

Segmentasi pada Plat Kendaraan Menggunakan Metode Deteksi Tepi Canny dan *Thresholding*

Ardian E. Rumetna¹, B. Yudi Dwiandiyanta², Patricia Ardanari³

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta

Jl. Babarsari No.43, Kabupaten Sleman 55281, Daerah Istimewa Yogyakarta, Indonesia

Email: ¹150708477@students.uajy.ac.id, ²yudi.dwiandiyanta@uajy.ac.id,

³patricia.ardanari@uajy.ac.id

Abstrak. Teknologi yang semakin lama semakin pesat di zaman sekarang ini membuat pengetahuan semakin jauh meningkat. Segmentasi plat kendaraan yang digunakan dalam penelitian ini untuk melihat perbandingan metode deteksi tepi canny dan *thresholding* pada plat kendaraan. Untuk melakukan segmentasi plat kendaraan tersebut akan dibuat model dengan menggunakan bahasa pemrograman *python* dan menggunakan tools *jupyter notebook*. Model digunakan untuk melihat perbandingan cara kerja, proses dan hasil segmentasi pada kedua metode tersebut. Segmentasi plat kendaraan dilakukan untuk membandingkan dua metode yang berbeda yang memiliki kekurangan dan kelebihan pada masing-masing metode. Dari hasil pengujian yang telah dilakukan dengan menggunakan metode deteksi tepi canny dan *thresholding* dengan jumlah data yang diuji sebanyak 100 gambar. Hasilnya adalah metode deteksi tepi canny sedikit lebih unggul dalam tingkat keakuratan yang tepat berjumlah 90 untuk metode deteksi tepi canny dan 85 untuk metode *thresholding*.

Kata Kunci: Segmentasi, Plat Nomor, Deteksi Tepi Canny, *Thresholding*, *Grayscale*.

1. Pendahuluan

Kendaraan merupakan alat transportasi yang dipakai untuk berpindah dari satu tempat ketempat lainnya. Baik itu kendaraan buatan manusia seperti mobil, motor, kereta, perahu dan pesawat. Maupun kendaraan yang bukan buatan manusia namun berupa hewan yang dijadikan sebagai kendaraan oleh manusia seperti kuda, unta dan lainnya. Secara umum kendaraan memiliki plat yang berfungsi untuk menjadi identitas dari sebuah kendaraan. Bentuk plat berupa potongan logam yang dipasang pada kendaraan sebagai identitas resmi. Membuat plat nomor atau Tanda Nomor Kendaraan Bermotor (TNKB) di Indonesia sudah diatur dalam Peraturan Pemerintah No. 60 tahun 2016 dengan masa berlaku yaitu 5 tahun. Aturan tersebut mengenai jenis dan tarif Penerimaan Negara Bukan Pajak (BNBP), dan sudah resmi diberlakukan pada tanggal 6 Januari 2017, dengan menggantikan Peraturan Pemerintah No.50 tahun 2010, dikutip dari situs surat kabar Kompas.

Masalah-masalah yang dihadapi oleh pengguna kendaraan antara lain, pencurian kendaraan yang marak terjadi dan tak kenal waktu serta tempat. Kendaraan harus berhenti dipintu tol untuk melakukan pembayaran, tentu ini memakan waktu dan juga memakan tenaga dari karyawan yang bekerja di pintu tol. Parkir di kampus, pusat perbelanjaan, hotel, dan lainnya yang belum otomatis, ini juga memakan waktu dan tenaga bagi penjaga pintu parkir. Identifikasi plat nomor kendaraan adalah bagian dari pemrosesan gambar digital yang banyak digunakan dalam sistem transportasi kendaraan untuk mengidentifikasi kendaraan. Sistem pengenalan plat nomor memiliki beragam aplikasi seperti pemeliharaan lalu lintas, pelacakan mobil curian, sistem pengumpulan tol elektronik otomatis, dan banyak lagi. Tetapi tujuan utamanya adalah untuk mengontrol sistem manajemen lalu lintas [1]. Perkembangan di zaman ini semakin banyak berubah terutama dalam bidang teknologi. Dengan berkembang pesat teknologi, ilmu juga akan semakin berkembang, banyak metode yang digunakan untuk menyelesaikan masalah namun tidak semua metode sempurna. Metode yang ada pasti memiliki

kelebihan dan kelemahan masing-masing maka dari itu tergantung pengguna metode menentukan dan memilih menggunakan metode yang sesuai dengan kasus yang dimiliki. Ada beberapa metode yang dapat digunakan untuk melakukan segmentasi pada plat kendaraan seperti *Automatic Plate License Plate Recognition (ALPR)* atau dikenal juga dengan sebutan *automatic vehicle identification, car plate recognition, automatic number plate recognition*, dan *optical character recognition (OCR)* [2].

