

**RANCANG BANGUN SISTEM PENGENALAN RAMBU
PETUNJUK ARAH BERBASIS RASPBERRY PI MENGGUNAKAN
METODE OCR (*OPTICAL CHARACTER RECOGNITION*)**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Rando

NIM: 135150301111061



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

RANCANG BANGUN SISTEM PENGENALAN RAMBU PETUNJUK ARAH BERBASIS
RASPBERRY PI MENGGUNAKAN METODE OCR (*OPTICAL CHARACTER
RECOGNITION*)

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Rando

NIM : 135150301111061

Skripsi ini telah diuji dan dinyatakan lulus pada
3 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Hurriyatul Fitriyah, S.T, M.Sc
NIP: 19851001 201504 2 003

Wijaya Kurniawan, S.T, M.T
NIP: 19820125 201504 1 00 2

Mengetahui
Ketua Jurusan Teknik Infomatika

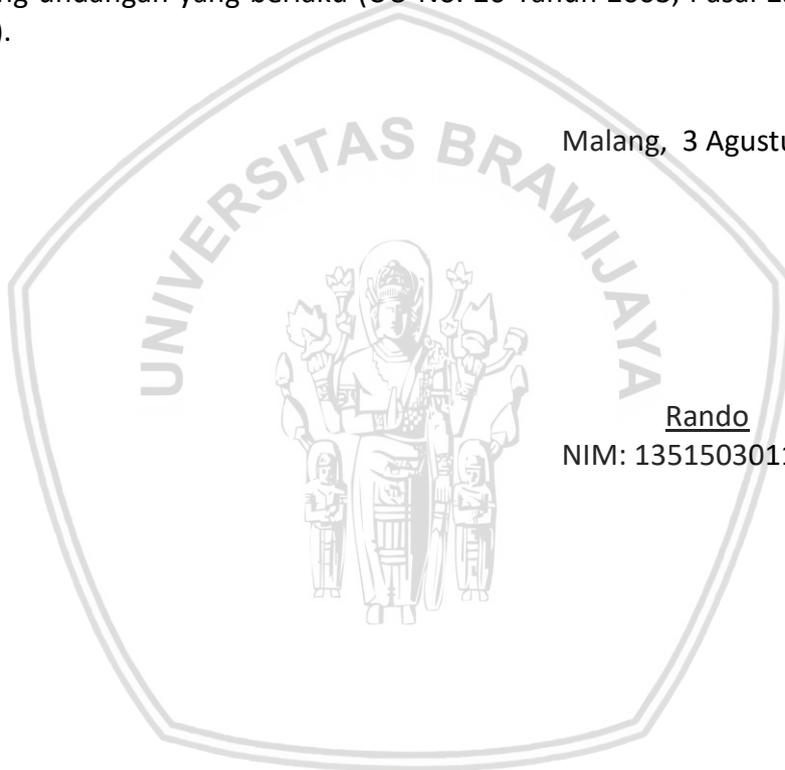
Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 3 Agustus 2018



Rando

NIM: 135150301111061

KATA PENGANTAR

Dengan menyebut nama Allah Subhanahu Wa Ta'ala Yang Maha Pengasih dan Maha Penyayang. Puji dan syukur penulis panjatkan kehadiran Allah SWT atas anugerah serta limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul "Rancang Bangun Sistem Pengenalan Rambu Petunjuk Arah Berbasis Raspberry Pi Menggunakan Metode OCR (*Optical Character Recognition*)". Skripsi ini disusun sebagai syarat memperoleh gelar sarjana pada Program Studi Teknik Komputer, Fakultas Ilmu Komputer Universitas Brawijaya.

Dalam proses penyelesaian skripsi ini penulis mendapatkan banyak bantuan, baik bantuan moral maupun materiil dari berbagai pihak. Oleh karena itu, penulis mengucapkan banyak terimakasih kepada:

1. Ibu Hurriyatul Fitriyah, S.T, M.Sc dan Bapak Wijaya Kurniawan, S.T, M.T selaku dosen pembimbing skripsi yang telah memberikan bimbingan, arahan, ilmu, dan masukan dalam penyelesaian skripsi ini.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Bapak Ir. Heru Nurwarsito, M.Kom, Bapak Drs. Marji, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Seluruh Dosen dan Staff Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Universitas Brawijaya dan selama penyelesaian skripsi ini.
4. Kedua Orangtua yang telah memberi support kepada saya untuk menyelesaikan skripsi ini, karena doa orangtua maka skripsi ini tidak akan selesai dengan orangtua akan memudahkan segala jalan.
5. Teman-teman Program Studi Teknik Komputer yang selalu memberikan semangat, dukungan, dan kebersamaan selama Penulis menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
6. Kepada semua teman "Label Merah", KAMMFIL, dan yang selalu memberi support dalam menyelesaikan skripsi dan tidak melupakan tentang nikmat Allah Ta'ala yang telah diberikan.

Penulis menyadari bahwa skripsi tidak terlepas dari kesalahan dan kekurangan dalam penulisan skripsi ini. Oleh karena itu, kritik dan saran dapat diterima dalam skripsi ini guna memberikan manfaat dan berguna kepada segala pihak.

Malang, 3 Agustus 2018

Penulis

randoago@gmail.com

ABSTRAK

Rambu petunjuk arah merupakan salah satu sarana yang memberikan petunjuk atau keterangan kepada pengemudi atau pemakai jalan lainnya, tentang arah yang harus ditempuh atau letak kota yang akan dituju lengkap dengan nama dan arah letak itu berada. Rambu petunjuk arah diperlukan agar pengendara fokus pada jalan ketika berkendara. Namun pengendara seringkali melewati rambu lalu lintas tanpa membaca pesan yang tersirat di dalamnya, dibutuhkan suatu sistem yang dapat mengolah citra dari rambu petunjuk arah agar pengendara fokus pada jalan maka informasi berupa suara kepada pengendara. Sehingga dalam penelitian ini dibuatlah sistem pengenalan rambu petunjuk arah berbasis raspberry pi menggunakan metode OCR (*Optical Character Recognition*). Perancangan sistem dimulai dari pembuatan berupa perangkat keras Raspberry pi dan kamera. Perangkat lunak yang digunakan menggunakan Bahasa pemrograman python dan library Opencv. Kemudian sistem akan memisahkan warna lain selain hijau disebabkan karena warna dari rambu petunjuk arah berwarna hijau, setelah itu sistem akan mencari citra yang berbentuk kotak, lalu pengolahan karakter huruf serta arah panah. Tahap terakhir perancangan sistem adalah merubah huruf yang sudah dideteksi pada proses sebelumnya dan kemudian dikenali menjadi suara. Setelah perancangan selesai, sistem tersebut diimplementasikan. Sistem yang telah diimplemasikan akan dilakukan pengujian dan analisis. Sistem menguji dengan mendeteksi karakter huruf dan arah panah kemudian dirubah menjadi suara. Waktu minimum dalam mengeksekusi gambar menjadi suara adalah 4.7 detik, maksimum 8.02 detik dan rata rata 6.402 detik. Berdasarkan hasil penelitian dapat disimpulkan bahwa metode *Optical Character Recognition* terbukti mampu mengenali citra yang dideteksi pada data latih. Sehingga mempercepat sistem dalam mengenali kharakter huruf pada citra yang telah dideteksi.

Kata kunci: Rambu Petunjuk Arah, Citra, *Optical Character Recognition*

ABSTRACT

Sign road are one means of providing solutions or the right to call or other road users, about the direction to be taken or the state of the city to be addressed with the name and direction of the place. Signs intended to be required for riders to focus on road compilation of driving. But the rider uses traffic signs without reading the messages implied in them, the information system that can process information from the direction for the rider to focus on the road then the information becomes voice to the rider. In this research was made the introduction system of raspberry pi guidance directions using OCR (Optical Character Recognition) method. Starting from the manufacture of Raspberry pi devices and cameras. Software that uses the Opencv python programming language and library. Then will be used colors that are different from the color of the destination beam, after which the system will look for a box-shaped image, then processing the character letters and arrows direction. The latter is a looping system that has been detected in the previous process and then recognized as sound. Once the design is complete, the system is implemented. Implemented systems will be tested and credited. The printing system by detecting the letters and arrow directions is then converted into sound. The minimum time to execute the image to sound is 4.7 seconds, a maximum of 8.0 seconds and an average of 6.402 seconds. Based on the results of research can be concluded that the Introduction of Optical Character proved able to be done image detected on the data train.

Keywords: Sign road, Imagery, Optical Character Recognition

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR PERSAMAAN.....	xii
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Dasar Teori.....	6
2.2.1 Rambu petunjuk arah.....	6
2.2.2 Pengolahan citra	7
2.2.3 OCR (<i>Optical Character Recognition</i>).....	9
2.2.4 <i>Grayscale</i>	11
2.2.5 Segmentasi.....	11
2.2.6 Metode <i>canny edge detection</i>	12
2.2.7 Otsu thresholding.....	12
2.2.8 Metode Hough Line.....	13
2.2.9 Pengertian Raspberry pi.....	15
2.2.10 Webcam Logitech c170.....	16
2.2.11 Earphone	16

2.2.12 OpenCV Library	17
BAB 3 METODOLOGI	18
3.1 Metode Penelitian	18
3.2 Identifikasi Masalah	18
3.3 Studi literatur	19
3.4 Analisis Kebutuhan	19
3.5 Perancangan Sistem.....	19
3.6 Implementasi	20
3.7 Pengujian	20
3.8 Kesimpulan.....	20
BAB 4 REKAYASA KEBUTUHAN.....	21
4.1 Gambaran Umum Sistem.....	21
4.2 Analisis Kebutuhan Sistem.....	21
4.2.1 Kebutuhan Fungsional.....	21
4.2.2 Kebutuhan Non-Fungsional	22
4.3 Batasan Desain Sistem	23
BAB 5 PERANCANGAN DAN IMPLEMENTASI	24
5.1 Perancangan Sistem.....	24
5.1.1 Perancangan Perangkat Keras	24
5.1.2 Perancangan perangkat lunak.....	24
5.2 Implementasi	32
5.2.1 Implementasi perangkat keras.....	32
5.2.2 Implementasi perangkat lunak	33
BAB 6 PENGUJIAN DAN ANALISIS.....	37
6.1 Pengujian dan analisis fungsional	37
6.1.1 Pengujian kamera webcam Logitech c 170.....	37
6.1.2 Pengujian deteksi rambu petunjuk arah	38
6.1.3 Pengujian Akurasi Pengenalan Karakter	41
6.1.4 Pengujian akurasi pengenalan posisi arah panah	42
6.1.5 Pengujian suara	44
6.2 Pengujian dan Analisis non fungsional	45
6.2.1 Pengujian dan Analisis waktu eksekusi	45



6.2.2 Hasil dan Analisis Pengujian.....	46
BAB 7 PENUTUP	47
7.1 Kesimpulan.....	47
7.2 Saran	47
Daftar pustaka.....	48
LAMPIRAN A KODE PROGRAM.....	50
LAMPIRAN B DATA LATIH RAMBU PETUNJUK ARAH	54



DAFTAR TABEL

Tabel 6. 1 Hasil pengujian deteksi rambu petunjuk arah	40
Tabel 6. 3 Hasil pengujian pengenalan karakter huruf	41
Tabel 6. 4 Hasil pengujian pengenalan posisi arah panah	43
Tabel 6. 5 Hasil pengujian waktu eksekusi sistem	46



DAFTAR GAMBAR

Gambar 2. 1 Contoh Rambu Petunjuk arah	6
Gambar 2. 2 Contoh Rambu Petunjuk arah	6
Gambar 2. 3 Contoh Rambu Petunjuk arah	7
Gambar 2. 4 (a) gambar agak gelap, (b) gambar burung yang telah pengolahan citra	7
Gambar 2. 5 Nilai Hue, Saturation, dan Value	8
Gambar 2. 6 (a) Offline Character Recognition (b) Online Character Recognition. 9	9
Gambar 2. 7 Metode OCR	10
Gambar 2. 8 Setelah melalui proses Canny Edge Detection.....	12
Gambar 2. 9 Citra setelah dilakukan <i>Hough Transform</i>	14
Gambar 2. 10 Citra sebuah garis	15
Gambar 2. 11 Hasil transformasi dari satu titik uji dalam ruang parameter $r - \theta$ 15	15
Gambar 2. 12 Raspberry Pi 3 Model B	16
Gambar 2. 13 webcam Logitech c170.....	16
Gambar 2. 14 <i>earphone</i>	16
Gambar 2. 15 OpenCV (Open Source Computer Vision Library)	17
Gambar 3. 1 Tahapan penelitian untuk pengenalan rambu petunjuk arah.	18
Gambar 4. 1 Diagram blok sistem	21
Gambar 5. 2 Perangkat keras dari sistem	24
Gambar 5. 3 diagram alir sistem	25
Gambar 5. 4 diagram alir perancangan inisialisasi citra	26
Gambar 5. 5 Contoh pengambilan citra	26
Gambar 5. 6 citra gambar pada ukuran asli.....	27
Gambar 5. 7 citra gambar setelah proses <i>resize</i>	27
Gambar 5. 8 gambar yang telah dilakukan <i>masking</i>	27
Gambar 5. 9 diagram alir pencarian objek.....	28
Gambar 5. 10 diagram alir perancangan pendeteksi karakter huruf dan panah . 29	29
Gambar 5. 11 diagram alir mendeteksi garis pada rambu petunjuk arah.....	30
Gambar 5. 12 diagram alir menentukan posisi dari arah panah	31
Gambar 5. 13 Perancangan <i>text to sound</i>	31
Gambar 5. 14 Hasil dari perancangan perangkat keras.....	32
Gambar 5. 15 Hasil akhir dari perangkat keras.....	32
Gambar 5. 16 Program inisialisasi <i>library</i>	33
Gambar 5. 17 Program inisialisasi warna hijau	33
Gambar 5. 18 Program pencarian objek berbentuk kotak	34
Gambar 5. 19 Program image to teks	35
Gambar 5. 20 Program penentuan posisi dari arah panah.....	36
Gambar 5. 21 Program <i>text to soud</i>	36
Gambar 6. 1 pengujian kamera webcam Logitech c 170.....	38
Gambar 6. 2 pengujian pengambilan gambar	38
Gambar 6. 3 Hasil pengujian koversi teks ke suara	45

DAFTAR PERSAMAAN

PERSAMAAN (2. 1)	8
PERSAMAAN (2. 2)	8
PERSAMAAN (2. 3)	9
PERSAMAAN (2. 4)	9
PERSAMAAN (2. 5)	11
PERSAMAAN (2. 6)	13
PERSAMAAN (2. 7)	14
PERSAMAAN (2. 8)	14
PERSAMAAN (2. 9)	14
PERSAMAAN (5. 1)	28



DAFTAR LAMPIRAN

LAMPIRAN A KODE PROGRAM.....	50
LAMPIRAN B DATA LATIH RAMBU PETUNJUK ARAH	54



BAB 1 PENDAHULUAN

1.1 Latar belakang

Rambu lalu lintas adalah sebuah sarana yang berfungsi sebagai peringatan, larangan, perintah, atau petunjuk bagi pengguna Jalan, berupa lambang, huruf, angka, dan kalimat. Menurut Peraturan Menteri Perhubungan Nomor 13 Tahun 2014 terdapat 5 rambu lalu lintas yang berlaku di Indonesia yaitu rambu peringatan, rambu larangan, rambu perintah, rambu petunjuk arah, rambu tambahan, dan rambu nomor rute jalan. Rambu lalu lintas berguna sebagai petunjuk dan mengurangi kecelakaan lalu lintas. Khususnya pada rambu petunjuk arah berguna untuk membantu pengguna jalan sampai ke tujuan.

