Jurnal Riset Aplikasi Ilmu Data dan Sistem 1 (2), 149–164 DOI: 0000
Oktober 2026

Deteksi dan Penghitungan Otomatis Ayam Broiler pada Kandang Menggunakan YOLOv8 dan YOLOv9 Berbasis Rekaman CCTV

Hendra Cahyono[1], Fadhilah Qalbi Annisa[2,*], Elly Matul Imah[2]

[1]*Program Studi Sarjana Sains Data, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Surabaya, Indonesia*

[2]*Program Studi Sarjana Kecerdasan Artifisial, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Surabaya, Indonesia*

Abstrak

Menghitung jumlah ayam broiler di kandang secara manual sulit dilakukan. Kesulitan ini disebabkan oleh beberapa hal, antara lain ayam yang berhimpitan, ayam yang tertutup ayam lain, dan pencahayaan rekaman *Closed-Circuit Television* (CCTV) yang berubah-ubah. Penelitian ini membandingkan dua model deteksi objek satu tahap, yaitu YOLOv8 dan YOLOv9 (*You Only Look Once*), untuk mendeteksi dan menghitung ayam broiler dari rekaman CCTV kandang. Seluruh ayam dianotasi sebagai satu kelas. *Dataset* berisi 220 *frame* dari 11 sudut kamera dan dibagi menurut sudut kamera menjadi data latih, validasi, dan uji dengan rasio 7:2:2 (7, 2, dan 2 sudut). Kedua model dilatih dengan konfigurasi yang sama. Kinerjanya dinilai memakai *precision*, *recall*, *mean Average Precision* (mAP), *Mean Intersection over Union* (IoU), *Mean Absolute Error* (MAE), dan *Frames Per Second* (FPS). Pada data uji, YOLOv9 mencapai mAP50 0,873, *recall* 0,794, dan *Mean* IoU 0,793, sedangkan YOLOv8 mencapai 0,871, 0,765, dan 0,786. *Precision* lebih tinggi pada YOLOv8, yaitu 0,850 dibanding 0,832. Pada ambang *confidence* 0,25, MAE YOLOv9 sebesar 21,7 ekor per *frame* dan MAE YOLOv8 sebesar 28,1 ekor per *frame*. Selisih ini hampir hilang pada ambang 0,40 yang dipilih dari data validasi. Pada pengukuran waktu terkontrol, YOLOv8 memproses 58,2 *frame* per detik dan YOLOv9 memproses 52,6 *frame* per detik. Melalui *Simple Additive Weighting* (SAW) atas delapan kriteria, YOLOv9 memperoleh skor 0,741 dan diusulkan sebagai model untuk memantau populasi ayam broiler.

Kata Kunci *ayam broiler; efisiensi komputasi; object detection; penghitungan objek; YOLO*

Abstract

Counting broiler chickens in a poultry house by hand is difficult. The difficulty arises from several factors, including crowded birds, birds hidden behind other birds, and changing lighting in Closed-Circuit Television (CCTV) footage. This study compares two single-stage object detectors, YOLOv8 and YOLOv9 (You Only Look Once), for detecting and counting broilers in CCTV footage of a poultry house. All chickens were annotated as a single class. The dataset contains 220 frames from 11 camera angles. It was split by camera angle into training, validation, and test sets at a 7:2:2 ratio (7, 2, and 2 angles). Both models were trained with the same configuration. They were evaluated with precision, recall, mean Average Precision (mAP), Mean Intersection over Union (IoU), Mean Absolute Error (MAE), and Frames Per Second (FPS). On the test set, YOLOv9 reached an mAP50 of 0.873, a recall of 0.794, and a Mean IoU of 0.793, against

*Corresponding author. Email: fadhilahannisa@unesa.ac.id or ellymatul@unesa.ac.id.

Received 26 Juli 2026; Revised 30 September 2026; Accepted 10 Oktober 2026

0.871, 0.765, and 0.786 for YOLOv8. Precision was higher for YOLOv8, at 0.850 against 0.832. At a confidence threshold of 0.25, the MAE was 21.7 birds per frame for YOLOv9 and 28.1 for YOLOv8. This gap nearly disappeared at a threshold of 0.40 chosen on the validation set. In a controlled timing test, YOLOv8 processed 58.2 frames per second and YOLOv9 processed 52.6. Using Simple Additive Weighting (SAW) over eight criteria, YOLOv9 obtained a score of 0.741 and is proposed as the model for monitoring broiler populations.

Keywords *broiler chicken; computational efficiency; object counting; object detection; YOLO*

1 Pendahuluan

OECD dan FAO (2025) memproyeksikan daging unggas sebagai penyumbang utama konsumsi protein daging hingga 2033. Kementerian Pertanian (2024) mencatat konsumsi daging ayam ras pedaging per kapita di Indonesia terus naik. Agar pasokan tetap stabil, konsistensi produksi di tingkat kandang perlu dijaga lewat pemantauan yang terukur (Astill dkk., 2020; Van Limbergen dkk., 2020). Kandang broiler komersial dihuni populasi yang sangat padat, sehingga pemeriksaan manual secara berkala sulit dilakukan dengan cepat dan konsisten (Guo dkk., 2023). Konsep *precision livestock farming* menawarkan jalan keluar dengan menempatkan pemantauan kontinu berbasis data sebagai dasar pengambilan keputusan (Berckmans, 2017). Kamera pengawas atau *Closed-Circuit Television* (CCTV) menjadi salah satu instrumen yang memungkinkan pengamatan tanpa mengganggu ternak (Schillings dkk., 2021).

Menerapkan visi komputer di kandang nyata tidak semudah pada lingkungan terkendali. Okinda dkk. (2020) mencatat tiga kendala yang berulang, yaitu pencahayaan yang berubah-ubah, objek yang saling menutupi, serta minimnya *dataset* unggas berlabel. Studi deteksi broiler berbasis *You Only Look Once* (YOLO) menambahkan persoalan *crowded scene*. Ayam berukuran relatif kecil, jumlahnya banyak, dan batas tubuhnya sering menyatu dengan latar kandang (Guo dkk., 2023; Xu dkk., 2023). Kondisi semacam ini melahirkan dua jenis kesalahan yang berlawanan arah. Satu ekor ayam bisa memperoleh lebih dari satu *bounding box* hingga jumlahnya membengkak. Sebaliknya, ayam bisa tidak terdeteksi sama sekali hingga hasil hitung menjadi kurang. Keduanya merusak estimasi populasi. Model yang dipilih perlu diuji dari konsistensi hasil hitungnya, di samping kemampuannya mengenali objek.

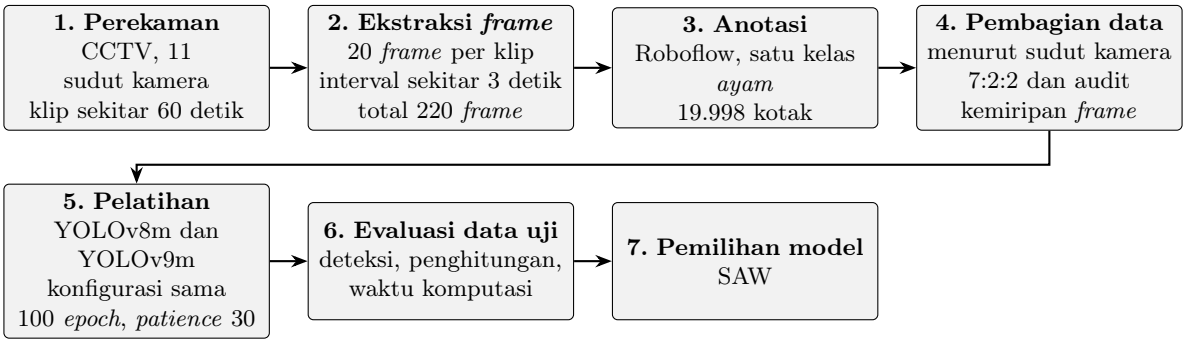
Dari tinjauan literatur tersebut, penelitian ini mengidentifikasi empat celah penelitian (*research gap*). Pertama, komparasi varian YOLO generasi baru sudah cukup banyak dilakukan, termasuk pada lingkungan peternakan. Namun, sebagian masih bertumpu pada data sintetis (Bumbálek dkk., 2025). Padahal literatur *sim-to-real* memperlihatkan bahwa kinerja model dapat merosot ketika dipindahkan ke data lapangan (Rüter dkk., 2024). Kedua, sebagian besar studi berhenti pada akurasi deteksi, misalnya *mean Average Precision* (mAP), tanpa menghubungkannya dengan ketepatan penghitungan populasi maupun kecepatan pemrosesan. Padahal ketiga hal itu tidak selalu bergerak seiring (Huang dkk., 2017). Ketiga, penelitian deteksi ayam broiler umumnya masih memakai anotasi multi-kelas yang memisahkan kondisi visual ayam. Akibatnya, kemampuan deteksi murni untuk keperluan menghitung populasi belum banyak diuji secara terpisah. Keempat, keputusan memilih model kerap disandarkan pada satu atau dua metrik saja, tanpa kerangka yang menilai akurasi, ketepatan hitung, dan efisiensi secara bersamaan (Peng dkk., 2011).

