

## Sistem Penegakan Speed Bump Berdasarkan Kecepatan Kendaraan yang Diklasifikasikan Haar Cascade Classifier

### Speed Bump Enforcement System Based on Vehicle Speed Classified by Haar Cascade Classifier

Muhammad Zulfikri<sup>\*)</sup>, Erni Yudaningtyas, Rahmadwati

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya  
Jl. M.T. Haryono No. 167, Lowokwaru, Malang, Indonesia 65145

---

**Cara sitasi:** M. Zulfikri, E. Yudaningtyas, and R. Rahmadwati, "Sistem Penegakan Speed Bump Berdasarkan Kecepatan Kendaraan yang Diklasifikasikan Haar Cascade Classifier," *Jurnal Teknologi dan Sistem Komputer*, vol. 7, no. 1, 2019. doi: 10.14710/jtsiskom.7.1.2019.12-18, [Online].

---

**Abstract - Driving at high speed is among the frequent causes of accidents. In this research, a warning system was developed to warn drivers when their speed beyond the safety limit. Haar cascade classifier was proposed for the detection system which comprises Haar features, integral image, AdaBoost learning, and cascade classifier. The system was implemented using Python OpenCV library and evaluated on road traffic video collected in one way traffic. As a result, the proposed method yields 97.92% of car detection accuracy in daylight and MSE of 2.88 in speed measurement.**

**Keywords** – speed bump; haar cascade classifier; detection; vehicle speed

**Abstrak – Berkendara dengan kecepatan tinggi merupakan salah satu penyebab terjadinya kecelakaan. Dalam penelitian ini, sistem peringatan dikembangkan untuk memperingatkan pengemudi ketika kecepatan mereka melanggar batas keamanan. Haar cascade classifier diusulkan untuk sistem deteksi yang terdiri dari Haar features, integral image, AdaBoost learning, dan cascade classifier. Sistem ini diimplementasikan menggunakan pustaka OpenCV Python dan dievaluasi pada video lalu lintas yang dikumpulkan dalam lalu lintas satu arah. Metode yang diusulkan menghasilkan akurasi deteksi mobil sebesar 97,92% di siang hari dan MSE 2,88 dalam pengukuran kecepatan kendaraan.**

**Kata Kunci** - speed bump; haar cascade classifier; deteksi; kecepatan kendaraan

#### I. PENDAHULUAN

Perkembangan teknologi yang semakin pesat menyebabkan berbagai terobosan dilakukan oleh para ahli guna mempermudah kinerja manusia dalam berbagai bidang. Salah satunya dalam bidang *Intelligent*

*Transport System* (ITS) yang merupakan integrasi antar sistem informasi dan teknologi komunikasi dengan infrastruktur transportasi, kendaraan dan pengguna jalan. Sistem ini diterapkan untuk mengelola dan mengendalikan lalu lintas, dsitribusi kendaraan dan infrastruktur untuk mencapai sistem transportasi yang lebih aman dan nyaman.

Di beberapa ruas jalan, banyak ditemukan fasilitas *speed bump* (polisi tidur) yang berfungsi memperingatkan pengemudi dalam memperlambat laju kendaraan. Namun pemasangan *speed bump* di beberapa tempat menjadi permasalahan tersendiri, yaitu dapat memberikan ketidaknyamanan bagi pengemudi yang melaju dengan kecepatan rendah. Sehingga perlunya mekanisme secara selektif dalam menegakkan *speed bump* sesuai dengan kecepatan kendaraan yang masuk.

