

Makalah Seminar

INTEGRASI *SINGLE SHOT MULTIBOX DETECTOR* DAN *ISOLATION FOREST* UNTUK IMPLEMENTASI SISTEM DETEKSI ANOMALI PERGERAKAN AYAM BERBASIS WEB

MATTHEW MARTIANUS HENRY (G64190032)*, SRI WAHJUNI, AURIZA RAHMAD AKBAR

ABSTRAK

Dalam usaha ternak ayam broiler, kesejahteraan hewan merupakan salah satu aspek yang perlu diperhatikan peternak. Aspek dalam kesejahteraan hewan diantaranya adalah *freedom from hunger or thirst* (kebebasan dari kelaparan dan haus) dan *freedom from pain, injury or disease* (kebebasan dari rasa sakit, luka dan penyakit). Proses pemberian makan secara terus menerus mengakibatkan meningkatnya massa ayam broiler yang membuat ayam broiler kesulitan bergerak untuk mengambil makanan maupun air sehingga ayam broiler menjadi sakit kemudian mati. Tersedianya sistem yang dapat mendeteksi perilaku tidak normal ayam broiler dari kejauhan akan mempermudah peternak dalam memantau pergerakan ayam broiler. Dalam penelitian ini, deteksi ayam yang kesulitan bergerak dilakukan melalui integrasi model deteksi objek *single shot multibox detector* (SSD) dan algoritma deteksi anomali *isolation forest* dari penelitian sebelumnya. Integrasi dilakukan menggunakan bahasa pemrograman Python, diuji menggunakan video dengan kombinasi resolusi dan *frame per second* (FPS) tertentu, kemudian diimplementasikan menjadi sebuah aplikasi web menggunakan *framework* Flask. Hasil analisis integrasi menunjukkan bahwa semakin besar resolusi dan FPS video masukan, maka waktu inferensi dan waktu eksekusi menjadi semakin besar dan diestimasi meningkat secara linear.

Kata Kunci: aplikasi web, ayam broiler, deteksi anomali, deteksi objek, kesejahteraan hewan

ABSTRACT

In the broiler farming business, animal welfare is an aspect that needs to be considered by breeders. Aspects of animal welfare include freedom from hunger or thirst and freedom from pain, injury or disease. The process of continuous feeding results in an increase of the broiler chickens mass, which makes it difficult for them to move in order to take food and water so that they become sick then die. The availability of a system that can detect abnormal behavior of broiler chickens from a distance will make it easier for farmers to monitor the movement of broiler chickens. In this study, the detection of chickens that had difficulty moving was carried out by integrating the single shot multibox detector (SSD) object detection model and the isolation forest anomaly detection algorithm from previous studies. Integration is carried out using the Python programming language, tested using video with a certain combination of resolution and frame per second (FPS), then implemented into a web application using the Flask framework. The integration analysis shows that the greater the resolution and FPS of the input video, the greater the inference and execution time. Both times are also estimated to increase linearly.

Keywords: animal welfare, anomaly detection, broiler chicken, object detection, web application

PENDAHULUAN

Latar Belakang

Budidaya unggas merupakan kegiatan pemeliharaan unggas dengan tujuan untuk membesarkan atau menggemukkan unggas. Unggas yang umumnya dijadikan sebagai budidaya adalah ayam broiler dan ayam petelur. Di Indonesia, terdapat 87 ribu peternakan budidaya unggas untuk ayam broiler dengan jumlah ayam broiler sebanyak 98,58 miliar ekor dan total nilai produksi sebesar 2,546 miliar rupiah (BPS 2021).

Budidaya unggas yang dilakukan tidak terlepas dari adanya permasalahan, salah satunya yang berkaitan dengan kesejahteraan hewan. Kesejahteraan hewan seperti dikutip dari World Organisation for Animal Health (2021) meliputi lima prinsip kebebasan yang dikenal sebagai 5F yaitu *freedom from hunger or thirst* (kebebasan dari kelaparan dan haus), *freedom from discomfort* (kebebasan dari rasa tidak nyaman), *freedom from pain, injury or disease* (kebebasan dari rasa sakit, luka dan penyakit), *freedom to express normal behaviour* (kebebasan untuk mengekspresikan tingkah laku alamiah) dan *freedom from fear and distress* (kebebasan dari rasa takut dan stres). Pemberian pakan yang dilakukan secara terus menerus dapat memicu *overweight* pada ayam broiler (Phibbs *et al.* 2021). Hal ini dapat mengakibatkan pembesaran otot tungkai pada ayam broiler, yang mengakibatkan ayam broiler kesulitan bergerak dan akhirnya mati (Kierończyk *et al.* 2017). Kesejahteraan hewan, dalam hal ini ayam broiler, perlu diperhatikan karena produktivitas peternakan meningkat seiring dengan meningkatnya kesejahteraan ayam broiler (Vetter *et al.* 2014). Kesejahteraan Hewan tersebut dapat diukur menggunakan salah satu metrik pengukuran kesejahteraan hewan yaitu Welfare Quality® (Blokhuys 2008).

