

# PERBANDINGAN METODE DETEKSI TEPI MENGGUNAKAN OPERATOR CANNY, KIRSCH & MARR-HILDRETH (LoG) PADA DETEKSI MARKA JALAN

## COMPARISON OF EDGE DETECTION OPERATOR USING CANNY, KIRSCH & MARR-HILDRETH (LoG) FOR ROAD MARKING DETECTION

<sup>1</sup>Kuncoro Triandono Mukti, <sup>2</sup>Agus Virgono, Ir.,M.T., <sup>3</sup>Randy Erfa Saputra,S.T.,M.T.

<sup>1,2,3</sup>Program Studi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>kuncorotm@student.telkomuniversity.ac.id, <sup>2</sup>avirgono@telkomuniversity.ac.id,

<sup>3</sup>...@telkomuniversity.ac.id

---

### Abstrak

Self-driving atau Autonomous driving, Advanced Driving Assistance System (ADAS) merupakan salah satu topik yang banyak digemari dalam berbagai penelitian yang berhubungan dengan keamanan kendaraan. Salah satu teknologi yang sangat berguna pada autonomous driving yaitu deteksi jalur atau lane detection yang menggunakan marka membujur (e.g. garis putus dan utuh) sebagai acuan agar kendaraan dapat berjalan tetap pada jalurnya. Berbagai operator pada deteksi tepi diajukan untuk mendapatkan hasil akurasi deteksi semaksimal mungkin. Namun, pergerakan marka garis antar frame akan bervariasi tergantung pada kecepatan kendaraan. Jika sistem gagal mendeteksi marka garis pada kecepatan tinggi, akan mengakibatkan sistem autonomous driving salah mengambil keputusan. Pada penelitian ini, akan dilakukan analisis perbandingan kemampuan operator Canny, Laplacian of Gaussian (Marr-Hildreth) dan Kirsch pada metode deteksi tepi untuk mendeteksi marka garis putus – putus pada kecepatan yang bervariasi. Hasil pengujian menunjukkan jika seluruh operator berhasil mencapai target minimum pendeteksian yaitu 80% dan didapatkan operator terbaik untuk pendeteksian marka garis yaitu Kirsch dengan persentase tertinggi pada kecepatan 30, 50 dan 80 km/j.

**Kata kunci :** Deteksi Tepi, Perbandingan, Kecepatan, Garis putus-putus

---

### Abstract

Self-driving or Autonomous driving, Advanced Driving Assistance System (ADAS) is one of the most popular topics in research related to vehicle safety. One of the most useful technologies in autonomous driving is lane detection that uses longitudinal marks (e.g. straight and dashed lines) as a reference to keep the vehicle running on lane. Various operators on edge detection are proposed to obtain the best accuracy of lane detection. However, the movement of the line marks between frames will vary depending on the speed of the vehicle. If the system fails to detect the line marker at high speed, will cause the autonomous driving system make a wrong decision. In this final task, we will perform a comparison analysis of Canny, Laplacian of Gaussian (Marr-Hildreth) and Kirsch's ability on edge detection methods to detect dashed line marks at varying speeds. The results showed that all operators succeeded in achieving the minimum detection target of 80% and obtained the best operators for line marker detection is Kirsch with the highest percentage at all speeds 30, 50 and 80 km/h.

**Keywords :** Edge Detection, Comparison, Speed, Dashed Line

---

### 1. Pendahuluan

Kecelakaan lalu lintas selalu menjadi masalah utama saat berkendara oleh karena itu banyak penelitian yang dilakukan untuk meningkatkan sistem keselamatan pada kendaraan sebagai solusi untuk mengurangi risiko kecelakaan lalu lintas diantaranya yaitu Self-driving atau Autonomous driving, Advanced Driving Assistance System (ADAS), Driving Assistance System (DAS). Salah satu teknologi yang sangat berguna pada sistem-sistem tersebut yaitu deteksi jalur atau lane detection [11] yang menggunakan marka membujur (e.g. garis putus dan utuh) sebagai acuan agar kendaraan tetap berjalan pada jalurnya.

Tujuan utama penelitian – penelitian sebelumnya terfokus pada dua topik [11]. Salah satunya adalah pendeteksian jalur secara efektif menggunakan berbagai metode pengolahan citra, seperti metode yang digunakan pada penelitian [1-4], [8], [9] yaitu edge detection. Berbagai operator pada edge detection diajukan untuk mendapatkan hasil akurasi deteksi semaksimal mungkin. Namun, pergerakan marka garis antar frame akan bervariasi tergantung pada kecepatan kendaraan. Tidak seperti garis utuh, garis putus – putus akan terlihat menyatu saat kecepatan kendaraan semakin tinggi. Pada penelitian [11] metode yang diajukan mendapatkan akurasi sebesar 96.84% pada kecepatan [5-45] km/j. Sedangkan pada penelitian [1-2], [4-5], [7-10], [12-17] tidak ada yang menjelaskan pada kecepatan berapa metode diuji, pada kasus tertentu mungkin saat kecepatan 45 km/j akurasi lane detection mencapai 90% tetapi pada kecepatan 80 km/j metode yang diajukan belum tentu dapat mendeteksi marka

garis tersebut. Jika sistem gagal mendeteksi marka tersebut pada kecepatan tinggi, akan mengakibatkan sistem autonomous driving salah mengambil keputusan.