Segmentasi gambar adalah proses pemartisian / pengelompokan gambar digital menjadi wilayah yang bermakna atau kelompok area piksel sehubungan dengan aplikasi tertentu. Segmentasi didasarkan pada pengukuran yang diambil dari gambar dan mungkin tingkat abu-abu, warna, tekstur, kedalaman atau gerakan. Hasil segmentasi gambar adalah sekumpulan segmen yang secara kolektif mencakup seluruh gambar. Semua piksel di wilayah tersebut mirip dengan yang terkait dengan beberapa karakteristik atau properti yang dihitung, seperti warna, intensitas, atau tekstur. Wilayah yang berdekatan berbeda sehubungan dengan karakteristik yang sama [1]. Mengenali objek dan lingkungan sekitarnya adalah tugas yang mudah bagi manusia. Tetapi jika penerapannya secara *artificial*, maka itu menjadi tugas yang sangat kompleks [3]. Pengenalan pola melakukan pengelompokan data, berupa data numerik dan simbolik. Data dikelompokkan dengan menggunakan komputer secara otomatis [4]. Pengelompokan ini bertujuan untuk dapat mengenali / mengidentifikasi ciri-ciri dari suatu objek/pola tertentu. Pola yang sudah dikelompokkan berdasarkan ciri-cirinya digunakan untuk dapat membedakan antara masing-masing pola. Deteksi tepi adalah alat mendasar dalam pemrosesan gambar dan *computer vision*, khususnya dibidang deteksi fitur dan ekstraksi fitur, yang bertujuan mengidentifikasi titik-titik dalam gambar digital di mana kecerahan gambar berubah tajam atau, lebih formal, memiliki diskontinuitas [5]. Suatu *edge* merupakan batas antara dua sisi yang memiliki grey level yang relatif berbeda.

Deteksi tepi canny dapat mendeteksi tepian dengan tingkat kesalahan minimum, deteksi tepi canny mempunyai perbedaan dengan operator lainnya karena menggunakan Gaussian Derivative Kernel yang dapat memperhalus tampilan citra [6]. Beberapa keunggulan deteksi tepi canny diantaranya, *good detection* dapat memaksimal *signal to noise ration (SNR)* yang berguna untuk semua tepi dapat terdeteksi dengan baik. *Good location* dapat meminimalkan jarak deteksi tepi yang dihasilkan dengan melalui pemrosesan, agar lokasi tepi dapat terdeteksi serupa dengan tepi secara nyata. Dan yang terakhir *one respon to single edge* yang dapat menghasilkan tepi tunggal, tidak memberi tepi yang tidak benar atau palsu. *Thresholding* adalah salah satu metode dalam melakukan segmentasi gambar, *Thresholding* memilih nilai batas abu-abu yang optimal dari sebuah gambar, agar dapat memisahkan objek yang menarik dalam sebuah gambar, dimana objek tersebut dipisahkan dari latar belakang gambar berdasarkan tingkat keabuannya atau gelap terangnya [7]. Dalam keluaran atau output dari segmentasi *Thresholding* adalah berupa citra biner. Dikarenakan pada prosesnya region atau wilayah citra yang gelap akan dibuat semakin gelap dengan nilai 0 atau sama saja dengan warna hitam. Begitu pula sebaliknya, bila wilayah citra terang maka akan dibuat semakin terang dengan nilai 1 atau sama saja dengan warna putih

2. Tinjauan Pustaka

2.1 Segmentasi

Segmentasi mengacu pada proses mempartisi gambar digital ke dalam segmen (set piksel, juga dikenal sebagai piksel super). Segmentasi gambar biasanya digunakan untuk menemukan objek dan batas (garis, kurva, tepi, dll.) Dalam suatu gambar. Lebih tepatnya, segmentasi gambar adalah proses pemberian label ke setiap piksel dalam suatu gambar sehingga piksel dengan label yang sama berbagi karakteristik visual tertentu. Hasil segmentasi gambar adalah seperangkat segmen yang secara kolektif mencakup seluruh gambar, atau seperangkat kontur yang diekstraksi dari gambar.

