

Deteksi *Zebra Cross* Pada Citra Digital Dengan Menggunakan Metode Hough Transform

Fitria Indriani¹, Fitri Utamingrum², Yuita Arum Sari³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹ia.fitri4@gmail.com, ² f3_ningrum@ub.ac.id, ³yuita@ub.ac.id

Abstrak

Tingginya angka kecelakaan yang mencederai pejalan kaki saat menyeberang disebabkan oleh kendaraan bermotor yang kurang berhati-hati. Kecelakaan yang tentunya tidak diinginkan dapat dicegah dan diminimalisir budaya tertib berlalu-lintas salah satu dengan menggunakan fasilitas seperti *zebra cross*. Pada penelitian ini, kami mengusulkan proses deteksi *zebra cross* pada citra digital menggunakan metode *Hough Transform*, dengan tujuan dapat diimplementasikan pada sistem navigasi kendaraan pintar dalam mengidentifikasi *zebra cross* guna peningkatan keselamatan baik pengendara maupun pengguna *zebra cross*. Proses deteksi *zebra cross* dimulai dari *pre-processing*, yang terdiri dari proses *grayscale*, *mean filtering*, *dilasi*, dan *histogram equalization*, untuk deteksi tepi kami menggunakan metode *canny* tahap selanjutnya adalah inversi gambar yang bertujuan untuk merubah piksel-piksel yang berwarna putih menjadi hitam, dan sebaliknya. Kemudian untuk deteksi garis pada *zebra cross* menggunakan metode *hough transform*. Berdasarkan pengujian, nilai akurasi tertinggi ketika nilai *threshold* 100 pada data uji kondisi pagi pertama sebesar 95,2%. Hasil dari pengujian variasi *structure element* didapatkan hasil maksimal dengan penggunaan bentuk *rectangle* memiliki nilai akurasi tertinggi yaitu 95,2% dibandingkan dengan penggunaan bentuk *structure element* yang lain. Pada hasil dari pengujian *sobel edge detection* memiliki nilai akurasi tertinggi yaitu 92,8%.

Kata kunci: *zebra cross*, *pre-processing*, *hough transform*.

Abstract

The high number of accidents that injure pedestrians while crossing is caused by motorists who are less cautious. Accidents of course undesirable can be prevented and minimized the culture of orderly traffic one by using facilities such as zebra cross. In this research, we propose the process of zebra cross detection on digital image using Hough Transform method, in order to be implemented in smart vehicle navigation system in identifying zebra cross in order to increase equality of both riders and zebra cross users. The zebra cross detection process starts from pre-processing, which consists of grayscale process, mean filtering, dilation, and histogram equalization, for our edge detection using the next stage canny method is the image inversion which aims to change the pixels of white to black, and vice versa. Then for line detection on zebra cross using hough transform method. Based on the test, the highest accuracy value when the 100 threshold value on the first morning condition test data is 95.2%. The result of testing the variation of the structure element obtained the maximum results with the use of rectangle has the highest accuracy value of 95.2% compared with the use of other structure element form. In the result of testing edge detection sobel has the highest accuracy value of 92.8%.

Keywords: *zebra cross*, *pre-processing*, *hough transform*

1. PENDAHULUAN

Tingginya angka kecelakaan yang mencederai pejalan kaki saat menyeberang disebabkan oleh kendaraan bermotor yang kurang berhati-hati. Kecelakaan yang tentunya tidak diinginkan dapat dicegah dan diminimalisir

budaya tertib berlalu-lintas dengan menggunakan fasilitas seperti *zebra cross*. Namun, pada kenyataannya meski jumlah *zebra cross* cukup banyak dan memiliki ukuran yang dapat dilihat seperti tidak berfungsi sebagaimana mestinya. Fenomena ini terjadi karena peningkatan jumlah kendaraan bermotor tidak

diiringi dengan peningkatan kesadaran dari masyarakat.

Manajemen lalu lintas semakin didukung dengan adanya teknologi informasi. Perkembangan teknologi memperkenalkan pendeteksian *zebra cross* pada sistem transportasi cerdas. *Zebra cross detection* adalah salah satu bagian penting dari kecerdasan berbasis visual navigasi kendaraan. Pengenalan *zebra cross* bisa dilakukan dengan menerapkan metode *Hough Transform*.

