Terbit online pada laman web jurnal: **http://journal.isas.or.id/index.php/JASENS**

JOURNAL OF APPLIED SMART ELECTRICAL NETWORK AND SYSTEMS (JASENS)

Vol. 6 No. 2 (2025) 69 - 75 ISSN Media Elektronik: 2723-5467

Simulasi Deteksi Marka Jalan Menggunakan Canny *Edge Detection* untuk Navigasi Kendaraan Otonom

Febrian Akbara[1], Ii Munadhif[2], Mohammad Abu Jami'in[3], Ryan Yudha Adhitya[4], Imam Sutrisno[5]
[1,2,3,4,5]Jurusan Teknik Kelistrikan Kapal, Program Studi Teknik Otomasi, Politeknik Perkapalan Negeri Surabaya
[1]febrianakbara@student.ppnc.ac.id, [2]iimunadhif@ppns.ac.id, [3]jammy@ppns.ac.id, [4]ryanyudhaadhitya@ppns.ac.id, [5]imam_sutrisno@ppns.ac.id

Abstract

This study develops an automatic *steering* control system based on image processing using the Canny *Edge Detection* method. The system is implemented on a small-scale autonomous vehicle prototype, utilizing Raspberry Pi 5 as the main processor and a Pi Camera as the visual sensor. Video *frame*s are processed through several stages, including color conversion, grayscale, Gaussian blur, *Edge Detection*, *Region of Interest* (RoI), and lane center estimation. The *steering* correction is determined based on the *error* between the lane center and the *frame* center. The system was tested using six videos representing various environmental conditions, including sunny, cloudy, rainy, and late-afternoon scenarios. The results show that the system achieved an average lane detection accuracy of 96%, with responsive and stable *steering* control, even when only one lane line was detected. The lane estimation mechanism proved effective in maintaining the vehicle's direction. This system demonstrates potential as an initial solution for lightweight autonomous vehicle development, although further improvement is needed for extreme lighting conditions and *real-time* processing.

Keywords: Autonomous Vehicle, Canny *Edge Detection*, Lane Detection, *Steering* Control, Raspberry Pi

Abstrak

Penelitian ini mengembangkan sistem kendali kemudi otomatis berbasis pengolahan citra menggunakan metode Canny *Edge Detection*. Sistem dirancang pada kendaraan otonom skala prototipe, dengan Raspberry Pi 5 sebagai pemroses utama dan Pi Camera sebagai sensor visual. Citra video diproses melalui beberapa tahap, termasuk konversi warna, grayscale, gaussian blur, deteksi tepi, *Region of Interest* (RoI), dan estimasi garis tengah jalur. Nilai *error* antara posisi garis tengah dan pusat *frame* kamera digunakan untuk menentukan koreksi arah kemudi kendaraan. Pengujian dilakukan pada enam video uji dengan kondisi lingkungan yang bervariasi, mencakup cuaca cerah, berawan, hujan, dan sore hari. Hasil menunjukkan sistem mampu mendeteksi marka jalan dengan akurasi rata-rata mencapai 96%, serta menghasilkan kendali kemudi yang responsif dan stabil, bahkan saat hanya satu sisi marka terdeteksi. Mekanisme estimasi posisi jalur terbukti efektif dalam menjaga arah kendaraan pada jalur yang benar. Sistem ini menunjukkan potensi sebagai solusi awal dalam pengembangan kendaraan otonom ringan, meskipun masih terdapat tantangan pada kondisi pencahayaan ekstrem dan kebutuhan peningkatan menuju sistem *real-time*.

Kata kunci: Otonom, Canny *Edge Detection*, Deteksi Marka Jalan, Pengendalian Kemudi, Raspberry Pi

Diterima Redaksi : 19-05-2025 | Selesai Revisi : 08-07-2025 | Diterbitkan Online : 31-12-2025

1. Pendahuluan

Kendaraan otonom merupakan salah satu terobosan teknologi otomotif yang terus berkembang dengan sangat pesat. Teknologi ini diprediksi akan mendominasi pasar transportasi komersial dalam waktu sekitar lima belas tahun ke depan[1-8]. Kendaraan otonom diklasifikasikan ke dalam enam tingkat otomatisasi, mulai dari level 0 (tanpa otomatisasi) hingga level 5 (otomatisasi penuh), yang masing-masing memiliki karakteristik dan tingkat kecanggihan sistem yang berbeda. Berbagai fitur dalam sistem *Advanced Driver Assistance Systems* (ADAS), seperti *Lane Keeping Assist* (LKA), telah dikembangkan untuk meningkatkan keselamatan dan kenyamanan saat berkendara[9-13].

