



TUGAS AKHIR TF-141581

**RANCANG BANGUN SISTEM DETEKSI RAMBU - RAMBU
LALU LINTAS MENGGUNAKAN JARINGAN SYARAF
TIRUAN BERBASIS *RASPBERRY PI***

**MOH. FIQIH TARMIDZI HAKIM
NRP 02311440000113**

**Dosen Pembimbing:
Ir. Apriani Kusumawardhani, M.Sc
Andi Rahmadiansah, S.T, M.T**

**DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

Halaman Ini Sengaja Dikosongkan



FINAL PROJECT TF-141581

***DESIGN SYSTEM DETECTION OF TRAFFIC SIGNS
RECOGNITION USING ATIFICIAL NEURAL
NETWORK BASED ON RASPBERRY PI***

**MOH. FIQIH TARMIDZI HAKIM
NRP 02311440000113**

Supervisors:

**Ir. Apriani Kusumawardhani, M.Sc
Andi Rahmadiansah, S.T, M.T**

***DEPARTMENT OF ENGINEERING PHYSICS
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018***

This Page Is Intentionally Left Blank

**RANCANG BANGUN SISTEM DETEKSI RAMBU –
RAMBU LALU LINTAS MENGGUNAKAN JARINGAN
SYARAF TIRUAN BERBASIS *RASPBERRY PI***

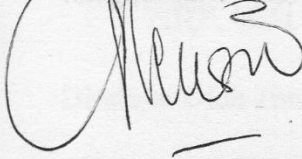
TUGAS AKHIR

Moh. Fiqih Tarmidzi Hakim
NRP. 0231 14 4000 0133

Surabaya, 25 Juli 2018

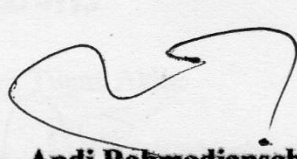
Mengetahui/Menyetujui

Pembimbing I



Ir. Apriani Kusumawardhani, M.Sc
NIP. 19530404 197901 2 001

Pembimbing II



Andi Rahmadiansah, S.T, M.T
NIP. 19790517 200312 1 002

**Kepala Departemen
Teknik Fisika FTI - ITS**



Agus Muhamad Hatta, S.T., M.Si., Ph.D.
NIP. 19780902 200312 1 002

Halaman Ini Sengaja Dikosongkan

**RANCANG BANGUN SISTEM DETEKSI RAMBU –
RAMBU LALU LINTAS MENGGUNAKAN
JARINGAN SYARAF TIRUAN BERBASIS
*RASPBERRY PI***

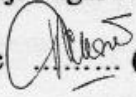

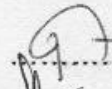


TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik Pada
Bidang Studi Fotonika
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya

Oleh:

Moh. Fiqih Tarmidzi Hakim
NRP. 0231 14 4000 0113

Disetujui Oleh Tim Penguji Tugas Akhir

1. Ir. Apriani Kusumawardhani, M.Sc  (Pembimbing I)
2. Andi Rahmadiansah, S.T., M.T  (Pembimbing II)
3. Agus M Hatta, S.T., M.Si., Ph.D  (Ketua Penguji)
4. Prof. Dr. Ir. Sekartedjo, M.sc  (Penguji)
5. Lizda Johar Mawarani, S.T., M.T  (Penguji)

SURABAYA, 25 JULI 2018

Halaman Ini Sengaja Dikosongkan

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Moh. Fiqih Tarmidzi Hakim
NRP : 02311 14 4000 0113
Departemen : Teknik Fisika FTI – ITS

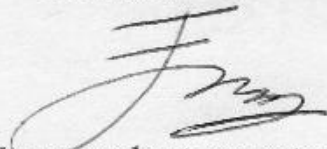
Dengan ini menyatakan bahwa tugas akhir saya yang berjudul:

RANCANG BANGUN SISTEM DETEKSI RAMBU – RAMBU LALU LINTAS MENGUNAKAN JARINGAN SYARAF TIRUAN BERBASIS *RASPBERRY PI*.

Adalah bebas dari plagiasi. Apabila pernyataan ini terbukti tidak benar, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian Surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 25 Juli 2018



Yang membuat pernyataan
Moh. Fiqih Tarmidzi Hakim

Halaman Ini Sengaja Dikosongkan

**RANCANG BANGUN SISTEM DETEKSI RAMBU – RAMBU
LALU LINTAS MENGGUNAKAN JARINGAN SYARAF
TIRUAN BERBASIS *RASPBERRY PI***

Nama Mahasiswa : Moh. Fiqih Tarmidzi Hakim
NRP : 0231 14 4000 0113
Departemen : Teknik Fisika FTI - ITS
Dosen Pembimbing : Ir. Apriani Kusumawardhani, M.Sc
Andi Rahmadiansah, S.T., M.T

Abstrak

Sistem Pengenalan rambu lalu lintas sedang menjadi topik yang menarik untuk dikembangkan. Penelitian ini bertujuan untuk membuat aplikasi untuk mendeteksi rambu-rambu lalu lintas berdasarkan *database* yang telah dibuat. Sistem pengenalan rambu lalu lintas merupakan bagian kecil dari sistem *Autonomous Driver Assistant*, yang merupakan sistem asisten kendali otomatis pada kendaraan mobil. Jaringan Syaraf Tiruan (JST) merupakan salah satu dari sekian metode pengenalan rambu lalu lintas yang mampu mengatasi permasalahan pada sistem pengenalan.

Prinsip kerja dari JST yaitu melalui *training* yang akan diberikan kepada objek gambar rambu lalu lintas yang menghasilkan *training* berupa *database*. Keunggulan dari metode yang dilakukan pada tugas akhir ini yakni bersifat fleksibel, artinya dapat menambahkan berapapun jumlah objek deteksi yang ingin ditambahkan, dan jika terdapat sebuah aturan baru pada rambu lalu lintas, maka metode ini sangat baik untuk diterapkan karena metode ini bersifat cepat, simple, dan efisien. Sistem ini dibagi menjadi dua tahapan, yakni tahap data *preprocessing* dan *pattern recognition*. *Training* yang dilakukan pada metode ini yakni dengan menggunakan Jaringan Syaraf Tiruan *Back Propagation*.

Metode ini mempunyai konsep dengan mencocokkan data yang dideteksi dengan *database* yang telah dibuat, untuk ditampilkan objek deteksi dari pengenalannya. Dengan hasil pengujian yang telah dilakukan pada sistem, dapat mendeteksi dan mengenali rambu-rambu lalu lintas dengan akurasi rata-rata deteksi mencapai 86%.

Kata Kunci – sistem pengenalan, rambu lalu lintas, *database*, JST

Halaman Ini Sengaja Dikosongkan

**DESIGN SYSTEM DETECTION SYSTEM OF TRAFFIC SIGNS
RECOGNITION USING ARTIFICIAL NEURAL NETWORK
BASED ON RASPBERRY PI**

Collage Student Name : Moh. Fiqih Tarmidzi Hakim
NRP : 0231 14 4000 0113
Departement : Teknik Fisika FTI-ITS
Supervisor : Ir. Apriani Kusumawardhani, M.Sc
Andi Rahmadiansah, S.T, M.T

Abstract

The Traffic Sign recognition system is an interesting topic to develop. This study aims to create an application to detect traffic signs based on a database that has been created. The traffic sign recognition system is a small part of the Autonomous Driver Assistant system, which is an automated control assistant system on automobile vehicles. Artificial Neural Network (JST) is one of the many methods of introducing traffic signs that can overcome problems in the recognition system.

The working principle of ANN is through training that will be given to the traffic image object that generates training in the form of database. The advantage of this method is flexible, meaning it can add any number of detection objects to be added, and if there is a new rule in traffic sign, this method is very good to be applied because this method is fast, simple, and efficient. The system is divided into two stages, namely the data preprocessing stage and pattern recognition. Training done on this method that is by using Back Propagation Neural Network.

This method has a concept by matching the detected data with the database that has been created, to be displayed object detection from the introduction. With the results of tests that have been done on the system, can detect and recognize traffic signs with an average detection accuracy of 86%.

Key-words – detections, traffic Signs, Artificial neural network

This Page Is Intentionally Left Blank

KATA PENGANTAR

Puji syukur senantiasa dipanjatkan kepada Allah SWT, Maha Agung dan Maha Bijaksana atas berkah, rahmat, petunjuk dan karunia-Nya serta sholawat kepada Nabi Muhammad sehingga penulis mampu menyelesaikan dan menyusun Tugas Akhir yang berjudul "**Rancang Bangun Sistem Deteksi Rambu – Rambu Lalu Lintas Menggunakan Jaringan Syaraf Tiruan Berbasis *Raspberry Pi***". Penulis telah banyak mendapatkan bantuan dari berbagai pihak untuk menyelesaikan Tugas Akhir ini. Untuk itu diucapkan terimakasih kepada:

1. Bapak Agus Muhammad Hatta, S.T., M.Si., Ph.D selaku Ketua Departemen Teknik Fisika ITS saat ini.
2. Bapak Hendra Cordova, S.T., M.T selaku Ketua Program Studi S-1 Teknik Fisika FTI – ITS saat ini.
3. Ibu Ir. Apriani Kusumawardhani, M.Sc dan Bapak Andi Rahmadiansah, S.T., M.T selaku Dosen Pembimbing yang telah memberikan arahan dan bimbingan dalam menyelesaikan Tugas Akhir ini.
4. Bapak Dr.rer.nat.Ir. Aulia Muhammad Taufiq Nasution M.Sc. Selaku dosen wali selama menempuh pendidikan di Teknik Fisika ITS.
5. Bapak dan Ibu dosen serta karyawan departemen Teknik Fisika.
6. Teman – teman Laboratorium Simulasi dan Komputasi yang telah membantu pengujian dan markas untuk membuat Tugas Akhir.
7. Teman – teman Laboratorium Fisika Rekayasa yang telah membantu dalam proses pengambilan data pengujian untuk Tugas Akhir.
8. Moh Turkan, Nur Cholifah, selaku kedua orang tua saya yang selalu mendukung secara moril dan secara spiritual dalam keadaan apapun.

9. Adik saya Nur Maulany Fauziah yang selalu memberikan semangat dalam keadaan apapun.
10. Mas Dodi Arif Hidayat, Pak Edwin Matatula, Galih Yoga Wardhana, Faris Mahdi Kholiq, Edo Sier Keusuma sebagai teman kosan *green house* 21/B yang selalu menghibur penulis ketika sedang jenuh dan bahkan membantu dalam perancangan *prototype*, meminjamkan kamar untuk pengambilan data, dan meminjamkan banyak alat serta sumbangsih lainnya sehingga Tugas Akhir ini dapat selesai tepat waktu.
11. Tarezqi Mochtar Rohma, Roni Harianto, Nur Syamsyi Inayah, yang telah mendukung penulis selama empat tahun kuliah di kerasnya Surabaya.
12. Mas Dan Sulaiman atas bimbingannya dalam penulisan Laporan Tugas Akhir
13. Kurnia Adiastana dalam membantu penulis untuk proses penyelesaian algoritma program
14. Teman-teman di HMTF yang telah membantu pembentukan karakter dalam hidup saya.
15. Tim TA Pak Andi Horee sebagai Tim seperjuangan dalam menyelesaikan Tugas Akhir bidang Rekayasa Bahan.
16. Teman-teman Teknik Fisika F-49 *Tree Fortrees* angkatanku yang tercinta yang tak bisa saya sebutkan satu-satu.

Penulis menyadari bahwa penulisan laporan Tugas Akhir ini masih banyak dalam kekurangan. Oleh karena itu sangat diharapkan kritik dan saran dari semua pihak sehingga mampu lebih baik lagi. Penulis berharap semoga laporan ini dapat menambah wawasan yang bermanfaat bagi pembacanya.

Surabaya, 25 Juli 2018

DAFTAR ISI

HALAMAN JUDUL.....	i
<i>TITLE PAGE</i>	iii
LEMBAR PENGESAHAN I	v
LEMBAR PENGESAHAN II	vii
PERNYATAAN BEBAS PLAGIASI	ix
ABSTRAK.....	xi
<i>ABSTRACT</i>	xiii
KATA PENGANTAR.....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian.....	3
BAB 2 DASAR TEORI	5
2.1 <i>Intelligent Transportation System</i>	5
2.2 Rambu Lalu Lintas	7
2.3 Pengelompokan Rambu Lalu Lintas	8
2.4 Citra Digital.....	10
2.5 Pengolahan Citra Digital	13
2.6 Jaringan Syaraf Tiruan (JST).....	25
2.7 <i>Raspberry Pi</i>	27
BAB III METODOLOGI	31
3.1 Study Literatur.....	32
3.2 Alat dan Bahan	32
3.3 Perancangan Sistem.....	33

3.4	Perancangan Jaringan Syaraf Tiruan <i>Back Propagation</i> .	36
3.5	Proses <i>Training</i>	38
3.7	Pengujian Alat	46
3.8	Analisa Data dan Pembahasan	47
3.9	Penulisan Laporan	47
BAB IV HASIL DAN PEMBAHASAN		49
4.1	Pengujian Perangkat Lunak (<i>Software</i>).....	49
4.2	Pembahasan	66
BAB V KESIMPULAN DAN SARAN		67
5.1	Kesimpulan.....	67
5.2	Saran.....	67
DAFTAR PUSTAKA		69
LAMPIRAN.....		71
Biodata Penulis		109

DAFTAR GAMBAR

Gambar 2.1	Elemen <i>Intelligent Transportation System</i>	5
Gambar 2.2	Rambu peringatan.....	8
Gambar 2.3	Rambu petunjuk	8
Gambar 2.4	Rambu larangan.....	9
Gambar 2.5	Rambu perintah	9
Gambar 2.6	Citra digital yang diperbesar	11
Gambar 2.7	Binerisasi citra; (a) Sebelum, (b) Sesudah.....	13
Gambar 2.8	Tulang daun dapat dianggap sebagai bagian daun	14
Gambar 2.9	Daun-daun yang bersinggungan dapat dipisahkan	15
Gambar 2.10	Contoh beberapa kernel	15
Gambar 2.11	Operasi kernel terhadap citra	16
Gambar 2.12	Perbandingan operasi erosi, <i>opening</i> , dan <i>closing</i>	17
Gambar 2.13	Piksel dan 8 piksel tetangga.....	18
Gambar 2.14	Tepi objek yang diperoleh melalui tepi biner	19
Gambar 2.15	Pemisahan objek daun terhadap latar belakang	19
Gambar 2.16	Segmentasi sebagai langkah awal sistem	21
Gambar 2.17	Gambaran operasi penggunaan <i>filter median</i>	22
Gambar 2.18	Citra (a) <i>normal difference</i> (b) <i>double difference</i>	25
Gambar 2.21	Neuron Y (Siang, 2009).....	26
Gambar 2.22	Komponen <i>Raspberry Pi</i>	28
Gambar 2.23	Fungsi <i>port GPIO raspberry pi zero</i> dan 3.....	29
Gambar 3.1	Diagram alir penelitian tugas akhir.....	31
Gambar 3.2	Ilustrasi kerja sistem.....	33
Gambar 3.3	Desain bidang uji.....	34
Gambar 3.4	Desain alat keseluruhan.....	35
Gambar 3.5	Desain perancangan tata letak kamera.....	36
Gambar 3.6	Diagram alir algoritma jaringan syaraf tiruan.....	37
Gambar 3.7	<i>Rawdata</i> dengan citra positif.....	38
Gambar 3.8	<i>Rawdata</i> dengan citra negatif.....	39
Gambar 3.9	Penambahan citra negative pada file <i>database</i>	39
Gambar 3.10	Proses <i>sampling</i> pada objek rambu lalu lintas.....	40
Gambar 3.11	Hasil dari <i>sampling</i> dan konversi citra ke vektor.....	40
Gambar 3.12	Proses <i>training</i> yang dilakukan pada citra gambar.....	41
Gambar 3.13	Perangkat keras keseluruhan (a) bentuk fisik keseluruhan.....	43