Berdasarkan celah tersebut, penelitian ini membandingkan YOLOv8 dan YOLOv9 untuk mendeteksi dan menghitung ayam broiler dari rekaman CCTV kandang padat. Anotasi memakai satu kelas *ayam* untuk seluruh ayam tanpa membedakan kondisinya. YOLOv8 dipakai seba-

gai *baseline* karena sudah luas diadopsi dan dokumentasinya lengkap. YOLOv9 dipilih karena merupakan generasi yang lebih baru (Wang dkk., 2024). Metrik evaluasi yang dipakai adalah mAP dan *Mean Intersection over Union* (IoU) untuk kualitas deteksi. Ketepatan penghitungan populasi diukur dengan metrik *Mean Absolute Error* (MAE), dan efisiensi komputasi diukur dengan metrik *Frames Per Second* (FPS). Model usulan ditentukan lewat skema pembobotan yang menggabungkan seluruh metrik tersebut. Sejauh penelusuran penulis, kajian yang menautkan ketiga aspek itu secara serentak pada peternakan unggas masih terbatas.

2 Metode Penelitian

Alur penelitian terdiri atas tujuh tahap yang diringkas pada Gambar 1. Rekaman CCTV dari 11 sudut kamera diubah menjadi 220 *frame* yang dianotasi, dibagi menurut sudut kamera, lalu dipakai untuk melatih dan mengevaluasi kedua model. Hasil evaluasi digabungkan dengan *Simple Additive Weighting* (SAW) untuk memilih model usulan.



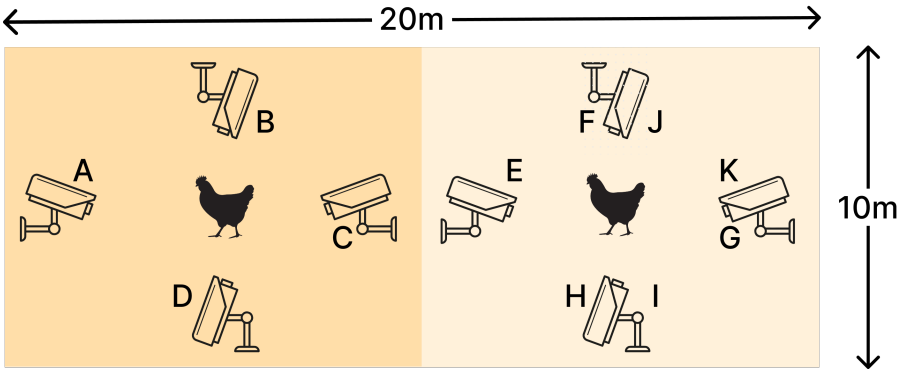
Gambar 1. Alur penelitian.

2.1 Akuisisi dan Anotasi *Dataset*

Data yang dipakai berupa rekaman CCTV satu kandang tertutup ayam broiler, dikumpulkan secara mandiri dan *offline*. Perekaman menggunakan Xiaomi Smart Camera C300 (model XMC01) beresolusi 2304×1296 piksel dengan kompresi H.265 (Xiaomi, 2025). Kamera dipasang pada ketinggian sekitar 60 cm dari lantai kandang supaya ayam tampak cukup besar dalam bidang pandang dan anotasi lebih konsisten. Pengambilan gambar dilakukan dari sebelas sudut kamera (Gambar 2) ketika ternak berumur 20–27 hari. Sudut disebar agar seluruh sisi kandang terjangkau. Ragam sudut dan rentang umur ini dipilih dengan sengaja. Tujuannya agar *dataset* memuat beragam orientasi tubuh, jarak ke kamera, tingkat tumpang tindih antar-ayam, dan variasi kepadatan yang memang dijumpai di kandang. Setiap rekaman berdurasi sekitar satu menit pada 20 *frame* per detik. Dari tiap rekaman diambil 20 *frame* dengan interval yang sama, sekitar tiga detik. Total *frame* pada seluruh *dataset* adalah 220.

Proses anotasi dikerjakan manual di Roboflow memakai *bounding box*. Seluruh ayam diberi satu label yang sama, yaitu *ayam*, tanpa membedakan kondisinya, sehingga *dataset* hanya memiliki satu kelas. Jumlah anotasi seluruhnya 19.998, dengan sebaran per *subset* pada Tabel 1. Kelas tunggal dipilih karena penelitian ini berfokus pada lokalisasi dan penghitungan. Ayam yang terpotong tepi *frame* atau tertutup sebagian tetap dianotasi selama masih dapat dikenali, supaya *dataset* mewakili kondisi berkerumun. Ayam yang terlalu jauh atau buram tidak diberi label

agar tidak membingungkan model saat belajar. Berkas anotasi diekspor ke format YOLO. Setiap citra memiliki satu berkas `.txt` yang berisi kelas dan koordinat *bounding box* ternormalisasi (`x_center`, `y_center`, `width`, `height`).



Gambar 2. Denah kandang dan posisi kesebelas sudut kamera (A sampai K).

2.2 Pembagian Data dan Pencegahan Kebocoran Data

Dataset dibagi menurut sudut kamera, bukan menurut *frame*. Tujuh sudut kamera (A–D dan I–K) menjadi data latih, dua sudut (E dan F) menjadi data validasi, dan dua sudut (G dan H) menjadi data uji. Rasio 7:2:2 merujuk pada jumlah sudut kamera: 7, 2, dan 2 dari 11 sudut. Setiap sudut menyumbang 20 *frame*, sehingga jumlah *frame* menjadi 140, 40, dan 40, atau 140:40:40 yang setara dengan 7:2:2. Kedua model memakai pembagian yang sama. Tabel 1 merinci jumlah *frame*, jumlah anotasi, dan sebaran jumlah ayam per *frame* pada tiap *subset*.

Tabel 1. Karakteristik *dataset* pada tiap *subset* (satu kelas *ayam*).

Subset	Sudut kamera	Waktu rekam	*Frame*	Anotasi	Ayam per *frame*
Latih	A–D, I–K (7)	28 Sep–5 Okt 2025	140	13.050	93,2 ± 33,4 (51–194)
Validasi	E, F (2)	24–25 Jan 2026	40	3.445	86,1 ± 2,0 (82–90)
Uji	G, H (2)	24–25 Jan 2026	40	3.503	87,6 ± 3,4 (82–99)
Total	11	—	220	19.998	90,9 ± 26,9 (51–194)

Catatan: waktu rekam mengikuti cap waktu pada video, dengan jam rekam 00.00–16.46 pada data latih serta 22.37–01.54 pada data validasi dan data uji. Ayam per *frame* berupa rerata ± simpangan baku dengan rentang dalam tanda kurung.

Kepadatan ayam berbeda antar sudut kamera. Rerata jumlah ayam per *frame* pada tiap sudut berkisar dari 54 sampai 151 ekor untuk data latih. Untuk data validasi dan data uji, kisarannya 86 sampai 89 ekor. Ukuran *bounding box* serupa antar-*subset*. Median lebar dan tingginya sekitar 105×83 piksel pada data latih, 116×89 piksel pada data validasi, dan 114×79 piksel pada data uji, pada citra beresolusi 2304×1296 piksel.

