

**PERANCANGAN *PROTOTYPE* SISTEM UNTUK PENGATURAN
RAMBU PUTAR BALIK (*U-TURN*) DENGAN METODE
BACKGROUND SUBTRACTION BERBASIS RASPBERRY PI**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Muchammad Irwan

NIM: 141510301111008



PROGRAM STUDI TEKNIK KOMPUTER

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2021

PENGESAHAN

PERANCANGAN *PROTOTYPE* SISTEM UNTUK PENGATURAN RAMBU PUTAR BALIK
(U-TURN) DENGAN METODE *BACKGROUND SUBTRACTION* BERBASIS
RASPBERRY PI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muchammad Irwan
NIM: 145150301111008

Skripsi ini telah diuji dan dinyatakan lulus pada
21 Juli 2021

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1



Fitri Utaminingrum, Dr. Eng., S.T., M.T.
NIP. 19820710 200812 2 001

Dosen Pembimbing 2



Hurriyatul Fitriyah, S.T., M.Sc
NIP. 19851001 201504 2003

Mengetahui
Ketua Jurusan Teknik Informatika




Achmad Basuki, S.T., M.MG., Ph.D.
NIP. 19741118 200312 1 002

PERNYATAAN ORISINALITAS

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

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

Malang, 08 Juli 2021



Muchammad Irwan
NIM: 145150301111008

UNIVERSITAS BR



ABSTRAK

Muchammad Irwan, Perancangan *Prototype* Sistem Untuk Pengaturan Rambu Putar Balik (*U-TURN*) Dengan Metode *Background Substraction* Berbasis Raspberry pi

Pembimbing : Fitri Utaminigrum, Dr. Eng., S.T., M.T.

Hurriyatul Fitriyah, S.T., M.Sc.

Mobil adalah salah satu moda transportasi yang cukup populer digunakan oleh masyarakat di Indonesia, dimana moda transportasi ini populer digunakan karena bisa membantu mobilitas masyarakat di Indonesia, seperti mengangkut hasil alam, kendaraan umum, atau kendaraan pribadi, lalu seiring dengan dengan berjalannya waktu moda transportasi ini juga bertambah dari tahun ke tahun sehingga harus di imbangi juga dengan penambahan peraturan berlalu lintas untuk dapat mengatur kendaraan terutama mobil supaya tertib dalam berlalu lintas, dan untuk menertibkan para pengendara mobil digunakan alat yang di namakan rambu lalu lintas, dimana rambu lalu lintas ada begitu banyak gambar, bentuk, dan fungsi, namun karena kurangnya kesadaran akan pengendara terhadap rambu lalu lintas yang ada maka timbul permasalahan yaitu kemacetan, dimana penyebab kemacetan adalah adanya penumpukan kendaran di suatu titik sehingga menyebabkan kendaraan yang lain menunggu dimana permasalahan ini salah satunya terjadi di area putar balik *u-turn* yaitu karena adanya beberapa pengendara yang menunggu untuk bisa melakukan putar balik karena dari arah yang berlawanan macet atau ramai sehingga pengendara yang ingin putar balik menunggu cukup lama yang mengakibatkan antrian mobil dan menyebabkan kemacetan di kedua arah, namun belum ada sistem atau alat yang di gunakan untuk mengurai kemacetan di area putar balik *u-turn*. Sebagai inovasi menggunakan teknologi solusi permasalahan ini diperlukan sebuah sistem yang berfungsi untuk mendeteksi sebuah mobil di area putar balik *u-turn* dengan berbagai warna mobil, berbagai posisi mobil, dan dengan berbagai kondisi pencahayaan yang berbeda nilai sebagai kondisi nyata di lapangan, dimana sistem untuk pengaturan rambu lalu lintas ini menggunakan kamera yang di hubungkan dengan raspberry pi 3 model b yang berfungsi untuk melihat dan mendeteksi mobil di area putar balik *u-turn*, dimana kamera akan mendeteksi objek mobil dan diolah menggunakan metode *background subtraction* dengan tingkat akurasi 100% di kondisi pencahayaan yang berbeda dan posisi mobil yang berbeda, alat ini jug menggunakan pemberitahuan ke pengendara modil berupa lampu led berwarna merah dan hijau sebagai petunjuk pengendara masih di perbolehkan antri pada area putar balik tersebut dengan ditambahkan sound sebagai pengingat berupa suara kepada pengendara mobil yang akan putar di area tersebut dengan keberhasilan *output* 100%.

Kata kunci : deteksi kendaraan, *background subtraction*, rambu lalu lintas dinamis

ABSTRACT

Muchammad Irwan, Perancangan *Prototype* Sistem Untuk Pengaturan Rambu Putar Balik (U-TURN) Dengan Metode *Background Subtraction* Berbasis Raspberry pi

Supervisor : Fitri Utaminigrum, Dr. Eng., S.T., M.T.

Hurriyatul Fitriyah, S.T., M.Sc.

Car is one of the transportation modes that are quite popular used by the public in Indonesia, where this mode of transportation is popularly used because it can help the mobility of people in Indonesia, such as transporting natural products, public transportation, or private vehicles, then along with the passage of time this mode of transportation also increases from year to year so it must be balanced also with the addition of traffic regulations to be able to regulate vehicles, especially supa cars yes orderly in traffic, and to discipline the motorists used a tool called traffic signs, where traffic signs there are so many pictures, shapes, and functions, but because of the lack of awareness of motorists to existing traffic signs then arises a problem that is congestion, where the cause of congestion is the buildup of vehicles at a point that causes other vehicles to wait where this problem one of them occurs in U-turn area is due to the presence of some motorists who wait to be able to do a U-turn because from the opposite direction is jammed or crowded so that motorists who want to turn around wait long enough that results in car queues and causes congestion in both directions, but there is no system or tools used to parse congestion in the U-turn area. As an innovation using technology this problem solution is required a system that serves to detect a car in the U-turn area with various colors of the car, various positions of the car, and with a variety of different lighting conditions value as a real condition in the field, where the system for setting traffic signs is using a camera connected with raspberry pi 3 model b that serves to see and detect the car in the u-tu reversing area rn, where the camera will detect the car object and processed using the background subtraction method with a 100% accuracy rate in different lighting conditions and different car positions, this tool uses notifications to motorists in the form of red and green led lights as a hint that motorists are still allowed to queue in the rewind area by adding sound as a reminder of the sound to motorists who will rotate in the area with a 100% success output.

Keywords : vehicle detection, background subtraction, traffic signs Dynamic

DAFTAR ISI

PERSETUJUAN.....	iii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	4
1.5 Batasan Masalah	4
1.6 Sistematika Pembahasan	4
1 Pendahuluan	4
2 Landasan Kepustakaan	5
3 Metodologi Penelitian	5
4 Rekayasa Kebutuhan	5
5 Perancangan dan Implementasi Sistem	5
6 Pengujian dan Analisis	5
7 Penutup	5
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Studi Literature	6
2.2 Landasan Kepustakaan	8
2.2.1 Rambu-Rambu Lalu Lintas	8
2.2.1.1 Rambu-Rambu Lalu Lintas Peringatan.....	8

2.2.1.2	Rambu-Rambu Lalu Lintas Larangan	10
2.2.1.3	Rambu-Rambu Lalu Lintas Perintah	11
2.2.1.4	Rambu-Rambu Lalu Lintas Petunjukv	12
2.3	Dasar Teori	14
2.3.1	<i>Digital Image Processing</i>	14
2.3.2	Segmentasi Warna	14
2.3.3	Citra	15
2.3.4	<i>Background Substraction</i>	16
BAB 3 METODOLOGI PENELITIAN		19
3.1	Tipe Penelitian	19
3.2	Strategi dan Rancangan Penelitian.....	19
3.2.1	Metode Umum Penelitian	19
3.2.2	Objek dan Lokasi Penelitian	21
3.2.3	TeknikPengumpulan Data	21
3.2.4	Tenik Analisis Data dan Pembahasan	21
3.2.5	Peralatan Kebutuhan yang di Gunakan	22
3.2.5.1	Kebutuhan Perangkat Keras.....	22
3.2.5.2	Kebutuhan Perangkat Lunak.....	22
BAB 4 REKAYASA KEBUTUHAN SISTEM		23
4.1	Gambaran Umum Sistem	23
4.1.1	Perspektif Sistem	24
4.1.2	Karakteristik Pengguna	24
4.1.3	Asumsi dan Ketergantungan.....	24
4.2	Kebutuhan Antarmuka Eksternal	24
4.2.1	Kebutuhan Antarmuka Perangkat Keras	24
4.2.2	Kebutuhan Antarmuka Perangkat Lunak	28
4.3	Kebutuhan Fungsional	29
4.4	Kebutuhan Non Fungsional	30
BAB 5 PERANCANGAN DAN IMPLEMENTASI		31
5.1	Perancangan Sistem	31

5.1.1	Perancangan Perangkat Keras	31
5.1.2	Perancangan Perangkat Lunak	35
5.1.2.1	Memberikan Input	36
5.1.2.2	Perancangan Pengambilan Data Kamera	36
5.1.2.3	Perancangan Prototype Alat	37
5.1.3	Implementasi System.....	38
5.1.3.1	Implementasi Perangkat Keras	38
5.1.3.2	Implementasi Perangkat Lunak	45
5.1.3.3	Implementasi Pemberian Input	45
5.1.3.4	Implementasi Kode Program	46
5.1.3.5	Implementasi Prototype Alat	51
BAB 6	PENGUJIAN DAN ANALISIS	54
6.1	Pengujian Kode Program Menggunakan Kamera Logitech.....	53
6.1.1	Tujuan Pengujian	53
6.1.2	Prosedur Pengujian	53
6.1.3	Hasil pengujian deteksi kendaraan (mobil) dengan warna objek berbeda	54
6.1.4	Hasil pengujian deteksi kendaraan (mobil) dengan intensitas cahaya yang berbeda	59
6.1.5	Hasil pengujian akurasi integrasi antara <i>input</i> dan <i>output</i> pada sistem.....	64
BAB 7	KESIMPULAN DAN SARAN	69
7.1	Kesimpulan	69
7.2	Saran	69
DAFTAR REFERENSI		71
LAMPIRAN		73

DAFTAR TABEL

Tabel 2.1 Studi literatur.....	6
Tabel 4.1 Spesifikasi AdvanceDUO-040	25
Tabel 4.2 Spesifikasi Raspberry pi 3.....	27
Tabel 5.1 Deskripsi pin raspberry Pi.....	33
Tabel 5.2 Kode program deklarasi library sistem.....	46
Tabel 5.3 Kode program deklarasi variabel.....	47
Tabel 5.4 Kode program deklarasi variable konfigurasi pin output.....	48
Tabel 5.5 Kode program pengolahan <i>background subtraction</i> dan <i>output</i> rambu lalu lintas dinamis.....	49
Tabel 6.1 Hasil Pengujian deteksi kamera dengan warna dan posisi objek (mobil) yang berbeda.....	55
Tabel 6.2 Hasil Pengujian deteksi kamera dengan intensitas cahaya yang berbeda.....	59
Tabel 6.3 Hasil Pengujian akurasi integrasi antara <i>input</i> dan <i>output</i> sistem.....	64
Tabel 6.4 <i>Source code</i> keseluruhan sistem.....	65

DAFTAR GAMBAR

Gambar 2.1 Rambu Lalu Lintas Peringatan.....	9
Gambar 2.1 Rambu Lalu Lintas Peringatan (lanjutan).....	10
Gambar 2.2 Rambu Lalu Lintas Larangan.....	11
Gambar 2.3 Rambu Lalu Lintas Perintah.....	12
Gambar 2.4 Rambu Lalu Lintas Petunjuk.....	13
Gambar 2.4 Rambu Lalu Lintas Petunjuk (lanjutan).....	14
Gambar 2.5 Proses pengolahan <i>background</i> dan <i>foreground</i>	17
Gambar 2.6 <i>flowchart background subtraction</i>	18
Gambar 3.1 Alur Metodologi Penelitian.....	20
Gambar 4.1 Blok Diagram Umum Sistem.....	23
Gambar 4.2 Lampu led.....	25
Gambar 4.3 Gambar ADVANCE DUO-040.....	26
Gambar 4.4 Gambar LOGITECH C 270H.....	27
Gambar 4.5 Gambar raspberry phi 3 model B.....	28
Gambar 5.1 Blok Diagram Perangkat Keras.....	31
Gambar 5.2 Sistematika Perangkat Keras.....	32
Gambar 5.3 Rangkain komponen penyusun sistem.....	34
Gambar 5.4 Diagram alir perancangan keseluruhan sistem.....	35
Gambar 5.5 Diagram alir perancangan pengambilan data dari kamera.....	36
Gambar 5.6 Desain Prototype Alat.....	38
Gambar 5.7 Menyambungkan Raspberry pi dengan Laptop.....	39
Gambar 5.8 Memeriksa jaringan wifi.....	40
Gambar 5.9 Memeriksa ip <i>raspberry pi</i>	40
Gambar 5.10 <i>Log in vnc viewers</i>	41
Gambar 5.11 Tampilan remote raspberry pi.....	41
Gambar 5.12 Memastikan tempat file coding sebelum program di jalankan.....	42

Gambar 5.13 Memindah direktori.....	43
Gambar 5.14 Menjalankan kode program.....	43
Gambar 5.15 Tampilan <i>foreground mask</i> dan frame saat program di jalankan.....	44
Gambar 5.16 Tampilan terminal saat program selesai di jalankan.....	45
Gambar 5.17 Ukuran replika mobil.....	46
Gambar 5.18 Implementasi <i>prototype</i> sistem.....	52
Gambar 6.1 Tampilan keseluruhan <i>prototype</i> sistem.....	73
Gambar 6.2 Tampilan fgmask deteksi objek <i>prototype</i> mobil.....	73
Gambar 6.3 Tampilan 2 objek yang terdeteksi dengan warna objek <i>prototype</i> mobil hijau dan biru.....	74
Gambar 6.4 Tampilan area rambu putar balik (<i>u-turn</i>) tanpa objek <i>prototype</i> mobil yang terdeteksi oleh frame.....	74
Gambar 6.5 Tampilan deteksi objek <i>prototype</i> mobil berwarna biru dengan output sistem bernilai macet.....	75
Gambar 6.6 Tampilan deteksi objek <i>prototype</i> mobil berwarna merah dengan output sistem bernilai macet.....	75
Gambar 6.7 Tampilan deteksi objek <i>prototype</i> mobil berwarna hijau dengan output sistem bernilai macet.....	76
Gambar 6.8 Tampilan deteksi objek <i>prototype</i> mobil berwarna abu-abu dengan output sistem bernilai macet.....	76
Gambar 6.9 tampilan kondisi <i>prototype</i> area putar balik (<i>u-turn</i>) tanpa ada objek yang terdeteksi dalam frame yang telah di tentukan.....	77

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pada saat ini perkembangan teknologi berkembang sangat pesat, hal ini dinilai cukup penting bagi kehidupan manusia dimasa sekarang ini, sehingga dapat berguna jika bisa di manfaatkan pada semua sektor tak terkecuali kegiatan keseharian masyarakat Indonesia, mengingat kepadatan penduduk Indonesia pada tahun 2020 adalah 270,20 juta jiwa dimana hal ini mengalami kenaikan 32,56% dibanding tahun sebelumnya (statistika, 2021).

Teknologi sangat di perlukan dalam mengatur keseharian masyarakat di Indonesia ini terutama pada bidang berlalu lintas dimana teknologi yang berkembang dapat di implementasikan pada bidang lalu lintas menggunakan kamera pengawas. Kamera pengawas atau CCTV (*Close Circuit Television*) berfungsi untuk mendeteksi adanya semua gerak dalam jarak pandang sehingga dapat digunakan untuk pengawasan kondisi suatu tempat (Alvin , et al., 2015), karena dengan jumlah penduduk yang terus bertambah tiap tahunnya maka jumlah kendaraan juga akan bertambah di tiap tahunnya dimana sistem ini nantinya berfungsi untuk mengatur masyarakat dalam berkendara supaya tidak terjadi kemacetan.

Pada tahun 2019 jumlah kendaraan bermotor di Indonesia mencapai 133 juta unit dimana jumlah ini mengalami peningkatan sebanyak 5.3 persen di bandingkan dengan tahun sebelumnya yaitu 2018 (gaikindo, 2021), mobil jenis penumpang adalah salah satu penyumbang kepadatan kendaraan di Indonesia saat ini dengan angka 11,6 persen dimana jumlah ini dinilai cukup tinggi mengingat ruang yang di butuhkan sebuah mobil cukup memakan tempat, sehingga hal ini sering memicu kemacetan di beberapa lokasi terutama jika ada pengendara mobil yang memberhentikan kendaraannya tidak sesuai prosedur sehingga kemacetan tidak dapat di hindari. Pemerintah sebenarnya sudah menyediakan banyak opsi untuk mencegah adanya titik penumpukan kendaraan atau dikenal dengan istilah kemacetan seperti menggalangkan upaya menggunakan kendaraan umum seperti kereta,bis dan sebagainya (theconversation, 2019).

Teknologi yang sekarang juga sudah ada beberapa yang di ciptakan untuk mengurai kemacetan seperti seperti *Area Traffic Control System (ATCS)* (Dishub LLAJ, 2010), *Traffic system Communication Algotithm (TSCA)* (Nadir Farhi, 2017), dan *Traffic Signal Time Manipulation Algorithm (TSTMA)* (Sameh Samra, 2013) dimana ketiganya digunakan untuk mengontrol lalu lintas yang bertujuan untuk mengurai kemacetan terutama di area lampu lalu lintas, sedangkan penyebab kemacetan di titik lain masih belum ada penelitian terkait itu seperti kemacetan akibat antrian mobil yang akan melakukan putar balik di area putar balik (*u-turn*), oleh sebab itu perlu ada penelitian yang digunakan untuk mengurai kemacetan juga pada area putar balik khususnya putar balik dengan jenis *u-turn*.

Teknologi yang berkembang sekarang dapat menjadi solusi untuk mengatasi masalah dalam mengurai kemacetan terutama khusus area putar balik

u-turn dimana sistem ini menggunakan teknologi komputer cerdas dengan pengolahan citra dengan metode *background subtraction*. Metode *background subtraction* dapat mendeteksi subtraksi pada *background* dengan mengubah citra menjadi citra biner dan menentukan tingkat kepekaan perubahan pixel *background*.

Metode *background subtraction* memiliki tingkat akurasi yang tinggi sehingga objek dapat dideteksi dengan baik. Proses yang kompleks pada metode *background subtraction* menjadikan pengolahan data memiliki banyak tahapan dalam pemrosesan citra sehingga program dikatakan sangat kompleks dan harus memiliki sebuah *hardware* yang memadai (Pamungkas, 2017). Data latih diperlukan untuk melatih keakuratan sistem dalam mendeteksi berbagai warna dan posisi mobil pada area putar balik *u-turn*. Hardware yang sering digunakan untuk project komputer vision adalah yaitu Raspberry pi, alat ini dapat digunakan sebagai proses mini komputer yang memuat program dan pemasangan komponen lain didalamnya menggunakan pin, kamera sebagai penangkap objek akan diteruskan ke mikrokomputer Raspberry pi 3 model B agar proses yang berjalan dapat memadai. Setelah objek dapat terdeteksi dan diolah oleh metode *background subtraction* maka akan diteruskan ke proses *output* suara dan lampu led merah dan lampu led hijau sehingga menjadikan alat ini petunjuk rambu lalu lintas yang dinamis dan interaktif bagi pengguna mobil yang akan melakukan putar balik supaya tidak menjadi penyebab kemacetan di area putar balik *u-turn*.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka penulis merancang beberapa poin untuk dijadikan rumusan masalah, sebagai berikut :

1. Bagaimana rata-rata hasil akurasi deteksi kendaraan (mobil) pada area putar balik menggunakan *background subtraction* dengan warna objek yang berbeda?
2. Bagaimana rata-rata waktu komputasi hasil deteksi kendaraan (mobil) pada area putar balik dengan tingkat pencahayaan yang berbeda?
3. Bagaimana akurasi integrasi antara input dan output *hardware* sistem pada rambu lalu lintas dinamis?

1.3 Tujuan Penelitian

Adapun berbagai macam tujuan penelitian yang ingin disampaikan dan dicapai oleh penulis melalui sistem ini. Tujuannya adalah :

1. Menghitung rata-rata akurasi dari deteksi kendaraan (mobil) pada area rambu putar balik dengan warna objek yang berbeda berupa mobil menggunakan metode *background subtraction*.

2. Menghitung rata-rata waktu hasil akurasi deteksi kendaraan (mobil) pada area rambu lalu lintas putar balik dengan tingkat percahayaan yang berbeda.

3. Mengetahui nilai tingkat akurasi sistem rambu lalu lintas dinamis pada komputer vision pada deteksi objek berupa mobil terhadap notifikasi *output* suara dari sound dan lampu led sebagai rambu lalu lintas dinamis

1.4 Manfaat Penelitian

Manfaat yang di peroleh dengan adanya penelitian ini adalah sebagai berikut:

1. rekomendasi yang diberikan diharapkan dapat di pakai sebagai bahan masukan dan sebagai landasan kebijakan bagi dinas terkait untuk meningkatkan optimalisasi penerapan rambu putar balik di jalan raya.
2. tampilan rambu-rambu lalu lintas lebih komunikatif, sehingga pengendara bisa melihat dalam segala kondisi di jalan raya.
3. System dapat menjadi salah satu solusi alternatif untuk dapat mengurangi kemacetan di jalan raya, terutama kemacetan yang di sebabkan oleh aktifitas putar balik.

1.5 Batasan Masalah

Sehubungan dengan adanya keterbatasan pada sistem ini dan keterbatasan pada penulis, baik dari segi waktu, biaya, pemikiran, maka dibutuhkan adanya batasan masalah terhadap sistem ini. Batasan masalahnya adalah sebagai berikut :

1. Penelitian ini menggunakan mikrokontroler *raspberry pi*
2. Penelitian ini menggunakan lampu LED berwarna merah, kuning, dan hijau sebagai prototype rambu lalu lintas putar balik dinamis dan spiker sebagai tambahan untuk media suara bagi pengendara mobil
3. Penelitian ini menggunakan prototype dengan skala 1:54,33 cm
4. Penelitian ini menggunakan aplikasi lux *light* meter di *handphone* untuk mengukur intensitas cahaya sesuai dengan berbagai kondisi di jalan raya seperti pagi hari, siang hari yang cerah, siang hari jika mendung, sore hari, dan malam hari.

1.6 Sistematika Pembahasan

Dalam pembahasan dan penelitian ini dibagi menjadi 7 bagian utama, yaitu adalah :

BAB 1 PENDAHULUAN

Bab pendahuluan menjelaskan tentang dasar landasan masalah atau urgensi yang terjadi, latar belakang dibuatnya penelitian, solusi dari identifikasi

masalah, tujuan penelitian, manfaat penelitian, batasan masalah penelitian dari sistem "Perancangan Prototype Sistem Untuk Pengaturan Rambu Putar Balik (U-Turn) Dengan Metode *Background Substraction* Berbasis Raspberry Pi"



BAB 2 LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan menjelaskan tentang landasan teori penelitian yang berhubungan dengan komputer vision khusus bidang *object detection*. Penjelasan deteksi objek diteliti menggunakan metode *Background Substraction* dalam pembuatannya.

