

Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode *YOLO* untuk *Mobile Robot*

Khairunnas, Eko Mulyanto Yuniarno dan Ahmad Zaini
Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: ekomulyanto@ee.its.ac.id; zaini@ee.its.ac.id.

Abstrak—Robot mobil saat ini sangat diminati bagi yang baru mulai mempelajari tentang robot. Hal ini karena membuat *mobile robot* tidak memerlukan kerja fisik yang berat. Pengembangan yang dilakukan oleh beberapa anak bangsa untuk mengembangkan *mobile robot* ini untuk beberapa fungsi, diantaranya ada robot pengikut garis (*Line follower*) atau robot pengikut dinding (*wall flower*) ataupun pengikut cahaya. Dari pengembangan tersebut hal yang perlu dikembangkan lagi adalah fungsi *mobile robot* pengikut manusia. Dalam fungsi ini robot mobil harus dapat mengenali objek “Manusia” untuk dapat mengikuti objek (manusia) tersebut. Pada studi ini akan dikembangkan sebuah sistem *mobile robot* pengikut manusia untuk dapat mengikuti manusia sebagai asisten yang dapat membantu pekerjaan manusia. Dengan sensor kamera serta menggunakan metode *YOLO* (*You Only Look Once*), robot mobil akan dapat mengenali objek manusia dan di klasifikasikan fungsinya. Dari hasil klasifikasi ini, diharapkan menghasilkan respon oleh robot *mobile* dan akan mengaktifkan aktuator pada *mobile robot* tersebut.

Kata Kunci—Identifikasi, *Mobile Robot*, *You Only Look Once* (*YOLO*), *Convolutional Neural Network* (*CNN*).

I. PENDAHULUAN

KEMAJUAN teknologi yang pesat dalam beberapa dekade ini mendorong pengembangan teknologi yang lebih kompleks pula. Salah satu bidang yang mengikuti arus perkembangan ini adalah teknologi robotika. Salah satu teknik pada robot saat ini adalah teknik pengolahan citra. Teknik pengolahan citra adalah teknik pengolahan gambar baik berupa gambar bergerak atau diam untuk memperoleh informasi dari gambar-gambar tersebut. Beberapa penerapan pengolahan citra diantaranya, *License Plate Recognition* (*LPR*), *face recognition*, dll.

Bidang robotika yang masih dikembangkan saat ini adalah *mobile robot*. Robot ini dapat bergerak karena biasanya terdiri dari komponen roda pada bagian bawahnya. Dalam perkembangannya *mobile robot* dapat dibuat sebagai pengikut garis (*line follower*) yang dapat berjalan dijalar hitam dengan permukaan lantai putih. Akan tetapi *mobile robot* yang dapat mengikuti manusia masih jarang ditemukan di Indonesia. Di Indonesia sendiri, penggunaan metode *you only look once* (*YOLO*) sebagai penerapan teknik pengolahan citra pada *mobile robot* untuk mendeteksi objek belum banyak digunakan dan di implementasikan [1], [2].

YOLO adalah bagian dari metode *Convolutional neural networks* (*CNN*) yang banyak diaplikasikan pada data citra. *YOLO* melihat seluruh gambar selama pelatihan dan waktu tes sehingga secara implisit menyandikan informasi kontekstual tentang kelas serta penampilan mereka.

Berdasarkan kasus diatas maka di lakukan penerapan metode *You Only Look Once* (*YOLO*) pada *mobile robot* untuk mendeteksi objek manusia tertentu dan mengikuti perpindahan manusia tersebut.

