

Akurasi Pengujian Model Hasil *Training* menggunakan YOLOv4 untuk Pengenalan Kendaraan di Jalan Raya

Ahmad Fali Oklilas*¹, Sukemi², Dinda Dwinta³, Ghinadhia Shofi⁴, Nanda Putri Mariza⁵,
Sri Arum Kinanti⁶, Yulia Amanda Sari⁷

^{1,2,3,4,5,6,7}Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya; Jalan Palembang – Prabumulih Km.32 Inderalaya Ogan Ilir Kode Pos 30662, (0711)379249, (0711)581700

¹Laboratium Elektronika dan Sistem Digital Fakultas Ilmu Komputer, Universitas Sriwijaya, Palembang

e-mail: *fali@ilkom.unsri.ac.id, sukemi@ilkom.unsri.ac.id, dindadwinta09@gmail.com,
ggshofi@gmail.com, nandapmariza@gmail.com, sriarumk02@gmail.com,
yuliaamanda358@gmail.com

Abstrak

Kemacetan lalu lintas saat ini merupakan masalah utama yang terjadi di kota kota besar di Indonesia. Analisis arus lalu lintas merupakan dasar penting untuk perencanaan kota. Pengelolaan Intelligent Transportation System (ITS) telah menjadi kebutuhan saat ini untuk mengelola masalah lalu lintas yang padat. Sistem transportasi cerdas menggunakan teknik computer vision semakin menarik perhatian untuk deteksi kepadatan lalu lintas. Dalam penelitian ini menggunakan metode deteksi objek You Only Look Once (YOLO) versi 4 untuk klasifikasi dan deteksi kendaraan agar mendapatkan sebuah model yang optimal. Pengujian Model YOLOv4 mendapatkan hasil mean average precision (mAP) sebesar 80,12%. Dalam pengujian video untuk medeteksi motor dan mobil mendapatkan total akurasi kendaraan sebesar 70,6% dan pada tingkat keyakinan kendaraan sebesar 78,7%.

Kata kunci—YOLOv4, pengenalan kendaraan, pengujian model, jalan raya

Abstract

Traffic congestion is currently the main problem that occurs in big cities in Indonesia. Traffic flow analysis is an important basis for urban planning. Management of Intelligent Transportation System (ITS) has become a necessity today to manage heavy traffic problems. Intelligent transportation systems using computer vision techniques are increasingly attracting attention for traffic density detection. This research uses the You Only Look Once (YOLO) version 4 object detection method for vehicle classification and detection to obtain an optimal model. Testing the YOLOv4 model results in a mean average precision (mAP) of 80.12%. In video testing to detect motorcycles and cars, the total vehicle accuracy is 70.6% and the vehicle confidence level is 78.7%.

Keywords—YOLOV4, vehicle recognition, model testing, highway

1. PENDAHULUAN

Kemacetan lalu lintas saat ini merupakan problem utama yang terjadi di kota-kota besar di Indonesia. Penyebab utama terjadinya kemacetan lalu lintas adalah karena tidak seimbangnya pertumbuhan jumlah kendaraan dengan kapasitas prasarana transportasi [1]. Perkembangan teknologi sangatlah pesat sehingga banyak aplikasi yang muncul guna

mempermudah kehidupan masyarakat. Dengan munculnya *Internet of Things* (IoT) yang telah menerapkan *machine learning*, membuat aplikasi menjadi lebih pintar [2]. Permasalahan kemacetan jalan raya tidak hanya dapat diselesaikan dengan meningkatkan kualitas dan kuantitas infrastruktur, namun juga manajemen lalu lintas [3].

Analisis arus lalu lintas merupakan dasar penting untuk perencanaan kota dan pengelolaan *Intelligent Transportation System* telah menjadi kebutuhan saat ini untuk mengelola masalah lalu lintas yang padat karena pertumbuhan transportasi jalan yang eksponensial [4]. Lalu lintas menggunakan teknik *computer vision* semakin menarik perhatian untuk pengembangan sistem transportasi cerdas. Maka menghitung volume lalu lintas berdasarkan sistem CCTV adalah salah satu aplikasi utama. Dengan memantau arus lalu lintas dari CCTV, dapat mengevaluasi dan memverifikasi kinerja sistem [5]. Sebagian kota-kota besar telah menerapkan CCTV untuk pemantauan arus lalu lintas dengan mengekstraksi informasi dari gambar seperti kecepatan, komposisi lalu lintas, kemacetan lalu lintas, bentuk kendaraan, jenis kendaraan, nomor identifikasi kendaraan, dan terjadinya pelanggaran lalu lintas atau kecelakaan di jalan [6].

