

## DETEKSI KEBERADAAN RAMBU PERINTAH DI JALAN MENGGUNAKAN *EUCLIDEAN COLOR FILTERING*

Rizal Pebianto<sup>1</sup>, Adriana Fanggalda<sup>2</sup> dan Yulianto Triwahyuadi Polly<sup>3</sup>

<sup>1,2,3</sup>Program Studi Ilmu Komputer, Fakultas Sains dan Teknik, Universitas Nusa Cendana,  
Jl. Adisucipto Penfui, Kupang, Indonesia

<sup>1</sup>Email: [rizalpebianto@gmail.com](mailto:rizalpebianto@gmail.com)

<sup>2</sup>Email: [adrianafanggalda@staf.undana.ac.id](mailto:adrianafanggalda@staf.undana.ac.id)

<sup>3</sup>Email: [yuliantopolly@staf.undana.ac.id](mailto:yuliantopolly@staf.undana.ac.id)

### ABSTRAK

Rambu perintah adalah salah satu jenis dari rambu lalu lintas yang sering kita jumpai saat berkendara di jalan, dan digunakan untuk menyatakan beberapa perintah saat berkendara. Ciri khas rambu perintah adalah memiliki warna latar biru. Kelalaian pengendara dalam memperhatikan keberadaan rambu perintah di jalan dapat berakibat pada terjadinya kecelakaan. Perlu adanya sebuah sistem yang dapat mendeteksi keberadaan rambu perintah di jalan sebagai sebuah solusi. Proses deteksi rambu perintah menggunakan metode *Euclidean color filtering* untuk memfilter warna dan BLOB untuk mencari piksel-piksel yang saling terhubung. Pada tahap filter warna dengan *Euclidean color filtering* digunakan nilai pusat warna RGB (10,110,200) dan rentang radius [90, 150]. Sistem diuji pada data primer sebanyak 150 citra, dan data sekunder sebanyak 25 citra. Pengambilan data primer dilakukan pada waktu pagi, siang, dan sore dengan jarak pengambilan 5–10 meter dan 5–10 meter. Rata-rata akurasi deteksi rambu perintah yang dihasilkan sistem dalam pengujian data primer sebesar 90,67% dan data sekunder sebesar 76%, sehingga rata-rata akurasi keseluruhan sistem sebesar 88,5%. Kegagalan sistem dalam mendeteksi keberadaan rambu perintah di jalan dikarenakan kondisi pencahayaan yang kurang atau berlebihan, rambu terhalang objek lain, sudut pengambilan citra yang terlalu miring, dan rambu berdampingan dengan objek lain yang memiliki warna sama atau mendekati sama dengan warna latar.

**Kata kunci:** rambu perintah, *Euclidean color filtering*, BLOB

### ABSTRACT

Command signs are one of the types of traffic signs that we often encounter when driving on the road and are used to express some commands while driving. A distinctive feature of command signs is that they have a blue background color. Driver negligence in paying attention to the presence of command signs on the road can result in accidents. There needs to be a system that can detect command signs on the road as a solution. The command sign detection process uses the Euclidean Color Filtering method to filter colors and BLOB to look for interconnected pixels. The RGB color center values (10,110,200) and the radius range [90, 150] are used in the color filter stage with Euclidean Color Filtering. The system was tested on 150 images of primary data and 25 images of secondary data. Primary data collection was carried out in the morning, afternoon, and evening with a distance of 5–10 meters and 10–15 meters. The system's average accuracy of command sign detection in testing primary data is 90.67%, and secondary data is 76%, so the average accuracy of the whole system is 88.5%. System failure in detecting the presence of command signs on the road is due to insufficient or excessive lighting conditions, signs blocked by other objects, angles that are too tilted, and signs adjoining other things with the same color or close to the background color.

**Keywords:** command signs, *Euclidean color filtering*, BLOB

### 1. PENDAHULUAN

Rambu lalu lintas adalah simbol aturan yang memiliki kekuatan hukum yang wajib ditaati oleh pengguna jalan. Rambu lalu lintas merupakan salah satu dari perlengkapan jalan yang memuat lambang, huruf, angka, kalimat, atau perpaduan diantaranya dengan tujuan memberi peringatan, larangan, perintah, ataupun petunjuk bagi pengguna jalan [1]. Rambu perintah merupakan salah satu jenis rambu lalu lintas yang digunakan untuk menyatakan perintah yang wajib dilakukan pengguna jalan agar tetap selamat selama perjalanan. Rambu perintah memiliki ciri berbentuk lingkaran dan persegi panjang, serta memiliki warna dasar biru dengan lambang atau tulisan berwarna putih.

