

Sistem Penegakan *Speed Bump* Berdasarkan Kecepatan Kendaraan pada Malam Hari yang Diklasifikasikan *Haar Cascade Classifier*

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

Muhammad Zulfikri¹, Sirojul Hadi², M. Najmul Fadli³
Universitas Bumigora, Mataram, Indonesia

Article Info

Article history:

Diterima 02 Februari 2023
Direvisi 08 Februari 2023
Disetujui 08 Februari 2023

Kata Kunci:

Deteksi
Haar cascade classifier
Kecepatan
Kendaraan
Speed bump

ABSTRAK

Berkendara dengan kecepatan tinggi merupakan salah satu factor penyebab sering terjadinya kecelakaan. Pemasangan *speed bump* sebagai alat peringatan kendaraan yang sering ditemui pada ruas jalan menjadi salah satu alat yang bertujuan membatasi kecepatan kendaraan yang melintas. Namun hal ini menimbulkan ketidak nyamanan bagi pengemudi kendaraan yang melaju dengan kecepatan rendah. Mekanisme secara selektif dalam mengaktifkan *speed bump* sesuai dengan kecepatan kendaraan yang terdeteksi diperlukan dengan memanfaatkan kamera dengan menggunakan pengolahan citra dan *machine learning* dalam deteksi kendaraan. Hasil deteksi yang didapatkan akan dihitung kecepatannya dan menjadi parameter dalam mengaktifkan *speed bump*. Dalam penelitian ini, sistem peringatan dikembangkan untuk memperingatkan pengemudi ketika kecepatan kendaraan melanggar batas keamanan. Sistem ini diimplementasikan dengan metode *Haar Cascade Classifier* dan bantuan library OpenCV Python dan dievaluasi dari video lalu lintas pada malam hari yang dikumpulkan dalam lalu lintas satu arah. Hasil yang didapatkan dengan metode yang diusulkan yaitu dengan akurasi 86% pada *frame* awal dan 90% pada *Frame* akhir, tingkat presisi yaitu 80% pada *frame* awal dan 86% pada *Frame* akhir, dan tingkat sensitifitas yaitu 92% pada kedua *frame*, dan didapatkan nilai 2,8 dalam pengukuran kecepatan MSE (*Mean Squared Error*).

ABSTRACT

Driving at high speed is one of the causes of frequent accidents. Installation of speed bumps as a vehicle warning device that is often found on roads is one of the tools aimed at limiting the speed of passing vehicles. However, this creates inconvenience for the driver of a vehicle traveling at low speed. Mechanisms to selectively activate speed bumps according to the detected vehicle speed are required by utilizing cameras using image processing and machine learning in vehicle detection. The detection results obtained will calculate the speed and become a parameter in activating the speed bump. In this research, a warning system is developed to warn the driver when the vehicle speed violates the safety limit. This system is implemented with the Haar Cascade Classifier method and the help of the OpenCV Python library and is evaluated from video traffic at night collected in one-way traffic. The results obtained with the proposed method are with an accuracy of 86% on first frame and 90% on last frame, a precision level of 80% on first frame and 86% on last frame, and a sensitivity level of 92% on both frame, and the obtained value 2.8 in the MSE (*Mean Squared Error*) speed measurement.



Copyright ©2023 JOMI: Journal of Millennial Informatics. This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

Penulis Korespondensi:

Muhammad Zulfikri
Ilmu Komputer dan Fakultas Teknik,
Universitas Bumigora, Mataram, Indonesia.
Email: mzulfikri@universitasbumigora.ac.id.

1 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) dimana pada prinsipnya adalah penerapan teknologi maju di bidang elektronika, komputer dan telekomunikasi yang dipadu dengan prinsip manajemen strategi untuk meningkatkan fungsi transportasi secara keseluruhan. Selain itu, ITS juga mampu memberikan informasi secara *real-time* [1]. Fasilitas *speed bump* (polisi tidur) sering ditemui di ruas jalan yang berfungsi memperingatkan pengemudi dalam memperlambat laju kendaraan. Namun pemasangan *speed bump* di beberapa tempat menjadi permasalahan tersendiri, yaitu dapat memberikan ketidak nyamanan bagi pengemudi yang melaju dengan kecepatan rendah. Sehingga perlunya mekanisme secara selektif dalam menegakkan *speed bump* sesuai dengan kecepatan kendaraan yang masuk [2], [3].