Pembagian menurut sudut kamera dipilih untuk mencegah kebocoran data (*data leakage*) antar-*subset*. *Frame* dari satu rekaman sangat mirip satu sama lain. Bila pembagian dilakukan menurut *frame*, *frame* yang nyaris kembar dapat masuk ke data latih dan data uji sekaligus, dan hasil uji menjadi terlalu optimis. Pada pembagian menurut sudut kamera, seluruh *frame* dari satu rekaman selalu berada pada satu *subset*. Penempatan sudut ke tiap *subset* tidak diacak, dan dimaksudkan agar model tidak menghafal tempat. Sudut E sampai H menempati posisi yang

berbeda dari sudut latih A sampai D. Antar-*subset*, yang membedakan adalah sudut kamera, waktu perekaman, atau keduanya.

Kemiripan antar-*frame* juga diperiksa dengan *difference hash* (dHash) 256 bit dan korelasi silang ternormalisasi (*normalized cross-correlation*, NCC) pada citra abu-abu. Pada 12.800 pasangan *frame* lintas *subset*, jarak dHash terkecil adalah 94 bit dan NCC terbesar 0,35. Pada pasangan *frame* dalam satu rekaman, jarak dHash median adalah 22 bit dan NCC mencapai 0,997. Tidak ada *frame* yang nyaris kembar di seberang batas *subset*.

Dari sisi pelatihan, augmentasi hanya diterapkan pada data latih, sedangkan bobot terbaik dan ambang penghitungan 0,40 (Subbab 2.5) dipilih dari data validasi, bukan dari data uji. Seluruh data berasal dari satu kandang, sehingga ayam pada *subset* yang berbeda bisa saja individu yang sama. Hasil pada data uji belum menggambarkan kemampuan generalisasi ke kandang lain.

2.3 Arsitektur Model

YOLOv8 dan YOLOv9 sama-sama berdiri di atas kerangka YOLO modern. Kerangka ini terdiri atas *backbone* sebagai pengekstraksi fitur bertingkat, *neck* sebagai penggabung fitur lintas skala, dan *head* sebagai penghasil prediksi *bounding box* beserta labelnya (Ali dan Zhang, 2024). Keduanya menerapkan gagasan deteksi satu tahap yang diperkenalkan Redmon dkk. (2016). Gagasan itu menyatukan lokalisasi dan klasifikasi dalam satu proses sehingga cocok untuk memproses *frame* CCTV secara beruntun. YOLOv8 bekerja tanpa *anchor box* dan memakai *decoupled head* serta modul C2f pada *backbone* (Li dkk., 2025; Ultralytics, 2026).

YOLOv9 (Wang dkk., 2024) merupakan generasi setelah YOLOv8 dan tersedia pada pustaka Ultralytics yang sama. Tabel 2 merangkum karakteristik struktural kedua model yang dilaporkan Ultralytics saat inisialisasi. YOLOv9 memiliki 348 lapisan, lebih banyak daripada YOLOv8 yang memiliki 170 lapisan. Parameternya justru lebih sedikit, 20,16 juta dibanding 25,86 juta. Beban komputasinya hampir setara, 77,55 dibanding 79,07 *giga floating-point operations* (GFLOPs).

Tabel 2. Perbandingan karakteristik arsitektur YOLOv8 dan YOLOv9.

Model	*Pretrained*	Jumlah Layer	Total Parameter	GFLOPs
YOLOv8	yolov8m.pt	170	25.856.899	79,07
YOLOv9	yolov9m.pt	348	20.159.043	77,55

2.4 Konfigurasi Pelatihan

Kedua model varian *medium* dilatih dari bobot *pretrained* pada *dataset Common Objects in Context* (COCO), yaitu `yolov8m.pt` dan `yolov9m.pt`, supaya pelatihan pada *dataset* yang terbatas tetap stabil. Seluruh pengaturan disamakan untuk menjaga keseimbangan perbandingan kedua model. Ukuran citra masukan adalah 640×640 piksel, dengan *batch size* 16, jumlah *epoch* maksimum 100, dan *seed* 0. *Early stopping* menghentikan pelatihan bila mAP50–95 validasi tidak membaik selama 30 *epoch* (*patience* 30). *Optimizer* diatur dengan mode *auto*, yang memilih AdamW. Augmentasi yang aktif adalah *mosaic*, skala acak, translasi, dan pembalikan horizontal, sedangkan augmentasi warna dimatikan. Augmentasi hanya bekerja pada data latih.

Validasi berjalan pada tiap *epoch*, dan bobot dengan mAP50–95 validasi tertinggi disimpan sebagai `best.pt`. Pelatihan dijalankan di Google Colaboratory dengan pustaka Ultralytics versi 8.4.102 pada *Graphics Processing Unit* (GPU) Tesla T4.

2.5 Metrik Evaluasi

Penilaian kinerja dibagi ke dalam empat kelompok. Kelompok pertama adalah kualitas deteksi. Kelompok ini diukur dengan *precision*, *recall*, mAP50, mAP50–95, dan *Mean* IoU untuk menakar ketepatan klasifikasi sekaligus lokalisasi *bounding box* (Padilla dkk., 2021). Ketepatan penghitungan menjadi kelompok kedua dan diukur dengan MAE pada Persamaan (1). MAE adalah rata-rata selisih mutlak antara jumlah ayam hasil prediksi dan jumlah sebenarnya di seluruh *frame* uji, dalam satuan ekor per *frame* (Willmott dan Matsuura, 2005). Karena berupa rata-rata, nilai MAE dapat berupa desimal. Kelompok ketiga adalah efisiensi komputasi, yang diukur dengan waktu per *frame* dan FPS pada Persamaan (2). Kelompok keempat adalah penentuan model usulan dengan skema pembobotan multi-kriteria pada Subbab 3.5.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{1}$$

$$\mathrm{FPS} = \frac{N}{T} \tag{2}$$

Pada Persamaan (1), N menyatakan jumlah *frame* uji, y_i menyatakan jumlah ayam sebenarnya pada *frame* ke-i, dan $\hat{y}_i$ menyatakan jumlah ayam hasil prediksi pada *frame* yang sama. Pada Persamaan (2), N menyatakan banyaknya *frame* yang diproses dan T menyatakan total waktu pemrosesan ujung ke ujung dalam detik.

Perhitungan mAP memakai pengaturan baku mode validasi Ultralytics dengan ambang *confidence* 0,001 mengikuti protokol COCO, supaya kurva *precision-recall* terbentuk utuh. Jumlah ayam pada satu *frame* dihitung dari *bounding box* yang tersisa setelah *Non-Maximum Suppression* (NMS) dengan ambang IoU 0,7 dan *confidence* di atas ambang penghitungan. Ambang penghitungan yang dipakai adalah 0,25 seperti pada inferensi praktis. MAE juga dihitung pada ambang 0,40. Ambang ini dipilih sebagai ambang bersama yang memberi jumlah MAE validasi terendah pada kedua model dari rentang 0,05 sampai 0,95. Metode *tracking* sengaja tidak dilibatkan karena penelitian ini menyoroti kemampuan deteksi murni per *frame*.

Waktu komputasi diukur pada GPU Tesla T4 dengan *batch* 1, ukuran citra 640 piksel, ambang *confidence* 0,25, dan NMS-IoU 0,7. Seluruh 40 *frame* uji dimuat sekali ke memori, sehingga waktu baca citra tidak ikut terhitung. Setelah pemanasan 20 *frame*, kedua model diukur bergantian selama 10 putaran dengan 40 *frame* per putaran, dan urutan model dibalik pada putaran genap. Pengukuran diulang pada dua sesi Colab yang terpisah. Hasilnya dilaporkan sebagai rerata dan simpangan baku antar-putaran untuk waktu inferensi dan waktu ujung ke ujung, serta FPS yang dihitung dari waktu ujung ke ujung.