Salah satu tantangan utama dalam pengembangan kendaraan otonom adalah bagaimana merancang sistem deteksi jalur yang handal dan presisi, serta pengendalian kemudi yang efektif berdasarkan data visual. Pengolahan citra digital menjadi salah satu solusi utama, dengan menggunakan berbagai teknik seperti transformasi tampilan *bird's eye view*, deteksi fitur tepi atau kontur, dan estimasi bentuk jalur. Dalam konteks ini, metode Canny *Edge Detection* sering

dipilih karena kemampuannya dalam menghasilkan deteksi tepi yang akurat dan detail, sehingga dapat membantu sistem dalam mengenali marka jalan secara lebih baik [13-16].

Penelitian-penelitian sebelumnya telah menunjukkan efektivitas penggunaan metode Canny *Edge Detection* dalam sistem kendaraan pintar dan aplikasi pengawasan lalu lintas. Namun, masih terdapat ruang untuk peningkatan, khususnya dalam hal responsivitas sistem dan keandalannya dalam berbagai kondisi jalan dan pencahayaan. Oleh karena itu, penelitian ini dilakukan dengan tujuan mengembangkan sistem deteksi dan pelacakan marka jalan menggunakan metode Canny *Edge Detection* yang dapat secara otomatis mendeteksi dan merespons arah pergerakan kendaraan secara *real-time*[17-19].

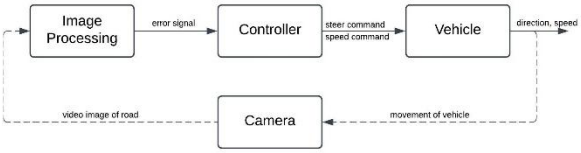
Dengan adanya pengembangan ini, diharapkan sistem navigasi kendaraan otonom menjadi lebih akurat dan handal dalam membaca kondisi marka jalan, sehingga meningkatkan keselamatan dan efisiensi kendaraan selama perjalanan. Penelitian ini juga diharapkan dapat memberikan kontribusi inovatif bagi perkembangan teknologi kendaraan pintar di masa depan[20-21].

2. Metode Penelitian

Penelitian ini dirancang sebagai bagian dari pengembangan sistem Kendaraan Otonom skala prototipe dengan pendekatan berbasis pengolahan citra digital. Metode yang digunakan menitikberatkan pada penerapan algoritma Canny *Edge Detection* untuk mendeteksi marka jalan, yang selanjutnya digunakan sebagai acuan dalam sistem kendali arah kendaraan secara otomatis. Sistem diuji dengan menggunakan video hasil perekaman kendaraan yang melintasi lintasan yang terdapat marka jalan berwarna putih/kuning. Kamera yang digunakan adalah Pi Camera, dipasang pada bagian depan kendaraan untuk mensimulasikan sudut pandang pengemudi. Video tersebut diproses secara *frame-by-frame* di Raspberry Pi 5 untuk mensimulasikan *real-time inference*, meskipun pemrosesan dilakukan secara *offline*.

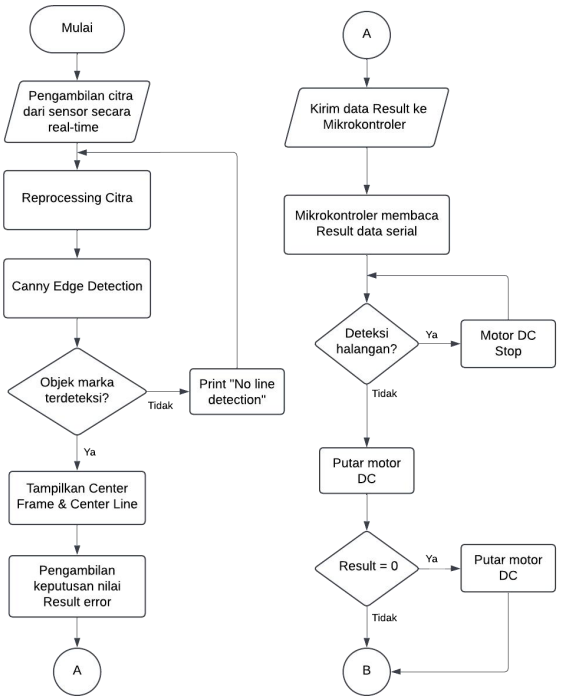
Secara umum, metode penelitian ini melibatkan beberapa tahapan, yaitu perancangan sistem perangkat keras dan lunak, pemrosesan data citra, pengembangan algoritma deteksi dan estimasi jalur, serta pengujian performa sistem. Pengembangan perangkat lunak dilakukan menggunakan bahasa Python pada Raspberry Pi dengan editor Thonny dan pustaka OpenCV, sedangkan pemrograman mikrokontroler dilakukan melalui Arduino IDE. Seluruh komponen bekerja secara sinkron untuk mendeteksi jalur dan mengoreksi arah laju kendaraan berdasarkan informasi visual secara otomatis secara simulasi. [4-6].