BAB 3 METODOLOGI PENELITIAN

Bab metodologi membahas tentang kerangka penelitian yang sebagai ulasan dari identifikasi masalah, rumusan masalah, dan tujuan penelitian yang akan dicapai. Pada bagian metodologi juga membahas perancangan sistem yang akan dikembangkan termasuk rancangan lingkungan penelitian dan metode evaluasi.

BAB 4 REKAYASA KEBUTUHAN SISTEM

Bab rekayasa kebutuhan sistem menguraikan gambaran umum sistem, analisis kebutuhan fungsional, analisis kebutuhan non fungsional berupa perangkat keras dan perangkat lunak, serta batasan desain sistem.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini akan menjabarkan proses perancangan sistem pada raspberry pi untuk deteksi objek dengan menggunakan *Background Substraction*. Kemudian, penelitian akan melanjutkan dengan mengimplmentasikan secara langsung pada hardware.

BAB 6 PENGUJIAN DAN ANALISIS

Bab pengujian pada penelitian ini dilakukan apabila perancangan dan implementasi pada sistem telah dilalui. Pengujian dilakukan untuk memperoleh data yang nantinya digunakan untuk proses analisis penelitian.

BAB 7 PENUTUP

Bab penutup berisikan kesimpulan dan saran dari hasil penelitian. Hasil penelitian tersebut diharapkan dapat membantu pengembangan penelitian yang terkait.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab Landasan kepastakaan memiliki uraian dasar teori pada penilitian dan sumber studi literatur. Kajian pustaka akan membahas pembahasan dari sumber penelitian lain yang pernah dikaji sehingga dasar ide dari penulis diambil dari percobaan yang telah dilakukan sebelumnya

2.1 Studi Literature

Dalam pengerjaan penelitian ini, penulis menggunakan penelitian sebelumnya untuk di jadikan literature. Dimana studi literature yang penulis pilih digunakan untuk mencari informasi yang ada sebelumnya mengenai metode- metode yang berkaitan dengan judul yang digunakan untuk mendapat landasan teori ilmiah.

Tabel 2.1 Studi literatur

No	Nama Penulis (Tahun) dan Judul Penelitian	Persamaan	Perbedaan	
			Penelitian Sebelumnya	Rencana Penelitian
1	Faisal Amir. (2016), Perancangan Aplikasi Komputer Pusat Untuk Mengendalikan Lampu Lalu Lintas Dinamis Secara Wireless	Mengkaji objek penelitian yaitu lintas dinamis	Menggunakan <i>hardware</i> berupa ardino IDE	Menggunakan Metode <i>Background subtraction</i> pada komputer vision pada raspberry phi 3 model B
2	Tjia may on. (2007), Sistem Pengaturan Lampu Lalu Lintas Secara Sentral Dari Jarak Jauh	Mengkaji deteksi objek gambar mobil melalui kamera pada komputer vision	Objek yang di deteksi merupakan mobil Dan gambar di olah secara manual	Obek deteksi merupakan mobil dengan berbagai warna dan posisi dengan <i>real time</i> pada hardware raspberry pi dan output suara dan lampu led
3	Teuku Hermansyah, M.Isya , M.Saleh. (2015), Keceratan Rambu dan Marka Terhadap Geometrik Jalan Pada Jalanan Antar Kota	Mengkaji tentang penempatan rambu lalu lintas	Objek yang di deteksi hanya penempatan rambu lalu lintas yang dinilai statis dan kurang sesuai kondisi jalan	Objek yang di deteksi berupa mobil pada area putar balik dengan output rampu lalu lintas dinamis

1. skripsi oleh faisal amir, mahasiswa S1 teknik komputer dari universitas brawijaya yang telah lulus sidang pada april 2016 dengan judul *perancangan aplikasi computer pusat untuk mengendalikan lampu lalu lintas dinamis secara wireless*.

Penelitian ini bersifat implementatif, dimana si penulis melakukan penelitian dengan tujuan untuk mengembangkan teknologi yang sudah ada saat ini, seperti *traffic system communication algorithm* (tsca), dan *traffic signal time manipulation algorithm* (tstma) dimana menurut penulis 2 system tersebut masih kurang efisien, karena komunikasi yang di lakukan masih melalui media kabel, dan masih terhitung lama pada saat pengiriman data dari computer pusat ke node yang ada pada tiap lampu lalu lintas, selain itu juga permasalahan ada pada pembagian waktu pada tiap lampu lalu lintas, sehingga penulis berfokus pada bagaimana management waktu dalam berkomunikasi tiap node, agar waktu komunikasi lebih singkat dengan tingkat akurasi yang sangat tinggi.

2. Jurnal teknik elektro oleh tjia may on, pono budi mardjoko, nato martanto dengan jurnal dari TESLA pada oktober 2007 dengan judul *system pengaturan lampu lalu lintas secara sentral dari jarak jauh*.

Pada penelitian ini juga bersifat implementatif, dimana penelitian di lakukan bertujuan untuk mengembangkan dari system yang sudah ada sebelumnya, yakni ATCS dimana pada ATCS yang di lakukan pada lampu lalu lintas adalah pengambilan sample dari tiap sisi jalan akan di olah manual oleh computer pusat melalui operator dan pengaturan time tiap rambu akan di atur secara manual oleh petugas terdekat, dan penulis memperbaiki system tersebut dengan komunikasi secara wireless, yaitu pengambilan data di ambil dari kamera yang terpasang pada lampu lalu lintas, data di tampung pada receiver terdekat dan di olah oleh computer pusat, jadi computer pusat hanya mengolah data dan menentukan waktu yang nantinya akan di tampilkan pada lampu lalu lintas, namun pada penelitian ini ada kendala yakni wireless yang di gunakan hanya sejauh 600m, jika lebih dari 600m maka keakuratan data yang di kirim akan berkurang, bahkan kemungkinan tidak sampai, jadi antara transmitter dan receiver dibutuhkan node yang dapat menjangkau jarak yang cukup jauh, sehingga keakuratan data dapat terjamin.

3. Jurnal teknik sipil pascasarjana universitas syah kuala oleh Teuku Hermansyah, M. Isya', Sofyan M. Saleh pada November 2015 dengan judul *KESERASIAN RAMBU DAN MARKA TERHADAP GEOMETRIK JALAN PADA JALAN ANTAR KOTA*.

Penelitian ini bersifat implemenatif, yaitu suatu pengembangan dari system sebelumnya dimana sang penulis menjelaskan mengenai kekurangan rambu lalu lintas dan marka jalan yang ada di Indonesia (pada

jurnal di fokuskan di daerah jalan banda aceh KM.77 batas pidle), menurut penelitian ini juga pada rambu lalu lintas ditemukan banyak kekurangan, yaitu berdasarkan ukuran rambu lalu lintas yang belum sesuai aturan, penempatan rambu lalu lintas yang belum sesuai, dan tinggi tiang yang belum sesuai juga, sehingga kurang maksimal dan kurang berguna adanya rambu lalu lintas di wilayah tersebut,

2.2 Landasan Kepustakaan

2.2.1 Rambu-Rambu Lalu Lintas

Indonesia banyak sekali rambu-rambu lalu lintas yang terpasang di jalan, yang tentunya berfungsi sebagai media informasi, media larangan, media peringatan dan yang pasti untuk menjaga agar terhindarnya terjadi kecelakaan di jalan. Akan tetapi meskipun banyak sekali rambu lalu lintas yang harus kita ketahui dan harus kita taati, adapun teknik pembagiannya supaya kita sebagai masyarakat tidak bingung bahwa rambu yang misalkan berwarna biru itu termasuk dalam rambu larangan, peringatan, petunjuk atau yang lainnya.

Menurut Keputusan Menteri Perhubungan No. KM 61 Tahun 1993 (Dishub, 1993) Tentang Rambu-Rambu Lalu Lintas di Jalan, rambu lalu lintas adalah salah satu alat perlengkapan jalan dalam bentuk tertentu yang memuat lambang, huruf, angka, kalimat dan atau perpaduan di antaranya, yang digunakan untuk memberikan peringatan, larangan, perintah dan petunjuk bagi pemakai jalan. Rambu lalu lintas dibuat untuk menciptakan kelancaran, keteraturan dan keselamatan dalam berkendara. Marka jalan dan rambu - rambu merupakan obyek untuk menyampaikan informasi atau perintah maupun petunjuk bagi pemakai jalan. Di penelitian atau karya tulis ilmiah ini akan dijelaskan pembagian rambu lalu lintas itu seperti apa, berikut penjelasannya.

2.2.1.1 Rambu Lalu Lintas Peringatan

Rambu peringatan berfungsi sebagai memberi peringatan kepada pengendara jalan bahwa mungkin di depan akan ada jalan bahaya, atau ada perempatan, atau ada jalan menyempit atau ada apapun yang kemungkinan bisa membahayakan masyarakat dalam berkendara (Fakhli, 2018). Rambu ini biasanya dipasang sekitar 50 meter atau dipasang di jarak tertentu sebelum sesuatu yang dianggap membahayakan sampai, karena melihat kondisi jalan, kondisi cuaca, atau kondisi geografis ditempat itu, makanya dipasang rambu dengan jarak yang sesuai dengan kebutuhan dan kondisi pengendara.

Bentuk dari rambu ini biasanya berbentuk bujur sangkar atau persegi yang diputar 45 derajat ke kiri maupun ke kanan. Dan biasanya pada rambu-rambu ini berwarna kuning dengan tulisan atau gambar berwarna hitam yang bertujuan

agar meskipun dalam kondisi gelap, rambu tersebut bisa tetap terlihat oleh pengendara.

Banyak sekali rambu-rambu peringatan yang terdapat di jalan, ada 70 macam rambu lalu lintas yang berwarna kuning yang sebenarnya dibuat oleh Menteri Perhubungan, ini bersumber dari Keputusan Menteri Perhubungan No. KM 61 Tahun 1993 lampiran 1 (Menhub, RAMBU-RAMBU LALU LINTAS DI JALAN, 1993). Akan tetapi mungkin hanya sedikit yang kita ketahui di jalan yang biasa kita lewati, karena tidak semua rambu lalu lintas harus dipasang di jalan, dipasang hanya jika jalan itu dibutuhkan peringatan, misalkan akan ada perempatan, tempat tertentu sering terjadi longsor, peringatan akan adanya rel kereta api, dan sebagainya yang masih banyak lagi. Gambar 2.1 merupakan contoh sampel gambar rambu lalu lintas peringatan.



Gambar 2.1 Rambu Lalu Lintas Peringatan (kakisentot, 2021)



Gambar 2.1 Rambu Lalu Lintas Peringatan (lanjutan) (kakisentot, 2021)

2.2.1.2 Rambu Lalu Lintas Larangan

Rambu larangan ini menunjukkan atau menggambarkan hal yang tidak boleh dilakukan oleh pengendara kendaraan, rambu ini dipasang dititik sedekat mungkin dengan tempat yang dilarang atau tepat ditempat yang tidak boleh untuk melakukan hal tersebut. Rambu lalu lintas ini bisa dilengkapi dengan rambu tambahan sesuai dengan kebutuhan (humaspolresbantul, 26).

Bentuk dari rambu lalu lintas ini berbagai macam, ada yang berbentuk segitiga terbalik, berbentuk bulat, berbentuk tanda X, empat persegi panjang, segi delapan sama sisi. Tetapi meskipun bentuk dari rambu ini berbagai macam, warna yang digunakan adalah sama dengan satu simbol itu berwarna dasar putih, memakai juga warna merah, dan misalkan ada gambar atau tulisan yang harus dipasang di rambu tersebut, biasanya berwarna hitam.

Begitu banyak bentuk dan ragam dari rambu lalu lintas, jika kita ketahui semua yang dibuat oleh Menteri Perhubungan jumlahnya adalah 49 macam, yang bisa kita ketahui dengan melihat Keputusan Menteri Perhubungan No. KM 61 Tahun 1993 lampiran 1 (Menhub, Rambu Rambu Lalu Lintas di Jalan, 1993). Gambar 2.2 merupakan contoh sampel gambar rambu lalu lintas larangan.



Gambar 2.2 Rambu Lalu Lintas Larangan (Dishub, 1993)

2.2.1.3 Rambu Lalu Lintas Perintah

Rambu lalu lintas perintah ini dibuat bertujuan supaya pengendara kendaraan tertib dalam menggunakan ruas jalan, dan juga agar terhindar dari kecelakaan, karena rambu lalu lintas ini mengatur segala hal yang harus kita

patuhi dalam berkendara, mulai dari arah berkendara, kecepatan maksimal berkendara, ruas jalan tersebut digunakan oleh kendaraan jenis apa atau untuk orang berjalan dan masih banyak lagi yang lainnya.

Rambu lalu lintas ini biasanya berbentuk bulat dan berwarna dasar biru, serta warna putih untuk menunjukkan perintah rambu lalu lintas dan warna merah garis serong untuk menunjukkan batas akhir perintah yang harus ditaati (carsome, 2021). Rambu lalu lintas ini dipasang dititik yang sedekat mungkin dengan area yang titik kewajiban atau perintah yang harus ditaati.

Rambu-rambu lalu lintas perintah ini tidak terlalu banyak macamnya dibandingkan rambu lalu lintas yang dibahas di sub bab sebelumnya, hanya ada 22 macam yang tercantum dalam Keputusan Menteri Perhubungan No. KM 61 Tahun 1993 lampiran 1 (Dishub, 1993). Gambar 2.3 merupakan contoh sampel gambar rambu lalu lintas perintah



Gambar 2.3 Rambu Lalu Lintas Perintah (Dishub, 1993)

2.2.1.4 Rambu Lalu Lintas Petunjuk

Rambu lalu lintas ini berfungsi sebagai rambu yang memberi informasi kepada pengguna jalan, baik yang berkendara maupun yang berjalan kaki mengenai jurusan jalan, nama jalan, situasi kota, suatu tempat, pengaturan jalan, fasilitas dan lain-lain. Rambu ini dibuat supaya para pemakai jalan bisa mengetahui beberapa informasi yang telah dicantumkan dan tidak bingung jikalau pengguna jalan tersebut tidak tahu arah tujuannya mau kemana dan seberapa jauh lagi fasilitas jalan apa yang dia butuhkan.

Rambu-rambu ini mempunyai warna bervariasi, ada rambu yang bertujuan memberi tahu nama suatu tempat, itu biasanya berwarna dasar biru kemudian ada tulisan yang berwarna putih. Ada rambu yang bertujuan untuk memberi informasi jurusan jalan, biasanya rambu tersebut berwarna dasar hijau kemudian terdapat tulisan yang berwarna putih dan terdapat arah kemana jurusan jalan tersebut mengarahkan pengguna jalan ke suatu tempat yang dituju. Ada juga rambu lalu lintas petunjuk yang berfungsi sebagai petunjuk seberapa jauh lagi fasilitas di jalan yang dibutuhkan oleh pengguna jalan, misalkan pengguna jalan membutuhkan masjid untuk melaksanakan ibadah, maka rambu tersebut dipasang sekitar 50 meter sebelum masjid tersebut berada.

Rambu petunjuk yang menyatakan tempat fasilitas umum, batas wilayah suatu daerah, situasi jalan, dan rambu berupa kata-kata serta tempat khusus dinyatakan dengan warna dasar biru. Rambu petunjuk pendahuluan jurusan rambu petunjuk jurusan dan rambu penegas jurusan yang menyatakan petunjuk arah untuk mencapai tujuan antara lain kota, daerah/ wilayah serta rambu yang menyatakan nama jalan dinyatakan dengan warna dasar hijau dengan lambang dan/atau tulisan warna putih. Khusus rambu petunjuk jurusan kawasan dan objek wisata dinyatakan dengan warna dasar coklat dengan lambang dan/atau tulisan warna putih.

Jumlah dari berbagai macam rambu lalu lintas petunjuk yang kita ketahui di jalan lumayan banyak, tapi perlu diketahui bahwa jumlah dari rambu lalu lintas petunjuk yang terdapat dalam Keputusan Menteri Perhubungan No. KM 61 Tahun 1993 lampiran 1 (Dishub, 1993) adalah sebanyak 64 macam rambu petunjuk. Gambar 2.4 merupakan contoh sampel gambar rambu lalu lintas petunjuk.



Gambar 2.4 Rambu Lalu Lintas Petunjuk (Dishub, 1993)



Gambar 2.4 Rambu Lalu Lintas Petunjuk (lanjutan) (Dishub, 1993)

Dalam penelitian ini penulis mengangkat topik rambu-rambu lalu lintas petunjuk yang spesifikasinya rambu lalu lintas petunjuk jurusan arah jalan. Rambu lalu lintas yang berfungsi sebagai penunjuk arah jalan, misalkan arah ke Surabaya belok ke kanan.

2.3 Dasar Teori

yaitu menjelaskan mengenai dasar yang menjadikan penelitian ini dilakukan ,yaitu :

2.3.1 Digital Image Processing

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud dalam penelitian ini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari kendaraan mobil *prototype*). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer.

2.3.2 Segmentasi Warna

Segmentasi warna adalah fase seleksi *pixel* berdasarkan *range* warna HSV yang telah ditentukan. Hasil dari fase ini adalah *Region of Interest (ROIs)*. Alasan

untuk menggunakan HSV adalah cara ruang warna HSV yang memisahkan nilai *hue*, *saturation*, dan *value* sehingga tidak seluruh warna perlu digunakan. Hanya warna dari rambu lalu lintas (perintah, larangan, peringatan). Berikut beberapa langkah untuk melakukan konversi ruang warna RGB menjadi ruang warna HSV.

Mencari nilai R, G, B melalui aplikasi edit warna kemudian mencari nilai R' , B' , G' melalui perbandingan masing-masing komponen warna.

- R' adalah untuk menormalkan tingkat warna merah/*red* dengan jangkauan 0 hingga 1 atau dengan kata lain dengan jangkauan minimal dengan maksimal.
- G' adalah untuk menormalkan tingkat warna hijau/*green* dengan jangkauan 0 hingga 1 atau dengan kata lain dengan jangkauan minimal dengan maksimal.
- H' adalah untuk menormalkan tingkat warna biru/*blue* dengan jangkauan 0 hingga 1 atau dengan kata lain dengan jangkauan minimal dengan maksimal.

$$R' = R / 255 \quad (2.1)$$

$$G' = G / 255 \quad (2.2)$$

$$H' = H / 255 \quad (2.3)$$

Mencari nilai c_{max} , c_{min} , dan delta :

- c_{max} adalah mencari nilai C dalam tingkatan maksimal pada R', G', B' yang sudah dicari pada persamaan 2.1 hingga 2.3
- c_{min} adalah mencari nilai c dalam tingkatan minimum pada R', G', B' yang sudah dicari pada persamaan 2.1 hingga 2.3
- Sedangkan delta adalah hasil pengurangan antara c_{max} terhadap c_{min} sehingga selisih itulah yang digunakan nilainya oleh delta

$$c_{max} = \max(R', G', B') \quad (2.4)$$

$$c_{min} = \min(R', G', B') \quad (2.5)$$

$$\Delta = c_{max} - c_{min} \quad (2.6)$$

2.3.3 Citra

Citra adalah sebuah fungsi intensitas cahaya dua dimensi $f(x,y)$ dimana x adalah posisi baris dan y adalah posisi kolom sedangkan F adalah fungsi intensitas atau kecerahan dari citra pada koordinat (x,y) . Pada umumnya citra berbentuk persegi panjang dan dimensi ukurannya dinyatakan sebagai (tinggi x lebar). Citra dengan tinggi N pixel, lebarnya M pixel, dan memiliki intensitas f dapat dipresentasikan sebagai suatu matriks yang berukuran N baris dan M kolom sebagai berikut :

- (x,y) : adalah koordinat pada bidang dwi matra

- $f(x,y)$: adalah intensitas cahaya (*brightness*) pada titik (x,y)

$$F = [f(x,y)] =$$

$$\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Indeks baris (x) dan indeks kolom (y) menyatakan suatu koordinat titik pada citra. Masing-masing titik (x,y) di citra disebut *picture element* atau *pixel*, sedangkan $f(x,y)$ merupakan intensitas (derajat keabuan) pada titik (x,y) .

Berikut adalah jenis-jenis citra berdasarkan nilai *pixel*-nya:

1. Citra Biner

Citra biner adalah citra digital memiliki dua kemungkinan nilai *Pixel* yaitu hitam(0) dan putih(1), yang disebut juga dengan citra B & W (*black and white*) atau monokrom. 1 bit digunakan untuk mewakili nilai setiap *Pixel* dari citra biner (Munir, 2019). Citra biner biasanya dihasilkan dari proses pengolahan seperti segmentasi, pengambangan ataupun morfologi.

2. Citra Grayscale

Citra grayscale adalah pada setiap *pixel* merepresentasi derajat intensitas cahaya atau keabuan. Terdapat 256 jenis derajat keabuan pada citra grayscale, mulai dari putih, hingga semakin gelap sampai warna hitam (achmadrizal, 2014). Oleh karena terdapat kemungkinan 256 derajat keabuan, maka setiap *Pixel* pada grayscale disimpan 1 byte memory (8 bits).