Implementasi model deteksi objek untuk mendeteksi ayam pada video telah dilakukan Khairunissa *et al.* (2021) menggunakan algoritma *single shot multibox detector* (Liu *et al.* 2016) dengan jaringan dasar MobileNet (Howard *et al.* 2017). Pada penelitian Khairunissa *et al.* (2021) tersebut, selain mendeteksi objek ayam pada video, algoritma penelusuran objek berupa *euclidean distance* (Nascimento *et al.* 1999) juga diterapkan untuk menelusuri pergerakan ayam yang terdeteksi dengan nilai *multi-object tracking precision* (MOTP) yang didapat sebesar 60,4%. Perilaku anomali berupa ayam broiler yang tidak banyak bergerak dapat dikenali melalui algoritma *isolation forest* (iForest) yang dikembangkan oleh Liu *et al.* (2008) yang diterapkan pada Khairunissa *et al.* (2023). Implementasi algoritma SSD pada penelitian Khairunissa *et al.* (2021) dan iForest pada penelitian Khairunissa *et al.* (2023) masih dilakukan secara manual dan belum ada sistem yang mengintegrasikan kedua algoritma tersebut untuk mendeteksi anomali pergerakan ayam. Adanya sistem yang mengintegrasikan kedua algoritma tersebut dapat mempermudah peternak untuk mengetahui apakah terdapat perilaku pergerakan ayam yang tidak normal di kandang sehingga tindakan yang diperlukan dapat segera dilakukan. Sistem tersebut kemudian dikembangkan menjadi sebuah aplikasi berbasis web. *Framework* yang digunakan untuk pengembangan aplikasi adalah Flask, karena sifatnya yang ringan dan cocok untuk aplikasi web yang tidak terlalu kompleks (Abbade *et al.* 2020).

Sebelum dijadikan sebuah aplikasi berbasis web, dilakukan pengujian kinerja terhadap integrasi kedua model untuk menganalisis waktu inferensi dan waktu sistem berdasarkan resolusi dan FPS. Penelitian Duan *et al.* (2021) menggunakan model YOLOv4 dan video pergerakan mobil di jalan raya menunjukkan bahwa semakin besar resolusi video dan FPS maka nilai metrik F1 akan semakin besar. Huang *et al.* (2017) seperti dikutip Duan *et al.* (2021) menyebutkan bahwa nilai metrik tersebut sejajar dengan waktu eksekusi model yang digunakan. Waktu inferensi dan waktu eksekusi dari sistem akan diuji menggunakan kombinasi video berdasarkan penelitian Sanjiwo (2021) menggunakan model regresi linear (Chatterjee dan Hadi 2012). Pengujian fungsional kemudian dilakukan setelah aplikasi web telah selesai untuk menguji keseluruhan fitur pada aplikasi web dapat berjalan dengan baik.

Perumusan Masalah

1. Bagaimana model deteksi anomali pergerakan ayam broiler menggunakan algoritma deteksi objek SSD MobileNets dengan penelusuran *centroid tracking* dari penelitian Khairunissa *et al.* (2021) dan algoritma klasifikasi iForest dari penelitian Khairunissa *et al.* (2023) dapat diintegrasikan menjadi sebuah aplikasi berbasis web?
2. Bagaimana kinerja dan fungsional dari sistem aplikasi berbasis web yang dibuat?

Tujuan Penelitian

1. Mengimplementasikan model deteksi anomali pergerakan ayam melalui integrasi algoritma deteksi objek SSD MobileNet dengan penelusuran *centroid tracking* (Khairunissa *et al.* 2021) dan algoritma klasifikasi iForest (Khairunissa *et al.* 2023) berbasis web.
2. Melakukan pengujian kinerja dan pengujian fungsional pada sistem yang dibuat.

Manfaat Penelitian

Aplikasi berbasis web untuk mendeteksi anomali pergerakan ayam diharapkan dapat membantu peternak untuk menindaklanjuti perilaku ayam broiler yang tidak normal yang dipantau dari jarak jauh melalui masukan sebuah video.

Ruang Lingkup Penelitian

1. Data video dan model yang diintegrasikan sama dengan yang digunakan pada penelitian Khairunissa *et al.* (2021) dan Khairunissa *et al.* (2023).
2. Integrasi model SSD dan iForest dilakukan menggunakan bahasa pemrograman Python dan dihubungkan dengan tampilan antarmuka menggunakan *framework* Flask.
3. Sistem yang diimplementasikan belum bersifat *real time*.

