

## **IMPLEMENTASI *OBJECT TRACKING* UNTUK MENDETEKSI DAN MENGHITUNG JUMLAH KENDARAAN SECARA OTOMATIS MENGGUNAKAN METODE *KALMAN FILTER* DAN *GAUSSIAN MIXTURE MODEL***

***Havez Vazirani Al Kautsar dan Kusworo Adi***

*Jurusan Fisika, Fakultas Sains dan Matematika, Universitas Diponegoro*

*Jl. Prof. Soedharto, SH., Tembalang, Semarang, Jawa Tengah*

*E-mail: alkautsar@st.fisika.undip.ac.id*

### **ABSTRACT**

*Traffic density can be controlled by obtaining and managing the data of the traffic flows on the highway. Generally, the process of data acquisition of the traffic flows which passing on the highway are still done manually by assigning some officers to be on the highway and count each of passing vehicle, then divided by a certain time frame. This manual counting are still have many weaknesses such as time of collecting data become longer, and need much amount of the human resources. Based on these conditions, needs an accurate automatic vehicle detection and counting system as traffic monitors, traffic controllers and traffic analysis. At this time, it has been developed a vehicle detection system using a hardware system such as using sensors, Radio Frequency Identifier or other hardware which integrated by software in the microcontroller and works automatically to detect the speed and count the number of passing vehicles on the highway. The weaknesses of these detectors can only detect at the narrow range, design of the system, the complexity of the operation, and also has a significant operational cost. Based on those system weaknesses, this study was developed with a focus of designing the detection system and the vehicle counter system using Kalman filter and Gaussian Mixture Models (GMM) method. This system get the most accurate results in the morning (10,000-25,000 lux illumination) with F1 Score value of 0.91111, while counting the vehicles most inaccurate happen at night (illumination from 0.27 to 1.0 lux) with F1 Score only 0.16071.*

**Keywords:** *Traffic flows data, vehicle counter system, Object Tracking, Gaussian Mixture Model, Kalman Filter.*

### **ABSTRAK**

*Kepadatan lalu lintas dapat dikontrol dengan cara memperoleh dan mengelola data volume kendaraan di jalan raya. Pada umumnya, proses akuisisi data volume kendaraan yang lewat di jalan raya masih dilakukan dengan cara manual yaitu dengan menugaskan beberapa orang untuk berada di lapangan dan menghitung setiap kendaraan yang lewat, kemudian dibagi dengan rentang waktu tertentu. Dalam penghitungan secara manual ini masih terdapat banyak kelemahan seperti pengumpulan data yang lama, dan jumlah sumber daya manusia yang dibutuhkan banyak. Berdasarkan pada kondisi tersebut dibutuhkan sistem penghitung dan deteksi kendaraan otomatis yang akurat sebagai pemantau kontrol lalu lintas dan analisa lalu lintas. Pada saat ini telah dikembangkan sistem deteksi kendaraan menggunakan sensor, Radio Frequency Identifier atau perangkat keras lain yang diintegrasikan oleh perangkat lunak pada mikrokontroler dan bekerja otomatis untuk mendeteksi kecepatan serta menghitung jumlah kendaraan yang melintas di jalan raya. Kelemahan detektor tersebut hanya dapat mendeteksi pada jangkauan yang sempit, perancangan sistem dan operasional yang rumit, serta biaya operasional yang besar. Berdasarkan pada beberapa hal tersebut, penelitian ini dikembangkan dengan fokus perancangan sistem deteksi dan penghitung jumlah kendaraan dengan metode Kalman Filter dan Gaussian Mixture Model (GMM). Sistem ini mendapatkan hasil yang paling akurat pada pagi hari (pencahayaan 10.000-25.000 lux) dengan nilai F1 Score sebesar 0,91111, sedangkan penghitungan kendaraan yang paling tidak akurat terjadi pada malam hari (pencahayaan 0,27-1,0 lux) dengan F1 Score 0,16071.*