Perancangan sistem untuk membandingkan operator deteksi tepi berdasarkan kecepatan kendaraan dapat dilakukan untuk menganalisis dan mengetahui kemampuan setiap operator saat mendeteksi marka garis putus – putus pada kecepatan yang bervariasi.

## 2. Perancangan

### 2.1. Perancangan Data Uji

Pengujian yang dilakukan bertujuan untuk menguji apakah algoritma pendeteksian marka garis putus yang digunakan mampu untuk mendeteksi marka pada masukkan video yang berisi sejumlah marka dengan persentase yang sudah ditentukan. Video tersebut diambil dengan menggunakan kecepatan kendaraan yang berbeda-beda yaitu pada kecepatan 30, 50, 80 km/j dengan tujuan untuk mengetahui sampai pada kecepatan berapakah sistem mampu mendeteksi marka garis putus. Kemudian pada video tersebut juga digunakan resolusi yang berbeda, untuk mengetahui apakah penggunaan resolusi yang berbeda akan mempengaruhi hasil deteksi marka garis. Ringkasan pengujian dapat dilihat pada tabel 2.1.

Tabel 2.1. Ringkasan data pengujian

Tempat/Lokasi	Toll / Jalan Raya		
Cuaca	Cerah		
Kecepatan (km/j)	30	50	80
Resolusi	320 x 180 px		800 x 450 px
Camera	Action Cam		
FOV (Field of View)	±120°		

### 2.2. Pengambilan Data Uji

Lokasi pengambilan data uji berada di Jalan Asera Boulevard, Harapan Indah II, Bekasi Utara. Jarak yang ditempuh yaitu 147,4 Meter pada kecepatan 30, 50 & 80 km/j. Cuaca cerah pukul 5 sore untuk kecepatan 30 dan 50 km/j, sedangkan untuk kecepatan 80 km/j dalam cuaca berawan pada pukul 7 pagi. Sesuai dengan perancangan yang telah dilakukan proses pemasangan kamera dilakukan pada bagian kaca diatas *dashboard* yang dapat dilihat pada gambar 2.1.

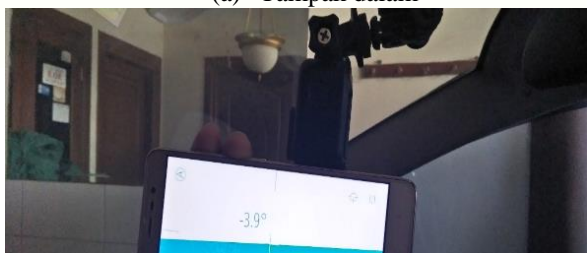
Untuk mengurangi pengambilan objek yang tidak diperlukan seperti cap mobil kamera akan dimiringkan keatas sebesar  $\pm 3,9^\circ$  jika kamera dimiringkan kebawah sebesar  $\pm 15^\circ$  seperti perancangan pada sub bab sebelumnya akan membuat cap mobil masuk kedalam *frame* dan mengurangi garis marka yang terlihat, agar tidak terdapat distorsi seperti efek *fisheye* pada gambar yang terrekam, kamera diatur agar mendapatkan gambar yang baik dan fokus terhadap marka garis, pada gambar 2.2 ditampilkan hasil pengambilan data uji.