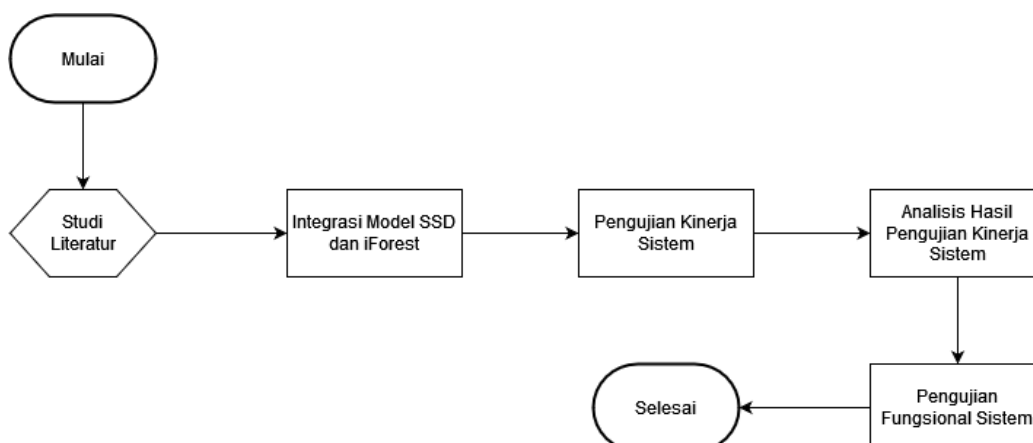
METODE

Data Penelitian

Dalam penelitian ini, digunakan model SSD yang dihasilkan dari penelitian Khairunissa *et al.* (2021) yang diimplementasikan dalam caffemodel dan model iForest yang sama dari penelitian Khairunissa *et al.* (2023). Video yang digunakan sebagai input juga merupakan video yang sama seperti dalam penelitian Khairunissa *et al.* (2021). Selain model dan video, kode program dari penelitian Khairunissa *et al.* (2023) dan Khairunissa *et al.* (2022) juga digunakan sebagai referensi dalam implementasi sistem yang dibuat.

Tahapan Penelitian

Tahapan penelitian terdiri dari lima tahapan yaitu studi literatur, integrasi model SSD dan iForest, pengujian kinerja sistem, analisis hasil pengujian kinerja sistem, dan diakhiri dengan pengujian fungsional sistem seperti pada Gambar 1.



Gambar 1 Tahapan penelitian

Studi Literatur

Studi literatur dilakukan untuk mengetahui parameter masukan dan keluaran dari model SSD (Khairunissa *et al.* 2021) dan iForest (Khairunissa *et al.* 2023) yang dikembangkan. Tahapan ini juga meliputi pembelajaran literatur terkait dengan SSD dan iForest untuk mendapatkan gambaran umum mengenai cara kerja kedua model.

Integrasi Model SSD dan iForest

Integrasi model SSD dan iForest dilakukan dengan menggunakan bahasa pemrograman Python seperti pada penelitian sebelumnya (Khairunissa *et al.* (2021) dan Khairunissa *et al.* (2023)). *Library* yang digunakan untuk integrasi adalah NumPy, Pandas, Matplotlib, Scikit-learn, dlib, SciPy, imutils dan OpenCV.

Pengujian Kinerja Sistem

Pengujian sistem dilakukan untuk mengukur waktu inferensi setiap model untuk mengetahui pengaruh resolusi dan FPS terhadap kinerja masing masing model yang diintegrasikan. Selain waktu inferensi, pengujian juga dilakukan untuk mengetahui waktu eksekusi dari keseluruhan proses pada sistem. Sistem akan diberi masukan 5 buah video dengan kombinasi resolusi dan FPS seperti pada Tabel 1. Pengukuran waktu inferensi dan waktu eksekusi dilakukan dengan fungsi *monotonic_ns()* dari modul *time* yang merupakan bawaan dari bahasa pemrograman Python.

Tabel 1 Skenario Pengujian Sistem

<i>Resolusi</i>	<i>FPS</i>
640 x 480	15, 25, 30
800 x 600	15, 25, 30
1280 x 720	15, 25, 30

Analisis Hasil Pengujian Kinerja Sistem

Berdasarkan hasil dari pengujian sistem, dilakukan analisis untuk mengetahui kemungkinan relasi antara resolusi dan FPS dengan lamanya waktu inferensi maupun waktu eksekusi. Hasil analisis akan dipertimbangkan sebagai saran untuk pengembangan sistem terutama pada bagian model deteksi objek dan model deteksi anomali. Analisis kinerja akan dilakukan menggunakan bahasa pemrograman Python dengan *library* Pandas dan Seaborn. Analisis regresi kemudian dilakukan menggunakan *library* Statsmodels.

