

Analisis Perbandingan Algoritma *Edge Detection* Pada Plat Kendaraan Bermotor

Anindita Saraswati¹, Miftahul Jannah²

^{1,2}STMIK Pelita Nusantara, Jl. Iskandar Muda No.1, Medan 20154, Indonesia

¹ aninditasaraswati30@yahoo.com; ² miftahuljannah0077@gmail.com

INFORMASI ARTIKEL

ABSTRAK

KataKunci:

Analisis Perbandingan
Algoritma Edge Detection
Plat Kendaraan Bermotor

Penelitian ini dilakukan guna menganalisis perbandingan algoritma deteksi tepi dalam mengidentifikasi pola nomor plat kendaraan bermotor, di dapatkan bahwa algoritma deteksi tepi memiliki beberapa perbedaan dalam mengenali citra dengan kualitas yang berbeda, kualitas citra yang digunakan yaitu *high defenition* serta blur. Algoritma deteksi tepi yang dipakai yaitu canny, sobel dan prewitt dikarenakan algoritma deteksi tepi tersebut sangat banyak digunakan namun tidak menunjukkan rating *detection rate* yang dihasilkan. Maka dari itu parameter pembanding yang digunakan pada penelitian ini adalah parameter *detection rate*. Kesimpulan dari penelitian ini adalah algoritma sobel merupakan algoritma terbaik dikarenakan memiliki nilai *detection rate* sebanyak 100% sedangkan canny sebanyak 92% dan sobel sebanyak 80%.

Keywords:

Comparative Analysis
Edge Detection Algorithm
Motor Vehicle License Plate

ABSTRACT

This research was conducted to analyze the comparison of edge detection algorithms in identifying motor vehicle plate number patterns. It was found that the edge detection algorithm has several differences in recognizing images with different quality, the image quality used is high definition and blur. The edge detection algorithms used are canny, sobel and prewitt because these edge detection algorithms are very widely used but do not show the resulting detection rate rating. Therefore, the comparative parameter used in this study is the detection rate parameter. The conclusion of this research is that the Sobel algorithm is the best algorithm because it has a detection rate of 100%, while the Prewitt algorithm is 92% and the Canny is 80%.

I. Pendahuluan

Banyak teknologi saat ini yang menggunakan pengolahan citra sebagai ilmu penopang untuk menyelesaikan permasalahan yang dihadapi. Di dalam pengolahan citra terdapat beberapa metode yang bisa dipakai salah satu nya deteksi tepi / konvolusi citra. Dalam metode ini, terdapat juga algoritma-algoritma yang bisa digunakan di antara nya algoritma sobel, canny dan prewitt. Namun pastinya setiap algoritma memiliki perbedaan, oleh karena itu pada penelitian ini akan dilakukan perbandingan kinerja dari setiap algoritma [1].

Perbandingan kinerja algoritma-algoritma tersebut akan dihitung berdasarkan *detection rate* dari setiap algoritma yang dipakai dengan menguji beberapa citra plat kendaraan bermotor. Kualitas citra kemungkinan akan berpengaruh terhadap kinerja algoritma deteksi tepi, seperti pada penelitian terdahulu yang dilakukan oleh Yashinta Oktayiana Legu pada tahun 2019 dengan judul skripsi Deteksi Plat Nomor Kendaraan Bermotor dengan Segmentasi Gambar, menggunakan algoritma prewitt dan sobel sebagai pendeteksi tepi citra plat kendaraan bermotor. Namun masih banyak kekurangan dalam penelitian tersebut karna citra yang digunakan berkualitas *high defenition*, yang hasil deteksi tepi nya pasti akan sangat jelas. Oleh karena itu pada kesempatan ini akan ada 2 kualitas citra yang digunakan sebagai kriteria penelitian yaitu citra dengan kualitas *high defenition* dan citra dengan kualitas blur dimana format citra yang digunakan adalah jpeg.

Tujuan dilakukannya penelitian ini adalah dapat menerapkan 3 algoritma *edge detection* yaitu canny, sobel dan prewitt untuk mengidentifikasi nomor plat kendaraan bermotor. Dan dapat melihat kinerja dari algoritma *edge detection* canny, sobel dan prewitt dalam mengidentifikasi nomor plat kendaraan bermotor.

