

Optimasi Sistem Deteksi Jalur pada Miniatur Kendaraan Otonom dengan menggunakan Algoritma *Hough Transform* berbasis *Computer Vision*

Silmi Awalya
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia

awalyasilmi@student.telkomuniversit
y.ac.id

Dr.-Ing. Fiky Y. Suratman, S.T., M.T
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia

fysuratman@telkomuniversity.ac.id

Dr. Eng. Arief Suryadi Satyawan, M. T
Fakultas Teknik Elektro Telkom
University Bandung, Indonesia

Arie021@brin.go.id

Abstrak --- Perkembangan kendaraan otonom telah menjadi fokus utama dalam dunia teknologi, khususnya dalam aspek deteksi jalur yang presisi pada kondisi lingkungan dinamis. Penelitian ini bertujuan untuk mengembangkan sistem deteksi jalur pada kendaraan otonom miniatur menggunakan algoritma *Hough Transform* yang dioptimasi dengan strategi deteksi multi-ruang warna berbasis *computer vision*. Sistem ini diimplementasikan pada modul *Duckiebot DB21J* yang dilengkapi dengan kamera *IMX219* dan pemrosesan oleh *Jetson Nano*. Metodologi yang digunakan meliputi studi literatur, perancangan sistem, implementasi perangkat keras dan perangkat lunak, serta pengujian komprehensif pada lima skenario berbeda dalam dua kondisi pencahayaan (305 lux dan 181 lux). Hasil menunjukkan bahwa sistem mampu mendeteksi jalur dengan akurasi 97.14% pada kondisi terang dan 96.40% pada kondisi standar, melampaui target minimal 90% dengan *detection rate* konsisten 99.26% dan 99.16%. Sistem menunjukkan presisi spasial yang sangat baik dengan kesalahan sudut 0.494° - 0.57° dan kesalahan posisi 1.19-1.31 cm, serta waktu pemrosesan rata-rata 213-230 ms. Penelitian ini membuktikan superioritas metode yang dikembangkan dengan peningkatan akurasi 6.40%-12.14% dibanding penelitian sebelumnya yang menggunakan *CNN+Hough Transform*, *Hough+Gaussian Filter*, dan *SVM Model*. Sistem ini berpotensi diterapkan pada platform robotika pendidikan dan penelitian lanjutan dalam otomasi kendaraan.

Kata kunci — kendaraan otonom, deteksi jalur, *computer vision*, *Hough Transform*, *Jetson Nano*, *Duckiebot*

I. PENDAHULUAN

Perkembangan teknologi kendaraan otonom telah menjadi perhatian besar dalam beberapa tahun terakhir, terutama dalam upaya mengintegrasikan sistem navigasi berbasis *computer vision* yang mampu beroperasi secara andal dalam lingkungan yang dinamis. Salah satu tantangan utama dalam teknologi ini adalah deteksi jalur yang presisi, terutama ketika kendaraan harus tetap berada di lintasannya meskipun menghadapi gangguan seperti perubahan pencahayaan, variasi permukaan jalan, serta noise visual dari lingkungan sekitar [1], [2]. Untuk mengatasi tantangan tersebut, berbagai pendekatan telah dikembangkan, salah satunya adalah penerapan algoritma *Hough Transform* yang dikenal unggul dalam mendeteksi bentuk geometris seperti garis dalam citra digital.

Dalam konteks miniatur kendaraan otonom, sistem deteksi jalur berbasis *Hough Transform* menawarkan keunggulan dari sisi efisiensi komputasi dan ketahanan terhadap *noise*, menjadikannya solusi ideal untuk sistem real-time berbasis kamera. Penelitian ini mengimplementasikan algoritma tersebut pada kendaraan mini berbasis *Duckiebot DB21J*, yang menggunakan kamera *IMX219* dan platform *Jetson Nano*. Dengan pengujian di berbagai kondisi pencahayaan dan konfigurasi lintasan, sistem ini dirancang untuk mengevaluasi akurasi deteksi, latensi pemrosesan, dan ketahanan sistem terhadap variabel lingkungan yang berubah-ubah [3], [4].