Dalam deteksi kecepatan kendaraan, metode yang sering digunakan yaitu menggunakan pistol radar, dengan memancarkan gelombang radar pada suatu objek dan dipantulkan kembali untuk dihitung kecepatannya, namun hasil deteksi bisa terpengaruh jika terdapat perangkat yang menghasilkan gelombang radio di sekitarnya [4]. Deteksi kecepatan kendaraan juga dapat menggunakan dua sensor tekanan dengan menghitung perbedaan waktu pemicu antara kedua sensor [5]. 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 [6]. 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 [7], [8].

Teknologi *image processing* merupakan salah satu bidang riset yang berkembang pesat beberapa tahun belakangan ini. Kemunculan metode-metode pengenalan pola, bentuk benda, dan pengukuran berbasis pengolahan citra telah dimanfaatkan pada beragam aplikasi kehidupan [9]. Pada penelitian oleh Andrew et al., 2017, melakukan deteksi kecepatan kendaraan berjalan di jalan menggunakan *OpenCV*, Tujuannya adalah untuk mengetahui perbedaan hasil deteksi kecepatan dengan berbagai nilai *Frame Per Second* (FPS) dengan *input* aplikasi dalam bentuk video (.avi). Pertama, sistem mengambil *Region of Interest* (ROI), dan selanjutnya sistem menggunakan metode *background subtraction*, membuat garis awal dan akhir, memperbarui posisi kendaraan, dan menyimpan hasil kecepatan rata-rata kendaraan ke berkas Excel (.xls). Pengujian dilakukan 5 kali setiap skenario, dan dibandingkan dengan hasil sebenarnya untuk mendapatkan nilai *error* pada sistem. *Error* terkecil yang dihasilkan sistem sebesar 2,75% dengan posisi koordinat garis akhir di (282,0) pada skenario 30 FPS [10], [11], [12]. Penelitian serupa juga dilakukan dengan membuat sistem penegakan *speed bump* berdasarkan kecepatan kendaraan yang diklasifikasikan *haar cascade classifier*. 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*, *Boost learning*, dan *cascade classifier* [13], [14]. Sistem ini diimplementasikan menggunakan pustaka *OpenCV Python* dan dievaluasi pada video lalu lintas pada lalu lintas satu arah dalam waktu siang hari. Metode yang diusulkan menghasilkan akurasi deteksi mobil sebesar 97,92% di siang hari dan MSE 2,88 dalam pengukuran kecepatan kendaraan [2]. Konsep yang digunakan adalah pengambilan ROI dari video. Dalam hal ini dilakukan proses deteksi berdasarkan bentuk kendaraan mobil secara keseluruhan. Namun pada penelitian tersebut hanya dapat melakukan pengklasifikasian pada siang hari [15].

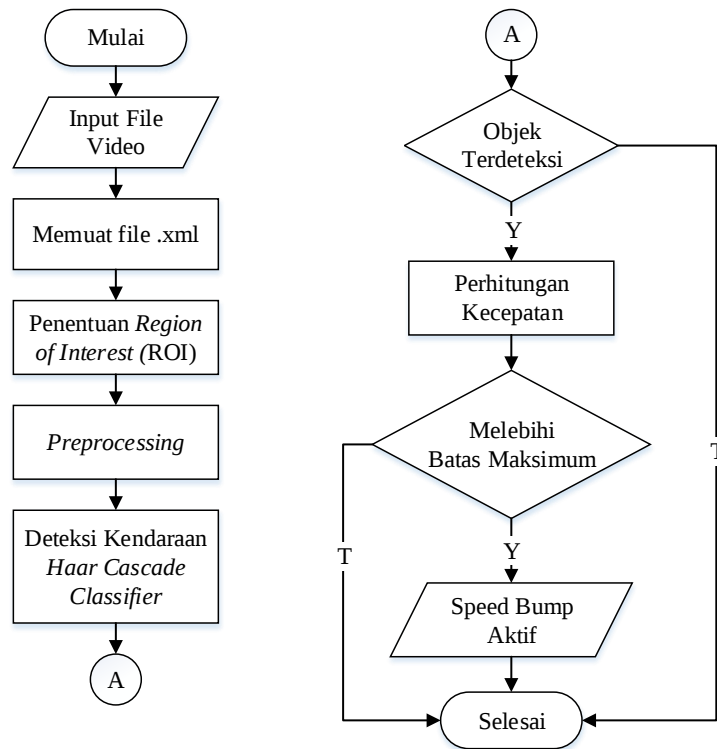
Berdasarkan permasalahan tersebut penulis tertarik untuk melakukan penelitian dengan menerapkan *Haar Cascade Classifier* pada klasifikasi kendaraan dalam sistem penegakan *speed bump* di malam hari, dimana proses deteksi kendaraan berdasarkan lampu depan kendaraan, yang diharapkan dapat melakukan klasifikasi kecepatan kendaraan pada intensitas cahaya yang rendah yaitu pada malam hari. Hasil klasifikasi kecepatan digunakan sebagai parameter untuk mengaktifkan *speed bump* dengan menentukan batas maksimum kecepatan, mengacu pada peraturan pemerintah Peraturan Pemerintah Republik Indonesia Nomor 79 Tahun 2013 Tentang Lalu Lintas yaitu batas kecepatan kendaraan pada pemukiman padat penduduk 30 km/jam.