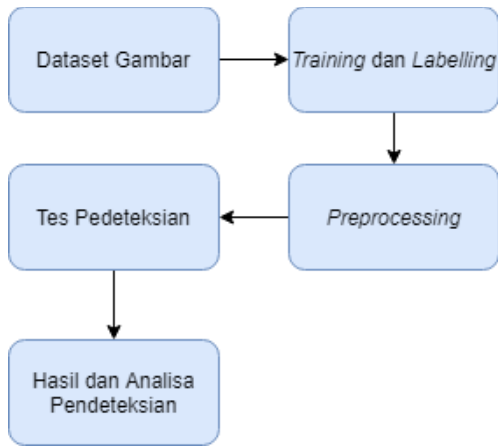
A. *Convolutional neural networks*

Salah satu Deep Learning untuk computer vision adalah *Convolutional neural network* (*CNN*). Sebuah *Convolutional neural network* terdiri dari sejumlah besar lapisan tersembunyi, yang masing-masingnya melakukan komputasi matematika pada input dan menghasilkan output yang dijadikan input ke lapisan selanjutnya [3]. *CNN* adalah metode yang sering digunakan untuk klasifikasi gambar. Contohnya pada *R-CNN* (*Region with CNN*) dimana ia bekerja dengan membuat kotak pembatas (*bounding boxes*) pada sebuah gambar dan kemudian menjalankan klasifikasi pada kotak tersebut. Metode lainnya yaitu *You Only Look Once* (*YOLO*). *YOLO* bekerja dengan cepat dan berbeda dari metode sebelumnya. *YOLO* tidak bekerja dengan cara seperti *sliding window* atau *R-CNN*. *YOLO* melihat seluruh gambar selama pelatihan dan waktu tes sehingga secara implisit menyandikan informasi kontekstual tentang kelas serta penampilan mereka. Secara umum layer pada *CNN* dapat dibedakan menjadi dua yaitu:

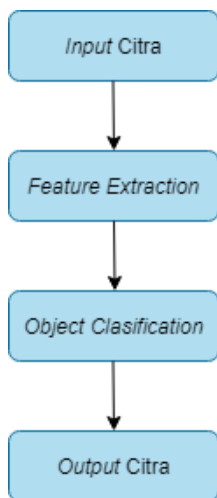
- 1) Layer ekstraksi fitur gambar, letaknya berada pada awal arsitektur tersusun atas beberapa layer dan setiap layer tersusun atas neuron yang terkoneksi pada daerah lokal (*local region*) layer sebelumnya. Layer jenis pertama adalah layer konvolusi dan layer kedua adalah layer *pooling*. Setiap layer diberlakukan fungsi aktivasi. Posisinya berselang-seling antara jenis pertama dengan jenis kedua. Layer ini menerima input gambar secara langsung dan memprosesnya
- 2) Layer klasifikasi, tersusun atas beberapa layer dan setiap layer tersusun atas neuron yang terkoneksi secara penuh (*fully connected*) dengan layer lainnya. Layer ini menerima input dari hasil keluaran layer ekstraksi fitur gambar berupa vektor kemudian ditransformasikan seperti *Multi Neural Networks* dengan tambahan beberapa *hidden layer*. Hasil keluaran berupa skoring kelas untuk klasifikasi.

B. *You Only Look Once* (*YOLO*)

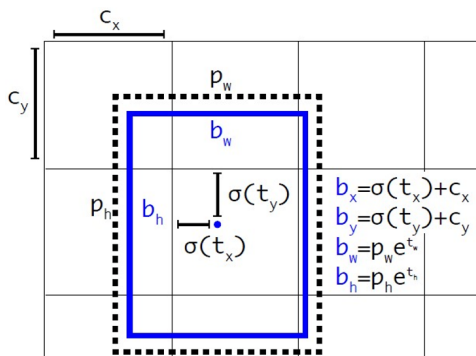
YOLO adalah sebuah metode untuk mendeteksi objek. *YOLO* memproses gambar secara *real-time* pada empat puluh lima (45) *frames per second*. Dibandingkan dengan sistem deteksi yang lainnya, *YOLO* membuat lebih banyak kesalahan pada praktiknya, namun dapat memprediksi



Gambar 1. Grafik Alur Pendeteksian Manusia.



Gambar 2. Deteksi Manusia Menggunakan YOLO.



Gambar 3. Prediksi Lokasi Bounding Box.

kesalahan pada background objek tersebut. *YOLO* dapat merepresentasi objek yang sangat umum [4], [5].

YOLO mendeteksi objek dengan menggunakan *unified model* dimana sebuah *single convolutional network* memprediksi beberapa *bounding boxes* (kotak pembatas) serta probabilitas kelas di dalam kotak-kotak tersebut secara bersamaan. Pertama-tama, sistem *YOLO* membagi citra input ke dalam grid $S \times S$. Jika pusat dari sebuah objek jatuh di dalam salah satu sel *grid*, maka sel *grid* itu bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel *grid* memprediksi *bounding boxes* dan *confidence score* dari tiap *bounding box* tersebut. *Confidence score* merefleksikan seberapa yakin dan akurat model bahwa terdapat sebuah objek di dalam kotak tersebut. Setiap *bounding box* terdiri dari 5 prediksi: x , y , w , h , dan *confidence*. Koordinat (x, y)



Gambar 4. Dataset dari Internet.