2.2 Deteksi Tepi

Operator deteksi tepi terutama dibagi ke dalam metode diferensial, yaitu metode operator yang optimal, dan metode yang didasarkan pada morfologi matematika. Pertama metode diferensial pesanan dari operator Robert, operator Prewitt, operator Sobel, Kirsch operator, dll. Operator diferensial urutan kedua seperti operator diferensial *Laplace*; metode operator terbaik adalah operator *Laplacian of Gaussian*, LoG dan Canny. Beberapa kriteria yang umum dalam deteksi tepi adalah: Deteksi tepi dengan tingkat kesalahan rendah, yaitu deteksi menangkap sebanyak mungkin tepi yang ditunjukkan pada gambar dan harus secara akurat. Titik tepi yang terdeteksi dari operator harus secara akurat melokalisasi di tengah tepi. Tepi yang diberikan pada gambar hanya boleh ditandai satu kali, dan jika memungkinkan, noise gambar tidak boleh membuat tepi yang salah. Detektor tepi Canny banyak digunakan dalam computer vision agar perubahan intensitas yang tajam dan batas-batas dalam gambar dapat ditemukan. Detektor tepi Canny mengklasifikasikan piksel sebagai tepi jika besarnya gradien piksel lebih besar daripada piksel di kedua sisi-sisinya ke arah perubahan intensitas maksimum [6].

2.3 Deteksi Tepi Canny

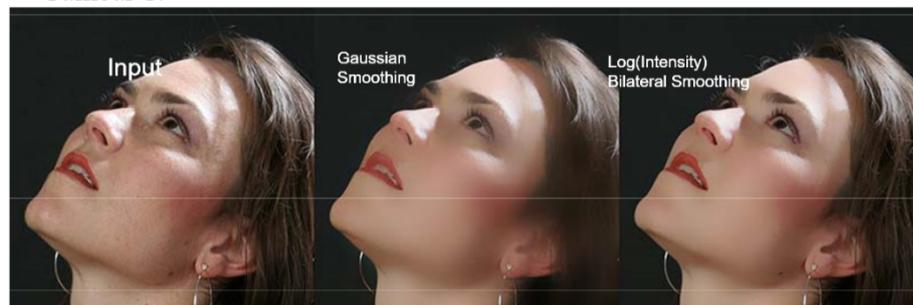
Deteksi tepi canny adalah suatu operator dalam deteksi tepi, canny ditemukan oleh John Canny pada tahun 1986. *Gaussian derivative kernel* dipakai canny untuk menghilangkan *noise* dan memperhalus citra. Salah satu tujuan dari deteksi tepi agar sebuah gambar dapat disederhanakan dan dikenali, melalui pendeteksian garis-garis yang terdapat dalam sebuah objek yang ada pada gambar [8]. Canny cukup terkenal karena termasuk dalam salah satu operator pendeteksian tepi yang baik. Ada beberapa tahapan dalam deteksi tepi canny: penghapusan *noise*, menghitung potensi gradien citra, perhitungan arah tepi, menghubungkan arah tepi, *non-maximal suppression*, *hysteresis threshoding*. Pada tahapan *filtering* merupakan tahapan yang paling penting dalam semua tahapan diatas [9].

Berikut adalah penjabaran dari beberapa tahapan diatas :

1. Penghapusan *Noise*

Noise akan dihapus dengan menggunakan Bilateral Filter, sehingga gambar pun akan menjadi lebih halus.

- Bilateral filter dapat mengembalikan kembali nilai piksel dengan rata-rata yang sama dengan nilai intensitas piksel yang sudah ada. *Gray level* akan dikombinasikan dengan fotometrik, piksel yang cerah akan diganti sedangkan piksel yang gelap akan diabaikan. Kemudian bila filter ada pada posisi piksel yang gelap maka piksel yang gelap diabaikan [8]. Proses ini dapat dilihat pada Gambar 1.



