

# IMPLEMENTASI *OBJECT RECOGNITION* PADA RAMBU-RAMBU DAN LAMPU LALU LINTAS DENGAN RASPBERRY PI DENGAN ALGORITMA YOLOV5

Agung Nugroho<sup>1</sup> dan Muhammad Ridwan Arif Cahyono<sup>2</sup>

<sup>1</sup>Manajemen Informatika, AMIK Mapan

<sup>2</sup>Teknik Elektronika, Politeknik Gajah Tunggal

<sup>1</sup>Jl. Gatot Subroto KM 9, Cimone, Tangerang, 15114

<sup>2</sup>Jl. Gatot Subroto KM 7, Jatiuwung, Tangerang, 15135

E-mail: agung.nugroho@amik.mapan.ac.id<sup>1</sup>, ridwan@poltek-gt.ac.id<sup>2</sup>

## ABSTRAK

Salah satu teknologi yang saat ini berkembang pesat khususnya untuk mendukung mobil otonom adalah fitur ADAS (*Advanced Driver Assistance System*). Salah satu fitur sistem ADAS yaitu kemampuan untuk mendeteksi dan mengenali rambu-rambu lalu lintas. Dalam penelitian ini akan dikembangkan dan diimplementasikan sistem pendeteksi lampu dan rambu lalu lintas berbasis kecerdasan buatan. Algoritma yang digunakan dalam penelitian ini yaitu dengan algoritma YOLO (*You Only Look Once*) versi 5. Sistem kecerdasan buatan ini akan diimplementasikan dalam perangkat Raspberry Pi yang dilengkapi dengan *webcam* dan *speaker*. Total jenis rambu yang digunakan sejumlah 12 jenis rambu lalu lintas dengan masing-masing data citra yang diolah sejumlah 100 data. Jenis rambu dan lampu lalu lintas yang diklasifikasikan meliputi : belok kanan, belok kiri, dilarang belok kanan, dilarang belok kiri, dilarang putar balik, dilarang berhenti, dilarang parkir, lampu merah, lampu hijau, lampu kuning, dan putar balik. Dilakukan variasi ukuran gambar dari 320x320, 416x416 dan 480x480, serta dilakukan variasi jumlah *training* yaitu 50, 100, 250 dan 500 *epoch* untuk mendapatkan nilai akurasi yang paling optimal. Didapatkan hasil yang optimal dengan ukuran gambar 320x320 pixel dan dengan jumlah *training* sejumlah 50 *epoch*. Dari hasil *training* didapatkan mAP (*mean Average Precision*) sebesar 89,50%, di mana ketika dilakukan pengujian gambar lampu dan rambu lalu lintas didapatkan ketepatan sebesar 88,88%. Sistem mampu diimplementasikan dalam perangkat Raspberry Pi 4 dan dipasang di mobil dan mampu berjalan dengan baik.

**Kata Kunci:** ADAS, YOLOv5, Traffic Light, Traffic Lamp, Rekognisi Objek, Raspberry Pi

## 1. PENDAHULUAN

Berdasarkan data tahun 2017 dari KORLANTAS Polri terdapat 37.204 kecelakaan lalu lintas yang terjadi (Ismail, 2019), salah satu faktor penyebab kecelakaan karena pengemudi yang kurang tertib atau melanggar rambu-rambu lalu lintas. Teknologi untuk meningkatkan keselamatan saat berkendara saat ini sudah mulai diterapkan oleh beberapa pabrikan otomotif, namun masih terbatas pada kendaraan pribadi dengan kategori mobil premium. Salah satu teknologi untuk tersebut adalah *Advanced Driver Assistance System* (ADAS). Mayoritas kendaraan yang beredar di masyarakat belum dilengkapi oleh fitur ADAS tersebut.

Dengan adanya kemajuan teknologi yang sangat pesat, semua pihak berlomba-lomba untuk menghasilkan atau mengembangkan teknologi yang sesuai khususnya terkait dengan keselamatan berkendara. Salah bentuk pengembangan sistem transportasi pintar adalah aplikasi *assistant driver*, di mana aplikasi ini bertujuan untuk membantu dan meningkatkan kinerja pengemudi sehingga mampu untuk meningkatkan keselamatan berkendara.

Salah satu teknologi yang saat ini dikembangkan oleh industri otomotif yaitu ADAS. Teknologi ini mampu membantu pengemudi dalam meningkatkan keselamatan di jalan, dan memberikan informasi kepada pengemudi tentang perlunya berhati-hati dan memperhatikan rambu-rambu. Saat

ini sistem ADAS hanya ditemukan di mobil-mobil penumpang tipe tertentu, masih banyak kendaraan yang belum dilengkapi fitur ini. Dengan adanya sistem yang mampu mendeteksi lampu dan rambu lalu lintas, diharapkan mampu menurunkan tingkat penaggaran lalu lintas dan menurunkan tingkat kecelakaan di jalan raya.