Rambu petunjuk arah seringkali terlewat dan sedikitnya mengganggu pada fokus pengendara ketika berkendara, perkembangan teknologi sistem cerdas dalam proses pengenalan rambu lalu lintas sudah banyak yang telah dilakukan dalam hal pengenalan gambar maupun video. Pengenalan rambu lalu lintas telah banyak dilakukan menggunakan kamera untuk menangkap gambar lalu lintas dalam melakukan pengenalan citra. Nama jalan atau petunjuk arah memiliki warna dasar hijau pada rambu petunjuk jalan. Dalam Peraturan Menteri Perhubungan Nomor 13 Tahun 2014 rambu petunjuk dengan warna dasar hijau, warna garis tepi putih, warna lambang putih dan warna huruf dan/atau angka putih.

Pada penelitian sebelumnya pengenalan rambu lalu lintas dilakukan oleh Greenhalgh & Mirmehd dengan membagi 3 bagian yang akan dideteksi yaitu atas, kiri dan kanan yang memungkinkan terdapat rambu lalu lintas menggunakan metode MSERs (*Maximally stable extremal regions*) dan menggunakan *hue*, *saturation* dan *value* dalam mendeteksi warna pada rambu petunjuk arah (Greenhalgh & Mirmehdi, 2015). Pada pendeteksian gambar dilakukan frame secara berurutan sehingga mendeteksi kemungkinan kata yang dideteksi pada gambar.

Pengenalan rambu petunjuk arah yang dilakukan pada tahun 2018 oleh Yingying Zhu dengan dilakukannya segmentasi dalam pengenalan rambu lalu lintas. Terdapat banyak cara yang dilakukan dalam pengenalan, satu diantaranya dengan membagi objek menjadi 3 yaitu berbasis warna, bentuk dan *sliding window*. Berbasis warna menggunakan warna RGB yang dinormalisasikan sehingga memisahkan warna yang akan dideteksi. Berbasis bentuk yang sering digunakan berbentuk segitiga, persegi dan lingkaran. *Sliding window* menggunakan pengklasifikasian bentuk yang terdapat bentuk rambu lalu lintas. Mendeteksi rambu lalu lintas menggunakan ROI (*Region Of Interest*) dengan melakukan segmentasi, terakhir dari citra yang dideteksi akan dilakukan *crop* dan dilakukan pendeteksi huruf (Yingying Zhu, 2018).

Pengenalan rambu petunjuk arah yang dilakukan oleh Satria Priambada dilakukan dengan memproses gambar menggunakan metode

ROI (*Region Of Interest*). Proses deteksi teks dalam video sudah banyak dilakukan dalam penelitian dan digunakan untuk mobil *self-driving* guna menunjukkan jalan yang sudah benar. Penentuan warna dilakukan dengan pemrosesan RGB ke HSV, sistem melakukan rentang warna hijau, petunjuk arah Indonesia menggunakan hue antara 80-240 (Satria Priambada, n.d.).

Berdasarkan latar belakang tersebut penulis mengusulkan penelitian yang berjudul “Rancang Bangun Sistem Pengenalan Rambu Petunjuk Arah Berbasis Raspberry Pi Menggunakan Metode OCR (*Optical Character Recognition*). Pada penelitian ini menggunakan kamera *webcam* dan Raspberry Pi sebagai perangkat keras. Sistem ini bertujuan untuk mengenali huruf dan arah panah yang terdapat pada rambu petunjuk lalu lintas, hasil akhir dari penelitian ini adalah dihasilkannya suara yang dikeluarkan melalui *earphone*.

1.2 Rumusan masalah

Berdasarkan latar belakang yang dibuat maka penulis membuat rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan metode OCR (*Optical Character Recognition*) untuk mendeteksi rambu petunjuk arah ?
2. Bagaimana tingkat akurasi dalam menemukan teks ditemukan pada rambu petunjuk arah yang dideteksi ?

1.3 Tujuan

Berdasarkan rumusan masalah yang disusun, adapun tujuan dari penulis adalah sebagai berikut:

1. Untuk mengetahui cara menerapkan metode OCR (*Optical Character Recognition*) pada *raspberry pi* dengan menggunakan *library opencv*.
2. Untuk mengetahui cara membangun rancang bangun untuk membaca rambu petunjuk arah.

1.4 Manfaat

Adapun manfaat yang ingin diberikan oleh penulis dalam penelitian ini adalah sebagai berikut:

1. Sebagai langkah untuk memberikan alternatif kemudahan *open source* untuk pengemudi dalam membaca rambu petunjuk arah.
2. Sebagai solusi untuk memberikan manfaat dalam pemberian informasi rambu petunjuk arah.

1.5 Batasan masalah

Batasan masalah berdasarkan rumusan masalah yang telah diuraikan, yaitu :

1. Implementasi penelitian ini sebatas rambu lalu lintas petunjuk arah yang mempunyai warna dasar hijau dan membedakan dengan objek yang lain.
2. Rambu petunjuk arah yang dideteksi hanya yang memiliki satu arah panah dalam rambu petunjuk arah.
3. Implementasi penelitian ini dilakukan pada saat pagi, siang dan sore yang memiliki cahaya matahari.
4. Batas kecepatan maksimal kendaraan 30 km/jam.

1.6 Sistematika pembahasan

BAB I PENDAHULUAN

Bab ini membahas masalah terkait dalam penelitian ini menggunakan metode *Optical Character Recognition* dalam mendeteksi rambu petunjuk arah. berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika pembahasan

BAB II LANDASAN KEPUSTAKAAN

Bab ini membahas tentang kajian putaka yang berisi studi literatur, pengumpulan data, teori yang digunakan, dasar teori.

BAB III METODOLOGI

Bab ini menjelaskan tentang gambaran umum dan implementasi perancangan penelitian ini yang digunakan dalam proses pengambilan rambu petunjuk arah serta melakukan pengambilan data serta metode yang digunakan dalam perancangan dalam bentuk pengujian dan analisis.

BAB IV REKAYASA KEBUTUHAN

Bab ini tentang analisis kebutuhan dalam proses perancangan dalam implementasi metode *Optical Character Recognition*. Pada bab ini menjelaskan tentang tahapan fungsional dan non-fungsional yang dibutuhkan dalam proses membangun sistem.

BAB V PERANCANGAN DAN IMPLEMENTASI

Bab ini tentang perancangan dan implementasi sistem proses pengujian terhadap sistem. Dalam bab ini membahas tentang kebutuhan fungsional dan non-fungsional yang diterapkan pada sistem.

BAB VI PENGUJIAN DAN ANALISIS

Bab ini membahas tentang pengujian sistem yang telah dilakukan implementasi kepada sistem dengan menggunakan metode *Optical Character Recognition* untuk mendeteksi rambu petunjuk arah.

BAB VII PENUTUP

Bab ini berisi tentang kesimpulan dari analisis yang telah dilakukan dari perancangan alat yang telah dilakukan dan sara yang berisi tentang evaluasi erancangan dan pengembangan selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada kajian pustaka untuk dilakukan untuk mengkaji beberapa penelitian yang sudah, berkaitan dengan pengenalan rambu penunjuk arah menggunakan metode OCR (*Optical Character Recognition*), pada kajian pustaka dilakukan perbandingan penelitian sebelumnya dan penelitian yang dilakukan.

Greenhalgh, dkk, 2015 penelitian ini yang berjudul "*Recognizing Text-Based Traffic Signs*" pengenalan rambu lalu lintas ketika berkendara. Pengenalan terdiri dari 2 tahap utama yaitu deteksi dan pengenalan. Tahap deteksi memanfaatkan struktur pemandangan yaitu ukuran dan lokasi jalan di dalam bingkai untuk menentukan daerah ditempatkan dan mencari tanda rambu lalu lintas, hasil ini dimasukkan kedalam *vanishing point* (VP). Selanjutnya rambu lalu lintas menentukan region dengan menggunakan kombinasi MSERs dan *hue, saturasi and value* (HSV) thresholding.

(R & Sharmila, 2016) penelitian ini berjudul "*Emergency exit sign detection system for visually impaired people*" penelitian ini berfokus kepada tanda darurat berbasis *smartphone* yang sering digunakan tiap hari. Penelitian ini digunakan terhadap orang yang mengalami gangguan penglihatan dengan dengan tujuan perangkat ini memberi informasi untuk jalur darurat yang dipakai menggunakan *smartphone*. Pada tanda keluar darurat terdiri latar belakang hijau tua dengan tulisan berwarna putih serta simbol manusia berlari dengan tanda anak panah tetapi simbol orang berlari berbeda disemua tempat dan tanda keluar darurat bermacam macam seperti pintu keluar darurat, pintu keluar api, pintu keluar dll. Aplikasi ini menangkap gambar menggunakan kamera ponsel. Citra ini kemudian diolah untuk menemukan arah jalan keluar. Papan nama harus terdeteksi terlepas dari parameter registrasi tertentu dan harus dideteksi dalam gambar kompleks.

Berdasarkan beberapa penelitian di atas, penulis mengusulkan suatu sistem pengenalan rambu petunjuk arah dengan menggunakan metode OCR (*Optical Character Recognition*) berbasis raspberry pi dan kamera. Perbedaan pada penulis pertama adalah peneliti menggunakan menggunakan metode MSERs dalam pengelompokan rambu penunjuk arah. Pada penulis kedua adalah penelitian tersebut dikhususkan pada penyandang tunanetra dan implementasi pada *smatphone* berbasis android. Diharapkan dalam penggunaan sistem ini dapat membantu pengendara dalam mengenali rambu penunjuk arah. Selanjutnya penguji akan mengenali rambu petunjuk arah penggunaan metode ini dapat membantu pengendara dalam menerjemahkan rambu petunjuk arah berupa suara.

2.2 Dasar Teori

Dasar teori merupakan teori yang akan dilakukan pada penelitian ini berupa metode dan algoritma yang digunakan pada penelitian ini.

2.2.1 Rambu petunjuk arah

Rambu petunjuk arah merupakan sebuah sarana yang berguna dalam memberi informasi letak sebuah lokasi atau tempat yang bertujuan membantu pengendara. Rambu petunjuk arah menggunakan huruf kapital pada huruf pertama, dan selanjutnya menggunakan huruf kecil dan/atau seluruhnya menggunakan huruf kapital dan/atau huruf kecil. Khusus rambu petunjuk jurusan kawasan dan objek wisata dinyatakan dengan warna dasar coklat dengan lambang dan/atau tulisan warna putih.

Rambu petunjuk yang menyatakan tempat fasilitas umum, batas wilayah suatu daerah, situasi jalan, dan rambu berupa kata-kata serta tempat khusus dinyatakan dengan warna dasar biru.



Gambar 2. 1 Contoh Rambu Petunjuk arah

Sumber : Peraturan Menteri Perhubungan RI (2014)



Gambar 2. 2 Contoh Rambu Petunjuk arah

Sumber : Peraturan Menteri Perhubungan RI (2014)



Gambar 2. 3 Contoh Rambu Petunjuk arah

Sumber : Peraturan Menteri Perhubungan RI (2014)

2.2.2 Pengolahan citra

Pengolahan citra adalah pemrosesan gambar, khususnya dengan menggunakan komputer, menjadi gambar yang mempunyai kualitas lebih baik (Maulida, 2010). Proses pengolahan gambar terdapat pada gambar 2.4



Gambar 2. 4 (a) gambar agak gelap, (b) gambar burung yang telah pengolahan citra

Sumber : Fauzi, 2013

Pada gambar 2.4 (a) terlihat bahwa gambar agak gelap menunjukkan kurang jika dilakukan pemrosesan gambar. Pada gambar 2.4 (b) dapat dilihat bahwa gambar yang telah dilakukan pengolahan citra akan terlihat lebih baik dengan dilakukan pengolahan citra dengan menaikkan kontras pada gambar (Fauzi, et al., 2013). Proses tersebut dapat dilakukan apabila :

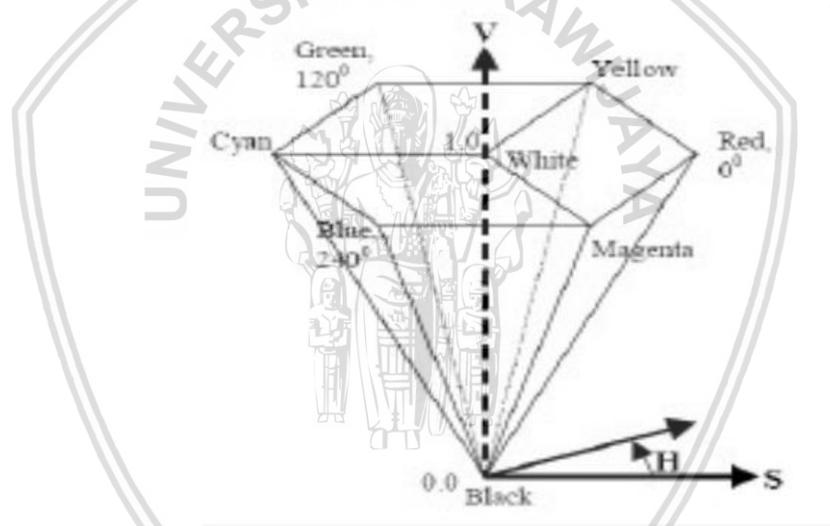
1. Perbaikan gambar dapat dilakukan untuk meningkatkan kualitas dari gambar yang dimana membutuhkan informasi dari gambar.
2. Nilai pada gambar perlu dikelompokkan, dicocokkan, atau diukur.
3. Sebagian gambar perlu dilakukan peggabungan dengan gambar lain sehingga menaikkan kualitas dari gambar.

2.2.2.1 Warna RGB

Warna RGB (*Red, Green, Blue*) adalah kombinasi warna primer yaitu merah, hijau, dan biru, yang biasa digunakan oleh monitor komputer atau televisi. Warna yang dihasilkan berasal dari kombinasi tiga warna dan masing - masing memiliki nilai 8 bit merah, 8 bit hijau, dan 8 bit biru. Campuran dari RGB ini dengan porposisi seimbang akan menghasilkan nuansa warna kelabu. Jika ketiga warna ini disaturasikan penuh, maka akan menghasilkan warna putih (Pratt, 2007).

2.2.2.2 Warna HSV

Model HSV (*Hue, Saturation, Value*) menunjukkan ruang warna dalam bentuk tiga komponen utama yaitu hue, saturation, dan value (atau disebut juga brightness). Hue adalah sudut dari 0 sampai 360 derajat. Biasanya 0 adalah merah, 60 derajat adalah kuning, 120 derajat adalah hijau, 180 derajat adalah cyan, 240 derajat adalah biru dan 300 derajat adalah magenta seperti pada Gambar 2.5. Hue menunjukkan jenis warna (seperti merah, biru, kuning), yaitu tempat warna tersebut ditemukan dalam spectrum warna. Saturasi (saturation) suatu warna adalah ukuran seberapa besar kemurnian dari warna tersebut akibat pengaruh dari warna putih. Seperti warna merah, dengan pengaruh warna putih, warna merah menjadi bervariasi dari warna merah menuju merah muda, yang artinya hue masih tetap bernilai merah tetapi nilai saturasinya berkurang. Komponen HSV berikutnya adalah nilai value atau disebut juga intensitas, yaitu ukuran seberapa besar kecerahan suatu warna atau seberapa besar cahaya datang dari suatu warna. Value memiliki nilai dengan jangkauan 0% sampai 100% (Pratt, 2007).