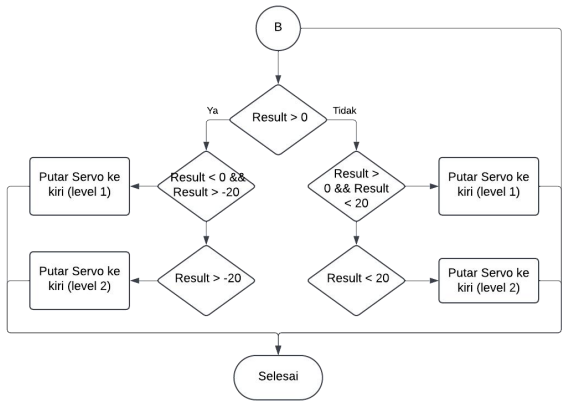
2.1. Perancangan Sistem



Gambar 1. Blok Diagram Sistem

Pada Gambar 1 menunjukkan blok diagram keseluruhan dari sistem yang dikembangkan. Sistem ini terdiri dari beberapa unit utama yang saling terintegrasi. Unit pemroses citra menggunakan Raspberry Pi 5 sebagai image processing yang menjalankan algoritma pengolahan citra dengan Canny *Edge Detection*, sementara Pi Camera berperan sebagai sensor visual untuk mendeteksi marka jalan secara *real-time*. Unit kendali kemudi (*steering controller unit*) dikendalikan oleh mikrokontroler, yang menerima perintah dari Raspberry Pi untuk mengatur arah laju kendaraan melalui aktuator kemudi.





Gambar 2. *Flowchart* Sistem

Alur proses sistem ditunjukkan dalam *Flowchart* pada Gambar 2. Input berupa kamera yang menangkap citra marka jalan. Citra yang diperoleh dari kamera akan diproses melalui tahapan *pre-processing*, yaitu tahap awal pengolahan citra digital yang bertujuan untuk meningkatkan kualitas citra agar lebih mudah dikenali oleh sistem deteksi tepi. Tahapan ini mencakup pengaturan resolusi dan ukuran citra agar sesuai dengan kebutuhan sistem deteksi. Konversi citra dari RGB ke Grayscale untuk menyederhanakan data warna menjadi hanya tingkat keabuan, sehingga proses deteksi tepi lebih efisien. Penerapan metode thresholding untuk memisahkan objek utama (garis marka) dari latar belakang. Setelah thresholding, dilakukan proses Canny *Edge Detection* untuk mendeteksi garis tepi secara detail. Metode ini efektif dalam menangkap tepi garis marka jalan yang menjadi referensi arah kendaraan.

Setelah tahap pre-processing dengan Canny *Edge Detection* selesai, citra kemudian dipotong menggunakan metode *Region of Interest* (RoI) yang berfokus pada area bawah gambar di mana marka jalan biasanya terlihat. Hal ini bertujuan untuk mengurangi *noise* dan mempercepat waktu pemrosesan. Selanjutnya, hasil dari proses deteksi tepi akan dianalisis untuk mengidentifikasi posisi center line dari marka jalan. Sistem kemudian akan menghitung nilai *error* antara posisi center line dan center *frame* (posisi tengah dari tampilan kamera). Nilai *error* ini digunakan untuk menentukan arah koreksi kemudi yang diperlukan.

Data hasil pengolahan citra kemudian dapat dikirimkan ke mikrokontroler yang bertugas sebagai pengendali aktuator kendaraan. Mikrokontroler ini akan mengatur arah kemudi menggunakan motor servo dan kecepatan kendaraan dengan mengatur motor DC, berdasarkan nilai *error* yang diterima.

2.2. Proses Algoritma Pengolahan Citra

Sistem ini menggunakan algoritma pengolahan citra digital untuk mendeteksi garis marka jalan dan mengontrol arah gerak kendaraan secara otomatis. Citra digital merupakan gambar dua dimensi yang dapat ditampilkan di layar komputer, terdiri dari sejumlah nilai digital diskrit yang dikenal sebagai pixel. Proses dimulai dari pengambilan input citra diperoleh dari video rekaman pergerakan kendaraan pada lintasan uji yang terdapat marka jalan. Video tersebut digunakan sebagai simulasi dari kondisi *real-time*, meskipun pemrosesan dilakukan secara *offline*, sistem dapat berjalan dan diuji. Penggunaan video memungkinkan pengujian efektivitas algoritma Canny *Edge Detection* dalam kondisi nyata dengan berbagai variasi pencahayaan, cuaca, dan kondisi fisik marka jalan.

