

## Klasifikasi Kepadatan Lalu Lintas Berbasis *Multitracking Object* Dengan Menggunakan Metode *Particle Filter*

### Classification of Traffic Density Based On *Multitracking Object* Using *Particle Filter* Method

Hafidz Al Djohari <sup>1</sup>, Tjokorda Agung Budi W, ST., MT <sup>2</sup>, Bedy Purnama, S.Si., MT <sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung 40257, Indonesia.

<sup>1</sup>djohari.hafidz@gmail.com, <sup>2</sup>cokagung@telkomuniversity.ac.id, <sup>3</sup>bedy.purnama@gmail.com

Abstrak— Penentuan kepadatan lalu lintas menggunakan kamera saat ini digunakan hanya sebatas pengawasan secara visual. Namun sebenarnya terdapat parameter yang menjadi acuan baku untuk menentukan karakteristik kepadatan arus lalu lintas salah satunya kecepatan rata-rata tiap kendaraan. Hal ini sudah diatur dalam Peraturan Menteri Perhubungan Nomor : KM 14 Tahun 2006. Untuk menentukan kecepatan kendaraan digunakan metode *object tracking* untuk mendapatkan jumlah frame yang dilalui tiap objek dari batas yang sudah ditentukan. *Particle Filter* digunakan sebagai metode *object tracking* karena dapat memperkecil wilayah telusur objek terutama dengan menggunakan *particle filter* berbasis warna. Selain itu juga dibutuhkan metode untuk mendeteksi kendaraan yang digunakan sebagai acuan dari *particle filter*. Metode yang digunakan untuk menangani masalah pendeteksian objek menggunakan *Gaussian Mixture Model*, metode ini memiliki hasil keluaran berupa citra hitam putih dengan citra putih merupakan citra yang dideteksi sebagai objek. Partikel yang menempel pada tiap objek digunakan untuk mendapatkan titik *centroid*. Titik *centroid* dari tiap objek diambil sebagai acuan posisi. Dengan menggabungkan dua metode ini didapatkan tingkat akurasi dalam menghitung jumlah kendaraan sebesar 91,22% dan akurasi dalam menentukan kecepatan kendaraan sebesar 94,41% sedangkan justifikasi tingkat kepadatan memiliki kecocokan yang tinggi dan tingkat kecocokan terhadap pelayanan arus lalu lintas sebesar 66,67% dari video yang diujikan.

Kata kunci : kepadatan lalu lintas, jalan tol, *object tracking*, *vehicle tracking*, *vehicle counting*, *particle filter*.

*Abstract— Determination of the density of traffic using the current camera used was limited to visual inspection. But actually there are parameters that become the standard reference for determining the characteristics of traffic density is a average speed of each vehicle. It is already stipulated in the Regulation of the Minister of Transportation No. KM 14 of 2006. To determine the vehicle speed object tracking method is used to get the number of frames that traversed each object of a specified limit. Particle Filter is used as a method of object tracking because it can reduce the search area, especially by using color-based particle filter. It also needed a method to detect the vehicle used as a reference of the particle filter. Meotde used to deal with object detection using Gaussian Mixture Models, this method has the output in the form of a black and white image with a white image is an image that is detected as an object. Particles which attached to each object is used to get the centroid point. The centroid of each object taken as a reference position. By combining these two methods, obtained accuracy in counting the number of vehicles amounted to 91.22% and the accuracy in determining the speed of vehicles by 94.41% while the justification density has a high compatibility and suitability to service traffic flows amounted to 66.67% of the video being tested.*

*Keywords: traffic density, road tolls, object tracking, vehicle tracking, vehicle counting, particle filter.*

#### 1. Pendahuluan

Perkembangan mobil di Indonesia saat ini bertumbuh pesat, terlihat dari data yang disajikan oleh Badan Pusat Statistik (BPS) pada [1]. Dengan semakin banyaknya jumlah mobil yang berada di jalan pasti membuat kondisi jalan raya semakin sesak dan berujung pada kemacetan. Keadaan ini mengakibatkan fungsi jalan tol sebagai jalan alternatif menghadapi kemacetan menjadi berkurang. Hal ini diakibatkan informasi mengenai arus lalu lintas kurang diketahui oleh pengguna jalan, sehingga banyak kendaraan yang menumpuk pada suatu jalur. Sedangkan ada kemungkinan jalan alternatif lain yang dapat diambil. Saat ini sistem pemantau statistik kepadatan kendaraan masih menggunakan sistem manual yaitu dengan monitoring pada kamera yang terpasang di jalan ataupun melakukan pemantauan langsung dilapangan. Diperlukan suatu metode yang dapat mengenali, menghitung, dan menganalisis kendaraan secara otomatis.