Bagi pengguna jalan khususnya pengendara kendaraan seringkali lalai dalam memperhatikan rambu lalu lintas yang disebabkan kecepatan kendaraan, dan fokus pengendara. Adanya sebuah sistem yang



2. Perhitungan jarak  
 Perhitungan jarak *dest* dari selisih warna RGB menggunakan persamaan 2.  

$$dest = \sqrt{(\Delta R)^2 + (\Delta G)^2 + (\Delta B)^2} \dots\dots\dots (2)$$
3. Mempertahankan warna  
 Persamaan 3 menentukan sebuah warna patut dipertahankan atau tidak berdasarkan nilai radius. Radius adalah batas ambang yang menentukan tingkat kesamaan suatu warna dengan pusat warna [9].  

$$\text{if } dest \leq radius \text{ then}$$

$$colorRGB[i, j] = [ColorR[i, j], ColorG[i, j], ColorB[i, j]] \dots\dots\dots (3)$$

$$\text{else}$$

$$ColorRGB[i, j] = [0,0,0]$$

**Binary Large Object (BLOB)**

BLOB adalah metode untuk menentukan sekumpulan piksel memiliki hubungan atau tidak. Prosesnya dilakukan dengan menganalisis setiap piksel yang memiliki relasi *k-neighbors*. Jika warna dari piksel tetangga memiliki warna yang sama atau mendekati warna piksel pusat, maka piksel tetangga akan diberi label yang sama dengan piksel pusat atau dinyatakan sebagai bagian dari objek yang sama [10]. Relasi *k-neighbors* dimana *k* = 8 dapat dilihat pada gambar 2.



Gambar 2. Relasi 8-neighbors

Algoritma relasi 8-neighbors dijelaskan sebagai berikut:

- a. Telusuri baris dan kolom
- b. Piksel A merupakan piksel pusat dari 9 piksel, cek piksel tetangga:
  - Jika piksel B memiliki warna yang sama atau mendekati piksel A, maka berikan label pada piksel B sama dengan label piksel A.
  - Pengecekan yang sama juga dilakukan pada piksel C, D, E, F, G, H, dan I.

Setiap objek dikatakan sebagai kandidat rambu perintah jika memenuhi syarat:

- a. Minimum tinggi dan lebar adalah masing-masing 60 piksel.
- b. Minimum jari-jari objek adalah 20 piksel.
- c. Minimum jumlah piksel objek adalah 3000 piksel

**Deteksi bentuk**

Setiap objek yang didapat dari BLOB selanjutnya dilakukan deteksi bentuk. *Simple Shape Checker* merupakan sebuah *class* pada *library* AForge.Math yang digunakan untuk melakukan deteksi bentuk. Dua *method* yang digunakan adalah *IsCircle* untuk mendeteksi bentuk lingkaran, dan *IsQuadrilateral* untuk mendeteksi bentuk persegi panjang.

**3. HASIL DAN PEMBAHASAN**

**Spesifikasi perangkat**

Sistem diimplementasikan menggunakan Visual Studio Community 2015 pada komputer dengan spesifikasi antara lain *processor* Intel Core i5 8250U, *memory* 8GB, Hard disk 256 GB, dan *VGA* Nvidia GeForce MX130. Pengambilan citra rambu dilakukan dengan menggunakan *smartphone* ASUS Zenfone Max Pro M1 dengan spesifikasi *processor* Qualcomm Snapdragon 636, RAM 3 GB, *internal memory*: 32 GB, dan resolusi kamera 13 MegaPiksel.

