

Implementasi Metode YOLOv8 Mendeteksi Komputer Aktif dengan Subjek Layar Monitor

Frisky Wijaya ^{(1)*}, Dedy Hermanto ⁽²⁾

Departemen Informatika, Universitas Multi Data, Palembang, Indonesia

e-mail : friskywijaya@mhs.mdp.ac.id, dedy@mdp.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 26 Agustus 2024, direvisi 19 Februari 2025, diterima 20 Februari 2025, dan dipublikasikan 30 September 2025.

Abstract

Computers are one example of technological advances used in education. The use of computers that are not turned off can cause damage to computer components, and the use of electrical energy can increase. Student disobedience in turning off school laboratory computers when finished using them causes teachers to conduct manual checks by visiting each computer laboratory in the school. Deep learning is a machine learning algorithm that uses artificial neural networks. Deep learning is usually used for image recognition, voice identification, and data pattern analysis. Therefore, this study will apply the Deep Learning method, specifically YOLOv8, which aims to detect active computers based on the subject of the monitor screen and is expected to provide information about computers that are still active in the school laboratory. Based on the study's results, which detected 10 active computers, the 200-epoch model was selected with 100% accuracy at a speed of 2 ms. Twenty active computers were selected, with 200 epoch models achieving 95% accuracy at a speed of 6 ms per epoch. Thirty active computers were selected, with 100 epoch models achieving 96.67% accuracy at a speed of 3 ms.

Keywords: Deep Learning, Computers, School Laboratory, YOLOv8, Monitor Screen

Abstrak

Komputer merupakan perangkat yang memiliki perkembangan teknologi yang cukup pesat serta digunakan dalam dunia pendidikan. Penggunaan komputer yang tidak dimatikan dapat menyebabkan kerusakan pada komponen-komponen komputer serta penggunaan energi listrik dapat meningkat. Ketidakpatuhan siswa dalam mematikan komputer laboratorium sekolah ketika selesai digunakan menyebabkan guru melakukan pengecekan manual dengan mengunjungi setiap laboratorium komputer sekolah. Deep learning merupakan algoritma pembelajaran mesin yang menggunakan jaringan saraf tiruan. Deep learning biasanya digunakan untuk pengenalan citra, identifikasi suara, dan analisis pola data. Penelitian kali ini akan menggunakan metode Deep Learning yaitu YOLOv8 bertujuan untuk mendeteksi komputer aktif dengan subjek layar monitor serta diharapkan dapat memberikan informasi terkait komputer yang masih aktif di laboratorium sekolah. Dari hasil penelitian, untuk mendeteksi 10 komputer aktif dipilih model 200 epoch memiliki akurasi 100% dengan kecepatan 2 ms. Pada skenario 20 komputer aktif, model dengan 200 epoch memberikan akurasi 95% dengan kecepatan 6 ms. Sementara itu, untuk 30 komputer aktif, model dengan 100 epoch memiliki akurasi 96,67% dengan kecepatan 3 ms.

Kata Kunci: Deep Learning, Komputer, Laboratorium Sekolah, YOLOv8, Layar Monitor

1. PENDAHULUAN

Dalam era digital saat ini, penggunaan komputer tidak terpisahkan dari berbagai aspek kehidupan seperti di lingkungan perkantoran, pendidikan, dan industri. Komputer digunakan untuk berbagai aktivitas mulai dari pengolahan data, komunikasi serta bisnis. Keberadaan komputer semakin banyak digunakan oleh pengguna karena manfaatnya (Monalia et al., 2022). Pada dasarnya, beberapa bagian komputer memiliki jangka waktu hidup terbatas, yang dapat berkurang jika digunakan secara terus menerus. Salah satu cara untuk mencegah terjadinya kerusakan komponen komputer tersebut dengan mematikan komputer ketika selesai digunakan. Penggunaan komputer di laboratorium sekolah menjadi masalah mengenai bagaimana tingkat kepatuhan siswa dalam mematikan komputer setelah selesai digunakan. Masih terdapat siswa



yang belum patuh dalam mematikan komputer setelah digunakan, sehingga guru melakukan pemantauan secara manual dengan mengunjungi setiap laboratorium komputer untuk mengetahui informasi tersebut.

Berdasarkan permasalahan yang telah dijelaskan, dibutuhkan sebuah perangkat lunak yang mampu memberikan informasi kepada guru terkait komputer yang belum dimatikan. Penelitian saat ini menggunakan Artificial Intelligence, sebuah teknologi yang berkembang sangat pesat dan mampu menyelesaikan masalah yang sulit diselesaikan dengan mudah. Secara umum, Artificial Intelligence dibagi menjadi dua bagian yaitu Deep Learning dan Machine Learning (Raup et al., 2022).

Pada penelitian berjudul "Klasifikasi Jenis Burung Menggunakan Metode CNN dan Arsitektur ResNet-50" memperoleh nilai akurasi 98% dengan *optimizer* SGD (Alberto & Hermanto, 2023). Adapun penelitian lain untuk pengenalan telapak tangan menggunakan Convolutional Neural Network (CNN) menggunakan metode Alexnet memperoleh nilai akurasi 97,01% pada epoch ke-7 (Hardi & Sundari, 2023). Penelitian lain untuk mengenali wajah orang juga telah dilakukan menggunakan algoritma HOG dengan *fitcecoc multiclass* SVM yang memperoleh tingkat akurasi 98,5714% (Laia et al., 2023). Dalam penelitian berjudul "Object Detection untuk Menentukan Citra Buah-Buahan dengan Metode YOLO" memperoleh nilai akurasi mAP sebesar 91% (Saputra et al., 2023).

