," *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, vol. 5, no. 2, pp. 63–68, 2023, doi: 10.26905/jasiek.v5i2.11605.
- [18] N. Gunawan, T. Waras, and S. Si, *KUASAI MACHINE LEARNING & COMPUTER VISION DALAM SEKEJAP*. Padang, 2024.
- [19] JOY HANS CHRISTABEL SINAGA, "EVALUASI KINERJA METODE KECERDASAN BUATAN BERBASIS YOLOv5 UNTUK DIAGNOSIS JENIS KANKER KULIT SECARA MOBILE REAL TIME," 2024.
- [20] M. Ikbali and R. A. Saputra, "PENGENALAN RAMBU LALU LINTAS MENGGUNAKAN METODE YOLOv8," *JIKA (Jurnal Inform. Univ. Muhammadiyah Tangerang)*, vol. 8, no. 2, pp. 204–212, 2024.
- [21] L. Rahma, H. Syaputra, A. H. Mirza, and S. D. Purnamasari, "Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once)," *J. Nas. Ilmu Komput.*, vol. 2, no. 3, pp. 213–232, 2021, doi: 10.47747/jurnalnik.v2i3.534.
- [22] N. Benarkah, M. F. Naufal, and B. Renatasiva, "YOLOv5 untuk Menghitung Sel Darah Merah dan Sel Darah Putih," *Keluwih J. Sains dan Teknol.*, vol. 5, no. 1, pp. 10–18, Apr. 2024, doi: 10.24123/saintek.v5i1.6291.

- [23] Moh. Faizin, "IMPLEMENTASI ALGORITMA YOLO PADA MODUL KAMERA UNTUK DETEKSI JENIS DAN KECEPATAN KENDARAAN," UNIVERSITAS JEMBER, 2023.
- [24] Q. Antoko Mohti, R. Wahyudi, and M. Habib Mustofa, "Penerapan Metode YoloV5 Dalam Mendeteksi Penyakit Tanaman Buah Naga," *Semin. Nas. Teknol. Sains*, vol. 3, no. 1, pp. 318–323, 2024.
- [25] N. J. Hayati, D. Singasatia, and M. R. Muttaqin, "Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 untuk Menghitung Kendaraan," *Komputa J. Ilm. Komput. dan Inform.*, vol. 12, no. 2, pp. 91–99, 2023, doi: 10.34010/komputa.v12i2.10654.