Gambar 1. Perbandingan Gaussian Dan Bilateral Filter

2. Menghitung Gradien Intensitas Citra

Setiap gradien piksel pada gambar yang telah diperhalus, akan di hitung potensi gradien citra menggunakan operator sobel. Magnitude gradien dihitung dengan matriks pada Persamaan 1.

$$G_x = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \quad G_y = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

Persamaan 1. Kernel (Gx) dan (Gy) dari Operator Sobel

Kemudian akan mencari gradien tepi dengan menggunakan rumus seperti pada Persamaan 2.

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

Persamaan 2. Menghitung Gradien Tepi

2. Perhitungan Arah Tepi

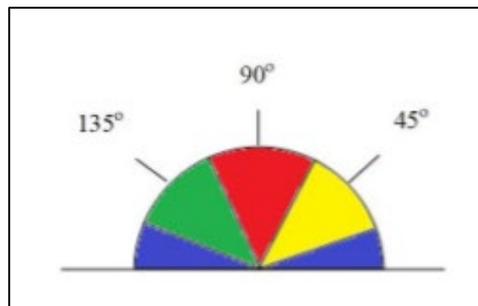
Tepian sebenarnya akan dihitung arah tepianya dengan persamaan 3 yaitu :

$$Angle (\theta) = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

Persamaan 3. Menghitung Arah Tepi

3. Menghubungkan Arah Tepi

Setelah mendapat arah tepi, kemudian akan dimasukkan kedalam kategori berdasarkan area yang terdapat pada Gambar 2.



Gambar 2. Pembagian Warna Berdasarkan Arah Tepian Canny

- Seluruh sudut tepi dengan kisaran antara 0 dan 22,5 serta 157,5 dan 180 derajat. Merupakan warna biru, akan diubah ke 0 derajat.
- Seluruh sudut tepi dengan kisaran antara 22,5 dan 67,5 derajat. Merupakan warna kuning, akan diubah ke 45 derajat.
- Seluruh sudut tepi dengan kisaran antara 67,5 dan 112,5 derajat. Merupakan warna merah, akan diubah ke 90 derajat.
- Seluruh sudut tepi dengan kisaran antara 112,5 dan 157,5 derajat. Merupakan warna hijau, akan diubah ke 135 derajat.

4. *Non-maximal Suppression*

Kemudian akan dilakukan penghilangan non-maksimum, proses ini akan dilakukan disepanjang tepian dan akan menghapus piksel-piksel yang tidak dirasa (anggap)

sebagai tepi. Seluruh piksel yang dianggap bukan tepi akan diubah pikselnya menjadi 0, sehingga akhirnya diperoleh tepi yang tipis.

5. Hysteresis Thresholding

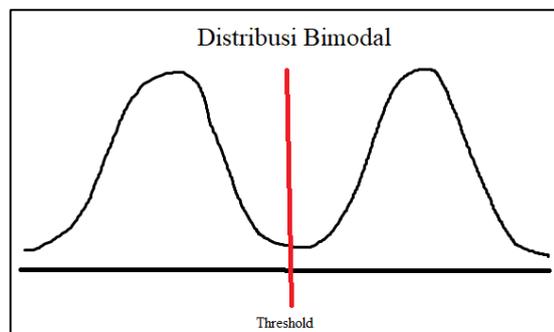
Larik (*array*) *Grad* yang berisi besaran dari gradien akan disalin dan dipindahkan ke *non-max*. Kemudian non-maksimum akan dihilangkan dengan melihat kedua titik tetangga yang terdapat pada arah tepi yang tersimpan. Sebuah contoh bila arah tepi sama dengan 0, dan titik yang diperhatikan adalah (r,c) , maka kedua tetangganya ialah $(r,c-1)$ dan $(r, c+1)$. Jika gradien titik yang diperhatikan lebih besar dari kedua gradien titik tetangganya maka nilainya tidak akan berubah. Tetapi bila nilai titik yang diperhatikan lebih kecil dari kedua gradien titik tetangganya, maka nilainya akan diubah ke 0. Pada prosesnya histeresis mempunyai dua ambang yaitu T1 yaitu ambang bawah dan T2 yaitu ambang atas. Untuk setiap nilai yang kurang dari T1 akan diubah ke 0 (hitam) dan nilai yang lebih dari T2 akan diubah ke 255 (putih). Sementara jika nilai tersebut kurang dari T2 dan lebih atau sama dengan T1 maka akan diubah ke 128, dimana dapat dikatakan akan diubah terlebih dahulu ke warna abu-abu [9].