Pada penelitian kali ini akan menerapkan Deep Learning dengan model Convolutional Neural Network (CNN) yang dikenal dengan "YOLO" (You Only Look Once) untuk deteksi komputer aktif melalui layar monitor di laboratorium sekolah. YOLO memiliki tiga bagian utama yaitu *backbone*, *neck*, dan *head* (Ali & Zhang, 2024; Gupta et al., 2025; Sadrawi et al., 2023). Model YOLO memiliki bobot ringan sehingga dapat berjalan secara *real-time* pada perangkat keras dengan memanfaatkan GPU untuk meningkatkan efisiensi deteksi (Alfarizi et al., 2023; Betti & Tucci, 2023). Deteksi dan pengenalan *real-time* menjadi keunggulan utama YOLO dengan kemampuan pengenalan mencapai 45 *frame* per detik serta memberikan kinerja yang sangat baik dalam situasi waktu nyata (Luthfi, 2021; Yaseen, 2024). Deteksi dan pengenalan objek dilakukan oleh YOLO dengan cepat dan akurat dengan kecepatan mencapai 22 ms per gambar (Amin & Obermaisser, 2025; Mulyana & Rofik, 2022). Pada penelitian ini diharapkan dapat mencegah terjadinya kerusakan komputer dan diharapkan dapat membantu mengurangi biaya pemakaian listrik dari komputer yang belum dimatikan.

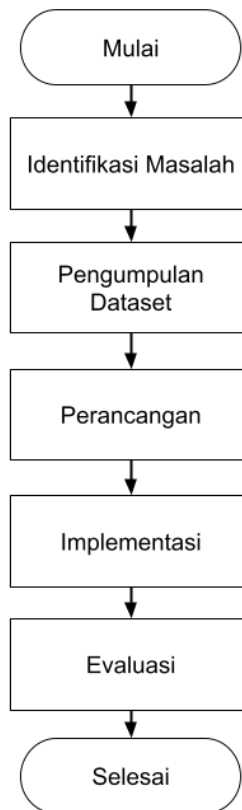
2. METODE PENELITIAN

Tahapan penelitian yang dilakukan untuk mendeteksi keberadaan komputer aktif menggunakan metode YOLOv8 ditunjukkan pada Gambar 1. Tahap awal yang dilakukan yaitu pengumpulan informasi yang berhubungan dengan topik penelitian, mempelajari jurnal yang berkaitan dengan deteksi keberadaan komputer aktif menggunakan metode You Only Look Once (YOLO). Tahap selanjutnya dilakukan pengumpulan *dataset* yang disusun secara *primer*. Data tersebut berupa gambar yang diambil dengan menggunakan *smartphone* Redmi Note 10S dengan spesifikasi kamera yaitu 64 Megapixel (MP). Pengambilan gambar dilakukan di laboratorium komputer, kemudian gambar yang diperoleh dipotong sesuai kebutuhan. Jumlah *dataset* yang terkumpul sebanyak 700 gambar, terdiri dari 490 data *train*, 140 data validasi, dan 70 data *test*. Contoh *dataset* yang diambil dalam penelitian ini ditunjukkan pada Gambar 2.

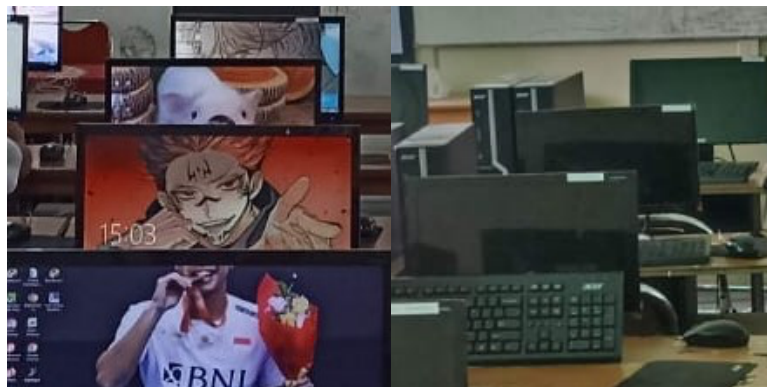
Pada tahap perancangan, dilakukan perancangan sistem yang diperlukan untuk melakukan penelitian yaitu penggunaan metode YOLO untuk deteksi keberadaan komputer aktif. Proses ini dimulai dengan membuat label dari setiap *dataset* yang telah di kumpulkan, sebagaimana yang ditunjukkan pada Gambar 3. Pengumpulan *dataset* dilakukan secara *primer*, kemudian dilakukan *labeling* dengan bantuan aplikasi Roboflow. Salah satu contoh proses pelabelan dapat dilihat pada Gambar 4. Setelah proses pelabelan selesai, *dataset* diekspor menjadi data *train* terdiri dari 490 gambar, data *test* terdiri dari 70 gambar, dan data validasi terdiri dari 140 gambar. Pemisahan *dataset* tersebut menghasilkan *source code* yang digunakan pada tahap pelatihan model. Proses pemisahan *dataset* ditunjukkan pada Gambar 5.



Pada tahap implementasi dilakukan dengan bantuan *Google Colab* dengan bahasa pemrograman *Python* menggunakan metode YOLO. Di mana akan melatih *dataset* yang telah disiapkan sebelumnya sehingga menghasilkan sebuah model. Model tersebut akan digunakan untuk *testing* deteksi keberadaan komputer aktif dalam pembuatan perangkat lunak berbasis *website* dengan bahasa pemrograman *Python* serta dengan bantuan *framework flask* dan *library OpenCV*. YOLO memproses gambar secara *real-time* pada empat puluh lima (45) *frames per second* (Khairunnas et al., 2021). Arsitektur YOLO terdiri atas dua puluh empat lapisan yang terhubung penuh, diikuti oleh empat lapisan dan dua lapisan yang terhubung penuh. YOLO menggambarkan identifikasi objek sebagai masalah regresi Tunggal, gambar ini akan dihubungkan dengan sebuah *bounding box* spasial yang terpisah dan probabilitas kelas yang terkait (Hutauruk et al., 2020).



