SISTEM DETEKSI MARKA MEMBUJUR GARIS UTUH DANPUTUS MENGGUNAKAN METODE THRESHOLDING DAN HOUGH TRANSFORMBERBASIS RASPBERY PI

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Putra Wijaya
NIM: 145150300111046

PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

SISTEM DETEKSI MARKA MEMBUJUR GARIS UTUH DAN PUTUS MENGGUNAKAN METODE THRESHOLDING DAN HOUGH TRANSFORM BERBASIS RASPBERY PI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :
Putra Wijaya
NIM: 145150300111046

Skripsi ini telah diuji dan dinyatakan lulus pada
11 Juli 2019
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Hurriyatul Fitriyah, S.T, M.Sc	Gembong Edhi Setyawan, S.T, M.T
NIP: 19851001 201504 2 003	NIK: 201208 761201 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

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

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

PERNYATAANORISINALITAS

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, 11 Juli 2018

Putra Wijaya

NIM: 145150300111046

UNIVERSITAS BRAWIJAYA

UNIVERSITAS BRAWIJAYA

PRAKATA

Puji syukur kehadirat Allah SWT, atas limpahan rahmat dan karunia-nya, sehingga penulis dapat menyelesaikan san menyusun laporan skripsi yang berjudul " SISTEM DETEKSI MARKA MEMBUJUR GARIS UTUH DAN PUTUS-PUTUS MENGGUNAKAN METODE THRESHOLDING DAN HOUGH TRANSFORM BERBASIS RASPBERRY PI ".

Penulis menyadari bahwa dalam proses penyusunan dan penyelesaian laporan skripsi ini tidak akan bisa terselesaikan tanpa campur tangan dari pihak-pihak tertentu.Oleh karena itu, penulis ingin mengucapkan rasa hormat dan ucapan terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer, Universitas Brawijaya
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika
3. Bapak Dahnial Syauqy, S.T., M.T., M.Sc. selaku Ketua Progam Studi Teknik Komputer
4. Ibu Hurriyatul Fitriyah, S.T, M.Sc. dan Bapak Gembong Edhi Setyawan, S.T,M.T. yang berkenan untuk meluangkan waktu, pikiran, dan tenaga dalam membimbing dan mengarahkan penulis sehingga proses skripsi ini dapat terselesaikan
5. Kedua orang tua saya, Bapak Drs.Djoko Purnomo dan Ibu Dra.Wiwik Sulistyowati yang selalu memberikan doa dan motivasi kepada penulis sehingga dapat menyelesaikan laporan skripsi
6. Keluarga besar Yamaha N-Max Club Indonesia Chapter Malang Raya yang telah bersedia membantu penulis untuk melakukan penelitian lapangan serta memberikan motivasi untuk menyelesaikan skripsi ini
7. Rekan-rekan WibuTekiKurau yang telah membantu penulis dalam berbagai hal
8. Seluruh civitas akademik Fakultas Ilmu Komputer Universitas Brawijaya yang sudah membantu penelitian selama menempuh studi di Universitas Brawijaya

Dalam pengerjaan skripsi ini, penulis menyadari bahwa skripsi ini masih belum sempurna, sehingga segala bentuk saran dan kritik akan sangat siterima oleh penulis.Akhir kata, penulis harap skripsi ini dapat bermanfaat bagi semua orang

Malang,

Putra Wijaya

wijaya162@gmail.com

repository.ub.ac.id

ABSTRAK

Putra Wijaya, Sistem Deteksi Marka Membujur Garis Utuh dan Putus Menggunakan Metode Thresholding dan Hough Transform Berbasis Raspberry Pi
Pembimbing: Hurriyatul Fitriyah, S.T, M.Sc dan Gembong Edhi Setyawan, S.T, M.T

Marka jalan dengan tipe garis membujur lurus merupakan marka jalan yang sering sekali kita temui di jalan raya,semua jalan raya selalu memiliki marka jenis ini. Meski marka ini memiliki fungsi membagai jalan menjadi dua arah serta membantu pengendara berkendara pada tempatnya, nyatanya masih banyak kecelakaan lalu lintas yang disebabkan tidak konsentrasi dan serta tidak sadar bahwa pengendara dalam posisi yang salah. Atas dasar tersebut sebagai usaha meningkatkan keamanan berkendara yang di padukan dengan teknologi komputer saat ini, maka penelitian ini dibuat sebuah alat yang dapat mendeteksi tipe marka jalan membujur lurus dengan tambahan fitur mengukur posisi kendaraan dengan marka serta memberikan peringatan kepada pengendaraan jika posisi kendaaan dalam posisi yang bahaya atau melanggar marka garis.Masukan dari sistem berupa video yang didapatkan dari penggunaan kamera yang dipasang pada bodi tegah sepeda motor yang sejajar dengan roda depan kendaraan . Metode yang digunakan untuk mengenali objek menggunakan perpaduan Thereshold dengan Hough transform sehingga akan dihasilkan sebuah citra yang dimana marka membujur garis dapat di deteksi serta dapat di hitung panjangnya dengan bantuan Raspberry Pi 3, kemudian hasil dari proses akan ditampilkan pada indikator berupa lampu LED beserta suara dari buzzer sebagai media keluaran. Dari hasil pengujian fungsional alat dapat mendeteksi objek marka serta dapat memberikan informasi kepada pengguna mengenai jarak kendaraan dengan marka garis ,serta sinyal suara jika pengendara berada pada kondisi yang berbahaya.Dalam pengujian akurasi didapatkan akurasi sebesar 85% dari sample video dengan empat kondisi jarak serta 2 kondisi marka jalan membujur garis. Dari pengujian waktu komputasi yang dimulai dari pengenalan objek hingga mengolah data memakan waktu dengan rata-rata 1,265 detik.

Kata kunci : Marka jalan membujur lurus, Keamanan berkendara, Kecelakaan lalu lintas, Threshold, Hough Transform

UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRACT

Putra Wijaya, Intactand Non-Intact Road Line Detection System Using ThresholdingandHough Transform Method Basedon Raspberry Pi
Supervisors : Hurriyatul Fitriyah, S.T, M.Sc dan Gembong Edhi Setyawan, S.T, M.T

Road lines with longitudinal line types are road lines that we often encounter on highways, all highways always have this type of marker. Although this marker has the function of dividing the road into two directions and helping motorists drive in place, in fact there are still many traffic accidents caused by not concentrating and are not aware that the driver is in the wrong position. On this basis as an effort to improve driving safety in combination with current computer technology, this study was made a tool that can detect the type of longitudinal road lines with the addition of features to measure the position of the vehicle with markers and provide a warning to driving if the position is in a position danger or violating line markings. Enter from the system in the form of video obtained from the use of a camera mounted on the body of a motorcycle that is parallel to the front wheels of the vehicle. The method used to recognize objects uses a Thereshold blend with Hough Transform so that an image will be produced where longitudinal line can be detected and can be calculated in length with the help of Raspberry Pi 3, then the results of the process will be displayed on the LED lights and sounds from buzzer as output media. From the results of functional testing the tool can detect markers objects and can provide information to users about the distance of vehicles with line markers, as well as sound signals if the driver is in a dangerous condition. In accuracy testing obtained an accuracy of 85% of the video sample with four distance conditions and two type longitudinal road lines conditions. From testing computation time starting from object recognition to processing data takes an average of 1,265 seconds.

Keywords:*Roadlines,driving safety, trafficaccident,Threshold, Hough Transform*

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
PRAKATA iv
ABSTRAK v
ABSTRACT vi
DAFTAR ISI vii
DAFTAR TABEL xi
DAFTAR GAMBAR xiii
DAFTAR LAMPIRAN xv
BAB 1 PENDAHULUAN 1
1.1Latar Belakang 1
1.2 Rumusan masalah 2
1.3 Tujuan 2
1.4 Manfaat 2
1.5 Batasan masalah 2
1.6 Sistematika pembahasan 3
BAB 2 LANDASAN KEPUSTAKAAN 4
2.1 Tinjauan Pustaka 4
2.2 Dasar Teori 6
2.2.1Marka Jalan Membujur Garis 6
2.2.2Citra Digital 7
2.2.3Jenis Citra Digital 7
2.2.3.1Citra Grayscale 7
2.2.4 Thresholding 8
2.2.5 Hough Transform 9
2.2.6Deteksi Tepi Canny 10
2.2.7ROI11
2.2.8Average Filter 12
2.2.9Open CV 13
2.2.10Raspberry Pi 13
BAB 3METODOLOGI PENELITIAN 15

3.1Studi Literatur 15
3.2 Rekayasa Kebutuhan 16
3.2.1 Penelitian Lapangan 16
3.2.2 Gambaran Umum Sistem 16
3.2.3Kebutuhan Pengguna 17
3.2.4 Kebutuhan Fungsional Sistem 17
3.2.5Kebutuhan Non Fungsional Sistem 17
3.2.6Kebutuhan Perangkat Keras 17
3.2.7Kebutuhan Parangkat Lunak 17
3.3 Perancangan Sistem 18
3.3.1 Perancangan Perangkat Keras 18
3.3.1 Perancangan Perangkat Lunak 18
3.4Implementasi Sistem 18
3.4.1 Implementasi Perangkat Keras 18
3.4.2 Implementasi Perangkat Lunak 18
3.5Pengujian dan Analisis 18
3.5.1Pengujian Fungsionalitas Sistem 19
3.5.2Pengujian Akurasi Deteksi Marka Membujur Garis Utuh dan Putus 19
3.5.3Pengujian Akurasi Deteksi Jarak Antara Kendaraan dengan Marka Membujur Garis 19
3.5.4Pengujian Waktu Deteksi Sistem 19
3.6Kesimpulan dan Saran 19

BAB 4 REKAYASA KEBUTUHAN 20
4.1 Penelitian Lapangan 20
4.2 Gambaran Umum Sistem 23
4.2.1 Tujuan 23
4.2.2 Karakteristik Pengguna 24
4.2.3Lingkungan Operasi 24
4.3 Kebutuhan Pengguna 24
4.4Kebutuhan Fungsional Sistem 24
4.5Kebutuhan Non Fungsional 25

4.6Kebutuhan Perangkat Keras 25
4.7Kebutuhan Perangkat Lunak 26
BAB 5 PERANCANGAN DAN IMPLEMENTASI 28
5.1 Perangkat Keras 28
5.1.1 Perancangan Perangkat Keras 28
5.1.1.1Perancangan Catu Daya 29
5.1.1.2Perancangan Papan Notifikasi 29
5.1.1.3Perancangan User Interface Pada Laptop 31
5.1.1.4 Perancangan Purwarupa 31
5.1.2 Penempatan Kamera 33
5.2 Perangkat Lunak 33
5.2.1 Perancangan Perangkat Lunak 33
5.2.1.1 Ekstraksi Video 34
5.2.1.2 Pengolahan Citra 36
5.2.1 3 Deteksi Marka Garis Lurus dan Deteksi Jarak 46
5.2.1.4Penentuan Status Marka dan Status Jarak 63
5.3 Implementasi Sistem 69
5.3.1 Implementasi Perangkat Keras 69
5.3.2 Implementasi Perangkat Lunak 71
5.3.2.1Ekstraksi Video 71
5.3.2.2 Pengolahan Citra 72
5.3.2.3 Deteksi Marka Garis Lurus dan Deteksi Jarak 74
5.3.2.4Penentuan Status Marka dan Status Jarak 79
BAB 6 PENGUJIAN DAN ANALISIS 81
6.1Pengujian Fungsional Sistem 81
6.1.1 Tujuan Pengujian 81
6.1.2 Pelaksanaan Pengujian 81
6.1.3Prosedur Pengujian 81
6.1.4Hasil Pengujian 82
6.1.5Analisis Hasil Pengujian 83
6.2Pengujian Akurasi Deteksi Marka Membujur Garis Utuh dan Putus .. 83
6.2.1 Tujuan Pengujian 83

6.2.2 Pelaksanaan Pengujian 83
6.2.3Prosedur Pengujian 83
6.2.4Hasil Pengujian 83
6.2.4.1Marka Membujur Garis Utuh 83
6.2.4.2Marka Membujur Garis Putus 87
6.2.4.3Akurasi Deteksi Marka Membujur Garis Utuh dan Marka Membujur Garis Putus 91
6.2.5Analisis Hasil Pengujian 92
6.3Pengujian Akurasi Deteksi Jarak Antara Kendaraan dengan Marka Membujur Garis 92
6.3.1 Tujuan Pengujian 92
6.3.2 Pelaksanaan Pengujian 92
6.3.3Prosedur Pengambilan Video 93
6.3.4Prosedur Pengujian 93
6.3.5Hasil Pengujian 93
6.3.5.1Status Aman 93
6.3.5.2Status Hati - hati 97
6.3.5.3Status Awas 101
6.3.5.4Status Bahaya 104
6.3.5.5Akurasi Deteksi Jarak Kendaraan dengan Marka Membujur Garis 108
6.3.6Analisis Hasil Pengujian 109
6.4Pengujian Waktu Deteksi 109
6.4.1 Tujuan Pengujian 109
6.4.2 Pelaksanaan Pengujian 110
6.4.3Prosedur Pengujian 110
6.4.4Hasil Pengujian 110
6.4.5Analisis Hasil Pengujian 110
BAB 7 PENUTUP 112
7.1 Kesimpulan 112
7.2 Saran 113
DAFTAR REFERENSI 114
LAMPIRAN A KODE PROGAM UTUH 117

repository.ub.ac.id

UNIVERSITAS BRAWIJAYA

DAFTAR TABEL

Tabel 2.1 Perbandingan Fokus Penelitian Referensi ..6
Tabel 2.2 Operasi Citra Biner ..9
Tabel 2.3 Hasil Operasi Citra Biner ..9
Tabel 2.4 Matriks Gaussian Filter ..10
Tabel 2.5 Kernel GX dan Gy Sobel ..11
Tabel 5.1 Perbandingan Ukuran Kernel ..39
Tabel 5.2 Perbandingan Nilai Ambang ..44
Tabel 5.3 Perbandingan Rangkaian Kordinat ..50
Tabel 5.4 Perbandingan Kominasi Nilai ..55
Tabel 5.5 Rekap Nilai Panjang Garis pada Marka Membujur Utuh ..61
Tabel 5.6 Penentu Batas Awas Satu ..64
Tabel 5.7 Penentu Batas Awas Dua ..64
Tabel 5.8 Penentu Batas Awas Tiga ..65
Tabel 5.9 Penentu Batas Hati – hati Satu ..66
Tabel 5.10 Penentu Batas Hati – hati Dua ..66
Tabel 5.11 Penentu Batas Hati – hati Tiga ..67
Tabel 5.12 Kode Progam Ekstraksi Video ..71
Tabel 5.13 Penjelasan Kode Progam Ekstraksi Video ..72
Tabel 5.14 Kode Progam Resize ..72
Tabel 5.15 Penjelasan Kode Progam Resize ..72
Tabel 5.16 Kode Progam Average Filter ..73
Tabel 5.17 Penjelasan Kode Progam Average Filter ..73
Tabel 5.18 Kode Progam RGB ke Grayscale ..73
Tabel 5.19 Penjelasan Kode Progam RGB ke Grayscale ..73
Tabel 5.20 Kode Progam Canny ..73
Tabel 5.21 Penjelasan Kode Progam Canny ..74
Tabel 5.22 Kode Progam ROI ..74
Tabel 5.23 Penjelsan Kode Progam ROI ..74
Tabel 5.24 Kode Progam Probabalistic Houghline Transform ..75
Tabel 5.25 Penjelasan Kode Progam Probabalistic Houghline Transform ..75
Tabel 5.26 Kode Progam Deteksi Marka ..76
Tabel 5.27 Penjelasan Kode Progam Deteksi Marka ..77
Tabel 5.28 Kode Progam Deteksi Jarak ..78
Tabel 5.29 Penjelasan Kode Progam Deteksi Jarak ..79
Tabel 5.30 Kode Progam Penentu Status Marka dan Status Jarak ..79
Tabel 5.31 Penjelasan Kode Progam Penentu Status Marka dan Status Jarak ..80
Tabel 6.1 Tabel Pengujian Fungsional Sistem ..82
Tabel 6.2 Deteksi Marka Membujur Garis Utuh ..84
Tabel 6.3 Deteksi Marka Membujur Garis Putus ..87
Tabel 6.4 Akurasi Deteksi Marka Membujur Garis ..91
Tabel 6.5 Status Aman ..94
Tabel 6.6 Status Hati-hati ..97
Tabel 6.7 Status Awas ..101

Tabel 6.8 Status Bahaya 105
Tabel 6.9 Akurasi Deteksi Jarak dari Marka Membujur Garis 108
Tabel 6.10 Waktu Deteksi Sistem 111

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR GAMBAR

Gambar 2.1 Posisi tidak Terdeteksi 5
Gambar 2.2 Marka Membujur Garis Utuh 6
Gambar 2.3 Marka Membujur Garis Putus 7
Gambar 2.4 Konversi RGB ke Grayscale 8
Gambar 2.5 Segmentasi Threshold 9
Gambar 2.6 Sudut Pembentuk Garis Citra 10
Gambar 2.7 ROI 12
Gambar 2.8 Average Filter 13
Gambar 2.9 Perangkat Pemroses 14
Gambar 3.1 Diagram Alir Metode Penelitian 15
Gambar 3.2 Denah Lokasi Penelitian 16
Gambar 4.1 Detail Jalan Raya 20
Gambar 4.2 Posisi Kondisi Aman 21
Gambar 4.3 Posisi Kondisi Hati-hati 21
Gambar 4.4 Posisi Kondisi Awas 22
Gambar 4.5 Posisi Kondisi Bahaya 22
Gambar 4.6 Blok Diagram Sistem 23
Gambar 5.1 Skema Perancangan dan implementasi 28
Gambar 5.2 Rancangan Catu Daya 29
Gambar 5.3 Rancangan Papan Notifikasi 30
Gambar 5.4 Rancangan User Interface 31
Gambar 5.5 Purwarupa Sisi Dalam 32
Gambar 5.6 Purwarupa Sisi Luar 32
Gambar 5.7 Penempatan Kamera 33
Gambar 5.8 Flowchart Perancangan Perangkat Lunak 34
Gambar 5.9 Flowchart Ekstraksi Video 35
Gambar 5.10 Flowchart Pengolahan Citra 36
Gambar 5.11 Flowchart Resize 37
Gambar 5.12 Average Filter 38
Gambar 5.13 Hasil Proses Average Filter 41
Gambar 5.14 Flowchart Konversi RGB ke Grayscale 42
Gambar 5.15 Hasil Konversi RGB ke Grayscale 42
Gambar 5.16 Flowchart Deteksi Tepi Canny 43
Gambar 5.17 Hasil Proses Deteksi Tepi Canny 46
Gambar 5.18 Flowchart Deteksi Marka Garis Lurus dan Deteksi Jarak 47
Gambar 5.19 Flowchart ROI 48
Gambar 5.20 Kordinat Gambar 49
Gambar 5.21 Hasil Proses ROI 52
Gambar 5.22 Flowchart Probabalistic Houghline Transform 53
Gambar 5.23 Hasil Proses Probabalistic Houghline Transform 58
Gambar 5.24 Flowchart Deteksi Marka 59
Gambar 5.25 Hasil Proses Deteksi Marka 61
Gambar 5.26 Flowchart Deteksi Jarak 62

Gambar 5.27 Flowchart Penentu Status Marka dan Status Jarak 68
Gambar 5.28 Hasil Proses Penentu Status Marka dan Status Jarak 68
Gambar 5.29 Perancangan Sistem dari Dalam 69
Gambar 5.30 Perancangan Sistem dari Luar 70
Gambar 5.31 Penempatan Kamera Tampak Depan dan Samping 71
Gambar 6.1Pengujian Fungsional Sistem 82
Gambar 6.2 Waktu Deteksi Sistem 110

DAFTAR LAMPIRAN

Kode Progam 117

repository.ub.ac.id

UNIVERSITAS BRAWIJAYA

BAB 1Pendahuluan

1.1 Latar Belakang

Banyaknya kecelakaan yang terjadi di Indonesia menimbulkan kekhawatiran bagi penguna jalan. menurut sumber dari website korlantas, total kecelakaan yang terjadi di indonesia pada september 2017 sampai desember 2017 tercatat sejumlah 26.291 kecelakaan (korlantas, 2018) . Salah satu penyumbang jumlah kecelakaan adalah pelanggaran marka jalan.

