

# Perbandingan Metode Haar Cascade, YoloV3, dan TinyYoloV3 Dalam Mendeteksi Kendaraan Bermotor Berbasis Video

Marzuarman<sup>1)</sup>, Stephan<sup>2)</sup>, Muharnis<sup>3)</sup>, Azizul<sup>4)</sup>, Doni Mirza Rinaldi<sup>5)</sup>, Bagas Prasetyo<sup>6)</sup>  
<sup>1,2,3,4,5,6</sup> Politeknik Negeri Bengkalis, Bengkalis, Indonesia

E-mail: <sup>1)</sup> marzuarman@polbeng.ac.id

**Abstract:** Motorized vehicles are one of the most important needs in everyday life. Every year in Indonesia there is always an increase in the number of motorized vehicles along with the increase in population. Many researchers in the field of information technology use image processing systems to investigate and develop systems that can be used on public roads, one of which is to detect motor vehicles. In general, the methods that are often used to detect objects are Haar cascade, YOLOV3, and TinyYOLOV3. In this study, a comparison was made, to determine the best accuracy of the three methods in detecting motorized vehicles. The test was carried out using Python 3.10 software that has been installed with OpenCV, where the test was carried out using a video with a duration of 1 minute 23 seconds which was downloaded from the youtube.com site. Based on the test results, the YOLOV3 method gets the best level of accuracy, which is 74%. For the Haar cascade method, the accuracy value is 41%, and TinyYOLOV3 produces an accuracy rate of 25%.

**Keywords:** Haar cascade, YOLOV3, TinyYOLOV3

**Abstrak:** Kendaraan bermotor termasuk kedalam kebutuhan yang sangat penting dalam kehidupan sehari-hari. Setiap tahun di Indonesia selalu terjadi peningkatan jumlah kendaraan bermotor seiring dengan meningkatnya jumlah penduduk. Banyak peneliti dibidang teknologi informasi memanfaatkan sistem *image processing* untuk menyelidiki dan mengembangkan sistem yang bisa digunakan pada jalan umum, salah satunya untuk mendeteksi kendaraan bermotor. Pada umumnya metode yang sering digunakan untuk mendeteksi objek adalah Haar cascade, YOLOV3, dan TinyYOLOV3. Pada penelitian ini dilakukan perbandingan, untuk mengetahui akurasi terbaik dari ketiga metode tersebut dalam mendeteksi kendaraan bermotor. Pengujian dilakukan dengan menggunakan *software* Python 3.10 yang sudah terpasang OpenCV, dimana pengujian dilakukan menggunakan video dengan durasi 1 menit 23 detik yang di download dari situs youtube.com. Berdasarkan hasil pengujian menunjukkan metode YOLOV3 mendapatkan tingkat akurasi yang paling baik yaitu sebesar 74%. untuk metode Haar cascade mendapatkan nilai akurasi sebesar 41%, dan TinyYOLOV3 menghasilkan tingkat akurasi sebesar 25%.

**Kata kunci:** Haar cascade, YOLOV3, TinyYOLOV3

## 1. Pendahuluan

Pada perkembangan zaman yang sangat modern seperti pada saat ini, kendaraan bermotor termasuk kedalam kebutuhan yang sangat penting dalam kehidupan sehari-hari [1]. Di Indonesia rata-rata setiap keluarga memiliki kendaraan bermotor, khususnya sepeda motor. Hal ini yang menyebabkan selalu meningkatnya jumlah kendaraan bermotor di Indonesia seiring dengan meningkatnya jumlah penduduk [2]. Beberapa peneliti dibidang teknologi informasi memanfaatkan momen tersebut untuk menyelidiki dan mengembangkan sistem yang bisa digunakan pada jalan umum, seperti sistem perhitungan jumlah kendaraan yang melewati suatu jalan, perhitungan kecepatan kendaraan yang lewat, mengklasifikasi jenis kendaraan, dan lain sebagainya [3]. Dimana seluruh penelitian itu banyak menggunakan *image processing* untuk mengolah seluruh data kendaraan bermotor tersebut [4]. Sebelum menghitung kecepatan dan mengklasifikasi kendaraan perlu adanya proses deteksi kendaraan, sehingga bisa membedakan kendaraan bermotor dengan objek lain [5]–[7].