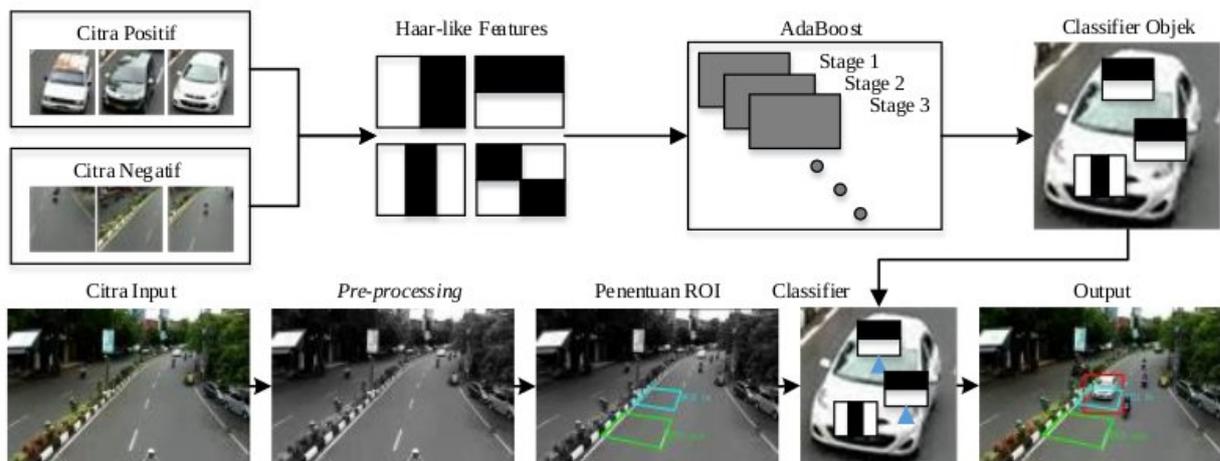
Untuk mendeteksi kecepatan kendaraan, metode yang umum digunakan adalah menggunakan pistol radar yang memancarkan gelombang radar pada suatu objek dan dipantulkan kembali untuk dihitung kecepatannya, namun hasil deteksi akan terpengaruh jika ada perangkat yang menghasilkan gelombang radio di sekitarnya [1]. Deteksi kecepatan kendaraan juga dapat menggunakan dua sensor tekanan dengan menghitung perbedaan waktu pemicu antara kedua sensor [2]. Meskipun begitu penggunaan sensor elektronik memiliki kelemahan, yaitu sensitif terhadap temperatur dan dalam proses instalasi pada permukaan jalan jelek akan memerlukan perbaikan dan pelapisan kembali, dapat membutuhkan instalasi ulang pada sensor [3].

Dari metode pendeteksian kecepatan kendaraan yang masih sepenuhnya belum sempurna, maka perlunya metode alternatif yang lebih baik dalam hal performa dan biaya. Penggunaan kamera dengan menggunakan pengolahan citra telah terbukti memberi hasil yang lebih handal dengan biaya yang lebih rendah [4], [5].

Untuk mengetahui kecepatan kendaraan menggunakan pengolahan citra, perlu adanya sistem untuk mendeteksi kendaraan. Metode yang banyak digunakan diantaranya *adaptive background subtraction* yang digunakan untuk memisahkan antara *background* dengan objek yang di deteksi atau *foreground* [6]-[8].

---

<sup>\*)</sup> Penulis korespondensi (Muhammad Zulfikri)  
Email: mzulfikri@student.ub.ac.id



**Gambar 1.** Proses pelatihan dan pengujian untuk metode *Haar cascade classifier*

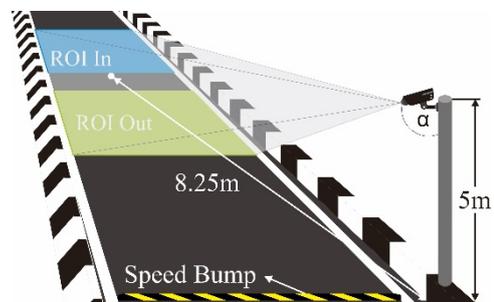
Metode ini berbasis gerakan yang dapat dilakukan secara *real time* dengan tingkat deteksi yang cukup tinggi, tetapi biasanya hanya mempertimbangkan gerak kendaraan dan mengabaikan fitur kendaraan lainnya. Metode berbasis gerakan ini sensitif terhadap *noise* dan biasanya dapat mencapai tingkat kesalahan deteksi yang tinggi [9].

Di sisi lain, pengenalan objek dapat dilakukan dengan metode *Haar cascade classifier* yang merupakan pemodifikasian sistem *face detection* dalam [10]. Metode ini mengenali objek berdasarkan nilai sederhana dari fitur, tetapi bukan merupakan nilai piksel dari citra objek tersebut dan memiliki kelebihan, yaitu komputasinya sangat cepat karena hanya bergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah citra [11]. Metode ini merupakan metode *learning* yang efektif dalam pendeteksian objek, yang menggabungkan beberapa konsep yaitu *Haar features*, *integral image*, *AdaBoost learning*, dan *cascade classifier* menjadi sebuah metode utama untuk mendeteksi objek [12], [13].