Gambar 4.1	Hasil pengujian pada rambu lalu lintas dilarang belok kiri.....	52
Gambar 4.2	Hasil pengujian pada rambu lalu lintas dilarang berhenti.....	53
Gambar 4.3	Hasil pengujian pada rambu lalu lintas dilarang berhenti.....	53
Gambar 4.4	Hasil pengujian pada rambu lalu lintas dilarang parkir	54
Gambar 4.5	Hasil pengujian pada rambu lalu lintas dilarang putar balik	55
Gambar 4.6	Hasil pengujian pada rambu lalu lintas dilarang belok kanan.....	55
Gambar 4.7	Hasil pengujian pada rambu lalu lintas tempat penyebrangan.....	56
Gambar 4.8	Hasil pengujian pada rambu lalu lintas perempatan.....	57
Gambar 4.9	Hasil pengujian pada rambu lalu lintas arah wajib diambil.....	57
Gambar 4.10	Hasil pengujian pada rambu lalu lintas arah wajib diambil.....	58
Gambar 4.11	Hasil pengujian pada rambu lalu lintas <i>stop</i>	59
Gambar 4.12	Hasil pengujian pada rambu lalu lintas jalur sepeda	59
Gambar 4.13	Hasil pengujian pada rambu lalu lintas tempat parkir	60
Gambar 4.14	Hasil pengujian pada rambu lalu lintas hati-hati.....	61
Gambar 4.15	Hasil pengujian variasi kecepatan pada rambu lalu lintas dilarang belok kiri	63
Gambar 4.16	Hasil pengujian variasi kecepatan pada rambu lalu lintas dilarang berhenti	64
Gambar 4.17	Hasil pengujian variasi kecepatan pada rambu lalu lintas dilarang putar balik	64
Gambar 4.18	Hasil pengujian variasi kecepatan pada rambu lalu lintas dilarang belok kanan	65
Gambar 4.19	Hasil pengujian variasi kecepatan pada rambu lalu lintas ambil arah disini.....	65
Gambar 4.20	Hasil pengujian variasi kecepatan pada rambu lalu lintas wajib arah kanan	66

DAFTAR TABEL

Tabel 2.1	Aplikasi Segmentasi pada Citra.....	20
Tabel 3.1	Parameter Algoritma Jaringan Syaraf Tiruan	36
Tabel 4.1	Pengujian Objek Rambu Lalu Lintas dengan Menggunakan	50
Tabel 4.2	Pengujian Objek Rambu Lalu Lintas dengan Menggunakan	62

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengolahan citra digital dapat didefinisikan sebagai akuisisi dan pengolahan informasi visual yang dilakukan oleh komputer. Pengolahan ini sekarang telah banyak digunakan oleh perusahaan mobil modern sebagai sistem *autonomous driver assistant*. Karena dapat memudahkan pengemudi dalam kegiatan berkendara, sistem ini memiliki peran penting. Dengan sistem tersebut kendaranya dapat mengenali objek dan dapat mengambil keputusan untuk sebuah tindakan yang sesuai dengan keadaan sekitarnya ^[1]. Disisi lain salah satu masalah adalah tingginya angka kecelakaan yang terjadi, kemudian kemacetan yang semakin parah, dan pelanggaran lalu lintas.

Untuk mengatasi masalah tersebut diperlukan suatu sistem yang dinamakan *Intelligent Transportation System*. Yakni sebuah sistem yang memadukan antara faktor manusia (*human*), jalan (*road*), dan kendaraan (*vehicles*) dengan memanfaatkan *state of the the art* teknologi informasi. Salah satu teknologi yang saat ini diterapkan dan sedang dikembangkan di dalam dunia otomotif yakni *Advanced Driver Assistance Systems*. Yakni sebuah sistem yang dapat digunakan untuk mengenali rambu lalu lintas di jalan raya, kemudian memberitahukan kepada pengemudi mengenai arti dari rambu-rambu lalu lintas tersebut untuk mengurangi atau menambah batas kecepatan dari mobil, memberitahu pengemudi adanya peringatan, adanya bahaya didepan, dan lain sebagainya.

Teknologi ini mempunyai kemampuan untuk meningkatkan sistem keamanan bagi pengemudi dengan selalu mengawasi pengemudi dan lingkungan sekitarnya yang ada pada kendaraan mobil ^[2]. Kendala utama dari sistem ini adalah sistem harus dapat mengenali kondisi kendaraan saat kendaraan sedang melaju dengan cepat, selain itu sistem harus tanggap dengan segala perubahan cuaca lingkungan sekitar dengan cepat dan tetap bekerja secara akurat dan secara terus menerus (*real time*). Beberapa

metode terkait pengenalan citra rambu lalu lintas yang diantaranya telah banyak diterapkan di *Advanced Driver Assistance Systems*, beberapa diantaranya yang telah dikembangkan adalah *SVM Method*, *HOG features*, *Ramer-Douglas-Peucker algorithm*, *SURF algorithm*, *Template Matching*.^[3]

Pada Tugas Akhir ini dibuat alat pengenalan rambu-rambu lalu lintas dengan memanfaatkan *USB Camera* dimana *output* dari pengenalan yang akan diimplementasikan pada sebuah *Autonomous Car*. Salah satu metode Pengenalan rambu yang sederhana dan mudah digunakan dalam mengenali citra adalah dengan menggunakan metode *Template Matching*. Metode tersebut bekerja dengan mencocokkan gambar yang diambil dengan *template*. Metode ini memiliki tingkat keberhasilan yang tinggi. Pada penelitian ini dilakukan metode yang akan menggunakan Jaringan Syaraf Tiruan. Metode ini juga memiliki tingkat keberhasilan yang tinggi mengingat metode ini menggunakan *database* yang lebih spesifik.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan yang diangkat pada Tugas Akhir ini adalah. Bagaimana membangun sistem pengenalan rambu-rambu lalu lintas menggunakan kamera yang dapat diterjemahkan menjadi bahasa komputer berbasis *raspberry pi* untuk sinyal pengenalan pada mobil dengan menggunakan Jaringan Syaraf Tiruan?

1.3 Tujuan

Adapun tujuan dilakukan penelitian ini adalah. Membangun sistem pengenalan rambu-rambu lalu lintas menggunakan kamera yang dapat diterjemahkan menjadi bahasa komputer berbasis *raspberry pi* untuk sinyal pengenalan pada mobil dengan menggunakan Jaringan Syaraf Tiruan.

1.4 Batasan Masalah

Batasan masalah yang ada pada penelitian ini adalah sebagai berikut:

1. File citra yang digunakan adalah format video *realtime*.
2. Rambu lalu lintas yang dideteksi dan dikenali yakni rambu lalu lintas yang berada di negara Indonesia dengan rambu-rambu Dilarang Belok Kanan, Kiri, Dilarang Berhenti, Dilarang Masuk, Dilarang Parkir, Dilarang Putar Balik, STOP, Lewat Kanan, Jalur Wajib Ambil, Jalur Sepeda, Hati-Hati, Penyebrangan, Perempatan, Tempat Parkir.
3. Standar ukuran, ketinggian, dan ketentuan rambu lalu lintas menggunakan panduan penempatan fasilitas perlengkapan jalan dari Departemen Perhubungan Direktorat Jenderal Perhubungan Darat dan Direktorat Bina Sistem Transportasi Perkotaan.
4. Batas kemiringan rambu adalah kurang lebih 15 derajat dari sudut pandang kamera yang terpasang pada mobil, sehingga didapatkan pengenalan objek yang baik.
5. Kamera yang digunakan yakni *webcam* Logitech C170 dan dipasang dimobil secara statis.
6. Software yang digunakan dalam implementasi pada penelitian ini adalah *Phyton* dan *OpenCV*
7. Mobil yang digunakan berupa *prototype* skala 1:16 dengan ukuran panjang mobil 25cm, lebar 11cm, dan tinggi mobil 7cm.

1.5 Manfaat Penelitian

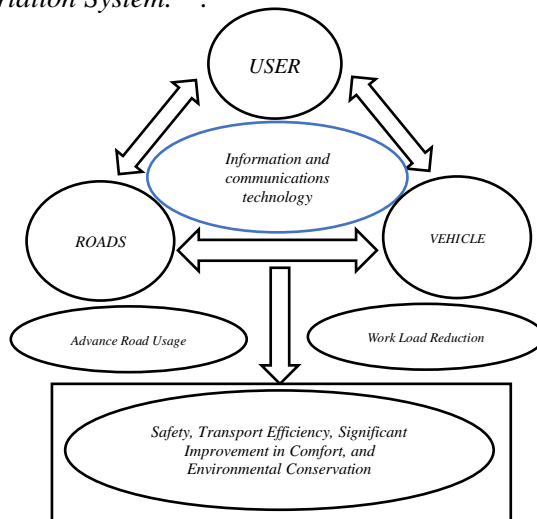
Manfaat yang diharapkan dari penelitian ini adalah dapat meningkatkan sistem keamanan bagi pengendara sebagai sistem yang memberitahu adanya bahaya dan sebagai peringatan di lingkungan sekitar untuk pengendara, penulis berharap kedepannya sistem ini dapat diimplementasikan kedalam *autonomous car*.

Halaman Ini Sengaja Dikosongkan

BAB 2 DASAR TEORI

2.1 *Intelligent Transportation System*

Intelligent Transportation System adalah suatu sistem yang memadukan antara faktor manusia (*people*), jalan (*road*), dan kendaraan (*vehicles*) dengan memanfaatkan *state of the art* teknologi informasi. *Intelligent Transportation System* merupakan sintesa dari beberapa teknologi seperti penentuan posisi, komunikasi, sistem informasi, kontrol, dan elektronik. Dalam kaitannya dengan teknologi pendukung sistem *Intelligent Transportation System*, GPS biasanya berperan sebagai teknologi penentuan posisinya dan *Geographic Information System* (GIS) berperan sebagai teknologi sistem informasinya. Sistem navigasi *Intelligent Transportation System* dapat diklasifikasikan dalam empat tipe, yakni: *Autonomous Intelligent Transportation System*, *Fleet Management Intelligent Transportation System*, *Advisory Intelligent Transportation System*, dan *Inventory Intelligent Transportation System*.^[4]



Gambar 2.1 Elemen *intelligent transportation system*^[5]

2.1.1 *Sistem Autonomous Intelligent Transportation System*

Sistem Autonomous Intelligent Transportation System terdiri dari sistem penentuan posisi dan sistem peta elektronik yang ditempatkan pada kendaraan dan dimaksudkan untuk memberikan kemampuan navigasi yang lebih baik bagi pengemudi kendaraan yang bersangkutan. Sarana ini tidak mempunyai komunikasi dengan sistem luar kendaraan kecuali menggunakan GPS untuk penentuan posisinya dimana dalam hal ini diperlukan antena untuk menerima sinyal GPS.

2.1.2 *Fleet Management Intelligent Transportation System*

Fleet Management Intelligent Transportation System berfungsi untuk mengelola kendaraan dari pusat pengontrol (*dispatch center*) melalui hubungan komunikasi. Dalam sistem ini kendaraan-kendaraan yang bersangkutan diperlengkapi dengan sistem penentuan posisi dan umumnya mereka tidak diperlengkapi dengan sistem peta elektronik. Kendaraan-kendaraan tersebut melaporkan posisinya kepusat pengontrol sehingga pusat pengontrol mempunyai kemudahan untuk mengelola pergerakan kendaraan tersebut. Disamping memberikan intruksi-intruksi mengenai pengarahannya, pusat pengontrol juga bertanggung jawab memberikan informasi yang diperlukan oleh pengemudi kendaraan seperti informasi cuaca dan keadaan lalu lintas.

2.1.3 *Sistem Advisory Intelligent Transportation System*

Untuk *Advisory Intelligent Transportation System* menggabungkan aspek penentuan posisi dan sistem peta elektronik dari sistem *autonomous intelligent transportation system* dengan aspek komunikasi dari arsitektur sistem *fleet management intelligent transportation system*, sistem *advisory intelligent transportation system* adalah *autonomous* dalam artian bahwa sistem ini tidak dikontrol oleh suatu pusat pengontrol

(*dispatch center*), tetapi pada saat yang sama sistem ini merupakan bagian dari armada kendaraan yang mendapat pelayanan dari pusat informasi lalu lintas. Pada beberapa sistem *advisory intelligent transportation system*, kendaraan-kendaraan tertentu berdiri sendiri sebagai *traffic probes*, yang memberikan kendaraan lainnya (yang tidak terdefiniskan oleh pusat informasi lalu lintas) informasi-informasi terbaru tentang kondisi lalu lintas dan cuaca.

2.1.4 Sistem *Inventory Intelligent Transportation System*

Yang terakhir adalah *Inventory Intelligent Transportation System*. Sistem ini biasanya terdiri atas kendaraan yang berdiri sendiri dan dilengkapi dengan kamera video digital untuk mengumpulkan data (lengkap dengan koordinat dan waktu pengambilan) yang terkait dengan jalan yang diperlukan antara lain untuk keperluan inventarisasi jalan, pemeliharaan jalan, serta penyelidikan objek-objek pengganggu lalu lintas. Kendaraan yang digunakan juga dilengkapi dengan alat penentu posisi, *data logger*, serta *pendisplay* data dalam bentuk peta elektronik.

2.2 Rambu Lalu Lintas

Rambu lalu lintas adalah salah satu alat perlengkapan jalan dalam bentuk tertentu yang memuat lambing, huruf, angka, kalimat dan perpaduan diantaranya yang digunakan untuk memberikan peringatan, larangan, perintah, dan petunjuk bagi pemakai jalan. Peranan akan adanya rambu lalu lintas sangat penting dimana jika ada pengendara yang kurang tahu akan daerah yang baru dilaluinya, maka rambu lalu lintas adalah petunjuk yang dapat digunakan untuk mendapatkan informasi. Rambu-rambu lalu lintas haruslah reflektif agar terlihat pada kondisi siang dan malam bahkan pada waktu hujan sehingga pengendara selalu terpadu dalam perjalanannya.

2.3 Pengelompokan Rambu Lalu Lintas

Berdasarkan Peraturan Pemerintah (PP) No.43 Tahun 1993. Pengelompokan rambu-rambu lalu lintas berdasarkan jenis pesan, fungsi dan tempat dibagi menjadi lima, yakni sebagai berikut:

- Rambu Peringatan

Rambu yang memperingatkan adanya bahaya agar para pengemudi berhati-hati dalam menjalankan kendaraannya. Rambu ini mempunyai warna dasar kuning dengan *list frame* berwarna hitam



Gambar 2.2 Rambu peringatan ^[6]

- Rambu Petunjuk

Rambu yang memberikan petunjuk atau keterangan kepada pengemudi atau pemakai jalan lainnya tentang petunjuk arah yang harus ditempuh atau petunjuk letak Kota yang akan dituju lengkap dengan nama dan arah letak itu berada. Rambu ini mempunyai warna dasar hijau dan tulisannya berwarna putih.



Gambar 2.3 Rambu petunjuk ^[6]

- **Rambu Larangan**

Rambu yang memberikan larangan kepada pengemudi dan kendaraan untuk memakai jalan, jurusan atau tempat-tempat tertentu. Rambu ini mempunyai warna dasar merah dengan tandanya berwarna putih.



Gambar 2.4 Rambu larangan ^[6]

- **Rambu Perintah**

Rambu yang memberikan perintah kepada semua jenis lalu lintas atau pemakai jalan untuk melaksanakan seperti rambu tersebut misalnya semua jenis kendaraan harus memutar terlebih dahulu untuk belok ke kanan atau ke kiri. Rambu ini mempunyai warna dasar biru dengan tulisannya berwarna putih.



Gambar 2.5 Rambu perintah ^[6]

2.4 Citra Digital

Citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik pengolahan citra. Istilah “Citra” yang digunakan dalam bidang pengolahan citra dapat diartikan sebagai suatu fungsi kontinu dari intensitas cahaya dalam bidang dua dimensi, Citra yang dimaksudkan adalah berupa objek gambar, foto, dan video baik citra video maupun video *realtime* (Berasal dari pengukuran sensor berupa kamera yang dipasang) Pengolahan digitalisasi terdiri dari dua proses, yaitu pencuplikan (*sampling*) posisi, dan kuantisasi intensitas dalam pengolahan citra diolah secara digital menggunakan *software* yang ada pada komputer. Secara matematis citra digolongkan sebagai fungsi yang kontinu (*continue*) dengan intensitas cahaya pada bidang dua dimensi, dan representasikan secara numerik dengan nilai-nilai diskrit. Hal ini ditujukan agar citra dapat diolah secara digital oleh *software* pada komputer. Citra digital dapat dinyatakan dalam matriks dua dimensi $f(x, y)$ dimana ‘ x ’ dan ‘ y ’ merupakan koordinat *pixel* dalam matriks dan ‘ f ’ merupakan derajat intensitas *pixel* tersebut. Citra digital berbentuk matriks dengan ukuran $M \times N$ akan tersusun sebagai berikut:

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & f(1,1) & \dots & f(1, M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix} \quad (2.1)$$

Suatu citra $f(x, y)$ dalam fungsi matematis dapat dituliskan sebagai berikut :

$$0 \leq x \leq M - 1$$

$$0 \leq y \leq N - 1$$

$$0 \leq (x, y) \leq G - 1$$

Dimana,

M = banyaknya baris pada *array* citra

N = banyaknya kolom pada *array* citra

G = banyaknya skala keabuan (*graylevel*)

Besarnya nilai M , N , dan G pada umumnya adalah perpangkatan dari dua.

$$M = 2^m; N = 2^n; G = 2^k \quad (2.2)$$

Dimana nilai m , n , dan k adalah bilangan bulat positif. Interval $(0, G)$ Interval $(0, G)$ disebut skala keabuan (*grayscale*). Besar G tergantung pada proses digitalisasinya. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan 1 (satu) menyatakan intensitas putih. Untuk citra 8-bit, nilai G sama dengan $2^8 = 256$ warna (derajat keabuan).