II. KAJIAN TEORI

A. Deteksi Jalur

Sistem deteksi jalur adalah teknologi berbasis *computer vision* yang dirancang untuk mengidentifikasi batas jalur pada permukaan jalan secara akurat. Sistem ini menggunakan pipeline pemrosesan citra yang terdiri dari *preprocessing*, *edge detection*, *feature extraction*, dan *post-processing*. Teknologi ini memanfaatkan gambar yang ditangkap oleh kamera untuk mendeteksi elemen geometris, seperti jalur atau garis yang digunakan kendaraan otonom sebagai panduan navigasi.

B. *Hough Transform*

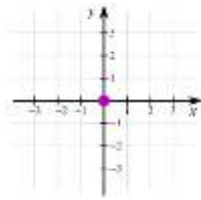
Penelitian ini menggunakan algoritma *Hough Transform* sebagai metode utama untuk deteksi jalur pada sistem kendaraan otonom miniatur. Algoritma ini dipilih karena kemampuannya dalam mendeteksi garis pada citra dengan ketahanan tinggi terhadap *noise* dan perubahan kondisi pencahayaan. Prinsip dasar *Hough Transform* adalah mengkonversi setiap titik dalam citra ke ruang parameter (r, θ) menggunakan persamaan garis polar, dimana sekumpulan titik yang kolinear dalam ruang citra akan menghasilkan kurva yang berpotongan di satu titik dalam ruang parameter [5] [16].

Persamaan :

$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (1)$$

DI mana :

- r adalah jarak dari titik asal (origin) ke garis terdekat.
- θ adalah sudut antara sumbu-x dan garis yang menghubungkan titik asal dengan titik terdekat pada garis.



Gambar 1

C. Open CV

OpenCV adalah library pemrograman *open source* yang dirancang untuk aplikasi *computer vision* secara real time. Library ini bebas digunakan, baik untuk keperluan akademis maupun komersial, dan mendukung berbagai bahasa pemrograman seperti C++, C, Python, dan Java. OpenCV kompatibel dengan beragam platform, termasuk *Windows*, *Linux*, *Android*, dan *Mac*. Teknologi ini digunakan dalam berbagai bidang, seperti seni interaktif, inspeksi tambang, hingga peta web berbasis robotik. Fitur-fitur utama OpenCV mencakup manipulasi data gambar, I/O video, manipulasi matriks dan vektor, pengolahan citra dasar seperti filter dan deteksi tepi, analisis struktur gambar, kalibrasi kamera, analisis gerakan, serta pengenalan objek [7].

D. Duckiebot DB21J

Duckiebot DB21J merupakan platform robotika open-source yang dirancang komunitas Duckietown untuk penelitian kendaraan otonom skala mini, mengintegrasikan kamera IMX219, Jetson Nano sebagai unit pemrosesan, dan motor DC untuk aktuasi gerak. Platform ini kompatibel dengan *framework Robot Operating System* (ROS), memungkinkan integrasi algoritma *Hough Transform* untuk deteksi jalur secara *real-time*. Dalam penelitian ini, citra lingkungan diproses oleh Jetson Nano untuk mengontrol arah gerak kendaraan berdasarkan posisi garis jalur yang terdeteksi, memungkinkan pengujian dengan skenario pencahayaan dan lintasan yang bervariasi [8].

