

Perhitungan Kecepatan Kendaraan Secara Otomatis Menggunakan Metode Frame Difference Berbasis Raspberry Pi

Faris Chandra Febrianto¹, Fitri Utamingrum²

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹farischandrafebrianto@gmail.com, ²f3_ningrum@ub.ac.id

Abstrak

Keberagaman alat transportasi sebagai perantara untuk berpindah tempat merupakan bentuk dari kemajuan teknologi. Beragamnya model kendaraan menarik peminat pengguna kendaraan itu sendiri, dengan makin banyaknya pengguna kendaraan yang juga beragam, seringkali menambah oknum-oknum yang tidak taat pada peraturan. Pelanggar peraturan ini dapat merugikan pengendara lain dan beresiko menambah jumlah angka kecelakaan, di jalan umum maupun di jalur bebas hambatan/TOL. Kecelakaan yang terjadi didasari oleh pelanggar rambu lalu lintas yang seringkali tidak terdeteksi pihak keamanan karena kurangnya bukti. Pengemudi juga mampu mengemudikan kendaraan dengan kecepatan yang berbeda-beda. Namun, kebanyakan pengemudi kendaraan cenderung tidak peduli dengan batas kendaraan yang tertera dengan jelas. Dari penjelasan diatas, penulis mempunyai gagasan berupa pembuatan artefak sistem yang tidak hanya mampu berguna sebagai CCTV melainkan dapat mengetahui kecepatan kendaraan secara otomatis. Penelitian ini menggunakan metode *frame difference* untuk pendeteksian objek dengan mengetahui selisih perbedaan *foreground* dengan *background* dan mampu mendeteksi kendaraan yang melintas dengan akurasi sebesar 90,1%. Kelebihan menggunakan *frame difference* ialah hanya menggunakan warna *grayscale* untuk pemrosesan gambar sehingga lebih ringan untuk diproses. Pengujian kecepatan yang dilakukan pada sistem, menghasilkan nilai rata-rata error sebesar 89,42% dengan waktu sistem mengeksekusi rata-rata tiap video selama 10,9 detik.

Kata kunci: *frame difference, kecepatan kendaraan, Raspberry Pi*

Abstract

The variation of transportation which can be a helper for moving from another place is a part of technology advances. There are many of this transport who attracted the rider itself, with more and more variation vehicle user, sometimes increasing the trespasser of the road. These trespasser can be risk to other rider and increase crash incident, in the public road or freeway. The driver is also able to drive vehicles with different speeds but, many driver mostly do not care with speed limit that already shown clearly. From the explanation above, writer have an idea to make an artifact that does not can be used as CCTV but can known by the vehicle speeds automatic. This study about using frame difference method to detect an object by looking for differences from foreground with background, can resulting detect moving vehicles with an accurate average 90,1%. The advances of this method is only used grayscale color for processing image so it won't burden while processed. For the testing result about vehicle speeds, system has average error by 89,58% with program execution for system to run single video is 10,9 seconds.

Keywords: *frame difference, vehicle speeds, Raspberry Pi*

1. PENDAHULUAN

Kemajuan teknologi yang saat ini dapat dikatakan telah berkembang dan banyak digunakan, salah satunya disektor transportasi. Salah satunya adalah kendaraan yang sering dipakai sehari-hari. Kendaraan sendiri

merupakan alat transportasi yang memudahkan manusia dalam berpindah tempat. Saat ini di Indonesia ada begitu banyak kasus kecelakaan kendaraan bermotor baik kendaraan pribadi seperti mobil, sepeda motor, sampai dengan angkutan umum seperti bus, angkutan dalam

kota dan lain sebagainya. Terdapat beberapa faktor penyebab kecelakaan mulai dari pengemudi, kendaraan, hingga faktor jalan dan cuaca (Samudra, 2018). Angka kecelakaan pada tahun 2016 di daerah Jawa Timur sendiri sejumlah 23.103 kasus. Angka kecelakaan ini meningkat menjadi 24.196 ditahun 2017 (Fauzi, 2017).