(a) Tampak dalam



(b) Tampak samping



(c) Kemiringan kamera



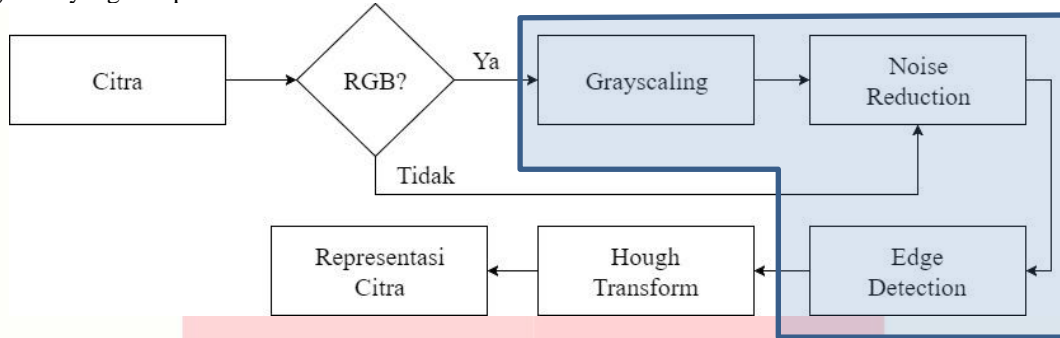
(d) Hasil pengambilan data uji

Gambar 2.1. Posisi pemasangan kamera

## 2.3. Perancangan Sistem Pendeteksian Marka

### 2.1.1 Sistem Pendeteksian

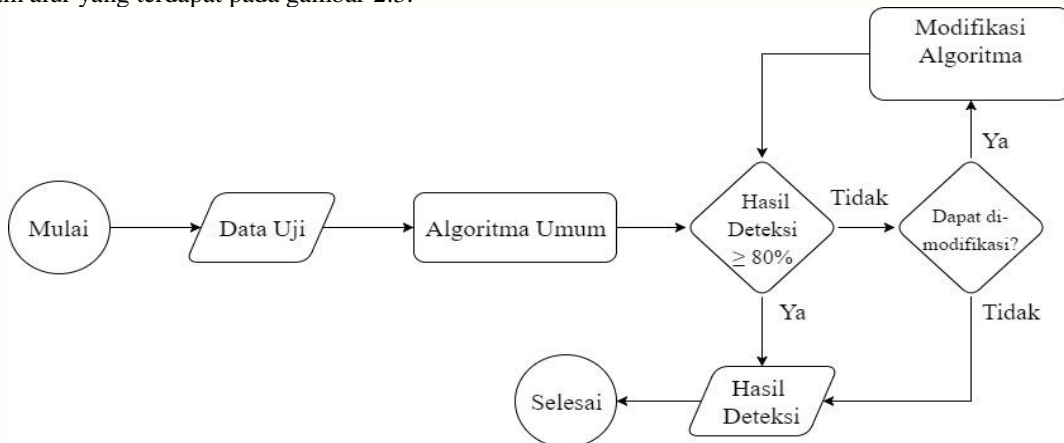
Sistem deteksi marka yang dirancang pada penelitian ini akan menghasilkan 3 sistem dengan operator deteksi tepi yang berbeda disetiap sistemnya, sistem pertama akan menggunakan Operator Canny pada proses deteksi tepi, selanjutnya Marr-Hildreth (LoG), dan yang terakhir Kirsch. Dapat dilihat pada gambar 2.2 tahap pendeteksian marka garis putus secara umum akan diterapkan pada setiap operator. Sistem ini dijalankan pada setiap *frame* yang didapatkan dari video secara *real-time*.



Gambar 2.2. Diagram blok algoritma pendeteksian umum

### 2.1.2 Modifikasi Sistem Pendeteksian

Pada bagian ini akan dijelaskan perancangan terhadap tahap modifikasi algoritma pendeteksian jika algoritma umum tidak dapat mencapai target yang telah ditentukan. Proses modifikasi algoritma ditampilkan pada diagram alur yang terdapat pada gambar 2.3.



Gambar 2.3. Diagram alur proses modifikasi algoritma pendeteksian

### 2.1.3 Perancangan Pengujian

Tujuan dari pengujian sistem yang telah dibangun yaitu untuk menghitung marka yang berhasil dideteksi dan membandingkan dengan jumlah marka garis sebenarnya. Pengujian dilakukan dengan menjalankan sistem deteksi pada masukan berupa video dan performansi sistem deteksi bisa didapatkan dengan membandingkan keluaran sistem deteksi dengan data posisi serta banyaknya objek marka yang telah diketahui sebelumnya. Oleh karena itu, untuk membandingkan keluaran sistem dengan data sebenarnya maka digunakan metode *Confusion Matrix* dengan menggunakan perhitungan *precision-recall* yang dapat dilakukan menggunakan persamaan 2.1 dan 2.2. Kemampuan sistem dikatakan “Baik” jika persentase *precision-recall*  $\geq 80\%$ .

$$Precision = \frac{TP}{TP + FP} \times 100 \quad (2.1)$$

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (2.2)$$

Dimana:

- *Precision*: Tingkat ketepatan antara marka garis yang terdeteksi dengan marka garis sebenarnya (%)
- *Recall*: Tingkat keberhasilan sistem dalam mendeteksi marka garis (%)
- TP: Jumlah deteksi yang benar
- FP: Jumlah deteksi yang salah
- FN: Jumlah objek (marka garis) yang tidak terdeteksi

Berbeda dengan marka garis utuh semakin jauh marka garis putus yang tertangkap kamera maka akan membuat garis semakin kecil dan seperti menyatu. Karena perhitungan yang dilakukan mengandalkan pengelihan manusia untuk mengurangi peluang terjadi kesalahan perhitungan maka perhitungan marka garis yang terdeteksi akan dibatasi. Ilustrasi permasalahan dan solusi dapat dilihat pada gambar 2.4.