Ruang lingkup masalah pada penelitian ini adalah penelitian ini dibuat untuk membandingkan 3 algoritma *edge detection* yaitu canny, sobel dan prewitt. Deteksi tepi nomor plat kendaraan bermotor menggunakan 3 algoritma *edge detection* yaitu canny, sobel dan prewitt tanpa menggunakan device pendukung.

Pembangunan sistem menggunakan software Microsoft Visual basic 2010. Pengujian algoritma *edge detection* canny, sobel dan prewitt menggunakan Parameter "*Detection Rate*". Kriteria citra yang dibutuhkan adalah gambar plat kendaraan bermotor dengan kualitas *high defenition* serta blur.

II. Metode

Nilai suatu pixel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0 - 255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer [2]. Pengolahan citra digital merupakan proses memanipulasi citra dengan bantuan komputer tujuannya yakni memperbaiki kualitas citra dan mengolah informasi pada gambar untuk keperluan pengenalan dan proses selanjutnya[3]. Deteksi tepi digunakan untuk menyaring citra dengan mengurangi banyaknya data tapi tidak menghilangkan informasi penting yang terdapat didalamnya [4]. Tujuan pendeteksian tepi adalah untuk menandai bagian yang menjadi detail citra dan untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra [5].

A. Operator Sobel

Tujuan operasi deteksi tepi adalah meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Operator gradien pertama yang digunakan untuk mendeteksi tepi di dalam citra, yaitu operator gradien selisih terpusat, operator Sobel, operator Prewitt, operator Roberts, operator Canny[6].

Operator sobel adalah salah satu operator yang menghindari adanya perhitungan gradient di titik interpolasi. Operator ini menggunakan kernel 3x3 pixel untuk perhitungan gradient sehingga perkiraan gradient berada tepat ditengah jendela. Misalkan susunan pixel-pixel disekitar pixel (x,y) adalah :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Berdasarkan susunan pixel tetangga tersebut, besaran gradient yang dihitung menggunakan operator sobel adalah :

$$M = \sqrt{S_x^2 + S_y^2}$$

Turunan parsial dihitung dengan :

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$S_y = (a_0 + ca_1 + a_{22}) - (a_6 + ca_5 + a_4)$$

Keterangan :

S_x = nilai keseluruhan sumbu x

S_y = nilai keseluruhan sumbu y

Dengan konstanta C=2 dalam bentuk maks, S_x dan S_y dapat dinyatakan sebagai :

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Area tepinya dihitung dengan persamaan :

$$\alpha(x,y) = \tan^{-1} \left\{ \frac{S_y}{S_x} \right\}$$

B. Operator Prewitt

Metode Prewitt merupakan pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF. Persamaan gradien pada operator prewitt sama dengan gradien pada operator sobel perbedaannya adalah pada prewitt menggunakan konstanta c = 1 [7].

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Kelebihan dari metode prewitt ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi. Peninjauan pengaturan pixel di sekitar pixelnya (x,y) adalah :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Operator prewitt adalah magnitudo dari gradien yang dihitung dengan :

$$M = \sqrt{Px^2 + Py^2}$$

Dalam hal ini turunan parsial dihitung dengan :

$$Px = (a_2 + ca_3 + a_4) - (a_0 + ca_1 + a_6)$$

$$Py = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$

Dengan konstanta $c = 2$ [8].

C. Operator Canny

Langkah-langkah dalam melakukan deteksi tepi Canny yang pertama adalah menghilangkan derau yang ada pada citra dengan menerapkan tapis Gaussian [9]. Proses ini menghasilkan citra yang tampak sedikit buram. Hal ini dimaksudkan untuk mendapatkan tepian citra yang sebenarnya. Bila tidak dilakukan maka garis-garis lurus juga akan dideteksi sebagai tepian. Selanjutnya melakukan deteksi tepi dengan salah satu operator deteksi tepi (*Operator Sobel* dan *Prewitt*) dengan melakukan pencarian secara horizontal (G_x) dan secara vertical (G_y) melalui rumus :

$$|G| = |G|$$

G_x = pencarian secara horizontal

G_y = pencarian secara vertikal

Untuk menghitung jarak *gradient*, digunakan persamaan berikut :