**Data penelitian**

Data yang digunakan dalam penelitian ini adalah data primer dan data sekunder. Data primer berupa citra rambu perintah yang diambil di Kota Kupang dengan format citra JPG, dan ukuran citra yang dihasilkan oleh *smartphone* yaitu 3840x2160 piksel. Rambu perintah yang diambil berjumlah 25 rambu, setiap rambu diambil sebanyak 3 kali yaitu pada waktu pagi, siang, dan sore dengan jarak 5–10 meter dan 10-15 meter sehingga total data primer adalah 150 citra. Data sekunder berupa citra rambu perintah yang didapatkan dari internet sebanyak 25 citra dengan format citra JPG, waktu dan jarak pengambilan citra tidak diketahui, rincian ukurannya dapat dilihat pada tabel 1.




Tabel 1. Rincian ukuran citra pada data sekunder

	Minimum	Maksimum	Rata-rata
Citra	660x400 piksel	4440x2807 piksel	2011x1536 piksel
File citra	64 KiloByte	3100 KiloByte	1071 KiloByte











**Penentuan pusat warna dan radius**

Mula-mula tentukan tiga warna dominan biru seperti yang terlihat pada tabel 2. Selanjutnya tiga warna dominan ini diuji terhadap pusat warna RGB. Pada pengujian ini digunakan sepuluh data pusat warna RGB yang diambil secara acak seperti yang terlihat pada tabel 3. Lakukan perhitungan jarak antara data warna RGB pada tabel 3 dan 2 menggunakan *Euclidean distance* sesuai persamaan 2, hasil perhitungan jarak dapat dilihat pada tabel 4. Jarak terpendek dimiliki oleh data 1 sehingga terpilih sebagai nilai pusat warna RGB yaitu  $centerR = 10$ ,  $centerG = 110$ , dan  $centerB = 200$ .

Tabel 2. Tiga warna dominan biru

Warna	R	G	B	Hasil warna
Biru muda	0	180	255	
Biru	0	0	255	
Biru tua	0	0	190	

Tabel 3. Sepuluh data pusat warna RGB

Data	R	G	B	Warna
1	10	110	200	
2	10	120	200	
3	20	110	200	
4	20	120	200	
5	30	110	200	
6	30	120	200	
7	40	110	200	
8	40	120	200	
9	50	110	200	
10	50	120	200	

Tabel 4. Hasil perhitungan jarak dari 10 data pusat warna RGB terhadap 3 warna dominan biru

Data	Jarakbiru_muda	Jarakbiru	Jarakbiru_tua	Total jarak
1	89,58	123,39	110,91	323,88
2	82,01	132,38	120,83	335,22
3	91,24	124,60	112,25	328,09
4	83,82	133,51	122,07	339,39
5	93,94	126,59	114,46	334,99
6	86,75	135,37	124,10	346,21
7	97,60	129,33	117,47	344,39
8	90,69	137,93	126,89	355,51
9	102,10	132,76	121,24	356,11
10	95,52	141,16	130,38	367,06

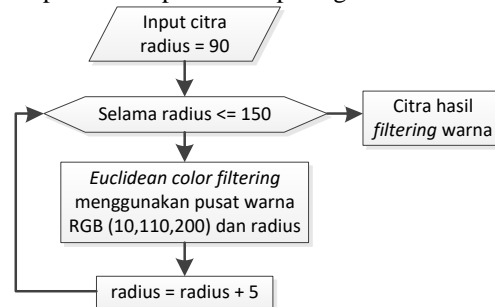
Parameter selanjutnya yang ditentukan adalah radius. Berdasarkan nilai pusat warna RGB (10,110,200) maka ditentukan radius mana saja yang dapat memfilter minimal satu dari tiga warna dominan biru, perhitungannya menggunakan persamaan 3. Dari tabel 5, dipilih nilai radius minimum adalah 90, dan nilai radius maksimum adalah 150 untuk membatasi agar warna lain tidak ikut terfilter.

