

PERANCANGAN *OBJECT DETECTION* AYAM BROILER MENGUNAKAN METODE *DEEP LEARNING*

*Y. Ananta Fajar Pramudya¹, M. Munadi², Ismoyo Haryanto²

¹Mahasiswa Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro

²Dosen Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro

Jl. Prof. Sudharto, SH., Tembalang-Semarang 50275, Telp. +62247460059

*E-mail: yohanananta@gmail.com

Abstrak

Kepadatan kandang merupakan kesesuaian antara luas kandang dengan jumlah ayam per 1 m². Ayam akan mudah stres akibat tingginya gas amonia di dalam kandang, sirkulasi udara menjadi buruk serta suhu dan kelembapan meningkat di kandang. Suhu dan kelembapan yang tinggi mengakibatkan konsumsi pakan akan berkurang sehingga hal ini berpengaruh terhadap pencapaian bobot yang tidak maksimal. Untuk mengatasi kepadatan ayam dalam kandang, dapat dilakukan proses penjarangan. Penjarangan ayam selama ini dilakukan secara manual dimana sering menyebabkan ayam stres. Dengan memanfaatkan *deep learning*, dapat dibuat suatu program *object detection* dan *object measurement* yang dapat membantu proses penjarangan ayam semakin efektif karena dapat mengukur ukuran ayam secara otomatis sehingga membantu mempercepat proses penjarangan. Pada penelitian ini dirancang sistem *object detection* yang dapat mendeteksi dan memprediksi ukuran ayam broiler menggunakan Raspberry Pi 4B sebagai komputer utama, kamera Logitech C930E, *frameworks* Darknet dan algoritma deteksi YOLO (*You Only Look Once*) sebagai sistem deteksi utama. *Training* data dilakukan pada Google Colab sebanyak 6000 iterasi dan menggunakan algoritma deteksi objek Yolov4-tiny. Pengujian jarak optimal kamera menunjukkan bahwa sistem *object detection* optimal pada jarak 100 cm. Hasil pengujian sistem *object detection* pada ayam broiler umur 3-25 hari menunjukkan eror, ayam 3 hari (5,19%), 14 hari (2,76%), 21 hari (3,23% dan 6,10%), 25 hari (5,05% dan 5,81%). Pengelompokan ukuran ayam berdasarkan pengamatan selama 25 hari secara berturut-turut kecil (18 – 22 cm), sedang (23 -27 cm), besar (28 - 32 cm).

Kata kunci : ayam broiler; *object detection*; *object measurement*; penjarangan; raspberry pi 4 model b; yolo; yolov4-tiny

Abstract

Cage density is the match between the area of the cage and the number of chickens per 1 m². Chickens will be easily stressed due to high ammonia gas in the cage, poor air circulation along with increased temperature and humidity in the cage. High temperatures and humidity result in reduced feed consumption so that this affects the lure of weight that is not optimal. To overcome the density of chickens in the cage, a thinning process can be carried out. So far, thinning of chickens has been done manually, which often causes stress for the chickens. By utilizing deep learning, an object detection and object measurement program can be made which can help the chicken thinning process to be more effective because it can measure the size of the chickens automatically thereby helping to speed up the thinning process. In this study an object detection system was designed that can detect and predict the size of broiler chickens using a Raspberry Pi 4B as the main computer, Logitech C930E camera, Darknet framework, and the YOLO (You Only Look Once) detection algorithm as the main detection system. Data training was carried out on Google Colab for 6000 iterations and using the Yolov4-tiny object detection algorithm. Testing the optimal camera distance shows that the object detection system is optimal at a distance of 100 cm. The test results of the object detection system on broiler chickens aged 3-24 days showed an error, chickens 3 days (5,19%), 14 days (2,76%), 21 days (3,23% and 6,10%), 24 day (5,05% and 5,81%). Classification of chicken size based on observations for 24 consecutive days small (18 – 22 cm), medium (23 -27 cm), large (28 - 32 cm).

