

# Pengenalan Objek Menggunakan YOLO pada Alat Bantu Tunanetra Berbasis Raspberry Pi

*Object Recognition Using YOLO on Visually Impaired Assistive Devices Based on Raspberry Pi*

Retnaning Tyas Utami<sup>1</sup>, Tedy Rismawan<sup>2</sup>, Rahmi Hidayati<sup>3</sup>

<sup>1,2,3</sup>Program Studi Rekayasa Sistem Komputer/Sistem Komputer, Universitas Tanjungpura

E-mail: <sup>1</sup>retnatyasutami@student.untan.ac.id, <sup>2</sup>tedyrismawan@siskom.untan.ac.id,

<sup>3</sup>rahmihidayati@siskom.untan.ac.id

## Abstrak

Keterbatasan sensorik yang melekat pada penyandang tunanetra menghalangi rutinitas mereka yang menyebabkan kesulitan dalam menjalankan aktivitas sehari-hari. Kurangnya teknologi pada alat bantu seperti tongkat yang menjadi kebutuhan para tunanetra menyebabkan kesulitan tunanetra menjalani aktivitas secara normal. Alat bantu yang mampu mengenali objek menjadi salah satu solusi untuk membantu para tunanetra. Dalam penelitian ini, dibangun sebuah sistem pengenalan objek menggunakan YOLO pada alat bantu tongkat tunanetra berbasis Raspberry Pi 4b. Sistem ini dibuat untuk membantu tunanetra mendapatkan informasi objek yang ada didepannya, sehingga tunanetra dapat menghindari objek tersebut karena menghalangi atau berpotensi mencelakai saat beraktivitas. Algoritma YOLO yang digunakan dalam penelitian ini adalah YOLOv4 dan YOLOv4-Tiny. Dataset yang digunakan memiliki 5 kelas objek yaitu kursi, meja, orang, pot tanaman, dan kulkas. Jumlah dataset yang digunakan untuk melatih model sebanyak 841 dataset yang dibagi menjadi 757 data latih dan 84 data uji. Pada pelatihan model yang dilakukan didapatkan hasil akurasi menggunakan mAP sebesar 99,74% untuk model YOLOv4 dan 67,78% untuk model YOLOv4-Tiny. Pada implementasi pengujian sistem pengenalan objek dengan jarak 1 hingga 3,5meter, didapatkan nilai akurasi sebesar 93,33% untuk model YOLOv4 dan 73,3% untuk model YOLOv4-Tiny.

Kata kunci: Alat Bantu, Pengenalan Objek, Tunanetra, YOLO.

## Abstract

*Sensory limitations that are inherent to the visually impaired hinder their routines causing difficulties in carrying out daily activities. The lack of technology in assistive devices such as canes that blind people need becomes a difficulty to carry out activities normally. Aids that recognize objects are one solution to help. This study built an object recognition system using YOLO was built on a Raspberry Pi 4b-based blind cane. This system is made to help blind people get information on objects in front of them, so they can avoid these objects because they hinder or potentially harm them during their activities. The YOLO algorithms used in this research are YOLOv4 and YOLOv4-Tiny. The dataset used has 5 object classes, namely chairs, tables, people, plant pots, and refrigerators. The number of datasets used to train the model is 841 datasets, which are divided into 757 training data and 84 test data. In the model training conducted, the accuracy results using mAP are 99.74% for the YOLOv4 model and 67.78% for the YOLOv4-Tiny model. In implementing object recognition system testing with a distance of 1 to 3.5 meters, the accuracy value is 93.33% for the YOLOv4 model and 73.3% for the YOLOv4-Tiny model.*

*Keywords: Visually Impaired, Objects, Assistive Devices, YOLO.*

## 1. PENDAHULUAN

Tunanetra merupakan sebuah kondisi ketika seseorang mengidap kebutaan total atau memiliki sisa penglihatan[1]. Menurut perkiraan data Kementerian Kesehatan RI, jumlah penyandang tunanetra di Indonesia adalah 1,5% dari total penduduk Indonesia atau sekitar 4 juta jiwa dari 270 juta jiwa yang terdiri dari kebutaan total dan gangguan penglihatan yang lebih ringan[2]. Dalam kehidupan sehari-hari penyandang tunanetra mengalami kesulitan dalam beraktivitas karena keterbatasan sensorik yang dimiliki sehingga membutuhkan alat bantu seperti tongkat untuk melakukan pendeteksian dan orientasi mobilitas saat berjalan[3]. Kebutuhan para penyandang tunanetra kepada tongkat menjadikan tongkat sebagai alat utama untuk melakukan aktivitas secara normal. Kurangnya teknologi pada alat bantu tongkat yang dapat mengenali objek atau hal-hal di depannya menjadi salah satu kesulitan penyandang tunanetra bergerak di tempat di mana suatu benda berada. Sehingga diperlukan sebuah teknologi yang diletakkan pada tongkat agar dapat mempermudah aktivitas para tunanetra.

Salah satu metode yang digunakan untuk melakukan pengenalan objek adalah algoritma *You Only Look Once* (YOLO). YOLO merupakan metode yang menggunakan jaringan konvolusional untuk mendeteksi jenis dan posisi objek pada suatu citra secara *realtime*. Algoritma ini dapat mendeteksi *bounding box* (kotak pembatas) dan kelas probabilitas dari setiap *frame*, sehingga memungkinkan deteksi objek yang akurat[4]. YOLO akan mengelompokkan gambar ke dalam sebuah *grid* berukuran SxS yang bergantung pada ukuran masukan yang digunakan. Masukan yang digunakan dapat berupa gambar maupun video secara *realtime* yang diambil melalui kamera. Masukan tersebut akan dideteksi dengan membagi masukan ke dalam suatu *grid* berukuran SxS. Lalu dideteksi dengan *bounding box* serta *confidence score* pada *bounding box* tersebut. Setelah itu YOLO akan mengklasifikasikan kelas objek-objek yang terdapat pada *bounding box* dengan probabilitasnya membentuk peta kelas probabilitas (*Class Probability Map*) untuk mendapatkan *bounding box* yang memiliki kelas objek dengan probabilitas tinggi dari semua deteksi dengan syarat hanya yang melewati *threshold* yang akan digunakan[5].