Gambar 2.6 Citra digital yang diperbesar ^[7]

Contoh citra digital ditunjukkan pada gambar 2.6, dimana sebagian dari citra ini diperbesar untuk memperlihatkan bahwa citra tersebut tersusun atas *pixel – pixel*. Jika kita memperhatikan citra digital secara seksama, kita dapat melihat titik-titik kecil berbentuk segi empat yang membentuk citra tersebut. Titik-titik tersebut merupakan satuan terkecil dari suatu citra digital disebut sebagai “*Picture Element*”, “*Pixel*”, atau “*PeI*”. Jumlah *pixel* persatuan panjang akan menentukan resolusi citra tersebut. Makin banyak *pixel* yang mewakili suatu citra, maka makin tinggi nilai resolusinya dan semakin halus gambarnya. Pada sistem dengan tampilan citra digital yang dirancang dengan baik (beresolusi tinggi), titik-titik kecil tersebut tidak teramati oleh kita yang melihatnya secara normal.

Citra berwarna dapat dinyatakan dengan banyak cara, salah satunya adalah dengan menggunakan sinyal *Red Green Blue* (RGB). Pada cara ini, sebuah citra berwarna dinyatakan sebagai gabungan dari tiga buah citra *monochrome* merah, hijau, dan biru yang berukuran sama. Warna untuk setiap *pixel* nya tergantung dari komposisi ketiga komponen pada koordinat tersebut. Konsep ini digunakan luas untuk berbagai aplikasi citra berwarna. Untuk menyederhanakan perhitungan, citra pada tugas akhir ini akan diolah ke dalam bentuk derajat keabuan (*graylevel*), dimana pada citra berwarna direpresentasikan dengan nilai yang sama pada ketiga komponen R-G-B-nya.

Hubungan formasi *geometri* citra dan representasi citra di dalam komputer adalah penting untuk memahami bagaimana citra digital disimpan dan diolah. Harus ada jembatan antara notasi matematika untuk mengembangkan algoritma pengolahan citra dan notasi algoritma yang digunakan dalam pembuatan program komputer. Komputer yang ada saat ini telah dilengkapi dengan sistem penyimpanan memori dua dimensi yang disebut larik (*array*) atau matriks memori.

Sebuah *pixel* adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Sebuah citra adalah kumpulan *pixel-pixel* yang disusun dalam larik dua-dimensi. Indeks baris dan kolom (x, y) dari sebuah *pixel* yang dinyatakan dalam bilangan bulat. *Pixel* (0, 0) terletak pada sudut kiri atas pada citra, indeks x bergerak ke kanan dan indeks y bergerak ke bawah. Konversi ini dipakai merujuk pada cara penulisan larik yang digunakan dalam pemrograman komputer.

Objek tertentu dapat dideteksi dengan menggunakan pengolahan citra digital. Salah satu metode yang digunakan adalah berdasarkan segmentasi warna. *Hue, Saturation, Value* adalah salah satu metode segmentasi warna yang memiliki beberapa kelebihan yakni mudah dibuat, proses yang cepat dan efektif pada objek rambu-rambu lalu lintas.

2.5 Pengolahan Citra Digital

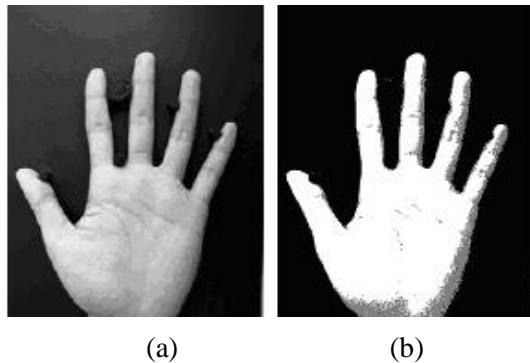
Pengolahan citra digital adalah mengolah citra yang bertujuan untuk mendapatkan suatu informasi dari citra tersebut. Pemrosesan data citra berupa proses binerisasi citra dan proses *morphologi*.

2.5.1 Binerisasi Citra

Binerisasi citra adalah mengubah citra dengan 256 derajat keabuan (*graylevel*) menjadi citra hitam-putih (*black-white*). Cara yang termudah untuk melakukan binerisasi citra adalah dengan menentukan nilai ambang batas (*Thresholding Value*), dan menggolongkan semua *pixel* dengan nilai diatas ambang (*Threshold*) sebagai warna putih, dan *pixel* yang lain sebagai warna hitam.

Permasalahan pada binerisasi citra adalah bagaimana menentukan nilai ambang yang benar. Pada kasus, sangat sulit mencari satu nilai ambang yang cocok untuk semua citra.

$$gray = \frac{R+G+B}{3} \quad (2.3)$$

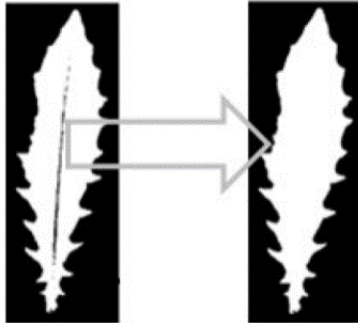


Gambar 2.7 Binerisasi citra; (a) sebelum, (b) sesudah ^[7]

2.5.2 Proses Morfologi

Operasi morfologi merupakan operasi yang umum dikenakan pada citra biner untuk mengubah struktur bentuk obyek yang terkandung dalam citra [8]. Sebagai contoh, lubang pada obyek yang berupa daun akibat dari tulang daun yang tidak tersegmentasi dapat ditutup melalui operasi morfologi sebagaimana ditunjukkan di gambar 2.8. Obyek-obyek daun yang saling berhimpitan pun dapat dipisahkan melalui morfologi, sebagaimana ditunjukkan di gambar 2.9. Beberapa contoh lain aplikasi morfologi adalah sebagai berikut [8].

- Membentuk filter spasial
- Memperoleh skeleton (rangka) objek
- Menentukan letak objek di dalam cita
- Memperoleh bentuk struktur objek

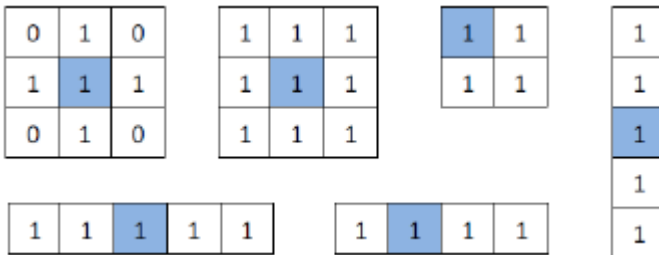


Gambar 2.8 Tulang daun dapat dianggap sebagai bagian daun melalui morfologi [8]

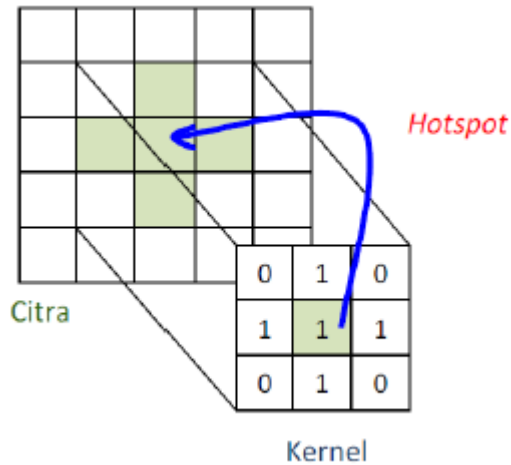


Gambar 2.9 Daun-daun yang bersinggungan dapat dipisahkan melalui morfologi, yang memperkecil ukurannya ^[8]

Inti operasi morfologi melibatkan dua larik piksel. Larik pertama berupa citra yang akan dikenai operasi morfologi, sedangkan larik kedua dinamakan kernel atau *structuring element* (elemen penstruktur) [8]. Contoh kernel ditunjukkan di gambar 2.10. Pada contoh tersebut, piksel pusat yang ditandai dengan warna abu-abu. Piksel pusat ini yang menjadi pusat dalam melakukan operasi terhadap citra, sebagaimana diilustrasikan di gambar 2.11.



Gambar 2.10 Contoh beberapa kernel ^[8]



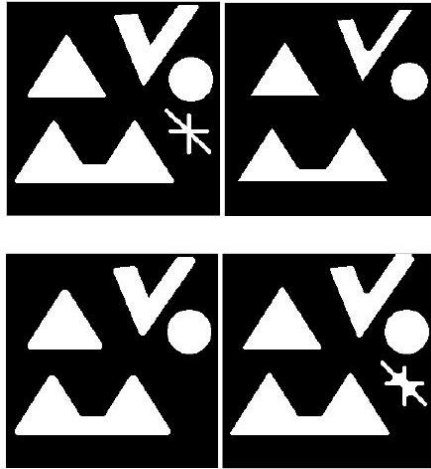
Gambar 2.11 Operasi kernel terhadap citra ^[8]

Operasi dasar pada morfologi ada dua yaitu dilasi dan erosi. Penelitian-penelitian yang menggunakan pengolahan citra biasanya menggunakan morfologi opening dan closing yang merupakan pengembangan dari operasi dasar dilasi dan erosi.

Operasi *opening* adalah operasi erosi yang diikuti dengan dilasi dengan menggunakan elemen penstruktur yang sama. Operasi ini berguna untuk menghaluskan kontur objek dan menghilangkan seluruh piksel di area yang terlalu kecil untuk ditempati oleh elemen penstruktur. Dengan perkataan lain, semua struktur latar depan yang berukuran lebih kecil daripada elemen penstruktur akan tereliminasi oleh erosi dan kemudian penghalusan dilakukan melalui dilasi ^[8]. Definisi operasi *opening* seperti berikut:

$$AoB = (A \ominus B) \oplus B \quad (2.3)$$

Contoh efek *opening* dapat dilihat pada gambar 2.12 di bawah ini



Gambar 2.12 Perbandingan operasi erosi, *opening*, dan *closing* ^[8]

Gambar 2.12 menunjukkan bahwa operasi erosi membuat objek mengecil dan bahkan ada yang hilang. Adapun operasi *opening* membuat ukuran objek relatif tetap Sama, walaupun juga menghilangkan objek berukuran kecil (kurus). Namun, perlu diketahui, operasi *opening* membuat penghalusan di bagian tepi. Perhatikan, ujung segitiga tidak tajam setelah dikenai operasi *opening*. Sebagai pembandingan, gambar 2.12 (d) menunjukkan hasil penggunaan operasi *closing* ^[8].

Operasi *opening* sering dikatakan sebagai *idempotent*. Artinya, jika suatu citra telah dikenai operasi *opening*, penguasaan *opening* dengan elemen penstruktur Sama tidak membawa efek apapun. Sifat ini dapat dituliskan secara matematis seperti berikut:

$$AoB = (A \ominus B) \oplus B \quad (2.4)$$

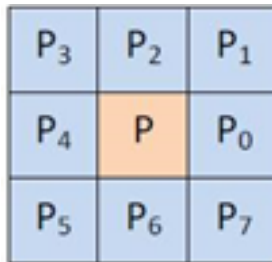
Operator *opening* dapat dimanfaatkan sebagai filter lolos-rendah, filter lolos-tinggi, maupun sebagai tapis lolos-bidang apabila elemen penstruktur yang digunakan berupa cakram. Berikut adalah rumusnya:

- Filter lolos-rendah (*low-pass*) = $A \circ B^h$;
- Filter lolos-tinggi (*high-pass*) = $A - (A \circ B^h)$;
- Filter lolos-bidang (*band-pass*) = $(A \circ B^{h1}) - (A \circ B^{h2})$,
dengan diameter $B^{h1} < B^{h2}$

2.5.3 Operasi pada Citra Biner

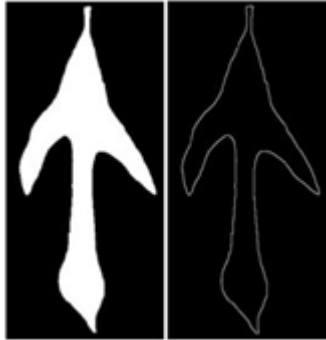
Beberapa pemrosesan citra mengacu pada citra biner. Sebagai contoh, dengan menggunakan citra biner, perbandingan panjang dan lebar objek dapat diperoleh. Di depan juga telah dibahas aplikasi citra biner pada morfologi. Namun, tentu saja masih banyak operasi lain yang memanfaatkan citra biner. Salah satu contohnya adalah ekstraksi tepi objek^[8].

Tepi objek pada citra biner dapat diperoleh melalui algoritma yang dibahas oleh Davis (1990). Pemrosesan dilakukan dengan menggunakan 8-ketetanggaan^[8]. Sebagai penjasas, dapat dilihat pada gambar 2.13.



Gambar 2.13 Piksel dan 8 piksel tetangga^[8].

Contoh ekstraksi tepi objek dapat dilihat pada gambar 2.14 di bawah ini.



Gambar 2.14 Tepi objek yang diperoleh melalui tepi biner ^[8].

2.5.4 Segmentasi Citra

Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek-objek yang terkandung di dalam citra atau membagi citra ke dalam beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut. Pada citra yang mengandung hanya satu objek, objek dibedakan dari latar belakangnya. Contoh ditunjukkan pada gambar 2.15. Untuk contoh penerapan segmentasi adalah untuk membuat fasilitas semacam “*Magic Wand*”, yang biasa terdapat pada perangkat pengedit foto ^[8].



Gambar 2.15 Pemisahan objek daun terhadap latar belakang ^[8].

Gambaran berbagai aplikasi segmentasi serta acuan yang digunakan dapat dilihat di tabel 2.1. Secara prinsip, segmentasi dilakukan untuk mendapatkan objek yang menjadi perhatian.

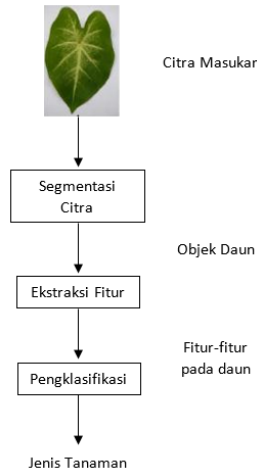
Tabel 2. 1 Aplikasi Segmentasi pada Citra [8].

<i>Objek</i>	<i>Citra</i>	<i>Kegunaan Segmentasi</i>	<i>Acuan yang Digunakan</i>
Mobil	Mobil, jalan, dan latar belakang	Pelacakan mobil	Gerakan dan warna
Struktur permukaan bumi	Foto satelit	Klasifikasi area	Tekstur dan warna
Wajah orang	Kumpulan orang di pasar	Pengenalan wajah	Warna, bentuk, dan tekstur
Apel	Kumpulan apel pada ban berjalan	Pemilahan buah apel berdasarkan ukuran	Bentuk, warna, dan ukuran

Segmentasi juga biasa dilakukan sebagai langkah awal untuk melaksanakan klasifikasi objek. Gambar 2.16 memperlihatkan hal ini. Setelah segmentasi citra dilaksanakan, fitur yang terdapat pada objek diambil. Sebagai contoh, fitur objek dapat berupa perbandingan lebar dan panjang objek, warna rata-rata objek, atau bahkan tekstur pada objek. Selanjutnya, melalui klasifikasi, jenis objek dapat ditentukan. Sebagai contoh, klasifikasi menyatakan bahwa daun termasuk golongan *Aglaonema* [8].