Dalam beberapa tahun terakhir, deteksi rambu lalu lintas telah menjadi subjek penelitian yang penting di bidang *computer vision*. Selain itu, penelitian tentang deteksi rambu lalu lintas berkembang pesat karena untuk mendukung sistem kemudi otomatis mobil, sistem ADAS, yang bertujuan untuk peningkatan sistem dukungan pengemudi dan meningkatkan keselamatan lalu lintas (Dewi dkk., 2020; Liu dkk., 2019; Luo dkk., 2018). Banyak aplikasi berbasis kecerdasan buatan seperti kendaraan otonom (Bansal, 2018), *smart surveillance* (Krishnaveni, 2022), dan kota pintar (Ahad, dkk., 2020), telah dianggap sebagai fondasi bagi pengembangan kota pintar. Deteksi objek memainkan peran penting dalam mengembangkan kota pintar dalam kondisi lalu lintas normal atau lingkungan otonom.

Dengan adanya kemajuan teknologi yang sangat pesat, semua pihak berlomba-lomba untuk menghasilkan atau mengembangkan teknologi yang sesuai khususnya terkait dengan keselamatan berkendara. Salah satu opsi untuk pengembangan transportasi pintar adalah aplikasi *assistant driver*, yang bertujuan untuk meningkatkan kinerja pengemudi dan meningkatkan keselamatan berkendara. Salah satu teknologi yang saat ini dikembangkan oleh industri otomotif yaitu ADAS. Teknologi ini mampu membantu pengemudi dalam meningkatkan keselamatan di jalan, dan memberikan informasi kepada pengemudi tentang perlunya berhati-hati dan memperhatikan rambu-rambu. Saat ini sistem ADAS hanya ditemukan di mobil-mobil penumpang tipe tertentu, masih banyak kendaraan yang belum dilengkapi fitur ini. ADAS adalah teknologi yang menyediakan informasi yang dibutuhkan pengemudi dan mengarah pada peningkatan keselamatan mobil secara keseluruhan. Beberapa dari teknologi ini telah terbukti meningkatkan pengalaman berkendara dan keamanan jalan yang lebih baik secara keseluruhan.

Penelitian mengenai identifikasi rambu lalu lintas sudah dilakukan sebelumnya, beberapa pendekatan yang digunakan antara lain artinya penggabungan *support vector machine* (SVM) serta *Knearest neighbor* (KNN) yang memperoleh akurasi sebesar 93,9% (Rahmad dkk., 2017), serta dengan metode *template matching* dengan akurasi pengenalan sebesar 88% (Chrisdwianto dkk., 2018). Lebih lanjut, klasifikasi dengan teknik *shape detection* yang dipergunakan untuk mengenali rambu lalu lintas berbentuk bulat memperoleh akurasi sebesar 80,7 % (Sitanggang dkk., 2018). Penelitian lain yang sudah dilakukan yaitu membandingkan metode *Circle Hough Transform* (CHT) dan *convolutional neural network* (CNN) di mana akurasi masing-masing sebesar 61,3 % dan 75 % (Triyadi & Utaminingrum, 2020).

Pendekatan CNN yang telah dilakukan pada penelitian tersebut belum maksimal, karena masih menghasilkan akurasi yang rendah. Hal ini dikarenakan terjadi *overfitting* pada objek yang diidentifikasi.

Beberapa tahun terakhir telah berkembang secara signifikan proses deteksi objek menggunakan metode *Deep Neural Network* (Geirhos, dkk., 2017, Lin dkk., 2017). Beberapa model *dataset* dalam pengklasifikasian objek yang sudah dikembangkan sebelumnya seperti MS COCO (Sonntag, dkk., 2017) dan PASCALVOC (Kong, dkk., 2017) sangat mendukung pengembangan aplikasi pendeteksian objek.

Pada sistem pendeteksi rambu lalu lintas, sangat penting untuk secara andal dan cepat mendeteksi objek kecil, di mana mobil yang bergerak dengan kecepatan tinggi. Selain itu, CNN dapat secara akurat mengenali dan mengkategorikan objek menggunakan teknik seperti YOLO (Du, 2018), *Single Shot Multibox Detector* (SSD), dan *Faster R-CNN* (Ren dkk., 2017). Ketika deteksi waktu yang lebih cepat diperlukan, algoritma YOLO memiliki kemampuan di mana memiliki akurasi yang baik dan cepat dalam mendeteksi objek. Selain itu algoritma YOLO mampu mendeteksi objek secara *realtime* (Redmon & Farhadi,

2017). Saat ini algoritma YOLO sudah mencapai versi 5 atau dikenal dengan YOLOv5.