Pelanggaran marka jalan berada di urutan nomor tiga dari sepuluh penyebab kecelakaan di indonesia, menurut Korlantas polri bapak Ajun Komisaris Besar Polisi Aldo Siahan diambil dari website Tempo (korlantas polri, 2017). Banyak pengguna jalan melakukan pelanggaran ini, karena kebanyakan pengendara lebih fokus pada kendaraan depan dan lebih menggunakan perasaan untuk memperhatikan marka jalan membujur lurus maupun putus-putus. Meskipun pemerintah sudah membuat Undang-Undang Nomor 22 tahun 2009 yaitu dengan ancaman kurungan dua bulan penjara atau denda sebesar 500 ribu rupiah bagi pelanggar(fitradian, 2018) , tetapi tetap saja banyak pengguna jalan melakukan pelanggaran.

Dengan kemajuan teknologi saat ini, kita mampu membangun perangkat pintar *embedded system* yang di kombinasiakan dengan teknologi pengolahan citra untuk memberikan peringatan kepada pengendara mengenai potensi potensi pelanggaran marka jalan khususnya marka jalan membujur. Perangkat ini nantinya dapat membantu pengguna jalan untuk mendeteksi jarak kendaraan dari marka jalan serta dapat memberikan peringatan kepada pengendara jika posisi kendaraan melanggar marka garis. Teknologi pengolahan citra membawa kemudahan dalam mengidentifikasi objek yang berada di jalan raya. Dengan metode *Threshold* dan *Hough Transform*,objek dibutuhkan oleh sistem dapat mendeteksiadanya tidaknya marka jalan, jarak kendaraan dengan marka,serta membedakan dua jenis marka jalan sesuai apa yang dibutuhkan oleh pengguna jalan untuk meminimalisir keterbatasan pengguna jalan. Keterbatasan penguna jalan adalah seringnya pengguna jalan tidak menyadari bahwa posisi kendaraan yang dikendarai telah melanggar peraturan yang ditetapkan karena beberapa faktor seperti mengantuk, kondisi badan tidak fit sehingga tidak fokus, serta melamun.

Dari keterbatasan pengendara tersebut maka fungsi dari perangkat *embedded system* ini dibutuhkan untuk membantu pengendara menempatkan posisi kendaraan yang benar dengan bantuan sebuah notifikasi dari alat ini, agar dapat meminimalisir terjadinya kecelakaan maupun pelanggaran jalan. Perangkat juga dapat dijadian asisten bagi pengendara yang tidak berkonsentrasi. Sistem nantinya dilakukan dalam tingkat simulasi dengan input video yang diambil pada jalan raya yang sebenarnya serta output berupa notifikasi.

1.2 Rumusan Masalah

Berdasarkan dari uraian pada latar belakang, maka dapat diambil beberapa rumusan masalah berikut:

1. Bagaimana cara merancang sistem deteksi marka membujur garis Utuh dan putus pada Raspberry Pi?
2. Bagaimana tingkat akurasi deteksi marka membujur garis lurus utuh dan putus?
3. Bagaimana tingkat akurasi sistem menentukan jarak antara kendaraan dengan marka membujur garis?
4. Berapa rata - rata waktu yang dibutuhkan untuk mendeteksi marka membujur garis utuh dan putus beserta menentukan jarak?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah dijabarkan, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Sistem dapat memberikan peringatan kepada pengendara
2. Sistem dapat mendeteksi marka membujur garis utuh dan putus.
3. Sistem dapat mengukur jarak antara kendaraan dengan marka membujur garis yang terdeteksi.
4. Sistem dapat menghitung rata - rata waktu deteksi.

1.4 Manfaat Penelitian

Manfaat yang didapatkan dari penelitian ini sebagai berikut:

1. Pengendara mendapatkan peringatan jika melewati marka membujur garis utuh.
2. Pengendara dapat memposisikan kendaraan dengan benar.
3. Meningkatkan hubungan interaksi manusia dengan komputer.
4. Meminimalisir kecelakaan yang disebabkan oleh marka membujurgaris.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini sebagai berikut:

1. Marka jalan yang dijadikan objek penelitian, hanyalah marka garis membujur yang berada pada jalan yang menjadi penelitian lapangan.
2. Marka jalan yang digunakan sebagai objek terdiri dari 2 jenis,yaitu marka membujur garis utuh, dan putus.
3. Resolusi Video input yang digunakan adalah 1280x720 piksel 30 fps
4. Tipe kendaraan yang digunakan untuk pengambilan video masukan adalah tipe kendaraan roda dua dengan transmisi otomatis dengan merek YAMAHA N-MAX.
5. Kendaraan berjalan pada kecepatan konstan 25km/jam
6. Waktu pengambilan video pada pukul 15.30 sampai 17.15 WIB

1.6 Sistematika Pembahasan

Sistematika pembahasan laporan sebagai berikut:

BAB 1 PENDAHULUAN

Pada bab ini penulis akan menjelaskan alasan mengapa melakukan penelitian ini beserta masalah yang diangkat pada penelitian.

BAB 2 LANDASAN KEPUSTAKA

Pada bab ini akan dijelaskan berbagai studi literatur terkait dengan penelitian ini serta dasar teori yang digunakan pada penelitian.

BAB 3 METODOLOGI

Pada bab berisikan penjelaskan tahap - tahap dari pengelolaan penelitian. Yang berisikan studi literatur, penelitian lapangan, rekayasa kebutuhan, perancangan sistem, tahap implementasi, sampai dengan penarikan kesimpulan.

BAB 4 REKAYASA KEBUTUHAN

Pada bab ini berisikan semua yang dibutuhkan untuk memperlancar proses perancangan

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini berisikan perancangan sistem perangkat keras yang berupa skematik yang berdasarkan pada kebutuhan perangkat keras, seperti skematik papan notifikasi pada mikrokontroller, Pada sisi perangkat lunak terdapat flowchart yang menjelaskan alur progam yang dijalankan pada mikrokontroller.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini, dijelaskan tahap pengujian yang dilakukan pada penelitian ini ,pengujian terdiri dari tujuan pengujian,pelaksanaan pengujian,prosedur pengujian ,hasil pengujian, hasil setiap pengujian akan dianalisis.

BAB 7 PENUTUP

Pada bab ini berisi kesimpulan keseluruhan mengenai isi dari penelitian ini. Selain itu berisi saran yang dapat digunakan untuk mengembangkan penelitian ini lebih lanjut.

DAFTAR PUSTAKA

Pada bab ini memuat keseluruhan dokumen yang dijadikan sebagai rujukan pada penelitian ini.

BAB 2Landasan Kepustakaan

Bab ini mengutaran referensi yang berasal dari penelitian terdahulu terkait dengan penelitian saat ini yang mendasari penelitian saya.

2.1 Tinjauan Pustaka

Kajian pustaka yang pertama adalah penelitian membuat sebuah alat yang dapat mendeteksi marka jalan menggunakan kamera secara live dengan format video(Pratama , 2015).Objek yang diteliti memiliki persamaan dengan penelitian yang saya kerjakan. Metode utama yang digunakan adalah *Threshold*, alasan menggunakan metode ini untuk mendeteksi marka jalan yang dimana objek*Interest* dan *Background* memiliki nilai itensitas yang sangat yang berbeda jauh sehingga dari rata rata objek*Interest,* hasil pengujian penulis, nilai *Threshold* yang didapat agar bisa mendeteksi marka jalan dengan baik sebesar 75 dari 100% ,Sedangakan pada metode *Regionprob* dapat berjalan dengan baik untuk membantu metode *Threshold* dengan nilai minimum sebesar 500 *blop* untuk melakukan pengkotakan. Kekurangan dari sistem yang dibuat adalah sistem hanya berada pada lingkup simulasi tidak berupa alat.

Sistem Deteksi Marka Pembatas Jalan Menggunakan Metode Voting dan Sampling (Pollo, 2013), menerangkan sebuah alat yang dapat mendeteksi marka jalan dengan menggunakan metode voting ,yang dimana sebelum menggunakan metode ini penulis mengumpulkan sample yang didapat dari penelitian dilokasi, seperti keadaan garis marka saat jalan lurus,berbelok, cahaya langsung, jalan dengan bayangan, Cahaya kurang.Semua sampel akan di kumpulkan dan dikelompokan. Sehingga dari pengambilan gambar secara berturut tersebu dapat menentukan batas daerah voting yang didapat letak posisi garis dengan mengambil data nilai dari kiri ketengah sebagai sisi kiri dan dari tengah ke kanan sebagai sisi kanan.tiap gambar yang diambil akan dihitung nilai samping kiri dan samping kanan.Selanjutnya hasil dari hitungan voting akan dimasukan kedalam persamaan yang dimana hasil dari persamaan akan menjadi parameter yang akan dimasukan kedalam sistem. Dari hasil pengujian penulis didapatkan sebuah kesimpulan bahwa sistem yang dibuat mempunya tingkat keberhasilan diatas 75% dengan pengambilan foto sebanyak 300 kali saat sistem berjalan, serta sistem telah berhasil mengenali 3 kondisi jalan yaitu lurus, berbelok, dan pindah jalur. Kekurangan sistem adalah belum adanya output perangkat keras.

Metode *Multiresolution Hough Transform digunakan sistem untuk* deteksi marka jalan dan posisi (Chandra,2013). Fungsi dari alat ini adalah memberikan informasi marka jalan pada pengguna mobil.proses pertama menggunakan *preprocesing* citra padagrayscale dan edge detection sebagai syarat dalam proses pada fungsi *multiresolution hough transform.* garis-garis yang muncul pada citra akan dicari menggunakan *multiresolution hough transform*.Langkah selanjutnya garis garis yang tidak dibutuhkan akan dihilangkan menggunakanproses *image cropping*.Setelah didapatkan garis maka akan dicari titik awal dan titik akhir pada

garis beserta panjang garis. Proses selanjutnya melakukan proses *inverse perspective mapping* yang berguna untuk memproyeksikan citra dari bidang citra ke jalan. Jika antara garis sesuai dengan jarak maka kedua garis tersebut dianggap sebagai marka. Contoh percobaan dapat dilihat pada Gambar 2.1.

Gambar 2.1 Percobaan Deteksi Marka

Setelah marka jalan berhasil terdeteksi, selanjutnya akan dilakukan perhitungan posisi marka jalan terhadap kamera. Hasil dari pengujian penulis didapatkah bahwa tingkat akurat mendeteksi marka mencapai 98% dan juga sistem dapat menentukan jarak kendaraan dengan marka jalan yang terdeteksi oleh sistem. Sistem belum dilengkapi sebuah perangkat peringatan bagi pemakai jika posisi kendaraan melewati garis marka mapun kondisi kondisi yang sangat *urgent*.

Sistem deteksi garis menggunakan metode *probalistic houghline transform* menggunakan raspberry pi (Patrik, 2017) menjelaskan bahwa sebuah objek garis dapat dideteksi dengan sempurna dengan menggunakan metode hougline transform dengan media embedded sistem raspberry pi. Objek yang diteliti pada penelitian ini memiliki objek *interest* dan perangkat keras yang sama dengan penelitian yang saya kerjakan. Kekurangan dari penelitian tersebut adalah tidak ditemukanya perangkat tambahan untuk memproses hasil dari pembacaan objek garis.

Dari keempat referensi yang dapat dari beberapa sumber memiliki objek penelitian yang sama dengan penelitian yang saya kerjakan, yaitu berhubungan dengan objek garis,marka jalan membujur, jalan raya. Pada referensi yang dicantumkan memiliki fokus penelitian yang berbeda yang dapat dilihat pada Tabel 2.2. Tingkat keberhasilan rata-rata dari semua referensi sebesar 80% dari target tiap penelitian yang dilakukan,sehingga penelitian ini nantinya dapat memiliki tingkat keberhasilan diatas 80 % .

Tabel 2.1Perbandingan Fokus Penelitian Referensi

Referensi Satu	Referensi Dua	Referensi Tiga	Referensi Empat
Fokuspenelitian adalah pada deteksi marka melintang jalan.	Fokus penelitian adalah mendeteksi marka jalan dalam 4 kondisi yaitu marka membujur garis utuh, melengkung, cahaya posisi didepan serta pada kondisi cahaya gelap	Fokus penelitian adalah mengukur jarak kendaraan dengan marka membujur garis pada marka buatan didalam ruangan	Fokus penelitian adalah mendeteksi garis dari tepian objek yang tertangkap pada webcame menggunakan mikrokontroller *Raspberry Pi*

2.2 Dasar teori

Pada bagian dasar teori menjelaskan tentang teori - teori yang digunakan untuk dapat mendeteksi objek penelitian seperti metode pada *computer vision*, perangkas pemroses,dan perangkat lunak pendukung dalam membangun sistem pada penelitian.

2.2.1 Marka Membujur Garis

Marka membujur garis adalah suatu tanda yang berada di permukaan jalan atau di atas jalan berbentuk garis membujur yang berfungsi untuk mengarahkan arus lalu lintas serta membagi ruas jalan. Jenis- Jenis marka membujur garis adalah sebagai berikut.

1. Marka membujur Garis Utuh

   Menurut keputusan menteri perhubungan nomor 60 tahun 1993 pasal 17 butir 1 menjelaskan marka membujur berupa garis utuh berfungsi sebagai larangan bagi kendaraan melintasi garis tersebut (keputusan menteri perhubungan, 2014) . Menurut undang undang diatas disimpulkan bahwa garis marka membujur utuh adalah garis tidak boleh dilewati oleh kendaraan karena pada daerah marka tersebut terdapat potensi bahaya bagi pengemudi.

Gambar 2.2 Marka Membujur Garis Utuh
Sumber: Korlantas (2000)

2. Marka Membujur Garis Putus

   Keputusan menteri perhubungan nomor 60 tahun 1993 pasal 18 butir 1 dan b menjelaskan marka membujur putus-putus berfungsi sebagai

mengarahkan lalu lintas, memperingatkan akan ada marka membujur garis utuh di depan,dan pembatas antara dua arah (keputusan menteri perhubungan, 2014) . Dari pernyataan tersebut dapat disimpulkan marka membujur putus-putus di gunakan sebagai peringatan jika kepada pengemudi jika didepan akan ada marka membujur utuh.

Gambar 2.3 Marka Membujur Garis Putus
Sumber: Korlantas (2000)

2.2.2 Citra Digital

Citra digital adalah sebuah gambaryang melalui bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling adalah besar ukuran kotak-kotak yang ditata pada baris dan kolom. Atau bisa disebut, sampling pada gambar adalah besar dan kecilnya ukuran *pixel* (titik) pada gambar, dan kuantisasi adalah nilai kecerahan pada piksel yang dinyatakan dalam nilai tingkat keabuan.Sesuai dengan jumlah bit biner yang digunakan oleh mesin, dengan kata lain kuantisasi pada gambar menyatakan jumlah warna yang ada pada gambar. (Basuki,2005).

2.2.3 Jenis Citra Digital

Suatupixel secara memiliki rentang nilai antara 0 -255, yang dimana nilai 0 adalah nilai minimum piksel dan 255 adalah nilai maksimun piksel.Namun diantara banyak piksel ada juga berbeda, tergantung dengan jenis nilai warna.Ada banyak tipe citrasalah satunya yang digunakan pada penelitian citra grayscale.

2.2.3.1 Citra*Grayscale*

Grayscale adalah jenis colorspace yang hanya memiliki 1 nilai intensitas dalam setiap pixelnya. Metode inidigambarkan dalam warna gradasi hitam ke putih. Warnahitam memiliki intensitasterendah yaitu 0, dan warna putih memiliki nilai itensitas warna tertinggi yaitu 255 (Fauzah, 2014). Metode yang digunakan adalah konversi RGB ke *grayscale*. Metode yang digunakan dalam menggunakan *Citra Grayscale* sebagai berikut:

a. Luminosity

Proses dari algoritma ini dengan mengalikan setiap nilai R G B dengan konstanta tertentu yang sudah ditetapkan nilainya, selanjutnya hasil perkalian seluruh nilai R GB dijumlahkan satu sama lain. Rumus matematisnya
adalah :

$$Grayscale = (0.299 * R) + (0{,}587 * G) + (0{,}114 * B) \quad (2.1)$$

Contoh gambar konversi RGB ke Grayscale seperti pada Gambar 2.4.

Gambar 2.4Konversi RGB ke Grayscale
Sumber: Mathwork (2018)

Bagian kiri adalah citra RGB(asli) sedangakan pada gambar sebelah kanan adalah konversi dari citra RGB ke citra *grayscale*.

2.2.4 *Thresholding*

*Threshodling*adalah metode segmentasi yang sederhana, yakni bernilai 1 padaintensitas yang masuk dalam ambang batas atas threshold dan warna putihmemberikan nilai 0 pada ambang batas bawah threshold warna hitam(Rahma, 2013). Parameter yang dibutuhkan di dalam *thresholding*adalah dengan memberikan nilai ambas batas atas dan ambang bawah seperti pada rumus berikut:

$$f_0(x,y) = \begin{cases} 0, & f_1(x,y) < (\text{'nilai threshold'}) \\ 255, & f_1(x,y) \geq (\text{'nilai threshold'}) \end{cases} \quad (2.2)$$

Metode ini hanya terdapat ada satu batas yaitu batas nilai batas threshold yang ditentukan, jika piksel mimiliki nilai dibawah nilai *Threshold*maka nilai akan katakan 0 tetapi jika nilai pixel berada diatas nilai *Threshold*maka dianggap 255. Contoh perhitungan dapat dilihat pada bagian berikut:

Contoh:
Sebuah citra mempunyai citra *grayscale* 255 warnadengan ukuran 5x5 piksel dengan batas bawah dan batas atas sebagai berikut:

Threshold: 125

Maka citra grayscale dengan piksel 5x5 diTabel 2.2.

Tabel 2.2Operasi cCtra Biner

40	140	30	20	70
30	50	70	150	200
80	140	100	130	60
150	100	120	50	140
60	129	30	20	30

Pada Tabel 2.2 adalah contoh piksel gambar asli yang akan di proses

Akan menjadi piksel 5x5 diTabel 2.2.

Tabel 2.3 Hasil Operasi Citra Biner

0	255	0	0	0
0	0	0	255	255
0	255	0	255	0
255	0	0	0	255
0	255	0	0	0

Pada Tabel 2.3 adalah hasil piksel dari proses *threshold*.
Sehingga didapatkan hasil menjadi citra threshold sesuai Gambar 2.5.