Teknik segmentasi citra didasarkan pada dua properti dasar nilai Aras keabuan: ketidaksinambungan dan kesamaan antar piksel. Pada bentuk pertama, pemisahan citra didasarkan pada perubahan mendadak pada arah keabuan. Contoh yang menggunakan pendekatan seperti itu adalah detektor garis dan

detektor tepi pada citra. Cara kedua didasarkan pada kesamaan antar piksel dalam suatu area. Termasuk dalam Cara kedua ini yaitu: pengembangan berdasarkan histogram, pertumbuhan area, pemisahan dan penggabungan area, pengelompokan atau klasifikasi, pendekatan teori graf, pendekatan yang dipandu pengetahuan atau berbasis aturan [8].



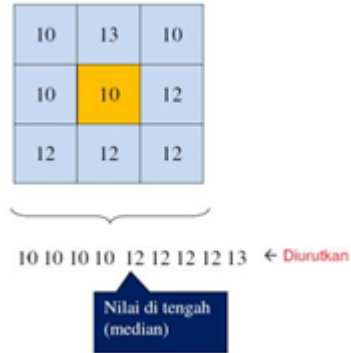
Gambar 2.16 Segmentasi sebagai langkah awal sistem klasifikasi [8].

2.5.5 Filter Median

Filter median sangat populer dalam pengolahan citra. *Filter* ini dapat dipakai untuk menghilangkan derau bintik-bintik. Nilai yang lebih baik digunakan untuk suatu piksel ditentukan oleh nilai median dari setiap piksel dan kedelapan piksel tetangga pada 8-ketetanggaan [9]. Secara matematis, filter dapat dinotasikan seperti berikut:

$$g(x, y) = \text{median} (f(y - 1, x - 1), f(y - 1, x), f(y - 1, x + 1), f(y, x - 1), f(y, x), f(y, x + 1), f(y + 1, x - 1), f(y + 1, x), f(y + 1, x + 1)) \quad (2.5)$$

Contoh untuk satu piksel ditunjukkan pada gambar 2.17



Gambar 2.17 Gambaran operasi penggunaan *filter* median ^[8].

Selain menghilangkan derau bintang-bintang, filter median juga cocok digunakan untuk menghilangkan derau impuls dan derau eksponensial negatif ^[8].

2.5.6 *Filter* Gaussian

Filter Gaussian tergolong sebagai *filter* lolos-rendah yang didasarkan pada fungsi Gaussian. Model dua dimensinya berupa:

$$G(y, x) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.6)$$

Dalam hal ini, σ adalah deviasi standar dan piksel pada pusat (y, x) mendapatkan bobot terbesar berupa 1.

Dengan mengatur nilai terkecil menjadi 1, maka setiap nilai di atas perlu dikalikan dengan 55 (diperoleh dari $1/0.0183$ dan kemudian hasilnya dibulatkan ke atas). Dengan demikian, diperoleh hasil seperti berikut, yang diperoleh dengan mengalikan nilai $G(x, y)$ di depan dengan 55 ^[8].

2.5.7 *Virtual Loop*

Metode *virtual loop* mendeteksi dan menghitung jumlah obyek berdasarkan posisi relatif obyek dengan *virtual loop*, yang

mana dalam penggunaannya begitu sederhana dan efektif, serta menjadi metode dasar untuk menghitung obyek yang bergerak. Ada dua aspek yang dibutuhkan untuk mendeteksi *virtual loop* yang efektif yaitu keberhasilan dalam mensegmentasi obyek yang bergerak dari latar belakangnya dan sebuah algoritma untuk menghindari penghitungan ulang suatu obyek [8].

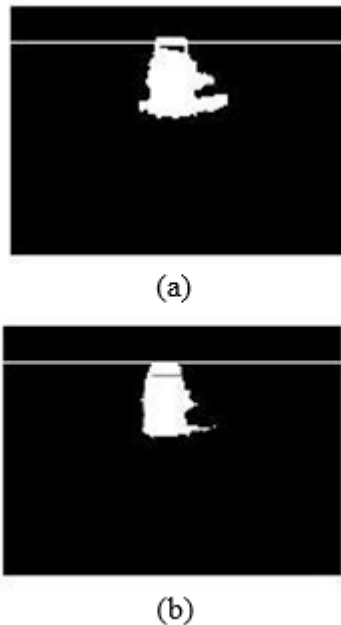
Kelemahan dari metode *virtual loop* tradisional adalah penghitungan yang salah dari obyek yang tidak bergerak, obyek yang bergerak lebih lambat atau obyek yang berjalan berhimpitan [8]. Untuk itu diperlukan algoritma tambahan yang berupa *frame differencing*. *Frame differencing* tradisional mengakuisisi kontur obyek yang bergerak dengan menggunakan perbedaan informasi antara dua frame berturut-turut, dan banyak digunakan dalam segmentasi gerakan karena karakteristiknya mudah diprogram. Selain itu *frame differencing* tradisional kurang peka terhadap perubahan kondisi pencahayaan, sehingga cocok digunakan pada sistem dengan pencahayaan yang tidak stabil. Tetapi pendekatan ini juga mengalami kekurangan seperti *foreground aperture* dan *ghosting*, serta yang utama karena sensitif terhadap kecepatan obyek dan *frame rate* dari video [8].

Untuk mengatasi ini diusulkan variasi metode ini, yang disebut metode “*the double difference*” yang menerapkan perbedaan ambang antara frame pada waktu t , $t-1$, dan $t+1$ dan menggabungkannya dengan logika AND [9]. Migliore, Matteucci, & Naccari mengusulkan sebuah teknik yang memanfaatkan kedua *frame* yaitu *frame difference* dan *background subtraction* [10]. Perbedaan antara *frame* aktual dan *frame* sebelumnya digabungkan dengan perbedaan antara *frame* aktual dan model sebenarnya dari latar belakang untuk mendeteksi obyek bergerak. Citra pertama dari urutan *frame* digunakan sebagai model latar belakang awal dan diperbarui secara dinamis dengan *frame* baru sesuai dengan segmentasi yang bergerak. Pendekatan ini menjanjikan deteksi gerakan obyek yang kuat asalkan model latar belakang memberikan informasi latar belakang yang *up-to-date* [8].

Posisi dan bentuk obyek yang benar merupakan faktor kunci penghitungan obyek yang akurat saat menggunakan metode *virtual loop*, karena sangat bergantung pada nilai piksel. Perbedaan kontur pada citra normal tidak mengekspresikan bentuk obyek dengan baik karena merupakan perpaduan bentuk obyek pada dua perbedaan waktu. Pendekatan “*double difference image*” menggunakan informasi dari *frame* sebelumnya dan *frame* berikutnya untuk memperkirakan bentuk dan posisi obyek, yang memberikan informasi posisi yang lebih akurat [8].

Untuk mendapatkan citra obyek “*double difference*”, yaitu dengan membedakan dua gambar yang dihasilkan dari dua gambar berturut-turut yang sesuai (*frame* pada waktu „ $t-1$ “, „ t “, dan „ $t+1$ “), kemudian dari perbedaan gambar dilakukan binerisasi dan operasi logika AND dieksekusi pada dua gambar ini. Obyek pada citra yang dihasilkan tetap mempertahankan bentuk dan posisinya pada saat „ t “. Pada saat yang Sama, untuk menghilangkan *noise* pada piksel dengan menggunakan operasi morfologi *opening*, *closing*, dan *dilate* pada citra biner yang telah dieksekusi [8].

Gambar 2.18 menunjukkan citra “*normal difference*” dan citra “*double difference*” dari obyek yang terdeteksi pada *frame*. Posisi obyek yang bergerak dideteksi dengan benar pada Gambar 2.18(b), ditandai dengan garis putih, sedangkan citra “*normal difference*” memberikan informasi yang salah. Meskipun perbedaan posisi pada kedua citra cukup kecil, namun memiliki efek yang besar apabila dilakukan estimasi posisi obyek menggunakan *virtual detector*.



Gambar 2.18 Citra (a) *normal difference* (b) *double difference* ^{[11][8]}.

2.6 Jaringan Syaraf Tiruan (JST)

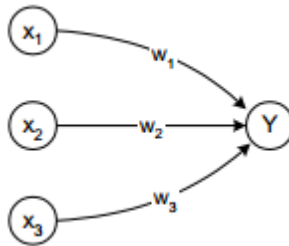
Jaringan syaraf tiruan atau JST adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi ^[12]. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa:

- Pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*)
- Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung
- Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal
- Untuk menentukan keluaran, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan input yang diterima. Besarnya

output ini selanjutnya dibandingkan dengan suatu batas ambang. JST ditentukan oleh tiga hal:

- Pola hubungan neuron (disebut arsitektur jaringan)
- Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algorithm*)
- Fungsi aktivasi

Sebagai contoh perhatikan gambar berikut ini.



Gambar 2.19 Neuron Y ^[12].

Y menerima input dari neuron x_1 , x_2 dan x_3 dengan bobot hubungan masing-masing adalah w_1 , w_2 dan w_3 . Ketiga impuls neuron yang ada dijumlahkan

$$net = x_1w_1 + x_2w_2 + x_3w_3 \quad (2.7)$$

Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $y = f(net)$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal Akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah bobot.

Beberapa aplikasi jaringan syaraf tiruan adalah sebagai berikut:

- Pengenalan Pola (*Pattern Recognition*)
Jaringan syaraf tiruan dapat dipakai untuk mengenali pola (missal huruf, angka, suara atau tanda tangan) yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu

tidak dijumpainya (mungkin wajah/bentuk tubuhnya sudah sedikit berubah).

- *Signal Processing*
Jaringan syaraf tiruan (model *Backpropagation*) dapat dipakai untuk menekan *noise* dalam saluran telpon.
- Peramalan
Jaringan syaraf tiruan juga dapat dipakai untuk meramalkan apa yang akan terjadi dimasa yang akan datang berdasarkan pola kejadian yang ada dimasa lampau. Ini dapat dilakukan mengingat kemampuan jaringan syaraf tiruan untuk mengingat dan membuat generalisasi dari apa yang sudah ada sebelumnya.

Disamping area-area tersebut, jaringan syaraf tiruan juga dilaporkan dapat menyelesaikan masalah dalam bidang kontrol, kedokteran dan lain-lain. Meskipun banyak aplikasi menjanjikan yang dapat dilakukan oleh jaringan syaraf tiruan, namun jaringan syaraf tiruan juga memiliki beberapa keterbatasan umum. Pertama adalah ketidak akuratan hasil yang diperoleh. Jaringan syaraf tiruan bekerja berdasarkan pola terbentuk pada inputnya.

2.7 Raspberry Pi

Raspberry Pi, sering disingkat dengan Nama Raspi, adalah komputer papan tunggal (*single-board circuit*; SBC) yang seukuran dengan kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi dikembangkan oleh yayasan nirlaba yaitu *Raspberry Pi Foundation*, yang digawangi sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris.

2.7.1 Spesifikasi *Raspberry Pi*

Raspberry Pi board dibuat dengan model yang berbeda yaitu *Raspberry Pi* type A, A+ *Raspberry Pi* type B, B+ *Raspberry pi* 2, *Raspberry pi* 3, *Raspberry Pi* zero. Perbedaannya antara lain pada RAM dan Port LAN. Type A RAM = 256 MegaBytes dan tanpa port LAN (*ethernet*), type B = 512

MegaBytes dan terpasang *port* untuk LAN. *Raspberry Pi Board* mempunyai input dan output antara lain:

- *HDMI*, dihubungkan ke LCD TV atau monitor *Personal Computer* yang mempunyai *port HDMI*.
- *Audio output*
- 2 buah *port USB*, digunakan untuk *keyboard* dan *mouse*
- 26 pin *I/O digital*
- *CSI port (Camera Serial Interface)*
- *DSI (Display Serial Interface)*
- *LAN port (Network)*
- *SD Card Slot* untuk *SD Card* memori yang menyimpan sistem operasi berfungsi seperti *hardisk* pada *Personal Computer/ Laptop*



Gambar 2. 20 Komponen *raspberry pi* ^[13]

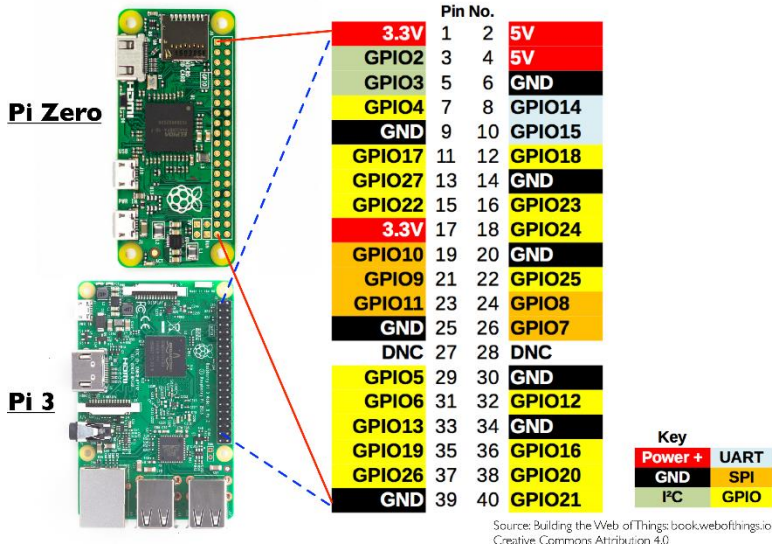
2.7.2 Pin Port GPIO

GPIO merupakan sederet *pin* yang terdiri dari 26 *pin* dengan berbagai fungsi diantaranya adalah sebagai berikut:

Bagian-bagian *Port GPIO* adalah sebagai berikut:

- *Port input-output* atau masukan pengambilan data dari sensor dan keluaran menuju actuator. Terdiri dari *port 7, 11, 12, 13, 15, 16, 18, 22*

- Keluaran berupa tegangan pada *port* 1 dan 2
- *Ground* sebagai pengaman pada *port* 6, 9, 14, 17, 20, 25
- *UART interface* pada *port* 8, 10
- *SP10 interface* pada *port* 19, 21, 23, 24, 26
- *12C interface* pada *port* 3, 5



Gambar 2. 21 Fungsi *port* GPIO raspberry pi zero dan 3 ^[13].

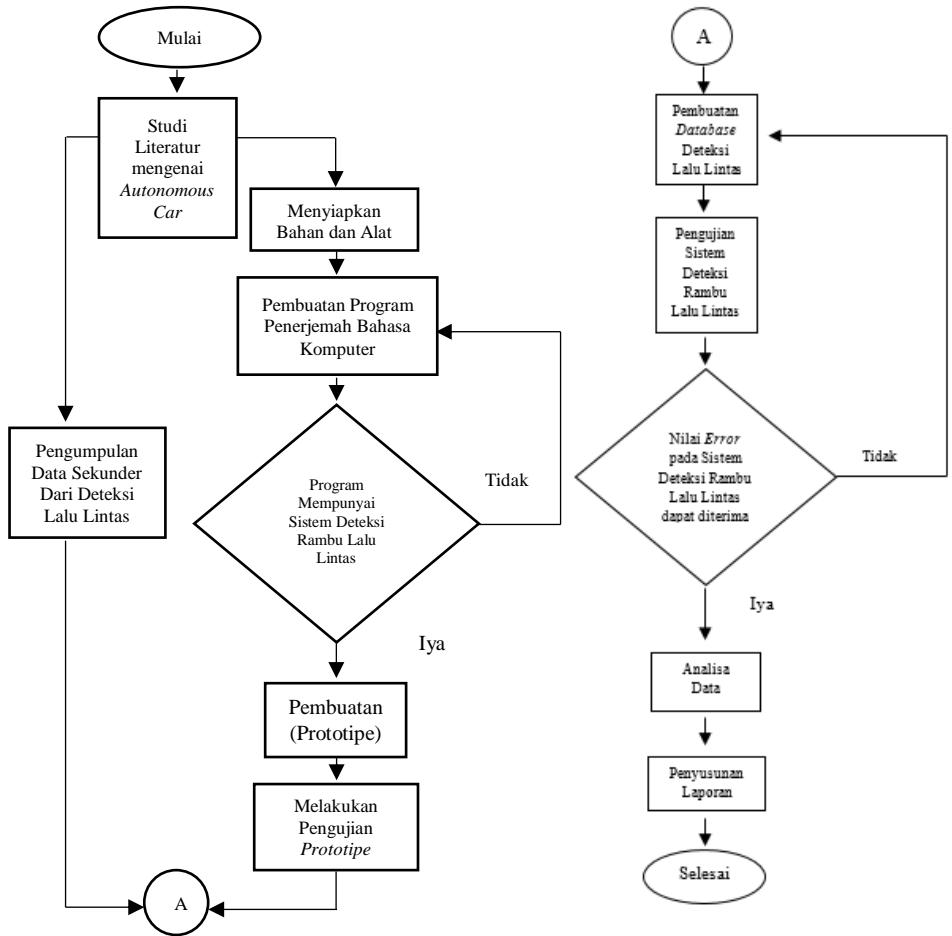
2.8.1 Sistem Operasi *Raspberry Pi*

OS atau Sistem Operasi yang bisa di jalankan di *Raseberry board* antara lain: *Arch Linux ARM*, *Debian GNU/Linux*, *Gentoo*, *Fedora*, *FreeBSD*, *NetBSD*, *Plan 9*, *Inferno*, *Raspbian OS*, *RISC OS*, dan *Slackware Linux*. Dalam menggunakan *microcomputer Raspberry Pi* menggunakan *Personal Computer* yang berbasis linux plus yang mempunyai *input-output* digital seperti yang ada di *board microcontroller*.