Penelitian ini bertujuan untuk merancang sebuah sistem ADAS yang mampu untuk mengurangi potensi pelanggaran rambu dan lampu lalu lintas. Sistem ADAS yang dirancang diharapkan mampu untuk mengenali berbagai rambu-rambu lalu lintas dan lampu lalu lintas, serta memberikan peringatan kepada pengemudi. Sistem ADAS dibangun dengan mengimplementasikan algoritma YOLOv5 untuk mendeteksi rambu-rambu dan lampu lalu lintas di perangkat *Raspberry Pi 4*. *Raspberry Pi* dipilih karena dapat dengan mudah dipindahkan ke mobil dan dapat mengambil catu daya DC 5V dari mobil.

## 2. RUANG LINGKUP

Dalam penelitian ini permasalahan mencakup:

### 2.1 Cakupan Permasalahan

Cakupan permasalahan dalam penelitian ini membahas tentang proses pengidentifikasian objek rambu-rambu dan lampu lalu lintas, di mana total jenis objek dan rambu-rambu yang akan diidentifikasi meliputi : belok kanan, belok kiri, dilarang belok kanan, dilarang belok kiri, dilarang putar balik, dilarang berhenti, dilarang parkir, lampu merah, lampu hijau, lampu kuning, dan putar balik. Jumlah sampel yang digunakan yaitu 100 gambar per kategori.

### 2.2 Batasan Penelitian

Dalam penelitian ini memiliki batasan, di mana algoritma yang digunakan yaitu YOLOv5 yang sudah dikembangkan oleh komunitas dan dapat diakses melalui repo Github dengan alamat berikut [github.com/ultralytics/yolov5](https://github.com/ultralytics/yolov5). Pada proses anotasi data menggunakan aplikasi *RoboFlow*, sedangkan untuk proses *training* menggunakan aplikasi *Google Collab*. Sistem diimplementasikan dalam *Raspberry Pi 4* dengan spesifikasi RAM 4 GB dan CPU 64-bit 1.5GHz. Bahasa pemrograman yang digunakan yaitu Bahasa Python.

### 2.3 Rencana Hasil yang Didapatkan

Rencana hasil dari penelitian ini berupa sistem alat bantu tambahan yang diimplementasikan dalam *Raspberry Pi* sebagai perangkat *driver assistance*.

## 3. BAHAN DAN METODE

Penelitian ini dimulai dengan persiapan dataset, dilanjutkan dengan proses *training* data dengan berbagai variasi untuk menghasilkan model deteksi objek terbaik, dan dilanjutkan dengan implementasi ke *Raspberry Pi*.

### 3.1 Persiapan Dataset

Penelitian ini dimulai dengan mengumpulkan data gambar lampu lalu lintas dan rambu lalu lintas. Di

mana total terdapat 12 jenis objek yang akan diidentifikasi, masing-masing terdiri dari 100 jenis gambar. Gambar ini diperoleh dari sumber gambar internet, mengambil gambar secara langsung dan mengambil dari aplikasi *Google Street View*. Selanjutnya masing-masing gambar tersebut dilakukan proses anotasi menggunakan aplikasi *Roboflow* untuk menjadi dataset. Proses anotasi adalah proses pengidentifikasian objek utama dalam satu *frame* gambar. Beberapa hasil anotasi gambar ditunjukkan pada Gambar 1.

### 3.2 Proses Training dengan Algoritma YOLOv5

Dataset rambu-rambu di *training* dengan menggunakan aplikasi *Google Collab*. *Google Collab* merupakan layanan dari Google yang menyediakan GPU dengan ukuran RAM sebesar 12 GB. Bahasa pemrograman yang digunakan yaitu menggunakan bahasa *Python*. Algoritma yang digunakan dalam mendeteksi objek yaitu dengan menerapkan algoritma YOLO (*You Only Look Once*), di mana YOLO yang digunakan merupakan YOLO versi 5. Penelitian sejenis juga dilakukan dalam (Novak dkk., 2020; Wang dkk., 2022). Pada penelitian tersebut masih mengadopsi YOLO v3 dan YOLO v4, sedangkan penelitian ini sudah menggunakan YOLOv5.

Pada proses *training* dilakukan variasi ukuran *pixel* gambar, di mana beberapa variasi yang digunakan yaitu 320 x 320, 416 x 416 dan 480 x 480 untuk dicari variasi dengan

tingkat kesalahan paling kecil. Selanjutnya dilakukan variasi jumlah *training* untuk menentukan objek dengan tingkat kesalahan paling minimal, di mana variasi yang dilakukan yaitu 50, 100, 250 dan 500 *epoch*. Keluaran dari proses *training* yaitu menghasilkan model *database* sebagai acuan dalam klasifikasi lampu dan rambu-rambu lalu lintas.

