

# Deteksi Marka Jalan Menggunakan *RGB Camera* Untuk *Lane Keeping Assist*

Ardiansyah Al Farouq <sup>1</sup>, Billy Montolalu <sup>2</sup> and Adinda Panca Mochamad <sup>2,\*</sup>

<sup>1</sup> Affiliation 1; [alfarouq@ittelkom-sby.ac.id](mailto:alfarouq@ittelkom-sby.ac.id)

<sup>2</sup> Affiliation 2; [billy@ittelkom-sby.ac.id](mailto:billy@ittelkom-sby.ac.id)

\* Correspondence: [adinda.panca.19@student.ce.ittelkom-sby.ac.id](mailto:adinda.panca.19@student.ce.ittelkom-sby.ac.id) ;

**Abstrak:** Penelitian ini bertujuan mengembangkan sistem Lane Keeping Assist berbasis image processing untuk meningkatkan keamanan dan kenyamanan pengemudi di Indonesia. Menghadapi meningkatnya risiko kecelakaan akibat human error, sistem ini mendeteksi garis marka jalan putih dan kuning pada sisi jalan dengan menggunakan data simulasi dan video real. Dalam pengujian simulasi, tingkat keberhasilan deteksi marka jalan mencapai 85%, sementara deteksi pelanggaran marka jalan mencapai 100%. Pada pengujian video real, error tertinggi terjadi pada kecepatan 95 km/jam dan 100 km/jam (15,625% dan 15,893%), sedangkan error terendah pada 60 km/jam (0,640%). Hasil ini menunjukkan potensi penerapan sistem dalam pengembangan kendaraan untuk meningkatkan keselamatan dan kenyamanan pengemudi. Dengan nilai rata-rata error 5,465% pada data video real, penelitian ini memberikan kontribusi berharga dalam mengatasi risiko kecelakaan di jalan raya dengan mengurangi human error melalui teknologi Lane Keeping Assist.

**Kata Kunci:** Garis Marka Jalan, *Human Error*, Kecelakaan, Transportasi .

## Lane Detection Using RGB Camera For Lane Keeping Assist

**Abstract:** This research aims to develop a Lane Keeping Assist system based on image processing to enhance driver safety and comfort in Indonesia. Confronting the escalating accident risks due to human error, the system detects white and yellow lane markings using both simulation and real video data. In simulation testing, lane marking detection achieved an 85% success rate, while violation detection reached 100%. In real video testing, the highest errors occurred at speeds of 95 km/h and 100 km/h (15.625% and 15.893%), with the lowest error at 60 km/h (0.640%). These results highlight the system's potential for vehicle development, enhancing driver safety and comfort. With an average error rate of 5.465% in real video data, this research contributes significantly to addressing road accident risks by mitigating human error through Lane Keeping Assist technology.

**Keywords:** Lane Markings, Human Error, Accidents, Transportation.

## 1. Pendahuluan

Kecelakaan lalu lintas merupakan ancaman serius bagi pengendara jalan raya, dengan potensi bahaya yang dapat mengancam nyawa kapan saja. Faktor-faktor seperti ketidakhati-hatian, kelalaian, mengantuk, dan kurangnya konsentrasi dari pengemudi dapat menyebabkan kecelakaan berbahaya. Data menunjukkan bahwa sekitar 61% kecelakaan pada awal tahun 2022 disebabkan oleh faktor *human error*[1]. Garis marka jalan memiliki peran krusial dalam membantu pengemudi menjaga kendaraannya pada jalur yang benar[2]. Namun, sering kali kendaraan keluar dari garis marka jalan yang ditetapkan, mengakibatkan potensi risiko bagi pengendara jalan dan pengguna jalan lainnya. Upaya perlindungan seperti pembatas beton di bahu jalan dan reflektor marka jalan telah diterapkan, tetapi belum sepenuhnya efektif dalam mengatasi masalah tersebut.