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Kemudian untuk menghitung arah dari garis tepi yang dihasilkan, digunakan persamaan berikut :

$$\theta = \arctan\left(\frac{G_x}{G_y}\right)$$

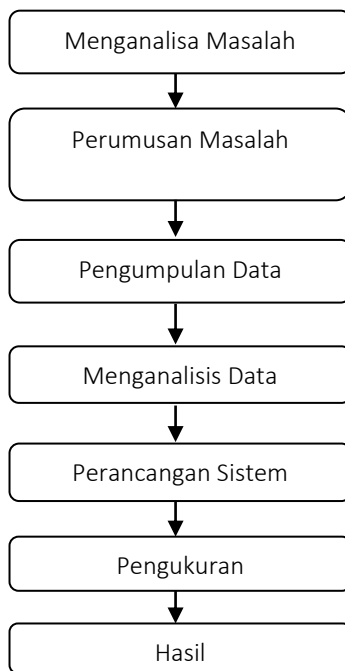
Langkah berikutnya adalah membagi garis-garis yang ada menjadi 4 warna terpisah dengan sudut masing-masing dengan ketentuan yaitu :

- 1) Derajat 0 - 22,5 dan 157,5 - 180 berwarna kuning.
- 2) Derajat 22,5 - 67,5 berwarna Hijau.
- 3) Derajat 67,5 - 157,5 berwarna Merah.

Kemudian memperkecil masing-masing garis tepi agar menjadi tipis (*non maximum surpression*). Langkah terakhir adalah melakukan proses binerisasi berdasarkan nilai *low & high threshold* yang diberikan [4].

D. Kerangka Kerja Penelitian

Kerangka penelitian membahas model pelaksanaan penelitian yang merupakan langkah-langkah yang akan diambil dalam menyelesaikan masalah penelitian. Tahapan dalam penelitian ini adalah sebagai berikut:

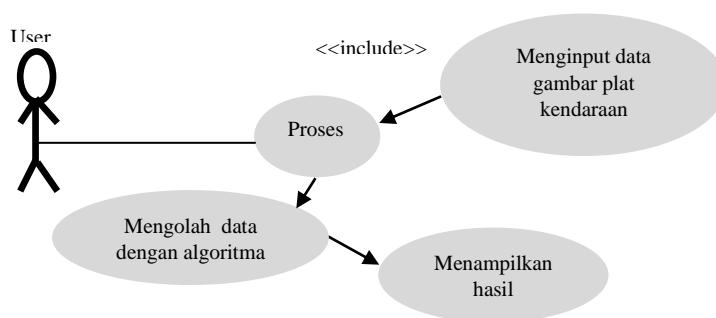


Gambar 1. Kerangka Kerja Penelitian

E. Perancangan Sistem

Dalam perkembangannya, standarisasi dalam metode UML dilakukan dengan standarisasi *proses Object Management Group (OMG)*, sehingga menjadikan UML sebagai bahasa standar dalam permodelan perangkat lunak yang saat ini banyak digunakan oleh berbagai kalangan.

Use case diagram digambarkan sebagai komponen yang memiliki fungsi untuk memodelkan proses kerja atau bisnis dari aplikasi yang akan dibangun. Use case merupakan abstraksi dari proses interaksi antara pengguna dengan fungsi atau tindakan yang dapat dilakukan terhadap sistem.[10].



Gambar 2. Use Case Diagram Sistem

Tabel 1. Penjelasan Use Case Diagram



Usecase Name	Description
Proses	Memproses data dengan algoritma
Menginput plat kendaraan bermotor	Memasukkan data berupa gambar plat kendaraan bermotor
Proses data dengan algoritma	Menghitung nilai piksel dengan algoritma terpilih
Menampilkan hasil	Menampilkan hasil dari deteksi tepi

III. Hasil dan Pembahasan

A. Data Uji









Dibawah ini adalah data citra asli yang akan diolah dengan algoritma deteksi tepi.