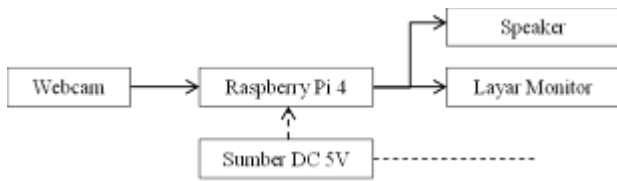
Model dengan tingkat kesalahan paling rendah, kemudian diujikan pada gambar rambu-rambu dan lampu lalu lintas yang lain. Dalam proses pengujian ini, masing-masing jenis rambu diujikan dengan 2 gambar yang berbeda dibandingkan dengan gambar untuk proses *training*.

### 3.3 Implementasi Sistem

Algoritma YOLOv5 merupakan algoritma yang diimplementasikan dalam bahasa *Python*. Algoritma ini diimplementasikan dalam perangkat *Raspberry Pi 4*, di mana memiliki spesifikasi RAM 4 GB dengan kapasitas penyimpanan 16 GB. Untuk mendeteksi gambar secara *real time*, digunakan *Webcam Full HD* seperti pada Gambar 2. Hasil pengujian ditampilkan dalam layar monitor 7" dan ditambahkan perangkat *speaker* untuk menampilkan suara ketika mendeteksi objek. Untuk catu daya diambil dari sumber tegangan 5V dari mobil.



Gambar 1. Proses anotasi pada 12 jenis gambar dataset



**Gambar 2. Konfigurasi hardware dalam implementasi YOLOv5**

#### 4. PEMBAHASAN

Algoritma YOLOv5 berhasil diimplementasikan untuk proses pendeteksian rambu dan lampu lalu lintas dengan menggunakan Bahasa Python. Jumlah data gambar yang digunakan sebagai *dataset* sudah cukup, karena sudah menghasilkan presisi yang baik. Sistem berhasil diimplementasikan dalam perangkat berbasis *Raspberry Pi* 4. Adapun detail pengujian untuk setiap komponen sebagai berikut:

##### 4.1 Pengaruh Ukuran Gambar Terhadap Akurasi

Variasi gambar yang digunakan yaitu gambar dengan ukuran 320x320, 416x416 dan 480x480. Resolusi gambar ini dipilih karena merupakan kelipatan 32. mAP (*mean Average Precision*) merupakan parameter yang menunjukkan tingkat akurasi. Hasil pengujian pengaruh variasi ukuran *pixel* gambar terhadap mAP ditunjukkan pada Tabel 1.

**Tabel 1. Nilai mAP terhadap variasi ukuran gambar**

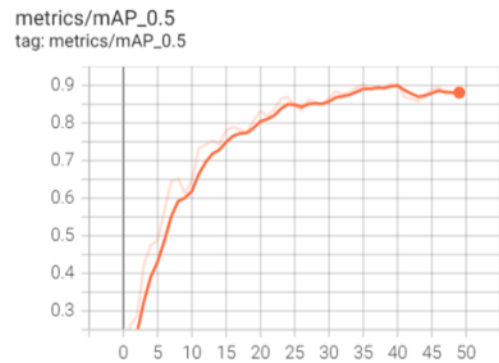
Jenis Rambu	320 x 320	416 x 416	480 x 480
Belok Kanan	0.783	0.687	0.569
Belok Kiri	0.684	0.662	0.668
Dilarang Belok Kanan	0.876	0.895	0.849
Dilarang Belok Kiri	0.995	0.978	0.995
Dilarang Berhenti	0.995	0.995	0.995
Dilarang Parkir	0.884	0.696	0.836
Dilarang Putar Balik	0.951	0.860	0.859
Kecepatan Mak. 80 km/jam	0.995	0.995	0.995
Lampu Hijau	0.843	0.995	0.845
Lampu Kuning	0.954	0.977	0.958
Lampu Merah	0.848	0.796	0.855
Putar Balik	0.931	0.976	0.944
mAP Keseluruhan	0.895	0.876	0.864

Dari Tabel 1 tersebut, terlihat bahwa semakin besar resolusi gambar maka tingkat kesalahan semakin besar. Akurasi terbaik didapatkan pada saat pengujian dengan ukuran gambar 320x320 *pixel* dengan nilai sebesar 0.895 atau 89.5%. Hasil ini lebih baik dari penelitian sebelumnya, di mana pada penelitian tersebut dihasilkan nilai mAP 87.98% (Hasibuan dkk., 2021) sedangkan pada penelitian lainnya menghasilkan mAP sebesar 82.5% (Mohd-Isa dkk., 2020).