**Kata Kunci:** *Data volume kendaraan, sistem penghitung kendaraan, Object Tracking, Gaussian Mixture Model, Kalman Filter.*

## PENDAHULUAN

Kepadatan lalu lintas harus selalu diawasi agar pengelolaan infrastruktur lintas dapat dikelola dengan baik, sehingga kemacetan lalu lintas dapat dikurangi [1]. Data volume kendaraan dapat digunakan untuk proses penghitungan kepadatan kendaraan, penghitungan frekuensi kendaraan, prediksi kemacetan, juga dapat digunakan sebagai referensi untuk perbaikan jalan yang sudah ada, pelebaran jalan, penambahan jalan baru pembuatan atau penataan rute baru, penataan rambu-rambu lalu lintas dan peningkatan infrastruktur transportasi lainnya [2].

Pada umumnya, proses akuisisi data volume kendaraan yang lewat di jalan raya masih dilakukan dengan cara manual yaitu dengan menugaskan beberapa orang untuk berada di lapangan dan menghitung setiap kendaraan yang lewat, kemudian dibagi dengan rentang waktu tertentu. Dalam penghitungan secara manual ini masih terdapat banyak kelemahan seperti pengumpulan data yang lama, dan jumlah sumber daya manusia yang dibutuhkan banyak [2].

Pada saat ini telah dikembangkan sistem deteksi kendaraan menggunakan sensor, *Radio Frequency Identifier* atau perangkat keras lain yang diintegrasikan oleh perangkat lunak pada mikrokontroler dan bekerja otomatis untuk mendeteksi kecepatan serta menghitung jumlah kendaraan yang melintas di jalan raya [3]. Kelemahan detektor tersebut hanya dapat mendeteksi pada jangkauan yang sempit, perancangan sistem dan operasional yang rumit, serta biaya operasional yang besar.

Pada bidang visi komputer telah dikembangkan sistem deteksi kendaraan otomatis secara visual yang digunakan untuk mendeteksi dan menghitung kendaraan yang bergerak dinamis secara efisien. Pengawasan visual yang cerdas untuk kendaraan di jalan raya adalah komponen penting untuk mengembangkan sistem transportasi cerdas [1]. Berdasarkan pada beberapa hal tersebut, penelitian ini

dikembangkan dengan fokus perancangan sistem deteksi dan penghitung jumlah kendaraan dengan metode *Kalmanfilter* dan *Gaussianmixture model (GMM)*.

## DASAR TEORI

### *ObjectTracking*

Dalam visi komputer, *objecttracking* (pelacakan objek) adalah suatu proses untuk melacak satu objek atau lebih dari suatu citra. *Objecttracking* termasuk dalam salah satu fungsi yang sangat penting di bidang visi komputer. Ada tiga langkah penting dalam analisa *video*: deteksi objek yang bergerak, mendeteksi beberapa objek di setiap *frame*, dan analisa objek yang dilacak untuk mengenali pergerakan objek pada citra. Dalam bentuk yang paling sederhana, *tracking* dapat didefinisikan sebagai suatu masalah untuk memperkirakan lintasan dari sebuah objek yang bergerak dalam sebuah gambar. Secara konsisten, pelacak memberikan label pada objek yang dilacak pada *frame-frame* yang berbeda dalam sebuah *video*. Berdasarkan dari pelacakan domain, sebuah pelacak juga dapat memberikan informasi suatu objek, seperti sebuah orientasi gerak, area, atau bentuk dari objek [4].

Satu hal yang dapat memudahkan pelacakan, yaitu dengan memberikan batasan secara paksa pada pergerakan dan/atau bentuk dari objek. Sebagai contoh, hampir semua algoritma *tracking* berasumsi bahwa objek bergerak secara tetap tanpa ada perubahan yang mendadak serta memaksakan kecepatan atau percepatan objek menjadi konstan berdasarkan informasi sebelumnya. Pembelajaran sebelumnya mengenai ukuran, tampilan, dan bentuk objek, juga dapat digunakan untuk memudahkan masalah [4].

