

**IMPLEMENTASI STRONGSORT DENGAN YOLOV5 UNTUK  
PENDETEKSIAN POSISI BANGKAI AYAM DARI DATA  
VIDEO**

**RAFIE FADLURAHMAN**



**DEPARTEMEN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
INSTITUT PERTANIAN BOGOR  
BOGOR  
2023**

## **PERNYATAAN MENGENAI SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA**

Dengan ini saya menyatakan bahwa skripsi dengan judul “Implementasi StrongSORT dengan YOLOv5 untuk Pendeteksian Posisi Bangkai Ayam dari Data Video” adalah karya saya dengan arahan dari dosen pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, 5 Juni 2023

Rafie Fadlurahman  
G64170074

## **ABSTRAK**

RAFIE FADLURAHMAN. Implementasi StrongSORT dengan YOLOv5 untuk Pendeteksian Posisi Bangkai Ayam dari Data Video. dibimbing oleh AGUS BUONO dan IDAT GALIH PERMANA

Dalam peternakan ayam, jika bangkai ayam tidak dibuang dengan cepat dari kandang ayam, akan meningkatkan resiko tersebarnya penyakit unggas. Oleh karena itu, suatu metode yang dapat mendeteksi bangkai ayam secara cepat akan sangat berguna untuk peternak ayam. Penelitian ini mengajukan untuk menggunakan *Object Tracking* sebagai metode pendeteksian bangkai ayam. Model *Object Tracking* yang digunakan adalah StrongSORT yang dibantu dengan metode *Object Detection* YOLOv5. Data video dari 8 kamera CCTV kandang ayam Fakultas Peternakan akan digunakan sebagai data pelatihan dan pengujian. *Object Tracking* akan dijalankan terhadap data tersebut. Akan disimulasikan data yang akan merepresentasikan data bangkai ayam. Hasil penelitian mendapatkan akurasi klasifikasi terbaik terendah 92% dan tertinggi 99%

Kata Kunci : bangkai ayam, *Object Tracking*, StrongSORT

## **ABSTRACT**

RAFIE FADLURAHMAN. Implementation of StrongSORT with YOLOv5 to Detect Chicken Carcass Position from Video Data. Supervised by AGUS BUONO and IDAT GALIH PERMANA

In poultry farming, if carcasses are not quickly disposed of, they will increase the risk of spreading diseases. Therefore, a method that can quickly detect chicken carcasses will be very useful to poultry farmers. This research proposes to use Object Tracking as said method. StrongSORT is chosen as the object tracking model with the help of YOLO v5 as an Object Detection model. Video Data from 8 CCTV from Faculty of Animal Science's chicken shed will be used as training and testing data. Object Tracking will be executed on said data. Data will be simulated which will represent chicken carcass data. The result of the research the best accuracy of classifiers at worst is 92% and at best 99%

Keywords : chicken carcass, object tracking, StrongSORT

© Hak Cipta milik IPB, tahun 2023  
Hak Cipta dilindungi Undang-Undang

*Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan atau menyebutkan sumbernya. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik, atau tinjauan suatu masalah, dan pengutipan tersebut tidak merugikan kepentingan IPB.*

*Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apa pun tanpa izin IPB.*

**IMPLEMENTASI STRONGSORT DENGAN YOLOV5 UNTUK  
PENDETEKSIAN POSISI BANGKAI AYAM DARI DATA  
VIDEO**

**RAFIE FADLURAHMAN**

Skripsi  
sebagai salah satu syarat untuk memperoleh gelar  
Sarjana pada  
Program Studi Ilmu Komputer

**DEPARTEMEN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
INSTITUT PERTANIAN BOGOR  
BOGOR  
2023**

Penguji pada Ujian Skripsi:  
Dr. Toto Haryanto, S.Kom., M.Si

Judul Skripsi : Implementasi StrongSORT dengan YOLOv5 untuk Pendeteksian  
Posisi Bangkai Ayam dari Data Video  
Nama : Rafie Fadlurahman  
NIM : G64170074

Disetujui oleh

Pembimbing 1:  
Prof. Dr. Ir Agus Buono, M.Si, M.Kom

---

Pembimbing 2:  
Dr. Ir. Idat Galih Permana, M.ScAgr

---

Diketahui oleh

Ketua Departemen Ilmu Komputer:  
Dr. Sony Hartono Wijaya, S.Kom, M.Kom  
NIP 19810809 200812 1 002

---

Tanggal Ujian : 5 Juni 2023

Tanggal Lulus :





## **PRAKATA**

Puji dan syukur penulis panjatkan kepada Allah *subhanaahu wa ta'ala* atas segala karunia-Nya sehingga karya ilmiah ini berhasil diselesaikan. Tema yang dipilih dalam penelitian yang dilaksanakan sejak bulan Januari 2021 sampai bulan Mei 2023 ini ialah Object Tracking, dengan judul “Implementasi StrongSORT dengan YOLOv5 untuk Pendeteksian Posisi Bangkai Ayam dari Data Video”.

Penulisan skripsi ini tidak mungkin dapat diselesaikan tanpa bantuan dari berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih terhadap pihak-pihak berikut:

1. Bapak Prof. Dr. Ir Agus Buono, M.Si, M.Kom sebagai ketua pembimbing dan Bapak Dr. Ir. Idat Galih Permana, M.ScAgr sebagai pembimbing kedua yang telah memberikan arahan dan saran-saran dalam penulisan skripsi ini.
2. Bapak Dr. Toto Haryanto, S.Kom., M.Si. sebagai moderator dan penguji yang juga telah memberikan banyak saran.
3. Teman-teman dari Departemen Ilmu Komputer yang menjadi pembahas dan mendukung penulisan skripsi ini.
4. Ayah, ibu, Uni Icha, Uni Anis serta keluarga besar yang telah memberikan dukungan, arahan dan kasih sayang.
5. Bapak/Ibu dosen dan staf pengajar yang telah memberikan ilmu selama masa perkuliahan ini.
6. Kang Khomeng dan pengurus kandang ayam yang lain yang telah membantu penulis dalam pengambilan data video CCTV kandang ayam.

Semoga karya ilmiah ini bermanfaat bagi pihak yang membutuhkan dan bagi kemajuan ilmu pengetahuan.

