Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.201961405
Vol. 6, No. 2, April 2019, hlm. 211-218 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 30/E/KPT/2018 e-ISSN: 2528-6579

SISTEM DETEKSI JUMLAH, JENIS DAN KECEPATAN KENDARAAN MENGGUNAKAN ANALISA BLOB BERBASIS RASPBERRY PI

Gembong Edhi Setyawan[1], Benny Adiwijaya[2], Hurriyatul Fitriyah[3]

Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]gembong@ub.ac.id, [2]bennywijaya18@gmail.ac.id, [3]hfitriyah@ub.ac.id

(Naskah masuk: 11 Desember 2018, diterima untuk diterbitkan: 07 Januari 2019)

Abstrak

Penghitungan kondisi lalu lintas guna analisa kualitas jalan raya umumnya dilakukan secara manual. Hal ini tentunya membutuhkan biaya dan SDM yang tinggi serta tidak dapat dianalisa secara langsung. Dalam penelitian ini telah dikembangkan metode pengenalan jenis, jumlah dan kecepatan kendaraan secara otomatis menggunakan pengolahan citra digital. Metode yang berdasarkan analisa terhadap *BLOB (Binary Large OBject)* tersebut ditanamkan pada sistem berbasis *Raspberry Pi.* Setiap blob merupakan *connected-component* yang diperoleh dari proses *thresholding* terhadap perubahan nilai pixel dari sebuah frame dan frame rujukan dalam metode *background subtraction.* Jenis kendaraan ditentukan oleh jumlah piksel dalam *bounding-box* setiap *blob.* Jumlah kendaraan yang melaju dihitung dengan memberikan garis virtual dimana jumlahnya akan bertambah jika *centroid* dari setiap *bounding-box* kendaraan melewatinya. Kecepatan kendaraan dihitung dengan membagi jarak sebenarnya dari koordinat awal hingga garis virtual sepanjang 12 meter yang dibagi dengan waktu *centroid* tersebut untuk menempuhnya. Algoritma tersebut diimplementasikan pada sistem berbasis *Raspberry Pi* dengan input kamera yang terhubung dengan *serial monitor* untuk menampilkan output penghitungan. Pengujian akurasi deteksi jenis kendaraan yakni sepeda motor, kendaraan ringan dan berat menghasilkan akurasi 93,39%. Pengujian jumlah kendaraan menghasilkan rata-rata akurasi 93,48% untuk semua jenis kendaraan. Pengujian laju kendaraan yang dideteksi dengan dibandingkan kecepatan pada spedometer kendaraan menunjukkan akurasi 93,9%.

Kata kunci: *Kondisi lalu lintas, Analisa Blob, Raspberry Pi*

Abstract

An analysis on traffic condition usually carried out manually by visual observation. This method demands high human resource and cannot be analysed immediately. This paper present an algorithm to analyse type, number and speed of vehicles that passing by a road automatically using BLOB (Binary Large Object) analysis. Each blob is a connected-component as a result of thresholding after background subtration process. Type of vehicles was determined by measuring pixel number of blob's bounding box. Number of vehicles was determined by drawing virtual line where the number was increased once a centroid of bounding box passed it. Speed of vehicles was determined using basic speed formula where 12 meters of actual distance between the beginning coordinate and virtual line was divided by time to travel it. The algorithm was embedded in Raspberry Pi where videos were acquired using attached web camera. The analysis result was shown in connected serial monitor. Testing on vehicles' type detection (motorcycle, light vehicle, heavy vehicle) result accuracy of 93.9%, number of vehicles result accuracy of 93.48%, whilst speed of vehicles result accuracy of 93.9%.

Keywords: *Traffic condition, Blob analysis, Raspberry Pi*

1. PENDAHULUAN

Dengan semakin bertambahnya jumlah kendaraan, maka dipastikan kondisi lalu lintas di jalan raya menjadi semakin padat. Kepadatan tersebut dapat berimbas pada kualitas jalan. Tingkat Pelayanan Jalan merupakan kualitas sebuah jalan dalam melayani arus lalu-lintas. Dalam Manual Kapasitas Jalan Raya (MKJI) dari Dirjen Bina Marga menyebutkan bahwa kecepatan dan derajat kejenuhan dapat menjadi indikator Tingkat Pelayanan Jalan tersebut. Derajat Kejenuhan sendiri merupakan perhitungan volume lalu lintas maksimum (Satuan mobil penumpang per jam) dibagi dengan kapasitas jalan (satuan mobil penumpang per jam). Menurut MKJI, derajat kejenuhan yang baik adalah kurang dari 0.75.

