

ANALISA ALGORITMA PENGHITUNG KENDARAAN RODA EMPAT DALAM KONDISI SIANG DAN MALAM HARI DENGAN METODE FRAME INTERSECTION

Brilliant Bagus Pakerti Utama¹⁾, Ratri Dwi Atmaja²⁾, Azizah³⁾

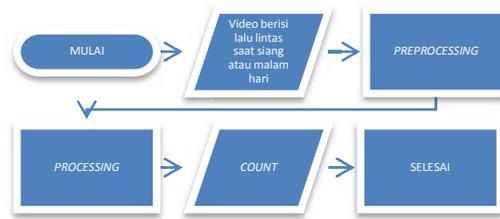
^{1),2),3)}SI Teknik Telekomunikasi, Universitas Telkom
Jl. Telekomunikasi No. 1, Bandung
Email : cinersa23@gmail.com

Abstrak . Sistem counting kendaraan atau objek bergerak yang melewati sebuah jalan merupakan suatu software untuk menghitung berapa banyak kendaraan yang bergerak pada suatu selang waktu tertentu di sebuah jalan. Pada pengerjaan penelitian sebelumnya, perhitungan kendaraan hanya dilakukan ketika dalam kondisi siang hari, dan objek yang digunakan berupa mainan yang berbentuk mobil. Namun, dalam pengerjaan penelitian ini akan dibuat sistem perhitungan kendaraan yang dapat berkerja dalam keadaan terang maupun gelap. Sistem perhitungan kendaraan yang akan dibuat menggunakan metode frame intersection. Metode ini bekerja dengan mencari irisan antara background frame dengan next frame, frame-frame tersebut sebelumnya akan diubah kedalam citra biner, Untuk mendapatkan irisan tersebut, sistem ini akan menggunakan logika 'AND'. Kemudian, didapatkan jumlah piksel hasil irisan. Untuk mengatasi kendala tingkat kecerahan yang kurang, akan dilakukan perubahan level luminance pada kondisi malam hari. Untuk menguji sistem ini, akan digunakan sebuah video yang bersifat non real-time. Diharapkan sistem dapat menghitung kendaraan dengan tingkat keakuratan sebesar 60%. Penelitian ini bersifat basic research atau hanya melakukan perancangan dan mengimplementasikan algoritma untuk menghitung kendaraan pada siang hari dan malam hari. Sehingga, diharapkan sistem ini akan dikembangkan untuk traffic monitoring atau parking monitoring.

Kata kunci: perhitungan kendaraan, frame intersection, previous frame, present frame, dan basic research.