1. Filter Warna
   Citra dikonversi dari BGR ke ruang warna HLS dan difilter untuk mendeteksi warna putih dan kuning yang umum digunakan sebagai marka jalan dengan batas bawah dan atas tertentu yang sesuai dengan persamaan (1).
   $$I_{HLS} = \text{convert}(I_{RGB}) \quad (1)$$
   Sebelum menentukan ambang batas, terlebih dahulu dilakukan konversi citra ke ruang warna HLS agar lebih mudah dalam membedakan warna tertentu, mengacu pada persamaan (2).
   $$M = \begin{cases} 1, & I_{HLS}(x,y) \in \text{rentang warna} \\ 0, & \text{lainnya} \end{cases} \quad (2)$$
2. Grayscale dan Gaussian blur
   Citra hasil filter diubah ke grayscale agar lebih sederhana secara data, lalu diterapkan Gaussian blur untuk mengurangi *noise* sebelum proses deteksi tepi.
   $$I_{grayscale}(x,y) = 0.299R + 0.587G + 0.114B \quad (3)$$
   Persamaan (3) digunakan untuk mengubah citra berwarna menjadi citra keabuan berdasarkan luminansi manusia terhadap warna RGB. Penilaian ini digunakan untuk menentukan ambang batas intensitas [7].
   $$I_{blur}(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} G(i,j) \cdot I_{gray}(x+i, y+j) \quad (4)$$
   Persamaan (4) digunakan untuk menghaluskan citra menggunakan konvolusi kernel Gaussian guna mereduksi *noise*.
3. Deteksi Tepi (Canny *Edge Detection*)
   Deteksi tepi (*Edge Detection*) dalam suatu citra adalah proses yang bertujuan untuk mengidentifikasi tepi-tepi dari objek-objek dalam citra tersebut, sehingga dapat menandai bagian-bagian yang memberikan detail penting pada citra [8]. Citra diolah dengan algoritma Canny untuk mengekstrak tepi garis marka berdasarkan perubahan intensitas.
   $$G = \sqrt{G_x^2 + G_y^2} \quad (5)$$
   $$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (6)$$
   Persamaan (5) merupakan deteksi gradient yang digunakan untuk menghitung besarnya gradien

(G) sedangkan persamaan (6) untuk arah tepi (θ) pada setiap piksel.

4. *Region of Interest* (ROI)
   Deteksi tepi dibatasi hanya pada area tertentu (bagian bawah citra) menggunakan masker poligonal yang merupakan segementasi citra untuk memisahkan atau menghilangkan latar belakang, sehingga sistem hanya fokus pada area jalan di depan kendaraan [20].
   $$I_{ROI}(x,y) = \begin{cases} I_{edge}(x,y), & \text{jika } (x,y) \in \text{area ROI} \\ 0, & \text{lainnya} \end{cases} \quad (7)$$
   Persamaan (7) digunakan untuk menyaring hasil deteksi tepi agar hanya mencakup bagian jalan yang relevan.
5. Hough Transform
   Digunakan untuk mendeteksi garis-garis lurus pada hasil Canny. Garis-garis ini dikelompokkan menjadi dua: garis kiri (kemiringan negatif) dan kanan (kemiringan positif) berdasarkan parameter ρ dan θ [10], yang mengacu pada persamaan (8).
   $$\rho = x\cos\theta + y\sin\theta \quad (8)$$
6. Rata-rata dan Stabilisasi Garis
   Posisi garis kiri dan kanan dihitung rata-rata dan distabilisasi dengan parameter alpha untuk menghindari fluktuasi mendadak akibat *noise* atau hilangnya garis sesaat.
   $$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (9)$$
   Persamaan (9) digunakan untuk menghitung kemiringan garis dari dua titik akhir garis deteksi.
   $$x_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (10)$$
   Persamaan (10) digunakan untuk menentukan rata-rata posisi garis kiri atau kanan yang terdeteksi
7. Estimasi Posisi Jalur Tengah
   Jika kedua garis terdeteksi, titik tengah di antara keduanya digunakan untuk menentukan jalur ideal. Bila hanya satu garis terdeteksi, sistem melakukan estimasi posisi jalur tengah berdasarkan arah garis dan asumsi lebar jalur.
   $$x_{\text{mid}} = \frac{x_{\text{left}} + x_{\text{right}}}{2} \quad (11)$$
   Persamaan (11) digunakan untuk menentukan estimasi jika mendeteksi adanya dua garis marka.
   $$x_{\text{mid}} = x_{\text{detected}} + \mathit{Offset} \quad (12)$$
   Persamaan (12) digunakan untuk menentukan estimasi jika mendeteksi hanya ada satu garis marka. *Offset* diasumsikan berdasarkan lebar jalur jika hanya satu sisi garis terlihat.