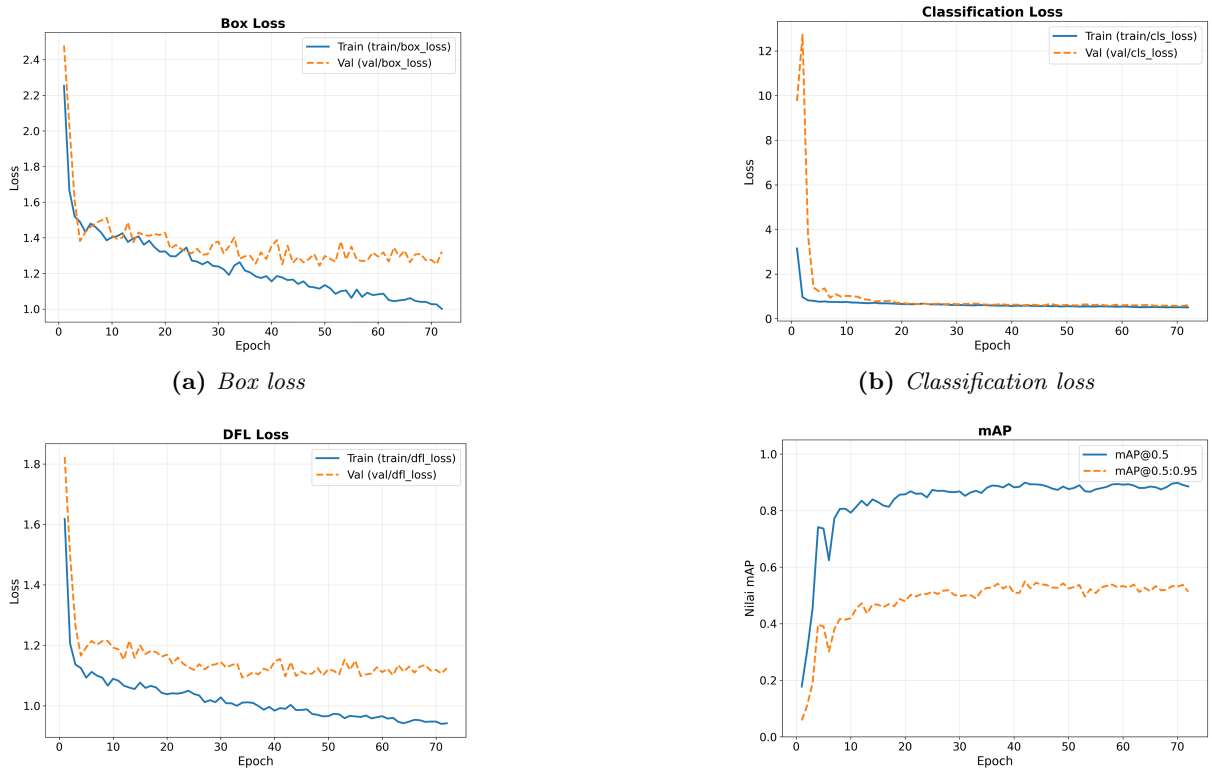
3 Hasil dan Pembahasan

3.1 Hasil Pelatihan Model

Jalannya pelatihan dipantau melalui kurva *box loss*, *classification loss*, *Distribution Focal Loss* (DFL), dan mAP pada data latih maupun validasi. *Box loss* dan DFL *loss* berkaitan dengan ketepatan lokalisasi *bounding box*, sedangkan *classification loss* berkaitan dengan kemampuan memisahkan ayam dari latar kandang. Kurva mAP tiap *epoch* dipakai untuk melihat perkembangan akurasi dan menentukan *checkpoint* terbaik. Pemantauan pada dua himpunan data sekaligus juga memudahkan pengenalan gejala *overfitting*, yaitu saat *loss* data latih terus turun tetapi *loss* data validasi justru naik.

Pada artikel ini, pelatihan disebut stabil bila *loss* validasi mendatar tanpa kenaikan yang berkelanjutan dan mAP validasi hanya berfluktuasi dalam rentang sempit. Konvergensi berarti tercapainya kondisi tersebut. Gambar 3 menampilkan keempat kurva pelatihan dan validasi untuk YOLOv8.

Pada YOLOv8, *box loss* (Gambar 3a) turun dari sekitar 2,1 di data latih dan 2,5 di data validasi menjadi 1,0–1,3 setelah *epoch* ke-50. *Classification loss* (Gambar 3b) sempat melonjak hingga mendekati 13 di data validasi pada *epoch* ke-2, lalu turun dan berada di bawah 1 mulai *epoch* ke-12. DFL *loss* (Gambar 3c) mengikuti pola serupa, naik ke sekitar 1,8 di awal sebelum turun ke kisaran 1,1–1,2. Kurva mAP (Gambar 3d) naik cepat pada *epoch* awal lalu mendatar di kisaran 0,85–0,90 untuk mAP@0,5 dan 0,45–0,55 untuk mAP@0,5:0,95. *Loss* data validasi turun searah dengan data latih lalu mendatar tanpa kenaikan yang berkelanjutan, sehingga tidak terlihat gejala *overfitting* yang berarti.

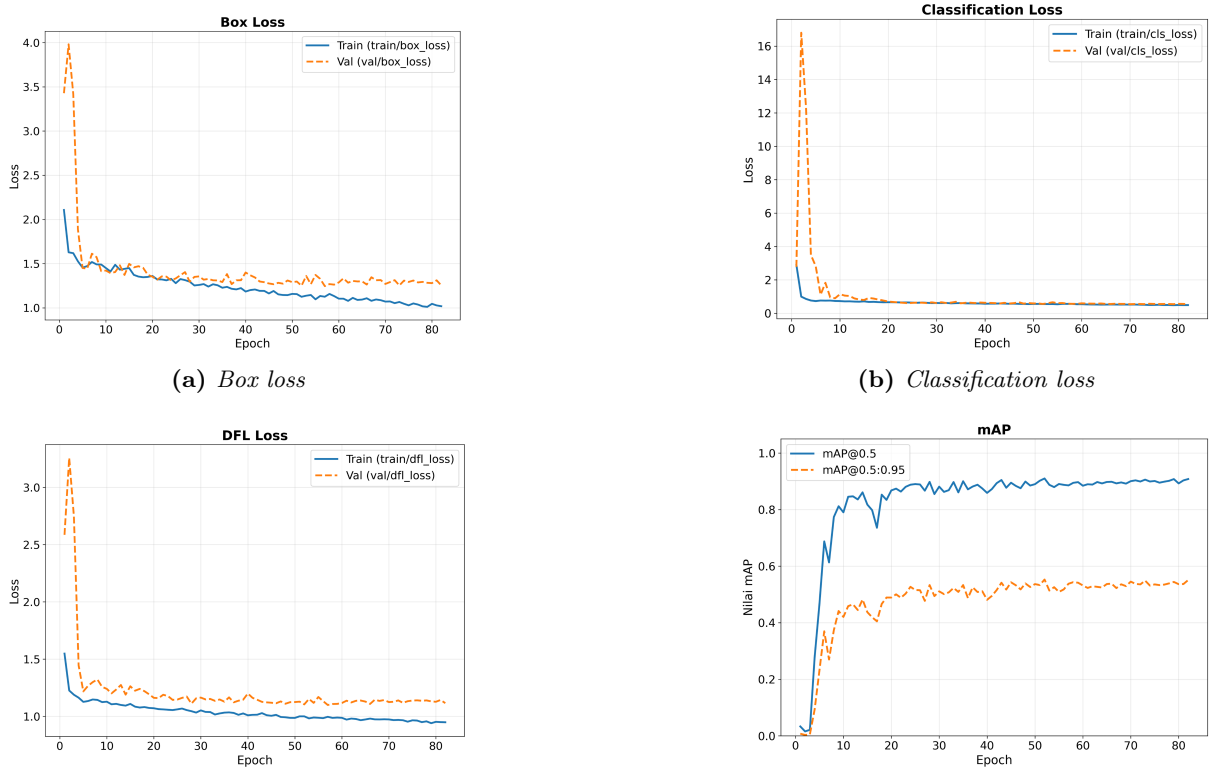


(a) *Box loss* **(b)** *Classification loss*

(c) DFL *loss* **(d)** mAP

Gambar 3. Kurva pelatihan dan validasi YOLOv8: (a) *box loss*, (b) *classification loss*, (c) DFL *loss*, dan (d) mAP.