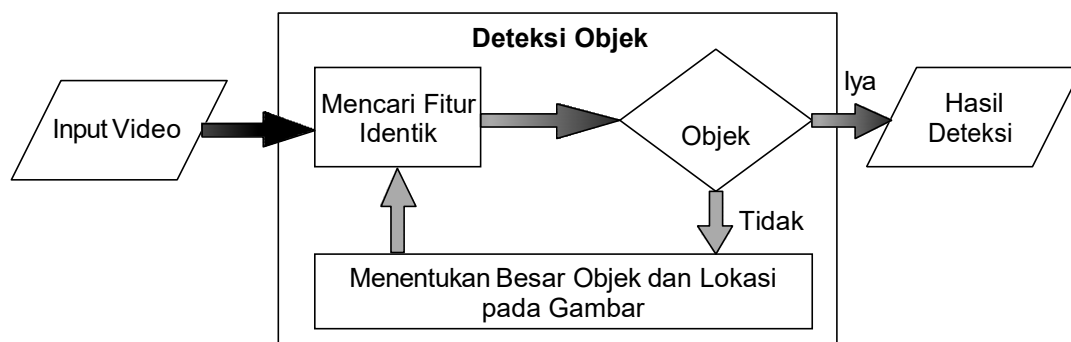
Banyak metode-metode dalam *image processing* untuk mendeteksi kendaraan bermotor. Beberapa diantaranya menggunakan metode *background subtraction*, haar cascade, You Only Look Once versi 3 (YOLO V3) beserta versi selanjutnya, YOLO V3 Tiny, dan masih banyak lagi metode lainnya [8]. Biasanya metode-metode ini diimplementasikan untuk membangun sistem yang dimanfaatkan di jalan umum. Pada penelitian ini dilakukan perbandingan metode haar cascade, YOLOV3, dan TinyYoloV3 untuk membedakan tingkat akurasi metode tersebut serta melihat kelebihan dan kekurangan metode tersebut dalam mendeteksi kendaraan bermotor di jalan raya. Karena ketiga metode tersebut sering digunakan dan memiliki akurasi yang sangat tinggi sehingga bisa diaplikasikan pada sistem pengukuran kecepatan kendaraan, sistem perhitungan volume kendaraan, dan

klasifikasi kendaraan. Metode-metode tersebut juga merupakan perkembangan metode terbaru di era modern seperti saat ini [8].

Pada penelitian ini dilakukan pengujian dengan menggunakan *file* video CCTV di jalan raya dan dihitung jumlah kendaraan yang terdeteksi oleh metode Haar cascade, YOLOV3, dan TinyYoloV3 dibandingkan dengan metode deteksi secara manual. Hasil perbandingan tingkat akurasi dipersentasekan, kemudian dijabarkan masing-masing kelebihan dan kekurangan metode tersebut sehingga bisa jadi referensi penelitian selanjutnya serta bisa digunakan metode mana yang terbaik dalam mendeteksi kendaraan untuk membangun sistem pengukuran kecepatan kendaraan di jalan, perhitungan volume kendaraan, klasifikasi jenis kendaraan, dan lain sebagainya.

## 2. Metode

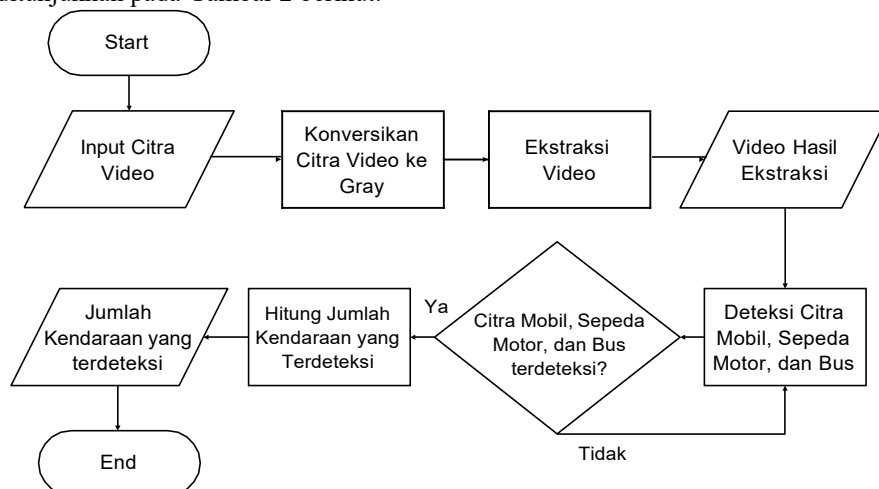
Pada sistem pendeteksian objek menggunakan metode Haar cascade, YOLOV3, dan TinyYoloV3 dapat diceritakan prosesnya melalui blok diagram seperti ditunjukkan pada Gambar 1. Dimana objek yang akan di deteksi bersumber dari video. Objek yang dideteksi didalam video berupa kendaraan bermotor dan objek lain yang terdeteksi oleh metode-metode tersebut. Pemrograman deteksi objek dibuat menggunakan *software* Python 3.10 yang sudah terinstall *library* opencv-python versi 4.1.2.30.