Sistem ini berbasiskan pada *object tracking* untuk menandai kendaraan sebagai acuan penghitungan jumlahnya seperti yang dilakukan pada [2-8,10-12]. *Tracking object* dimulai dengan deteksi objek, kemudian menelusuri gerak objek dengan menggunakan metode *Particle Filter*.

Sistem ini terdiri dari beberapa tahap dimulai dari preprocessing, vehicle detecting, tracking and counting, vehicle density estimation. Pada tahap preprocessing dilakukan dengan mengubah video menjadi gambar-gambar yang berkelanjutan. Selanjutnya proses vehicle detecting, untuk mendapatkan sebuah citra latar yang menampilkan citra foreground.

## 2. Dasar Teori dan Perancangan

### 2.1. Gaussian Mixture Model

Gaussian Mixture Model (GMM) merupakan metode yang digunakan untuk memodelkan foreground berdasarkan dengan perubahan pada distribusi warna yang terjadi. Dalam memodelkan foreground dipengaruhi oleh model GMM. Semakin besar jumlah model GMM maka akan semakin sensitif dalam memodelkan foreground. Untuk mendapatkan hasil permodelan foreground dilakukan pencocokan input terhadap distribusi sehingga nantinya menghasilkan citra dalam bentuk foreground dan non-foreground.

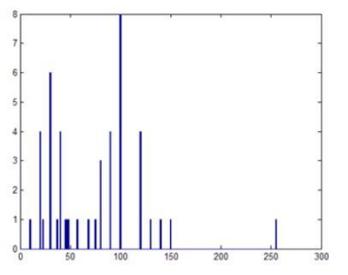
Input dicocokkan dengan seluruh distribusi sehingga menghasilkan distribusi yang paling cocok. Sedangkan suatu piksel dikatakan cocok masuk dalam distribusi apabila nilai dari piksel tersebut masuk kedalam jarak 2,5 standar deviasi dari distribusi sehingga memenuhi persamaan berikut.

(1)

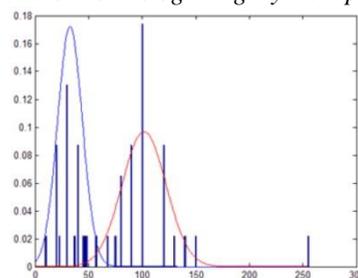
Dimana :

$X_t$  : vektor dari warna piksel (R,G,B) pada waktu t  
 $\mu_k$  : vektor nilai mean (R,G,B) dari Gaussian ke k-th  
 $\sigma_k$  : standar deviasi dari Gaussian ke k-th

Apabila piksel yang ditunjuk tidak cocok dengan distribusi yang ada maka piksel tersebut dianggap sebagai foreground kemudian dibuat suatu distribusi baru dengan menggantikan nilai nilai dari distribusi sebelumnya yang tidak mencerminkan foreground. Distribusi yang baru memiliki nilai mean sesuai dengan nilai piksel, nilai variansi yang tinggi dan nilai weight yang kecil. Sehingga nilai awal yang diberikan pada variabel tersebut akan sangat mempengaruhi performa dari algoritma GMM. Nilai awal dari dimaksud merupakan distribusi yang diperoleh dari hasil banyaknya frame latih pada saat awal.



Gambar 1 Ilustrasi histogram gray level pada frame



Gambar 2 Ilustrasi distribusi Gaussian yang didapat

Pada gambar 2-3 dapat dilihat bahwa terdapat pemisahan distribusi yang digambarkan dengan garis berwarna biru dan merah.