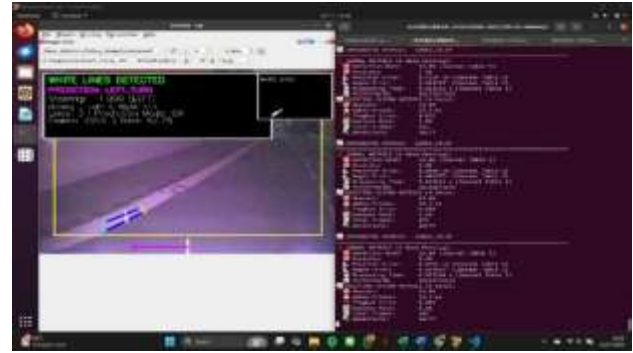
III. METODE

Penelitian ini mengembangkan sistem deteksi jalur pada miniatur kendaraan otonom menggunakan algoritma Hough Transform berbasis computer vision. Sistem diimplementasikan pada platform Duckiebot DB21J yang dilengkapi dengan kamera IMX219 dan diproses menggunakan Jetson Nano untuk mencapai deteksi jalur secara real-time dengan target akurasi minimal 90% dan latensi pemrosesan maksimal 150 ms.

Persiapan sistem dimulai dengan mempersiapkan lintasan deteksi jalur dilakukan menggunakan platform miniatur kendaraan otonom dengan modul Duckiebot DB21J sebagai sistem akuisisi dan pemrosesan data. Untuk menilai performa secara komprehensif, metodologi pengujian menggunakan pendekatan evaluasi multi-dimensi yang mengkombinasikan confusion matrix elements, computational performance, dan spatial error measurements. Algoritma *Hough Transform* merupakan inti dari sistem deteksi garis pada penelitian ini. Beberapa parameter kritis dikonfigurasi untuk mengoptimalkan deteksi garis dalam kondisi real-time dengan tingkat akurasi yang memadai.



GAMBAR 2
(Lintasan Uji coba)



GAMBAR 3
(Tampilan proses pengambilan data)

Pada gambar 2 dan 3 merupakan lintasan dan tampilan proses pengambilan data yang dilakukan untuk pengujian sistem deteksi jalur berbasis kendaraan otonom

Algoritma *Hough Transform* merupakan inti dari sistem deteksi garis pada penelitian ini. Beberapa parameter kritis dikonfigurasi untuk mengoptimalkan deteksi garis dalam kondisi real-time dengan tingkat akurasi yang memadai.

Parameter	Nilai	Deskripsi
Hough Threshold	5	Minimum votes untuk mendeteksi garis
Min Line Length	20	Panjang minimum garis yang valid
Max Line Gap	20	Gap maksimum yang diizinkan dalam satu garis
Rho Resolution	1	Resolusi jarak dalam piksel
Theta Resolution	$\pi/180$	Resolusi sudut dalam radian

TABEL 1
(Parameter Hough Transform)

Dalam implementasi sistem deteksi jalur berbasis Hough Transform, beberapa parameter dikonfigurasi secara manual untuk mengoptimalkan deteksi garis putih dalam berbagai kondisi pencahayaan dan lingkungan.

Parameter	Nilai	Deskripsi
Threshold HSV Lower	[5, 60, 60]	Batas bawah deteksi warna putih dalam ruang warna HSV
Threshold HSV Upper	[25, 255, 255]	Batas atas deteksi warna putih dalam ruang warna HSV
Grayscale Threshold	210	Threshold untuk deteksi putih pada citra grayscale
LAB L-channel Threshold	180	Threshold untuk deteksi putih pada channel L dalam ruang LAB

TABEL 2
(Parameter Deteksi Garis Putih)

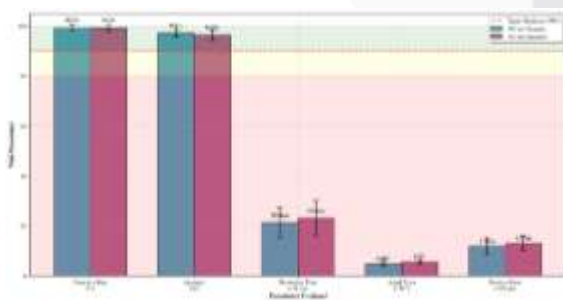


GAMBAR 4
(Flowchart)

Gambar 4 menunjukkan flowchart keseluruhan sistem deteksi jalur. Sistem menghasilkan output perintah kontrol roda & debug image real-time melalui publikasi ROS topics untuk *wheel commands* dan *visualization*. Proses berlanjut secara kontinyu melalui titik keputusan lanjutan yang memungkinkan sistem kembali ke tahap input citra untuk *frame processing* berikutnya, menciptakan closed-loop real-time lane detection dan navigation system hingga sistem dihentikan atau mencapai kondisi termination.

