

SEGMENTASI KENDARAAN MENGGUNAKAN *IMPROVE BLOB ANALYSIS* (BA) PADA VIDEO LALU LINTAS

Sutrisno¹, Imam Cholissodin², Rina Christanti³, Candra Dewi⁴, Nurul Hidayat⁵

¹²³⁴⁵Program Studi Informatika dan Ilmu Komputer, Universitas Brawijaya
Email: ¹trisno@ub.ac.id, ²imamcs@ub.ac.id, ³rnyun@gmail.ac.id, ⁴dewi_candra@ub.ac.id, ⁵ntayadih@ub.ac.id

(Naskah masuk: 29 Januari 2015, diterima untuk diterbitkan: 17 Februari 2015)

Abstrak

Penggunaan citra digital untuk keperluan penelitian sudah banyak dilakukan, salah satunya yaitu segmentasi. Segmentasi berfungsi untuk mendeteksi objek - objek yang terdapat pada citra, sehingga hasil segmentasi sangat penting untuk proses selanjutnya. Pada penelitian ini diusulkan teknik optimasi hasil *background subtraction* menggunakan kombinasi *frame difference* (FD) atau *difference image* dengan filter SDGD dan *running average* (RA) atau *background updating* dengan filter SDGD untuk diterapkan pada *blob analysis*. Alasan utama menggunakan penggabungan kedua metode tersebut adalah karena seringkali terdapat piksel objek yang tidak mampu dideteksi sehingga akan mengurangi tingkat optimasi pengenalan objek. Hasil pengujian akurasi dari 10 data uji yang masing – masing terdiri dari 30 *frame* menunjukkan bahwa aplikasi ini memiliki nilai akurasi tertinggi yakni 90% untuk pengujian *threshold* dan 100% untuk pengujian ukuran *structure element*. Sehingga dapat disimpulkan bahwa aplikasi ini mampu melakukan segmentasi kendaraan dengan baik.

Kata kunci: *filter SDGD, blob analysis, video lalu lintas, background subtraction.*

Abstract

The use of digital images for the purposes of research has been often applied, one of them is segmentation. Segmentation is used to detect objects contained in the image, so the segmentation result is very important for further processing. In this study, the results of the optimization technique proposed background subtraction using a combination of frame difference (FD) or a difference image with filter SDGD and running average (RA) or background updating with SDGD filter to be applied blob analysis. The main reason to use the merger of these two methods is that often there are pixels that are not able to detect objects that will reduce the level of optimization object recognition. The results of accuracy testing using 10 data testing for each data consisting of 30 frames shows that the system proposed in this paper has best accuracy of 90% for testing the threshold and 100% for testing the size of structure element. So it can be concluded that this system capable to segmentation the vehicle properly.

Keywords: *filter SDGD, blob analysis, traffic video, background subtraction*

1. PENDAHULUAN

Indonesia merupakan negara yang memiliki banyak kota besar. Akan tetapi, dari sekian banyak kota tersebut ternyata sering mengalami permasalahan dalam bidang transportasi. Salah satu permasalahan tersebut adalah kemacetan. Kemacetan merupakan problem utama dalam transportasi. Meskipun telah setiap ruas jalan telah dilengkapi dengan alat pengatur lalu lintas yang lengkap, namun alat tersebut hanya mampu untuk mengatur berhenti dan berjalannya kendaraan tanpa adanya dukungan teknologi yang mampu untuk mengestimasi tingkat kepadatan kendaraan disetiap ruas jalan untuk membuat waktu berhenti dan berjalannya secara adaptif. Selain itu, bertambahnya jumlah kendaraan roda dua maupun roda empat juga merupakan faktor utama dalam kemacetan lalu lintas. Hal ini memang dipicu pula dengan sedikitnya rencana pembangunan jalan raya disetiap daerah di Indonesia, sehingga membuat tidak seimbang antara kapasitas jalan raya

dengan jumlah kendaraan yang ada. Dan tentunya, permasalahan transportasi ini pada beberapa tahun mendatang dapat dipastikan akan semakin kompleks jika tidak segera diatasi sedini mungkin.