Gambar 5. Dataset dari Lokasi

mewakili pusat dari kotak relatif ke batas sel *grid*. (w, h) atau lebar dan tinggi mewakili pusat dari kotak relatif ke gambar. Dan terakhir adalah *confidence* yang mewakili *Intersection over Union (IoU)* antara kotak prediksi dan kotak *ground-truth*. Setiap sel *grid* juga memprediksi probabilitas kelas. Probabilitas dikondisikan pada sel *grid* yang memuat objek dan hanya satu kelas probabilitas yang dideteksi per sel *grid* tanpa memperhitungkan jumlah *bounding boxes*.

C. Mobile robot

Robot mobil atau *mobile robot* adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan-badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain [3].

II. DESAIN SISTEM

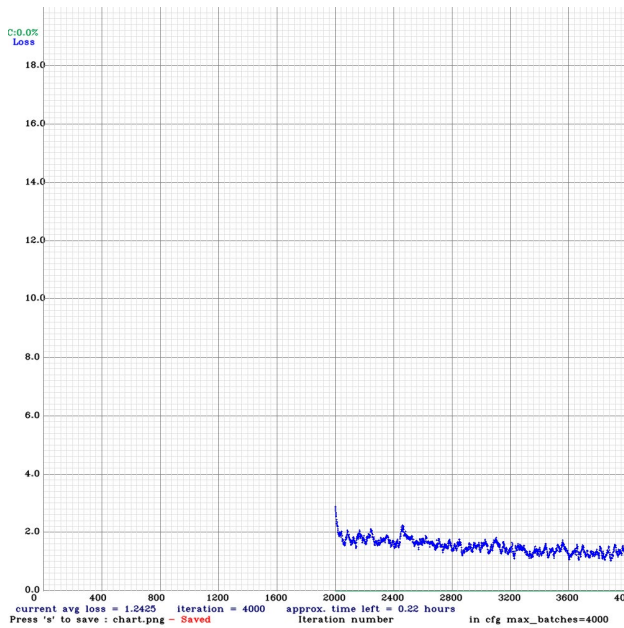
Pada Gambar 1 ditunjukkan alur pembuatan deteksi manusia menggunakan *Convolutional Neural Network*.

A. Dataset Gambar

Tahap dataset gambar dilakukan dengan pengambilan gambar dari *cocodataset.org*. Gambar tersebut selanjutnya dilakukan *labeling* untuk mendapatkan hasil *training* menggunakan *YOLO*.

B. Labeling Dataset

Labeling dataset atau dapat disebut anotasi adalah proses pemberian informasi pada berbagai pergerakan manusia.



Gambar 6. Grafik Hasil *Training* menggunakan *YOLOv4*.



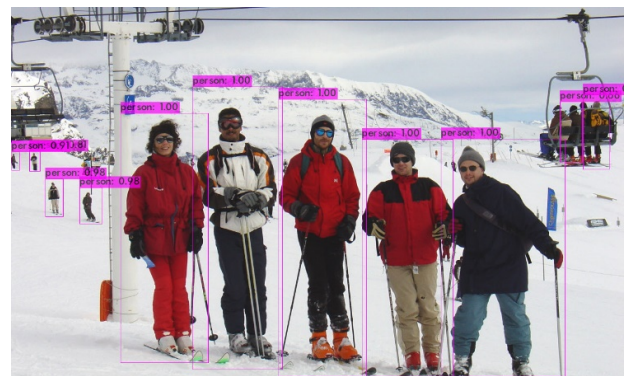
Gambar 7. Hasil Deteksi di Lokasi.

Dataset gambar yang di peroleh di berikan label satu persatu untuk memperoleh koordinat *ground-truth bounding box* yang akan dibandingkan dengan *predicted bounding box*. Dengan membandingkan kedua *ground box* itu akan diperoleh nilai *Intersection over Union (IoU)*.

Proses *labeling* menggunakan *source code* dari bahasa Python yang kemudian di compile menjadi sebuah aplikasi *labeling*. Hasil *labeling* ini akan di *export* ke dalam *YOLO* dengan format *.txt per file. Jika ada file yang tidak berisi gambar manusia maka file *.txt tidak di butuhkan.