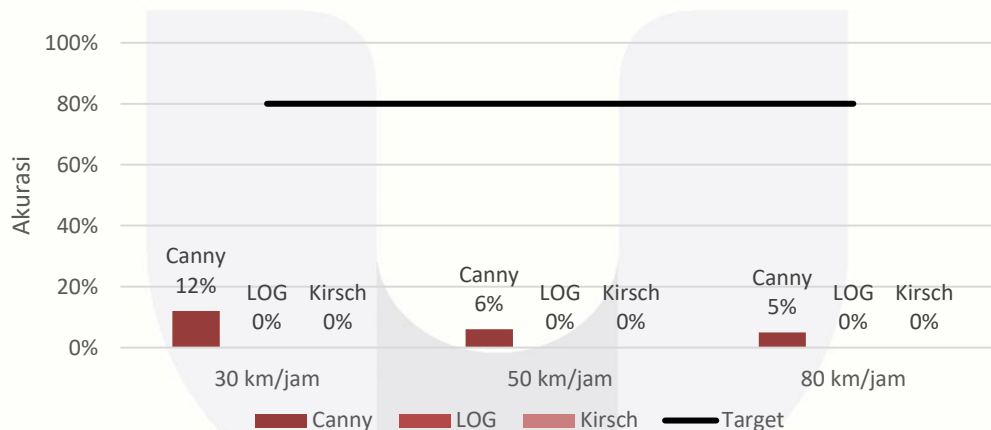


Gambar 2.4. Ilustrasi permasalahan dan solusi perhitungan

### 3. Pengujian

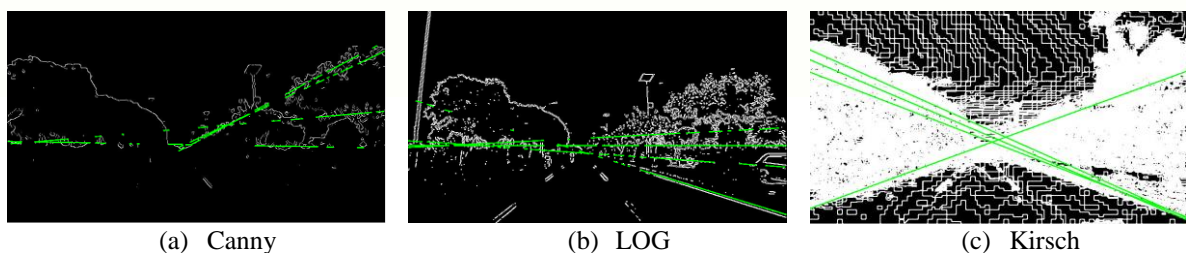
Pengujian ini dilakukan dengan cara simulasi secara *real-time*, simulasi dilakukan dengan cara memutar video jalanan yang telah didapatkan pada aplikasi SparkoCam Video yang berguna untuk menggantikan kamera laptop, selanjutnya sistem akan membaca adaptor SparkoCam video sebagai kamera laptop jika kamera dijalankan maka kamera akan menampilkan video jalanan seakan sedang merekam jalanan pada video tersebut. Pengujian akan dilakukan sebanyak 3 kali, pengujian pertama bertujuan untuk mengetahui kemampuan algoritma umum. Pengujian kedua dilakukan untuk mengetahui kemampuan pada algoritma modifikasi, sedangkan pada pengujian ketiga dilakukan untuk mengetahui pengaruh resolusi terhadap hasil pendeteksian. Video data uji memiliki spesifikasi sebagai berikut, 3 video berbeda dengan kecepatan 30, 50 dan 80 km/j dengan jarak 147.4 meter, resolusi masing-masing video 800x450 piksel dengan 30 fps. Inisialisasi nilai pada setiap metode menggunakan pengaturan *default* MATLAB, Nilai yang akan dibandingkan yaitu waktu komputasi, persentase *recall* dan *precision*.

#### 3.1. Pengujian Sistem Pendeteksian



Gambar 3.1. Perbandingan hasil pendeteksian algoritma umum

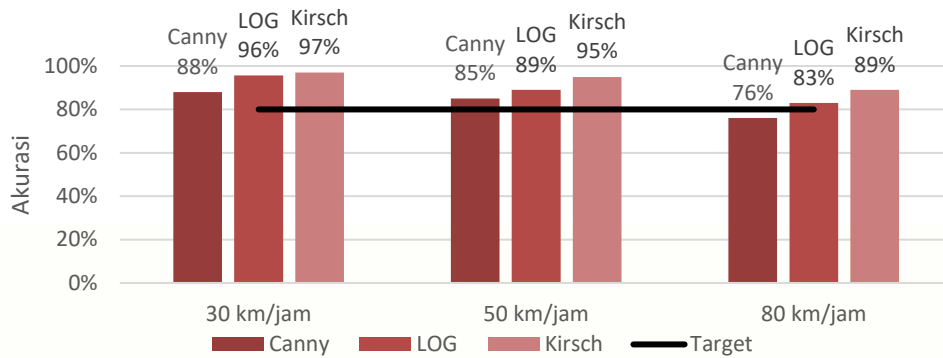
Dapat dilihat pada gambar 3.1, tidak ada operator yang berhasil mencapai minimum hasil pendeteksian sebesar 80%, rendahnya persentase hasil deteksi disebabkan oleh hasil deteksi tepi masing – masing operator yang lebih dominan kepada pepohonan dibandingkan dengan jalanan seperti yang ditampilkan pada gambar 3.2. Hal ini mengakibatkan proses voting objek berbentuk garis pada transformasi hough lebih dominan terhadap garis yang berada di pepohonan dari pada marka garis yang berada di jalanan seperti yang ditampilkan pada gambar 3.2.