Pengujian Fungsional Sistem

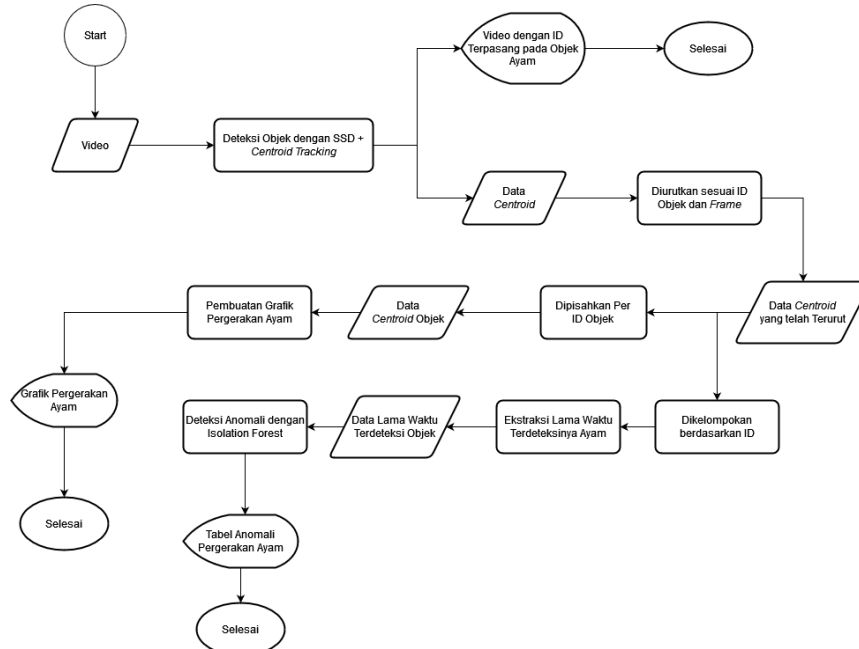
Pada tahap ini, dilakukan pengembangan tampilan antarmuka web yang akan menjadi hasil akhir dari penelitian. Pengembangan tersebut dilakukan dengan HTML, CSS dan bahasa pemrograman Javascript. Tampilan antarmuka yang telah jadi akan digabungkan dengan hasil integrasi model SSD dan iForest yang berfungsi sebagai *backend* dari aplikasi web yang dikerjakan menggunakan *framework* Flask.

Pengujian fungsional kemudian dilakukan untuk menguji seluruh fitur yang dibuat pada web dapat berjalan dengan baik. Keseluruhan fitur tersebut yaitu seluruh tombol untuk berpindah ke halaman lain, kotak untuk *upload* video, tombol *submit* untuk memulai analisis dan tombol untuk *download* hasil analisis yang dilakukan sistem.

HASIL DAN PEMBAHASAN

Studi Literatur

Berdasarkan pembelajaran kode program yang digunakan pada penelitian sebelumnya (Khairunissa *et al.* (2021) dan Khairunissa *et al.* (2022)), didapatkan skema berjalannya program untuk mendapatkan hasil yang diinginkan seperti pada Gambar 2. Skema ini akan digunakan sebagai dasar untuk pembuatan modul dalam tahap integrasi model SSD dan iForest.



Gambar 2. Skema kerja program yang menjadi dasar integrasi model SSD dan iForest

Integrasi Model SSD dan iForest

Integrasi dilakukan dengan membuat modul yang sesuai untuk setiap tahapan yang dilalui. Perubahan ini dilakukan agar implementasi pada tahap pengembangan aplikasi web dapat menjadi lebih mudah dengan hanya melakukan instansiasi kelas dan memanggil fungsi dari modul yang diperlukan. Berdasarkan Gambar 3, akan terdapat 5 modul yang dibuat, yaitu

1. Modul *inputs* yang berisikan kode program Python untuk kelas Video dengan atribut dan fungsi untuk instansiasi video, membaca tiap *frame* pada video, menuliskan *frame* hasil ke tempat penyimpanan, menghentikan *pointer* video dan memberikan *stream* video ke model SSD yang telah diinstansiasi. Instansiasi kelas Video dilakukan dengan memberikan sebuah video dengan format MP4.
2. Modul *detector* berisikan sebuah folder dengan nama *mobilenet_ssd* yang menampung *pretrained model* SSD dalam format *caffemodel* dan *prototxt*, kode program Python untuk kelas *MobileNetSSD* dengan atribut dan fungsi untuk instansiasi model, mendeteksi objek, memberikan *tracker*, memperbaharui posisi *tracker*, melakukan perhitungan untuk menentukan posisi *centroid* baru, membuat hasil tabular dan menyimpan video hasil deteksi dan penelusuran yang telah dilengkapi *centroid*. *Pretrained model* SSD pada folder *mobilenet_ssd* akan dipanggil ketika kelas *MobileNetSSD* diinstansiasi. Selain *pretrained model* SSD, *stream* video dari kelas Video yang telah diinstansiasi juga akan digunakan oleh kelas *MobileNetSSD*.
3. Modul *tracker* berisikan 2 buah kode program Python. Kode program pertama berupa kelas *CentroidTracker* yang memiliki atribut dan fungsi untuk melakukan inisiasi ID pada objek terdeteksi, terminasi ID pada objek terdeteksi dan pembaharuan ID pada objek terdeteksi. Kode program kedua berupa kelas *TrackableObject* untuk menyimpan ID objek yang terdeteksi. Implementasi kedua program ini dilakukan pada fungsi yang terdapat pada kelas *MobileNetSSD* pada modul *detector*.
4. Modul *preprocessing* berisikan 4 kode program Python untuk menghitung waktu awal dan akhir sebuah objek terdeteksi, plot pergerakan objek, mengurutkan objek

berdasarkan *frame* mulai terdeteksinya objek dan memisahkan objek terdeteksi berdasarkan ID dan menyimpannya ke dalam sebuah file CSV dengan banyaknya file sesuai dengan banyaknya objek. Keempat kode program tersebut seluruhnya merupakan sebuah fungsi.