Gambar 2. 5 Nilai Hue, Saturation, dan Value

Sumber : Prat, 2007

Konversi warna RGB ke HSV seperti pada Persamaan 1 sampai dengan 5 (Gonzales, 2008).

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (2. 1)$$

$$V = \max(r, g, b) \quad (2. 2)$$



$$H = \begin{cases} 0 & \text{If } S = 0 \\ \frac{60 * (g - b)}{S * V} & \text{If } V = r \\ 60 * \left[2 + \frac{(b - r)}{S * V} \right] & \text{If } V = g \\ 60 * \left[2 + \frac{(r - g)}{S * V} \right] & \text{If } V = b \end{cases} \quad (2.3)$$

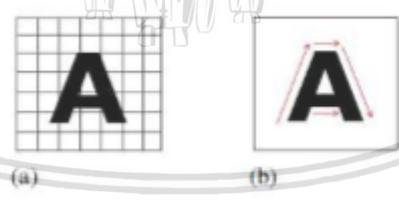
$$S = \begin{cases} 0 & \text{If } V = 0 \\ V - \frac{\min(r, g, b)}{V} & \text{If } V > 0 \end{cases} \quad (2.4)$$

$$H = H + 360 \quad \text{jika } H < 0$$

2.2.3 OCR (Optical Character Recognition)

Optical Character Recognition yaitu sistem yang akan menerjemahkan huruf yang dapat dibaca baik dari tulisan cetak maupun tulisan tangan. Pada sistem pengenalan huruf akan meningkatkan nilai fleksibilitas ataupun kemampuan dan kecerdasan sistem komputer sehingga pengenalan huruf yang memiliki tingkat pengenalan yang lebih baik.

Character Recognition bertugas untuk mengenali tulisan didalam mengenali karakter tulisan dalam gambar dan merubahnya kedalam American Standad Code for Information Interchange (ASCII) atau bahasa mesin lainnya yang setara dan dapat diedit. Terdapat dua macam Character Recognition, antara lain: Offline dan Online Character Recognition (Rao, Sasrty, ChakrCarthy, & Kalyanchakravarthi, 2016).



Gambar 2. 6 (a) Offline Character Recognition (b) Online Character Recognition

Sumber : Rao, Sasrty, ChakrCarthy, & Kalyanchakravarthi (2016).

Secara umum metode OCR pada proses pengolahan citra :



Gambar 2. 7 Metode OCR

Sumber : Sandhika (2014)

a. *File Input*

File input berupa gambar yang memiliki format bmp atau jpg.

b. *Preprocessing*

Preprocessing merupakan proses untuk menghilangkan bagian yang tidak perlu pada gambar yang telah dimasukkan.

c. *Segmentasi*

Segmentasi yaitu proses untuk mendapatkan Region of Interest (ROI). Proses ini akan memisahkan foreground yang merupakan ROI atau objek yang dicari dengan background yang merupakan region selain objek yang dicari. Proses segmentasi sebuah citra akan menghasilkan nilai biner 0 dan 1 pada gambar dengan ketentuan.

d. *Normalisasi*

Normalisasi adalah proses merubah dimensi *region* tiap karakter dan ketebalan karakter.

e. *Ekstraksi ciri*

Ekstraksi ciri adalah proses untuk mengambil ciri-ciri tertentu dari karakter yang diamati.

f. *Recognition*

Recognition merupakan proses untuk mengenali karakter yang diamati dengan cara membandingkan ciri-ciri karakter yang diperoleh dengan ciri-ciri karakter yang ada pada basis data.

2.2.4 Grayscale

Pada proses *graycalling* dilakukan untuk mempermudah dalam proses pengolahan citra. Proses ini mengubah citra dengan warna yang beragam menjadi citra 8 bit keabuan. Persamaan yang digunakan (Syafi'i, et al., 2015) :

$$\text{Gray} = 0,2989 * R + 0,5870 * G + 0,1140 * B \quad (2.5)$$

Dimana :

R : Nilai intensitas komponen warna merah

G : Nilai intensitas komponen warna hijau

B : Nilai intensitas komponen warna biru

Proses ini dilakukan pada setiap gambar yang memiliki 3 komponen warna yaitu *red, green, blue* (RGB). Dengan persamaan diatas akan membuat warna baru yang memiliki entitas 0 – 255, ini digunakan untuk karena gambar yang digunakan akan bernilai 8 bit sehingga terdapat 28 warna atau 255. (Fauzi, et al., 2013).

2.2.5 Segmentasi

Proses segmentasi citra dilakukan untuk membentuk objek citra. Proses segmentasi mengelompokkan piksel-piksel bertetangga berdasarkan kemiripan nilai piksel. Algoritma segmentasi dapat dikelompokkan menjadi tiga kelompok, yaitu:

a. Spectral-based segmentation

Spectral-based segmentation mengelompokkan piksel berdasarkan analisis data pada *feature space*. Contoh umum dari pendekatan ini adalah *clustering* dan *histogram thresholding*.

b. Region-based segmentation

Edge detection segmentation menentukan batas antara dua region yang memiliki homogenitas dalam karakteristiknya. Contoh dari pendekatan ini adalah algoritma *watershed*, di mana batas segmen ditentukan melalui proses iterasi dengan menganggap nilai piksel sebagai data ketinggian dan batas segmen ditentukan dengan logika seperti memisahkan daerah aliran sungai.

c. Edge detection.

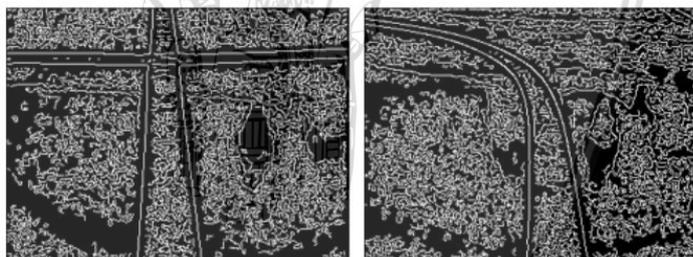
Region-based segmentation dapat dibagi menjadi *region growing/merging*, *region dividing/splitting* dan *hybrid method*. *Region growing/merging* mengambil beberapa piksel sebagai "*seed*" yang kemudian tumbuh (*grow*) menjadi region di sekitarnya berdasarkan kriteria homogenitas tertentu. *Region dividing/splitting* secara berulang

memecah citra menjadi sekumpulan region terpisah yang setiap region tersebut koheren secara internal.

2.2.6 Metode *canny edge detection*

Edge Detection Batas garis atau Lane Boundaries didefinisikan oleh kontras antara permukaan lintasan dan garis pada jalan. Kontras ini adalah edges dari gambar. Oleh karena itu edge detectors sangat penting untuk menentukan dimana batas garis atau lane boundaries. Hal itu juga mengurangi jumlah pembelajaran data yang dibutuhkan untuk menyederhanakan gambar, jika outline dari sebuah lintasan dapat diekstrak dari sebuah gambar.

Edge detector yang digunakan pada algoritma ini dan satu-satunya yang dapat menghasilkan gambar edge terbaik dari semua edge detector adalah 'canny' edge detector. Sangat penting memiliki pendeteksi pinggiran atau edge detection yang dapat memilih threshold secara otomatis, Tetapi threshold otomatis yang ada pada standar canny menghasilkan informasi pinggiran atau edge yang terlalu banyak. Dengan membuat sedikit modifikasi terhadap edge detection yang dihasilkan oleh canny memberikan hasil yang lebih baik. Canny edge detector juga memiliki karakteristik yang membuatnya tidak menghasilkan noise seperti pendekatan-pendekatan yang lainnya. Dibawah adalah gambar yang dihasilkan dari Canny Detection. (Abdulhakam.AM.Assidiq, et al., 2008)



Gambar 2. 8 Setelah melalui proses Canny Edge Detection

Sumber : Abdulhakam.AM.Assidiq, et al., 2008

2.2.7 Otsu thresholding

Otsu *thresholding* merupakan suatu metode untuk yang membantu untuk melakukan segmentasi citra digital dengan cara mencari threshold menggunakan analisa statistik histogram. Metode Otsu thresholding diperkenalkan pertama kali oleh Nobuyuki Otsu, dalam jurnal ilmiahnya yang berjudul "A Threshold Selection Method from Grayscale Histogram" pada tahun 1979 Secara definitif, proses tresholding adalah proses untuk mengubah citra grayscale multi value, menjadi citra biner bivalued (dua nilai saja). Proses tersebut dapat di simbolkan dengan notasi matematis berikut:

$$g(x, y) \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (2.6)$$

Di mana $g(x,y)$ adalah nilai pixel hasil threshold pada baris x dan kolom y , $f(x,y)$ adalah nilai pixel dari citra input asal pada baris x dan kolom y , sedangkan T adalah nilai threshold sebagai batas nilai keabuan untuk mengkonversi nilai pixel $f(x,y)$ menjadi nilai pixel $g(x,y)$. Otsu thresholding didasarkan pada ide untuk menemukan threshold yang meminimasi bobot variansi within-class. Hal tersebut sama saja dengan memaksimalkan variansi between-class. Otsu *thresholding* bekerja pada citra dengan model warna grayscale (Aristyagama, December 2015).

2.2.8 Metode Hough Line

Hough Transform adalah cara yang paling populer dan metode yang paling efisien dalam mendeteksi garis pada gambar. Untuk setiap *edge pixel* (x,y) pada $x - y$ *plane*, nilai dari p terhadap setiap ρ dihitung dan dilakukan vote ke grid yang relevan. Parameter untuk menentukan kandidat garis lurus dihasilkan dengan mendeteksi *local peak* dari *cumulative matrix*.

Dalam penggunaan *Hough Transform* parameter solusi dalam menggunakan metode Hough Transform didapatkan dari pencarian secara menyeluruh, yang akan menyebabkan peningkatan beban komputasi. Untuk mengurangi waktu pengerjaan dan menggunakan secara maksimal informasi yang didapatkan. Sebuah improvisasi *Hough Transform* digunakan untuk mempercepat perhitungan. Ketika menghitung parameter dari setiap poin, nilai ρ menurut arah dari kontur pixelnya digunakan untuk mengecilkan ruang pencarian.

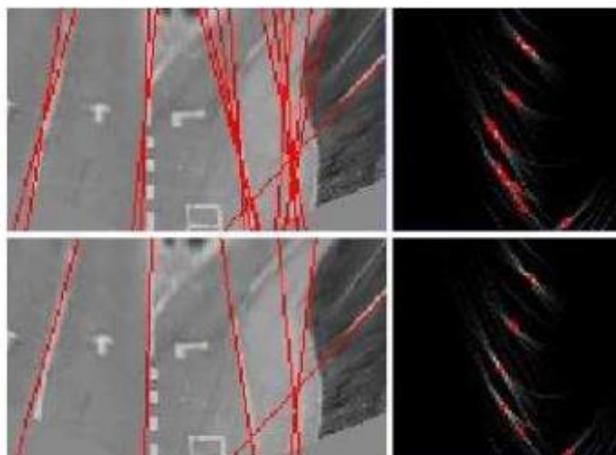
Berikut adalah langkah-langkah *Hough Transform* :

- Hitung *Hough Transform*
- Cari *local peak* dari *cumulative matrix*

Nilai dari semua *grid* dihitung untuk mencari tahu *local peaks*. Setiap *peak* menunjukkan kandidat garis lurus.

- *Grouping pada Hough Space*

Penggunaan teknik *grouping* pada *Hough Space* sangat dibutuhkan untuk menghilangkan beberapa respon yang berasal dari garis yang mirip atau sama. Ketika lebih dari satu *peak* ditemukan dalam daerah yang kecil, rata-rata dari *peak* tersebut dihitung untuk menjadikan *peak-peak* sebelumnya menjadi satu.



Gambar 2. 9 Citra setelah dilakukan *Hough Transform*

Sumber : Halimatus Sa'diyah, 2011

Hasil dari langkah diatas ada pada gambar diatas. Gambar pada baris pertama mendeteksi garis lurus dan Hough Space sebelum di grouping. Pada baris kedua hasil dari garis lurus dan Hough Space setelah di grouping. Respon – respon yang berasal dari garis yang sama telah dikelompokkan pada langkah ini.

Transformasi Hough digunakan dalam bentuk parametrik dan melakukan pendeteksian gambar terbanyak dengan menentukan nilai dari parameter yang digunakan. Apabila dalam gambar memiliki beberapa garis yang saling memotong dalam satu titik, titi tersebut akan dilakukan transformasi kedalam ruang parameter m - c akan didapatkan bahwa transformasi dalam parameter yaitu garis lurus dengan persamaan garis lurus dapat dinyatakan pada persamaan sebagai berikut (Halimatus Sa'diyah, 2011)

$$y_i = mx_i + c \quad (2.8)$$

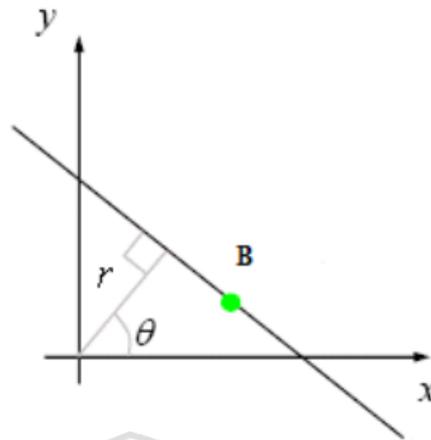
Sebaliknya jika dalam citra terdapat sebuah garis lurus, dimana garis tersebut mempunyai persamaan seperti persamaan (2.2), maka apabila ditransformasi ke dalam ruang parameter $m - c$ akan diperoleh bahwa transformasi dalam ruang parameter $m - c$ adalah beberapa garis yang saling berpotongan dalam suatu titik. Persamaan transformasi diperoleh dengan cara memanipulasi persamaan garisnya yaitu persamaan (2.2) menjadi bentuk parametrik menjadi persamaan.

$$c = y_i - mx_i \quad (2.7)$$

Perhitungan transformasi dilakukan dengan cara menghitung tiap titik dalam citra kedalam semua variasi nilai $m - c$. Dalam kenyataannya, apabila ditemui sebuah garis vertikal, maka akan terjadi masalah dalam penghitungannya dikarenakan garis vertikal mempunyai nilai gradien kemiringan m yang besarnya tak berhingga ∞ . Sebagai alternatifnya digunakan persamaan (2.4).

$$r = x \cos \theta + y \sin \theta \quad (2.9)$$

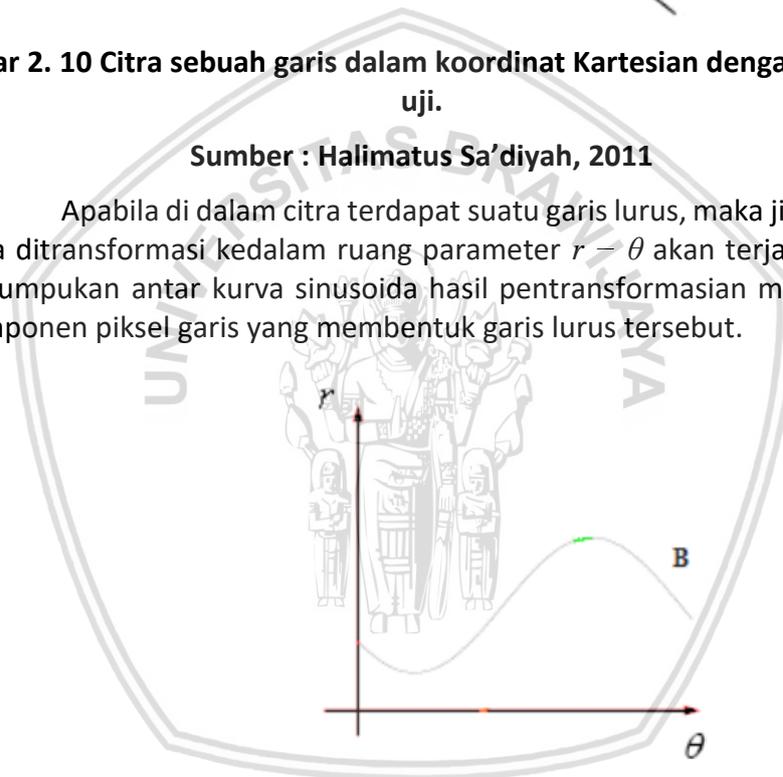
dimana gambar koordinat kartesiannya ditunjukkan pada Gambar 2.8.