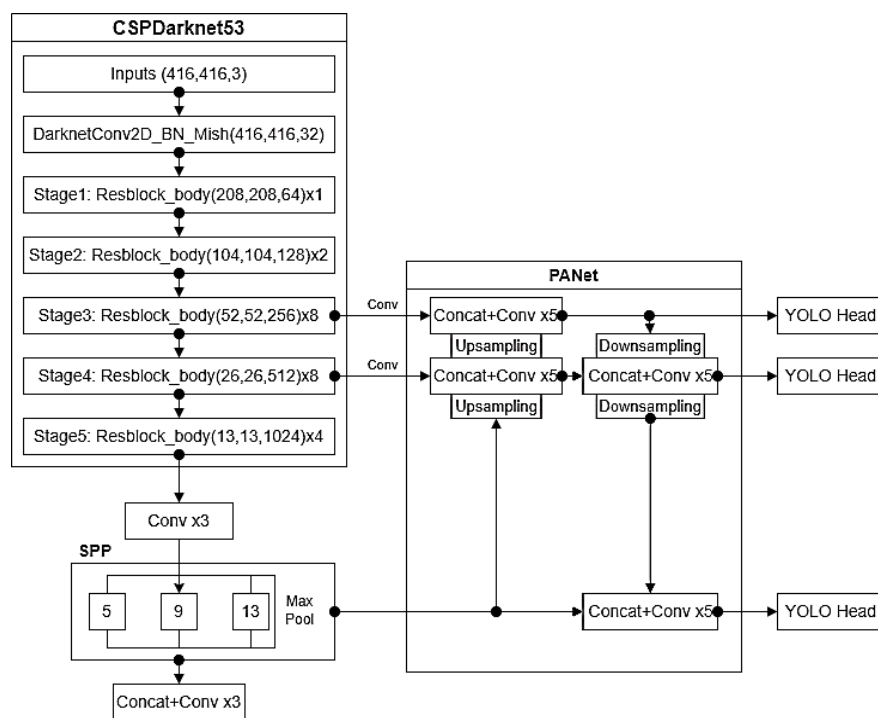
Pada penelitian sebelumnya telah dilakukan menggunakan metode *Single Shot MultiBox Detector* (SSD) yang dilengkapi dengan *buzzer* dan *vibrator*[3] dan algoritma MobileNet untuk menghasilkan sistem yang mampu mendeteksi objek buku, gelas, botol, telepon seluler, dan orang dengan rata-rata keberhasilan pengujian sebesar 40%[6]. Penelitian lainnya terkait algoritma YOLO telah dilakukan dengan menerapkan algoritma tersebut sebagai penentu pindah tanaman cabai melalui notifikasi bot telegram [4], sebagai pendeteksi adanya konten pornografi pada video[5], sebagai algoritma untuk mengenali objek pada citra makanan cepat saji dengan mAP sebesar 100% dan mampu mengenali objek dengan akurasi 63% hingga 100%[7]. Kemudian penelitian untuk pemantauan arus lalu lintas melalui CCTV dengan akurasi deteksi paling besar pada siang hari sebesar 79,8% menggunakan algoritma YOLOv5[8], sebagai algoritma pengklasifikasi dan pendeteksi kendaraan untuk mencari model yang optimal[9], sebagai pendeteksi pelanggaran helm dan masker serta nomor plat kendaraan secara otomatis menggunakan Tesseract-OCR[10], sebagai pengklasifikasi 3 kategori sampah menggunakan YOLOv4 yang di improvisasi[11], sebagai pendeteksi objek dengan menghasilkan keluaran suara menggunakan gTTS[12], sebagai pendeteksi orang dan klasifikasi jarak individu menggunakan kamera termal[13], dan sebagai sistem deteksi cacat pada permukaan kayu dan kayu karet menggunakan dengan menerapkan algoritma YOLOv5[14].

Berdasarkan penelitian terkait, dilakukan penelitian ini dengan mengembangkan sebuah alat bantu berupa tongkat tunanetra yang dilengkapi dengan algoritma YOLOv4 dan YOLOv4-Tiny untuk melakukan pendeteksian objek dan menghasilkan keluaran berupa suara informasi objek. Penelitian ini dikembangkan menggunakan algoritma YOLOv4 dan YOLOv4-Tiny yang sudah dilatih hingga menjadi sebuah model yang diletakkan ke dalam Raspberry Pi 4b. Raspberry Pi 4b kemudian diletakkan pada tongkat tunanetra untuk melakukan deteksi objek dengan menghubungkan Raspberry Pi 4b ke *webcam* dan *earphone*. Hasil deteksi objek kemudian dikeluarkan melalui *earphone* berupa informasi objek yang terdeteksi. Penelitian ini diharapkan mampu membantu para tunanetra untuk menjalankan aktivitas secara normal dengan menghindari objek-objek yang berpotensi mencelakai.

## 2. METODE PENELITIAN

YOLO mampu melakukan pendeteksian objek secara *realtime* berbasis *Convolutional Neural Network* (CNN). Pendekatan jaringan syaraf tunggal digunakan YOLO untuk melakukan pendeteksian pada sebuah citra. YOLO menggunakan semua fitur gambar untuk memprediksi setiap kotak pembatas (*bounding box*) yang mampu melakukan prediksi pada kotak-kotak pembatas dan probabilitas dalam satu evaluasi secara langsung. YOLO mampu melakukan proses pendeteksian gambar secara *realtime* dalam waktu *45 frames per second*[9].