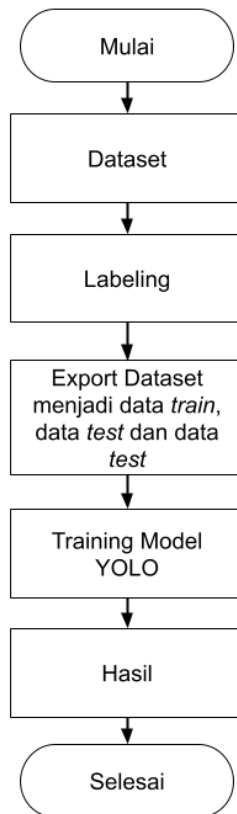
Gambar 1 Tahapan Penelitian



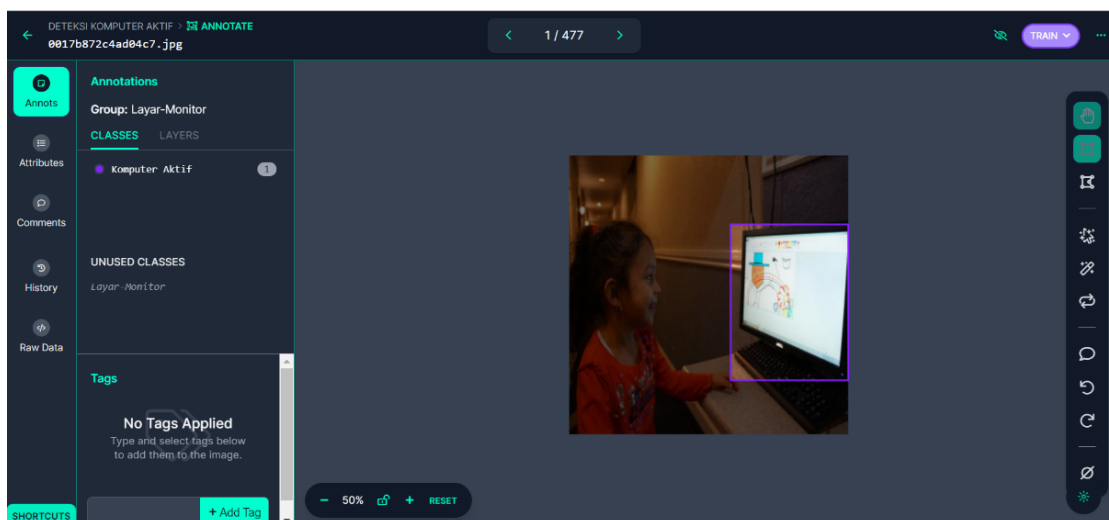
Gambar 2 Contoh Dataset Komputer



Proses dimulai dengan menambahkan dan menginstall *dependency* yang dibutuhkan, sebagaimana ditunjukkan pada potongan kode program yang ditunjukkan pada Gambar 6. Kemudian lakukan proses *training* model dengan epoch yang ditentukan, misalnya 100 epoch, menggunakan file data.yaml yang telah diunduh dan disimpan di lokasi yang ditentukan seperti yang ditunjukkan pada potongan kode program pada Gambar 7. Hasil dari proses *training* tersebut, seperti yang ditunjukkan pada Gambar 8, akan disimpan dan dapat diunduh pada folder *runs/detect/train/weights/last.pt*.

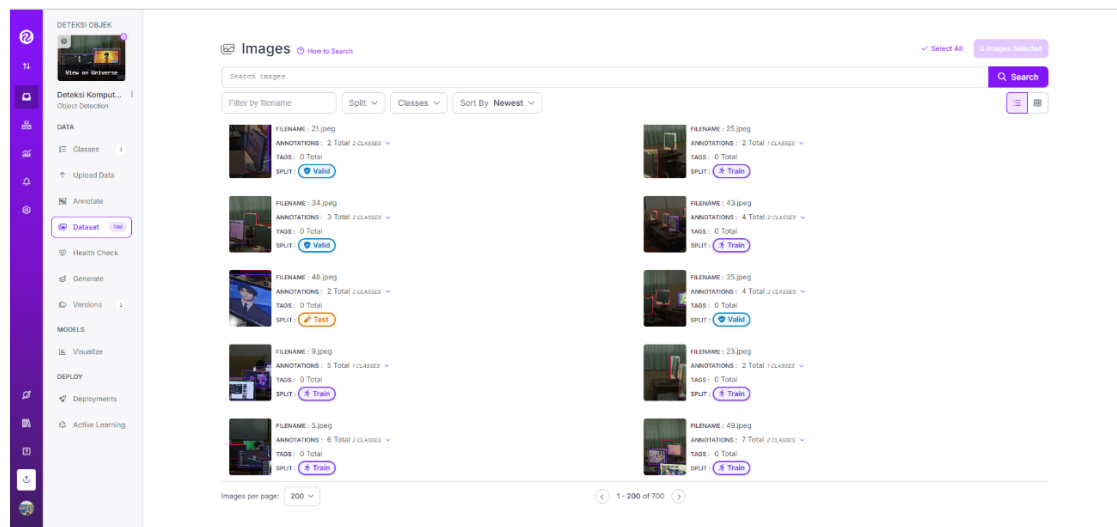


Gambar 3 Perancangan Model YOLO



Gambar 4 Contoh *Labeling* dengan Bantuan Roboflow



Gambar 5 Contoh Hasil *Export Dataset*

```
!pip install ultralytics==8.0.196
from IPython import display
display.clear_output()
import ultralytics ultralytics.checks()
from ultralytics import YOLO
from IPython.display import display, Image
```

Gambar 6 Kode Program Instalasi *Dependency*

```
%cd {HOME}
!yolo task=detect mode=train model=yolov8s.pt
data={dataset.location}/data.yaml epochs=100
```