### 2.2. Particle Filter

“Algoritma Particle Filter atau disebut juga *sequential Monte Carlo (MC)* merupakan metode yang memiliki dasar pada representasi partikel dari probabilitas kepadatan. Metode ini digunakan sebagai solusi dari metode yang pernah ada sebelumnya seperti *nonlinear* dan *non-Gaussian* yang masih memiliki masalah-masalah dalam penerapannya. Particle Filter ini dalam penerapannya dibagi menjadi 3 (tiga) tahapan penting yaitu : generasi partikel baru, perhitungan bobot partikel, dan resampling. Pada tahap resampling perlu menangani masalah degenerasi Particle Filter yang merupakan tantangan utama dalam pengembangan metode ini.” (Xiaoyan Fu and Yingmin Jia, 2010). Particle Filter menurut [12] merupakan teknik *Bayesian Sequential* dengan sampel sebagai hal penting, yang secara rekursif mendekati distribusi posterior menggunakan *finite sets* dari nilai sampel.

Pada dasarnya terdiri dari 2 (dua) langkah, yaitu : prediksi dan *update*. Dengan observasi yang digunakan yaitu hingga waktu  $t - 1$ , pada tahap prediksi menggunakan model sistem transisi probabilistik  $(x_t | x_{t-1})$  untuk memprediksi *posterior* pada saat waktu  $t$  seperti berikut,

$$p(x_t | z_{1:t-1}) = \int p(x_t | x_{t-1}) p(x_{t-1} | z_{1:t-1}) dx_{t-1} \quad (2)$$

saat  $t$  dan observasi  $z_t$ , untuk meng*update state* dapat menggunakan aturan Bayes,

$$p(x_t | z_{1:t}) = \frac{p(z_t | x_t) p(x_t | z_{1:t-1})}{\int p(z_t | x) p(x | z_{1:t-1}) dx} \quad (3)$$

dimana  $p(x_t | z_t)$  dijelaskan dengan persamaan observasi.

Pada *particle filter*, *posterior*  $p(x_t | z_{1:t})$  didekati oleh sebuah himpunan berhingga dari sampel  $N$  dengan bobot kepentingannya  $w_t^{(k)}$ .

### 2.3. Particle Filter Berbasis Warna

Sesuai dengan [6,8] Particle Filter ini menggunakan warna sebagai target  $Y = (r, g, b)$  penempatan partikel dengan melalui 2 tahap.

#### 1. Update

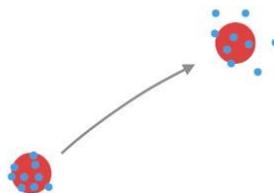
Pada tahap update ini melakukan penempatan posisi partikel dengan memperhatikan waktu sebelumnya ( $t-1$ ) dan saat ini ( $t$ ) dengan menggunakan rumus berikut.

$$x_t^{(k)} = x_{t-1}^{(k)} + \epsilon_t \quad (4)$$

$X_{n|m}^{(k)}$  : Estimator state partikel ke- $n$  dengan menggunakan informasi hingga waktu ke  $m$  pada partikel ke  $k$ .

$W_n$  : Random number dari model matematika.

Dari perhitungan diatas menghasilkan  $X$  dengan isi lokasi dan *speed* dari objek yang terdeteksi  $(x, y, \dot{x}, \dot{y})$ .



Gambar 3 Ilustrasi update partikel pada frame selanjutnya

#### 2. Filtering

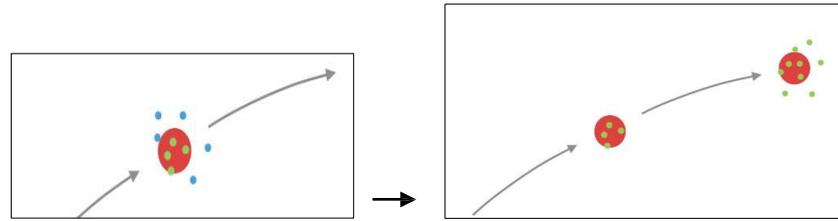
Filtering merupakan bagian dari *resample* partikel yang sudah terdefinisi berdasarkan objek sesuai dengan nilai kemiripan warna. Dengan menggunakan rumus berikut ini.