Gambar 1. Blok diagram deteksi objek.

### 2.1. Metode Haar Cascade

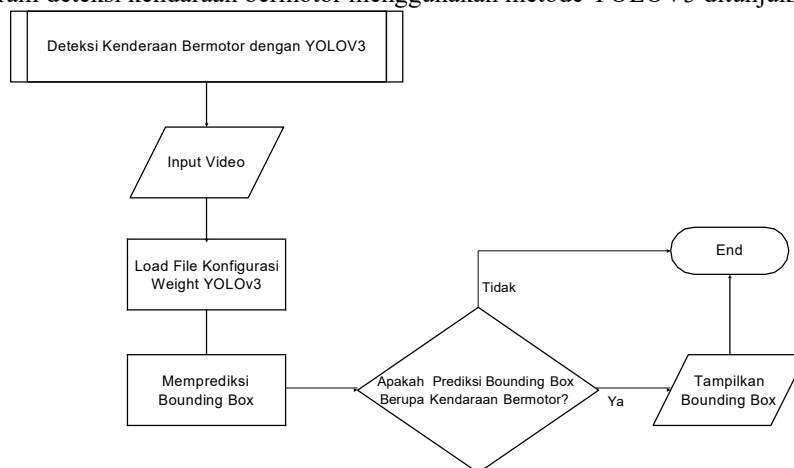
Metode Haar cascade adalah metode deteksi yang ditemukan oleh Paul Viola dan Michael Jones pada tahun 2001. Prinsipnya adalah terdiri dari beberapa kumpulan fungsi Haar-Like yang digabungkan untuk membentuk klasifikasi yang berprinsip pada jumlah nilai piksel putih dikurangkan dengan nilai piksel hitam, atau dengan kata lain metode Haar cascade merupakan gabungan piksel hitam dan piksel putih yang berbentuk kotak [9]. Pada sistem Haar-Like memproses gambar didalam kotak, yang mana ada beberapa piksel berada dalam satu bingkai [5]. Untuk proses pada setiap kotak menghasilkan nilai yang berbeda-beda sebagai penanda area gelap dan terang. Lalu nilai tersebut digunakan untuk proses pengolahan citra. Adapun *flowchart* metode Haar cascade pada penelitian ini ditunjukkan pada Gambar 2 berikut.



Gambar 2. Flowchart sistem deteksi kendaraan dengan metode haar cascade.

## 2.2. Metode You Only Look Once Versi 3 (TinyYOLOV3)

Algoritma YOLO ini adalah merupakan *Real Object Deteciton* yang baru-baru ini sangat populer untuk dikembangkan. YOLO merupakan keluarga *Convolutional Neural Networks* (CNN) yang mencapai hasil optimal dengan model ujung-ke-ujung tunggal yang dapat melakukan deteksi objek secara *real time*. YOLOV3 merupakan salah satu *object detector* dengan tingkat akurasi yang relatif baik dengan menggunakan metode *anchor-based detection* dimana metode ini menggunakan kotak jangkar (*anchor box*) [8]. Pada penelitian ini metode YOLOV3 berfungsi untuk mendeteksi kendaraan bermotor pada video dan mendapatkan posisi dari kendaraan tersebut. Adapun blok diagram deteksi kendaraan bermotor menggunakan metode YOLOV3 ditunjukkan pada Gambar 3.