1. Pendahuluan^{[7],[12],[15]}

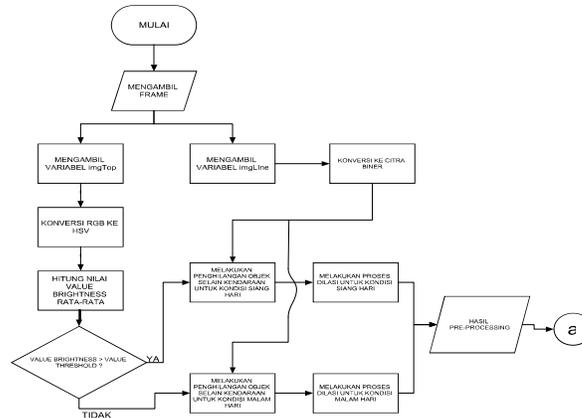
Sistem yang dirancang tidak hanya akan mengembangkan algoritma yang sudah dikerjakan pada penelitian sebelumnya mengenai sistem monitoring parkir tetapi sistem ini dapat digunakan sebagai solusi dari perhitungan kendaraan yang secara manual dengan menggunakan alat penghitung mekanik yang dioperasikan secara langsung oleh manusia. Sehingga diharapkan sistem ini akan menggantikan tugas manusia sebagai penghitung kendaraan di jalan yang terkadang mengalami permasalahan ketidakakuratan perhitungan karena keterbatasan penglihatan manusia. Berdasarkan beberapa permasalahan tersebut, pada penelitian ini akan dibuat sebuah program pada sebuah software komputasi berbasis matriks yang akan menghitung banyaknya kendaraan pada sebuah gambar bergerak atau video, dengan pengambilan gambar atau video yang diambil dari atas suatu jalan dengan kondisi siang hari dan malam hari. Adapun rumusan masalah pada Penelitian ini adalah membuat sistem perhitungan jumlah kendaraan berbasis video processing, membuat sistem nilai akurasi yang optimal, dan mengukur tingkat akurasi dari sistem yang akan dibuat sesuai harapan. Adapun tujuan dari pembuatan penelitian ini adalah sebagai membuat suatu sistem yang dapat menghitung suatu kendaraan pada suatu video yang berguna sebagai input dari sistem tersebut, membuat suatu sistem perhitungan jumlah kendaraan pada malam hari dengan tingkat akurasi yang optimal sebesar 60%, dan menganalisis metode Intersection Frame pada sistem penghitung kendaraan. Beberapa batasan masalah pada penelitian ini agar tidak menyimpang dari permasalahan adalah video yang digunakan Non Real Time, kendaraan yang dihitung hanya pada satu lajur saja, Penelitian ini bersifat Basic Research, video yang digunakan menggambarkan jalan dengan jarak antar kendaraan tidak kurang dari 50 meter atau dalam kondisi lancer, dan pada kondisi malam hari, kondisi jalan hanya mendapatkan penerangan dari lampu mobil atau tanpa lampu jalan. Tahap-tahap dari metode ini adalah studi literatur, pengumpulan data, proses melakukan simulasi, proses pengujian dan analisis, dan pembuatan laporan. Berikut adalah diagram alir sistem :



Gambar 1.1 Diagram Alir Sistem

Dari diagram alir diatas, sistem secara garis besar memiliki dua proses utama, antara lain :

- *Pre-processing*



Gambar 1.2 Diagram Alir *Pre-Processing*

1. Proses inisiasi video

Dalam proses ini, video yang berisi kondisi lalu lintas dengan format *colourspace Red, Green, and Blue*(RGB) akan diambil tiap layer warna merah, hijau, dan biru. Namun, jika video yang digunakan sebagai input adalah video dengan format *YCbCr* maka sebaiknya dilakukan konversi format *colourspace* ke dalam bentuk RGB lalu diubah lagi ke bentuk HSV. Setelah itu, akan diambil *layer Value* saja yang memuat nilai *brightness* pada setiap piksel. Kemudian dilakukan perhitungan nilai rata-rata *layer Value*.

2. Penentuan *Pre-processing*

Pada proses sebelumnya, nilai rata-rata *layer Value* telah didapatkan. Kemudian, hasil tersebut akan ditentukan apakah *frame* berada dalam kondisi siang hari atau malam hari. Jika, nilai rata-rata *Value* berada di bawah nilai *Value Threshold* maka *frame* tersebut akan terdeteksi dalam kondisi malam hari. Begitupun jika nilai rata-rata *Value* berada di atas nilai *Value threshold* maka *frame* tersebut akan terdeteksi dalam kondisi siang hari. Berikut adalah variable atau parameter yang terdapat pada *pre-processing* :

- *Value Threshold*

Parameter ini merupakan syarat tentang nilai minimum dari nilai *Value Brightness*. Parameter ini berguna pada proses *pre-processing* untuk membedakan apakah suatu *frame* berada dalam kondisi siang atau malam hari dan penentuan itu akan menentukan bagaimana suatu *frame* akan diperlakukan dalam mengubah *frame* menjadi citra biner.