Untuk mengatasi hilangnya fitur yang disebabkan oleh kompresi gambar atau video resolusi tinggi selama tahap normalisasi, dengan kemampuan pendeteksian yang cepat dan akurat hingga dua kali lipat dibandingkan beberapa metode lain yaitu menggunakan algoritma deteksi objek *You Only Look Once* (YOLO) yang disarankan untuk pra-pemrosesan data dan tahap deteksi [6][7]. YOLO ini sendiri bertujuan untuk klasifikasi dan deteksi kendaraan yang akan dianalisa performansinya sebagai pendeteksi kendaraan supaya untuk mendapatkan model yang optimal pada saat mengimplementasi sistem pengendalian di lalu lintas [8].

Dengan hasil model *training* YOLO bisa dilakukan mendapatkan kondisi jalan menggunakan berbagai metode yaitu *Artificial Neural Network* (ANN), *Decision Tree*, *K-Nearest Neighbor* (KNN), dan *Random Forest*. Metode Artificial neural network merupakan sebuah metode yang mengadopsi keterkaitan antar neuron pada jaringan syaraf manusia. Artificial neural network dikembangkan untuk mencoba membuat suatu sistem yang dapat melakukan proses pembelajaran sendiri layaknya manusia tanpa harus setiap saat mengubah kondisi masukan atau sistem dan memiliki kecenderungan semakin pandai dalam melakukan keputusan selanjutnya. Artificial neural network terdiri atas neuron-neuron yang saling terhubung satu sama lain [9].

Metode *Decision Tree* atau yang disebut dengan pohon keputusan merupakan model dari klasifikasi dengan bentuknya yang seperti struktur pohon merepresentasikan atribut setiap data yang diproses. Istilah *Decision Tree* yaitu mendeskripsikan setiap kelas untuk menemukan pola atau fungsi dengan tujuan untuk melakukan klasifikasi atau prediksi data yang belum mempunyai kelas. *Decision tree* adalah algoritma *Supervised Machine Learning* yang digunakan untuk memecahkan masalah klasifikasi. Tujuan utama mampu menghasilkan model prediksi secara spesifik dalam bentuk aturan yang mudah untuk diimplementasikan [10]. *Decision tree* menggunakan teknik “membagi dan menaklukkan” untuk membagi ruang pencarian masalah menjadi himpunan masalah. Proses adalah mengubah bentuk data tabel menjadi sebuah model *tree* lalu akan menghasilkan rule dan disederhanakan [11].

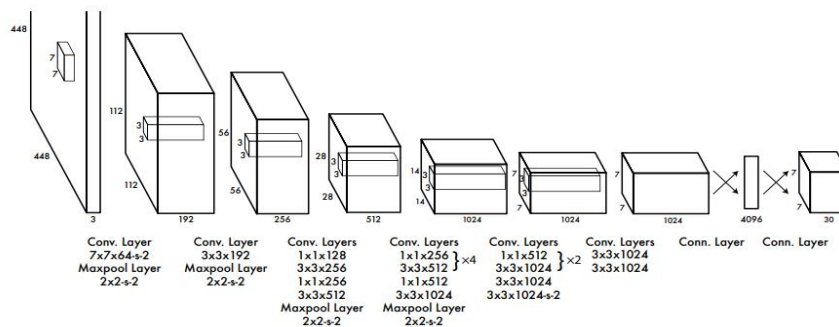
Algoritma KNN adalah algoritma *supervised machine learning* yang sederhana dan mudah diterapkan, dapat digunakan untuk menyelesaikan keduanya masalah klasifikasi dan regresi. Strategi KNN lebih kuat daripada model deret waktu parametrik karena kecenderungannya yang tidak memerlukan pemahaman tentang gagasan hubungan antara fitur dan hasil. Metode KNN dapat mempertimbangkan beberapa jumlah pengamatan (juga disebut sebagai tetangga terdekat) dari *dataset* historis dan kemudian memprediksi variabel masa depan berdasarkan himpunan tetangga terdekat [12]. Untuk metode *Random Forest* adalah sebuah algoritma *machine learning* tipe *ensemble* pohon berstruktur untuk klasifikasi dan regresi yang pertama kali diusulkan oleh Breiman pada tahun 2001.