Gambar 3.2. Hasil deteksi tepi frame ke-5 dan voting transformasi hough

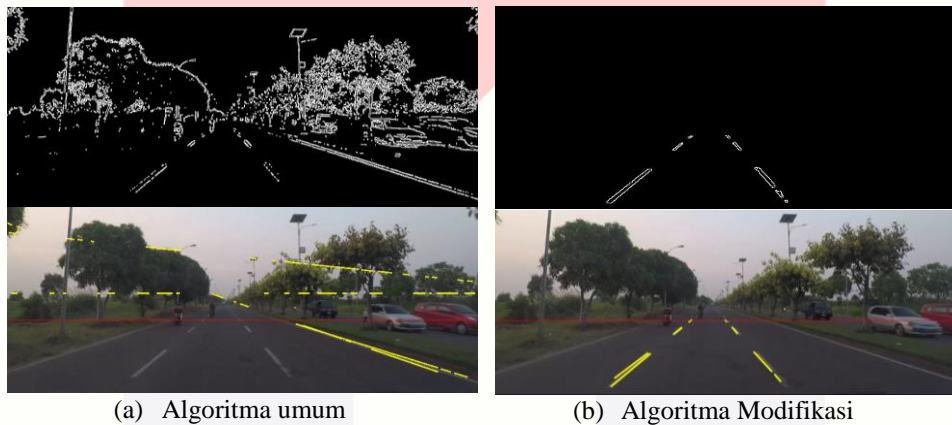


### 3.2. Pengujian Modifikasi Sistem Pendeteksian



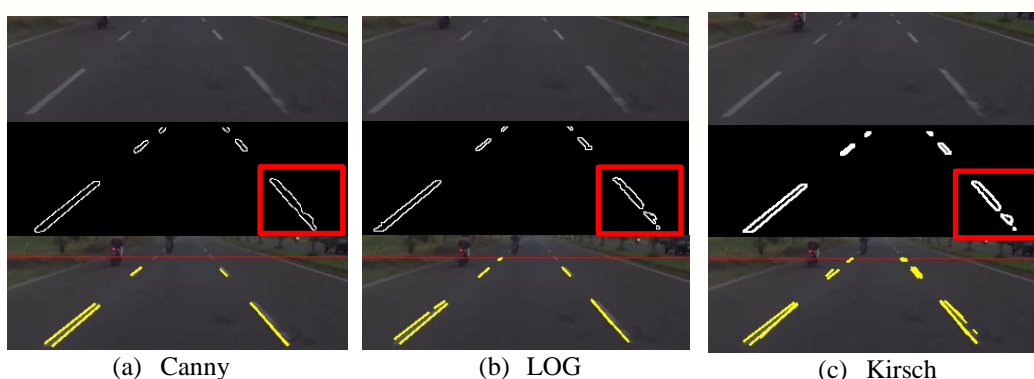
Gambar 3.3. Perbandingan hasil pendeteksian algoritma modifikasi

Berbeda dengan algoritma umum, dilakukan beberapa modifikasi pada algoritma pendeteksian yaitu peningkatan kontras, pencarian piksel putih dan *Region of Interest* (ROI) yang digunakan untuk membatasi area pengolahan transformasi hough pada area jalanan saja, dapat dilihat pada gambar 4.5.b hasilnya transformasi hough hanya mendeteksi marka garis sebagai objek garis dibandingkan dengan algoritma umum hasil transformasi hough lebih dominan pada garis dipepohonan ini yang menyebabkan akurasi pendeteksian meningkat.



Gambar 3.4. Perbandingan hasil deteksi tepi algoritma umum dan modifikasi

Pada grafik perbandingan hasil deteksi, operator canny berhasil mendeteksi 88% pada 30 km/j, 85% pada 50 km/j dan 76% pada 80 km/j sedangkan operator LOG berhasil mendeteksi 96% pada kecepatan 30 km/j, 89% pada 50 km/j dan 83% pada 80 km/j lalu operator Kirsch memiliki hasil tertinggi pada seluruh kecepatan yaitu 97% pada kecepatan 30 km/j, 95% pada 50 km/j dan 89% pada kecepatan 80 km/j. Semua operator mencapai target minimum pendeteksian kecuali operator canny pada kecepatan 80 km/j, pada kecepatan 80 km/j operator Canny hanya dapat mendeteksi 76% dari total marka garis hasil ini membuat canny tidak memenuhi target minimum pendeteksian, berdasarkan teori operator canny memiliki kelebihan yaitu menghasilkan tingkat kesalahan tepi yang rendah dan hanya memberikan satu tanggapan untuk satu tepi sedangkan operator LOG memiliki kelebihan tidak terpengaruh oleh derau dan operator Kirsch menghasilkan tepi dari 8 arah mata angin yang mengakibatkan hasil deteksi terlalu tebal [18].