Analisa Derajat Kejenuhan Jalan membutuhkan penghitungan terhadap jumlah kendaraan yang melintas. Setiap jenis jalan memiliki daya tampung maksimal, sehingga jika nilai daya tampung sudah buruk maka perlu dilakukan evaluasi lanjutan. Penghitungan jumlah kendaraan yang melintas juga membutuhkan detil jenis kendaraan berikut kecepatannya terutama untuk perhitungan Karakteristik arus lalu lintas.

Penghitungan jumlah, jenis dan kecepatan kendaraan yang melintas dapat dilakukan secara manual, yakni dengan mengirim petugas untuk menghitung jumlah kendaraan yang melintas. Penelitian oleh (Merentek, 2016) melakukan analisa kualitas jalan Manado-Bitung dengan penghitungan manual selama 7 hari dengan interval 15 menit selama jam 06.00-21.00. Pencatatan manual tersebut dibantu peralatan berupa pita ukur, *speed gun*, *stop watch*, *counter*, dan formulir survei. Analisa kualitas jalan menggunakan pengamatan secara manual juga dilakukan oleh (Reswantomo, 2007) yang mengamati jalan di pusat Kota Depok. Pengamatan dilakukan pada pagi dan sore hari yang merupakan jam padat lalu lintas. Pengamatan secara manual tentunya membutuhkan sumber daya manusia dan tidak efektif.

Beragam penelitian telah dilakukan untuk mendeteksi kendaraan di jalan secara otomatis. (Sutrisno, dkk., 2015) menggunakan *blob analysis* dari hasil *background subtraction* pada video lalu lintas. Hasil *blob* terdeteksi telah di-*improve* menggunakan *Second-Derivative in Gradient Direction* (SDGD) untuk meningkatkan akurasi pengenalan kendaraan. Penelitian (Dewantoro, dkk., 2015) juga melakukan analisa pengaruh parameter pada setiap langkah pemrosesan citra hasil *background subtraction* guna mendeteksi kendaraan yang melintas. Penelitian lain mendeteksi kendaraan menggunakan *Haar-Cascade* dengan menggunakan data latih berupa citra kendaraan (Lazaro, dkk, 2017), (Irfan, dkk., 2017).

Jenis kendaraan umumnya dikenali menggunakan metode klasifikasi. Pada penelitian (Irfan, dkk., 2017) menggunakan *Multi-Layer Perceptron (MLP)* untuk mengkategorikan kendaraan ke dalam kelas mobil, bus atau truk. Sedangkan (Wibowo, dkk., 2013) menggunakan *Fuzzy-C-Means (FCM)* untuk melabeli kendaraan yang terdeteksi menjadi kelas mpv, motor, bus, sedan dan truk. Kedua metode tersebut membutuhkan keberadaan data latih yang banyak untuk mendapatkan akurasi yang baik.

Penghitungan kecepatan kendaraan berdasarkan video telah dilakukan oleh (Fajriyah, 2016) dan (Sadewo, dkk., 2015) dimana keduanya menggunakan *frame-frame* pada video untuk menghitung lama waktu yang dicapai oleh sebuah kendaraan dari satu tempat ke tempat lain. Keduanya menggunakan koordinat pixel yang ditentukan sebelumnya untuk dijadikan titik mula dan titik akhir penghitungan waktu. Waktu dihitung sejak titik pusat dari *blob* kendaraan terdeteksi mencapai koordinat mula hingga mencapai koordinat akhir. Jarak sebenarnya antara kedua pixel ditentukan dengan rasio terhadap jarak sebenarnya. Kecepatan kendaraan tersebut kemudian dihitung dengan rumus kecepatan sederhana yakni jarak tempuh dibagi dengan waktu tempuh.

Tidak hanya berupa penelitian teoritis, beberapa penelitian telah mengimplementasikan metode pemantauan kondisi jalan raya pada perangkat computer. (Rahmadina, F., dkk 2016) telah membuat sistem implementatif berbasis *Raspberry Pi* untuk menghitung jumlah kendaraan yang melewati sebuah jalan raya menggunakan metode *background subtraction.* Penelitian tersebut menempatkan kamera pada posisi samping menghadap ke jalan. Citra yang ditangkap oleh kamera kemudian diproses secara *realtime* pada *Raspberry Pi* dan hasil pendeteksian kendaraan dikirim ke server untuk ditampilkan pada sistem berbasis web.