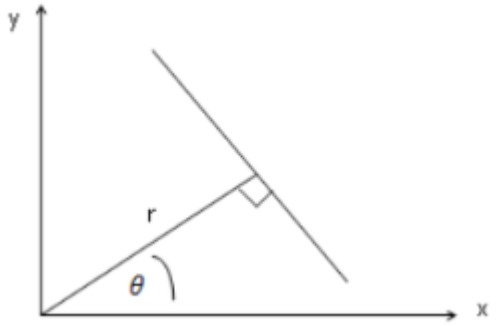
Gambar 2.5Segmentation Thresholding
Sumber: Mathwork (2018)

Pada bagian kiri adalah citra RGB , sedangkan bagian kanan adalah hasil konversi dari citra RGB ke citra segmentasi *thresholding.*

2.2.5 *HoughlineTransform*

HoughlineTransform awal mula dikenalkan oleh Paul Hough pada tahun 1962 untuk mendeteksi garis lurus.Untuk mengisolasi suatu objek pada citra dengan menemukan batas-batasnya boundary detection digunakan teknik

transformasi citra yaitu *Hough transform*(Opencv, 2017). Tujuan dari sebuah transformasi adalah untuk menemukan suatu objek bentuk kurva seperti garis.



Gambar 2.6Sudut Pembentuk Garis Citra
Sumber: Houghline (2018)

Probabalistic Houghline Transform metode ini memiliki proses yang sama dengan *Hough Transform* telah dijelaskan tersebut, yang membedakan adalah pada metode ini proses menampilkan objek garis yang dicari dengan membatasi panjang pikselnya dengan dua variabel tambahan yaitu *minLineLength* yaitu panjang piksel minimal yang ditampilkan dan *maxLineGap* yaitu rentang panjang maksimal yang akan dinyatakan sebuah baris. Jika objek yang di proses memiliki panjang piksel dibawah *MinLineLenght* maka bukan dianggap garis dan jika nilai panjang piksel diatas MaxLineGap tidak diagap sebuah garis.

2.2.6 Deteksi Tepi *Canny*

Tepian dari suatu objek adalah informasi penting bagi citra yang bersangkutan. Tepian citra adalah objek yang terkandung dalam citra tersebut, seperti bentuk dan ukuranya serta informasi tentang teksturnya . metode yang digunakan untuk mendeteksi tepi adalah deteksi *canny*. Metode ini pertama di kembangkan oleh (John F.Canny, 1986) menggunakan algoritma multi-tahap digunakan untuk menemukan tepi dalam citra . dalam implementasinya metode *canny* memiliki beberapa tahap yaitu sebagai berikut:

1. Pertama adalah menghilangkan *noise* yang muncul pada citra dengan menggunakan *gaussian filter.* Penghalusandengan *gasussian filter* bertujuan untuk mengurangi piksel yang tidak perlu. Hasilnya citra akan terlihat lebih buram. Hal ini di maksudkan untuk mendapatkan hasil tepian citra yang sebenarnya. Jika tidak dilakukan proses ini maka garis garis halus akan terdeteksi pada Tabel 2.4.

Tabel 2.4Matriks*Gaussian* Filter

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	8	12	8	4

3	4	5	4	3

2. Langkah kedua dengan melakukan deteksi tepi menggunakan operator deteksi *Sobel* ,dengan mencari horizontal (*Gx*) dan vertikal (*Gy*).selanjutnya adalah mengalikan matriks *Gaussian Filter* dengan kernel Gx dan Gy pada Tabel 2.5.

Tabel 2.5 Kernel Gx dan Gy *Sobel*

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Setelah mengalikan kedua matriks *Gaussian Filter pada tabel 2.4* dengan kernel *Gx* dan *Gy*pada tabel 2.5, maka langkah selanjutnya adalah mentukan nilai gradien dan arah gradien pada piksel.Dengan persamaan 2.3.

$$G = \sqrt{G\frac{2}{x} + G\frac{2}{y}} \quad (2.3)$$

Keterangan
$G : gradien$
$Gx : sumbu\ x(horizontal)$
$Gy : sumbu\ y\ (vertikal)$

3. Langkah ketiga dengan memperkecil garis tepi dengan menerapkan fungsi non maximum suppression, maka dihasilkan garis tepi yang ramping. *NonMaksimum suppression*berfungsi untuk mencari titik tertinggi pada besaran besaran tepi.

4. Tahap ke-empat dari proses Deteksi tepi *canny*adalah melakukan dobel *threshold* pada tiga kondisi :

- Jika nilai gradien piksel lebih besar dari batas atas, maka piksel dianggap sebagai tepi piksel

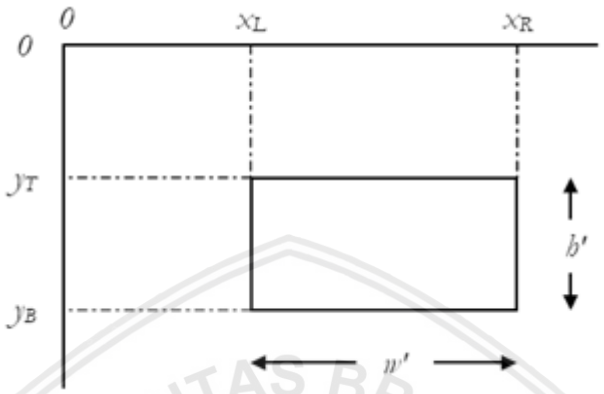
- Jika nilai gradien piksel lebih kecil batas bawah , maka piksel tersebut bukan tepi piksel

- Jika nilai gradien piksel diantara nilai batas bawah dan batas atas dan piksel terhubung sebuah tepi piksel yang dimana piksel yang terhubung adalah piksel yang berada di antara ambang batas atas dan batas bawah maka juga dianggap tepi piksel

2.2.7*ROI*

ROI (*region of interest*) adalah sebuah proses yang digunakan untuk mencari objek yang akan diteliti, metode ini memanfaatkan titik kordinat dari piksel

gambar yang akan diproses untuk dilakukan proses gambar, sesuai banyak titik yang ditentukan. Frame yang akan dilakukan proses gambar adalah salinan dari gambar asli yang telah dikosongkan tetapi memiliki resolusi yang sama. Gambar2.7 adalah contoh cara mendapatkan titik kordinat untuk melakukan proses gambar.



Gambar 2.7Titik Kordinat ROI
Sumber: ROI (2013)

Pada Gambar 2.7 digambarkan terdapat empat titik kordinat yaitu xl,xr dan yr,yb agar keempat titik membentuk bidang yang menjadi objek yang diteliti maka ke-empat titik tersbut harus dicari titik gambarnya. Hasilnya didapatkan empat titik potong yang membentuk suatu bidang sebagai berikut:

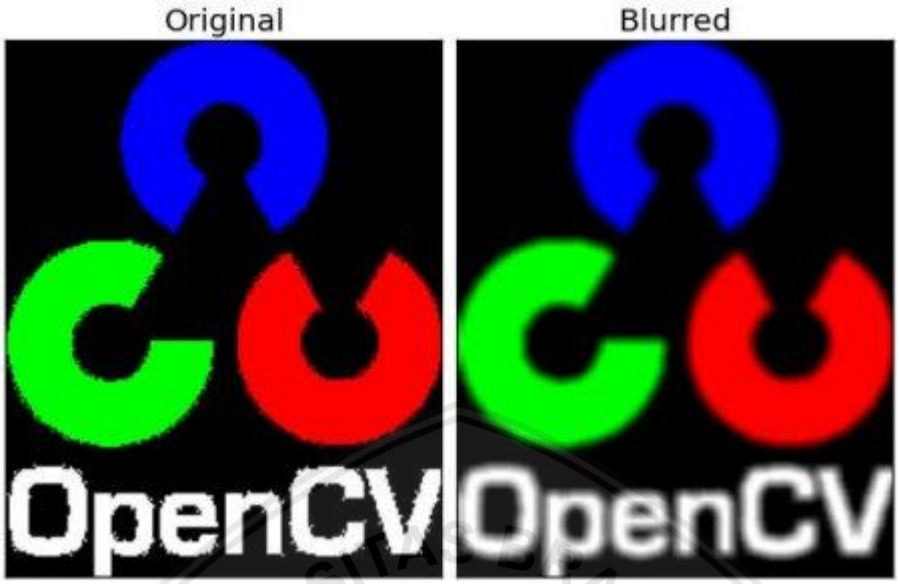
a. Titik gambar antara titik kordinat xl dengan yr sehingga menjadi titiik gambar (xl,yr).
b. Titik gambar antara titik kordinat xl dengan yr sehingga menjadi titiik gambar (xl,yb).
c. Titik gambar antara titik kordinat xr dengan yr sehingga menjadi titiik gambar (xl,yr).
d. Titik gambar antara titik kordinat xr dengan yb sehingga menjadi titiik gambar(xl,yb).

Proses selanjutnya ada hasil gambar yang telah di masking warna hitam pada bidang gambar sesuai kordinat diatas akan di tumpuk pada gambar asli, secara proses kedua piksel gambar sebenarnya dilakukan proses perkalian dua gambar. Sehingga bagian yang tertutup oleh bidang gambar akan nampak hitam dan daerah yang tidak tertutup gambar bidang nampak gambar aslinya, sehingga didapatkan gambar yang seolah olah telah dipotong.

2.2.8*Average Filter*

Average filter juga disebut dengan *mean* filter yaitu filter rata-rata. Average filter mengganti nilai piksel pada posisi(x,y) dengan nilai rata-rata dari nilai piksel yang berada tetangga disekitarnya(Andre Wedianto,2016). Luasan nilai piksel ditentukan dari besar kecilnya *kernel* yang digunakan, rata rata kernel yang digunakan pada metode ini memiki nilai ganjil karena kernel bagian tengah akan

menjadi pusat lusaan dari proses *average filter* ini. Contoh hasil dari proses *average filter* dapat dilihat pada Gambar 2.8.



Gambar 2.8Average Filter
Sumber: Average Filter (2013)

2.2.9 *OpenCV*

OpenCVadalah sebuah progam *open source* antar muka untuk melakukan proses pengolahan citra*computer vision*. OpenCV banyak digunakan pada penelitian citra yang menggunakan mikrokontroller *Raspberry Pi*. Aplikasi ini dapat berjalan pada bahasa pemrogaman C, C++,Python,dan Matlab serta pada OS windows ,linux,MacOS dan android. Contoh implementasi yang dapat dilakukan dengan OpenCV seperti pengenalan wajah, pengenalan jalan raya serta implementasi lain yang masuk dalam kategori computer vision. OpenCV memang didesai untuk melakukan aplikasi real time,dan memiliki akuisisi yang cukup baik pada gambar dan video. Awal kemunculan OpenCV di keluarkan oleh perusahaan intel menggunkan library dari intel image processing library,kemudian depency tersebut dihilangkan sehinga terlahirlah OpenCV sebagai *stand alone library*

2.2.10 *Raspberry Pi*

Raspberry pi adalah sistem embeddedyang digunakan pada penelitian. Mikrokontroller ini mempunyai spesifikasi prosesor khusus dari Broadcom BCM2837 disertai prosesor ARM Cortex-A53 ya empat core berkecepatan1,2 GHz. Cache memori 1 32Kb dan memori 2 512Kb. Grafiknya VideoCore IV, dan memori 1 GB.

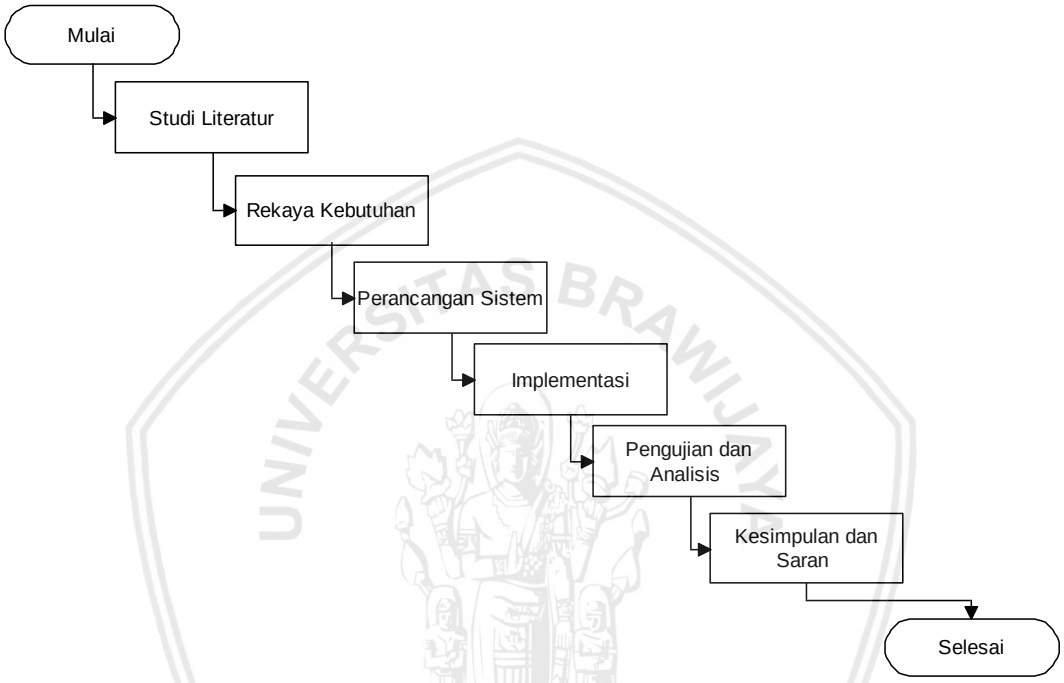
Untuk *pin input output* pada rasberry pi 3 menggunakan GPI sebanyak 26, 2 pin DC 3.3v,2 pin DC 5V, 8 pin ground, dan 2 pin I2C ID EEPROM. Dalam penomoran GPIO terdapat dua jenis sistem yaitu BCM penomoran pin oleh software dan GPIO Board penomoran langsung yang tampak pada board.Tetapi hanya salah satu sistem penomoran saja yang didapat digunakan pada sistem.

Untuk Koneksifitas rasberry pi 3 menggunakan Broadcom BCM43438 yang menyediakan LAN 2,4GhZ 802.11n dan dukungan Bluetooth 4.1 rendah daya. Serta chip USB SMSC LAN9514 sehingga mendukung konekstifitas ethernet dan USB empat channel pada satu board. Tampilan dari *Raspberry Pi* dapat dilihat pada Gambar 2.9.

Gambar 2.9Perangkat Pemroses
Sumber: Rasberry Pi 3 Model B (2016)

BAB 3METODOLOGI PENELITIAN

Pada bab ini dijelaskan tentang langkah-langkah yang dilakukan dalam pengerjaan penelitian. Tahapan metodologi penelitian dapat dilihat pada



Gamabar 3.1

Gambar 3.1 Diagram Alir Metode Penelitian

3.1Studi Literatur

Studi Literatur dilakukan untuk mempelajari teori, konsep,metodedari literatur ilmiah yang berkaitan. Pada studi literatur akan mempelajari konsep dari objek penelitian serta proses pengolahan gambar.

a. Lalu Lintas

   Pada bagian ini membahas tentang teori yang terkait dengan Marka jalan, seperti marka membujur garis utuhdan putus yang nantinya akan menjadi objek pada penelitian ini.

b. Pengelolahan Citra Digital

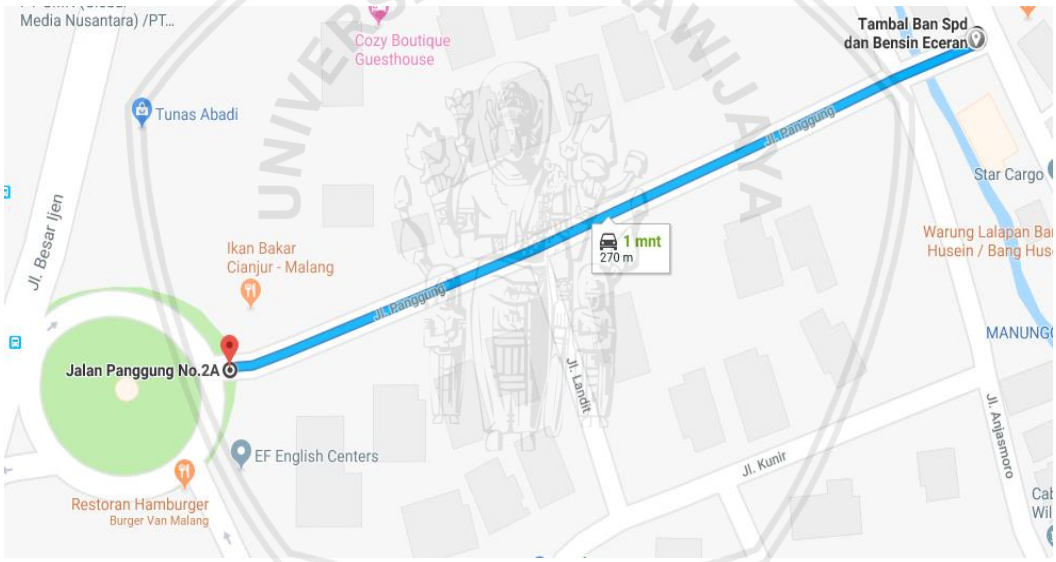
   Pada Bagian ini dilakukan kajian tentang teori yang dibutuhkan untuk melakukan pengolahan citra digital, diantaranya adalah proses *thresholding*,*houghline transform*, serta metode pendukung pengolahan citra yang dapat membantu proses mendeteksi marka membujur garis ,serta referensi yang terkait dengan computer vision dan Open CV Library.

3.2 Rekayasa Kebutuhan

Rekayasa kebutuhan Sistem dilakukan untukmenganalisis kebutuhan apa saja yang dibutuhkan untuk merancang sistem deteksi marka membujur garis utuh dan putus,

3.2.1 Penelitian Lapangan

Penelitian lapangan dilakukan untuk mengetahui kondisi marka jalan raya yang sebenarnya, sehingga hasil dari penelitian yang dilakukan di lapangan nantinya akan dibawa untuk menunjang keberhasilan alat menangkap objek yang diteliti. Pengambilan data yang diambil meliputi jarak jalan pada kondisi marka membujur lurus utuh, jarak jalan pada kondisi lurus pada posisi marka jalan garis lurus putus putus dan lebar permukaan jalan raya.



Gambar 3.2 Denah Lokasi Penelitian
Sumber : Google Map, (2017)

Pada Gambar 3.2 diambil dari aplikasi google map, kondisi jalan yang menjadi tempat penelitian memiliki suasana relatif sepi namun memiliki kualitas garis marka yang cukup baik,serta lebar jalan yang panjang. Dari semua kondisi diatas maka kemanan penguji dan keberhasilan pengambilan data dapat membantu tingkat keberhasilan sistem.

3.2.2 Gambaran Umum Sistem

Padagambaran umum sistem akan dijelaskan keseluruhan sistem deteksi marka membujur garis utuh danputus. Sistem terdiri dari input, pemroses,dan

output. Pada Gambaran umum sistem akan dijelaskan dalam bentuk diagram yang terbagi dalam tiga blo yaitu *input*, proses,dan *output*.

Blok input berisi semua data masukan yang dibutuhkan sistem untuk di proses,blok proses berisi pengolahan data yang diterima dari blok input agar nantinya menjadi data yang dikirimkan pada blok output,dan blok output berisikan dua perangkat yaitu lampu LED dan *Buzzer.* Dari semua blok yang dijelaskan diatas akan menjadi satu kesatuan pada sub bab gambaran umum sistem.