Gambar 7 Kode Program Training Model

```
100 epochs completed in 0.366 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126358 parameters, 0 gradients, 28.4 GFLOPs
```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95
all	140	622	0.955	0.957	0.982	0.796
Komputer Aktif	140	344	0.943	0.942	0.975	0.805
Komputer TidakAktif	140	278	0.966	0.971	0.99	0.786

```
Speed: 0.3ms preprocess, 6.5ms inference, 0.0ms loss, 6.6ms postprocess per image
```

Gambar 8 Hasil Model Proses Pelatihan

Tabel 1 Confusion Matrix

Prediksi	Aktual		
	Positif		Negatif
	Positif	True Positive (TP)	False Positive (FP)
Negatif	False Negative (FN)	True Negative (TN)	

Setelah implementasi, tahap evaluasi dilakukan menggunakan Confusion Matrix. Confusion Matrix memberikan pemahaman tentang kesalahan model dengan mengungkapkan saling ketergantungan antara kelas yang diberi label dengan yang diprediksi (Erbani et al., 2024).



Evaluasi ini melibatkan beberapa komponen yaitu akurasi, *precision*, dan *recall*. *Accuracy* digunakan untuk menggambarkan akurasi dari suatu model yang dalam mengklasifikasikan dengan benar (Romadloni et al., 2022). Akurasi merupakan prediksi data benar (positif dan negatif) dengan keseluruhan data (Rahayu et al., 2021), seperti yang ditunjukkan pada Pers. (1). *Precision* digunakan untuk menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model (Romadloni et al., 2022). *Precision* merupakan prediksi data benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif (Rahayu et al., 2021), seperti yang ditunjukkan pada Pers. (2). *Recall* digunakan untuk menggambarkan keberhasilan model dalam menemukan Kembali sebuah informasi (Romadloni et al., 2022). *Recall* merupakan prediksi data benar positif yang berbeda dari keseluruhan data benar positif (Rahayu et al., 2021), seperti yang ditunjukkan pada Pers. (3).

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

3. HASIL DAN PEMBAHASAN

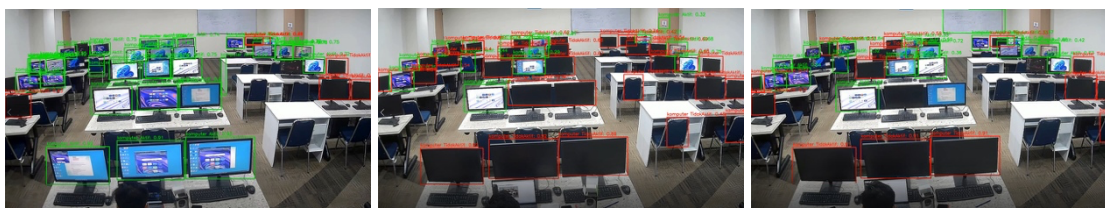
Pengujian dilakukan menggunakan metode YOLOv8 dengan beberapa epoch yaitu 100 epoch, 150 epoch, dan 200 epoch. Pengujian tersebut dilakukan pada salah satu laboratorium komputer yang memiliki jumlah komputer 36 komputer dengan tiga kondisi yaitu 10, 20, dan 30 komputer aktif. Dari hasil pengujian tersebut dilakukan analisis, kemudian akan dipilih epoch yang memiliki akurasi tertinggi pada setiap kondisi dan waktu tercepat.

3.1 Pengujian 100 Epoch

Pada pengujian dengan 100 epoch, hasil pengujian dari ketiga kondisi dapat dilihat pada Tabel 2 dan Gambar 9. Pada kondisi pertama, jumlah komputer aktif sebanyak 10 unit, seluruhnya berhasil terdeteksi dengan benar. Jumlah komputer tidak aktif sebanyak 26 unit, dengan 24 unit terdeteksi benar dan 2 unit tidak terdeteksi. Pada kondisi kedua, jumlah komputer aktif sebanyak 20 unit, namun hanya 17 unit yang berhasil terdeteksi, sedangkan 3 unit tidak terdeteksi. Jumlah komputer tidak aktif sebanyak 16 unit, dengan 13 unit terdeteksi benar dan 3 unit tidak terdeteksi. Sementara itu, pada kondisi ketiga, jumlah komputer aktif sebanyak 30 unit, dengan 29 unit berhasil terdeteksi dan 1 unit tidak terdeteksi. Jumlah komputer tidak aktif sebanyak 6 unit, dengan 4 unit terdeteksi benar dan 2 unit tidak terdeteksi.

Tabel 2 Pengujian 100 Epoch

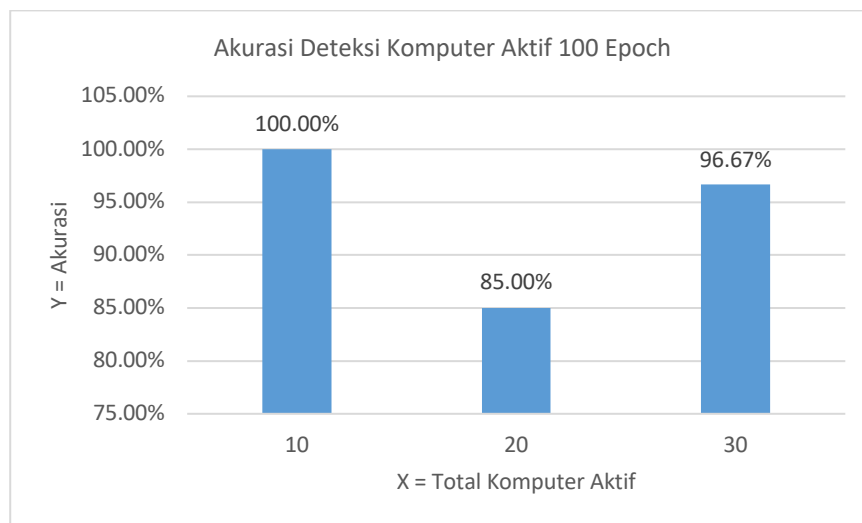
Kondisi	Komputer Aktif	Komputer Tidak Aktif	Terdeteksi Aktif	Terdeteksi Tidak Aktif	Tidak Terdeteksi	Terdeteksi Salah	Lama Deteksi
1	10	26	10	24	2	-	4 ms
2	20	16	17	13	6	-	2 ms
3	30	6	29	4	3	-	3 ms