2.4 Thresholding

Thresholding menjadi salah satu teknik yang dipakai dalam segmentasi citra/gambar. Pada dasarnya *thresholding* akan memilih nilai ambang batas dengan tingkat keabuan yang optimal untuk dapat memisahkan *region of interest* (objek yang menjadi perhatian) dengan latar belakang dari sebuah gambar, tingkat abu-abu dari sebuah histogram pada sebuah gambar merupakan alat yang efisien dalam pengembangan algoritma *thresholding*. Keluaran dari proses *thresholding* ini akan adalah berupa gambar biner, dikarenakan pada prosesnya *region* atau wilayah citra yang gelap akan dibuat semakin gelap dengan nilai 0 atau sama saja dengan warna hitam. Begitu pula sebaliknya, bila wilayah citra terang maka akan dibuat semakin terang dengan nilai 1 atau sama saja dengan warna putih [7].

Nilai ambang batas dari sebuah gambar *Grayscale* (gambar abu-abu) didapatkan dengan menggunakan metode otsu. Metode otsu *thresholding* dapat melakukan segmentasi pada area objek dengan cukup akurat menggunakan histogram *grayscale* [18]. Nobuyuki Otsu adalah orang yang memperkenalkan metode otsu *thresholding* pada tahun 1979. Langkah pertama adalah membuat histogram. Dari histogram kita dapat mengetahui jumlah piksel untuk setiap tingkat abu-abu, i sampai L adalah tingkat keabuan gambar. Level i dimulai dengan 1, yakni pixel 0. Sedangkan maksimum level dari L adalah 256 bernilai 255 pada pikselnya [10].

Gambar 3 menunjukkan bahwa sebuah gambar memiliki histogram bimodal (bimodal adalah gambar yang histogramnya memiliki dua puncak), kedua puncak adalah latar belakang dan depan dari sebuah gambar. *Thresholding* akan menempatkan nilai ambang (*threshold*) diantara kedua puncak.



Gambar 3. Distribusi Bimodal

3. Metodologi Penelitian

Penelitian ini dimulai dengan dilakukannya perancangan penelitian yang akan dilakukan. Penulis harus paham terlebih dahulu kasus yang akan di bahas dan diselesaikan. Untuk melakukan proses dan mengetahui hasilnya tentu penulis harus mengerti apa saja yang dibutuhkan untuk melakukan penelitian tersebut. Penulis mengambil objek plat kendaraan yang akan dijadikan objek permasalahan. Permasalahan yang harus di selesaikan dalam penelitian ini adalah melakukan segmentasi plat kendaraan menggunakan metode. Penulis memilih untuk menggunakan metode deteksi tepi canny dan tresholding. Sehingga penulis dapat memberikan gambaran proses penelitian sampai pada hasilnya. Penulis juga membutuhkan model untuk melihat penerapan metode yang digunakan dan melihat alur kerja algoritma pemrosesan.

Proses penerapan metode yang dilakukan pada penelitian ini akan menghasilkan perbedaan antara dua metode yang digunakan oleh penulis. Semua metode akan memiliki kekurangan dan kelebihan masing-masing. Maka dari itu penulis akan memaparkan apa kekurangan dan kelebihan yang di dapat jika menggunakan kedua metode yang penulis gunakan.