Pola pelatihan YOLOv9 mirip dengan YOLOv8, hanya saja nilai awal beberapa *loss* lebih tinggi. Gambar 4 menyajikan keempat kurva pelatihan dan validasi untuk YOLOv9. *Box loss* (Gambar 4a) melonjak lebih tinggi di data validasi, mendekati 4,0 pada *epoch* ke-2, sebelum turun dan menetap di kisaran 1,1–1,3. *Classification loss* (Gambar 4b) sempat menyentuh angka mendekati 17 di awal pelatihan, lalu turun cepat dan berada di bawah 1 mulai *epoch* ke-13. DFL *loss* (Gambar 4c) menurun konsisten dari sekitar 3,2 menuju kisaran 1,1–1,2 pada data validasi. Kurva mAP (Gambar 4d) naik tajam di awal dan bertahan pada kisaran 0,85–0,90 untuk mAP@0,5 sampai *epoch* ke-82, sedangkan mAP@0,5:0,95 berada di kisaran 0,50–0,55.



(a) *Box loss* **(b)** *Classification loss*

(c) DFL *loss* **(d)** mAP

Gambar 4. Kurva pelatihan dan validasi YOLOv9: (a) *box loss*, (b) *classification loss*, (c) DFL *loss*, dan (d) mAP.

Setelah *epoch* ke-30, kedua model memenuhi kriteria stabil tersebut. mAP50–95 validasi berfluktuasi di sekitar 0,525 pada YOLOv8 dan 0,527 pada YOLOv9, dengan simpangan baku 0,014 dan 0,016. *Box loss* validasi mendatar di sekitar 1,30 dengan simpangan baku 0,04 dan 0,03. *Checkpoint* terbaik YOLOv8 dicapai pada *epoch* ke-42 dan YOLOv9 pada *epoch* ke-52. *Early stopping* menghentikan pelatihan tepat 30 *epoch* setelah *checkpoint* itu, pada *epoch* ke-72 untuk YOLOv8 dan *epoch* ke-82 untuk YOLOv9. Pada rentang stabil tersebut, *epoch* dengan mAP tertinggi sangat dipengaruhi fluktuasi acak, sehingga selisih sepuluh *epoch* antara kedua model tidak dapat dikaitkan dengan arsitektur.

Durasi pelatihan total adalah 9,9 menit untuk YOLOv8 dan 14,4 menit untuk YOLOv9 pada GPU Tesla T4. Selisihnya berasal dari sepuluh *epoch* tambahan dan waktu per *epoch* yang lebih lama, 10,5 detik dibanding 8,3 detik.

Tabel 3 merangkum performa validasi pada *checkpoint* terbaik masing-masing model. Nilai YOLOv9 lebih tinggi pada *precision*, *recall*, mAP50, dan mAP50–95. *Classification loss* validasi YOLOv9 juga lebih rendah, 0,558 dibanding 0,603. *Box loss* validasi keduanya nyaris sama, 1,251 untuk YOLOv8 dan 1,250 untuk YOLOv9. *DFL loss* validasi YOLOv9 sedikit lebih tinggi, 1,105 dibanding 1,097. *Precision* YOLOv9 pada data validasi adalah 0,872 dibanding 0,851, dan hasil ini dibandingkan lagi dengan data uji pada Subbab 3.2.

Tabel 3. Perbandingan performa validasi pada *checkpoint* terbaik (`best.pt`).

Metrik	YOLOv8	YOLOv9	Keterangan
Epoch terbaik	42	52	—
Box loss validasi	1,2509	1,2496	Lebih rendah lebih baik
Classification loss validasi	0,6032	0,5575	Lebih rendah lebih baik
DFL *loss* validasi	1,0973	1,1045	Lebih rendah lebih baik
Precision	0,8514	0,8724	Lebih tinggi lebih baik
Recall	0,8142	0,8254	Lebih tinggi lebih baik
mAP50	0,899	0,910	Lebih tinggi lebih baik
mAP50–95	0,550	0,553	Lebih tinggi lebih baik

3.2 Hasil Pengujian Deteksi Model

Tabel 4 dan Gambar 5 menampilkan hasil pada data uji. *Recall* YOLOv9 mencapai 0,794, sedangkan YOLOv8 mencapai 0,765. Perbandingan mAP50 adalah 0,873 untuk YOLOv9 dan 0,871 untuk YOLOv8, dan perbandingan mAP50–95 adalah 0,515 dan 0,504. *Mean* IoU YOLOv9 sebesar 0,793 dibanding 0,786 pada YOLOv8, dan F1-*score*-nya 0,812 dibanding 0,805. Hanya *precision* yang lebih tinggi pada YOLOv8, yaitu 0,850 dibanding 0,832. Hasil ini berbeda dari data validasi pada Tabel 3, tempat *precision* YOLOv9 lebih tinggi.

Selisih mAP50 dan mAP50–95 antara kedua model pada data uji lebih kecil daripada simpangan baku mAP validasi antar-*epoch*, sebesar 0,010–0,012 untuk mAP50 dan 0,014–0,016 untuk mAP50–95. Setiap model juga hanya dilatih sekali. Kualitas deteksi kedua model dinilai sebanding.

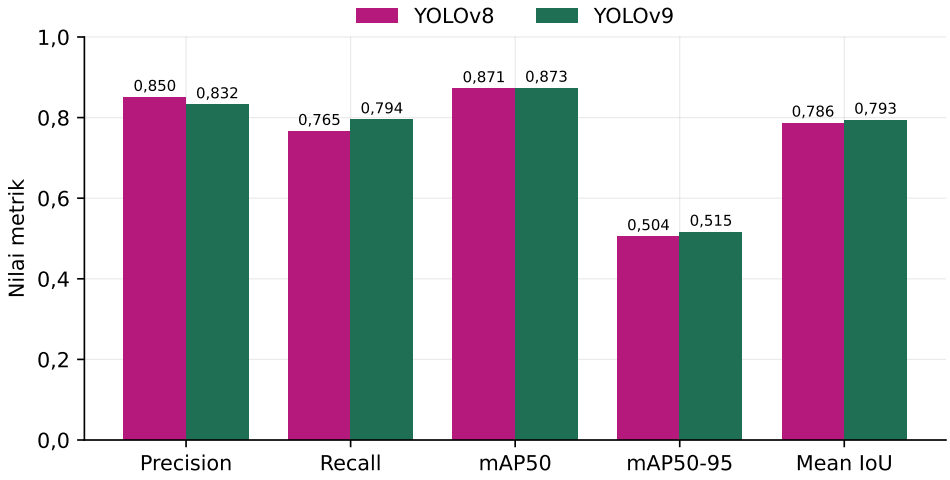
Tabel 4. Hasil pengujian deteksi dan penghitungan ayam broiler pada data uji.

Model	*Precision*	*Recall*	mAP50	mAP50–95	*Mean* IoU	F1	MAE (0,25)	MAE (0,40)
YOLOv8	0,850	0,765	0,871	0,504	0,786	0,805	28,1	8,9
YOLOv9	0,832	0,794	0,873	0,515	0,793	0,812	21,7	8,8

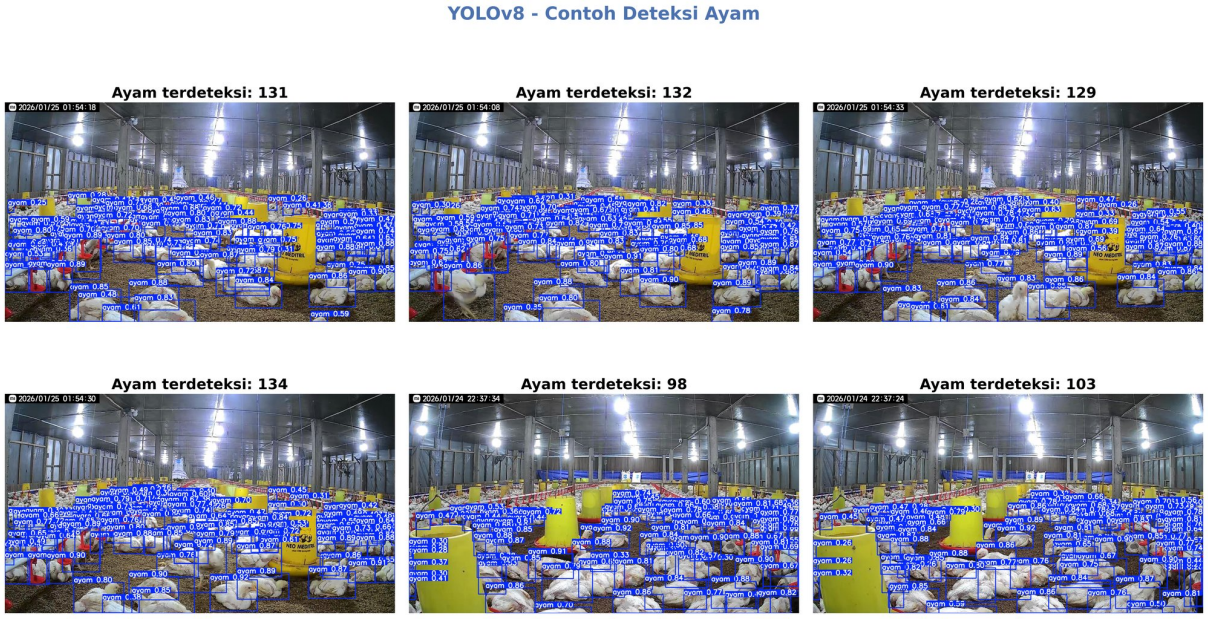
Catatan: seluruh metrik dihitung untuk kelas tunggal *ayam* pada data uji. MAE dalam ekor per *frame*, dengan ambang *confidence* 0,25 dan 0,40 pada tanda kurung.

