

# Deteksi Plat Nomor Menggunakan *Connected Components* dan Pra-Pengolahan untuk Mempercepat Proses Deteksi

**Arlia Anggraeni<sup>1</sup>, Adhika Pramita Widyassari<sup>2</sup>**

<sup>2</sup>Teknik Elektro, Sekolah Tinggi Teknologi Ronggolawe Cepu E-mail:

<sup>1</sup>libtihajilhama@gmail.com, <sup>2</sup>dikasari9@gmail.com

**Abstrak** — Seiring meningkatnya jumlah kendaraan di Indonesia, semakin meningkat pula kejahatan yang terjadi, mengakibatkan peningkatan masalah pelanggaran lalu lintas dan meningkatnya jumlah aktivitas parkir kendaraan. Melihat pentingnya deteksi plat nomor guna mengatasi hal tersebut maka diperlukan suatu sistem yang mampu mendeteksi plat nomor dengan cepat dan tepat. Proses segmentasi menggunakan *Connected Components* memiliki kinerja yang lebih bagus bila dibandingkan dengan *Diff* dan metode *Fast-RNN*. Pra-pengolahan adalah langkah penting sebelum melakukan deteksi plat nomor. diawali dengan mengubah ukuran citra dengan menggunakan fungsi *resizing* dan dengan mengubah citra RGB ke dalam citra grayscale yang bertujuan memperkecil ukuran citra, sehingga mempercepat proses selanjutnya. Kemudian proses segmentasi menggunakan metode *connected components*. Hasil uji coba pada 10 citra uji menghasilkan akurasi 80 %. Dari segi waktu deteksi menunjukkan penurunan. Proses deteksi plat nomor tanpa menggunakan pra-pengolahan membutuhkan waktu rata-rata 3,721519 detik, sedangkan bila memanfaatkan pra-pengolahan hanya 1,45731 detik. Dapat disimpulkan bahwa penambahan pra-pengolahan mampu mempercepat proses deteksi.

**Kata Kunci** - *connected components, deteksi, morfologi, pra-pengolahan.*

**Abstract** — As the number of vehicles in Indonesia increases, the number of crimes that occur also increases, resulting in an increase in the problem of traffic violations and an increase in the number of vehicle parking activities. Seeing the importance of number plate detection to overcome this, a system that is able to detect number plates quickly and accurately is needed. The segmentation process using *Connected Components* has better performance when compared to *Diff* and *Fast-RNN* methods. Pre-processing is an important step before carrying out a number plate detection. begins by changing the image size using the *resizing* function and by converting the **RGB** image into a grayscale image which aims to reduce the image size, thereby speeding up the next process. Then the segmentation process uses the *connected components* method. The test results on 10 test images produce an accuracy of 80%. In terms of detection time, it shows a decrease. The number plate detection process without using pre-processing takes an average of 3.721519 seconds, whereas when using pre-processing only 1.45731 seconds. It can be concluded that the addition of pre-processing is able to speed up the detection process.

**Keywords** — *connected components, detection, morphology, pre-processing*

## 1. PENDAHULUAN

Seiring meningkatnya jumlah kendaraan di Indonesia, semakin meningkat pula kejahatan yang terjadi. Salah satunya adalah pencurian kendaraan bermotor (Achmad Solichin dan Zulfikar Rahman, 2017). Meningkatnya populasi kendaraan, juga mengakibatkan peningkatan masalah yang terjadi pada pelanggaran lalu lintas (Erwin Dwika Putra, 2017). Puspaningrum dkk (Puspaningrum

et al,2018) juga menyatakan dengan semakin banyaknya kendaraan juga meningkatkan jumlah aktivitas parkir. Pencatatan nomor kendaraan merupakan salah satu aktivitas parkir, dengan banyaknya jumlah kendaraan yang ada membuat perekaman dan pencatatan nomor kendaraan mengalami kesulitan dalam kegiatan pencatatan dan perekaman nomor kendaraan. Deteksi plat nomor merupakan solusi dalam mengatasi masalah di atas. Melihat pentingnya deteksi plat nomor, maka diperlukan suatu sistem yang mampu mendeteksi plat nomor dengan cepat dan tepat.