Sistem penghitung jumlah dan kecepatan kendaraan lain juga telah dikembangkan oleh (Resilawati, R., 2015). Sensor yang digunakan adalah 3 buah sensor ultrasonik yang dipasang secara horizontal. Jika sensor 1 terhalangi maka *counter* akan diaktifkan untuk menunjukkan sebuah kendaraan tengah melintasi jalan. Waktu tempuh dihitung menggunakan *timer* antara sensor 2 dan 3 yang aktif saat terhalang kendaraan. Kecepatan kendaraan dihitung dengan membagi jarak tempuh yang tetap antara sensor 2 dan 3 dibagi waktu tempuh anta keduanya.

Berdasarkan kajian terhadap penelitian-penelitian diatas, maka penelitian ini membangun sistem penghitungan jumlah, jenis dan kecepatan kendaraan yang juga berbasis pemrosesan citra. Kamera memiliki kelebihan dimana pengambilan data dapat dilakukan secara non-kontak. Kamera juga dapat dipasang pada ketinggian dan lokasi tertentu di jalan yang tidak memungkinkan dijangkau oleh manusia, contoh di atas jalan. Metode utama pendeteksian yang digunakan dalam penelitian ini adalah *background subtraction*. Dengan metode tersebut, kendaraan dengan mudah dideteksi tanpa terpengaruh pencahayaan luar ruangan yang beragam dibandingkan saat menggunakan metode segmentasi objek berdasarkan intensitas. Metode tersebut ditanamkan pada *Raspberry Pi* yang mengolah inputan gambar dari kamera.

2. PERANCANGAN DAN IMPLEMENTASI

Secara umum, algoritma yang dibangun memiliki beberapa tahapan yakni *background subtraction*, segmentasi, pemfilteran dan ekstraksi fitur. Tahap *background subtraction* dan segmentasi bertujuan untuk mendeteksi keberadaan BLOB kendaraan. *Binary Large Object* dalam penelitian ini merujuk kepada satuan objek terdeteksi yakni kendaraan baik kendaraan berjenis motor, ringan dan berat. Penentuan jumlah, jenis dan kecepatan kendaraan dilakukan dengan menganalisa beragam fitur dari *blob* yang terdeteksi.

Algoritma tersebut diprogram dalam *Raspberry Pi* yang mengambil video kondisi jalan menggunakan kamera *webcam*. Video diambil dalam resolusi ukuran 320×240 piksel. Resolusi tersebut dipilih dengan pertimbangan kejelasan gambar dan

kemampuan mikrokomputer *Raspberry Pi* untuk memproses. Kamera diletakkan setinggi 5 meter dari jalan dengan posisi tetap atau *fix* diletakkan pada jembatan penyeberangan. Sudut kemiringan kamera adalah 30° dari sisi horizontal.

2.1. Deteksi *Blob*

Sistem penghitung kondisi jalan yang dibangun dimulai dengan *background subtration* antara *frame* citra *grayscale* saat ini dengan citra *grayscale* referensi saat kondisi jalan kosong. Hasil *background subtraction* adalah citra keabuan skala 0-255. Hasil tersebut kemudian disegmentasi untuk mendapatkan *blob* kendaraan. Beberapa filter diterapkan untuk menghilangkan kesalahan pendeteksian yang umumnya muncul setelah hasil segmentasi. Deteksi *blob* memiliki urutan langkah yakni: (1) *Bacground subtraction*, (2) Segmentasi, (3) *Filtering*, (4) Pemberian label untuk masing-masing *blob*.

a. Background subtraction

Setiap *frame* dari video yang diambil akan dilakukan konversi dari ruang warna RGB ke *grayscale* menggunakan metode *luma* (Gambar 1(a)). Hasil konversi tersebut akan dikurangkan dengan *frame* referensi (Gambar 1(b)) sehingga menghasilkan citra hasil pengurangan. Nilai hasil pengurangan berada pada rentang 0-255 (Gambar 1(c)). Selain *robust* terhadap invariansi pencahayaan, *background subtraction* juga dipilih karena menggunakan operasi pengurangan sederhana yang rendah dalam waktu komputasi.

(a) (b) (c)

Gambar 1. Proses *Background Subtraction*; (a) Citra terkini, (b) Citra referensi, (C) Hasil *background subtration*

b. Segmentasi untuk mendeteksi BLOB kendaraan

Citra hasil *background subtraction* kemudian disegmentasi untuk mendapatkan *region of interest* berupa kendaraan saja. Proses *thresholding* dipilih sebagai metode segmentasi karena sederhana dan membutuhkan sedikit memori karena menggunakan kondisi *if-else* saja saat deprogram. Batas yang ditentukan adalah 20 (dari skala 0-255) yang dipilih secara manual. Gambar 2 menujukkan proses segmentasi dimana Gambar 2(b) merupakan hasil proses *thresholding* dari Gambar 2(a). Hasil segmentasi adalah citra biner dimana nilai 1 diberikan pada piksel dengan nilai keabuan 20-255, dan nilai 0 untuk selain *range* tersebut.