2 METODE PENELITIAN

Proses utama penelitian ini yaitu dimulai dari proses deteksi kendaraan menggunakan metode *haar cascade classifier* pada jalan raya. Proses awal dimulai dari memasukkan citra dalam bentuk video, yang diambil menggunakan kamera. Pemrosesan awal pada citra dilakukan dengan merubah citra asli yaitu *Red Green Blue* (RGB) menjadi citra grayscale, dengan tujuan untuk menyederhanakan model citra. Selanjutnya akan ditentukan posisi ROI untuk membatasi area deteksi kendaraan sehingga deteksi lebih fokus dan akurat. Selanjutnya metode *haar cascade classifier* diterapkan dalam proses deteksi kendaraan. Metode ini memerlukan proses pelatihan *classifier* dari beberapa sampel citra dari objek, yang terdiri dari citra positif (kendaraan) dan citra negatif (latar belakang). *Classifier* yang dihasilkan dari proses pelatihan ini digunakan untuk proses deteksi objek.

Proses perhitungan kecepatan akan dilakukan untuk mengetahui kecepatan kendaraan yang hasilnya akan digunakan sebagai parameter untuk mengaktifkan *speed bump*. Dalam Peraturan Pemerintah Republik Indonesia No. 79 Tahun 2013 Tentang Jaringan Lalu Lintas dan Angkutan Jalan, telah menetapkan batas kecepatan paling tinggi yang diterapkan pada jalan di kawasan permukiman yaitu 30 km/jam, sehingga kecepatan tersebut menjadi ambang batas dalam mengaktifkan *speed bump* [16]. Dalam melakukan implementasi dari *classifier* yang akan dibuat dan diujikan kedalam sebuah sistem yang ditambahkan

beberapa fitur yang meliputi deteksi dan perhitungan kecepatan kendaraan dengan mengikuti alur proses sistem seperti pada Gambar 1.

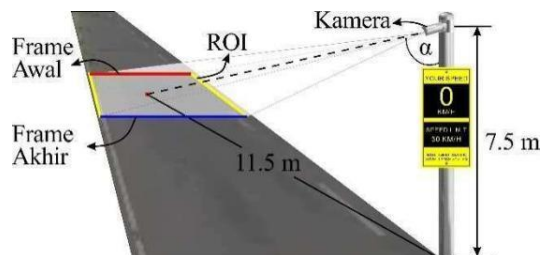


Gambar 1. Alur proses sistem

2.1. Akuisisi Citra (Video)

Tabel dan Gambar disajikan di tengah, seperti yang ditunjukkan di bawah ini dan dikutip dalam manuskrip.

Proses awal dari penelitian ini adalah proses memasukkan citra ke 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 kebawah, seperti pada Gambar 2.



Gambar 2. Posisi akuisi citra (video)

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, yang ditunjukkan pada persamaan 1.