Tujuan dari penelitian ini adalah merancang implementasi dari *safety guide system*, terutama dalam bidang *Lane Keeping Assist*. Sistem ini bertujuan untuk mengurangi kecelakaan yang disebabkan oleh *human error* dengan membantu pengemudi mendeteksi garis marka jalan dan memberikan panduan untuk menjaga kendaraan pada jalur yang benar [3]. Algoritma dalam sistem ini akan mendeteksi garis marka jalan, menampilkan panduan posisi jalur, dan memberikan peringatan jika kendaraan melenceng dari jalur yang seharusnya.

## 2. Tinjauan Pustaka

### 2.1 Kajian Penelitian Terkait

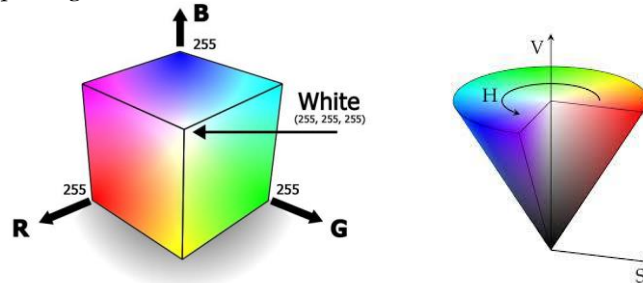
Penelitian ini merangkum beberapa penelitian terdahulu yang relevan dengan fokus pada deteksi marka jalan dan teknologi Lane Keeping Assist. Pertama, Deon Sampurna (2020) dari Universitas Semarang merancang Lane Keeping Assist menggunakan metode homografi[4]. Meskipun berhasil memprediksi arah kendaraan, keterbatasan dalam data citra video menjadi hambatan. Selanjutnya, Bhakti Yudho et al. (2020) dari Universitas Sriwijaya mengembangkan sistem deteksi jalan dan kendaraan dengan memanfaatkan ruang warna HSV. Pendekatan ini berhasil mengatasi masalah perubahan pencahayaan dan variasi warna [5]. Moveh Samuel et al. (2021) dari Universitas Teknologi Malaysia fokus pada algoritma berbasis tepi untuk meningkatkan akurasi deteksi garis marka jalan. Walaupun efektif, kinerja komputasi yang intensif menjadi pertimbangan [6]. Pravin T. Mandlik et al. (2016) membahas sistem peringatan keluar jalur berbasis pengolahan citra menggunakan Raspberry Pi. Sistem ini menampilkan peringatan real-time kepada pengemudi [7]. Terakhir, Charles Edison Chandra et al. mengeksplorasi deteksi marka jalan dan estimasi posisi kendaraan menggunakan metode multiresolution Hough Transform. Penelitian ini berfokus pada pengembangan metode deteksi marka jalan berbasis warna RGB dan HSV, penggunaan transformasi perspektif citra, serta implementasi titik panduan dan peringatan pelanggaran marka jalan dalam teknologi Lane Keeping Assist [8].

### 2.2 Teori Dasar

#### 2.2.1 Ruang Warna

Warna adalah hasil dari cahaya yang diterima oleh mata manusia atau perangkat penglihatan lainnya. Objek yang menerima cahaya akan menyerap dan memantulkan panjang gelombang tertentu, yang kemudian diinterpretasikan sebagai warna oleh retina mata. Ruang warna adalah sistem yang digunakan untuk mengatur dan merepresentasikan warna secara sistematis. Setiap ruang warna memiliki metode dan parameter yang berbeda untuk menggambarkan warna. Misalnya, dalam model warna RGB (*Red-Green-Blue*), warna dibentuk melalui kombinasi intensitas cahaya merah,

hijau, dan biru. Jenis ruang warna umum lainnya termasuk LUV, CMYK, HSL, dan HSV. Ruang warna RGB memanfaatkan tiga warna dasar (merah, hijau, biru) untuk membentuk warna. Setiap komponen warna memiliki rentang nilai 0 hingga 255, menggambarkan intensitas relatif dari setiap warna. Ruang warna RGB konsisten dengan perangkat elektronik, seperti monitor komputer, yang menggunakan model ini untuk tampilan gambar [9].