(a) (b)

Gambar 2. Proses Segmentasi; (a) Citra keabuan, (a) Citra biner hasil segmentasi

c. Filter Berdasarkan Bentuk dan Ukuran

Umumnya proses segmentasi akan menghasilkan kesalahan binerisasi, yakni piksel mobil yang dikenali sebagai *background* dan sebaliknya. Untuk menghilangkain kesalahan tersebut, maka diterapkan *morpohological filter* yang menggunakan bentuk sebagai *template* pembanding. Filter tersebut menggunakan operator logika (AND, OR) antara piksel citra dengan piksel *structure element* sebagai *template* pembanding untuk mengubah nilai biner. Bentuk dari *structure element* yang digunakan adalah *square* berukuran 3×3 yang dipilih secara manual. Penelitian ini menerapkan filter *Opening* dan *Closing* untuk menghilangkan kesalahan binerisasi. Hasil pemfilteran dari Gambar 2(b) ditunjukkan pada Gambar 3(a). Citra hasil pemfilteran menunjukkan bentuk mobil yang lebih utuh dimana nilai 0 pada piksel mobil berubah menjadi 1 dan nilai 1 pada *background* berubah menjadi 0.

Pemfilteran selanjutnya dilakukan berdasarkan ukuran. Setiap kumpulan piksel putih atau yang disebut dengan *blob* dihitung jumlah pikselnya. *Blob* yang memiliki jumlah piksel kurang dari batas tertentu tidak akan dianggap sebagai kendaraan. Contoh hasil pemfilteran berdasarkan ukuran piksel 200 terlihat pada gambar 3(b) dimana hanya *blob* berukuran diatas batas yang akan dianalisa lebih lanjut. Batas minimum jumlah piksel per *blob* yang dianggap sebagai kendaraan dilakukan melalui Pengujian Jenis Kendaraan.

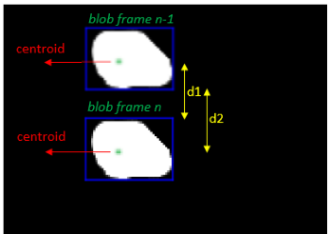
(a) (b)

Gambar 3. Proses Filter; (a) Citra biner, (b) Hasil erosi, (c) Hasil dilasi

d. Pemberian label kendaraan

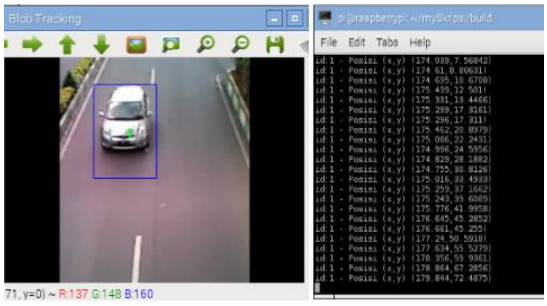
Setiap *blob* yang terdeteksi kemudian dihitung *centroid* dan koordinat *bounding-box* nya. Di jalan raya yang dapat dilalui lebih dari satu kendaraan pada saat yang sama, maka dibutuhkan sebuah pemberian label *blob* yang dapat membedakan apakah sebuah *blob* antar *frame* video merupakan satu kendaraan

yang sama atau kendaraan yang berbeda. Penentuan pelabelan kendaraan ini dilakukan dengan menghitung jarak *d1* dan *d2* dimana *d1* adalah jarak *centroid* dari *frame* saat ini ke koordinat *bounding-box* di *frame* sebelumnya. Sedangkan *d2* adalah jarak *centroid* dari *frame* sebelumnya ke koordinat *bounding-box* saat ini. Nilai batas *d1* dan *d2* ditentukan sebagai 40. Dimana jika salah satu dari jarak tersebut kurang dari 40 maka dianggap sebagai satu kendaraan yang sama, dan sebaliknya. Ilustrasi jarak *d1* dan *d2* tersebut diperlihatkan pada Gambar 4.



Gambar 4. *Blob Tracking* dengan Perhitungan *d1* dan *d2*.