### *Gaussian Mixture model*

*GaussianMixture model (GMM)* adalah bagian penting dari penghitung kendaraan, yang fungsinya digunakan untuk melakukan segmentasi citra. Berdasarkan teori *GMM*, satu distribusi *Gaussian*

memungkinkan untuk membuat suatu pemodelan *background* ketika sebuah *background* pada tingkat kecerahan tertentu. Akan tetapi, kecerahan dari sebuah *background* bervariasi secara bertahap dari waktu-waktu. Dengan demikian, distribusi *Gaussian* yang adaptif ini digunakan untuk membuat pemodelan tersebut. Jika terdapat nilai piksel yang tidak sesuai dengan distribusi *Gaussian* pada *background*, maka disebut dengan *foreground*. Ketika terdapat distribusi *Gaussian*, kumpulan contoh piksel pada *background* akan menjadi  $\{X_1, \dots, X_t\}$ , sehingga diberikan fungsi probabilitas yang ditunjukkan pada persamaan (1),

$$P(X_t) = \prod_{i=1}^K w_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

Dimana  $w_{i,t}$  menunjukkan bobot kovarian ke-  $i$  dari pemodelan *Gaussian* pada waktu  $t$  dalam jam,  $\mu_{i,t}$  adalah nilai rata-rata kovarian ke-  $i$  dari pemodelan *Gaussian* pada waktu  $t$  dalam jam, sedangkan  $\Sigma_{i,t}$  adalah matriks kovarian ke-  $i$  dari pemodelan *Gaussian* pada waktu  $t$  dalam jam.  $\eta$  adalah fungsi probabilitas kerapatan atau *probability density function* (pdf),  $\eta$  dinyatakan dalam persamaan (2),

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \quad (2)$$

Dimana  $X$  adalah variabel yang memberikan jumlah piksel dalam *frame* [5].

### Kalman Filter

Teknik *Kalmanfilter* adalah kumpulan persamaan matematika yang menyediakan perhitungan komputasi yang efisien (rekursif) untuk memperkirakan suatu keadaan dari sebuah proses, dengan cara meminimalkan nilai *error* kuadrat rata-rata antara citra asli dengan citra hasil penyisipan yang biasa disebut dengan *Mean Squared Error* (MSE). *Filter* ini

dapat memperkirakan suatu keadaan di masa sebelumnya, sekarang, dan yang akan datang, bahkan *filter* ini dapat memperkirakan secara tepat sebuah sistem pemodelan yang belum diketahui [6].

Dalam visi komputer, *Kalmanfilter* biasa digunakan untuk *tracking* suatu objek dalam citra hasil segmentasi. Teknik *Kalmanfilter* biasa digunakan untuk memperkirakan keadaan suatu sistem linier dengan asumsi keadaan tersebut telah terdistribusi oleh *Gaussian*. *Kalmanfilter* melacak suatu objek dengan cara memprediksi posisi objek dari informasi sebelumnya dan memeriksa keberadaan objek pada posisi yang diprediksi [7].

*Kalmanfilter* memperkirakan sebuah proses dalam bentuk kendali umpan balik (*feedback control*). Persamaan *Kalmanfilter* terbagi dalam dua kelompok, persamaan pembaruan waktu dan persamaan pembaruan pengukuran. Persamaan pembaruan waktu bertugas untuk memproyeksikan keadaan berikutnya (dalam waktu) dan kovarian *error*-nya diperkirakan untuk menghasilkan perkiraan sebelumnya untuk langkah selanjutnya. Persamaan pembaruan pengukuran bertugas sebagai umpan balik. Persamaan ini digunakan untuk memasukkan sebuah pengukuran baru menjadi perkiraan sebelumnya untuk menghasilkan sebuah perkiraan selanjutnya yang lebih baik. Persamaan pembaruan waktu dapat juga disebut dengan persamaan prediksi, sedangkan persamaan pembaruan pengukuran disebut sebagai persamaan koreksi [7].