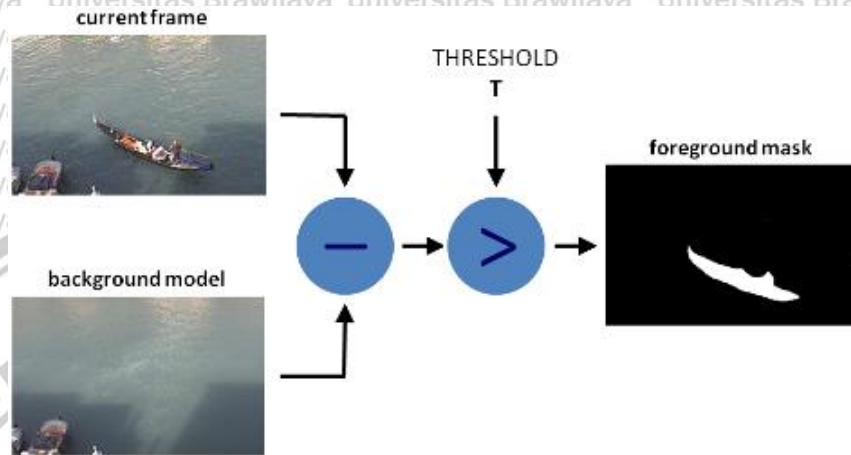
3. Citra Warna

Citra warna terdiri dari 3 layar *matrix*, yaitu *R-layer*, *G-layer*, *B-layer*. Sistem warna RGB (Red, Green, Blue) menggunakan sistem tampilan grafik kualitas tinggi yaitu 24 bit. Setiap komponen warna merah, hijau, biru masing-masing mendapatkan alokasi 8 bit untuk menampilkan warna

2.3.4 Background subtraction

Metode *background subtraction* atau yang juga dikenal sebagai *Foreground Detection*, adalah salah satu teknik pada bidang pengolahan citra dan *computer vision* yang bertujuan untuk mendeteksi/mengambil *foreground* dari *background* untuk diproses lebih lanjut (seperti pada

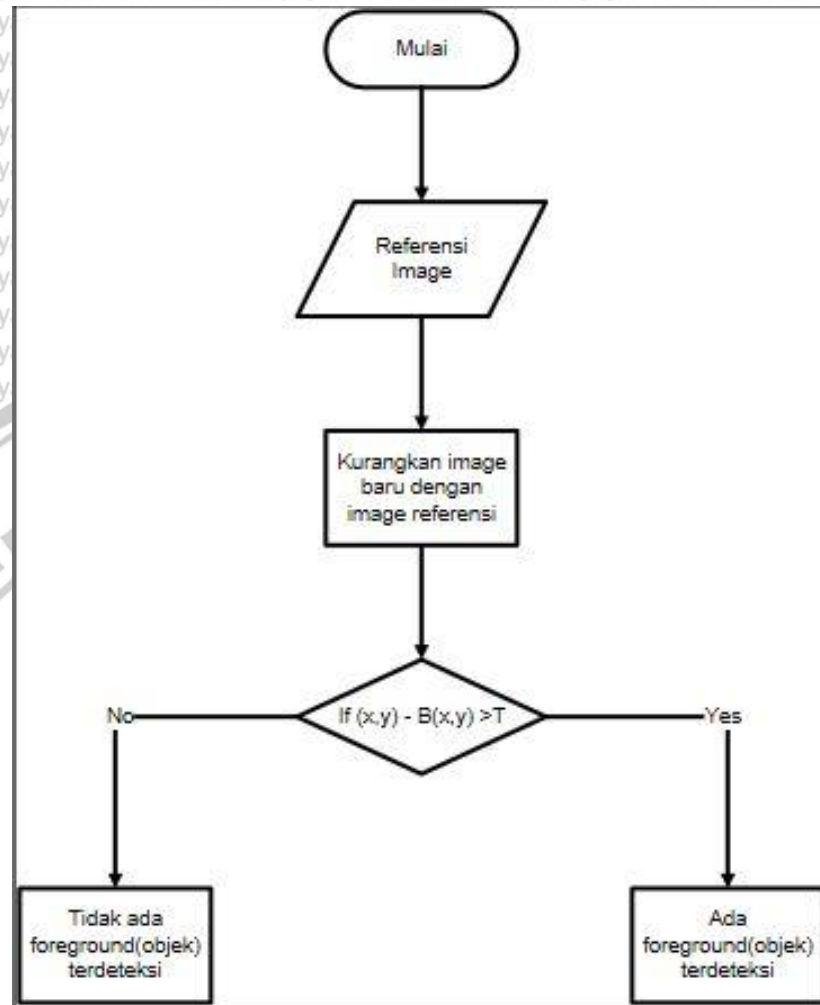
proses *object recognition* dll). Umumnya *foreground* yang diinginkan adalah berupa objek manusia, mobil, teks, dll. *Background subtraction* merupakan metode yang umumnya digunakan untuk mendeteksi objek bergerak pada video dari kamera statis (*stationary camera*). Proses deteksi objek bergerak dengan metode *background subtraction* didasarkan pada perbedaan antara *background* referensi dengan *frame*.



Gambar 2.5 Proses pengolahan *background* dan *foreground*

(Miss Helly M Desai, 2014) mengatakan bahwa skenario utama backgroundsubtractor adalah jika selisih jarak antar Frame yang melebihi nilai ambang batas maka akan ditandai sebagai objek dan jika tidak maka akan diabaikan.

Dan berikut flowchart *background subtraction* menurut (Miss Helly M Desai, 2014) terlihat pada gambar 2.7 dibawah ini :



Gambar 2.6 *flowchart background subtraction*

BAB 3 METODOLOGI PENELITIAN

3.1 Tipe Penelitian

Tipe penelitian ini adalah impelementatif (Perancangan) dimana terdapat prinsip-prinsip rekayasa secara sistimatis untuk menunjang penelitian antara lain, analisis, perancangan, implementasi dan pengujian.

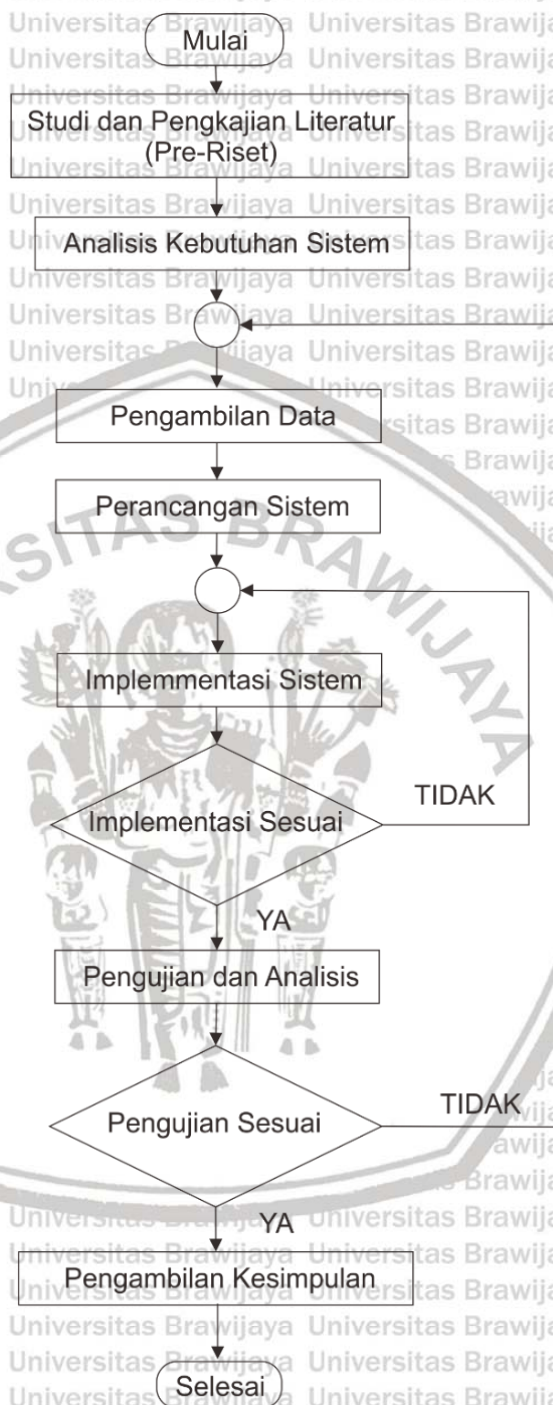
3.2 Strategi dan Rancangan Penelitian

Strategi dan rancangan penelitian ini dibagi dalam beberapa sub-bab agar penjelasan menjadi detail. Hal-hal yang perlu di jelaskan antara lain adalah, metode umum penelitian, subjek penelitian, lokasi penelitian, Teknik pengambilan data, teknik analisis data beserta pembahasannya, dan peralatan pendukung.

3.2.1 Metode Umum Penelitian

Di dalam bab ini membahas tentang langkah-langkah apa saja yang kan diambil untuk mewujudkan tercapainya suatu penelitian atau berhasilnya sebuah penelitian ini. Di bab ini menjelaskan tentang studi literatur apa saja yang diperlukan untuk menunjang berhasilnya sebuah penelitian ini yang studi literatur itu adalah langkah awal dari pengerjaan suatu penelitian. Tidak hanya membahas tentang studi literatur, melainkan juga membahas semua proses sampai akhir bagaimana penelitian bisa berjalan dan berhasil dikerjakan. Berikut alur metodologi yang akan dilaksanakan.





Gambar 3.1 Alur Metodologi Penelitian

Berdasarkan Gambar 3.1 dapat dilihat semua proses dilakukan secara berurutan dimulai dari studi dan pengkajian literatur sebagai pre-riset sampai

dengan kesimpulan. Dari beberapa tahapan di atas ada pula tahapan yang memiliki syarat untuk melanjutkan ke tahap selanjutnya. Syarat ini memiliki pengertian apabila pada suatu tahapan tertentu tidak memiliki hasil atau *output* seperti yang diharapkan, maka tahapan tersebut akan diulang. Ketika tahapan yang dijalankan sesuai dengan apa yang diinginkan oleh user, maka akan dilanjutkan ke tahapan selanjutnya. Hal ini terlihat pada tahapan implementasi sistem, ketika sesuai akan dilanjutkan ke tahap selanjutnya yaitu tahap analisis, ketika hasil tidak sesuai maka tahapan implementasi sistem akan diulang sampai memiliki hasil seperti yang diinginkan.

3.2.2 Objek dan Lokasi Penelitian

Yang akan digunakan sebagai objek dalam penelitian ini adalah prototype yang meliputi mobil dan area putar balik dengan skala 1:54,33cm beserta lampu LED merah, hijau, dan *sound* sebagai prototype rambu lalu lintas dinamis.

Dimana prototype di buat menyerupai lokasi area putar balik di Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur yang mana area ini berada tepat di depan politeknik negeri malang, dengan objek berupa kendaraan mobil dengan berbagai kondisi cuaca seperti pagi hari, siang hari, mendung dan malam hari dimana penulis menyesuaikan pencahayaan pada prototype dengan lux meter berdasarkan pencahayaan di beberapa kondisi tersebut.

3.2.3 Teknik Pengumpulan Data

Pengambilan data digunakan untuk mendapatkan *sample* gambar yang berfungsi sebagai inputan sistem. Pengambilan data tersebut dilaksanakan dengan menggunakan prototype. Data tersebut diambil secara *random* dengan posisi mobil yang berbeda-beda, melihat begitu banyak kemungkinan posisi kendaraan di jalan raya yang sebenarnya.

Dimana pada pengumpulan data ini, prototype nantinya akan di uji kan juga dengan kondisi pencahayaan yang berbeda sesuai dengan intensitas cahaya pada pagi hari cerah, siang hari cerah, mendung, sore hari dan malam hari sehingga secara tidak langsung pengujian akan dilakukan dengan intensitas cahaya yg bervariasi dimana pencahayaan di beberapa kondisi tersebut sebelumnya sudah di ambil sample intensitas cahayanya oleh penulis dengan lux meter dan akan di sesuaikan ukuran intensitas cahayanya pada saat pengujian pada *prototype*.

3.2.4 Teknik Analisis Data dan Pembahasan

Analisis data dan pembahasan sistem bertujuan untuk mengetahui apakah sistem telah mencapai tujuan atau belum. Data yang telah diolah pada sistem berupa hasil *capture* gambar dari kendaraan yang di ambil oleh kamera yang berada di salah satu sisi area putar balik. Umumnya kamera di letakkan tidak jauh

dari area putar balik dan berposisi di atas dengan sudut 180 derajat atau tegak lurus keatas dengan ketinggian kamera pada prototype yaitu 20cm sehingga cakupan kamera menentukan juga keberhasilan sistem dalam mendeteksi kendaraan yang antri, sedangkan posisi rambu lalu lintas dinamis berada pada posisi rambu lalu lintas konvensional sebelumnya, sehingga pengemudi lebih mudah untuk mengetahui keberadaan rambu pada kondisi pagi, siang, malam, ataupun saat hujan. Pengukuran dalam pendeteksian pada kendaraan mobil menggunakan pencahayaan yang bervariasi yakni kurang lebih 1 lux hingga 4000 lux di sesuaikan dengan kondisi pagi hari, siang hari, mendung, sore, dan malam hari.

3.2.5 Peralatan Kebutuhan yang Digunakan

Pada penelitian membutuhkan perangkat keras dan perangkat lunak untuk mendukung sistem ini supaya berjalan dengan lancar, berikut kebutuhannya :

3.2.5.1 Kebutuhan Perangkat Keras

Adapun beberapa perangkat keras yang dibutuhkan untuk menunjang keberhasilan terhadap penelitian ini, perangkat keras yang dibutuhkan adalah :

1. Raspberry pi 3 model B V1.2
2. Sound Sistem / Speaker (dalam penelitian ini yang digunakan adalah sound yang akan di pasang di rambu lalu lintas dinamis)
3. Prototype mobil
4. Prototype lampu LED warna merah dan hijau
5. Prototype jalan raya
6. Logic shifter converter 3,3v-5v

3.2.5.2 Kebutuhan Perangkat Lunak

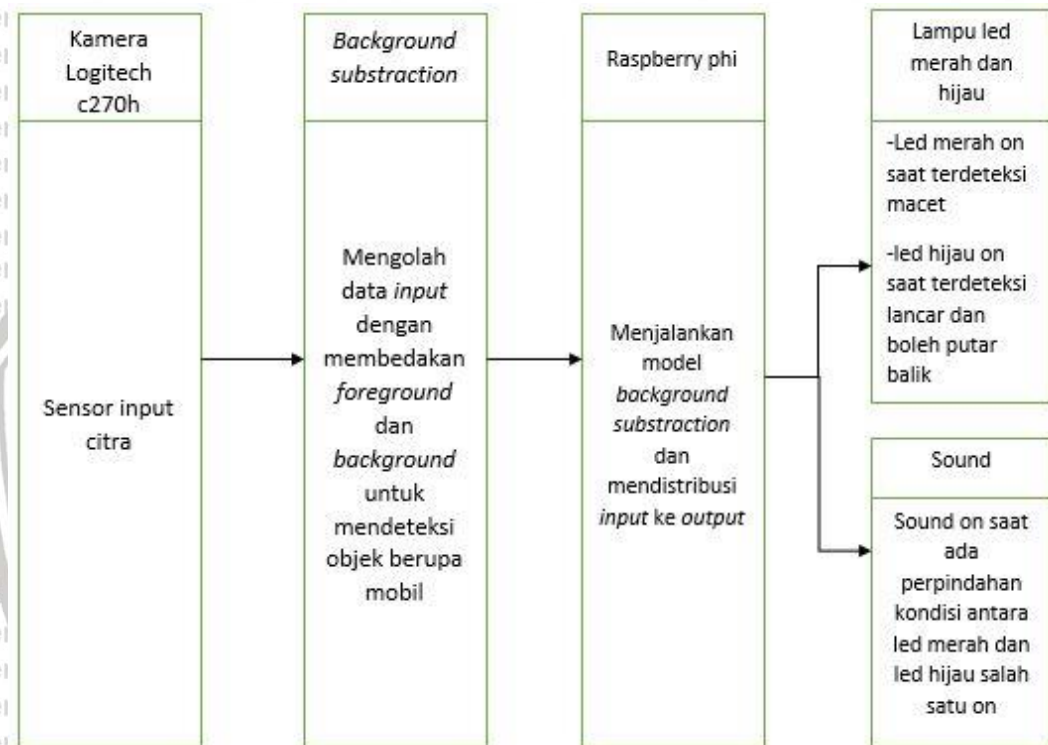
Adapun beberapa perangkat lunak yang dibutuhkan untuk menunjang keberhasilan terhadap penelitian ini, perangkat lunak yang dibutuhkan adalah :

1. VNC Viewer
2. Python versi 2.7.13

BAB 4 REKAYASA KEBUTUHAN SISTEM

Pada bab ini akan di jelaskan tentang kebutuhan system secara umum. Dimulai dari gambaran system, kebutuhan system dan alur kerja system sehingga system dapat berjalan dengan baik dan sesuai dengan harapan.

4.1 Gambaran Umum Sistem



Gambar 4.1 Blok Diagram Umum Sistem

Gambar 4.1 merupakan diagram blok gambaran umum dari sistem yang akan dibuat. Dimana inputan gambar yang di ambil dari prototype melalui kamera pada sistem tersebut akan dimasukkan ke *Raspberry Pi* yang kemudian akan di proses menggunakan *background subtraction*.

Setelah inputan tersebut dibaca dan di olah menggunakan *background subtraction*, maka akan di proses oleh sistem pada raspberry pi yang kemudian akan menghasilkan output berupa lampu led dan suara dari sound yang berfungsi untuk mengarahkan pengemudi mobil supaya dia bisa tetap antri untuk putar balik atau harus putar balik di rambu berikutnya untuk meminimalisir kemacetan.

Dimana pada kondisi ini jumlah mobil maksimal yang diolah oleh sistem untuk menghasilkan kondisi macet adalah 2 mobil, namun pada kondisi 1 mobil pun bisa di anggap macet saat seluruh badan mobil masuk frame kamera.

4.1.1 Perspektif Sistem

Sistem ini dikatakan bisa berjalan dengan semestinya apabila sesuai dengan yang diharapkan yaitu sistem *background subtraction* mampu mengubah gambar input dari prototype menjadi lampu lalu lintas dinamis yang interaktif kepada pengguna di berbagai situasi seperti pag, siang, dan malam hari.

4.1.2 Karakteristik Pengguna

Pengguna system dalam hal ini adalah semua pengendara terutama kendaraan berjenis mobil, sebelum system beroperasi pengendara akan di hadapkan pada kondisi antrian yang di akibatkan oleh kemacetan pada arah sebaliknya.

4.1.3 Asumsi dan Ketergantungan

Sistem merupakan sistem yang belum terlalu sempurna, masih ada asumsi dan ketergantungan pada sistem ini, yaitu ;

1. Pada sistem yang dibangun kemungkinan tidak bisa mendeteksi dengan baik apabila diujikan pada kondisi malam hari, sehingga dibutuhkan pencahayaan yang baik untuk mendeteksi mobil dengan berbagai warna yg tersedia saat ini.
2. Begitu juga walaupun terdapat pencahayaan yang baik tapi kondisi diluar sedang hujan, maka sistem kami juga masih belum bisa secara sempurna melakukan pemfilteran terhadap kondisi tangkapan gambar dengan *noise* hujan terutama saat hujan lebat.

4.2 Kebutuhan Antarmuka Eksternal

Pada subbab ini akan dibahas kebutuhan antarmuka dari sistem yang akan dirancang dengan ketentuan sebagai berikut :

4.2.1 Kebutuhan Antarmuka Perangkat Keras

1. Lampu LED

Sebagai prototype rambu lalu lintas dinamis, maka lampu yang di gunakan harus sedemikian rupa interaktif ke pengguna jalan agar dapat dimengerti secara cepat, selain itu lampu yang di gunakan untuk rambu lalu lintas ini juga kecerahannya tidak boleh berlebihan agar tidak mengganggu penglihatan pengguna jalan, sehingga digunakan lampu led dengan warna merah dan hijau

dimana bermaksud warna merah untuk berhenti dan hijau bisa di artikan untuk bisa terus jalan.



Gambar 4.2 Lampu led

2. Sound atau speaker

ADVANCE DUO-040 merupakan sebuah speaker yang handal dengan kemampuan memberikan sura yang tajam. Perangkat ini terdiri dari dua buah *speaker* yang bekerja secara maksimal, yang dapat memberikan suara keras, sesuai yang Anda harapkan. Perangkat ini dapat disambungkan ke perangkat komputer dengan menggunakan kabel USB.

Tabel 4.1 Spesifikasi Advance DUO-040

TYPE	DUO-040
DAYA	6W
TEGANGAN	DC 5V
FREKUENSI	50/60Hz

Pada kabelnya, dilengkapi sebuah tombol kontrol, yang berfungsi untuk mengecilkan atau membesarkan volume suara. Selain kualitas suaranya yang baik, perangkat ini pun mendukung fitur plug and play, yang dapat digunakan langsung pada komputer tanpa perlu menginstal program bawaan dari CD, lebih praktis, dan dapat digunakan secara instan.



Gambar 4.3 Gambar ADVANCE DUO-040 (phi, 2016)

3. *Logic shifter*

Logic shifter merupakan salah satu modul yang di gunakan untuk mengubah tegangan dari 3,3v menjadi 5v dimana hal ini karena perbedaan tegangan antara tegangan pada raspberry pi dengan hardware lain, dalam skripsi ini penulis menggunakan *logic shifter* untuk menghubungkan raspberry phi 3 dengan lampu LED merah dan hijau dimana lampu LED ini digunakan sebagai *prototype* lampu lalu lintas dinamis.

4. Kamera Logitech c270h

Kamera Logitech adalah salah satu kamera yang cukup simple jika di gunakan, dimana pada kamera Logitech memiliki spesifikasi dan bentuk yang cukup bagus, selain itu juga kamera Logitech c270 cukup murah di banding kamera modul dari raspberry pi nya sendiri, untuk fiturnya pun cukup lengkap seperti :

1. Bisa melakukan panggilan video dengan resolusi 1280x720 pixels
2. Tangkapan video juga sama yaitu bisa sampai resolusi 1280x720 pixels
3. Kualitas foto bisa sampai 3.0 megapixels
4. USB sudah kecepatan tinggi dengan port 2.0 dan sudah *recommended*
5. Terdapat fitur *face tracking*
6. *Built-in mic with RightSound* dimana suara yang berkualitas dan bisa mengurangi *noise*
7. *RightLight technology* berguna juga dimana pencahayaan bisa diatur secara manual atau otomatis

6. Motion detection, dan masih banyak lagi.



Gambar 4.4 Gambar LOGITECH C 270H (phi, 2016)

sehingga dengan berbagai fitur di atas penulis menilai kamera Logitech c270h cukup untuk menunjang penelitian ini.

5. Raspberry Pi 3

Raspberry Pi 3 merupakan generasi ketiga dari keluarga Raspberry Pi. Raspberry Pi 3 memiliki RAM 1GB dan grafis Broadcom VideoCore IV pada frekuensi clock yang lebih tinggi dari sebelumnya yang berjalan pada 250MHz. Raspberry Pi 3 menggantikan Raspberry Pi 2 model B pada bulan Februari 2016. Untuk spesifikasi lengkap ada pada tabel berikut

Tabel 4.2 Spesifikasi Raspberry pi 3 (Raspberry Pi, 2016)

SoC	Broadcom BCM2837
CPU	4x ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1 GB LPDDR2 (900 MHz)
Networking	10/100 Ethernet, 2.4 GHz 802.11n wireless
Bluetooth	Bluetooth 4.1 Classic, Bluetooth Low Energy
Storage	MicroSD
GPIO	40-pin header, populated

Ports	HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), display Serial Interface (DSI)
--------------	---



Gambar 4.5 Gambar raspberry phi 3 model B (phi, 2016)

4.2.2 Kebutuhan Antarmuka Perangkat Lunak

1. VNC Viewer

Virtual network computing (VNC) adalah software remote-control yang memungkinkan untuk mengontrol komputer lain melalui koneksi network. Pencetakan keyboard dan mouse click dikirimkan dari satu komputer ke komputer lainnya sehingga seseorang dapat mengelola sebuah dekstop, server dan alat yang terhubung jaringan tanpa harus di lokasi yang sama.

VNC bekerja pada model client/server. Sebuah VNC Viewer (atau client) diinstall pada komputer lokal dan dihubungkan dengan server yang harus diinstall di komputer remote. Server mengirim duplikasi dari display komputer remote ke viewer (client). Server juga menerjemahkan command dari viewer dan menerapkannya pada komputer remote.

VNC adalah platform independent dan compatible dengan operating system apapun (aliridhos, 2013). Komputer harus berada di jaringan TCP/IP dan memiliki port yang terbuka untuk traffic dari IP address suatu alat yang akan mengontrol.

4.3 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang harus dipenuhi dalam sebuah sistem. Berikut beberapa kebutuhan fungsional dari sistem yang harus berfungsi agar dapat bekerja dengan baik.

1. Sistem dapat mengambil nilai dari prototype mobil

Prototype mobil akan di jalankan dan di berhentikan, dimana batas sudah di tentukan sebelumnya yaitu untuk mendapatkan kondisi macet maka batas mobil yang bisa masuk kedalam frame maksimal 2 mobil dan hasilnya akan di input ke sistem. Kemudian sistem akan mengolah dan mendeteksi tiap mobil yang lewat apakah mobil tersebut berhenti atau tidak, dan saat berhenti maka akan di lanjutkan ke proses *background subtraction* dan waktunya di hitung yang kemudian akan Kembali di proses untuk dapat mengeluarkan output rambu lalu lintas dinamisnya.

