

Sistem Deteksi dan Pengenalan Jenis Rambu Lalu Lintas Menggunakan Metode *Shape Detection* Pada Raspberry Pi

Olivia Rumiris Sitanggang¹, Hurriyatul Fitriyah², Fitri Utaminingrum³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹oliviasitanggangr@gmail.com, ²hfitriyah@ub.ac.id, ³f3_ningrum@ub.ac.id

Abstrak

Pengenalan Rambu Lalu Lintas (*Traffic Sign Recognition*) adalah teknologi pengolahan citra digital yang digunakan untuk mengenali rambu secara *real-time*. Teknologi ini diterapkan dalam *Driver Assistance System* (DAS). Pengenalan Rambu Lalu Lintas terdiri dari dua fase utama, yaitu deteksi rambu dan pengenalan. Deteksi merupakan fase untuk menemukan kemungkinan area gambar dimana rambu berada. Keluaran dari proses deteksi adalah gambar hasil segmentasi berisi *Region of Interest* (ROIs) yang dapat mengenali daerah potensial rambu jalan berada. Daerah potensial tersebut akan mempengaruhi *input* proses pengenalan (*recognition*). Sehingga dibangun sebuah sistem pendeteksi dan pengenalan jenis rambu. Sistem ini diimplementasikan pada Raspberry Pi dan bekerja *real-time* saat mengolah citra dari rambu dari *webcam*. Algoritma deteksi dibagi menjadi tiga bagian utama, yaitu segmentasi warna, *shape detection*, dan klasifikasi rambu. Metode yang diterapkan dalam penelitian ini adalah metode pengenalan bentuk. Metode ini didukung oleh jumlah titik dari objek sebagai representasi dari tiap-tiap bentuk dan perbandingan luas objek kontur dengan *bounding rectangle*. *Output* dari sistem ini berupa notifikasi suara jenis rambu bagi pengemudi. Diharapkan dengan metode ini proses pendeteksian untuk menemukan regional rambu lebih akurat sehingga memperkecil jumlah informasi yang perlu diolah dalam tahap pengenalan. Tingkat keberhasilan deteksi jenis rambu perintah, larangan, dan rambu peringatan yaitu 80.7%, hasil pengujian warna dari ketiga rambu mencapai angka 85.45%, dan hasil persentase mengenali bentuk dari rambu adalah 80.7%. durasi mendeteksi jenis rambu ini menghasbiskan 0.5 detik untuk satu frame atau 2 frame perdetiknya dengan jarak deteksi 2-5 meter.

Kata kunci: rambu lalu lintas, *shape detection*, deteksi

Abstract

The traffic sign recognition is the digital image processing technology that used to recognize the sign in real-time. This technology applied in the Driver Assistance System. The road sign recognition consist of 2 main phase, they are road detection and recognition. Detection is the phase to find the possibility of picture area where the sign is located. The output from the detection process is the result picture segmentation that contain region of interest that can recognize the potential area of where the road sign being located. Those potential area will be affected the input of recognition process. So built a system of detection and recognition of the type of signs. This system is implemented on raspberry pi and real-time when processing the image of road sign from webcam. The detection of algorithm consist into three main part, they are color segmentation, shape detection, and road classification. The method that being applied in this research is shape recognition method. This method is supported by the amount of point from the object as a representation of the amount of side from every shape and the comparison of object area with the bounding rectangle. And the output of this system is a kind of the sign notification for drivers. It is expected with this method the detection process to find the accurate regional sign recognition. The level of success in detecting kind of command signs, prohibition, and warning sign are 80.7%, the result of color examination from the three signs reach the number of 85.45%, and the result of presentation in recognizing the shape of sign is 80.7%. the duration of detecting of traffic signals is 0.5 seconds (for each frame) or 2 frames per second with detection distance 2-5 meters.

Keywords: traffic road sign, shape detection, detection.

1. PENDAHULUAN

Teknologi *Traffic Sign Recognition* merupakan teknologi yang berguna untuk mendeteksi dan mengenali rambu jalan secara *real-time* dengan pemrosesan citra digital. Implementasi dari teknologi ini adalah *Driver Assistance System* (DAS). Beberapa sub sistem yang berada dalam *Driver Assistance System* diantaranya pendeteksian garis lane pada jalan, bantuan parkir otomatis, monitoring kondisi dan gangguan pengemudi, *auto pilot* atau sistem kemudi otomatis, dan GPS. Tujuan dari teknologi pengenalan rambu ini adalah membantu pengemudi saat melewati dan tidak mengetahui keberadaan rambu. rambu adalah bagian visualisasi perbuhan pada kondisi jalan. Pengemudi yang lalai sangat membahayakan dan dapat menimbulkan kecelakaan.(Ilmuti.org)