Pada penelitian sebelumnya yang berkaitan dengan deteksi *zebra cross* secara otomatis dilakukan oleh Riveiro (2015) menjelaskan kinerja *Hough Transform* dapat mendeteksi 30 penyeberangan zebra. Tes itu menunjukkan akurasi sebesar 83%. Penelitian selanjutnya tentang pengenalan *zebra cross* yang dilakukan Sun (2016) menggunakan metode *Hough Transform* untuk ekstraksi dan rekonstruksi *zebra cross* dengan akurasi tertinggi sebesar 98,4%..

Berdasarkan uraian di atas, peneliti mengusulkan penelitian dengan menggunakan *Hough Transform* untuk deteksi *zebra cross* pada citra digital.

2. TINJAUAN KEPUSTAKAAN

2.1 Pre-Processing

2.1.1 Konversi Grayscale

Proses awal yang banyak dilakukan dalam *image processing* adalah mengubah citra berwarna menjadi citra *grayscale*, hal ini bertujuan untuk menyederhanakan model citra (Hendry, 2011). Dalam praktik, citra berwarna seringkali harus dikonversi ke dalam bentuk citra berskala keabuan mengingat banyak pemrosesan citra yang bekerja pada skala keabuan Kadir dan Susanto (2013). Tahap *grayscale* dilakukan dengan cara mengkalikan piksel-piksel pada setiap komponen RGB dengan masing-masing konstanta *grayscale*. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing *r*, *g* dan *b* menjadi citra keabuan dengan nilai *grayscale*, maka konversi dapat dilakukan sesuai dengan Persamaan 1 Kanan & Cottrell (2012) dan Kadir & Susanto (2013).

$$Grayscale = (0,2989 \times r) + (0,5870 \times g) + (0,1140 \times b) \tag{1}$$

Dimana *grayscale* merupakan nilai citra keabuan, *r* merupakan nilai piksel komponen

merah, *g* merupakan nilai piksel komponen hijau, dan *b* merupakan nilai piksel komponen biru.

2.1.2 Mean Filter

Mean filter adalah salah satu teknik *filtering* citra *non linear* yang berfungsi untuk memperhalus suatu citra dan menghilangkan *noise* atau gangguan berupa bintik putih. Proses *mean filtering* secara umum dapat dilihat pada Persamaan 2.4 (Yuwono, 2010). Misalkan S_{xy} menyatakan himpunan koordinat dalam sebuah *window subimage* berukuran $m \times n$ yang berpusat di titik (x, y) . Proses *arithmetic mean filtering* menghitung nilai rata-rata dari citra ber-*noise* $g(x, y)$ dalam area yang didefinisikan oleh S_{xy} . Atau dapat dinyatakan dengan Persamaan 2.

$$\bar{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(x, y) \tag{2}$$

Dimana \bar{f} merupakan *mean* atau nilai rata-rata, mn merupakan jumlah data, $g(x, y)$ merupakan nilai piksel citra inputan.

2.1.3 Dilasi

Dilasi adalah transformasi yang menghasilkan gambar yang bentuknya sama seperti aslinya, namun begitu ukuran yang berbeda (Sreedhar & Panlal, 2012). Bila suatu objek (citra input) dinyatakan dengan A dan *struture element* dinyatakan dengan B serta B_x menyatakan translasi B sedemikian sehingga pusat B terletak pada x. operasi dilasi A dengan B dapat dinyatakan sebagai berikut:

$$D(A, B) = A \oplus B = \{x: +B_x \cap A \neq \emptyset\} \tag{3}$$

Dengan \emptyset menyatakan himpunan kosong.

Proses dilasi dilakukan dengan membandingkan setiap piksel citra masukan dengan nilai pusat *struture element* dengan cara melapisi (*superimpose*) *struture element* dengan citra sehingga pusat *struture element* tepat dengan posisi piksel citra yang diproses. Jika paling sedikit ada 1 piksel pada *struture element* sama dengan nilai piksel objek (*foreground*) citra maka piksel input diset nilainya dengan nilai piksel *foreground* dan bila semua piksel yang berhubungan adalah *background* maka input piksel diberi nilai piksel *background*. Proses serupa dilanjutkan dengan menggerakkan (translasi) *struture element* piksel demi piksel pada citra input.