Efek secara tidak langsung dari kemacetan ini adalah terkait dengan penurunan produktifitas dari setiap pengguna jalan raya. Karena waktu mereka akan habis diperjalanan untuk setiap hari – harinya. Kemudian terkait dengan polusi udara di jalan raya pun akan semakin meningkat dan tidak baik bagi kesehatan serta pada lingkungan. Atau bahkan akan menimbulkan pelanggaran lalu lintas dengan menentang arus untuk cepat sampai di tempat tujuan tanpa memperdulikan marka jalan, misalnya melewati trotoar dan menentang arus jalan, yang tanpa disadari oleh pengendara kadang menimbulkan kecelakaan atau membahayakan pengendara lain maupun pejalan kaki. Jika dianalisis dari beberapa kejadian di lalu lintas, maka dapat dikatakan bahwa perbandingan sebenarnya waktu tempuh normal dengan waktu tempuh karena adanya kemacetan bisa mencapai 2 sampai 3 kali lipat. Hal inilah yang

kadang – kadang membuat seseorang terlambat sampai di tempat kerja, terlambat mengantarkan pesanan, sampai pada terlambat untuk mengikuti ujian dan lain sebagainya. Fakta – fakta di atas merupakan beberapa permasalahan utama dalam transportasi yang harus segera ditangani dengan serius. Oleh karena itu perlu dibuat suatu sistem yang mampu bekerja secara otomatis dan dapat membantu dalam mengatasi permasalahan – permasalahan tersebut. Sistem tersebut akan dilengkapi dengan kecerdasan buatan dari mesin pembelajaran yang dapat melatih secara mandiri dari setiap kejadian – kejadian yang menjadi data masukan dan mampu memproses, sehingga mendapatkan keputusan yang optimal.

Pada penelitian sebelumnya yang dilakukan oleh Thou-Ho (2007), telah dicoba dibuat sistem penghitungan jumlah kendaraan mobil dan sepeda pada jalan raya menggunakan *blob analysis*. Dari hasil uji coba yang dilakukan didapatkan hasil rata-rata akurasi sebesar 91.7% pada dua kondisi lalu lintas kendaraan, yaitu *flow1* (*bi-directional*/dua arah), *flow2* dan *flow3* (*uni-directional*/satu arah). *Blob analysis* tersebut didapatkan dari proses *learning* citra *sequence* yang diambil dari data *input video* dalam *frame* tertentu. Proses *learning* yang dilakukan pertama kali adalah dengan melakukan segmentasi objek bergerak, dalam hal ini adalah kendaraan, yang didasarkan pada *background subtraction* dari penghitungan selisih *current frame* dengan *background* awal. *Background* awal ini nantinya akan terus menerus di-update secara *real-time* menggunakan model *updating* dengan faktor pembobot terhadap *current frame*. Namun faktor pembobot tersebut harus di-*set* secara manual, dan tidak bersifat adaptif. Kemudian, pada proses *tracking* objek menggunakan fungsi *match* yang digunakan untuk menandai bahwa objek sedang bergerak adalah objek bergerak sebelumnya, sehingga sistem akan tetap mengenali objek – objek tersebut secara unik (Thou-Ho dkk. 2007).

Penelitian tentang *adaptive background subtraction* yang telah dilakukan Ruolin Zhang dan Jian Ding (2012), dicoba untuk mendapatkan nilai *threshold* yang adaptif untuk hasil citra *subtraction* dari citra di *current frame* dengan citra *background*. Namun nilai *threshold* tersebut juga masih menggunakan cara *trial-and-error* beberapa kali percobaan. Mereka mengatakan bahwa nilai *threshold* harus disesuaikan dengan lingkungan yang ada pada sekitar objek untuk menjamin hasil *background subtraction* yang optimal. Farah Yasmin Abdul Rahman dkk (2013) mengembangkan model *background subtraction* (BGS) menggunakan *Second Derivative in Gradient Direction Filter* (SDGD). Model tersebut dibuat dengan teknik integrasi dengan filter *gradient-based edge detector*. Filter SDGD tersebut digunakan untuk memperbaiki hasil BGS yang masih belum sempurna yang dilakukan dengan empat teknik yang sebelumnya sudah ada, yaitu

frame difference (FD), *approximate median* (AM), *running average* (RA), dan *running Gaussian average* (RGA). Hal yang mendasari pengembangan filter tersebut adalah karena terdapat intensitas warna dari objek yang lebih kecil dibandingkan dengan nilai *threshold* dan ukuran *blob* hasil *tracking*-nya terlalu kecil dan kurang sesuai dengan ukuran sebenarnya dari objek bergerak. Selain itu, filter SDGD juga digunakan untuk memperbaiki hasil deteksi tepi dari teknik dasar BGS yang ada sebelumnya terhadap piksel – piksel yang hilang. Dari hasil pengujian, terbukti bahwa kombinasi antara FD dengan filter SDGD memberikan hasil *pre-proses* terbaik pada algoritma jaringan syaraf tiruan (JST) untuk digunakan dalam membedakan antara objek manusia dan bukan manusia dengan akurasi terbaik sebesar 98.75% (Rahman, F. Y. A. dkk. 2013).