4. Hasil dan Diskusi

4.1 Implementasi Metode Deteksi Tepi Canny

Implementasi merupakan sebuah proses pelaksanaan atau penerapan dari rencana (analisis) yang sudah dilakukan sebelumnya untuk kemudian diterapkan dengan rinci dan matang. Akan dilakukan implementasi model yang akan digunakan untuk melakukan segmentasi plat nomor kendaraan dengan menggunakan metode Deteksi Tepi Canny dan *Thresholding*. Implementasi kode program untuk Deteksi Tepi Canny dapat dilihat pada Gambar 4. Implementasi pada masing-masing model memiliki tingkat akurasi dan ketepatan yang hampir sama, namun tetap memiliki perbedaan diantara kedua model ini, sehingga dengan demikian kedua model dapat dibandingkan untuk kemudian menghasilkan pertimbangan untuk menentukan model yang lebih baik.

```
1 import numpy as np
2 import cv2
3 import imutils
4
5 # Baca file gambar
6 image = cv2.imread('20.jpg')
7
8 # Ubah Lebar menjadi 500
9 image = imutils.resize(image, width=500)
10
11 # Tampilkan gambar asli
12 cv2.imshow("[1] Original Image", image)
13
14 # Konversi RGB ke Grayscale
15 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
16 cv2.imshow("[2] Grayscale Conversion", gray)
17
18 # Penghapusan noise dengan iterative bilateral filter
19 gray = cv2.bilateralFilter(gray, 11, 17, 17)
20 cv2.imshow("[3] Bilateral Filter", gray)
21
22 # Temukan Tepi dari gambar grayscale
23 edged = cv2.Canny(gray, 170, 200)
24 cv2.imshow("[4] Canny Edges", edged)
25
26 # Temukan kontur berdasarkan Tepian
27 (cnts, _) = cv2.findContours(edged.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
28 cnts=sorted(cnts, key = cv2.contourArea, reverse = True)[:30] #sortir kontur berdas
29 NumberPlateCnt = None
30
```

Gambar 4. Code Deteksi Tepi Canny

Pada Gambar 4, terdapat beberapa fungsi yaitu ada `imread` yang digunakan untuk membaca file gambar, lalu ada `imshow` yang digunakan untuk menampilkan gambar, lalu gambar berwarna akan diubah menjadi gambar abu-abu dengan menggunakan fungsi `cv2.cvtColor`, setelah itu gambar akan dihaluskan dengan menggunakan fungsi `cv2.bilateralFilter`, kemudian deteksi tepi Canny akan diterapkan dengan menggunakan fungsi `cv2.Canny`, kontur dari gambarpun akan dibaca dengan menggunakan fungsi `findContours`.

4.2 Implementasi Metode *Thresholding*

```

1  import numpy as np
2  import cv2
3  import imutils
4
5  # Baca file gambar
6  image = cv2.imread('7.jpg')
7
8  # Tampilkan gambar asli
9  cv2.imshow("[1] Original Image", image)
10
11 # Konversi RGB ke Grayscale
12 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
13 cv2.imshow("[2] Grayscale Conversion", gray)
14
15 # Penghapusan noise dengan iterative bilateral filter
16 gray = cv2.bilateralFilter(gray, 11, 17, 17)
17 #gray = cv2.GaussianBlur(gray,(5,5),0)
18 cv2.imshow("[3] Bilateral Filter", gray)
19
20 # Menerapkan Otsu thresholding
21 ret, thresh1 = cv2.threshold(gray, 120, 255, cv2.THRESH_BINARY +cv2.THRESH_OTSU)
22 cv2.imshow("[4] Otsu Threshold", thresh1)
23
24 # Temukan kontur berdasarkan Tepian
25 (cnts,_) = cv2.findContours(thresh1.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
26 cnts=sorted(cnts, key = cv2.contourArea, reverse = True)[:30] #sortir kontur berdasar
27 NumberPlateCnt = None

```

Gambar 5. Code *Thresholding*

Pada tahap implementasi pada Gambar 5, terdapat beberapa fungsi yaitu ada `imread` yang digunakan untuk membaca file gambar, lalu ada `imshow` yang digunakan untuk menampilkan gambar, lalu gambar berwarna akan diubah menjadi gambar abu-abu dengan menggunakan fungsi `cv2.cvtColor`, setelah itu gambar akan dihaluskan dengan menggunakan fungsi `cv2.bilateralFilter`, kemudian *thresholding* akan diterapkan dengan menggunakan fungsi `cv2.threshold`, kontur dari gambarpun akan dibaca dengan menggunakan fungsi `findContours`.

4.3 Hasil Pengujian

Hasil dari pengujian ini adalah untuk melakukan segmentasi atau memotong objek plat nomor kendaraan dari sebuah gambar, dengan sebelumnya dilakukan *preprocessing* untuk *data cleaning*, agar data siap dipakai pada tahap selanjutnya dengan menggunakan kedua metode yaitu deteksi tepi canny dan *thresholding*. Pada Tabel 1 dan Tabel 2 menunjukkan hasil pengujian menggunakan kedua metode yaitu deteksi tepi canny dan *thresholding*.