$$X_{pred_k} = A \cdot X_{k-1} + B \cdot U_k + W_{k-1} \quad (3)$$

$$P_{pred_k} = A \cdot P_{k-1} \cdot A + Q \quad (4)$$

$X_{pred_k}$  adalah vektor dari proses suatu keadaan yang terprediksi dalam waktu  $k$ .  $X$  adalah sebuah vektor 4 dimensi  $(x, y, dx, dy)$ , dimana  $x$  dan  $y$  adalah koordinat dari pusat objek, sedangkan  $dx$

dan  $dy$  mewakili kecepatan.  $X_{k-1}$  adalah vektor dari proses suatu kondisi dalam waktu  $k - 1$ .  $A$  adalah sebuah proses matriks transisi  $4 \times 4$  dari bentuk [7].

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad (5)$$

$U_k$  adalah sebuah vektor kendali dan  $B$  terhubung secara opsional dengan vektor kendali  $U_k$  dalam kondisi ruang.  $W_{k-1}$  adalah suatu proses derau (*noise*).  $Ppred_k$  adalah kovariansi *error* yang terprediksi pada waktu  $k$ .  $P_{k-1}$  adalah sebuah matriks yang mewakili kovarian *error* dalam kondisi prediksi pada waktu  $k - 1$ , dan  $Q$  adalah proses kovarian derau. Setelah memprediksi kondisi  $Xpred_k$  dan kovarian *error*-nya pada waktu  $k$  menggunakan langkah pembaruan waktu, selanjutnya *Kalmanfilter* menggunakan pengukuran untuk mengoreksi prediksi *Kalman* pada saat langkah pembaruan pengukuran.

$$K_k = Ppred_k \cdot X_{k-1} + H^T \cdot (H \cdot Ppred_k + H^T + R)^{-1} \quad (6)$$

$$X_k = Xpred_k + K_k \cdot (Z_k - H \cdot Xpred_k) \quad (7)$$

$$P_k = (I - K_k \cdot H) \cdot Ppred_k \quad (8)$$

Pada persamaan (6),  $K_k$  adalah penguat *Kalman*.  $H$  adalah sebuah matriks yang merubah kondisi ruang menjadi ruang pengukuran dan  $R$  adalah kovarian pengukuran derau. Sulit untuk menentukan  $R_k$  sebagai pengatur pengukuran, banyak implementasi *Kalman* yang secara *stateistik* menganalisa pelatihan data untuk menentukan nilai  $R$  yang tetap pada semua pembaruan waktu yang akan datang. Pada persamaan (7)  $X_k$  adalah sebuah proses kondisi sebenarnya. Dengan menggunakan  $K_k$  dan pengukuran  $Z_k$ , kondisi proses  $X_k$  dapat dilakukan pembaruan.  $Z_k$  yang paling memungkinkan adanya koordinat  $x$  dan  $y$  objek target dalam *frame*. Langkah terakhir dari *Kalmanfilter* adalah memperbarui kovarian *error*  $Ppred_k$  yang menjadi  $P_k$

sebagaimana diberikan pada persamaan (8). Setelah setiap pembaruan waktu dan pengukuran berpasangan, proses diulangi dengan perkiraan lanjutan sebelumnya untuk memproyeksikan dan memprediksi perkiraan sebelumnya yang baru [7].

### Pencahayaan pada ObjectTracking

Banyak sekali teknik deteksi objek bergerak tetapi sangat sedikit dari teknik tersebut dapat mengatasi perubahan pencahayaan. Akan tetapi deteksi *background* berderau memberikan masalah pada tingkatan yang lebih rumit. Pencahayaan adalah elemen penting dari suatu gambar, adanya perubahan cahaya *objecttracking* menjadi sangat kompleks, karena bentuk dari suatu objek akan berubah-ubah bergantung pada objek yang dikenai cahaya [8].