2.1.4 Histogram Equalization

Tujuan dari teknik ini adalah untuk menghasilkan histogram citra yang seragam. Teknik ini dapat dilakukan pada keseluruhan citra atau pada beberapa bagian citra saja Putra (2010).

Histogram hasil proses ekualisasi tidak akan seragam atau sama untuk seluruh intensitas. Teknik ini hanya melakukan distribusi ulang terhadap distribusi intensitas dari histogram awal. Jika histogram awal memiliki beberapa puncak dan lembah maka histogram hasil ekualisasi akan tetap memiliki puncak dan lembah. Akan tetapi puncak dan lembah tersebut akan digeser. Histogram hasil ekualisasi akan lebih disebar (spreading) Putra (2010).

Pada *histogram equalization* terdapat beberapa langkah yaitu Wang (2016):

1. Mencari derajat keabuan i kemudian menjumlahkan frekuensi $n(i)$ masing-masing derajat keabuan ke- i yang terdapat pada matriks citra.
2. Menghitung nilai kepadatan probabilitas pada masing-masing tingkat abu-abu atau PDF (*Probability Density Function*). Seperti yang ditunjukkan pada Persamaan 4.

$$h_x(i) = \frac{n_i}{N} \quad (4)$$

Dimana $h_x(i)$ merupakan nilai kepadatan probabilitas skala keabuan ke- i dari citra, n_i merupakan derajat keabuan ke- i , N merupakan jumlah piksel.

3. Menghitung nilai CDF (*Cumulative Density Function*). Dengan cara menjumlahkan tiap nilai PDF dari tiap *range* menggunakan Persamaan 5

$$y' = H_x(i) = \sum_{i=0}^j h_x(i) \quad (5)$$

Dimana $H_x(i)$ merupakan nilai CDF, $h_x(i)$ merupakan nilai kepadatan probabilitas skala keabuan ke- i dari citra, j merupakan maksimal derajat keabuan, i merupakan minimal derajat keabuan.

4. Hasil dari CDF pada kemudian dikalikan dengan nilai derajat keabuan yang terbesar dan dibulatkan.

2.2 Deteksi Tepi Canny

Metode Canny merupakan metode deteksi tepi yang digunakan pada penelitian ini. Metode ini pertama kali dikembangkan oleh John F. Canny menggunakan algoritme multi-tahap untuk mendeteksi berbagai *edge* dalam citra.

Implementasi dari *Canny Edge Detection*, memiliki beberapa tahap yaitu:

1. Langkah pertama adalah mengilangkan *noise* yang ada pada citra dengan mengimplementasikan *Gaussian Filter Smoothing* dengan *Gaussian filter*, tujuannya adalah untuk mengurangi respon terhadap *noise* Hermawati (2013).

$K = 1/159$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Gambar 1. Kernel Gaussian Filter 5 x 5

2. Langkah kedua adalah melakukan deteksi tepi dengan salah satu operator deteksi seperti *Robert*, *Prewit* atau *Sobel* dengan melakukan pencarian secara horizontal (G_x) dan secara vertical (G_y). Pada penelitian ini menggunakan operator deteksi *Sobel*.

- a. Lakukan proses perkalian matriks citra hasil Gaussian Filtering antara kernel G_x dan G_y .