IV. HASIL DAN PEMBAHASAN

Pengujian algoritma *Hough Transform* untuk deteksi jalur dilakukan menggunakan platform Duckiebot DB21J dengan kamera IMX219-160 dan Jetson Nano. Evaluasi dilakukan pada lima skenario berbeda dalam dua kondisi pencahayaan: terang (305 lux) dan standar (181 lux). Setiap skenario pengujian dilakukan sebanyak lima kali untuk memastikan validitas statistik dan konsistensi hasil. Untuk menilai performa secara komprehensif, metodologi pengujian menggunakan pendekatan evaluasi multi-dimensi yang mengombinasikan *confusion matrix elements*, *computational performance*, dan *spatial error measurements*.



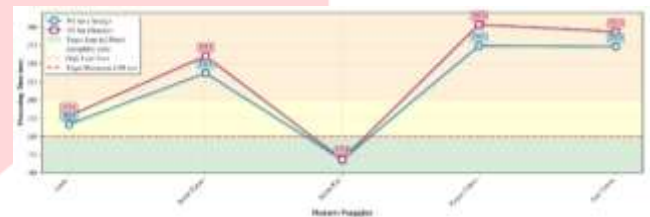
GAMBAR 3
(Multi-Parameter Performance Comparison)

Pada gambar 3 merupakan visualisasi untuk perbandingan *performance* algoritma *Hough transform* antara kondisi pencahayaan terang (305 lux) dan standar (181 lux) menunjukkan *robustness* yang *excellent* dengan hasil yang mengejutkan pada *detection rate*.

Pada gambar 3 merupakan visualisasi untuk perbandingan *performance* algoritma *Hough transform* antara kondisi

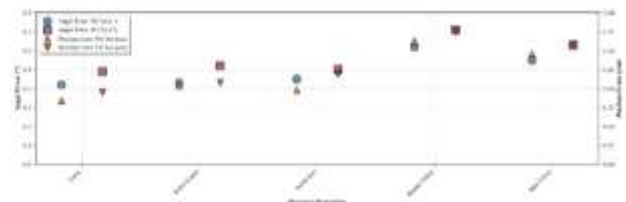
pencahayaan terang (305 lux) dan standar (181 lux) menunjukkan *robustness* yang *excellent* dengan hasil yang mengejutkan pada *detection rate*. *Detection rate* justru mengalami peningkatan 0,54% dari 99,26% menjadi 99,8%, sementara *accuracy* mengalami *degradation* minimal 0,74% dari 97,14% menjadi 96,4%. *Spatial error* mengalami peningkatan terkontrol sebesar 15,4% untuk *angel error* (0,494° menjadi 0,57°) dan 10,3% untuk *position error* (1,19 cm menjadi 1,312 cm). Peningkatan *detection rate* pada kondisi standar mengindikasikan fenomena menarik dimana penerangan yang berkurang justru mengoptimalkan distribusi kontras tepi pada skenario yang spesifik, memberikan keuntungan pada *Hough space accumulator* untuk mendeteksi *white line* dengan lebih konsisten. Stabilitas dan *improvement* ini disebabkan oleh implementasi multi-color space detection strategy yang menggabungkan grayscale, HSV, dan LAB detection, menciptakan adaptabilitas yang superior terhadap kondisi pencahayaan yang bervariasi dimana setiap metode saling mengkompensasi ketika kondisi pencahayaan berubah.