Halaman Ini Sengaja Dikosongkan

BAB III METODOLOGI

Penelitian tugas akhir ini dilaksanakan berdasarkan langkah-langkah yang sistematis seperti ditunjukkan pada diagram alir di bawah ini:



Gambar 3.1 Diagram alir penelitian tugas akhir

Langkah langkah penelitian yang ditunjukkan oleh diagram alir di atas Akan dijabarkan pada masing-masing sub Bab berikut:

3.1 Study Literatur

Study literatur Akan difokuskan pada pengenalan rambu lalu lintas dan pembuatan algoritma program pengenalan rambu lalu lintas karena tugas akhir ini mengenai bagaimana membangun sistem pengenalan rambu-rambu lalu lintas. Sumber study literatur dapat berupa penelitian sebelumnya yang bersangkutan, jurnal dan buku.

3.2 Alat dan Bahan

Adapun alat-alat yang digunakan pada tugas akhir ini adalah sebagai berikut:

- a. *Raspberry Pi 3*
- b. *WebCam Logitech C170*
- c. *Display Monitor*
- d. *SD Card*
- e. Kabel USB
- f. Charger HP
- g. *Keyboard*
- h. *Mouse*
- i. Kabel HDMI
- j. Laptop
- k. Marka jalan A0 Skala 1:16

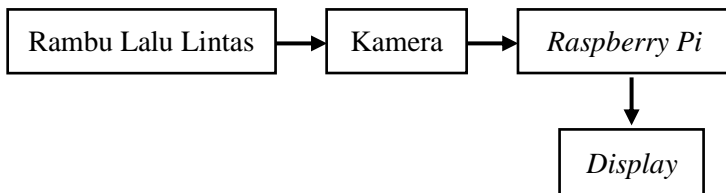
Bahan-bahan yang digunakan pada tugas akhir ini adalah sebagai berikut:

- a. OpenCV
- b. Raspi
- c. OS
- d. Rambu lalu lintas Skala 1:16

3.3 Perancangan Sistem

Dalam perancangan sistem ini dijelaskan perancangan sistem mulai dari menyiapkan alat dan bahan, perancangan *hardware* yang meliputi perancangan pembuatan bentuk fisik alat hingga perancangan *software* yang meliputi perancangan algoritma agar alat dapat mengenali rambu-rambu lalu lintas.

Pada pembuatan tugas akhir ini, algoritma *traffic signs recognition* digunakan untuk mengenali rambu-rambu lalu lintas dengan mempertimbangkan prinsip kerja yang Sama antara obyek yang berupa mobil dengan obyek yang berupa rambu lalu lintas dengan skala 1:16. Algoritma dan *library* program pada alat ini menggunakan referensi dari forum www.github.com yang ditulis oleh *Rem-nonsym* ^[14].



Gambar 3.2 Ilustrasi kerja sistem

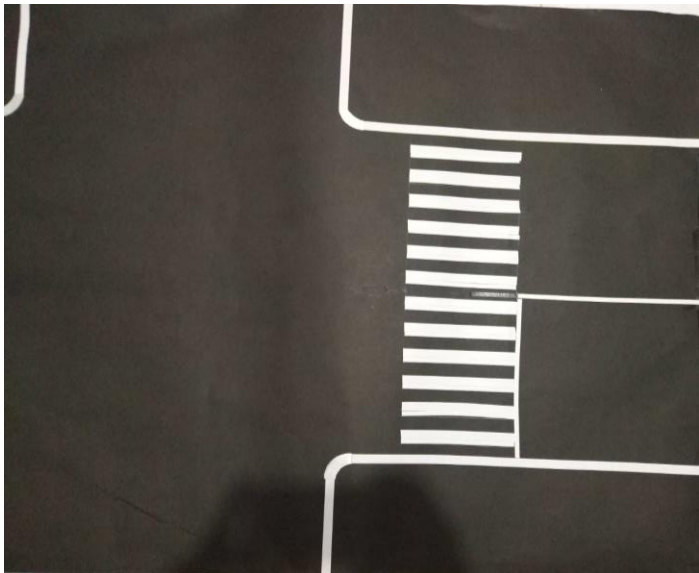
Ilustrasi kerja dari sistem dapat digambarkan pada gambar 3.2. Dari gambar tersebut, dapat dilihat Cara kerja dari masing-masing Blok. Kamera berfungsi sebagai sensor dengan mengambil video secara *real time* yang kemudian diproses oleh *Raspberry Pi* hingga diperoleh hasil pengenalan rambu-rambu lalu lintas yang Akan ditampilkan pada *display*.

3.3.1 Perancangan *Hardware*

Dalam perancangan *hardware* dibagi menjadi dua bagian yang meliputi perancangan desain fisik alat, dan perancangan tata letak kamera.

- Perancangan Desain Fisik Alat

Sistem didesain dengan menggunakan raspberry pi dengan disambungkan dengan menggunakan *webcam* Logitech C107. Kemudian alat tersebut di letakkan di dalam *case raspberry* dengan diletakkan diatas mobil prototipe. Hal tersebut dimaksudkan agar citra yang ditangkap kamera memiliki hasil yang Sama di setiap pengambilan data. Selain itu bidang uji didesain agar dapat diatur kemiringannya dengan tujuan supaya pendeteksian rambu-rambu lalu lintas yang dideteksi oleh sensor kamera pada raspberry dapat mengambil citra dengan baik. Untuk bidang uji sendiri memiliki ukuran 84 cm x 118 cm. Dengan lebar marka 22 cm x 84 cm, Desain fisik alat keseluruhan dapat dilihat pada gambar



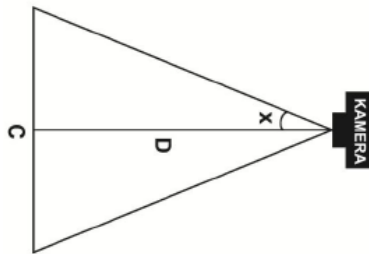
Gambar 3.3 Desain bidang uji



Gambar 3.4 Desain alat keseluruhan

- Perancangan Tata Letak Kamera

Perancangan tata letak kamera sangat penting dalam pengambilan citra, pada perancangan tata letak kamera ini harus mempertimbangkan posisi kamera. Posisi kamera didesain tegak lurus dengan bidang uji dan selalu mengikuti kemiringan bidang uji. Hal tersebut dimaksudkan agar citra dari obyek yang tertangkap kamera memiliki ukuran yang sama dengan obyek tersebut yang ada di bidang uji. Adapun untuk menghitung ketinggian kamera terhadap bidang uji harus dengan mempertimbangkan ukuran dari bidang uji agar seluruh obyek dapat tertangkap kamera. Bidang uji didesain dengan ukuran 84 cm x 118 cm, sehingga ukuran bidang pandang kamera harus melebihi bidang uji. Jika diasumsikan sudut lebar pandang kamera (α) adalah 15° maka ketinggian kamera (D) dapat dicari dengan menggunakan rumus *Pythagoras*. Untuk lebih jelasnya dapat dilihat pada gambar 3.5 di bawah ini.



Gambar 3.5 Desain perancangan tata letak kamera

3.4 Perancangan Jaringan Syaraf Tiruan *Back Propagation*

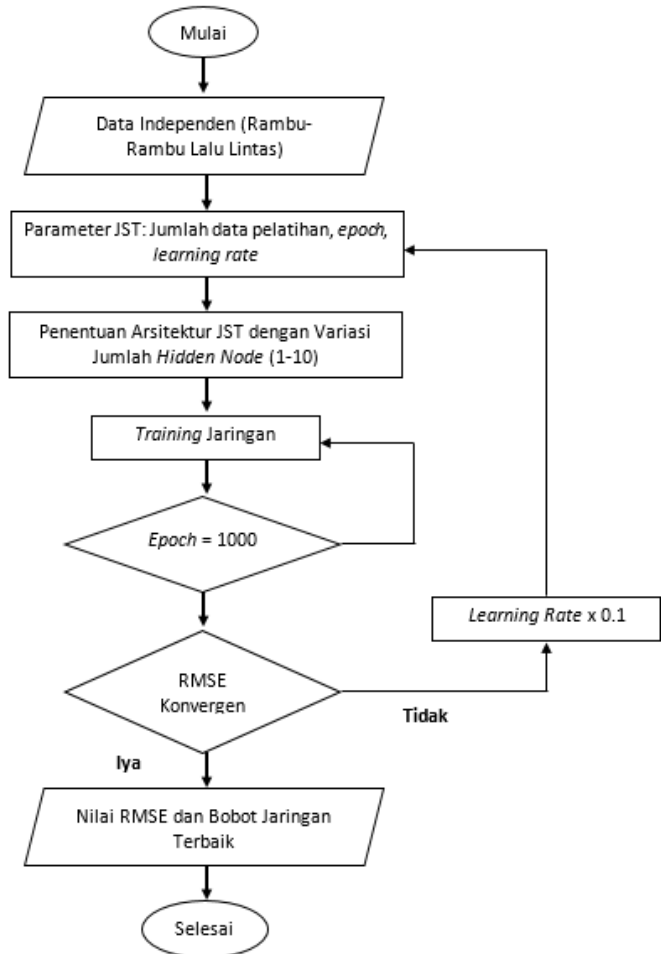
Perancangan algoritma *Back Propagation* Jaringan saraf tiruan ini dilakukan dengan bantuan perangkat lunak *OpenCv*. Pada percobaan ini digunakan metode Jaringan saraf tiruan *multilayer* dengan *back propagation algorithm*. Data-data yang telah dikumpulkan akan digunakan oleh algoritma ini untuk berlatih memahami pola pengenalan pada objek rambu-rambu lalu lintas, yang sangat berguna dalam menentukan tingkat akurasi pengenalan.

Pencarian model terbaik didapatkan melalui *training* yang dilakukan 10 kali percobaan untuk mendapatkan model terbaik. Variabel JST yang divariasikan adalah berupa jumlah *hidden node* (1-10 *node*) dan jumlah *epoch/iterasi* (1000 iterasi). Berikut merupakan hasil percobaan dengan variasi *learning rate* dan jumlah iterasi.

Tabel 3.1 Parameter Algoritma Jaringan Syaraf Tiruan

Parameter	Nilai
<i>Training</i>	<i>Back Propagation</i>
Jumlah Data Pelatihan	200
<i>Epoch</i>	1000
<i>Learning Rate</i>	0.001

Arsitektur JST terdiri dari 3 lapisan; *input layer*, *hidden layer*, dan *output layer*. *Input layer* merupakan masukan data independen (rambu-rambu lalu lintas). Sedangkan *output layer* berisi pengenalan pola rambu lalu lintas yang dikenali. Proses algoritma JST secara lengkapnya dapat dilihat diagram alir pada Gambar 3.6.



Gambar 3.6 Diagram alir algoritma jaringan syaraf tiruan

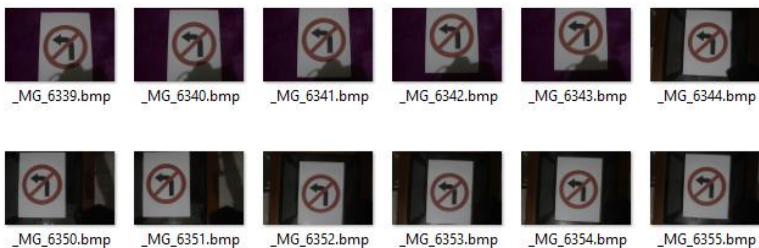
3.5 Proses Training

Dalam pembuatan sistem deteksi rambu lalu lintas, dengan menggunakan metode Jaringan Syaraf Tiruan yang perlu dilakukan adalah proses *training* yang akan dilakukan secara garis besar dibagi menjadi tiga yakni penambahan citra gambar positif dan negatif, kedua proses *sampling* objek secara spesifik, dan yang terakhir yakni proses *training*. Pada objek rambu lalu lintas yang awalnya format file nya berupa citra gambar setelah dilakukan *training* file berubah format menjadi *database* dengan format file .xml, dimana file output *training* ini akan diaplikasikan kedalam perangkat keras (*hardware*) dan perangkat lunak (*software*) yang akan mengartikan dari *database* yang telah dibuat.

Proses *training* yang akan dilakukan pada sistem rambu lalu lintas ini yakni menggunakan *OpenCV* Algoritma dan *library* program pada alat ini menggunakan referensi dari forum www.github.com yang ditulis oleh *Rem-nonsym* ^[14]. Dimana tahapan dari proses *training* dapat dilihat lebih jelas lagi pada tahapan dibawah ini.

- **Pembuatan Rawdata Dengan Menggunakan Citra Gambar Positif dan Negatif**

Untuk pembuatan raw data citra foto yang akan digunakan yakni dengan membuat beberapa gambar positif format .bmp dengan besar pixel 467 x 312 dan diletakkan di dalam folder positif/ *rawdata*.



Gambar 3.7 Rawdata dengan citra positif

Setelah itu secara default, terdapat 200 gambar negatif (latar belakang) di folder *negative* dalam hal ini digunakan 420 gambar dengan citra negatif. Juga dapat ditambahkan lebih banyak gambar di dalamnya dalam bentuk format apa pun, dengan catatan gambar yang ditambahkan citranya negatif.



Gambar 3.8 Rawdata dengan citra negatif

Setelah menambahkan sampel negatif, dilakukan penamaan pada sistem dengan menjalankan program '*create_list.bat*', hal ini bertujuan untuk memberikan penamaan pada *rawdata* negatif menjadi file *database*.

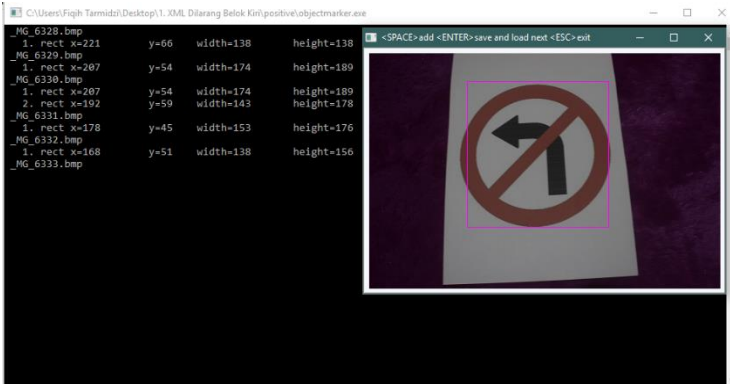
```
C:\Users\Fiqih Tarmidzi\Desktop\dasar_haartrain\negative\nama (1).jpg
C:\Users\Fiqih Tarmidzi\Desktop\dasar_haartrain\negative\nama (10).jpg
C:\Users\Fiqih Tarmidzi\Desktop\dasar_haartrain\negative\nama (100).jpg
C:\Users\Fiqih Tarmidzi\Desktop\dasar_haartrain\negative\nama (101).jpg
C:\Users\Fiqih Tarmidzi\Desktop\dasar_haartrain\negative\nama (102).jpg
```

Gambar 3.9 Penambahan citra negative pada file *database*

- **Proses *Sampling* Pada Objek Gambar Sebelum Dilakukan *Training***

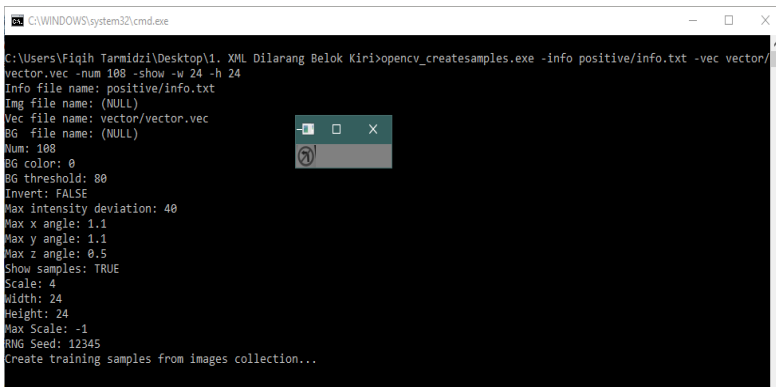
Pada proses ini proses yang akan dilakukan dengan menggunakan '*objectmarker.exe*' dengan memotong bagian dari gambar pada citra sampel positif. Pada program ini tombol spasi berfungsi untuk menambahkan area yang telah dipilih. Dan tombol enter untuk pindah ke gambar berikutnya. Jika terdapat kesalahan saat proses pemilihan area, maka dapat memilih area kembali tanpa menekan tombol spasi. Proses ini bertujuan untuk

memfokuskan objek deteksi yang akan menjadi ciri khas dalam rambu lalu lintas secara spesifik. Yang akan ditunjukkan pada gambar dibawah ini.