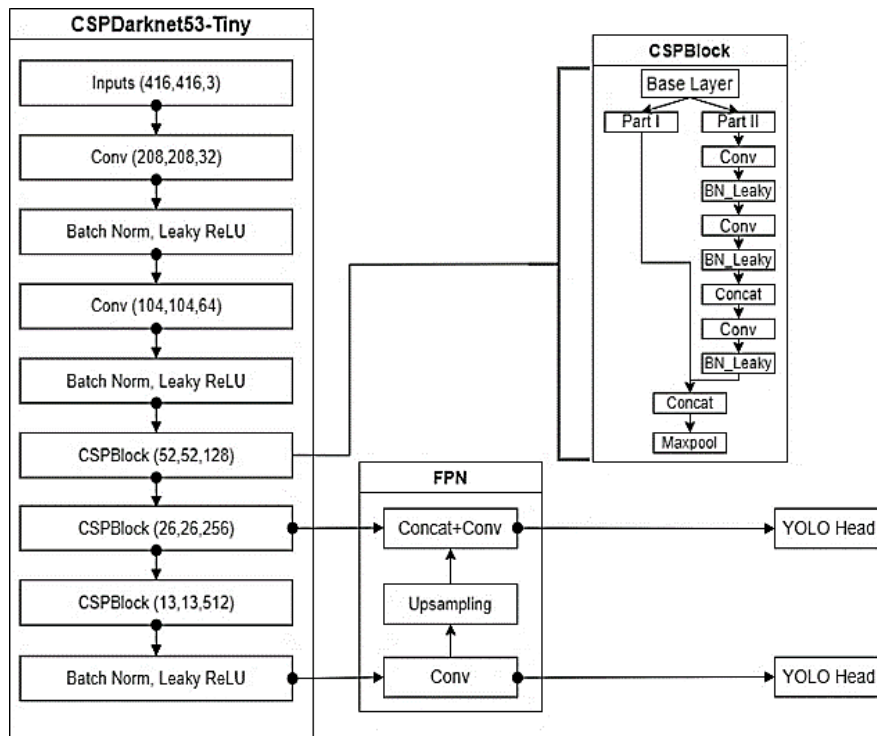
Algoritma YOLOv4 akan melakukan pengenalan objek yang berfokus pada pengoptimasian komputasi paralel dan kecepatan operasi yang difokuskan untuk penggunaan GPU agar mencapai hasil pelatihan yang berkualitas tinggi dan *realtime*[15]. YOLOv4 menggunakan teknik pengembangan yang disebut *Bag-of-Freebies* (BoF) dan *Bag-of-Special* (BoS) pada fase pelatihan yang bertujuan untuk meningkatkan efisiensi dan akurasi model tanpa mempengaruhi waktu pemrosesan selama pelatihan[16]. YOLOv4 memiliki tiga komponen penting yaitu *backbone* (blok tulang punggung), *neck* (blok leher), dan *head* (blok kepala). *Backbone* berfungsi untuk mengekstrak fitur gambar, dimana pada YOLOv4 akan menggunakan CSPDarknet53 sebagai pengekstrak gambar[15]. Sedangkan *neck* berfungsi untuk mendapatkan fitur yang lebih efektif dengan menambahkan lapisan ekstra di antara *backbone* dan *head*. Kemudian *Head* atau blok kepala berfungsi untuk membuat *bounding box* (kotak pembatas) dan mengklasifikasikan objek yang ada pada tiap kotak serta menghasilkan skor kepercayaan[15]. Pada arsitektur YOLOv4 yang dapat dilihat pada Gambar 1, keluaran yang akan dihasilkan berupa *bounding box regression loss* menggunakan Ciou-Loss dan *classification loss* menggunakan Diou\_nms. Ciou-Loss akan memperhitungkan area yang mengalami tumpang tindih antara kotak prediksi dan kotak target, jarak antara titik pusat, dan aspek rasio yang dapat meningkatkan ketepatan regresi kotak prediksi[16].



Gambar 1 Arsitektur YOLOv4 [11]

YOLOv4-Tiny merupakan pengembangan dari YOLOv4 yang memiliki kecepatan pendeteksian yang lebih baik. Pada arsitektur YOLOv4-Tiny menggunakan CSPDarknet-53-tiny

sebagai *backbone*, dimana modul CSPBlock menggunakan jaringan *cross stage partial*. Modul CSPBlock dapat meningkatkan kemampuan belajar jaringan konvolusi dibandingkan dengan Modul ResBlock. Meskipun waktu komputasi yang dimiliki meningkat hingga 10% -20%, hal ini mampu meningkatkan akurasi pada YOLOv4-Tiny[17]. Algoritma ini juga menggunakan LeakyReLU sebagai fungsi aktivasi, berbeda dengan YOLOv4 yang menggunakan fungsi aktivasi Mish. Algoritma YOLOv4-Tiny menggunakan jaringan piramida fitur untuk mengekstrak peta fitur dengan skala yang berbeda untuk meningkatkan kecepatan deteksi objek, tanpa menggunakan pengumpulan piramida spasial dan agregasi jalur yang digunakan dalam metode YOLOv4. Pada saat yang sama, YOLOv4-Tiny menggunakan dua peta fitur dengan skala yang berbeda yaitu  $13 \times 13$  dan  $26 \times 26$  untuk memprediksi hasil deteksi. Jika dibandingkan dengan YOLOv4, akurasi yang dimiliki oleh YOLOv4-Tiny relatif lebih rendah[12]. Struktur FPN yang ada pada YOLOv4-Tiny digunakan untuk menggabungkan fitur-fitur dari berbagai lapisan yang dapat menyimpan data semantik dari *deep network* dan informasi geometrik dari *low-level network*. Pada arsitektur YOLOv4-Tiny yang dapat dilihat pada Gambar 2, keluaran yang dihasilkan pada bagian Yolo Head berupa *bounding box*, label kelas, dan nilai kepercayaan deteksi objek[13].