- *Strel*

Pada parameter ini terdapat dua jenis *strel* yang digunakan pada pengerjaan penelitian. *Strel Small Object* yang berguna untuk menghilangkan objek-objek kecil selain kendaraan seperti garis pada jalan dengan mengombinasikan *tools imopen* pada Matlab. Sehingga, diharapkan penggunaan *strel* ini dapat menghasilkan citra biner berupa kendaraan saja. Selanjutnya adalah *Strel Dilate* yang berguna untuk menghubungkan dua sinar lampu kendaraan dan berguna juga untuk menyempurnakan hasil pendeteksian kendaraan dalam kondisi siang hari.

- *imgTop*

Variable ini berguna untuk menyimpan secara sementara hasil *crop* pada *frame* bagian tengah hingga atas yang memuat kondisi langit, dan digunakan untuk menentukan *frame* pada kondisi siang ataupun malam hari.

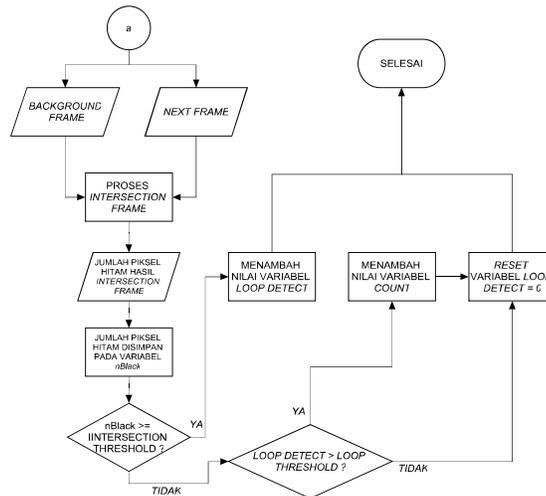
- *imgLine*

Variable ini berguna untuk menyimpan secara sementara hasil *crop* pada *frame* yang memuat kendaraan pada salah satu jalur. Dan variabel inilah yang akan diolah dan dilakukan proses *Intersection Frame*.

3. Citra Biner

Frame-frame yang sudah di inisiasi dan ditaruh pada variabel *imgLine*, selanjutnya akan diubah menjadi citra biner agar mempermudah proses pengirisan antar *frame*. Objek yang terdeteksi sebagai kendaraan akan diberikan direpresentasikan sebagai piksel hitam, sedangkan untuk objek yang diidentifikasi selain kendaraan akan direpresentasikan sebagai piksel putih.

- *Processing*



Gambar 1.3 Diagram Alir *Processing*

- *nBlack*

Variabel ini akan menyimpan secara sementara jumlah piksel hitam pada hasil *Intersection Frame*.

- *Loop Detect*

Variabel ini berfungsi untuk menunjukkan banyaknya proses *loop* dari sistem dalam menemukan suatu objek atau pada sistem ini dideteksi sebagai kendaraan roda empat pada sebuah *frame*.

- *Count Vehicle*

Variable ini digunakan untuk memberitahukan berapa banyak kendaraan yang terdeteksi. Nilai variabel ini saat sistem baru dijalankan bernilai '0'. Variabel akan bertambah nilainya jika pada sebuah *loop* proses didapatkan nilai *Intersected Frame* berada diatas nilai *Intersected Threshold* dan *Loop Detect* berada diatas nilai *Loop Threshold*.

- *Intersection Threshold*

Intersection Threshold merupakan batas nilai maksimum dari jumlah piksel hitam pada *frame* yang didapatkan dari hasil dari salah satu operasi pada proses *Intersection Frame* agar *frame* tersebut dapat didefinisikan sebagai *frame* yang telah berhasil terdeteksi sebagai kendaraan.

- *Loop Threshold*

Loop Threshold merupakan syarat jumlah minimum dari jumlah *frame-frame* yang telah memenuhi syarat bahwa terdapat objek yang diinginkan pada *frame-frame* tersebut secara berurutan pada proses perhitungan.