3.2.3 Kebutuhan Pengguna

Kebutuhan pengguna menjelaskan tentang apa saja yang dibutuhkan oleh pengguna pada sistem. Kebutuhan pengguna adalah alasan utama pembuatan sistem ini, sehingga nantinya perancangan sistem ini dapat sesuai dengan kebutuhan pengguna.

3.2.4 Kebutuhan Fungsional Sistem

Menganalisa semua kebutuhan fungsi sistem yang harus dicapai,kebutuhan fungsional sebagai berikut:

a. Sistem dapat membedakan dua jenis marka garis membujur lurus
b. Sistem dapat menginformasikan tiga kondisi jarak kendaraan pengguna dengan marka jalan.
c. Sistem dapat memberikan notifikasi kepada pengguna jika melanggar marka garis membujur lurus utuh.

3.2.5 Kebutuhan Non Fungsional Sistem

Menganalisa batasan yang di perlukan oleh sistem, agar sistem dapat bekerjadengan baik yaitu batas aman tengangan yang diperlukan oleh mikrokomputer untuk dioprasikan agar tidak merusak mikrokomputer serta komponen outputnya.

3.2.6 Kebutuhan Perangkat Keras

Menganalisa semua kebutuhan perangkat keras yang dibutuhkan untuk membangun sistem. Perangkat keras yang dibutuhkan sebagai berikut:

a. Mikrokomputer
b. Power Supply
c. Papan *Output*
d. Kabel data tipe – B
e. Kabel RJ45 Croz
f. Laptop
g. Action Camera

3.2.7 Kebutuhan Perangkat Lunak

Menganalisa semua kebutuhan perangkat lunak yang dibutuhkan untuk membangun sistem. Perangkat lunak yang dibutuhkan sebagai berikut:

a. Open CV Library
b. Sistem Operasi Raspbian
c. Python 2.7
d. VNC View
e. IP Scanner

3.3Perancangan Sistem

Hasil analisa kebutuhan perangkat keras dan lunak akan dilakukan perancangan menjadi satu kesatuan yang dapat mendeteksi marka membujur garis.Pada proses ini terbagi dua perancangan, perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras berisikan perancangan catu daya, perancangan papan notifikasi, serta penempatan kamera. Pada perangkat lunak terdiri dari proses urutan metode proses sistem yang terdiri dari ekstraksi video, pengolahan citra, deteksi marka dan jarak,serta penentu status marka dan status jarak.

3.3.1 Perancangan Perangkat Keras

Tahapan perancangan perangkat keras untuk membangun sistem yang dapat deteksi marka membujurgaris utuh dan putus, terdiri dari perancangan catu daya yaitu hubungan antara *power supply* dengan *Raspberry Pi*, Selanjutnya adalah perancangan papan notifikasi, pada rancangan ini semua komponen pendukung perangkat notifikasi dirancang menjadi satu kesatuan,pada tahap terakhir adalah penempatan kamera yang digunakan untuk mengambil data video pada jalan raya yang sebenarnya.

3.3.2 Perancangan Perangkat Lunak

Tahapan perancangan perangkat lunak untuk sistem yang dapat deteksi marka membujur garis lurus utuh dan putus, terdiri diagram alir proses pengolahan data input yang didapat dari video, data video akan diproses melalui proses ekstraksi video,pengolahan citra, deteksi marka dan jarak, penentu status marka dan status jarak.

3.4Implementasi Sistem

Proses menerapkanrancangan sistem yang sudah dirancang pada perancangan perangkat keras dan perancangan perangkat lunak. Dimulai dari Implementasi perangkat keras,dan dilanjutkan pada implementasi perangkat lunak.

3.4.1 Implementasi Perangkat Keras

Hasil dari perancangan perangkat keras untuk mendeteksi marka membujur garis lurus utuh dan putus, Berupa gambar alat dari sisi dalam maupun luar casing dan foto penempatan kamera.

3.4.2 Implementasi Perangkat Lunak

Hasil Perancangan perangkat lunak untuk mendeteksi marka membujur garis utuh dan putus, ditampilkan dalam bentuk tabel kode progam beserta penjelasannya

3.5 Pengujian Sistem dan Analisis

Dilakukan untuk mengetahui hasil dari perancangan serta implementasi sistem. Terdiri dari pengujian fungsionalitas sistem, tingkat akurasi deteksi marka membujur garis utuh dan putus, tingkat akurasi deteksi jarak kendaraan dengan marka membujur garis, serta waktu rerata deteksi sistem. Pengujian terdiri dari tujuan pengujian, pelaksanaan pengujian, prosedur pengujian,hasil pengujian, dan analisis hasil pengujian.

3.5.1 Pengujian Fungsionalitas Sistem

Untuk mengetahui tingkat fungsionalitas sistem yang telah dibuat, pengujian meliputi kinerja dari lampu LED berwarna merah, kuning hijau yang masing masing terdiri dari empat biji, dan kinerja dari buzzer. Semua komposisi dari sistem akan di lakukan pengujian sesuai status marka dan status jarak hasil dari proses sistem.

3.5.2 Pengujian Akurasi Deteksi Marka Membujur Garis Utuh danPutus

Untuk mengetahui tingkat akurasi sistem dalam mendeteksi marka yang ditangkap oleh sistem, meliputi marka marka membujur garis utuh dan putus. Hasil akurasi pada setiap tipe marka membujur garis akan dicari rata rata totalnya

3.5.3 Pengujian Akurasi deteksi Jarak Antara Kendaraan dengan Marka Membujur Garis

Untuk mengetahuiakurasi dari pembacaan sistem dalam mengukur jarak kendaraan dengan garis marka yang dideteksi oleh sistem. terdiri empat status jarak yaitu status aman,status hati-hati,status awas, status bahaya sesuai dengan penelitian lapangan. Hanya pada pengujian ini komponen pengujian ditambahkan dengan prosedur pengambilan video. Hasil rata-rata akurasi pada setiap deteksi status jarak akan dicari rata rata totalnya.

3.5.4 Pengujian Waktu Deteksi Sistem

Untuk mengetahui lama yang dibutuhkan sistem melakukan proses sekali *loop*, Tujuannya untuk menegatahui waktu rerata yang dibutuhkan oleh sistem mendeteksi marka membujur garis beserta status jaraknya. Waktu rata proses dari tiap kondisi status jarak beserta status marka akan di cari rata-rata totalnya.

3.6 Kesimpulan dan Saran

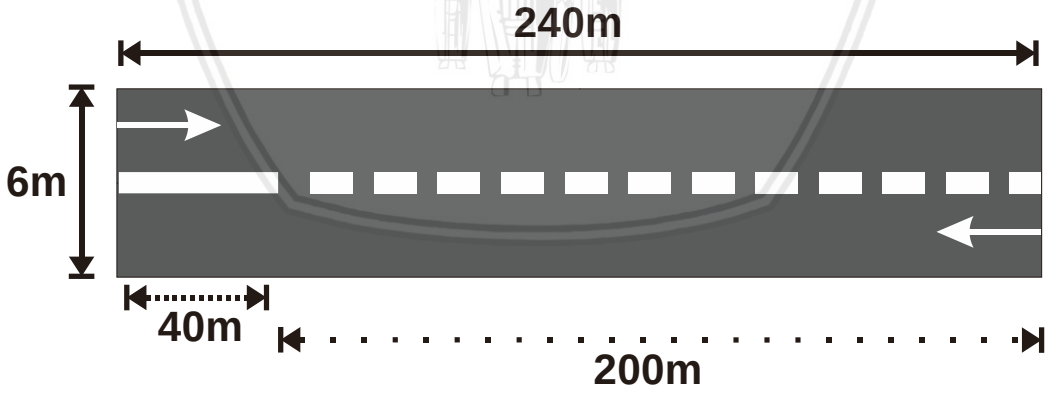
Hasil pegujian dan analisis yang telah dilakukan berguna untuk menjawab permasalahan yang mendasari penulis melakukan penelitian ini, serta saran diberikan untuk pengembangan kedepan yang lebih baik.

BAB 4REKAYASA KEBUTUHAN

Pada bab rekayasa kebutuhan ini menjelaskan tentang penelitian lapangan,gambaran umum sistem, kebutuhan pengguna, kebutuhan fungsional sistem,kebutuhan non fungsional sistem,kebutuhan perangkat keras,dan kebutuhan perangkat lunak yang dibutuhkan dalam menunjang penelitian.

4.1 Penelitian Lapangan

Pada penelitian lapangan yang dilakukan di jalan panggungkota malang ini memiliki kondisi garis lurus utuh dan garis lurus putus-putussepanjang 240 M dengan rincian panjang garis lurus utuh sepanjang 40M dan garis lurus putus-putus sepanjang 200M, serta lebar total jalan sepanjang 6 M dengan tipe jalan dua arah.Kondisi jalan relatif sepi sehingga sangat cocok untuk pengambilan sample video untuk di proses selanjutnya.Untuk lebih detail dapat dilihat pada Gambar 4.1.



Gambar 4.1Detail Jalan Raya

Keterangan:

: Panjang marka garis utuh

: Panjang marka putus – putus

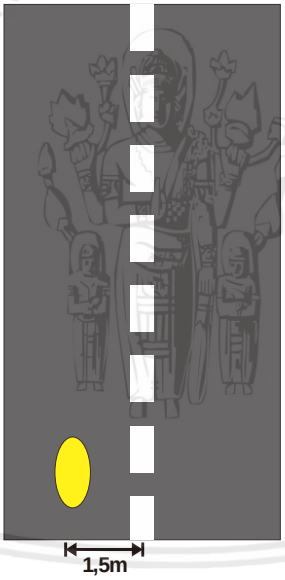
: Arah jalur kendaraan

Setelah melakukan diskusi dengan pengendara motor yang tergabung dalam Club Yamaha N-Max Club Indonesia chapter Malang Raya, akhirnya didapatkan posisi pengambilan sampel video pada empat posisi berbeda, posisi pertama menjauhi marka di sisi dalam dengan jarak antara marka tengah dengan posisi kendaraan sejauh 1,5m masuk kategori kondisi 'Aman', posisi kedua mendekati marka disisi dalam dengan jarak antara marka jalan dengan posisi kendaraan sejauh 0,75m masuk kedalam kondisi 'Hati-hati',posisi ketiga tepat diatas marka jalan masuk kedalam kondisi 'Awas',posisi ke-empat menjauhi marka disisi luar dengan jarak antara 0m sampai dengan 1,5m masuk dalam kondisi 'Bahaya'. Untuk lebih detail dapat dilihat pada Gambar 4.2,Gambar 4.3,Gambar 4.4,dan Gambar 4.5 berikut:

Keterangan :

= Posisi Kendaraan

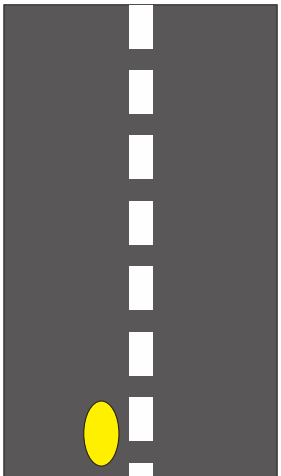
Pada Gambar4.2 adalah posisi pengambilan video pada kondisi aman.



Gambar 4.2Posisi Kondisi Aman

Pada gambar 4.2 posisi kendaraan pembawa kamera perekam di ilustrasikan dengan bidang ellips berwarna kuning berada pada sisi dalam marka dengan jarak antara kendaraan dengan marka membujur garis sebesar 1,5 meter dan posisi tersebut masuk dalam kategori kondisi Aman.

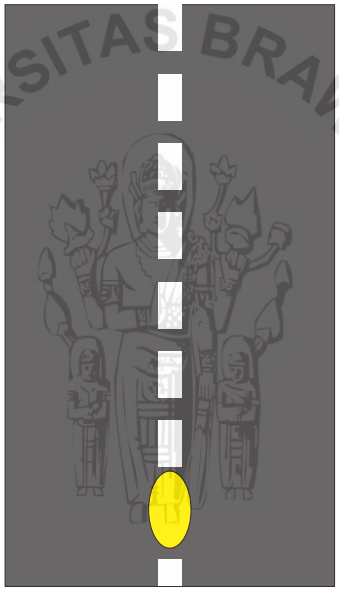
Pada Gambar 4.3 adalah posisi pengambilan video pada kondisi hati-hati.



Gambar 4.3 Kondisi Hati-hati

Pada gambar 4.3 posisi kendaraan pembawa kamera perekam di ilustrasikan dengan bidang ellips berwarna kuning berada pada sisi dalam marka dengan jarak antara kendaraan dengan marka membujur garis sebesar 0,75 meter dan pada posisi tersebut masuk dalam kategori kondisi Hati-hati.

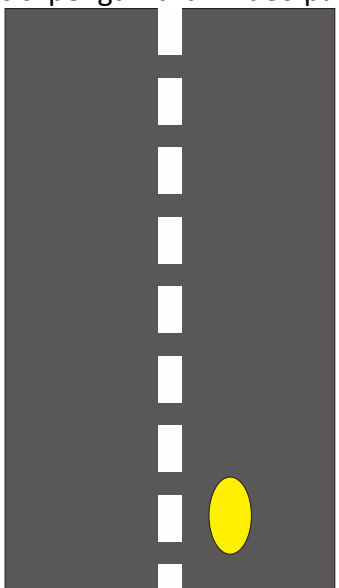
Pada Gambar 4.4 adalah posisi pengambilan video pada kondisi awas.



Gambar 4.4Posisi Kondisi Awas

Pada gambar 4.4 posisi kendaraan pembawa kamera perekam di ilustrasikan dengan bidang ellips berwarna kuning berada tepat diatas marka membujur garis ,pada posisi tersebut masuk dalam kategori kondisi Awas.

Pada Gambar 4.5 adalah posisi pengambilan video pada kondisi bahaya

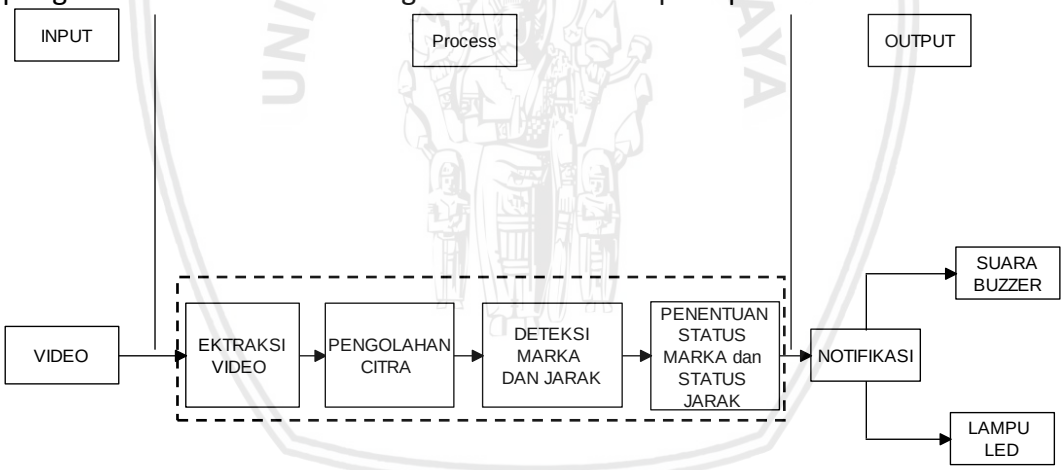


Gambar 4.5Posisi Kondisi Bahaya

Pada gambar 4.5 posisi kendaraan pembawa kamera perekam di ilustrasikan dengan bidang ellips berwarna kuning berada pada sisi luar marka dengan jarak antara kendaraan dengan marka membujur garis sebesar 0 - 1,5 meter dan posisi tersebut masuk dalam kategori kondisi Bahaya.

4.2 Gambaran Umum Sistem

Penelitian ini bertujuan untuk membangun sebuah sistem yang mendukung teknologi pengenalan marka jalan yang nantinya dapat digunakan pada kendaraan roda empat. Sistem yang terbentuk dalam sistem deteksi dan pengenalan marka jalan memiliki keluaran berupa suara dan signal lampu bagi pengendara.Berikut adalah diagram blok sistem seperti pada Gambar 4.6.



Gambar 4.6 Blok Diagram Sistem

Blok diagram diatas memaparkan bentuk umum dari sistem, ada tiga bagian utama dari sistem, yaitu blok input, proses,dan output. Video yang didapat pada penelitian lapangan adalah bagian dari blok input.Berikutnya adalah pada bagian blok proses yang terdiri dari empat bagian yaitu ekstraksi video, pengolahan citra, deteksi marka dan jarak,Penentuan output.Pada bagian ekstraksi video merupakan proses menangkap frame pada video.Pada bagian pengolahan citra dilakukan proses pengolahan citra sehingga dapat membantu proses selanjutnya, pada bagian deteksi marka dan jarak, merupakan proses mendeteksi tipe marka jalan serta jarak kendaraan dengan marka dari marka jalan melalui proses pengenalan citra.Serta pada bagian penentuan status marka dan status jarak merupakan tahap pengumpulan data dari proses sebelumnya sehingga hasil dari proses ini adalah status marka jalan,status jarak,dan perintah yang dikirimkan pada bagian output.Dan pada bagian terakhir adalah bagian notifikasi, pada

bagian ini menerima perintah dari proses sebelumnya untuk mengaktifkan dan mematikan suara yang berasal dari buzzerdan disertai signal lampu dari LED.

4.2.1 Tujuan

Sistem ini memiliki tujuan agar dapat mendeteksi marka jalan garis lurus, dapat mendetaksi kendaraan pengguna saat melewati marka jalan, menghitung jarak kendaraan pengguna terhadap marka jalan dengan memanfaatkan teknologi computer vision.Jenis marka jalan garis lurus dibagi menjadi dua yaitu marka jalan garis lurus utuh dan marka jalan garis lurus putus-putus.dengan adanya sistem ini maka pengguna kendaraa dapat memantau posisi kendaraan dengan marka jalan.

4.2.2 Karakteristik Pengguna

Karakteristik pengguna dalam sistem ini adalah sebagai user, yang dimana user akan menerima info dari sistem berupa nyala LED dan suara dari *Buzzer*.saat sistem mendeteksi adanya marka jalan dengan disertai keempat konsisi maka user akan mendapatkan informasi dari nyala lampu LED dan jika sistem mendeteksi adanya pelanggaran marka garis membujur lurus maka user mendapatkan informasi tambahan berupa suara dari buzzer.

4.2.3 Lingkungan Operasi

Lingkungan operasi sangat mendukung keberhasilan berjalannya penelitian,penelitian ini dilakukan padasiang sampai sore hari dengan kondisi cuaca cerahdan hanya dilakukan pada satu lokasi yang telah disebutkan pada penelitian lapangan. Lingkungan operasi juga harus pada kondisi minim kendaraan karena jika dipaksakan dilakukan saat kendaraan ramai maka hasil penelitian akan tidak maksimal.

4.3 Kebutuhan Pengguna

Kebutuhan pengguna pada sistem yang akan dibuat pada penelitian ini adalah sebagi berikut:

1. Sistem mampu memberikan informasi jika pengendara melewati garis marka jalan lurus utuh

   Pada sistem ini pengguna akan diberikan peringatan dalam bentuk buzzer jika posisi kendaraan telah melewati marka jalan garis lurus utuh,sehingga pengguna dapat segera memposisikan kendaraan pada posisi yang sesuai.