Sistem pengenalan pada rambu lalu lintas memiliki dua fase utama, yaitu fase deteksi dan pengenalan rambu. fase deteksi adalah fase untuk menyeleksi objek pada sebuah gambar dengan tujuan menemukan rambu itu sendiri. Melalui proses segmentasi maka hasil dari fase ini adalah *Region Of Interest* (ROIs) atau objek rambu tersebut. Hasil tersebut akan menjadi masukan untuk tahap pengenalan rambu (*recognition*). Fase ini merupakan fase untuk mendefenisikan arti dan tujuan dari rambu tersebut menggunakan *template* yang telah dipersiapkan atau metode lainnya. (I. Sebanja, 2010)

Sebuah penelitian dengan judul *Road Sign Detection and Recognition System for Real-Time Embedded Applicatios* berhasil mendeteksi dan melakukan pengenalan pada rambu hingga menghasilkan akurasi sebesar 90.7%. deteksi rambu dibagi kedalam tiga tahap yaitu segmentasi warna, *shape detection*, dan juga klasifikasi rambu. kemudian tahap pengenalan rambu dengan menghitung luas objek dan keliling dari *Region Of Interest* dari rambu. (S.S.M. Sallah, 2014)

Hasil akurasi deteksi mencapai angka 90% dan hasil pengenalan rambu sebesar 88%. Dari hasil tersebut dapat disimpulkan bahwa keluaran hasil deteksi sangat mempengaruhi hasil dari pengenalan karena hasil deteksi berupa region of interest dari menggunakan segmentasi wana adalah *input* ke proses pengenalan. Penelitian oleh (Sheikh, 2016) yang berjudul *Traffic Sign Detection and Classification using Colour*

Feature and Neural Network dibagi kedalam 3 modul utama, yaitu *pre-processing*, *detection*, dan *recognition*. Fase *pre-processing* adalah tahap untuk menghilangkan *noise* sehingga kualitas gambar menjadi tinggi.

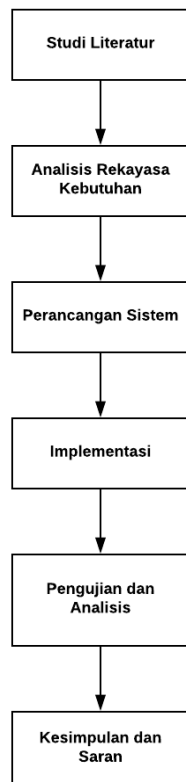
Bersumber pada kedua penelitian sebelumnya, proses deteksi dan objek dari hasil proses ini sangat mempengaruhi ke tahap pengenalan (*recognition*). Penelitian ini akan dibangun sebuah sistem deteksi pada rambu lalu lintas (larangan, peringatan, dan perintah) pada Raspberry Pi menggunakan metode pengenalan bentuk (*shape detection*) yang bekerja secara *real-time*. Video hasil rekaman oleh kamera dipecah menjadi gambar-gambar sebagai *input* dari sistem. Algoritma deteksi dikateogirkan menjadi empat bagian, yaitu ekstraksi *input* video, segmentasi warna, pengenalan bentuk, dan klasifikasi rambu.

Metode pengenalan bentuk adalah gabungan dari fungsi *approximation*, luas kontur, dan aspek rasio. fungsi *approximation* adalah fungsi untuk menemukan jumlah titik puncak atau *vertice* pada sebuah bangun datar. Kedua adalah luas kontur yang berasal dari jumlah pixel pada objek. Dan terakhir aspek rasio adalah perbandingan panjang dan lebar dari suatu gambar. Keluaran dari sistem berupa notifikasi suara bagi pengendara. Dengan metode ini diharapkan sistem deteksi rambu lebih akurat sehingga dapat mendukung ke tahap proses pengenalan rambu. (OpenCV, n.d.)

2. METODOLOGI PENELITIAN

Tahapan metodologi dimulai dengan studi literatur sebagai dasar teori dan literatur penelitian ini. Tahap berikutnya adalah analisis kebutuhan penelitian mencakup kebutuhan sistem, kebutuhan fungsional dan non fungsional. Perancangan dibagi menjadi dua bagian, yaitu perancangan perangkat keras dan perangkat lunak (algoritma deteksi). Implementasi adalah bentuk nyata dari perancangan sistem deteksi rambu lalu lintas.

Setelah sistem terbentuk, pengujian sangat diperlukan untuk memastikan ketepatan kerja sistem berdasarkan keakuratan hasil keluaran sistem tersebut. Terakhir adalah pengambilan kesimpulan dari hasil pengujian dan saran untuk pengembangan lanjut. Keseluruhan tahap diatas dapat digambarkan ke dalam diagram alir seperti pada Gambar 1.