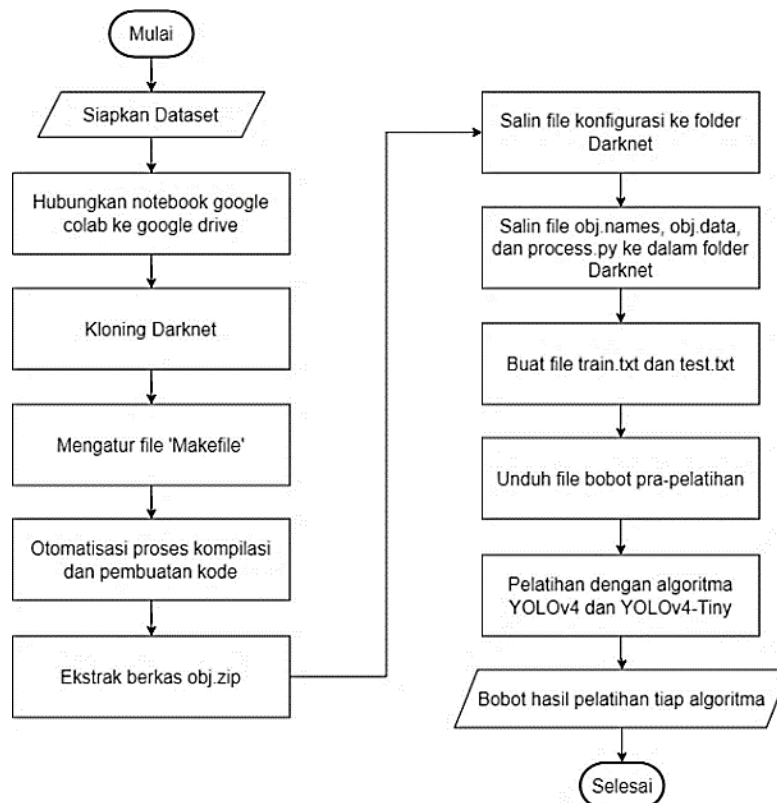


Gambar 2 Arsitektur YOLOv4-Tiny [13]

Proses pelatihan YOLOv4 dan YOLOv4-Tiny melibatkan pemanfaatan dataset yang telah dianotasi dan diberi label yang dilakukan menggunakan *platform* Roboflow. Dataset yang dianotasi juga menghasilkan titik koordinat objek yang disimpan dalam berkas berindeks .txt dan dikompres menjadi berkas berindeks .zip. Pada saat pelatihan, total dataset yang digunakan adalah 1682 dataset yang terdiri dari 841 dataset gambar dan 841 dataset berisi titik koordinat objek yang ada pada gambar. Data berindeks .txt berisi koordinat relatif dari sudut kiri atas dan sudut kanan bawah dalam format yang telah ditentukan. Data tersebut digunakan agar model dapat belajar cara menentukan *bounding box* yang optimal untuk membatasi objek yang ada dalam sebuah gambar sehingga model mampu mengenali dan menandai objek dengan akurasi yang lebih baik.

*Google Colaboratory* digunakan untuk melakukan pelatihan menggunakan algoritma pra-pelatihan YOLOv4 dan YOLOv4-Tiny. *Google Colaboratory* dihubungkan dengan *Google*

Drive agar dapat melakukan replikasi berkas Darknet dan menyimpan bobot hasil pelatihan. *Google Drive* juga digunakan untuk menyimpan bobot sementara dari tiap lipatan 100 iterasi yang digunakan apabila terjadi interupsi pada proses pelatihan. Pelatihan dilakukan dengan memanfaatkan pengembangan yang dilakukan oleh Alexey Botchkovsky dengan menyalin atau mereplikasi berkas Darknet yang telah didokumentasikan. Pada proses pelatihan, dataset akan dibagi menjadi 757 gambar untuk *training* dan 84 gambar untuk *test*. Kemudian penyesuaian parameter-parameter yang ada pada file konfigurasi yang diperoleh dari berkas Darknet seperti penyesuaian *batch*, *subdivisi*, lebar, tinggi, *max\_batches*, dan *steps*, serta mengaktifkan beberapa pustaka seperti OpenCV, CUDNN, CUDNN\_HALF, GPU, LIBSO. Penyesuaian parameter tersebut dilakukan berdasarkan jumlah gambar yang akan dilatih dalam satu iterasi, jumlah kelas yang digunakan, dan ukuran inputan yang ingin digunakan. Saat pelatihan dilakukan, terdapat resiko terjadinya *error* atau interupsi yang mengakibatkan berhentinya proses pelatihan. Hal ini dapat diatasi dengan mengulang proses pelatihan dengan menjalankan tahap menghubungkan *Google Colabotaory* ke *Google Drive* dan diikuti proses selanjutnya tanpa melakukan proses replikasi berkas Darknet dan memanggil berkas bobot sementara yang berisi lipatan 100 iterasi terakhir yang dilakukan. Proses pelatihan akan menghasilkan bobot yang akan digunakan untuk melakukan deteksi objek pada Raspberry Pi 4b. Diagram alur proses pelatihan YOLOv4 dan YOLOv4-Tiny dapat dilihat pada Gambar 3.



Gambar 3 Diagram Alur Proses Pelatihan YOLOv4 dan YOLOv4-Tiny

Penilaian kinerja model YOLOv4 dan YOLOv4-Tiny dilakukan menggunakan *Mean Average Precision* (mAP). mAP merupakan rata-rata *Average Precision* (AP) secara keseluruhan untuk semua kelas yang digunakan sebagai parameter evaluasi. Nilai AP diperoleh dari dua perhitungan, yaitu perhitungan *precision* dan *recall*. *Precision* berfungsi untuk menunjukkan persentase kesalahan yang terdeteksi dengan benar terhadap semua jumlah kesalahan yang terdeteksi. Sedangkan *Recall* berfungsi untuk mengukur seberapa banyak objek pada kelas tersebut yang berhasil dideteksi oleh model[17]. *Precision* dan *Recall* dihitung menggunakan nilai TP, FP, dan FN dari model yang dilatih.

