

DOI: doi.org/10.21009/03.SNF2022.01.FA.18

SISTEM PENDETEKSIAN KECEPATAN KENDARAAN MENGGUNAKAN *HAAR CASCADE* BERBASIS RASPBERRY PI

Panji Ahmad Nurhusni^{a)}, Hadi Nasbey^{b)}, Riser Fahdiran^{c)}

Program Studi Fisika, FMIPA, Universitas Negeri Jakarta, Jl. Rawamangun Muka, Kec. Pulo Gadung, Kota Jakarta Timur, DKI Jakarta, 13220, Indonesia

Email: ^{a)}panjihmahmadnurhusni_3225163503@mhs.unj.ac.id, ^{b)}hadinasbey@unj.ac.id, ^{c)}riser-fahdiran@unj.ac.id

Abstrak

Penelitian ini bertujuan mengembangkan sistem yang dapat mengidentifikasi kendaraan di jalan raya dan mendeteksi kecepatan kendaraan-kendaraan tersebut. Sistem ini dikembangkan pada Raspberry Pi menggunakan Python 3.7. Metode pada penelitian ini menggunakan metode *Haar cascade* untuk melakukan pengidentifikasian dan pelacakan obyek kendaraan pada rekaman video sehingga nilai kecepatan kendaraan dapat diestimasi. Pengujian dilakukan dengan menggunakan rekaman-rekaman video kendaraan di jalan raya dengan variasi nilai kecepatan yang berbeda. Dua metode analisis digunakan untuk mengevaluasi performa sistem dalam mengidentifikasi kendaraan dan mendeteksi kecepatannya. Nilai kecepatan asli dan jumlah kendaraan pada rekaman video dibandingkan dengan nilai yang dideteksi untuk mengukur performa sistem. Pengidentifikasian kendaraan dengan metode *Haar cascade* memperoleh nilai *precision* sebesar 96,32% dan nilai *recall* sebesar 96,91%. Pendeteksian kecepatan kendaraan yang dilakukan dengan program ini memperoleh nilai rata-rata *error* sebesar 3,37% dengan rata-rata rentang sebesar $\pm 1,72$ km/jam.

Kata-kata kunci: pendeteksian citra, pendeteksian kecepatan kendaraan, *Haar cascade*, Raspberry Pi.

Abstract

This study aimed to develop a system to identify vehicles on the highway and detect their speed. The system is developed on Raspberry Pi 2 using Python 3.7. In this study, Haar cascade is used to identify and track the vehicle objects on video records so that the speed can be measured. The program was tested using videos that show vehicle movement on the highway with various speed values. Two analysis methods were used to evaluate the performance of the system in identifying vehicles and detect their speed. The actual vehicle speed and the actual amount of vehicles on the video records were compared with the detected values to measure the system's performance. Vehicle identification using Haar cascade on this research obtains 96,32% precision and 96,91 recall. Vehicle speed detection using this program obtains 3,37% average error with interval of $\pm 1,72$ km/hours.

Keywords: image detection, vehicle speed detection, Haar cascade, Raspberry Pi.

PENDAHULUAN

Berkendara dengan kecepatan tinggi merupakan salah satu penyebab kecelakaan lalu lintas yang paling umum. Di Indonesia, kecelakaan yang disebabkan oleh faktor pengemudi merupakan penyebab kecelakaan tertinggi. Pada 2013, terdapat 103.784 kasus kecelakaan di Indonesia, dengan kecelakaan yang disebabkan pelanggaran batas kecepatan merupakan penyebab dari 13.273 atau 13%

jumlah kecelakaan tersebut [1]. Dikarenakan banyaknya kecelakaan lalu lintas yang terjadi karena pengemudi yang mengebut, maka dibutuhkan aturan yang mengatur kecepatan kendaraan di jalan raya sehingga pengemudi kendaraan dapat berkendara dengan kecepatan dalam tahap aman. Pembatasan kecepatan kendaraan merupakan unsur krusial dalam mengurangi frekuensi dan tingkat kecelakaan kendaraan. Salah satu cara untuk membatasi kecepatan kendaraan adalah dengan mendeteksi kecepatan kendaraan di jalan raya sehingga dapat ditindak dan kecelakaan dapat dicegah.

Untuk pendeteksian kecepatan, terdapat berbagai metode yang dapat digunakan. Salah satu metode yang umum digunakan adalah dengan menggunakan radar kecepatan yang memanfaatkan efek Doppler yang mengirim gelombang radio pada kendaraan yang bergerak setiap saat untuk menghitung kecepatan kendaraan tersebut. Namun, radar kecepatan mempunyai beberapa kendala teknis, seperti interferensi radio yang umum terjadi di perkotaan juga dapat menyebabkan kesalahan deteksi pada radar kecepatan [2]. Selain itu, radar kecepatan juga mempunyai harga yang cukup mahal sehingga sulit untuk digunakan secara massal. Maka, diperlukan alat pendeteksian kecepatan alternatif, salah satunya adalah dengan menggunakan pengolahan citra digital.