Bogor, 5 Juni 2023

*Rafie Fadlurahman*

## DAFTAR ISI

DAFTAR ISI	viii
DAFTAR TABEL	ix
DAFTAR GAMBAR	ix
I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
1.5 Ruang Lingkup Penelitian	2
II TINJAUAN PUSTAKA	3
2.1 DeepSORT	3
2.2 StrongSORT	4
2.3 YOLO v5	4
III METODE	6
3.1 Tahapan Penelitian	6
3.2 Pengumpulan Data	6
3.3 Tracking Menggunakan StrongSORT dengan YOLOV5	7
3.4 Penyisipan Data Simulasi Ayam Mati	8
3.5 Ekstraksi Fitur	8
3.6 Algoritma Classifier	9
3.7 Tuning Hyperparameter	9
3.8 Evaluasi Classifier	9
IV HASIL DAN PEMBAHASAN	10
4.1 Hasil Ekstraksi Fitur	10
4.2 Hasil Tuning Hyperparameter	11
4.3 Evaluasi Model	13
V KESIMPULAN DAN SARAN	16
5.1 Kesimpulan	16
5.2 Saran	16
VI DAFTAR PUSTAKA	17
VII RIWAYAT HIDUP	18

## Daftar Tabel

1	Hasil tuning <i>Random Forest Classifier</i> dengan dataset yang menggunakan simulasi normal pertama	12
2	Hasil tuning <i>Nearest Neighbors Classifier</i> dengan dataset yang menggunakan simulasi normal pertama	12
3	Hasil tuning <i>Support Vector Classifier</i> dengan dataset yang menggunakan simulasi normal pertama	12
4	Hasil tuning <i>Random Forest Classifier</i> dengan dataset yang menggunakan simulasi normal kedua	13
5	Hasil tuning <i>Nearest Neighbors Classifier</i> dengan dataset yang menggunakan simulasi normal kedua	13
6	Hasil tuning <i>Support Vector Classifier</i> dengan dataset yang menggunakan simulasi normal kedua	13
7	<i>Classification Report</i> untuk <i>Random Forest Classifier</i> dengan dataset yang menggunakan simulasi normal pertama	14
8	<i>Classification Report</i> untuk <i>Nearest Neighbors Classifier</i> dengan dataset yang menggunakan simulasi normal pertama	14
9	<i>Classification Report</i> untuk <i>Support Vector Classifier</i> dengan dataset yang menggunakan simulasi normal pertama	14
10	<i>Classification Report</i> untuk <i>Random Forest Classifier</i> dengan dataset yang menggunakan simulasi normal kedua.	15
11	<i>Classification Report</i> untuk <i>Nearest Neighbors Classifier</i> dengan dataset yang menggunakan simulasi normal kedua	15
12	<i>Classification Report</i> untuk <i>Support Vector Classifier</i> dengan dataset yang menggunakan simulasi normal kedua	15
1	Infrastruktur YOLOv5	4
2	Tahapan penelitian	6
3	Contoh data citra yang sudah diekstrak dari video	7
4	Contoh teks hasil tracking	7
5	Hasil tracking setelah disisipkan data simulasi	8
6	Vektor gerakan ayam	9
7	Scatter plot dari fitur ayam dengan simulasi normal pertama yang memperlihatkan seluruh data	10
8	Scatter plot dari fitur ayam dengan simulasi normal pertama dengan focus pada range [0,1-0,8; 0,1-0,6]	10
9	Scatter plot dari fitur ayam dengan simulasi normal pertama dengan focus pada data mendekati titik (0,0)	11
10	Scatter plot data dengan simulasi normal kedua	11

# I PENDAHULUAN

## 1.1 Latar Belakang

Selama lima tahun terakhir secara makro Indonesia mampu menyediakan pangan yang cukup untuk memenuhi kebutuhan warganya, namun kualitas konsumsi pangan rata-rata masyarakat Indonesia masih di bawah rekomendasi para ahli gizi (Suryana, 2014).

Konsumsi ayam ras atau broiler per kapita penduduk lebih tinggi dari daging lainnya dan setiap tahun selalu mengalami kenaikan. Rata-rata konsumsi rumah tangga per kapita seminggu tahun 2020 untuk daging ayam ras meningkat sebesar 6,42 persen (Kementerian Pertanian, 2021). Akibatnya produksi daging ayam ras akan perlu dinaikkan di tahun berikutnya agar dapat memenuhi permintaan. Peningkatan produksi ayam broiler perlu didampingi dengan pengelolaan yang baik agar kualitas daging tidak menurun. Salah satu masalah dalam pengelolaan ternak ayam adalah pemantauan kesehatan ayam.

Menurut panduan peternakan ayam Food and Agriculture Organization of United Nations (FAO), bangkai ayam harus segera disingkirkan dari kandang secepat mungkin dengan cara dikuburkan untuk mencegah tersebarnya penyakit. (Sonaiya dan Swan, 2004)

Salah satu cara memandang masalah menemukannya lokasi bangkai ayam adalah dengan memandangnya sebagai masalah *Multi Object Tracking*. MOT adalah suatu masalah mengidentifikasi dan melacak pergerakan beberapa benda di dalam suatu video secara akurat. MOT juga sebelumnya sudah diterapkan di bidang peternakan. Cowton et al. (2019) mengusulkan metode memonitor perilaku babi secara individual menggunakan *object tracking*. Khairunissa (2020) mengimplementasikan *multi-object tracking* untuk menganalisis perilaku ayam. Ali et al. (2022) mengusulkan algoritma berbasis *Faster R-CNN* untuk mendeteksi dan melacak unggas.

Salah satu metode Object Tracking adalah StrongSORT (Yunhao et al 2022). StrongSORT adalah model Object Tracking baru yang didapat dari penyempurnaan DeepSORT yang sebelumnya telah diusulkan oleh Wojke et al. (2017). StrongSORT adalah metode object tracking yang bekerja dengan cara tracking-by-detection dan maka dari itu, memerlukan algoritma tambahan yang akan mendeteksi benda yang akan dilakukan object tracking. Dalam implementasi sebelumnya, metode ini digunakan dengan Metode Detection YOLOv5. StrongSORT dipilih karena StrongSORT adalah salah satu algoritma object tracking terbaik yang juga bersifat real time. Dengan ini, program dapat melakukan pendeteksian bangkai ayam secara *real time* dan langsung dari video stream CCTV kandang ayam.

Dalam *Object Detection*, YoloV5 hanya menentukan keberadaan ayam di posisi tertentu pada gambar. YoloV5 belum bisa menentukan apakah ayam tersebut hidup atau mati. Hal ini disebabkan karena penampilan ayam yang baru mati tidak jauh dari penampilan ayam yang masih hidup. Untuk itu akan diperlukan suatu metode yang dapat membedakan bangkai ayam dengan ayam hidup menggunakan sifat selain dari penampilannya.