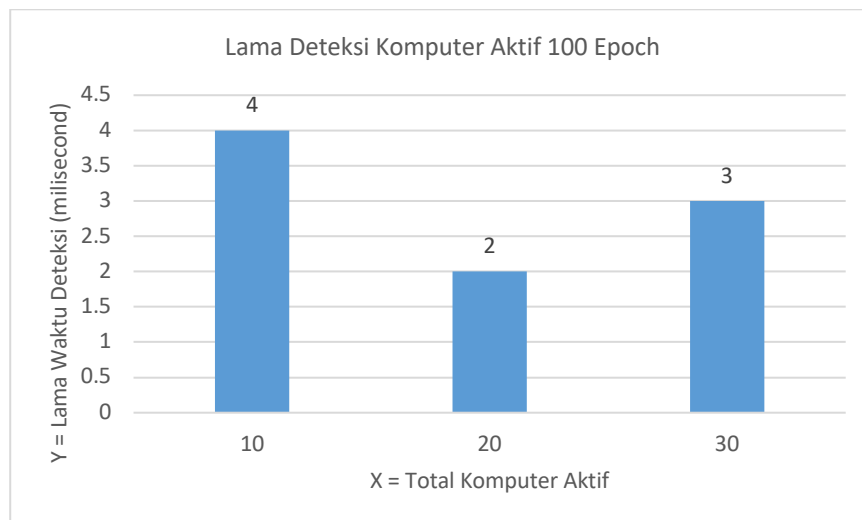
Gambar 9 Pengujian 100 Epoch



Nilai akurasi pada pengujian dengan 100 epoch menunjukkan hasil yang bervariasi pada setiap kondisi. Pada kondisi pertama, dengan total 10 komputer aktif yang seluruhnya berhasil terdeteksi, diperoleh nilai akurasi sebesar 100%. Pada kondisi kedua, dengan total 20 komputer aktif namun hanya 17 komputer yang berhasil terdeteksi, diperoleh nilai akurasi sebesar 85%. Sementara itu, pada kondisi ketiga dengan total 30 komputer aktif dan 29 komputer berhasil terdeteksi, nilai Akurasi yang dicapai adalah 96,67%. Visualisasi nilai akurasi dari ketiga kondisi tersebut dapat dilihat pada Gambar 10. Selanjutnya, waktu deteksi juga berbeda pada setiap kondisi pengujian. Pada kondisi pertama, dengan total 10 komputer aktif, waktu deteksi yang dibutuhkan adalah 4 ms. Pada kondisi kedua, dengan 20 komputer aktif, waktu deteksi tercatat 2 ms. Sedangkan pada kondisi ketiga, dengan 30 komputer aktif, waktu deteksi yang dicatat adalah 3 ms. Perbandingan lama deteksi pada ketiga kondisi tersebut dapat dilihat pada Gambar 11.



Gambar 10 Diagram Akurasi Deteksi Komputer Aktif 100 Epoch



Gambar 11 Diagram Lama Deteksi Komputer Aktif 100 Epoch

3.2 Pengujian 150 Epoch

Pada pengujian dengan 150 epoch, hasil pengujian dari ketiga kondisi dapat dilihat pada Tabel 3 dan Gambar 12. Pada kondisi pertama, jumlah komputer aktif sebanyak 10 unit, seluruhnya berhasil terdeteksi dengan benar. Jumlah komputer tidak aktif sebanyak 26 unit, dengan 25 unit



terdeteksi benar dan 1 unit terdeteksi salah. Pada kondisi kedua, jumlah komputer aktif sebanyak 20 unit, dengan 19 unit berhasil terdeteksi dan 1 unit tidak terdeteksi. Jumlah komputer tidak aktif sebanyak 16 unit, dengan 14 unit terdeteksi benar, 1 unit terdeteksi salah, dan 2 unit tidak terdeteksi. Sementara itu, pada kondisi ketiga, jumlah komputer aktif sebanyak 30 unit, seluruhnya berhasil terdeteksi dengan benar. Jumlah komputer tidak aktif sebanyak 6 unit, dan seluruhnya terdeteksi benar.

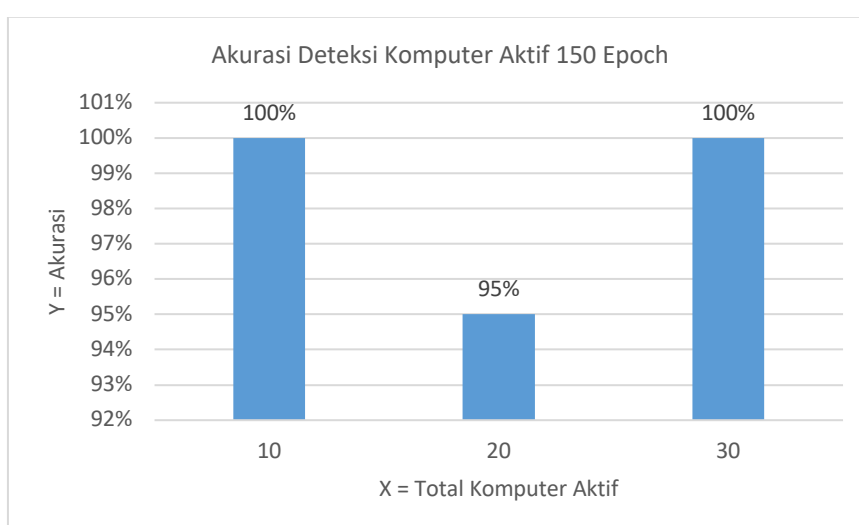
Tabel 3 Pengujian 150 Epoch

Kondisi	Komputer Aktif	Komputer Tidak Aktif	Terdeteksi Aktif	Terdeteksi Tidak Aktif	Tidak Terdeteksi	Terdeteksi Salah	Lama Deteksi
1	10	26	10	25	-	1	5 ms
2	20	16	19	14	2	1	7 ms
3	30	6	30	6	-	-	3 ms