-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

(a) (b)

Gambar 2. (a) Kernel Sobel G_x (b) Kernel Sobel G_y

- b. Menentukan ukuran gradien dan arah gradien pada tiap piksel. Dengan Persamaan 6 seperti di bawah ini:

$$G = \sqrt{G_x^2 + G_y^2} \quad (6)$$

3. Langkah selanjutnya adalah memperkecil garis tepi yang muncul dengan menerapkan *non maximum suppression* sehingga menghasilkan garis tepi yang lebih ramping.
4. Tahap keempat dari proses *Canny Edge Detection* adalah melakukan *hyteresis thresholding* dengan 3 kondisi sebagai berikut:

- a. Jika gradien suatu piksel lebih besar daripada batas atas, maka piksel tersebut dianggap sebagai *'edge pixel'*.
- b. Jika gradien suatu piksel lebih kecil daripada batas bawah nilai ambang, maka piksel tersebut bukan *'edge pixel'*.
- c. Jika gradien suatu piksel diantara nilai

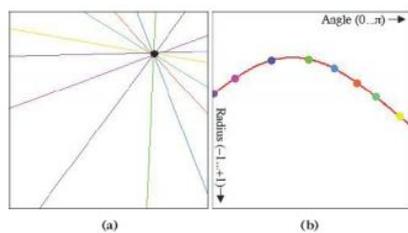
batas bawah dan atas maka piksel terhubung dengan sebuah 'edge pixel' jika dan hanya jika piksel tersebut terhubung dengan sebuah 'edge pixel' secara langsung atau melalui piksel-piksel antara batas bawah dan atas.

2.3 Inversi citra

Proses selanjutnya yang dilakukan yaitu inversi citra hasil *Canny* yang telah didapat. Konversi citra dilakukan pada citra hasil *Canny* karena deteksi tepi yang dihasilkan berwarna putih dan background berwarna hitam, sehingga diperlukan pembalikan nilai agar tepi yang dideteksi berwarna hitam.

2.4 Hough Transform

Setelah citra hasil inversi didapat, maka proses selanjutnya yaitu deteksi garis atau *zebra cross* dengan menggunakan metode *Hough Transform*. Tujuan penerapan metode ini yaitu untuk mendeteksi setiap pikselnya termasuk piksel penyusun *zebra cross* atau bukan *zebra cross*. Beberapa prinsip yang dilakukan pada *Hough Transform* adalah: (a) untuk setiap titik pada gambar dalam bidang-xy akan membentuk sebuah kurva sinusoidal di dalam ruang Hough (r, θ). (b) setiap garis yang melalui suatu titik yang sama dalam bidang-xy merupakan titik-titik penyusun sebuah kurva sinusoidal di dalam ruang Hough. Titik-titik perpotongan di dalam ruang Hough mengidentifikasi garis-garis di dalam bidang-xy seperti diperlihatkan dalam Gambar 5 (Russ, 2011).



Gambar 3. Prinsip Transformasi Hough

Langkah-langkah dalam membentuk transformasi Hough adalah Hendry (2011):

1. Langkah pertama yaitu melakukan *pre-processing*.
2. Untuk mempercepat komputasi maka perlu dilakukan deteksi tepi.
3. Misalkan ukuran dari citra digital adalah x dan y maka nilai d didapatkan dari Persamaan 2.7:

$$d = \sqrt{x^2 + y^2} \tag{7}$$

Nilai d tersebut akan menjadi ukuran maksimal (ρ_{max}) dari nilai ρ .

4. Selanjutnya menentukan nilai ρ_{min} , ρ_{max} , θ_{min} dan θ_{max} .
5. Langkah keenam membuat sebuah matriks $A(\rho, \theta)$ dengan nilai ρ berada di antara ρ_{min} dan ρ_{max} sedangkan nilai θ berada di antara θ_{min} dan θ_{max} . Perlu diingat bahwa nilai ρ_{max} sudah ditentukan di atas.
6. Berikan nilai nol untuk $A(\rho, \theta)$ pada tiap elemennya.
7. Lalu kumpulkan koordinat dari citra yang telah dideteksi tepi yang memiliki nilai selain 0 sehingga komputasi akan semakin sedikit dan bisa mempercepat proses.
8. Untuk mendapatkan nilai ρ dari θ_{min} hingga θ_{max} menggunakan Persamaan 8.

$$\rho = x \cdot \cos \theta + y \cdot \sin \theta \tag{8}$$

Perlu diingat bahwa nilai x dan y merupakan koordinat hasil deteksi tepi, bukan lagi koordinat sebenarnya dari citra digital.