Gambar 1. Diagram Alir Metodologi Penelitian

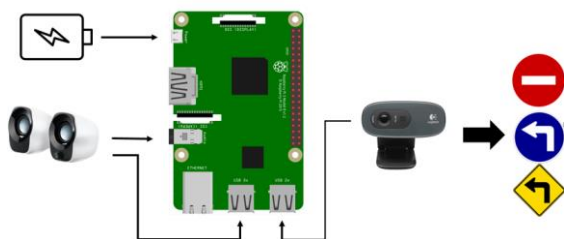
2.1. Perancangan Perangkat Keras

Sistem terdiri dari Raspberry Pi, kamera, dan *speaker*. Sistem akan berjalan ketika sistem tersambung dengan sumber tegangan.

Kamera langsung mulai merekam kondisi jalan dan program pada Raspberry Pi akan mengolah *frame-frame* untuk mendeteksi keberadaan rambu pada *video*. Hasil *output* tersebut direpresentasikan ke bentuk suara yang keluar melalui *speaker* seperti pada Gambar 2. Sistem bekerja secara *real-time*.

2.2. Perancangan Perangkat Lunak

Video hasil *capture* kamera yang terpasang pada Raspberry Pi adalah *input* yang akan diolah di sistem. Library OpenCV dan bahasa pemrograman *python* sebagai pondasi perancangan *software*. Algoritma deteksi terdiri dari 4 bagian yaitu perancangan ekstraksi pada video, segmentasi warna, *shape detection*,



Gambar 2. Perangkat Keras

dan klasifikasi. Ekstraksi video adalah tahap perubahan video menjadi *frame*. Sedangkan *segmentasi* hingga klasifikasi adalah tahap *frame* menjadi keluaran suara.

2.2.1. Perancangan Ekstraksi Video

Ekstraksi *video* memiliki tujuan untuk memecah sebuah *video* menjadi *frame-frame* secara sekuensial. Kemudian *frame* yang telah dipisah akan diolah satu persatu. Proses ini disebut dengan *frame acquisition*. Seperti Gambar 3, berikut diagram alir proses *frame acquisition*.

2.2.2. Segmentasi Warna

Gambar RGB dikonversi kedalam bentuk HSV. Segmentasi warna adalah fase seleksi piksel berdasarkan range warna hsv yang telah ditentukan. Hasil dari fase ini adalah *Region Of Interest* (ROIs). Alasan untuk menggunakan HSV adalah cara ruang warna HSV yang memisahkan nilai *hue*, saturasi, dan *value* sehingga tidak seluruh warna perlu digunakan. Hanya warna dari rambu lalu lintas (perintah, larangan, peringatan). Berikut beberapa langkah untuk melakukan konversi ruang warna RGB menjadi ruang warna HSV.

Mencari nilai R, G, B melalui aplikasi edit warna kemudian mencari nilai R' , B' , G' melalui perbandingan masing-masing komponen warna seperti Persamaan 1, Persamaan 2, dan Persamaan 3.

$$R' = R/255 \quad (1)$$

$$G' = G/255 \quad (2)$$

$$B' = B/255 \quad (3)$$

Mencari nilai c_{max} pada Persamaan 4, c_{min} pada Persamaan 5, dan delta pada Persamaan 6.

$$c_{max} = \max(R', G', B') \quad (4)$$

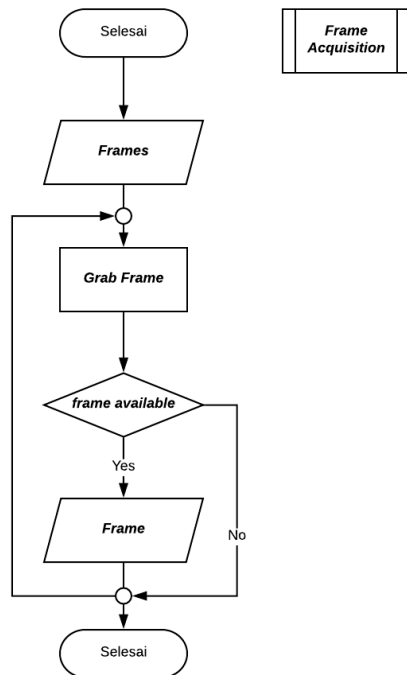
$$c_{min} = \min(R', G', B') \quad (5)$$

$$\Delta = c_{max} - c_{min} \quad (6)$$

Mencari nilai *hue* didasarkan hasil c_{max} seperti pada Persamaan 7. Untuk nilai delta sama dengan 0 maka nilai *H* sama dengan 0.

$$H = \begin{cases} H = 60^\circ \times \left(\frac{G' - B'}{\Delta} + 0 \right), & c_{max} = R' \\ H = 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right), & c_{max} = G' \\ H = 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right), & c_{max} = B' \end{cases} \quad (7)$$

Menemukan nilai saturasi dipengaruhi oleh nilai c_{max} seperti pada Persamaan 8. Untuk