Untuk melihat pola yang tidak tertangkap angka mAP, Gambar 6 dan 7 menampilkan contoh keluaran YOLOv8 dan YOLOv9 pada enam *frame* CCTV yang sama. *Frame* dipilih agar mewakili variasi kepadatan dan sudut pengambilan gambar. Pada keenam *frame* tersebut, kandang tampak dipadati ayam, pencahayaan buatan tidak merata, dan peralatan kandang seperti tempat pakan dan minum berpotensi dikira objek ayam. Jumlah total tertera pada judul tiap panel, dan kerapatan *bounding box* dapat diamati langsung.

Pada keenam *frame* itu YOLOv8 melaporkan berturut-turut 131, 132, 129, 134, 98, dan 103 ekor, sedangkan YOLOv9 melaporkan 114, 116, 116, 117, 98, dan 108 ekor. Jumlah YOLOv8 lebih besar pada empat *frame*, dengan jarak terlebar 17 ekor pada perbandingan 134 dan 117. Pada satu *frame* keduanya menghasilkan angka yang sama, 98 ekor, dan pada satu *frame* lainnya YOLOv8 justru lebih rendah, 103 dibanding 108. Selisih rata-rata jumlah deteksi antara kedua model pada keenam *frame* mencapai sekitar 9,7 ekor. Nilai ini jauh lebih besar daripada selisih metrik deteksinya, yang paling besar hanya sekitar 3 poin persentase. Metrik deteksi saja belum cukup untuk menilai kelayakan model pada tugas menghitung populasi (Huang dkk., 2017). MAE perlu ditinjau sebagai pelengkap.



Gambar 5. Perbandingan metrik deteksi ayam broiler antara YOLOv8 dan YOLOv9 pada data uji.

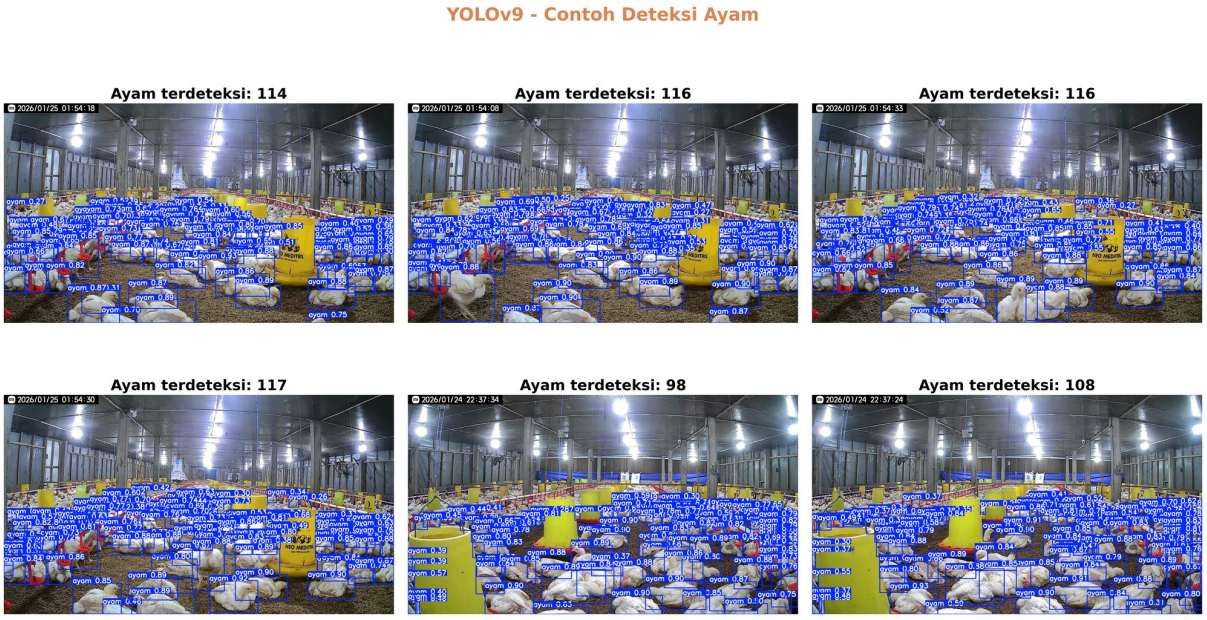


Gambar 6. Contoh hasil deteksi YOLOv8 pada enam *frame* CCTV kandang berkepadatan tinggi.

3.3 Hasil Penghitungan Populasi (MAE)

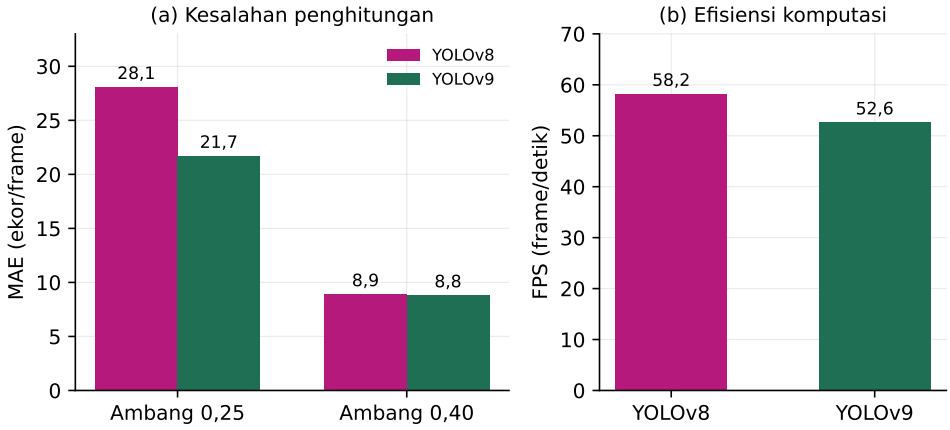
Pada ambang *confidence* 0,25, MAE YOLOv9 tercatat 21,7 ekor per *frame*, sedangkan MAE YOLOv8 tercatat 28,1 ekor per *frame* (Tabel 4 dan Gambar 8a). Selisihnya sekitar 23 persen, jauh lebih lebar daripada selisih metrik deteksi. Hasil hitung dipakai untuk menaksir jumlah ternak, sehingga perbedaan ini penting bagi sistem pemantauan. Metrik deteksi dan MAE menakar hal yang berbeda dan tidak dapat saling menggantikan.