Jika dilabeli sebagai satu kendaraan maka posisi *blob* kendaraan tersebut disimpan dan diperbarui seperti ditunjukkan pada Gambar 5.



(a) (b)

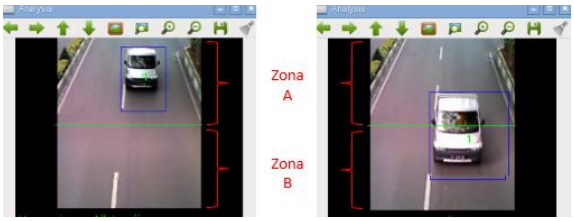
Gambar 5. Pelacakan Posisi Kendaraan; (a) Kendaraan terdeteksi, (b) Posisi kendaraan diperbarui.

2.2. Algoritma Penghitung Jumlah, Jenis dan Kecepatan

Setelah ditemukan *blob* kendaraan, maka luas area dan titik pusat *blob* tersebut dihitung untuk dilakukan analisa terhadap jenis dan kecepatan. Setiap metode yang dipilih selalu mementingkan komputasi yang sederhana agar proses komputasi di mikrokomputer *Raspebrry Pi* dapat berjalan lebih cepat. Kecepatan analisa dibutuhkan karena kondisi lalu lintas jalan raya berubah dengan sangat cepat.

a. Jumlah Kendaraan

Sebuah garis virtual dibuat pada koordinat $(260, 120)$ untuk mengamati apakah ada kendaraan yang melintas melewati garis tersebut. Lokasi sebelum garis virtual disebut Zona A, sedangkan lokasi sesudah arus virtual disebut Zona B. Jika *centroid* dari sebuah *blob* yang telah melewati garis virtual maka akan ditambahkan dalam variabel Jumlah Kendaraan. Ilustrasi penghitungan kendaraan berikut hasil *increment* Jumlah Kendaraan dalam porgram ditampilkan pada Gambar 5. Pada Gambar tersebut, garis virtual kedua ditunjukkan sebagai warna hijau. Gambar 5(a) menunjukkan sebuah kendaraan masuk ke zona A sebelum garis virtual. Sedangkan Gambar 5(b) menunjukkan sebuah kendaraan melewati garis virtual dan masuk zona B untuk kemudian menambah nilai Jumlah Kendaraan.



(a) (b)

Gambar 5. Penghitungan Jumlah Kendaraan; (a) Kendaraan terdeteksi sebelum garis vritual (b) Kendaraan melewati garis virtual

b. Jenis kendaraan

Jenis kendaraan ditentukan berdasarkan luas area *bounding-box* dari setiap *blob* kendaraan yang terdeteksi. Luas area dihitung menggunakan jarak antara posisi minimum dan maksimum dari koordinat sumbu x dan y. Gambar 6 menunjukkan perhitungan luas area dari 2 *blob* kendaraan yang kemudian tercatat pada variabel Area untuk masing-masing label kendaraan (id). Penentuan besar luas area untuk setiap jenis kendaraan yakni sepeda motor, kendaraan ringan (contoh: mobil) dan kendaraan berat (contoh: bis, truk) dilakukan melalui Pengujian Jenis Kendaraan.

(a) (b)

Gambar 6. Penghitungan Luas Area Kendaraan, (a) Ilustrasi pengukuran luas, (b) Luas area kendaraan diukur

c. Kecepatan kendaraan

Kecepatan kendaraan dihitung dengan menggunakan rumus kecepatan sederhana yakni jarak dibagi waktu tempuh. Dalam penelitian ini, jarak adalah ukuran sebenarnya dari ujung frame hingga ke garis virtual $(y_2 - y_1)$yang diukur secara manual dan ditemukan berjarak 12 meter. Sedangkan waktu tempuh adalah *timestamp* saat *centroid* kendaraan terdeteksi hingga *centroid* tersebut

melewati garis virtual. Gambar 7 menunjukkan ilustrasi penghitungan kecepatan dari sebuah kendaraan yang terdeteksi.

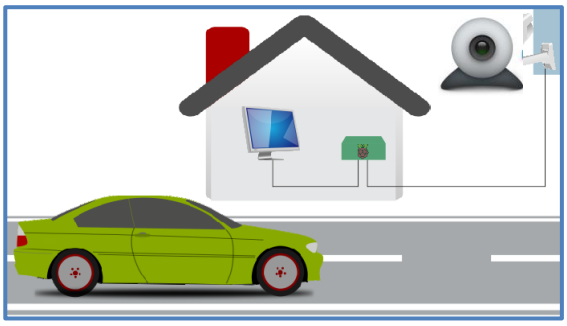
(a) (b)

Gambar 7. Penghitungan Kecepatan Kendaraan; (a) Kendaraan awal terdeteksi, (b) Kendaraan melewati garis virtual berjarak sebenarnya 12 meter.