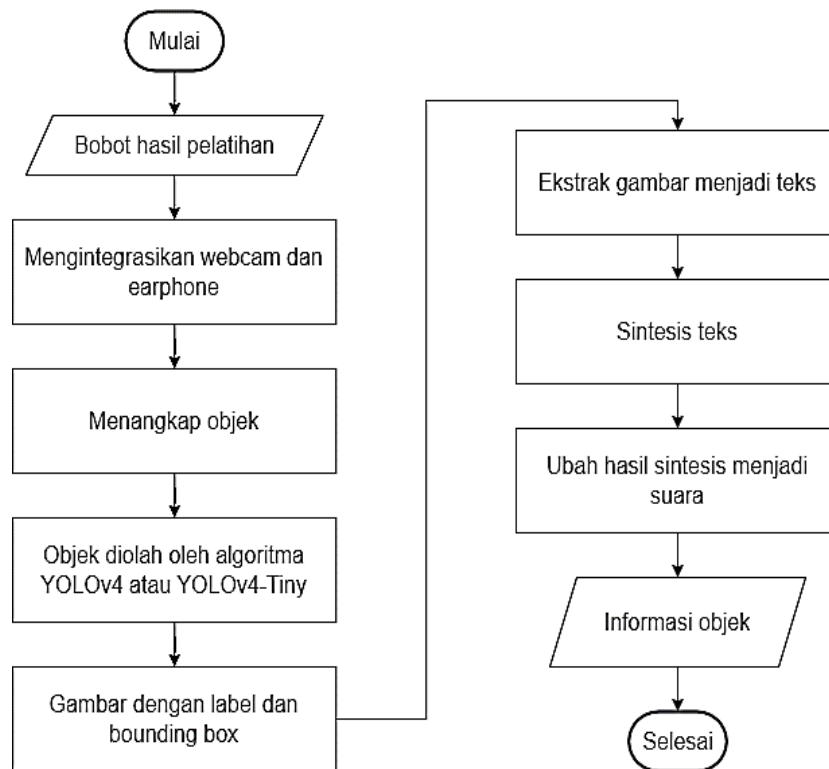
$$mAP = \sum_{i=1}^N \frac{AP(i)}{N} \times 100\% \quad (1)$$

Persamaan di atas merupakan persamaan untuk mencari mAP, dimana:

AP = *Average Precision*

N = Banyaknya kelas yang dimiliki model yang telah dilatih

Proses deteksi pada Raspberry Pi 4b diimplementasikan melalui skrip Python yang berfungsi untuk memanggil file bobot hasil pelatihan, file konfigurasi, dan berkas lainnya yang disimpan ke dalam Raspberry Pi 4b. *Webcam* dihubungkan melalui *port* USB untuk melakukan pengenalan objek, sementara *earphone* dihubungkan ke *port audio output jack* untuk mengeluarkan suara hasil pengenalan objek. *Webcam* digunakan untuk menangkap citra atau gambar dengan satu atau lebih objek kursi, meja, orang, pot tanaman, dan kulkas di dalamnya. Hasil tangkapan objek dari *webcam* akan diolah oleh YOLOv4 dan YOLOv4-Tiny yang sudah dilatih. Data yang berasal dari tangkapan objek, berupa *frame* berukuran 416x416 piksel, akan diproses melalui 161 lapisan. Hasilnya adalah *frame* yang telah dilengkapi dengan *bounding box*, label kelas, dan skor kepercayaan. Dalam penelitian ini, *output* yang dihasilkan berupa suara yang memberikan informasi mengenai objek yang terdeteksi. Proses transformasi gambar dilakukan dengan mengambil informasi dari label kelas hasil deteksi objek menjadi suara menggunakan perangkat lunak sintesis suara eSpeak dan MBROLA. Kemudian didapatkan *output* berupa suara yang akan memberikan informasi mengenai objek yang terdeteksi oleh *webcam*. Diagram alur proses deteksi objek pada Raspberry Pi 4b dapat dilihat pada Gambar 4.



Gambar 4 Diagram Alur Proses Pengenalan Objek pada Raspberry Pi 4b

$$Akurasi = \frac{(TP + TN)}{(TP + FN + FP)} \times 100\% \quad (2)$$

Pada pengenalan objek yang dilakukan pada Raspberry Pi 4b, akurasi digunakan untuk mengukur sejauh mana *record* data diklasifikasikan dengan tepat setelah pengujian model menggunakan persamaan diatas, dimana:

TP = *True Positive*

FP = *False Positive*

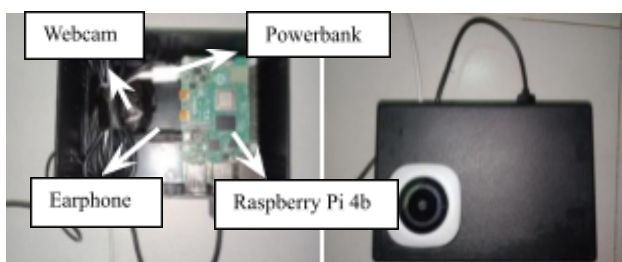
FN = *False Negative*

Penelitian ini bertujuan untuk mengembangkan sebuah sistem yang memiliki kemampuan untuk mengenali objek menggunakan model YOLOv4 dan YOLOv4-Tiny yang telah dilatih sebelumnya. Model tersebut kemudian diletakkan ke dalam Raspberry Pi 4b yang terhubung dengan *webcam* dan *earphone* sebagai sarana untuk mendapatkan masukan dan keluaran dari hasil pengenalan objek. Rancangan penggunaan sistem yang telah dibuat dapat dilihat pada Gambar 5.



Gambar 5 Rancangan Penggunaan Sistem

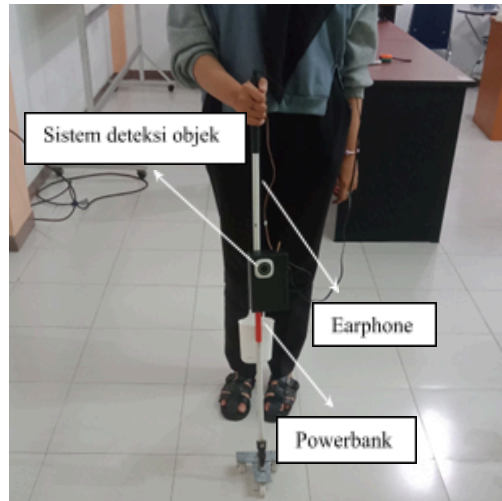
Implementasi rancangan penggunaan sistem merupakan implementasi *webcam*, *earphone*, dan *powerbank* pada Raspberry Pi 4b yang akan digunakan untuk melakukan proses penangkapan objek atau citra yang diletakkan pada tongkat tunanetra. Raspberry Pi 4b menggunakan *powerbank* sebagai sumber daya dengan baterai sebesar 10000 mAH yang dapat digunakan selama kurang lebih 7,1 jam. *Webcam*, *powerbank*, dan *earphone* masing-masing akan dihubungkan ke *port USB*, *port power*, dan *port audio jack* pada Raspberry Pi 4b. Implementasi visualisasi perangkat keras yang dilakukan dapat dilihat pada Gambar 6.