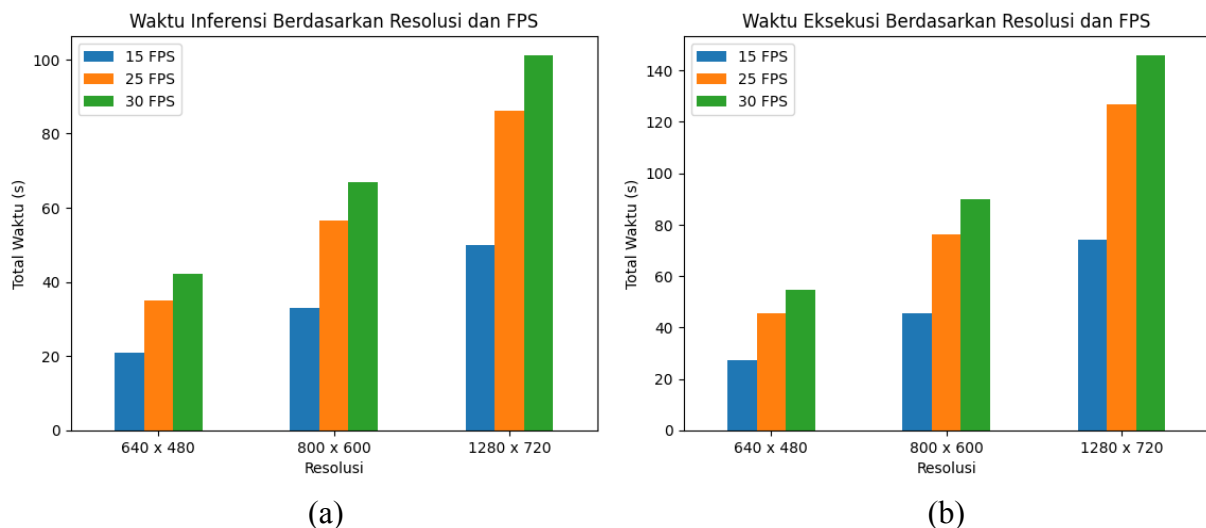
5. Modul *anomaly_detection* yang berisikan sebuah kode program untuk melakukan deteksi anomali dengan model *isolation forest* menggunakan *library* Scikit-learn.

Pengujian Kinerja Sistem

Kinerja kedua model yang telah diintegrasikan menjadi sebuah sistem diuji dengan menginstansiasi 5 video pergerakan ayam di kandang ayam dengan kombinasi resolusi 640 x 480, 800 x 600 serta 1280 x 720 dengan FPS 15, 25 dan 30. Atribut yang diteliti berupa resolusi video, FPS video, waktu inferensi, waktu eksekusi. Hasil pengujian disimpan dalam sebuah file CSV dengan banyaknya observasi sebanyak 45 observasi dan banyaknya atribut sebanyak 8 atribut.

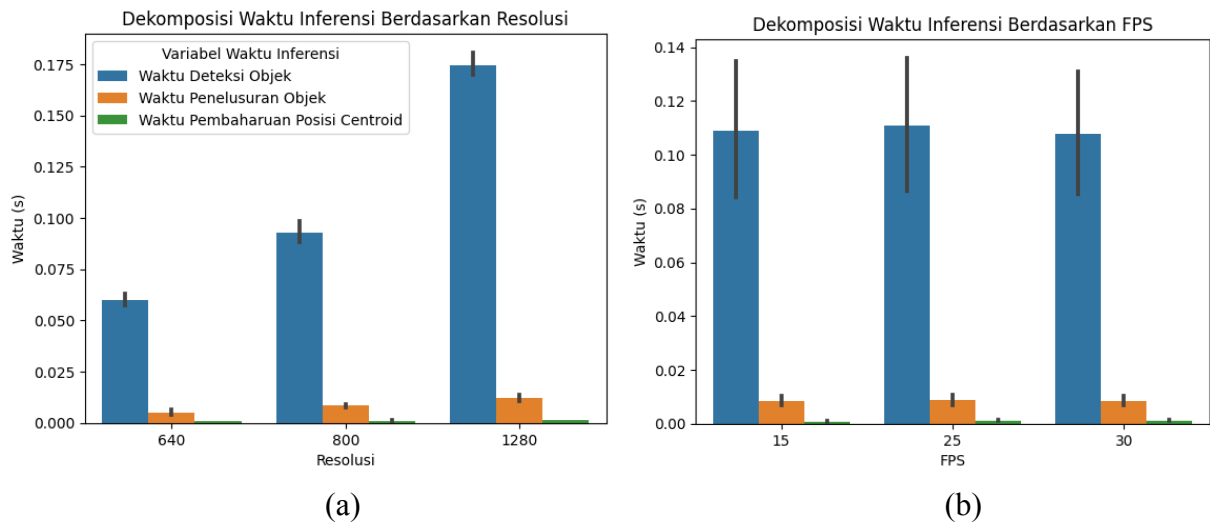
Analisis Hasil Pengujian Kinerja Sistem

Berikut merupakan hubungan resolusi video dan FPS terhadap rata rata waktu inferensi dan waktu eksekusi (Gambar 3).