### METODOLOGI PENELITIAN

Pada langkah awal dilakukan *survey* jalan raya dengan arus lalu lintas sedang atau tidak terlalu padat untuk menentukan tempat dan waktu untuk akuisisi data. Jalan raya harus dilalui jembatan penyeberangan atau sejenisnya untuk tempat peletakan *webcam* dan *tripod*, sudut yang diperlukan untuk akuisisi citra sebesar  $65^\circ - 90^\circ$  dari jalan datar agar mendapat sudut gambar yang baik. Kamera diatur sedemikian rupa agar tidak bergerak/ selalu pada kondisi stabil agar hasil akuisisi mendapatkan yang baik. Akuisisi data dilakukan pada 4 variasi waktu pagi, siang, sore, dan malam untuk mengetahui pengaruh pencahayaan pada hasil *processing*.

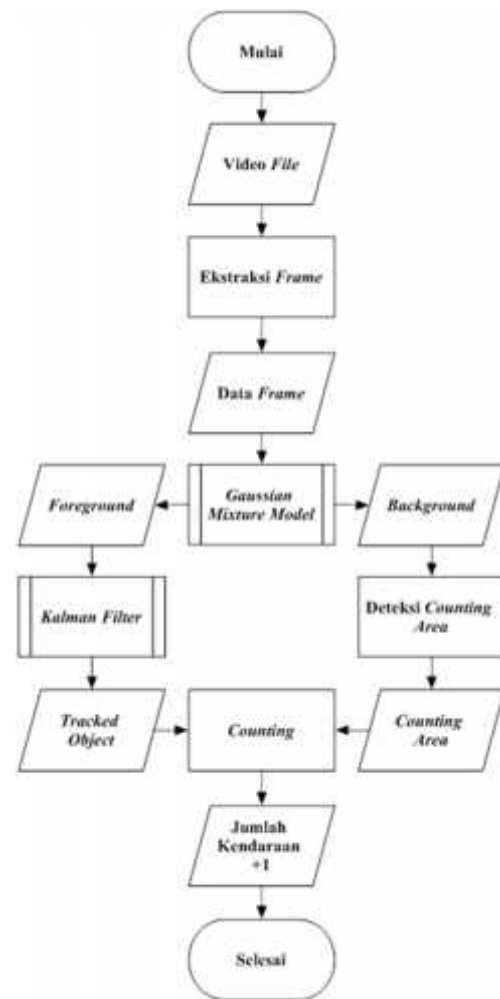
Setelah menentukan tempat dan waktu untuk akuisisi, kemudian dilakukan pengukuran nilai intensitas cahaya setiap waktu (pagi, siang, sore, dan malam hari) yang dilanjutkan dengan akuisisi data. Akuisisi yang dilakukan berupa perekaman gambar menggunakan *webcam*. *Webcam* yang tersambung dengan *PC* akan menyimpan *filevideo* hasil akuisisi citra.

Kemudian hasil akuisisi citra diolah menggunakan metode segmentasi citra dengan fungsi *GMM* (*Gaussian mixture model*) untuk mendeteksi citra *background* dan citra *foreground*. Citra *background* diidentifikasi dengan bagian *video* yang diam, sedangkan citra *foreground* adalah objek yang bergerak pada *video*. Pembuatan *detecting lane* serta inisialisasi citra *background* agar terbagi 2 area, area A dan B. *Foreground* akan diberi ID oleh *GMM* sebagai identitas *frame*.

Fungsi *Kalman filter* diberikan untuk prediksi *frameforeground* yang sudah diberi ID oleh *GMM*. Pada saat satu ID bergerak, *Kalman filter* akan memprediksi pergerakan ID setiap *frame*-nya dari area A sampai area B atau sebaliknya, karena *background* yang terinisialisasi dapat digunakan dua arah (dari A ke B atau B ke A).