Penelitian ini menyajikan sistem penegakan *speed bump* berdasarkan kecepatan kendaraan roda empat (mobil). Deteksi kendaraan dilakukan sebagai parameter dalam perhitungan kecepatan dengan menggunakan metode *Haar cascade classifier* seperti dalam [10]. Penelitian ini mengembangkan beberapa aspek yang telah dilakukan sebelumnya, salah satunya dalam perhitungan kecepatan kendaraan yang masih menggunakan sensor elektronik atau radar.

## II. METODE PENELITIAN

Dalam penelitian ini diusulkan proses utama yaitu proses deteksi kendaraan menggunakan metode *Haar cascade classifier* pada jalan raya, seperti yang ditunjukkan pada Gambar 1. Proses dimulai dari masukan citra dalam bentuk video (diambil menggunakan kamera). Setelah itu, dilakukan *preprocessing* pada citra dengan mengubah citra asal yaitu RGB menjadi citra *grayscale* untuk menyederhanakan model citra. Citra yang telah diubah



**Gambar 2.** Posisi pengambilan citra (video)

menjadi *grayscale* akan ditentukan posisi ROI sebagai batas pendeteksian dengan tujuan proses deteksi objek kendaraan lebih fokus dan akurat. Hasil dari tahap tersebut selanjutnya dilakukan proses deteksi kendaraan menggunakan metode *Haar cascade classifier*. Secara khusus, dalam metode ini diperlukan proses *training classifier* menggunakan sampel citra dari objek. Sampel terdiri dari citra positif (kendaraan) dan citra negatif (*background*). Setelah itu, *classifier* yang dihasilkan dari proses pelatihan (*training*) ini digunakan untuk proses deteksi objek.

Proses terakhir dalam sistem ini adalah perhitungan kecepatan terhadap kendaraan yang terdeteksi. Hasil perhitungan akan menjadi parameter dalam penegakan *speed bump*. Sesuai Peraturan Pemerintah Republik Indonesia No. 79 Tahun 2013 Tentang Lalu Lintas dan Angkutan Jalan, batas kecepatan yang diberikan pada jalan untuk kawasan permukiman yaitu paling tinggi 30 km/jam, sehingga kecepatan tersebut menjadi batas maksimal dalam mengaktifkan *speed bump* [14].

### A. Masukan Citra (Video)

Proses awal dari proses utama penelitian ini adalah proses memasukkan citra dalam program. Data citra yang digunakan adalah video jalan raya pada satu jalur yang diambil menggunakan kamera dengan posisi kamera dari samping jalan dan menghadap ke bawah, seperti ditunjukkan pada Gambar 2.



**Gambar 3.** Penentuan *Region of Interest* (ROI)

Penentuan tata letak kamera dilakukan, dengan dimiringkan dan lurus dengan area deteksi atau ROI (*Region of Interest*) untuk mendapatkan citra yang mencakup bagian ROI yang digunakan. Sudut kemiringan kamera ditentukan menggunakan fungsi untuk menghitung invers tangen (*arctan*) yang merupakan sebuah fungsi trigonometri (Persamaan 1). Parameter  $L$  menyatakan jarak tiang kamera dengan titik tengah ROI dan  $H$  tinggi tiang penempatan kamera. Nilai  $\alpha$  diperoleh sebesar  $50,53^\circ$ , yaitu  $\alpha = \tan^{-1}(8,25\text{ m}/5\text{ m}) = 50,53^\circ$ .

$$\alpha = \tan^{-1}(L/H) \quad (1)$$

Ukuran resolusi video yang diperoleh dan dimasukkan dalam program akan diatur kembali menjadi  $320 \times 240$  piksel sehingga lebih proporsional dan proses komputasi dapat berjalan lebih cepat. Citra video selanjutnya dikonversi dari warna asal RGB menjadi *grayscale* menggunakan fungsi *cv2.cvtColor* yang telah disediakan pada *OpenCV library*. Video kemudian dibaca secara *frame demi frame*.