##### 4.2 Pengaruh Jumlah Training terhadap Akurasi

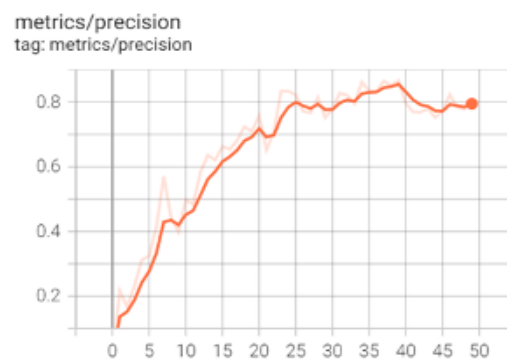
Dalam pengujian ini, ukuran gambar yang digunakan yaitu 320x320 *pixel*, dengan variasi *training* data sejumlah 50, 100, 250, dan 500. Semakin banyak jumlah *training* maka durasi pelatihan semakin lama. Hasil pengujian

ditunjukkan pada Tabel 2. Dari Tabel 2, terlihat bahwa jumlah pelatihan tidak sebanding dengan tingkat akurasi. Akurasi paling optimal didapatkan dengan jumlah perulangan sebanyak 50 kali. Semakin banyak perulangan belum tentu menghasilkan objek dengan akurasi yang lebih baik.



**Gambar 3. Hubungan mAP dengan jumlah pelatihan**

Dari Gambar 3 terlihat bahwa akurasi meningkat seiring dengan bertambahnya jumlah *training*. Di mana sumbu x adalah banyaknya jumlah *training* dan sumbu y adalah mAP. mAP merupakan salah satu parameter yang digunakan untuk menentukan presisi dari model dalam sistem kecerdasan buatan. Nilai mAP maksimum adalah 1, semakin besar mAP, maka model tersebut semakin presisi. Dari Gambar 3 terlihat bahwa akurasi terbaik didapatkan ketika jumlah *training* sebanyak 40 kali, kemudian akurasi menurun ketika jumlah *training* sebesar 43 kali, kemudian naik kembali ketika *training* sama dengan 50 kali.



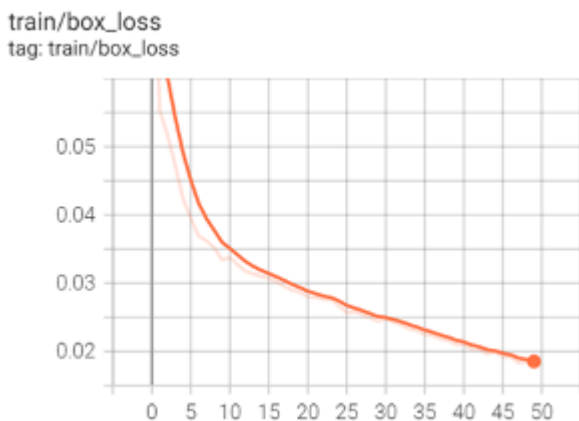
**Gambar 4. Hubungan antara jumlah training dengan presisi**

Dari Gambar 4 terlihat bahwa model yang digunakan sudah memiliki presisi yang baik, di mana presisi yang didapat sudah diatas 0,8 atau 80%. Hal ini lebih baik dari penelitian lainnya, di mana didapatkan presisi antara 0,5 – 0,6 dengan perulangan sebanyak 40 kali (Cengil & Çinar, 2021). Pada Gambar 5, terlihat bawa *box loss* semakin turun ketika jumlah pelatihan semakin banyak.

**Tabel 2. Nilai mAP terhadap variasi banyaknya pelatihan**

Jenis Rambu	Jumlah Pelatihan			
	50	100	250	500
Belok Kanan	0.783	0.691	0.620	0.614
Belok Kiri	0.684	0.582	0.639	0.608
Dilarang Belok Kanan	0.876	0.853	0.836	0.909
Dilarang Belok Kiri	0.995	0.978	0.978	0.995
Dilarang Berhenti	0.995	0.995	0.995	0.995
Dilarang Parkir	0.884	0.887	0.912	0.890
Dilarang Putar Balik	0.951	0.919	0.837	0.840
Kecepatan Mak. 80 km/jam	0.995	0.995	0.995	0.995
Lampu Hijau	0.843	0.995	0.945	0.945
Lampu Kuning	0.954	0.945	0.968	0.968
Lampu Merah	0.848	0.894	0.978	0.838
Putar Balik	0.931	0.918	0.918	0.944
mAP Keseluruhan	0.895	0.888	0.882	0.887

Sedangkan semakin kecil *box loss*, maka akurasi semakin baik. Hasil *box loss* didapatkan kurang dari 0.02, di mana lebih baik dari penelitian lainnya dengan *box loss* sebesar 0.04 (Yu dkk., 2021). Sehingga model yang akan diimplementasikan yaitu menggunakan *training* dengan perulangan 50 kali.