Tabel 5. Berhasil atau gagalnya filter 3 warna dominan biru di setiap radius

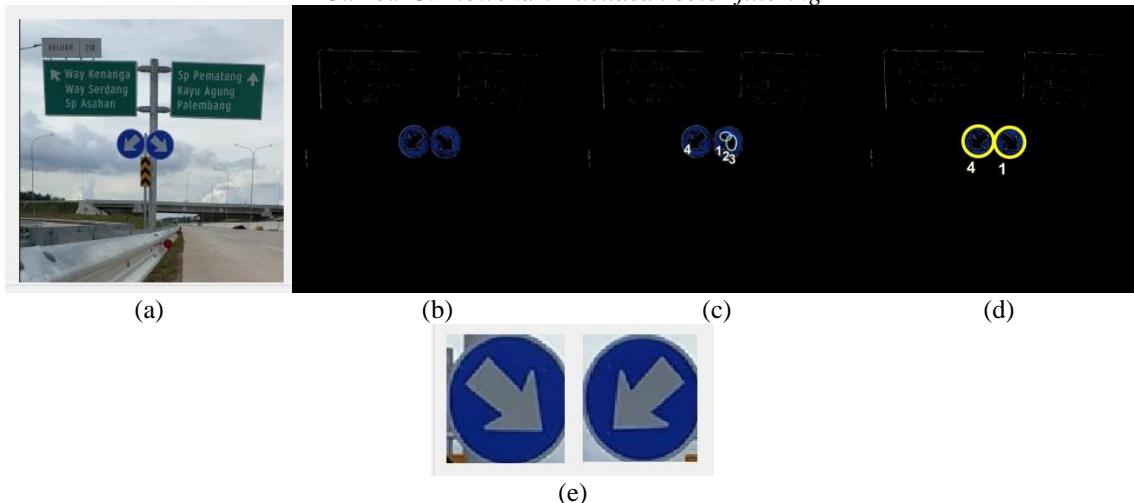
R	G	B	Jarak	Radius								
				80	90	100	110	120	130	140	150	160
0	180	255	89,58	x	✓	✓	✓	✓	✓	✓	✓	✓
0	0	255	123,39	x	x	x	x	x	✓	✓	✓	✓
0	0	190	110,91	x	x	x	x	✓	✓	✓	✓	✓

**Pengujian sistem**

Nilai pusat warna RGB (10,110,200) dan rentang radius [90,150] yang didapat digunakan dalam *Euclidean color filtering* untuk mendapatkan citra hasil *filtering* warna, seperti yang terlihat pada gambar 3. Citra ini selanjutnya dilakukan BLOB dan *simple shape checker* untuk didapatkan rambu perintah. Hasil pengujian terhadap data primer dan sekunder dapat dilihat pada tabel 6, dan contoh hasil citra di setiap tahapan dalam mendapatkan rambu perintah dapat dilihat pada gambar 4.



Gambar 3. Flowchart *Euclidean color filtering*



Gambar 4. Tahapan pemrosesan citra: (a) Citra asli, (b) Citra hasil *Euclidean color filtering*, (c) Citra hasil BLOB (d) Citra hasil deteksi bentuk, dan (e) Citra rambu perintah yang terdeteksi

Tabel 6. Hasil pengujian data primer dan sekunder

Waktu	Jarak (meter)				Akurasi (%)
	5-10		10-15		
	Berhasil	Gagal	Berhasil	Gagal	
Data Primer (150 Data)					
Pagi	24	1	22	3	92
Siang	24	1	25	0	98
Sore	21	4	20	5	82
Akurasi (%)	92	8	89	11	Rata-rata = 90,67
Data Sekunder (25 Data)					
Waktu dan jarak tidak diketahui	Berhasil		Gagal		
	19		6		76
<b>Rata-rata akurasi keseluruhan data (%)</b>					<b>88,5</b>

Citra pada data primer dan sekunder memiliki beragam kondisi yaitu kondisi baik, miring, pudar, rusak, terhalang, berbayang, berdampingan dengan rambu lain, dan dalam satu citra terdapat beberapa

rambu perintah dalam bentuk yang sama maupun berbeda. Contoh citra yang berhasil maupun gagal dideteksi sebagai rambu perintah dapat dilihat pada gambar 5.



Gambar 5. Hasil dari metode yang diusulkan: (a) Citra asli (b) Citra hasil

#### Pembahasan

Pada tabel 6, pengambilan citra di waktu dan jarak yang berbeda memberikan rata-rata akurasi deteksi rambu perintah di data primer sebesar 90,67%, sehingga dapat dikatakan sistem tidak rentan terhadap perbedaan waktu dan jarak pengambilan. Untuk data sekunder, akurasi deteksi rambu perintah sebesar 76%, walaupun waktu dan jarak tidak diketahui tetapi citra yang digunakan memiliki kondisi lebih beragam daripada data primer. Kegagalan sistem dalam melakukan deteksi, dominan disebabkan karena kondisi pencahayaan yang kurang atau berlebihan, rambu terhalang objek lain, sudut pengambilan citra yang terlalu miring, dan rambu berdampingan dengan objek lain yang memiliki warna sama atau mendekati sama dengan warna latar.