2.2. Implementasi pada Raspberry Pi

Algoritma yang sudah dibangun tersebut kemudian ditanamkan pada sistem pendeteksi berbasis *Raspberry Pi 2 Model B*. Algoritma diprogram menggunakan bahasa pemrograman *python* dan *library openCV*.

Sebuah kamera *webcam* dengan resolusi 0.5MP dan 30 *fps frame-rate* digunakan untuk menangkap video dari kondisi jalan. Kamera tersebut terkoneksi ke *Raspberry* via USB port. Untuk mengambil gambar, webcam tersebut diletakkan di atas jembatan layang sehingga memudahkan dalam pemasangan dan dapat menghasilkan video dengan posisi yang stabil. Output dari *Raspberry* yakni berupa *streaming* video dan penghitungan jumlah, jenis, dan kecepatan kendaraan ditampilkan ke serial monitor menggunakan kabel HDMI. Mikrokomputer *Raspberry* mendapatkan catu daya dari baterai *portable*. Gambar 8 menunjukkan ilustrasi sistem pendeteksi yang dibuat.



Gambar 8. Ilustrasi Implementasi Sistem Pendeteksi

Pada serial monitor diberi antar muka berisi *streaming* video dan keterangan hasil penghitungan jumlah, jenis dan kecepatan dari kendaraan yang melewati jalan raya. Informasi tersebut seperti yang diperlihatkan pada Gambar 9. Dapat dilihat pada Gambar 9 bahwa sistem juga dapat menghitung jumlah kendaraan yang melintas secara bersamaan.

Gambar 9. Antarmuka Sistem Pada *Serial monitor*

3. PENGUJIAN DAN ANALISA

Pengujian pertama dilakukan terhadap jumlah minimum piksel *blob* untuk dapat dialabeli sebagai kendaraan. Pengujian selanjutnya adalah terhadap jumlah piksel *blob* dari masing-masing jenis kendaraan. Pengujian selanjutnya adalah pada akurasi penghitungan jumlah dan kecepatan kendaraan. Pengujian dilakukan dengan mengambil data video sebanyak 5 kali dengan masing-masing durasi antara 120-140 detik. Video diambil pada kondisi pencahayaan siang hari yang cerah.

3.1. Batas Jumlah Piksel *Blob* Kendaraan

Untuk menenentukan jumlah piksel minimum dari sebuah *blob* untuk ditetapkan sebagai sebuah kendaraan maka dilakukan pengujian variasi nilai batas. Pengujian dilakukan pada 5 video dengan kenaikan batas setiap 5 piksel dari 5 hingga 50. Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian Batas Piksel *Blob*

Thresh.	**Akurasi (%) Video ke-**					**Rata-rata Akurasi(%)**
	1	2	3	4	5	
5	75	56,1	70,2	67,9	62,9	66,42
10	77,7	65,8	76,5	81,1	77,4	75,75
15	88,8	73,1	93,6	92,4	87,1	87,04
20	**94,4**	**73,1**	**91,4**	**94,3**	**91,9**	**89,07**
25	91,6	73,1	91,4	94,3	90,4	88,95
30	83,3	73,1	91,4	94,3	96,7	87,82
35	88,8	78,8	89,3	84,9	96,7	87,59
40	75	73,1	82,9	75,4	95,1	80,35
45	52,7	63,4	74,4	67,9	83,8	68,44
50	36,1	53,6	61,7	49,1	74,1	54,92

Dari Tabel 1 diatas didapatkan akurasi tertinggi yakni sebesar 89,07% pada batas jumlah piksel 20. Nilai tersebut digunakan pada proses pemfilteran berdasarkan ukuran. Nilai tersebut merupakan nilai tertinggi namun masih di bawah 90%, sehingga akuasi tersebut dapat menurunkan akurasi pada penghitungan jenis dan jumlah kendaraan.