Gambar 2. 10 Citra sebuah garis dalam koordinat Kartesian dengan satu titik uji.

Sumber : Halimatus Sa'diyah, 2011

Apabila di dalam citra terdapat suatu garis lurus, maka jika garis lurus citra ditransformasi kedalam ruang parameter $r - \theta$ akan terjadi suatu titik penumpukan antar kurva sinusoida hasil pentransformasian masing-masing komponen piksel garis yang membentuk garis lurus tersebut.



Gambar 2. 11 Hasil transformasi dari satu titik uji dalam ruang parameter $r - \theta$

Sumber : Halimatus Sa'diyah, 2011

2.2.9 Pengertian Raspberry pi

Raspberry Pi adalah modul mikro komputer yang juga mempunyai nilai input dan output digital port seperti pada *board microcontroller*. Pada Raspberry pi merupakan mikro komputer telah disediakan 4 port USB yang dapat digunakan untuk keyboard, flashdisc dan lainnya, dilengkapi dengan port HDMI guna display dapat digunakan pada TV ataupun monitor, jack *earphone* untuk output suara. Raspberry Pi mempunyai sistem operasi berbasis os linux yang bersifat *open source*.





Gambar 2. 12 Raspberry Pi 3 Model B

Sumber : Raspberry.org

2.2.10 Webcam Logitech c170

Webcam Logitech c170 adalah webcam memiliki spesifikasi 5 Megapiksel, menggunakan koneksi via USB. Merekam video yang memiliki resolusi XVGA (1024 x 768). USB 2.0 dalam penggunaan koneksi ke perangkat yang digunakan.



Sumber : www.logitech.com

Gambar 2. 13 webcam Logitech c170

2.2.11 Earphone

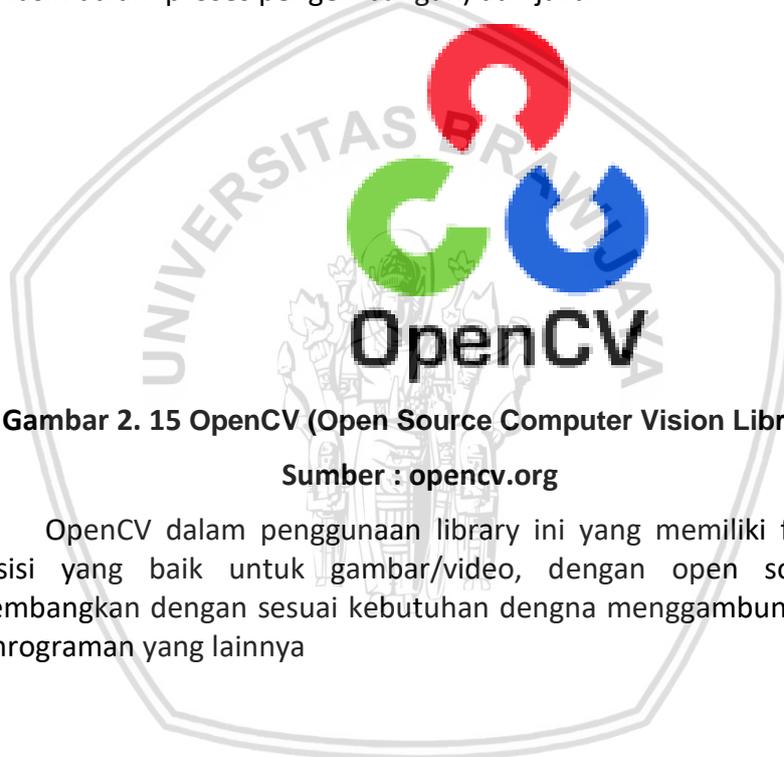
Earphone merupakan perangkat keras yang mempunyai guna untuk mengubah energi listrik menjadi suara. Digunakan untuk mendengarkan suara dari komputer .



Gambar 2. 14 earphone

2.2.12 OpenCV Library

OpenCV merupakan library gratis yang dikembangkan oleh Intel Corporation. Library mempunyai fungsi sebagai pengolahan citra (*image processing*) computer vision dan API (Application Programming Interface) untuk *image processing* high level maupun low level sebagai optimasi aplikasi realtime. OpenCV sudah sangat sering digunakan dalam computer vision dan bidang *image processing* karena library ini merupakan open source yang digunakan secara gratis dan dapat dikembangkan sesuai dengan kebutuhan maupun kemampuan. openCV telah dikembangkan dengan para developer yang dimana telah berbagai macam versi bahasa pemrograman yaitu C/C++, C#(masih dalam proses pengembangan) dan java.



Gambar 2. 15 OpenCV (Open Source Computer Vision Library)

Sumber : opencv.org

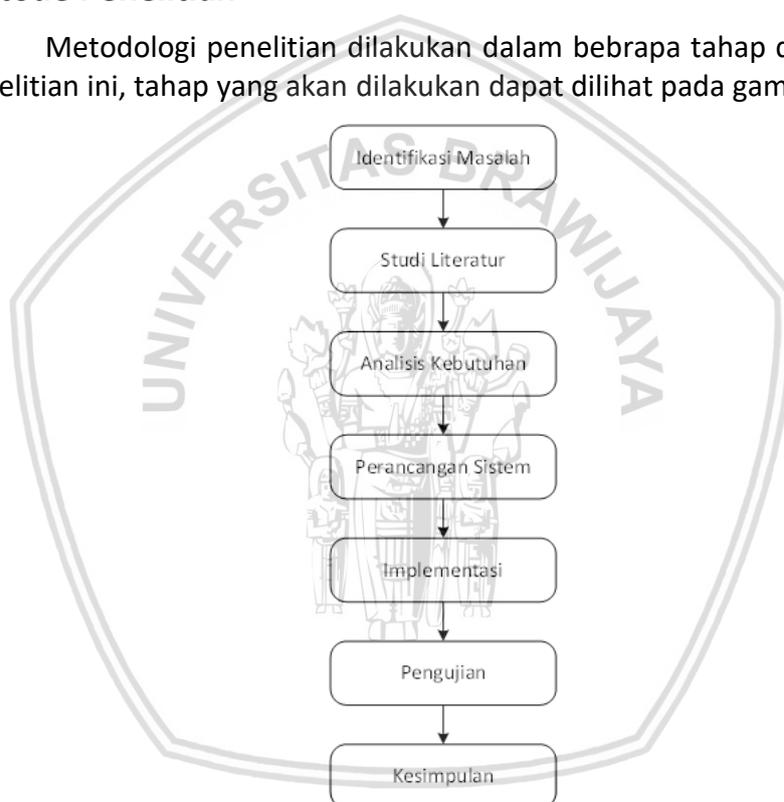
OpenCV dalam penggunaan library ini yang memiliki fungsi fungsi akusisi yang baik untuk gambar/video, dengan open source dapat dikembangkan dengan sesuai kebutuhan dengna menggabungkan bahasa pemrograman yang lainnya

BAB 3 METODOLOGI

Pada bab metodologi menjelaskan tentang tahapan dalam melakukan penelitian ini yang berjudul Rancang Bangun Sistem pengenalan rambu petunjuk arah berbasis raspberry pi menggunakan metode OCR (*Optical Character Recognition*). Bab ini menjelaskan tentang langkah langkah dalam proses pengenalan rambu, penjelasan langkah langkah dalam penelitian ini akan dilakukan beberapa tahap agar peneliti dapat menyelesaikan permasalahan dalam bab sebelumnya. Ada beberapa tahapan yang akan dilakukan peneliti yaitu sebagai berikut

3.1 Metode Penelitian

Metodologi penelitian dilakukan dalam beberapa tahap dalam proses penelitian ini, tahap yang akan dilakukan dapat dilihat pada gambar 3.1.



Gambar 3. 1 Tahapan penelitian untuk pengenalan rambu petunjuk arah.

3.2 Identifikasi Masalah

Tahap ini adalah awal dari penelitian adalah identifikasi masalah yang bertujuan untuk menentukan penelitian yang akan dilakukan dengan tahap ini akan menemukan objek penelitian yang akan dibahas dalam penelitian. Objek yang akan digunakan sebagai permasalahan dalam penelitian akan dimuat dalam latar belakang, solusi yang akan dilakukan dalam penelitian dan penelitian yang telah dilakukan sehingga penelitian yang dilakukan tidak berulang lagi dalam penelitian ini. Masalah yang diambil dalam penelitian ini adalah cara membaca rambu petunjuk arah yang akan di *convert text to sound*

pengendara akan mendapatkan peringatan suara dari gambar rambu petunjuk arah yang telah dideteksi.

3.3 Studi literatur

Tahap ini yaitu untuk mempelajari literature yang berhubungan dengan penelitian ini, dengan mempelajari hasil dari penelitian sebelumnya akan membuat penelitian yang sesuai dengan masalah yang akan diselesaikan. Ada beberapa pustaka yang telah melakukan penelitian berkaitan dengan alat pembaca rambu petunjuk arah. Pada penelitian ini menggunakan metode OCR untuk melakukan pengolahan citra. Pustaka yang berkaitan dengan penelitian ini yaitu :

- a. *Recognizing Text-Based Traffic Signs*
- b. *Emergency exit sign detection system for visually impaired people*

3.4 Analisis Kebutuhan

Tahap ini yaitu untuk melakukan analisa kebutuhan-kebutuhan yang akan digunakan dalam perancangan sistem penelitian ini. Analisis kebutuhan ini merupakan tahapan setelah menemukan objek masalah yang akan diselesaikan dalam penelitian sehingga tepat pada menentukan kebutuhan-kebutuhan sistem. Dimana kebutuhan-kebutuhan yang akan digunakan antara lain

1. Kebutuhan Hardware antara lain :
 - a. Raspberry Pi
 - b. Layar Monitor
 - c. Keyboard
 - d. Mouse
 - e. Webcam 5 pixel
 - f. Earphone
2. Kebutuhan software antara lain :
 - a. Noobs (OS Raspberry pi)
 - b. OpenCV 3.2
 - c. Numpy
 - d. Pyttsx

3.5 Perancangan Sistem

Tahap ini adalah perancangan sistem yang akan dilakukan setelah semua kebutuhan yang dilakukan sebelum tahap ini dengan menggabungkan semua kebutuhan kebutuhan yang diperlukan untuk penelitian. Pada

perancangan terbagi 2 yaitu perancangan perangkat lunak dan perancangan perangkat keras.

Pada perancangan perangkat keras bertujuan untuk memenuhi kebutuhan sistem pada pengenalan rambu petunjuk arah yaitu berupa kamera Logitech c170, Raspberry pi 3, dan earphone. Selanjutnya perancangan perangkat lunak bertujuan agar sistem dapat mengenali rambu petunjuk arah agar memenuhi penelitian ini.

3.6 Implementasi

Pada tahapan ini merupakan tahap membangun sistem sesuai dengan masalah yang akan diselesaikan sebelumnya dengan menggunakan semua pada tahap sebelumnya dikerjakan pada objek yang dituju. Objek yang dituju adalah rambu petunjuk arah yang hanya memiliki satu arah panah yang akan dilakukan pengolahan citra dengan camera yang digunakan untuk menangkap gambar rambu petunjuk arah. Implementasi sistem ini menggunakan bahasa python dengan melakukan *thresholding* pada gambar yang telah ditangkap. Dieksekusi pada library OpenCV.

3.7 Pengujian

Pada proses pengujian adalah dilakukan percobaan sistem pada implementasi yang telah dilakukan dengan perhitungan dengan mengubah gambar menjadi biner. Diharapkan dapat sesuai dengan hasil yang diharapkan. Tujuan dari pengujian ini adalah untuk menjamin bahwa hasil citra yang dilakukan sesuai dengan gambar yang dicari. Pengujian sistem sesuai dengan huruf yang dideteksi sesuai dengan keluar suara yang dibaca oleh sistem. (Tauchid, et al., 2015)

3.8 Kesimpulan

Kesimpulan yang akan didapatkan dari hasil pengujian pengenalan rambu petunjuk arah berbasis raspberry pi menggunakan metode Optical Character Recognition terhadap sistem yang dibuat. Pada tahapan terakhir ini bertujuan untuk hasil kesimpulan akhir dari sistem dan dapat dilakukan perbaikan pada kesalahan penelitian.

BAB 4 REKAYASA KEBUTUHAN

Pada rekayasa kebutuhan menjelaskan tentang gambar umum sistem yang akan dilakukan sebelum implementasi pada bab selanjutnya dan dilakukan analisis kebutuhan perangkat keras dan perangkat lunak sertabatasan desain dari sistem pengenalan rambu petunjuk arah menggunakan metode *Optical Character Recognition*.

4.1 Gambaran Umum Sistem

Sistem pengenalan rambu petunjuk arah digunakan dalam mendeteksi setiap rambu petunjuk arah lalu lintas yang khususnya yang memiliki warna *background* hijau. Metode dalam sistem menggunakan metode ORC dengan mendeteksi rambu petunjuk arah yang berwarna *background* hijau. Merubah gambar RGB menjadi HSV dengan melakukan masking warna hijau untuk memisahkan objek lain dengan rambu lalu lintas. Sistem menggunakan kamera untuk menangkap gambar yang akan diproses, dengan misahkan warna hijau dan mendeteksi citra yang berbentuk kotak. Kemudian sistem mengeluarkan hasil yang dideteksi dan akan mengubah citra tersebut ke teks dan merubah menjadi suara hasil dari pendeteksi citra tersebut. Pada gambar berikut merupakan diagram blok dari sisitem.



Gambar 4. 1 Diagram blok sistem

Perancangan diagram blok sistem dibagi menjadi 3 yaitu kamera, raspberry pi dan suara. Kamera digunakan untuk melakukan input gambar secara *real time* dari sistem dan akan diproses oleh raspberry pi. Hasil akhir dari sistem berupa suara dari gambar yang diproses pada sistem pengenalan rambu petunjuk arah.