Salah satu sifat yang dapat membedakan ayam hidup dengan ayam mati adalah pergerakannya. Ayam yang hidup walaupun tidak banyak bergerak dalam lingkungan kandang, akan sewaktu-waktu bergerak jika terganggu oleh ayam lain atau saat ingin makan atau minum. Sedangkan ayam yang mati tidak akan bergerak sama sekali kecuali dipindahkan oleh pengurus kandang. Oleh karena itu, kecepatan dari suatu ayam dapat digunakan untuk menentukan apakah ayam tersebut hidup atau mati.

## **1.2 Perumusan Masalah**

Perumusan masalah penelitian ini adalah sebagai berikut:

1. Bagaimana performa *tracking* ayam dengan menggunakan StrongSORT?
2. Peubah apa yang akan menjadi patokan sebagai pembeda ayam hidup dengan ayam mati?
3. Apakah bangkai ayam dapat dideteksi secara akurat?

## **1.3 Tujuan Penelitian**

Penelitian ini bertujuan untuk mengimplementasi StrongSORT dengan YOLOv5 untuk mendeteksi keberadaannya bangkai ayam dan mengevaluasi performanya.

## **1.4 Manfaat Penelitian**

Pendeteksian keberadaan dan posisi ayam mati dapat membantu pengelola kandang ayam untuk lebih cepat menyingkirkan bangkai ayam tersebut. Jika akurasi prediksi cukup baik, bisa juga digunakan untuk menghitung jumlah ayam yang mati pada jangka waktu tertentu.

## **1.5 Ruang Lingkup Penelitian**

Kandang ayam broiler milik Fakultas Peternakan (FAPET) serta sistem CCTV yang sudah dipasang di tempat tersebut akan digunakan dalam penelitian ini. Data berupa video-video dari CCTV di dalam kandang ayam milik FAPET IPB. Citra yang didapat berwarna dari pagi hingga sore karena lampu kandang dinyalakan sedangkan saat gelap citra akan berupa monokrom.

## II TINJAUAN PUSTAKA

### 2.1 DeepSORT

DeepSORT adalah algoritma tracking online yang menggunakan *framework Simple Online and Realtime Tracking* dan model CNN yang sudah dilatih terlebih dahulu. DeepSORT bekerja dengan metode tracking-by-detection yang mempertimbangkan bounding box hasil deteksi object dan informasi visual objek yang sedang dilacak. Algoritma ini adalah algoritma tracking online yang berarti untuk membuat prediksi, algoritma ini hanya mempertimbangkan informasi citra kini dan yang di masa lalu.

Di awal rekaman, yaitu, di frame pertama, ID trek unik ditetapkan ke setiap kotak pembatas yang mewakili ayam dan memiliki nilai kepercayaan lebih tinggi dari ambang batas yang ditetapkan. Algoritma Hungarian digunakan untuk menetapkan deteksi dalam bingkai baru ke trek yang ada sehingga fungsi biaya penugasan mencapai minimum global.

Fungsi biaya melibatkan jarak spasial (Mahalanobis)  $d^{(1)}$  dari kotak pembatas yang terdeteksi dari posisi yang diprediksi menurut posisi objek yang diketahui sebelumnya, dan jarak visual  $d^{(2)}$  yang mempertimbangkan penampilan objek yang terdeteksi dan sejarah penampilan objek yang dilacak. Fungsi biaya untuk menetapkan objek yang terdeteksi ke trek diberikan oleh fungsi:

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1 - \lambda) d^{(2)}(i,j) \quad (1)$$

Dimana  $\lambda$  adalah parameter yang dapat diatur untuk menentukan pengaruh jarak spasial  $d^{(1)}$  dan jarak visual  $d^{(2)}$ .

Jarak spasial  $d^{(1)}$  didapat dari fungsi:

$$d^{(1)}(i,j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \quad (2)$$

Dimana  $y_i$  dan  $S_i$  mewakili rata-rata dan pengamatan kotak pembatas matriks kovarians untuk track ke- $i$ , dan  $d_j$  mewakili bounding box yang terdeteksi ke- $j$ .

Jarak visual  $d^{(2)}$  bergantung pada deskriptor fitur penampilan dan diberikan oleh ekspresi:

$$d^{(2)}(i,j) = \min \left\{ 1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in \mathcal{R}_i \right\} \quad (3)$$

Dimana  $r_j$  adalah deskriptor tampilan yang diekstraksi dari bagian citra di dalam bounding box yang terdeteksi ke- $j$ , dan  $\mathcal{R}_i$  adalah kumpulan dari 100 deskriptor penampilan terakhir  $r_k^{(i)}$  asosiasikan dengan track  $i$ .

Pengukuran  $d^{(2)}$  menggunakan jarak kosinus antara deteksi ke- $j$  dan jalur ke- $i$  untuk memilih jalur yang secara visual mendeteksi deteksi yang paling mirip sebelumnya dengan deteksi saat ini.

Pendeskripsi tampilan diekstraksi menggunakan jaringan saraf sisa lebar yang terdiri dari dua lapisan konvolusi diikuti oleh enam blok sisa yang menghasilkan vektor 128 elemen. Jaringan tersebut telah dilatih sebelumnya pada kumpulan data identifikasi ulang seseorang lebih dari satu juta citra dari 1.261 pejalan kaki (Wojke et al., 2017). Vektor fitur dinormalisasi agar sesuai dengan hypersphere unit sehingga jarak kosinus dapat digunakan.

Track ID baru dihasilkan setiap kali ada lebih banyak deteksi dalam bingkai daripada pemain yang sudah dilacak, yaitu, ketika pemain baru terdeteksi dalam bingkai. Track baru juga dibuat ketika deteksi tidak dapat ditetapkan ke track yang

ada, karena deteksi terlalu jauh dari trek yang ada, atau tidak mirip secara visual dengan deteksi sebelumnya. Ini dikendalikan oleh ambang batas yang menetapkan  $d^{(1)}$  dan  $d^{(2)}$  maksimum saat penugasan masih memungkinkan. Sebuah track dapat ditinggalkan jika tidak ada penugasan yang dibuat untuk itu selama  $n$  frame berturut-turut. Dalam hal ini, track ID baru akan ditetapkan jika objek yang sama muncul kembali nanti di video.

Informasi penampilan digunakan khususnya untuk membantu dalam re-identifikasi dan mencegah pembuatan ID baru untuk objek yang tidak terlacak selama beberapa waktu, baik karena mereka terhalang dari pandangan kamera, keluar dari pandangan kamera, maupun karena tidak terdeteksi akibat kesalahan detektor.