$$w_t^{(k)} = \frac{p(z_t | x_t^{(k)}) p(x_t^{(k)} | z_{1:t-1})}{\sum_{k=1}^N p(z_t | x_t^{(k)}) p(x_t^{(k)} | z_{1:t-1})} \quad (5)$$

Dengan  $(r, g, b)$  merupakan kemungkinan dari  $Y$  pada state  $(x, y)$ .  
Sehingga,

$$d = \sqrt{(r - r_t)^2 + (g - g_t)^2 + (b - b_t)^2} \quad (6)$$

dimana,  $d = \sqrt{(r - r_t)^2 + (g - g_t)^2 + (b - b_t)^2}$  dengan  $(r, g, b)$  merupakan warna yang ditempati partikel dan  $(r_t, g_t, b_t)$  merupakan warna dari target.



Gambar 4 Ilustrasi Filtering Partikel

2.4. Kepadatan Lalu Lintas

Untuk menghitung intensitas kepadatan pada lalu lintas, sesuai dengan peraturan yang berlaku mengacu pada dua hal utama yaitu volume (flow rate) serta kecepatan rata-rata kendaraan (average speed). Dengan menggunakan rumus berikut,

(7)

Dimana :

- D = Density (Kepadatan) dengan satuan kendaraan/km.
- F = Flow Rate (Volume) dengan satuan kendaraan/jam.
- S = Speed (Kecepatan) dengan satuan km/jam.

Flow rate dapat diartikan sebagai jumlah kendaraan yang melewati suatu titik pada jalan dalam kurun waktu tertentu. Flow rate sendiri dapat diperoleh dengan menggunakan persamaan berikut ini.

(8)

Dengan :

- F = Flow rate (jumlah kendaraan yang melewati satu titik dalam kurun waktu tertentu (kend/jam)
- n = Jumlah kendaraan
- t = Waktu pengamatan

2.5. Klasifikasi Kepadatan Lalu Lintas

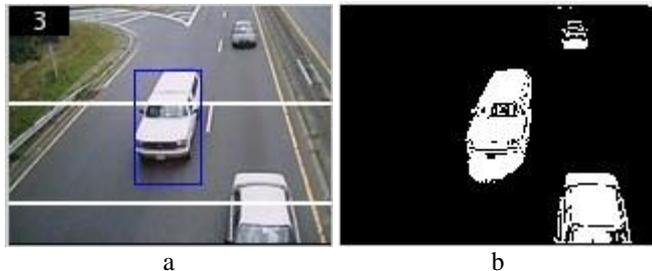
Dari nilai kepadatan lalu lintas yang didapat dari hasil komputasi berdasarkan rumus (7). Untuk menentukan tipe kepadatan yang terjadi, sudah terdapat peraturan pemerintah berdasarkan [9] tersedia pada tabel dibawah ini.