2. Sistem Mampu memberikan indikator jarak antara pengendara dengan marka jalan

   Pada sistem ini pengguna akan diberikan indikator dalam bentuk suara dan ketipan lampu sebagai indikator jarak antara kendaraan dengan marka jalan. Semakin dekat dengan marka maka bunyi buzzer dan lampu led semakin cepat dan semakin jauh semakin lama lambat.

4.4 Kebutuhan Fungsional Sistem

1. Sistem dapat membedakan dua jenis marka jalan

Fitur ini mengharuskan sistem untuk mendeteksi marka jalan dalam dua jenis yaitu garis marka garis membujue lurus utuh dengan marka garis membujur lurus putus-putus.

2. Sistem dapat mengukur jarak kendaraan pengguna dengan marka jalan

   Fitur ini mengharuskan sistem untuk dapat mengukur jarak antara kendaraan pengguna dengan marka jalan yang dideteksi.Jarak yang diukur dibagi menjadi 4kondisi yaitu kondisiAman, kondisiHati-hati, kondisiAwas, dan kondisiBahaya.

3. Sistem dapat mendeteksi jika posisi kendaraan melewati garis marka lurus utuh

   Fitur ini mengharuskan sistem untuk mendeteksi pengguna kendaraan jika berada posisi melanggar marka garis membujur lurus utuh.

4. Sistem dapat memberikan notifikasi kepada pengguna

   Fitur ini mengharuskan sistem untuk dapat memberikan informasi kepada pengguna mengenai jarak kondisi kendaraan dengan marka garis membujur garis serta peringatan jika pengguna telah melanggar marka membujur garis utuh dengan indikator lampu LED dan Buzzer.

4.5 Kebutuhan Non Fungsional Sistem

Untuk keselamatan oprasional maka, mikrokomputer harus mendapatkan tegangan optimal sebesar 5*volt* dengan rentang arus 1.2 sampai 2.5 *Amphere*. Tegangan yang melebihi maupun kurang dari rentang nilai yang dipatok tersebut dapat menyebabkan kerusakan maupun tidak berfungsinya komponen yang telah dirangkai.

4.6 Kebutuhan Perangkat Keras

1. Mikrokomputer

   Mikrokomputer yang digunakan adalah Raspberry Pi 3 Model B yang digunakan sebagai otak atau pusat pemroses dari sistem.

2. *Power Supply*

   Power Supply yang digunakan untuk memenuhi catu daya sistem menggunakan tipe Power Bank yang memiliki kapasitas sebesar 5200 mAh dengan rentang arus sebesar 0,8A sampai 1A, dengan rentang arus sebesar itu cukup memenuhi keselamatan oprasional dari mikrokomputer.

3. Papan Notifikasi

   Papan Notifikasi adalah sebuah papan board yang terdiri rangkaian elekronika yang digunakan untuk memberikan informasi kepada pengguna sistem. Pada papan ini terdiri dari komponen LED warna merah, kuning, hijau

yang dimana masing masing warna terdiri dari 4 biji, 3 resistor dengan daya 30 Ohm,50 Ohm,dan 150 Ohm serta lima pin *Male*.

4. Kabel data tipe –mikro B

Kabel data ini banyak digunakan sebagai konektor catu daya dari perangkat elektronik termasuk juga pada mikrokomputer Raspberry Pi3 yang menggukan port daya mikro B. Panjang kabel yang dibutuhkan sepanjang 10 cm.

5. Kabel LAN RJ45 tipe Croz

Kabel UTP dengan konektor keduanya RJ45 digunakan untuk menyambungkan mikrokomputer dengan laptop sebagai media *User Interface*, penggunaan kabel ini untuk menggantikan media User Interface yang menggunakan media WIFI, hal ini dilakukan untuk menekan pemakaian daya dari power supply sehingga masa pakai alat bisa lebih lama serta menghindari gangguan sinyal yang dapat terjadi kapan saja.

6. Laptop

Laptop digunakan sebagai media interface dari *Raspberry Pi 3*, sehingga semua data yang diterima dari *Raspberry Pi3* dapat dipantau dari laptop. Hal ini dikarenakan pada paket penjualan *Raspberry Pi3 basic* belum disertai perangakat user interface, melainkan hanya dalam bentuk *board* saja. Berikut adalah spesifikasi laptop yang digunakan pada penelitian ini:

a. Merek : ASUS
b. Tipe : K43TA
c. Prosesor : AMD APU A6-3400M 1,4 GHz ~ 2.3 GHz
d. Chipset : AMD A70M
f. Memori : 8GB DDR3 1333MHZ SDRAM
g. Storage : 650GB 2.5” SATA
h. Grafis : AMD RADEON HD 6720G2 1GB DDR3 VRAM
i. Interface : 1x USB 3.0
1x VGA port
1x RJ45 LAN

7. Action Camera

Action camera digunakan untuk media pengambilan objek data jalanraya yang terdapat marka garis membujur lurus. *Actioncam* dipilih sebagai media, dikarenakan kamera ini memiliki bentuk kecil dan dapat ditempelkan dengan kuat pada bagian sepeda motor yang dirasa perlu untuk menunjang keberhasilan penelitian, serta memiliki lensa yang lebar sehingga dapat mencakup semua bidang permukaan jalan yang diperlukan. Berikut spesifikasi *action camera* yang digunakan:

a. Merek : YI Tecnology
b. Tipe : YI –Lite
c.Chipset : Hisilicon Hi3556
d. Lensa : 16 MP Wide Angle 150” aperture f/2.8
e.*Image Sensor* : SONY IMX206
f. Storage : 32GB

4.7 Kebutuhan Perangkat Lunak

1. Open CV Library

OpenCV library terdiri dari kumpulan dari beberapa fungsi yang dapat digunakan dalam pemgolahan citra yang dimana dalam pengoprasian kita tidak perlu mendefinisikan semua proses matematika serta proses kerjanya secara manual, yang perlu kita kerjakan hanyalah mengambil fungsi yang sudahada pada library serta mamasukan variabel yang dibutuhkan agar fungsi yang kita butuhkan dapat di proses, hal ini lah yang mendasari saya membutuhkanperangkat lunak Open CV Library dalam mengolah data yang saya peroleh dari penelitian lapangan.

2. Sistem Operasi Raspbian

Sistem operasi berfungsi sebagai media *user interface* antara mikrokomputer dengan saya, sekaligus sebagai manajemen proses dalam pengolahan citra yang dibutuhkan. Raspbian adalah OS terbaik yang didukung langsung secara resmi sehingga OS ini ringan. OS ini banyak digunakan oleh pengguna dalam menjalankan atau memproses suatu citra maupun video dan mampu menjalankan suatu proses pengolhan citra dengan sangat baik.

3. Python 2.7

Python versi 2.7 berfungsi sebagai kode pemrogaman yang dibutuhkan untuk membuat sebuah progam sebagai jembatan antara OS dari mikrokomputer dengan proses pengolaha citra menggunAkan perangkat lunak OpenCV. Versi 2.7 masih sering digunakan pada forum forum pengolahan citra yang menggunakan perangkat lunak OpenCV, alasan menggunkan versi ini adalah banyaknya referensi pengolhan citra yang menggunkan versi ini sedangkan pada Python veri 3.0 memiliki banyak perbedaan dari versi sebelumya terutama dari segi pemrogamnya.

4. VNC View

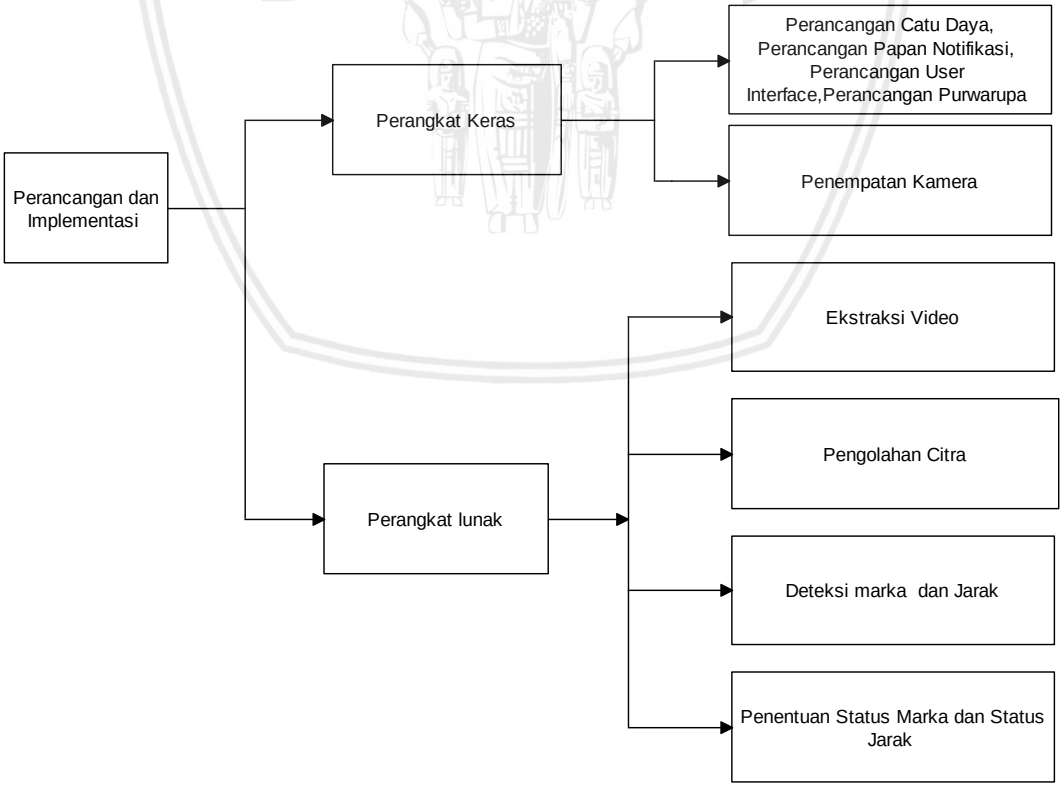
VNC View yaitu aplikasi remote yang berfungsi untuk mengontrol komputer lain melalui komputer utama, hal ini butuhkan oleh mikrokontroler sejenis *raspberry pi*, karena pada board *basiknya*, *raspberry pi3* tidak disertakan dengan layar monitor sebagai*user interface*, sehingga dengan aplikasi ini ,user dapat mengontrol *raspberry pi3* dengan komputer yang memiliki layar seperti PC maupun Laptop.

5. *IP Scanner*

Ip scaner merupakan aplikasi yang berfungsi mencari ip address yang terhubung dengan komputer, hal ini berfungsi untuk mengenali ip address dari raspberry pi3 yang tersambung dengan komputermenggunakan jaringan LAN. Ip dari raspberry pi3 dibutuhkan agar komputerdapat tersambung dengan perangkat sebagai remote.

BAB 5PERANCANGAN DAN IMPLEMENTASI SISTEM

Bab ini menjelaskan tahap perancangan dan implementasi sistem deteksi marka jalan menggunkan Raspberry Pi. Tahap tahap perancangan dan implementasi dibagi menjadi dua bagian yaitu perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras terdiri dari dua bagian yaitu pada bagian pertama terdiri Raspberry Pi, papan notifikasi, dan purwarupa. Pada bagian kedua yaitu penempatan Kamera. Sedangkan perancangan perangkat lunak terdiri dari empat proses yaitu ekstraksi video, pengolahan citra, deteksi marka dan jarak,dan penentuan status marka dan status jarak pada skema Gambar 5.1.



Gambar 5.1Skema Perancangan dan Implementasi

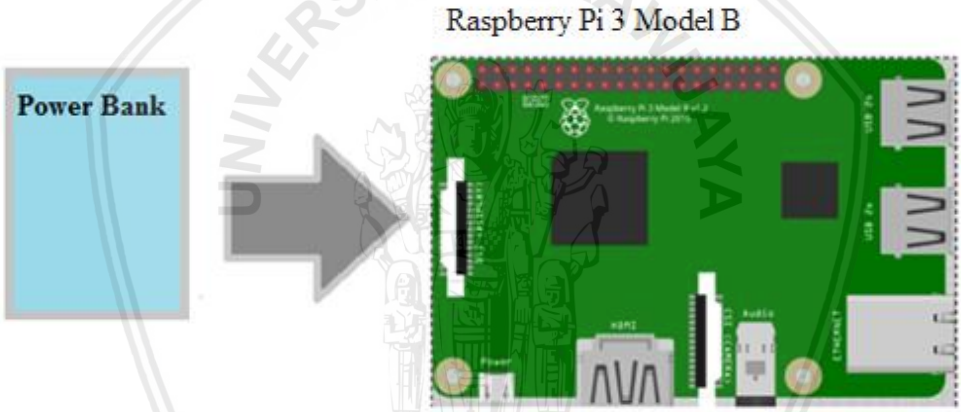
5.1 Perangkat Keras

5.1.1 Perancangan Perangkat Keras

Pada bagian ini akan menjelaskan bagaimana bentuk rancangan dari perangkat keras dari sistem deteksi marka jalan,yang dimana komponen yang ada didalamnya terdiri tiga bagian, yang pertama hubungan antara*power suply* dengan mikrokontroler, hubungan antara mikrokontroler dengan papan notifikasi,dan hubungan antara sistem dengan perangkat *user interface* sebagai medianya adalah laptop.

5.1.1.1 Perancangan Catu Daya

Perancangan Catu Daya meliputi *Power Bank* dengan *Raspberry*. Kedua perangkan tersebut akan dikombinasikan senjadi satu kesatuan dalam sistem. Hal ini dilakukan agar sistem dapat berjalan dengan baik serta perangkat *output* dapat bekerja dengan baik. Rancangan catu daya dari sistem dapat dilihat pada Gambar 5.2



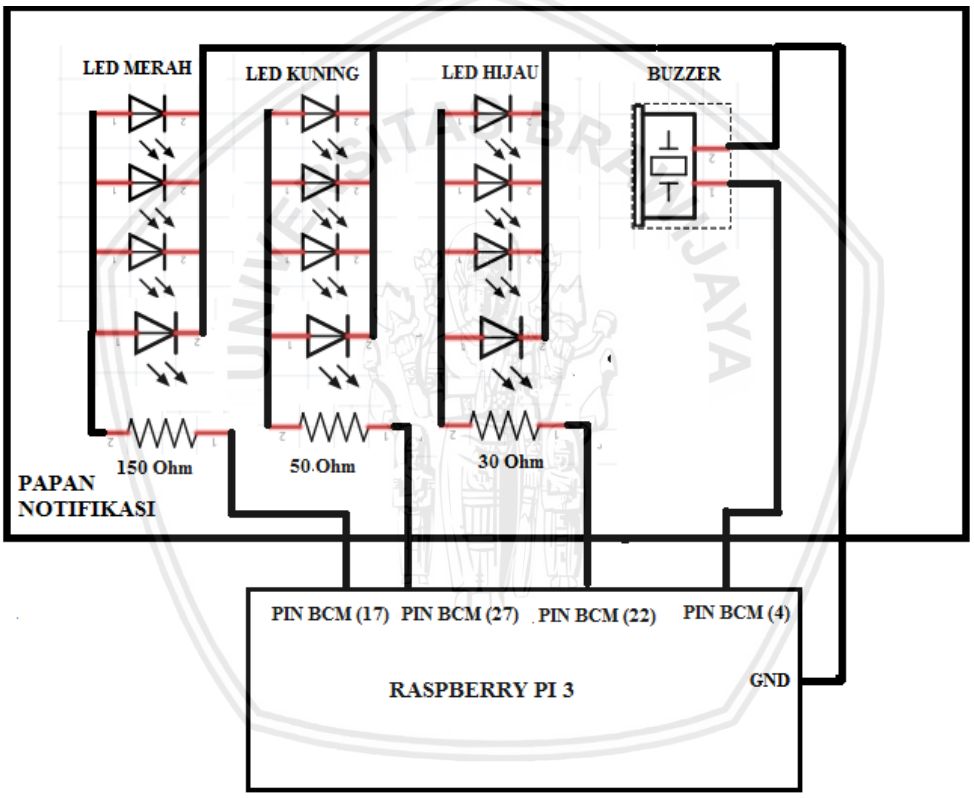
Gambar 5.2 Rancangan Catu Daya

Pada perancangan catu daya sumber power dari sistem berasal dari power supply dengan jenis *power bank* yang memiliki kapasitas sebesar 5200 mAh dengan rentang arus 0,8 A sampai 1 A dengan tengangan sebesar 5 Volt, yang dimana pada rentang arus tersebut adalah rentang arus aman untuk mengoprasikan mikrokomputer serta papan notifikasi yang berfungsi sebagai output sistem.

5.1.1.2 Perancangan Papan Notifikasi

Perancangan papan notifikasi meliputi hubungan antara *Raspberry*dengan Papan notifikasi,Papan notifikasi berperan sebagai*output* dari sistem. Perancangan ini dibutuhkan agar hubungan antara Raspberry dengan Papan notifikasi dapat terhubung dan terkoneksi dengan baik, karena jika dilihat pada Raspberry terdapat banyak pin yang dimana terdapat berbagai macam fungsi yang berbeda pada pin tertentu.

Rancangan papan notifikasi dapat dilihat pada Gambar 5.3.



Gambar 5.3Rancangan Papan Notifikasi

Dalam perancanganya papan notifikasi memiki komponen empat biji lampu LED warna merah, kuning, hijau yang dimana setiap warna dirangkai pada rangkaian seri dan dihubungkan pada yang resistor tiap warnanya memiliki hambatan sebesar 150 Ohm,50 Ohm, dan 30 Ohm, serta terdapat komponen penghasil suara yaitu *buzzer* yang terangkai didalam sebagai notifikasi suara dari output sistem. Perhitungan nilai resistor didapat dari rumus sebagai berikut sebagai berikut:

$$R = (Vs - Vl)/I \quad (5.1)$$

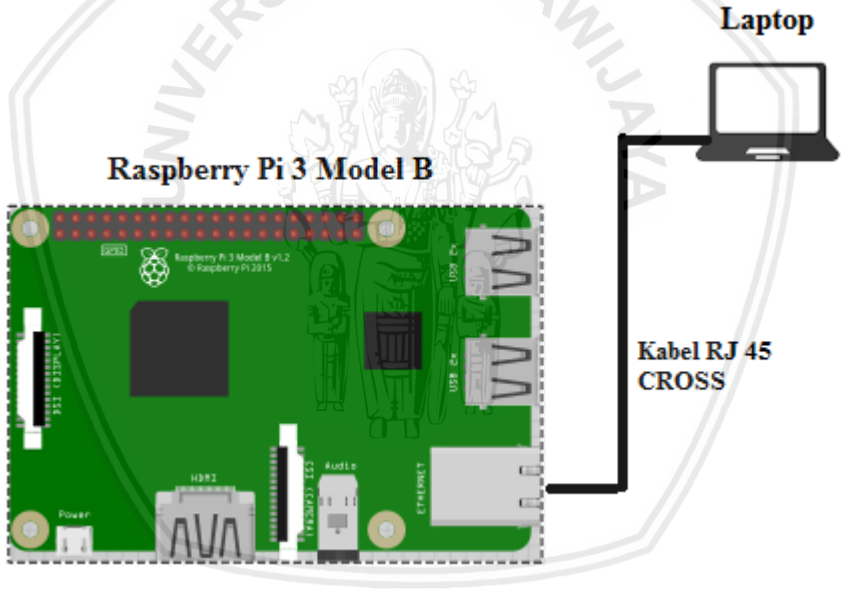
Keterangan

R = nilai resistor yang diperlukan
Vs = Tegangan input (Volt)
Vl = Tegamgan LED (Volt)
I = Arus maju LED (Amphere)

Jika nilai resistor yang keluar dari hasil perhitungan tidak ada di pasaran maka gunakankan nilai hanbatan yang diatasnya yang tersedia dipasaran karena jika arus yang masuk ke LED melebihi dari arus maksumum led maka lampu akan rusak dan tidak bisa digunakan lagi.