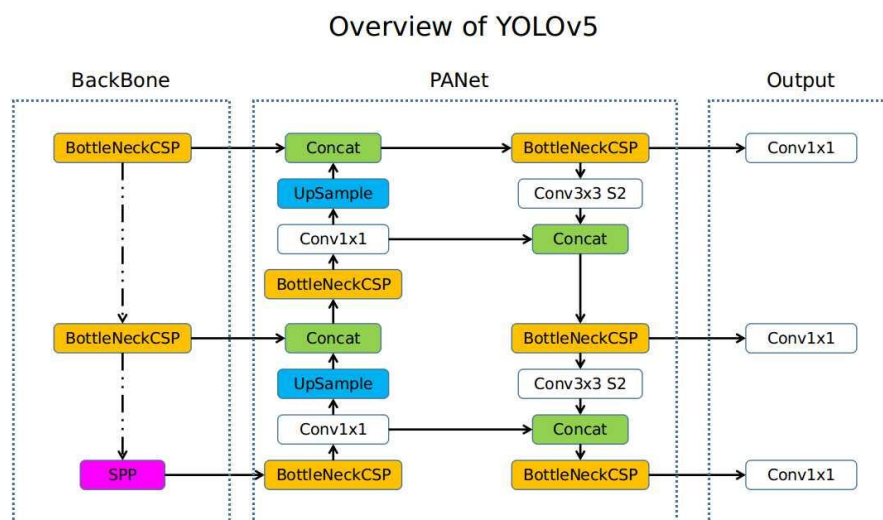
## 2.2 StrongSORT

Peningkatan StrongSORT terhadap DeepSORT dapat dibagi menjadi dua cabang: Untuk cabang penampilan peningkatan dilakukan dengan menggantikan CNN dengan ekstraktor fitur penampilan ResNeSt50 (Zhang et al, 2004) yang digabungkan dengan trik-trik pelatihan yang digunakan oleh studi lainnya (Luo et al, 2015). Selain itu StrongSORT juga menggantikan bank fitur DeepSORT dengan strategi pembaharuan fitur. Dalam cabang gerakan, Enhanced Correlation Coefficient Maximization digunakan untuk mengkompensasi gerakan kamera.

## 2.3 YOLO v5

YOLO v5 adalah detektor objek satu tahap yang memiliki tiga bagian penting seperti detektor objek satu tahap lainnya.

1. Tulang Punggung Model
2. Leher Model
3. Kepala Model



Gambar 1 Infrastruktur YOLOv5

Tulang Punggung Model terutama digunakan untuk mengekstrak fitur penting dari citra input yang diberikan. Di YOLO v5, *Cross Stage Partial Networks* (CSP) digunakan sebagai tulang punggung untuk mengekstrak kaya fitur informatif dari citra input. CSPNet telah menunjukkan peningkatan yang signifikan dalam waktu pemrosesan dengan jaringan yang lebih dalam.

Leher Model terutama digunakan untuk menghasilkan piramida fitur. Piramida fitur membantu model untuk digeneralisasi dengan baik pada penskalaan objek. Struktur ini akan membantu dalam mengidentifikasi objek yang sama dengan ukuran dan skala yang berbeda.

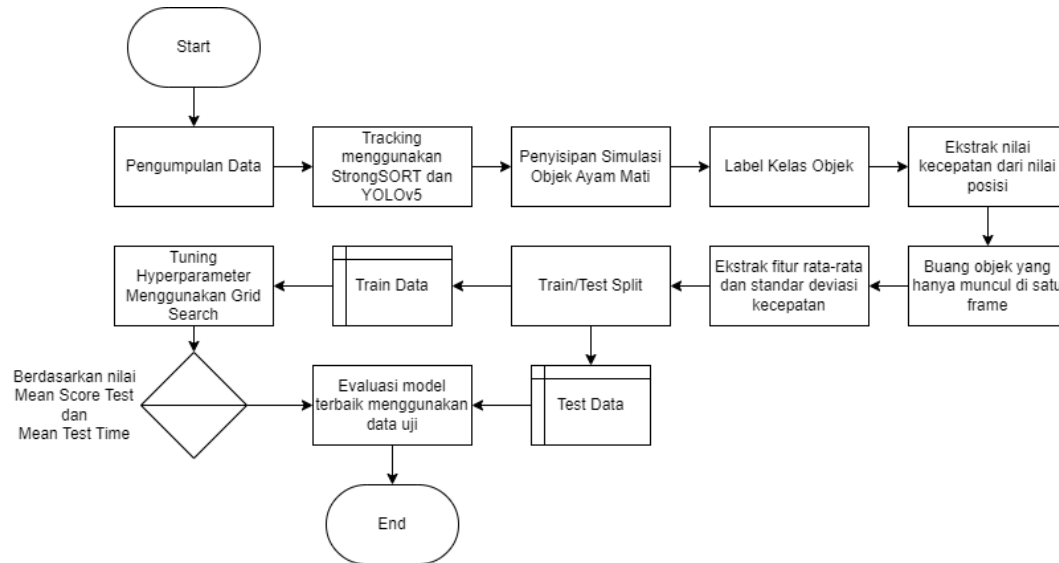
Piramida fitur sangat berguna dan membantu model untuk bekerja dengan baik pada data yang tidak terlihat. Ada model lain yang menggunakan berbagai jenis teknik piramida fitur seperti FPN, BiFPN, PANet, dll. Di YOLO v5 PANet digunakan sebagai leher untuk mendapatkan fitur piramida.

Kepala Model terutama digunakan untuk melakukan bagian deteksi akhir. Ini menerapkan kotak jangkar pada fitur dan menghasilkan vektor keluaran akhir dengan probabilitas kelas, skor objektivitas, dan kotak pembatas. Pada kepala model YOLO v5 sama dengan versi YOLO V3 dan V4 sebelumnya.



### III METODE

#### 3.1 Tahapan Penelitian



Gambar 2 Tahapan Penelitian

Tahapan penelitian dimulai dengan pengumpulan data, praproses data, pelatihan model, dan diakhiri dengan evaluasi model.

#### 3.2 Pengumpulan Data

Data penelitian yang akan digunakan adalah data primer yang diambil dari kamera kandang ayam FAPET. Kamera yang digunakan disambung ke Digital Video Recorder DVR-216G-K1. Data memiliki dua bentuk yaitu berbentuk video dan citra yang diakses secara online melalui aplikasi HiLook. Tampak dari data video dan citra dapat berupa berwarna dan monokrom, tergantung dengan waktu pengambilan data. Data citra yang akan digunakan dalam pelatihan model akan dianotasi menggunakan *website* anotasi citra yang disediakan Roboflow. Untuk data video yang akan dipakai untuk klasifikasi ayam mati, video yang diambil akan ada setidaknya satu ayam yang mati.