Sistem ini ditujukan pada jalan satu jalur dan lajur yang bertujuan dalam penegakan *speed bump* hanya untuk satu mobil pada satu kali penegakan. ROI ditentukan dengan membatasi area deteksi, sehingga mengurangi resiko kesalahan deteksi dan proses deteksi menjadi lebih fokus dan akurat. Penentuan ROI dilakukan dengan menentukan titik-titik koordinat ( $x, y$ ) sehingga didapatkan ROI dalam bentuk persegi panjang. Penentuan ROI ditunjukkan pada Gambar 3, yaitu terdapat dua ROI yang dibedakan dengan warna biru (ROI In) dan warna merah (ROI Out). Kedua ROI tersebut diberikan untuk membedakan proses input (ROI In) dan output (ROI Out) dari deteksi mobil yang digunakan dalam proses perhitungan kecepatan mobil. Pengambilan video dilakukan di atas jembatan penyeberangan Jalan Jenderal Basuki Rahmat, Malang.

## B. Deteksi Kendaraan

Pada proses deteksi, terdapat beberapa tahap yang dilakukan yaitu *Haar-like features*, *AdaBoost (Adaptive Boosting)*, dan *cascade classifier*.

### *Haar-like Features*

Tahap pertama adalah memilih fitur *Haar* yang ada pada citra tersebut yang dalam algoritma *Viola-Jones*

disebut dengan *Haar-like features* [10]. Teknik yang dilakukan adalah dengan mengkotak-kotakkan setiap daerah pada citra dari mulai ujung kiri atas sampai kanan bawah. Proses ini dilakukan untuk mencari apakah ada fitur kendaraan pada area tersebut.

Pada setiap kotak-kotak fitur tersebut terdiri dari beberapa piksel dan akan dihitung selisih intensitas piksel pada kotak terang dan kotak gelap. Intensitas piksel pada daerah kotak yang berdekatan dengan citra akan dihitung perbedaannya untuk membentuk *Haar-like features*. Setiap fitur dipasangkan dengan nilai *threshold* dan keputusan klasifikasi ditentukan dengan membandingkan fitur dengan nilai *threshold*. Dalam penelitian ini, empat jenis fitur *Haar* yang digunakan yaitu *edge feature*, *line feature*, dan *four-rectangle feature* seperti yang ditunjukkan pada Gambar 1.

Dalam proses *training* dan pengujian, citra positif dan negatif disatukan. Setiap detektor (satu citra mobil) di *training* menggunakan set sampel positif dan negatif yang berbeda di mana positif merupakan objek mobil, sedangkan negatif mencakup semua objek lainnya selain objek mobil atau citra *background*.

### *AdaBoost (Adaptive Boosting)*

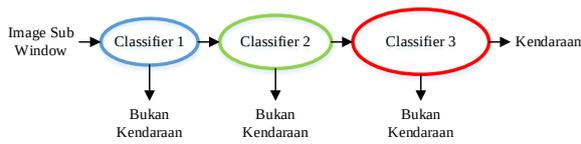
*AdaBoost* berfungsi untuk melakukan pemilihan fitur *haar* dalam jumlah banyak dengan hanya memilih fitur *haar* tertentu. *AdaBoost* ini digunakan untuk memilih fitur spesifik yang akan digunakan dalam mengatur *threshold*. *AdaBoost* menggabungkan banyak *weak classifier* dengan tingkat kebenaran kurang akurat untuk membentuk sebuah *strong classifier*. *AdaBoost* memilih sejumlah *weak classifier* untuk disatukan dan menambahkan bobot pada setiap *classifier* sehingga akan terbentuk sebuah *strong classifier*. Metode untuk dapat beradaptasi dengan *weak classifier* adalah dengan membatasi *weak classifier* ke himpunan klasifikasi fungsi yang masing-masing bergantung pada fitur tunggal.

### *Cascade Classifier*

*Cascade classifier* sebagai sebuah *classifier* yang telah terlatih dengan ribuan contoh objek yang terdiri dari objek positif dan objek negatif memproses banyak fitur yang terorganisir dalam bentuk klasifikasi bertingkat. Dalam algoritma *Viola-Jones* dilakukan penggabungan atau kombinasi *cascade classifier* supaya kecepatan dari proses pendeteksian dapat meningkat, yaitu dengan cara memusatkan perhatian pada daerah yang berpeluang dalam citra. Hal ini berguna untuk menentukan letak objek yang dicari pada suatu citra.