2.3. Perhitungan *Error* Kemudi Sistem

Error kemudi adalah selisih antara posisi garis tengah jalur yang dideteksi (berdasarkan posisi garis kiri dan kanan) dengan posisi tengah dari *frame* kamera. Nilai *error* yang dihasilkan dari selisih antara xmid dan xcenter kemudian dikonversi menjadi nilai sudut kemudi. Nilai *error* ini menunjukkan seberapa jauh kendaraan menyimpang dari jalur idealnya. Sehingga, digunakan persamaan (13).

$$\mathit{Error} = x_{\text{mid}} - x_{\mathit{center}} \quad (13)$$

Dengan x_{mid} adalah posisi jalur tengah, sedangkan x_{center} merupakan posisi tengah *frame* kamera.

$$\theta = \tan^{-1}\left(\frac{x_2 - x_1}{y_2 - y_1}\right) \cdot \left(\frac{180}{\pi}\right) \quad (14)$$

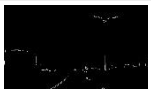
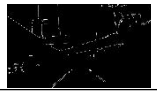
Persamaan (14) digunakan untuk memperkirakan arah kemudi dalam satuan derajat berdasarkan arah kemiringan garis.

3. Hasil dan Pembahasan

3.1. Hasil Deteksi Marka

Pada tahap ini, pengujian deteksi marka dilakukan menggunakan enam video lintasan uji yang direkam dalam kondisi lingkungan berbeda. Tujuan utama pengujian adalah untuk mengevaluasi efektivitas algoritma Canny *Edge Detection* dalam mengekstrak tepi marka jalan pada situasi pencahayaan dan cuaca yang beragam. Secara keseluruhan, sistem diuji terhadap total 16.608 *frame*, dengan rata-rata 2700+ *frame* per video. Proses dilakukan secara *frame*-by-*frame* menggunakan Raspberry Pi 5 sebagai pemroses utama.

Tabel 1.Hasil Pengujian Deteksi Tepi Canny

Citra Asli	Citra Canny	Tepi	RoI
			
			
			
			
			

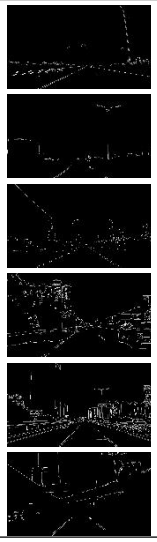
Hasil menunjukkan bahwa sistem mampu mendeteksi garis marka secara konsisten, terutama pada kondisi pencahayaan terang dan kontras tinggi antara marka dan permukaan jalan. Pada kondisi cahaya redup atau marka yang pudar, sistem tetap mampu mendeteksi tepi, meskipun kualitasnya menurun.

Penggunaan tahapan grayscale dan Gaussian blur terbukti efektif dalam mengurangi *noise*, sehingga algoritma Canny dapat bekerja lebih stabil. Citra hasil deteksi tepi kemudian difokuskan pada area *Region of Interest* (RoI), yang secara signifikan membantu dalam membatasi area pemrosesan hanya pada bagian penting, yakni area bawah citra tempat marka biasanya berada

3.2. Hasil Tracking Marka

Pada tahap pengujian ini, selanjutnya dilakukan pengujian untuk memastikan sistem dapat melacak marka jalan secara berkelanjutan saat kendaraan bergerak. Uji coba dilakukan pada beberapa kondisi pencahayaan yang berbeda. Sistem melakukan identifikasi posisi kiri dan kanan marka, lalu menghitung jalur tengah (*centerline*) sebagai referensi utama untuk arah kendaraan.

Tabel 2.Hasil Pengujian Tracking Marka Jalan

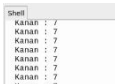
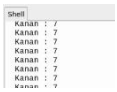
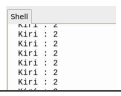
Citra Asli	Citra Tepi Canny	Masking
		

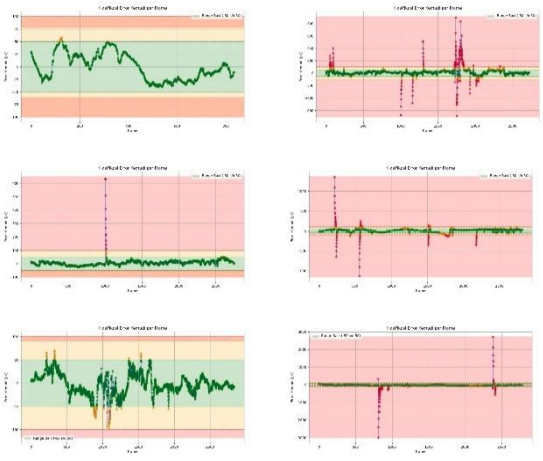
Setelah hasil masking dan deteksi tepi diperoleh, sistem melakukan analisis lanjutan untuk menentukan posisi kiri dan kanan garis marka. Posisi jalur tengah dihitung dari dua garis tersebut, atau dengan estimasi jika hanya satu garis terdeteksi. Hasil menunjukkan bahwa sistem mampu menjaga posisi *centerline* secara stabil pada jalur lurus dan dapat menyesuaikan saat terjadi belokan. Bahkan ketika hanya satu sisi garis terdeteksi, sistem tetap melakukan estimasi posisi jalur berdasarkan data sebelumnya dan asumsi lebar jalur, yang menunjukkan kemampuan prediksi yang cukup adaptif

3.3. Hasil Kemudi Sistem

Pada pengujian ini fokus pada penerapan hasil *error* jalur sebagai dasar pengendalian kemudi. Nilai *error* posisi dihitung sebagai selisih antara *xmid* (titik tengah marka) dan *xcenter* (tengah *frame* kamera), lalu dikonversi menjadi perintah sudut belok *steering*.