Berikut adalah penjelasan dari proses perhitungan kendaraan :

1. Penentuan *Frame* Background

Pada tahap ini sistem akan mengambil *frame* sebagai background apabila *frame* tersebut telah diproses pada loop proses sebelumnya, atau *frame* awal pemrosesan

2. Pengambilan *Frame* Baru
Setelah *frame* background sudah di dapatkan, selanjutnya sistem akan mengambil *frame* yang baru dan menamainya sebagai *Next frame* dan akan menaruh *frame* pada *loop* sebelumnya sebagai *Background frame*.
3. Proses Pemisahan *Frame*
Proses ini bermaksud untuk membagi suatu *frame* menjadi dua. Bagian pertama adalah hasil *crop* pada area tertentu dari sebuah *frame* yang berisikan kendaraan atau salah satu jalur jalan dan hasil tersebut akan dinamakan sebagai *imgLine*. Pada bagian kedua akan diambil area tertentu lainnya dari *frame* berupa bagian tengah hingga atas dari sebuah *frame* yang mengandung informasi apakah langit dalam kondisi siang atau malam, hasil dari proses tersebut akan disimpan pada variabel *imgTop*.
4. Konversi RGB ke HSV
Kemudian setiap *frame* yang sudah diambil bagian dari tengah hingga atas akan dilakukan konversi dari format RGB ke HSV, setelah itu akan dilakukan proses perhitungan nilai rata-rata dari *Value Brightness*. Jika nilai rata-rata *Value Brightness* lebih besar dari *Value Threshold*, maka sistem akan menganggap *frame* tersebut berada pada kondisi siang hari, begitupun jika nilai rata-rata *Value Brightness* lebih kecil atau samadengan dari *Value Threshold*, maka sistem akan menganggap *frame* tersebut berada pada kondisi malam hari.
5. *Intersection Frame*
Pada tahap ini *Next frame* dan *Background frame* akan dicari irisan nilainya pada setiap *piksel*. Dengan demikian akan didapatkan persamaan:

$$|Background\ frame|, |Next\ frame| \quad (1.1)$$

6. Perhitungan piksel hitam
Setelah melalui proses *Frame Intersection* dari kedua *frame*, maka akan didapat hasil berupa *frame* irisan. Dalam *frame* ini berisi piksel-piksel yang berwarna hitam. Selanjutnya, dari hasil tersebut akan dilakukan pada tahapan selanjutnya.
7. Penghitungan kendaraan
Pada tahap ini, hasil dari tahap sebelumnya akan dibandingkan dengan *Intersection Threshold*, jika jumlah piksel hitam pada *frame* irisan kurang dari nilai *Intersection Threshold* selanjutnya akan dibandingkan lagi nilai dari variable *Loop Detect* dengan *Loop Threshold*. Perbandingan ini bertujuan untuk menunjukkan kekonsistenan dari syarat sebelumnya yang terpenuhi. *Loop Threshold* merupakan nilai minimum jumlah *frame* irisan yang telah memenuhi syarat pertama untuk kemudian disimpulkan bahwa memang terdapat objek pada *frame* selisih tersebut, sedangkan variabel *Loop Detect* merupakan nilai jumlah *frame* irisan yang telah memenuhi syarat pertama tersebut. Nilai *Loop Detect* akan terus bertambah jika *frame* irisan yang masuk selalu memenuhi syarat secara berturut-turut. Jika dalam proses penambahan nilai variabel *Loop Detect* yang belum mencapai nilai *Loop Threshold* terdapat 1 *frame* irisan yang tidak memenuhi syarat, maka nilai variabel *Loop Detect* akan langsung berubah menjadi 0. Hal ini menunjukkan bahwa nilai *frame* irisan tidak konsisten.