Pada penelitian ini diusulkan teknik optimasi hasil *background subtraction* menggunakan kombinasi *frame difference* (FD) atau *difference image* (pada paper sebelumnya oleh Thou-Ho, 2007) dengan filter SDGD dan *running average* (RA) atau *background updating* (pada paper sebelumnya oleh Thou-Ho, 2007) dengan filter SDGD untuk diterapkan *blob analysis*. Alasan utama menggunakan penggabungan kedua metode tersebut adalah karena seringnya terdapat piksel objek yang tidak mampu dideteksi sehingga akan mengurangi tingkat optimasi pengenalan objek. Diharapkan dengan menggunakan teknik tersebut akan diperoleh hasil uji coba yang optimal.

2. TINJAUAN PUSTAKA

2.1. *Second Derivative In Gradient Direction* (SDGD) Filter

Dalam studi pengolahan citra, peneliti menggunakan derivatif pertama dan kedua untuk mendeteksi tepi objek berdasarkan gradien nya. Dengan menggunakan turunan pertama, lokasi tepi didefinisikan pada posisi maksimum yang curam dan menurun (*descent*) (Young, I. J. dkk. 2007). Metode deteksi tepi tradisional, seperti Prewitt, Sobel, dan Roberts, mengkonvolusi gambar dengan kernel tertentu (Young, I. J. dkk. 2007, Narendra, V. G. dkk. 2009). Namun, teknik ini menjadi sensitif terhadap *noise* dan kurang akurat (Narendra, V. G. dkk. 2009). Pada tahun 1986, detektor tepi Canny diperkenalkan, yang mewakili perbaikan atas metode tradisional (Narendra, V. G. dkk. 2009, Persoon, M. P. dkk. 2003). Detektor menerapkan *smoothing* Gaussian untuk mengurangi *noise*, segmen yang tidak diinginkan, tekstur, *nonmaximum suppression*, dan *hysteresis thresholding* untuk menemukan tepi (Young, I. T. 1996).

Pendekatan *second-order derivative* mendefinisikan piksel tepi berdasarkan perubahan kecerahan atau persimpangan nol (*zero crossing*) pada bidang gambar (Young, I. T. 1996, Anonymous. 2011). SDGD adalah operator nonlinear yang dapat dinyatakan dalam turunan pertama dan kedua. Selain

mirip dengan Canny, SDGD juga dikombinasikan dengan *low pass filter* Gaussian untuk tujuan *smoothing* (Verbeek, P. W. 1994). Selain itu, operator Laplace digunakan untuk menyederhanakan operasi SDGD (Hagara, M. & Moravcik, J. 2002).

Operator Laplacian didefinisikan pada Persamaan (1)

$$\nabla^2 = \frac{\partial^2 a}{\partial x^2} + \frac{\partial^2 a}{\partial y^2} = (h_{2x} \otimes a) + (h_{2y} \otimes a) \quad (1)$$

dimana h_{2x} dan h_{2y} , adalah filter turunan kedua (*second derivative*) dan a adalah citra.

Cara mendasar dalam filter turunan kedua diberikan Persamaan (2).

$$\begin{aligned} [h_{2x}] &= [h_{2y}] = [1 \quad -2 \quad 1], \\ [h_2] &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{aligned} \quad (2)$$

Mengaitkan Persamaan (1) dengan hasil filter Gaussian

$$b = g_\sigma \otimes (h_2 \otimes a) = (g_\sigma \otimes h_2) \otimes a \quad (3)$$

dimana b adalah hasil dari operasi *Gaussian Filter*, g_σ adalah *Gaussian low pass filter* dan h_2 hasil dari laplacian.