Gambar 3.10 Proses *sampling* pada objek rambu lalu lintas

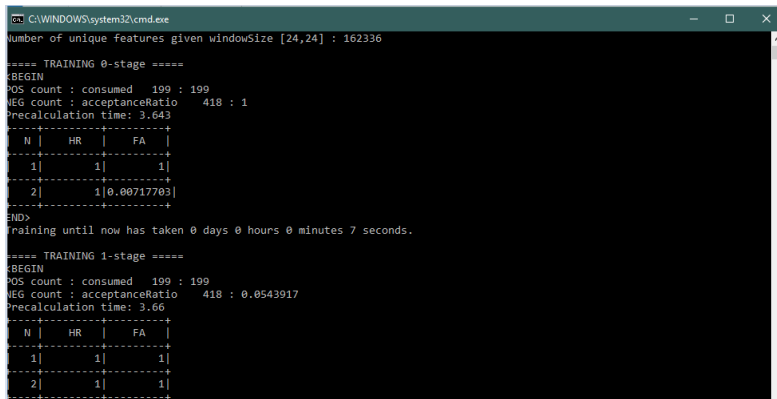
Setelah itu dijalankan program '*sample_cration.bat*' dalam hal ini akan membuat file vektor positif pada folder vektor, dan juga untuk melihat berapa banyak file yang dapat digunakan untuk dilanjutkan dalam proses terakhir yakni proses *training*. Yang akan ditunjukkan pada gambar dibawah ini.



Gambar 3.11 Hasil dari *sampling* dan konversi citra ke vektor

- **Proses *Training* Untuk Mendapatkan File .XML**

Dengan menjalankan proses terakhir yakni dengan menjalankan program '*haarTraining.bat*' dengan menyesuaikan jumlah file yang dapat digunakan untuk *training* hal ini dapat dilihat pada proses *sampling* yang menunjukkan jumlah gambar yang dapat digunakan untuk di *training* dan juga disesuaikan jumlah citra gambar negatif yang digunakan.



```

C:\WINDOWS\system32\cmd.exe
Number of unique features given windowSize [24,24] : 162336

==== TRAINING 0-stage ====
BEGIN
POS count : consumed 199 : 199
NEG count : acceptanceRatio 418 : 1
Precalculation time: 3.643
-----+-----
N | HR | FA |
-----+-----
1 | 1 | 1 |
-----+-----
2 | 1|0.00717703|
-----+-----
END>
training until now has taken 0 days 0 hours 0 minutes 7 seconds.

==== TRAINING 1-stage ====
BEGIN
POS count : consumed 199 : 199
NEG count : acceptanceRatio 418 : 0.0543917
Precalculation time: 3.66
-----+-----
N | HR | FA |
-----+-----
1 | 1 | 1 |
-----+-----
2 | 1 | 1 |
-----+-----

```

Gambar 3.12 Proses *training* yang dilakukan pada citra gambar

Setelah menunggu proses *training* selama beberapa menit atau bahkan beberapa jam untuk sistem menyelesaikan proses *training*. Dijalankan program terakhir '*convert.bat*' yang akan menghasilkan hasil terakhir yakni file output .XML, file inilah merupakan hasil akhir dari citra positif, negatif, dan beberapa bagian yang telah diseleksi secara spesifik. File ini merupakan file *database* yang akan diterjemahkan oleh *software* python untuk didapatkan sistem deteksi rambu lalu lintas.

3.6 Perakitan dan Pembuatan Sistem

Dalam pembuatan sistem, yang perlu dilakukan adalah perancangan pada perangkat keras (*hardware*) dan perangkat lunak (*software*) yang Akan dirancang. Pembuatan *hardware* yang Akan digunakan dalam implementasi alat ini meliputi: posisi kamera,

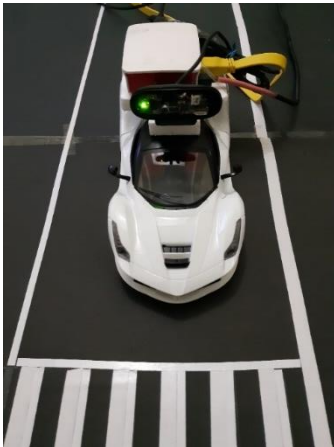
kemiringan derajat kamera, dan jarak bidang uji. Pembuatan *software* Akan dilakukan pada laptop atau Raspberry Pi yang sudah terinstall *software* python IDLE. Video *real time* yang ditangkap oleh kamera Akan diproses pada Raspberry Pi. Oleh karena itu, Raspberry Pi yang digunakan untuk mengolah citra harus diprogram terlebih dahulu agar dapat memproses data dari kamera. Setelah dilakukan pengolahan citra melalui mini komputer didapatkan hanya obyek yang berupa rambu lalu lintas yang terdeteksi, sehingga rambu lalu lintas dapat dikenali dengan tepat.

3.4.1 Pembuatan Hardware

Dalam pembuatan *hardware* dibagi menjadi dua bagian yaitu pembuatan bentuk fisik alat, dan tata letak kamera dan Raspberry Pi.

- Pembuatan Fisik Alat

Dalam pembuatan fisik alat yang harus diperhitungkan adalah Jarak kamera ke bidang uji, kemiringan kamera pada objek uji. Untuk dimensi prototype yang telah dibuat memiliki panjang 25 cm, lebar 11 cm, dan tinggi 7 cm.



(a)



(b)



(c)

Gambar 3.13 Perangkat keras keseluruhan (a) bentuk fisik keseluruhan alat, (b) *Raspberry Pi* (c) Kamera *webcam C107*

3.4.2 Pembuatan *Software*

Pada pembuatan tugas akhir ini, terdapat beberapa fungsi yang diperlukan dalam pembuatan sistem ini agar algoritma program mampu mengenali rambu-rambu lalu lintas dengan baik. Di antaranya adalah fungsi untuk mendapatkan variabel dari obyek, fungsi untuk mendeteksi obyek, fungsi untuk membandingkan objek dengan *library*, fungsi untuk mengenali sebuah objek dan fungsi utama.

- **Fungsi untuk menangkap objek rambu lalu lintas**

Dalam pembuatan sistem deteksi objek sangat diperlukan pembuatan penangkapan objek agar objek dapat dideteksi dengan sempurna. Berikut adalah *script* program fungsi yang digunakan untuk mendapatkan obyek yang diinginkan:

```

video_capture = cv2.VideoCapture(0)

while True:
    # Menangkap frame-by-frame
    ret, frame = video_capture.read()

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    aseribu = aCascade.detectMultiScale(gray, 1.3, 5)
    bseribu = bCascade.detectMultiScale(gray, 1.3, 5)
    asepuhrib = cCascade.detectMultiScale(gray, 1.6, 5)
    bsepuhrib = dCascade.detectMultiScale(gray, 1.6, 5)
    alimapuhrib = eCascade.detectMultiScale(gray, 1.3, 5)
    blimapuhrib = fCascade.detectMultiScale(gray, 1.3, 5)
    aduaribu = gCascade.detectMultiScale(gray, 1.3, 5)
    bduaribu = hCascade.detectMultiScale(gray, 1.2, 5)
    aseratusrb = iCascade.detectMultiScale(gray, 1.1, 5)
    bseratusrb = jCascade.detectMultiScale(gray, 1.1, 5)
    alimaribu = kCascade.detectMultiScale(gray, 1.1, 5)
    blimaribu = lCascade.detectMultiScale(gray, 1.1, 5)
    aduapuhrib = mCascade.detectMultiScale(gray, 1.2, 5)
    bduapuhrib = nCascade.detectMultiScale(gray, 1.2, 5)

```

- **Fungsi untuk mendeteksi dan mengartikan hasil file *training* rambu lalu lintas**

Setelah menangkap objek tahap selanjutnya adalah mendeteksi objek. Pendeteksian objek yang digunakan adalah dengan mengartikan program *training* yang telah dilakukan. Yakni berupa format file .XML. file ini berfungsi untuk memanggil *database* yang telah dibuat, yakni ketika objek rambu lalu lintas dideteksi oleh sensor kamera *webcam C107* dengan begitu maka dapat ditampilkan hasil dari deteksi. Berikut adalah *script* program fungsi yang telah dibuat untuk menerjemahkan hasil file *training* yang digunakan untuk mendeteksi obyek:

```

aCascade = cv2.CascadeClassifier('1.xml')
bCascade = cv2.CascadeClassifier('2.xml')
cCascade = cv2.CascadeClassifier('3.xml')
dCascade = cv2.CascadeClassifier('4.2.xml')
eCascade = cv2.CascadeClassifier('5.xml')
fCascade = cv2.CascadeClassifier('6.xml')
gCascade = cv2.CascadeClassifier('7.xml')
hCascade = cv2.CascadeClassifier('8.xml')
iCascade = cv2.CascadeClassifier('9.xml')
jCascade = cv2.CascadeClassifier('10.xml')
kCascade = cv2.CascadeClassifier('11.xml')
lCascade = cv2.CascadeClassifier('12.xml')
mCascade = cv2.CascadeClassifier('13.xml')
nCascade = cv2.CascadeClassifier('14.xml')

```

- **Fungsi untuk utama**

Fungsi utama ini adalah fungsi inti dari seluruh fungsi karena sebagai tempat untuk memanggil fungsi sebelumnya dan membuat bingkai pada objek Berikut adalah *script* program fungsi utama yakni:

```

# Menampilkan dan menggambar kotak penanda pengenalan rambu lalu lintas
for (x, y, w, h) in aseribu:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Dilarang Belok kiri',(x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in bseribu:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Dilarang Berhenti',(x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in asepuhurb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Dilarang Masuk',(x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in bsepuhurb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Dilarang Putar Parkir',(x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in alimapulurb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Dilarang Putar Balik',(x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

```

```

for (x, y, w, h) in blimapuluhrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Dilarang Belok Kanan', (x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in aduaribu:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Penyebrangan', (x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in bduaribu:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Perempatan', (x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in aseratusrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Lewat Sini', (x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in bseratusrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Lewat Kanan', (x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in alimaribu:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'STOP', (x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in blimaribu:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Jalur Sepeda', (x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in aduapuluhrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Tempat Parkir', (x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in bduapuluhrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Hati-Hati', (x,y), font, 1.5, (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

# Menampilkan Frame Display
cv2.imshow('Video', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):
    break

```

3.7 Pengujian Alat

Untuk mendapatkan spesifikasi dari sistem pendeteksi rambu lalu lintas maka dilakukan pengujian yang meliputi:

- Pengujian *software*

Pengujian ini dilakukan untuk menguji algoritma program apakah program dapat mendeteksi dan mengenali obyek rambu

lalu lintas yang telah dilakukan *training*. Pengujian ini dilakukan dengan menggunakan rambu lalu lintas dengan skala 1:16.

- Pengaruh kemiringan kamera

Pengujian ini dilakukan untuk mendapatkan hasil yang terbaik pada kemiringan bidang uji. Kemiringan yang digunakan yakni pada kemiringan 15° .

- Keakurasian dalam pengenalan rambu lalu lintas

Pengujian ini ditujukan untuk menguji keakurasian sistem pendeteksi rambu lalu lintas dalam mengenali rambu lalu lintas. Pengujian ini menggunakan kemiringan bidang uji yang sebelumnya sudah diuji, yakni rambu-rambu yang telah disebutkan diatas. Dari pengujian tersebut dapat dilihat besar *error* dari masing-masing rambu lalu lintas yang diuji.

3.8 Analisa Data dan Pembahasan

Analisa data dilakukan terhadap hasil-hasil pengukuran yang diperoleh. Dari hasil tersebut, Akan dihitung berapa persen tingkat keberhasilan dan keakuratan dari sistem deteksi sehingga didapatkan spesifikasi dan kemampuan sistem *image processing* untuk mengenali rambu lalu lintas.

3.9 Penulisan Laporan

Tahap akhir dari penelitian ini adalah penyusunan laporan. Laporan tersebut merupakan rekaman dan bentuk pertanggung jawaban dari berbagai kegiatan yang dilakukan selama proses penelitian kepada pihak terkait. Setelah itu hasilnya Akan disampaikan baik secara tulisan dan lisan.

Halaman Ini Sengaja Dikosongkan

BAB IV

HASIL DAN PEMBAHASAN

Pada Bab ini, akan dijelaskan mengenai analisa data pada setiap objek rambu lalu lintas yang telah dilakukan pengujian. Pengujian yang telah dilakukan yakni pengenalan objek rambu-rambu lalu lintas pada jarak tertentu, pengenalan objek rambu-rambu lalu lintas pada kecepatan tertentu.

4.1 Pengujian Perangkat Lunak (*Software*)

Pengujian ini bertujuan untuk menguji dan mengetahui perangkat lunak (*software*) yang telah dibangun algoritmanya apakah dapat berjalan dengan baik dan semestinya dalam pendeteksian dan pengenalan objek rambu-rambu lalu lintas. Jumlah rambu-rambu lalu lintas yang digunakan pada pengujian sebanyak 14 rambu lalu lintas. Pengujian dilakukan sebanyak 20 kali untuk pengujian pengenalan rambu-rambu lalu lintas dengan berbagai variasi jarak, dengan jarak yang diujikan 50 cm, 100 cm, 150 cm, 200 cm. Berdasarkan pada pengujian yang dilakukan pada variasi jarak didapatkan rata-rata *error* sebesar 0% seperti yang ditunjukkan pada tabel 4.1. Dengan hasil tersebut dapat dinyatakan bahwa program berhasil dalam mengenali objek rambu-rambu lalu lintas dengan baik. Untuk lebih jelasnya pada pengujian variasi jarak dapat dilihat dibawah ini, yakni sebagai berikut.







4.1.1 Pengujian Rambu-rambu Lalu Lintas Variasi Jarak

Pengujian pengenalan rambu-rambu lalu lintas ini bertujuan mengukur jarak minimal dan maksimal antara sensor yang dipasang pada mobil prototype dengan objek rambu lalu lintas. Dimana pada proses ini jarak minimal dan maksimal didapat antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Hal tersebut dilakukan sebanyak 10 kali untuk dilakukan pengujian pada tiap jenis rambu lalu lintas, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.