Gambar 9 memperlihatkan empat contoh *frame* uji, Frame A sampai D. Kedua model cenderung melebihkan jumlah ayam, tetapi besar selisihnya berbeda antar-*frame*. Pada Frame A yang berisi 91 ekor, YOLOv8 menyebut 131 ekor dan meleset 40 ekor, sedangkan YOLOv9 menyebut 114 ekor dengan selisih 23 ekor. Pada Frame C yang berisi 90 ekor, YOLOv9 cen-



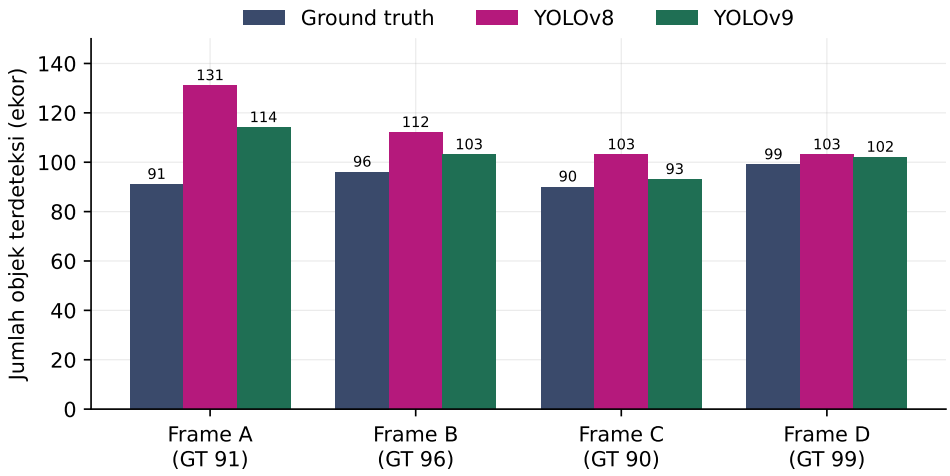
Gambar 7. Contoh hasil deteksi YOLOv9 pada *frame* CCTV yang sama dengan Gambar 6.

derung mendekati nilai aktual dengan 93 ekor, sedangkan YOLOv8 menyebut 103 ekor. Frame B dan Frame D memperlihatkan pola serupa dengan selisih yang lebih kecil. Pada seluruh 40 *frame* uji, YOLOv9 lebih dekat ke jumlah sebenarnya pada 30 *frame*, YOLOv8 pada 8 *frame*, dan 2 *frame* berimbang. Salah satu *frame* yang tidak ditampilkan berisi 93 ekor, dengan YOLOv8 menyebut 99 ekor dan YOLOv9 menyebut 104 ekor.



Gambar 8. Perbandingan (a) kesalahan penghitungan (MAE) pada ambang *confidence* 0,25 dan 0,40 serta (b) efisiensi komputasi (FPS) antara YOLOv8 dan YOLOv9.

Selisih MAE tersebut bergantung pada ambang *confidence*. Pada ambang bersama 0,40 yang dipilih dari data validasi, MAE YOLOv8 turun menjadi 8,9 ekor per *frame* dan MAE YOLOv9 menjadi 8,8 ekor per *frame*. Pada data validasi dengan ambang 0,25, urutannya bahkan terbalik: 15,5 untuk YOLOv8 dan 17,4 untuk YOLOv9. Keunggulan YOLOv9 pada ambang 0,25 belum dapat dianggap tetap. Pemilihan arsitektur untuk sistem penghitungan sebaiknya diuji langsung pada tingkat kepadatan yang menyerupai kondisi lapangan, bukan hanya pada metrik deteksi



Gambar 9. Ilustrasi *over-counting*: perbandingan jumlah objek *ground truth* dengan hasil prediksi YOLOv8 dan YOLOv9 pada empat contoh *frame* uji berkepadatan tinggi.

seperti mAP (Deng dkk., 2024).

3.4 Hasil Pengujian Waktu Komputasi (FPS)

Waktu komputasi diukur dengan protokol pada Subbab 2.5 dan diulang pada dua sesi Colab yang terpisah. Tabel 5 memuat hasilnya, dan Gambar 8b merangkum FPS rerata kedua sesi. YOLOv8 lebih cepat pada kedua sesi, dengan waktu ujung ke ujung 16,8 dan 17,6 ms per *frame* atau 59,6 dan 56,9 FPS. Waktu YOLOv9 adalah 18,2 dan 19,8 ms per *frame* atau 54,9 dan 50,4 FPS. Waktu ini lebih lama 8,5 persen pada sesi pertama dan 12,8 persen pada sesi kedua.

YOLOv9 memiliki parameter dan GFLOPs yang lebih sedikit (Tabel 2), tetapi jumlah lapisannya lebih dari dua kali lipat, 348 dibanding 170. Penyebab waktu yang lebih lama tidak diuji dalam penelitian ini.

Tabel 5. Waktu komputasi pada GPU Tesla T4 dengan protokol terkontrol (rerata ± simpangan baku antar-putaran, 10 putaran × 40 *frame*).

Sesi	Model	Inferensi (ms/*frame*)	Ujung ke ujung (ms/*frame*)	FPS
1	YOLOv8	12,8 ± 0,3	16,8 ± 0,9	59,6
1	YOLOv9	14,4 ± 0,7	18,2 ± 1,3	54,9
2	YOLOv8	13,5 ± 0,2	17,6 ± 0,6	56,9
2	YOLOv9	15,6 ± 1,0	19,8 ± 1,7	50,4

Catatan: waktu ujung ke ujung mencakup pra-proses, inferensi, NMS, dan *overhead* di luar model. FPS dihitung dari rerata waktu ujung ke ujung.

3.5 Pemilihan Model Usulan

Memilih model terbaik menjadi tidak sederhana ketika kriteria penilaiannya saling bersaing, karena satu model bisa unggul pada akurasi tetapi lebih lambat. Persoalan ini termasuk pengambilan keputusan multi-kriteria atau *Multi-Criteria Decision Making* (MCDM). Metode ini telah

lama dipakai untuk memeringkat algoritma pembelajaran mesin berdasarkan beberapa kriteria sekaligus (Peng dkk., 2011; Ali dkk., 2017). Huang dkk. (2017) menyatakan bahwa metrik akurasi baku seperti mAP tidak memadai sebagai satu-satunya dasar keputusan penerapan. Waktu proses dan kebutuhan sumber daya ikut menentukan kelayakan model di lapangan. Penelitian ini memakai SAW, salah satu metode MCDM yang paling ringkas. SAW menormalkan tiap kriteria ke rentang yang sebanding, lalu menjumlahkannya sesuai bobot kepentingan (Afshari dkk., 2010). Normalisasi diperlukan karena satuan dan rentang tiap kriteria berlainan, dari mAP yang bernilai 0 sampai 1 hingga FPS yang bernilai puluhan (Hwang dan Yoon, 1981).

Delapan kriteria dilibatkan dalam penilaian, yaitu mAP50, mAP50–95, *precision*, *recall*, *Mean* IoU, FPS, waktu per *frame*, dan MAE. Enam kriteria bersifat *benefit*, artinya nilai yang lebih tinggi menunjukkan kinerja yang lebih baik. Dua kriteria lainnya, yaitu waktu per *frame* dan MAE, bersifat *cost*, artinya nilai yang lebih rendah menunjukkan kinerja yang lebih baik. Kriteria *cost* dibalik lebih dulu sebelum dinormalkan, agar arah penilaian seluruh kriteria seragam. Setelah itu setiap kriteria dinormalkan dengan skema *min-max* ke rentang 0 sampai 1, lalu dijumlahkan sesuai bobotnya menjadi satu skor akhir (Tabel 6). Nilai FPS dan waktu per *frame* berasal dari rerata dua sesi pada Tabel 5, dan MAE berasal dari ambang 0,25.

Tabel 6. Skema pembobotan multi-kriteria (SAW) untuk pemilihan model usulan.

Aspek pertimbangan	YOLOv8	YOLOv9	Unggul	Keterangan
mAP50	0,8714	0,8725	YOLOv9	Kualitas deteksi pada data uji
mAP50–95	0,5043	0,5152	YOLOv9	Evaluasi lebih ketat pada *bounding box*
Precision	0,8498	0,8315	YOLOv8	Ketepatan deteksi
Recall	0,7653	0,7942	YOLOv9	Kemampuan menemukan objek
Mean IoU	0,7860	0,7932	YOLOv9	Ketepatan lokalisasi *bounding box*
FPS	58,2	52,6	YOLOv8	Kecepatan pemrosesan
Waktu per *frame* (ms)	17,2	19,0	YOLOv8	Waktu ujung ke ujung
MAE (ekor/*frame*)	28,1	21,7	YOLOv9	Ketepatan penghitungan, ambang 0,25
Skor terbobot (0–1)	**0,259**	**0,741**	**YOLOv9**	Model usulan

YOLOv9 unggul pada lima dari delapan kriteria, yaitu mAP50, mAP50–95, *recall*, *Mean* IoU, dan MAE. YOLOv8 unggul pada *precision*, FPS, dan waktu per *frame*. Skor akhir YOLOv9 adalah 0,741 dan skor YOLOv8 adalah 0,259, sehingga YOLOv9 ditetapkan sebagai model usulan. Karena hanya dua model yang dibandingkan, normalisasi *min-max* memberi nilai penuh kepada model yang unggul, sekecil apa pun selisihnya. Skor ini juga bergantung pada bobot, yang ditetapkan berdasarkan kebutuhan pemantauan kandang. Penelitian lanjutan dapat menurunkan bobot secara sistematis, misalnya melalui perbandingan berpasangan, atau menguji kepekaan peringkat terhadap perubahan bobot.

