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

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

Muhammad Zulfikri[1], Erni Yudaningtyas[2], Rahmadwati[3]
[1,2,3]Magister Teknik Elektro, Fakultas Teknik, Universitas Brawijaya
Jl. M.T. Haryono No.167, Lowokwaru, 65145, Malang, Indonesia
e-mail: [*1]mzulfikri@student.ub.ac.id , [2]erni@ub.ac.id, [3]rahma@ub.ac.id

Abstrak

Kecelakaan lalu lintas sering terjadi disebabkan kendaraan yang melaju dengan kecepatan tinggi. Penelitian ini mengembangkan sistem penegakan speed bump (polisi tidur) yang dapat memberikan peringatan bagi pengemudi dalam memperlambat laju kendaraan dan memberikan kenyamanan saat melaju dengan kecepatan rendah. Penegakan speed bump dilakukan berdasarkan kecepatan kendaraan yang terdeteksi menggunakan metode Haar Cascade Classifier, yang merupakan gabungan beberapa konsep yaitu Haar Features, Integral Image, AdaBoost Learning, dan Cascade Classifier. Pengujian dilakukan menggunakan video jalan raya pada satu jalur. Sistem dibuat menggunakan interpreter python dengan library OpenCV. Pendeteksian didapatkan hasil yang cukup baik apabila dilakukan pada intensitas cahaya tinggi, dan didapatkan tingkat akurasi deteksi sebesar 97,92%. Perhitungan kecepatan kendaraan didapatkan dengan membandingkan hasil kecepatan pada sistem dengan video dalam keadaan real time, yang dibuktikan dari tingkat error dengan nilai MSE yaitu 2,88.

Kata kunci— speed bump, haar cascade classifier, deteksi, kecepatan, kendaraan

Abstract

Traffic accidents are frequent due to vehicles which drove at high speed. This research has developed a speed bump enforcement system which can give a warning for drivers in slowing the vehicle and gives comfort when driving at low speeds. Speed bump enforcement is based on the vehicle speed detected using the Haar Cascade Classifier method is used in detection, which is a combination of several concepts namely Haar Features, Integral Image, AdaBoost Learning, and Cascade Classifier. Testing is conducted using road traffic video on one way. The system is created using the python interpreter with the OpenCV library. Detection is obtained good results when carried out at high light intensity, and the detection accuracy rate is 97.92%. The calculation of vehicle speed is obtained by comparing the results of the speed on the system with the video in real time, as evidenced by the error rate with the MSE value of 2.88.

Keywords— *speed bump, haar cascade classifier, detection, speed, vehicle*

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) 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* atau polisi tidur yang berfungsi

memperingatkan pengendara agar memperlambat laju kendaraan dan memperhatikan daerah sekitarnya. Namun pemasangan speed bump dibeberapa tempat menjadi permasalahan tersendiri, seperti yang terpasang di lingkungan sekolah atau persimpangan tertentu dapat memberikan ketidaknyamanan bagi pengemudi kendaraan yang melaju dengan kecepatan rendah. Diharapkan ada mekanisme secara selektif dalam menegakkan speed bump sesuai dengan kecepatan kendaraan yang masuk.

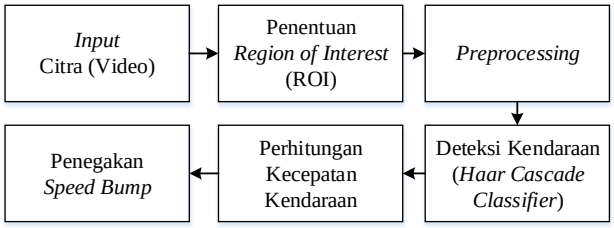
Untuk mendeteksi kecepatan kendaraan, metode yang umum digunakan adalah menggunakan pistol radar. Pistol radar bekerja dengan memancarkan gelombang radar yang diarahkan pada suatu objek yang bergerak dan dipantulkan kembali untuk dihitung kecepatan dari objek tersebut. Teknologi radar memberikan hasil yang menjanjikan dan sangat akurat, namun harganya sangat mahal dan jika ada perangkat yang menghasilkan gelombang radio di sekitarnya, dapat mempengaruhi hasil deteksi [1]. Deteksi kecepatan kendaraan juga dapat menggunakan dua sensor tekanan dengan menghitung perbedaan waktu pemicu antara kedua sensor [2]. Meskipun begitu penggunaan sensor elektronik tersebut memiliki kelemahan, yaitu sensitif terhadap temperatur dan dalam proses instalasi pada permukaan jalan jelek yang 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]. 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* [5]. 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. Oleh karena itu, metode berbasis gerakan sensitif terhadap *noise* dan biasanya dapat mencapai tingkat kesalahan deteksi yang tinggi [6].