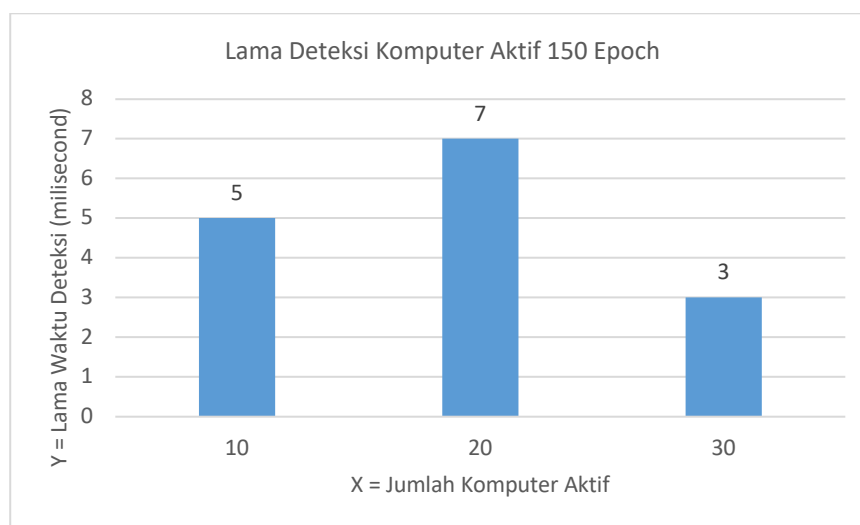
Gambar 12 Pengujian 150 Epoch

Nilai akurasi pada pengujian dengan 150 epoch menunjukkan hasil yang sangat baik. Pada kondisi pertama, dengan total 10 komputer aktif yang seluruhnya berhasil terdeteksi, diperoleh nilai akurasi sebesar 100%. Pada kondisi kedua, dengan total 20 komputer aktif namun hanya 19 komputer yang berhasil terdeteksi, diperoleh nilai akurasi sebesar 95%. Sementara itu, pada kondisi ketiga dengan total 30 komputer aktif yang seluruhnya berhasil terdeteksi, nilai akurasi mencapai 100%. Visualisasi nilai akurasi dari ketiga kondisi tersebut dapat dilihat pada Gambar 13. Selanjutnya, waktu deteksi juga berbeda pada setiap kondisi pengujian. Pada kondisi pertama, dengan total 10 komputer aktif, waktu deteksi yang dibutuhkan adalah 5 ms. Pada kondisi kedua, dengan 20 komputer aktif, waktu deteksi tercatat 7 ms. Sedangkan pada kondisi ketiga, dengan 30 komputer aktif, waktu deteksi yang dicatat adalah 3 ms. Perbandingan lama deteksi pada ketiga kondisi tersebut dapat dilihat pada Gambar 14.



Gambar 13 Diagram Akurasi Deteksi Komputer Aktif 150 Epoch





Gambar 14 Diagram Lama Deteksi Komputer Aktif 150 Epoch

3.3 Pengujian 200 Epoch

Pada pengujian dengan 200 epoch, hasil pengujian dari ketiga kondisi dapat dilihat pada Tabel 4 dan Gambar 15. Pada kondisi pertama, jumlah komputer aktif sebanyak 10 unit, seluruhnya berhasil terdeteksi dengan benar. Jumlah komputer tidak aktif sebanyak 26 unit, dengan 21 unit terdeteksi benar dan 5 unit tidak terdeteksi. Pada kondisi kedua, jumlah komputer aktif sebanyak 20 unit, dengan 19 unit berhasil terdeteksi dan 1 unit tidak terdeteksi. Jumlah komputer tidak aktif sebanyak 16 unit, dengan 15 unit terdeteksi benar dan 1 unit tidak terdeteksi. Sementara itu, pada kondisi ketiga, jumlah komputer aktif sebanyak 30 unit, dengan 27 unit berhasil terdeteksi dan 3 unit tidak terdeteksi. Jumlah komputer tidak aktif sebanyak 6 unit, dengan 5 unit terdeteksi benar dan 1 unit tidak terdeteksi.

Tabel 4 Pengujian 200 Epoch

Kondisi	Komputer Aktif	Komputer Tidak Aktif	Terdeteksi Aktif	Terdeteksi Tidak Aktif	Tidak Terdeteksi	Terdeteksi Salah	Lama Deteksi
1	10	26	10	21	5	-	2 ms
2	20	16	19	15	2	-	6 ms
3	30	6	27	5	4	-	4 ms