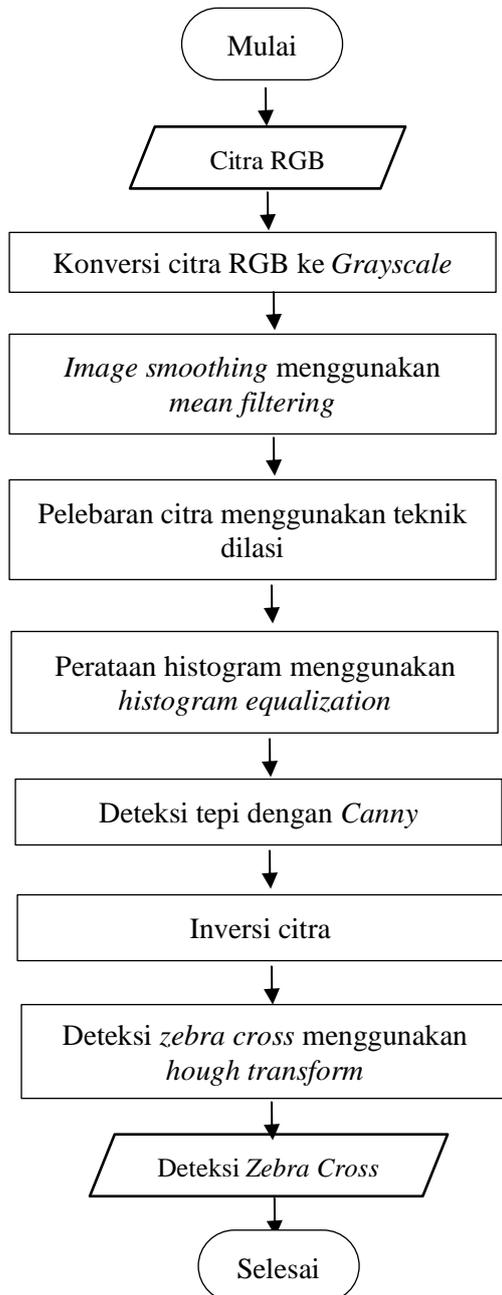
9. Bulatkan nilai ρ tersebut ke nilai terdekat.
10. Hitunglah inkremen dari $A(\rho, \theta)$. Nilai $A(\rho, \theta)$ yang terbesar menandakan banyaknya titik-titik yang membentuk garis. Sehingga nilai θ pada A tersebut merupakan nilai sudut yang kita inginkan. Dengan menggunakan nilai θ tersebut, kita bisa melakukan rotasi citra ke arah yang sebaliknya. Inkremen yang dimaksud disini merupakan aktifitas *vote* seperti yang telah dijelaskan di atas.

3. METODOLOGI

Penyelesaian masalah deteksi pada *zebra cross* dalam hal ini menggunakan metode *Hough Transform*. Tahap awal yang dilakukan yaitu *pre-processing* yang meliputi proses konversi dari citra RGB ke *grayscale*. Setelah tahap pertama dilanjutkan proses *mean filtering*. Output dari proses *mean filtering* akan dijadikan input pada proses dilasi dan proses selanjutnya akan dilakukan *histogram equalization*. Setelah proses *pre-processing* selesai dilanjutkan proses deteksi tepi dengan menggunakan metode *Canny*, output dari deteksi tepi ini akan dilakukan inversi pada citra hasil *Canny* yang fungsinya untuk merubah piksel hitam menjadi

putih dan putih menjadi hitam pada citra. Pada proses terakhir dilakukan proses deteksi garis pada *zebra cross* dengan *Hough Transform*.

Diagram alir sistem secara umum ditunjukkan pada Gambar 4.



Gambar 4. Algoritme untuk deteksi *zebra cross*

3.1. Dataset

Data yang digunakan untuk mengevaluasi algoritme pada penelitian ini didapat dari observasi yang dilakukan peneliti. Data yang didapat berupa video beberapa ruas jalan kota Malang. Data yang digunakan adalah citra

berwarna. Data pada program sebanyak 4 video uji. Data dibagi dalam dua kondisi, masing-masing terdapat kondisi pagi dan sore hari. Terdapat 2 video dengan kondisi pagi dan 2 video dalam kondisi sore hari. Pada masing-masing video kondisi pagi terdiri dari 300 frame dan 240 frame. Pada video pertama dikondisi sore terdapat 180 dan video kedua terdapat 493 frame. Citra ini diambil menggunakan kamera iPhone 5s dengan sudut 90 derajat. Setiap citra yang akan diproses disimpan dengan ukuran 300x169 piksel.