C. *Labeling Dataset*

Middle point atau titik tengah menjadi acuan pergerakan mobile robot berdasarkan citra yang ditangkap oleh mata mobile robot. Terdapat dua titik tengah yaitu titik tengah relatif terhadap sisi sel dan titik tengah relatif terhadap bounding box. Untuk menghitung sudut, hal pertama yang harus ditentukan adalah ukuran *frame* untuk mendeteksi objek. Pada rumus 3 dan 4 ukuran layar ditentukan dengan variabel l_x dan l_y . Koordinat titik tengah relatif terhadap sisi sel di hitung dengan menjumlahkan t_x dan t_w sehingga menghasilkan nilai C_x serta menjumlahkan nilai t_y dan t_h untuk mendapatkan nilai C_y berdasarkan persamaan 1 dan 2. Setelah mendapatkan nilai C_x dan C_y , selanjutnya dapat dihitung koordinat titik tengah relatif terhadap *bounding box* berdasarkan persamaan 3.3 dan 3.4. Hasil dari nilai S_x , dan S_y



Gambar 8. Hasil Deteksi Data dari Internet.

Table 1. Spesifikasi *Hardware* yang Digunakan

Processor	Intel(R) Core (TM) i5-6200 CPU @ 2.30GHz
RAM	12.26 GB
Storage	128 GB
Graphic Card	NVIDIA GeForce 930M

Table 2. *YOLOv4 Performance Testing*

Load Model	YOLOv4
AP	87.03%
Person Images	904
Targets	563
Processing Time (s)	116
Precision	0.83
Recall	0.86
f1-score	0.84
mAP@0.5	87.03

selanjutnya dihitung nilai arctangent yang merupakan sudut antara titik tengah mata robot dan objek manusia menurut persamaan 5.

$$C_x = \frac{t_x + t_w}{2} \tag{1}$$

$$C_y = \frac{t_y + t_h}{2} \tag{2}$$

$$S_x = \left(\frac{l_x}{2} - C_x \right) + 1 \tag{3}$$

$$S_y = (l_y - C_y) + 1 \tag{4}$$

$$Angle = \frac{180 \times \arctan \frac{S_x}{S_y}}{\pi} \tag{5}$$

D. *Training dan Deteksi Menggunakan YOLO*

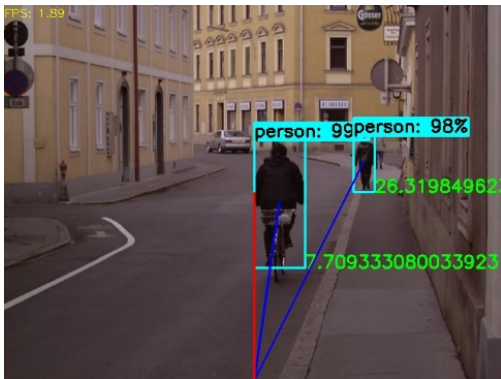
Dataset yang sudah dilabeli pada tahap sebelumnya akan dilatih sehingga membentuk sebuah pola yang hasilnya berbentuk Bobot. Bobot tersebut akan digunakan untuk mendeteksi objek di dalam citra. *Training* akan dilakukan menggunakan *You Only Look Once (YOLO)* dimana metode ini berbasis CNN. Perangkat yang digunakan dalam proses pengambilan data *training* adalah *Google Colaboratory*. Sedangkan *YOLO* yang digunakan adalah *YOLOv4*.

Berdasarkan diagram pada Gambar 2, *input* citra baik berupa gambar atau video akan dilewatkan ke dalam *YOLO*. Output yang diperoleh berupa confidence, nama kelas, serta posisi koordinat bounding box dari manusia yang terdeteksi oleh system.

- *Pengolahan Input*: Untuk mendeteksi objek, *YOLO* membutuhkan input gambar yang ukuran pikselnya berkelipatan 32. Semakin besar ukuran gambar *input*, maka semakin akurat pula hasil pendeteksian yang di



Gambar 9. Hasil Deteksi Sudut Pada Satu Objek Manusia.



Gambar 10. Hasil Deteksi Sudut Pada Beberapa Objek Manusia



Gambar 11. Hasil Deteksi Range Berdasarkan Sudut.



Gambar 12. Hasil Deteksi Berdasarkan ID.

peroleh, namun komputasi akan membutuhkan waktu yang lebih lama. Hal ini juga berlaku sebaliknya. Oleh karena itu, ukuran piksel dari gambar *input* disesuaikan dengan perangkat yang digunakan untuk *training* dan deteksi.