4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk perancangan sistem pengenalan rambu petunjuk arah ini agar dipenuhi. Dalam melakukan analisis kebutuhan sistem dilakukan 2 kebutuhan, yaitu kebutuhan fungsional dan kebutuhan non-fungsional

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional dari sistem untuk memenuhi kebutuhan dalam penelitian ini yaitu :

1. Sistem mengambil gambar secara *real time*

Sistem dapat mengambil citra secara *real time* dengan pada saat perjalanan.

2. Sistem dapat memisahkan mendeteksi rambu petunjuk arah

Sistem harus bisa membedakan antara objek lain dan rambu petunjuk arah, banyaknya objek yang ditangkap oleh kamera diharuskan sistem membedakan citra yang harus dideteksi untuk hasil citra. Citra rambu lalu lintas dengan berwarna hijau dan kotak.

3. Sistem dapat mendeteksi huruf dan arah yang terdapat pada rambu petunjuk arah

Sistem harus dapat mendeteksi huruf yang terdapat pada rambu lalu lintas berupa arah dari rambu lalu lintas. Sistem mendeteksi dan merubah dalam bentuk teks.

4. Sistem dapat mengubah teks menjadi suara

Sistem bisa menerjemahkan huruf yang telah dideteksi kedalam suara.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional pada sistem akan dibagi menjadi 2 kebutuhan yaitu perangkat keras dan perangkat lunak.

4.2.2.1 Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan pada sistem pengenalan rambu petunjuk arah dijelaskan sebagai berikut:

1. Raspberry pi 3 Model B

Raspberry pi 3 merupakan modul *mikro* komputer yang mempunyai nilai input output. Raspberry pi merupakan komponen utama untuk melakukan pemrosesan citra.

2. Webcam Logitech c 170

Webcam Logitech c 170 digunakan sebagai perangkat keras akan mengambil gambar merupakan input pada pengenalan rambu petunjuk arah

3. Personal Computer

Personal computer berfungsi untuk pembuatan program yang akan digunakan oleh sistem.

4.2.2.2 Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang dibutuhkan dalam sistem pengenalan rambu petunjuk arah dijelaskan sebagai berikut :

1. Noobs V2

Noobs v2 digunakan sebagai sistem operasi yang digunakan pada perangkat raspberry pi.

2. Opencv

Opencv merupakan *library* python yang digunakan pada proses pengolahan gambar, fungsi tampilan, fungsi menyimpan gambar dan lain lain.

3. Python *command line*

Python command line digunakan untuk memproses sistem, dalam sistem menggunakan bahasan pemrograman python.

4. Pytesseract

Pytesseract merupakan *library* python yang digunakan dalam pengenalan huruf pada gambar.

5. Pytsx

Pytsx merupakan *library* python yang digunakan untuk merubah huruf ke dalam suara yang akan didengarkan menggunakan *earphone*.

4.3 Batasan Desain Sistem

Dalam pembuatan sistem pendeteksi rambu lalu lintas petunjuk arah terdapat beberapa batasan yang digunakan dalam sistem ini sehingga pembahasan, perancangan dan implementasi dari sistem tidak meluas :

1. Sistem mengambil rambu petunjuk arah yang berwarna hijau.
2. Sistem hanya menerjemahkan rambu petunjuk arah yang memiliki satu arah panah.
3. Sistem hanya mendeteksi huruf dan posisi arah panah

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini menjelaskan tentang perancangan sistem dan implementasi sistem. Perancangan sistem digunakan untuk perancangan diagram alir sistem dalam kebutuhan perangkat keras maupun perangkat lunak dan implementasi dari penelitian rancang bangun sistem pengenalan rambu petunjuk arah berbasis raspberry pi menggunakan metode *Optical Character Recognition*.

5.1 Perancangan Sistem

Pada pembahasan perancangan sistem akan menjelaskan diagram alir sistem dan proses yang dilakukan sistem.

5.1.1 Perancangan Perangkat Keras

Perangkat keras sistem yang dibutuhkan dalam pembuatan sistem, pada gambar dapat dilihat terdapat komponen utama pada sistem yang akan digunakan.

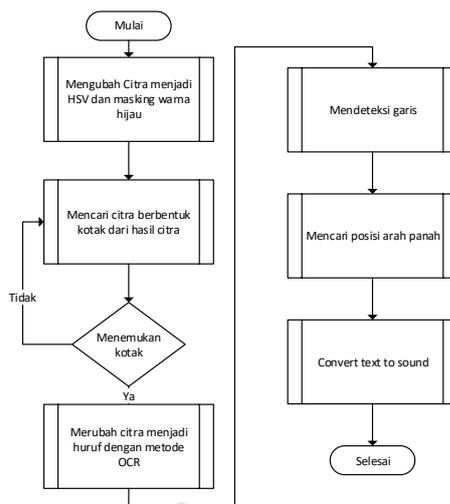


Gambar 5. 1 Perangkat keras dari sistem

Pada gambar 5.2 diatas menjelaskan mengenai pemasangan webcam Logitech c 170 pada raspberry pi menggunakan port USB yang dihubungkan pada port USB pada raspberry pi. Webcam Logitech c 170 berfungsi untuk menangkap gambar rambu petunjuk arah yang akan diproses pada raspberry pi dan akan memberi output berupa hasil teks dari rambu petunjuk arah.

5.1.2 Perancangan perangkat lunak

Dalam diagram alir ini menjelaskan tentang perancangan sistem, berikut ini alur kerja perangkat lunak dari sistem secara garis besar digambarkan pada gambar 5.3.

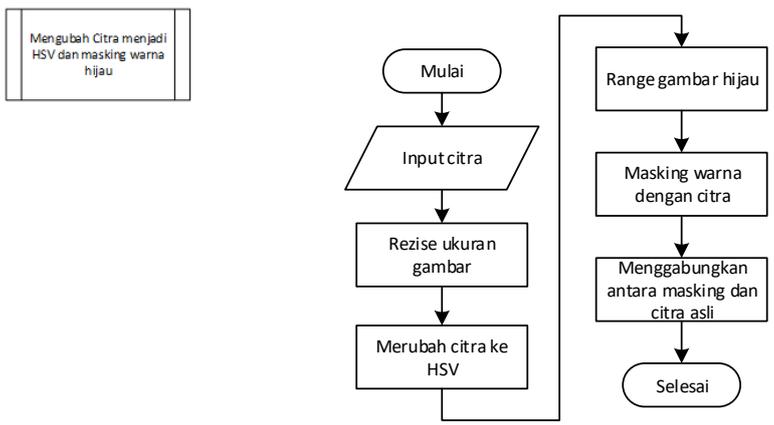


Gambar 5. 2 diagram alir sistem

Pada gambar 5.3 menjelaskan tentang keseluruhan dari perangkat lunak yang akan digunakan pada sistem pengenalan rambu petunjuk arah yang akan diklasifikasikan berdasarkan warna dan bentuk dan hasil akhir dari sistem adalah berupa suara.

5.1.2.1 Perancangan inialisasi citra pada HSV

Pada sub bab ini perancangan sistem dengan memisahkan warna lain selain hijau disebabkan karena warna dari rambu petunjuk arah yaitu hijau. Warna pada citra didefinisikan range warna hijau, range warna tersebut akan dilakukan masking pada citra yang asli sebelum dilakukannya konversi citra ke hsv. HSV mendefinisikan warna dalam terminologi Hue (Warna Sebenarnya), Saturation (Kemurnian Warna) dan Value (Kecerahan Warna) Keuntungan HSV adalah terdapat warna-warna yang sama dengan yang ditangkap oleh indra manusia, sedangkan warna yang dibentuk model lain seperti RGB merupakan hasil campuran dari warna-warna primer. Masking bertujuan untuk melakukan pengolahan citra penggabungan gambar yang telah diinisialisasi pada *range* warna yang ditetapkan pada *hue*, *saturation*, dan *value*. Masking akan menampilkan warna yang telah diberi *range* hijau saja, dijelaskan pada gambar 5.3



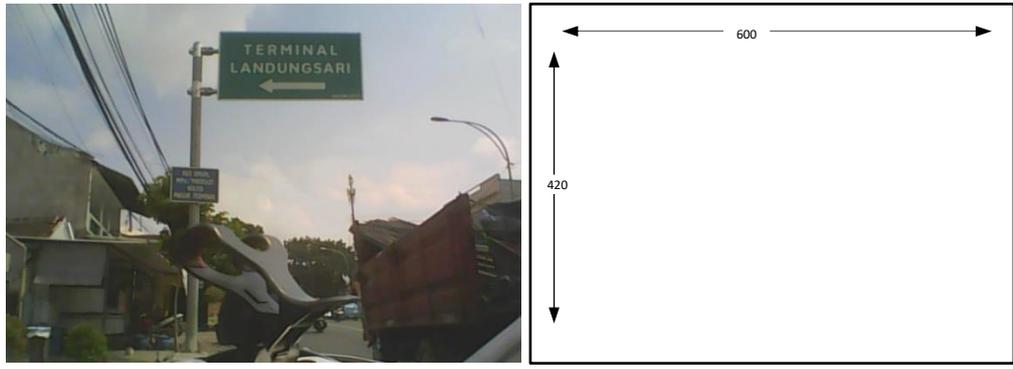
Gambar 5. 3 diagram alir perancangan inisialisasi citra

Pada sub bab ini menjelaskan tentang sistem melakukan pemilihan warna yang akan diselect untuk proses. Sistem ini akan memilih hanya warna hijau untuk dideteksi memisahkan antara objek lain. Pemilihan objek untuk membedakan dengan objek lain, rambu petunjuk arah berada diatas dari citra

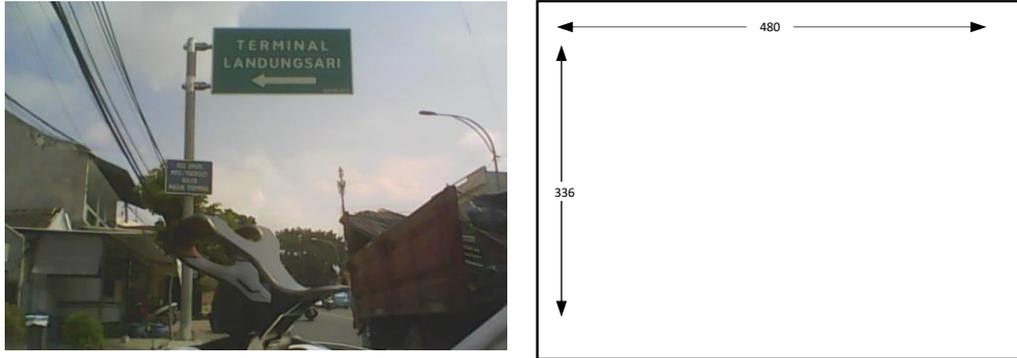


(a) (b)
Gambar 5. 4 Contoh pengambilan citra

Pada gambar rambu petunjuk lalu lintas berada diatas dari citra yang ditangkap pada perancangan sistem dilakukan *resize* pada citra untuk mengurangi citra yang tidak perlu di insialisasi citra. Citra yang diambil berukuran tinggi 420 pixel dan lebar 600 pixel, akan dilakukan *resize* sebesar 80% dari citra untuk meningkatkan proses pengenalan citra, maka ukuran yang telah *resize* berukuran tinggi 480 pixel dan lebar 336 pixel.



Gambar 5. 5 citra gambar pada ukuran asli



Gambar 5. 6 citra gambar setelah proses *resize*

Setelah gambar *direzise* gambar dilakukan *masking* warna hijau dengan penentuan *range* warna hijau yang akan dideteksi, gambar yang telah dideteksi. Gambar yang telah dimasking hanya akan memunculkan warna hijau saja yang ditampilkan pada sistem dapat dilihat pada gambar 5.7.

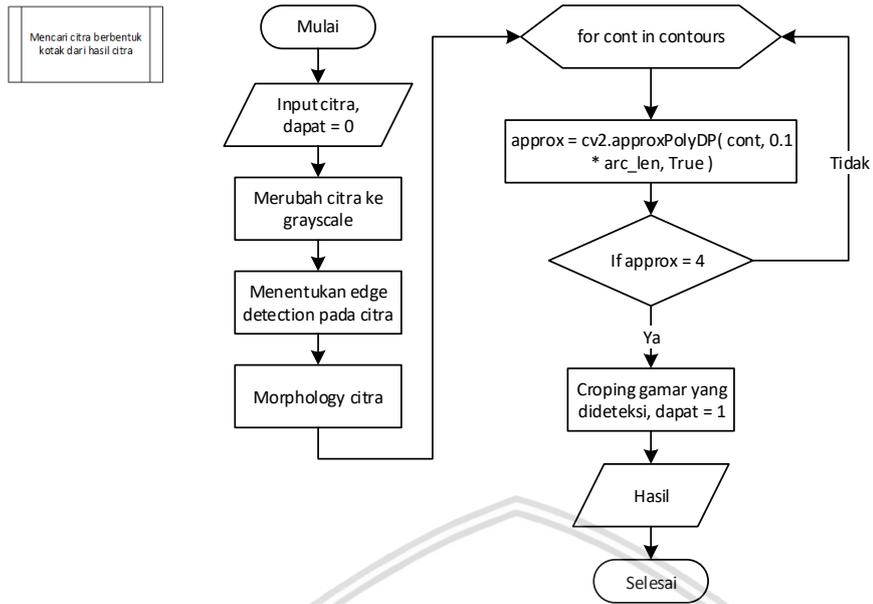


Gambar 5. 7 gambar yang telah dilakukan *masking*

5.1.2.2 Perancangan pencarian objek

Pada perancangan pencarian objek merupakan pencarian dari citra, pada rambu lalu lintas petunjuk arah yang sudah ditangkap akan dilakukan inisialisasi objek yang berbentuk kotak. Sistem akan mengambil objek yang berbentuk kotak, perancangan dapat digambarkan pada diagram alir berikut ini.