Oleh karena itu, dalam penelitian ini penulis menyajikan sistem penegakan *speed bump* berdasarkan kecepatan kendaraan. Kendaraan yang akan di deteksi adalah kendaraan roda empat (mobil). Penelitian ini bertujan mengembangkan beberapa aspek yang telah dilakukan oleh para peneliti sebelumnya, salah satunya dalam perhitungan kecepatan kendaraan yang masih menggunakan sensor elektronik atau radar. Selain itu deteksi kendaraan dilakukan sebagai parameter dalam perhitungan kecepatan yang menggunakan metode *Haar cascade classifier* yang merupakan pemodifikasian sistem *face detection* yang disusulkan oleh Viola dan Jones. Prinsipnya adalah mengenali objek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari *image* objek tersebut. Metode ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah *image* 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 [8].

2. METODE PENELITIAN

Dalam penelitian ini, diusulkan proses utama yaitu proses deteksi kendaraan menggunakan metode *Haar cascade classfier* pada jalan raya yang telah ditentukan *Region of Interest* (ROI) dan telah dilakukan proses *preprocessing*. Setelah kendaraan terdeteksi, sistem akan melakukan perhitungan kecepatan kendaraan yang menjadi parameter di dalam penegakan *speed bump*.



Gambar 1 Skema penegakan *speed bump*

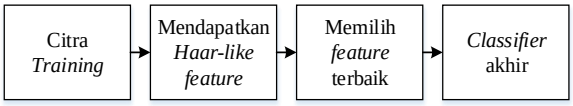
Gambar 1 menunjukkan bahwa proses dimulai dari *input* citra dalam bentuk video (diambil menggunakan kamera). Dalam prosesnya, ditentukan posisi ROI sebagai batas pendeteksian dengan tujuan proses deteksi objek kendaraan lebih fokus dan akurat. Setelah itu, dilakukan *preprocessing* pada citra di dalam ROI dalam bentuk proses *rescale* dan *grayscale*. Hasil dari *preprocessing* selanjutnya akan 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 *training* ini digunakan untuk proses deteksi objek.

Langkah terakhir yaitu perhitungan kecepatan terhadap kendaraan yang terdeteksi. Hasil perhitungan akan menjadi parameter di 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*.

Adapun tahapan lebih jelas dari penelitian akan dijelaskan sebagai berikut:

A. *Classifier Training*

Gambar 2 menggambarkan proses *classifier training* yang dilakukan sebelum proses utama untuk mendapatkan *classifier* yang dibutuhkan. Proses *training* menggunakan dua jenis data. Data pertama adalah ratusan citra dari objek kendaraan yang diambil dalam berbagai kondisi, yang disebut sampel positif (Gambar 3). Data kedua adalah ratusan gambar yang tidak mengandung objek kendaraan, juga diambil dalam berbagai kondisi, yang disebut sampel *background* atau sampel negative (Gambar 4). Ratusan citra ini akan diolah menjadi sampel dengan ukuran yang telah ditentukan. Selain itu, *classifier* akan di *training* sampai jumlah tahap yang telah ditentukan.



Gambar 2 Proses *classifier training*

Dalam proses *training*, sistem akan mencari *haar-like feature* yang menonjol pada citra sampel. *Haar-like feature* memproses citra dalam kotak-kotak, dimana dalam satu kotak terdapat beberapa piksel. Setiap kotak kemudian diproses dan didapatkan perbedaan nilai ambang (*threshold*) yang menandakan daerah gelap dan terang. Jika perbedaan lebih besar dari *threshold*, maka daerah tersebut dianggap sebagai fitur dari objek.