Gambar 6 Implementasi Visualisasi Perangkat Keras

Setelah seluruh perangkat keras saling terhubung dan diletakkan ke dalam kotak, selanjutnya perangkat keras akan diimplementasikan pada tongkat tunanetra seperti pada Gambar 7.





Gambar 7 Implementasi Perangkat Keras pada Tongkat Tunanetra

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Hasil Pelatihan Model YOLOv4 dan YOLOv4-Tiny

Setelah pelatihan model dilakukan, baik pada model YOLOv4 maupun YOLOv4-Tiny didapatkan hasil kinerja model yang dihitung menggunakan mAP seperti yang tersaji pada Tabel 1.

Tabel 1 Nilai AP Pelatihan Model

Kelas Objek	YOLOv4 (%)	YOLOv4-Tiny (%)
Kursi	100	78,20
Meja	99,97	66,90
Orang	98,75	63,28
Pot Tanama	100	67,58
Kulkas	100	62,95
<b>Hasil mAP</b>	<b>99,74</b>	<b>67,78</b>

Pada Tabel 1 dapat dilihat bahwa nilai mAP yang dihasilkan model YOLOv4 setelah pelatihan adalah 99,74%. Hasil mAP yang diperoleh menjadikan model algoritma YOLOv4 mampu mendeteksi objek yang ditentukan dengan baik dengan sedikit kesalahan deteksi. Hal ini disebabkan oleh tujuan YOLOv4 yang memfokuskan model agar mendapatkan akurasi yang tinggi sehingga model menambahkan parameter yang mampu meningkatkan akurasi. Parameter seperti *learning rate*, *batch size*, *epochs*, dan parameter lainnya sangat jelas dapat meningkatkan akurasi model. Banyaknya lapisan yang digunakan untuk mengekstraksi gambar objek juga mampu meningkatkan akurasi. Selain itu, kualitas data yang baik juga menjadi pendukung untuk mendapatkan akurasi yang baik.

Pada Tabel 1 juga didapatkan hasil mAP untuk model YOLOv4-Tiny setelah pelatihan sebesar 67,78%. Nilai tersebut menyebabkan masih banyaknya kesalahan yang dilakukan model untuk mendeteksi objek. Hal ini disebabkan karena kompromi yang dilakukan oleh model untuk mempercepat waktu inferensi. Faktor yang menyebabkan rendahnya akurasi YOLOv4-Tiny adalah jumlah lapisan yang lebih sedikit sehingga mengurangi kapasitas model untuk mendapatkan representasi gambar yang lebih kompleks. Selain itu, jumlah parameter yang lebih sedikit juga dapat membatasi kapasitas model untuk memahami dan menangkap pola yang rumit dalam data.

#### 3.2 Hasil Pengujian Model YOLOv4 dan YOLOv4-Tiny

Pengujian model pada alat bantu tongkat tunanetra dilakukan dengan jarak 1-3,5 meter dengan satu kali pengujian untuk masing-masing jarak. Pada pengujian model, akurasi dihitung dengan mengklasifikasikan hasil keluaran menggunakan *confusion matrix*. Dimana data yang berhasil mengenali atau mendeteksi objek yang sebenarnya dan melibatkan eksekusi suara yang



sesuai dengan identifikasi objek diklasifikasikan ke dalam *True Positive*. Sedangkan data yang merujuk pada situasi di mana sistem mengenali objek yang sebenarnya tidak ada, namun tetap menghasilkan suara dengan merinci kelas objek tertentu dapat diklasifikasikan ke dalam *False Positive*. Sementara itu, data yang tergolong dalam *False Negative* mencakup keadaan di mana objek aktual sebenarnya ada, namun sistem gagal mendeteksi atau mengenali objek tersebut dan sebagai hasilnya sistem tidak mengeluarkan suara. Dalam konteks deteksi objek multikelas, nilai *True Negative* tidak memiliki relevansi signifikan karena fokus deteksi objek cenderung terpusat pada kelas-kelas tertentu.

Tabel 2 Hasil Pengujian Dengan Model YOLOv4

N o	Jarak (m)	Hasil Yang Diharapkan	Hasil Deteksi Objek	Hasil Pengujian	Waktu Komputasi (s)
1	1	Terdeteksi Kursi	Terdeteksi Kursi	<i>True Positive</i>	35,04
2		Terdeteksi Meja	Terdeteksi Meja	<i>True Positive</i>	06,90
3		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	28,24
4		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	35,43
5		Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	50,61
6	1,5	Terdeteksi Kursi	Terdeteksi Kursi	<i>True Positive</i>	42,31
7		Terdeteksi Meja	Terdeteksi Meja	<i>True Positive</i>	07,75
8		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	19,68
9		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	34,30
10	2	Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	37,27
11		Terdeteksi Kursi	Terdeteksi Kursi	<i>True Positive</i>	57,28
12		Terdeteksi Meja	Terdeteksi Kulkas	<i>False Positive</i>	18,43
13		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	39,79
14		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	38,49
15	2,5	Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	34,70
16		Terdeteksi Kursi	Terdeteksi Kursi	<i>True Positive</i>	53,28
17		Terdeteksi Meja	Terdeteksi Meja	<i>True Positive</i>	18,91
18		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	32,60
19		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	38,81
20	3	Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	33,01
21		Terdeteksi Kursi	Terdeteksi Kursi	<i>True Positive</i>	46,02
22		Terdeteksi Meja	Terdeteksi Meja	<i>True Positive</i>	31,11
23		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	38,23
24		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	37,72
25	3,5	Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	30,75
26		Terdeteksi Kursi	Terdeteksi Kursi	<i>True Positive</i>	36,56
27		Terdeteksi Meja	Terdeteksi Kulkas	<i>False Positive</i>	48,39
28		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	52,34
29		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	46,00
30		Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	16,65
<b>Rata-Rata</b>					<b>34,88</b>