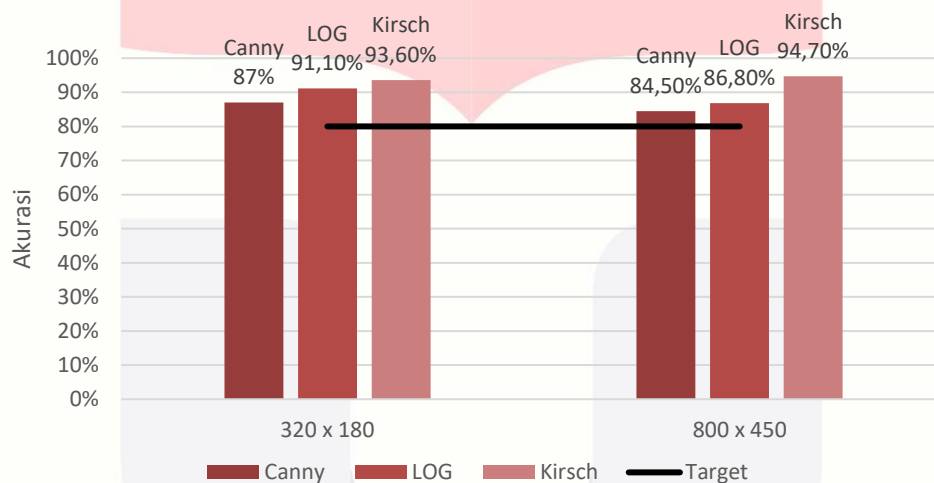
Gambar 3.5. Perbandingan hasil deteksi tepi algoritma modifikasi

Dapat dilihat pada gambar 3.5, hasil ekstraksi tepi masing-masing operator sesuai dengan teori diatas, jika dibandingkan secara visual hasil pendeteksian Canny, LOG dan Kirsch terlihat hampir sama, namun dengan menggunakan `syntax nnz(edge_detection)` dapat diketahui jumlah piksel tepi pada suatu citra, operator Canny menghasilkan ekstraksi tepi sebanyak 494 piksel, operator LOG sebanyak 575 dan Kirsch sebanyak 1100, dengan jumlah tepi yang lebih sedikit dari pada operator lain Canny dapat mengekstraksi tepi pada marka garis yang pudar sedangkan operator LOG menghasilkan deteksi tepi yang salah pada marka garis pudar dan operator Kirsch menghasilkan tepi yang hampir menutupi marka garis.

Penyebab kecilnya hasil pendeteksian Canny disebabkan oleh kelebihanannya menghasilkan tepi yang sangat tipis dengan tingkat kesalahan yang kecil walaupun hasil deteksi tepinya sangat baik tetapi pada kecepatan 80 km/j marka garis terakhir (garis yang bersilangan dengan garis batas berwarna merah) pada semua *frame* tidak dianggap sebagai objek garis oleh transformasi hough sedangkan hasil deteksi tepi LOG dan Kirsch yang hampir menutupi marka garis dianggap sebagai garis. Hasil pengujian ini sesuai dengan teori yang mengatakan berbagai teknik deteksi tepi bekerja dengan cara yang berbeda, masing-masing memiliki kelebihan (crane, 1997) itulah sebabnya eksperimen pada suatu aplikasi dengan menggunakan berbagai teknik deteksi tepi perlu dilakukan untuk mendapatkan hasil (operator) yang terbaik [18].

Berdasarkan hasil analisis diatas maka didapatkan kesimpulan jika aplikasi pendeteksian marka garis membutuhkan hasil deteksi tepi yang tegas seperti hasil deteksi tepi Kirsch. Jika diurutkan berdasarkan hasil pendeteksian tertinggi operator terbaik adalah Kirsch, LOG, Canny.