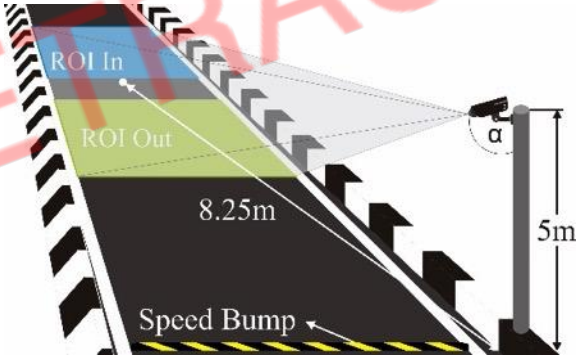
Gambar 3 Sampel positif yang digunakan dalam *training*

Gambar 4 Sampel negatif yang digunakan dalam *training*

Dalam video, perhitungan dan penjumlahan *pixel* dilakukan secara terus-menerus dan membutuhkan waktu yang lama. Sehingga, perhitungan diganti dengan *integral image* untuk mendapatkan hasil yang lebih cepat. Sebuah *image* dapat memiliki jumlah *haar-like feature* yang besar, sehingga perlu memilih fitur terbaik dalam klasifikasi objek. Pemilihan dilakukan dengan menggunakan algoritma *AdaBoost*. Dalam algoritma ini, salah satu fitur dipilih dari *haar-like feature* yang diperoleh. Fitur ini kemudian akan digunakan untuk menentukan apakah sampel berlabel positif atau negatif. Semakin banyak fitur yang digunakan, akan didapatkan hasil yang semakin akurat. Dalam prosesnya, *haar-like feature* yang banyak diatur di dalam *cascade classifier*. Sampel yang diberi label dengan benar oleh *classifier* pada tahap pertama akan digunakan untuk melatih *classifier* untuk tahap berikutnya. Dengan setiap tahap, fitur yang digunakan untuk membangun *classifier* akan meningkat dan menyebabkan proses klasifikasi menjadi ketat. *Training* akan terus berlanjut sampai memperoleh hasil klasifikasi dengan sejumlah tahapan yang ditentukan. Semua hasil klasifikasi tersebut kemudian akan digabungkan ke dalam *classifier* yang lebih besar. Sehingga hasil dari *cascade classifier* ini, hasil klasifikasi *classifier* awal akan diteruskan ke *classifier* pada tahap berikutnya

B. Input 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 kebawah, seperti pada Gambar 5.



Gambar 5 Posisi pengambilan citra (video)

Penentuan tata letak kamera dilakukan, dengan dimiringkan dan lurus dengan titik tengah ROI untuk mendapatkan citra yang mencakup bagian ROI yang telah ditentukan. Sudut kemiringan kamera ditentukan menggunakan fungsi untuk menghitung invers tangen (arctan) yang merupakan sebuah fungsi trigonometri, yang ditunjukkan pada persamaan 1.

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

dimana,
L = jarak tiang kamera dengan titik tengah ROI
H = tinggi tiang penempatan kamera

Sehingga didapatkan,

$\alpha = tan^{-1}(8{,}25m/5m)$
$\alpha = 50{,}53°$

Ukuran resolusi video yang didapatkan dan dimasukkan dalam program akan diatur kembali menjadi 320x240 piksel, sehingga lebih proporsional dan proses komputasi dapat berjalan lebih cepat. Setelah itu, video dibaca secara *frame by frame*. Batasan masalah diberikan yaitu sistem ini diperuntukkan pada jalan satu jalur dan lajur, dengan tujuan 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 6, dimana 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.