**Gambar 5. Hubungan antara jumlah *training* dengan kesalahan**

#### 4.3 Pengujian Model dengan File Gambar

Pengujian dengan file gambar terhadap model YOLOv5 ditunjukkan pada Gambar 6 dan Gambar 7. Pada kondisi pengujian tertentu, dalam 1 gambar terdapat beberapa rambu-rambu, seperti pada Gambar 6c, pada pengujian rambu dilarang belok kanan terdapat dua buah rambu dan lampu merah, di mana berhasil terdeteksi semuanya. Dari 14 jenis rambu dan lampu lalu lintas pada Gambar 6, tidak ditemukan kesalahan dalam pengklasifikasian gambar.

Sedangkan pada Gambar 7, total terdapat 15 objek jenis rambu dan lampu lalu lintas, dari hasil pengujian didapatkan

3 kesalahan, yaitu pada Gambar 7b di mana seharusnya berupa gambar belok kiri, tetapi terdeteksi sebagai belok kanan dan pada Gambar 7l, dilarang belok kanan dibaca sebagai dilarang belok kiri dan mendeteksi objek dilarang parkir. Hal ini dapat dijelaskan sebagaimana Tabel 2, di mana untuk pendeteksi rambu belok kanan dan belok kiri memiliki akurasi yang paling rendah, di mana berturut-turut 0.783 dan 0.684. Apabila dibandingkan dengan rambu dan lampu lalu lintas lainnya, memiliki akurasi diatas 0.8. Hasil ketepatan pengujian seperti pada Gambar 6 dan Gambar 7 dapat dilihat pada Tabel 3. Di mana secara rata-rata dengan pengujian gambar rambu dan lampu lalu lintas didapatkan ketepatan pendeteksian sebesar 88,88%.

**Tabel 3. Ketepatan pendeteksian YOLOv5 terhadap lampu dan rambu lalu lintas**

Jenis Rambu	Benar	Salah	Ketepatan
Belok Kanan	2	0	100%
Belok Kiri	1	1	50%
Dilarang Belok Kanan	4	0	100%
Dilarang Belok Kiri	2	0	100%
Dilarang Berhenti	2	0	100%
Dilarang Parkir	2	0	100%
Dilarang Putar Balik	2	0	100%
Kecepatan Mak. 80 km/jam	2	0	100%
Lampu Hijau	3	0	100%
Lampu Kuning	2	0	100%
Lampu Merah	2	2	50%
Putar Balik	2	0	100%
<b>Ketepatan Total</b>	<b>24</b>	<b>3</b>	<b>88.88%</b>

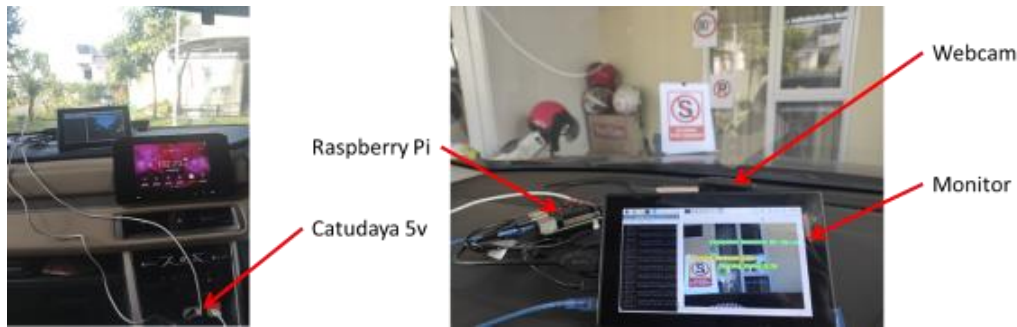




**Gambar 6. Pengujian 1 dengan objek gambar rambu dan lampu lalu lintas**



**Gambar 7. Pengujian 2 dengan objek gambar rambu dan lampu lalu lintas**



**Gambar 8. Implementasi YOLO 5 dalam Raspberry Pi di mobil**

#### 4.4 Implementasi dalam Raspberry Pi

Setelah dilakukan pengujian, sistem diimplementasikan dalam Raspberry Pi 4. Raspberry Pi sebelumnya juga digunakan dalam penelitian (M.R. dkk., 2019) untuk mendeteksi wajah, penelitian (Ansor & Afrianto, 2020) untuk mendeteksi penggunaan masker, dan penelitian (Asmara dkk., 2020) untuk mendeteksi kepadatan lalu lintas. Sebelumnya Raspberry Pi 4 yang digunakan sudah diinstall dengan OS 64 bit dan file *library-library* yang diperlukan untuk menjalankan YOLOv5 sudah diinstall. Raspberry Pi ditempatkan di dalam mobil, sumber catu daya berasal dari catu daya 5V di mobil. Implementasi sistem ditunjukkan seperti pada Gambar 8.