Keywords: broilers; *object detection*; *object measurement*; raspberry pi model 4b; thinning; yolo; yolov4-tiny

1. Pendahuluan

Ayam broiler memiliki sifat karakteristik badan yang besar, berlemak, memiliki gerak yang lamban dan memiliki pertumbuhan yang cepat, serta menghasilkan daging dengan kandungan protein yang tinggi. Hal ini menjadikan usaha ayam broiler dinilai sangat potensial untuk di jadikan usaha [1]. Kandang ayam broiler berdasarkan tipe dinding (ventilasi) dapat dibedakan menjadi kandang tertutup (*closed house*) dan kandang terbuka (*opened house*) [2].

Computer vision bekerja dengan menggunakan algoritma dan sensor optik untuk merangsang visualisasi manusia untuk secara otomatis mengekstrak informasi penting dari suatu objek [3]. Dalam *machine learning*, program komputer ditugaskan untuk melakukan beberapa tugas dan dikatakan bahwa mesin telah belajar dari pengalamannya, performa meningkat seiring dengan semakin banyaknya pengalaman-pengalaman yang didapatkan [4].

Algoritma *machine learning* yang sering digunakan di antaranya adalah *linear classifier*, *logistic regression*, Naïve Bayes, *Bayesian Network*, SVM (*Support Vector Machine*), Random Forest, k-NN (k-Nearest Neighbor), dan ANN (*Artificial Neural Network*) [5]. *Artificial neural networks* (ANNs) terdiri dari neuron yang saling berhubungan yang mengambil *input* dan melakukan tindakan pemrosesan tertentu pada data *input*, akhirnya meneruskan *output* lapisan saat ini ke lapisan berikutnya [6]. ANN yang memiliki dua atau lebih lapisan *hidden layer* bisa digolongkan ke dalam *deep learning* atau *deep neural network*.

Convolutional Neural Network (CNN) adalah salah satu *neural networks* yang mengekstrak fitur gambar asli melalui *convolution layer* dan *pooling layer*. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data gambar [7]. Arsitektur CNN tersusun menjadi *Input Layer*, *Convolution Layer* melakukan operasi konvolusi untuk mengekstraksi fitur dari gambar *input*. *Pooling layer* adalah lapisan yang berfungsi untuk mengurangi ukuran spasial dari fitur konvolusi. Ada dua jenis *pooling layer*, yaitu *max pooling* dan *average pooling* [8].

Object Detection adalah prosedur untuk menentukan turunan dari kelas tempat objek tersebut berada dan memperkirakan lokasi objek dengan mengeluarkan kotak pembatas (*bounding box*) di sekitar objek [9]. *Object detection* bekerja dengan mengidentifikasi keberadaan objek di kamera video atau di gambar [10]. *Object detection and measurement* menggunakan algoritma deteksi untuk mendeteksi objek dan untuk pengukuran objek digunakan OpenCV. Dengan bantuan titik referensi yang dibuat dengan *bounding box grid* dapat digunakan menghitung tinggi dan lebar objek [10].

OpenCV (*Open Source Computer Vision Library*) dibuat untuk menyediakan infrastruktur yang umum digunakan pada aplikasi *computer vision* dan mempercepat aplikasi yang terkait dengan *machine perception* untuk produk komersial [11]. Darknet adalah *high performance open source frameworks* untuk implementasi *neural networks*. Implementasi lanjutan dari *deep neural networks* dapat dilakukan menggunakan Darknet [12]. Implementasi ini termasuk *You Only Look Once* (YOLO) untuk deteksi objek *real-time* [13]. YOLO menerapkan jaringan saraf tunggal ke gambar penuh. Jaringan ini membagi gambar menjadi beberapa bagian dan memprediksi *bounding box* dan probabilitas untuk setiap bagian. Setiap sel kisi memprediksi *bounding box* dan *confidence score* untuk kotak tersebut [14].

Kepadatan kandang merupakan kesesuaian antara luas kandang dengan jumlah ayam per 1 m². Apabila kondisi nyaman dari kepadatan tidak tercapai, maka dapat memberikan dampak negatif terhadap performa ayam. Ayam akan mudah stres akibat tingginya gas amonia di dalam kandang. Untuk mengatasi kepadatan ayam dalam kandang, dapat dilakukan proses penjarangan. Penjarangan ayam selama ini dilakukan secara manual dimana sering menyebabkan ayam stres. Dengan memanfaatkan *deep learning*, dapat dibuat suatu program *object detection* dan *object measurement* yang dapat membantu proses penjarangan ayam semakin efektif karena dapat mengukur ukuran ayam secara otomatis sehingga membantu mempercepat proses penjarangan dan mengurangi stres pada ayam.