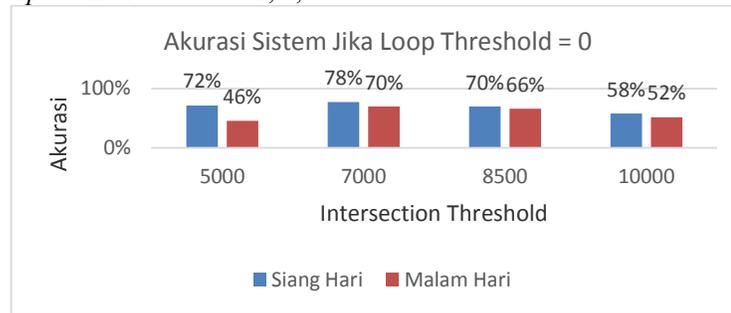
2. Pembahasan

Skenario pengujian dan analisis unjuk kerja sistem ini akan dilakukan sebanyak 3 skenario dengan video kendaraan pada siang hari dengan 24 *frames per second*, dan video kendaraan pada malam hari dengan 14 *frames per second*, dimana kedua video kondisi tersebut sudah diambil tiap *framena* dan disimpan dalam beberapa folder sebanyak masing-masing 10 buah folder dan setiap folder terdapat 5 kendaraan yang lewat. Pada tahap *pre-processing* akan diambil bagian tengah hingga bagian atas dari *frame* sebesar 263x355 piksel. Kemudian akan diambil area tertentu dari sebuah *frame* yang berisikan kendaraan atau salah satu jalur jalan dan hasil tersebut ukuran sebesar 127x141 piksel.

Tabel 2.1 Parameter yang Digunakan pada Pengujian Sistem

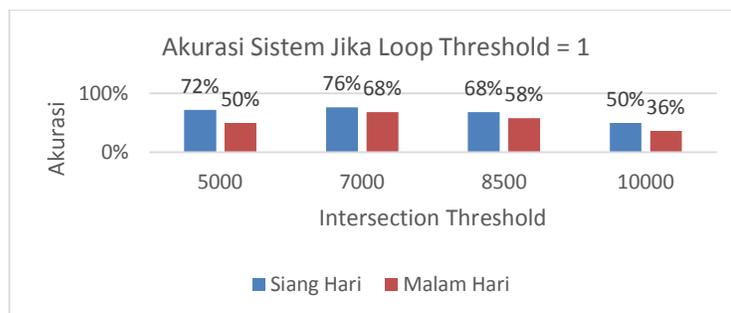
Kondisi	Value Threshold	Level Luminance	Ukuran Strel Small Object	Ukuran Dilasi
				[Baris Kolom]
Siang Hari	0.02	0.4	10	[55 1]
Malam Hari		0.8	1	[1 45]

Setelah ditetapkan parameter pengujian seperti pada Tabel 3.1, kemudian akan dicari nilai akurasi sistem terhadap *Intersection Threshold* sebesar 5000, 7000, 8500, dan 10000 dengan merubah-ubah nilai parameter *Loop Threshold* sebesar 0, 1, dan 2.



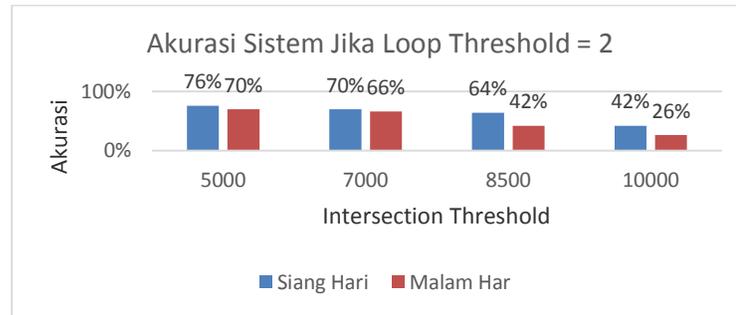
Gambar 2.1 Grafik Akurasi Sistem Jika *Loop Threshold* = 0