Gambar 3. Flowchart Ekstraksi Video

Setiap nilai c_{max} sama dengan nol maka nilai *saturation* adalah 0° .

$$S = \begin{cases} 0, & c_{max} = 0 \\ \frac{\Delta}{c_{max}}, & c_{max} \neq 0 \end{cases} \quad (8)$$

Nilai *value* sama dengan nilai c_{max} Persamaan 9.

$$V = c_{max} \quad (9)$$

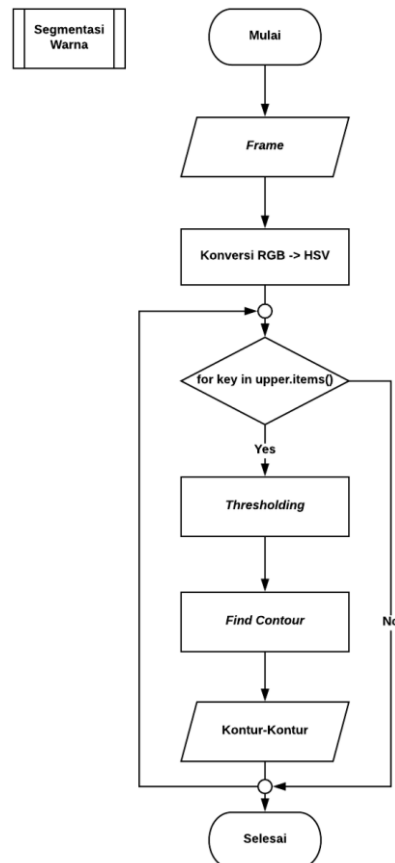
Pustaka OpenCV memberikan range HSV yaitu 0-179. Menggunakan format integer 8bit maka derajat *hue* dibagi kedalam dua bagian dimana nilai 110 pada pustaka OpenCV sama dengan 220° (OpenCV, 2018).

Range HSV yang dipakai untuk melakukan seleksi objek rambu, yaitu warna merah (170,50,50) dan (190,255,255), range minimum biru (100,50,50) dan maksimum (120,255,255). Dan terakhir untuk warna kuning batas minimum HSV (20,100,100) dan maksimum (40,255,255). (E. Petlenkov, 2015)

Piksel dengan nilai diantara *range* tersebut dalam akan bernilai 1 dan piksel lainnya dianggap 0. selanjutnya menemukan kontur dari piksel-piksel putih. kontur sebagai *Region Of Interest* diolah satu persatu seperti pada flowchart pada Gambar 4. (OpenCV, 2018)

2.2.3. Shape Detection

Terdapat dua faktor penentuan jenis bentuk pada sistem, yaitu *vertice*, aspek rasio dan luas objek. *vertice* atau titik puncak adalah jumlah



Gambar 4. Flowchart Segmentasi Warna

Titik pada sebuah bangun datar. Titik ini merepresentasikan jenis bentuknya. Jika kontur biner memiliki 3 titik pada bentuk tersebut maka dapat disebutkan bahwa objek ini adalah bangun segitiga. Untuk mencari titik puncak objek digunakan fungsi *approximation* dari pustaka OpenCV. Titik puncak atau *vertice* pada sebuah bangun datar dapat dijabarkan seperti pada Tabel 1. (Puri, 2018)

Faktor kedua adalah aspek rasio. aspek rasio adalah perbandingan panjang dan lebar bangun datar. Menggunakan bantuan *bouding rectangle* disekeliling luas minimum objek maka ditemukan nilai panjang dan lebar bentuk kotak (*square*) dari nilai *width* dan *height bounding rectangle* seperti pada persamaan 10.

- *Square*

$$\text{Aspect ratio square} = \frac{\text{width}}{\text{height}} \quad (10)$$

Daun rambu berbentuk bulat memerlukan dua cara untuk mengenali bentuknya. Pertama adalah aspek rasio seperti pada Persamaan 11.

Kedua adalah perbandingan luas kontur objek dengan luas kontur pada Persamaan 13 menggunakan radius dengan bantuan *bouding rectangle* yang mengelilingi objek pada Persamaan 12. (OpenCV, 2018)