2. Bahan dan Metode Penelitian

2.1. Komponen utama

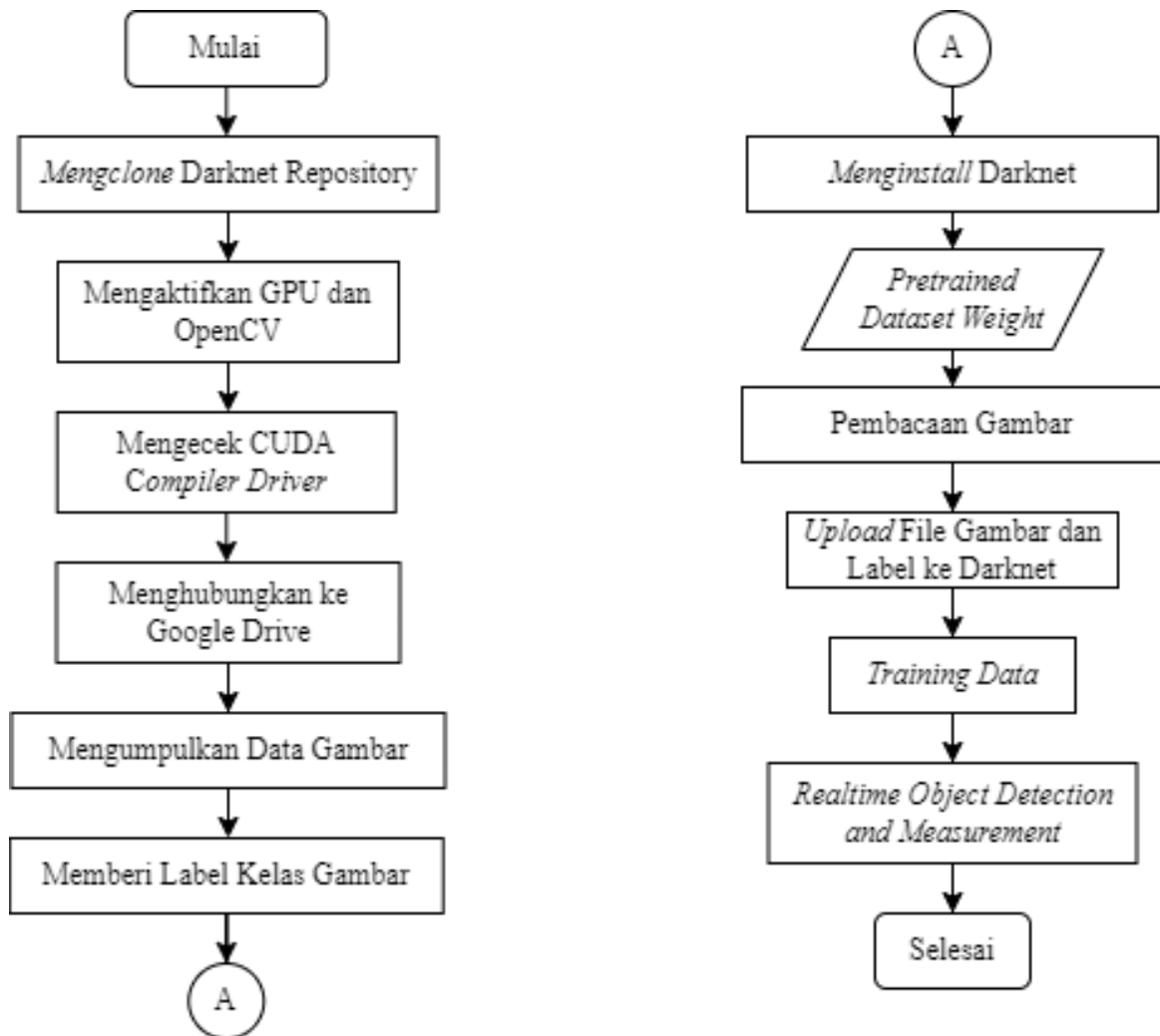
Raspberry Pi adalah komputer *mini* (terdiri dari: prosesor, *graphics card*, dan memori dalam satu paket). Menggunakan *Board* Raspberry Pi ini, seluruh sistem dapat dioperasikan hanya dari *power source* yang diperlukan untuk *Board* Raspberry Pi. Raspberry Pi menggunakan kartu SD untuk *booting* dan penyimpanan. Kartu SD harus berisi sistem operasi (Linux), program dan data yang dibutuhkan untuk menjalankan Raspberry Pi [15]. Kamera yang digunakan adalah Logitech C930E yang mempunyai resolusi 3 Megapixel 1080p/30fps dan 720p/30fps, USB A, panjang kabel 1,5 m.