Proses deteksi plat nomor dapat dilakukan dengan menggunakan pengolahan citra *digital*. Proses perekaman plat nomor diawali dengan mencari atau mendeteksi posisi plat nomor terdahulu, tepatnya pendeteksian posisi plat nomor mempengaruhi proses segmentasi karakter pada plat nomor. Deteksi menurut KBBI adalah usaha menemukan dan menentukan keberadaan, anggapan, atau kenyataan. Mazen dkk (Mazen et al, 2016) menyatakan fungsi pendeteksian di antaranya adalah menghitung, mengindeks, melabeli, menentukan ukuran objek, *grouping* objek dan sebagainya. Deteksi plat nomor merupakan kegiatan yang memisahkan area plat nomor dengan area yang bukan plat nomor, kegiatan ini disebut dengan segmentasi. dengan masalah yg diselesaikan, ulasan penelitian yang pernah dilakukan sebelumnya oleh peneliti lain yang relevan dengan penelitian yang dilakukan.

Penelitian yang membahas deteksi posisi plat nomor telah banyak dilakukan, antara lain penelitian yang dilakukan oleh Eva Yulia Puspaningrum, dkk (Puspaningrum et al, 2018) yang menggunakan metode *Wavelets Transforms* menghasilkan 11 gambar mampu dideteksi dengan benar dari 20 citra data uji. Peneliti menyatakan bahwa metode *Wavelets Transforms* kurang cocok digunakan dalam melakukan deteksi plat nomor, hal ini dipengaruhi dalam pengambilan gambar yang harus tegak lurus dengan garis horizontal dan vertikal.

Penelitian yang dilakukan Daniel Onoro, dkk (Daniel onoro, 2013) menyatakan bahwa proses segmentasi dengan menggunakan *Connected Components* memiliki kinerja yang lebih bagus bila dibandingkan dengan Diff dan metode Fast-RNN. Metode *Connected Components* dapat mengklasifikasikan daerah dan/atau *region* suatu objek pada citra digital. Piksel-piksel dalam *region* disebut *connected* karena adanya keterhubungan satu sama lain, yakni mematuhi aturan *adjacency* atau kedekatan suatu piksel. (Gonzales et al, 2004).

Proses deteksi plat nomor selain dipengaruhi proses segmentasi, juga dipengaruhi oleh proses pra-pengolahan. Soon.C, dkk (Soon.C et al, 2012) menyatakan bahwa pra-pengolahan adalah langkah penting sebelum melakukan deteksi plat nomor. Metode pra-pengolahan di antaranya adalah *histogram equalization*, binerisasi, *noise reduction*, dan *resizing*.

*Resizing* adalah proses pengubahan piksel citra menjadi lebih besar dan/atau kecil dari ukuran asli citra. Semakin sedikit ukuran data maka data akan kian cepat untuk diproses (Wakhidah, 2012). Hal tersebut sesuai dengan pernyataan bahwa pemrograman yang baik adalah program yang mampu mengurangi waktu proses (M.Bister et al, 2007).

*Grayscale* adalah tahapan mengubah citra RGB (*Red Green Blue*) ke dalam skala abu. Metode ini sering digunakan karena dapat menyederhanakan algoritma dan mengurangi waktu komputasi. Karena terkadang banyak informasi yang tidak dibutuhkan dari citra berwarna. Selain itu banyaknya jumlah warna juga berpengaruh terhadap peningkatan data latih (Kanan C dan Cottrell GW, 2012). *Grayscale* juga merupakan proses sebelum dilakukan deteksi tepi.

## 1. METODE PENELITIAN

### 2.1. Metode Pra-Pengolahan

Pra —pengolahan yang digunakan adalah *Resizing* dan *Grayscale*. Pra-pengolahan yang digunakan adalah interpolasi bilinear sebesar 0.6 (im2-imresize,0.6, 'bilinear') Fungsi interpolasi bilinear adalah sebagai berikut (M.Bister et al, 2007)

$$L(x,y) = \frac{([x] - x)([y] - y)([y] - y)L([x] , [y]) + ([x] - x)(y - [y]) + L([x], [y]) + (x -$$

Dengan

$$|x| (|y| - y) L (|x|, |y|) + (x - |x|)(y - |y|) L (|x|, |y|)$$

Keterangan : s=skala *resizing*

i dan j : integer

$$(x,y) \left( \frac{i}{s}, \frac{j}{s} \right) \dots\dots\dots (1)$$

Sementara *Grayscale* menggunakan fungsi *rgb2gray* dengan persamaan:

$$0.2989 * R + 0.5870 * G + 0.1140 * B \dots\dots\dots(2)$$

Pembobotan tersebut didasarkan pada BT.601, yaitu sebuah parameter yang dapat diakses secara daring mengenai televisi *digital* pada tahun 2011 serta mencakup standar warna untuk proses *encode* citra. Sedangkan R, G, dan B secara berturut-turut menyatakan nilai komponen ruang warna R (merah), G (hijau), dan B (biru) dari nilai setiap piksel citra berwarna pada posisi (x,y) (Teguh, et al, 2014).