Terdapat tiga buah klasifikasi untuk menentukan apakah ada atau tidak fitur kendaraan pada fitur yang sudah dipilih. Pada klasifikasi filter pertama, tiap subcitra akan diklasifikasi menggunakan satu fitur. Jika hasil nilai fitur dari filter tidak memenuhi kriteria yang diinginkan, hasil tersebut akan ditolak.

Algoritma bergerak ke *sub-window* selanjutnya dan menghitung nilai fitur kembali. Jika didapat hasil sesuai dengan *threshold* yang diinginkan, maka dilanjutkan ke



Gambar 4. Alur cascade classifier



Gambar 5. Hasil pendeteksian

tahap filter selanjutnya. Hingga jumlah *sub-window* yang lolos klasifikasi akan berkurang hingga mendekati citra yang dideteksi. Proses rangkaian filter yang dilalui oleh setiap *classifier* ditunjukkan dalam Gambar 4.

Setelah proses pemilihan fitur dan klasifikasi bertingkat, maka didapatkan sebuah hasil pendeteksian, yang bisa berupa kendaraan atau bukan. Saat proses klasifikasi bertingkat dilakukan, objek yang terdeteksi sebagai sebuah kendaraan akan ditandai sebuah *rectangle* pada daerah kendaraan, seperti yang ditunjukkan pada Gambar 5.

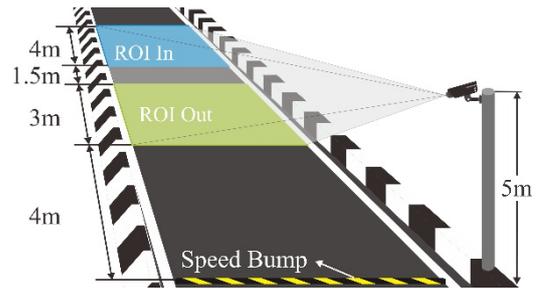
### C. Deteksi Kecepatan

Kendaraan yang terdeteksi dihitung kecepatannya, dengan cara membagi panjang daerah dari ROI *In* dengan waktu kendaraan. Dalam hal ini, ROI *In* diukur panjangnya pada keadaan yang sesungguhnya. Gambar 6 menunjukkan penempatan kamera pada ketinggian 5 meter dari permukaan.

Penghitungan kecepatan kendaraan dilakukan dengan Persamaan 2. Parameter  $V_m$  menyatakan kecepatan kendaraan dalam meter/detik,  $s$  panjang daerah yang ditempuh (meter) dan  $t$  waktu tempuh kendaraan (detik). Untuk menghitung kecepatan video, waktu tempuh kendaraan dapat dikembangkan menjadi Persamaan 3. Parameter  $ft$  menyatakan jumlah frame yang dihitung sejak kendaraan masuk sampai keluar ROI dan  $fps$  menyatakan kecepatan frame dari video (frame/detik). Substitusi Persamaan 2 ke Persamaan 3 menghasilkan persamaan untuk menghitung kecepatan kendaraan pada video (Persamaan 4). Jumlah frame kendaraan ( $ft$ ) dihitung dengan menggunakan Persamaan 5 dengan  $t$  menyatakan waktu kendaraan masuk sampai keluar ROI.

$$V_m = \frac{s}{t} \quad (2)$$

$$t = \frac{ft}{fps} \quad (3)$$



Gambar 6. Tampilan dari kamera pada daerah ROI

$$\text{kecepatan kendaraan} = \frac{s \cdot fps}{ft} \quad (4)$$

$$ft = t \times fps \quad (5)$$

### D. Pengujian Sistem

Efektivitas pengujian diukur dengan dua aspek. Aspek pertama adalah tingkat akurasi dari deteksi kendaraan. Yang kedua adalah tingkat kesalahan rata-rata (MSE) dari perhitungan kecepatan kendaraan.