Gambar 1. Ruang Warna RGB dan HSV

Ruang warna HSV (*Hue, Saturation, Value*) lebih sesuai dengan persepsi manusia terhadap warna. Hue mengacu pada rentang warna, saturation mengukur kejenuhan warna, dan value menggambarkan tingkat kecerahan. Ruang warna HSV memberikan informasi lebih akurat tentang karakteristik warna sesuai dengan persepsi manusia [10].

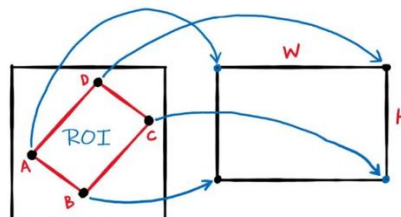
## 2.2.2 Pengolahan Citra

### 2.2.2.1 Segmentasi Warna Citra HSV

Segmentasi warna citra menggunakan ruang warna HSV (*Hue, Saturation, Value*) berfungsi untuk memisahkan objek berdasarkan komponen warna. Ruang warna HSV adalah hasil transformasi dari model warna RGB (*Red, Green, Blue*) dan mencerminkan intensitas, cahaya, atau kecerahan warna dari 0 hingga 100. Ruang warna HSV membantu dalam masking, di mana objek dapat dianggap sebagai *foreground* atau *background* berdasarkan rentang nilai komponen warna [11].

### 2.2.2.2 Transformasi Perspektif Citra

Transformasi perspektif citra adalah metode untuk mengubah sudut pandang dalam citra. Ini melibatkan operasi seperti translasi, rotasi, skala, dan distorsi geometris. Transformasi ini berguna dalam pengolahan gambar, grafika komputer, *augmented reality*, dan pemetaan citra. Langkah-langkahnya meliputi menggeser koordinat, melakukan rotasi dan skala, serta memproyeksikan gambar ke bidang dua dimensi. Teknik ini memungkinkan melihat objek dari sudut pandang yang berbeda dan menciptakan efek visual menarik.



Gambar 2. Transformasi Perspektif Citra

### 2.2.2.3 Masking

Masking adalah teknik pengolahan citra yang memisahkan area tertentu berdasarkan kriteria, seperti rentang warna atau intensitas. Dalam masking, operasi *bitwise AND* dilakukan antara citra asli dan mask, menjaga piksel yang sesuai dengan mask sambil menghapus piksel lainnya. Teknik ini memungkinkan pemisahan objek atau area yang ingin dianalisis atau diproses terpisah dari latar belakang atau bagian yang tidak relevan.

### 2.2.3 Morfologi

Transformasi morfologi dalam pengolahan citra adalah metode yang digunakan untuk mengubah tampilan objek dalam citra dengan tujuan meningkatkan kualitas gambar, mengurangi gangguan berupa *noise*, dan mengungkapkan fitur-fitur objek yang ada. Terdapat beberapa operasi dasar dalam transformasi morfologi, seperti *dilasi* (memperluas area objek), *erosi* (mempersempit area objek), *opening* (erosi diikuti dilasi untuk menghilangkan *noise*), dan *closing* (dilasi diikuti erosi untuk menutupi lubang-lubang kecil dalam objek).