- *YOLOv4* menggunakan *feature extraction* yang bernama CSPDarknet53 sebagai bagian dari *backbone*-nya. *Feature extraction* ini memiliki 29 *convolutional layers* 3×3 , 725×725 *receptive field* dan 27.6 M parameter. Berdasarkan nilai ini, CSPDarknet53 merupakan model yang optimal sebagai *feature extraction* atau *backbone* dari detektor [6].
- Pengolahan *Output*: *YOLO* menghasilkan prediksi berupa koordinat *bounding box* (t_x, t_y, t_w, t_h), *confidence* dan *class probability*. Koordinat (t_x, t_y) merepresentasikan pusat kotak relatif terhadap sisi sel, sedangkan (t_w, t_h) memprediksi keseluruhan tinggi dan lebar gambar [7]. *Confidence* merupakan representasi dari *Intersection over Union (IoU)* antara *predicted box* dan *ground truth box*. Nilai *confidence* biasanya digambarkan sesuai persamaan 6. Untuk mendapatkan prediksi final, faktor penentunya adalah *class confidence score* yang didapat, berdasarkan probabilitas kondisional kelas dan *box confidence score*. *Class confidence score* mengukur nilai kepercayaan pada klasifikasi dan lokalisasi objek. *Class confidence score* memberi nilai kepercayaan kelas spesifik untuk setiap kotak, yang mengkodekan kemungkinan kelas yang muncul di kotak dan seberapa sesuai kotak yang diprediksi dengan objek [4]. Jika tidak ada objek yang terdeteksi maka nilai *confidence* adalah nol [7].

$$Confidence\ Score = P_r(Class_i) \times IoU_{pred}^{truth} \quad (6)$$

Intersection over Union (IoU) dapat dihitung dengan membandingkan *ground-truth bounding box* dan *predicted bounding box* yang dapat ditulis dengan persamaan:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (7)$$

Selain nilai *IoU*, di peroleh pula nilai rata-rata *IoU* atau dapat disebut dengan *mean average precision (mAP)*. Pada penelitian ini *mAP@IoU* membutuhkan nilai *threshol* yang melebihi 0.5 untuk diakui keberhasilannya. Jika nilai tersebut kurang dari 0.5 maka hasil dapat dianggap salah. Untuk ketentuan *threshol* 0.5 maka diketahui jika:

- Jika $IoU \geq 5$ maka klasifikasikan objek sebagai *True Positive (TP)*.
- Jika $IoU \leq 5$ maka klasifikasikan objek sebagai *False Positive (FP)*.
- Jika *ground truth* menampilkan gambar dan model gagal mendeteksi objek maka klasifikasikan sebagai *False Negative (FN)*.

Setiap bagian citra yang tidak dideteksi kalsifikasikan sebagai *True Negative (TN)*.

Nilai diatas dapat digunakan untuk menghitung *precision* dan *recall* menggunakan rumus:

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

Dari hasil penghitungan *IoU* didapatkan nilai *True Positive*, *False Positive*, dan *False Negative* yang digunakan untuk mencari nilai *precision* dan *recall* dari hasil pendeteksian manusia. Nilai *precision* dan *recall* kemudian digambarkan ke dalam sebuah kurva yang dinamakan *precision-recall curve*. Nilai *Average Precision (AP)* sendiri didapatkan dari hasil penghitungan area di bawah kurva dari tiap kelas yang terdeteksi oleh sistem. Sedangkan untuk nilai *mean Average Precision (mAP)* didapatkan dengan cara merata-ratakan nilai *AP* dari seluruh kelas yang terdeteksi.

III. HASIL DAN PENGUJIAN

Pada tahap ini akan dipaparkan hasil pengujian dari persiapan desain yang sebelumnya telah dilakukan.

A. Pembuatan Sistem Pendeteksian Manusia

Metode yang digunakan untuk mendeteksi manusia pada penelitian ini adalah metode *YOLO*. Pertama dataset yang diperoleh, di *labeling* dalam satu kelas yaitu objek manusia. Setelah *labeling* selesai, lalu masuk ke tahap *preprocessing* untuk menentukan sudut, dimana sudut ini menjadi acuan posisi dari objek manusia. Jika objek keluar dari sudut yang telah ditentukan sistem tidak akan mendeteksi objek. Selain sudut terdapat juga *ID tracking* dimana untuk mengklasifikasi objek yang terdapat didalam *frame*, sehingga *mobile robot* nantinya dapat mengambil keputusan untuk *tracking ID* tertentu.