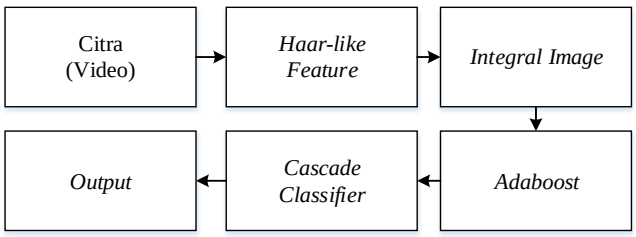
Gambar 6 Penentuan *Region of Interest* (ROI

C. *Pre-Processing*

Citra video selanjutnya akan dikonversi dari warna asal *Red Green Blue* (RGB) menjadi *grayscale* menggunakan fungsi *cv2.cvtcolor* yang telah disediakan pada *OpenCV library*. Pada proses *preprocessing* ini akan menghasilkan *output* berupa *Region of Interest subcitra grayscale*, yang merupakan hasil dari konversi *grayscale* pada daerah ROI.

D. Deteksi Kendaraan

Pada proses deteksi kendaraan menggunakan metode *Haar Cascade Classifier*, terdapat beberapa proses untuk mendapatkan *output* kendaraan yang terdeteksi. Proses tersebut yaitu *Haar-Like Featrure*, *Integral image*, *Adaboost (Adaptive Boosting)*, dan *Cascade Classifier*. Proses dari setiap tahap dalam deteksi kendaraan ditunjukkan pada Gambar 7.



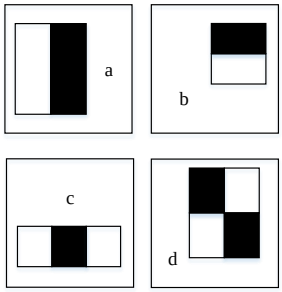
Gambar 7 Proses deteksi *Haar Cascade Classifier*

Detail dari tiap tahap dalam proses deteksi kendaraan pada Gambar 7 adalah sebagai berikut:

1. *Haar-like Feature*

Citra yang telah dirubah menjadi citra *grayscale*, selanjutnya dilakukan proses memilih fitur *Haar* yang ada pada citra tersebut yang dalam algoritma *Viola-Jones* disebut dengan *Haar-*

Like feature. Teknik yang dilakukan yaitu dengan cara 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. Dalam algoritma *Viola-Jones*, ada beberapa jenis fitur yang bisa digunakan seperti *Edge-feature*, *Line feature*, dan *Four-rectangle feature* seperti yang ditunjukkan pada Gambar 8.

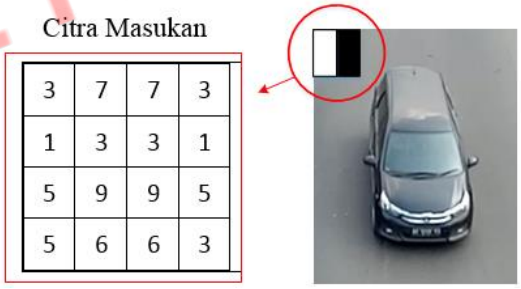


Gambar 8 *Haar-like feature*, (a, b) *Edge-feature*, (c) *Line-feature*, (d) *Four-rectangle feature*

Pada setiap kotak-kotak fitur tersebut terdiri dari beberapa pixel dan akan dihitung selisih antara nilai pixel pada kotak terang dengan nilai piksel pada kotak gelap. Apabila nilai selisih antara daerah terang dengan daerah gelap di atas *threshold*, maka daerah tersebut dinyatakan memiliki fitur. Pada citra kendaraan yang akan dideteksi, kondisinya menghadap ke depan. Untuk mempermudah dan mempercepat proses perhitungan nilai *haar* pada sebuah citra, dalam algoritma *Viola-Jones* digunakan sebuah perhitungan yang disebut dengan *Integral Image*.

2. *Integral Image*

Integral image adalah reperesentasi citra baru yang digunakan untuk menentukan keberadaan fitur *haar* pada sebuah citra secara efisien. *Integral image* melakukan proses perhitungan hanya dengan satu kali *scan* dengan waktu yang cepat dan akurat. *Integral image* digunakan untuk menghitung hasil penjumlahan nilai piksel pada daerah yang dideteksi oleh fitur *haar*. Nilai-nilai piksel yang dihitung merupakan nilai piksel dari sebuah citra masukan dan dilalui oleh fitur *haar* pada saat pencarian fitur kendaraan. Pada setiap jenis fitur *haar* yang digunakan dan pada setiap kotak-kotaknya terdiri dari beberapa piksel, seperti pada Gambar 9.