Gambar 3. Waktu inferensi (a) dan waktu eksekusi (b) berdasarkan resolusi dan FPS

Berdasarkan Gambar 3a dan 3b, terlihat bahwa semakin besar resolusi maka baik waktu inferensi maupun waktu eksekusi menjadi semakin besar. Hal yang sama juga berlaku untuk atribut FPS. Waktu inferensi merupakan bagian dari waktu eksekusi, sehingga hal tersebut juga menjelaskan penyebab kedua grafik pada gambar 3a dan 3b memiliki bentuk yang kurang lebih serupa. Hal ini sesuai dengan kesimpulan dari penelitian Duan *et al.* (2017) yang menyatakan secara tidak langsung hubungan waktu eksekusi yang meningkat seiring bertambahnya besarnya resolusi video dan FPS. Berdasarkan Gambar 3a dan 3b tersebut, dapat disimpulkan juga bahwa video dengan resolusi yang sama tetapi memiliki FPS berbeda akan memiliki waktu inferensi dan waktu eksekusi yang berbeda, dimana semakin besar FPS maka waktu inferensi dan eksekusi semakin besar.



Gambar 4. Dekomposisi waktu inferensi berdasarkan resolusi (a) dan FPS (b)

Apabila dilihat lebih detail pada waktu inferensi, diketahui berdasarkan Gambar 4a bahwa waktu deteksi objek merupakan variabel yang paling berpengaruh dalam waktu inferensi dibandingkan waktu penelusuran objek dan waktu pembaharuan posisi *centroid*. Rata rata waktu deteksi objek terlihat semakin besar seiring bertambahnya resolusi video, sementara waktu penelusuran objek dan waktu pembaharuan *centroid* tidak meningkat secara signifikan. Berdasarkan Gambar 4b, terlihat bahwa rata rata waktu deteksi objek, waktu penelusuran objek dan waktu pembaharuan posisi *centroid* tidak jauh berbeda untuk masing masing FPS.

Hubungan peningkatan waktu inferensi dan waktu eksekusi tersebut dianalisis menggunakan model regresi linear berganda. Peubah yang digunakan adalah resolusi video dan FPS yang dilakukan *encoding* karena keduanya merupakan peubah kategorik. Peubah 15 dan 25 melambangkan peubah FPS yang telah dilakukan *encoding*, begitu juga dengan peubah 640 x 480 dan 800 x 600 yang melambangkan peubah resolusi video. Masukan angka untuk peubah yang telah dilakukan *encoding* hanya dapat bernilai 1 atau 0. Tabel 2 merupakan hasil analisis regresi untuk waktu inferensi dengan atribut resolusi video dan FPS.

Tabel 2 Analisis Regresi Linear Resolusi Video dan FPS terhadap Waktu Inferensi

	<i>Koefisien</i>	<i>Standar Deviasi</i>	<i>t</i>	<i>P > t </i>
Konstanta	94.5510	4.096	23.086	0.000
15	-35.5488	4.486	-7.924	0.000
25	-10.8607	4.486	-2.421	0.020
640	-46.3893	4.486	-10.340	0.000
800	-26.9306	4.486	-6.003	0.000
R-squared : 0.813		F-statistic : 43.45		

Berdasarkan Tabel 2, didapatkan model regresi linear untuk waktu inferensi seperti pada persamaan 2.

$$y = 94.55 - 35.55x_1 - 10.86x_2 - 46.39x_3 - 26.93x_4 + \epsilon \quad (2)$$

Nilai y menyatakan waktu inferensi, x_1 menyatakan apakah video memiliki *frame rate* sebesar 15 FPS, x_2 menyatakan apakah video memiliki *frame rate* sebesar 25 FPS, x_3 menyatakan

apakah video beresolusi 640 x 480 dan x_4 menyatakan apakah video beresolusi 640 x 480. Berdasarkan hasil analisis tersebut, dapat diinterpretasikan bahwa konstanta berperan sebagai waktu inferensi terbesar yang dimiliki video dengan resolusi 1280 x 720 dan FPS sebesar 30. Koefisien yang dimiliki oleh peubah lainnya merepresentasikan pengurangan waktu sesuai dengan resolusi video atau FPS. Sebagai contoh, video dengan resolusi 640 x 480 dan 15 FPS akan memiliki estimasi rata rata waktu inferensi sebesar $94.5510 - 46.3893 - 35.5488 = 12.6129$ sekon, yang merupakan waktu terkecil dari seluruh kombinasi resolusi video dan FPS. Berdasarkan hasil analisis dan interpretasi tersebut, dapat disimpulkan bahwa model regresi linear dapat digunakan untuk memperkirakan rata rata waktu inferensi dari sistem.