Pada grafik 2.1 didapatkan tingkat akurasi tertinggi pada kisaran nilai *Intersection Threshold* 7000 dan 8500 karena nilai dari jumlah piksel hitam pada data uji siang hari maupun malam hari berada pada nilai 5000 dan 9000, sehingga jumlah piksel hitam hasil proses *intersection frame* tidak berada jauh pada nilai tersebut. Dan oleh sebab itu juga, nilai akurasi cenderung lebih tinggi untuk kedua kondisi jika diberikan nilai *Intersection Threshold* dibawah 10000. Jika dilihat kembali grafik diatas, ketika nilai *Intersection Threshold* 5000, tingkat akurasi menurun menjadi sebesar 46% .Hal tersebut terjadi karena sistem mengalami kenaikan sensitivitas yang tinggi khususnya pada kondisi malam hari dengan ditandai banyak objek kendaraan yang dideteksi lebih dari satu kali khususnya pada video ke-2 (malam hari). Kemudian, ketika nilai *Intersection Threshold* sebesar 10000 untuk siang dan malam hari penurunan tingkat akurasi terjadi karena terdapat objek/kendaraan pada masing-masing video yang tidak terdeteksi dan terhitung, bahkan pada kondisi malam hari pada video ke-1 dan ke-2 sama sekali tidak ada objek yang terdeteksi.



Gambar 2.2 Grafik Akurasi Sistem Jika *Loop Threshold* = 1

Pada grafik 2.2 dengan perubahan *Loop Threshold* sebesar 1 tingkat akurasi tertinggi masih terjadi pada nilai *Intersection Threshold* sebesar 7000 khususnya pada kondisi siang hari meskipun terjadi penurunan tingkat akurasi jika dibandingkan dengan tingkat akurasi sistem pada skenario pengujian ke-2 (Metode Penyempurnaan Bentuk). Kemudian, hal tersebut menunjukkan bahwa dengan penambahan nilai *Loop Threshold* sebesar 1 mampu mengurangi kesalahan sistem dalam membaca satu objek/kendaraan terdeteksi dan terhitung lebih dari sekali. Kemudian, pada grafik diatas didapatkan tingkat akurasi tertinggi untuk kondisi siang hari sebesar 76% dan untuk kondisi malam hari sebesar 68%.



Gambar 2.3 Grafik Akurasi Sistem Jika *Loop Threshold* = 2

Dari Gambar 2.3, dapat dilihat bahwa akurasi tertinggi untuk kondisi siang dan malam hari terjadi ketika nilai *Intersection Threshold* sebesar 5000. Kemudian nilai akurasi terendah terjadi ketika nilai *Intersection Threshold* sebesar 10000. Hal tersebut menunjukkan semakin besar nilai *Loop Threshold* akan semakin menurunkan nilai akurasi jika nilai *Intersection Threshold* diatas rata-rata jumlah piksel hitam pada hasil proses *Intersection Frame*. Namun, akan meningkatkan nilai akurasi untuk nilai-nilai *Intersection Threshold* yang menyebabkan sensitivitas sistem menjadi tinggi atau pada nilai *Intersection Threshold* yang rendah. Sehingga, jika sistem ingin menggunakan *Loop Threshold* sebesar 2 maka sebaiknya juga diikuti dengan perubahan nilai *Intersection Threshold* sebesar 5000.