Gambar 3 Contoh Data citra yang sudah diekstrak dari video

### 3.3 Tracking Menggunakan StrongSORT dengan YOLOV5

*Tracking* dilakukan kepada data video yang sudah dikumpulkan. Untuk YOLOV5 menggunakan weight yang sudah dilatih di dalam riset ChickTrack (Neethirajan, 2022). Untuk model ReID, akan digunakan model Resnet50 yang sudah disediakan di dalam repositori. Hasil *tracking* akan berupa text yang berisikan data objek yang ditemukan di setiap frame video. Bagian Object Detection dan Tracking hanya menentukan apakah suatu objek adalah ayam atau tidak, dan belum menentukan apakah ayam tersebut hidup atau mati.

```
3 1 193 741 71 106 -1 -1 -1 0
3 2 192 894 68 84 -1 -1 -1 0
3 3 53 707 88 120 -1 -1 -1 0
3 4 593 747 53 92 -1 -1 -1 0
3 5 172 1011 72 68 -1 -1 -1 0
3 6 265 810 66 81 -1 -1 -1 0
3 9 153 918 64 88 -1 -1 -1 0
3 10 414 814 68 91 -1 -1 -1 0
3 11 705 741 53 64 -1 -1 -1 0
```

Gambar 4 Contoh teks hasil tracking

Setiap baris teks akan berisikan 10 nilai yang merepresentasikan suatu *bounding box*. Kolom pertama adalah nomor frame kemunculan objek. Kolom kedua adalah nomor ID objek dari *bounding box* yang ditemukan. Kolom ketiga dan keempat merupakan nilai posisi pojok kiri atas suatu *bounding box*, posisi horizontal untuk kolom ketiga, dan posisi vertikal untuk kolom keempat. Kolom kelima dan keenam merepresentasikan besar dari *bounding box*, lebar untuk kolom kelima dan tinggi untuk kolom keenam. Kolom lainnya tidak dipakai oleh StrongSORT, hanya sebagai format untuk MOT.

Karena tidak dapat diverifikasi apakah ada ayam mati di video yang didapatkan, akan dianggap bahwa semua ayam yang dideteksi dan dilakukan tracking pada tahap ini adalah ayam hidup, untuk data ayam mati akan dijelaskan di tahap berikut.

### 3.4 Penyisipan Data Simulasi Ayam Mati

Sebelum dilakukan *fitting*, akan disisipkan data yang mensimulasikan objek ayam mati ke dalam semua hasil *tracking*. Untuk setiap hasil *tracking* akan disisipkan satu objek simulasi ini. Posisi dan ukuran dari *bounding box* objek tersebut akan diambil dari rata-rata posisi dan ukuran ayam-ayam yang berada di video. Ayam yang hidup akan diberikan label 0 sedangkan ayam mati diberikan label 1.

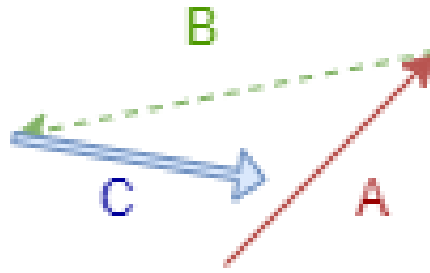
3	0	383	812	83	121	1
3	1	438	769	88	137	0
3	3	501	829	83	108	0
3	4	409	321	48	97	0
4	0	383	812	83	121	1
4	1	437	768	88	138	0
4	3	501	830	82	106	0
4	4	408	320	50	98	0
5	0	383	812	83	121	1
5	1	437	768	89	138	0
5	3	502	830	82	106	0

Gambar 5 Hasil tracking setelah disisipkan data simulasi

Selain itu juga diciptakan data yang menirukan data yang sebelumnya dihasilkan dari rata-rata posisi dan besar ayam hidup menggunakan distribusi normal. Data yang disimulasikan dengan cara ini akan dilakukan dua kali dengan parameter berbeda. Untuk simulasi dengan distribusi normal pertama, standar deviasi dari fitur *mean* dan *std* akan dinaikkan sebesar 0.01. Untuk yang kedua, fitur *mean* akan disimulasikan dengan rata-rata 0,2 dan standar deviasi 0,2, sedangkan fitur *std* akan memiliki rata-rata 0,1 dan standar deviasi 0,1. Data ayam mati simulasi normal kedua akan lebih tersebar dan memiliki daerah yang tumpang tindih dengan data ayam hidup daripada simulasi normal pertama. Hal ini dilakukan untuk mensimulasikan ketidakakuratan *Object Tracking*.

### 3.5 Ekstraksi Fitur

Dari hasil *tracking* tersebut, akan dihitung nilai kecepatan objek di setiap frame. kecepatan objek untuk frame pertama akan diberikan nilai 0. Lalu akan diambil rata-rata dan standar deviasi dari kecepatan objek sebagai fitur yang akan dipertimbangkan dalam klasifikasi. Untuk meningkatkan akurasi klasifikasi dan karena standar deviasi suatu data yang hanya memiliki satu datapoint bernilai undefined, objek yang hanya muncul dalam satu frame tidak akan dilakukan ekstraksi fitur. Kecepatan memiliki satuan pixel per frame.



Gambar 6 Vektor gerakan ayam

Gambar diatas adalah contoh gerakan suatu objek ayam. Panah menunjukan perubahan posisi objek dalam satu frame. Dari data ini dapat diambil kecepatannya dengan dengan fungsi:

$$v_n = \frac{x_n - x_{n-1}}{t}$$

Dimana  $v_n$  merupakan kecepatan objek pada deteksi ke- $n$ ,  $x_n$  adalah posisi objek pada deteksi ke  $n$ , dan  $t$  adalah selisih waktu dalam satuan frame antara  $n$  dan  $n-1$ . Umumnya  $t$  bernilai 1, tetapi bisa bernilai lebih jika terdapat jangka waktu dimana objek tidak terdeteksi. Dari nilai ini dapat diambil rata-rata dan standar deviasinya untuk dijadikan sebagai fitur yang akan dibandingkan oleh *classifier*.

### 3.6 Algoritma Classifier

Penelitian ini akan mempertimbangkan 3 algoritma yaitu K-Nearest Neighbour Classifier, Random Forest Classifier, dan Support Vector Machine Classifier. Data akan dibagi menjadi data latih dan data uji dengan rasio 7:3.

### 3.7 Tuning Hyperparameter

Tuning hyperparameter akan dilakukan menggunakan metode grid search cross validation dengan 5-fold cross validation. Karena data sangat tidak seimbang dimana jumlah kelas “ayam hidup” jauh lebih banyak dari pada “ayam mati”, akan digunakan nilai balanced accuracy score untuk scoring. Balanced accuracy adalah rata-rata dari recall masing-masing kelas.