2. Sistem dapat mengolah data dari inputan prototype dengan menggunakan *background subtraction*

Sistem pengolahan dapat berjalan ketika mendapatkan input dari prorotype yang masuk ke sistem. Gambar yang tertangkap kamera akan di proses menggunakan *background subtraction* yang selanjutnya akan di mengeluarkan pernyataan bahwa jalan termasuk ke dalam kategori macet atau lancar yang kemudian akan di keluarkan juga dalam bentuk lampu LED dan *text to speech*

3. Sistem dapat mengolah inputan berupa protype mobil menjadi rambu lalu lintas dinamis menggnakan *background subtraction*

Dengan menggunakan metode *background subtraction* hasil tangkapan kamera dimana dalam penelitian ini tangkapan kamera berupa prototype mobil akan di bedakan kedalam kategori *background* dan *foreground* dimana jalan raya yang diam di anggap sebagai *background* dan mobil yang bergerak di anggap sebagai *foreground* dan pengolahan ini akan di ditampilkan berupa rambu lalu lintas dinamis dalam bentuk lampu LED merah dan hijau.

4. Sistem dapat menyesuaikan output berupa suara dari hasil proses *background subtraction* yang diubah menjadi *text to speech*

Data inputan berupa gambar yang telah diubah oleh metode *background subtraction* menjadi tulisan, data tersebut akan melalui proses selanjutnya yang merupakan proses terakhir, yaitu proses *text to speech*. Pada proses ini, gambar yang telah diubah menjadi tulisan akan dibaca oleh sistem *text to speech* sehingga akan mengeluarkan suara, yang merupakan output terakhir yang diinginkan oleh penulis

4.4 Kebutuhan Non Fungsional

Pada sub bab ini dijelaskan beberapa kebutuhan non fungsional pada sistem yang akan dirancang, yaitu ;

1. Tegangan listrik yang sesuai dengan kebutuhan

Pada sistem ini dibutuhkan tegangan maksimal $5\text{ V} = 2\text{ A}$ yang diambil dari terminal listrik yang akan dihubungkan dengan *raspberry pi* guna untuk daya utama supaya bisa menyala

2. Output dari sistem yang sesuai dengan input

Sistem dapat berjalan dengan lancar dan dapat menghasilkan output yang bagus ketika sesuai metode *background subtraction* yang telah ditentukan kondisinya. Dan perhitungan data bisa akurat sesuai dengan jumlah *sample* yang dihitung dan jumlah kesuksesan metode *background subtraction* dalam mengubah gambar menjadi tulisan.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

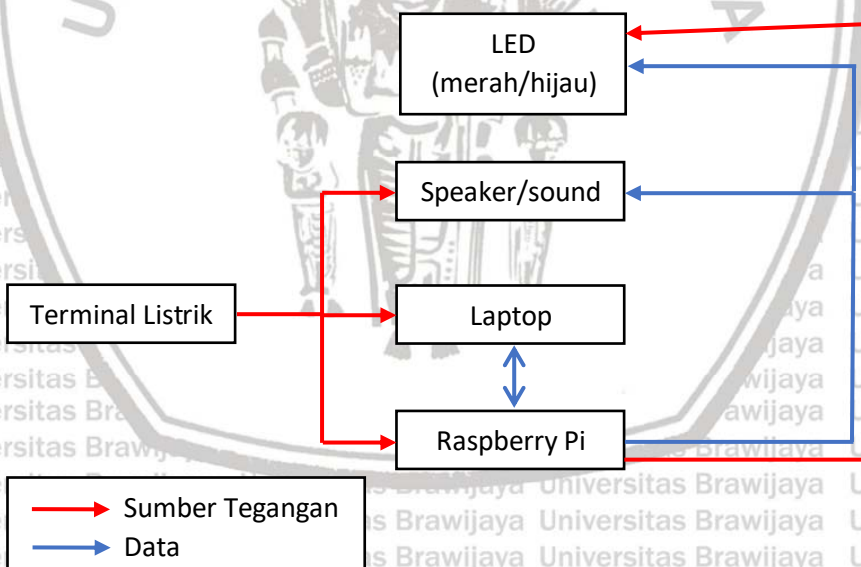
Pada bab ini menjelaskan mengenai proses perancangan dan implementasi sistem secara terperinci baik perancangan dan implementasi dari sisi perangkat keras maupun dari sisi perangkat lunak

5.1 Perancangan Sistem

Berdasarkan metode penelitian diatas, perancangan sistem dilakukan setelah melakukan analisa kebutuhan sistem. Perancangan penelitian ini dilakukan untuk merencanakan tahapan-tahapan yang harus dilakukan dalam proses melakukan penelitian. Tahapan penelitian yang harus dilakukan adalah perancangan perangkat lunak dan perangkat keras. Diharapkan dengan dilakukannya tahapan perancangan penelitian dapat memperlancar proses pelaksanaan penelitian.

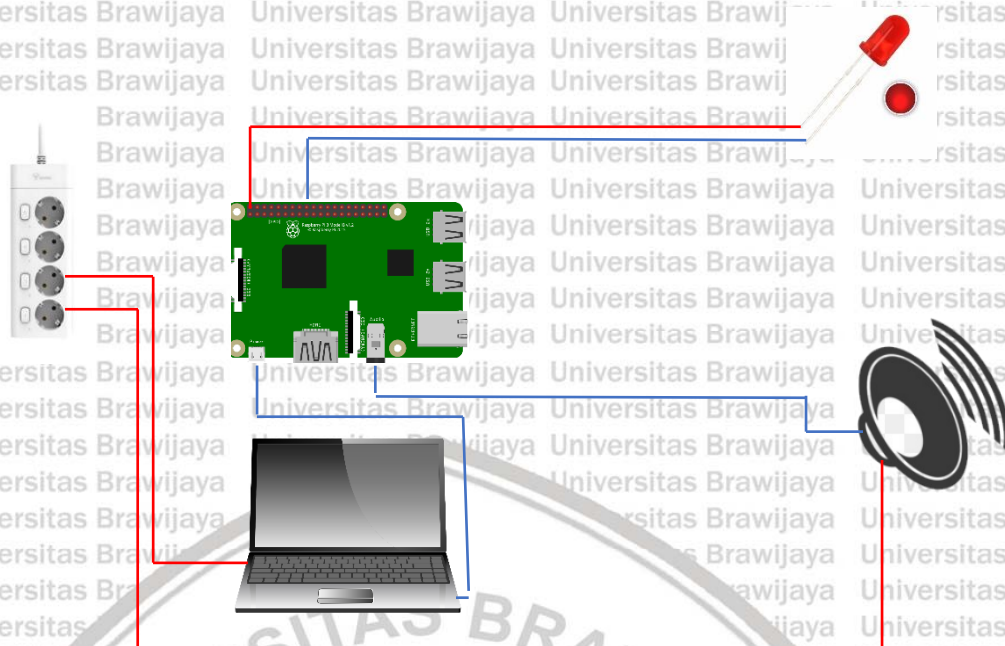
5.1.1 Perancangan Perangkat Keras

Perancangan perangkat keras keseluruhan pada sistem ini dapat dilihat pada diagram blok gambar 5.2 berikut ini :



Gambar 5.1 Blok Diagram Perangkat Keras

Pada diagram blok Gambar 5.2, garis merah merupakan sumber tegangan yang berfungsi sebagai jalur tenaga agar bisa menyala, sementara garis biru merupakan jalur transmisi data dari satu perangkat ke perangkat yang lain.



Gambar 5.2 Sistematika Perangkat Keras

Skematik keseluruhan tentang perangkat keras dapat dilihat pada Gambar 5.3. Kabel berwarna merah mewakili aliran arus daya tegangan yang dibutuhkan oleh perangkat untuk menyalakan perangkat tersebut, khusus untuk arus aliran listrik yang ke arah *raspberry pi* mempunyai batas $5\text{ V} == 2\text{ A}$, jika melebihi itu akan terjadi konslet dan bisa menyebabkan perangkat meledak. Sedangkan kabel berwarna biru mewakili aliran data dari satu perangkat ke perangkat yang lain.

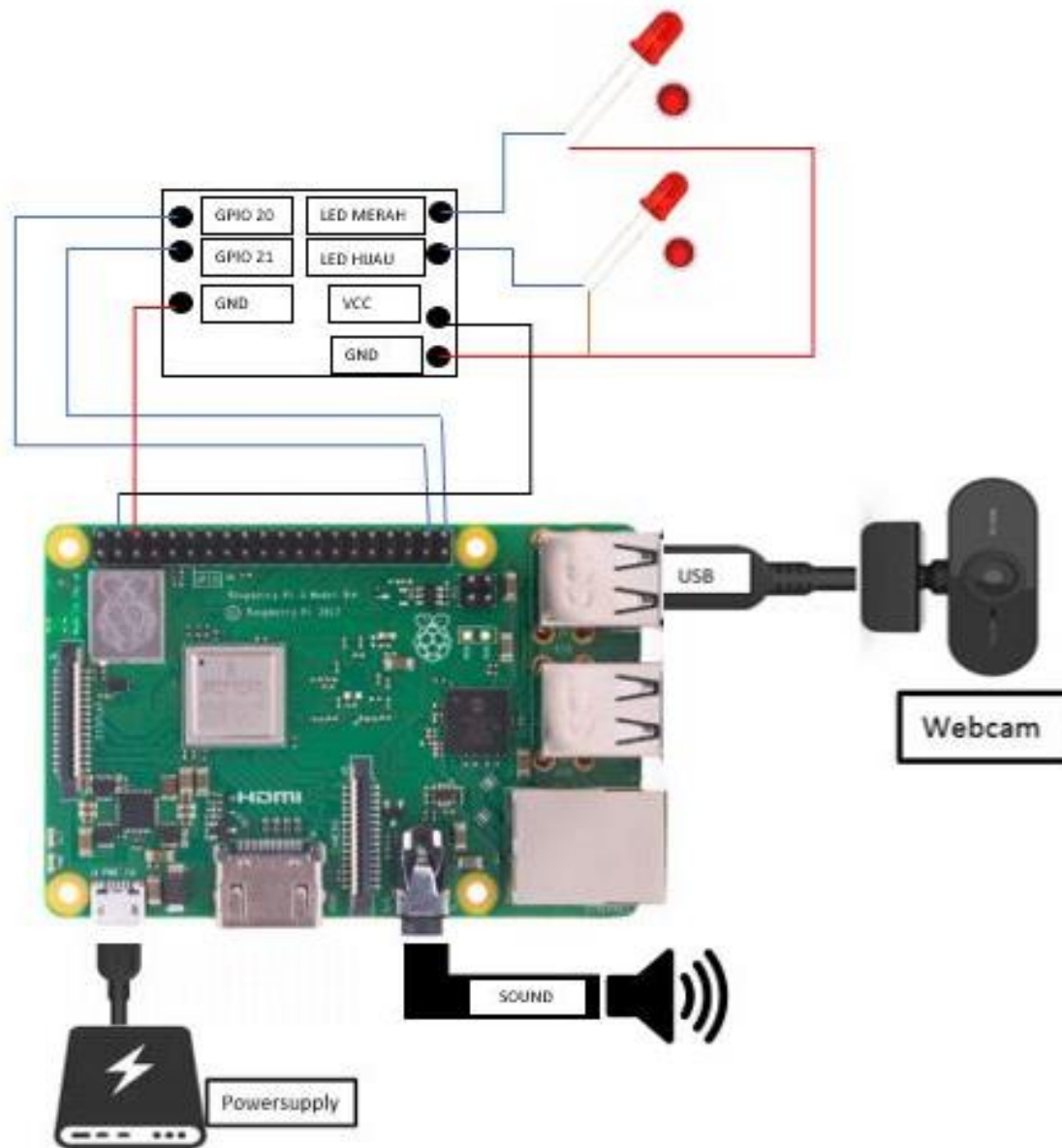
Dapat di perhatikan pada skema gambar di atas bahwa perangkat keras *raspberry pi*, laptop, LED dan juga speaker mendapatkan arus listrik langsung dari terminal listrik. Sedangkan untuk arus aliran data, data berawal dari laptop yang disitu terdapat inputan data berupa gambar dari prototype. Kemudian gambar tersebut akan di proses di *raspberry pi* dan akan di baca oleh metode *background subtraction*. Setelah itu data yang sudah melewati tahap metode *background subtraction* akan dikirim ke speaker untuk mengeluarkan data output berupa suara, selain itu data yang di olah juga di salurkan ke lampu LED dimana dalam hal ini lampu LED berfungsi sebagai rambu lalu lintas dinamis.

Raspberry pi 3 model B memiliki sebanyak 40 pin yang memiliki fungsi yang berbeda digunakan dalam membangun sistem. Pada sistem deteksi ini pin raspberry berguna untuk menghubungkan bagian proses ke komponen *output* yaitu lampu led dan sound . Pin yang digunakan yaitu pin GPIO 21 untuk led hijau dan GPIO 20 untuk led merah , power dan ground. Pin GPIO (*General purpose Input/output*) merupakan pin untuk input sebuah fungsi yang diberikan logika dalam proses . Data untuk pin raspberry dapat dilihat pada Tabel 5.1 dibawah

Tabel 5.2 Deskripsi pin raspberry Pi (www.raspberry.org, 2020)

Deskripsi	Nomor Pin		Deskripsi
VCC 3 volt (<i>Power</i>)	1	2	VCC 5 volt (<i>Power</i>)
GPIO2	3	4	VCC 5 volt (<i>Power</i>)
GPIO3	5	6	Ground
GPIO4	7	8	GPIO14
Ground	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	Ground
GPIO22	15	16	GPIO23
VCC 3 volt (<i>Power</i>)	17	18	GPIO24
GPIO10	19	20	Ground
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
Ground	25	26	GPIO7
GPIO8	27	28	GPIO1
GPIO5	29	30	Ground
GPIO6	31	32	GPIO12
GPIO13	33	34	Ground
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
Ground	39	40	GPIO21

Untuk design perancangan *hardware* sistem rambu lalu lintas dinamis dapat terlihat pada ilustrasi Gambar 5.2 dibawah :



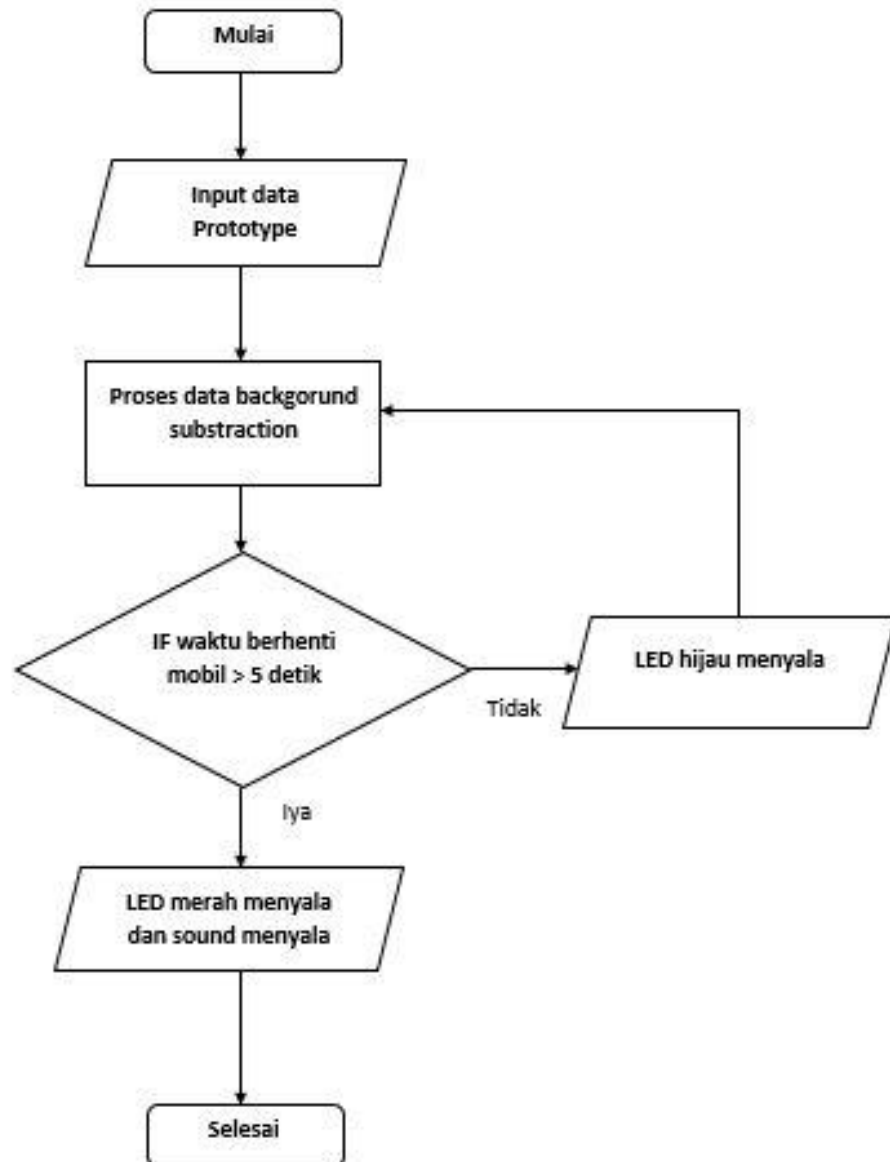
Gambar 5.3 Rangkaian komponen penyusun sistem

Adapun pin yang digunakan pada *raspberry pi* disambungkan ke perangkat lain untuk mendapatkan hasil output yang diinginkan, pin yang digunakan adalah sebagai berikut :

1. Micro USB power input, digunakan untuk mendapatkan daya supaya *raspberry pi* bisa menyala dan digunakan
2. 3.5 mm 4-pole composite video and audio output jack, digunakan untuk disambung ke speaker untuk mengeluarkan suara dari hasil pemrosesan
3. Pin 20 dan 21 pada *raspberry pi* yang di gunakan untuk menyalakan lampu led merah dan hijau.

5.1.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak digunakan untuk menjelaskan proses-proses yang berjalan pada perangkat lunak. Dimulai dari memberikan *input*, membaca data kamera, mengumpulkan data, pengolahan background subtraction, hingga menampilkan hasil.



Gambar 5.4 Diagram alir perancangan keseluruhan sistem

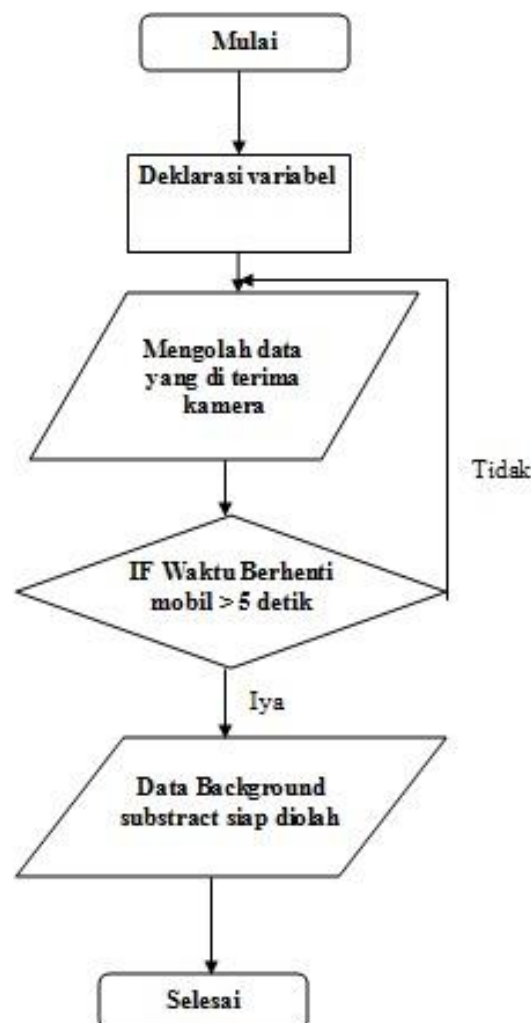
Pada Gambar 5.3 diatas adalah diagram alir keseluruhan sistem. Dari gambar diatas dapat dilihat bahwa agar sistem dapat berjalan sesuai dengan yang diharapkan, maka langkah pertama yang harus dilakukan yaitu memberikan *input* pada sistem berupa replika mobil pada prototype. Dengan adanya input tersebut maka sistem akan melakukan proses pembacaan data oleh kamera yang

digunakan yaitu kamera Logitech c270H. Setelah data didapatkan maka nilai-nilai tersebut akan di kumpulkan dan di olah menggunakan metode background subtraction dimana pengambilan data di ambil dari selisih waktu replika mobil saat berhenti. Tahap terakhir yaitu sistem dapat menampilkan hasil pengolahan gambar pada dekstop yang selanjutnya *output* juga berupa lampu LED dan suara pada Sound. Untuk penjelasan lebih detail dapat dilihat pada subbab di bawah ini.

5.1.2.1 Memberikan Input

Input yang digunakan pada penelitian ini yaitu replika mobil, ukuran replika mobil yang di gunakan adalah Panjang 7 cm dan lebar mobil 3 cm dengan asumsi ukuran mobil memakai skala 1:54,33cm.

5.1.2.2 Perancangan Pengambilan Data Kamera



Gambar 5.5 Diagram alir perancangan pengambilan data dari kamera

Proses perancangan perangkat lunak yang ditunjukkan oleh gambar 5.4 diatas bertujuan untuk menentukan hasil pengolahan dari kamera yang nantinya akan di olah oleh raspberry phi menggunakan metode background subtraction,karena disini mikrokontroller melakukan pengolahan data yang di terima dari kamera secara terus menerus atau realtime.

A. Mulai, sistem mulai melakukan pengambilan dan pengiriman data.

B. Deklarasi library,dan variable yang di gunakan.

a) Deklarasi library, mendeklarasikan library yang digunakan pada sistem. Library yang digunakan pada sistem ini yaitu library numpy as np, imutils, time,cv2 dimana penjelasan setiap *library* akan dijelaskan pada bab selanjutnya.

b) Deklarasi variable, mendeklarasikan besaran pixel yang akan di gunakan dimana pixel yang digunakan pada sistem ini berukuran 2000 pixel dan variable array untuk menyimpan nilai yang di olah dengan metode background subtraction.