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

dimana,

L = jarak tiang kamera dengan titik tengah ROI

H = tinggi tiang penempatan kamera

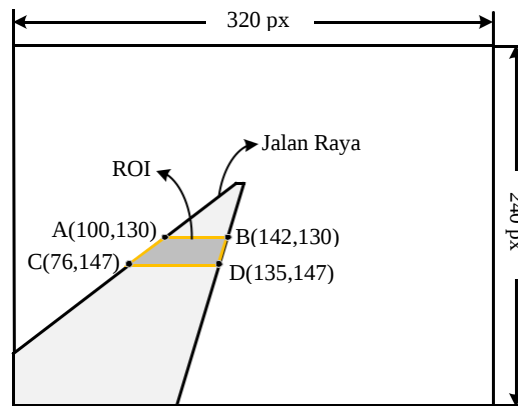
Sehingga didapatkan,

$$\alpha = \tan^{-1}(11,5m/7,5m)$$

$$\alpha = 56,89^\circ$$

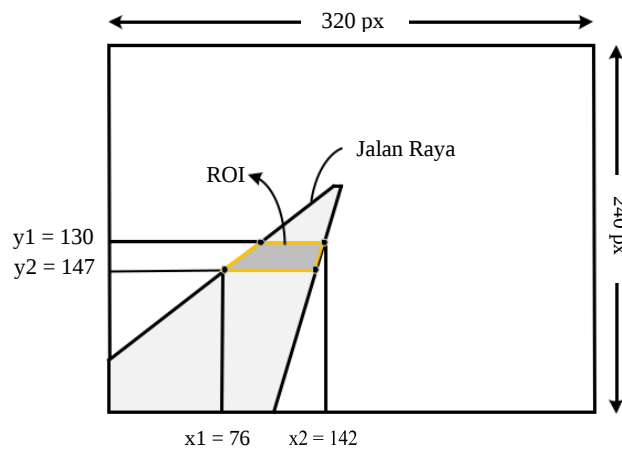
Ukuran resolusi video yang didapatkan dan dimasukkan dalam program akan diatur kembali menjadi 320x240 piksel, sehingga lebih proporsional dan proses komputasi dapat berjalan ringan. Citra video selanjutnya dikonversi dari warna asal *Red Green Blue* (RGB) menjadi citra keabuan (*grayscale*) menggunakan fungsi `cv2.cvtColor` yang telah disediakan pada *OpenCV library*. Setelah itu, video dibaca secara *frame by frame*.

Batasan masalah diberikan yaitu sistem ini diperuntukkan pada jalan satu jalur dan lajur, dengan tujuan dalam mengaktifkan *speed bump*, hanya diperuntukkan 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.



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

Untuk menentukan resolusi dari ROI (x,y), ditentukan terlebih dahulu koordinat (x,y), seperti pada Gambar 4.



Gambar 4. Penentuan koordinat (x, y) ROI

Setelah penentuan koordinat (x, y) ROI, resolusi ditentukan dengan persamaan (2).

$$ROI(x, y) = ((x2 - x1), (y2 - y1)) \quad (2)$$

Sehingga didapatkan resolusi dari ROI yaitu 66x17 piksel.

2.2. Pemrosesan Awal (*Preprocessing*)

Citra video selanjutnya akan dikonversi dari warna asal RGB menjadi *grayscale* menggunakan fungsi `cv2.cvtColor` yang telah disediakan pada *OpenCV library*, dan dilanjutkan pembacaan video secara *frame by frame*. Pada proses pemrosesan awal ini akan menghasilkan hasil berupa ROI *subcitra grayscale*, yang merupakan hasil dari konversi *grayscale* pada daerah ROI.

2.3. Deteksi Kendaraan

Pada proses deteksi kendaraan menggunakan *haar cascade classifier*, terdapat beberapa proses yang dilakukan sebelum akhirnya didapatkan kendaraan yang terdeteksi pada sebuah citra, yaitu sebagai berikut:

1. Pemilihan Fitur

a. Haar-like Features

Citra yang sudah dirubah menjadi citra *grayscale*, selanjutnya dilakukan pememilihan fitur *haar* yang disebut juga dengan *haar-like feature*. Proses ini dilakukan dengan cara mengkotak-kotakkan setiap daerah pada citra, mulai dari bagian kiri atas sampai kanan bawah, dengan tujuan untuk mencari apakah ada fitur kendaraan pada area tersebut.

Setiap kotak-kotak fitur terdiri dari beberapa piksel dan akan dihitung selisih antara nilai piksel pada kotak putih dengan kotak hitam. Apabila nilai selisih antara daerah putih dengan daerah hitam di atas nilai ambang (*threshold*), maka daerah tersebut dinyatakan memiliki fitur. Pada citra kendaraan yang akan dideteksi, kondisinya menghadap ke depan dan untuk mempermudah dan mempercepat proses perhitungan nilai *haar* pada sebuah citra, *haar cascade classifier* menggunakan sebuah perhitungan yang disebut dengan *integral image*.

b. Integral Image

Integral image merupakan representasi citra baru yang digunakan dalam menentukan keberadaan fitur *haar* pada sebuah citra secara efisien. *Integral image* membantu proses perhitungan hanya dengan satu kali pindai dan dengan waktu yang cepat dan akurat. Nilai piksel yang dihitung merupakan nilai piksel dari sebuah citra masukan yang dilalui oleh fitur *haar* pada saat pencarian fitur kendaraan. Pada setiap jenis fitur yang digunakan pada setiap kotak terdiri dari beberapa piksel, seperti pada Gambar 5.