4. HASIL DAN PEMBAHASAN

Pada pengujian ini terdapat 3 macam pengujian yang dilakukan. Pengujian pertama dilakukan dengan tujuan untuk mengetahui pengaruh proses penentuan nilai *threshold*. Kemudian untuk pengujian kedua dilakukan untuk mengetahui pengaruh proses perbaikan citra pada dilasi dengan penggunaan bentuk *structure element*. Dan terakhir adalah pengujian untuk mengetahui pengaruh dari dilakukannya proses deteksi tepi menggunakan *Sobel* dan tanpa menggunakan deteksi tepi *Sobel*. Tidak semua frame pada video yang dipakai dalam pengujian, tetapi hanya beberapa frame yang terdapat *zebra cross*. Jumlah frame yang akan diuji secara detail dapat dilihat pada Tabel 1.

Tabel 1. Rancangan Analisis Komputasi

Kondisi	Jumlah Frame
Pagi 1	63
Pagi 2	48
Sore 1	26
Sore 2	98

4.1 Pengujian Variasi *Threshold*

Nilai *threshold* yang digunakan dalam pengujian ini adalah $t = 50$, $t = 100$, $t = 200$. Pada pengujian ini menggunakan jumlah data uji yang terdapat *zebra cross* pada setiap *frame*, data uji kondisi pagi 1 sejumlah 63, data uji kondisi pagi 2 sejumlah 48, data uji kondisi sore 1 sejumlah 26, dan data uji kondisi sore 2 sejumlah 98. Hasil terbaik dari pengujian nilai *threshold* ini akan digunakan untuk melakukan pengujian variasi *structure element*.

Tabel 2. Pengujian Variasi Nilai *Threshold*

Data Uji : jumlah frame	50	100	200
Pagi 1	85,7%	95,2%	69,84%

Pagi 2	72,9%	84,09%	68,75%
Sore 1	84,26%	69,23%	46,15%
Sore 2	92,8%	74,26%	50,49%

Berdasarkan hasil pada pengujian variasi nilai *threshold*, diperoleh hasil dari nilai *threshold* 50 dengan nilai akurasi pada data uji kondisi pagi 1, pagi 2, sore 1, sore 2 sebesar 85,7%, 72,9%, 84,26%, dan 92,8%. Hasil akurasi tertinggi terdapat pada data kondisi sore kedua dan akurasi terendah dimiliki oleh kondisi pagi yang kedua.

Pada pengujian variasi nilai *threshold* 100 dengan nilai akurasi pada data uji kondisi pagi 1, pagi 2, sore 1, sore 2 sebesar 95,2%, 84,09%, 69,23%, dan 74,26%. Hasil akurasi tertinggi terdapat pada data kondisi pagi pertama dan akurasi terendah dimiliki oleh kondisi sore yang pertama. Hasil tersebut dapat terjadi karena intensitas cahaya pada data uji pagi yang pertama cenderung rendah, sehingga ketika diberikan nilai ambang 100 informasi yang diperlukan untuk deteksi tetap ada, sedangkan *noise* atau informasi yang tidak diperlukan hilang. Sedangkan untuk hasil terendah dapat terjadi karena data pada kondisi sore kedua intensitas cahaya lebih tinggi. Akibatnya adalah informasi posisi zebra lebih banyak yang hilang. Berdasarkan hasil pada pengujian variasi nilai *threshold* 200 dengan nilai akurasi pada data uji kondisi pagi 1, pagi 2, sore 1, sore 2 sebesar 69,84%, 68,75%, 46,15%, dan 50,49%. Hasil akurasi tertinggi terdapat pada data kondisi pagi pertama dan akurasi terendah dimiliki oleh kondisi sore yang pertama. Hasil pengujian tersebut dikatakan rendah karena nilai *threshold* 200 terlalu tinggi untuk semua frame yang diuji. Akibatnya banyak informasi citra yang hilang termasuk posisi *zebra cross*.