*Frameforeground* ID yang ditapis oleh *Kalman filter* dan *GMM* diberikan *ROI* (*Region of Interest*) berupa segi empat untuk membatasi piksel *frame* ID dan sebagai visualisasi *tracking*. Jika *frame* ID yang di-*tracking* dari area awal telah melewati *detecting lane*, maka ID akan dihitung satu secara otomatis karena ID telah berada di area berikutnya. Kemudian sistem akan melakukan *updatetracking* kendaraan ID baru untuk meneruskan iterasi, sehingga jumlah kendaraan (ID) akan diakumulasi dan ditampilkan di *video*.

Sistem bekerja otomatis mulai dari segmentasi citra, melakukan identifikasi *background-foreground* citra, *tracking*, *counting*, hingga menampilkan hasil dari pengolahan citra. Hasil yang ditampilkan berupa *video* hasil pengolahan data dan *video* biner untuk menunjukkan fungsi *Kalman* dan *Gaussian*. Diagram alir prosedur penelitian ditunjukkan pada Gambar 1,



**Gambar 1. Diagram alir *tracking*, dan *counting* kendaraan**

## HASIL DAN PEMBAHASAN

### Hasil *Tracking* dan *Counting*

Penelitian ini mendapatkan hasil kualitatif *F1 Score* dengan hubungan antara *counting manual* (CM) dan *counting otomatis* (CO) dengan parameter *truepositive* (TP), *falsepositive* (FP) serta *falsenegative* (FN). FP dan FN *counting* adalah nilai *error* dari sistem.

Nilai *F1 Score* berkisar antara 0 hingga 1, semakin mendekati nilai 1 maka semakin akurat. *F1 Score* adalah hasil dari tes uji akurasi. TP (*True positive*) adalah *pixel target* yang dikenali sebagai objek. FP (*False positive*) adalah *noise* yang terdeteksi sebagai objek. FN (*False negative*) adalah *target* yang tidak terdeteksi sebagai objek

(Lipton dkk, 2014). Untuk menentukan nilai *F1 Score* ditunjukkan pada persamaan (9).

$$F1\ Score = \frac{TP}{2TP + FP + FN} \quad (9)$$

Hasil *counting* kendaraan secara keseluruhan ditunjukkan pada Tabel 1.

**Tabel 1. Hasil keseluruhan *Counting* secara manual dan otomatis.**

Waktu	C O	C M	T P	F P	F N	F1 Score
Pagi	43	47	41	2	6	0,9111 1
Siang	82	86	76	6	10	0,9047 6
Sore	63	79	63	0	16	0,8873 2
Malam	55	57	9	46	48	0,1607 1

Salah satu contoh *true positive counting* ada pada Gambar 2.



**Gambar 2. *True positive counting* pada pagi hari, a) citra hasil GMM, dan b) citra pada saat *processing***

*False positive counting* ditunjukkan pada Gambar 3.



**Gambar 3. *False positive counting* pada pagi hari, a) citra hasil GMM, dan b) citra pada saat *processing***

Tingkat akurasi penghitungan kendaraan pada penelitian ini dipengaruhi oleh beberapa faktor. Diantaranya adalah,

1. **Pencahayaannya**

Pencahayaannya adalah faktor paling penting karena hasil yang diberikan sangat bergantung pada pencahayaannya. Pencahayaannya yang berhasil diukur oleh *luxmeter* hanya pada *filevideo* sore hari, hal ini dikarenakan *rangeluxmeter* yang digunakan antara 1 hingga 2.000 *lux*.

2. **Noise**

*Noise* memberikan gangguan yang signifikan pada *tracking*. Seperti pada pantulan cahaya dengan intensitas tinggi dari kendaraan yang melintas menyebabkan terbentuknya kumpulan *noise* dengan ukuran *pixel* yang sesuai dengan batasan *pixel*.

3. **Bertumpuknya arus lalu lintas**

Arus lalu lintas yang cukup padat dengan kecepatan rendah menghasilkan bertumpuknya kendaraan dan berhimpitan karena adanya bayangan dari kendaraan, sehingga kendaraan yang berhimpit tersebut dikenali sebagai satu kendaraan yang utuh.