- Circle

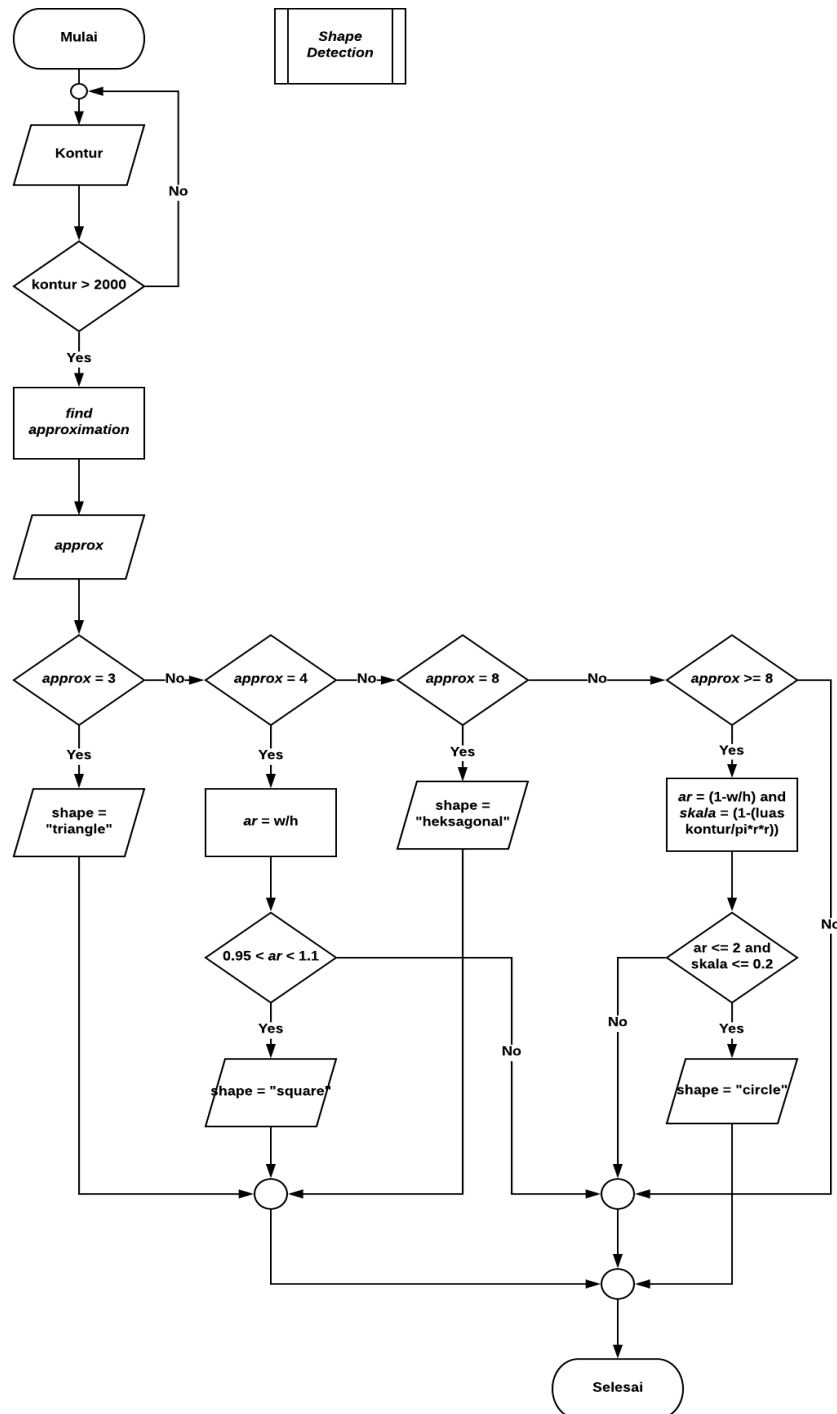
$$\text{Aspect ratio} = 1 - \frac{\text{width}}{\text{height}} \quad (11)$$