Tabel 1 Klasifikasi Kepadatan Lalu lintas

Tingkat Pelayanan	Karakteristik Operasi Terkait	Klasifikasi Kepadatan (D)
A	<ul style="list-style-type: none"> <li>• Arus bebas</li> <li>• Kecepatan <math>\geq 100</math> km/jam</li> <li>• Volume max 1400 kend/jam</li> </ul>	<ul style="list-style-type: none"> <li>• Kepadatan <math>\leq 14</math> kend/km</li> <li>• Lancar</li> </ul>
B	<ul style="list-style-type: none"> <li>• Arus stabil dengan kecepatan tinggi</li> <li>• Kecepatan <math>\geq 90</math> km/jam</li> <li>• Volume max 2000 kend/jam</li> </ul>	<ul style="list-style-type: none"> <li>• Kepadatan <math>\leq 22</math> kend/km</li> <li>• Lancar</li> </ul>
C	<ul style="list-style-type: none"> <li>• Arus masih stabil</li> <li>• Kecepatan <math>\geq 80</math> km/jam</li> <li>• Volume max 3000 kend/jam</li> </ul>	<ul style="list-style-type: none"> <li>• Kepadatan <math>\leq 37</math> kend/km</li> <li>• Sedang</li> </ul>
D	<ul style="list-style-type: none"> <li>• Arus mendekati tidak stabil dan peka terhadap perubahan kondisi</li> <li>• Kecepatan 60 – 80 km/jam</li> <li>• Volume max 3600 kend/jam</li> </ul>	<ul style="list-style-type: none"> <li>• Kepadatan 38 – 60 kend/km</li> <li>• Sedang</li> </ul>
E	<ul style="list-style-type: none"> <li>• Arus tidak stabil</li> <li>• Kecepatan 50 – 60 km/jam</li> <li>• Volume max 4000 kend/jam</li> </ul>	<ul style="list-style-type: none"> <li>• Kepadatan 61 – 80 kend/km</li> <li>• Padat</li> </ul>
F	<ul style="list-style-type: none"> <li>• Arus tertahan</li> <li>• Kecepatan <math>&lt; 50</math> km/jam</li> <li>• Volume <math>&gt; 4000</math> kend/jam</li> </ul>	<ul style="list-style-type: none"> <li>• Kepadatan <math>&gt; 80</math> kend/km</li> <li>• Padat</li> </ul>

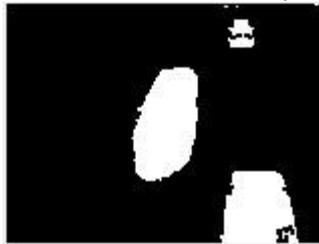
### 3. Pembahasan

Pembangunan sistem ini menggunakan aplikasi MATLAB R2014a untuk membangun dan mengujinya. Input dari sistem ini merupakan video yang memiliki ekstensi .avi. untuk membaca inputan yang berupa video ini menggunakan *tools* dari Matlab yaitu dengan menggunakan *library* vision.VideoFileReader. Setelah video tersebut dibaca, kemudian masuk kedalam tahap *preprocessing* yang ditangani oleh metode GMM. Metode ini memiliki tahap training dengan menggunakan sejumlah frame dari seluruh video. Jumlah frame latihan yang terbaik adalah sekitar 0,05% dari jumlah seluruh frame pada video. Hasil output pada tahap preprocessing ini yaitu citra biner dengan warna putih merupakan citra yang mewakili objek. Berikut merupakan contoh hasil keluaran dari GMM.



Gambar 5. a. Gambar pada frame asli, b. Gambar hasil proses GMM

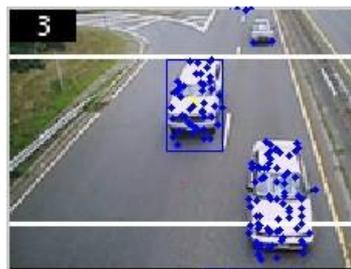
Setelah didapatkan citra dari hasil *preprocessing* menggunakan GMM selanjutnya dilakukan proses penegasan *blob* dengan menggunakan library dari Matlab yaitu *imfill* serta morfologi citra seperti *closing* dan *opening*. Hal ini dilakukan untuk mengisi bagian hitam pada objek serta memisahkan antar objek antara yang satu dengan yang lain apabila terlalu dekat. Berikut ini hasil dari library *imfill* dan morfologi yang digunakan.



Gambar 6 Hasil dari pemrosesan *imfill* dan *closing*

Gambar 2. Hasil dari pemrosesan *imfill* dan *closing*

Selanjutnya setelah didapatkan *blob* yang penuh, hasil citra ini kemudian dijadikan target warna untuk *tracking* menggunakan *particle filter*. Berikut ini merupakan hasil dari pemrosesan dengan menggunakan *particle filter*.