Lima derivatif parsial yang digunakan dalam filter SDGD adalah sebagai berikut:

$$\begin{aligned} A_{xx} &= \frac{\partial^2 a}{\partial x^2}, & A_{xy} &= \frac{\partial^2 a}{\partial x \partial y}, & A_x &= \frac{\partial a}{\partial x}, \\ A_{yy} &= \frac{\partial^2 a}{\partial y^2}, & A_y &= \frac{\partial a}{\partial y}. \end{aligned} \quad (4)$$

Oleh karena itu,

$$SDGD = \frac{A_{xx}A_x^2 + 2A_{xy}A_xA_y + A_{yy}A_y^2}{A_x^2 + A_y^2} \quad (5)$$

Penjelasan detail tentang SDGD dapat ditemukan di (. 2011, Verbeek, P. W. 1994, El-Glaly, Y. N. 2007).

Berdasarkan kajian yang dilakukan oleh Persoon dkk (2003), filter SDGD memberikan hasil lokalisasi permukaan yang lebih baik, terutama di daerah yang sangat melengkung, dibandingkan dengan teknik deteksi tepi Canny. Dengan demikian, filter tersebut diadopsi dalam dalam penelitian ini. Selain itu, Persoon dkk (2003) menunjukkan bahwa SDGD menjamin minimal detail *smoothing* yang menyebabkan visualisasi yang lebih baik dari polip di *computed tomography* (CT) scan data. Selanjutnya, kajian yang dilakukan oleh Nader El-Glaly (2007) menggunakan filter SGD dalam mengembangkan parsial-diferensial pada algoritma digital *inpainting* untuk mencari data yang hilang dalam gambar digital.

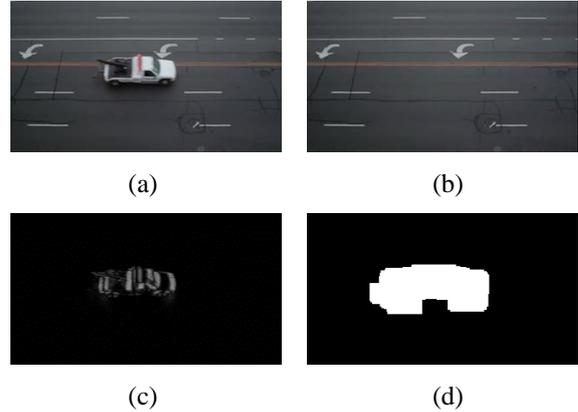
2.2. Algoritma Vehicle Analysis

Pada umumnya, kamera lalu lintas diletakkan pada tempat tertentu, karenanya *background*-nya diam. Oleh karena itu, metode *background subtraction* (Thou-Ho dkk. 2007) sangat cocok untuk mendeteksi kendaraan yang bergerak melalui perbedaan yang

dihasilkan untuk setiap deteksi perubahan gambar. Fungsi deteksi ditunjukkan sebagai berikut:

$$D_i(x, y) = C_i(x, y) - B_i(x, y) \quad (6)$$

dimana $D_i(x,y)$ adalah hasil *difference image*, $C_i(x,y)$ adalah *current image* pada Gambar 1(a) dan $B_i(x,y)$ adalah *background image* pada Gambar 1(b).



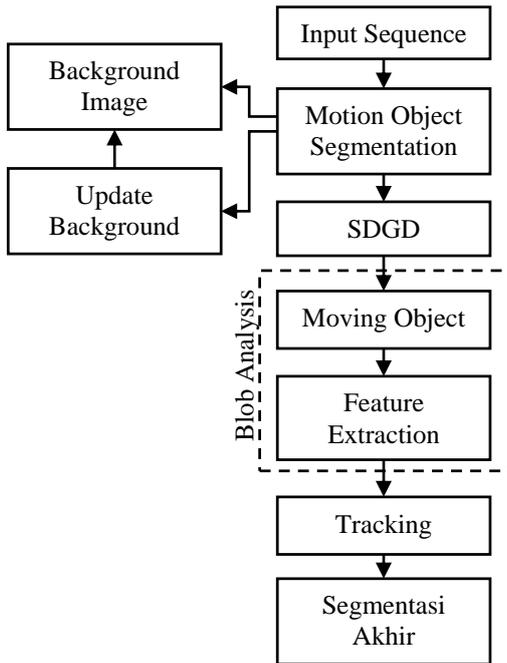
Gambar 1. Hasil dari *Vehicle Segmentation*