C. Mengolah data yang di terima kamera, dalam sistem ini terdapat 1 kamera yaitu Logitech c270H.

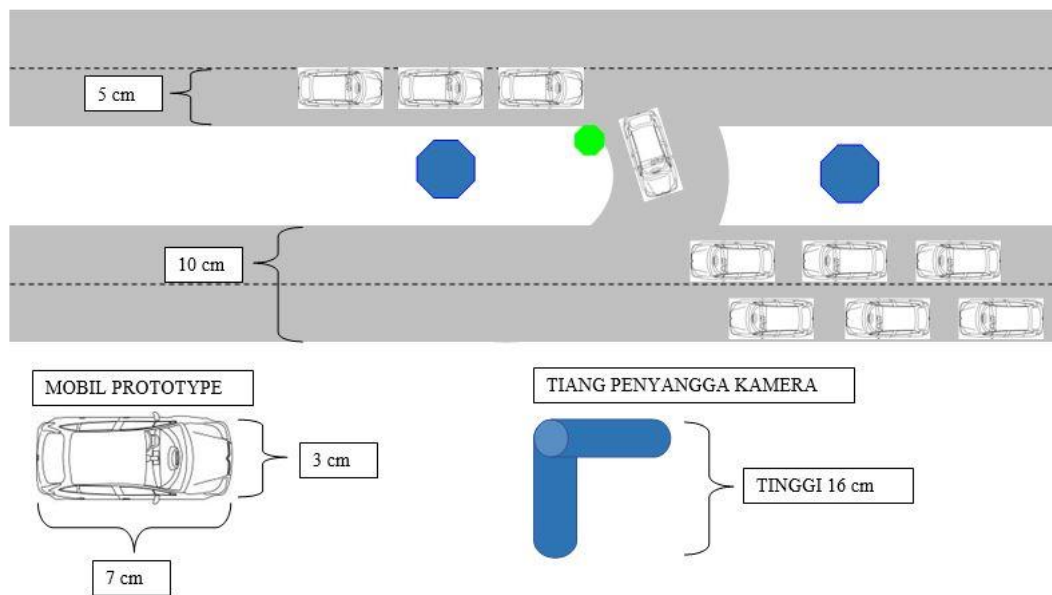
D. If mobil waktu berhentinya kurang dari 5 detik maka kondisi jalan raya di area putar balik di nilai lancar oleh sistem,tapi saat ada mobil yang berhenti di area putar balik lebih dari 5 detik maka area tersebut dinilai macet oleh sistem

E. Data background subtraction siap diolah,setelah mendapatkan nilai terakhir mengenai koordinat sumbu x dan y maka data tersebut di gunakan menggunakan metode background subtraction.

F. Selesai, sistem telah selesai melakukan proses pembacaan dan pengiriman data.

5.1.2.3 Perancangan prototype alat

Dalam Melakukan perancangan prototype alat untuk sistem pengaturan rambu lalu lintas (u-turn) ini perlu di perhatikan ukuran alat yang akan di buat dan peletakan masing-masing komponen.sistem ini akan dirancang pada lembaran kertas kardus dan kertas karton putih yang di beri warna hitam sebagai simulasi jalan raya dengan skala 1:54,33cm,penggunaan ukuran tersebut di maksudkan agar *prototype* bersifat *portable* namun tidak menghilangkan kondisi nyata yang ada pada jalan raya saat ini,selain itu skala yang di gunakan 1:54,33cm supaya tidak terlalu banyak space yang ada pada prototype ini,untuk menggambarkan rancangan pembuatan sistem ini,penulis menggunakan Microsoft word 2016,sebagaimana bentuk prototype alat di tunjukkan pada gambar 5.1 berikut .



Gambar 5.6 Desain Prototype Alat

Mobil prototype yang di gunakan berada di bawah tiang kamera, dan cakupan kamera hanya di batasi 2 baris mobil, yang fungsinya untuk mengatasi penumpukan barisan mobil yang ingin putar balik atau untuk menentukan indikator macet atau tidak nya ruas jalan raya yang sudah di tentukan sebelumnya, selain itu tinggi tiang juga di sesuaikan dengan perhitungan program yang gunanya untuk memaksimalkan deteksi mobil yang berada di area cakupan kamera tersebut.

5.1.3 Implementasi System

Implentasi sistem merupakan tahap untuk mengimplementasikan pembuatan sistem berdasarkan semua perancangan yang sudah dilakukan sebelumnya. Pada subbab ini akan dijelaskan secara rinci terkait implementasi *prototype* alat, implementasi perangkat keras, hingga implementasi perangkat lunak.

5.1.3.1 Implementasi Perangkat Keras

Dalam mengimplementasikan perangkat keras sistem, pengaturan rambu lalu lintas (u-turn) mencakup beberapa komponen, antara lain yaitu *raspberry pi* yang berfungsi sebagai otak pada sistem ini, sound card yang berfungsi sebagai penghubung antara pemrosesan sistem yang sudah selesai dengan perangkat yang akan menjadi *output*, *speaker* yang berfungsi sebagai mengeluarkan hasil dari pemrosesan sistem yang telah dibuat, dan lampu led juga yang berfungsi sebagai *output* sistem yang telah di buat sebagai rambu lalu lintas dinamis, lalu semua komponen tersebut

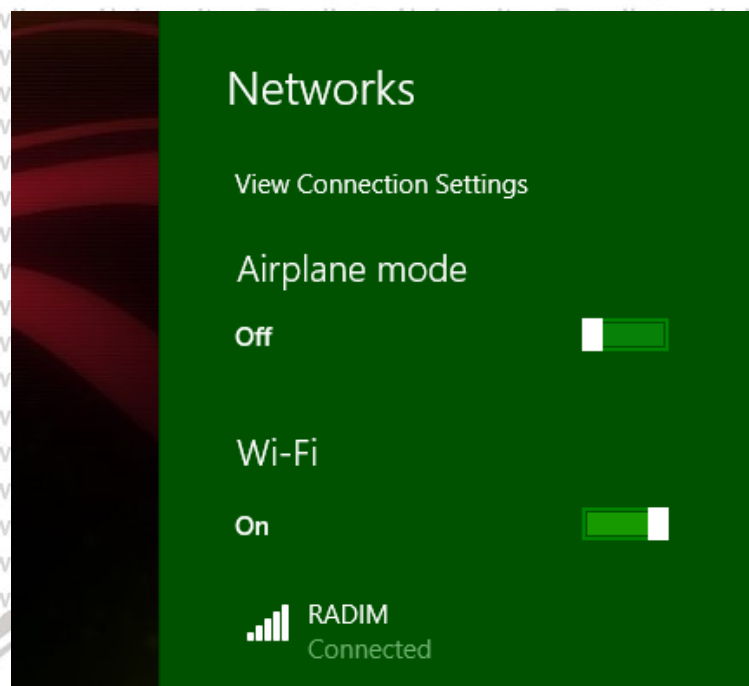
dihubungkan sesuai dengan rancangan yang sudah di jelaskan pada subbab 5.1.2.

Rangkaian yang pertama adalah perangkat *raspberry pi* yang akan disambungkan dengan laptop. Harus di sambungkan dengan laptop karena disini laptop sebagai perangkat yang memberi perintah awal kepada vnc dan juga sebagai display dari *raspberry pi* itu sendiri



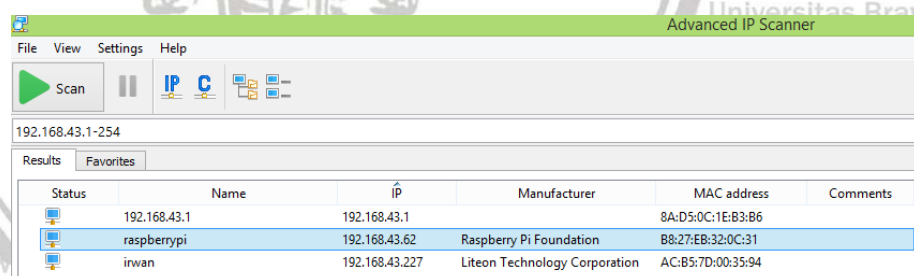
Gambar 5.7 Menyambungkan Raspberry pi dengan Laptop

Pada gambar di atas dapat kita lihat *raspberry pi* di hubungkan dengan laptop, dimana penulis menggunakan jaringan wifi berupa *hotspot* untuk *raspberry pi* agar bisa terhubung ke laptop, dimana sebelumnya *raspberry pi* harus diatur atau di setting supaya *raspberry pi* dan laptop berada pada 1 jaringan wifi *hotspot*



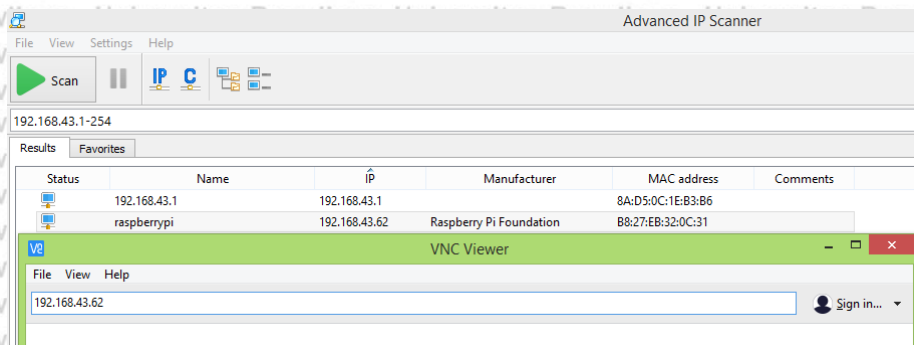
Gambar 5.8 Memeriksa jaringan wifi

Pada gambar di atas adalah nama jaringan wifi *hotspot* dimana antara laptop dan *raspberry pi* harus berada pada 1 jaringan tersebut, karena jika beda jaringan maka vn tidak akan bisa terhubung dengan *raspberry pi*, pada penelitian ini penulis menggunakan jaringan wifi dengan nama RADIM



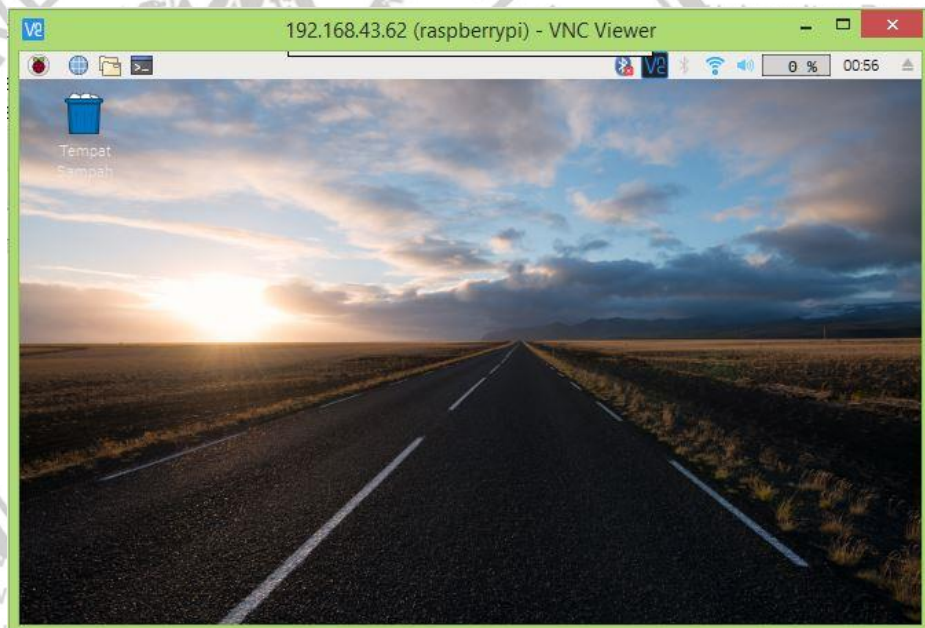
Gambar 5.9 Memeriksa ip raspberry pi

Setelah *raspberry pi* dan laptop sudah masuk kedalam jaringan wifi yang sama, maka penulis cek apakah *raspberry pi* sudah benar-benar masuk kedalam wifi yang sama atau tidak, disini penulis menggunakan aplikasi pc bernama *ip scanner*, dengan aplikasi ini penulis dapat mengetahui apakah *raspberry pi* sudah terhubung dan untuk cek juga ip dari *raspberry pi* tersebut



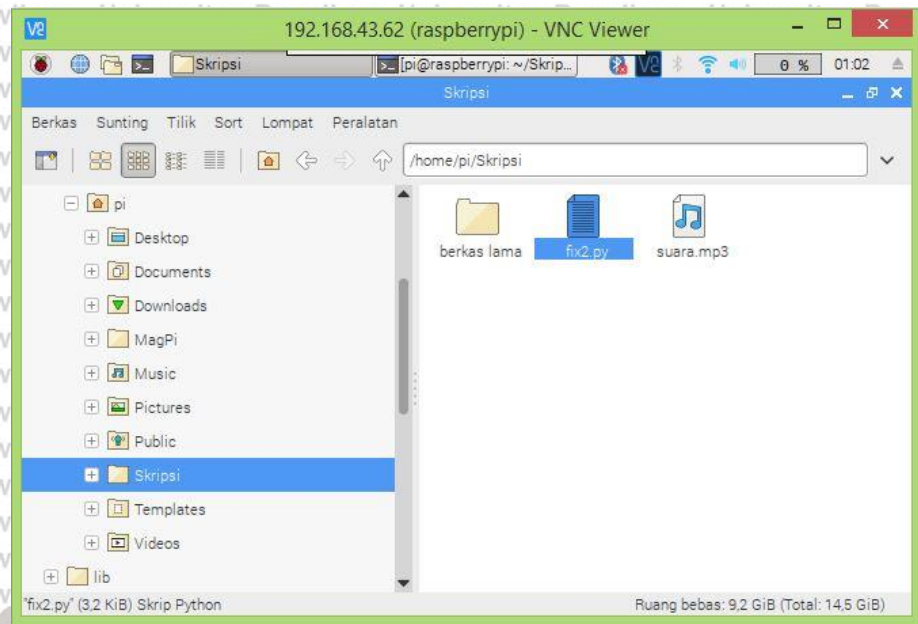
Gambar 5.10 Log in vnc viewers

Lalu, setelah ip dari *raspberrypi* sudah di dapatkan maka selanjutnya penulis buka *vnc viewers* dimana *vnc* ini di gunakan untuk dapat menghubungkan juga antarmuka laptop dengan antarmuka *raspberrypi*, lalu jika *vnc viewers* sudah terbuka penulis memasukkan ip *raspberrypi* yang sebelumnya sudah dapat di ketahui dari aplikasi *ip scanner*



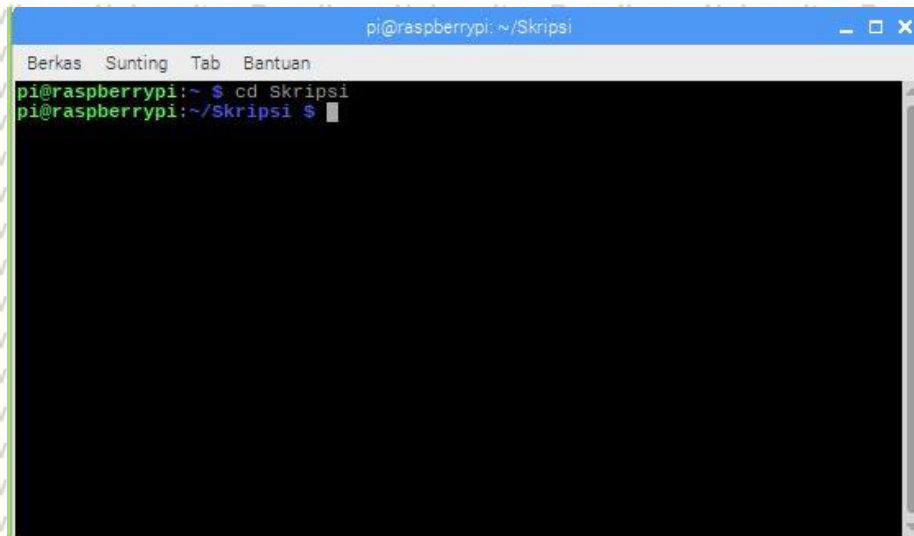
Gambar 5.11 Tampilan remote raspberrypi

Saat ip pada *raspberrypi* sudah di masukkan kedalam *vnc connect* maka tampilan remote laptop akan seperti gambar 5.11 di atas, dimana ini adalah tampilan default pada *raspberrypi 3b*, selanjutnya hanya perlu mencari file folder tempat coding



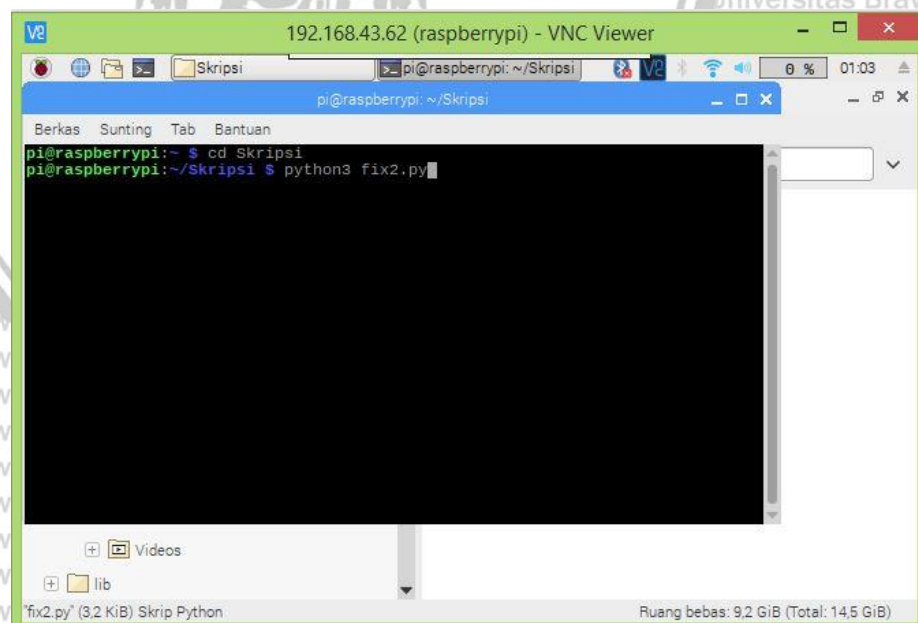
Gambar 5.12 Memastikan tempat file coding sebelum program di jalankan

Pada gambar 5.12 di atas menunjukkan lokasi atau tempat file coding, dimana file berada pada folder pi dan oleh penulis di beri nama skripsi, dimana di dalam folder skripsi terdapat *source code* yang akan di jalankan bernama fix2.py dan file suara.mp3 yang nantinya file suara ini di gunakan jika program mendeteksi kemacetan pada prorotype, dan file *source code* dan suara harus berada pada 1 folder karena saat program di jalankan dan menjalankan file suara maka yang di panggil oleh program yang berada pada 1 file dengan *source code* yang di jalankan



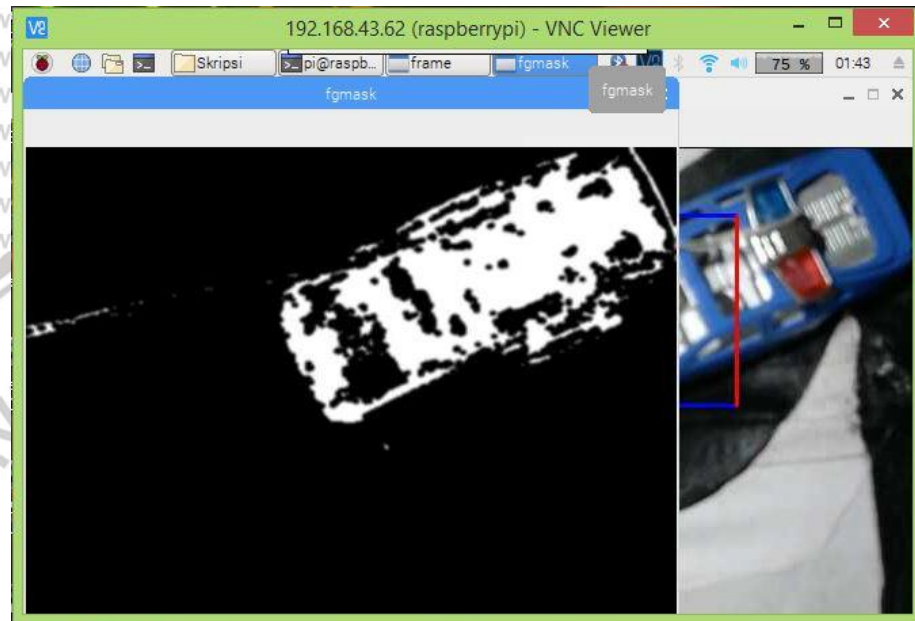
Gambar 5.13 Memindah direktori

Pada saat file source code sudah di letakkan pada folder pi subfolder skripsi maka selanjutnya menjalankan terminal dengan cara klik lambang terminal di tampilan layar atas, pada gambar 5.13 terlihat bahwa direktori awal berada pada pi, maka untuk masuk ke direktori skripsi maka diperlukan perintah `cd skripsi` supaya bisa masuk ke direktori skripsi dan menjalankan *source code* nya dan file lainnya seperti suara.mp3



Gambar 5.14 Menjalankan kode program

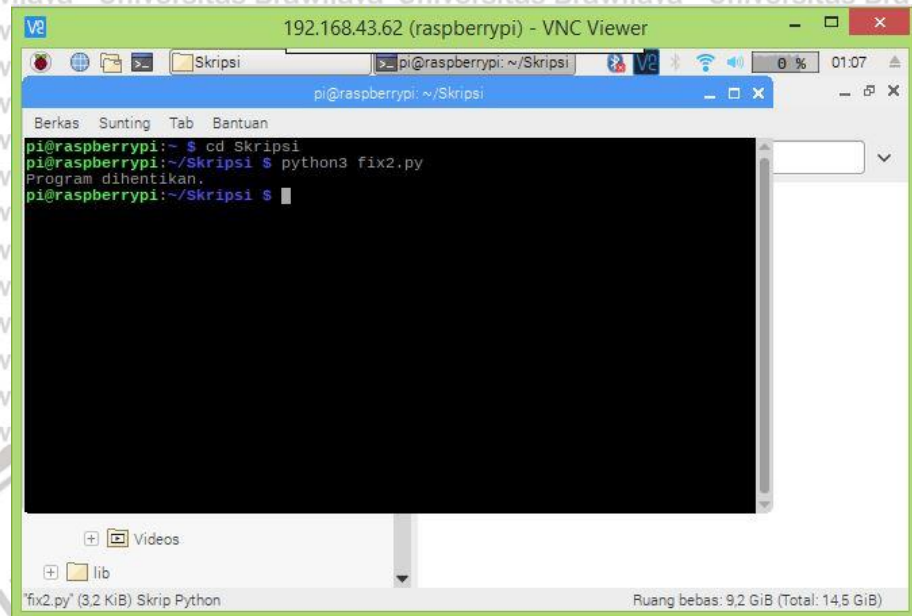
Setelah masuk ke direktori skripsi tempat source code ditempatkan maka selanjutnya menjalankan source code nya, dimana disini penulis menggunakan python sebagai bahasa pemrograman dengan versi python ke 3, jadi pada saat run program maka harus diawali dengan python3 dan diikuti oleh nama file nya, pada gambar 5.14 terlihat menjalankan source code dengan cara mengetikkan python3 fix2.py dimana sudah dijelaskan juga pada gambar sebelumnya bahwa fix2.py merupakan nama file source code nya program pada rambu lalu lintas dinamis ini



Gambar 5.15 Tampilan *foreground mask* dan frame saat program di jalankan

Pada saat program di jalankan maka akan menampilkan fgmask dan frame dimana fgmask menampilkan background subtraction dari program Perancangan *Prototype* Sistem Untuk Pengaturan Rambu Putar Balik (U-Turn) Dengan Metode *Background Subtraction* Berbasis *Raspberry Pi* ini, dan tampilan bernama frame menampilkan frame yang nantinya digunakan untuk mendeteksi objek apakah nantinya dengan adanya objek tersebut maka keadaan dianggap macet atau lancar oleh program, dimana pada tampilan frame ada 2 warna garis yaitu merah dan hijau dimana 2 garis ini nantinya digunakan untuk deteksi masuk dan keluarnya objek dengan selisih waktu yang sudah ditentukan sebelumnya yaitu 5 detik, maka saat objek berupa *prototype* mobil terdeteksi di garis merah dan selama 5 detik belum terdeteksi di garis hijau maka kondisi area putar balik dianggap macet, dan sound mengeluarkan suara yang berisi perintah bahwa pengendara di luar area putar balik bisa melakukan putar balik di

rambu berikutnya supaya tidak terjadi penumpukan kendaraan yang mengakibatkan kemacetan di salah satu sisi jalan.