Karena struktur yang sederhana dan memiliki kinerja luar biasa di bandingkan algoritma *machine learning* lainnya, sehingga banyak digunakan di berbagai bidang. Algoritma ini merupakan kombinasi pohon keputusan dengan dua program keputusan dengan dua program

prosedur pemilihan acak yang dapat mengurangi variasi dan meninggalkan bias secara tetap. Terdapat tiga aspek utama dalam menjalankan algoritma random forest diantaranya adalah melakukan *bootstrap* sampling untuk membangun pohon prediksi, setiap dengan mengombinasikan hasil dari setiap pohon keputusan dengan cara *majority vote* untuk melakukan klasifikasi [13].

2. METODE PENELITIAN

Metode yang digunakan penulis dalam melakukan penelitian ini adalah metode *You Only Look Once* (YOLO) versi 4. YOLOv4 adalah sebuah metode untuk mendeteksi objek. YOLO memproses gambar secara real-time pada empat puluh lima (45) *frames* per second. YOLO menggunakan pendekatan jaringan syaraf tunggal (*Single neural network*) untuk melakukan pendeteksian objek pada sebuah citra. Jaringan ini menggunakan fitur dari semua gambar untuk memprediksi setiap *bounding box* yang dapat melakukan prediksi pada kotak-kotak pembatas dan probabilitas secara langsung dalam satu evaluasi [14][15]



Gambar 1 Arsitektur YOLO

Jaringan deteksi YOLO memiliki 24 lapisan konvolusi (*convolutional layer*) yang diikuti oleh 2 lapisan yang terhubung penuh (*fully connected layer*). Beberapa lapisan konvolusi menggunakan lapisan reduksi 1x1 sebagai alternatif dalam mengurangi kedalaman feature maps yang diikuti oleh 3x3 lapisan konvolusional (*convolutional layer*) [14]. Metode YOLO terdiri dari beberapa tahap seperti yang ditunjukkan pada gambar 2, metode ini berfokus untuk mengetahui tingkat keakuratan sebuah objek secara *real time*.

Tahap awal dalam penelitian ini adalah pengumpulan dataset, pengumpulan dataset ini dianggap sebagai poin pertama dalam perancangan yang berfokus pada sebuah objek. Dalam pelaksanaan tahap awal ini penulis melakukan pengumpulan data dengan cara foto lapangan dengan fokus pada gambar mobil dan gambar motor. Kemudian penulis melakukan observasi keberbagai ruas jalan yang ada di kota Palembang. Data yang digunakan pada penelitian ini terdiri dari dua objek kendaraan yaitu gambar mobil dan gambar motor yang diambil secara manual, dengan total dataset yang ada dalam penelitian ini berjumlah 4.224 dataset file gambar.

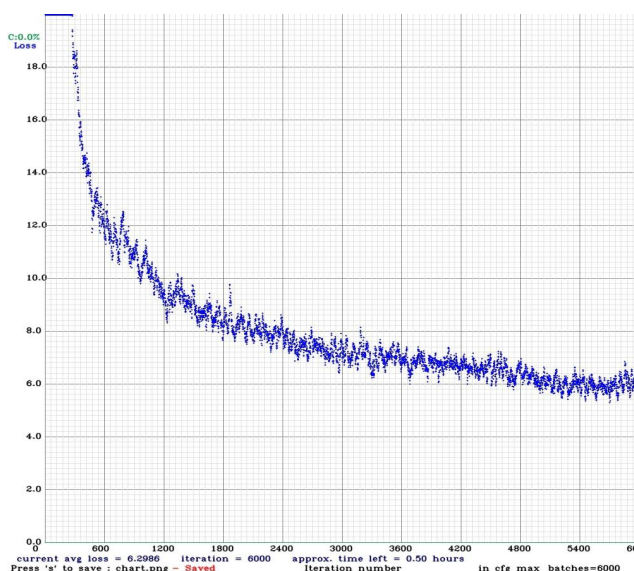
Tahap selanjutnya perancangan *preprocessing*, *preprocessing* merupakan proses mengklasifikasi objek gambar secara manual dengan *software LabelImg*. User bisa mengklasifikasi semua objek dan menyimpan hasil gambar beserta .txt ke folder dalam data sebelum masuk ke program YOLOv4. Terdapat beberapa proses pada preprocessing yaitu:

1. Proses pertama yang dilakukan yaitu proses *cleaning* dan *resize* gambar.
2. Pelabelan data citra menggunakan software *labelImg* dengan kelas 0 untuk sepeda motor dan 1 untuk mobil.
3. Data citra yang sudah dilabel kemudian disimpan dengan format .txt dan digabungkan dengan data yang sudah dilabel lainnya ke dalam satu folder.