### 2.2.4 Kontur

Kontur dalam pengolahan citra adalah visualisasi dari garis tepi atau batas yang memisahkan objek dari latar belakang dalam gambar. Ini terjadi karena perbedaan intensitas piksel di sekitar objek dalam hal perubahan warna atau kecerahan. Pengenalan kontur memungkinkan identifikasi objek, segmentasi, pengukuran dimensi, dan bentuk. *Library OpenCV* menyediakan berbagai fitur terkait kontur, termasuk perhitungan momen citra untuk pusat massa, luas daerah objek, dan lainnya. "*Contour Area*" mengukur luas area kontur berdasarkan jumlah piksel di dalamnya. Fitur "*Bounding Rectangle*" memungkupi area kontur dengan persegi. "*Point Polygon Test*" mengukur jarak titik dari kontur dan memberikan nilai positif jika dalam kontur, nol jika pada garis kontur, dan negatif jika di luar kontur. Dalam OpenCV, ini dapat diakses melalui perintah "*cv2.pointPolygonTest*".

### 2.2.5 Robot Operating System (ROS)

ROS (*Robot Operating System*) adalah sistem operasi *open-source* untuk pengembangan perangkat lunak robot. Ini berfungsi sebagai perantara antara perangkat keras robot dan sistem operasi komputer, memungkinkan kolaborasi dan pembangunan bersama dalam komunitas pengembang [12]. Gazebo adalah aplikasi simulasi *open-source* yang akurat, memungkinkan pengembang merancang, menguji, dan mengembangkan algoritma robot serta sistem pengendalian sebelum implementasi fisik. Gazebo juga terintegrasi dengan ROS, memungkinkan pengguna untuk mengontrol robot, melakukan pemodelan fisik, dan mensimulasikan sensor [13].

## 3. Metode dan Pemodelan

### 3.1 Pengumpulan Dataset Garis Marka Jalan

Tujuan dari pengumpulan dataset ini adalah untuk melatih dan menguji model deteksi garis marka jalan secara akurat dan handal. Video dataset garis marka jalan menggunakan spesifikasi sebagai berikut:

Tabel 1. Spesifikasi Dataset Video

Spesifikasi	Deskripsi
Resolusi Video	1280p x 720 p
Kecepatan Frame	30 fps
Durasi	65 detik
Variasi Marka	<ul style="list-style-type: none"><li>• Garis Marka Jalan Kuning Lurus Utuh</li><li>• Garis Marka Jalan Putih Lurus Utuh</li><li>• Garis Marka Jalan Putih Putus-Putus</li></ul>
Bidang Pandang	75° dFoV
Cuaca	Cerah Siang Hari

### 3.2 Perancangan Purwarupa Simulasi

Purwarupa marka ini memiliki ukuran rasio 1:5 dari marka asli yang ada di lingkungan nyata. Model purwarupa dibuat menggunakan format .skp (*SketchUp*) dengan *software* 3D untuk menghasilkan model yang akurat. Dalam rancangan purwarupa marka, warna hitam aspal digunakan sebagai lapisan dasar yang mewakili permukaan jalan. Model ini juga mencakup marka putih utuh, marka putih putus-putus, dan marka kuning. Tujuannya adalah mensimulasikan berbagai kondisi marka yang umum ditemukan di jalan raya.

### 3.3 Pengkodean Pengolahan Citra

Kode akan ditulis dalam Python 3.9 dan fokus pada pengolahan citra video dari kamera simulasi ROS Gazebo dan juga dataset video garis marka yang telah dikumpulkan. Tujuannya adalah mendeteksi dan menganalisis marka jalan, klasifikasi marka, pencarian midpoint, serta memberikan peringatan pelanggaran marka. Proses dimulai dengan inisiasi kamera dan pengambilan gambar video yang diolah menggunakan algoritma pengolahan citra, termasuk segmentasi warna, deteksi kontur, dan transformasi perspektif untuk meningkatkan kualitas citra dan memisahkan marka dari latar belakang.

#### 3.3.1 Kalibrasi Data Sensor Kamera

Tahap awal melibatkan penentuan spesifikasi pengambilan data dari kamera sensor di simulasi Gazebo ROS. Ini termasuk resolusi gambar, kecepatan pembaruan *frame* (FPS), format warna citra, dan sudut pandang horizontal. Dalam script yang diberikan, format warna citra adalah R8G8B8 (8 bit untuk komponen merah, hijau, dan biru), resolusi gambar adalah 1280 x 720 piksel, pembaruan frame dilakukan dengan kecepatan 30 FPS, dan sudut pandang horizontal adalah 1.089°.