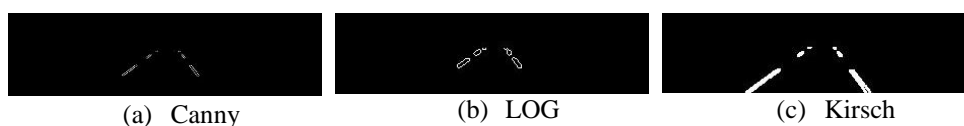
### 3.3. Pengujian Resolusi



Gambar 3.6. Perbandingan hasil pendeteksian resolusi 320x180 dan 800x450

Pada gambar 4.10 ditampilkan perbandingan akurasi antara setiap operator pada resolusi yang berbeda. Pada resolusi 800x450 piksel operator Canny dan LOG memiliki nilai *precision* 100% dimana hasil deteksi garis tidak mengalami kesalahan pendeteksian pada objek lain, sedangkan Kirsch mendapatkan nilai *precision* 94,3% yang berarti jika hasil deteksi kirsch terdapat kesalahan pendeteksian objek, sesuai dengan teori yang dijelaskan pada buku teori dan aplikasi pengolahan citra [18] jika hasil deteksi canny dan LOG menghasilkan tepi yang lebih baik dibandingkan dengan Kirsch dikarenakan operator Kirsch menghasilkan tepi yang sangat tebal dan memungkinkan terjadinya kesalahan pendeteksian dan membuat sistem salah mendeteksi garis hal ini yang menyebabkan operator Canny dan LOG dapat mendeteksi dengan baik walaupun dengan kualitas citra yang menurun.

Pada resolusi 320x180 piksel operator Canny meningkat sebesar 2,5% dari resolusi 800x450 piksel begitu juga dengan operator LOG yang meningkat sebesar 4,2%, sedangkan operator Kirsch mengalami penurunan akurasi sebesar 1,1% walaupun mengalami penurunan operator Kirsch tetap memiliki akurasi pendeteksian lebih tinggi dibandingkan operator lainnya, tingginya akurasi disebabkan oleh hasil ekstraksi tepi pada operator Kirsch sangat tebal dan menutupi marka garis dapat dilihat pada gambar 4.11 perbandingan hasil ekstraksi tepi operator Canny, LOG dan Kirsch. Berdasarkan analisis diatas maka operator Canny, LOG dan Kirsch dapat digunakan pada saat resolusi 320x180 piksel dengan hasil akurasi tertinggi pada operator Kirsch.



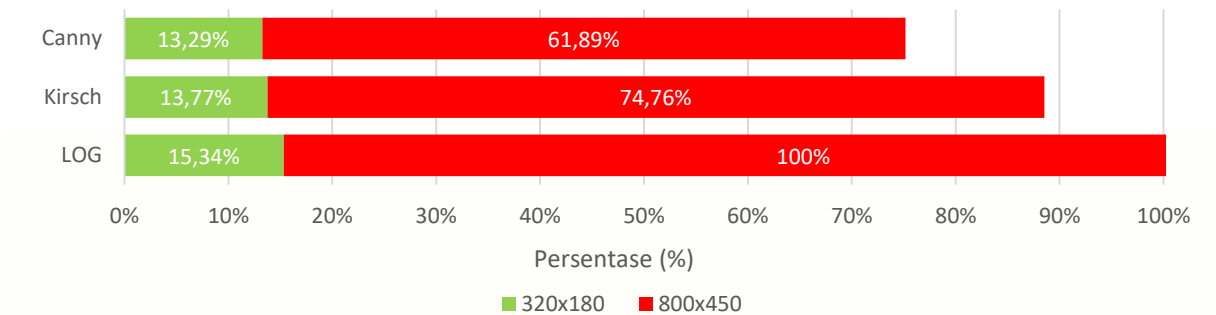
(a) Canny

(b) LOG

(c) Kirsch

Gambar 3.7. Perbandingan hasil ekstraksi tepi pada resolusi 320x180 piksel

Resolusi 320x180 piksel dapat mereduksi waktu komputasi pada resolusi 800x450 piksel sebesar 78% pada operator Canny, 84,7% pada operator LOG dan 81,6% pada operator Kirsch. Dapat dilihat pada gambar 3.8 grafik perbandingan waktu antara resolusi 800x450 piksel dan resolusi 320x180 piksel yang ditampilkan dengan acuan waktu komputasi terlama yaitu LOG dalam skala persentase (%).



Gambar 3.8. Perbandingan waktu komputasi pada resolusi

Berdasarkan hasil analisis diatas didapatkan kesimpulan jika pada resolusi 320x180 piksel operator deteksi tepi masih dapat mendeteksi marka garis bahkan meningkatkan akurasi pendeteksian pada beberapa operator dan memiliki waktu komputasi yang sangat cepat. Jika menggunakan resolusi 320x180 piksel operator Canny dapat mendeteksi  $\frac{1 \text{ detik}}{0,0576 \text{ detik}} = 17 \text{ frame/detik}$ , operator LOG 15 *frame/detik* dan Kirsch dapat mendeteksi 16 *frame/detik*.

#### 4. Kesimpulan

- Pengaplikasian deteksi tepi pada pendeteksian marka jalan membutuhkan hasil ekstrasi tepi yang tegas dan tebal agar transformasi hough tidak salah mengenali objek garis.
- Operator Kirsch merupakan operator yang paling cocok digunakan untuk pendeteksian marka jalan hal ini dikarenakan hasil dan waktu komputasinya mengungguli operator Canny dan Log
- Operator tercepat yaitu Kirsch, LOG, Canny.
- Setiap operator memiliki cara dan Teknik masing-masing untuk mengekstrak citra oleh karena itu dibutuhkan eksperimen untuk mencari tahu operator terbaik yang cocok untuk bidang tersebut.
- Jika sistem akan diterapkan pada kehidupan sehari-hari maka dengan resolusi 320x180 piksel operator sudah mampu menyajikan informasi dengan *delay* hanya  $\pm 0.05$  detik.