2.2. Tahap Pengolahan Citra

2.2.1. Deteksi Tepi

Deteksi Tepi sangat penting untuk mengetahui garis batas objek dalam suatu citra. Juga dengan adanya proses deteksi tepi akan menjadi langkah awal pendeteksian. Deteksi Tepi yang digunakan peneliti menggunakan Operator Sobel. Operator Sobel memiliki kelebihan untuk mengurangi noise atau derau sebelum melakukan proses deteksi tepi. Deteksi Tepi dengan Operator Sobel dapat dilakukan dengan menuliskan fungsi `ime=edge(im2,'sobel')` pada Matlab. Operator Sobel mempunyai garis vertikal dan horizontal yang dapat dihitung menggunakan matriks. (Muthukrishnan.R dan Radha.M, 2011).

-1	-2	-1
0	0	0
+1	*2	+1

Gx

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

Gy

Keterangan :

Gx : Tepi Vertikal Gy :Tepi Horizontal

2.2.2. Morfologi Citra

Morfologi Citra adalah operasi yang umum dikenakan pada citra biner untuk mengubah struktur objek yang terkandung dalam citra. (Santony.J et al, 2015). Pada citra biner hanya mengenal angka 1 dan 0. Angka 1 untuk objek dan angka 0 untuk *background*. Operasi yang paling umum digunakan pada Morfologi adalah erosi, dilasi dan *filling holes*. Rumus erosi, dilasi dan *filling holes* (Gonzales, 2004) adalah sebagai berikut:

$$A \ominus B = (z | (B \cap D) \cap A) \dots\dots\dots(3)$$

Operasi Dilasi adalah kebalikan dari Erosi, proses ini bertujuan untuk memperbesar segmen objek dalam citra. Persamaan untuk dilasi adalah:

$$A \oplus B = [z|(B) \cap A^c \cup O] \dots\dots\dots(4)$$

Tujuan dari operasi *Filling Holes* adalah mengisi keseluruhan *region* dengan nilai 1. Operasi ini menggunakan acuan berdasarkan nilai piksel tetangganya Fungsi matematis *Filling Holes* adalah:

$$f_m(x, y) = \begin{cases} 1 - f(x, y) & \dots\dots\dots(5) \\ 0 \end{cases}$$

2.2.3. *Connected Components*

*Connected Components* adalah salah satu teknik yang mudah untuk dipahami dan diterapkan. (Akmal.R et al,2010). *Connected Components* merupakan teknik yang digunakan untuk mengklasifikasikan daerah dan/atau *region* suatu objek pada citra *digital*. Piksel-piksel dalam *region* disebut *connected* karena adanya keterhubungan satu sama lain; yakni mematuhi aturan *adjacency* atau kedekatan suatu piksel. (Gonzales et al, 2004). Ada dua jenis konektivitas yang sering digunakan yaitu 4-Konektivitas/4-*Connected Neighborhood* dan 8-Konektivitas/8-*Connected Neighborhood*. Perbedaannya terletak pada sifat 4- Konektivitas yang menganggap piksel yang bersinggungan sebagai dua objek yang berbeda, sementara pada 8- Konektivitas, piksel yang bersinggungan hanya akan dianggap sebagai satu objek yang sama.

Cara menentukan *Connected Components* dari Rosenfeld-Pfaltz (1996) seperti yang dikutip dari Tetsuo A.dan Hiroshi T (2010)

1. Cek nilai piksel pertama pada citra masukan yaitu padabaris 1 kolom ke 1. Apabila nilai tersebut 0 (*background*), lewati dan pindah ke kolom selanjutnyapada baris yang sama. Apabila piksel bernilai 1, maka piksel tetangganya dicek. Apabila tetangga terdekatnya (*neighbour*) belum terlabel, berikan label pada piksel yang ditinjau dengan label 1.
2. Apabila memiliki satu tetangga terdekat yang sudah terlabel, beri label yang sama dengan label tetangga. Dan jika memiliki lebih dari satu tetangga terdekat dengan label berbeda, beri label menggunakan label terkecil dari tetangga-tetangga tersebut.
3. Setiap selesai melakukan pelabelan dan bertemu kembali dengan *background*, perbarui indeks label.

2.2.4. *Structuring Elements*

*Structuring elements* (strel) merupakan himpunan kecilyang dipakai untuk meneliti citra dalam pembelajaran properti. Piksel yang terletak pada bagian tengah dari *structuring elements* biasa disebut dengan *origin*. Kemudian *origin* akan mengidentifikasi piksel yang menjadi inti. Ada beberapa bentuk *structuring element* yang biasa digunakan, yaitu *rectangle, square, disk, linear, dan diamond*. Dandalam penelitian ini digunakan strel berbentuk *disk* dan *line*.