Gambar 5. Nilai piksel pada sebuah fitur

2. Klasifikasi Bertingkat

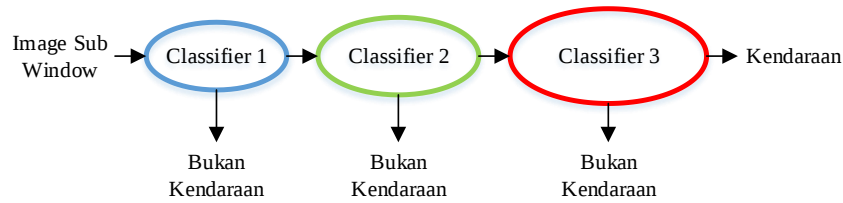
a. Adaptive Boosting (Adaboost)

Adaptive boosting merupakan teknik yang digunakan untuk mengkombinasikan banyak *classifier* lemah untuk membentuk suatu gabungan *classifier* yang lebih baik. Proses dari *adaptive boosting* akan menghasilkan sebuah *classifier* yang kuat dari *classifier* dasar. Satuan dari *classifier* dasar tersebut disebut dengan *weak learner*. Setelah sebelumnya dilakukan pemilihan fitur *haar*, pada proses selanjutnya dalam deteksi yaitu menggunakan algoritma *adaboost*, dimana fitur pada sebuah citra akan dideteksi kembali. Tujuannya untuk mengetahui apakah ada fitur kendaraan pada daerah dengan klasifikasi fitur yang lemah. Pada *classifier* lemah akan dilakukan perhitungan dan dibandingkan dengan *classifier* lainnya secara acak. Selanjutnya dilakukan kombinasi atau penggabungan pada *classifier* lemah untuk membentuk suatu kombinasi yang linier.

b. Cascade classifier

Cascade classifier melakukan proses dari banyak fitur-fitur untuk mengorganisir dengan bentuk klasifikasi bertingkat. Terdapat tiga buah klasifikasi untuk menentukan apakah terdapat fitur kendaraan pada fitur yang sudah dipilih. Pada klasifikasi penyaringan pertama, tiap sub-citra akan diklasifikasi menggunakan satu fitur. Jika hasil nilai fitur dari filter tidak memenuhi kriteria yang diinginkan, hasil tersebut akan ditolak.

Setelah itu, algoritma bergerak ke *sub window* selanjutnya dan menghitung nilai fitur kembali. Jika hasil yang didapatkan sesuai dengan *threshold* yang diinginkan, maka penyaringan dilanjutkan ke tahap selanjutnya hingga jumlah *sub window* yang lolos klasifikasi akan berkurang, mendekati citra yang dideteksi. Pada Gambar 6 menunjukkan proses rangkaian penyaringan yang dilalui oleh setiap *classifier*.

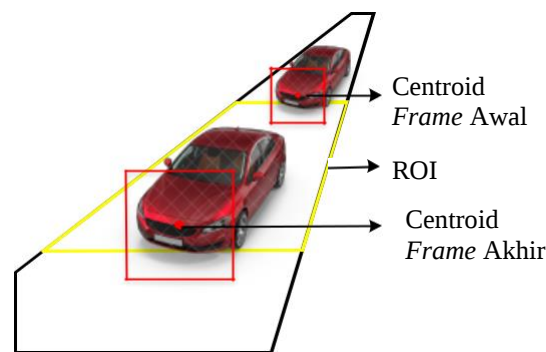


Gambar 6. Alur Cascade Classifier

Setelah proses pemilihan fitur dan klasifikasi bertingkat, akan didapatkan sebuah hasil pendeteksian yang bisa berupa mobil atau bukan. Pada saat proses klasifikasi bertingkat dilakukan maka, pada citra tersebut akan ditandai dengan sebuah kotak pembatas pada daerah kendaraan yang terdeteksi.