Tabel 4.1 Pengujian Objek Rambu Lalu Lintas dengan Menggunakan Variasi Jarak

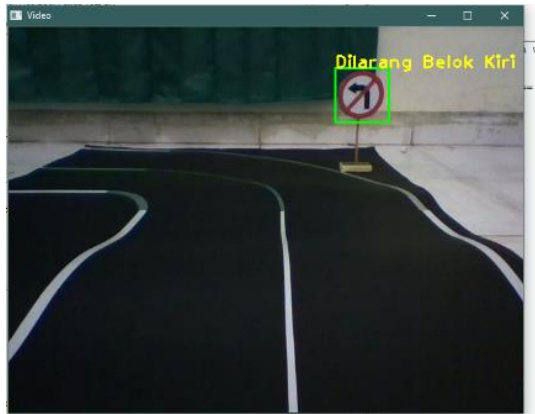
Jenis Rambu Lalu Lintas	Variasi Jarak (Cm)	Jumlah Percobaan	Keberhasilan (%)
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100

Lanjutan Tabel 4.2 Pengujian Objek Rambu Lalu Lintas dengan Menggunakan Variasi Jarak

Jenis Rambu Lalu Lintas	Variasi Jarak (Cm)	Jumlah Percobaan	Keberhasilan (%)
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100
	50	10 Kali	100
	100	10 Kali	100
	150	10 Kali	100
	200	10 Kali	100

- Rambu Lalu Lintas Dilarang Belok Kiri

Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas dilarang belok, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.1 Model rambu lalu lintas dilarang belok kiri

- Rambu Lalu Lintas Dilarang Berhenti

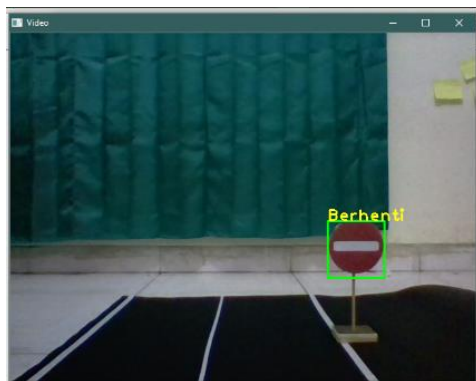
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas dilarang berhenti, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.2 Model rambu lalu lintas dilarang Berhenti

- Rambu Lalu Lintas Berhenti

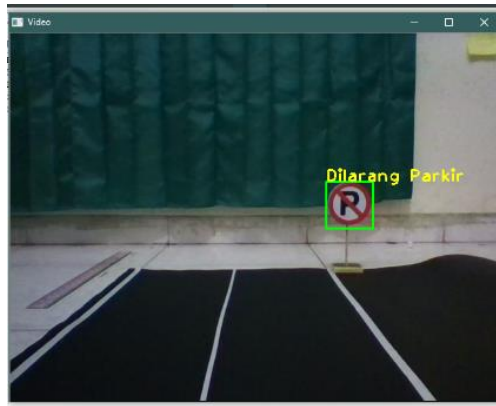
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas berhenti, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.3 Model rambu lalu lintas dilarang berhenti

- **Rambu Lalu Lintas Dilarang Parkir**

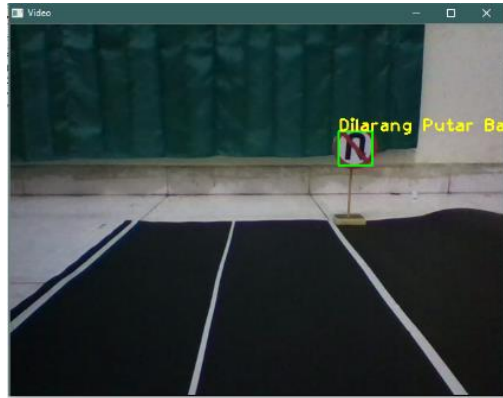
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas dilarang parkir, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.4 Model rambu lalu lintas dilarang parkir

- **Rambu Lalu Lintas Dilarang Putar Balik**

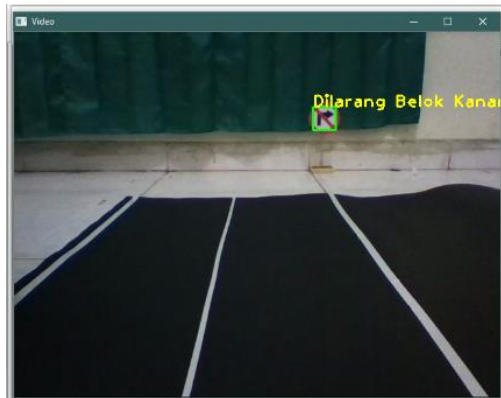
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas dilarang putar balik, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.5 Model rambu lalu lintas dilarang putar balik

- Rambu Lalu Lintas Dilarang Belok Kanan

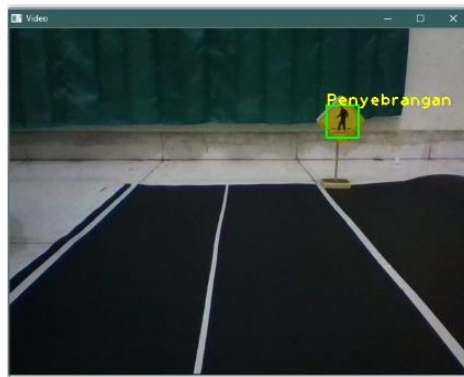
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas dilarang belok kanan, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.6 Model rambu lalu lintas dilarang belok kanan

- **Rambu Lalu Lintas Tempat Penyebrangan**

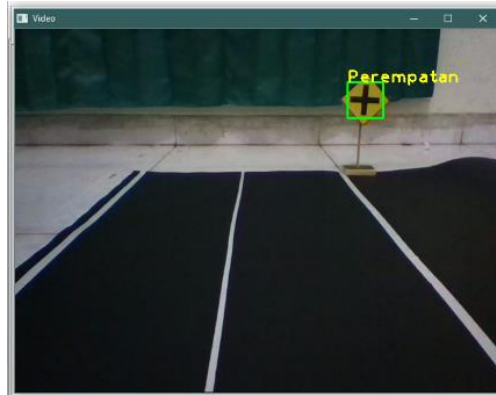
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas tempat penyebrangan, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.7 Model rambu lalu lintas tempat penyebrangan

- **Rambu Lalu Lintas Perempatan**

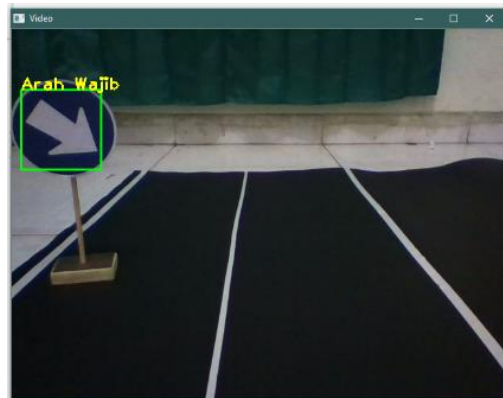
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas perempatan, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.8 Model rambu lalu lintas perempatan

- Rambu Lalu Lintas Arah Wajib Diambil

Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas arah wajib diambil, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.9 Model rambu lalu lintas arah wajib diambil

- **Rambu Lalu Lintas Wajib Arah Kanan**

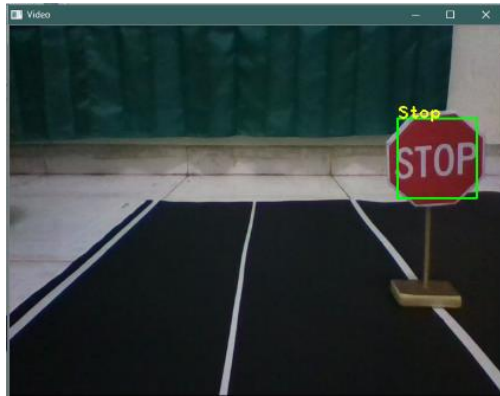
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas wajib arah kanan, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.10 Model rambu lalu lintas arah wajib kanan

- **Rambu Lalu Lintas *Stop***

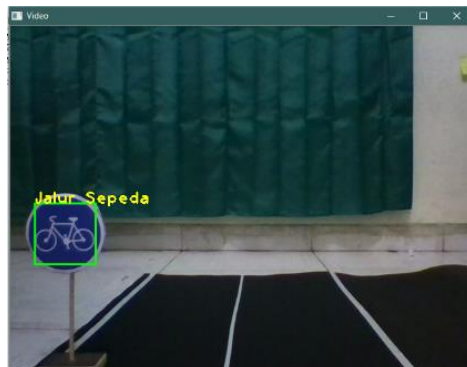
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas *stop*, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.11 Model rambu lalu lintas *stop*

- Rambu Lalu Lintas Jalur Sepeda

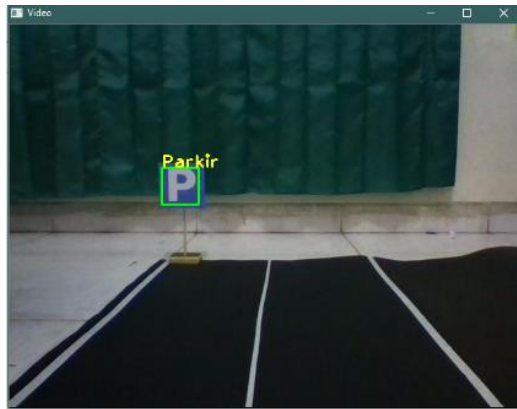
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas jalur sepeda, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.12 Model rambu lalu lintas jalur sepeda

- **Rambu Lalu Lintas Tempat Parkir**

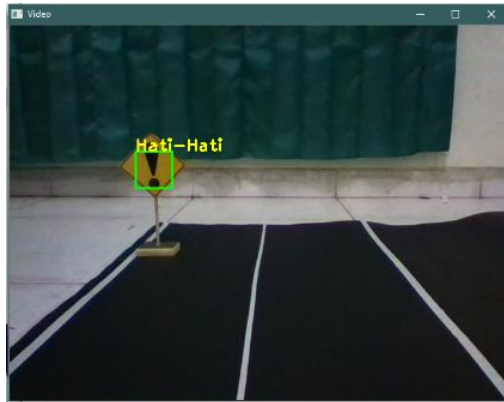
Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas tempat parkir, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.



Gambar 4.13 Model rambu lalu lintas tempat parkir

- **Rambu Lalu Lintas Hati-hati**

Pada rambu lalu lintas ini jarak minimal dan maksimal yang di berikan yakni antara 50 Cm, 100 Cm, 150 Cm, dan 200 Cm dari sensor kamera. Dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas hati-hati, kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.




Gambar 4.14 Model rambu lalu lintas hati-hati

4.1.2 Pengujian Rambu-rambu Lalu Lintas Variasi Kecepatan

Variasi kecepatan diberikan bertujuan untuk mengetahui apakah sistem deteksi rambu lalu lintas masih mengenali objek. Jika diberikan variasi kecepatan pada saat pengukuran secara *realtime*. Dimana pada proses ini diberikan variasi kecepatan pada mobil yakni pada mobil prototype diberikan 4 buah baterai tipe AA dengan 1.2 Volt kecepatan yang dicapai menyentuh nilai 157.14 cm/s. Variasi kecepatan kedua diberikan 3 buah baterai tipe AA kecepatannya mencapai 110 cm/s. Variasi kecepatan ketiga diberikan 2 buah baterai tipe AA yang dipasang secara seri dengan kecepatannya mencapai 55 cm/s. Dan yang terakhir diberikan 1 buah baterai dengan kecepatan yang dicapai sebesar 44 cm/s. Hal tersebut dilakukan sebanyak 10 kali untuk dilakukan pengujian pada beberapa jenis rambu tertentu saja, tidak semua rambu seperti yang dilakukan pada pengujian variasi oleh jarak dan juga dihitung banyaknya keberhasilan dan kegagalan dari sistem. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu.

- Variasi Kecepatan 157.14 cm/s, 110 cm/s, 55 cm/s, 44 cm/s
Adapun pada variasi kecepatan 1, 2, 3, dan 4 maka, dilakukan pengujian pada rambu lalu lintas yang ada di depan mobil secara langsung, atau rambu lalu lintas yang menghadap di depan sensor kamera, pengujian dilakukan sebanyak 10 kali untuk jenis rambu lalu lintas dilarang belok kiri, berhenti, dilarang putar balik, dilarang belok kanan, wajib ambil arah disini, wajib ambil arah kanan, dan kemudian dihitung banyaknya keberhasilan dan kegagalan dari sistem pengenalan. Berikut adalah hasil pengujian dan tabel yang dihasilkan dari proses estimasi jarak pengenalan rambu dibawah ini.

Tabel 4.3 Pengujian Objek Rambu Lalu Lintas dengan Menggunakan Variasi Kecepatan

Jenis Rambu Lalu Lintas	Variasi Kecepatan (Cm/sec)	Jumlah Percobaan	Keberhasilan (%)
	157.14	10 Kali	80
	110	10 Kali	90
	55	10 Kali	100
	44	10 Kali	100
	157.14	10 Kali	40
	110	10 Kali	60
	55	10 Kali	70
	44	10 Kali	80
	157.14	10 Kali	80
	110	10 Kali	90
	55	10 Kali	100
	44	10 Kali	100
	157.14	10 Kali	90
	110	10 Kali	100
	55	10 Kali	100
	44	10 Kali	100

Lanjutan Tabel 4.4 Pengujian Objek Rambu Lalu Lintas dengan Menggunakan

Jenis Rambu Lalu Lintas	Variasi Kecepatan (Cm/sec)	Jumlah Percobaan	Keberhasilan (%)
	157.14	10 Kali	80
	110	10 Kali	90
	55	10 Kali	100
	44	10 Kali	100
	157.14	10 Kali	80
	110	10 Kali	90
	55	10 Kali	100
	44	10 Kali	100

❖ Rambu Lalu Lintas Dilarang Belok Kiri



Gambar 4.15 Model rambu lalu lintas dilarang belok kiri pada pengujian variasi kecepatan

❖ Rambu Lalu Lintas Dilarang Berhenti



Gambar 4.16 Model rambu lalu lintas dilarang berhenti pada pengujian variasi kecepatan

❖ Rambu Lalu Lintas Dilarang Putar Balik



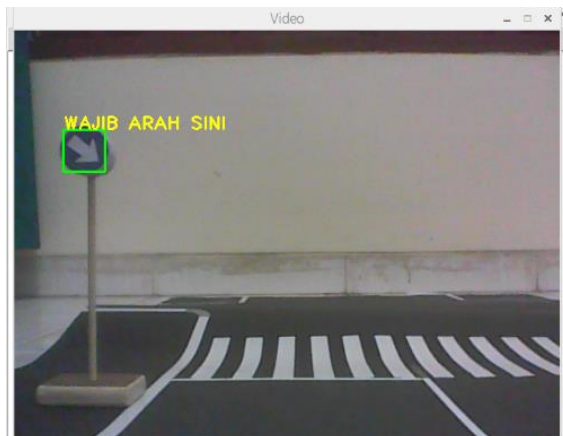
Gambar 4.17 Model rambu lalu lintas dilarang putar balik pada pengujian variasi kecepatan

❖ Rambu Lalu Lintas Dilarang Belok Kanan



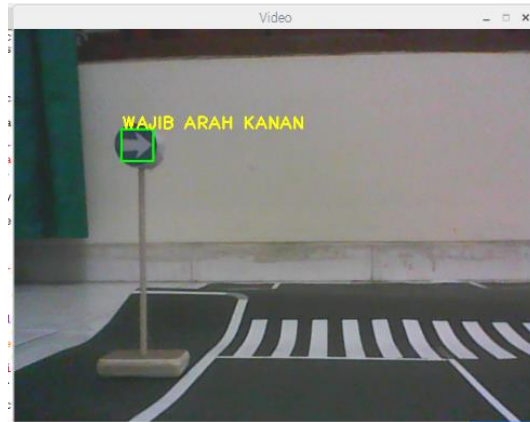
Gambar 4.18 Model rambu lalu lintas dilarang belok kanan pada pengujian variasi kecepatan

❖ Rambu Lalu Lintas Wajib Arah Disini



Gambar 4.19 Model rambu lalu lintas wajib ambil disini pada pengujian variasi kecepatan

❖ Rambu Lalu Lintas Wajib Arah Kanan



Gambar 4.20 Model rambu lalu lintas wajib arah kanan pada pengujian variasi kecepatan

4.2 Pembahasan

Proses pengambilan data pada variasi jarak dengan asumsi bahwa kondisi kamera yang terpasang pada mobil tegak lurus dengan posisi rambu dan batas kemiringan kurang dari 15 derajat. Posisi rambu lalu lintas juga statis atau tidak bergerak. Berdasarkan tabel diatas dapat dianalisa sistem pengenalan rambu lalu lintas yang telah dilakukan mempunyai keakuratan <88.60% sehingga sistem pengenalan rambu lalu lintas menggunakan Jaringan Syaraf Tiruan berhasil dan baik dalam sistem deteksi rambu lalu lintasnya. Sedangkan pada proses pengambilan data diatas dengan kondisi kamera yang terpasang pada mobil tegak lurus dengan posisi rambu dan batas kemiringan kurang dari 15 derajat. Posisi rambu lalu lintas juga statis atau tidak bergerak. Berdasarkan tabel diatas dapat dianalisa sistem pengenalan rambu lalu lintas yang telah dilakukan mempunyai keakuratan <80% sehingga sistem pengenalan rambu lalu lintas menggunakan Jaringan Syaraf Tiruan berhasil dan baik dalam sistem deteksi rambu lalu lintasnya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan percobaan pada tugas akhir ini, maka dapat disimpulkan bahwa:

1. Sistem pengenalan rambu-rambu lalu lintas mampu mendeteksi dan mengenali objek rambu lalu lintas dengan baik
2. Alat mampu mendeteksi dan mengenali rambu-rambu lalu lintas dengan baik. Dengan nilai *error* pada percobaan variasi jarak sebesar 0 %.
3. Alat mampu mendeteksi dan mengenali objek rambu-rambu lalu lintas bahkan ketika diberikan variasi kecepatan pada saat pendeteksian dengan nilai persentasi keberhasilan sistem mengenali objek mencapai 80 %.
4. Jaringan Syaraf Tiruan berhasil diterapkan pada sistem pengenalan rambu-rambu lalu lintas.