Gambar 7. Hasil dari pemrosesan *particle filter*

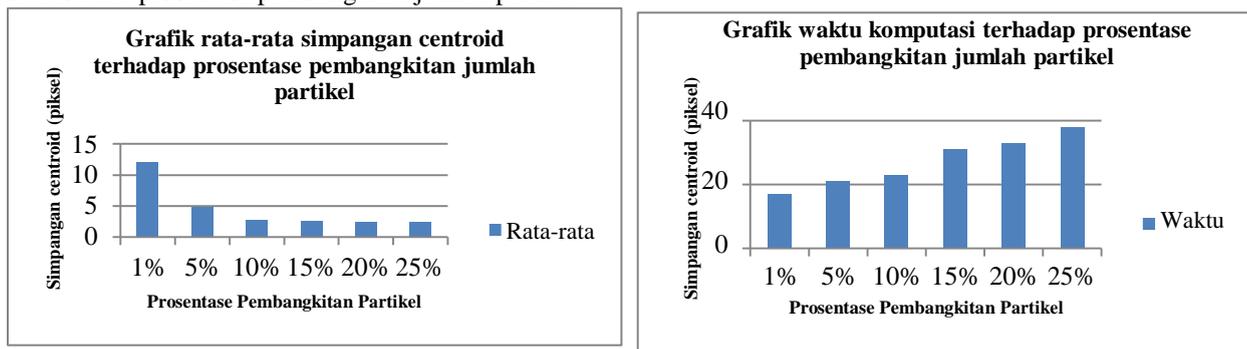
Pada tiap objek dapat dilihat terdapat *particle* yang menempel, dari partikel-partikel ini kemudian dicari nilai rata-rata untuk memperoleh koordinat centroid tiap objek. Pada gambar 3 centroid ditandai dengan titik berwarna kuning. Jumlah partikel yang dibangkitkan mempengaruhi letak dari centroid pada tiap objek karena jumlah partikel yang menempel pada tiap objek memiliki jumlah yang berbeda-beda. Untuk itu, dilakukan pengujian terhadap stabilitas posisi centroid yang diperoleh antara centroid objek yang sesungguhnya dengan yang diperoleh berdasarkan partikel pada tiap objek. Jumlah partikel yang dibangkitkan berdasarkan prosentase dari luas dimensi frame yang digunakan pada video. Berikut hasil yang diperoleh mengenai stabilitas posisi centroid. Prosentase jumlah partikel yang digunakan yaitu 1%, 5%, 10%, 15%, 20%, dan 25% dari luas frame.

Tabel 2 Selisih Posisi Koordinat Antar Tiap Objek

Video 1 (viptraffic)	Rata - rata selisih											
	1%		5%		10%		15%		20%		25%	
Kendaraan	x	y	x	Y	x	y	x	y	x	y	x	y
1	2,814	4,42	2,979	4,346	2,0904	2,621	1,796	3,242	1,543	2,473	2,139	3,464
2	3,967	2,94	2,234	2,502	1,4488	3,426	1,824	3,627	1,949	2,639	1,836	2,863
3	4,313	5,437	1,824	4,724	1,9784	4,2	1,721	3,791	2,229	3,223	1,818	3,173
4	19,91	12,36	20,1	8,442	3,022	3,574	1,596	2,931	1,798	3,019	1,923	2,972
5	2,646	4,473	2,757	2,743	2,8036	3,684	2,868	3,757	2,5	2,706	2,25	2,82
6	29,07	12,25	3,11	2,522	1,6998	3,745	2,245	3,74	1,657	2,96	2,408	3,047
7	2,702	2,976	1,631	3,1	1,3958	2,942	1,675	1,835	1,567	3,034	1,108	2,464
8	28,18	14,74	15,73	6,853	2,4277	5,2	3,078	3,226	2,528	3,62	2,145	3,202
9	12,53	7,372	1,378	3,384	1,2884	2,145	1,412	3,135	1,493	2,894	1,554	2,716
10	50,02	18,8	2,804	4,211	2,8279	3,478	2,314	3,374	2,216	3,681	2,28	2,917
Rata-rata	15,61	8,576	5,454	4,283	2,0983	3,502	2,053	3,266	1,948	3,025	1,946	2,964
Rata-rata simpangan	12,1		4,869		2,8		2,659		2,487		2,455	
Waktu Komputasi I (detik)	34		42		55		81		93		110	
Waktu Komputasi II (detik)	17		21		23		31		33		38	

Dari tabel di atas dihasilkan nilai rata-rata simpangan centroid, dengan waktu komputasi. Waktu komputasi ada dua macam yaitu waktu komputasi pertama adalah waktu komputasi sistem dengan menampilkan partikel pada visualisasinya sedangkan waktu komputasi kedua tanpa menampilkan partikel tersebut. Dari data yang diperoleh, dapat dilihat bahwa rata-rata penyimpangan posisi centroid yang memiliki dengan prosesntase pembangkitan mulai 10% memiliki perubahan tingkat penyimpangan yang realtif sudah tidak terlalu mencolok, namun waktu komputasi yang dibutuhkan antara 10% dengan 15% memiliki jarak yang relatif besar.