Hasil dari *difference image* pada Gambar 1(c) akan diubah menjadi citra biner dengan persamaan berikut:

$$R_i(x, y) = \begin{cases} 0, & \text{if } |D_i(x, y)| < T \\ 255, & \text{otherwise} \end{cases} \quad (7)$$

dimana $R_i(x,y)$ adalah hasil citra biner dan T adalah nilai *threshold* yang nantinya ditetapkan dari hasil beberapa kali percobaan. Dan pada Gambar 1(d) merupakan hasil SDGD kemudian dilakukan operasi morfologi dengan menerapkan teknik erosi dan dilasi. Detail operasi algoritma Segmentasi Kendaraan dengan *Improve BA* dapat dilihat pada Gambar 2.

Pada Gambar 2 dijelaskan cara kerja algoritma tersebut adalah serangkaian masukkan objek akan disegmentasi, objek tersebut akan mengalami *update background* secara terus – menerus sampai seluruh rangkaian masukkan objek tersegmentasi seluruhnya agar membentuk *reliable background* dan setelah itu dilakukan filter SDGD. Kemudian melakukan *blob analysis* yaitu ekstraksi fitur seperti perhitungan *dispersedness*, *aspect ratio*, dan *area ratio* pada objek bergerak. Setelah itu dilakukan *tracking* dan segmentasi akhir.



Gambar 2. Algoritma Segmentasi Kendaraan dengan *Improve BA*

2.3. Ekstraksi Fitur untuk *Tracking*

Pada Gambar 2 dibagian *blob analysis*, ada banyak fitur yang dari objek yang menjadi target bergerak (*moving target*) seperti tekstur, warna, bentuk, dll. Fitur-fitur ini secara garis besar dapat diklasifikasikan menjadi dua bagian: fitur spasial dan fitur temporal. Fitur spasial digunakan untuk membedakan objek yang berbeda pada waktu yang sama, dan fitur temporal digunakan untuk mengenali objek yang sama pada waktu yang berbeda. Untuk mengenali objek yang berbeda, maka perlu untuk mendapatkan fitur tertentu yang bermakna dan bersifat diskriminatif.

Ketika kendaraan bergerak, fitur yang diekstrak adalah *perimeter* dan *area*, yang kemungkinan dapat berubah pada waktu ekstraksi yang berbeda. Untuk mengatasi masalah fitur yang nilainya sering berubah diambil dari objek yang bergerak menggunakan *bounding-box* (Lin, B. & Zhou, H. 2002). Oleh karena itu fitur lainnya seperti *dispersedness*, *aspect ratio*, dan *area ratio* juga diukur untuk menyediakan fitur yang stabil pada objek yang bergerak, seperti dijelaskan dalam Persamaan (8), (9), dan (10).

$$Dispersedness = \frac{Perimeter^2}{Area} \tag{8}$$

$$AspectRatio = \frac{Height}{Width} \tag{9}$$

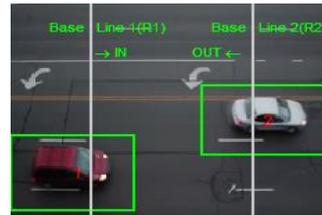
$$AreaRatio = \frac{Area}{ROI} \tag{10}$$

Dalam persamaan di atas, *Perimeter* berarti batas objek bergerak dan *Area* menunjukkan wilayahnya, tetapi *Height*, *Width* dan *ROI* berarti ketinggian, lebar dan luas (yaitu, *Height*Width*) dari *bounding-box* masing – masing. Kemudian

menghitung *centroid* dari masing – masing objek sesuai dengan Persamaan (11).

$$x_0 = \frac{\sum_{(x,y) \in R} x}{\sum_{(x,y) \in R} 1}, y_0 = \frac{\sum_{(x,y) \in R} y}{\sum_{(x,y) \in R} 1} \tag{11}$$

Pada Gambar 3, masing-masing objek ditentukan titik (x_0, y_0) yang merupakan *centroid* objek dan R adalah himpunan *pixel* objek.