Pengolahan citra digital merupakan bidang dalam ilmu komputer yang sangat berkembang pada saat ini. Terdapat banyak penelitian dalam pendeteksian kecepatan yang memanfaatkan berbagai metode-metode pada pengolahan citra digital. Salah satunya adalah *frame difference*. Metode *frame difference* merupakan teknik yang dilakukan dengan mengurangi *frame* akhir dengan *frame* sebelumnya sepanjang urutan video. Metode *frame difference* dapat melakukan pendeteksian kendaraan dengan akurasi rata-rata sebesar 90,1% dan mempunyai *error* sebesar 10,58% dari 10 kali pengujian [3]. Meskipun teknik ini mempunyai nilai akurasi yang tinggi dan nilai *error* yang rendah, namun teknik ini membutuhkan memori yang tinggi untuk pendeteksian [4].

Terdapat juga metode *Gaussian Mixture Model* (GMM). GMM adalah model yang terdiri dari beberapa fungsi komponen Gaussian yang dapat digunakan untuk proses ekstraksi karena dapat beradaptasi terhadap perubahan kondisi dan cahaya dengan baik. Metode GMM yang dikombinasikan dengan *Kalman filter* dapat memperoleh akurasi sebesar 97,22% dan *recall* sebesar 94,44% untuk lalu lintas yang lengang dan juga dapat memperoleh akurasi sebesar 79,63% dan sensitivitas sebesar 88,89% untuk lalu lintas yang ramai dengan konsistensi pelacakan obyek hingga 100% [5].

Penelitian ini menggunakan metode *machine learning* bernama *Haar cascade*. *Haar cascade* merupakan metode yang dikembangkan oleh Viola dan Jones pada tahun 2001. *Haar cascade* merupakan metode pengolahan citra digital yang digunakan untuk mengidentifikasi dan mengklasifikasi obyek pada suatu citra dengan cukup baik. *Haar cascade* dapat mengidentifikasi obyek dengan mencari ciri atau *feature* bernama ciri Haar pada citra-citra yang mengandung obyek yang ingin diidentifikasi. Pencarian ciri dilakukan dengan menggunakan data-data latih yang dipelajari oleh model program. Metode ini dapat melakukan pengidentifikasian obyek citra dengan kecepatan olah sebesar 0,067 detik [6]. Hasil ini menunjukkan bahwa metode *Haar cascade* lebih cepat dari metode-metode pengidentifikasian obyek lainnya, seperti metode berbasis *neural network*, yaitu Baluja-Kanade [7] dan metode Schneiderman-Kanade [8]. Maka karena itu, tingginya kecepatan pengolahan citra yang dilakukan metode *Haar cascade* dapat menjadi metode yang baik untuk mengidentifikasi kendaraan dan melacak pergerakan kendaraan.

METODOLOGI

Metode yang digunakan pada penelitian ini adalah metode berupa penelitian dan pengembangan suatu sistem alat. Sistem alat yang diteliti dan dikembangkan ini adalah sistem yang dapat mendeteksi kecepatan kendaraan menggunakan Raspberry Pi 2 dan metode *Haar cascade*. Program ini diimplementasikan menggunakan bahasa pemrograman Python dan *library* OpenCV.

Data yang digunakan pada penelitian ini adalah 10 sampel rekaman video pergerakan kendaraan dengan variasi nilai kecepatan yang berbeda-beda. Data kecepatan asli diperoleh dengan mencatat nilai kecepatan pada *speedometer* pada kendaraan yang dikemudikan oleh penulis setiap kali melewati jalan yang sedang direkam. Penelitian dibatasi pada kecepatan kendaraan yang tidak lebih dari 65 km/jam. Perekaman dilakukan dengan posisi diagonal terhadap pergerakan mobil di jalan raya dari atas jembatan penyeberangan orang (JPO).

Sampel-sampel data yang telah dikumpulkan digunakan untuk menguji performa program. Terdapat dua hal yang dianalisis pada penelitian ini, yaitu menganalisis performa program dalam mengidentifikasi kendaraan, dan menganalisis performa program dalam mendeteksi kecepatan kendaraan tersebut. Analisis performa program dalam mengidentifikasi kendaraan dilakukan dengan membandingkan jumlah kendaraan pada rekaman video dengan jumlah kendaraan sesungguhnya pada video. Nilai jumlah kendaraan sesungguhnya dihitung secara manual. Pengukuran performa program dalam mengidentifikasi kendaraan dilakukan dengan menghitung nilai *precision* dan nilai *recall*. Nilai *precision* menunjukkan seberapa baik pengidentifikasian yang dilakukan program, sedangkan nilai *recall* menunjukkan seberapa baik program dalam mengenali obyek kendaraan yang ingin dideteksi. Nilai *precision* dan *recall* dapat dihitung menggunakan persamaan di bawah ini.