GAMBAR 4
(Analisis Processing Time per Skenario)



Evaluasi processing time yang divisualisasikan pada gambar 4.6 (a) menunjukkan peningkatan efisiensi sebesar 7,4% dari 230,4 ms (kondisi 181 lux) menjadi 213,4 ms (kondisi 305 lux), mengindikasikan bahwa *enhanced illumination conditions* memberikan keunggulan komputasional yang signifikan. Peningkatan performa ini disebabkan oleh optimized image quality yang mengurangi preprocessing overhead, dimana higher contrast levels memungkinkan *Canny edge detection* parameters (50-150) bekerja lebih efisien tanpa memerlukan penyaringan kebisingan intensif. Kondisi pencahayaan optimal (305 lux) memfasilitasi *Hough transform parameters* (threshold=5, rho=1, theta= $\pi/180$) untuk mencapai konversi yang lebih cepat, sementara operasi morfologi dengan ellipse kernel memproses cleaner input images dengan kompleksitas komputasi yang berkurang. *Superior illumination* juga meminimalkan *false edge detections*, sehingga keterbatasan validasi garis yang komprehensif dapat berfokus pada actual line features tanpa proses penyaringan yang luas. Hasil analisis ini memvalidasi hypothesis bahwa optimal *lighting conditions* tidak hanya meningkatkan *detection accuracy*, tetapi juga *significantly improves computational efficiency*, memberikan dual advantage untuk *real-world autonomous vehicle implementation* dengan *faster response time* dan *reliable performance*.

GAMBAR 5

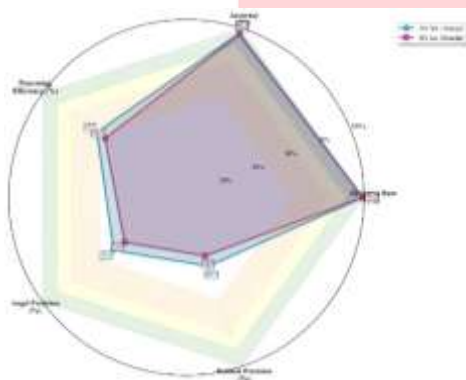


(Analisis Spatial Error per Skenario)

Analisis *spatial error* pada Gambar 4.6 (b) menunjukkan *degradation* terkontrol antara kondisi pencahayaan 305 lux dan 181 lux dengan pattern yang konsisten *across all scenarios*. Kondisi terang (305 lux) mencapai *angel error* rata-rata 0,494° dan *position error* 1,19 cm, sementara kondisi standar (181 lux) mengalami

peningkatan menjadi $0,57^\circ$ dan $1,31$ cm, merepresentasikan degradation $15,4\%$ dan $10,1\%$ masing-masing. *Pattern degradation* ini disebabkan oleh *reduced edge quality* pada *lower illumination* yang mempengaruhi *accuracy* dalam *line detection* dan *center calculation*, dimana camera resolusi 640×480 menghasilkan *edge pixels* yang kurang sharp pada kondisi pencahayaan berkurang. Meskipun terjadi peningkatan error, *spatial accuracy* tetap dalam *range sub-degree* dan sub-2cm level yang *excellent* untuk *autonomous navigation*, mengkonfirmasi *robustness* algoritma dengan *precision spatial* yang superior pada kedua kondisi pencahayaan sesuai implementasi *steering angle calculation* menggunakan formula $|\text{steering_angle}| \times 5.0^\circ$ untuk *angel error* dan $|\text{steering_angle}| \times 10.0$ cm untuk *position error*.

Hasil analisis ini memvalidasi *hypothesis* bahwa optimal *lighting conditions* tidak hanya meningkatkan *detection accuracy*, tetapi juga *significantly improves computational efficiency*, memberikan dual *advantage* untuk *real-world autonomous vehicle implementation* dengan *faster response time* dan *reliable performance*.