Sistem ini merupakan pengembangan dari penelitian sebelumnya dimana peletakkan kamera untuk mendeteksi kendaraan diletakkan pada tempat yang tinggi. Peletakkan kamera yang terlampaui tinggi mengurangi efisiensi dalam hal perbaikan atau *maintenance*. Hasil penelitian yang dilakukan oleh Nur Hilman Tsani memiliki nilai ketidak pastian sebesar ± 1 km/jam dan mempunyai nilai *mean square error* sebesar 12,632. Oleh karena itu, sistem ini dibuat secara portable supaya lebih mudah peletakkannya maupun perbaikannya dengan menggunakan *raspberry pi* sebagai pemrosesan gambar sekaligus perhitungan kendaraan untuk menguji seberapa besar akurasi yang dihasilkan apabila sistem ditempatkan pada ketinggian ± 110 cm.

Dalam kasus ini, penulis mempunyai gagasan membuat sistem untuk mempermudah aparat kepolisian dalam mencegah terjadinya hal yang tidak diinginkan seperti kecelakaan dan mampu memberi sesuatu yang dapat membuat jera bagi pelanggar pengendara yang tidak disiplin. Penulis menggunakan metode *frame difference* untuk mendeteksi kecepatan kendaraan yang sedang melaju dalam video. Teknik ini menggunakan urutan frame video dan melakukan perbandingan antar *frame* gambar dengan menyeleksi *foreground* dan *background* suatu video untuk mendapatkan informasi yang menentukan adanya gerakan atau tidak (Nur Hilman Tsani, 2017). Dengan memanfaatkan kamera yang dipasang disisi jalan, sistem ini bisa digunakan secara *multifungsi*, tidak hanya sebagai CCTV atau keamanan dijalan, namun juga bisa dimanfaatkan untuk deteksi kecepatan kendaraan yang alatnya tidak harus dipegang dan diarahkan, melainkan masing-masing kendaraan yang terekam dan terdeteksi dapat diketahui kecepatannya secara otomatis.

2. METODE FRAME DIFFERENCE

Pengolahan citra digital atau digital *image processing* adalah proses memanipulasi citra secara matematis untuk meningkatkan, menganalisa dan mengekstraksi fitur dari sebuah

citra digital (Hurriyatul Fitriyah, 2016). Penelitian akan difokuskan pada akurasi ketepatan sistem dalam mendeteksi objek yang terdeteksi sebagai objek yang bergerak. Citra digital memiliki beberapa model yang dapat diolah antara lain citra 2 dimensi dengan citra video.

Salah satu metode yang dapat digunakan untuk mendeteksi objek yang bergerak selain metode *template matching* atau *tracking* adalah metode *frame difference*. Metode *frame difference* merupakan bentuk pengurangan latar belakang yang sederhana (Nur Hilman Tsani, 2017). Proses metode ini tergolong segmentasi *background subtraction* yang hanya mengurangi frame sekarang dengan frame sebelumnya sepanjang jumlah urutan frame video. Jika nilai mutlak piksel dari hasil pengurangan ini lebih besar dari threshold yang ditentukan maka akan dipertimbangkan sebagai latar depan atau objek bergerak. Secara matematis, metode *frame difference* dapat dijelaskan pada persamaan (1).

$$D(x, y, t + 1) = \begin{cases} 1 & |f(x, y, t) - f(x, y, t + 1)| > Th \\ 0 & otherwise \end{cases} \quad (1)$$

Dimana $f(x, y, t)$ adalah frame video pada saat waktu t . Sedangkan $f(x, y, t + 1)$ adalah frame berikutnya pada saat waktu $t + 1$. Nilai Th merepresentasikan nilai threshold untuk pengambilan keputusan.