2.2. Menginstall Paket Perangkat Lunak

Untuk menjalankan program *object detection and measurement* dibutuhkan beberapa *software deep learning*. Git Bash untuk *mengclone* data ke penyimpanan lokal. OpenCV *library* untuk *software* yang mengaplikasikan *computer vision* dan *machine learning*, Darknet untuk implementasi *neural networks*, CUDA dan cuDNN untuk meningkatkan dalam kinerja komputasi dengan memanfaatkan kekuatan GPU, YOLO v4-tiny untuk algoritma deteksi utama.

2.3. Pelatihan Data

Pelatihan data dilakukan pada Google Colaboratory supaya mempercepat waktu pelatihan data. Data *training* dimulai dengan *mengclone* Darknet ke Google Colab sebagai *frameworks* untuk algoritma deteksi YOLO yang akan digunakan untuk sistem deteksi objek.



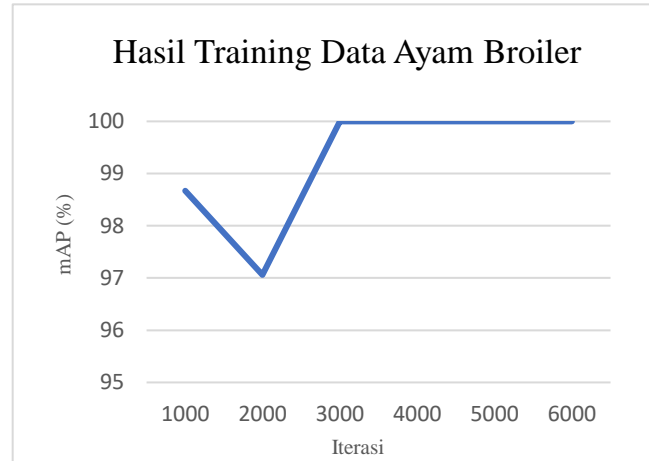
Gambar 1. Diagram Alir Algoritma *Training Data*

Gambar 1. menunjukkan diagram alir proses pelatihan data pada Darknet. Semua *software* yang digunakan dijalankan pada Google Colaboratory. Gambar yang akan digunakan diberi nama label kelas sesuai dengan objek yang digunakan. Keseluruhan gambar kemudian diunggah kedalam Darknet untuk melakukan proses *training data* sebanyak 6000 iterasi yang akan digunakan dalam sistem deteksi. Hasil training data akan tersimpan secara otomatis dalam Google Drive yang sudah dihubungkan setiap 1000 iterasi. Setelah training data selesai, dilakukan pemrograman sistem deteksi dan pengukuran objek secara realtime yang dapat membaca ukuran panjang ayam broiler. Program tersebut dijalankan melalui Raspberry Pi 4 Model B.