Sistem sudah dapat diimplementasikan dengan baik pada Raspberry Pi 4 dan mampu mengidentifikasi rambu-rambu lalu lintas dengan baik. Rambu-rambu yang digunakan dalam implementasi ini masih berupa rambu yang dicetak dalam kertas berukuran A4. Ketika mendeteksi objek, sistem akan mengeluarkan peringatan suara melalui *Speaker*. Dalam implementasi pada Gambar 8, menggunakan *Webcam* yang *build-in Speaker*. Kendala yang dihadapi pada saat proses implementasi yaitu FPS (*Frame Per Second*) yang dihasilkan masih relatif rendah, di mana rata-rata kurang dari 8 FPS. Hal ini berdampak pada waktu pendeteksian objek yang lebih lama dan terkesan terjadi *lag* pada gambar.

#### 5. KESIMPULAN

Pada penelitian ini dirancang dan diimplementasikan salah satu fitur ADAS (*Advanced Driving Assistant System*) yaitu mendeteksi lampu dan rambu lalu lintas. Algoritma YOLOv5 berhasil diimplementasikan dalam perangkat Raspberry Pi 4 sebagai *object recognition* untuk mendeteksi rambu dan lampu lalu lintas. Jenis lampu dan rambu lalu lintas yang digunakan meliputi: belok kanan, belok kiri, dilarang belok kanan, dilarang belok kiri, dilarang putar balik, dilarang berhenti, dilarang parkir, lampu merah, lampu hijau, lampu kuning, dan putar balik. Dari 12 jenis *dataset* tersebut, didapatkan mAP (*mean Average Precision*) sebesar 0.895 dengan ukuran gambar terbaik yaitu 320 x 320 pixel dan dengan 50 perulangan. Dari hasil pengujian didapatkan ketepatan pendeteksian objek sebesar 88.88%. Sistem berhasil diimplementasikan ke dalam mobil dan memberikan pesan peringatan suara Ketika mendeteksi objek.

#### 6. SARAN

Pada penelitian selanjutnya, diharapkan untuk meningkatkan FPS dari Raspberry Pi 4, saat ini FPS yang dihasilkan masih rendah. Selain itu agar lebih memudahkan pengguna, dapat dikembangkan sistem deteksi ini ke dalam aplikasi *smartphone*.

#### 7. DAFTAR PUSTAKA

- Ahad, M. A., Paiva, S., Tripathi, G., & Feroz, N. (2020). Enabling technologies and sustainable smart cities. *Sustainable cities and society*, 61, 102301.
- Ansor, A., & Afrianto, Y. (2020). Mask Detection Using Framework Tensorflow and Pre-Trained CNN Model Based on Raspberry Pi. *Jurnal Mantik*, 4(3), 1539–1545. <https://iocscience.org/ejournal/index.php/mantik>
- Asmara, R. A., Syahputro, B., Supriyanto, D., & Handayani, A. N. (2020). Prediction of traffic density using yolo object detection and implemented in raspberry pi 3b + and intel ncs 2. *4th International Conference on Vocational Education and Training, ICOVET 2020*, 391–395. <https://doi.org/10.1109/ICOVET50258.2020.9230145>
- Bansal, P., & Kockelman, K. M. (2018). Are we ready to embrace connected and self-driving vehicles? A case study of Texans. *Transportation*, 45(2), 641-675.
- Cengil, E., & Çınar, A. (2021). Poisonous Mushroom Detection using YOLOV5. Dalam *Turkish Journal of Science & Technology* (Vol. 16, Issue 1).
- Chrisdwianto, T. O., Fitriyah, H., & Widasari, E. R. (2018). Perancangan Sistem Deteksi dan Pengenalan Rambu Peringatan Menggunakan Metode Template Matching. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(3), 1265–1274. <http://j-ptiik.ub.ac.id>
- Dewi, C., Chen, R. C., & Tai, S. K. (2020). Evaluation of robust spatial pyramid pooling