$$\text{radius} = \frac{\text{width}}{2} \quad (12)$$

$$\text{skala} = 1 - \frac{\text{luas kontur}}{\pi \text{radius}^2} \quad (13)$$

Tabel 1. Bentuk dan Jumlah Vertice

Bentuk	Vertice (titik)
circle	>=8
Square	4
triangle	3
heksagonal	8



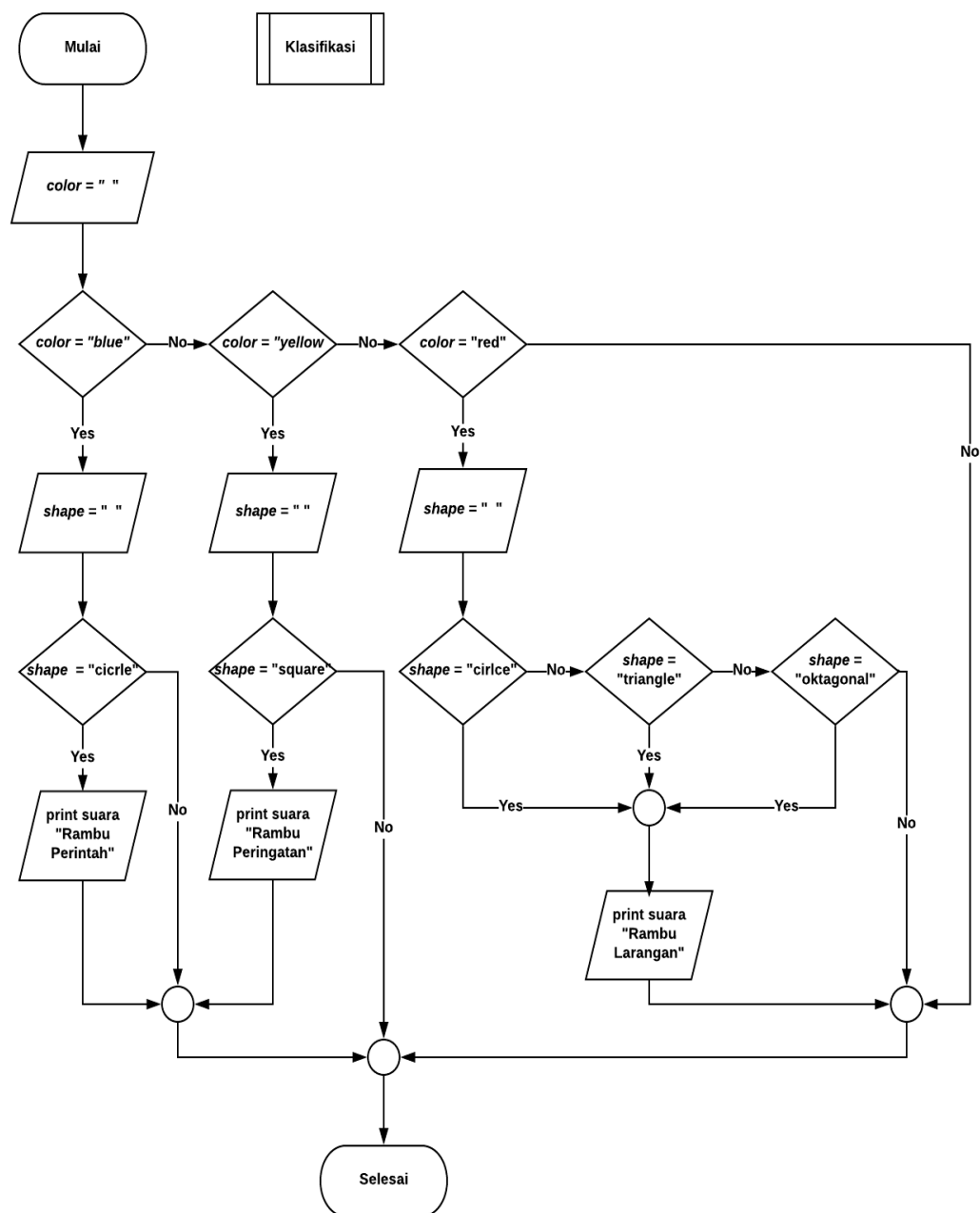
Gambar 5. Diagram Alir Shape Detection

Berdasarkan ketentuan dari masing-masing bentuk (*approximation*, aspek rasio, dan luas objek), maka untuk mengklasifikasikan bentuk dari kontur tersebut dibutuhkan diagram alir seperti pada Gambar 5.

2.2.4. Klasifikasi

Rambu memiliki dua ciri khas, yaitu warna dan bentuk. Kedua ciri membantu rambu menjadi lebih informatif dan dapat dikategorikan

menjadi 3 jenis, yaitu rambu perintah, rambu peringatan, dan rambu larangan. Informasi warna dan bentuk rambu ditentukan berdasarkan peraturan Kementerian Perhubungan tahun 2014. Dari seluruh kombinasi warna dan bentuk maka dapat disimpulkan dalam *flowchart* pada Gambar 6. Maka dengan begitu kontur atau objek yang memiliki kecocokan dengan aturan yang telah ditentukan dapat disebut sebagai rambu lalu lintas. (Kementerian Perhubungan, 2014)



Gambar 6. Flowchart Klasifikasi

3. PENGUJIAN DAN HASIL

3.3.1 Hasil Pengujian Ketepatan Deteksi

Penilaian performa sistem didasari pengujian deteksi terhadap warna, bentuk, dan jenis rambu (peringatan, perintah, dan larangan). Pengujian dilaksanakan sepanjang jarak 2-5 meter antara kamera dan rambu.

a. Terdeteksi Warna

Gambar RGB dikonversi ke bentuk ruang warna HSV. Kemudian pencarian rambu dilakukan dengan menyeleksi berdasarkan nilai *hue*, saturasi, dan *value* dari masing-masing warna (biru, merah, kuning). Pada penelitian ini, warna adalah syarat awal pencarian objek rambu, jika ada objek berada pada range yang telah ditentukan, maka ada kemungkinan objek tersebut adalah rambu dan akan memunculkan tulisan diatas rambu seperti pada Gambar 7.

b. Terdeteksi Bentuk

Saat warna tidak terdeteksi maka otomatis tidak akan mengenali bentuk. Namun saat warna dikenali maka ada dua kemungkinan yaitu bentuk dari kontur merupakan salah satu bentuk rambu atau tidak sama sekali (*undetected*) sesuai dengan Gambar 8.

c. Terdeteksi Jenis

Deteksi jenis dipengaruhi oleh hasil deteksi warna dan bentuk objek. Jika warna merah dengan bentuk segitiga, bulat, dan heksagonal amaka disebut dengan rambu larangan.