Berdasarkan pengujian yang dilakukan menggunakan model YOLOv4 yang dapat dilihat pada Tabel 2, terdapat 28 data pengujian yang diklasifikasikan ke dalam *True Positive* dan 2 data pengujian yang diklasifikasikan ke dalam *False Positive*. Akurasi dihitung menggunakan persamaan 2:

$$Akurasi = \frac{(28 + 0)}{(28 + 0 + 2)} \times 100\% = 93,33\%$$

Pada pengujian yang dilakukan, akurasi dihitung untuk mengukur kinerja model yang sudah dilatih dalam mengenali objek. Pada model YOLOv4 didapatkan hasil sebesar 93,33% dengan 28 data hasil pengujian memiliki nilai *True Positive* yang artinya hasil pengenalan objek sesuai dengan objek yang dikenali. Kemudian 2 data hasil pengujian memiliki nilai *False Positive* yang artinya objek yang dikenali dan hasil pengenalan objek berbeda, namun sistem tetap mengeluarkan hasil pengenalan objek melalui suara. Kesalahan deteksi yang dilakukan sistem disebabkan oleh model yang memang tidak sempurna karena mAP yang dihasilkan tidak mencapai 100%. Pada model YOLOv4 memiliki jumlah lapisan dan bobot yang lebih banyak sehingga rata-rata waktu komputasi yang dilakukan untuk melakukan deteksi objek secara realtime memakan waktu yang lebih lama. Rata-rata waktu yang dibutuhkan model YOLOv4

untuk mengenali objek saat pengujian adalah 34,88 detik.

Tabel 3 Hasil Pengujian Dengan Model YOLOv4-Tiny

No	Jarak (m)	Hasil Yang Diharapkan	Hasil Deteksi Objek	Hasil Pengujian	Waktu Komputasi (s)
1	1	Terdeteksi Kursi	Terdeteksi Kursi	<i>True Positive</i>	29,36
2		Terdeteksi Meja	Terdeteksi Kulkas	<i>False Positive</i>	06,38
3		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	6,64
4		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	31,71
5		Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	12,71
6	1,5	Terdeteksi Kursi	Terdeteksi Kursi	<i>False Positive</i>	40,17
7		Terdeteksi Meja	Terdeteksi Kulkas	<i>False Positive</i>	15,56
8		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	4,08
9		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	10,46
10		Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	6,65
11	2	Terdeteksi Kursi	Terdeteksi Kursi	<i>True Positive</i>	56,67
12		Terdeteksi Meja	Terdeteksi Kulkas	<i>False Positive</i>	9,82
13		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	5,59
14		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	4,37
15		Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	8,59
16	2,5	Terdeteksi Kursi	Terdeteksi Kulkas	<i>False Positive</i>	44,03
17		Terdeteksi Meja	Terdeteksi Kulkas	<i>False Positive</i>	31,69
18		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	2,42
19		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	3,84
20		Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	7,62
21	3	Terdeteksi Kursi	Terdeteksi Kursi	<i>True Positive</i>	52,31
22		Terdeteksi Meja	Terdeteksi Kulkas	<i>False Positive</i>	04,93
23		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	7,15
24		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	4,40
25		Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	9,9
26	3,5	Terdeteksi Kursi	Terdeteksi Kursi	<i>True Positive</i>	24,38
27		Terdeteksi Meja	Terdeteksi Kulkas	<i>False Positive</i>	42,03
28		Terdeteksi Orang	Terdeteksi Orang	<i>True Positive</i>	3,47
29		Terdeteksi Pot Tanaman	Terdeteksi Pot Tanaman	<i>True Positive</i>	5,83
30		Terdeteksi Kulkas	Terdeteksi Kulkas	<i>True Positive</i>	4,78
<b>Rata-Rata</b>					<b>16,58</b>

Berdasarkan pengujian yang dilakukan menggunakan model YOLOv4-Tiny yang dapat dilihat pada Tabel 3, terdapat 22 data pengujian yang diklasifikasikan ke dalam *True Positive* dan 8 data pengujian yang diklasifikasikan ke dalam *False Positive*. Akurasi dihitung menggunakan persamaan 2:

$$Akurasi = \frac{(22 + 0)}{(22 + 0 + 8)} \times 100\% = 73,3\%$$

Akurasi yang dihasilkan pada pengujian model YOLOv4-Tiny mendapatkan nilai sebesar 73,3% dengan 22 data hasil pengujian memiliki nilai *True Positive*. Lalu 8 data hasil pengujian memiliki nilai *False Positive*. Pada model YOLOv4-Tiny memiliki rata-rata waktu komputasi yang lebih cepat dari YOLOv4 yang telah diuji sebelumnya karena jumlah lapisan dan bobot yang lebih sedikit dibandingkan algoritma sebelumnya. Rata-rata waktu komputasi yang lebih cepat ini disebabkan karena algoritma YOLOv4-Tiny mengorbankan beberapa fitur dan kompleksitasnya. Rata-rata waktu yang dibutuhkan oleh model YOLOv4-Tiny untuk mengenali objek saat pengujian dilakukan adalah 16,58 detik.

#### 4. KESIMPULAN DAN SARAN

Pengenalan objek menggunakan algoritma YOLO melalui *webcam* dapat dilakukan dengan baik. *Webcam* akan menangkap objek kemudian melakukan proses pengenalan dan mengklasifikasikan objek sesuai kelas objek. Hasil dari pengenalan tersebut kemudian dikeluarkan melalui *earphone* berupa suara yang berisi informasi objek yang dikenali. Keluaran suara yang dihasilkan diolah menggunakan perangkat lunak sintesis teks menjadi suara yaitu

eSpeak dan MBROLA sehingga menjadikan penelitian ini berbeda dari penelitian terkait.

Hasil akurasi menggunakan mAP yang didapatkan setelah pelatihan pada model YOLOv4 yaitu sebesar 99,74%, sedangkan pada model YOLOv4-Tiny didapatkan nilai mAP sebesar 67,78%. Sementara itu, hasil akurasi sistem saat pengujian yang diletakkan pada alat bantu tongkat tunanetra menggunakan model YOLOv4 yaitu sebesar 93,33% dengan rata-rata waktu komputasi yang dihasilkan sebesar 34,88 detik dan akurasi model YOLOv4-Tiny sebesar 73,3% disertai rata-rata waktu komputasi yang dihasilkan sebesar 16,58 detik. Penelitian ini menggunakan YOLOv4 dan YOLOv4-Tiny yang menghasilkan akurasi pengujian yang lebih baik dibandingkan penelitian terkait yang menghasilkan rata-rata keberhasilan pengujian sebesar 40% menggunakan algoritma MobileNet untuk mendeteksi objek pada alat bantu tunanetra.