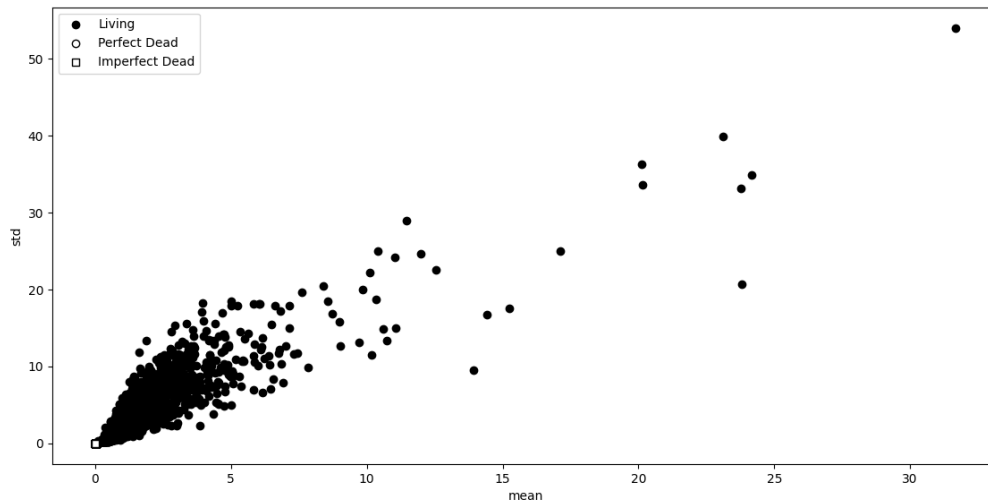
### 3.8 Evaluasi Classifier

Classifier terbaik lalu akan dipilih untuk dinilai performanya dalam melakukan klasifikasi terhadap data latih yang didapat dari hasil train/test splitting sebelumnya. Classifier terbaik adalah classifier dengan nilai test dari grid search yang paling tinggi. Jika terjadi seri, classifier dengan waktu test terkecil yang akan diambil.

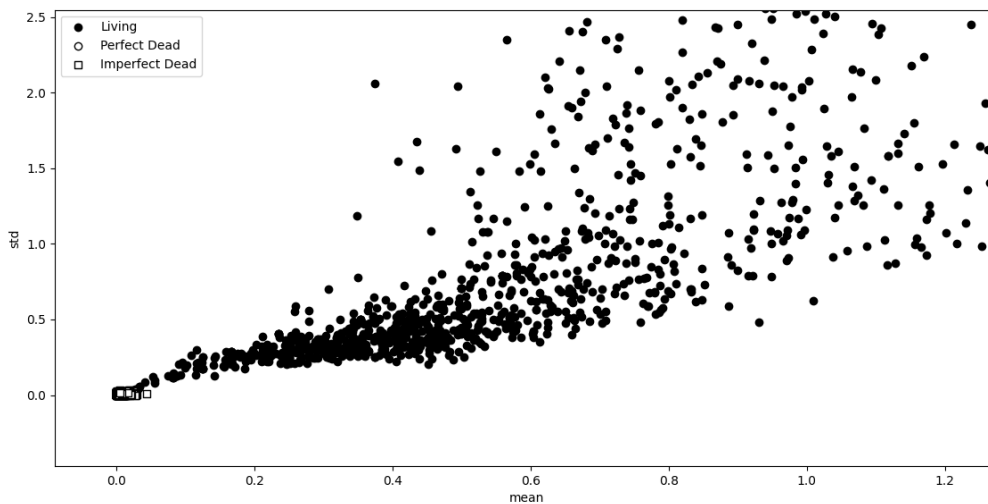
## IV HASIL DAN PEMBAHASAN

### 4.1 Hasil Ekstraksi Fitur

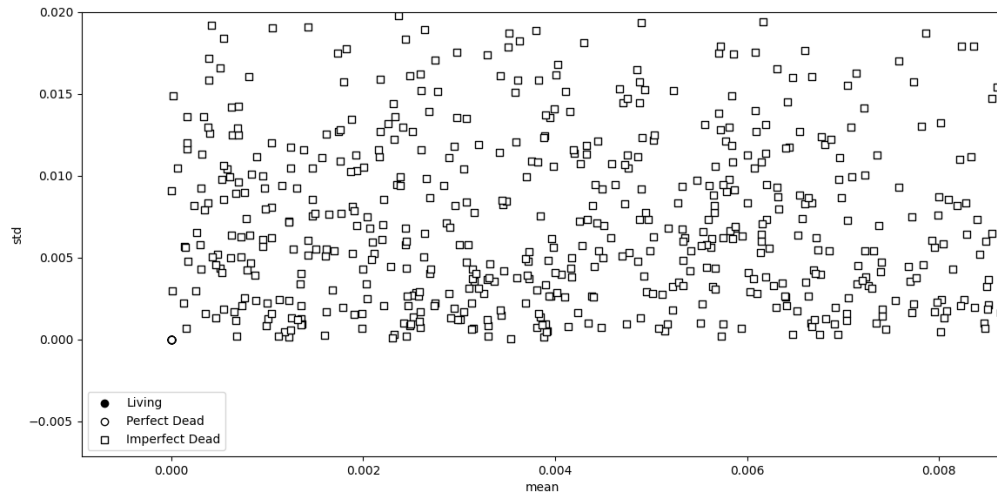
Grafik-grafik berikut akan menampilkan *scatter plot* dimana setiap *datapoint* merepresentasikan objek ayam. Data yang dilabelkan ‘Living’ merepresentasikan ayam hidup, ‘Perfect Dead’ adalah simulasi ayam mati yang sempurna dimana objeknya tidak bergerak sama sekali, dan ‘Imperfect Dead’ adalah simulasi ayam mati dimana nilai standar deviasi fitur dinaikkan. Mayoritas *datapoint* ayam hidup berada di range  $[0,1-0,8; 0,1-0,6]$ . Datapoint objek ‘perfect dead’ berada pada titik (0,0)