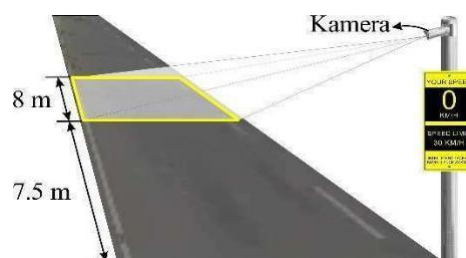
2.4. Perhitungan Kecepatan

Kendaraan yang terdeteksi selanjutnya diberikan *bounding box* dan *centroid* (titik tengah) sebagai titik referensi lokasi suatu objek dan membantu dalam mengetahui jarak mobil yang bergerak dalam *frame* secara berturut-turut, sehingga jumlah *frame* dari setiap pergerakan yang direkam akan diketahui, dan memungkinkan perhitungan kecepatan. *Centroid* yang diperoleh pada *frame* awal ROI dan *frame* akhir ROI digunakan untuk mencari nilai perpindahan jarak antar *frame*, dengan cara mencari selisih resultan antara koordinat *frame* akhir dengan koordinat *frame* awal, seperti pada Gambar 7.



Gambar 7. Tampilan dari kamera pada daerah ROI

Kendaraan yang terdeteksi akan dihitung kecepatannya, dengan cara membagi panjang daerah dari *frame* awal dengan waktu kendaraan. Dalam hal ini, *frame* awal diukur panjangnya pada keadaan yang sesungguhnya. Pada Gambar 8 menunjukkan penempatan kamera pada ketinggian 5 meter dari permukaan.



Gambar 8. Tampilan dari kamera pada daerah ROI

Untuk menghitung kecepatan kendaraan, persamaan yang digunakan adalah:

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

dengan,

V_m = kecepatan kendaraan (meter/detik)

s = panjang daerah yang ditempuh (meter)

t = waktu tempuh kendaraan (detik)

Sedangkan untuk menghitung kecepatan dalam video, waktu tempuh kendaraan dapat dikembangkan menjadi:

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

dengan,

ft = jumlah *frame* yang dihitung sejak kendaraan masuk sampai keluar ROI

fps = jumlah *frame rate* dari video

Selanjutnya dengan mensubstitusi persamaan 2 ke persamaan 3 maka akan diperoleh persamaan 4 untuk menghitung kecepatan kendaraan pada video yaitu :

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

Untuk mencari jumlah *frame* kendaraan (ft), dapat dihitung dengan menggunakan persamaan 5. Seperti yang sudah diketahui, dimana panjang ROI (s) = 4 meter dan *frame rate* video = 30 fps. Untuk mencari nilai ft , dapat dimisalkan jika diketahui dalam sistem mendeteksi kendaraan masuk sampai keluar ROI dengan waktu (t) = 1.5s, maka:

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

$$ft = 1.5s \times 30fps = 70\ frame$$

Sehingga didapatkan jumlah *frame* kendaraan (ft) adalah 70 *frame*.

2.5. Pengujian Sistem

Efektivitas pengujian diukur dengan dua aspek. Aspek pertama adalah tingkat akurasi dari deteksi kendaraan. Yang kedua adalah tingkat kesalahan rata-rata atau *Mean Square Error* (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.

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

dimana,

jumlah data benar = jumlah (mobil + objek lain) terdeteksi dengan benar

keseluruhan data = jumlah total mobil + objek lain

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

Setelah kendaraan terdeteksi, proses selanjutnya yaitu perhitungan kecepatan yang menjadi parameter dalam penegakan *speed bump*. Pengujian dilakukan dengan mengukur tingkat keakuratan performa dari sistem menggunakan metode *Mean Square Error* (MSE). MSE digunakan untuk mengetahui perbedaan antara estimator dengan hasil estimasi. MSE digunakan sebagai tolok ukur suatu estimator. Hasil yang diperoleh selalu berupa angka positif. Semakin mendekati nol maka semakin baik kinerja estimator tersebut. Untuk menghitung nilai MSE, digunakan persamaan 7.

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

dengan,

N = jumlah sampel

y_t = nilai aktual indeks

$\hat{y}_t$ = nilai prediksi indeks

3 HASIL DAN ANALISIS

Dalam pengujian, video jalan raya yang telah diambil dimasukkan ke dalam program pengujian. Pengujian dilakukan menggunakan laptop dengan prosesor i3-4030U 1.90GHz, RAM 6 GB, dan sistem operasi Windows 10. Sistem dibuat menggunakan *interpreter python* dengan *library OpenCV*. Video yang telah dimasukkan akan diproses dengan tahap-tahap yang telah ditentukan. Jika dalam proses deteksi adanya objek berupa mobil terdeteksi pada kedua *frame*, maka akan ditandai dengan sebuah kotak merah, seperti yang ditunjukkan pada Gambar 9.