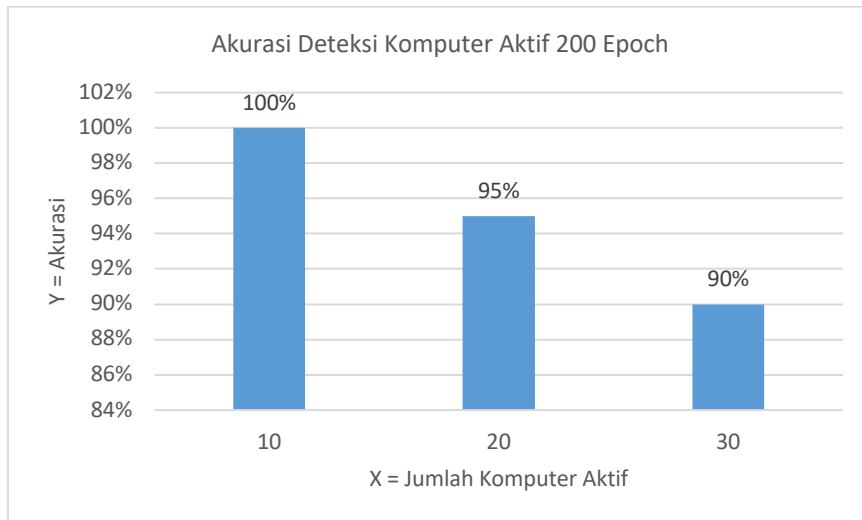


Gambar 15 Pengujian 200 Epoch

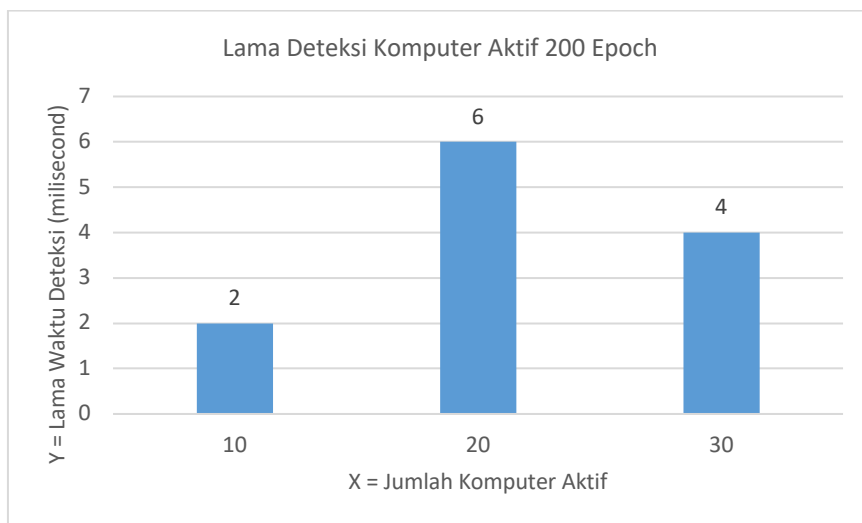
Nilai akurasi pada pengujian dengan 200 epoch juga menunjukkan hasil yang beragam pada setiap kondisi. Pada kondisi pertama, dengan total 10 komputer aktif yang seluruhnya berhasil terdeteksi, diperoleh nilai akurasi sebesar 100%. Pada kondisi kedua, dengan total 20 komputer aktif namun hanya 19 komputer yang berhasil terdeteksi, diperoleh nilai akurasi sebesar 95%. Sementara itu, pada kondisi ketiga dengan total 30 komputer aktif dan hanya 27 komputer berhasil terdeteksi, nilai akurasi yang dicapai adalah 90%. Visualisasi nilai akurasi dari ketiga kondisi tersebut dapat dilihat pada Gambar 16. Selanjutnya, waktu deteksi juga bervariasi pada



setiap kondisi. Pada kondisi pertama, dengan total 10 komputer aktif, waktu deteksi yang dibutuhkan adalah 2 ms. Pada kondisi kedua, dengan 20 komputer aktif, waktu deteksi tercatat 6 ms. Sedangkan pada kondisi ketiga, dengan 30 komputer aktif, waktu deteksi yang dicatat adalah 4 ms. Perbandingan lama deteksi pada ketiga kondisi tersebut dapat dilihat pada Gambar 17.



Gambar 16 Diagram Akurasi Deteksi Komputer Aktif 200 Epoch



Gambar 17 Diagram Lama Deteksi Komputer Aktif 200 Epoch

3.4 Hasil Pengujian

Hasil pengujian model YOLO pada 100, 150, dan 200 epoch menunjukkan bahwa performa model berbeda-beda sesuai jumlah komputer aktif. Untuk 10 komputer aktif, model 200 epoch memberikan akurasi tertinggi 100% dengan waktu deteksi tercepat 2 ms. Pada 20 komputer aktif, model 200 epoch juga unggul dengan akurasi 95%, meskipun waktu deteksi sedikit lebih lama (6 ms). Untuk 30 komputer aktif, model 100 epoch dipilih karena memberikan kombinasi terbaik antara akurasi (96,67%) dan waktu deteksi (3 ms), meskipun model 150 epoch mampu mencapai akurasi 100%, terdapat beberapa komputer yang salah atau tidak terdeteksi. Secara keseluruhan, hasil pengujian menekankan *trade-off* antara jumlah epoch, akurasi, dan kecepatan deteksi. Model dengan epoch lebih tinggi cenderung lebih akurat untuk jumlah komputer yang



lebih sedikit, sedangkan untuk jumlah komputer yang lebih banyak, model dengan epoch lebih rendah menawarkan keseimbangan optimal antara akurasi dan kecepatan.

4. KESIMPULAN

Berdasarkan pengujian di salah satu laboratorium komputer dengan menggunakan beberapa model YOLO, diperoleh kesimpulan sebagai berikut. Untuk pengujian dengan jumlah komputer aktif 10 unit, dipilih model 200 epoch yang memiliki akurasi 100% dalam mengenali komputer aktif dengan waktu 2 ms. Pengujian dengan jumlah komputer aktif 20 unit, dipilih model 200 epoch yang memiliki akurasi 95% dalam mengenali komputer aktif dengan waktu 6 ms. Pengujian dengan jumlah komputer aktif 30 unit, model 150 epoch memiliki akurasi tertinggi 100% dalam mengenali komputer aktif meskipun masih ada beberapa komputer terdeteksi salah. Oleh karena itu, dipilih model 100 epoch dengan akurasi 96,67% dengan waktu 3 ms sebagai model terbaik untuk kondisi ini.