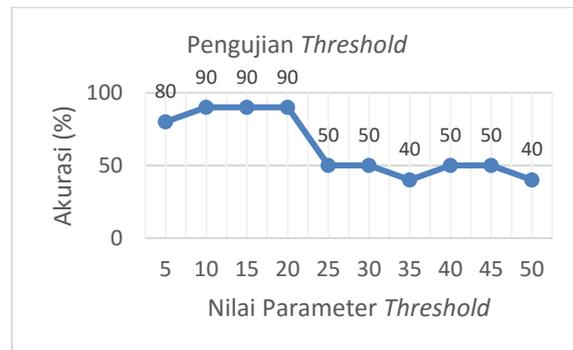
Gambar 3. Hasil Segmentasi Kendaraan

3. HASIL PENGUJIAN DAN ANALISIS

Data yang digunakan untuk pengujian berupa file video (.avi) yang diunduh dari internet dengan *frame rate* 24 fps dan ukuran video 1280x720 piksel. Data video kemudian diekstraksi menjadi gambar (.png) dan diperkecil menjadi 320x280 piksel. Spesifikasi komputer yang digunakan dari perangkat keras yakni prosesor Intel® Core™ i3-2330M CPU @ 2.20GHz, memori 4 GB, dan dari perangkat lunak menggunakan sistem operasi Windows 10 Pro 64 bit serta aplikasi Matlab. Waktu yang digunakan untuk proses pengujian satu data uji (30 *frame*) adalah 3 detik.

3.1. Pengujian *Threshold*

Pengujian *threshold* dilakukan dengan tujuan untuk mengetahui nilai *threshold* terbaik sehingga dapat menyeleksi citra hasil *background subtraction* dan membentuk gambar biner dengan baik. Pengujian ini dilakukan dengan mengganti nilai *threshold* dengan 10 nilai parameter *threshold*.



Gambar 4. Analisis Pengujian *Threshold*

Berdasarkan hasil pengujian *threshold* dengan 10 data uji, yakni 1 data uji dengan 30 *frame* dan 10 nilai parameter *threshold* didapatkan bahwa *threshold* 10, 15 dan 20 memiliki nilai akurasi

tertinggi yakni 90%. Dari hasil ini, dapat disimpulkan bahwa *threshold* 10, 15 dan 20 merupakan *threshold* terbaik.

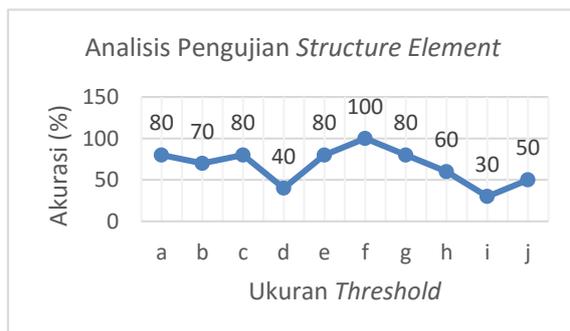
Nilai akurasi dari *threshold* mengalami kenaikan dan penurunan akurasi. Dari Gambar 4 dapat dilihat bahwa nilai parameter *threshold* yang mendekati *threshold* 10, 15 dan 20 akurasinya semakin tinggi dan nilai akurasi yang menjauhi *threshold* 10, 15 dan 20 akurasinya semakin rendah. Hal ini disebabkan karena jika nilai *threshold* semakin kecil atau semakin besar maka sistem tidak mampu menyeleksi *background* dengan baik. Jika *threshold* semakin kecil, maka *background* dapat terdeteksi sebagai objek. Sedangkan jika *threshold* semakin besar, *blob – blob* objek akan menjadi lebih besar, sehingga dapat menyatu dengan *blob* yang lain.

3.2. Pengujian Structure Element

Pengujian *structure element* dilakukan dengan tujuan untuk mengetahui ukuran *structure element* terbaik sehingga dapat membentuk *blob* dengan baik. Pengujian ini dilakukan dengan mengganti nilai *structure element* dengan 10 ukuran *structure element* pada Tabel 1.

Tabel 1. Ukuran Structure Element

SE	Ukuran	
	Erosi	Dilasi
a	5x1	10x20
b	10x1	10x20
c	15x1	10x20
d	5x1	20x40
e	10x1	20x40
f	15x1	20x40
g	5x2	15x25
h	10x2	15x25
i	15x2	15x25
j	5x2	20x40



Gambar 5. Analisis Pengujian Structure Element

Berdasarkan hasil pengujian *structure element* dengan 10 data uji pada Gambar 5, yakni 1 data uji dengan 30 *frame* dan 10 ukuran *structure element* (Tabel 1) didapatkan bahwa *structure element* dengan erosi 15x1 dan dilasi 20x40 memiliki nilai akurasi tertinggi yakni 100%. Dari hasil ini, dapat disimpulkan bahwa ukuran *structure element* dengan

erosi 15x1 dan dilasi 20x40 merupakan ukuran *structure element* terbaik.