Gambar 5.16 Tampilan terminal saat program selesai di jalankan

Setelah program selesai di jalankan maka tampilan pada terminal akan seperti gambar 5.16 dimana pada terminal keluar tulisan program di hentikan,yaitu jika penulis menekan tombol Esc pada keyboard laptop.

5.1.3.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak pada subbab ini menjelaskan proses implementasi sistem pengaturan rambu lalu lintas (u-turn). Untuk dapat mengimplementasikan perangkat lunak ini, terdapat software python dan vnc viewers.

5.1.3.3 Implementasi Pemberian Input

Input yang di gunakan adalah prototype mobil atau mobil mainan yang ukurannya sudah di sesuaikan dengan kondisi mobil atau ukuran mobil yang asli dengan skala 1:54,33cm, selain itu *input* berupa mobil mainan ini juga dilakukan dengan berbagai macam warna mobil untuk mengetahui seberapa jauh error program terhadap warna kendaraan yang berbeda pada saat di kondisi nyata nantinya.



Gambar 5.17 Ukuran replika mobil

5.1.3.4 Implementasi Kode Program

Dalam melakukan implementasi untuk membaca nilai dari kamera, maka perlu di lakukan deklarasi variable berdasarkan perancangan perangkat keras serta perancangan perangkat lunak yang telah di jelaskan sebelumnya. Karena proses pembuatan kode program sepenuhnya dilakukan di python 3.8 maka pada awal program dilakukan deklarasi library terlebih dahulu, selanjutnya implementasi kode program pengambilan data yang di tangkap oleh jangkauan kamera.

Tabel 5.2 Kode program deklarasi library system

Baris	Kode program
1	<code>import numpy as np</code>
2	<code>import imutils</code>
3	<code>import time</code>
4	<code>import cv2</code>

Deklarasi kamera perlu di lakukan dalam membuat sebuah program. Hal ini dimaksudkan agar memudahkan dalam melakukan pemrograman beberapa fungsi tertentu. Pada tabel 5-6 menunjukkan pengimplementasian *library* pada sistem ini, yang pertama adalah *library* "numpy as np" yang memiliki kegunaan untuk operasi vector dan matriks dimana jika pada matlab fungsinya hampir sama seperti mengelola array dan arra multidimensi, yang kedua adalah *library* "imutils" dimana *library* ini memiliki fungsi untuk proses *resize*, yang ketiga *library* "time" dimana

library ini digunakan untuk mengolah perhitungan waktu yang nantinya di gunakan untuk menghitung lama waktu miniatur mobil berhenti pada prototype jalan raya, yang keempat adalah library “cv2” dimana pada library ini digunakan untuk membaca gambar pada open cv karena program ini di jalankan dan pada open vc pada raspberry phi v3.

Tabel 5.3 Kode program deklarasi variabel

Baris	Kode program
1	waktu = 5
2	s_start = (10,300)
3	s_end = (230,340)
4	start_point = (50,50)
5	end_point = (350,190)
6	rect_color = (255,0,0)
7	inx = start_point[0]
8	iny = start_point[1]
9	inw = end_point[1] - iny
10	outx = end_point[0]
11	outy = start_point[1]
12	outw = end_point[1] - outy
13	minArea = 2000
14	inEn = 1
15	income = 0
16	outcome = 0
17	timerIn = 0
18	timerOut = 0
19	status = "Lancar"
20	currentTime = 0
21	prevTime = 0
22	fgbg = cv2.createBackgroundSubtractorMOG2(history=100,detectShadows=True)
23	kernelA = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5))

Setelah deklarasi *library* selesai dilakukan, maka langkah selanjutnya yaitu melakukan deklarasi variabel. Tabel 5.7 menunjukkan bahwa pada pada baris ke-1 merupakan deklarasi variable yang di gunakan untuk menentukan batas waktu sebuah kendaraan dimana dalam prototype ini bertujuan untuk menentukan kondisi area rambu lalu lintas apakah macet atau ancar dengan batas berhenti 5 detik, dimana saat sebuah kendaraan atau objek lain berhenti di atas waktu 5 detik maka objek yang menutupi frame akan di nlai kedalam kategori macet. Pada tabel 5.7 juga di baris ke-2 hingga baris ke-12 juga penulis mendekalarasikan variable dimana variable-variabel ini bertujuan untuk mengatur nilai frame seperti warna garis, Panjang dan lebar frame dimana ini sangat penting pada penelitian ini karena untuk menentukan cakupan pandangan kamera yang nantinya akan di olah ke dalam metode background subtraction itu sendiri sehingga input bisa sesuai dengan harapan yang di inginkan. Selanjutnya pada baris ke-13 hingga baris ke-21 dimana deklarasi variable di tujukan untuk perhitungan counter waktu dan output yang di gunakan

Tabel 5.4 Kode program deklarasi variabel konfigurasi pin output

Baris	Kode program
1	lampu_hijau = 21
2	lampu_merah = 20
3	GPIO.setmode(GPIO.BCM)
4	GPIO.setwarnings(False)
5	GPIO.setup(lampu_hijau, GPIO.OUT)
6	GPIO.setup(lampu_merah, GPIO.OUT)
7	GPIO.output(lampu_hijau, GPIO.HIGH)
8	GPIO.output(lampu_merah, GPIO.LOW)
9	cam = cv2.VideoCapture(0)
10	pygame.mixer.init()
11	pygame.mixer.music.set_volume(0.35)

Setelah deklarasi *library* counter dan waktu selesai dilakukan, maka langkah selanjutnya yaitu melakukan deklarasi konfigurasi pin output. Tabel 5.8 menunjukkan deklarasi variable untuk output sebagai contoh pada baris ke-1 dan baris ke-2 dimana variable bernama lampu_hijau dan

lampu_merah di buat untuk mendeklarasikan pin 20 dan 21 pada raspberry pi yang nantinya di salurkan kepada lampu LED merah dan hijau sebagai rambu lali lintas dinamis, selain itu pada baris ke-3 sampai baris ke-6 instruksi yang di gunakan untuk memberi perintah pada LED dimana salah satu lampu harus menyala yaitu nyala hijau atau nyala merah sebagai penanda pengendara bahwa bisa melakukan putar balik atau tidak, selain itu tidak lupa juga penulis deklarasikan port kamera sebagai input yaitu pada baris ke-9 dimana port kamera ini digunakan untuk jalur komunikasi memasukkan data melalui kamera kepada raspberry pi, dan juga deklarasi speaker sebagai pengingat pengendara juga selain rambu lalu lintas dinamis, dimana terlihat pada baris ke-10 dan ke-11 bahwa deklarasi variabel dilakukan untuk menyiapkan speaker dimana speaker nanti diatur volumenya juga sebesar 0,35

Tabel 5.5 Kode program pengolahan background subtraction dan output rambu lalu lintas dinamis

Baris	Kode program
1	if timerIn == 1:
2	currentTime = time.time()
3	if currentTime - prevTime >= waktu_macet:
4	status = "Macet"
5	GPIO.output(lampu_hijau, GPIO.LOW)
6	GPIO.output(lampu_merah, GPIO.HIGH)
7	if speakFlag == 1:
8	pygame.mixer.music.load("suara.mp3")
9	pygame.mixer.music.play()
10	speakFlag = 0
11	if timerOut == 1:
12	currentTime = time.time()
13	if currentTime - prevTime >= waktu_lancar:
14	status = "Lancar"
15	GPIO.output(lampu_hijau, GPIO.HIGH)
16	GPIO.output(lampu_merah, GPIO.LOW)
17	if speakFlag == 0:
18	speakFlag = 1
19	ret, frame = cam.read()
20	frame = imutils.resize(frame,width=480)

```

21 frame = imutils.rotate(frame,180)
22 gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
23 fgmask = fgbg.apply(gray)
24 fgmask[fgmask==127] = 255
25 fgmask = cv2.morphologyEx(fgmask,cv2.MORPH_CLOSE,kernelA)
26 cv2.rectangle(frame,start_point,end_point,rect_color,2)
27 cnts,_ =
28 cv2.findContours(fgmask.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX
29 X_SIMPLE)
30 for c in cnts:
31     if cv2.contourArea(c) < minArea:
32         continue
33     (x,y,w,h) = cv2.boundingRect(c)
34     xCen = (x+x+w)/2
35     yCen = (y+y+h)/2
36     if inEn == 1:
37         income = counterCheck(xCen,yCen,inx,iny,inw)
38         if income == 1:
39             inEn = 0
40             prevTime = time.time()
41             timerIn = 1
42             timerOut = 0
43     else:
44         outcome =
45         counterCheck(xCen,yCen,outx,outy,outw)
46         if outcome == 1:
47             inEn = 1
48             prevTime = time.time()
49             timerIn = 0
50             timerOut = 1
51     cv2.rectangle(frame,s_start,s_end,(255,255,255),-1)
52     cv2.putText(frame,"Status : {}".format(status),(20,330),
53                 cv2.FONT_HERSHEY_SIMPLEX,0.8,(0,0,0),2)
54
55 cv2.imshow('frame',frame)
56 cv2.imshow('fgmask',fgmask)

```

```
57 if cv2.waitKey(1) & 0xFF == 27:
58     break
```

Setelah deklarasi *library* pada tabel 5.6 hingga tabel 5.8 maka di tabel 5.9 ini adalah proses pengolahan background subtraction hingga output pada rambu lalu lintas dinamis, terlihat pada baris ke-1 hingga baris ke-18 bahwa kendaraan yang masuk ke dalam frame akan di hitung atau *counter* waktunya dimana saat waktu mobil berhenti maka status sistem yang semula lancar seperti baris ke-14 maka akan berubah menjadi macet seperti pada baris ke-4, dan setiap perubahan status akan di ikuti juga dengan perubahan nyala pada lampu LED yaitu saat status dari lancar berubah menjadi macet maka lampu LED juga akan berubah dari warna hijau menjadi warna merah yang menyala sebagai penanda kepada pengendara nantinya bahwa tidak bisa antri di area putar balik dan perintah ini akan di tambah juga dengan bunyi nya sound yang berisi perintah untuk pengendara melakukan putar balik pada rambu berikutnya dimana suara pada sound dapat di lihat pada baris ke-8 dimana file suara pada sound di beri nama suara.mp3.

Pada baris ke-19 hingga baris ke-26 untuk deklarasi pembuatan frame yang digunakan untuk deteksi kendaraan (mobil) dimana saat mobil masuk frame inilah yang nantinya mobil akan di olah oleh system sehingga counter waktu akan berjalan untuk menentukan apakah kondisi area rambu lalu lintas dinilai macet atau lancar. Dimana pada baris ke-30 sampai baris ke-50 digunakan untuk menghitung seberapa lama objek (mobil) yang masuk kedalam frame, dimana saat ada ada objek yang terdeteksi di dalam frame kepadatannya lebih dari 5 detik maka akan di nilai macet oleh sistem, dimana dengan kata lain ada baris ke-30 sampai baris ke-50 ini digunakan untuk menentukan koordinat deteksi agar tidak semua objek yang masuk kedalam frame kamera bisa sepenuhnya di olah oleh system sehingga system bisa berjalan sesuai yang di harapkan yaitu hanya objek yang masuk kedalam frame yang di tentukan saja yang nantinya akan di olah oleh syrtem menggunakan *background subtraction*.

5.1.3.5 Implementasi prototype alat

Pada implementasi prototype alat untuk perancangan sistem untuk pengaturan rambu putar balik (u-turn) menggunakan metode background subtraction yang mengacu pada sub bab 5.1.1 maka hasilnya seperti yang terlihat pada gambar 5.8 dibawah ini, bahan yang di gunakan adalah kardus, kertas karton, plat besi 4 biji dan miniatur mobil mainan dengan skala 1:54,33cm



Gambar 5.18 Implementasi *prototype* sistem



BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai pengujian serta analisis dari hasil pengujian yang telah dilakukan berdasarkan sistem yang dibuat. Tujuan dari pengujian dan analisis ini yaitu untuk mengetahui apakah sistem yang sudah dibuat sesuai dengan analisis kebutuhan yang diinginkan. Pengujian terhadap sistem dilakukan dalam beberapa tahapan yang sesuai dengan kebutuhan yang diinginkan. Sedangkan analisis dilakukan untuk menarik kesimpulan dari penelitian yang sudah dilakukan.

6.1 Pengujian Kode Program Background Substraction Menggunakan Kamera Logitech C270h

Pengujian dilakukan dengan kamera Logitech C270h untuk mengetahui lama waktu yang dibutuhkan untuk komunikasi antara kamera dengan raspberry pi dan tingkat keakuratan kamera dalam menangkap gambar replika mobil dengan perbandingan waktu yang sudah ditentukan sebelumnya, sehingga akan diperoleh nilai error dari percobaan kamera yang digunakan oleh sistem.

6.1.1 Tujuan pengujian

Pada pengujian kali ini berfungsi untuk mengetahui nilai atau *output* dari *inputan* pada *prototype* hal ini dikarenakan pada posisi yang sebenarnya di area rambu putar balik kendaraan berada di posisi yang beraneka macam dengan warna kendaraan yang beraneka macam juga, lalu dari segi pencahayaan juga bermacam-macam nilai nya sehingga pasti akan mempengaruhi penangkapan gambar oleh kamera yang akan mempengaruhi *output* juga.

6.1.2 Prosedur pengujian

Berikut prosedur yang dilakukan untuk menguji kamera Logitech c270h dan sistem:

- Menghubungkan sistem dengan laptop.
- Run program python3 pada cmd *vnc viewers*.
- Mengukur keakuratan sistem pada setiap replika mobil yang berhenti dimana yang diukur adalah kesesuaian sistem dalam mendeteksi objek mobil yaitu jumlah mobil, warna mobil, dan dengan pencahayaan yang berbeda
- Menyesuaikan dan mengamati keakuratan sistem yang outputnya dari terminal *vnc viewers* ke lampu led dan *sound*

- e. Nilai output dari masing-masing pembacaan deteksi diambil nilai selisihnya untuk di lakukan perhitungan presentase error dalam bentuk decimal.

Cara untuk mengukur presentase error yaitu dengan menggunakan persamaan (6.1) berikut:

$$\text{Presentase Error} = \frac{\text{jumlah error}}{\text{jumlah sample percobaan}} \times 100\% \quad (6.1)$$

Sedangkan untuk mencari waktu rata-rata nya menggunakan rumus persamaan (6.2) berikut:

$$R = \frac{\sum P}{n} \quad (6.2)$$

keterangan :

R = waktu rata-rata

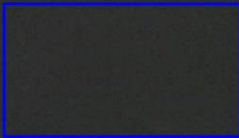
$\sum P$ = Jumlah hasil waktu percobaan

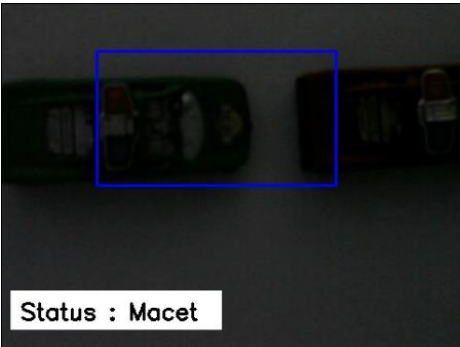
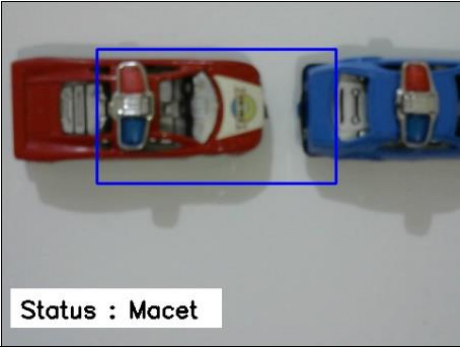
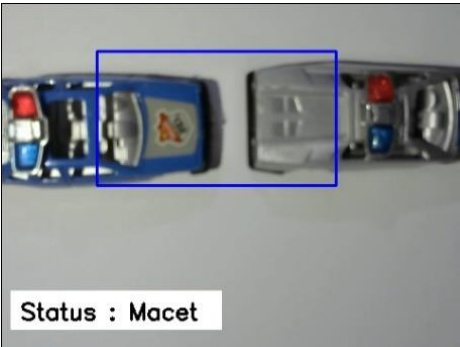
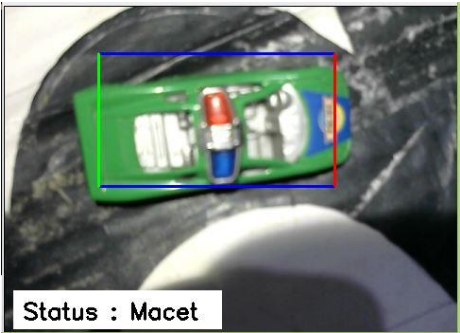
n = jumlah data sampel

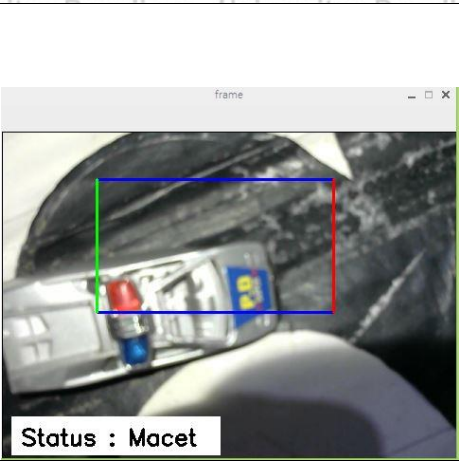
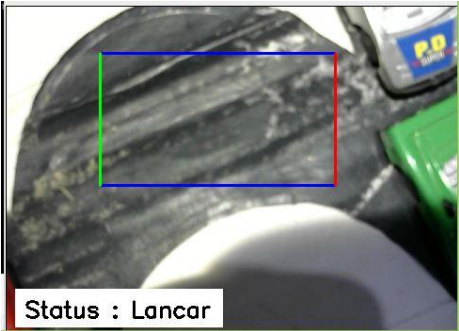
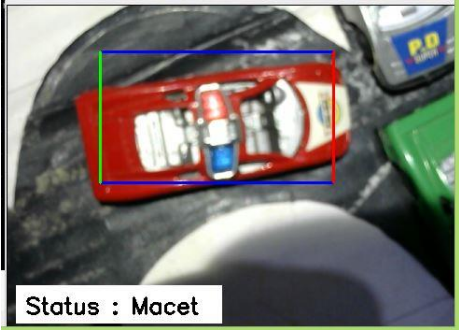
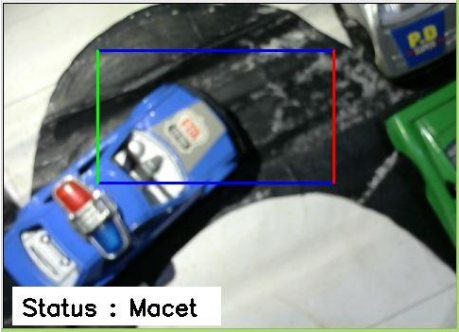
6.1.3 Hasil pengujian deteksi kendaraan (mobil) dengan warna objek berbeda

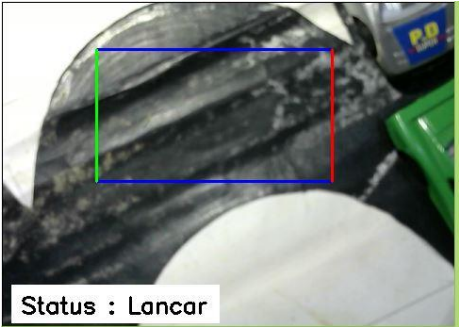
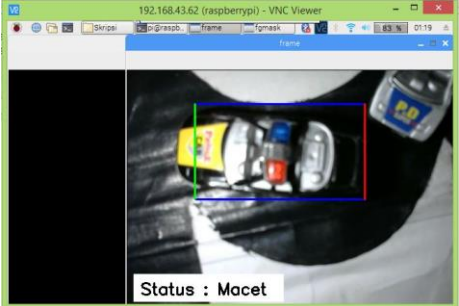
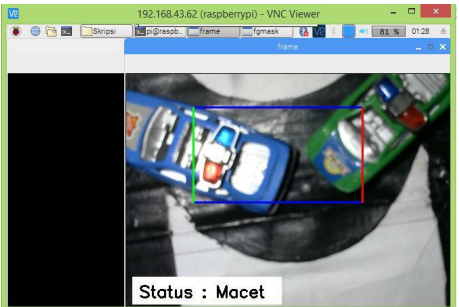
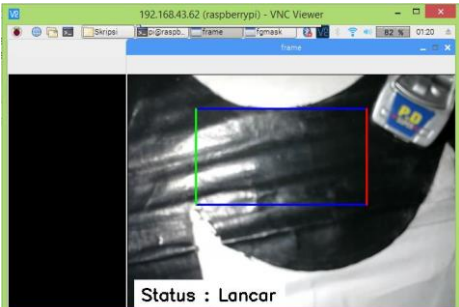
Pengujian deteksi kendaraan (mobil) dengan warna yang berbeda adalah pengujian pada area rambu putar balik *u-turn* menggunakan sensor input webcam secara langsung menggunakan objek mobil *prototype* dengan warna dan posisi mobil yang berbeda secara *realtime* untuk mengetahui apakah sistem bekerja dengan baik. Proses deteksi dan percobaan dilakukan dengan 15 data contoh, dan pengujian dapat dilihat pada tabel 6.1 dibawah ini :

Tabel 6.3 Hasil Pengujian deteksi kamera dengan warna dan posisi objek (mobil) yang berbeda

TABEL DETEKSI KAMERA DENGAN WARNA DAN POSISI MOBIL YANG BERBEDA				
JUMLAH MOBIL TERDETEKSI				
NO	DI KAMERA	WARNA MOBIL	KESIMPULAN	GAMBAR
1	0	Tidak ada objek	SESUAI	 Status : Lancar
2	0	Tidak ada objek	SESUAI	 Status : Lancar
3	0	Tidak ada objek	SESUAI	 Status : Lancar

4	2	1 merah 1 hijau	SESUAI	 Status : Macet
5	2	1 merah 1 biru	SESUAI	 Status : Macet
6	2	1 biru 1 abu-abu	SESUAI	 Status : Macet
7	1	1 hijau	SESUAI	 Status : Macet

8	1	1 abu-abu	SESUAI	 Status : Macet
9	0	Tidak ada Objek yang Terdeteksi	SESUAI	 Status : Lancar
10	1	1 merah	SESUAI	 Status : Macet
11	1	1 biru	SESUAI	 Status : Macet

12	0	Tidak ada Objek yang Terdeteksi	SESUAI	
13	1	1 hitam	SESUAI	
14	2	1 hijau 1 biru	SESUAI	
15	0	Tidak ada Objek yang Terdeteksi	SESUAI	

Untuk menentukan presentase error pada setiap pengujian berdasarkan persamaan (6.1),diperoleh rata-rata deteksi kendaraan error sebesar 0%. untuk mendapatkan presentase error sebagai berikut :

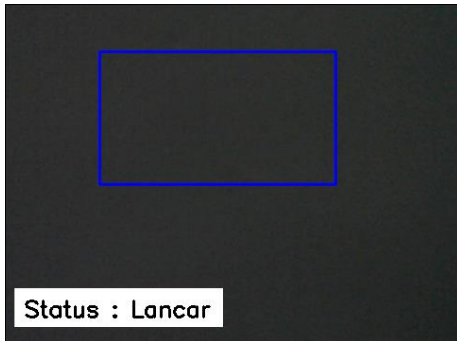
$$\begin{aligned} \text{Presentase Error} &= \frac{\text{jumlah error}}{\text{jumlah sample percobaan}} \times 100\% \\ &= \frac{0}{15} \times 100\% \\ &= 0\% \end{aligned}$$

Dari perhitungan di atas dapat diketahui bahwa pada penelitian ini untuk hasil deteksi kendaraan dengan berbagai posisi dan warna tidak ada kendala atau semua kendaraan yang masuk pada frame kamera dapat terdeteksi dengan baik.