3. Hasil dan Pembahasan

3.1. Hasil Training Data Ayam broiler

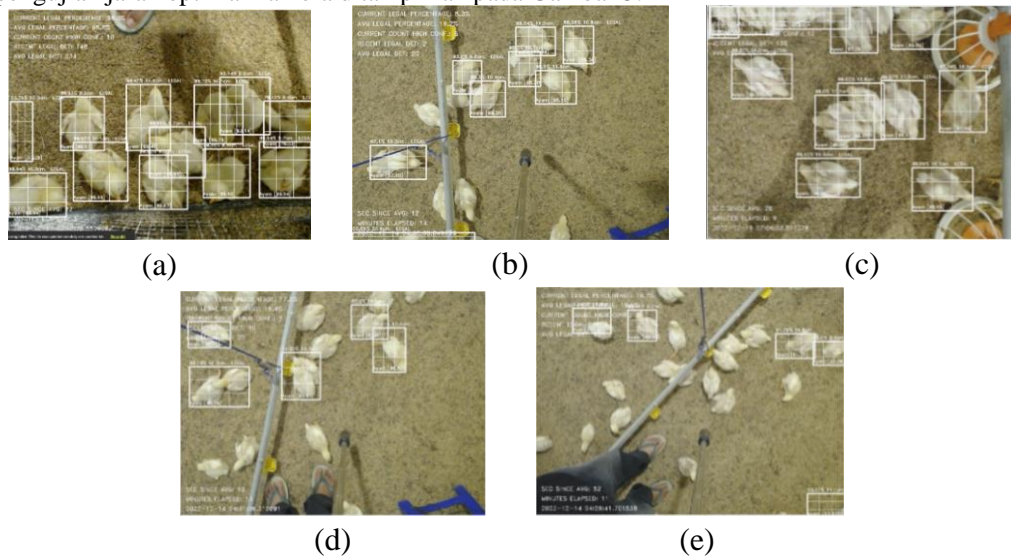
Training data dilakukan menggunakan YOLO v4-tiny sebagai algoritma deteksi objek didapatkan *Mean Average Precision* (mAP) melebihi 50% yang berarti data yang di training mempunyai akurasi yang tinggi sehingga program dapat mengenali objek secara cepat dan akurat.



Gambar 2. Hasil Training Data Ayam Broiler

3.2. Hasil Pengujian Jarak Optimal Kamera

Pengujian jarak kamera dilakukan dengan mengubah jarak kamera dengan objek sesuai dengan variasi jarak yang diuji. Hasil pengujian jarak optimal kamera ditampilkan pada Gambar 3.



Gambar 3. Hasil pengujian jarak kamera pada jarak (a) 50 cm, (b) 75 cm, (c) 100cm, (d) 125 cm (e) 150 cm