## Daftar Pustaka

- [1] Yadi Li, Liguang Chen, Haibo Huang, Xiangpeng Li, Wenkui Xu, Liang Zheng, Jiaqi Huang. 2016. *Nighttime lane markings recognition based on Canny detection and Hough transform*. 2016 IEEE International Conference on Real-time Computing and Robotics.
- [2] Yassin Kortli. 2016. *Efficient Implementation of a Real-Time Lane Departure Warning System*. IEEE 2016 INTERNATIONAL IMAGE PROCESSING APPLICATIONS AND SYSTEMS CONFERENCE 2016.
- [3] Danding Adhi Priutomo. 2016. *Simulasi dan Analisis Sistem Smart Traffic Light Berbasis Pengolahan Citra Digital Dengan Metode Deteksi tepid an Segmentasi*. e-Proceeding of Engineering: Vol.3, No.1.
- [4] Ari Purno Wahyu Wibowo, Ardiles Sinaga. 2016. *Implementasi Teknologi LDWS Pada Transportasi sebagai pendukung sarana dan keamanan berkendara*. Seminar Nasional Telekomunikasi dan Informatika 2016 (SELISIK 2016).
- [5] Shabana Habib, Mohammad A Hannan. 2013. *Lane Departure Detection and Transmisson using Hough Transform Method*. Journal of Electronic Review: Vol 89, No. 5.
- [6] Mohsen Sharifi, Mahmoud Fathy, Maryam Tayefeh Mahmoudi. 2002. *A Classified and Comparative Study of Edge Detection Algorithms*, International Conference on Information Technology: Coding and Computing.
- [7] Worawit Phueakjeen, Nattha Jindapetch, Leang Kuburat, and Nikom Suvanvorn. 2011. *A Study of the Edge Detection for Road Lane*. The 8th Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand - Conference 2011.
- [8] Priyadarshi Kanungo, Sandeep Kumar Mishra, Subhajyoti Mahapatra, Uddesh Ranjan Sahoo, Uma Shankar Kr. Sah and Varsha Taunk. 2014. *Road Scene Segmentation Using Thresholding*. International Journal of Scientific & Engineering Research, Volume 5, Issue 5, May-2014. Tahun 2014.
- [9] Oki Januar Insani Mulyana. 2016. *Implementasi Lane Detection dengan metode Hough Transform untuk penilaian mengemudi berdasarkan marka jalan (Studi kasus Sukses Mandiri)*. Jurnal Ilmiah Komputer dan Informatika (KOMPUTA).
- [10] Mohammed Aly. 2008. *Real time Detection of Lane Markers in Urban Streets*. Intelligent Vehicles Symposium, 2008 IEEE.
- [11] Danilo Cáceres Hernández, Laksono Kurnianguro, Alexander Filonenko and Kang Hyun Jo. 2016. *Real-Time Lane Region Detection Using a Combination of Geometrical and Image Features*. Sensors.
- [12] Seongrae Kim, Junhee Lee, and Youngmin Kim. 2016. *Speed-Adaptive Ratio-Based Lane Detection Algorithm for Self-Driving Vehicles*. ISSOC 2016.
- [13] Pallavi V. Ingale, Prof. K. S. Bhagat, 2016. *Comparative Study of Lane Detection Techniques*. International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), Volume: 4 Issue: 5.
- [14] Jamel Baili1, Mehrez Marzougui1, Ameer Sboui, Samer Lahouar, Mounir Hergli, J.Subash Chandra Bose, Kamel Besbes, 2016. *Lane Departure Detection Using Image Processing Techniques*. 2017 2nd International Conference on Anti-Cyber Crimes (ICACC).
- [15] Yassin Kortli, Mehrez Marzougui, Belgacem Bouallegue, J. Subash Chandra Bose, Paul Rodrigues, Mohamed Atri. *A Novel Illumination-Invariant Lane Detection System*. 2nd International Conference on Anti-Cyber Crimes (ICACC).
- [16] Jie Guo, Zhihua Wei, Duoqian Miao. 2015. *Lane Detection Method Based on Improved RANSAC Algorithm*. IEEE Twelfth International Symposium on Autonomous Decentralized Systems.
- [17] Amjad Jaleel Humaidi, Mohammed Abdurraheem Fadhel. 2016. *Performance Comparison for Lane Detection and Tracking with Two Different Techniques*. Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA).
- [18] Abdul Kadir & Adhi Susanto, Teori dan Aplikasi Pengolahan Citra. Andi, Yogyakarta, 2013.