Berikut grafik yang menggambarkan nilai simpangan centroid dan waktu kmputasi yang dibutuhkan berdasarkan prosentase pembangkitan jumlah partikel.



Gambar 7 Grafik Nilai Simpangan Centroid dan Waktu Komputasi Terhadap Prosentase Pembangkitan Jumlah Partikel

Setelah didapatkan jumlah partikel yang sesuai dalam hal ini yang tingkat stabilitasnya tinggi, centroid ini digunakan sebagai patokan posisi dari tiap objek. Dengan menggunakan centroid tersebut, kita dapat menghitung jumlah frame yang dilalui oleh setiap objek. Jumlah frame ini digunakan untuk menghitung kecepatan dari tiap objek yang diperlukan untuk mendapatkan nilai kecepatan rata-rata dari seluruh objek yang ada pada video. Dan dengan menggunakan rumus (9) maka akan didapatkan nilai kepadatan dimana nilai tersebut merupakan parameter untuk menentukan karakteristik dari pelayanan lalu lintas dan justifikasi dari kepadatan arus lalu lintas berdasarkan pada tabel 1.

Untuk menghitung jumlah kendaraan, yaitu dengan berdasarkan kendaraan yang terdeteksi melewati garis awal yang sudah ditentukan. Berikut ini merupakan hasil perhitungan terhadap jumlah kendaraan yang terdeteksi pada sistem.

Tabel 3 Hasil Perhitungan Jumlah Kendaraan Yang Dihasilkan

Video	Jumlah Kendaraan		Selisih	Akurasi
	Sistem	Nyata		
1	10	10	0	100 %
2	163	138	25	81,88%
3	18	17	1	94,12%
4	27	24	3	88,89%
Rata-rata akurasi				91,22%

Dari sistem juga dihasilkan nilai kecepatan rata-rata 3 dari 4 video diatas. Berikut ini hasil kecepatan rata-rata yang dihasilkan oleh sistem.

Tabel 4 Hasil Akurasi Kecepatan Rata-rata

No	Video	$\Pi$ Kec. Nyata	$\Pi$ Kec. Sistem	Selisih	Akurasi
1	Viptraffic	101,31	99,91	1,4	98,61%
2	test_cut(1)	101,77	108,94	7,17	93,41%
3	test_cut(3)	90,60	99,34	8,74	91,20%
Rata-rata		97,89	102,73	5,77	94,41%

Dari hasil diatas, pada tabel 3 perhitungan jumlah kendaraan masih terdapat kesalahan yaitu kendaraan yang dideteksi terhitung lebih banyak ini dikarenakan adanya interferensi terhadap citra *foreground* sehingga ada citra yang bukan objek namun terdeteksi sebagai objek selain itu juga dikarenakan bayangan yang juga masih terdeteksi sebagai objek. Berdasarkan data kecepatan diatas, maka tingkat kepadatan yang diperoleh adalah sebagai berikut.

Tabel 5 Hasil Tingkat Kepadatan Lalu Lintas

No	$\Pi$ Kec. Nyata	Kepadatan Nyata	Karakteristik	$\Pi$ Kec. Sistem	Kepadatan Sistem	Karakteristik	Status
1	101,31	44,4181	Sedang	99,91	45,0405	Sedang	Benar
2	101,77	33,41	Sedang	108,94	33,04	Sedang	Benar
3	90,60	38,32	Sedang	99,34	34,94	Sedang	Benar

#### 4. Kesimpulan & Saran

Kesimpulan yang dapat diambil berdasarkan pembahasan yang telah dilakukan yaitu tingkat kepadatan yang dihasilkan oleh sistem memiliki tingkat kecocokan yang tinggi hal ini dihasilkan dengan tingkat akurasi terhadap jumlah kendaraan dan kecepatan rata-rata sebesar 91,22% dan 94,41%. Sedangkan jumlah partikel yang dibangkitkan sebesar 10% hal ini dikarenakan semakin banyak jumlah partikel yang dibangkitkan maka posisi centroid semakin stabil dan waktu komputasi yang tidak ada perbedaan yg terlalu jauh.