Tabel 3.Hasil Pengujian Kemudi Sistem

Tracking Marka	Output Deteksi	Keterangan
		Marka Lurus
		Marka Belok Kanan
		Marka Belok Kiri



Gambar 3. Grafik *Error* Kemudi Per-*Frame*

Hasil uji menunjukkan bahwa sistem mampu memberikan respon kemudi yang dinamis dan mengikuti arah marka, bisa dilihat pada tabel 3. Ketika jalur membelok ke kanan atau kiri, sudut kemudi berubah secara proporsional mengikuti nilai *error*. Saat berada di jalur lurus, servo cenderung mempertahankan arah tetap. Sistem berhasil melakukan koreksi arah secara kontinu sepanjang video berjalan. Gambar 3 menunjukkan distribusi nilai *error* kemudi secara per-*frame* dari enam video uji dengan kondisi lingkungan yang berbeda. Secara umum, grafik menunjukkan bahwa sistem mampu menjaga *error* dalam batas toleransi (ditandai area hijau), terutama pada video dengan pencahayaan baik seperti kondisi cerah. Beberapa lonjakan *error* (area merah) terjadi pada kondisi berawan, hujan, atau cahaya redup (sore hari), yang disebabkan oleh gangguan visual seperti bayangan, marka pudar, atau pantulan cahaya.

Meskipun terjadi fluktuasi pada kondisi ekstrem, sistem menunjukkan kemampuan koreksi arah yang cukup stabil dan adaptif sepanjang lintasan. Hal ini mengindikasikan bahwa pendekatan deteksi tepi dan estimasi jalur yang digunakan masih efektif untuk skala prototipe.

3.4 Analisis Pembahasan Hasil

Secara keseluruhan, sistem menunjukkan performa yang baik dalam mengolah citra untuk deteksi, pelacakan, dan pengendalian arah kendaraan. Penggunaan video sebagai input citra memberikan gambaran nyata terhadap berbagai kondisi lingkungan, seperti perbedaan pencahayaan, cuaca, dan kondisi fisik marka jalan. Dari hasil pengujian pada enam video dengan total lebih dari 16.000 *frame*, sistem terbukti mampu mempertahankan akurasi deteksi dan pelacakan marka pada sebagian besar kondisi. Ketika hanya satu sisi marka terdeteksi, sistem secara adaptif

menggunakan pendekatan estimasi posisi jalur tengah berdasarkan lebar jalur dan data posisi sebelumnya. Pendekatan ini memungkinkan sistem untuk tetap menjaga arah kendaraan secara stabil, meskipun akurasi deteksi menurun.

Tabel 4.Hasil Jumlah Jalur Terdeteksi

No	Total *Frame* Citra Marka	*Error* Terdeteksi	Rata-rata *error* kemudi (px)	Akurasi (%)
1	2706	62	24.54	97.76
2	2723	160	34.46	94.45
3	2750	23	14.07	99.17
4	2813	118	35.06	95.97
5	2837	97	17.40	96.72
6	2779	68	39.72	97.61
Rata-rata				96.95

Berdasarkan hasil deteksi, video 3 yang diambil pada kondisi cuaca cerah menunjukkan performa terbaik dengan akurasi tertinggi sebesar 99,17% dan *error* kemudi terkecil (14,07 piksel). Hal ini disebabkan oleh kontras tinggi antara marka dan permukaan jalan yang memudahkan proses deteksi tepi. Sebaliknya, video 2 (berawan gelap) dan video 4 (berbayang) mengalami penurunan akurasi hingga masing-masing 94,45% dan 95,97%, dengan *error* kemudi rata-rata yang lebih tinggi. Penyebabnya adalah pencahayaan yang tidak merata dan keberadaan bayangan yang mengganggu visibilitas marka. Pada video 5 (hujan ringan), meskipun terdapat pantulan cahaya dan *noise* dari permukaan jalan yang basah, sistem tetap mampu menjaga akurasi yang baik sebesar 96,72%. Hal ini menunjukkan efektivitas tahap Gaussian blur dan segmentasi *Region of Interest* (RoI) dalam mereduksi gangguan visual. Video 6 yang direkam pada sore hari menunjukkan tantangan signifikan akibat cahaya yang redup dan bayangan panjang, sehingga menghasilkan *error* kemudi tertinggi sebesar 39,72 piksel, walaupun akurasinya masih tergolong tinggi, yaitu 97,61%.