4. Proses selanjutnya yaitu membagi data menjadi dua yaitu data *training* 80% dan *testing* 20%.

Selanjutnya *configuration software*, *configuration software* adalah proses yang dilakukan untuk memaksimalkan aktivitas dalam suatu aplikasi. Dengan dilakukan konfigurasi, sebuah aplikasi atau program dapat diatur untuk memenuhi kebutuhan pengguna. *Software* yang dilakukan *configuration* untuk penelitian ini ialah OpenCV, openCV Contrib, CUDA, cuDNN. Dalam proses konfigurasi dilakukan menggunakan aplikasi *software* CMake dan dilanjutkan dengan Visual Studio 2022. Pada tahap *configuration software* proses yang digunakan dalam konfigurasi yaitu proses *training* dengan menggunakan YOLOv4.

Selanjutnya tahap *training data*, *training data* yaitu proses yang bertujuan untuk melatih komputer dengan cara mengolah gambar dan anotasi yang sudah dibuat sehingga terbentuk pola atau karakteristik dari setiap kelas yang akan menjadi bahan pertimbangan komputer dalam mencapai sebuah keputusan atau prediksi. Proses *training data* yang dilakukan maka akan menghasilkan sebuah *chart* yang merupakan hasil dari *training data* tersebut.



Gambar 3 Hasil chart *training data*

Kemudian tahap pengujian model, pengujian model merupakan tahap akhir untuk mendapatkan hasil *training*. Setelah dilakukan proses *training*, maka akan mendapatkan hasil model YOLOv4 berbentuk file *weights*. Hasil tersebut akan yang nantinya digunakan untuk membedakan pendeteksian objek motor dan mobil.

Tahap terakhir ialah hasil training, setelah melakukan proses *training* dengan menggunakan arsitektur YOLOv4 dan telah melakukan tahap pengujian model. Tahap selanjutnya adalah melakukan hasil *training* uji coba pendeteksian terhadap objek motor dan mobil. Objek gambar motor dan mobil akan di tandai dengan sebuah kotak dengan nilai akurasi. Pedekteksian objek dilakukan dengan mengambil video yang ada diberbagai ruas jalan. Hasil deteksi video akan terbagi menjadi dua yaitu perhitungan masing-masing untuk kendaraan motor dan mobil. Adanya perhitungan akurasi dan persentase tingkat keyakinan yang didapatkan dari hasil *training*. Pengujian model dilakukan bertujuan untuk mengetahui tingkat keakurasian pada YOLOv4. Dalam melakukan pengujian model, dibutuhkan dataset testing yang telah diberi anotasi. Dataset testing merupakan dataset yang belum pernah digunakan sebagai dataset training. Dataset yang valid digunakan pada penelitian ini sebesar 20% dari jumlah dataset yang digunakan.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini dilakukan pengujian model berupa proses *training* dengan menggunakan YOLOv4. Dimana pengujian model ini akan menghasilkan suatu model yang nantinya model tersebut akan digunakan pada saat pengenalan jenis kendaraan motor dan mobil. Setelah semua proses *training* data selesai maka didapatkan hasil model YOLOv4. Berikut gambar hasil model YOLOv4.

```
detections_count = 171028, unique_truth_count = 48584
class_id = 0, name = Motor, ap = 75.57% (TP = 14441, FP = 7150)
class_id = 1, name = Mobil, ap = 84.68% (TP = 22182, FP = 5915)

for conf_thresh = 0.25, precision = 0.74, recall = 0.75, F1-score = 0.75
for conf_thresh = 0.25, TP = 36623, FP = 13065, FN = 11961, average IoU = 57.84 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.801220, or 80.12 %
Total Detection Time: 612 Seconds

Set -points flag:
`-points 101` for MS COCO
`-points 11` for PascalVOC 2007 (uncomment `difficult` in voc.data)
`-points 0` (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset
```