2.2.5. *Matlab*

Matlab atau *Matrix Laboratory* adalah sebuah perangkat lunak pengolah yang dikembangkan oleh Mathworks, Inc. Perangkat lunak ini mampu menjalankan teknik komputasi numerik dan berbagai pengolahan yang dibutuhkan dengan operasi matematika (Wakhidah, 2012). MATLAB sendiri banyak digunakan untuk:

- a. Komputasi dan Matematika
- b. Pemrograman simulasi, *prototype*
- c. Analisa data, numerik dan statistik

## 2. HASIL DAN PEMBAHASAN

### 1.1. Dataset

Dalam penelitian ini kami menggunakan 10 data citra. Gambar Berikut adalah salah satu contoh citra yang digunakan.



Gambar 3.1 Data Set (Citra 1)

### 1.2. Hasil

Dari hasil pengujian proses deteksi plat nomor, didapatkan perbedaan waktu deteksi dengan dan tanpa pra-pengolahan. Dapat dilihat pada Gambar 3.2 bahwa pra-pengolahan mempercepat hasil deteksi.

```
Elapsed time is 3.286453 seconds.
>> case
n =
    1
Elapsed time is 1.397692 seconds.
```

Gambar 3.2 Tangkapan Layar Perbedaan Waktu Komputasi Tabel 3.1 menunjukkan percepatan waktu komputasi.

Hasil rata-rata kecepatan waktu dengan pra-pengolahan (*resizing* dan *grayscale*) adalah sebesar 1,45731 detik, sedangkan tanpa pra-pengolahan rata-rata sebesar 3,721519 detik. Dari 10 (sepuluh) citra uji, hanya delapan citra yang berhasil dideteksi. Hal ini dipengaruhi oleh intensitas cahayadan posisi pada saat pengambilan citra. Berdasar jumlah platdan keberhasilan hasil uji, dapat disimpulkan bahwa akurasi penggunaan *Connected Components* sebagai berikut:

$$\frac{\text{Jumlah Plat yang berhasil dideteksi}}{\text{Jumlah plat uji}} \times 100\% \dots\dots\dots (6)$$

Sehingga didapatkan hasil sebagai berikut:

$$\frac{8}{10} \times 100\% = 80\%$$

Hasil uji menunjukkan bahwa pernyataan pra- pengolahan mempercepat jalannya proses deteksi plat nomor adalah benar (Wakhidah, 2012). Hal ini selaras dengan pernyataan bahwa pemrograman yang baik adalah program yang dapat dijalankan secara cepat (M.Bister,2007). Dapat disimpulkan pula bahwa pra-pengolahan adalah langkah penting sebelum deteksi plat nomor (Soon.C et al, 2012).

Tabel 3.1 Hasil Pengolahan

No	Data Uji	Hasil	Waktu komputasi tanpa pra- pengolahan(dalam detik)	Waktu komputasi dengan pra-pengolahan (dalam detik)
1	Citra 1	Berhasil	5.418814	1.840519
2	Citra 2	Berhasil	6.401825	3.414187
3	Citra 3	Berhasil	4.744928	1.539978
4	Citra 4	Berhasil	3.362250	1.101627
5	Citra 5	Tidak Berhasil	3.198699	1.038317
6	Citra 6	Berhasil	3.278893	1.362885
7	Citra 7	Berhasil	3.552732	1.332049
8	Citra 8	Berhasil	3.583700	1.653616
9	Citra 9	Berhasil	3.223926	1.129535
10	Citra 10	Tidak Berhasil	4.4942016	1.603887
	Rata—rata		3.721519	1.45731

Berikut ini adalah gambar contoh tampilan hasil deteksi



Gambar 3.3 Contoh Tampilan Hasil Deteksi

### 3. SIMPULAN

Dari pengujian yang telah dilakukan, kesimpulan dari penelitian ini antara lain:

- a. Pra-pengolahan berpengaruh terhadap waktu deteksi, yaitu mempercepat proses program ketika dijalankan. Hal ini sesuai dengan pernyataan pada penelitian sebelumnya bahwa semakin sedikit ukuran data maka data akan kian cepat untuk diproses, hal ini ditunjukkan dengan rata-rata deteksi tanpa pra- pengolahan sebesar 3,721519 detik dan rata-rata deteksi dengan pra-pengolahan sebesar 1,45731detik.
- b. Plat nomor berhasil dideteksi menggunakan *Connected Components* dengan akurasi

sebesar 80% setelah melalui tahap pra-pengolahan *Resizing* dan *Grayscale* serta tahap pengolahan yakni Deteksi Tepi, Dilasi, *Filling Holes* dan *Opening* diperoleh, kelebihan dan kekurangannya, serta kemungkinan pengembangan selanjutnya.