Dalam pemrosesan citra digital diperlukan adanya teknik segmentasi untuk mengetahui sebuah pola atau region yang ada pada objek di video yang diproses. Berdasarkan tujuan dari segmentasi sendiri adalah menemukan karakteristik khusus yang dapat dideteksi dalam suatu citra

Beberapa pendekatan yang digunakan pengujian dalam proses segmentasi citra pada sistem adalah sebagai berikut:

1. Teknik *threshold*, segmentasi citra yang berguna untuk pengelompokan sesuai distribusi piksel pada penyusunan citra. Perbedaan citra yang telah diproses dengan segmentasi *threshold* dapat diperhatikan pada Gambar 1. Ketiga gambar berikut telah disegmentasi dengan pewarnaan 2 channel, hitam dan putih dan ditresholding untuk menghilangkan noise background.



Gambar 1. Teknik tresholding



Gambar 2. Perbedaan RGB ke Grayscale

2. *Grayscale*, teknik segmentasi *grayscale* berfungsi untuk meringankan beban kerja dari *rapsberry pi* karena hanya memiliki satu *channel* warna saja. Hasil tampilan segmentasi *grayscale* dapat dilihat pada Gambar 2.

Pencarian nilai kontur diperlukan sebagai parameter objek yang terdeteksi dengan mengetahui panjang, lebar, koordinat x dan y. Nilai kontur dapat dijelaskan sebagai bentuk dari beberapa kurva yang menghubungkan poin-poin atau titik koordinat pixel antar satu sama lain secara terus menerus atau kontinu yang dimiliki pada gambar, dan memiliki intensitas dari *channel* warna yang serupa. Kontur dapat digunakan untuk menganalisis bentuk dan mengenali objek (OpenCV, 2018).

3. KECEPATAN KENDARAAN

Objek yang bergerak dan telah diketahui kemudian dicari kecepatannya dengan mencari jarak total yang ditempuh objek dari awal bergerak sampai akhir objek bergerak dibagi dengan total waktu tempuh objek dalam melakukan perpindahan tempat. Untuk mendapatkan hasil tersebut dapat digunakan persamaan 2 seperti berikut

$$V = \frac{D}{T} \tag{2}$$

Keterangan:

V (velocity)= kecepatan objek bergerak dalam melakukan perpindahan

D (distance)= jarak yang ditempuh oleh objek

T (time) = waktu tempuh perpindahan objek

Berdasarkan rumus perhitungan kecepatan diatas, agar mendapat hasil kecepatan laju kendaraan dari rekaman video, nilai jarak yang ditempuh objek saat mulai terdeteksi sampai akhir telah ditentukan dan waktu yang diperoleh

didapat dari selisih waktu awal hingga akhir objek terdeteksi. Dari penjelasan diatas didapat persamaan 3

$$Kecepatan = \frac{Jarak\ Euclidean \times total\ frame \times 3600}{\Delta t_{detik} \times \Delta t_{frame} \times 1000} km/h \tag{3}$$

Dimana:

Jarak Euclidean = jarak yang telah ditentukan dalam penelitian

Total frame = banyaknya frame dalam satu video

Δt_{detik} = selisih waktu saat objek muncul sampai objek menghilang

Δt_{frame} = selisih waktu awal video hingga selesai

4. OBJEK TERDETEKSI

Objek yang akan diteliti berupa kendaraan berjenis mobil akan dicari dan dideteksi dari video yang telah diproses pada *rapsberry pi*. Pendeteksian objek dilakukan dengan mengambil parameter panjang, lebar, dan koordinat x dan y pada objek yang dideteksi oleh metode *frame difference*. Hasil akurasi ketepatan sistem dalam mendeteksi mobil didapat dari persamaan (4)

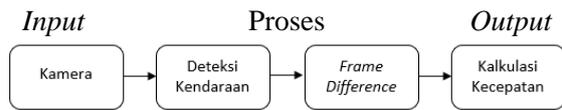
$$Akurasi = \frac{\Delta frame_{1,2}}{\Sigma frame\ objek\ terdeteksi} \times 100\% \tag{4}$$

Perhitungan akurasi objek terdeteksi digunakan untuk mencari tahu seberapa akurat sistem mendeteksi kendaraan.