Sementara itu, hasil analisis regresi untuk waktu eksekusi seperti terlihat pada Tabel 3.

Tabel 3 Analisis Regresi Linear Resolusi Video dan FPS terhadap Waktu Eksekusi

	<i>Koefisien</i>	<i>Standar Deviasi</i>	<i>t</i>	<i>P > t </i>
Konstanta	136.3256	4.385	31.087	0.000
15	-47.7563	4.804	-9.941	0.000
25	-13.9333	4.804	-2.900	0.006
640	-73.3279	4.804	-15.264	0.000
800	-45.2956	4.804	-9.429	0.000
R-squared : 0.895			F-statistic : 85.46	

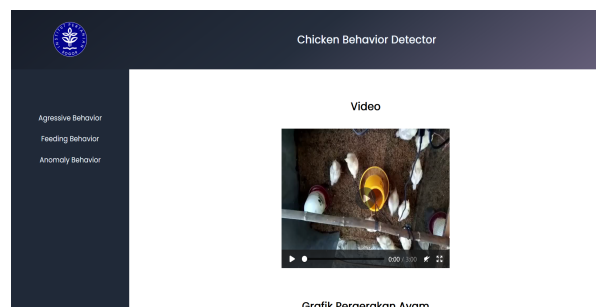
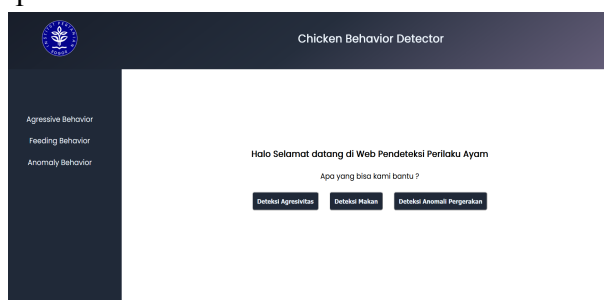
Berdasarkan Tabel 3, didapatkan model regresi linear untuk waktu eksekusi seperti pada persamaan 3.

$$y = 136.33 - 47.76x_1 - 13.93x_2 - 73.33x_3 - 45.30x_4 + \epsilon \quad (3)$$

Representasi variabel pada persamaan 3 sama seperti pada persamaan 2. Interpretasi analisis regresi waktu eksekusi sama dengan interpretasi analisis regresi waktu inferensi. Konstanta berperan sebagai waktu eksekusi resolusi video terbesar dan FPS terbesar, yaitu resolusi video 1280 x 720 dan FPS sebesar 30. Koefisien yang dimiliki oleh peubah lainnya merepresentasikan pengurangan waktu sesuai dengan resolusi video atau FPS seperti halnya analisis regresi waktu inferensi. Berdasarkan hasil analisis dan interpretasi tersebut, dapat disimpulkan bahwa model regresi linear dapat digunakan untuk memperkirakan rata rata waktu eksekusi dari sistem.

Pengujian Fungsional Sistem

Fitur deteksi anomali pergerakan ayam memiliki 3 tampilan antarmuka, yaitu halaman utama, halaman untuk memasukan video pergerakan ayam dan halaman untuk menampilkan hasil deteksi anomali pergerakan ayam. Contoh beberapa hasil antarmuka tersebut dapat dilihat pada Gambar 5 berikut.



ID	First Occurance	Last Occurance	Detection Time	Anomaly
0	3.421875	106.203125	102.78125	1
1	11.3125	94.796875	83.484375	1

Gambar 5. Cuplikan hasil implementasi sistem ke dalam sebuah aplikasi berbasis web

Setelah tampilan antarmuka dikembangkan dan digabungkan dengan integrasi sistem, dimasukkan sebuah video untuk menguji apakah sistem dapat berjalan dengan baik dan memberikan hasil seperti yang diharapkan. Berdasarkan video masukan tersebut, diketahui bahwa sistem telah berjalan dengan baik sesuai yang diharapkan. Akan tetapi, waktu eksekusi dapat menjadi berbeda dari hasil pengujian kinerja sistem karena beberapa hal yang baru diterapkan saat pembuatan web, seperti konversi video AVI ke MP4 menggunakan ffmpeg, penggunaan bahasa pemrograman Javascript yang tidak berada dalam ruang lingkup pengujian kinerja, dan sebagainya.

KESIMPULAN DAN SARAN

Kesimpulan

1. Integrasi model deteksi objek dan deteksi anomali pergerakan ayam dilakukan dengan menggunakan bahasa pemrograman Python, kemudian dihubungkan dengan antarmuka aplikasi web menggunakan *framework* Flask.
2. Semakin besar resolusi dan FPS pada video masukan, maka waktu inferensi dan waktu eksekusi dari sistem juga akan semakin besar. Apabila dilakukan dekomposisi untuk waktu inferensi, maka semakin besar resolusi video, waktu deteksi objek semakin besar sementara semakin besar FPS maka waktu deteksi objek, penelusuran objek dan pembaharuan posisi *centroid* tidak akan banyak terpengaruh.