Gambar 9 Nilai piksel pada sebuah fitur

3. *Adaboost (Adaptive Boosting)*

Adaboost merupakan algoritma yang berfungsi untuk melakukan pemilihan fitur *haar* dalam jumlah banyak dengan hanya memilih fitur *haar* tertentu, dan merupakan tahap ketiga dalam proses pendeteksian objek *Viola-Jones*.

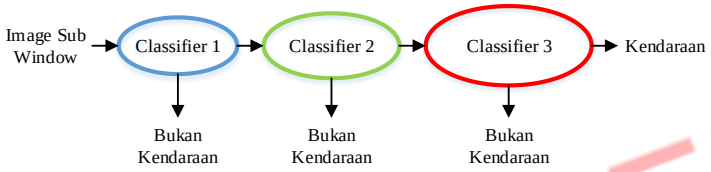
Adaboost digunakan untuk memilih fitur spesifik yang akan digunakan dalam mengatur *treshold*. *Adaboost* menggabungkan banyak *weak classifier* untuk membentuk sebuah *strong classifier*. *Weak classifier* merupakan suatu jawaban dengan tingkat kebenaran yang kurang akurat. *Adaboost* memilih sejumlah *weak classifier* untuk disatukan dan menambahkan bobot pada setiap *classifier* sehingga akan terbentuk sebuah *strong classifier*. Salah satu metode yang

cepat untuk dapat beradaptasi dengan *weak classifier* adalah dengan membatasi *weak classifier* ke-set klasifikasi fungsi yang masing-masing bergantung pada fitur tunggal.

4. *Cascade classifier*

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

Cascade classifier memproses banyak fitur-fitur yang terorganisir dalam bentuk klasifikasi bertingkat. 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 kemudian bergerak ke sub window selanjutnya dan menghitung nilai fitur kembali. Jika didapat hasil sesuai dengan *threshold* yang diinginkan, maka dilanjutkan ke tahap filter selanjutnya. Hingga jumlah sub window yang lolos klasifikasi akan berkurang hingga mendekati citra yang dideteksi. Pada Gambar 10 menunjukkan proses rangkaian filter yang dilalui oleh setiap *classifier*.



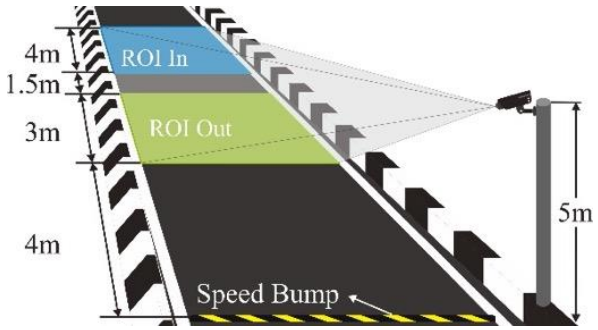
Gambar 10 Alur *Cascade Classifier*

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 11.

Gambar 11 Hasil pendeteksian

E. Deteksi Kecepatan

Kendaraan yang terdeteksi akan dihitung kecepatannya, dengan cara membagi panjang daerah dari ROI *In* dengan waktu kendaraan. Dalam hal ini, ROI *In* diukur panjangnya pada keadaan yang sesungguhnya. Pada Gambar 12 menunjukkan penempatan kamera pada ketinggian 5 meter dari permukaan.



Gambar 12. Tampilan dari kamera pada daerah ROI

Untuk menghitung kecepatan kendaraan, persamaan yang digunakan adalah:

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

dengan,

Vm = 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:

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

dimana,

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

fps = jumlah frame rate dari video

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

$kecepatan\ kendaraan = \frac{s\ .\ fps}{ft}$ .....(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:

$ft = t\ x\ fps$.....(5)

$ft = 1.5s\ x\ 30fps$

$ft = 70\ frame$

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

3. HASIL DAN PEMBAHASAN

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*. Sistem yang dirancang disesuaikan untuk dapat diterapkan secara *real time* dalam *embedded system* menggunakan *raspberry pi* yang terintegrasi dengan kamera sebagai sensor visual untuk mengambil citra dan juga terhubung dengan motor servo sebagai penggerak *speed bump*.