3.2. Jumlah Piksel untuk Setiap Jenis Kendaraan

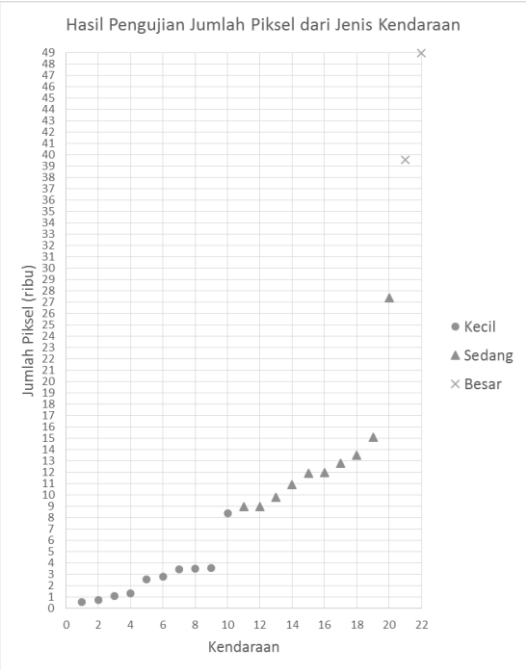
Pengujian dilakukan dengan mengukur jumlah piksel dari setiap kendaraan yang teridentifikasi. Selama pengujian ditemukan 10 sepeda motor, 10 kendaraan ringan, dan 2 kendaraan berat. Hasil pengukuran tersebut dapat dilihat pada gambar 10. Jumlah piksel ditunjukkan dalam satuan ribu. Dari pengujian tersebut dapat dikelompokkan bahwa kendaraan sepeda motor memiliki jumlah piksel < 9,000, kendaraan sedang 9,000 – 39,000, dan kendaraan besar > 39,000.

Batas jumlah piksel dari setiap kendaraan tersebut tentunya hanya berlaku pada peletakan kamera yang ditetapkan dalam penelitian ini. Ketinggian dan sudut kamera yang berbeda memerlukan penghitungan ulang. Namun demikian, hasil pengujian menunjukkan bahwa jenis kendaraan dapat dibedakan berdasarkan jumlah pixel dari setiap *blob.*

3.3. Pengujian Jenis Kendaraan

Penetapan batas antar jenis kendaraan pada pengujian 3.2 tersebut kemudian diuji pada 5 buah video. Hasil pengujian ditunjukkan pada Tabel 2. Rata-rata akurasi untuk mendeteksi jenis kendaraan sepeda motor adalah 85.86%, kendaraan ringan 94.56% dan kendaraan berat 100%. Rata-rata akurasi penghitungan jenis kendaraan adalah 93.39%.

Motor memiliki akurasi pengenalan jenis kendaraan yang paling rendah karena jumlah piksel hasil *background subtraction* yang sangat kecil dibanding batas jumlah piksel blob kendaraan yang kemudian menjadi hilang ketika diberikan filter.



Gambar 10. Hasil pengujian jumlah piksel setiap jenis kendaraan.

Tabel 2. Hasil Pengujian Jenis & Jumlah Kendaraan

Vid.	Jenis	Jumlah Kendaraan		Akurasi (%)
		Sebenarnya	Sistem	
1	Motor	28	26	92,85
	Ringan	8	8	100
	Berat	0	0	100
	Total	**36**	**34**	**97,62**
2	Motor	27	18	66,66
	Ringan	13	11	84,61
	Berat	1	1	100
	Total	**41**	**30**	**83,76**
3	Motor	34	30	85,18
	Ringan	13	13	100
	Berat	0	0	100
	Total	**47**	**43**	**96,08**
4	Motor	37	35	94,59
	Ringan	15	14	93,33
	Berat	1	1	100
	Total	**53**	**50**	**95,98**
5	Motor	23	20	71,42
	Ringan	39	37	97,43
	Berat	0	0	100
	Total	**62**	**57**	**93,94**
	Rata-rata akurasi			**93,48**

3.4. Akurasi Jumlah Kendaraan

Hasil pengujian pada jumlah kendaraan yang melintas dapat dilihat pada Tabel 2. Dari Tabel tersebut didapati rata-rata akurasi penghitungan jumlah kendaraan adalah sebesar 93,48%. Nilai akurasi terkecil dimiliki oleh video 2 dimana hanya 18 sepeda motor yang terdeteksi dari total 27. Hal ini menunjukkan bahwa untuk mendapatkan sistem pendeteksian jenis dan jumlah kendaraan dengan akurasi tinggi sangat bergantung pada metode pemfilteran. Umumnya *blob* sepeda motor sangat kecil hamper menyerupai perubahan cahaya dan objek kecil di pinggir jalan yang bergerak.