**Gambar 3.** Flowchart sistem deteksi kendaraan dengan metode YOLOV3.

Data *weight* yang digunakan diperoleh dari <https://pjreddie.com/darknet/yolo/>. Data tersebut sangat baik digunakan untuk deteksi objek dengan tingkat akurasi yang cukup tinggi. Sistem dari metode YOLO akan menghasilkan nilai koordinat *bounding box* dan nilai tingkat keyakinannya. Setelah itu sistem ini akan menampilkan objek yang terdeteksi dengan menampilkan *bounding box*. Data train pada sistem ini berasal dari *file COCO trainval* yang memiliki *mean average precision* yang lumayan baik.

## 2.3. Metode Tiny You Only Look Once Versi 3 (TinyYOLOV3)

TinyYoloV3 merupakan pengembangan dari YOLOV3 dalam versi yang lebih ringan dan cepat. Metode TinyYOLOV3 di train menggunakan kumpulan data dalam *file COCO trainval* yang mampu mendeteksi hingga 80 kelas. Jumlah piksel citra *input* pada TinyYoloV3 yaitu berdimensi 832x480 piksel dan nilai *stride* yaitu sebesar 32, maka jumlah *grid cell* yang dihasilkan adalah 26x15. *Grid cell* merupakan garis berwarna hitam pada citra setelah citra dibagi ke dalam beberapa *grid cell*, lalu tahap selanjutnya adalah TinyYoloV3 memprediksi *anchor box* sebanyak B (nilai B dapat di deklarasikan) pada setiap *grid cell* [8]. Selanjutnya, dari *anchor box* yang sudah ada TinyYoloV3 akan memprediksi *bounding box* pada setiap objek. Dari *bounding box* yang sudah ada, TinyYoloV3 akan memilih kotak pembatas mana yang paling sesuai dalam memprediksi objek kendaraan. Untuk *flowchat* sistem deteksi kendaraan TinyYoloV3 sama untuk versi YOLOV3 seperti ditunjukkan pada Gambar 3.

## 3. Hasil Dan Pembahasan

Hasil yang telah dilakukan menjalankan aplikasi yang telah dibuat menggunakan *software* Python 3.10. Pada program deteksi kendaraan menggunakan metode Haar cascade, *file path* dalam bentuk format .xml untuk deteksi mobil, sepeda motor, dan bus. Masing-masing *path* kendaraan tersebut berada pada *file* terpisah dan di *download* pada *link* <https://github.com/AdityaPai2398/Vehicle-And-Pedestrian-Detection-Using-Haar-Cascades>. Sedangkan pada YOLOV3 dan TinyYOLOV3 *file weight*, *file* konfigurasi dan *file COCO* di *downlaod* pada *link* <https://pjreddie.com/darknet/yolo/>. Sebelum melakukan pemrograman, dilakukan instalasi *library* opencv pada Python melalui pip dengan menggunakan cmd. Adapun perintah instalasi opencv pada cmd adalah “pip install opencv-python”. Setelah menginstalasi opencv selanjutnya adalah instalasi *library* numpy dengan menggunakan perintah “pip install numpy”.

```

mobil_cascade = cv2.CascadeClassifier('mobil.xml') #Path to cars.xml
motor_cascade = cv2.CascadeClassifier('motor.xml') #Path to motor.xml
bus_cascade = cv2.CascadeClassifier('bus.xml') #Path to motor.xml

```

**Gambar 4.** Deklarasi variabel *path* deteksi kendaraan menggunakan haar cascade

Untuk memulai pemrograman dengan metode Haar cascade dengan memanggil *file path* dalam format .xml dan dideklarasikan menjadi variabel seperti terlihat pada Gambar 4. Video pengujian diambil dari website youtube.com dengan durasi 1 menit 23 detik. Adapun hasil tampilan video pada menit 0 dan detik ke-21 ditunjukkan pada Gambar 5.