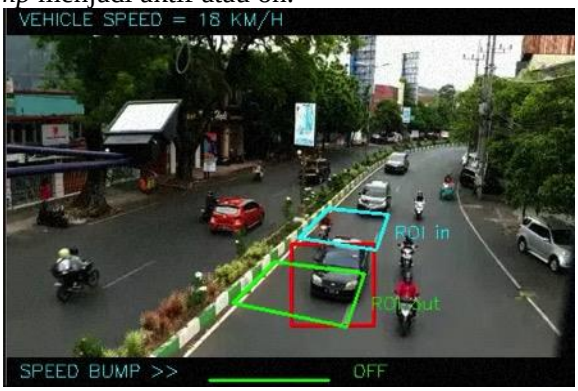
Video yang telah dimasukkan akan diproses dengan tahap-tahap yang telah ditentukan. Jika dalam proses deteksi adanya objek berupa mobil terdeteksi pada kedua ROI, maka akan ditandai dengan sebuah kotak merah, seperti yang ditunjukkan pada Gambar 13.

(a)

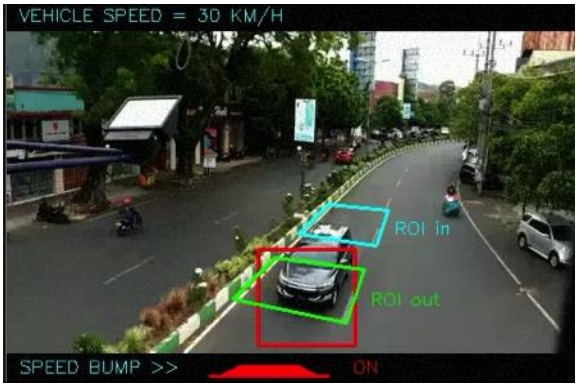
(b)

Gambar 13 Hasil deteksi objek pada (a) ROI In (objek masuk), dan (b) ROI Out (objek keluar)

Hasil deteksi yang telah dilakukan menentukan hasil perhitungan kecepatan. Sehingga hanya mobil yang terdeteksi dalam kedua ROI yang akan dihitung kecepatannya. Dengan menggunakan rumus yang telah ditentukan, 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 akan diketahui dan menjadi parameter dalam proses selanjutnya untuk mengaktifkan *speed bump*. Gambar 14 (a) menunjukkan hasil perhitungan kecepatan pada kondisi kecepatan mobil >30 Km/Jam, sehingga *speed bump* tetap pada posisi tidak aktif atau off. Sedangkan Gambar 14 (b) menunjukkan hasil perhitungan kecepatan pada kondisi kecepatan mobil ≤30 Km/Jam, sehingga posisi *speed bump* menjadi aktif atau on.



(a)



(b)

Gambar 14 Tampilan hasil perhitungan kecepatan mobil (a) >30 Km/Jam, dan (b) ≤30 Km/Jam

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 kesulurah 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.

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

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

Tabel 1. Hasil Deteksi

Jumlah objek mobil	18
Jumlah objek lain	30
Mobil terdeteksi dengan benar	18
Mobil terdeteksi sebagai objek lain	0
Objek lain terdeteksi sebagai mobil	1
Objek lain terdeteksi dengan benar	29

Dengan menggunakan persamaan 6, didapatkan:

$$Akurasi = \frac{47}{48} \text{x } 100\%$$

$$Akurasi = 97{,}91\%$$

Berdasarkan hasil yang didapat, tingkat akurasi menggunakan metode *haar casacade classifier* yaitu 97,91%. Hasil yang belum sempurna tersebut didapatkan karena adanya objek lain yang terdeteksi sebagai mobil, seperti ditunjukkan pada Gambar 15.

Gambar 15 Tampilan objek lain terdeteksi sebagai mobil

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 [9]. 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.