- 1) Pembuatan Dataset: Pembuatan dataset bertujuan untuk menyiapkan data-data yang akan digunakan untuk *training*. Data berupa gambar manusia baik perseorangan ataupun berkelompok di keramaian yang di peroleh dengan cara mengambil data baik secara manual atau dari internet. Gambar 5 merupakan salah satu contoh pengambilan data langsung dilokasi dan Gambar 4 merupakan salah satu contoh pengambilan data dari internet. Data yang diperoleh berjumlah 902. Data tersebut dibagi menjadi 614 *image train* dan 288 *image test*.
- 2) *Training Data*: *Training* data bertujuan untuk menghasilkan sebuah bobot yang akan digunakan dalam proses deteksi manusia. Proses *training* data dilakukan menggunakan *YOLOv4* dengan *pytorch*. *Training* dilakukan menggunakan *Google Colaboratory*. Proses *training* berlangsung selama 9 jam dengan *training* menggunakan 4000 *max batch*, dengan *img size* 416 dan *batch size* 16 dengan jumlah 1 kelas. Gambar 6 adalah grafik dari hasil *training YOLOv4*. Berdasarkan grafik tersebut, *mAP* terbaik yang diperoleh *YOLOv4* sebesar 87.03%.
- 3) Hasil Deteksi Manusia: Deteksi Manusia menggunakan data yang diperoleh langsung di lokasi di tunjukkan pada Gambar 7 sedangkan untuk data yang di peroleh dari internet ditunjukkan pada Gambar 8. Pada gambar tersebut terlihat *bounding box* dari hasil deteksi serta nilai persepsi objek manusia pada pojok kiri atas.

B. Pengujian Performa YOLOv4

Pengujian performa dilakukan bertujuan untuk mengetahui tingkat keakurasian pada *YOLOv4*. Dalam melakukan uji performa, dibutuhkan *dataset test* yang telah diberi anotasi dimana *dataset test* merupakan dataset yang belum pernah digunakan sebagai dataset *training*. Valid dataset yang dibutuhkan sekitar 10% sampai 20% dari jumlah dataset *training*.

Valid dataset yang sudah diberi anotasi dan kelas tersebut diolah sebagai *groundtruth box* dibandingkan dengan *predicted box* yang kemudian menghasilkan *Confusion Matrix*, dan kemudian dikalkulasi untuk mendapatkan nilai *Precision*, *Recall*, *Average Precision (AP)*, *F1- score*, *Intersection over Union (IoU)* dan *Mean Average Precision* sesuai rumus yang telah diberikan pada desain dan implementasi sistem. Pada pengujian performa, masing-

masing model menggunakan *Google Colaboratory* dengan spesifikasi *hardware* seperti pada Tabel 1.

Pengujian Performa *YOLOv4*, dari hasil pengujian pada masing-masing model, didapatkan hasil seperti pada Tabel 2. Pada pengujian ini didapatkan *mean Average Precision (mAP)* pada *YOLOv4* dengan nilai 87,03%. Waktu pemrosesan dataset test, *YOLOv4* dapat memproses deteksi 904 dataset dalam waktu 116 detik.

C. Hasil Deteksi Sudut

Deteksi sudut didapat dengan menambahkan *center point* atau titik tengah. *Center point* ditandai dengan garis merah di tengah *frame*. Deteksi sudut pada satu objek manusia ditunjukkan pada Gambar 9 serta deteksi sudut pada beberapa objek manusia ditunjukkan pada Gambar 10.

Pada Gambar 9 dan Gambar 10 terlihat nilai sudut dibagian bawah *bounding box* merupakan nilai dalam satuan derajat yang sebelumnya di hitung berdasarkan persamaan:

$$\text{Angle} = \frac{180 \times \arctan \frac{S_x}{S_y}}{\pi} \quad (10)$$

Sudut ini akan berguna untuk memberi batas deteksi manusia berdasarkan *range* yang telah di berikan pada desain sistem. Area ini akan memberikan sebuah *clue* pada *mobile robot* tentang keberadaan posisi objek manusia. Dengan sistem yang seperti ini kecerdasan *mobile robot* dalam hal mengolah data dari objek yang di deteksi akan lebih meningkat.