#### 4. KESIMPULAN DAN SARAN

Deteksi keberadaan rambu perintah di jalan dengan tahapan awal memfilter warna menggunakan *Euclidean color filtering* memerlukan parameter pusat warna dan radius. Penelitian ini mendapatkan nilai pusat warna RGB (10,110,200), dan rentang radius [90,150] yang mampu memfilter warna biru dengan baik. Berdasarkan nilai pusat warna dan radius ini, dilakukan pengujian terhadap data primer, diperoleh

rata-rata akurasi deteksi rambu perintah sebesar 90,67%. Pengujian pada data sekunder memberikan akurasi deteksi rambu perintah sebesar 76%, ini dikarenakan citra yang digunakan lebih beragam dari data primer. Pada penelitian berikutnya diharapkan penentuan nilai pusat warna RGB dan radius juga dilakukan pada rambu lalu lintas lainnya.

#### DAFTAR PUSTAKA

- [1] A. R. Himamunanto and S. Tora, "Alogaritma Quatree Untuk Segmentasi Citra Digital Rambu Lalu Lintas," *Jurnal Infact*, vol. 1, no. 2, p. 9, Nov. 2016.
- [2] O. R. Sitanggang, H. Fitriyah, and F. Utamingrum, "Sistem Deteksi dan Pengenalan Jenis Rambu Lalu Lintas Menggunakan Metode Shape Detection Pada Raspberry Pi," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 12, pp. 6108–6117, Dec. 2018.
- [3] H. Fleyeh, "Color detection and segmentation for road and traffic signs," in *IEEE Conference on Cybernetics and Intelligent Systems, 2004.*, Singapore, 2004, vol. 2, pp. 809–814. doi: [10.1109/ICCIS.2004.1460692](https://doi.org/10.1109/ICCIS.2004.1460692).
- [4] S. Wali, "Shape Matching and Color Segmentation Based Traffic Sign Detection System," *ELECTROTECHNICAL REVIEW*, vol. 1, no. 1, pp. 38–42, Jan. 2015, doi: [10.15199/48.2015.01.06](https://doi.org/10.15199/48.2015.01.06).
- [5] M. E. Baz, T. Zaki, and H. Douzi, "Red Color Segmentation based Traffic Signs Detection," *International Journal on Emerging Technologies*, vol. 11, no. 5, pp. 121–127, 2020.
- [6] F. Santi Wahyuni, "Penerapan BLOB (Binary Large Object) Analysis Pada Sistem Pengenalan Rambu-Rambu Lalu Lintas," *mnemonic*, vol. 1, no. 2, pp. 62–66, Dec. 2019, doi: [10.36040/mnemonic.v1i2.40](https://doi.org/10.36040/mnemonic.v1i2.40).
- [7] D. Avianto and W. S. Utami, "Segmentasi Berbasis Warna Pada Plat Nomor Kendaraan Umum Di Indonesia," presented at the Seminar Nasional Teknologi Informasi, Jakarta, 2017.
- [8] A. Basu and A. Mukherjee, "Automated Elephant Tracker," *International Journal of Electronics and Communication Engineering*, vol. 6, no. 1, pp. 17–22, Jan. 2015.
- [9] R. Efendi, E. P. Purwandari, and E. T. Mareta, "Segmentasi Warna Untuk Pendeteksian Rambu Lalu Lintas," in *Inovasi dan Teknologi untuk Pembangunan Indonesia Berkelanjutan*, Bengkulu, Sep. 2018, p. 6. [Online]. Available: <http://senitia.ft.unib.ac.id/prosiding/2018f-1-18-rusdi-efendi/>
- [10] N. Sarashadarti, B. Hidayat, and Suhardjo, "Sintesa Penelitian Deteksi Kista Periapikal Radiograf Dengan Metode Binary Large Object (BLOB) Dan Metode Gray Level Co- Occurrence Matrix (GLCM)," *e-Proceeding of Engineering*, vol. 5, no. 3, p. 5538, Dec. 2018.