GAMBAR 6
(Radar Chart Robustness Assesment)

Untuk memberikan visualisasi komprehensif terhadap *robustness* algoritma yang telah dianalisis secara menyeluruh pada perbandingan kondisi pencahayaan. Gambar 6 menunjukkan radar *chart assesment* performa keseluruhan dengan normalisasi semua parameter evaluasi pada skala 0-100%.

V. KESIMPULAN

Penelitian ini berhasil mengembangkan metode optimasi algoritma *Hough Transform* yang mencapai *accuracy* 97.14% pada kondisi pencahayaan terang (305 lux) dan 96.40% pada kondisi pencahayaan standar (181 lux), melampaui target minimal 90% dengan margin 7.14% dan 6.40% . *Detection rate* yang dicapai mencapai 99.26% dan 99.16% dengan penurunan minimal hanya 0.10% antar kondisi pencahayaan, membuktikan sistem tahan terhadap perubahan pencahayaan dengan degradasi akurasi minimal 0.74% . Implementasi dan evaluasi sistem deteksi jalur pada

platform *Duckiebot* DB21J berhasil mencapai presisi spasial yang sangat baik dengan kesalahan sudut 0.494° - 0.57° dan kesalahan posisi 1.19 - 1.31 cm, serta *processing time* rata-rata 213 - 230 ms yang mendukung navigasi waktu nyata pada sistem kendaraan otonom miniatur.

Penelitian ini memberikan kontribusi signifikan dengan membuktikan superioritas metode yang dikembangkan dibanding penelitian sebelumnya, mencapai peningkatan *accuracy* 6.40% - 7.14% terhadap *CNN+Hough Transform* [18] (90%), 2.58% - 3.32% terhadap *Probabilistic Hough Transform* [6] (93.82%), dan 6.80% - 7.54% terhadap *Hough Transform* [19] (89.6%). Kombinasi strategi deteksi multi-ruang warna dan optimasi parameter *Hough Transform* terbukti menciptakan solusi yang robust dan efisien untuk aplikasi deteksi jalur pada platform robotika miniatur, menunjukkan bahwa optimasi algoritma tradisional dengan *preprocessing* yang tepat dapat menghasilkan performa yang unggul dengan beban komputasi yang dapat diterima untuk implementasi *real-time*.

REFERENSI

- [1] M. IKBAL and R. A. Saputra, "Pengenalan Rambu Lalu Lintas Menggunakan Metode Yolov8," *JIKA (Jurnal Inform.*, vol. 8, no. 2, p. 204, 2024, doi: 10.31000/jika.v8i2.10609.
- [2] A. B. Pratama, R. Effendi, A. Kadir, and A. Hady, "81667-165267-1-Pb," vol. 11, no. 1, 2022.
- [3] M. A. L. Furqon, S. Fikri, J. T. Elektro, F. Teknik, and U. Sriwijaya, "DAN DEEP LEARNING SECARA REALTIME SEBAGAI INPUT STEERING ANGLE PADA," 2024.
- [4] Arducam, "Native Camera, 8MP IMX219." [Online]. Available: <https://docs.arducam.com/Nvidia-Jetson-Camera/Introduction-to-ArducamJetson-Cameras/>
- [5] Wikipedia, "Transformasi Hough," Wikimedia Project. [Online]. Available: https://en.wikipedia.org/wiki/Hough_transform
- [6] M. Marzougui, A. Alasiry, Y. Kortli, and J. Baili, "A Lane Tracking Method Based on Progressive Probabilistic Hough transform," *IEEE Access*, vol. 8, pp. 84893–84905, 2020, doi: 10.1109/ACCESS.2020.2991930.
- [7] OpenCV.org, "OpenCV Library Features," OpenCV Official Website. Accessed: Dec. 24, 2024. [Online]. Available: <https://opencv.org/about/>
- [8] A. Paull, L. Saeedi, M. Seto, and H. Li, "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Singapore, 2017, pp. 1497–1504.