#### 4. SARAN

Saran untuk penelitian selanjutnya berdasar penelitian kali ini adalah:

1. Adanya penambahan pra-pengolahan lain untuk meningkatkan akurasi pendeteksian, misalnya mengurangi intensitas cahaya pada citra kendaraan bermotor yang terlalu terang karena dapat mempengaruhi hasil deteksi.
2. Mendeteksi selain plat nomor kendaraan pribadi (*background* hitam dan tulisan putih)
3. Menambah metode yang mampu mendeteksi karakter pada plat nomor-saran untuk penelitian lebih lanjut untuk menutupi kekurangan penelitian. Tidak memuat saran-saran diluar untuk penelitian lanjut.

#### DAFTAR PUSTAKA

- Achmad Solichin dan Zulfikar Rahman.(2015) Identifikasi Plat Nomor Kendaraan Berbasis Mobile dengan Metode Learning Vector Quantization. *Jurnal TICOM*(3) ISSN 2302 - 3252
- Anita, S. (2017) Implementasi Teknik Threshoding Pada Segmentasi Citra Digital. *Jurnal Mantik Penusa*. (1) 48-51. p-ISSN 2088-3943, e-ISSN 2580-9741
- Akmal.R et al (2010) Connected Component Labeling Using Components Neighbors-Scan Labeling Approach. *Journal of Computer Science* (6) 1070-1078. ISSN 1549-3636
- Dr. Ziad AlQadi, Dr. Rushdi S. Abu Zneit, dan Dr. Mohammad Abu Zalata (2017) Extracting Individual Objects from RGB Color Image. *International Journal of Computer Science and Mobile Computing* 6 190-196.ISSN 2320-088X
- Daniel Onoro Rubio, Artem Lenskiy dan Jee-Hwan Ryu. 2015. Connected Components for a Fast and Robust 2DLIDAR Data Segmentation. *Korea University of Technology and Education*
- Dianne et al. (2005). *Biophysical Journal* (88) 761-762. doi:10.1529/biophysj.105.059600
- Erwin Dwika Putra 1 dan Stefanus Santosa. (2017) Optimasi Kemampuan Segmentasi Otsu Pada Identifikasi Plat Nomor Kendaraan Indonesia Menggunakan Metode Gaussian. *Jurnal Pseudocode*. ISSN 2355-5920
- Puspiningrum et al (2018). Deteksi Letak Plat Nomor Kendaraan Dengan Metode Wavelets Transform. *SCAN XIII*. (1) ISSN : 1978-0087.
- G. Jyothi et al, 2015. Luminance Based Conversion of Gray Scale Image to RGB Image. *International Journal of Computer Science and Information Technology Research* (3) 279- 283. ISSN 2348-1196.
- Gonzales et al. Digital Image Processing. (2004) ISBN-13: 978-0201180756, ISBN-10: 0201180758
- Huning, Sri et al (2010) Estimasi Bentuk Structuring Element Berdasar Representasi Obyek. *Jurnal Ilmiah WATSON*.(3) 157-165. ISSN 0 2 1 -6- 0544
- Kanan C, Cottrell GW (2012) Color-to-Grayscale: Does the Method Matter in Image Recognition? *PLoS ONE* 7(1):e29740. doi:10.1371/ journal.pone.0029740
- K.S. Mahalakshmi, et al (2013) Identification Technique in Defence Sector Using Digital Signal Processing and Digital Image Processing. *International Journal of Technology and*

*Engineering Science(I)* 1376-1379.ISSN: 2320— 8007

- M.Bister dan KH Ng (2007) Increasing The Speed Of Medical Image Processing In Matlab. *Biomedical Imaging and Intervention Journal*. doi: 10.2349/bij.3.1.e9
- Mazen A.Hamdan, Dr. Ziad AlQadi, dan Bassam M.Subaih (2016). A Methodology to Analyze Objects in Digital Image using Matlab. *International Journal of Computer Science and Mobile Computing* (5) 21-28. ISSN 2320-088X
- Munir, R. (2004). Pengolahan Citra Digital dengan Pendekatan Logaritmik. Bandung: Informatika. Muhamad Masjun Efendi. (2018) Metode Deteksi Tepi Block Jpeg Terkompresi Untuk Meningkatkan Akurasi Analisis Manipulasi Splicing Pada Citra Berekstensi