Gambar 5. 8 diagram alir pencarian objek

Pada gambar 5.9 menjelaskan tentang bagaimana sistem mencari kotak pada gambar yang diproses. Warna yang telah dilakukan *masking* warna hijau sebelumnya dilakukan perubahan warna ke *grayscale* agar mempermudah dalam pencarian bentuk kotak. Selanjutnya dilakukan *edge detection* untuk melakukan deteksi garis pada gambar. Setelah dilakukan *morphology* pada gambar dilakukan perulangan pada gambar untuk mencari kotak dengan parameter memiliki 4 sudut setelah dilakukan garis pada *edges detection*

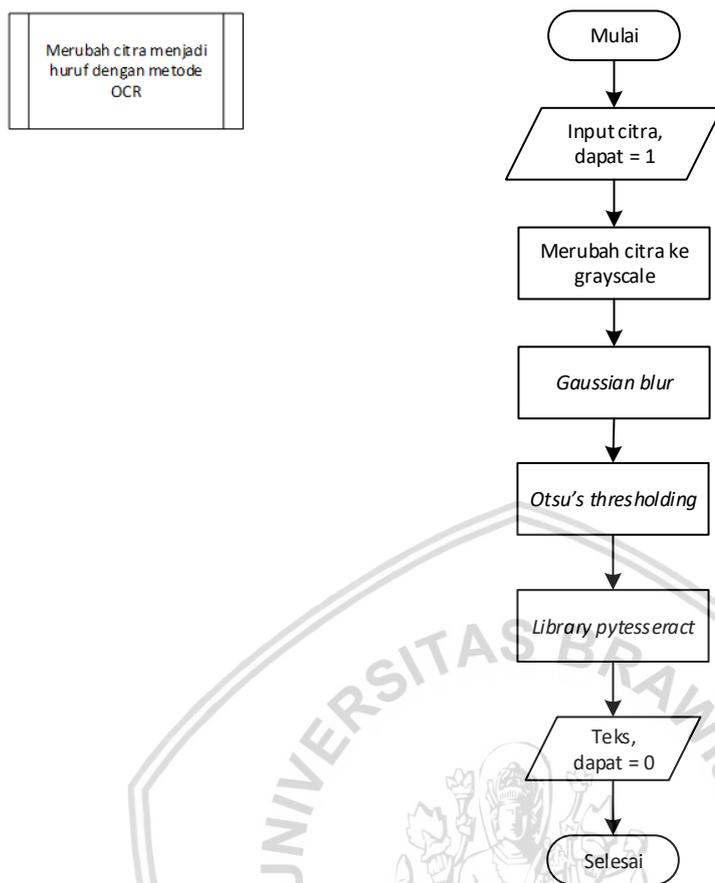
5.1.2.3 Perancangan deteksi karakter huruf

Pada perancangan ini setelah mendapatkan citra dari rambu petunjuk arah, selanjutnya sistem melakukan dua proses pengolahan citra yaitu karakter huruf dan arah panah. Pada perancangan ini melakukan *grayscale* agar objek menjadi hitam putih, selanjutnya proses *Gaussian blur* adalah memperbaiki citra dengan membuang *noise* pada citra. *Otsu's Threshold* bertujuan untuk menentukan titik tepi pada citra dengan menghasilkan tepi dari gambar, selanjutnya menentukan minimal nilai ambang dari citra, *otsu's thresholding* dapat dituliskan persamaan berikut :

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{5. 1}$$

OCR melakukan gambar ke teks. Proses tersebut dapat dijabarkan pada diagram alir berikut.

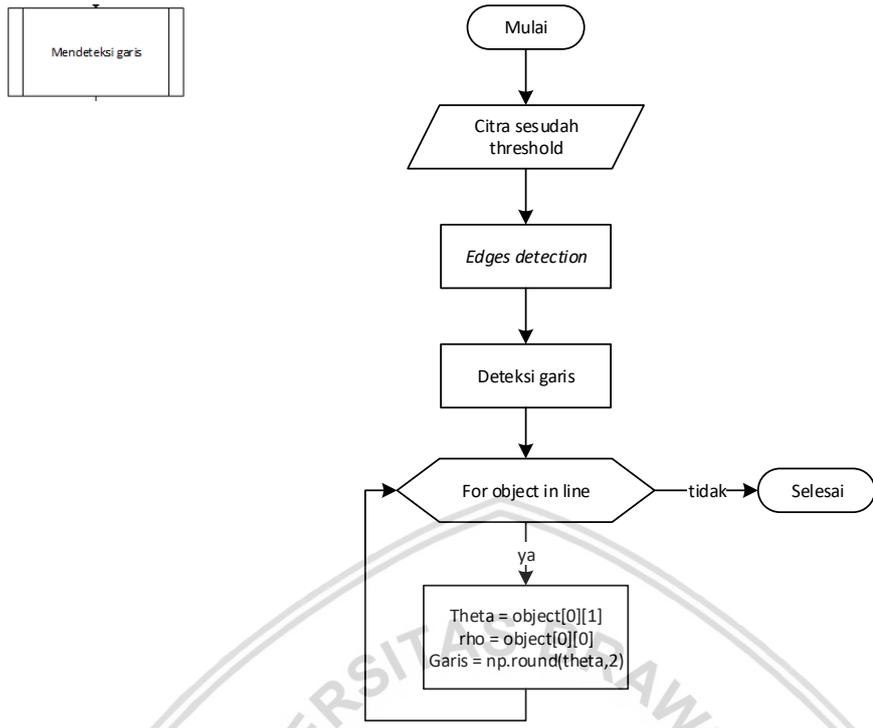




Gambar 5. 9 diagram alir perancangan pendeteksi karakter huruf dan panah

5.1.2.4 Perancangan deteksi garis lurus

Pada perancangan deteksi arah panah menggunakan citra yang sudah dideteksi pada subbab sebelumnya, perancangan arah panah menggunakan *edges detection* untuk mendeteksi tepi pada gambar arah panah, selanjutnya akan dilakukan deteksi garis menggunakan *Houghlines*. Menentukan arah dari sudut yang dideteksi oleh houghline, proses analisa arah panah dapat dijelaskan pada diagram alir pada gambar 5.11.

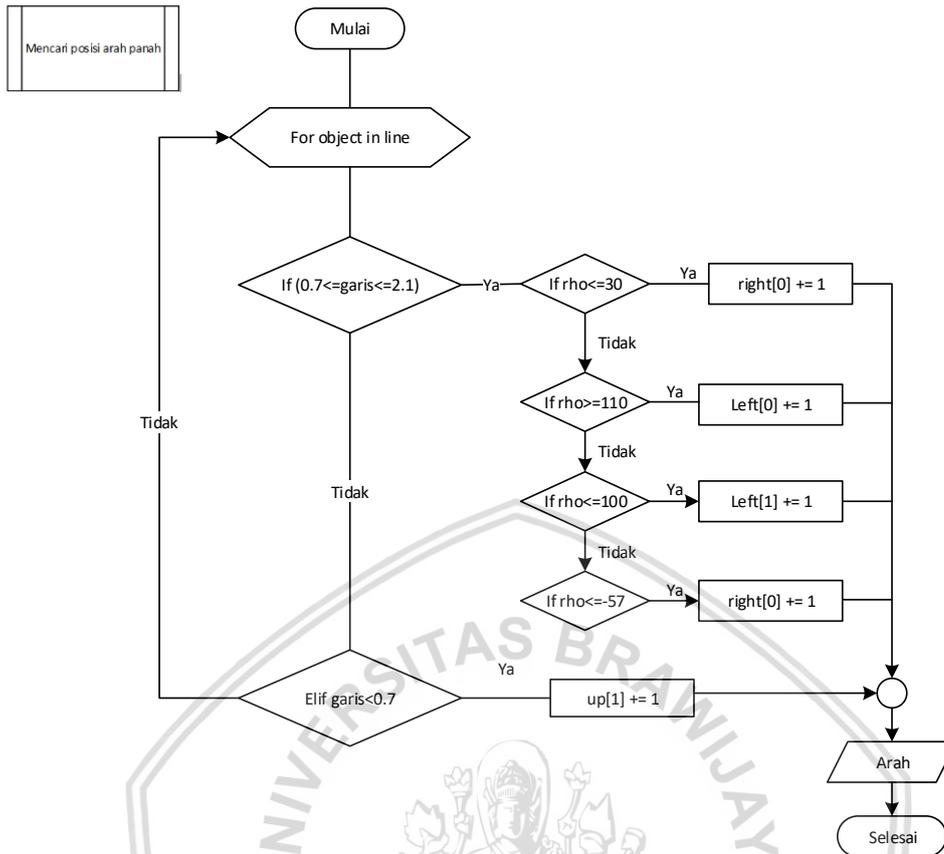


Gambar 5. 10 diagram alir mendeteksi garis pada rambu petunjuk arah

5.1.2.5 Perancangan deteksi posisi arah panah

Selanjutnya setelah dilakukan transformasi hough mendeteksi garis bekerja dengan memproyeksikan objek dari koordinat x dan y ke koordinat lingkaran. Pada analisis posisi arah panah menganalisa sudut dari pada arah panah, dilakukan *increment* pada nilai array.jika kedua nilai array. Pada analisis berikut dijelaskan diagram alir pada gambar 5.12.

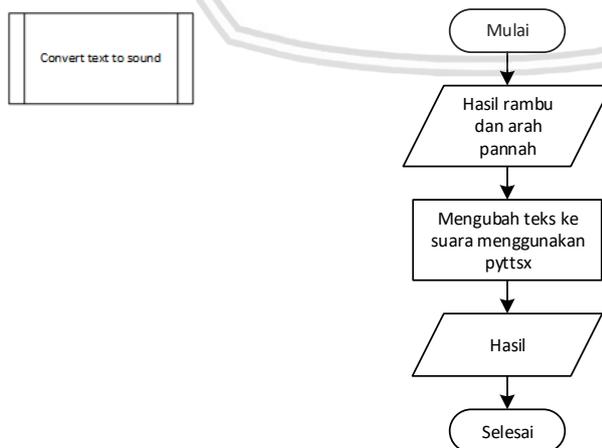




Gambar 5. 11 diagram alir menentukan posisi dari arah panah

5.1.2.6 Perancangan *text to sound*

Tahap terakhir dari sistem adalah merubah huruf yang sudah dideteksi pada proses pengenalan huruf dan arah panah yang telah dikenali menjadi suara, yang hasilnya akan dikeluarkan dari sistem. Sistem menggunakan *library* pada python yaitu *pyttsx*, yang akan merubah hasil dari teks tersebut menjadi suara.



Gambar 5. 12 Perancangan *text to sound*

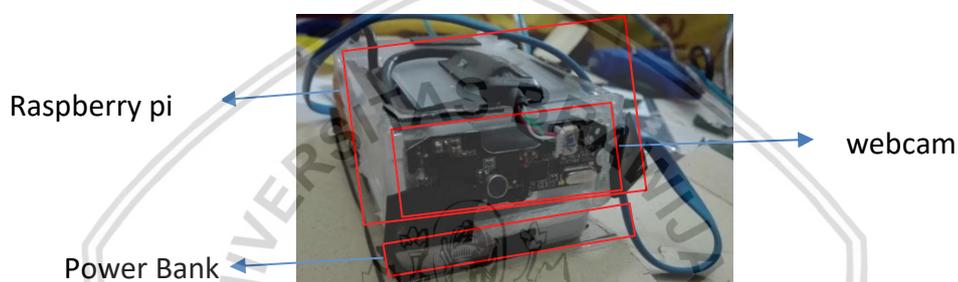


5.2 Implementasi

Implementasi menjelaskan tentang hasil perencanaan yang dilakukan. Pada bagian terdapat beberapa yang akan dibahas dalam bab ini yaitu implementasi perangkat keras dan perangkat lunak pada sistem pengenalan rambu petunjuk arah.

5.2.1 Implementasi perangkat keras

Pada implementasi perangkat keras merupakan hasil dari perencanaan pada perangkat keras yang telah dirancang. Perangkat yang digunakan yaitu webcam logitech c170 dihubungkan ke port usb pada raspberry pi 3 Model B. power yang digunakan untuk perangkat yaitu menggunakan power bank. Hasil dari perangkat dapat dilihat pada gambar 5.14.



Gambar 5. 13 Hasil dari perancangan perangkat keras

Pada gambar 5.14 adalah bentuk implementasi perangkat keras pengenalan rambu petunjuk arah. Perangkat keras yang digunakan dalam sistem yaitu webcam logitech c170, raspberry pi, power bank sebagai daya untuk sistem dan beberapa perangkat keras yang dibutuhkan pada sistem. hasil akhir dari implementasi perangkat keras pada pengenalan rambu petunjuk arah dapat dilihat pada gambar 5.15.



Gambar 5. 14 Hasil akhir dari perangkat keras

Pada gambar 5.15 dapat dilihat hasil akhir dari perangkat keras sudah dapat digunakan pengenalan rambu petunjuk arah yang telah menggunakan kotak dan ditambahkan beberapa perangkat keras berupa *mount action camera* yang digunakan untuk pengait pada kendaraan bermotor dan *earphone* yang berfungsi sebagai output peringatan suara kepada pengendara.

5.2.2 Implementasi perangkat lunak

Implementasi perangkat lunak adalah hasil implementasi dari perancangan sistem yang dijelaskan pada perancangan. Menjelaskan tentang perangkat lunak yaitu berupa *source code* yang digunakan dalam sistem ini. sistem ini menggunakan *library* pada python yang digunakan untuk melakukan proses dalam melakukan pengenalan rambu lalu lintas yaitu pada gambar 5.15.

1	import cv2
2	import time
3	import numpy as np
4	import pytesseract
5	import pyttsx

Gambar 5. 15 Program inialisasi *library*

5.2.2.1 Implementasi pengenalan citra

Sub bab ini menjelaskan tentang implementasi pengenalan citra yang akan melakukan pemisahan warna yang ditangkap oleh kamera. Program tersebut terdapat pada Gambar 5.16.

1	ret, cap = video.read()
2	start_time = time.time()
3	ukuranbaru = cv2.resize(cap, size, interpolation = cv2.INTER_AREA)
4	#ubah ke grayscale
5	ubah = cv2.cvtColor(ukuranbaru, cv2.COLOR_BGR2HSV)
6	
7	#range warna hijau
8	lower_green = np.array([35,50,50])
9	upper_green = np.array([102,255,255])
10	
11	#masking warna hijau
12	mask = cv2.inRange(ubah, lower_green, upper_green)
13	res = cv2.bitwise_and(ukuranbaru, ukuranbaru, mask= mask)

Gambar 5. 16 Program inialisasi warna hijau

Pada *source code* awalnya membaca video dengan dibaca secara langsung, gambar diberi ukuran baru dengan 600x420 agar mempermudah dalam proses *resize* gambar, selanjutnya gambar akan dikoversi ke HSV pada *library opencv* dan melakukan *cropping* gambar yang telah dijelaskan pada sub bab perancangan sistem. Menggunakan *library opencv* untuk melakukan pemilihan warna yang akan diproses. pengenalan citra akan melakukan metode HSV untuk dapat memisahkan warna yang akan dipilih, dengan mendefinisikan *range* warna hijau yang akan diambil oleh sistem, menggunakan numpy dengan membuat *range* dari warna hijau.

Setelah melakukan range pada gambar maka dilakukan masking untuk mengidentifikasi warna yang terdapat pada citra, masking tersebut akan memberntuk pola pada warna yang telah di inialisasi, hasil dari masking tersebut menggabungkan tepi luar hasil masking dengan citra asli menggunakan operasi *bitwise_and*. Hasil gambar akan membentuk warna hijau yang telah dibuat *range* dan citra asli yang telah di *mask*.