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{1}$$

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{2}$$

Penelitian ini menggunakan klasifikasi biner untuk mengelompokkan hasil pengidentifikasian kendaraan. Metrik klasifikasi biner yang digunakan pada penelitian ini terdiri dari tiga metrik, yaitu *true positive*, *false negative*, dan *false positive*. Nilai *true positive* menunjukkan jumlah kendaraan roda empat atau lebih yang berhasil dideteksi. Nilai *false positive* menunjukkan jumlah obyek bukan kendaraan roda empat yang dideteksi sebagai kendaraan. Nilai *false negative* menunjukkan jumlah kendaraan roda empat yang tidak berhasil dideteksi.

Analisis performa pendeteksian kecepatan kendaraan dapat dilakukan dengan membandingkan nilai kecepatan asli dan nilai kecepatan yang dideteksi program. Nilai kecepatan asli diperoleh dengan mencatat nilai kecepatan pada *speedometer*, sedangkan nilai kecepatan deteksi diperoleh dengan mencatat nilai kecepatan yang dideteksi oleh program setiap kali kendaraan tersebut muncul pada rekaman video. Pengukuran performa program dalam mendeteksi kecepatan kendaraan dapat dilakukan dengan menghitung nilai persentase *error* pada masing-masing sampel kecepatan. Nilai persentase *error* dari masing-masing sampel kecepatan kemudian dikumpulkan dan dihitung untuk memperoleh nilai rata-rata persentase *error* seluruh sampel kecepatan. Nilai persentase *error* tersebut dapat dihitung dengan menggunakan persamaan di bawah ini.

$$error = \left(\frac{KA - KD}{KA} \right) \times 100\% \tag{3}$$

HASIL DAN PEMBAHASAN

Data rekaman video yang telah dikumpulkan digunakan untuk menguji performa program dalam mengidentifikasi obyek kendaraan dan mendeteksi kecepatan kendaraannya. Pengujian pengidentifikasian obyek kendaraan dilakukan dengan menghitung jumlah kendaraan pada masing-masing video secara manual, kemudian nilai jumlah kendaraan tersebut dibandingkan dengan nilai jumlah kendaraan yang dideteksi oleh program. Metrik klasifikasi biner digunakan untuk mengelompokkan hasil pengujian pengidentifikasian kendaraan, yang terdiri dari *true positive*, *false positive*, dan *false negative*.

TABEL 1. Tabel klasifikasi biner dari jumlah kendaraan yang berhasil dideteksi. Nilai berjumlah 162 merupakan nilai *true positive*, nilai berjumlah 5 merupakan nilai *false negative*, dan nilai berjumlah 6 merupakan nilai *false positive*.

		Deteksi	
		Mobil	Bukan Mobil
Asli	Mobil	162	5
	Bukan Mobil	6	-

Pada tabel klasifikasi biner di atas, terlihat bahwa program dapat mengidentifikasi kendaraan roda empat atau lebih dengan sangat baik dan dengan sedikit nilai *false*. Pada pengujian ini, nilai *false* kemungkinan muncul karena kemiripan ciri citra obyek pada video dengan ciri kendaraan roda empat

yang dikenali program. Selain itu, nilai-nilai *false* juga terjadi ketika kendaraan yang telah terdeteksi terhalang oleh obyek kendaraan lainnya, sehingga program tidak menemukan kendaraan yang telah terdeteksi tersebut.

Pengukuran performa program dalam mendeteksi kendaraan dapat dilakukan dengan menghitung nilai *precision* dan *recall*. Dengan menghitung nilai *precision* dan *recall* menggunakan Persamaan 1 dan 2, ditunjukkan bahwa program sudah berhasil melakukan pengidentifikasian yang sangat baik dengan masing-masing nilai *precision* dan *recall* sebesar 96,32% dan 96,91%. Didapatkannya nilai *precision* yang sangat baik ini menunjukkan bahwa program dapat mengidentifikasi kendaraan roda empat dengan baik sehingga kemungkinan obyek-obyek selain kendaraan roda empat dideteksi itu sangat kecil. Selain itu, nilai *recall* yang sangat baik menunjukkan bahwa program dapat mengenali kendaraan roda empat dengan baik sehingga kemungkinan obyek-obyek kendaraan roda empat tidak berhasil dideteksi itu sangat kecil.