#### 3.3.2 Segmentasi Warna Garis Marka

Proses segmentasi warna HSV adalah menerapkan segmentasi pada citra yang dilakukan dengan menggunakan data informasi warna yang telah ditentukan sebelumnya untuk memisahkan objek marka jalan dari objek lainnya. Dalam hal ini, digunakan dua rentang warna, yaitu rentang warna kuning (*ylw\_lower* dan *ylw\_upper*) serta rentang warna putih (*wht\_lower* dan *wht\_upper*), sesuai dengan jenis garis marka jalan yang sedang diteliti.

#### 3.3.3 Transformasi Perspektif Citra

Transformasi perspektif citra dilakukan untuk mengubah sudut pandang menjadi pandangan dari atas (*bird's eye view*). Proses ini melibatkan penggunaan matriks transformasi untuk memetakan koordinat titik-titik pada citra asli ke koordinat titik-titik pada citra hasil

transformasi. Dari matriks transformasi ini, diperoleh data titik-titik transformasi yang digunakan dalam proses transformasi perspektif citra.

### 3.3.4 Mendeteksi Kontur Warna Garis Marka

Proses deteksi kontur garis marka jalan melibatkan beberapa langkah. Setelah mendapatkan data warna dari segmentasi warna HSV, dilakukan pembuatan *mask* untuk warna kuning dan putih. Operasi *dilasi* dan *bitwise AND* digunakan untuk menghubungkan kontur dan memisahkan marka jalan.

### 3.3.5 Klasifikasi Garis Marka Kuning dan Putih

Tahap klasifikasi garis marka melibatkan penggambaran kontur marka kuning dan putih pada citra asli. Teks "Yellow Lane" dan "White Lane" ditambahkan untuk memberikan informasi tentang jenis garis marka yang terdeteksi dan membantu mengidentifikasi jenis garis marka jalan.

### 3.3.6 Membuat Titik Panduan Kendaraan

Berikutnya, dilakukan pencarian nilai titik tengah pada garis marka (garis marka simulasi) sebagai acuan kendaraan. Nilai ini digunakan untuk menentukan offset dan titik panduan kendaraan terhadap garis marka jalan. Titik tengah purwarupa garis marka dihitung dengan mengambil rata-rata koordinat horizontal ( $x$ ) dari garis marka kuning dan putih pada purwarupa. Dengan nilai titik tengah purwarupa, kita dapat menghitung offset kendaraan dari garis marka jalan.

### 3.3.7 Membuat Fungsi Peringatan Pelanggaran

Dalam tahap memasukkan peringatan pelanggaran marka jalan, dilakukan pengecekan apakah titik depan tengah kendaraan berinteraksi dengan kontur garis marka jalan yang terdeteksi. Jika terjadi interaksi, akan ditampilkan notifikasi "WARNING" yang menunjukkan pelanggaran garis marka jalan oleh kendaraan. Jika tidak ada interaksi, akan ditampilkan notifikasi "AMAN" yang menunjukkan bahwa kendaraan berada di jalur yang benar.

## 3.4 Pengujian Sistem

Pengujian model dilakukan untuk memverifikasi kinerja dan akurasi deteksi marka, pemberian titik panduan, serta identifikasi pelanggaran marka jalan. Pengujian ini melibatkan penggunaan dataset video jalan tol dan simulasi melalui ROS Gazebo. Dataset video jalan tol digunakan untuk menguji model dalam kondisi lapangan yang nyata. Model dijalankan pada video jalan tol untuk menguji kemampuannya dalam mengenali marka, memberikan titik panduan, dan mendeteksi pelanggaran marka jalan dalam situasi sebenarnya. Selain itu, pengujian juga dilakukan melalui simulasi menggunakan ROS Gazebo. Tujuan utamanya adalah memastikan bahwa model yang dikembangkan mampu memberikan hasil deteksi marka jalan yang akurat dan konsisten.