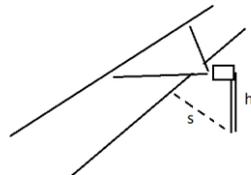
5. IMPLEMENTASI DAN GAMBARAN UMUM SISTEM

Tujuan dilakukannya penelitian ini yaitu untuk menciptakan sebuah sistem secara artefak yang mampu mendukung dalam bidang pendeteksian objek yang dihasilkan dari perekaman menggunakan kamera. Artefak sistem yang akan dikembangkan dan dibangun adalah sistem deteksi dan pengenalan dari objek dengan keluaran berupa hasil komputasi kecepatan kendaraan yang melintas didepan kamera. Diagram blok pada Gambar 3 menunjukkan bahwa sistem ini dibagi menjadi tiga bagian utama, yaitu blok *input*, proses, dan *output*.

Pemasangan kamera untuk proses *input* difokuskan pada bagaimana caranya untuk mendapatkan kondisi kendaraan melaju yang terekam. Perancangan artefak sistem dirancang secara maksimal supaya dari hasil perekaman memiliki *output* yang mampu mendeteksi kecepatan kendaraan sesuai dengan yang diharapkan.



Gambar 3. Diagram blok sistem



Gambar 4. Ilustrasi penempatan Kamera

Perancangan penempatan kamera diilustrasikan pada Gambar 4 dengan posisi kamera berjarak ±4 meter dari jalan dan tinggi dari tripod kamera setinggi 1,24 meter.

Dimana:

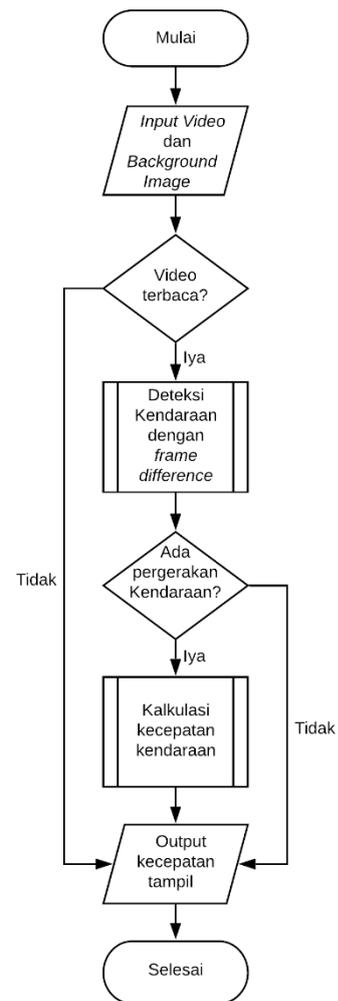
s = merupakan jarak kamera ketepi jalan

h = tinggi tripod

Kamera yang ditempatkan dibahu jalan bertujuan supaya sistem dapat diletakan dimana saja dengan batas ketinggian 1,24 meter untuk menangkap citra mobil dari depan supaya terlihat jelas di video. Peletakan kamera dapat diletakan tegak lurus maupun sejajar dengan jalur kendaraan yang melintas. Peletakan kamera digunakan diletakkan sedemikian rupa untuk melihat apakah mobil yang melaju terdeteksi sepenuhnya pada kamera. Peletakan sejauh 4 meter dari jalan digunakan supaya mobil yang terekam dikamera dapat terlihat bagian-bagiannya.

5.1 Program Utama Sistem

Proses perancangan program secara umum ditampilkan dalam bentuk flowchart pada Gambar 5 dibawah. Program dimulai dari input video dan foto sebagai langkah awal dari sistem saat pertama kali dieksekusi. Sistem mengambil video hasil rekaman kamera yang telah ditempatkan di bahu jalan sebelumnya dengan posisi kamera menghadap ke jalan. Sistem kemudian membaca bila video yang diinputkan terbaca maka program jalan ke langkah berikutnya yaitu proses deteksi pergerakan dengan bantuan metode *frame difference* deteksi *contour* yang diberi *bounding box*. Apabila video tidak terbaca oleh sistem atau tidak ada input sama sekali maka sistem langsung menghentikan proses dengan menampilkan kecepatan bernilai null.