Gambar 4 Hasil Pengujian Model Dengan YOLOv4

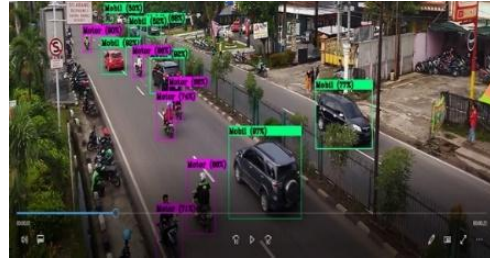
Berdasarkan hasil model YOLOv4 tersebut dapat diketahui bahwa pada kelas motor jumlah *true positive* sebesar 14441 lebih besar dibandingkan jumlah *false positive* sebesar 7150 dengan nilai AP untuk motor sebesar 75,57%. Pada kelas mobil jumlah *true positive* sebesar 22182 lebih besar dibandingkan dengan jumlah *false positive* sebesar 5915 dengan nilai AP sebesar 84,68%. Dari hasil model YOLOv4 tersebut juga diketahui hasil *mean average precision*(mAP) sebesar 0,801220 atau sebesar 80,12%. Terdapat total waktu deteksi sebesar 612 detik. Sedangkan *recall* dalam menemukan semua objek positif mencapai nilai 0,75 sementara untuk *F1-score* mencapai 0,75. Berikut tabel hasil pengujian model.

Tabel 1 Hasil Pengujian Model

Load Model		yolov4-obj_last.weights
Motor	AP	75,57%
	TP	14441
	FP	7150
Mobil	AP	84,68%
	TP	22182
	FP	5915
FN		11961
Total Waktu Deteksi		612 Detik
Recall		0,75
Precision		0,74
F1-Score		0,75
IoU		57,84%
mAP@0.50		80,12%



Gambar 4 Pengujian 1



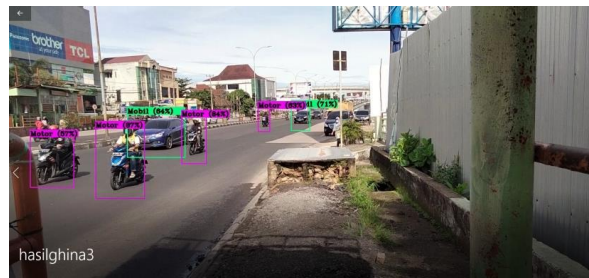
Gambar 5 Pengujian 2



Gambar 6 Pengujian 3



Gambar 7 Pengujian 4



Gambar 8 Pengujian 5

Tabel 2 Hasil Akurasi Dan Tingkat Keyakinan Motor dan Mobil

NO	Motor		Mobil	
	Akurasi	Tingkat Keyakinan	Akurasi	Tingkat Keyakinan
Pengujian 1	75 %	76 %	80 %	84 %
Pengujian 2	38 %	75 %	57 %	69 %
Pengujian 3	100 %	79 %	88 %	75 %
Pengujian 4	68 %	73 %	83 %	90 %
Pengujian 5	67 %	80 %	50 %	86 %
Rata-Rata Akurasi dan Tingkat Keyakinan	69,6 %	76,6 %	71,6 %	80,8 %

Dari tabel diatas dapat diketahui bahwa hasil rata-rata akurasi untuk motor sebesar 69,6 % dan rata-rata tingkat keyakinan sebesar 76,6 %. Sedangkan untuk mobil hasil rata-rata akurasi sebesar 71,6 % dan rata-rata tingkat keyakinan sebesar 80,8 %.

4. KESIMPULAN

Kesimpulan dari hasil pendeteksian kendaraan motor dan mobil dengan menggunakan YOLO terutama pada penelitian ini menggunakan YOLOv4 dinilai dapat bekerja dengan baik. Dimana setelah dilakukan deteksi dengan menggunakan video sistem telah dapat mengenal objek dalam video tersebut yang ditandai dengan adanya *frame - frame* berwarna dengan menghasilkan nilai akurasi yang berbeda-beda setiap *frame* nya selama video tersebut terus berjalan.

Untuk hasil rata-rata akurasi dan tingkat keyakinan yang dihasilkan dari setiap objek yang telah di deteksi hampir semuanya menunjukkan rata-rata diatas 50%. Dengan total akurasi kendaraan motor dan mobil sebesar 70,6% dan pada tingkat keyakinan kendaraan motor dan mobil sebesar 78,7%.