Gambar 7 Deteksi Warna



Gambar 8 Deteksi Bentuk

Objek berwarna kuning dan berbentuk kotak disebut dengan rambu peringatan. Dan rambu berwarna biru dan berbentuk bulat adalah rambu perintah. Jika salah satu dari antara kedua parameter (warna dan bentuk) tidak ditemukan atau salah maka deteksi jenis dianggap gagal atau tidak ada keluaran yang diberikan seperti Gambar 9.

Seluruh hasil deteksi dibagi kedalam tabel berisi kolom warna, bentuk, dan jenis. Keberhasilan deteksi dihitung per *frame* nya. Kolom total *frame* adalah jumlah total *frame* yang terekam untuk masing-masing percobaan.

Akurasi total pengujian adalah jumlah seluruh akurasi dibagi dengan jumlah percobaan seperti pada persamaan 15.

$$\text{total akurasi} = \frac{\text{akurasi}}{\text{total percobaan}} \quad (15)$$



Gambar 9 Deteksi Jenis

Rambu peringatan adalah rambu dengan kotak (*square*) dengan daun rambu warna kuning. Pada gambar rambu akan muncul kedua parameter (kuning dan *square*) pada *frame* dan terminal. Hasil akurasi jenis mencapai 80.55%, sedangkan untuk kedua parameter yaitu warna dan bentuk masing-masing 86.06% dan 80.55%. hasil ketiga pengujian tersebut dapat dijabarkan seperti pada Tabel 2.

Setiap *frame* dapat terdeteksi keberadaan rambu perintah jika objek pada frame memiliki bentuk bulat (*circle*) dan warna biru. Bagian atas objek rambu akan muncul tulisan *circleblue*.

Rata-rata akurasi pengujian warna dari rambu perintah yaitu 86.67%, sedangkan bentuk kotak terdeteksi hingga 81.4%, setelah keduanya ditemukan, maka persentasi jenis rambu peringatan 81.4%. ketiga persentase tersebut didapatkan berdasarkan hasil pada Tabel 3.

Ketika pada frame ditemukan bentuk *circle*, *triangle* atau heksagonal dan daun rambu berwarna merah maka objek kemungkinan adalah rambu larangan. Untuk memastikan kebenarannya maka keluaran *frame* pada *display* dijadikan titik acuan kecocokan deteksi pada terminal dan kondisi asli. Berdasarkan hasil pengujian pada Tabel 4, maka ditemukan hasil akurasi warna 83.61%, bentuk 80.15% dan persentase jenis sama halnya dengan bentuk yaitu 80.15%.

Tabel 2. Persentase Percobaan Rambu Peringatan

Percobaan	Akurasi Warna(%)	Akurasi Bentuk(%)	Akurasi Jenis(%)
1	77.1	76.1	76.1
2	81.2	81.2	81.2
3	73.33	73.33	73.33
4	88.57	82.86	82.86
5	80	80	80
6	86.67	86.67	86.67
7	83.2	83.2	83.2
8	97.95	81.63	81.63
9	98.89	76.67	76.67

10	96.77	83.87	83.87
Rata-rata Akurasi	86.06	80.55	80.55

Tabel 3. Persentase Percobaan Rambu Perintah

Percobaan	Akurasi Warna(%)	Akurasi Bentuk(%)	Akurasi Jenis(%)
1	85.94	84.37	84.37
2	82.93	79.27	79.27
3	94.02	75.54	75.54
4	94.59	85.14	85.14
5	92.63	87.37	87.37
6	81.36	74.58	74.58
7	88.37	83.72	83.72
8	74.02	74.02	74.02
9	88.40	85.51	85.51
10	84.48	84.48	84.48
Rata-rata Akurasi	86.67	81.4	81.4

Tabel 4. Persentase Percobaan Rambu Larangan

Percobaan	Akurasi Warna(%)	Akurasi Bentuk(%)	Akurasi Jenis(%)
1	88	86.67	86.67
2	85.46	83.64	86.64
3	79.17	77.08	77.08
4	82.93	73.17	73.17
5	76.4	75.28	75.28
6	83.33	74.4	74.4
7	85.36	82.92	82.92
8	93.15	84.93	83.93

9	83.33	83.33	83.33
10	78.95	78.07	78.07
Rata-rata Akurasi	83.61	80.15	80.15

Rambu berhasil terdeteksi jika keseluruhan objek memenuhi nilai parameter (*range* HSV, luas kontur, titik puncak, perbandingan luas yang telah ditentukan. Ketika paparan cahaya yang mengenai daun rambu tidak merata kondisi cuaca buruk (hujan) dapat mengganggu pendeteksian.