**Gambar 5.** Tampilan video pengujian pada waktu 0 menit 21 detik

Adapun hasil pengujian deteksi kendaraan menggunakan metode Haar cascade ditunjukkan pada Gambar 6, dimana sampel video yang diambil pada waktu 0 menit 21 detik. Dari hasil pengujian tersebut menunjukkan jumlah kendaraan yang terdeteksi adalah sebanyak 4 kendaraan, dimana berdasarkan perhitungan manual seperti terlihat pada Gambar 6 menunjukkan ada sebanyak 15 kendaraan yang terlihat jelas dari jarak dekat. Dari data tersebut diperoleh tingkat akurasi deteksi sebesar 26% dan dari hasil *learning* menghasilkan kecepatan gambar 18.2 FPS. Untuk data pengujian keseluruhan pada video ditunjukkan pada Tabel 1.



**Gambar 6.** Hasil pengujian menggunakan metode Haar cascade pada waktu 0 menit 21 detik

Hasil pengujian deteksi kendaraan menggunakan YOLOV3 ditunjukkan pada Gambar 7, dimana video pengujian diambil pada waktu yang sama dengan waktu pengujian menggunakan metode Haar cascade yaitu pada waktu 0 menit 21 detik. Berdasarkan hasil pengujian menunjukkan jumlah kendaraan yang terdeteksi sebanyak 8 kendaraan dengan dibandingkan dengan hasil perhitungan secara manual menunjukkan ada sebanyak 15

kendaraan yang terlihat jelas dari jarak dekat. Dari data tersebut diperoleh tingkat akurasi metode deteksi yaitu sebesar 53%. Walaupun tingkat akurasi YOLOV3 lebih tinggi dibandingkan menggunakan Haar cascade namun dalam proses *learning* metode YOLOV3 membuat beban kerja PC menjadi lebih besar sehingga menurunkan kecepatan gambar menjadi 0.16 FPS.



**Gambar 7.** Hasil pengujian menggunakan metode YOLOV3 pada waktu 0 menit 21 detik

Untuk hasil pengujian menggunakan TinyYOLOV3 ditunjukkan pada Gambar 8. Hasil dibandingkan pada waktu yang sama dengan menggunakan metode Haar cascade dan YOLOV3. Untuk hasil yang didapatkan ada 2 objek kendaraan yang terdeteksi. Dari hasil tersebut diperoleh tingkat akurasi pendeteksian sebesar 13%. Untuk kecepatan pergerakan gambar pada pengujian lebih baik dibandingkan menggunakan YOLOV3 yaitu sebesar 1.57 FPS.



**Gambar 8.** Hasil pengujian menggunakan metode TinyYOLOV3 pada waktu 0 menit 21 detik Untuk

hasil seluruh pengujian dengan durasi video pengujian 1 menit 23 detik ditunjukkan pada Tabel 1.

**Tabel 1.** Perbandingan Hasil Deteksi Keseluruhan

Hitung Manual	Haar Cascade	YOLOV3	TinyYOLOV3
43 Kendaraan	18 Kendaraan	32 Kendaraan	11 Kendaraan

Berdasarkan data pada Tabel 1 menunjukkan perbandingan hasil deteksi ketiga metode ditambah perhitungan secara manual. Hasil tersebut menunjukkan bahwa tingkat akurasi terbaik dalam mendeteksi objek kendaraan adalah menggunakan metode YOLOV3 dengan nilai akurasi sebesar 74%. Metode Haar cascade memiliki tingkat

akurasi terbaik kedua dengan nilai 41% dan metode TinyYOLOV3 mendapatkan hasil tingkat akurasi terendah dengan nilai 25%