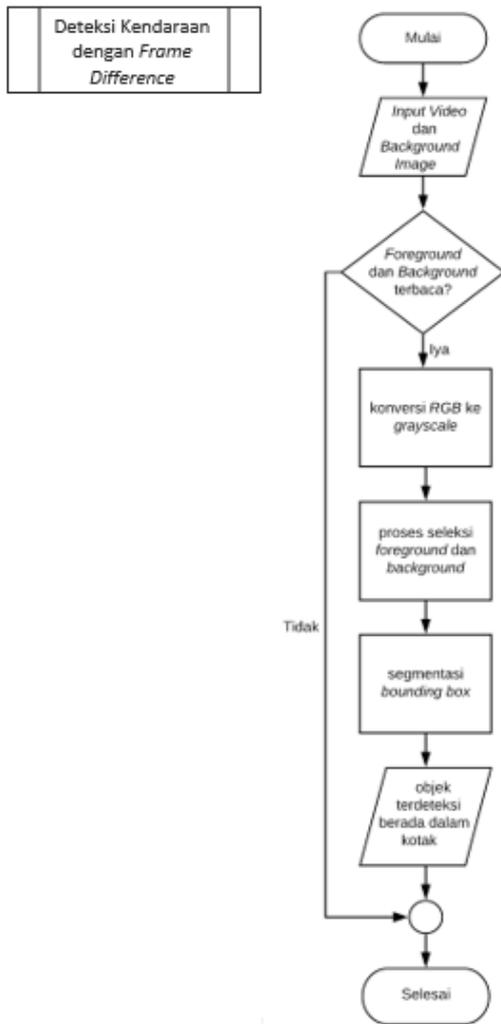


Gambar 5. Flowchart utama sistem

Setelah sistem mengeksekusi program deteksi pergerakan dan mendapat nilai data dari seleksi *frame*, sistem akan melanjutkan ke tahap selanjutnya saat menemui perubahan gerak objek. Sistem mulai menghitung kecepatan kendaraan bermotor dan hasil output kecepatan tersebut ditampilkan pada LCD.

5.2 Flowchart Deteksi Kendaraan Dengan *Frame Difference*

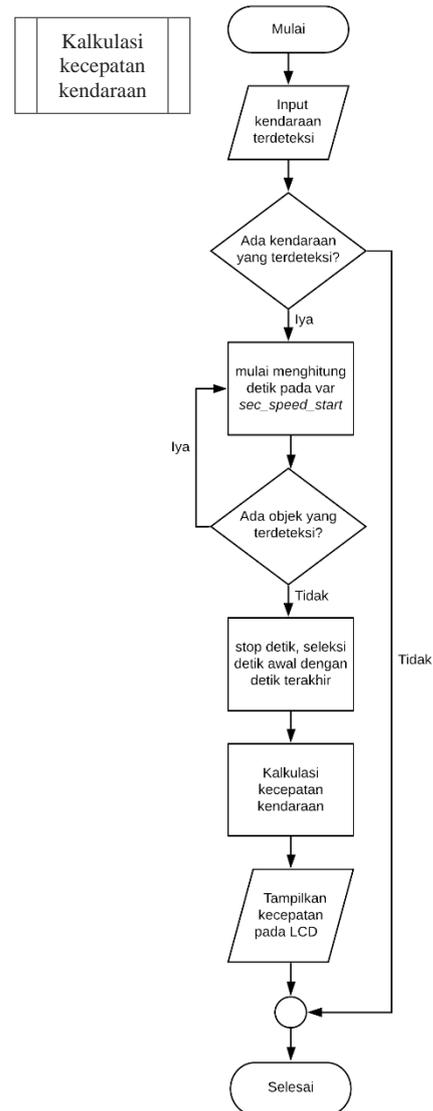
Pada diagram alir Gambar 6 dibawah, terdapat beberapa langkah yang untuk mendeteksi sebuah objek yang bergerak. Inputan foto dan video dari program utama, diinisialisasi dengan foto sebagai *background* dan video sebagai *foreground* dimana kedua variabel ini dirubah kedalam *grayscale* untuk memperingan kerja Raspberry dalam memproses segmentasi video.