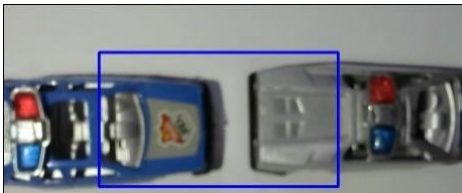
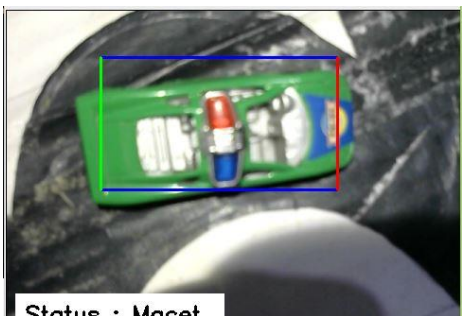
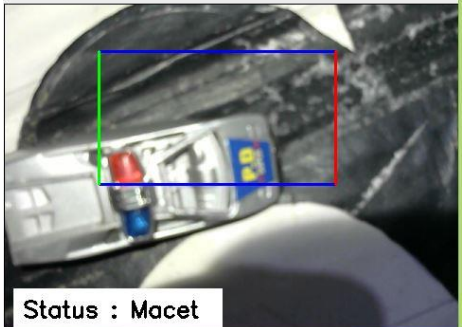
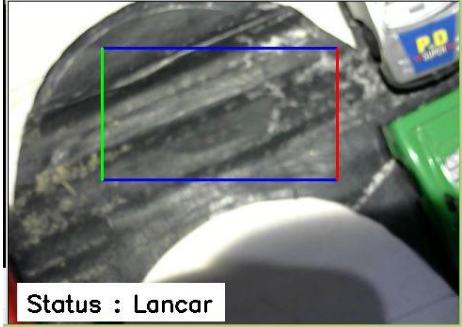
6.1.4 Hasil pengujian deteksi kendaraan (mobil) rata-rata waktu dengan intensitas cahaya yang berbeda

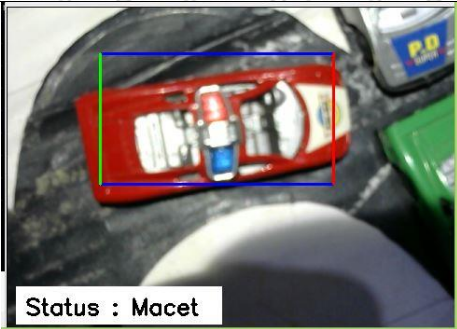
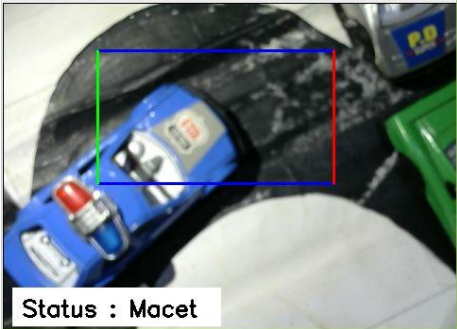
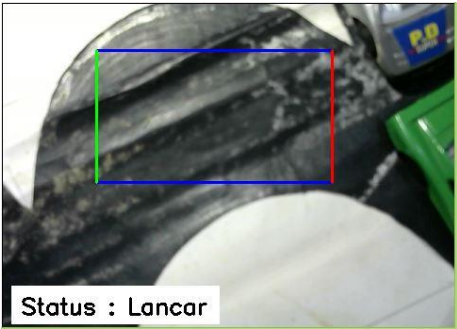
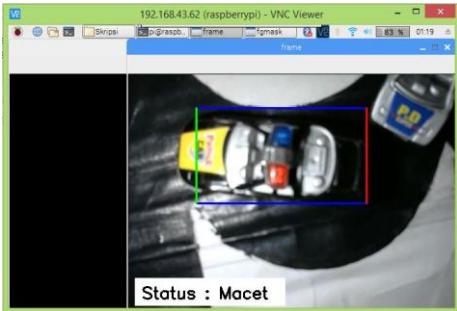
Hasil pengujian deteksi *prototype* mobil ini menggunakan tingkat pencahayaan yang berbeda dimana pada percobaan ini dilakukan sebanyak 15 kali dengan tingkat pencahayaan yaitu 1 lux hingga lebih dari 2000 lux dimana percobaan ini dilakukan juga secara realtime sebagai bentuk *prototype* kondisi jalan *real* dengan kondisi cuaca terang, mendung, pagi hari, siang hari, dan malam hari, pengujian deteksi kendaraan mobil pada area putar balik dengan intensitas cahaya yang berbeda dapat dilihat pada tabel 6.2 dibawah ini :

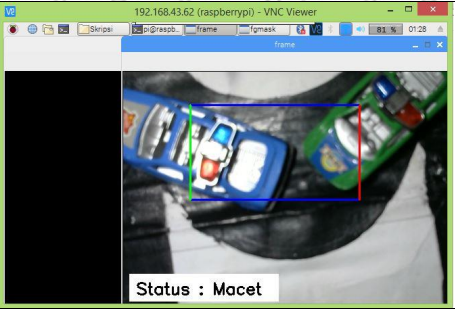
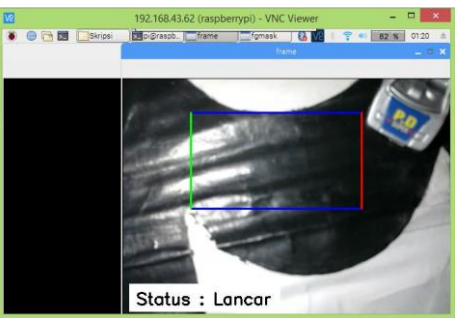
Tabel 6.2 Hasil Pengujian rata-rata waktu deteksi kamera dengan intensitas cahaya yang berbeda

TABEL DETEKSI KAMERA MENGGUNAKAN INTENSITAS CAHAYA BERBEDA			
NO	NILAI BESARAN INTENSITAS CAHAYA (LUX)	WAKTU DETEKSI	GAMBAR
1	1lux	5 DETIK	

2	19 lux	5 DETIK	 Status : Lancar
3	150 lux	5 DETIK	 Status : Lancar
4	1 lux	5.11 DETIK	 Status : Macet
5	19 lux	5.14 DETIK	 Status : Macet

6	150 lux	5.25 DETIK	 Status : Macet
7	540 lux	5.25 DETIK	 Status : Macet
8	540 lux	5.15 DETIK	 Status : Macet
9	540 lux	5 DETIK	 Status : Lancar

10	625 lux	5.11 DETIK	
11	815 lux	5.11 DETIK	
12	821 lux	5 DETIK	
13	2213 lux	5.20 DETIK	

14	2213 lux	5.21 DETIK	
15	2213 lux	5.06 DETIK	

Setelah deteksi kendaraan dilakukan perhitungan error maka selanjutnya dilakukan perhitungan rata-rata nilai pada saat kendaraan terdeteksi dengan kondisi pencahayaan yang berbeda, dimana hal ini dilakukan untuk mengetahui keakuratan sistem dalam mendeteksi kendaraan dari sisi waktu yang sebelumnya sudah di tentukan, berikut perhitungannya berdasarkan persamaan (6.2) :

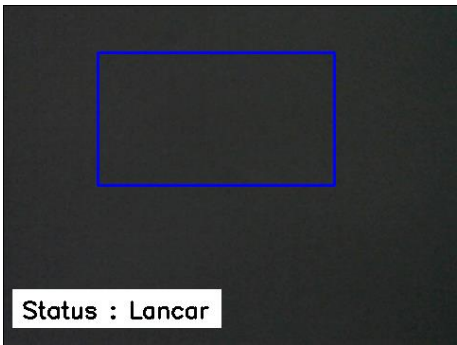
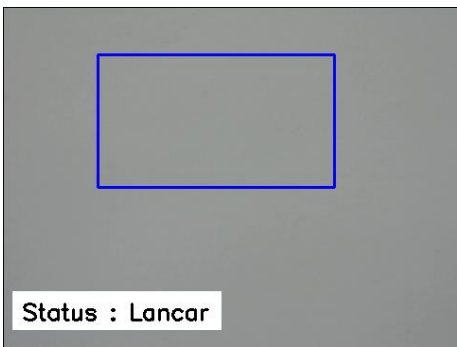
$$\begin{aligned}
 R &= \frac{\sum P}{n} \\
 &= \frac{5+5+5+5.11+5.14+5.25+5.25+5.15+5+5.11+5.11+5+5.20+5.21+5.06}{15} \\
 &= \frac{76.59}{15} \\
 &= 5.106 \text{ detik}
 \end{aligned}$$

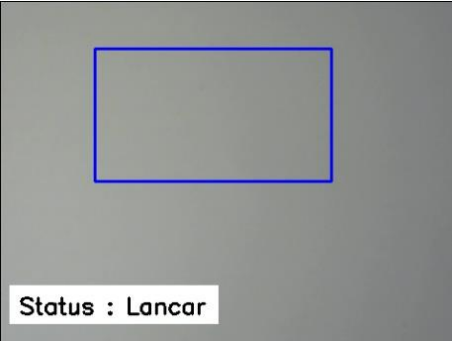
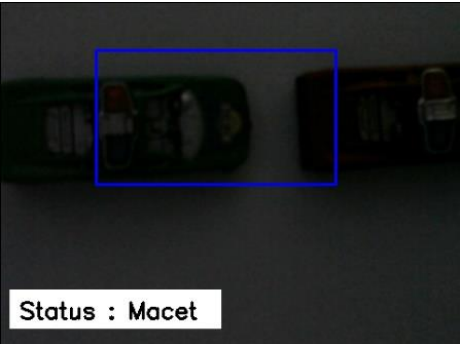
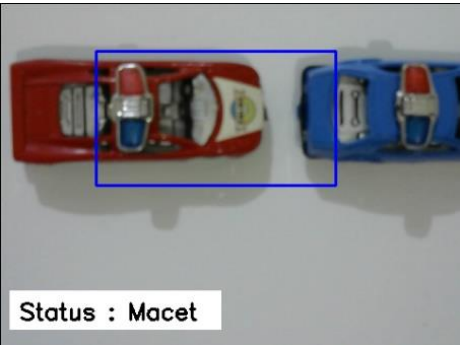
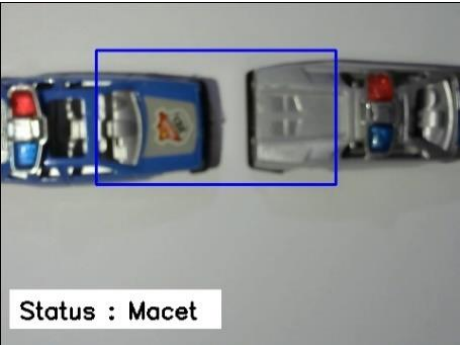
Dari perhitungan di atas di ketahui jika rata-rata waktu yang di butuhkan sistem untuk dapat mendeteksi kemacetan di area putar balik dengan kondisi nilai pencahayaan yang berbeda adalah sebesar 5.106 detik, dimana pada sistem waktu yang sudah di tentukan adalah 5 detik, sehingga rata-rata dari 15 percobaan ada penambahan waktu sebesar 0.106 detik.

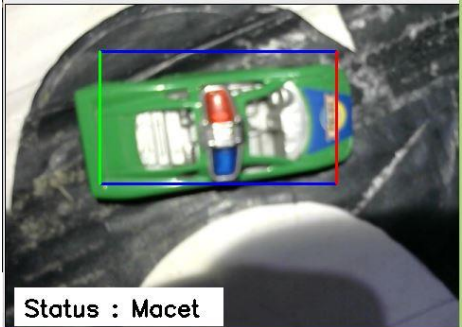
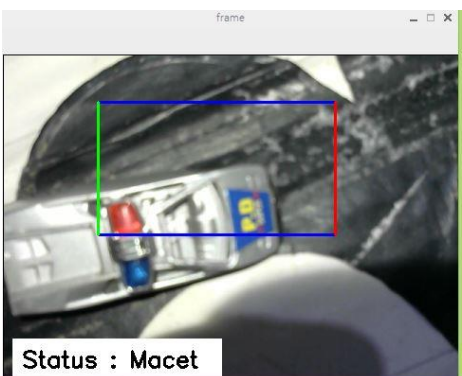
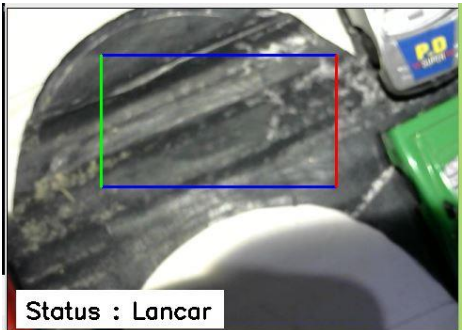
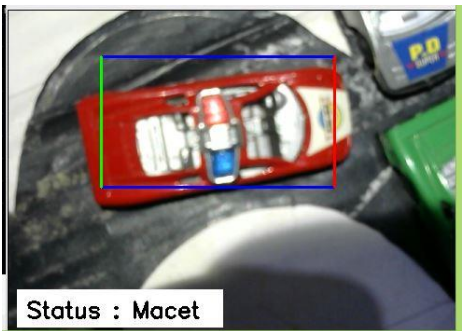
6.1.5 Hasil pengujian akurasi integrasi antara *input* dan *output* pada sistem

Berdasarkan hasil uji yang di lakukan pada sistem untuk melihat akurasi integrasi antara input berupa kamera webcam yang diolah menggunakan *background subtraction* yang kemudian di keluarkan pada output berupa lampu led merah, hijau, dan suara pada sound maka dari diperlukan sedikitnya 15 sample data yang di gunakan untuk mengetahui akurasi sistem rambu lalu lintas dinamis, 15 sample percobaan dapat dilihat pada tabel 6.3 dibawah ini :

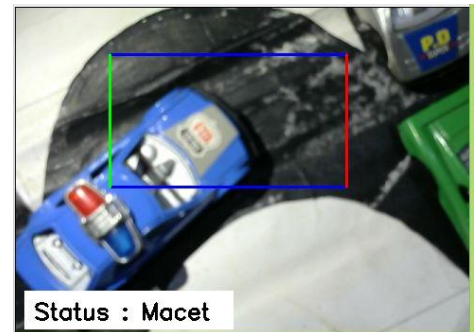
Tabel 6.3 Hasil Pengujian akurasi integrasi antara *input* dan *output* sistem

TABEL DETEKSI KAMERA MENGGUNAKAN INTENSITAS CAHAYA BERBEDA DENGAN OUTPUT LED						
NO	JUMLAH MOBIL TERDETEKSI DI KAMERA	LAMPU LED MERAH	LAMPU LED HIJAU	SOUND	KESIMPULAN	GAMBAR
1	0	OFF	ON	OFF	SESUAI	 Status : Lancar
2	0	OFF	ON	OFF	SESUAI	 Status : Lancar

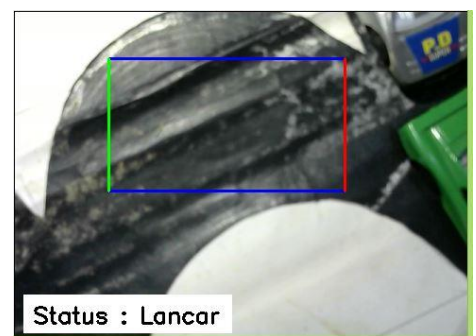
3	0	OFF	ON	OFF	SESUAI	 Status : Lancar
4	2	ON	OFF	MENGELUARKAN SUARA UNTUK MENGATASI KEMACETAN MAKA PENGGUNA MOBIL HARUS MELAKUKAN PUTAR BALIK DI AREA PUTAR BALIK BERIKUTNYA		 Status : Macet
5	2	ON	OFF	MENGELUARKAN SUARA UNTUK MENGATASI KEMACETAN MAKA PENGGUNA MOBIL HARUS MELAKUKAN PUTAR BALIK DI AREA PUTAR BALIK BERIKUTNYA		 Status : Macet
6	2	ON	OFF	MENGELUARKAN SUARA UNTUK MENGATASI KEMACETAN MAKA PENGGUNA MOBIL HARUS MELAKUKAN PUTAR BALIK DI AREA PUTAR BALIK BERIKUTNYA		 Status : Macet

7	0	ON	OFF	MENGELUARKAN SUARA UNTUK MENGATASI KEMACETAN MAKA PENGGUNA MOBIL HARUS MELAKUKAN PUTAR BALIK DI AREA PUTAR BALIK BERIKUTNYA	SESUAI		
8	1	ON	OFF	MENGELUARKAN SUARA UNTUK MENGATASI KEMACETAN MAKA PENGGUNA MOBIL HARUS MELAKUKAN PUTAR BALIK DI AREA PUTAR BALIK BERIKUTNYA	SESUAI		
9	0	OFF	ON		OFF	SESUAI	
10	1	ON	OFF	MENGELUARKAN SUARA UNTUK MENGATASI KEMACETAN MAKA PENGGUNA MOBIL HARUS MELAKUKAN PUTAR BALIK DI AREA PUTAR BALIK BERIKUTNYA	SESUAI		

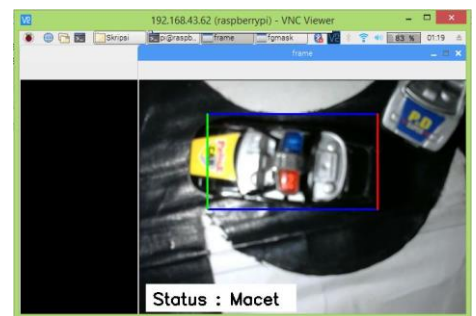
11	1	ON	OFF	MENGELUARKAN SUARA UNTUK MENGATASI KEMACETAN MAKA PENGGUNA MOBIL HARUS MELAKUKAN PUTAR BALIK DI AREA PUTAR BALIK BERIKUTNYA	SESUAI
----	---	----	-----	--	--------



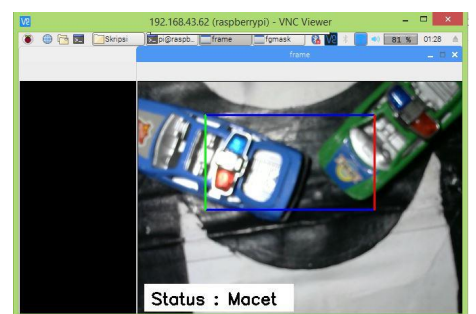
12	0	ON	OFF	OFF	SESUAI
----	---	----	-----	-----	--------

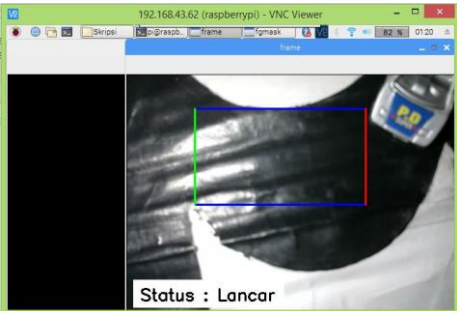


13	1	ON	OFF	MENGELUARKAN SUARA UNTUK MENGATASI KEMACETAN MAKA PENGGUNA MOBIL HARUS MELAKUKAN PUTAR BALIK DI AREA PUTAR BALIK BERIKUTNYA	SESUAI
----	---	----	-----	--	--------



14	2	ON	OFF	MENGELUARKAN SUARA UNTUK MENGATASI KEMACETAN MAKA PENGGUNA MOBIL HARUS MELAKUKAN PUTAR BALIK DI AREA PUTAR BALIK BERIKUTNYA	SESUAI
----	---	----	-----	--	--------



15	0	OFF	ON	OFF	SESUAI	
----	---	-----	----	-----	--------	---

Untuk menentukan presentase error pada setiap pengujian berdasarkan persamaan (6.1), diperoleh rata-rata integrasi sistem antara input dan output error sebesar 0%. untuk mendapatkan presentase error sebagai berikut :

$$\begin{aligned}
 \text{Presentase Error} &= \frac{\text{jumlah error}}{\text{jumlah sample percobaan}} \times 100\% \\
 &= \frac{0}{15} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dari perhitungan di atas dapat diketahui bahwa pada penelitian ini untuk hasil deteksi kendaraan pada area putar balik dimana antara input berupa kamera webcam dan output berupa lampu led merah, hijau, dan sound sebagai rambu lalu lintas dinamis sistem sesuai dimana setiap kendaraan yang terdeteksi kondisinya akan sesuai dengan sistem yaitu dinilai macet atau lancar sehingga diharapkan sistem ini dapat digunakan untuk mengurai kemacetan terutama dari antrian kendaraan yang ingin melakukan putar balik.

BAB 7 KESIMPULAN DAN SARAN

Bab ini berisi penarikan kesimpulan berdasarkan tahap-tahap yang telah dilakukan sebelumnya. Serta berisi saran dari peneliti yang diharapkan dapat digunakan untuk pengembangan penelitian serupa pada penelitian-penelitian selanjutnya.

7.1 Kesimpulan

Berdasarkan rumusan masalah yang diajukan di awal penelitian serta berdasarkan hasil pengujian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Implementasi metode *background subtraction* melalui proses pengujian deteksi pada *input* kamera dengan objek mobil di area putar balik (*u-turn*) dilakukan 15 kali percobaan dengan pengujian objek mobil diposisikan berbeda tiap percobaan dan dengan warna objek yang berbeda mendapatkan nilai error 0%
2. Analisa rata-rata waktu deteksi objek berupa mobil pada area putar balik (*u-turn*) dengan nilai intensitas cahaya dari 1 lux hingga 2213 lux mendapat rata-rata waktu 5.106 detik dimana ada selisih rata-rata waktu dari batas sistem yaitu 5 detik sehingga selisih rata-ratanya 0.106 detik pada setiap percobaan.
3. Proses pengujian hasil integrasi antara *input* dan *output* sistem dimana input berupa kamera webcam dan output berupa lampu led merah dan hijau mendapatkan hasil akurasi error 0% sehingga untuk komunikasi antara *input* dan *output* sistem dinilai berhasil.