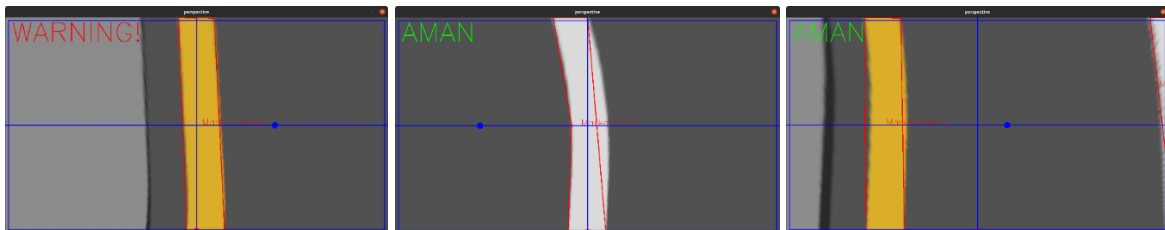
## 4. Hasil dan Analisa

### 4.1 Hasil dan Analisis Deteksi Garis Marka Jalan

Pengujian hasil deteksi marka jalan melibatkan parameter dari proses pengolahan citra, termasuk deteksi marka, posisi dalam lajur jalan, dan sudut kendaraan. Terdapat 13 percobaan dengan variasi parameter posisi dan sudut yang berbeda, dengan menggunakan dua sampel marka yaitu garis kuning dan garis putih. Tabel berikut menunjukkan persentase sampel yang berhasil terdeteksi sebagai garis marka jalan, berdasarkan variasi posisi dan sudut kamera.

**Tabel 2.** Pengambilan Data

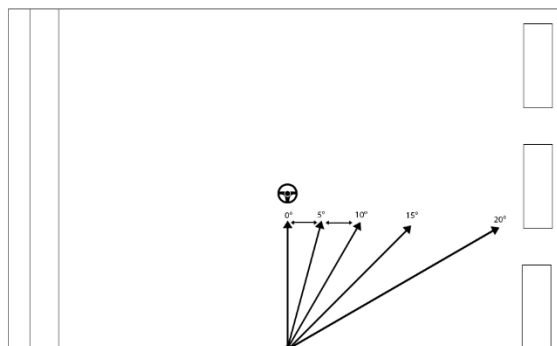
Posisi dan Sudut	Persentase Sample yang Terdeteksi Sebagai Marka Jalan
Posisi 0 cm, Sudut 0	50%
Posisi 0 cm, Sudut 20	100%
Posisi 5 cm, Sudut 0	100%
Posisi 5 cm, Sudut 10	100%
Posisi 5 cm, Sudut -10	100%
Posisi 7 cm, Sudut 0	100%
Posisi 7 cm, Sudut 10	100%
Posisi 7 cm, Sudut -10	100%
Posisi 10 cm, Sudut 0	100%
Posisi 10 cm, Sudut 10	100%
Posisi 10 cm, Sudut -10	100%
Posisi 15 cm, Sudut 0	50%
Posisi 15 cm, Sudut -20	100%



**Gambar 3.** Pengujian Deteksi Marka Jalan

#### 4.2 Hasil dan Analisis Titik Panduan Kendaraan

Pengujian titik panduan kendaraan melibatkan penggunaan parameter dari proses pengolahan citra, termasuk titik tengah lajur jalan, titik tengah kendaraan, dan variasi sudut kendaraan. Pengambilan data titik panduan melibatkan parameter posisi sudut, selisih nilai piksel x sebelum dan setelah perubahan sudut posisi, serta persentase error. Kalibrasi piksel dan sudut derajat digunakan untuk menghubungkan lokasi piksel pada layer dengan nilai sudut derajat yang sesuai. Gambar berikut mengilustrasikan sudut pengambilan data kendaraan dan titik panduan.