5.1.1.3Perancangan User Interface pada laptop

Perancangan user interface antara mikrokontroller dengan laptop berfungsi sebagai media koding dan melihat log progam dari sistem yang dibuat sehingga data proses dapat di analisis langsung pada laptop sebagai salah satu faktor pendukung keberhasilan penelitian. Rancangan user interface dapat dilihat paa Gambar 5.4.

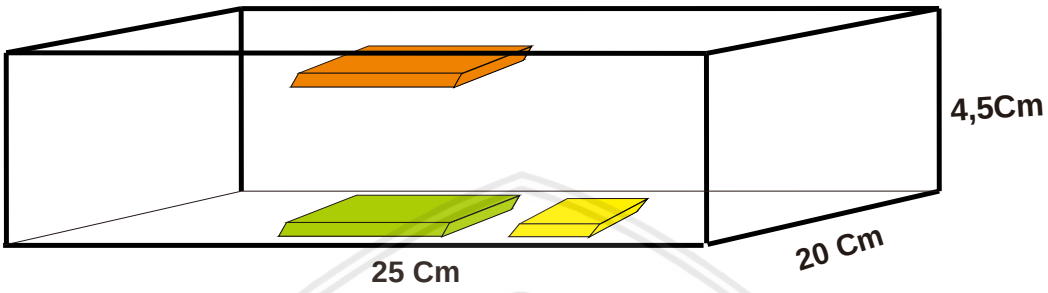


Gambar 5.4 Rancangan Sistem User Interface

Dalam perancanganya dibutuhkan kabel UTP dengan ujungnya berbentuk RJ 45 dengan tipe posisi kabel yaitu cross, tipe cross digunakan khusus untuk melakukan pengiriman dua arah dimana data dari mikrokomputer akan dikirimkan ke laptop untuk dibaca dan juga sebaliknya dari laptop mengirimkan sebuah data perintah yang akan diterima oleh mikrokomputer, sebenaenya kabel tipe *straight* dapat digunakan untuk pengiriman dua arah tetapi tingkat terjadi crash sering berbeda dengan tipe cross yang memang khusus digunkan sebagi pengiriman dua arah.

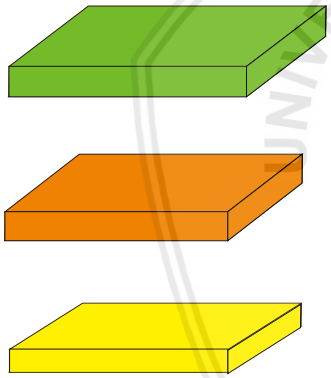
5.1.1.4 Perancangan Purwarupa

Pada bagian akan menjalaskan bentuk dari purwarupa dari sistem deteksi marka jalan, Dalam perancangan purwarupa terdapat dua buah rancangan yauti rancangan sisi dalam perwarupa dan rancangan sisi luar purwarupa yang akan di jelaskan pada Gambar 5.5 dan Gambar 5.6. Material yang digunakan untuk membentuk perwarupa sistem berupa kertas karton coklat dengan ketebalan keras 0,3mm,sedangkan perekat yang digunakan adalah tipe perekat dua sisi agar kertas dapat dibangun dalam bentuk tiga dimensi.



Gambar 5.5 Purwarupa Sisi Dalam

Keterangan:

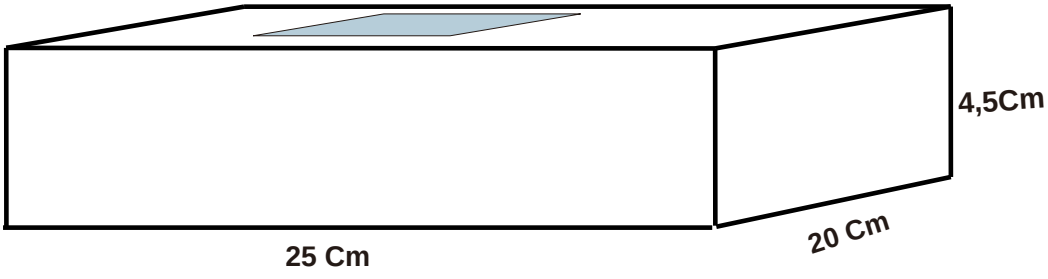


= *Raspberry pi*

= Papan Notifikasi

= Power Bank

Pada Gambar 5.5 dijelaskan penempatan Raspbery pi diumpamakan dengan balok dengan warna hijau dan penempatan papan notifikasi diupamakan balok berwarna orange serta power bank ddiumpamakan balok berwarna kuning. Penempatan balok warna orange ditempatkan mendekati sisi atas kubus karena pada sisi atas kubus akan dilubangi sesuai besar papan notifikasi agar output yang ditampilkan oleh balok warna orange dapat dilihat oleh pengguna.



Gambar 5.6 Purwarupa Sisi Luar

Keterangan:

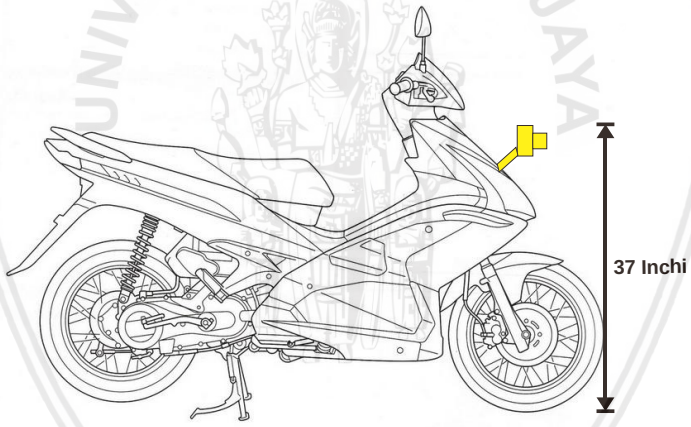
= Lubang dengan lapisan mika

Besar mika (panjang 3,5 Cm lebar 4,5 Cm)

Pada Gambar 5.6 menjelaskan tentang perancangan purwarupa bagian luar,terdiri dari lubang bagaian atas untuk melihat notifikasi sistem. Lubang pada sisi atas dilapisi dengan mika transparan sehingga tampilan lebih rapi dan secara kasat mata terlihat seperti kaca.

5.1.2 Penempatan Kamera

Penempatan Kamera dibutuhkan karena objek yang diambil adalah jalan raya, sehingga saat melakukan proses pengambilan video bisa berjalan dengan lancar dan aman. Dikarenakan motor yang digunakan adalah tipe automatis maka headlamp berada pada dada depan motor,bukan berada pada stir motor. Penempatan kamera di rekatkan pada posisi yang telah ditentukan dengan menggunakan perekat dua sisi yang kuat, hal ini digunakan agar saat pengambilan video getaran pada kamera akan direduksi. Letak penempatan kamera dapat dilihat pada Gambar5.7



Gambar 5.7Penempatan Kamera

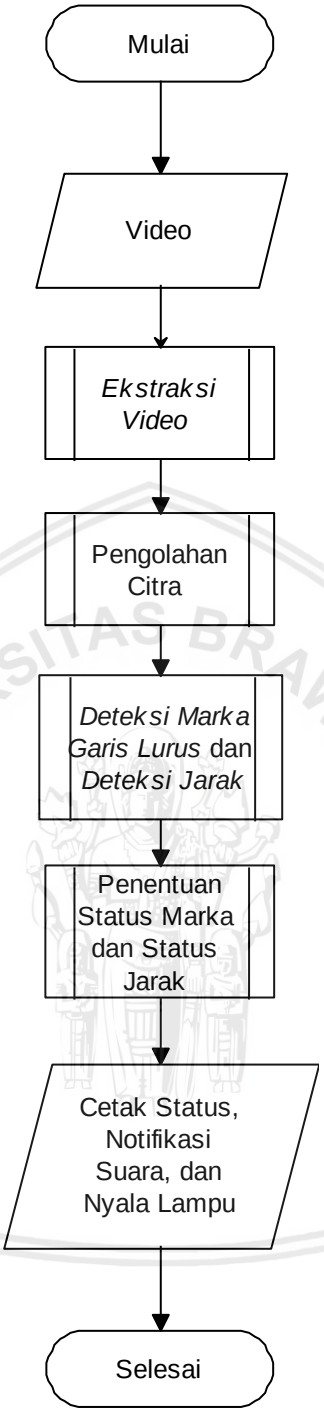
Penempatan kamera berada diatas lampu dan dibawah windshield pada motor yang sebenarya, hal ini dilakukan karenamenurut penelitian yang dilakukan semakin tinggi penempatan kamera maka semakin baik pula objek yang ditangkap, tetapi karena keterbatasan penempatan kamera maka pada posisi di Gambar 5.7 adalah posisi paling baik untuk menangkap objek. Jika penempatan kamera berada di windshield maka efek getaran dari mesin kendaraan akan banyak diserap oleh windshield,karena bahan pembuat windshield adalah plastik ABS yang dimana memiliki sifat lentur, sehingga getaran mesin akan banyak diserap oleh windshield, sehingga video yang ditangkap oleh kamera akan banyak memiliki noise getaran yang berasal dari getaran mesin.

5.2 Perangkat Lunak

Pada bab ini kan menjelaskan tentang bagaimana rancangan perangkat lunakdidalam sistem serta menjelaskan tentang kerja dari metode yang digunakan oleh sistem.

5.2.1 Perancangan Perangkat Lunak

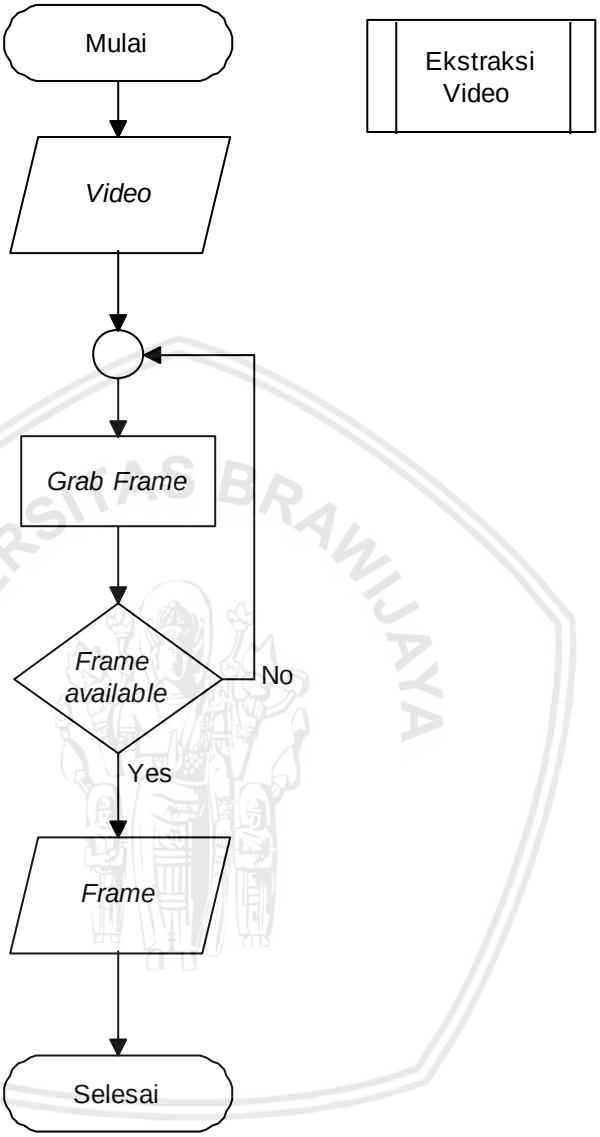
Input dari proses deteksi marka jalan adalah berupa video dengan format ’mp4’, pemutaran video input menggunakan *library OpenCV* berbasis *Python.* Proses yang akan dilewati oleh video adalah*Frame Acquitision* dan Resize,Pengolahan citra,Deteksi garis dan jarak dan Penentu status marka dan status jalan. Flowchart perancangan Perangkat lunak dapat dilihat pada Gambar 5.8.



Gambar 5.8 Flowchart Perancangan Perangkat Lunak

5.2.1.1 Ektraksi Video

Video adalah sekumpulan gambar sekeuensial dalam waktu tertentu dengan tujuan untuk membentuk gambar bergerak. Mengakses masing – masing frame ini disebut *frame acquisition*. Dalam frame acquitision kecepatan frame yang



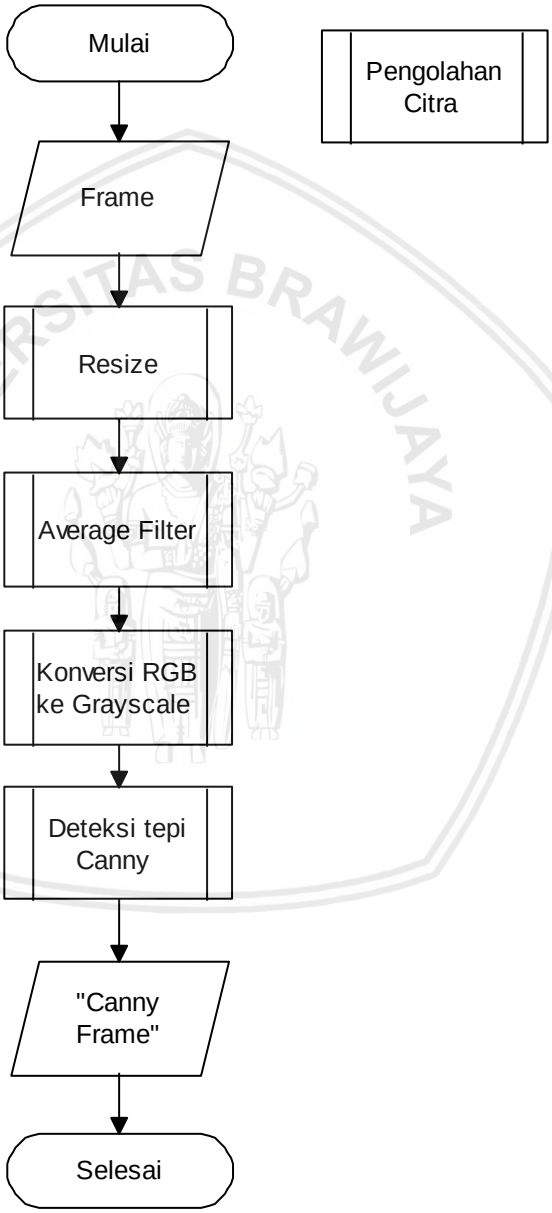
diambil untuk di proses dapat diatur sesuai kebutuhan tetapi *default* dari *OpenCV*sebesar 25 milidetik. Diagram frame acuisition dapt dilihat pada Gambar 5.9.

Gambar 5.9 Flowchart *Ektraksi Video*

Frame acquisiton menerima data dari video, saatvideo berjalan pada waktu *grab time*(waktu ambil) maka*grabframe(ambil gambar)* yang didapat pada wakru tersebut akan diambil,setelah *frame* didapatkan maka *frame* akan dilakukan cek,jika terdapat frame pada saat *grab time*ada maka frame akan diolah pada proses berikutnya. Saat *frame* tidak didapatkan saat proses *grab frame*, maka akan kembali pada proses *grap time* dan *grap frame*. Proses ini akan berlanjut sampai video berakhir.

5.2.1.2 Pengolahan Citra

Pengolahan citra adalah proses untuk mengolah frame yang didapat dari ektraksi video menjadi frame yang nantinya dapat dideteksi pada proses deteksi marka jalan lurus beserta dengan jarak. Proses pengolahan citra di mulai dari Resize,Gaussian blur,Konversi RGB ke Grayscale,Canny. Flowchart pengolahan citra dapat dilihat pada Gambar 5.10.



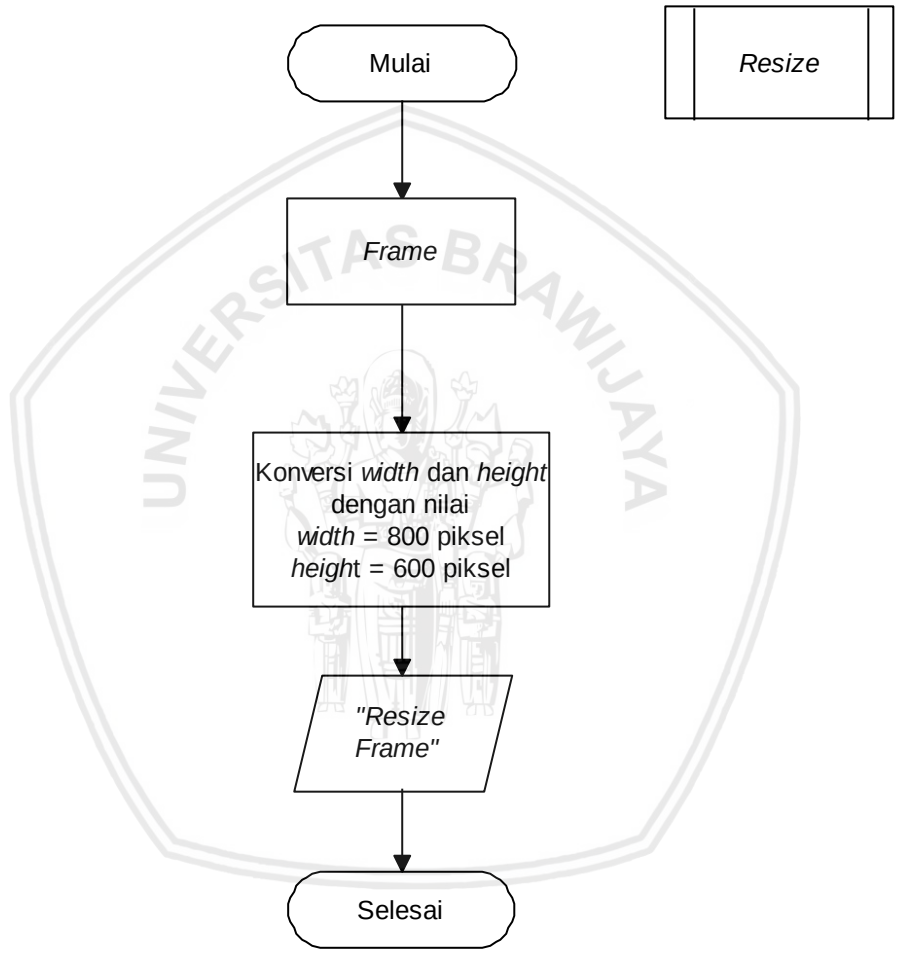
Gambar 5.10 FlowchartPengolahan Citra

A. Resize

Resize berfungsi untuk mengatur ulang resolusi *frame* yang di dapat dari video, karena resolusi video relatif besar maka perlu melakukan penyesuaian

ulang, sehingga frame yang di proses oleh sistem dapat lebih efisien dan mempercepat waktu proses, karena semakin besar resolusi *frame*maka semakin lama juga waktu prosesnya.