5.2.2.2 Implementasi pencarian objek

Tahap selanjutnya mencari objek berbentuk kotak, sistem akan melakukan perubahan warna agar bisa diidentifikasi, akhir dari proses adalah menemukan citra yang berbentuk kotak. Program tersebut dapat dilihat pada gambar 5.17.

```

1  gray = cv2.cvtColor( res, cv2.COLOR_BGR2GRAY)
2  gray = cv2.bilateralFilter( gray, 1, 10, 120 )
3  edges = cv2.Canny( gray, 100, 255 )
4  kernel = cv2.getStructuringElement( cv2.MORPH_RECT, ( 7, 7 ) )
5  closed = cv2.morphologyEx( edges, cv2.MORPH_CLOSE, kernel )
6  contours, h = cv2.findContours( closed, cv2.RETR_EXTERNAL,
   cv2.CHAIN_APPROX_SIMPLE )
7  for cont in contours:
8      if cv2.contourArea( cont ) > 1000 :
9          arc_len = cv2.arcLength( cont, True )
10         approx = cv2.approxPolyDP( cont, 0.1 * arc_len, True )
11         if ( len( approx ) == 4 ):
12             dapat = 1
13             pts_src = np.array( approx, np.float32 )
14             h, status = cv2.findHomography( pts_src, pts_dst )
15             out = cv2.warpPerspective( ukuranbaru, h, ( int( width + margin * 2 ), int(
   height + margin * 2 ) ) )
16             cv2.drawContours( ukuranbaru, [approx], -1, ( 255, 0, 0 ), 2 )
17
18         else : pass

```

Gambar 5. 17 Program pencarian objek berbentuk kotak

Gambar dengan menggunakan *grayscale* mengubah objek yang sudah difilter menggunakan HSV, *grayscale* berfungsi mempermudah menemukan bentuk kotak pada image dengan merubah gambar dengan menentukan tepi pada objek yang sudah dideteksi. Pada gambar tersebut yang sudah menghasilkan tepi-tepi objek dari gambar tersebut, dengan memperbaiki citra yang kabur untuk memudahkan dalam proses pengenalan objek, selanjutnya menentukan objek yang berbentuk kotak dengan menggunakan *countours* menghitung jumlah sudut gambar berjumlah 4. Gambar yang sudah ditangkap akan dilakukan perhitungan sudut dengan menggunakan tepi dari hasil gambar yang di tangkap.

5.2.2.3 Implementasi deteksi karakter huruf

Pada tahap implementasi karakter huruf, gambar yang telah dideteksi terlebih dahulu dilakukan normalisasi dengan segmentasi pada gambar agar citra dapat dideteksi, dilakukan normalisasi agar citra yang memiliki cahaya yang berlebih akan susah untuk dideteksi dan segmentasi guna untuk menghilangkan *noise* pada gambar, selanjutnya akan diproses dengan *library* OCR untuk mendeteksi huruf. Dapat dilihat pada gambar 5.18.

```

1  gmbr= cv2.resize(out, size, interpolation = cv2.INTER_AREA)
2  gmbr= cv2.cvtColor(gmbr, cv2.COLOR_BGR2GRAY)
3  blur = cv2.GaussianBlur(gmbr,(5,5),0)
4  # mencari normalized_histogram
5  hist = cv2.calcHist([blur],[0],None,[256],[0,256])

```

```

6 hist_norm = hist.ravel()/hist.max()
7 Q = hist_norm. cumsum()
8 bins = np.arange(256)
9 fn_min = np.inf
10 thresh = -1
11 for i in xrange(1,256):
12     p1,p2 = np.hsplit(hist_norm,[i])
13     q1,q2 = Q[i],Q[255]-Q[i]
14     b1,b2 = np.hsplit(bins,[i])
15     # mencari means and variances
16     m1,m2 = np.sum(p1*b1)/q1, np.sum(p2*b2)/q2
17     v1,v2=np.sum(((b1-m1)**2)*p1)/q1,np.sum(((b2-m2)**2)*p2)/q2
18
19     # menghitung minimal
20     fn = v1*q1 + v2*q2
21     if fn < fn_min:
22         fn_min = fn
23         thresh = i
24
25     # mencari nilai dari otsu thresholding
26     ret,otsu=cv2.threshold(blur,0,255,cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
27     kata = pytesseract.image_to_string(th2)

```

Gambar 5. 18 Program image to teks

Pada baris 2-6 hasil dari deteksi pada gambar akan dilakukan perbaikan citra menggunakan *otsu's threshold* untuk menemukan tepi pada tiap huruf untuk memberi tepi pada gambar, terlebih dahulu dilakukan *resize* karena citra kembali ke ukuran citra yang sebenarnya. Pada tahapan selanjutnya dilakukan *grayscale* untuk menghasilkan citra hitam putih yang terdiri dari 8 bit. Selanjutnya dilakukan *median filtering* berfungsi meningkatkan kualitas citra. Tahapan selanjutnya pada baris 13 - 25 dalam proses ini dilakukan *otsu's threshold* untuk mengklasifikasikan nilai-nilai pixel, selanjutnya dilakukan nilai *threshold* pada citra lalu tahapan terakhir mengkonversi gambar menjadi biner mencari nilai minimum pada citra.

Setelah gambar telah dilakukan segmentasi selanjutnya menggunakan *library OCR python* yaitu *pytesseract* untuk mendeteksi huruf. *Pytesseract* merupakan akan memanggil *Tesseract* yang dieksekusi sebagai *script* eksternal.

5.2.2.4 Perancangan deteksi posisi arah panah

Citra yang telah dideteksi sebelumnya terdapat 2 bagian yaitu arah panah dan huruf, pada OCR gambar hanya menerjemahkan huruf saja, sehingga gambar arah panah tidak dapat diterjemahkan. Pada pendeteksi posisi arah panah ini akan mengeluarkan hasil dari posisi arah panah kanan, kiri, depan, dan bawah. Sistem akan mendeteksi garis lurus yang terdapat pada gambar, menggunakan *Hough Line* mendeteksi garis lurus pada gambar arah panah. Arah panah memiliki dominan garis lurus daripada citra pada huruf sehingga citra huruf pada gambar akan dapat dibedakan oleh sistem.

```

1 edges = cv2.Canny(th2,10,150,apertureSize = 3)
2 lines = cv2.HoughLines(edges,1,np.pi/180,20)

```

3	for object in lines:
4	theta = object[0][1]
5	rho = object[0][0]
6	garis = np.round(theta,2)
7	if (0.7<= garis <= 2.1):
8	if (rho <= 30):
9	left[0] += 1
10	elif (rho <= 100):
11	left[1] +=1
12	elif (rho <= -57):
13	right[0] +=1
14	elif (rho >= 110):
15	right[1] +=1
16	
17	elif (garis < 0.7):
18	up[1] +=1
19	if left[0] >= 1 or left[1] >= 1:
20	arah = "belok kiri"
21	elif right[0] >= 1 or right[1] >= 1:
22	arah = "belok kanan"
23	elif up[1] >= 1:
24	print arah

Gambar 5. 19 Program penentuan posisi dari arah panah

Dapat dilihat pada baris 1 citra akan dilakukan deteksi tepi pada gambar dan dilakukan konversi citra sehingga citra berbentuk garis tepi, mendeteksi garis tepi dari citra, dari tepi citra tersebut akan dilakukannya deteksi garis pad citra. Citra yang dideteksi akan menggunakan garis yang dideteksi lebih banyak sehingga membedakan citra hasil dari citra huruf yang dilakukan *edges detection*. Pada tahapan selanjutnya pada baris 2 menentukan arah panah dengan menggunakan *houghline* mendeteksi garis lurus pada gambar. Gambar diwakilkan oleh 2 komponen, yakni jari-jari (rho) dan sudut (theta).

Pada proses pendeteksi posisi arah panah menentukan nilai dari rho dan theta untuk mengetahui posisi pada citra yang telah dideteksi.

5.2.2.5 Perancangan *text to sound*

1	engine.say(kata)
2	engine.say(arah)
3	engine.runAndWait()
4	time.sleep(DELAY)

Gambar 5. 20 Program *text to soud*

Pada tahapan terakhir dari sistem mengubah kata yang dideteksi pada rambu petunjuk arah dan posisi dari gambar arah panah menggunakan *library* python pyttsx. Library ini digunakan untuk *text-to-sound* yang akan mengubah huruf yang dideteksi menjadi suara dan teks posisi arah panah



BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang pengujian sistem serta melakukan analisis yang dilakukan sistem yang telah di bahas dalam implementasi pada bab sebelumnya. Pengujian dan analisis ini bertujuan untuk menguji sistem secara fungsional dan non fungsional dilakukan pada perangkat keras dan perangkat lunak.

6.1 Pengujian dan analisis fungsional

Pada pengujian dan analisis fungsional membahas mengenai pembacaan arah panah dan akurasi gambar yang ditangkap.

6.1.1 Pengujian kamera webcam Logitech c 170

Pada pengujian ini melakukan pengujian pada kamera dan raspberry. Raspberry pi digunakan sebagai operasi sistem utama yang memproses gambar yang akan ditangkap oleh kamera. Pengujian pada webcam logitech dapat mengambil citra berupa gambar secara *realtime* terus menerus.

6.1.1.1 Tujuan Pengujian

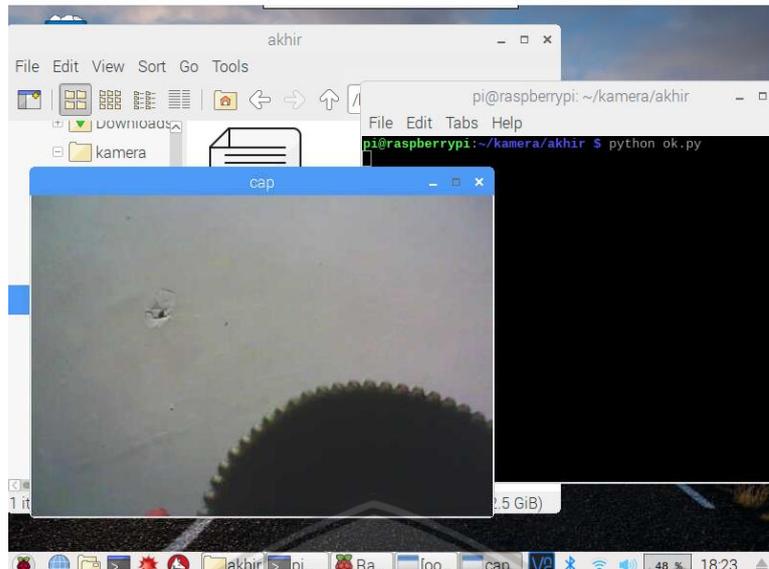
Tujuan pengujian ini adalah untuk mengetahui modul kamera dapat dilakukan pengambilan gambar oleh raspberry pi dan kamera Logitech yang telah terhubung menggunakan USB secara *real time*.

6.1.1.2 Prosedur pengujian

1. Menghubungkan webcam Logitech c 170 pada raspberry pi 3 menggunakan USB.
2. Melakukan perintah pengambilan gambar.
3. Gambar ditangkap secara realtime

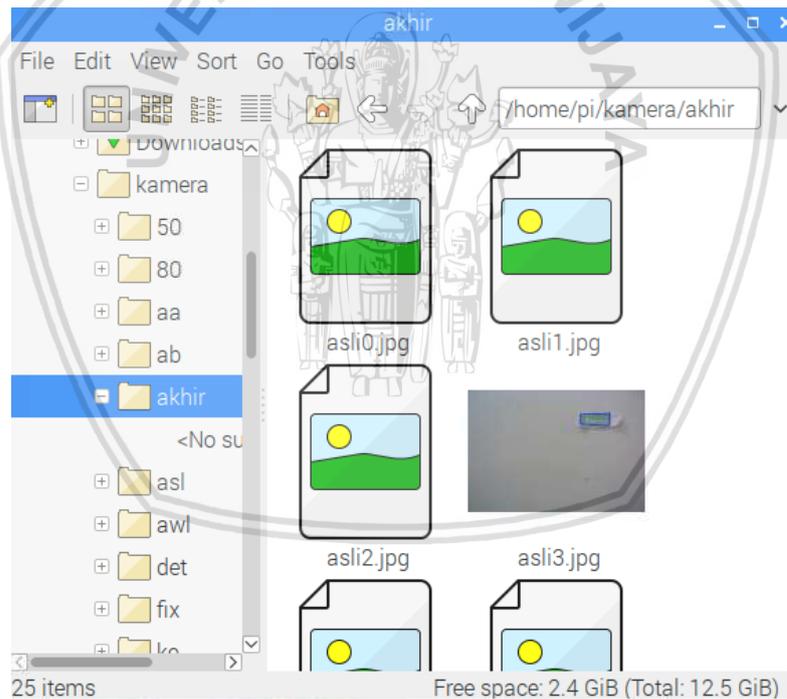
6.1.1.3 Hasil dan Analisa pengujian

Hasil dan analisi dari pengujian webcam logitech c 170 akan mengaktifkan *interface* kamera pada raspberry pi, sehingga kamera dapat digunakan pada raspberry pi 3. Melakukan perintah untuk mengaktifkan kamera dan menampilkan hasil dari tangkapan kamera dapat dilihat pada gambar



Gambar 6. 1 pengujian kamera webcam Logitech c 170

Setelah kamera telah diaktifkan, pengujian selanjutnya melakukan pengujian melakukan input dari kamera.



Gambar 6. 2 pengujian pengambilan gambar

Pengujian yang telah dilakukan, dapat dilihat bahwa webcam Logitech c 170 telah dapat terhubung dengan raspberry pi sehingga kamera dapat dilakukan secara *real time* dan input gambar.

6.1.2 Pengujian deteksi rambu petunjuk arah

Pada pengujian ini pengambilan citra diambil akan diproses untuk memisahkan antara gambar yang tidak perlu dideteksi, gambar yang dideteksi



kotak berbentuk hijau pada citra yang diambil. Pengujian ini dilakukan dengan cara menangkap gambar jalan secara *realtime*.

6.1.2.1 Tujuan Pengujian

Pada pengujian akurasi deteksi ini bertujuan untuk mengetahui akurasi deteksi rambu petunjuk arah yang digunakan oleh sistem dan keberhasilan dari sistem pada gambar. Sistem diharapkan dapat mendeteksi rambu petunjuk arah yang berwarna hijau dan mampu membedakan dengan objek lainnya.

6.1.2.2 Prosedur pengujian

1. Memberikan sistem citra jalan raya secara realtime
2. Sistem menerapkan program untuk mendeteksi kotak yang berwarna hijau

6.1.2.3 Hasil dan Analisa pengujian

No	Gambar	Hasil deteksi
1.		
2.		
3.		
4.		

5.		
6.		
7.		
8.		
9.		
10.		

Tabel 6. 1 Hasil pengujian deteksi rambu petunjuk arah

Pada tabel 6.1 dapat dilihat hasil deteksi rambu penunjuk arah dengan mengubah warna hijau terlebih dahulu setelah itu terdapat rambu petunjuk arah. Setelah dideteksi berbentuk kotak maka hasil gambar yang telah dideteksi maka akan dilakukan *crop* pada gambar yang dideteksi.

6.1.3 Pengujian Akurasi Pengenalan Karakter

Pengujian ini dilakukan untuk mendeteksi karakter tiap huruf yang terdapat pada rambu petunjuk arah.

6.1.3.1 Tujuan Pengujian

Tujuan pengujian ini adalah untuk mengetahui pengenalan karakter pada sistem untuk memenuhi kebutuhan fungsional sistem. Pada rambu petunjuk tersebut diharapkan hasil mendeteksi huruf sesuai dengan yang terdapat pada rambu petunjuk arah

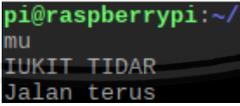
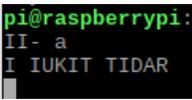
6.1.3.2 Prosedur Pengujian

1. Memberikan sistem berupa gambar kotak yang telah dideteksi.
2. Sistem akan melakukan perbaikan citra dan *thresholding* untuk memperbaiki citra sebelum dilakukan pengenalan huruf.
3. Melakukan deteksi huruf dengan tesseract pada *python library*. Sistem akan mengeluarkan hasil dari huruf yang dideteksi.