Namun demikian, dalam kondisi ekstrem yang belum diuji pada penelitian ini, seperti hujan deras, jalan malam hari tanpa penerangan, atau marka putus-putus, metode deteksi perlu disempurnakan. Tantangan terbesar terjadi ketika marka tidak terlihat sama sekali. Meski begitu, sistem estimasi garis tengah tetap mampu menjaga arah kendaraan dalam batas toleransi.

Secara umum, hasil ini membuktikan bahwa algoritma Canny *Edge Detection* cukup andal untuk skenario pencahayaan bervariasi dan dapat dijadikan pendekatan awal untuk sistem kendaraan otonom skala ringan. Penggabungan dengan metode segmentasi berbasis pembelajaran mesin dapat menjadi langkah pengembangan berikutnya untuk meningkatkan akurasi pada kondisi yang lebih ekstrem.

4. Kesimpulan

Penelitian ini berhasil mengimplementasikan sistem deteksi dan pelacakan marka jalan serta kendali kemudi otomatis pada kendaraan otonom skala prototipe menggunakan metode Canny *Edge Detection*. Sistem menunjukkan performa yang baik, dengan rata-rata akurasi deteksi mencapai 96% dan respon kemudi yang stabil terhadap perubahan arah jalur. Proses pelacakan berlangsung secara konsisten, bahkan saat hanya satu sisi garis marka terdeteksi. Mekanisme estimasi jalur tengah terbukti efektif dalam menjaga arah kendaraan tetap berada di jalurnya, sehingga sistem mampu beroperasi dalam kondisi visual yang tidak ideal.

Keterbatasan utama sistem terletak pada ketergantungannya terhadap kualitas citra yang baik agar metode *Canny Edge Detection* dapat mendeteksi tepi marka secara presisi. Dalam kondisi pencahayaan rendah atau ketika marka jalan tampak pudar, kontras antara marka dan permukaan jalan menurun secara signifikan, sehingga hasil deteksi tepi menjadi tidak konsisten. Akibatnya, algoritma pelacakan gagal menentukan posisi jalur tengah secara akurat, yang berdampak pada kestabilan arah kendaraan. Untuk mengatasi keterbatasan tersebut, pengembangan sistem selanjutnya akan difokuskan pada integrasi metode segmentasi berbasis *deep learning*, seperti *Convolutional Neural Network* (CNN), yang mampu mengenali pola marka secara lebih adaptif meskipun kontur visual tidak terlihat secara jelas. Selain itu, peningkatan efisiensi pemrosesan data akan dilakukan melalui teknik *multithreading* dan penyesuaian resolusi input guna menjaga waktu respon sistem tetap *real-time* ketika digunakan dalam kondisi lingkungan nyata yang dinamis.

Daftar Rujukan

[1] Abdiansyah, F, K A Santoso, and A Kamsyakawuni. 2021. "Perbandigan Image RGB Dan Grayscale Pada Pengkodean Image Dengan Metode 3D Playfair." PRISMA, Prosiding Seminar Nasional Matematika 4: 524–33. https://journal.unnes.ac.id/sju/index.php/prisma/.

[2] Adhayanti, Nurul, Andika Muhamad, and Romdhoni Susiloatmadja. 2024. ICIT Journal PERBANDINGAN DETEKSI TEPI (*EDGE DETECTION*) CITRA DIGITAL DENGAN MENGGUNAKAN GUI MATLAB.

[3] Almusawi, Husam A, Mohammed Al-Jabali, Amro M Khaled, Korondi Péter, and Husi Géza. 2022. "Self-Driving Robotic Car Utilizing Image Processing and Machine Learning." IOP Conference Series: Materials Science and Engineering 1256(1): 012024. doi:10.1088/1757-899x/1256/1/012024.

[4] Aria, Muhammad. 2020. "Metode Perencanaan Jalur Kendaraan Otonom Di Lingkungan Perkotaan Dari Sudut Pandang Filosofi Kuhn Dan Filosofi Popper." Telekontran : Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan 7(2). doi:10.34010/telekontran.v7i2.2627.

[5] Bathla, Gourav, Kishor Bhadane, Rahul Kumar Singh, Rajneesh Kumar, Rajanikanth Aluvalu, Rajalakshmi Krishnamurthi, Adarsh Kumar, R. N. Thakur, and Shakila Basheer. 2022. "Autonomous Vehicles and Intelligent Automation: Applications, Challenges, and Opportunities." Mobile Information Systems 2022. doi:10.1155/2022/7632892.

[6] Damanik, Marni. 2022. 1 Journal of Informatics, Electrical and Electronics Engineering Aplikasi Pengenalan Identitas Gambar

Dengan Menggunakan Metode Backpropagation. https://djournals.com/jieee.

[7] Filsa, Nisfal, Widodo, and Bambang Prasetya Adhi. 2019. "Kinerja Metode Canny Untuk Mendeteksi Tepi Dalam Mengidentifikasi Tulisan Pada Citra Digital Meme." PINTER : Jurnal Pendidikan Teknik Informatika dan Komputer 3(1): 45–53. doi:10.21009/pinter.3.1.8.