Salah satu faktor yang menyebabkan kegagalan dalam proses deteksi zebra cross adalah proses pengembangan yang gagal, artinya proses pengembangan menyebabkan titik-titik penyusun garis *zebra cross* tidak terbaca, sehingga saat melakukan variasi *threshold* tidak dapat menemukan adanya zebra cross dalam citra, akibatnya tidak ada objek yang terdeteksi atau objek lain terdeteksi sebagai *zebra cross*. Faktor-faktor yang menyebabkan proses pendeteksian tidak tepat antara lain:

1. Kondisi pengambilan citra, seperti pencahayaan (penyebaran intensitas) yang tidak merata, yang disebabkan karena pantulan cahaya yang tajam pada bagian tertentu citra atau citra

yang kabur.

2. Kondisi dari *zebra cross*.

4.2 Pengujian Variasi *Structure Element*

Pengujian ini menjelaskan tentang pengujian untuk mengetahui pengaruh proses perbaikan citra pada dilasi dengan penggunaan bentuk *structure element*. Data latih dan data uji sama-sama dilakukan proses *pre-processing* namun dilakukan variasi bentuk *structure element* pada tahap dilasi. Pada pengujian ini dilakukan variasi 3 bentuk *structure element* yang berbeda. Yaitu bentuk 1 dilakukan pengujian dengan bentuk *structure element* kotak (*Rectangle*), dan selanjutnya pada bentuk *structure element* dilakukan pengujian dengan bentuk *structure element* bulat. Terakhir bentuk *structure element* yang diuji adalah *cross*. Hasil pengujian pengaruh penggunaan bentuk *structure element* pada tahap dilasi dapat dilihat pada Tabel 3.

Tabel 3. Pengujian Variasi *Structure Element*

Data Uji : jumlah frame	Rect	Ellipse	Cross
Pagi 1:63	95,2%	93,6%	92,06%

Berdasarkan pengujian untuk mengetahui pengaruh bentuk *structure element* (*structure element*) pada proses perbaikan citra tahap dilasi diperoleh hasil dari penggunaan kotak sebesar 95,2%, dan hasil pada *ellipse* sebesar 93,6%, sedangkan hasil pada *cross* sebesar 92,06%. Dari perbandingan nilai tersebut dapat dikatakan bahwa penggunaan *structure element* bentuk kotak mendapatkan hasil yang jauh lebih baik dibandingkan dengan *structure element* lainnya. Untuk penggunaan *structure element ellipse* dan *cross* lebih rendah disebabkan oleh lebih banyaknya kemunculan derau atau *noise*, akibatnya adalah tidak dapat mendeteksi posisi *zebra cross* dengan tepat.

4.3 Pengujian Menggunakan *Sobel Edge Detection* dan tanpa *Sobel Edge Detection*

Tahap ini menjelaskan tentang pengujian untuk mengetahui pengaruh penggunaan *Sobel Edge Detection* dan tanpa menggunakan *Sobel Edge Detection*. Data latih dan data uji sama-sama dilakukan proses *pre-processing*, deteksi tepi *Canny* dan inversi gambar namun dilakukan variasi *Sobel Edge Detection* dan tanpa *Sobel Edge Detection* pada tahap setelah deteksi tepi *Canny*. Pengujian ketiga dilakukan dengan tujuan untuk mengetahui pengaruh penggunaan bentuk deteksi tepi Sobel terhadap hasil deteksi *zebra*

cross. Hasil pengujian pengaruh penggunaan *Sobel Edge Detection* dan tanpa *Sobel Edge Detection* pada tahap deteksi tepi dapat dilihat pada tabel 4.

Tabel 4. Pengujian Menggunakan *Sobel Edge Detection* dan tanpa *Sobel Edge Detection*

Data Uji : jumlah frame	<i>Sobel Edge Detection</i>	Tanpa <i>Sobel Edge Detection</i>
Pagi 1:63	95,8%	95,2%

Pengujian dengan menggunakan *Sobel Edge Detection* memiliki hasil deteksi tepi pada *zebra cross* lebih jelas dan lebih tegas. Hal ini disebabkan metode deteksi tepi *sobel* lebih sensitif terhadap tepi diagonal dari pada tepi horizontal dan vertikal. Untuk deteksi tepi horizontal menggunakan kernel sumbu y dan untuk deteksi tepi vertikal menggunakan kernel sumbu x.