4. **Kecepatan kendaraan**

Kendaraan yang melesat sangat cepat memberikan masalah pada *counting*, meskipun kendaraan berhasil di-*tracking*. Hal ini dikarenakan proses komputasi dari *counting* membutuhkan waktu 0,1 detik secara real time dan *counting* dilakukan tepat pada saat berada pada *counting* area.

5. **Sudut pengambilan gambar**

Sudut pengambilan gambar sangat berpengaruh, karena dengan sudut

yang berbeda menghasilkan perspektif yang berbeda sehingga mempengaruhi ukuran kendaraan. Sudut pengambilan gambar yang dilakukan adalah  $\pm 60^\circ$  dari jalan dengan ketinggian  $\pm 10$  meter.

Hasil penelitian ini menunjukkan bahwa *errorcounting* terbesar ada pada *video* malam hari karena hanya menghasilkan *F1 Score* sebesar 0,16071. *Counting* pada *video* sore hari menghasilkan *F1 Score* sebesar 0,88732. *Counting* pada *video* siang hari menghasilkan *F1 Score* sebesar 0,90476. *Counting* pada *video* pagi hari paling baik karena menghasilkan *F1 Score* tertinggi dengan nilai sebesar 0,91111.

## KESIMPULAN

Berdasarkan pada hasil penelitian dapat disimpulkan bahwa, implementasi GMM dan kalman filter untuk deteksi dan penghitung kendaraan mendapatkan hasil yang paling akurat pada pagi hari (pencahayaan 10.000-25.000 *lux*) dengan nilai *F1 Score* sebesar 0,91111, sedangkan penghitungan kendaraan yang paling tidak akurat terjadi pada malam hari (pencahayaan 0,27-1,0 *lux*) dengan *F1 Score* 0,16071.

## DAFTAR PUSTAKA

- [1] Daigavane, P. M. dan Bajaj, P. R. (2010) *Real Time Vehicle Detection and Counting Method for Unsupervised Traffic Video on Highways*, International Journal of Computer Science and Network Security, Vol.10, No.8.
- [2] Tarnoto, L. (2014) *Rancang Bangun Penghitung Kendaraan Secara Otomatis Berbasis Client Server*, Skripsi, Fakultas Ilmu Komputer, Universitas Gunadarma, Depok.
- [3] Ghozi, M. F., Yuliana, M. dan Sudibyo, R. W., (2013) *Pembuatan Sistem Pendeteksi Kecepatan Kendaraan untuk Mengatasi Kemacetan Lalu Lintas sebagai Bagian Dari Intelligent Transportation System (ITS)*, Jurnal Elektro PENS, Vol. 2, No. 1.
- [4] Yilmaz, A., Javed, O., dan Shah, M. (2006) *Object Tracking: A Survey*, ACM Computing Surveys, Vol. 38 No. 4, Article 13.
- [5] Jang, H., Won, In-Su dan Jeong, Dong-Seok, (2014) *Automatic Vehicle Detection and Counting Algorithm*, International Journal of Computer Science and Network Security, Vol.14, No.9.
- [6] Welch, G. dan Bishop, G. (2006) *An Introduction to the Kalman filter*, University of North Carolina, Chapel Hill.
- [7] Patel, H. A. dan Thakore, D. G. (2013) *Moving Object Tracking Using Kalman Filter*, International Journal of Computer Science and Mobile Computing, Vol.2, Issue. 4, pg. 326-332.
- [8] Ranipa, K. R. dan Bhatt, K. (2014) *Illumination Condition Effect on Object Tracking: A Review*, Global Journal of Computer Science and Technology: F Graphics & Vision. Vol. 14, Issue 5, Ver. 1.0.
- [9] Lipton, Z. C., Elkan, C. dan Naryanaswamy, B. (2014) *Thresholding Classifiers to Maximize F1 Score*, Machine Learning and Knowledge Discovery in Databases, Vol. 8725 pp 225-239.