Adapun saran yang dapat dilakukan untuk pengembangan dan penelitian yang selanjutnya adalah sebagai berikut.

1. Dataset yang digunakan sebaiknya diperoleh sendiri sehingga dapat mengatur pengaturan posisi kamera sehingga dapat mengetahui jarak sesungguhnya yang dilewati oleh objek yaitu antara garis awal dan garis akhir.
2. Metode pendeteksian objek dapat menggunakan metode lainnya yang lebih akurat dan stabil sehingga tidak terjadi gangguan terutama dalam pendeteksian objek.
3. Metode shadow removal dapat digunakan untuk menghilangkan pendeteksian objek yang salah yaitu terhadap bayangan yang terjadi.
4. Perhitungan tingkat kepadatan dapat menggunakan parameter yang lainnya sehingga dapat memberikan variasi dan dapat menjadi perbandingan.
5. Dapat dilakukan pengembangan sistem sehingga dapat mendeteksi kepadatan secara *realtime*.

## 5. Daftar Pustaka

- [1] [http://bps.go.id/tab\\_sub/view.php?kat=2&tabel=1&daftar=1&id\\_subyek=17&notab=12](http://bps.go.id/tab_sub/view.php?kat=2&tabel=1&daftar=1&id_subyek=17&notab=12), di akses terakhir 03 – 10 – 2014
- [2] Alessandro Bof de Oliveira and Jacob Scharcanski, *-Vehicle Counting and Trajectory Detection Based on Particle Filtering*||, IEEE 23rd conference, 2010.
- [3] Changjiang Yang, Ramani Duraiswami and Larry Davis, *-Fast Multiple Object Tracking via a Hierarchical Particle Filter*||, University of Maryland, College Park, USA.
- [4] Christiano Bouvie et al., *-Tracking and Counting Vehicles in Traffic Video Sequences Using Particle Filtering*||, DIGICON.
- [5] Jacob Scharcanski, Alessandro Bof de Oliveira, *-A Particle-Filtering Approach for Vehicular Tracking Adaptive to Occlusions*||, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 60, NO. 2, FEBRUARY 2011.
- [6] Katja Nummiaro, Esther Koller-Meier and Luc Van Gool, *-A Color-based Particle Filter*||, Katholieke Universiteit Leuven, ESAT/PSI Visics, Belgium.
- [7] M. Jaward, L. Mihaylova, N. Canagarajah, D. Bull, *-Multiple Object Tracking Using Particle Filters*||, Department of Electrical and Electronic Engineering, University of Bristol, UK.
- [8] P. Perez, C.Hue, J. Vermaak, and M. Gangnet, *-Color-Based Probabilistic Tracking*||, Spinger-Verlag Berlin Heidelberg 2012.
- [9] Peraturan Menteri Perhubungan Nomor : KM 14 Tahun 2006.
- [10] Sri Lakshmi Gogulamudi and B M Kusuma Kumari, *-EFFICIENT MULTIPLE OBJECTS DETECTION AND TRACKING USING PARTICLE FILTER*||, International Journal of Information Technology, Control and Automation (IJITCA) Vol.2, No.4, October 2012.
- [11] W.L. Khong, W.Y. Kow, Y.K. Chin, I. Saad, K.T.K. Teo, *-Overlapping Vehicle Tracking via Adaptive Particle Filter with Multiple Cues*||, IEEE International Conference, 2011.
- [12] Xiaoyan Fu and Yingmin Jia, *-An Improvement on Resampling Algorithm of Particle Filters*||, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 58, NO. 10, OCTOBER 2010.