Tingkat akurasi hasil deteksi kendaraan didapatkan dari perbandingan objek kendaraan yang terdeteksi dengan benar, dengan keseluruhan citra yang digunakan pada data. Objek kendaraan terdiri dari dua bagian, yaitu objek mobil dan objek lain (sepeda motor). Akurasi data dihitung menggunakan Persamaan 6. Akurasi jumlah data benar menyatakan jumlah mobil dan objek lain yang terdeteksi dengan benar dan keseluruhan data menyatakan jumlah total mobil dan objek lain.

$$\text{Akurasi} = \frac{\text{jumlah data benar}}{\text{keseluruhan data}} \times 100\% \quad (6)$$

Nilai kesalahan pada hasil deteksi didapat melalui selisih dari keseluruhan data dan hasil deteksi kendaraan yang dikenali dengan baik. Nilai kesalahan meliputi jumlah kendaraan yang dikenali dengan hasil yang salah, dan tidak dapat dikenali.

Setelah kendaraan terdeteksi, kecepatan kendaraan dihitung dan menjadi parameter dalam penegakan *speed bump*. Pengujian dilakukan dengan mengukur tingkat keakuratan performa dari sistem menggunakan metode MSE sebagai tolok ukur suatu estimator. Semakin mendekati nol, maka semakin baik kinerja estimator tersebut. Nilai MSE dihitung menggunakan Persamaan 7. Parameter  $N$  menyatakan jumlah sampel,  $y_t$  menyatakan nilai aktual indeks dan  $\hat{y}_t$  menyatakan prediksi indeks.

$$MSE = \frac{1}{N} \sum_{i=h}^N (y_t - \hat{y}_t)^2 \quad (7)$$

### III. HASIL DAN PEMBAHASAN

Dalam pengujian, video jalan raya yang telah diambil dimasukkan ke dalam program pengujian. Pengujian dilakukan menggunakan laptop dengan prosesor i3-



**Gambar 7.** Hasil deteksi objek pada (a) ROI In (objek masuk), dan (b) ROI Out (objek keluar)



**Gambar 8.** Tampilan hasil perhitungan kecepatan mobil (a) >30 Km/Jam, dan (b) ≤30 Km/Jam

4030U 1.90GHz, RAM 6 GB, dan sistem operasi Windows 10. Sistem dibuat menggunakan *interpreter python* dengan *library OpenCV*. Sistem yang dirancang disesuaikan untuk dapat diterapkan secara *real time* dalam sistem *embedded* menggunakan *Raspberry pi* yang terintegrasi dengan kamera sebagai sensor visual untuk mengambil citra dan juga terhubung dengan motor servo sebagai penggerak *speed bump*.

Video yang telah dimasukkan akan diproses dengan tahap-tahap yang telah ditentukan. Jika dalam proses deteksi adanya objek berupa mobil terdeteksi pada kedua ROI, maka akan ditandai dengan sebuah kotak merah, seperti yang ditunjukkan pada Gambar 7.

Hasil deteksi yang telah dilakukan menentukan hasil perhitungan kecepatan. Hanya mobil yang terdeteksi dalam kedua ROI yang akan dihitung kecepatannya. Parameter pertama yang diperlukan dalam perhitungan adalah jarak dari ROI In yang diukur panjangnya pada keadaan sesungguhnya. Parameter kedua adalah jumlah frame yang dihitung sejak mobil terdeteksi pada ROI In sampai dengan saat mobil terdeteksi pada ROI Out. Dengan kedua parameter tersebut, kecepatan mobil dapat diketahui dan menjadi parameter dalam proses selanjutnya untuk mengaktifkan *speed bump*.

Gambar 8(a) menunjukkan hasil perhitungan kecepatan pada kondisi kecepatan mobil >30 km/jam, sehingga *speed bump* tetap pada posisi tidak aktif atau *off*. Gambar 8(b) menunjukkan hasil perhitungan kecepatan pada kondisi kecepatan mobil ≤30 km/jam, sehingga posisi *speed bump* menjadi aktif atau *on*.