3.3.2 Hasil Pengujian Waktu Deteksi

Perhitungan waktu deteksi didapat berdasarkan selisih waktu akhir (*endtime*) dan waktu awal (*starttime*). Waktu pengolahan dihitung untuk setiap *frame* seperti pada Tabel 5. Sehingga diperoleh rata-rata waktu yang diperlukan untuk melakukan deteksi dan pengenalan jenis rambu adalah 0.5 detik untuk setiap *frame* atau 2 *frame* untuk setiap detiknya.

Tabel 5. Waktu Deteksi Rambu Lalu Lintas

Percobaan	Rambu Peringatan (detik)	Rambu Perintah (detik)	Rambu Larangan (detik)
1	0.41	0.40	0.47
2	0.40	0.50	0.48
3	0.87	0.48	0.49
4	0.84	0.59	0.45
5	0.83	0.45	0.49
6	0.4	0.36	0.6
7	0.5	0.49	0.4
8	0.40	0.48	0.48
9	0.82	0.41	0.4
10	0.5	0.4	0.57
Rata-rata waktu	0.59	0.45	0.48

4. KESIMPULAN

Berdasarkan hasil seluruh tahap penelitian yang telah dilakukan maka terdapat tiga kesimpulan untuk menjawab rumusan masalah sebelumnya antara lain:

Tahap perancangan perangkat lunak (segmentasi warna, *shape detection*, dan klasifikasi jenis rambu) yang didukung oleh pustaka OpenCV berbasis python berhasil merancang sistem deteksi rambu lalu lintas pada Raspberry Pi.

Kontur hasil segmentasi warna berupa gambar biner akan diseleksi berdasarkan 3 kategori, yaitu jumlah *vertice* yang didapat dari menggunakan fungsi *approximation*. Untuk setiap objek yang memiliki jumlah titik sebanyak tiga artinya berbentuk segita maka kemungkinan adalah jenis rambu larangan. kedua perbandingan panjang dan lebar kontur atau disebut dengan aspek rasio dan terakhir perbandingan luas kontur dengan bantuan *bounding rectangle* di sekitar objek.

Sebuah frame dapat diolah rata-rata menghabiskan waktu 0.5 detik artinya 2 frame tiap detiknya.

Ciri khas dari suatu rambu terletak pada bentuk dan warnanya. Sehingga tingkat keakuratan keduanya sangat mempengaruhi pendeteksian rambu oleh sistem. Maka dengan begitu dari hasil pendeteksian warna dari rambu lalu lintas diperoleh persentase sebanyak 85.45% dan hasil pengenalan bentuk didapat sebesar 80.7%. sehingga dari hasil keduanya diperoleh nilai akurasi deteksi jenis dari keseluruhan percobaan sebesar 80.7%. pengujian sistem dianggap sudah cukup baik dalam mendeteksi rambu.

Saran yang dapat membangun pengembangan lanjut untuk penelitian sejenis antara lain adalah jarak deteksi rambu menggunakan kamera diharapkan melebihi jarak 5 meter. Melanjutkan penelitian deteksi rambu menjadi pengenalan rambu lalu lintas. Penerapan metode yang baik untuk mendukung klasifikasi rambu yang lebih baik. Dan terakhir penambahan jumlah jenis rambu yang akan dideteksi sistem.

5. DAFTAR PUSTAKA

E. Petlenkov, A. T. (2015). Raspberry Pi based System for Visual Object Detection and Tracking. Tallin: hgpu.

- I. Sebanja, D. M. (2010). Automatic Detection and Recognition of Traffic Road Signs for Intelligent Autonomous Unmanned Vehicles for Urban Surveillance and Rescue. 1-7.
- Ilmuti.org. (n.d.). *Ilmuti.org*. Retrieved September 7, 2017, from http://ilmuti.org/wp-content/uploads/2017/05/Tavip-Prahasta_CaraKerjaSistemAutopilotPadaMobil.pdf
- Kementerian Perhubungan. (2014, April 14). *Peraturan Menteri Perhubungan Nomor PM 13 Tahun 2014*. Retrieved September 2018, from http://jdih.dephub.go.id/produk_hukum/view/VUUwZ01UTWdWRUZJVIU0Z01qQXhOQT09
- OpenCV. (2018). *Contour : Getting Started*. Retrieved May 15, 2018, from https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=findcontours#findcontours
- OpenCV. (2018). *Morphological Transformation*. Retrieved May 14, 2018, from https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html
- OpenCV. (2018). *Contour : Getting Started*. Retrieved May 15, 2018, from https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=findcontours#findcontours
- Puri, R. (2018). Contour, Shape & Color Detection using OpenCV-Python. 3.