5. KESIMPULAN

Dari hasil penelitian dan pembahasan pada bab sebelumnya dapat disimpulkan bahwa penggunaan operasi dilasi sangat bergantung pada *structuring element* yang digunakan. Penggunaan bentuk *struture element* memungkinkan sistem mengenali posisi *zebra cross*, tetapi kegagalan yang ditimbulkan juga semakin tinggi. Jadi pemilihan *struture element* harus disesuaikan dengan domain aplikasinya dan harus meminimalisir kegagalan sistem. Pada tahap pengujian data uji menghasilkan persentase pendeteksian 95,2% untuk keseluruhan data uji pada kondisi pagi yang pertama. Terdapat beberapa faktor yang mempengaruhi hasil pendeteksian posisi *zebra cross* yaitu cara pengambilan citra, tingkat pencahayaan yang tidak merata (penyebaran intensitas), rusak atau tidaknya *zebra cross*, hal ini dapat dapat diatasi salah satunya dengan penentuan *threshold*. *Threshold* yang dipakai sangat berpengaruh pada deteksi tepi untuk kemudian dijadikan masukkan pada deteksi garis *zebra cross*. Saran yang dapat diberikan untuk penelitian lebih lanjut adalah untuk meminimalisir pengaruh tingkat kecerahan pada citra *zebra cross*, sebaiknya diperlukan algoritma untuk menentukan kecerahan secara otomatis (adaptif) seperti algoritma Contrast-limited adaptive histogram equalization (CLAHE) agar pada proses perubahan ke citra biner tidak mengalami kendala. Sebaiknya mengimplementasikan metode *filling* pada

proses ekstraksi fitur untuk meningkatkan akurasi pengenalan *zebra cross*, dan untuk menyempurnakan bentuk *zebra cross* yang terputus-putus karena kondisi pencahayaan atau konstruksi *zebra cross* yang tidak sempurna.

6. DAFTAR PUSTAKA

Fadliyah, 2007. *Computer Vision*. Yogyakarta: ANDI.

Hendry, J., 2011. *Ocr Document De-Skew Using Hough Transform*.

Indriastuti, A. K., Fauziah, Y. & Priyanto, E., 2011. *Karakteristik Kecelakaan Dan Audit Keselamatan Jalan Pada Ruas Ahmad Yani Surabaya*. Volume 5.

Kadir, A. & Susanto, A., 2013. *Teori dan Aplikasi Pengolahan Citra*. 1 penyunt. Yogyakarta: ANDI.

Kanan, C. & Cottrell, G. W., 2012. *Color-to-Grayscale: Does the Method Matter in Image*.

Putra, D., 2010. *Pengolahan Citra Digital*. Mengwitani: ANDI.

Reveiro, B. et al., 2015. *Automatic Detection of Zebra Crossing from Mobile LIDAR Data*. *Science Direct*, 70(Optics & Laser Technolog), pp. 63-70.

Sreedhar, K. & Panlal, B., 2012. *Enhancement Of Images Using Morphological Transformations*. *International Journal of Compter Science & Information Technology (IJCSIT)*, Volume 4, pp. 33-50.

Stephen, S., 2000. *Zebra-crossing Detection for the Partially Sighted*. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, Volume vol.2, pp. 211 - 217.

Sun, Y., Zhang, F., Gao, Y. & Huang, X., 2016. *Extraction and Reconstruction of Zebra Crossings from High Resolution Aerial Images*. *International Journal of Geo-Information*, Volume 5, pp. 1-20.

Sun, Y., Zhang, F., Gao, Y. & Huang, X., 2016. *Extraction and Raconstruction of Zebra Crossings from High Resolution Aerial Image*.

Wang, M. et al., 2016. *Zebra-Cross Automatic Recognition and Early Warning for Intelligent Driving*. Volume 9 .