5. SARAN

Dalam penelitian yang telah dilakukan oleh peneliti, terdapat saran yang dapat ditambahkan agar selanjutnya mendapatkan hasil yang lebih baik. Dapat menambahkan perhitungan otomatis pada YOLO untuk mendapatkan jumlah kendaraan yang terdeteksi. Menggunakan algoritma deteksi objek lainnya seperti YOLOv5, YOLOv6, YOLOv7 dan penambahan metode seperti *machine* atau *deep learning* untuk peningkatan akurasi yang lebih akurat.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Laboratorium Elektronika Sistem Digital dan semua pihak yang telah memberi dukungan dan membantu terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] R. Suyuti, "Implementasi "Intelligent Transportation System (Its)" Untuk Mengatasi Kemacetan Lalu Lintas Di Dki Jakarta," *Konstruksia*, vol. 3, pp. 13–21, 2012.
- [2] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, "A review of machine learning and IoT in smart transportation," *Futur. Internet*, vol. 11, no. 4, 2019, doi: 10.3390/FI11040094.
- [3] E. Harahap, A. Suryadi, R. Ridwan, D. Darmawan, and R. Ceha, "Efektifitas Load Balancing Dalam Mengatasi Kemacetan Lalu Lintas," *Matematika*, vol. 16, no. 2, 2017, doi: 10.29313/jmtm.v16i2.3665.
- [4] P. B. Kumar and K. Hariharan, "Time Series Traffic Flow Prediction with Hyper-Parameter Optimized ARIMA Models for Intelligent Transportation System," *J. Sci. Ind. Res. (India)*, vol. 81, no. 4, pp. 408–415, 2022, doi: 10.56042/jsir.v81i04.50791.
- [5] K. H. Nam Bui, H. Yi, and J. Cho, "A multi-class multi-movement vehicle counting framework for traffic analysis in complex areas using CCTV systems," *Energies*, vol. 13, no. 8, 2020, doi: 10.3390/en13082036.
- [6] M. Harahap et al., "Sistem Cerdas Pemantauan Arus Lalu Lintas Dengan YOLO (You Only Look Once v3)," *Semin. Nas. APTIKOM*, p. 2019, 2019.
- [7] Z. Chen, L. Cao, and Q. Wang, "YOLOv5-Based Vehicle Detection Method for High-Resolution UAV Images," *Mob. Inf. Syst.*, vol. 2022, 2022, doi: 10.1155/2022/1828848.
- [8] N. N. Hasibuan, M. Zarlis, and S. Efendi, "Detection and tracking different type of cars with YOLO model combination and deep sort algorithm based on computer vision of traffic controlling," *J. dan Penelit. Tek. Inform.*, vol. 6, no. 1, pp. 210–220, 2021,

- [Online]. Available: <https://doi.org/10.33395/sinkron.v6i1.11231>.
- [9] Y. R. Rifa'i, S. Sumaryo, and ..., "Pemodelan Dan Simulasi Kontrol Adaptif Lampu Lalu Lintas Menggunakan Algoritma Jaringan Syaraf Tiruan," *eProceedings ...*, vol. 5, no. 3, pp. 3997–4004, 2010, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/8130>.
- [10] A. Franseda, W. Kurniawan, S. Anggraeni, and W. Gata, "Integrasi Metode Decision Tree dan SMOTE untuk Klasifikasi Data Kecelakaan Lalu Lintas," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 3, p. 282, 2020, doi: 10.26418/justin.v8i3.40982.
- [11] S. Bahri and A. Lubis, "Metode Klasifikasi Decision Tree Untuk Memprediksi Juara English Premier League," *J. Sintaksis*, vol. 2, no. 1, pp. 63–70, 2020.
- [12] F. I. Rahman, "Short Term Traffic Flow Prediction Using Machine Learning - Knn, Svm and Ann With Weather Information," *Int. J. Traffic Transp. Eng.*, vol. 10, no. 3, pp. 371–389, 2020, doi: 10.7708/ijtte.2020.10(3).08.
- [13] P. Dewi, P. Purwono, and S. Dwi, "Pemanfaatan Teknologi Machine Learning pada Klasifikasi Jenis Hipertensi Berdasarkan Fitur Pribadi," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 11, no. 3, pp. 377–387, 2022, doi: 10.30591/smartcomp.v11i3.3721.
- [14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
- [15] K. Khairunnas, E. M. Yuniarno, and A. Zaini, "Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot," *J. Tek. ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373539.v10i1.61622.