**Gambar 4.** Sudut Pengambilan Data Kendaraan dan Titik Panduan

Hasil percobaan deteksi titik panduan kendaraan menunjukkan rata-rata nilai error setelah 24 percobaan sebesar 3,1337%. Persentase error tertinggi terjadi pada kendaraan dengan sudut 4° (8,4% error), -14° (7,8% error), dan 20° (4,2% error). Sementara itu, persentase error terendah terjadi pada sudut -22° (0,187% error), 22° (0,5627% error), dan -22° (0,5628% error).

### 4.3 Hasil dan Analisis Fungsi Peringatan Pelanggaran Marka

Hasil deteksi fungsi peringatan pelanggaran marka menampilkan notifikasi "WARNING" saat marka kuning dilanggar dan "AMAN" saat marka tidak dilanggar. Dalam hal ini, dilakukan pengecekan interaksi antara titik depan tengah kendaraan dengan kontur garis marka jalan yang terdeteksi. Jika terjadi interaksi atau melewati antara titik kendaraan dan marka kuning, akan dianggap sebagai pelanggaran marka, dan sebaliknya akan dianggap aman.



Gambar 5. Pengujian Peringatan Pelanggaran Marka

### 4.4 Hasil dan Analisis Deteksi Garis Marka Jalan Berdasarkan *Dataset Real*

Pengujian sistem deteksi marka jalan dilakukan secara komprehensif dengan 9 uji coba. Setiap uji coba menggunakan dataset nyata yang mengakomodasi variasi kecepatan dari 9 variabel yang berbeda. Tujuannya adalah menguji performa sistem dalam mengatasi berbagai tingkat kecepatan, untuk memastikan kemampuan sistem dalam bekerja akurat dan konsisten dalam berbagai kondisi nyata yang beragam.

Tabel 3. Pengujian Sistem pada *Dataset Real*

Kecepatan	Total FrameVideo	Frame Error	Nilai Error
60 km	2029 frame	13 frame	0,640 %
65 km	2303 frame	139 frame	6,03 %
70 km	2549 frame	170 frame	6,66 %
75 km	2035 frame	76 frame	3,70 %
80 km	2061 frame	77 frame	3,73 %
85 km	2144 frame	85 frame	3,964 %
90 km	2092 frame	69 frame	3,298 %
95 km	1824 frame	285 frame	15,625 %
100 km	2932 frame	466 frame	15,893 %
60 km	2029 frame	13 frame	0,640 %
65 km	2303 frame	139 frame	6,03 %
70 km	2549 frame	170 frame	6,66 %
75 km	2035 frame	76 frame	3,70 %



Gambar 6. Dataset Video Real Garis Marka



## 5. Kesimpulan

Pada hasil percobaan tugas akhir ini dalam pengujian simulasi garis marka jalan, deteksi marka kuning dan putih mencapai tingkat keberhasilan 85%, dengan beberapa percobaan menunjukkan tingkat deteksi yang lebih rendah. Sistem juga berhasil merancang titik panduan kendaraan sebagai referensi pengemudi, dengan rata-rata error 3,1337% dalam pengujian simulasi. Pendeteksian pelanggaran marka mencapai 100% dalam simulasi. Melalui pengujian komprehensif menggunakan dataset nyata pada berbagai tingkat kecepatan, sistem berhasil menunjukkan kinerja yang baik dan konsisten. Rata-rata error deteksi marka jalan dalam pengujian adalah 5,465%, dengan hasil yang bervariasi tergantung pada kecepatan kendaraan.

Untuk penelitian selanjutnya, disarankan untuk meningkatkan FPS dan resolusi pada kamera guna meningkatkan akurasi deteksi. Kalibrasi ulang kamera juga perlu dilakukan untuk hasil yang lebih akurat. Pertimbangkan tambahan algoritma deteksi dan fokus pada perbaikan sistem, terutama pada kecepatan tertentu dan lokasi khusus. Akhirnya, pengembangan sistem deteksi marka jalan penting untuk mendukung kendaraan otonom dan keselamatan di jalan.