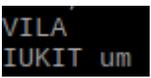
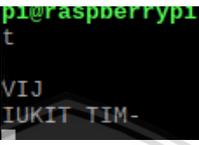
6.1.3.3 Hasil dan Analisis Pengujian

Hasil pengujian sistem yang telah mendeteksi karakter pada citra pada table berikut :

Tabel 6. 2 Hasil pengujian pengenalan karakter huruf

No	Citra	Hasil	Keberhasilan	Persentasi
1			Total huruf = 19 Terdeteksi = 19	100%
2			Total huruf = 12 Terdeteksi = 9	75%
3			Total huruf = 19 Terdeteksi = 7	36%
4			Total huruf = 12 Terdeteksi = 9	75%



5			Total huruf = 6 Terdeteksi = 1	16%
6			Total huruf = 12 Terdeteksi = 8	66%
7			Total huruf = 12 Terdeteksi = 8	66%
8			Total huruf = 18 Terdeteksi = 9	50%
9			Terdeteksi = 3	0.13%
10			Terdeteksi = 5	0.21%
Rata rata				48.4%

Pada table 6.2 terdapat hasil tertinggi dari pengujian karakter huruf adalah 100% dengan gambar yang memiliki warna dan cahaya yang maksimal, terdapat 0.13% pendeteksian karakter huruf pada rambu petunjuk arah karena terdapat banyak noise dan mempunyai tulisan yang kurang mampu dideteksi oleh kamera karena keterbatasan besaran pixel. hasil rata rata dari sistem 48.4%, terdapat banyak *noise* pada citra karena keterbatasan dari kamera sehingga gambar yang dideteksi banyak terdapat *noise*.

6.1.4 Pengujian akurasi pengenalan posisi arah panah

Pengujian ini dilakukan untuk mendeteksi karakter tiap huruf yang terdapat pada rambu petunjuk arah.

6.1.4.1 Tujuan Pengujian

Tujuan pengujian ini adalah untuk mengetahui pengenalan arah panah pada rambu petunjuk arah guna kebutuhan fungsional sistem. Pada rambu petunjuk tersebut diharapkan hasil mendeteksi arah panah sesuai dengan yang terdapat pada rambu petunjuk arah



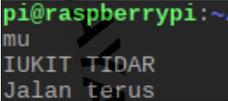
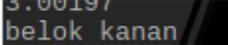
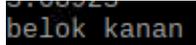
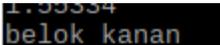
6.1.4.2 Prosedur Pengujian

1. Memberikan sistem berupa gambar kotak yang telah dideteksi.
2. Sistem akan melakukan perbaikan citra dan *otsu's thresholding* untuk memperbaiki citra sebelum dilakukan pengenalan huruf.
3. Melakukan deteksi garis menggunakan hough line dengan mendeteksi garis pada arah panah.

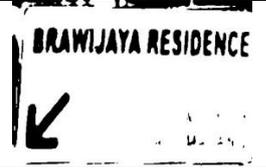
6.1.4.3 Hasil dan Analisis Pengujian

Hasil pengujian sistem yang telah mendeteksi posisi arah panah pada citra pada table berikut :

Tabel 6. 3 Hasil pengujian pengenalan posisi arah panah

No	Citra	Hasil
1		
2		
3		
4		
5		
6		



7		17.0 0.0 Jalan terus
8		1.5708 belok kiri 7 8245088
9		ML" Ant-um M l-T'auru Jalan terus
10		3.12414 belok kanan
Total		70%

Dapat dilihat dari table diatas keberhasilan dari mendeteksi posisi arah panah sebanyak 70% dari jumlah total data yang diuji. Terdapat banyak *noise* dalam citra yang dideteksi sehingga banyak garis yang dideteksi oleh *Hough line* terjadi kesalahan pada proses pendeteksian garis.

6.1.5 Pengujian suara

Pengujian ini dilakukan untuk merubah hasil dari teks ke suara sehingga dapat didengar oleh pengguna.

6.1.5.1 Tujuan Pengujian

Tujuan pengujian untuk mengetahui kinerja sistem untuk melakukan perubahan teks ke suara, sistem ini diharapkan dapat memberi informasi berupa suara kepada sistem

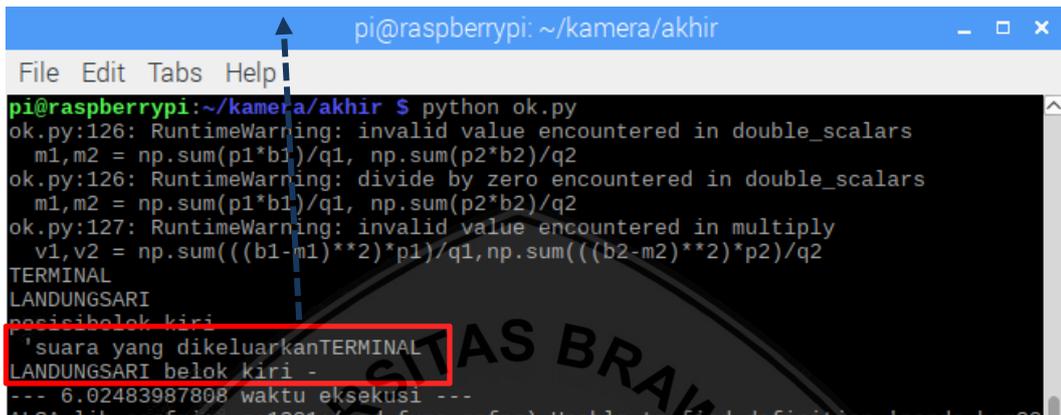
6.1.5.2 Prosedur Pengujian

1. Memberikan sistem hasil teks pada rambu dan posisi arah panah yang telah dideteksi.
2. Sistem akan melakukan konversi teks ke suara
3. Mengamati hasil yang telah dilakukan konversi teks ke suara

6.1.5.3 Hasil dan Analisis Pengujian

Hasil pengujian sistem yang telah melakukan konversi teks ke suara pada gambar :

Output dari sistem berupa suara yang dikeluarkan sesuai dengan huruf yang dideteksi OCR



```
pi@raspberrypi: ~/kamera/akhir
File Edit Tabs Help
pi@raspberrypi:~/kamera/akhir $ python ok.py
ok.py:126: RuntimeWarning: invalid value encountered in double_scalars
  m1,m2 = np.sum(p1*b1)/q1, np.sum(p2*b2)/q2
ok.py:126: RuntimeWarning: divide by zero encountered in double_scalars
  m1,m2 = np.sum(p1*b1)/q1, np.sum(p2*b2)/q2
ok.py:127: RuntimeWarning: invalid value encountered in multiply
  v1,v2 = np.sum(((b1-m1)**2)*p1)/q1,np.sum(((b2-m2)**2)*p2)/q2
TERMINAL
LANDUNGSARI
posisibelok kiri
'suara yang dikeluarkanTERMINAL
LANDUNGSARI belok kiri -
--- 6.02483987808 waktu eksekusi ---
```

Gambar 6. 3 Hasil pengujian koversi teks ke suara

Dapat dilihat dari gambar bahwa sistem dapat melakukan konversi dari teks ke suara sehingga dapat didengar oleh pengguna dan akhir dari sistem yang digunakan. Hasil dari output dari suara yaitu dikeluarkan melalui *earphone*.

6.2 Pengujian dan Analisis non fungsional

Pengujian dan analisis non fungsional ini dilakukan pengujian pada sistem, dalam subab ini menjelaskan tentang analisis dari waktu eksekusi sistem.

6.2.1 Pengujian dan Analisis waktu eksekusi

Pada pengujian dan analisis waktu yang dibutuhkan untuk eksekusi gambar setiap kali mendeteksi sampai *ouput* berupa huruf. Pada subab ini menghitung jumlah waktuyang dibutuhkan.

6.2.1.1 Tujuan Pengujian

Pada pengujian ini bertujuan mengetahui waktu untuk mendeteksi rambu lalu lintas hingga akhir.

6.2.1.2 Prosedur pengujian

Berikut ini adalah prosedur pengujian dalam pengujian waktu eksekusi sistem:

1. Menyalakan sistem.
2. Melakukan deteksi rambu petunjuk rambu lalu lintas.

3. Menghitung waktu eksekusi yang diperlukan sampai output berupa suara menggunakan *library* python.

6.2.2 Hasil dan Analisis Pengujian

Hasil pengujian sistem :

Tabel 6. 4 Hasil pengujian waktu eksekusi sistem

No	Citra	Waktu (detik)
1	Percobaan 1	6.70
2	Percobaan 2	6.20
3	Percobaan 3	8.02
4	Percobaan 4	5.82
5	Percobaan 5	5.41
6	Percobaan 6	4.70
7	Percobaan 7	5.92
8	Percobaan 8	6.95
9	Percobaan 9	7.50
10	Percobaan 10	6.80
	Maksimum	8.02
	Minimum	4.70
	Rata – rata	6.402

Pada Tabel 6.4 dapat dilihat hasil dari pengujian waktu eksekusi sistem waktu yang dibutuhkan dalam mengeksekusi satu gambar sampai hasil akhir berupa suara. Waktu minimum dalam mengeksekusi gambar 4.7 detik dalam mengeksekusi gambar, maksimum 8.02 detik dan rata rata 6.402 detik dalam mengeksekusi gambar.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil penelitian pengenalan rambu petunjuk arah menggunakan metode *Optical Character Recognition* dapat disimpulkan sebagai berikut :

1. Menggunakan metode *Optical character Recognition* dapat diterapkan pada pengenalan huruf pada rambu petunjuk arah, terdapat banyak *library* pada python untuk mengenali karakter huruf pada citra yang dideteksi. Sebelum dilakukannya deteksi huruf harus melakukan normalisasi citra agar sistem dapat mengenali huruf pada citra. *Optical Character Recognition* terbukti mampu mengenali citra yang dideteksi pada data latih. Penambahan *threshold* pada citra berguna untuk mengklasifikasikan nilai-nilai pixel, selanjutnya dilakukan nilai maksimum pada citra lalu tahapan terakhir mengkonversi gambar menjadi biner. Sehingga mempercepat sistem dalam mengenali kharakter huruf pada citra yang telah dideteksi.
2. Berdasarkan dari hasil dari pengujian, mendeteksi rambu petunjuk arah dengan kecepatan maksimum 30 km/jam dan deteksi rambu petunjuk arah tergantung pada pencahayaan pada rambu petunjuk arah yang cukup, jika cahaya pada rambu petunjuk arah cukup maka citra dapat dideteksi sedangkan jika citra kekurangan cahaya maka sistem yang telah dilakukan inialisasi pada citra tidak dapat dideteksi. Pengujian juga dipengaruhi pada spesifikasi pada kamera yang digunakan pada sistem sehingga citra yang deteksi kurang detail. Hasil pengujian dari data uji, akurasi tertinggi mendeteksi rambu petunjuk arah adalah 100% dan terendah 0.13%.

7.2 Saran

Terdapat beberapa saran yang terdapat pada rancang bangun sistem pengenalan rambu petunjuk arah menggunakan metode *Optical character Recognition* diantaranya :

1. Penelitian ini mendeteksi kata kurang akurat karena spesifikasi dari kamera yang digunakan 5 megapixel sehingga akurasi dari citra kurang dalam mendeteksi huruf yang terdapat pada citra. Dapat menggunakan kamera lebih tinggi spesifikasi dari kamera.
2. Penelitian ini hanya mendeteksi rambu petunjuk arah yang hanya memiliki satu arah panah. Pada penelitian selanjutnya diharapkan dapat mendeteksi rambu petunjuk arah yang mempunyai banyak arah panah sehingga dapat menemukan lebih banyak citra yang akan dideteksi dengan menggunakan spesifikasi kamera lebih dari 5 megapixel dan ditambahkan database rambu petunjuk arah sehingga pengujian lebih baik dari penelitian ini.

DAFTAR PUSTAKA

- Abdelhamid, M., Azzedine , B. & El, H. K., 2016. MSER-Based Text Detection and Communication Algorithm for Autonomous Vehicles. *IEEE Symposium on Computers and Communication (ISCC)*, Issue 16.
- Aristyagama, Y. H., December 2015. Pengenalan Karakter Sintaktik menggunakan Algoritma Otsu dan Zhang-Suen.
- Bahri, R. S. & Maliki, I., 2015. PERBANDINGAN ALGORITMA TEMPLATE MATCHING DAN FEATURE EXTRACTION PADA OPTICAL CHARACTER RECOGNITION Jurnal Komputer dan Informatika (KOMPUTA). Volume 1.
- Brown, E. W., 1992. Character Recognition by Feature Point Extraction..
- Faisal Mohammad, J. A. M. S. P. G., 2014. Optical Character Recognition Implementation Using Pattern Matching. (*IJCSIT*) *International Journal of Computer Science and Information Technologies*, Volume 5 , p. 2.
- Fauzi, M. R., Nugroho, A. & Ajulian, A., 2013. MENGUBAH TULISAN TANGAN MENJADI TEXT DIGITAL OCR (OPTICAL CHARACTER RECOGNITION) DENGAN MENGGUNAKAN METODE SEGMENTASI DAN KORELASI. Volume 2.
- Greenhalgh, J. & Mirmehdi, M., 2015. Recognizing Text-Based Traffic Signs. *TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, Volume 16.
- G, V. & Y, R. B., January 2014. Vision Based Assistive System for Label Detection with Voice Output. *International Journal of Innovative Research in Science, Engineering and Technology*, Volume III.
- Halimatus Sa'diyah, R. I. A. H., 2011. APLIKASI TRANSFORMASI HOUGH UNTUK DETEKSI GARIS LURUS. *Diponegoro University Institutional Repository*.
- INDONESIA, M. P. R., 2014. RAMBU LALU LINTAS. *PM 13 TAHUN 2014*.
- Maula, A. R. et al., 2014. OPTICAL CHARACTER RECOGNITION DENGAN METODE NAIVE BAYES.
- Maulida, B. A., 2010. *Sistem Analog VS Sistem Digital*. [Online] Available at: blog.ub.ac.id [Accessed 21 Agustus 2017].
- Otsu, N., 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE*, Volume IX.
- Pratt, W., 2007. Digital Image Processing. *Wiley Interscience, A John Wiley & Sons, Inc.*
- R, M. S. & Sharmila, D. T. S., 2016. Emergency exit sign detection system for visually impaired people.

- Satria Priambada, D. H. W., n.d. Levensthein Distance as a Post-Process to Improve the Performance of OCR in Written Road Signs.
- Syafi'i, S. I., Wahyuningrum, R. T. & Muntasa, A., 2015. SEGMENTASI OBYEK PADA CITRA DIGITAL MENGGUNAKAN METODE OTSU THRESHOLDING. *Jurnal Informatika*, Volume 13.
- Tauchid, N. A., Rumani, R. & Irawan, B., 2015. ANALISIS PERFORMANSI METODE KNN (K-NEAREST NEIGHBOR) UNTUK PENGENALAN KARAKTER PADA PLAT NOMOR KENDARAAN DI RASPBERRY PI. *e-Proceeding of Engineering*, 2(2), p. 3575.
- Yingying Zhu, M. L. ,. M. Y. ,. a. W. L., 2018. Cascaded Segmentation-Detection Networks for Text-Based Traffic Sign Detection. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 19(1).
- Yuhang , H., Shi , C., Yifeng, P. & Kai, N., 2014. Using Edit Distance and Junction Feature to Detect and Recognize. *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Issue 14.