Tabel 1. Hasil Pengujian Menggunakan Metode Deteksi Tepi Canny

Nama File	Foto Asli	Grayscale	Bilateral Filter	Canny	Hasil Deteksi	Potong Plat
1.jpg						

Tabel 2. Hasil Pengujian Menggunakan Metode Thresholding.

Nama File	Foto Asli	Grayscale	Bilateral Filter	Thresholding	Hasil Deteksi	Potong Plat
1.jpg						

Dapat dilihat dari Tabel 1 dan Tabel 2 bahwa plat nomor kendaraan dapat tersegmentasi dengan baik dengan menggunakan metode deteksi tepi canny dan *thresholding*.

4.4 Analisis Hasil Pengujian

Dari hasil pengujian yang telah dilakukan pada penelitian ini dapat diketahui bahwa objek plat nomor kendaraan yang diuji tidak boleh rusak, karena sangat mempengaruhi hasil akhir dari segmentasi dan deteksi yang dilakukan pada pengujian ini. Pengujian metode *Thresholding* menggunakan sebuah ambang batas yang digunakan untuk memisahkan latar belakang dan latar depan dari sebuah gambar, setiap piksel akan dilihat nilainya, jika nilai piksel lebih kecil dari ambang maka diganti dengan 0, namun jika nilai piksel lebih besar sama dengan ambang maka diganti dengan 255. Sehingga setelah diuji *thresholding* cukup baik dalam memisahkan latar depan dan belakang gambar yang diuji dalam penelitian ini.

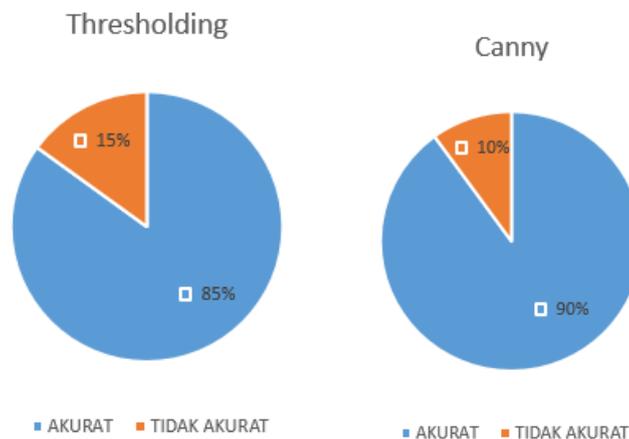
Pengujian menggunakan metode deteksi tepi canny lebih baik dibanding menggunakan metode *thresholding*, karena canny memiliki banyak langkah yang membuat pengujian gambar menjadi lebih detail. Di dalam algoritma Canny juga untuk menemukan tepi menggunakan sobel operator, sobel menemukan gradien dalam arah horisontal (Gx) dan vertikal (Gy). Kedua arah ini adalah matriks atau kernel berukuran 3x3 yang akan digunakan untuk mengkonvolusikan matriks yang menjadi inputan. Canny dapat melakukan pembagian warna berdasarkan arah tepian, seluruh sudut tepi dengan kisaran antara 0 dan 22,5 serta 157,5 dan 180 derajat, merupakan warna biru, akan diubah ke 0 derajat. Seluruh sudut tepi dengan kisaran antara 22,5 dan 67,5 derajat, merupakan warna kuning, akan diubah ke 45 derajat. Seluruh sudut tepi dengan kisaran antara 67,5 dan 112,5 derajat, merupakan warna merah, akan diubah ke 90 derajat. Seluruh sudut tepi dengan kisaran antara 112,5 dan 157,5 derajat, merupakan warna hijau, akan diubah ke 135 derajat.