Gambar 6. Flowchart deteksi kendaraan dengan frame difference

Background dan foreground yang telah dirubah warnanya, dilakukan proses seleksi dengan frame difference. Hasil yang keluar dari proses seleksi frame difference berupa objek yang terdeteksi berbeda dengan background. Objek yang telah terdeteksi selanjutnya diproses untuk mencari titik tengah dari objek tersebut, yang mana objek tersebut nantinya diberi bounding box sebagai penanda jika ada objek yang terdeteksi dan akan diproses untuk dicari kecepatannya.

Pada diagram alir Gambar 7, masuk dalam bagian sistem untuk mendeteksi kecepatan kendaraan bermotor yang terekam. Inputan dari proses ini diambil dari proses sebelumnya yang mana saat ada objek yang terdeteksi maka menjalankan detik yang diinput dari library time. Detik ini dijadikan acuan untuk selisih detik (Δt) pada Persamaan 3.



Gambar 7. Flowchart kalkulasi kecepatan kendaraan

Selisih detik didapat dari awal objek terdeteksi dalam frame dikurangi dengan detik saat objek keluar/tidak tampak dari frame. Ketika selisih waktu telah didapatkan, sistem selanjutnya melakukan perhitungan dengan Persamaan 3 dengan frame rate yang telah didapat dari total frame dalam satu video dan selisih waktu antar frame dan jarak lintasan dari objek yang telah diketahui sebelumnya. Kecepatan kendaraan yang diketahui selanjutnya ditampilkan dalam bentuk angka pada LCD 16x2.

6. PENGUJIAN DAN ANALISIS

Pengujian dengan 10 video berbeda yang mengolah warna RGB dikonversi ke grayscale dapat dilihat pada Tabel 1 dibawah ini.

Tabel 1. Hasil pengujian objek kendaraan terdeteksi

Video uji ke-	Σ frame	Σ Frame Objek Terdeteksi	Σ Hasil Deteksi Objek Selain Kendaraan Pada Sistem	Persentase Akurasi Mobil Terdeteksi
1	166	161	10	93%
2	111	111	33	70%
3	23	21	0	100%
4	71	71	3	95%
5	158	158	25	84%
6	100	99	19	80%
7	63	58	5	92%
8	59	59	3	94%
9	48	47	0	100%
10	32	32	2	93%
Rata-rata				90.1%

Bentuk konversi warna dalam salah satu video dapat dilihat pada Gambar 8. Perhitungan hasil deteksi objek dilakukan supaya akurasi hasil deteksi sistem dapat diketahui. Perhitungan akurasi deteksi objek pada sistem menggunakan persamaan 4. Jumlah keseluruhan frame akan dibandingkan dengan frame yang terdeteksi sebagai objek dimana, frame-frame yang terdeteksi objek dihitung selisihnya terlebih dahulu. Frame-frame yang diketahui terdapat objek akan terlihat ditandai dengan *bounding box* pada objek.

Pengujian kecepatan mobil pada sistem dilakukan setelah objek terdeteksi diketahui sebagai mobil dan diambil kecepatannya. Hasil akurasi sistem membaca error kecepatan kendaraan disajikan pada Tabel 2.