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

dimana,

N = jumlah sampel

y_t = nilai aktual indeks

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

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

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	

Berdasarkan hasil pengujian perhitungan kecepatan, didapatkan nilai MSE sebesar 2.88. Dalam perhitungan nilai MSE, hasil yang mendekati sempurna adalah nilai error yang sangat kecil yaitu mendekati nol. Sehingga nilai MSE yang didapatkan pada penelitian ini masih belum sempurna, disebabkan karena 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.*

4. KESIMPULAN

Metode *haar casacade classifier* yang diterapkan dalam deteksi kendaraan pada penelitian ini bekerja dengan baik pada intensitas cahaya tinggi, dimana akurasi deteksi yang didapatkan yaitu 97,91%. Sedangkan perhitungan kecepatan 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. Untuk kedepannya, perlunya sistem yang dalam proses deteksinya dilakukan pada intensitas cahaya rendah, dengan tujuan adanya sistem yang dapat bekerja pada kondisi siang dan malam. Hal tersebut juga dapat didukung dengan metode yang lebih baik dalam deteksi objek.

5. SARAN

Kedepannya, perlunya sistem yang dapat dilakukan pada malam hari atau dengan kondisi instensitas 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.

DAFTAR PUSTAKA

[1] 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.

[2] 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. 2016–May, pp. 1669–1674, 2016.

[3] L. E. Y. Mimbela, “A Summary of Vehicle Detection and Surveillance Technologies used in IN INTELLIGENT TRANSPORTATION SYSTEMS,” 2000 dalam https://www.fhwa.dot.gov/ohim/tvtw/vdstits.pdf diakses pada 11 Oktober 2018.

[4] D. Bhargava, Kritika; Goyal, “A Video Surveillance System for Speed Detection of Vehicles and Law Enforcement using Automatic Number Plate Recognition,” *Int. Journal of Digital Application & Contemporary research.,* vol. 2, no. 10, 2014.

[5] P. K. Thadagoppula and V. Upadhyaya, “Speed detection using image processing,” *2016 Int. Conf. Comput. Control. Informatics its Appl.*, pp. 11–16, 2016.

[6] 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.

[7] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition. CVPR 2001*, vol. 1, p. I-511-I-518, 2001.

[8] Andrew, J. L. Buliali, and Y. Wijaya, “Deteksi Kecepatan Kendaraan Berjalan di Jalan Menggunakan OpenCV,” *J. Tek. ITS*, vol. 6, no. 2, pp. 379–384, 2017.

[9] and T. E. U. of I. Office for Mathematics, Science, “Mean Square Error,” 2004.

[10] D. K. Ulfa and D. H. Widyantoro, "Implementation of haar cascade classifier for motorcycle detection," *IEEE Int. Conf. Cybern. Comput. Intell. Cybern. 2017 - Proc.*, vol. 2017–Novem, pp. 39–44, 2018.

[11] S. Choudhury, "Vehicle Detection and Counting using Haar Feature- Based Classifier," *Proc. IEEE Ann. Ind. Auto. and Electromec. Eng. Conf. (IEMECON).*, pp. 106–109, 2017.

[12] N. H. Tsani, I. B. D. M. T, and A. L. P. S. T, "Implementasi Deteksi Kecepatan Kendaraan Menggunakan Kamera Webcam dengan Metode Frame Difference The Implementation of Vehicle Speed Detection using Webcam with Frame Difference Method," *e-Proc. of Eng.*, vol. 4, no. 2, pp. 2373–2381, 2017.

[13] A. Varghese, "Background Subtraction for Vehicle Detection," *Glob. Conf. Commun. Technol. 2015) Backgr.*, no. Gcct, pp. 380–382, 2015.

[14] B. Juhendaja and E. Petlenkov, "Vehicle detection methods and their comparison in the context of Tallinn's traffic surveillance videos," *Bachelor's thesis of Tallinn University of Technology*., 2017.

[15] 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.

[16] K. M. R. Indonesia, "Keputusan Menteri Perhubungan No 3 Tahun 1994 tentang Alat Pengendali dan Pengaman Pemakai Jalan." Jakarta, 1994.