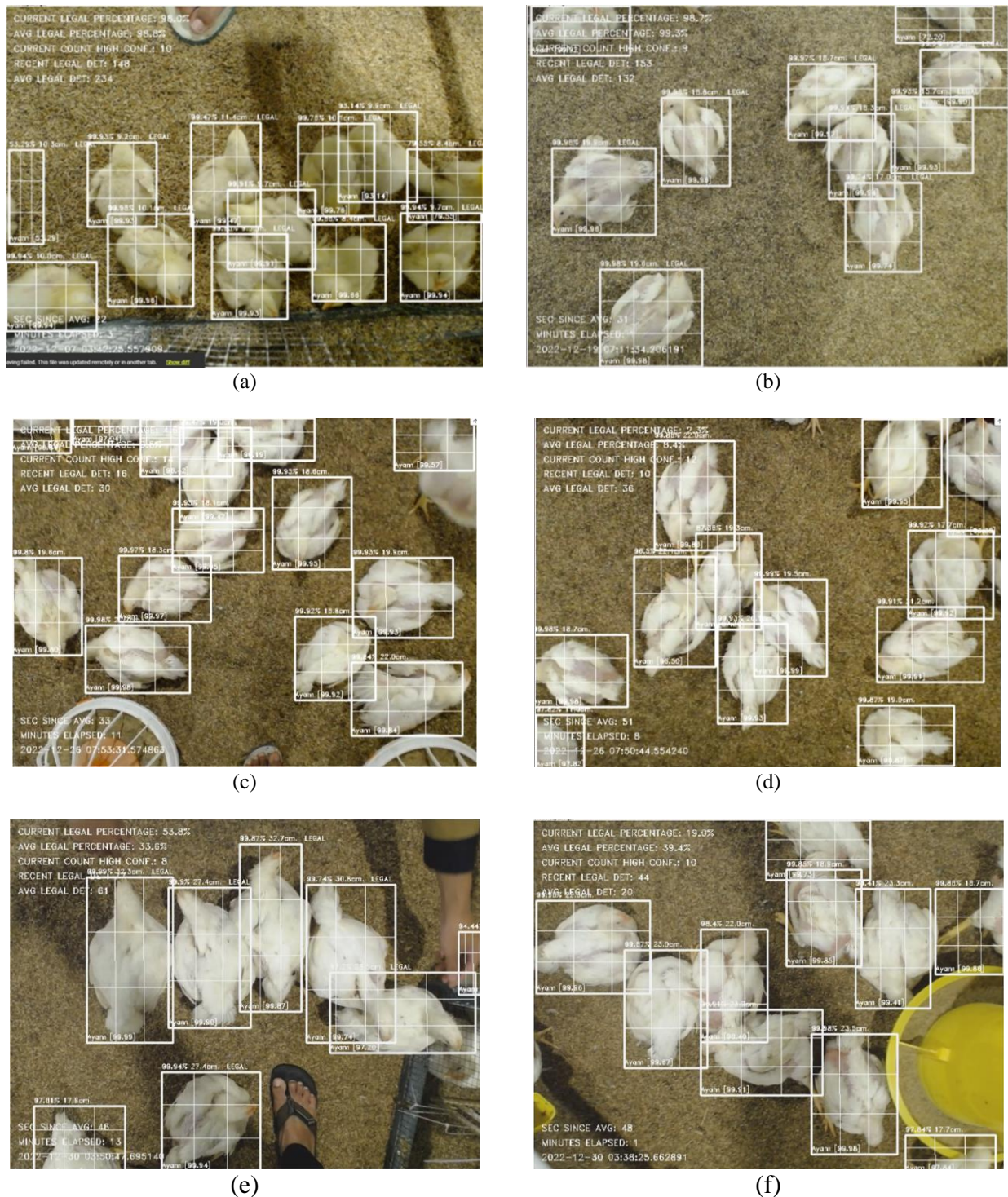
Tabel 1. Pengujian variasi jarak kamera

No	Jarak (cm)	Hasil
1.	50	Berhasil mendeteksi semua objek, konsisten
2.	75	Berhasil mendeteksi semua objek, konsisten
3.	100	Berhasil mendeteksi semua objek, konsisten
4.	125	Berhasil Mendeteksi beberapa objek, kurang konsisten
5.	150	Tidak berhasil mendeteksi semua objek

Berdasarkan hasil pengujian jarak optimal kamera pada Gambar 3 dapat dilihat bahwa program deteksi berhasil mendeteksi semua objek pada variasi jarak 50 –100 cm dan konsisten dalam mendeteksi semua objek. Pada variasi jarak 125 cm, program masih dapat mendeteksi semua objek tetapi tidak konsisten dalam mendeteksi semua objek. Pada variasi jarak 150 dan 170 cm program hanya dapat mendeteksi beberapa objek

3.3. Hasil Pengujian Program *Object Detection*

Pengujian program *object detection* pada ayam broiler dilakukan pada ayam umur 3 - 24 hari. Pengujian program dilakukan dengan membandingkan hasil jumlah deteksi program dengan jumlah ayam sebenarnya. Hasil pengujian ditampilkan dalam Gambar 4. dan Tabel 2.



Gambar 4. Hasil program pendeteksi pada ayam umur (a) 3 hari, (b) 14 hari, (c) 21 hari *frame 1*, (d) 21 hari *frame 2*, (e) 24 hari *frame 1*, (f) 24 hari *frame 2*

Tabel 2. Hasil deteksi program *object detection*

No	Umur Ayam (hari)	Jumlah Deteksi	Jumlah Sebenarnya
1.	3	11	10
2.	14	10	10
3.	21 frame 1	13	14
4.	21 frame 2	12	13
5.	24 frame 1	8	7
6.	24 frame 2	10	10

Berdasarkan hasil pengujian program pada Gambar 4, dapat dilihat hasil deteksi objek ada beberapa yang kurang sesuai dengan jumlah ayam sebenarnya. Ketidaksesuaian jumlah ini diakibatkan ayam yang bergerak terlalu cepat sebelum program objek deteksi mampu untuk memprediksi bahwa objek tersebut adalah ayam. Terjadi juga *false detection* dimana program salah mendeteksi objek dan *beckground* objek nya.