3. Simpulan

1. Sistem penghitung kendaraan roda empat dalam kondisi siang dan malam hari dengan metode *frame intersection* masih memiliki keterbatasan terhadap parameter tertentu khususnya pada parameter *Intersection Threshold*, dimana hal tersebut dapat mempengaruhi tingkat akurasi.
2. Tingkat akurasi terbaik berdasarkan pengujian yang dilakukan adalah ketika parameter *Intersection Threshold* = 7000 dengan *Loop Threshold* = 0.
3. Sistem memiliki kendala dalam menghitung ketika dua kendaraan berada dalam jarak yang dekat, sehingga kedua kendaraan tersebut masih terdeteksi sebagai satu kendaraan.
4. Sistem yang direalisasikan sudah mampu menghilangkan objek-objek selain kendaraan khususnya garis jalan meskipun pada beberapa *frame* masih terdapat sedikit garis jalan.
5. Kemudian, dari percobaan yang dilakukan didapatkan bahwa untuk kondisi malam hari, lampu dari kendaraan dapat digunakan sebagai objek yang dapat dideteksi dan diproses (dihitung).
6. Untuk kondisi malam hari, proses dilasi dapat dilakukan dengan memperbesarnilai kolom sehingga dua sinar lampu dapat digabung menjadi satu objek.
7. Tingkat akurasi sistem tidak mencapai 100% karena dipengaruhi oleh spesifikasi perangkat keras yaitu kamera yang berasal dari smartphone atau memiliki kualitas video yang belum baik jika dibandingkan kamera webcam ataupun kamera konvensional, alur dari pre-processing masih belum baik dalam mendeteksi kendaraan karena tidak melakukan pengolahan gambar pada variable *imgLine* pada domain RGB, dan sifat acak pada kendaraan di setiap video.

Daftar Pustaka

- [1] Anonim, *Operation Manual : SONY VEGAS PRO 11.0*, SONY Corporation, Japan, 2011
- [2] Anonim, *Edit Colors : Paint 6.2(Build 9200)*, Microsoft Corporation, United States, 2012
- [3] Anonim, *Product Help : MATLAB 7.8.0 (R2009a)*, The MathWorks, United States, 2009
- [4] Kuo, Sen, M dkk. 2006. *Real Time Signal Processing*. Edisi ke 2. London.
- [5] Prasetyo, Eko. 2012. *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*. Yogyakarta. Penerbit Andi.
- [6] Ryanto. "Praktikum 2 Dasar Pengolahan Citra (1)". diunduh pada 30, November, 2013. <http://lecturer.eepis-its.edu/~riyanto/citra-bab2.pdf>.
- [7] Sistama,Ranggi.2016. Simulasi Monitoring Objek yang Masuk dan Keluar untuk Mengontrol Ketersediaan Lahan Menggunakan *Video Processing*.Bandung : Telkom University
- [8] Wijaya, Marvin Ch dan Agus Priyono. 2007. *Pengolahan Citra Digital Menggunakan MatLab Image Processing Toolbox*. Bandung. Informatika.

- [9] Wikipedia. "HSL and HSV". 16 Desember 2016.
<http://www.lps.usp.br/hae/apostila/basico/HSI-wikipedia.pdf>
- [10] Wade, I. "Hue-Saturation-Value Representation and Correlation of Multispectral/Multi-Modal Datasets". 16 Desember 2016.
http://thesis.library.caltech.edu/6027/10/10_Appendix_F_LWade.pdf
- [11] Fisher, R, dkk. "Dilation". 16 Desember 2016.
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>
- [12] Dewantoro, Adrian Kurnia. 2015. Simulasi dan Analisis Sistem Penghitung Kepadatan Lalu Lintas dan Klasifikasi Kendaraan Berbasis Webcam dengan Metode *Background Substraction*. Bandung : Telkom University .
- [13] Syukur, Muhammad. 2009. Menentukan Kepadatan Lalu Lintas dengan Penghitungan Jumlah Kendaraan Berbasis *Video Processing*. Bandung : Telkom University.
- [14] Putri, Riski Dahlia I. 2010. Desain dan Simulasi Penghitung dan Pengidentifikasian Panjang Antrian pada Lampu Lintas dengan Metode Pengolahan *Digital*. Bandung : Telkom University.
- [15] Berlian W., Marnala. 2010. *Simulasi dan Analisis Sistem Penghitung Kepadatan Lalu Lintas Berbasis Kamera Digital dan Pengolahan Citra Digital*. Bandung : Telkom University.
- [16] Wildan. 2011. Sistem Penghitung Kecepatan Sesaat Kendaraan Berbasis *Camcorder* dengan Metode *Background Substraction*. Bandung : Telkom University.