Gambar 9. Hasil deteksi objek pada ROI

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

Gambar 10 menunjukkan hasil perhitungan kecepatan pada kondisi kecepatan mobil >30 Km/Jam, sehingga *speed bump* menjadi aktif atau *on*.



Gambar 10. Tampilan hasil perhitungan kecepatan mobil >30 Km/Jam

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

Tabel 1. Hasil Deteksi

No	Objek Uji	Deteksi pada ROI	
		Frame awal	Frame akhir
1	Mobil 1	Terdeteksi	Terdeteksi
2	Mobil 2	Terdeteksi	Terdeteksi
3	Mobil 3	Terdeteksi	Terdeteksi
4	Mobil 4	Terdeteksi	Terdeteksi
5	Mobil 5	Terdeteksi	Terdeteksi
6	Mobil 6	Terdeteksi	Terdeteksi
7	Mobil 7	Terdeteksi	Terdeteksi
8	Mobil 8	Tidak Terdeteksi	Terdeteksi
9	Mobil 9	Terdeteksi	Terdeteksi
10	Mobil 10	Terdeteksi	Terdeteksi
11	Mobil 11	Terdeteksi	Terdeteksi
12	Mobil 12	Tidak Terdeteksi	Tidak Terdeteksi
13	Mobil 13	Tidak Terdeteksi	Tidak Terdeteksi
14	Mobil 14	Terdeteksi	Terdeteksi
15	Mobil 15	Terdeteksi	Terdeteksi

Setelah dilakukan pengujian dan didapat hasilnya lalu dilakukan pengujian *performance model* yang sudah dibuat dengan menggunakan tabel *Confusion Matrix*, untuk mengetahui tingkat akurasi, presisi dan sensitifitas.

Tabel 2. Hasil Pengujian Performa

Performa	Frame awal	Frame akhir
Akurasi	86%	90%
Presisi	80%	86%
Sensitifitas	92%	92%

Setelah objek mobil terdeteksi, diperlukan pengujian lebih lanjut untuk membuktikan keakuratan perhitungan kecepatan yang dihasilkan pada sistem dengan perhitungan kecepatan dari video.

Tabel 4 Hasil perhitungan. kecepatan

Mobil	Kec. Program (F_t)	Kec. Video (A_t)	Error ($A_t - F_t$)	Square of Error ($(A_t - F_t)^2$)	Total	MSE
1	21	20	-1	1	5	2,8
2	23	25	2	4		
3	30	30	0	0		
4	40	40	0	0		
5	53	50	3	9		

Berdasarkan hasil pengujian perhitungan kecepatan, nilai rata-rata MSE yang didapatkan yaitu 2,8. Hasil tersebut masih jauh dari kata sempurna dalam perhitungan kecepatan, dimana nilai yang sempurna yaitu nilai *error* yang sangat kecil yaitu mendekati 0 (nol).

4 KESIMPULAN (10 PT)

Dari pengujian yang telah dilakukan pada penelitian ini, didapatkan bahwa penerapan metode *haar cascade classifier* dalam deteksi kecepatan kendaraan pada malam hari berjalan dengan baik hal ini ditandai dengan tingkat akurasi pengujian sistem yaitu 86% pada *Frame* awal dan 90% pada *Frame* akhir, tingkat presisi yaitu 80% pada *Frame* awal dan 86% pada *Frame* akhir, dan tingkat sensitifitas yaitu 92% pada kedua *frame*. Namun, nilai MSE perhitungan kecepatan pada sistem yang dibandingkan dengan video didapatkan nilai MSE yang masih belum sempurna atau jauh mendekati nol yaitu 2,8, hal tersebut terjadi karena proses komputasi, sehingga dibutuhkan sistem yang berjalan pada perangkat dengan spesifikasi yang lebih tinggi sehingga dapat melakukan proses komputasi yang lebih baik.

Pada penelitian selanjutnya, diharapkan sebuah pengembangan yang dapat melakukan pendeteksian pada kendaraan roda dua, dan perlunya penambahan citra pada dataset yang digunakan, sehingga diharapkan dapat menambah tingkat akurasi sistem.