Tabel 2. Contoh Data

No	Image	Quality	Size
1		HD	10x5
2		HD	10x5
3		Blur	5x5
4		Blur	10x5
5		Blur	10x5
6		HD	8x5

Setelah data di inputkan kedalam sistem, maka hasil yang didapatkan adalah sebagai berikut :

Tabel 3. Tabel Hasil Pengujian

No	Gambar	Kualitas	Alg. Sobel	Alg. Prewitt	Alg. Canny
1		HD			
2		HD			



Setelah didapatkan data hasil pengujian, maka selanjutnya menghitung parameter perbandingan. Ketahanan algoritma deteksi tepi terhadap *noise* dapat diukur dengan parameter *detection rate* sebagai berikut :

$$p = \frac{y}{z} \times 100\%$$

dimana :

p = algoritma yang akan dihitung

y = jumlah data yang garis tepinya terdeteksi

z = jumlah data keseluruhan yang garis tepinya akan dideteksi

Nilai p akan menunjukkan kinerja dari algoritma yang dipakai untuk mendeteksi tepi citra. Nilai akhir akan berbentuk persentase. Perbandingan algoritma dengan parameter ini akan lebih mudah karena hasil terbaik dan kinerja yang akurat dari sebuah algoritma akan ditunjukkan dengan melihat nilai persentase yang paling tinggi. Maka hasil perbandingan dapat dilihat dalam tabel berikut :

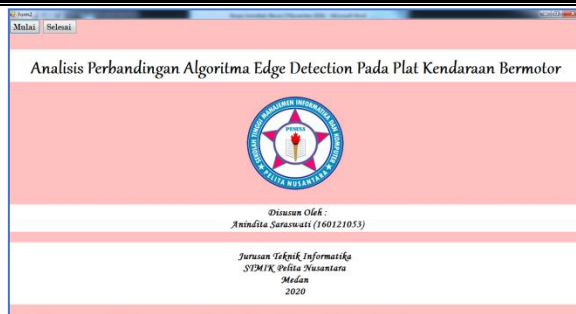
Tabel 4. Hasil Perhitungan Parameter Perbandingan

No	Algoritma	Jumlah Data Keseluruhan	Jumlah Data yang Terdeteksi	Jumlah Data yang Tidak Terdeteksi/Gelap	Persentase
1	Sobel	25	25	0	100%
2	Prewitt	25	23	2	92%
3	Canny	25	20	5	80%

Berdasarkan tabel di atas, maka dapat disimpulkan bahwa algoritma sobel dapat dikategorikan algoritma terbaik dengan persentase deteksi tepi 100% karena dapat mendeteksi citra dengan kualitas blur dan *high defenition* dengan baik, sedangkan algoritma prewitt berada di posisi kedua dengan nilai persentase 92%. Dan canny di posisi terakhir dengan nilai persentase 80%.

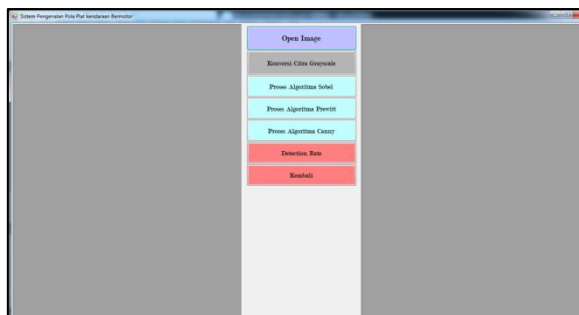
B. Tampilan Interface

Berikut ini adalah tampilan interface dari sistem yang telah dibangun :



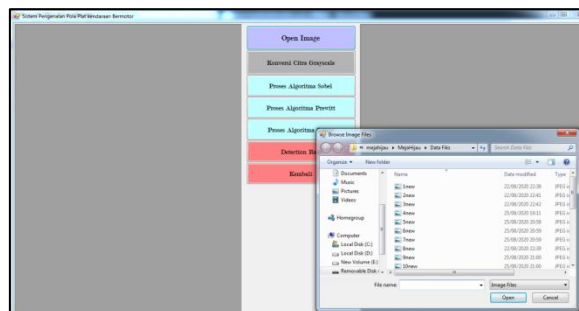
Gambar 3. Tampilan Utama

Gambar di atas menunjukkan tampilan utama dari sistem, dengan 2 menu utama yaitu mulai dan selesai. Jika menu mulai dipilih maka tampilan yang muncul sebagai berikut :



Gambar 4. Tampilan Menu Mulai

Pada menu ini terdapat beberapa option button seperti open image, konversi citra grayscale, dan lainnya. Setiap button memiliki proses masing-masing.



Gambar 5. Tampilan Button Open Image

Jika open image dipilih maka akan muncul tampilan windows explorer dimana pengguna harus menginputkan sebuah citra yang akan digunakan untuk proses selanjutnya.