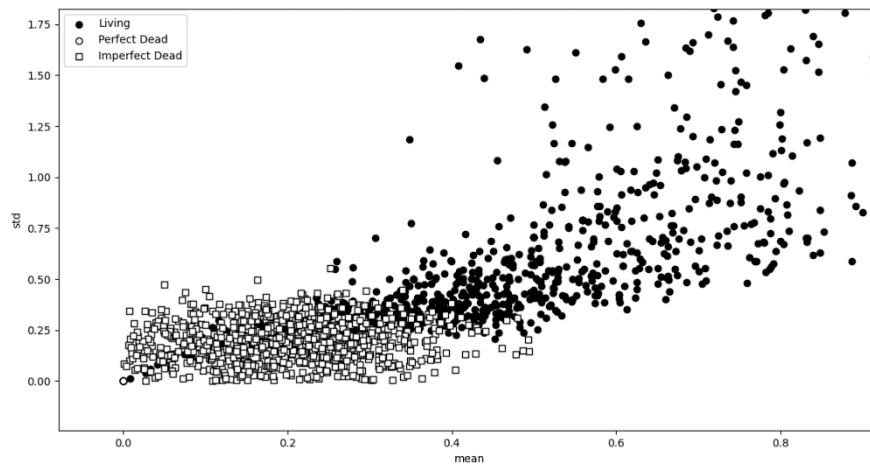
Gambar 7 Scatter plot dari fitur ayam dengan simulasi normal pertama yang memperlihatkan seluruh data



Gambar 8 Scatter plot dari fitur ayam dengan simulasi normal pertama dengan focus pada range  $[0,1-0,8; 0,1-0,6]$



Gambar 9 Scatter plot dari fitur ayam dengan simulasi normal pertama dengan focus pada data mendekati titik (0,0)



Gambar 10 Scatter plot data dengan simulasi normal kedua

## 4.2 Hasil Tuning Hyperparameter

Metode *Grid Search* akan membandingkan 18 konfigurasi *hyperparameter* untuk *Random Forest Classifier*, 8 konfigurasi *hyperparameter* untuk *Nearest Neighbors Classifier*, dan 24 konfigurasi *hyperparameter* untuk *Random Forest Classifier*.

Tabel-tabel berikut akan menampilkan hasil dari tuning *hyperparameter* dengan dataset yang menggunakan simulasi normal pertama. *Classifier* yang dimasukan ke dalam tabel adalah 5 *classifier* dengan nilai mean test score terbaik. Karena nilai *mean test score* untuk algoritma *random forest classifier* sama untuk kelimanya, akan diambil berdasarkan nilai *mean score time* terendah.

Tabel 1 Hasil tuning Random Forest Classifier dengan dataset yang menggunakan simulasi normal pertama

Hyperparameter			Mean Score Time	Mean Test Score
Class weight	Max Depth	n-Estimators		
None	None	2	0.002	0.997
Balanced	None	2	0.002	0.997
Balanced	5	2	0.002	0.997
Balanced	3	2	0.002	0.997
None	3	2	0.003	0.997

Tabel 2 Hasil tuning Nearest neighbors Classifier dengan dataset yang menggunakan simulasi normal pertama

Hyperparameter		Mean Score Time	Mean Test Score
n-Neighbors	Weights		
20	Distance	0.004	0.995
20	Uniform	0.025	0.994
5	Distance	0.004	0.962
5	Uniform	0.024	0.962
10	Distance	0.004	0.962

Tabel 3 Hasil tuning Support Vector Classifier dengan dataset yang menggunakan simulasi normal pertama

Hyperparameter			Mean Score Time	Mean Test Score
C	Class weight	Kernel		
100	None	Linear	0.002	0.995
10	Balanced	Linear	0.002	0.994
100	Balanced	Linear	0.002	0.994
10	None	Linear	0.002	0.994
100	None	Sigmoid	0.003	0.994

Tabel-tabel berikut akan menampilkan hasil dari tuning *hyperparameter* dengan dataset yang menggunakan simulasi normal kedua.

Tabel 4 Hasil tuning Random Forest Classifier dengan dataset yang menggunakan simulasi normal kedua

Hyperparameter				
Class weight	Max depth	n- Estimators	Mean score time	Mean test score
Balanced	3	10	0,006123	0,913321
Balanced	5	5	0,005641	0,913184
Balanced	5	10	0,006028	0,911646
Balanced	3	5	0,0067	0,911459
None	5	10	0,007532	0,908642

Tabel 5 Hasil tuning Nearest neighbors Classifier dengan dataset yang menggunakan simulasi normal kedua

Hyperparameter			
n-Neighbors	Weights	Mean Score Time	Mean Test Score
20	Distance	0,020067	0,900854
10	Distance	0,010143	0,900468
20	Uniform	0,056101	0,896711
5	Uniform	0,062549	0,893142
3	Uniform	0,041937	0,891354

Tabel 6 Hasil tuning Support Vector Classifier dengan dataset yang menggunakan simulasi normal kedua

Hyperparameter				
C	Class weight	Kernel	Mean Score Time	Mean Test Score
10	Balanced	RBF	0,055606	0,920119
100	Balanced	RBF	0,042525	0,918733
1	None	RBF	0,056682	0,917805
10	None	RBF	0,062444	0,911093
100	None	RBF	0,038175	0,907939

### 4.3 Evaluasi Model

Berikut adalah tabel-tabel hasil pengujian classifier terbaik. Untuk *Nearest Neighbors Classifier* dan *Support Vector Classifier* berdasarkan nilai akurasiya sedangkan untuk *Random Forest Classifier* karena banyak yang seri, berdasarkan kecepatan pengujian di tahap *tuning hyperparameter*.



Tabel 7 Classification report untuk Random Forest Classifier dengan dataset yang menggunakan simulasi normal pertama

Class	Precision	Recall	F1 score	Support
0	1.00	0.99	0.99	569
1	0.85	0.97	0.91	36
Accuracy			0.99	605
Macro Average	0.93	0.98	0.95	605
Weighted Average	0.99	0.99	0.99	605

Tabel 8 Classification report untuk Nearest Neighbors Classifier dengan dataset yang menggunakan simulasi normal pertama

Class	Precision	Recall	F1 score	Support
0	1.00	0.99	0.99	569
1	0.86	1.00	0.92	36
Accuracy			0.99	605
Macro Average	0.93	0.99	0.96	605
Weighted Average	0.99	0.99	0.99	605

Tabel 9 Classification report untuk Support Vector Classifier dengan dataset yang menggunakan simulasi normal pertama

Class	Precision	Recall	F1 score	Support
0	1.00	0.99	0.99	569
1	0.84	1.00	0.91	36
Accuracy			0.99	605
Macro Average	0.92	0.99	0.95	605
Weighted Average	0.99	0.99	0.99	605

Dapat dilihat bahwa akurasi dari ketiga *classifier* dengan nilai 99%. Ada juga *classifier* dengan nilai *recall* untuk kelas ayam mati 100% yang menunjukkan bahwa semua data ayam mati klasifikasikan dengan benar sebagai ayam mati.