Pada pemrogaman OpenCV terdapat fungsi dengan nama ***cv2.Resize***yang memiliki fungsi untuk melakukan seting ulang resolusi gambar sesuai resolusi yang diinginkan. Fungsi ini memiliki tiga variabel masukan yaitu variabel gambar yang akan dirubah, variabel nilai width dan height yang diiginkan serta variabel dari library inter=cv2.INTER_AREA.Flowchart *resize* dapat dilihat pada Gambar



5.11.

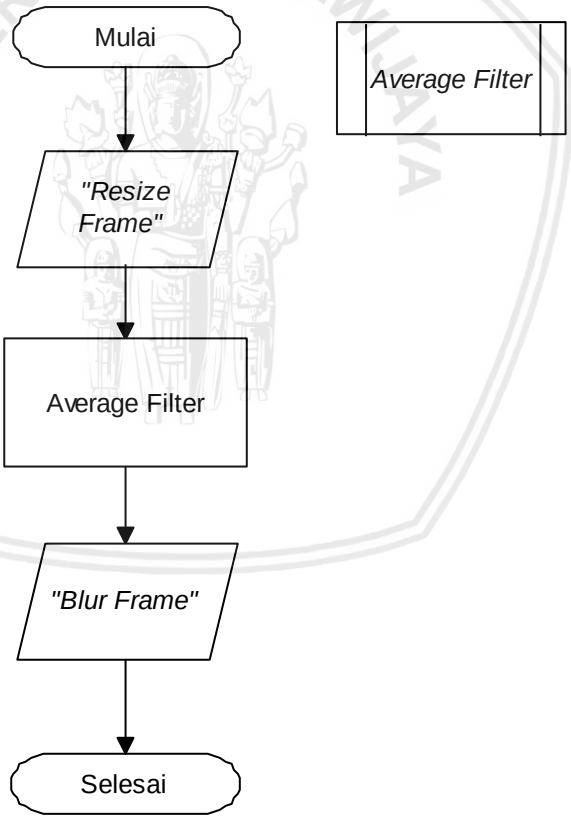
Gambar 5.11*Flowchart Resize*

Proses Resize pada gambar 5.11 dimulai dari hasil *frame* proses *frame acquition,* selanjutnyadilakukan perubahan resolusi sesuai kebutuhan peneliti, peneliti membutuhkan agar resolusi frame yang didapat dari *frame acquition menjadi frame yang memiliki* lebar 800 dan tinggi 600 piksel,sehingga pada fungsi **cv2.*resize*** pada OpenCV, nilai masukan fungsi *width diberikan nilai sebesar* 800 piksel dan nilai *height* diberikan nilai sebesar 600. Hasil dari proses resize menghasilkan framedengan resolusi sebesar 800x600 piksel.

B. Average Filter

Average filter merupakan salah satu metode yang digunakan untuk melakukan pengkaburan pada gambar dengan metode membagi semua nilai yang ditangkap oleh kernel sama rata dan hasil dari pembgaian akan menjadi nilai matrik yang berada di tengah. Hal ini dilakukan untuk menghilangkan *noise* pada frame yang didapat. Karena objek yang di teliti adalah jalan raya maka metode ini sangat dibutuhkan dalam sistem, karena jalan raya memiliki bentuk kontur yang tidak halus,serta bahan dasar pembuat jalan raya salah satunya adalah kerikil kecil yang berguna untuk menambah *traksi* pada ban kendaraan sehingga jalan raya memiliki permukaan yang tidak rata.

Library OpenCV telah menyediakan metode *average filter* dengan nama ***cv2.blur.*** Untuk menggunakan fungsi ini, diperlukan dua variabel data yang harus dimasukan pada fungsi tersebut, yang pertama adalah variabel data gambar yang akan dikaburkan yaitu variabel Gray Frame, selanjutnya diperlukan nilai kernel yang digunakan untuk melakukan pengkaburan pada gambar. Flowchart *Average filter* dapat dilihat pada Gambar 5.12.

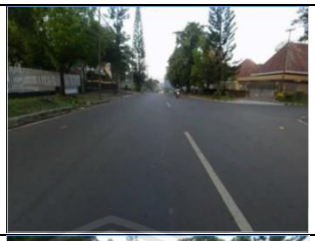
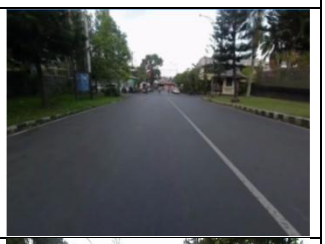
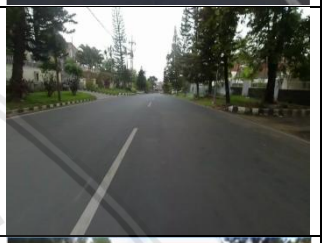
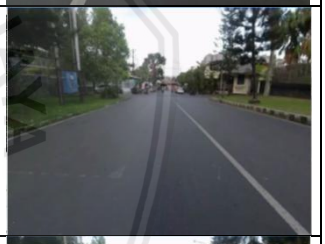
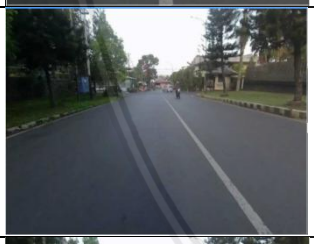
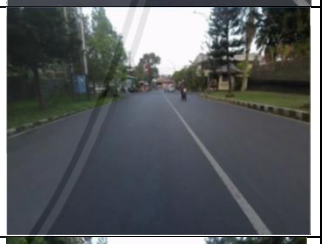
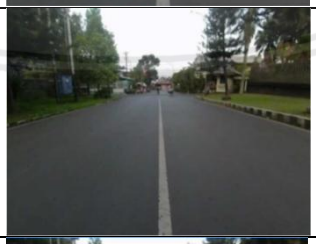
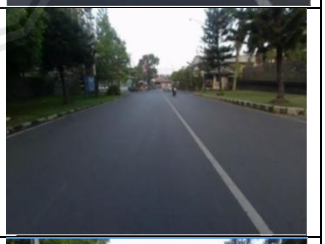
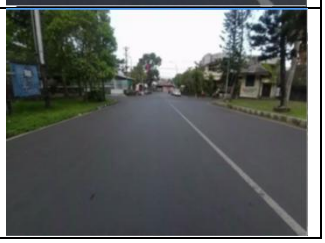


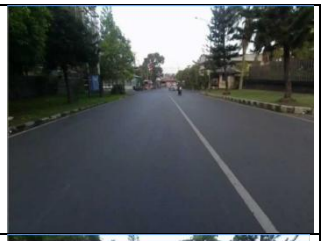
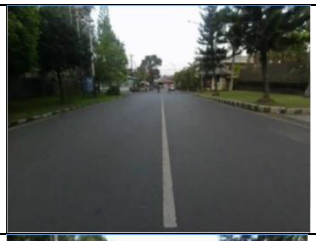
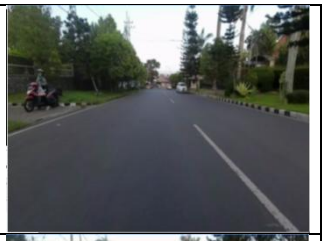
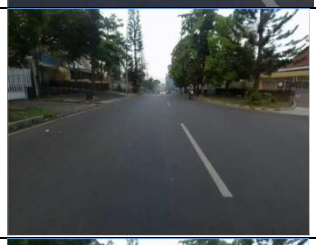
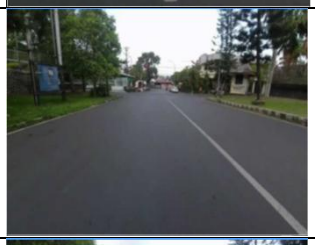
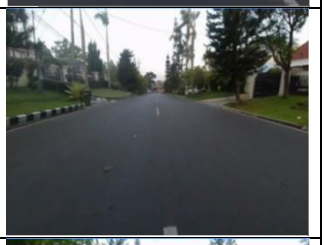
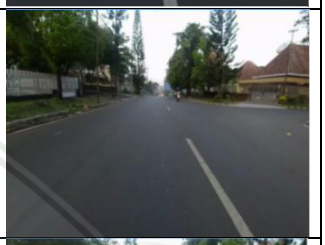
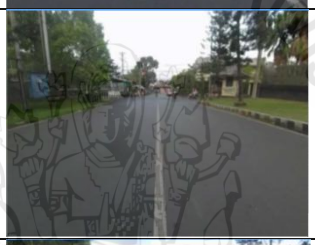
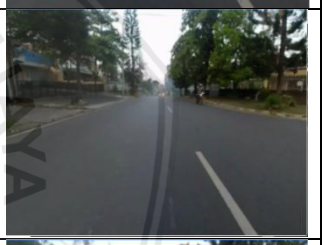
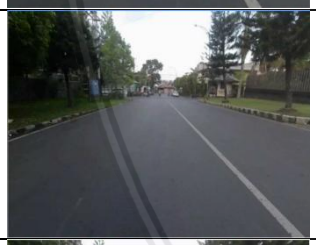
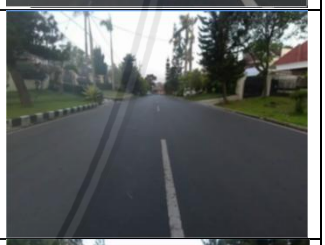
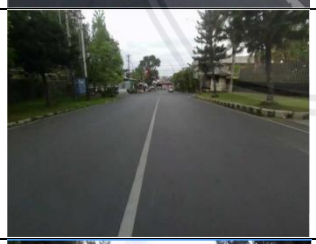
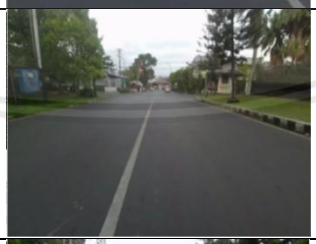
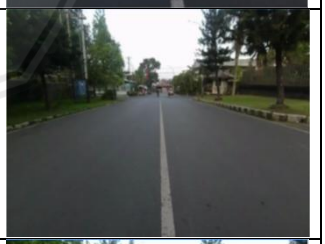
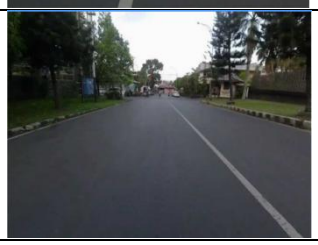
Gambar 5.12*Average Filter*

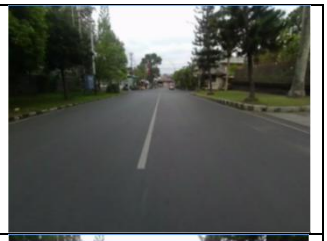
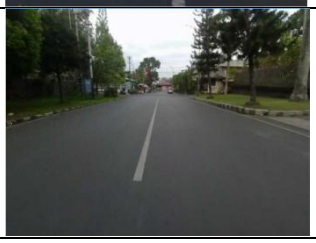
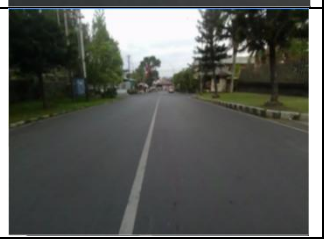
Proses *average filter* memiliki proses meratakan nilai piksel tetangga yang tertangkap oleh kernel, nilai rata-rata tetanggaya akan dicari dan nilai rata-rata akan diberikan pada pusat kernel. Ukuran Kernel yang digunakan adalah 3x3 . Hasil dari proses ini akan dimasukan pada variabel "*Blur Frame"*. Ukuran kernel didapat dari serangkaian perbandingan terlebih dahulu dengan

membandingakan dengan kernl 5x5 dan kernel 7x7 Hasilnya dapat dilihat pada Tabel 5.1.

Tabel 5.1 Perbandingan Ukuran Kernel

NO	3x3	5x5	7x7
1.			
2.			
3.			
4.			
5.			
6			

7			
8			
9			
10			
11			
12			
13			

14			
15			

Hasil dari proses Average Filter dapat dilihat pada Gambar 5.13.

Gambar 5.13 Hasil proses *Average Filter*

C. Konversi RGB ke Grayscale

Konversi ini adalah mengubah gambar RGB menjadi gambar keabuan. Metode yang digunakan dalam pemrogaman OpenCV adalah Luminosity. Cara kerja metode ini adalah mengalikan nilai dari masing-masing RGB pada gambar sengan nilai yang sudah ditentukan dalam rumus 5.1,sehingga hasil dari pengalian rumus didapatkan gambar yang memiliki warna keabuan.

$$Grayscale = (0.299 \times R) + (0.587 \times G) + (0.144 \times B) \quad (5.1)$$

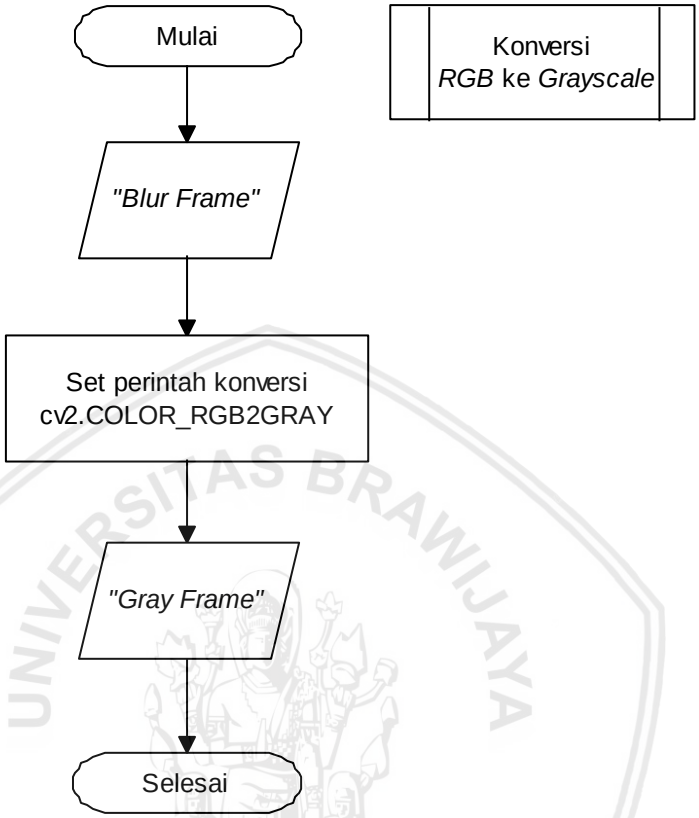
keterangan:

R : nilai piksel red

G : nilai piksel green

B : nilai piksel blue

Pada pemrogaman OpenCV terdapat library konversi gambar RGB ke *grayscale* dinamakan fungsi ***cv2.cvtcolor*.** Untuk melaukan proses ini membutuhkan dua masukan yaitu fata gambar yang akan dikonversi serta perintah konversi yang digunakan.Flowchart konversi RGB ke Grayscale dapat dilihat pada Gambar 5.14.



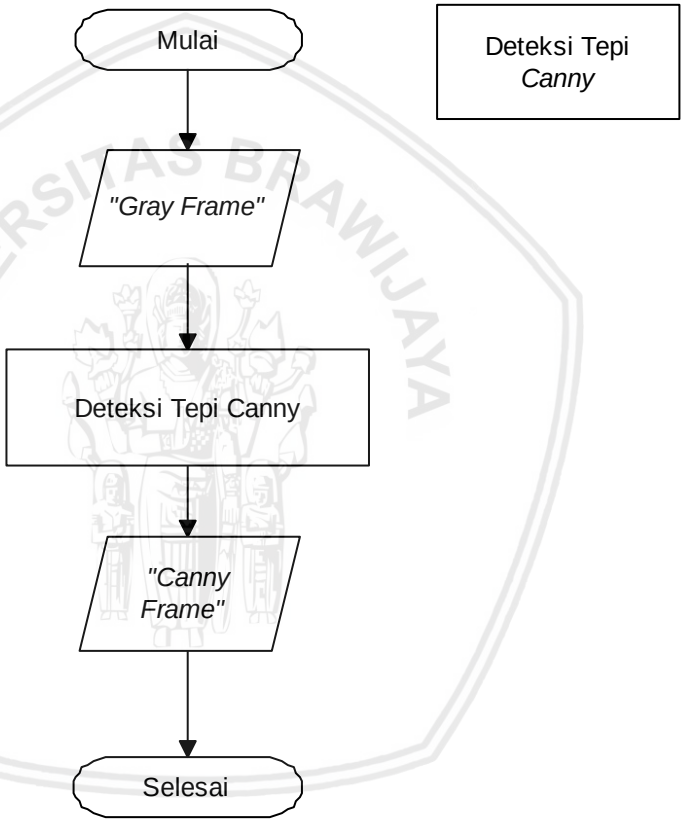
Gambar 5.14*Flowchart*Konversi *RGB ke Grayscale*

Didalam perintah ***cv2.COLOR_RGB2GRAY***sudah tertanam rumus 5.1, sehingga kita tidak perlu memasukan nilai pengali dari setiap piksel RGB secara manual.sehingga secara otomatis jika data gambar dimasukan dalam parameter fungsi cv2.cvtcolor dengan perintah ***cv2.COLOR_RGB2GRAY*** maka nilai tiap piksel gambar akan melakukan proses pengalian dan menghasilkan gambar keabuan. Hasil perubahan citra dapat dilihat pada Gambar 5.15.

Gambar 5.15 Hasil *Konversi RGB ke Grayscale*

D. Deteksi Tepi Canny

Deteksi tepi Canny merupakan metode yang banyak digunakan untuk mendapatkan tepian dari suatu objek dari gambar. Metode ini membutuhkan input data yang berwarna keabuan untuk prosesnya. Pada pemrogaman OpenCV terdapat library dari metode canny yaitu ***cv2.canny***. Untuk menggunakan fungsi ini membutuh tiga buah masukan, yang pertama yaitu gambar keabuan,kedua ambang bawah, ambang atas. Flowchart Deteksi Tepi *canny* dapat dilihat pada Gambar 5.16.