Tabel 2. Hasil pengujian pembacaan kecepatan mobil

Video uji ke-	Pembacaan Sistem	Pembacaan Speedometer	Perbedaan	Persentase Error
1	20.47	20	0.47	2.35%
2	8.93	10	1.07	10.7%
3	14.15	10	4.15	41%
4	22.18	20	2.18	10.9%
5	21.44	20	1.44	7.2%
6	31.18	30	1.18	3.9%
7	38.52	40	1.48	3.7%
8	22.99	20	2.99	14.9%
9	31.62	30	1.62	5.4%
10	42.35	40	2.35	5.8%
Rata-rata				10.58%

Tabel 3. Hasil pengujian waktu eksekusi program

Video uji ke-	Waktu eksekusi (s)	Waktu video asli (s)	Selisih waktu (s)
1	13	2	11
2	25	5	20
3	16	2	14
4	27	5	22
5	14	2	12
6	9	4	5
7	7	1	6
8	14	1	13
9	5	1	4
10	3	1	2
Ratar-rata			10.9

Berdasarkan dari 10 video yang berbeda, hasil perhitungan sistem memiliki beda atau error sebesar 10,58% untuk menghitung kendaraan yang terdeteksi melintas didepan kamera.

Pengujian waktu eksekusi ditujukan untuk mengetahui seberapa lama waktu yang diperlukan dalam memproses kecepatan.

Berdasarkan dari 10 video yang berbeda, hasil perhitungan sistem memiliki beda atau error sebesar 10,58% untuk menghitung kendaraan yang terdeteksi melintas didepan kamera.

Pengujian waktu eksekusi ditujukan untuk mengetahui seberapa lama waktu yang diperlukan dalam memproses kecepatan kendaraan menggunakan metode *frame difference* disatu video Tabel 3.

Waktu eksekusi program akan tampil pada layar laptop. Salah satu pengujian yang telah ditangkap layar dapat dilihat pada Gambar 8 dibawah ini.



Gambar 8. Hasil perhitungan waktu komputasi program



Gambar 9. Keluaran hasil perhitungan sistem

7. PENUTUP

Program sistem pendeteksi kecepatan kendaraan secara otomatis yang terimplementasi dan telah diuji pada penelitian ini mempunyai akurasi rata-rata sebesar 90,1% untuk pendeteksian kendaraan. Pada pengujian kesesuaian kecepatan kendaraan, rata-rata sistem menghitung kecepatan mobil yang melintas memiliki nilai error sebesar 10,58% dari 10 kali pengujian video yang berbeda. Hasil waktu komputasi sistem dalam mengolah tiap video-video pengujian, menghasilkan nilai rata-rata waktu sebesar 10,9 detik dari hasil pengolahan video menggunakan *Raspberry Pi*.

Saran yang perlu diperhatikan untuk penelitian lebih lanjut, supaya memanfaatkan kamera seperti *webcam* atau IP-Camera sebagai perekam kendaraan secara *realtime* dengan komunikasi pengiriman data menuju *raspberry pi* secara *wireless* sebagai pembanding komputasi waktu.

8. DAFTAR PUSTAKA

- A.B., T., 2004. *Computer Science Handbook*. Massachusetts: CRC Press.
- Fauzi, P., 2017. *Angka Kecelakaan Lalu Lintas di Jawa Timur Sepanjang 2017 Meningkat*, Surabaya: Tribun Jatim.
- Hurriyatul Fitriyah, S. M., 2016. MODUL AJAR KOMPUTASI CITRA DAN SUARA DIGITAL. In: Malang: s.n., pp. 1-28.
- Nur Hilman Tsani, B. D. A. L. P., 2017. Implementasi Deteksi Kecepatan Kendaraan Menggunakan Kamera Webcam dengan Metode Frame Difference. *e-Proceeding of Engineering : Vol.4*, p. 1.

OpenCV, 2018. *Contour : Getting Started*. [Online] [Accessed 06 Agustus 2018].

Samudra, A., 2018. *4 Faktor Utama Penyebab Kecelakaan di Jalan Raya*, s.l.: GridOto.