4. Kesimpulan

Pada penelitian perancangan *object detection* menggunakan metode *deep learning* untuk pengukuran ayam broiler berhasil merancang dan membuat sistem *object detection* menggunakan metode *deep learning* untuk ayam broiler. Keseluruhan sistem dapat berjalan dengan baik. YOLO yang digunakan sebagai algoritma deteksi mampu untuk mendeteksi ayam broiler dengan *Mean Average Precision* terkecil sebesar 97,06% dan terbesar 100% yang membuat program deteksi dalam mendeteksi objek cepat dan akurat. YOLO yang digunakan sebagai *object detection* dan *object measurement* dapat berjalan dengan optimal pada jarak 100 cm. Pengujian sistem *object detection* dan *object measurement* pada ayam umur 3 - 24 hari.

5. Daftar Pustaka

- [1] Beddu, H., Aisya, S., Muis, M., Sirajuddin, A., & Abdul, H. M. (2022). Tingkat Pengetahuan Peternak Terhadap Perbedaan Berat Badan Ayam Broiler Dengan Variasi Kelembaban Dalam Sistem Pemeliharaan Closed House. *Jurnal Agrisistem: Seri Sosek dan Penyuluhan*, 18(1), 55-59.
- [2] Nuryati, T. (2019). Analisis Performans Ayam Broiler Pada Kandang Tertutup Dan Kandang Terbuka Performance Analysis Of Broiler In Closed House And Opened House. *Jurnal Peternakan Nusantara*, 5(2), 77-86.
- [3] Wiley, V., & Lucas, T. (2018). Computer vision and image processing: a paper review. *International Journal of Artificial Intelligence Research*, 2(1), 29-36.
- [4] Ray, S. (2019). *A Quick Review of Machine Learning Algorithms*. 2019 *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. doi:10.1109/comitcon.2019.8862451
- [5] Shinde, P. P., & Shah, S. (2018). A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)* (pp. 1-6). IEEE.
- [6] Razzak, M. I., Naz, S., & Zaib, A. (2017). *Deep Learning for Medical Image Processing: Overview, Challenges and the Future*. *Classification in BioApps*, 323–350. doi:10.1007/978-3-319-65981-7_12
- [7] Liu, X., Jia, Z., Hou, X., Fu, M., Ma, L., & Sun, Q. (2019). Real-time marine animal images classification by embedded system based on mobilenet and transfer learning. In *OCEANS 2019-Marseille* (pp. 1-5). IEEE.
- [8] Alwanda, M. R., Ramadhan, R. P. K., & Alamsyah, D. (2020). Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle. *Jurnal Algoritme*, 1(1), 45-56.
- [9] Pathak, A. R., Pandey, M., & Rautaray, S. (2018). Application of deep learning for object detection. *Procedia computer science*, 132, 1706-1717.
- [10] Mogre, N., Bhagat, S., Bhoyar, K., Hadke, H., & Ingole, P. (2022). REAL TIME OBJECT DETECTION AND HEIGHT MEASUREMENT. *International Research Journal of Modernization in Engineering Technology and Science*, 132, 3825–3829.
- [11] Mahesh, M., Reddy, V., Reddy, A., & Reddy, C. (2022). *OBJECT DETECTION AND DIMENSIONING USING OPENCV*, 10(6 June 2022), b60–b68.
- [12] Reddy, B. K., Bano, S., Reddy, G. G., Kommineni, R., & Reddy, P. Y. (2021). Convolutional Network based Animal Recognition using YOLO and Darknet. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)* (pp. 1198-1203). IEEE.
- [13] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo Algorithm Developments. *Procedia Computer Science* 199, 199, 1066–1073. <https://doi.org/10.1016/j.procs.2022.01.135>
- [14] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11), 3212-3232.
- [15] Rohit, S. L., & Tank, B. V. (2018). Iot based health monitoring system using raspberry PI-review. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 997-1002). IEEE.