Setelah pembulatan arah tepi, akan dibandingkan nilai setiap piksel terhadap dua piksel tetangga dalam arah gradien. Jika piksel tersebut merupakan piksel yang memiliki nilai maksimum lokal maka akan dipertahankan sebagai piksel tepi, namun bila tidak maka akan diganti nilai pikselnya dengan nilai 0. Lalu langkah selanjutnya terakhir adalah membuat dua ambang batas T1 dan T2, setiap tepi dengan intensitas lebih besar dari T1 maka adalah tepi yang pasti, setiap tepi dengan intensitas kurang dari T2 maka bukan merupakan sebuah tepi, dan

setiap tepi dengan iutensitas yang berada diantara T1 dan T2 serta saling terhubung, maka dapat diklasifikasikan sebagai sebuah tepi. Pada hasil pengujian yang telah dilakukan kepada kedua metode dengan menguji 100 data, maka dapat diperoleh nilai akurasi dengan tingkat keakuratan yang tepat berjumlah 90 untuk metode deteksi tepi canny dan 85 untuk metode *thresholding*. Sedangkan hasil pengujian yang tidak akurat berjumlah 10 untuk metode deteksi tepi canny dan 15 untuk metode *thresholding*. Dengan presentase yang akurat adalah 90% untuk metode deteksi tepi canny dan 85% untuk metode *thresholding*. Hasil akurasi dapat dilihat pada Tabel 3 dan Gambar 6.

Tabel 3. Hasil Akurasi Data Segmentasi Plat Nomor Kendaraan Menggunakan Metode Canny dan Thresholding

Metode	Akurat	Tidak Akurat	Total
Canny	90	10	100
Thresholding	85	15	100



Gambar 6. Diagram Presentase Hasil Akurasi Data Segmentasi Plat Nomor Kendaraan Menggunakan Metode Canny dan Thresholding

5. Kesimpulan

Dari hasil penelitian dan pengujian data citra plat nomor kendaraan, untuk dapat mendeteksi objek plat nomor kendaraan yang terdapat pada sebuah citra. Plat nomor kendaraan dapat terbaca dengan baik, dengan membaca garis putih yang terdapat pada tepian plat nomor kendaraan. Didapatkan juga hasil yang baik ketika melakukan pengujian terhadap plat nomor kendaraan menggunakan kedua metode didapatkan bahwa model dapat melakukan segmentasi secara baik, dengan presentase akurat sebesar 90% untuk metode deteksi tepi canny dan 85% untuk metode *thresholding*. Dari hasil penelitian yang sudah didapat maka metode deteksi tepi canny unggul untuk melakukan segmentasi terhadap plat nomor kendaraan, dengan jumlah data yang diuji adalah 100 data.

Referensi

- [1] J. M. Sourav Roy, Amitava Choudhury, "An approach towards detection of Indian number plate from vehicle," *Int. J. Innov. Technol. Explor. Eng.*, vol. 2, no. 4, pp. 2278–3075, Mar. 2013.

- [2] Y. T. Rada, "Pengenalan pola pada fisik mobil menggunakan persamaan diferensial deteksi tepi (edge detection)," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 7, no. 1, pp. 57, Mar. 2018.
- [3] S. Asht and R. Dass, "Pattern recognition techniques : A Review," *Int. J. Comput. Sci. Telecommun.*, vol. 3, no. 8, pp. 25–29, Aug. 2012.
- [4] N. Syafitri, "Pengenalan Pola Untuk Deteksi Uang Koin," *SNTIKI* pp. 18–24, 2011.
- [5] M. K. Vairalkar and S. U. Nimbhorkar, "Edge detection of images using sobel operator," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 1, pp. 291–293, Jan. 2012.
- [6] J. W. Yodha and A. W. Kurniawan, "Perbandingan penggunaan deteksi tepi dengan metode laplace , sobel dan prewit dan canny pada pengenalan pola," *Techno.COM*, vol. 13, no. 3, pp. 189–197, Aug. 2014.
- [7] S. R. Halwa, J. Wójtowicz, J. Szuman, B. Pawlak, W. Adamczyk, and Z. Lorkiewicz, "A review on otsu image segmentation algorithm," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 25, no. 5–6, pp. 403–408, Jun. 1982.
- [8] A. Essra, "Analisis deteksi tepi canny pada citra dengan gaussian filtering dan bilateral," *Isd*, vol. 2, no. 1, pp. 34–39, Jan. 2017.
- [9] B. Derviş, "Pengolahan citra digital menggunakan metode canny dan euclidean distance untuk mengukur tingkat kemiripan sketsa wajah," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.
- [10] N. Otsu, "Threshold Selection Method From Gray-Level Histograms.," *IEEE Trans Syst Man Cybern*, vol. SMC-9, no. 1, pp. 62–66, 1979.