Selain itu ada juga hasil dari klasifikasi untuk dataset yang menggunakan simulasi normal kedua. Hasil dari klasifikasi dataset tersebut dapat dilihat di tabel-tabel berikut:

Tabel 10 Classification report untuk Random Forest Classifier dengan dataset yang menggunakan simulasi normal kedua.

Class	Precision	Recall	F1 score	Support
0	0.98	0.89	0.93	568
1	0.81	0.97	0.89	294
Accuracy			0.92	862
Macro Average	0.9	0.93	0.91	862
Weighted Average	0.93	0.92	0.92	862

Tabel 11 Classification report untuk Nearest Neighbors Classifier dengan dataset yang menggunakan simulasi normal kedua

Class	Precision	Recall	F1 score	Support
0	0.95	0.93	0.94	568
1	0.87	0.9	0.88	294
Accuracy			0.92	862
Macro Average	0.91	0.92	0.91	862
Weighted Average	0.92	0.92	0.92	862

Tabel 12 Classification report untuk Support Vector Classifier dengan dataset yang menggunakan simulasi normal kedua

Class	Precision	Recall	F1 score	Support
0	1	0.84	0.91	568
1	0.76	1	0.86	294
Accuracy			0.89	862
Macro Average	0.88	0.92	0.89	862
Weighted Average	0.92	0.89	0.9	862

## V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Ketiga classifier menghasilkan nilai akurasi sebesar 99% untuk dataset pertama dan akurasi tertinggi 92% untuk dataset kedua. Nearest Neighbors Classifier memiliki nilai Precision dan Recall terbesar untuk kelas ayam mati. Dari 15 classifier yang dengan nilai tertinggi, rata-rata waktu tertinggi untuk melakukan fitting saat dilakukannya Hyperparameter Tuning adalah 0.024 detik. Dengan ini, dapat disimpulkan bahwa *object tracking* menggunakan StrongSORT dengan YOLOv5 dapat digunakan untuk mendeteksi ayam mati secara akurat.

### 5.2 Saran

Penelitian ini hanya mengamati akurasi klasifikasi dari ayam yang sudah terdeteksi. Ada kemungkinan terdapat ayam yang tidak terdeteksi sebagai objek oleh objek detector. Hal ini dapat mengakibatkan tingkat akurasi dari metode ini untuk menjadi lebih rendah. Untuk itu perlu dilakukan pelatihan object detector yang disesuaikan dengan lingkungan dan sistem kamera yang digunakan dalam kandang ayam.

Selain itu, Ultralytics juga sudah menerbitkan versi terbaru dari YOLO yaitu YOLOv8. Versi ini tidak digunakan untuk penelitian ini karena weight yang digunakan untuk object detection memakai YOLOv5. Jika penelitian selanjutnya akan melatih model YOLO sendiri, akan lebih baik menggunakan versi YOLO yang terbaru untuk meningkatkan tingkat akurasi dan kecepatan model.

## VI DAFTAR PUSTAKA

Ali R, Dorozynski M, Stracke J, Mehlretter M. 2022. Deep Learning-Based Tracking of Multiple Objects in the Context of Farm Animal Ethology. *Int Arch Photogramm Remote Sens Spat Inf Sci*. XLIII-B2-2022:509–516. doi:10.5194/isprs-archives-xliii-b2-2022-509-2022.

Broström, M. Real-time multi-object, segmentation and pose tracking using Yolov8 with DeepOCSORT and LightMBN (Version 8.0) [Computer software]. <https://doi.org/https://zenodo.org/record/7629840>

Cowton J, Kyriazakis I, Bacardit J. 2019. Automated Individual Pig Localisation, Tracking and Behaviour Metric Extraction Using Deep Learning. *IEEE Access*. 7:108049–108060. doi:10.1109/ACCESS.2019.2933060.

Host K, Ivašić-Kos M, Pobar M. 2020. Tracking handball players with the Deepsort algorithm. Di dalam: *ICPRAM 2020 - Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods*. hlm 593–599.

Hou X, Yi W, Lap-Pui C. 2019. Proceedings of AVSS 2018 : 2018 IEEE International Conference on Advanced Video and Signal-based Surveillance : Auckland, New Zealand, 27-30 November 2018. *2019 16th IEEE Int Conf Adv Video Signal Based Surveill.*, siap terbit.

Kapania S, Saini D, Goyal S, Thakur N, Jain R, Nagrath P. 2020. Multi object tracking with UAVs using deep SORT and YOLOv3 RetinaNet detection framework. Di dalam: *PervasiveHealth: Pervasive Computing Technologies for Healthcare*. ICST.

Kementrian Pertanian RI. *Statistik Peternakan dan Kesehatan Hewan 2021*. Ramadhany A, Ermansyah L, editor. Republik Indonesia: Direktorat Jenderal Peternakan dan Kesehatan Hewan.

Khairunissa J. 2020. Implementasi Algoritme Multi-Object Tracking Untuk Analisis Perilaku Ternak Ayam. Bogor: Institut Pertanian Bogor.

Neethirajan, S. 2022. ChickTrack – A Quantitative Tracking Tool for Measuring Chicken Activity. *Measurement*. 191:110819. doi:10.1016/j.measurement.2022.110819

Qiu Z, Zhao N, Zhou L, Wang M, Yang L, Fang H, He Y, Liu Y. 2020. Vision-based moving obstacle detection and tracking in paddy field using improved yolov3 and deep sort. *Sensors (Switzerland)*. 20(15):1–15. doi:10.3390/s20154082.

Suryana A. 2014. Menuju Ketahanan Pangan Indonesia Berkelanjutan 2025: Tantangan dan Penanganannya. *Forum Penelitian Agro Ekonomi*. Volume 32 No 2: hlm 123 - 135.

Wojke N, Bewley A, Paulus D. 2017 Mar 21. Simple Online and Realtime Tracking with a Deep Association Metric. <https://doi.org/10.48550/arXiv.1703.07402>

## **VII RIWAYAT HIDUP**

Penulis dilahirkan di kota Jakarta pada 1 Agustus 1999 sebagai anak pertama dari pasangan bapak Iman Fungsiawan dan ibu Heilany. Pendidikan sekolah menengah keatas (SMA) ditempuh di SMAN 101 Jakarta, dan lulus pada tahun 2017. Pada tahun yang sama, penulis diterima sebagai mahasiswa program sarjana (S-1) di program studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam di IPB.

Selama Mengikuti program S-1, penulis pernah melakukan Kuliah Kerja Nyata dalam rangka melaksanakan Cloud Computing Training di Kelurahan Sudimara Jaya.