D. Deteksi Objek Di Dalam Range

Penambahan range deteksi dapat memperluas wawasan olah data dari *mobile robot*. Deteksi didalam *range* ini didapat dengan mengubah beberapa baris kode pada pemrograman dengan bahasa *Python* pada *YOLOv4*. Perubahan kode ini dengan cara membuat kondisi ketika objek manusia berada pada *range* 10 derajat dari titik tengah maka deteksi objek manusia akan bekerja.

E. Deteksi Objek Berdasarkan ID

Didalam modul yang berbeda, objek manusia dapat dideteksi berdasarkan ID. Deteksi berdasarkan ID ini adalah hasil dari metode *deep sort*. ID merupakan nilai unik yang terdapat di bagian kanan atas *bounding box*. ID tidak akan berubah selama objek manusia masih terdapat didalam *frame* deteksi oleh *mobile robot*.

Berdasarkan Gambar 12, dapat dilihat pada dua gambar bawah terdapat ID unik di kanan atas *bounding box*. ID ini yang kemudian digunakan untuk *object tracking* sehingga bagaimanapun pergerakan objek manusia didalam *frame* yang dideteksi oleh *mobile robot*, objek tersebut akan terdeteksi sebagai objek yang sama. Hal ini sangat dibutuhkan untuk meningkatkan intelektual dari *mobile robot* dengan kemampuan yang kompleks.

Pada Gambar 12 bagian bawah kanan merupakan objek pada posisi pertama sebelum bergerak. Objek manusia ini kemudian bergerak dan melakukan *crossing* yang di tunjukkan pada Gambar 12 bagian atas. Pada gambar itu terlihat *bounding box* objek manusia yang berada di belakang hilang, namun ID untuk objek manusia yang di depan masih menunjukkan angka dua. Hal ini menunjukkan bahwa ID ini akan tetap walaupun pergerakan objek manusia berubah-ubah. Ketika objek manusia selesai melakukan *crossing*

seperti terlihat pada Gambar 12 bagian kiri bawah, ID untuk objek yang *bounding box*-nya hilang sebelumnya masih tetap menunjukkan angka satu.

5) Deteksi Objek berdasarkan ID berhasil mengklasifikasikan objek berdasarkan ID selama objek masih berada dalam *frame*.

IV. KESIMPULAN

Dari pelaksanaan dan pengujian sistem yang sudah dilakukan, penulis berhasil mengimplementasikan sistem *object tracking* dan deteksi objek manusia. Kemudian untuk lebih detail dapat ditarik beberapa kesimpulan sebagai berikut:

- 1) Pada penelitian ini modul deteksi objek manusia berhasil mendeteksi objek manusia.
- 2) Berdasarkan hasil uji performa *YOLOv4* diperoleh nilai *mAP* sebesar 87,03% dan waktu pemrosesan selama 116 detik dengan jumlah total gambar sebanyak 904 gambar.
- 3) Pada pengujian menghitung sudut deteksi objek manusia, modul dapat mendeteksi baik *single object* ataupun *multiple object*.
- 4) Pengujian deteksi didalam *range* sudut tertentu berhasil mengklasifikasikan objek yang berada di dalam *range* dan yang di luar *range*.

DAFTAR PUSTAKA

- [1] R. Kusumanto and A. N. Tompunu, "Pengolahan citra digital untuk mendeteksi obyek menggunakan pengolahan warna model normalisasi RGB," 2011.
- [2] K. Buhler, J. Lambert, and M. Vilim, "YoloFlow Real-time Object Tracking in Video CS 229 Course Project." 2016.
- [3] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," 2018, doi: 10.1109/ICENGTECHNOL.2017.8308186.
- [4] J. S. W. Hutaaruk, T. Matulatan, and N. Hayaty, "Deteksi kendaraan secara real time menggunakan metode YOLO berbasis Android," *J. Sustain. J. Has. Penelit. dan Ind. Terap.*, vol. 9, no. 1, pp. 8–14, 2020, doi: 10.31629/SUSTAINABLE.V9I1.1401.
- [5] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement." 2018, Accessed: Aug. 13, 2021. [Online]. Available: <https://arxiv.org/abs/1804.02767>.
- [6] E. W. Felten, M. Raj, and R. Seamans, "A method to link advances in Artificial Intelligence to occupational abilities," *AEA Pap. Proc.*, vol. 108, pp. 54–57, 2018, doi: 10.1257/PANDP.20181021.
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.