3.5. Akurasi Kecepatan Kendaraan

Pengujian kecepatan kendaraan dilakukan dengan membandingkan kecepatan yang dihitung oleh program dengan kecepatan kendaraan sebenarnya. Kendaraan yang digunakan adalah motor dimana nilai kecepatan sebenarnya didapatkan dari indicator *speedometer*. Kendaraan tersebut melaju dengan kecepatan konstan pada jalan yang dipantau oleh sistem. Hasil pengujian kecepatan dari 5 kali percobaan dapat dilihat pada Tabel 3. Rata-rata akurasi penghitungan kecepatan kendaraan adalah 93,9%.

Tabel 3. Hasil Pengujian Kecepatan Kendaraan

Data	Kecepatan sebenarnya (Km/Jam)	Kecepatan Hasil Program (Km/Jam)	Akurasi (%)
1	40	43	92,5
2	50	51	98
3	50	53	94
4	60	53	95
5	60	66	90
Rata-rata akurasi			93,9

Akurasi penghitungan kendaraan tersebut adalah tinggi dan hanya memberikan rata-rata error 4 Km/jam. Sehingga penggunaan garis virtual dan rumus kecepatan sederhana terbukti berhasil menghitung kecepatan kendaraan.

4. KESIMPULAN

Dalam penelitian ini telah dikembangkan sistem yang dapat mendeteksi jenis, jumlah dan kecepatan kendaraan secara otomatis sehingga memudahkan analisa kondisi jalan secara langsung dan otomatis. Akurasi yang diperoleh adalah tinggi sehingga memungkinkan untuk diimplementasikan secara nyata. Sistem dapat dikembangkan lebih lanjut untuk bisa berkomunikasi secara jarak jauh untuk memungkinkan pengamatan secara terpusat dan *live.*

5. DAFTAR PUSTAKA

MERENTEK, T. G. S., SENDOW, T. K. & MANOPPO, M. R. E. 2017. Evaluasi Perhitungan Kapasitas Menurut Metode KMJI 1997 dan Metode Perhitungan Kapasitas dengan Menggunakan Analisa Perilaku Karakteristik Arus Lalu Lintas pada Ruas Jalan Antar Kota (Studi Kasus manado – Bitung). *Jurnal Sipil Statik*, 4(3), 187-201.

RESWANTOMO, B. S. 2008. Identifikasi Kinerja beberapa Ruas jalan Raya Utama menuju Pusat Kota Depok Tahun 2007. *Skripsi*. Teknik Sipil FT UI.

SUTRISNO, CHOLISSODIN, I., CHRISTIANTI, R., DEWI, C., HIDAYAT, N., 2015 Segmentasi Kendaraan Menggunakan Improve Blob Analysis (BA) Pada Video Lalu Lintas. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, (2(1), 67-72.

DEWANTORO, A.K., IWUT, I., SUSATIO, E., 2015. Simulasi dan Analisis Sistem Penghitung Kepadatan Lalu Lintas dan Klasifikasi Kendaraan Berbasis Webcam dengan Metode Background Subtraction. *E-Proceeding of Engineering*. 2(2). 2833-2840.

LAZARO, A., BULIALI, J. L., AMALIAH, B., 2017. Deteksi Jenis Kendaraan di jalan Menggunakan OPENCV. *Jurnal Teknik.* 6(2). 2337-3520.

IRFAN, M., SUMBODO, B. A. A., CANDRADEWI, I., 2017. Sistem Klasifikasi Kendaraan Berbasis Pengolahan Citra Digital dengan Metode Multilayer Perceptron. *Indonesian Journal of Electronics and Instrumentation System (IJEIS).* 7(2). 139-148.

FAJRIYAH, F., 2016, Pengembangan Perangkat Lunak Deteksi Kecepatan kendaraan Bergerak Berbasis Pengolahan Citra Digital. *Skripsi*. FMIPA Institut Teknologi Sepuluh Nopember, Surabaya

SADEWO, S. S. SUMIHARTO, R., CANDRADEWI, I., 2015. Sistem Pengukur Kecepatan Kendaraan Berbasis Pengolahan Video. *Indonesian Journal of Electronics and Instrumentation System (IJEIS)*. 5(2). 177-186.

RAHMADINA, F., ZAINI, 2016. Sistem Informasi kepadatan Lalu Lintas berbasis Raspberry Pi PC Board. *Jurnal Nasional Teknik Elektro*. 5(1). 151-155.

RESILAWATI, R., 2015. Penghitung Jumlah Kendaraan dan pengukur Kemacetan Menggunakan Sensor Ultrasonik Berbasis Arduino UNO. *Skripsi*. D3 Elektronika dan Instrumentasi SV UGM, Yogyakarta.