5.2 Saran

Adapun saran untuk penelitian selanjutnya agar lebih baik adalah Dalam penelitian tugas akhir ini, terdapat beberapa hal yang perlu diperbaiki. Oleh karena itu, validasi dan penelitian lebih lanjut sangat diperlukan. Saran yang dapat diberikan penulis terkait dengan pengembangan penelitian ini adalah sebagai berikut

1. Skala yang digunakan pada penelitian berikutnya menggunakan skala yang sebenarnya.
2. Untuk kedepannya bisa digunakan mikrokontroler selain *raspberry pi* sehingga dapat dibandingkan antara hasil dari deteksi menggunakan *raspberry pi* dengan mikrokontroler yang lainnya.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] M. D. Hasan Fleyeh., "Road and traffic sign detection and recognition," *Department of Computer Engineering, Dalarna University, Sweden.*, pp. 645-652, 2014.
- [2] Ihsan, "Rambu Lalu Lintas – Penjelasan dan Arti dari Gambar Rambu-rambu Jalan," *Ceritaihsan*, [Online]. Available: www.ceritaihsan.com/rambu-lalu-lintas/.html. [Accessed 21 Januari 2018].
- [3] A. R. W. D. H. E. P. W. Muhamad Afif Fanno, "Sistem Pengenalan Rambu Lalu Lintas menggunakan Raspberry Pi," *STMIK GI MDP*, 2017.
- [4] A. K. d. A. Susanto, *Teori dan Aplikasi Pengolahan Citra*, Jogjakarta: Andi Offset, 2013.
- [5] R. H. J. Nakatsu, *Entertainment Computing Technologies and Application*, Nishihara: IFIP International Federation for Information Processing, 2003.
- [6] M. M. M. N. Davide A. Migliore, "A revaluation of frame difference in fast and robust motion detection," *VSSN '06 Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pp. 215-218 , 2006.
- [7] A. R. S. M. d. I. A. K. M. Firsan Noviza Duanto, "Perancangan Image Processing System Berbasis Raspberry Pi untuk Menghitung Benih Ikan secara Otomatis dan Real-Time," *POMITS*, 2015.
- [8] J. J. Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB, ANDI*, 2009.
- [9] E. O. Sheldon Waite, "FPGA-Based Traffic Sign Recognition for Advanced Driver Assistance Systems.," *Journal of Transportation Technologies*, 2013, 3, 1-16, pp. 1-16, 2013.
- [10] Rem-nonsym, "Dasaradh Makes," 30 September 2016. [Online]. Available: http://www.mediafire.com/file/1aq02tpidk105fv/dasar_haartrain.rar. [Accessed 10 Juli 2018].

- [11] K. V. I. E. V. S. D. M. Filatov., "Neural Network System of Traffic Signs Recognition," *Saint Petersburg Electrotechnical University "LETI"*, 2017.
- [12] H. F. E. R. W. Thomas Oddy Chrisdwianto., "Perancangan sistem deteksi dan pengenalan rambu peringatan menggunakan metode template matching," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 1265-1274, 2018.
- [13] M. E. H. Saber Elotmani., "A light traffic signs recognition system," *Labo LETAS, FS, university*, 2014.
- [14] H. M. R. E. B. a. M. H. Nadra Ben Romdhane., "Combined 2d/3d traffic signs recognition and distance estimation," *Intelligent Vehicles Symposium (IV)*, pp. 19-22, 2016.
- [15] R. T. R. B. L. V. G. Markus Mathias., "Traffic sign recognition - how far are we from the solution?," *Computer Vision Lab, D-ITET, ETH Zurich*, 2011.
- [16] Y. H. Y. C. H. B. Kwangyong Lim., "Real-time traffic sign recognition based on a general purpose GPU and deep-learning," *PLoS ONE*, p. 12 (3), 2017.
- [17] M. R. D. S. Ericks Rachmat Swedia., "Aplikasi Pendeteksi Rambu Lalu-Lintas Menggunakan Operator Sobel dan Metode Hamming," *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*, p. 1907 – 5022, 2017.
- [18] I. S. David Soendoro, "Traffic sign recognition with color-based method, shape-arc estimation and SVM," *International Conference on Electrical Engineering and Informatics*, pp. j2-2, 2011.

LAMPIRAN

SCRIPT SISTEM PROGRAM PENGENALAN RAMBU LALU LINTAS

Program Utama

```

import cv2
import sys

x = 0
y = 0
faceCascade = cv2.CascadeClassifier('filenya.xml')

video_capture = cv2.VideoCapture(0)

while True:
    # Capture frame-by-frame
    ret, frame = video_capture.read()

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray
    )

    # Draw a rectangle around the faces
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    if len(faces)>0:
        cv2.putText(frame, "BELOK KIRI", (x,y),
            cv2.FONT_HERSHEY_PLAIN, 1.5,
            (11,255,255),thickness=2)
    else:

```

```

cv2.putText(frame, "", (x,y),
            cv2.FONT_HERSHEY_PLAIN, 1.5,
            (11,255,255),thickness=2)
# Display the resulting frame
cv2.imshow('Video', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# When everything is done, release the capture
video_capture.release()
cv2.destroyAllWindows()

```

Program Secara Keseluruhan

```

import cv2
import sys

aCascade = cv2.CascadeClassifier('1.xml')
bCascade = cv2.CascadeClassifier('2.xml')
cCascade = cv2.CascadeClassifier('3.xml')
dCascade = cv2.CascadeClassifier('4.2.xml')
eCascade = cv2.CascadeClassifier('5.xml')
fCascade = cv2.CascadeClassifier('6.xml')
gCascade = cv2.CascadeClassifier('7.xml')
hCascade = cv2.CascadeClassifier('8.xml')
iCascade = cv2.CascadeClassifier('9.xml')
jCascade = cv2.CascadeClassifier('10.xml')
kCascade = cv2.CascadeClassifier('11.xml')
lCascade = cv2.CascadeClassifier('12.xml')
mCascade = cv2.CascadeClassifier('13.xml')
nCascade = cv2.CascadeClassifier('14.xml')

video_capture = cv2.VideoCapture(0)

```

while True:

Menangkap frame-by-frame

ret, frame = video_capture.read()

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

asribu = aCascade.detectMultiScale(gray, 1.3, 5)

bseribu = bCascade.detectMultiScale(gray, 1.3, 5)

asepulhrb = cCascade.detectMultiScale(gray, 1.6, 5)

bsepulhrb = dCascade.detectMultiScale(gray, 1.6, 5)

alimapulhrb = eCascade.detectMultiScale(gray, 1.3, 5)

blimapulhrb = fCascade.detectMultiScale(gray, 1.3, 5)

aduaribu = gCascade.detectMultiScale(gray, 1.3, 5)

bduaribu = hCascade.detectMultiScale(gray, 1.2, 5)

aseratusrb = iCascade.detectMultiScale(gray, 1.1, 5)

bseratusrb = jCascade.detectMultiScale(gray, 1.1, 5)

alimaribu = kCascade.detectMultiScale(gray, 1.1, 5)

blimaribu = lCascade.detectMultiScale(gray, 1.1, 5)

aduapulhrb = mCascade.detectMultiScale(gray, 1.2, 5)

bduapulhrb = nCascade.detectMultiScale(gray, 1.2, 5)

Menampilkan dan menggambar kotak penanda pengenalan rambu lalu lintas

for (x, y, w, h) in asribu:

font = cv2.FONT_HERSHEY_PLAIN

cv2.putText(frame, 'Dilarang Belok kiri', (x, y), font, 1.5, (11, 255, 255), thickness=2)

cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in bseribu:

font = cv2.FONT_HERSHEY_PLAIN

cv2.putText(frame, 'Dilarang Berhenti', (x, y), font, 1.5, (11, 255, 255), thickness=2)

cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

```

for (x, y, w, h) in asepuuluhrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Dilarang Masuk',(x,y), font, 1.5,
        (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in bsepuuluhrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Dilarang Putar Parkir',(x,y), font, 1.5,
        (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in alimapuluhrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Dilarang Putar Balik ',(x,y), font, 1.5,
        (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in blimapuluhrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Dilarang Belok Kanan',(x,y), font, 1.5,
        (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in aduaribu:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Penyebrangan',(x,y), font, 1.5,
        (11,255,255),thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

for (x, y, w, h) in bduaribu:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame,'Perempatan',(x,y), font, 1.5,
        (11,255,255),thickness=2)

```



```
cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

```
for (x, y, w, h) in aseratusrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Lewat Sini', (x,y), font, 1.5,
                (11,255,255), thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

```
for (x, y, w, h) in bseratusrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Lewat Kanan', (x,y), font, 1.5,
                (11,255,255), thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

```
for (x, y, w, h) in alimaribu:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'STOP', (x,y), font, 1.5,
                (11,255,255), thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

```
for (x, y, w, h) in blimaribu:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Jalur Sepeda', (x,y), font, 1.5,
                (11,255,255), thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

```
for (x, y, w, h) in aduapuluhrb:
    font = cv2.FONT_HERSHEY_PLAIN
    cv2.putText(frame, 'Tempat Parkir', (x,y), font, 1.5,
                (11,255,255), thickness=2)
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

```
for (x, y, w, h) in bduapuluhrb:
    font = cv2.FONT_HERSHEY_PLAIN
```

```
cv2.putText(frame, 'Hati-Hati', (x, y), font, 1.5,  
            (11, 255, 255), thickness=2)  
cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

```
# Menampilkan Frame Display
```

```
cv2.imshow('Video', frame)
```

```
if cv2.waitKey(1) & 0xFF == ord('q'):  
    break
```

```
# Akhir dari Program, Close dan berakhir
```

```
video_capture.release()
```

```
cv2.destroyAllWindows()
```

LAMPIRAN
TABEL HASIL PERCOBAAN VARIASI JARAK

- Tabel Hasil Percobaan Rambu Lalu Lintas Dilarang Belok Kiri

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian dengan objek rambu lalu lintas Dilarang Berhenti

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian yang Dilakukan dengan Objek Rambu Lalu Lintas Dilarang Masuk

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian yang Dilakukan dengan Objek Rambu Lalu Lintas Dilarang Parkir

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian yang Dilakukan dengan Objek Rambu Lalu Lintas Dilarang Putar Balik

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian dengan Objek Rambu Lalu Lintas Dilarang Belok Kanan

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian yang Dilakukan dengan Objek Rambu Lalu Lintas Tempat Penyebrangan

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian yang Dilakukan dengan Objek Rambu Lalu Lintas Perempatan

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian yang Dilakukan dengan Objek Rambu Lalu Lintas Arah Wajib Diambil

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian yang Dilakukan dengan Objek Rambu Lalu Lintas Wajib Arah Kanan

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian yang dilakukan dengan objek rambu lalu lintas *Stop*

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian yang Dilakukan dengan Objek Rambu Lalu Lintas Jalur Sepeda

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

- Tabel Pengujian dengan Objek Rambu Lalu Lintas Tempat Parkir

Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	


- Tabel Pengujian yang Dilakukan dengan Objek Rambu Lalu Lintas Hati-hati


Variasi Jarak	Percobaan Ke	Hasil	Error (%)	Rata-rata (%)
50 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
100 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	

150 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	
200 Cm	1	Berhasil	0	0
	2	Berhasil	0	
	3	Berhasil	0	
	4	Berhasil	0	
	5	Berhasil	0	
	6	Berhasil	0	
	7	Berhasil	0	
	8	Berhasil	0	
	9	Berhasil	0	
	10	Berhasil	0	


LAMPIRAN TABEL HASIL PERCOBAAN VARIASI KECEPATAN

- Tabel Pengujian dengan Variasi Kecepatan 1, 2, 3, dan 4 pada Objek Rambu Lalu Lintas Dilarang Belok Kiri

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 1 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	157.14 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Tidak
	Keenam		Berhasil
	Ketujuh		Tidak
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			80%

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 2 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	110 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Tidak
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			90%

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 3 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	55 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 4 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	44 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%

- Tabel Pengujian dengan Variasi Kecepatan 1, 2, 3, dan 4 pada Objek Rambu Lalu Lintas Dilarang Masuk


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 1 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	157.14 cm/s	Tidak
	Kedua		Tidak
	Ketiga		Tidak
	Keempat		Berhasil
	Kelima		Tidak
	Keenam		Berhasil
	Ketujuh		Tidak
	Kedelapan		Berhasil
	Kesembilan		Tidak
	Kesepuluh		Berhasil
Rata-rata Keakuratan			40%

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 2 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	110 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Tidak
	Keenam		Berhasil
	Ketujuh		Tidak
	Kedelapan		Tidak
	Kesembilan		Tidak
	Kesepuluh		Berhasil
Rata-rata Keakuratan			60%


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 3 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	55 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Tidak
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Tidak
	Kesembilan		Tidak
	Kesepuluh		Berhasil
Rata-rata Keakuratan			70%

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 4 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	44 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Tidak
	Keenam		Berhasil
	Ketujuh		Tidak
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			80%

- Tabel Pengujian dengan Variasi Kecepatan 1,2,3, dan 4 pada Objek Rambu Lalu Lintas Dilarang Putar Balik


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 1 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	157.14 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Tidak
	Kesepuluh		Tidak
Rata-rata Keakuratan			80%

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 2 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	110 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Tidak
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			90%


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 3 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	55 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 4 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	44 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%

- Tabel Pengujian dengan Variasi Kecepatan 1 pada Objek Rambu Lalu Lintas Dilarang Belok Kanan


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 1 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	157.14 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Tidak
Rata-rata Keakuratan			90%


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 2 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	110 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 3 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	55 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 4 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	44 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%

- Tabel Pengujian dengan Variasi Kecepatan 1 pada Objek Rambu Lalu Lintas Wajib Arah Sini


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 1 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	157.14 cm/s	Berhasil
	Kedua		Tidak
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Tidak
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			80%

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 2 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	110 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Tidak
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			90%


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 3 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	55 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%


Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 4 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	44 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%

- Tabel Pengujian Dengan Variasi Kecepatan 1 Pada Objek Rambu Lalu Lintas Wajib Arah Kanan

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 1 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	157.14 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Tidak
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Tidak
	Kesepuluh		Berhasil
Rata-rata Keakuratan			80%

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 2 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	110 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Tidak
	Kesembilan		Berhasil
	Kesepuluh		Tidak
Rata-rata Keakuratan			90%

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 3 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	55 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%

Jenis Rambu	Percobaan Yang Dilakukan	Variasi Kecepatan 4 (cm/s)	Keberhasilan Mendeteksi Rambu
	Pertama	44 cm/s	Berhasil
	Kedua		Berhasil
	Ketiga		Berhasil
	Keempat		Berhasil
	Kelima		Berhasil
	Keenam		Berhasil
	Ketujuh		Berhasil
	Kedelapan		Berhasil
	Kesembilan		Berhasil
	Kesepuluh		Berhasil
Rata-rata Keakuratan			100%

Halaman Ini Sengaja Dikosongkan

Biodata Penulis



Penulis mempunyai Nama lengkap Moh. Fiqih Tarmidzi Hakim, lahir di Gresik pada tanggal 21 Desember 1996, merupakan anak pertama dari dua bersaudara. Penulis menempuh pendidikan sekolah dasar di M.I Nurul Ulum Gumeng, pendidikan menengah di SMP Negeri 1 Bungah dan melanjutkan pendidikan menengah atas di SMA Negeri 1 Sidayu, yang kemudian dilanjutkan ke ITS Surabaya Departemen Teknik Fisika tahun 2014 dengan NRP 0231144000113. Hingga pada semester genap tahun 2018 penulis akhirnya melaksanakan Tugas Akhir sebagai syarat untuk menyelesaikan study S-1 di Departemen Teknik Fisika dengan judul “**Rancang Bangun Sistem Deteksi Rambu – Rambu Lalu Lintas Menggunakan Jaringan Syaraf Tiruan Berbasis *Raspberry Pi***”. Bagi pembaca yang memiliki saran, kritik atau ingin berdiskusi lebih lanjut tentang Tugas Akhir ini, maka bisa menghubungi penulis melalui email FiqihTarmidzi.17@gmail.com atau menghubungi ke alamat email kedua dari penulis moh.fiqih.t.h14@mhs.ep.its.ac.id.