Dari pengujian yang telah dilakukan, didapatkan hasil deteksi yang ditunjukkan pada Tabel 1. Dengan

**Tabel 1.** Hasil deteksi objek

Kelas Deteksi	Terdeteksi Mobil	Terdeteksi Objek lain	Presi Kelas
Mobil	18	0	100%
Objek lain	1	29	96.67%
Class Recall	94.74%	100.00%	



**Gambar 9.** Tampilan objek lain terdeteksi sebagai mobil

Persamaan 6, tingkat akurasi deteksi menggunakan metode *haar cascade classifier* yaitu 97,91%. Hasil yang belum sempurna tersebut didapatkan karena adanya objek lain yang terdeteksi sebagai mobil, seperti ditunjukkan pada Gambar 9. Dalam pengujian ini, adanya objek lain yang terdeteksi sebagai mobil dipengaruhi oleh sebuah faktor. Faktor tersebut adalah adanya pengaruh intensitas cahaya, dimana saat intensitas cahaya rendah akan mengakibatkan sistem mendeteksi objek lain yang memiliki kemiripan fitur

**Tabel 2.** Hasil perhitungan kecepatan

Mobil	Kec. pada program	Kec. pada video	Perbedaan	MSE
1	39	36	3	2,88
2	20	19	1	
3	28	27	1	
4	20	19	1	
5	25	23	2	
6	25	24	1	
7	23	25	-2	
8	30	32	-2	
9	43	41	2	
10	24	23	1	
11	25	27	-2	
12	27	24	3	
13	23	22	1	
14	19	17	2	
15	28	27	1	
16	19	18	1	
17	20	21	-1	
18	19	18	1	

dengan mobil. Kemiripan fitur tersebut bisa diminimalisir dalam proses *training* yang membutuhkan sampel citra (positif dan negatif) dalam jumlah yang lebih banyak sehingga objek bisa terdeteksi secara akurat dengan membedakan objek lain dan mobil.

Nilai MSE hasil perhitungan kecepatan ditunjukkan dalam Tabel 2. Pengujian dilakukan dengan membandingkan hasil perhitungan kecepatan dari sistem dengan video secara *real time* menggunakan 18 objek mobil yang terdeteksi.

Berdasarkan hasil pengujian perhitungan kecepatan, dengan menggunakan Persamaan 7 didapatkan nilai MSE sebesar 2,88. Nilai MSE yang didapatkan pada penelitian ini masih belum sempurna. Hal ini disebabkan beberapa masalah yaitu adanya proses komputasi yang berat sehingga menghasilkan perbedaan kecepatan pada sistem dibandingkan dengan kecepatan pada video. Sebaliknya, jika proses komputasi berjalan normal dapat menghasilkan kecepatan pada sistem berjalan normal sehingga dapat menghasilkan perhitungan kecepatan yang sesuai dengan video secara *real time*.

Penggunaan kamera sebagai sensor deteksi pada penelitian ini, memberikan hasil yang lebih baik dari pada menggunakan sensor radar [1] dan sensor elektronik [2], dimana penggunaan kamera dapat digunakan untuk memonitor dan mendeteksi kendaraan pada beberapa jalur. Selain itu, kamera dapat menangkap informasi seperti tekstur, warna dan kontras dengan tingkat detail yang baik sehingga memungkinkan dalam proses klasifikasi dan deteksi kendaraan [3]-[5].

Dalam proses klasifikasi dan deteksi kendaraan, penggunaan metode *Haar cascade classifier* dalam sistem ini memiliki kelebihan dibandingkan dengan metode-metode deteksi yang diterapkan dalam penelitian sebelumnya [6]-[8]. *Haar cascade classifier* mampu

memberikan hasil yang baik dalam penerapannya untuk klasifikasi dan deteksi kendaraan pada jalur lalu lintas dengan akurasi 97,91% dan dapat bekerja secara baik secara *real time* yang didukung dengan intensitas cahaya yang baik [9]-[11].

Deteksi kendaraan menggunakan metode ini memberikan hasil yang akurat dan menjanjikan, seperti halnya dalam [12], [13]. Tetapi adanya hasil yang belum sempurna menunjukkan metode tidak bekerja dengan baik dengan berbagai orientasi kendaraan pada video lalu lintas, yang memberikan adanya kesalahan deteksi (objek lain terdeteksi salah). Pemberian ROI merupakan salah satu solusi untuk memperkecil kesalahan deteksi dan memberikan hasil yang akurat.