Saran

1. Jumlah video pengujian dapat ditambah untuk mendapatkan hasil analisis regresi yang lebih akurat.
2. Atribut lain selain resolusi video dan FPS dapat digunakan untuk memperkuat hasil analisis regresi yang didapat, seperti jumlah objek terdeteksi.

DAFTAR PUSTAKA

- Abbade LR, da Cruz MAA, Rodrigues JJPC, Lorenz P, Rabelo RAL, Al-Muhtadi J. 2020. Performance comparison of programming languages for Internet of Things middleware. *Trans. Emerg. Telecommun. Technol.* 31(12).doi:10.1002/ett.3891. [diunduh 2023 Apr 20]. Tersedia pada: <https://onlinelibrary.wiley.com/doi/10.1002/ett.3891>
- Blokhuis HJ. 2008. International cooperation in animal welfare: the Welfare Quality® project. *Acta Vet. Scand.* 50(S1):S10.doi:10.1186/1751-0147-50-S1-S10.
- [BPS] Badan Pusat Statistik. 2021. Statistik Perusahaan Peternakan Unggas, 2021. Jakarta: Badan Pusat Statistik.
- Chatterjee S, Hadi AS. 2012. Regression Analysis by Example. Fifth edition. John Wiley and Sons Inc.
- Duan Z, Yang Z, Samoilenko R, Oza DS, Jagadeesan A, Sun M, Ye H, Xiong Z, Zussman G, Kostic Z. 2021. Smart City Traffic Intersection: Impact of Video Quality and Scene Complexity on Precision and Inference. Di dalam: *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. [internet] *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th*

- Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*; Haikou, Hainan, China. Haikou, Hainan, China: IEEE. hlm. 1521–1528. [diunduh 2023 Mar 22]. Tersedia pada: <https://ieeexplore.ieee.org/document/9781018/>
- Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. [diunduh 2022 Des 7]. Tersedia pada: <http://arxiv.org/abs/1704.04861>
- Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, *et al.* 2017. Speed/accuracy trade-offs for modern convolutional object detectors. [diunduh 2023 Mar 22]. Tersedia pada: <http://arxiv.org/abs/1611.10012>
- Khairunissa J, Wahjuni S, Soesanto IRH, Akbar AR, Rahmawan H. 2023. Multi-Object Tracking Algorithm for Poultry Behavior Anomaly Detection. *Int. J. Adv. Soft Comput. Its Appl.* 15:18.doi:10.15849/IJASCA.230320.11.
- Khairunissa J, Wahjuni S, Soesanto IRH, Wulandari W. 2021. Detecting Poultry Movement for Poultry Behavioral Analysis using The Multi-Object Tracking (MOT) Algorithm. Di dalam: *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*. [internet] *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*; Kuala Lumpur, Malaysia. Kuala Lumpur, Malaysia: IEEE. hlm. 265–268. [diunduh 2022 Des 7]. Tersedia pada: <https://ieeexplore.ieee.org/document/9467144/>
- Kierończyk B, Rawski M, Józefiak D, Świątkiewicz S. 2017. Infectious and non-infectious factors associated with leg disorders in poultry – a review. *Ann. Anim. Sci.* 17(3):645–669.doi:10.1515/aoas-2016-0098.
- Liu FT, Ting KM, Zhou Z-H. 2008. Isolation Forest. Di dalam: *2008 Eighth IEEE International Conference on Data Mining*. [internet] *2008 Eighth IEEE International Conference on Data Mining (ICDM)*; Pisa, Italy. Pisa, Italy: IEEE. hlm. 413–422. [diunduh 2022 Des 7]. Tersedia pada: <http://ieeexplore.ieee.org/document/4781136/>
- Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC. 2016. SSD: Single Shot MultiBox Detector. Vol. 9905. hlm. 21–37. [diunduh 2022 Des 7]. Tersedia pada: <http://arxiv.org/abs/1512.02325>
- Nascimento JC, Abrantes AJ, Marques JS. 1999. An algorithm for centroid-based tracking of moving objects. Di dalam: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*. [internet] *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*; Phoenix, AZ, USA. Phoenix, AZ, USA: IEEE. hlm. 3305–3308 vol.6. [diunduh 2022 Des 7]. Tersedia pada: <http://ieeexplore.ieee.org/document/757548/>
- [OIE] World Organization of Animal Health. 2021. Terrestrial Animal Health Code. Twenty-ninth edition. Paris: World Organization for Animal Health.
- Phibbs DV, Groves PJ, Muir WI. 2021. Leg health of meat chickens: impact on welfare, consumer behaviour, and the role of environmental enrichment. *Anim. Prod. Sci.* 61(12):1203.doi:10.1071/AN19511.
- Sanjiwo SH. 2021. Pembangunan Sistem Pemantauan Jarak Jauh Otomatis untuk Perilaku Ayam Menggunakan Kamera Inframerah [Skripsi]. Bogor: Institut Pertanian Bogor.
- Vetter S, Vasa L, Ózsvári L. 2014. Economic Aspects of Animal Welfare. *Acta Polytech. Hung.* 11(7).