DAFTAR PUSTAKA

- Alberto, J., & Hermanto, D. (2023). Bird Species Classification Using CNN Method and ResNet-50 Architecture. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 10(3), 34–36. <https://doi.org/10.35957/jatisi.v10i3.4558>
- Alfarizi, D. N., Pangestu, R. A., Aditya, D., Setiawan, M. A., & Rosyani, P. (2023). Penggunaan Metode YOLO pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis. *Jurnal Artificial Intelligent dan Sistem Penunjang Keputusan*, 1(1), 54–63. <https://jurnalmahasiswa.com/index.php/aidanspk/article/view/144>
- Ali, M. L., & Zhang, Z. (2024). The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection. *Computers*, 13(12), Article ID: 336. <https://doi.org/10.3390/computers13120336>
- Amin, R. Al, & Obermaisser, R. (2025). Real-Time Object Detection and Classification Using YOLO for Edge FPGAs. *ArXiv Preprint*. <http://arxiv.org/abs/2507.18174>
- Betti, A., & Tucci, M. (2023). YOLO-S: A Lightweight and Accurate YOLO-like Network for Small Target Detection in Aerial Imagery. *Sensors*, 23(4), Article ID: 1865. <https://doi.org/10.3390/s23041865>
- Erbani, J., Portier, P.-É., Egyed-Zsigmond, E., & Nurbakova, D. (2024). Confusion Matrices: A Unified Theory. *IEEE Access*, 12, 181372–181419. <https://doi.org/10.1109/ACCESS.2024.3507199>
- Gupta, C., Gill, N. S., Gulia, P., Kumar, A., Karamti, H., & Moges, D. M. (2025). An Optimized YOLO NAS Based Framework for Realtime Object Detection. *Scientific Reports*, 15(1), Article ID: 32903. <https://doi.org/10.1038/s41598-025-17919-w>
- Hardi, N., & Sundari, J. (2023). Pengenalan Telapak Tangan Menggunakan Convolutionall Neural Network (CNN). *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 4(1), 10–15. <https://doi.org/10.31294/reputasi.v4i1.1951>
- Hutauruk, J. S. W., Matulatan, T., & Hayaty, N. (2020). Deteksi Kendaraan Secara Real Time Menggunakan Metode YOLO Berbasis Android. *Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan*, 9(1), 8–14. <https://doi.org/10.31629/sustainable.v9i1.1401>
- Khairunnas, K., Yuniarno, E. M., & Zaini, A. (2021). Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot. *Jurnal Teknik ITS*, 10(1), A50–A55. <https://doi.org/10.12962/j23373539.v10i1.61622>
- Laia, F. H., Rosnelly, R., Naswar, A., Buulolo, K., & Lase, M. C. (2023). Deteksi Pengenalan Wajah Orang Berbasis AI Computer Vision. *Jurnal Teknologi Informasi Mura*, 15(1), 62–72. <https://jurnal.univbinainsan.ac.id/index.php/jti/article/view/2024>
- Luthfi, A. (2021). Pendeteksi Senjata Berbahaya pada Percobaan Tindakan Kriminal dengan Menggunakan Metode YOLO (You Only Look Once) [Universitas Islam Negeri Sultan Syarif Kasim Riau]. In *Pekanbaru*. <https://repository.uin-suska.ac.id/41304/>
- Monalia, M., Asfiyanti, N. A., & Putri, S. E. (2022). Computers and Information Technology as a Source of Learning Media for Elementary School Teachers. *International Journal of Natural Science and Engineering*, 5(3), 96–103. <https://doi.org/10.23887/ijnse.v5i3.41862>



- Mulyana, D. I., & Rofik, M. A. (2022). Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan di Indonesia Menggunakan Metode YOLOV5. *Jurnal Pendidikan Tambusai*, 6(3), 13971–13982. <https://doi.org/10.31004/jptam.v6i3.4825>
- Rahayu, W. I., Prianto, C., & Novia, E. A. (2021). Perbandingan Algoritma K-Means dan Naïve Bayes untuk Memprediksi Prioritas Pembayaran Tagihan Rumah Sakit Berdasarkan Tingkat Kepentingan pada PT. Pertamina (Persero). *Jurnal Teknik Informatika*, 13(2), 1–8. <https://ejurnal.ulbi.ac.id/index.php/informatika/article/view/1383>
- Raup, A., Ridwan, W., Khoeriyah, Y., Supiana, S., & Zaqiah, Q. Y. (2022). Deep Learning dan Penerapannya dalam Pembelajaran. *JlIP - Jurnal Ilmiah Ilmu Pendidikan*, 5(9), 3258–3267. <https://doi.org/10.54371/jiip.v5i9.805>
- Romadloni, P., Adhi Kusuma, B. A., & Baihaqi, W. M. (2022). Komparasi Metode Pembelajaran Mesin untuk Implementasi Pengambilan Keputusan dalam Menentukan Promosi Jabatan Karyawan. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), 622–628. <https://doi.org/10.36040/jati.v6i2.5238>
- Sadrawi, M., Fugaha, D. R., Heerlie, D. M., Lorell, J., Gautama, N. R. P., & Aminuddin, M. Z. (2023). Artificial Intelligence Based Brain Tumor Localization Using YOLOv5. *Indonesian Journal of Life Sciences*, 5(1), 1–9. <https://doi.org/10.54250/ijls.v5i01.176>
- Saputra, D. H., Imran, B., & Juhartini. (2023). Object Detection untuk Mendeteksi Citra Buah-Buahan Menggunakan Metode YOLO. *Jurnal Kecerdasan Buatan dan Teknologi Informasi*, 2(2), 70–80. <https://doi.org/10.69916/jkbt.v2i2.18>
- Yaseen, M. (2024). What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. *ArXiv Preprint*. <http://arxiv.org/abs/2408.15857>