Penelitian ini dapat dikembangkan menggunakan metode atau algoritma YOLO lainnya seperti YOLOv5 hingga YOLOv8 agar dapat menghasilkan tingkat akurasi dan waktu komputasi yang lebih baik. Kemudian memperluas cakupan kelas objek yang dapat dideteksi untuk memberikan ragam yang lebih luas dalam upaya mendukung tunanetra menjalankan aktivitas sehari-hari.

#### DAFTAR PUSTAKA

- [1] A. Praptaningrum, "Penerapan Bahan Ajar Audio untuk Anak Tunanetra Tingkat SMP di Indonesia," *Jurnal Teknologi Pendidikan*, vol. 5, no. 1, pp. 1–19, Oct. 2020.
- [2] E. P. Herlambang and M. S. Suteja, "Perancangan Fasilitas Pembinaan dan Rekreasi Tunanetra dengan Pendekatan Indera," *Jurnal Sains, Teknologi, Urban, Perancangan, Arsitektur (Stupa)*, vol. 5, no. 2, pp. 1765–1778, Oct. 2023, doi: 10.24912/stupa.v5i2.24315.
- [3] S. Fuady, N. Nehru, and G. Anggraeni, "Deteksi Objek Menggunakan Metode Single Shot Multibox Detector Pada Alat Bantu Tongkat Tunanetra Berbasis Kamera," *Journal of Electrical Power Control and Automation (JEPCA)*, vol. 3, no. 2, p. 39, Dec. 2020, doi: 10.33087/jepca.v3i2.38.
- [4] F. Ramasari, F. Firdaus, S. Nita, and K. Kartika, "Penggunaan Metode You Only Look Once dalam Penentu Pindah Tanaman Cabai Besar Ternetifikasi Telegram," *Elektron : Jurnal Ilmiah*, pp. 45–52, Nov. 2021, doi: 10.30630/eji.13.2.229.
- [5] H. H. Al Asyhar, S. A. Wibowo, and G. Budiman, "Implementasi Dan Analisis Performansi Metode You Only Look Once (Yolo) Sebagai Sensor Pornografi Pada Video Implementation And Performance Analysis Of You Only Look Once (Yolo) Method As Porn Censorship In Video," *e-Proceeding of Engineering*, vol. 7, no. 2, pp. 3631–3638, Aug. 2020.
- [6] S. Ariyani, A. B. Nugroho, and A. S. T. Mubarak, "Alat Bantu Pendeteksi Objek Untuk Tuna Netra Berbasis Ai Mobilenet Pada Raspberry Pi 3B," *Jurnal Teknik Elektro dan Komputasi (ELKOM)*, vol. 4, no. 1, pp. 73–90, Mar. 2022, doi: 10.32528/elkom.v4i1.4951.
- [7] O. E. Karlina and D. Indarti, "Pengenalan Objek Makanan Cepat Saji Pada Video Dan Real Time Webcam Menggunakan Metode You Look Only Once (YOLO)," *Jurnal Ilmiah Informatika Komputer*, vol. 24, no. 3, pp. 199–208, Dec. 2019, doi: 10.35760/ik.2019.v24i3.2362.
- [8] R. Dwiyanto, D. W. Widodo, and P. Kasih, "Implementasi Metode YOLOv5 untuk Klasifikasi Kendaraan Pada CCTV kab. Tulungagung," *Seminar Nasional Inovasi Teknologi*, pp. 102–104, Jul. 2022.
- [9] A. Fali Oklilas, D. Dwinta, G. Shofi, N. Putri Mariza, S. Arum Kinanti, and Y. Amanda Sari, "Akurasi Pengujian Model Hasil Training menggunakan YOLOv4 untuk Pengenalan Kendaraan di Jalan Raya," *Jurnal JUPITER*, vol. 15, pp. 799–806, Apr. 2023.
- [10] R. Syamsul Huda, R. Wulanningrum, and D. Swanjaya, "Pemanfaatan YOLOv4 Untuk Deteksi Pelanggaran Helm Dan Masker Serta Identifikasi Pelat Nomor Menggunakan Tesseract-Ocr," *JOUTICA*, vol. 7, no. 2, pp. 592–598, 2022.
- [11] Q. Chen and Q. Xiong, "Garbage Classification Detection Based on Improved YOLOV4," *Journal of Computer and Communications*, vol. 08, no. 12, pp. 285–294, 2020, doi: 10.4236/jcc.2020.812023.

- [12] A. Ospanova *et al.*, “Real Time Object Recognition with Voice Feedback using YOLOv4-tiny Deep Learning Model,” *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, vol. 11, no. 4, pp. 3932–3937, 2022, doi: 10.15680/IJIRSET.2022.1104137.
- [13] S. Saponara, A. Elhanashi, and Q. Zheng, “Developing a real-time social distancing detection system based on YOLOv4-tiny and bird-eye view for COVID-19,” *J Real Time Image Process*, vol. 19, no. 3, pp. 551–563, Jun. 2022, doi: 10.1007/s11554-022-01203-5.
- [14] F. Akhyar, L. Novamizanti, and T. Riantiarni, “Sistem Inspeksi Cacat pada Permukaan Kayu menggunakan Model Deteksi Obyek YOLOv5,” *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 10, no. 4, pp. 990–1005, Oct. 2022, doi: 10.26760/elkomika.v10i4.990.
- [15] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 2020, [Online]. Available: <http://arxiv.org/abs/2004.10934>
- [16] Q. Aini, N. Lutfiani, H. Kusumah, and M. S. Zahran, “Deteksi Dan Pengenalan Objek Dengan Model Machine Learning: Model YOLO,” *CESS(Journal of Computer Engineering System and Science)*, vol. 6, no. 2, pp. 2502–714, Jul. 2021.
- [17] Z. Jiang, L. Zhao, S. Li, Y. Jia, and Z. Liquan, “Real-Time Object Detection Method for Embedded Devices”.