## Referensi

- [1] A. Rahman Hakim, "Jasa Raharja: 61 Persen Penyebab Kecelakaan Karena Kelalaian Manusia," *www.liputan6.com*, Feb. 22, 2022. <https://www.liputan6.com/bisnis/read/4882287/jasa-raharja-61-persen-penyebab-kecelakaan-karena-kelalaian-manusia> (accessed Jan. 29, 2023).
- [2] Roderta, "Rancang Bangun Sistem Peringatan Keamanan Pada Mobil Berdasarkan Garis Markah Jalan Menggunakan Kamera," 2014.
- [3] Detik News, "Mengenal Toyota Safety Sense, Fitur Canggih yang Bikin Avanza Naik Kelas," *oto.detik.com*, Sep. 24, 2021. <https://oto.detik.com/mobil/d-5738965/mengenal-toyota-safety-sense-fitur-canggih-yang-bikin-avanza-naik-kelas> (accessed Feb. 08, 2023).
- [4] Deon Sampurna, "DETEKSI GARIS SEBAGAI PREDIKSI ARAH GERAK KENDARAAN BERBASIS PENGLIHATAN KOMPUTER DENGAN METODE HOMOGRAFI," *SPIE*, Jul. 2020.
- [5] B. Y. Suprpto, A. Ghaida, H. Hikmarika, and S. Dwijayanti, "Road and Vehicles Detection System Using HSV Color Space for Autonomous Vehicle," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 6, no. 1, p. 42, Jul. 2020, doi: 10.26555/jiteki.v16i1.16949.
- [6] M. Samuel, M. Mohamad, S. M. Saad, and M. Hussein, "Development of Edge-Based Lane Detection Algorithm using Image Processing," *JOIV: International Journal on Informatics Visualization*, vol. 2, no. 1, p. 19, Jan. 2018, doi: 10.30630/joiv.2.1.101.
- [7] Pravin T. Mandlik and Prof. A. B. Deshmukh, "Raspberry-Pi Based Real Time Lane Departure Warning System using Image Processing," *International Journal of Engineering Research and*, vol. V5, no. 06, Jun. 2016, doi: 10.17577/IJERTV5IS060725.
- [8] C. E. Chandra, H. Jufry, and S. Tan, "DETEKSI MARKA JALAN DAN ESTIMASI POSISI MENGGUNAKAN MULTIREOLUTION HOUGH TRANSFORM," 2013. Accessed: Jan. 29, 2023. [Online]. Available: <http://eprints.binus.ac.id/id/eprint/30679>
- [9] Muh. A. R. B. R. Haidar Azmi Rabbani, "Perbandingan Ruang Warna RGB dan HSV dalam Klasifikasi Kematangan Biji Kopi," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, pp. 2243–2248, Jun. 2021.
- [10] P. Ganesan, V. Rajini, B. S. Sathish, and K. B. Shaik, "HSV color space based segmentation of region of interest in satellite images," in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, IEEE, Jul. 2014, pp. 101–105. doi: 10.1109/ICCICCT.2014.6992938.
- [11] Andi Asvin Mahersatillah and Suradi, "Deteksi Jalan Berdasarkan Segmentasi Warna HSV Dalam Penerapan Mobil Otonom (Autonomous Car)," 2021.
- [12] Enrique Fernandez, Luiz Sanchez, Anil Mahtani, and Aaron Martinez, *Learning ROS for Robotics Programming second edition*. Birmingham B3 2PB: Packt Publishing Ltd. Livery Place, 2015.
- [13] Aaron Martinez and Enrique Fernandez, *Learning ROS for Robotics Programming*. Birmingham B3 2PB: Packt Publishing Ltd. Livery Place, 2013.



22 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).