#### 4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Hasil pengujian pada waktu sampel 0 menit 21 detik menunjukkan metode YOLOV3 mampu mendeteksi kendaraan bermotor dengan baik sebanyak 8 objek yang terdeteksi dari 15 objek yang terlihat secara jelas pada gambar dengan nilai akurasi sebesar 53%. Untuk metode Haar cascade mendapatkan tingkat akurasi tertinggi kedua sebesar 26%. Sedangkan pada metode TinyYOLOV3 mendapatkan tingkat akurasi terendah dengan nilai 13%.
2. Sedangkan pada pengujian keseluruhan video yang berdurasi 1 menit 23 detik menunjukkan metode YOLOV3 mampu mendeteksi sebanyak 32 kendaraan dari 43 kendaraan yang ada pada video tersebut dengan nilai akurasi sebesar 74%. Pada metode Haar cascade mendapatkan nilai akurasi sebesar 41%, dan TinyYOLOV3 menghasilkan nilai akurasi sebesar 25%.
3. Dari 3 metode yang digunakan pada penelitian ini terdapat masing-masing kekurangan dan kelebihan. Pada metode YOLOV3 menghasilkan akurasi yang baik, namun kecepatan pergerakan gambar lambat akibat proses *learning* data yang lama sehingga menambah beban kerja dari komputer yang digunakan. Pada metode Haar cascade menghasilkan kecepatan tampilan gambar yang baik, namun tingkat akurasi deteksi masih kurang baik dibandingkan YOLOV3. Sedangkan pada TinyYOLOV3 menghasilkan kecepatan tampilan gambar yang lebih baik dari YOLOV3, namun tingkat akurasi masih kalah dengan YOLOV3 dan Haar cascade.

#### Ucapan Terima Kasih

Dengan terselesainya artikel ini, penulis mengucapkan terima kasih kepada Lembaga Penelitian dan Pengabdian Masyarakat (LP2M) Politeknik Negeri Bengkalis, atas bantuan dana yang diberikan dalam penelitian ini, selain itu penulis juga mengucapkan terima kasih kepada seluruh Panitia ABEC 2022 dan semua pihak yang terlibat dalam menyelesaikan penelitian ini.

#### Rujukan

- [1] S. A. Caroline, "Pentingnya Transportasi Umum untuk Kepentingan Publik," *J. Adm. Publik*, vol. 12, no. April, pp. 19–34, 2015.
- [2] M. N. Prayudyanto, "Perbandingan Kinerja Buy The Services Angkutan Umum Massal Kota Metropolitan dengan Metode Biaya Operasional Kendaraan dan Indeks Sustainability," *J. Penelit. Transp. Darat*, vol. 23, no. 1, pp. 55–71, 2021, [Online]. Available: [http://trainingadvokasi.smeru.or.id/cso/file/4\\_3.pdf](http://trainingadvokasi.smeru.or.id/cso/file/4_3.pdf).
- [3] S. S. Sadewo, R. Sumiharto, and I. Candradewi, "Sistem Pengukur Kecepatan Kendaraan Berbasis Pengolahan Video," *IJEIS (Indonesian J. Electron. Instrum. Syst.)*, vol. 5, no. 2, p. 177, 2015, doi: 10.22146/ijeis.7641.
- [4] F. R. Satura, A. A. Chandra, and F. D. Adhinata, "Pengukur Kecepatan Kendaraan Menggunakan Algoritma Image Subtracting," *J. ICTEE*, vol. 2, no. 2, p. 35, 2021, doi: 10.33365/jictee.v2i2.1287.
- [5] M. Zulfikri, K. Abd Latif, R. Hammad, M. Syahrir, and P. Studi, "Deteksi dan Estimasi Kecepatan Kendaraan dalam Sistem Pengawasan Lalu Lintas Menggunakan Pengolahan Citra Detection and Estimation of Vehicle Speed in Traffic Control Systems Using Image Processing," *Agustus*, vol. 20, no. 3, pp. 455–467, 2021.
- [6] P. Hartoto, "Sistem Deteksi Kecepatan Kendaraan Bermotor pada Real Time Traffic Information System," *Fak. Tek. Inst. Teknol. Sepuluh Novemb.*, 2011.
- [7] M. K. A. Fauzi, B. Firman, and A. Novianta, "Pengukur Kecepatan Kendaraan Di Kawasan Pemukiman Menggunakan Algoritma Image Subtracting Berbasis Opencv," *J. Elektr.*, pp. 38–45, 2017.
- [8] A. S. Riyadi, I. P. Wardhani, M. S. Wulandari, and S. Widayati, "Perbandingan Metode ResNet, YoloV3, dan TinyYoloV3 pada Deteksi Citra dengan Pemrograman Python," *Petir*, vol. 15, no. 1, pp. 135–144, 2022, doi: 10.33322/petir.v15i1.1302.
- [9] G. Aprilian Anarki, K. Auliasari, and M. Orisa, "Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 5, no. 1, pp. 179–186, 2021, doi: 10.36040/jati.v5i1.3214.