Ukuran *structure element* yang tepat akan menghasilkan segmentasi *blob* yang baik. Jika ukuran erosi besar dan tidak diimbangi dengan dilasi yang tepat, maka *blob* akan terlihat lebih kecil dari ukuran sebenarnya atau dapat menyebabkan sebuah *blob* akan terpecah menjadi beberapa *blob*. Tetapi jika ukuran erosi terlalu kecil juga tidak mampu menghilangkan *noise* yang tidak diperlukan. Sebaliknya jika ukuran dilasi terlalu besar, maka akan menyebabkan ukuran *blob* lebih besar dari pada objek yang sebenarnya dan dapat menyebabkan beberapa *blob* menyatu menjadi satu *blob*.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

1. Implementasi *Blob Analysis* dan SDGD dapat digunakan untuk segmentasi kendaraan pada video lalu lintas.
2. Hasil ekstraksi fitur bergantung pada *blob* yang dihasilkan sehingga ukuran *threshold* dan *structure element* harus tepat untuk menghasilkan *blob* yang baik. *Threshold* terbaik adalah *threshold* 10, 15, dan 20 dengan akurasi 90%, sedangkan untuk ukuran *structure element* terbaik adalah erosi 15x1 dan dilasi 20x40 dengan akurasi 100%.

4.2 Saran

1. Diharapkan pada penelitian berikutnya dapat menggunakan data video yang lebih banyak dan dapat mewakili sebagian besar dari variasi jalan dan kendaraan serta dapat ditambahkan juga data dalam bentuk video yang memiliki kualitas resolusi yang tinggi untuk menunjang kelengkapan data sekaligus dapat juga digunakan analisis dalam bentuk data video pada sembarang posisi dengan harapan hal ini akan mampu mengoptimalkan hasil dari nilai akurasi yang didapatkan.

5. DAFTAR PUSTAKA

- ANONYMOUS. 2011. Image processing fundamentals: derivative-based operation. <http://www.mif.vu.lt/atpazinimas/dip/FIP/fip-Derivati.html>.
- EL-GLALY, Y. N. 2007. Development of PDE-Based Digital Inpainting Algorithm Applied to Missing Data in Digital Images [M.S. thesis]. Ain Shams University.
- HAGARA, M & MORAVCIK, J. 2002. PLUS operator for edge detection in digital images. in *Proceedings of the International Conference of Radioelektronika*. 467–470.

- LIN BO, ZHOU HEQIN. 2003. Using object classification to improve urban traffic monitoring system. *IEEE, International Conference on Neural Networks and Signal Processing*, Vol. 2. 1155-1159.
- NARENDRA, V. G. & HAREESH, K. S. 2009. Study and comparison of various image edge detection techniques. *International Journal of Image Processing*, vol. 4, no. 2, article 83.
- PERSOON, M. P., SERLIE, I. W. O., POST, F. H., TRUYEN, R., & VOS, F. M. 2003. Visualization of noisy and biased volume data using first and second order derivative techniques. *in Proceedings of the 14th IEEE Visualization Conference*. 379–385.
- RAHMAN, F. Y. A. 2013. Enhancement of Background Subtraction Techniques Using a Second Derivative in Gradient Direction Filter. Hindawi Publishing Corporation, *Journal of Electrical and Computer Engineering*.
- THOU-HO, YU-FENG, L., & TSONG-YI, C. 2007. Intelligent Vehicle Counting Method Based on Blob Analysis in Traffic Surveillance. *IEEE*, 0-7695-2882-1/07 \$25.00 ©2007.
- YOUNG, I. J., GERBRANDS, J.J., & VAN VLIET, L. J. 2007. *Fundamentals of Image Processing*. Version 2. 3.
- YOUNG, I. T. 1996. Generalized convolutional filtering. *in Proceedings of the 19th CERN School of Computing*. 51–65.
- VERBEEK, P. W. & VAN VLIET, L. J. 1994. Location error of curved edges in low-pass filtered 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 7. 726–733.