Selanjutnya, pengujian pendeteksian kecepatan kendaraan dilakukan. Nilai kecepatan kendaraan deteksi pada setiap video dibandingkan dengan nilai kecepatan asli. Perbandingan ini dilakukan dengan menghitung persentase *error*-nya sehingga performa pendeteksian yang dilakukan oleh program dapat diketahui.

TABEL 2. Hasil pendeteksian kecepatan. Nilai KA merupakan nilai kecepatan asli, yaitu nilai kecepatan pada *speedometer* kendaraan yang dicatat pada pengumpulan data. Nilai KD merupakan nilai kecepatan deteksi, yaitu nilai kecepatan yang dideteksi oleh program.

No.	Kecepatan Asli (km/jam)	Kecepatan Deteksi (km/jam)	KA-KD	Persentase Error (%)
1	30	29,90	0,10	0,33
2	40	41,81	1,81	4,53
3	50	46,74	3,26	6,52
4	60	59,12	0,88	1,47
5	60	62,71	2,71	4,52
6	45	46,74	1,74	3,87
7	65	66,10	1,11	1,69
8	60	62,71	2,71	4,52
9	65	66,10	1,11	1,69
10	40	41,81	1,81	4,53
Rata-rata Nilai Error			1,72	3,37

Pendeteksian kecepatan yang dilakukan program mendapatkan rata-rata nilai selisih yang cukup baik, yaitu sebesar $\pm 1,72$ dari nilai kecepatan asli. Hal ini ditunjukkan juga dari rata-rata nilai *error*-nya, yaitu sebesar 3,37%. Sedangkan, pendeteksian hasil *error* terkecil adalah sebesar 0,33% untuk nilai kecepatan 30 km/jam. Hasil dari rata-rata nilai *error* ini jauh lebih baik dari yang diharapkan, yaitu maksimal 10%, dan hasil ini juga menunjukkan bahwa program melakukan pendeteksian kecepatan kendaraan dengan cukup baik.

Namun, terdapat beberapa hal yang mungkin mempengaruhi persentase *error*, yaitu pengurangan *frame rate* pada saat pengolahan video dan juga ketidakstabilan kamera pada saat perekaman video sehingga membuat perhitungan program menjadi kurang akurat. Meskipun begitu, pendeteksian ini tetap dapat menghasilkan persentase *error* yang cukup baik dengan lalu lintas kendaraan yang cukup ramai.

SIMPULAN

Pada penelitian ini, telah dikembangkan sistem pendeteksian kecepatan kendaraan yang menggunakan Raspberry Pi dan metode *Haar cascade*. Berdasarkan hasil dan pembahasan, telah ditunjukkan bahwa pengembangan sistem pendeteksian kecepatan kendaraan menggunakan metode ini dapat dengan baik mengidentifikasi obyek kendaraan pada video dan juga dapat mendeteksi kecepatannya dengan cukup baik. Pengidentifikasian kendaraan menggunakan *Haar cascade* memperoleh nilai *precision* sebesar 96,32% dan memperoleh nilai *recall* sebesar 96,91%. Sedangkan, pendeteksian kecepatan kendaraan menggunakan *Haar cascade* memperoleh nilai persentase *error* sebesar 3,37% dan dapat dengan baik mendeteksi kecepatan kendaraan pada lalu lintas yang ramai

REFERENSI

- [1] Korlantas Polri, "Polantas dalam Angka Tahun 2013," *Korlantas Polri*, Jakarta Selatan, Jakarta, 2013.
- [2] M. E. Goodon, "Technical Shortcomings of Doppler Traffic Radar," in *Journal of Forensic Sciences*, Lewisville, TX, pp. 1186-1193, 1985.
- [3] F. C. Febrianto & F. Utaminingrum, "Perhitungan Kecepatan Secara Otomatis Menggunakan Metode Frame Difference Berbasis Raspberry Pi," in *Jurnal Fakultas Ilmu Komputer*, state, pp. 10968-10974, 2019.
- [4] S. S. Mohamed, N. M. Tahir & R. Adnan, "Background Modelling and Background Substraction Performance for Objecte Detection," in *6th International Colloquium on Signal Processing & Its Applications*, Melaka, pp. 236-241, 2010.
- [5] Indrabayu *et al.*, "Vehicle Detection and Tracking using Gaussian Mixture Model and Kalman Filter," in *2016 International Conference on Computational Intelligence and Cybernetics*, Makassar, pp. 115-119, 2016.
- [6] P. Viola & M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Cambridge, pp. 1-9, 2001.
- [7] H. Schneiderman & T. Kanade, "A Statistical Method for 3D Object Detection Applied to Faces and Cars," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head SC, pp. 746-751, 2000.
- [8] H. A. Rowley, S. Baluja & T. Kanade, "Neural Network-based Face Detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Toronto, pp. 22-38, 1998.