- based on convolutional neural network for traffic sign recognition system. *Electronics (Switzerland)*, 9(6). <https://doi.org/10.3390/electronics9060889>
- Du, J. (2018, April). Understanding of object detection based on CNN family and YOLO. Di *Journal of Physics: Conference Series* (Vol. 1004, No. 1, p. 012029). IOP Publishing.
- Geirhos, R., Janssen, D. H., Schütt, H. H., Rauber, J., Bethge, M., & Wichmann, F. A. (2017). Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*.
- Hasibuan, N. N., Zarlis, M., & Efendi, S. (2021). Detection and tracking different type of cars with YOLO model combination and deep sort algorithm based on computer vision of traffic controlling. *Jurnal Dan Penelitian Teknik Informatika*, 6(1). <https://doi.org/10.33395/sinkron.v6i1.11231>
- Ismail, S. (2019). Analisis Probabilitas Pengemudi Mobil Melakukan Driving Distractions di Kota Medan.
- Kong, T., Sun, F., Yao, A., Liu, H., Lu, M., & Chen, Y. (2017). Ron: Reverse connection with objectness prior networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5936-5944).
- Krishnaveni, P., & Sutha, J. (2020). Novel deep learning framework for broadcasting abnormal events obtained from surveillance applications. *Journal of Ambient Intelligence and Humanized Computing*, 1-15.
- Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).
- Liu, C., Li, S., Chang, F., & Wang, Y. (2019). Machine Vision Based Traffic Sign Detection Methods: Review, Analyses and Perspectives. *IEEE Access*, 7, 86578–86596. <https://doi.org/10.1109/ACCESS.2019.2924947>
- Luo, H., Kong, Q., & Wu, F. (2018). Traffic Sign Image Synthesis with Generative Adversarial Networks. *2018 24th International Conference on Pattern Recognition (ICPR)*, 2540–2545.
- Mohd-Isa, W. N., Abdullah, M. S., Sarzil, M., Abdullah, J., Ali, A., & Hashim, N. (2020, Oktober 26). Detection of Malaysian Traffic Signs via Modified YOLOv3 Algorithm. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020*. <https://doi.org/10.1109/ICDABI51230.2020.9325690>
- M.R., S., Ghate R, G., Khan, F., & S., H. (2019). Object Detection and Human Identification using Raspberry Pi. *ICAIT - 2019 : First IEEE International Conference on Advances in Information Technology : Proceedings : 24th - 27th July 2019*.
- Novak, B., Ilić, V., & Pavković, B. (2020). YOLOv3 Algorithm with additional convolutional neural network trained for traffic sign recognition. *2020 Zooming Innovation in Consumer Technologies Conference, ZINC 2020*, 165–168. <https://doi.org/10.1109/ZINC50678.2020.9161446>
- Rahmad, C., Rahmah, I. F., & Asmara, R. A. (2017). Deteksi dan Pengenalan Rambu Lalu Lintas di Indonesia Menggunakan RGBN dan Gabor. *SEINTRINOV*, 13–22.
- Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. *IEEE Conference on Computer Vision and Pattern Recognition*, 7263–7271. <http://pjreddie.com/yolo9000/>
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6), 1–9. <https://github.com/>
- Sitanggang, O. R., Fitriyah, H., & Utaminingrum, F. (2018). Sistem Deteksi dan Pengenalan Jenis Rambu Lalu Lintas Menggunakan Metode Shape Detection Pada Raspberry Pi. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(12), 6108–6117. <http://j-ptiik.ub.ac.id>
- Sonntag, D., Barz, M., Zacharias, J., Stauden, S., Rahmani, V., Fóthi, Á., & Lőrincz, A. (2017). Fine-tuning deep cnn models on specific ms coco categories. *arXiv preprint arXiv:1709.01476*.
- Triyadi, A., & Utaminingrum, F. (2020). Pengembangan Sistem Rekognisi Rambu Kecepatan Menggunakan Circle Hough Transform dan Convolutional Neural Network Berbasis Raspberry Pi. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(1), 56–64. <http://j-ptiik.ub.ac.id>
- Wang, Q., Zhang, Q., Liang, X., Wang, Y., Zhou, C., & Mikulovich, V. I. (2022). Traffic lights detection and recognition method based on the improved yolov4 algorithm. *Sensors*, 22(1). <https://doi.org/10.3390/s22010200>
- Yu, Y., Sun, Y., Zhao, C., & Qu, C. (2021). Research on Defect Detection of Electric Energy Metering Box Based on YOLOv5. *Journal of Physics: Conference Series*, 2087(1). <https://doi.org/10.1088/1742-6596/2087/1/012081>

#### UCAPAN TERIMA KASIH

Ucapan terima kasih kami sampaikan kepada Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi karena telah mendanai penelitian ini dalam skema Penelitian Dosen Pemula (PDP) tahun 2022.