4 Kesimpulan

Penelitian ini membandingkan YOLOv8 dan YOLOv9 untuk mendeteksi dan menghitung ayam broiler pada rekaman CCTV satu kandang tertutup dengan anotasi satu kelas. Pada data uji, YOLOv9 lebih tinggi pada *recall*, mAP50, mAP50–95, dan *Mean* IoU, sedangkan YOLOv8 lebih tinggi pada *precision*. Selisihnya kecil dan berada dalam fluktuasi mAP validasi antar-*epoch*, sehingga kualitas deteksi kedua model dinilai sebanding. Pada penghitungan populasi dengan ambang *confidence* 0,25, MAE YOLOv9 sebesar 21,7 ekor per *frame* dan MAE YOLOv8 sebesar

28,1 ekor per *frame*. Selisih ini hampir hilang pada ambang 0,40 yang dipilih dari data validasi, dengan MAE 8,8 dan 8,9 ekor per *frame*. Pada pengukuran waktu terkontrol, YOLOv9 membutuhkan waktu per *frame* 8,5 sampai 12,8 persen lebih lama daripada YOLOv8.

Melalui skema SAW, YOLOv9 memperoleh skor 0,741 dibanding 0,259 dan ditetapkan sebagai model usulan untuk pemantauan populasi ayam broiler di kandang padat. Hasil ini menjawab celah penelitian pada bagian Pendahuluan. Celah itu adalah belum adanya kerangka evaluasi yang menautkan akurasi deteksi, ketepatan penghitungan, dan efisiensi komputasi secara serentak pada studi deteksi ayam broiler berbasis YOLO. Pilihan ini bergantung pada bobot yang mengutamakan akurasi dan ketepatan hitung. Bila kecepatan diutamakan, YOLOv8 dapat lebih sesuai.

Penelitian ini masih memiliki keterbatasan. Seluruh data berasal dari satu kandang, dan data uji hanya terdiri atas 40 *frame* dari dua sudut kamera. Setiap model dilatih sekali dengan satu nilai *seed*. Yang dibandingkan adalah dua model utuh, sehingga selisih hasil belum dapat dikaitkan dengan komponen arsitektur tertentu. Penelitian berikutnya dapat memperbesar volume dan ragam data, termasuk kandang dan sudut kamera lain, serta melakukan ablasi komponen arsitektur. Strategi *post-processing* dapat diperbaiki melalui penyetelan ambang *confidence*, IoU, dan NMS. Metode *tracking* juga dapat ditambahkan agar hasil hitung tidak sepenuhnya bergantung pada penghitungan *bounding box* per *frame*. Pengujian pada beragam perangkat komputasi juga diperlukan untuk menilai kelayakan penerapan pada sistem pemantauan kandang secara *real-time*.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Program Studi Sarjana Sains Data Universitas Negeri Surabaya serta seluruh pihak yang telah memberikan dukungan dan masukan selama pelaksanaan penelitian ini.

Daftar Pustaka

Afshari, A., Mojahed, M., dan Yusuff, R. M. (2010). Simple additive weighting approach to personnel selection problem. *International Journal of Innovation, Management and Technology*, 1(5):511–515.

Ali, M. L. dan Zhang, Z. (2024). The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. *Computers*, 13(12):336.

Ali, R., Lee, S., dan Chung, T. C. (2017). Accurate multi-criteria decision making methodology for recommending machine learning algorithm. *Expert Systems with Applications*, 71:257–278.

Astill, J., Dara, R. A., Fraser, E. D. G., Roberts, B., dan Sharif, S. (2020). Smart poultry management: Smart sensors, big data, and the internet of things. *Computers and Electronics in Agriculture*, 170:105291.

Berckmans, D. (2017). General introduction to precision livestock farming. *Animal Frontiers*, 7(1):6–11.

Bumbálek, R., Umurungi, S. N., Ufitikirezi, J. d. D. M., Zoubek, T., Kuneš, R., Stehlík, R., Lin, H.-I., dan Bartoš, P. (2025). Deep learning in poultry farming: Comparative analysis of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 for dead chickens detection. *Poultry Science*, 104(9):105440.

Deng, B., Lu, Y., dan Li, Z. (2024). Detection, counting, and maturity assessment of blueberries in canopy images using YOLOv8 and YOLOv9. *Smart Agricultural Technology*, 9:100620.

Guo, Y., Aggrey, S. E., Yang, X., Oladeinde, A., Qiao, Y., dan Chai, L. (2023). Detecting broiler chickens on litter floor with the YOLOv5-CBAM deep learning model. *Artificial Intelligence in Agriculture*, 9:36–45.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., dan Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hwang, C.-L. dan Yoon, K. (1981). *Multiple Attribute Decision Making: Methods and Applications — A State-of-the-Art Survey*, volume 186 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin.

Kementerian Pertanian (2024). Outlook daging ayam ras pedaging 2024. Technical report, Pusat Data dan Sistem Informasi Pertanian, Sekretariat Jenderal Kementerian Pertanian.

Li, H., Li, H., Shi, L., Yu, T., dan Su, Y. (2025). YOLOv8-SST lightweight small target fault detection system integrating SENet attention mechanism and Tr-OCR. *Cluster Computing*, 28(8):501.

OECD dan FAO (2025). OECD-FAO agricultural outlook 2025–2034. Technical report, OECD Publishing.

Okinda, C., Nyalala, I., Korohou, T., Okinda, C., Wang, J., Achieng, T., Wamalwa, P., Mang, T., dan Shen, M. (2020). A review on computer vision systems in monitoring of poultry: A welfare perspective. *Artificial Intelligence in Agriculture*, 4:184–208.

Padilla, R., Passos, W. L., Dias, T. L. B., Netto, S. L., dan da Silva, E. A. B. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics*, 10(3):279.

Peng, Y., Kou, G., Wang, G., dan Shi, Y. (2011). FAMCDM: A fusion approach of MCDM methods to rank multiclass classification algorithms. *Omega*, 39(6):677–689.

Redmon, J., Divvala, S., Girshick, R., dan Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Rüter, J., Durak, U., dan Dauer, J. C. (2024). Investigating the sim-to-real generalizability of deep learning object detection models. *Journal of Imaging*, 10(10):259.

Schillings, J., Bennett, R., dan Rose, D. C. (2021). Exploring the potential of precision livestock farming technologies to help address farm animal welfare. *Frontiers in Animal Science*, 2:639678.

Ultralytics (2026). YOLOv8 vs. YOLOv5: A comprehensive technical comparison. https://docs.ultralytics.com/compare/yolov8-vs-yolov5/. Accessed: 2026.

Van Limbergen, T., Sarrazin, S., Chantziaras, I., Dewulf, J., Ducatelle, R., Kyriazakis, I., McMullin, P., Méndez, J., Niemi, J. K., Papasolomontos, S., Szeleszczuk, P., Van Erum, J., dan Maes, D. (2020). Risk factors for poor health and performance in european broiler production systems. *BMC Veterinary Research*, 16(1):287.

Wang, C.-Y., Yeh, I.-H., dan Liao, H.-Y. M. (2024). YOLOv9: Learning what you want to learn using programmable gradient information. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–21.

Willmott, C. J. dan Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*,

30:79–82.

Xiaomi (2025). Xiaomi smart camera C300. Product specification.

Xu, H., Wang, X., Wang, D., Duan, B., dan Rui, T. (2023). Object detection in crowded scenes via joint prediction. *Defence Technology*, 21:103–115.