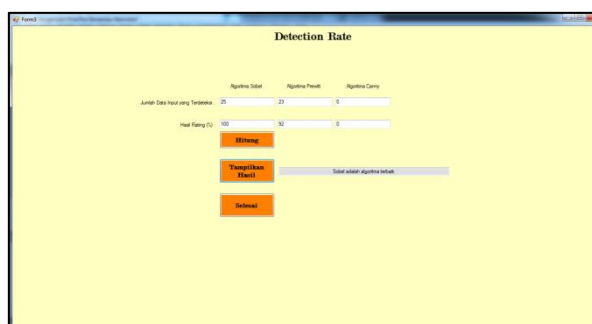
Gambar 6. Tampilan Konversi Citra Grayscale

Setelah citra diinputkan maka user harus mengkonversikan citra rgb ke dalam bentuk citra grayscale dengan memilih button konversi citra grayscale.



Gambar 7. Tampilan Proses Algoritma

Setelah dikonversikan ke dalam citra grayscale maka selanjutnya adalah memilih button proses algoritma yang telah disediakan untuk mulai mendeteksi tepi citra.



Gambar 8. Tampilan Detection rate

Dan setelah semua data terdeteksi, user bisa mulai menganalisis algoritma dengan memilih button detection rate yang nantinya akan memberikan solusi dan persentase kinerja dari sebuah algoritma yang telah dipakai.

IV. Kesimpulan

Setelah melakukan penelitian terhadap beberapa algoritma deteksi tepi, maka dapat disimpulkan beberapa hal sebagai berikut :

- 1) Kualitas citra blur dapat mempengaruhi hasil deteksi tepi yang dilakukan oleh sistem.
- 2) Pada citra dengan kualitas blur beresolusi rendah, deteksi tepi bisa dilakukan dengan lebih cepat jika dibandingkan dengan citra kualitas blur beresolusi tinggi.
- 3) Hasil deteksi algoritma sobel lebih tegas daripada algoritma prewitt dan canny.
- 4) Jika ditinjau dari kinerja, algoritma sobel dikategorikan algoritma terbaik untuk mendeteksi tepi dengan rating tertinggi dengan nilai deteksi 100%.

Daftar Pustaka

- [1] D. Avianto, "Pengenalan Pola Karakter Plat Nomor Kendaraan Menggunakan Algoritma Momentum Backpropagation Neural Network," J. Inform., 2016.
- [2] M. R. Kumaseh, L. Latumakulita, And N. Nainggolan, "Segmentasi Citra Digital Ikan Menggunakan Metode Thresholding," J. Ilm. Sains, 2013.
- [3] A. Ambarwati, R. Passarella, And Sutarno, "Segmentasi Citra Digital Menggunakan Thresholding Otsu Untuk Analisa Perbandingan Deteksi Tepi," Annu. Res. Semin. 2016, 2016.

-
- [4] Y. O. L. Rema, "Deteksi Plat Nomor Kendaraan Bermotor Dengan Segmentasi Gambar," J. Saintek Lahan Kering, 2019.
 - [5] Y. Qin, Y. Wu, Q. Wang, And S. Yu, "Method For Pests Detecting In Stored Grain Based On Spectral Residual Saliency Edge Detection," Grain Oil Sci. Technol., 2019.
 - [6] A. Padmo A.M And M. Murinto, "Segmentasi Citra Batik Berdasarkan Fitur Tekstur Menggunakan Metode Filter Gabor Dan K-Means Clustering," J. Inform., 2016.
 - [7] M. Zainuddin, L. T. Sianturi, And R. K. Hondro, "Implementasi Metode Robinson Operator 3 Level Untuk Mendeteksi Tepi Pada Citra Digital," J. Ris. Komput., 2017.
 - [8] A. Pranata And E. Z. Astuti, "Pengolahan Citra Berbasis Deteksi Tepi Prewitt Pada Gambar Gigi Manusia," Eksplora Inform., 2017.
 - [9] L. Wang, Y. Shen, H. Liu, And Z. Guo, "An Accurate And Efficient Multi-Category Edge Detection Method," Cogn. Syst. Res., 2019.
 - [10] D. Faitelson And S. Tyszberowicz, "Uml Diagram Refinement (Focusing On Class-And Use Case Diagrams)," In Proceedings - 2017 Ieee/Acm 39th International Conference On Software Engineering, Icse 2017, 2017.