5 UCAPAN TERIMA KASIH

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

REFERENSI

- [1] H. Mandaku and M. Tukan, "Studi Penerapan Intelligent Transportation System (ITS) Di Kabupaten Seram Bagian Barat," *Arika*, vol. 4, no. 1, pp. 33–42, 2010.
- [2] M. Zulfikri, E. Yudaningsy, K. Kendaraan, H. Cascade, and J. Teknologi, "Sistem Penegakan Speed Bump Berdasarkan Kecepatan Kendaraan yang Diklasifikasikan Haar Cascade Classifier," *J. Teknol. dan Sist. Inf.*, vol. 7, no. 1, pp. 12–18, 2019, doi: 10.14710/jtsiskom.7.1.2019.12-18.
- [3] C. Y. Ho, H. Y. Lin, and L. T. Wu, "Intelligent speed bump system with dynamic license plate recognition," *Proc. IEEE Int. Conf. Ind. Technol.*, vol. 5, no. 1, pp. 1669–1674, 2016, doi: 10.1109/ICIT.2016.7475013.
- [4] S. L. Jeng, W. H. Chieng, and H. P. Lu, "Estimating speed using a side-looking single-radar vehicle detector," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 607–614, 2014, doi: 10.1109/TITS.2013.2283528.
- [5] C. Y. Ho, H. Y. Lin, and L. T. Wu, "Intelligent speed bump system with dynamic license plate recognition," *IEEE Int. Conf. Ind. Technol.*, pp. 1669–1674, 2016, doi: 10.1109/ICIT.2016.7475013.
- [6] L. E. Y. Mimbela, "A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems," *Fed. Highw. Adm. Intell. Transp. Syst. Progr. Off.*, 2007.
- [7] D. Bhargava, Kritika; Goyal, "A Video Surveillance System for Speed Detection of Vehicles and Law Enforcement using Automatic Number Plate Recognition," vol. 2, no. 10, 2014.
- [8] N. H. Tsani, I. B. D. M. T, and A. L. P. S. T, "Implemmentasi Deteksi Kecepatan Kendaraan Menggunakan Kamera Webcam dengan Metode Frame Difference The Implementation of Vehicle Speed Detection using Webcam with Frame Difference Method," vol. 4, no. 2, pp. 2373–2381, 2017.
- [9] 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.
- [10] A. Andrew, J. L. Buliali, and A. Y. Wijaya, "Deteksi Kecepatan Kendaraan Berjalan di Jalan Menggunakan OpenCV," *J.*

- Tek. ITS*, vol. 6, no. 2, 2017, doi: 10.12962/j23373539.v6i2.23489.
- [11] A. Varghese and G. Sreelekha, "Background Subtraction for Vehicle Detection," *Glob. Conf. Commun. Technol.*, pp. 380–382, 2015, doi: 10.1109/GCCT.2015.7342688.
- [12] N. A. Mandellos, I. Keramitsoglou, and C. T. Kiranoudis, "A background subtraction algorithm for detecting and tracking vehicles," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 1619–1631, 2011, doi: 10.1016/j.eswa.2010.07.083.
- [13] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, pp. 511–518, 2001, doi: 10.1109/CVPR.2001.990517.
- [14] D. K. Ulfa and D. H. Widyantoro, "Implementation of haar cascade classifier for motorcycle detection," *2017 IEEE Int. Conf. Cybern. Comput. Intell. Cybern. 2017 - Proc.*, vol. 2017-Novem, pp. 39–44, 2018, doi: 10.1109/CYBERNETICSCOM.2017.8311712.
- [15] M. Zulfikri, K. A. Latif, H. Hairani, A. Ahmad, R. Hammad, and M. Syahrir, "Deteksi dan Estimasi Kecepatan Kendaraan dalam Sistem Pengawasan Lalu Lintas Menggunakan Pengolahan Citra," *Techno.Com*, vol. 20, no. 3, pp. 455–467, 2021, doi: 10.33633/tc.v20i3.4588.
- [16] PP No 79 tahun 2013, "PP Republik Indonesia Nomor 79 Tahun 2013 Tentang Jaringan Lalu Lintas dan Angkutan Jalan," *Jaringan Lalu Lintas Dan Angkutan Jalan*, no. Peraturan pemerintah republik Indonesia. pp. 1–97, 2013.