7.2 Saran

Adapun beberapa saran dari peneliti yang dapat dijadikan sebagai acuan untuk pengembangan penelitian ini atau penelitian serupa kedepannya yaitu:

1. menggunakan penerangan yang khusus dan tetap atau tidak menggunakan penerangan ruangan sehingga hasil deteksi lebih akurat.
2. menambahkan baterai atau accu sehingga system bisa portable dan raspberry phi tidak khawatir off jika ada pemadaman pada lokasi yang gunakan untuk pengujian system ini.
3. membuat prototype yang lebih real, bisa di tambah lego, atau miniature tambahan lain seperti tumbuhan, pohon, dan lain sebagainya supaya prototype menjadi menyerupai kondisi asli dan lebih hidup.
4. untuk mikrokontroler bisa menggunakan versi yang lebih tinggi lagi, dimana pada penelitian kali ini penulis menggunakan raspberry phi 3

model b dan bisa di ganti dengan yang lebih terbaru seperti raspberry phi empat atau versi lebih terbaru lagi, dimana pada versi terbaru bisa mengolah data lebih akurat dan efisien dalam segi waktu mengolah dan sebagainya.



DAFTAR REFERENSI

achmadrizal. (2014, juni 19). *Pengolahan Citra*. Retrieved from telkomuniversity.ac.id:

<https://achmadrizal.staff.telkomuniversity.ac.id/pengolahan-citra/>

aliridhos. (2013, May 21). *VNC (VIRTUAL NETWORK COMPUTING)*. Retrieved from

<https://marstechnoo.wordpress.com/>:

<https://marstechnoo.wordpress.com/2013/05/21/203/>

carsome. (2021, februari 24). *Rambu Lalu Lintas dan Artinya Berdasarkan Kategorinya*.

Retrieved from carsome.id: [https://www.carsome.id/news/item/rambu-lalu-](https://www.carsome.id/news/item/rambu-lalu-lintas-dan-pengertiannya)

[lintas-dan-pengertiannya](https://www.carsome.id/news/item/rambu-lalu-lintas-dan-pengertiannya)

Dishub LLAJ. (2010). *Tentang_ATCS*. Retrieved from atcs-dishub.bandung.go.id:

http://atcs-dishub.bandung.go.id/Tentang_ATCS

Dishub, M. (1993). *RAMBU-RAMBU LALU LINTAS DI JALAN MENTERI PERHUBUNGAN*.

<https://jnk tollroad.com/wp-content/uploads/2017/08/Keputusan-Menteri-Perhubungan-No.-KM-61-Tahun-1993-tentang-Rambu-Rambu-Lalu-Lintas-Di-Jalan.pdf>.

Fakhli. (2018, Agustus 2018). *Jenis Rambu Peringatan dan Artinya*. Retrieved from

<https://www.kumpulengineer.com/>:

<https://www.kumpulengineer.com/2018/12/jenis-rambu-peringatan-dan-artinya.html>

gaikindo. (2021, maret 21). Retrieved from hasil sensus BPS: jumlah kendaraan bermotor

di indonesia tembus 133 juta unit: <https://www.gaikindo.or.id/data-bps-jumlah-kendaraan-bermotor-di-indonesia-tembus-133-juta-unit/>

humaspolresbantul. (26, Agustus 2014). *ARTI DAN LAMBANG RAMBU LALU LINTAS*.

Retrieved from <http://humaspolresbantul.blogspot.com/>:

<http://humaspolresbantul.blogspot.com/2014/08/arti-dan-lambang-rambu-lalu-lintas.html>

kakisentot. (2021, 3 1). *Rambu-rambu Lalu Lintas*. Retrieved from workamerica.co:

<https://workamerica.co/rambu-lalu-lintas/>

Menhub. (1993). *Rambu Rambu Lalu Lintas di Jalan*.

<http://hubdat.dephub.go.id/km/tahun-1993/121-km-61-tahun-1993>. Retrieved from <http://hubdat.dephub.go.id/km/tahun-1993/121-km-61-tahun-1993>

Menhub. (1993). *RAMBU-RAMBU LALU LINTAS DI JALAN*. [https://jnk tollroad.com/wp-](https://jnk tollroad.com/wp-content/uploads/2017/08/Keputusan-Menteri-Perhubungan-No.-KM-61-Tahun-1993-tentang-Rambu-Rambu-Lalu-Lintas-Di-Jalan.pdf)

[content/uploads/2017/08/Keputusan-Menteri-Perhubungan-No.-KM-61-Tahun-1993-tentang-Rambu-Rambu-Lalu-Lintas-Di-Jalan.pdf](https://jnk tollroad.com/wp-content/uploads/2017/08/Keputusan-Menteri-Perhubungan-No.-KM-61-Tahun-1993-tentang-Rambu-Rambu-Lalu-Lintas-Di-Jalan.pdf).

Miss Helly M Desai, M. V. (2014). *Background Subtraction Techniques. A Survey:*

Background Subtraction Techniques, volume 5.

Munir, R. (2019, May 4). *Citra Biner*. Retrieved from Interpretasi dan Pengolahan Citra:
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2019-2020/15-Citra-Biner.pdf>

Nadir Farhi, H. H.-S.-P. (2017). Traffic system Communication Algorithm. A vehicle-to-infrastructure communication based algorithm for urban traffic control,
<https://arxiv.org/pdf/1703.08408.pdf>.

Pamungkas, A. (2017, oktober 21). *Background Subtraction (Foreground Detection)*. Retrieved from <https://pemrogramanmatlab.com/>:
<https://pemrogramanmatlab.com/2015/10/12/background-subtraction-foreground-detection/>

phi, r. (2016). *raspberrypi 3b*. Retrieved from www.pricebook.co.id:
<https://www.pricebook.co.id>

Sameh Samra, A. E.-M. (2013). algorithm provides for an efficient dynamic programming formulation of the state space. *A Linear-Time and Space Algorithm for Optimal Traffic Signal Durations at an Intersection*,
<https://arxiv.org/ftp/arxiv/papers/1311/1311.0362.pdf>.

statistika, b. p. (2021, januari 21). *hasil sensus penduduk 2020*. Retrieved from bps.go.id:
<https://www.bps.go.id/pressrelease/2021/01/21/1854/hasil-sensus-penduduk-2020.html>

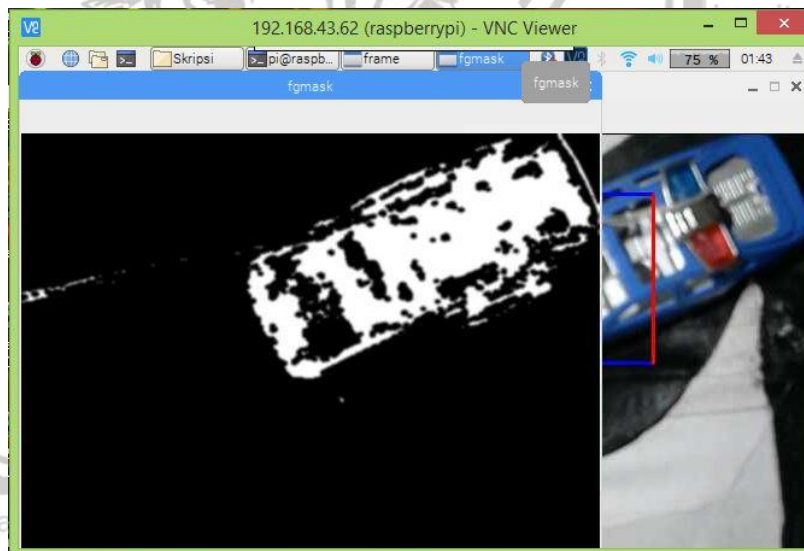
theconversation. (2019, desember 16). *Paradoks kemacetan perkotaan dan solusi untuk mengatasinya* . Retrieved from theconversation.com:
<https://theconversation.com/paradoks-kemacetan-perkotaan-dan-solusi-untuk-mengatasinya-127021>

VNC (VIRTUAL NETWORK COMPUTING). (2013, may). Retrieved from
<https://marstechnoo.wordpress.com/>.

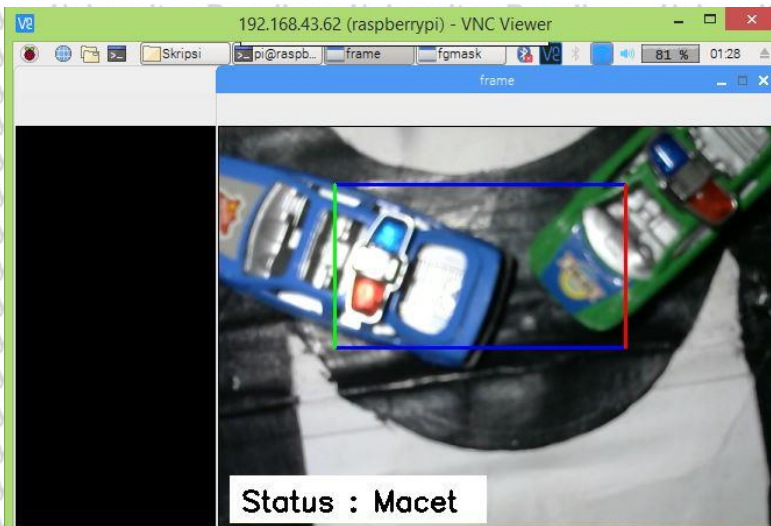
LAMPIRAN HASIL UJI COBA



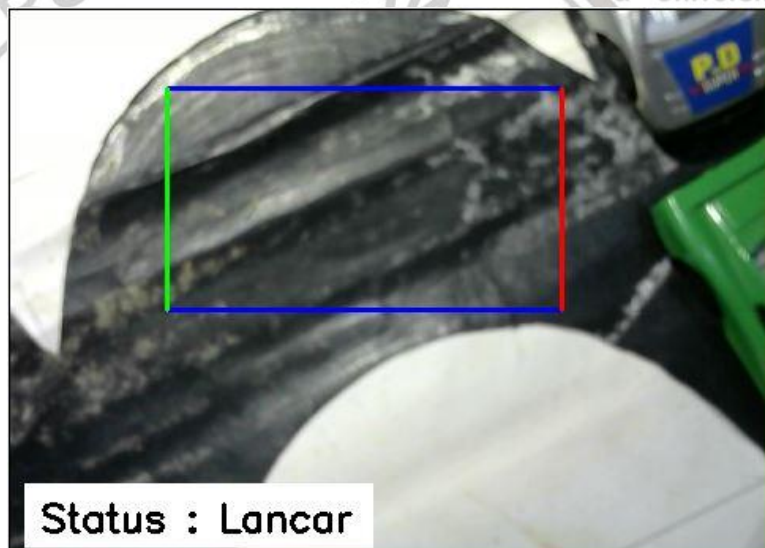
Gambar 6.1 Tampilan keseluruhan *prototype* sistem



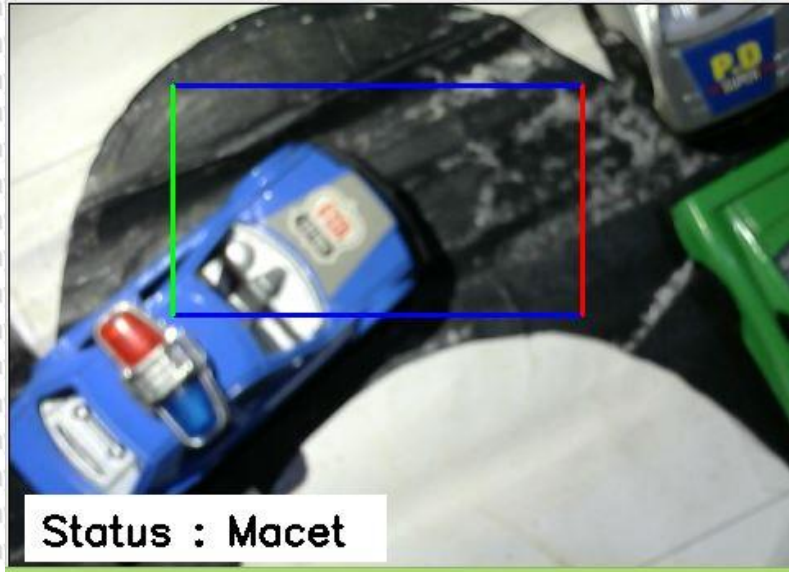
Gambar 6.2 Tampilan fgmask deteksi objek *prototype* mobil



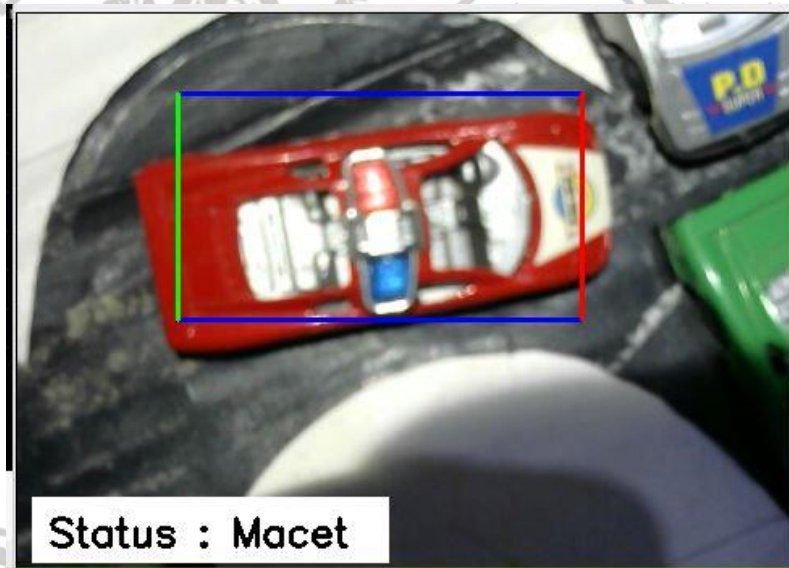
Gambar 6.3 Tampilan 2 objek yang terdeteksi dengan warna objek *prototype* mobil hijau dan biru



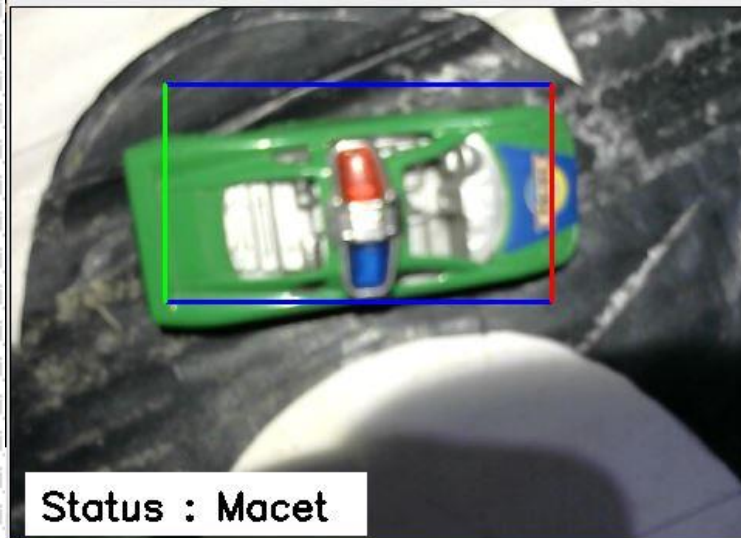
Gambar 6.4 Tampilan area rambu putar balik (*u-turn*) tanpa objek *prototype* mobil yang terdeteksi oleh frame



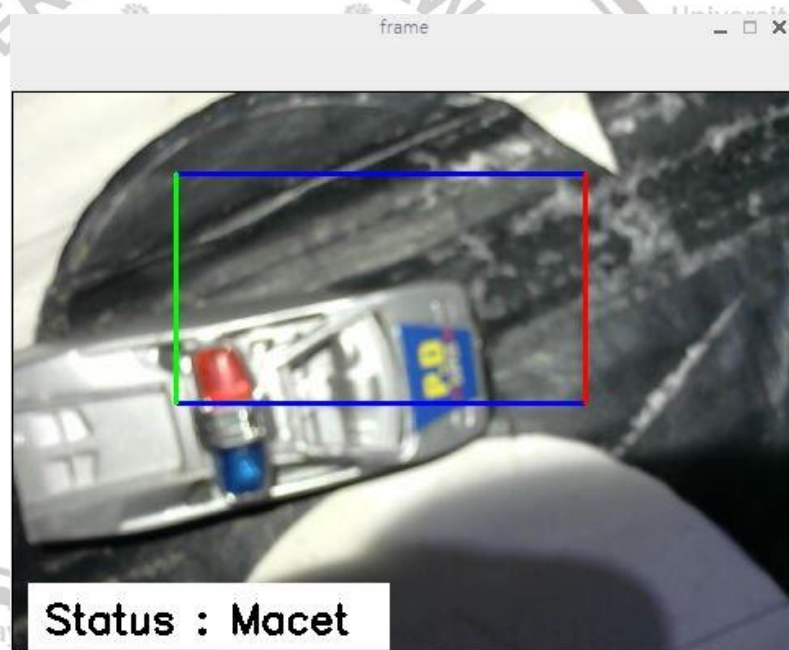
Gambar 6.5 Tampilan deteksi objek *prototype* mobil berwarna biru dengan output sistem bernilai macet



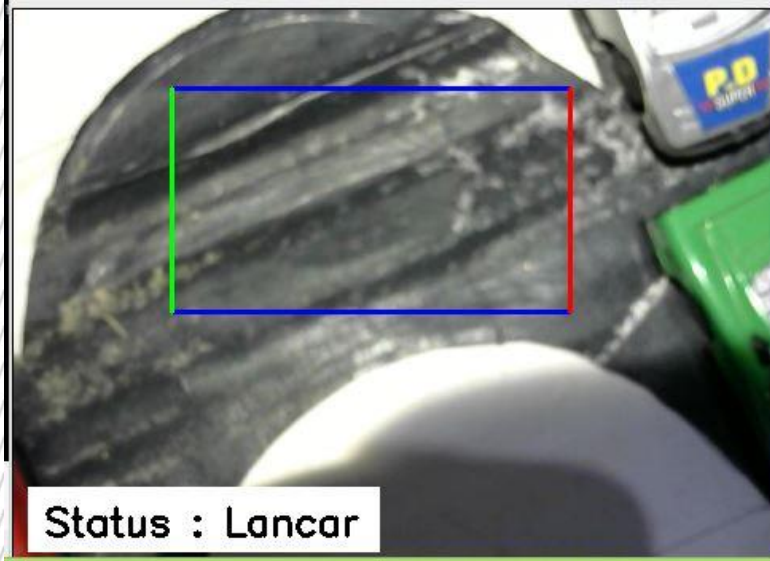
Gambar 6.6 Tampilan deteksi objek *prototype* mobil berwarna merah dengan output sistem bernilai macet



Gambar 6.7 Tampilan deteksi objek *prototype* mobil berwarna hijau dengan *output* sistem bernilai macet



Gambar 6.8 Tampilan deteksi objek *prototype* mobil berwarna abu-abu dengan *output* sistem bernilai macet



Gambar 6.9 tampilan kondisi *prototype* area putar balik (*u-turn*) tanpa ada objek yang terdeteksi dalam frame yang telah di tentukan

Tabel 6.4 Source Code keseluruhan sistem

Kode program
<pre>import cv2 import numpy as np import imutils import time import RPi.GPIO as GPIO import pygame waktu_macet = 5 waktu_lancar = 1 lampu_hijau = 21 lampu_merah = 20 GPIO.setmode(GPIO.BCM) GPIO.setwarnings(False) GPIO.setup(lampu_hijau, GPIO.OUT) GPIO.setup(lampu_merah, GPIO.OUT) GPIO.output(lampu_hijau, GPIO.HIGH) GPIO.output(lampu_merah, GPIO.LOW)</pre>

```

cam = cv2.VideoCapture(0)

pygame.mixer.init()
pygame.mixer.music.set_volume(0.35)

s_start = (10,300)
s_end = (230,340)

start_point = (100,50)
end_point = (350,190)
rect_color = (255,0,0)

inx = start_point[0]
iny = start_point[1]
inw = end_point[1] - iny

outx = end_point[0]
outy = start_point[1]
outw = end_point[1] - outy

minArea = 2000
inEn = 1
income = 0
outcome = 0
timerIn = 0
timerOut = 0
status = "Lancar"
currentTime = 0
prevTime = 0
speakFlag = 1

fgbg = cv2.createBackgroundSubtractorMOG2(history=100,detectShadows=True)
kernelA = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5))

def counterCheck(dotx,doty,x,y,w):
    dist = abs(dotx-x)

```

```

if dist <= 5 and doty >= y and doty <= y+w:
    return 1
else:
    return 0

while True:
    if timerIn == 1:
        currentTime = time.time()
        if currentTime - prevTime >= waktu_macet:
            status = "Macet"
            GPIO.output(lampu_hijau, GPIO.LOW)
            GPIO.output(lampu_merah, GPIO.HIGH)
            if speakFlag == 1:
                pygame.mixer.music.load("suara.mp3")
                pygame.mixer.music.play()
                speakFlag = 0
        if timerOut == 1:
            currentTime = time.time()
            if currentTime - prevTime >= waktu_lancar:
                status = "Lancar"
                GPIO.output(lampu_hijau, GPIO.HIGH)
                GPIO.output(lampu_merah, GPIO.LOW)
                if speakFlag == 0:
                    speakFlag = 1
            ret, frame = cam.read()
            frame = imutils.resize(frame,width=480)
            frame = imutils.rotate(frame,180)
            gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
            fgmask = fgbg.apply(gray)
            fgmask[fgmask==127] = 255
            fgmask = cv2.morphologyEx(fgmask,cv2.MORPH_CLOSE,kernelA)
            fgmask = cv2.GaussianBlur(fgmask,(5,5),0)
            cv2.rectangle(frame,start_point,end_point,rect_color,2)
            cv2.line(frame,(inx,inx),(inx,inx+inw),(0,255,0),2)
            cv2.line(frame,(outx,outx),(outx,outx+outw),(0,0,255),2)
            cnts, _ =
            cv2.findContours(fgmask.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)

```

```

for c in cnts:
    if cv2.contourArea(c) < minArea:
        continue
    (x,y,w,h) = cv2.boundingRect(c)
    xCen = (x+x+w)/2
    yCen = (y+y+h)/2
    if inEn == 1:
        income = counterCheck(xCen,yCen,inx,iny,inw)
        if income == 1:
            inEn = 0
            prevTime = time.time()
            timerIn = 1
            timerOut = 0
        else:
            outcome = counterCheck(xCen,yCen,outx,outy,outw)
            if outcome == 1:
                inEn = 1
                prevTime = time.time()
                timerIn = 0
                timerOut = 1
    cv2.rectangle(frame,s_start,s_end,(255,255,255),-1)
    cv2.putText(frame,"Status : {}".format(status),(20,330),
                cv2.FONT_HERSHEY_SIMPLEX,0.8,(0,0,0),2)

    cv2.imshow('frame',frame)
    cv2.imshow('fgmask',fgmask)
    if cv2.waitKey(1) & 0xff == 27:
        break

cam.release()
GPIO.cleanup()
print("Program dihentikan.")
cv2.destroyAllWindows()

```