[8] Hairudin, Muhammad Samsul Arip, Mia Kusmiati, and Ika Rachmawati. 2023. "Kajian Autonomous Car Untuk Mengatasi Kemacetan Dan Kecelakaan Lintas Di Jakarta."

[9] Kuo, C. Y., Y. R. Lu, and S. M. Yang. 2019. "On the Image Sensor Processing for Lane Detection and Control in Vehicle Lane Keeping Systems." Sensors (Switzerland) 19(7). doi:10.3390/s19071665.

[10] Muhammad Rizky Alditra Utama, Kgs, Rusydi Umar, and Anton Yuhdana. 2022. "*Edge Detection* Comparative Analysis Using Roberts, Sobel, Prewitt, and Canny Methods." Jurnal Teknologi dan Sistem Komputer 10(2): 67–71. doi:10.14710/jtsiskom.2022.14209.

[11] Pramudiya, Relin, Cerwyn Asyraq, Aldo Kadafi, and Ricky Putra Sardika. 2024. "ANALISIS GAMBAR MENGGUNAKAN METODE GRAYSCALE HSV (HUE, SATURATION, VALUE)." Just IT : Jurnal Sistem Informasi, Teknologi Informasi dan Komputer Vol. 14, No. 3.

[12] Ramisetty, Umamaheswari, M Grace Mercy, V Nooka Raju, N Jagadesh Babu, P Ashok Kumar, and Vempalle Rafi. 2024. "*REAL-TIME* LANE DETECTION USING RASPBERRY PI FOR AN AUTONOMOUS VEHICLE." 19(7). www.arpnjournals.com.

[13] Rasdiyanti, Aprilia Dwi, Hayat, and Suyeno. 2024. "Indonesia Government Policy Strategy to Support Autonomous Vehicles Implementation for Public Transport Strategi Kebijakan Pemerintah Indonesia Dalam Mendukung Penerapan Kendaraan Otonom Untuk Transportasi Publik." Jurnal Aristo (Social, Politic, Humaniora 12(2): 452–74.

[14] Salkiawati, Ratna, Hendarman Lubis, and Allan Desi Alexander. 2021. "Implementasi Canny *Edge Detection* Pada Aplikasi Pendeteksi Jalur Lalu Lintas." JURNAL MEDIA INFORMATIKA BUDIDARMA 5(1): 164. doi:10.30865/mib.v5i1.2502.

[15] Siagian, Novita. 2023. "Perancangan Aplikasi Pengolahan Citra Digital Untuk Penajaman Sisi Citra Hasil Fingerprint Menggunakan Metode Fourier Phase Only Synthesis." Teknologi Dan Informasi 1(2): 66–75. https://journal.grahamitra.id/index.php/jurikti.

[16] Sita, Tisara, and Dian Rusmanawati. 2024. "Marka Jalan Berpendar Dalam Gelap: Inovasi Preservasi Jalan Berkelanjutan." Jurnal Ilmiah Teknik Sipil Politeknik Negeri Samarinda Vol. XVI No. 1.

[17] Supiyandi Supiyandi, Muhammad Abdul Mujib, Khairul Azis, Rahmat Abdillah, and Salsa Nabila Iskandar. 2024. "Penerapan Teknologi Pengolahan Citra Dalam Analisis Data Visual Pada Tinjauan Komprehensif." Jurnal Kendali Teknik dan Sains 2(3): 179–87. doi:10.59581/jkts-widyakarya.v2i3.3796.

[18] Wang, Jun, Li Zhang, Yanjun Huang, and Jian Zhao. 2020. "Safety of Autonomous Vehicles." Journal of Advanced Transportation 2020. doi:10.1155/2020/8867757.

[19] Wicaksono, Damar, Putra Almeyda, Irfan Mikola, Muldiyanto Putra, and Letty Malihatuningrum. 2024. Analisis Perbandingan Metode Pra Pemrosesan Citra Untuk Deteksi Tepi Canny Pada Citra Berbagai Kondisi Jalan Menggunakan Bahasa Pemrograman Python. JUTIKOMP.

[20] Widiyanto, Didit. 2020. "Tinjauan Metode RoI (*Region of Interest*) Dengan Metode Pengambangan Otsu Dan Klasterisasi K-Mean; Hasil Dan Tantangannya."

[21] Zahra Salsa Azizah, Allyssa, Fuaz Nazar To Yalis, Ikade Shiva Narayana, Karen Azelia Syalom, and Perani Rosyani. 2024. 1 Jurnal Artificial Inteligent dan Sistem Penunjang Keputusan Pengolahan Citra Digital Dengan Penerapan Teknik Ambang Batas: Studi Kasus Menggunakan Opencv. https://jurnalmahasiswa.com/index.php/aidanspk.