#### IV. KESIMPULAN

Metode *haar cascade classifier* yang diterapkan dalam deteksi kendaraan pada penelitian ini bekerja dengan baik pada intensitas cahaya tinggi dengan akurasi deteksi sebesar 97,91%. Penghitungan kecepatan kendaraan dengan menerapkan batas area deteksi menggunakan dua ROI (In dan Out) menunjukkan kinerja yang baik, yang dibuktikan dengan nilai MSE mendekati nol yaitu 2,88. Ke depannya, diperlakukan sistem yang dapat dilakukan pada malam hari atau dengan kondisi intensitas cahaya rendah sehingga penambahan lampu penerangan sebagai cahaya tambahan di sekitar area deteksi diperlukan untuk mendapatkan hasil deteksi yang baik. Selain itu, sistem deteksi tidak terbatas hanya pada kendaraan roda empat (mobil), tetapi juga untuk sepeda motor, yang ditunjang dengan metode deteksi objek yang lebih baik sehingga penegakan *speed bump* bisa berlaku untuk semua jenis kendaraan.

#### UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada Universitas Brawijaya atas dukungannya terhadap penelitian ini.

#### DAFTAR PUSTAKA

- [1] S. L. Jeng, W. H. Chieng, and H. P. Lu, "Estimating Speed using a Side-Looking Single-Radar Vehicle Detector," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 607-614, 2014.
- [2] C. Y. Ho, H. Y. Lin, and L. T. Wu, "Intelligent Speed Bump System with Dynamic License Plate Recognition," in *2016 IEEE International Conference on Industrial Technology*, Taipei, Taiwan, 14-17 March 2016, pp. 1669-1674.
- [3] L. E. Y. Mimbela, "A Summary Of Vehicle Detection and Surveillance Technologies Used in Intelligent Transportation Systems," in *International Transport Forum*, Leipzig, Germany, May. 2000.
- [4] D. Bhargava, Kritika; Goyal, "A Video Surveillance System for Speed Detection of

- Vehicles and Law Enforcement using Automatic Number Plate Recognition,” *International Journal of Research in Computer Engineering and Electronics*, vol. 3, no. 3, 2014.
- [5] N. H. Tsani, “Implementasi Deteksi Kecepatan Kendaraan Menggunakan Kamera Webcam dengan Metode Frame Difference,” Skripsi, Universitas Telkom, Bandung, Indonesia, 2017.
- [6] P. K. Thadagoppula and V. Upadhyaya, “Speed Detection using Image Processing,” in *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, Tangerang, Indonesia, Oct. 2016, pp. 11-16.
- [7] A. Varghese, “Background Subtraction for Vehicle Detection,” in *2015 Global Conference on Communication Technologies (GCCT)*, Thuckalay, India, Apr. 2015, pp. 380-382.
- [8] N. A. Mandellos, I. Keramitsoglou, and C. T. Kiranoudis, “A Background Subtraction Algorithm for Detecting and Tracking Vehicles,” *Expert Systems with Applications*, vol. 38, no. 3, pp. 1619-1631, 2011.
- [9] Q. Wu, W. Kang, and X. Zhuang, “Real-Time Vehicle Detection with Foreground-Based Cascade Classifier,” *IET Image Process.*, vol. 10, no. 4, pp. 289-296, 2016.
- [10] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” in *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, USA, Dec. 2001, pp. I-511-I-518.
- [11] Andrew, J. L. Buliali, and Y. Wijaya, “Deteksi Kecepatan Kendaraan Berjalan di Jalan Menggunakan OpenCV,” *Jurnal Teknik ITS*, vol. 6, no. 2, pp. 379-384, 2017.
- [12] D. K. Ulfa and D. H. Widyantoro, “Implementation of Haar Cascade Classifier for Motorcycle Detection,” in *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, Phuket, Thailand, Nov. 2017, pp. 39-44.
- [13] S. Choudhury, “Vehicle Detection and Counting using Haar Feature-Based Classifier,” in *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, Bangkok, Thailand, Aug. 2017, pp. 106-109.
- [14] *Keputusan Menteri Perhubungan No 3 Tahun 1994 tentang Alat Pengendali dan Pengaman Pemakai Jalan*, Menteri Perhubungan Republik Indonesia, Jakarta, 1994.