Gambar 5.16*Flowchart Deteksi Tepi Canny*

Pada proses Deteksi Canny di perlukan dua buah data yaitu dua nilai ambang atas (maksimum) dengan nilai 200dan nilai ambang bawah (minimum) dengan nilai 50. dua nilai tersebut nantinya digunakanuntuk mendeteksi apakah objek yang dideteksi adalah tepi atau bukan, dengan proses sebagai berikut:

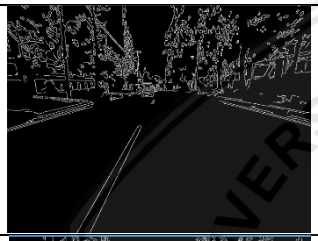
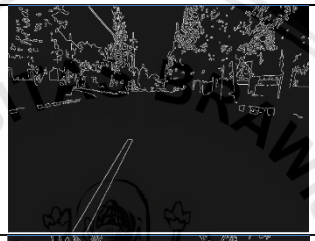
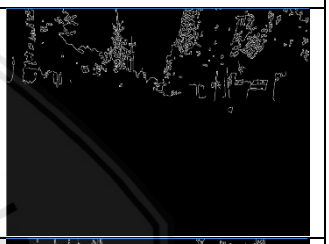
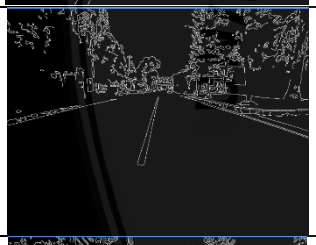
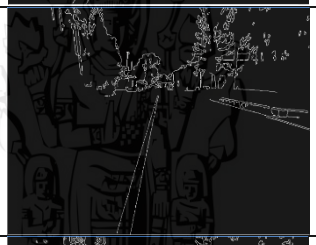
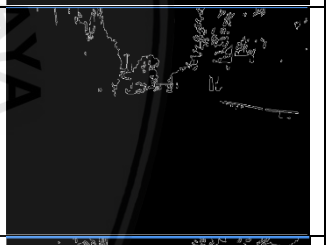
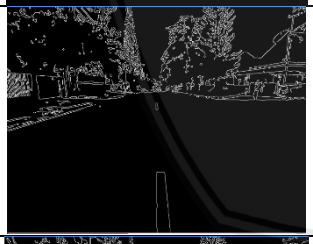
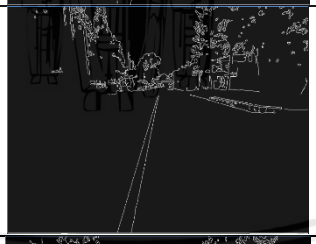
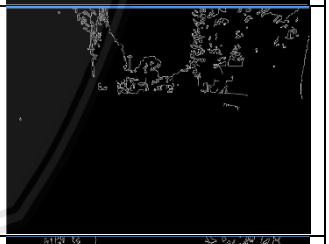
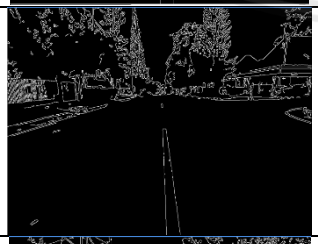
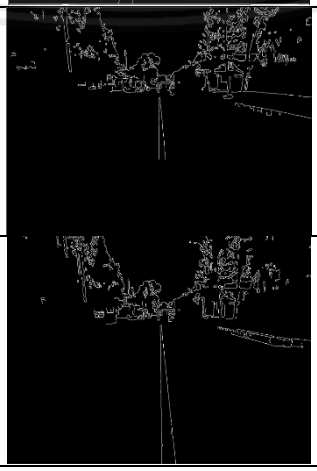
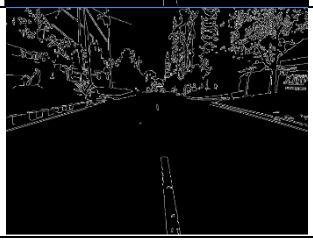
1. jika setiap sisi dengan intensitas gradien lebih dari max pasti itu adalah tepi.

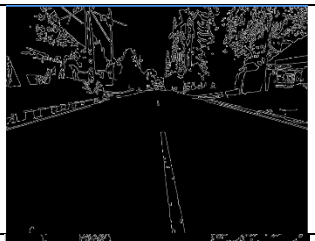
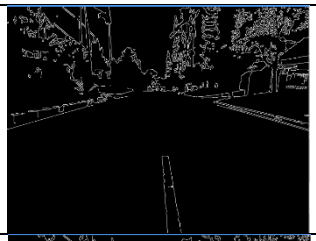
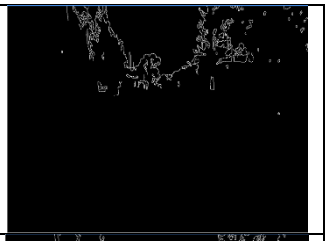
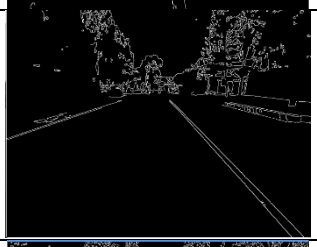
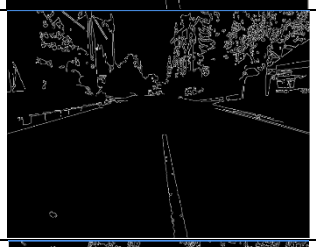
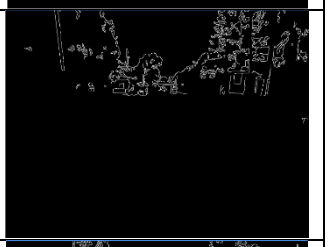
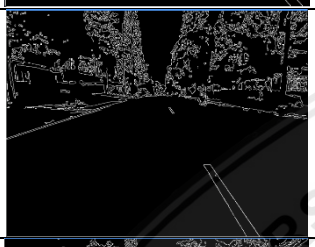
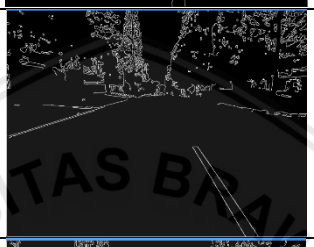
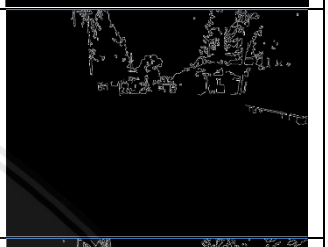
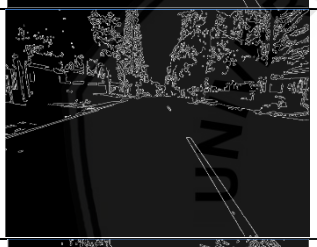
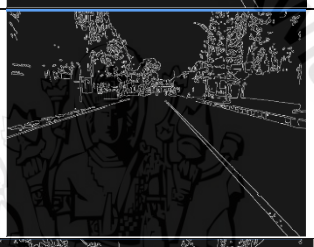
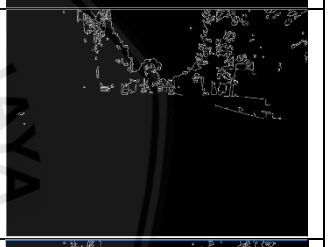
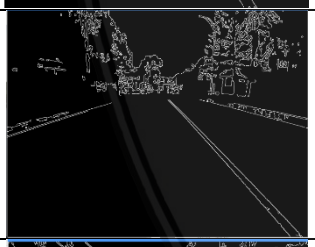
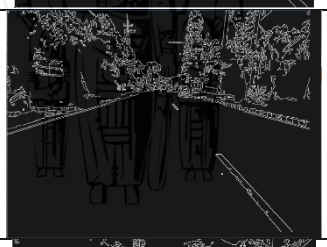
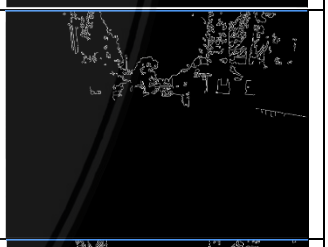
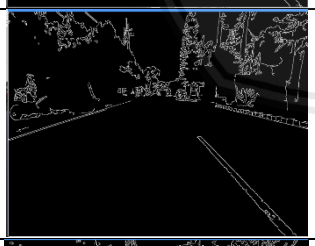
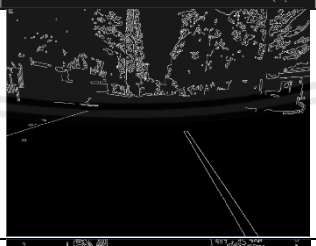
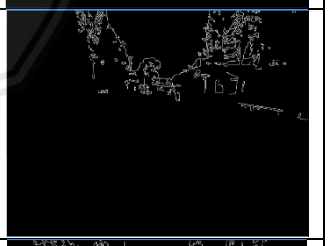
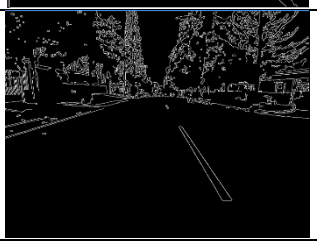
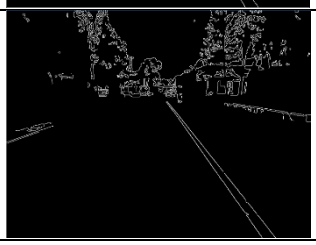
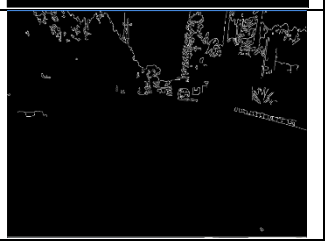
2. jika berada dibawah min maka akan dihilangkan.

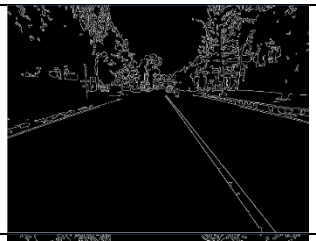
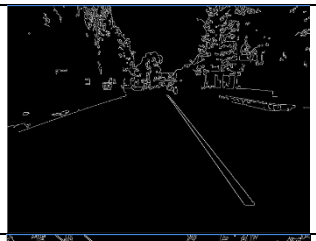
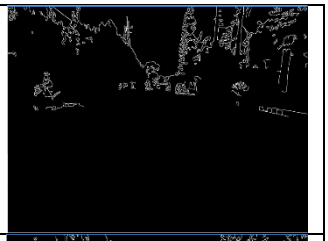
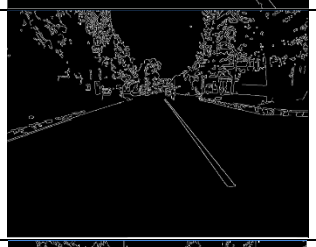
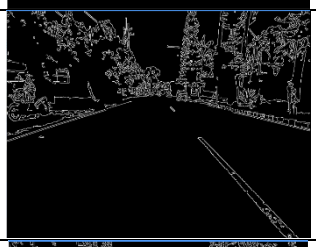
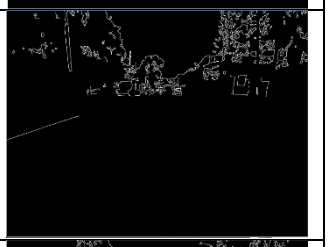
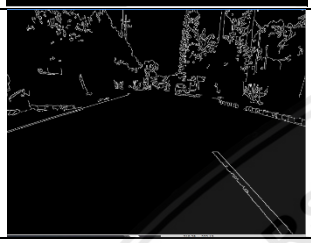
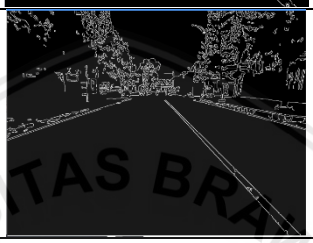
3. Mereka yang berada diantara dua ambang ini akan diklasifikasikan sebagai tepi atau bukan tepi dari konektivitasnya, jika terhubung ke piksel maka dinyatakan tepi jika tidak maka akan dibuang.

Untuk menentukan nilai kedua ambang tersebut, maka perlu dilakukan denganperbandingan dengan nilai ambang lain. Nilai ambang minimum bernilai 50 dan ambang minimum dengan nilai 200 akan dibandingkan dengan nilai 30, 120 dan 70,280. Hasil Perbandingan dapat dilihat pada Tabel 5.2

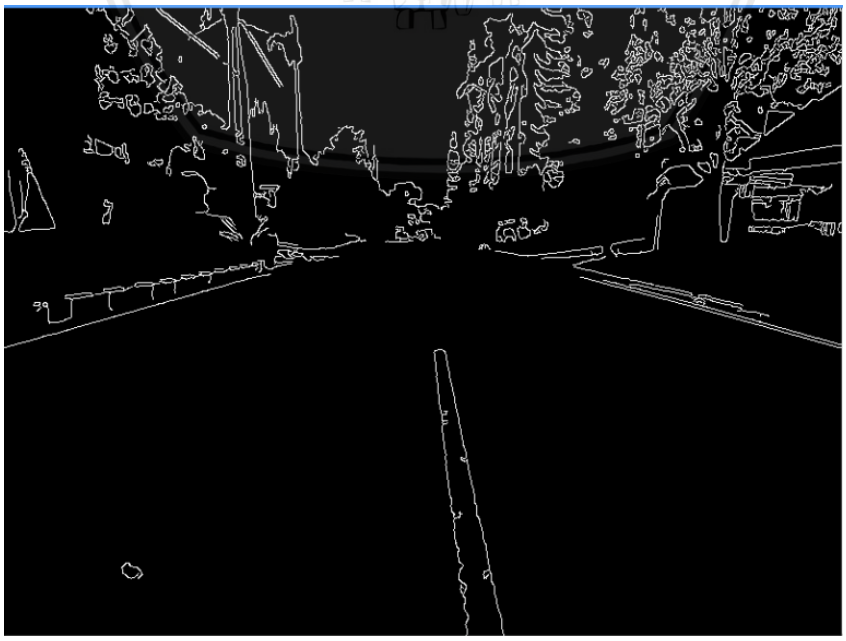
Tabel 5.2 Perbandingan Nilai Ambang

NO	Minimum:30 Maksimum: 120	Minimum:50 Maksimum: 200	Minimum:70 Maksimum: 280
1.			
2.			
3.			
4.			
5.			

6.			
7.			
8.			
9.			
10.			
11.			
12.			

13.			
14.			
15.			

Hasil yang didapatkan dari percobaan Tabel 5.2 bahwa nilai ambang yang paling cocok dengan peneltian adalah nilai maksimum 200 dan minimum 50 sebesar menjelaskan objek pada huruf A melebihi max sehingga dianggap sebagai tepi, sedangkan pada objek huruf B berada diantara kedua ambang dan tidak memiliki konekfitas sehingga bukan dianggap sebagai tepi dan juga akan dihilangkan,selanjutnya objek huruf C berada pada kedua ambang dan memiliki konektivitas dengan maka dapat dianggap sebagai tepi.Hasil ari proses deteksi tepi canny dapat dilihat pada Gambar5.17.



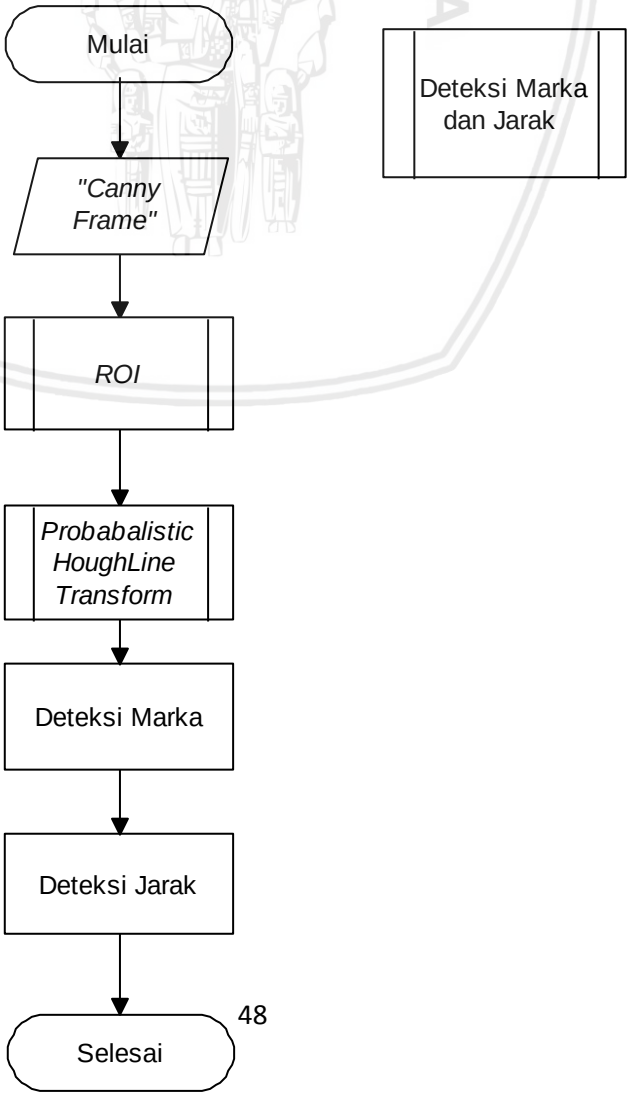
Gambar 5.17 Hasil Proses Deteksi Tepi Canny

5.2.1.3 Deteksi Marka dan Deteksi Jarak

Untuk mendeteksi sebuah objek garis yang berasal dari marka garis lurus dijalan maka di perlukan sebuah proses yang dapat mencari semua objek garis di jalan beserta berapa ukuran panjang garis yang terdeteksi, hal ini berguna sebagai acuan untuk sistem menentukan status dan tipe marka jalan yang terdeteksi.

Metode yang dibutuhkan untuk pengenalan marka garis lurus dan jarak diawali dariproses*cropping atau disebut proses mencari ROI(Region of interest)*. Proses *cropping atau masking*adalah proses menutupbidang gambar yang tidak dibutuhkan oleh sistem, sehingga hanya meninggalkan objek yang hanya dibutuhkan oleh sistem.Setelah daerah ROI didapatkan maka dilanjutkan dengan proses deteksi garis menggunakan metode *Probabalistic Houghline Transform.*

Kelebihan dari metode *Probabalistic Houghline Transform* adalah tidak semua garis yang dideteksi pada gambar akan ditampilkan hanya garis yang memiliki syarat khusus yang ditampilkan sesuai masukan yang dibutuhkan pada fungsinya. Hasil dari metode ini berupa garis dalam kordinat (x1 y1 , x2 y2), dari kedua kordinat inilah yang akan di proses pada proses berikutnya yaitu *hypot*untuk mendeteksi panjang garis dan proses deteksi jarak. Flowchart Deteksi Marka Garis lurus dan deteksi jarak dapat dilihat pada Gambar 5.18.

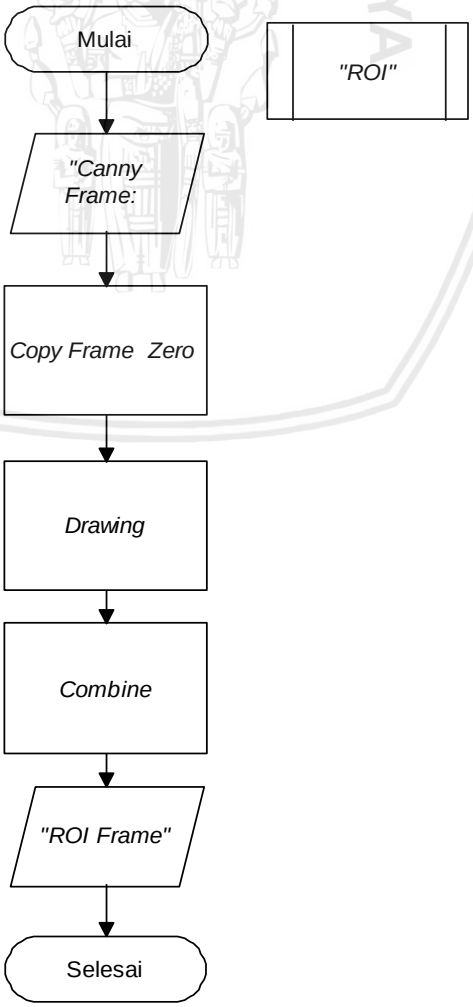


Gambar 5.18 Flowchart Deteksi Marka Garis Lurus dan Deteksi Jarak

A. ROI(Region of Interest)

ROI berfungsi untuk mencari objek yang dibutuhkan oleh sistem menggunakan teknik gambar dan tumpuk , konsepnya sama dengan *cropping*, bedanya jika pada cropping gambar akan dipotong sesuai kordinat yang dimasukan, sedangkan pada proses ini gambar asli masih utuh hanya saja gambar asli akan tumpuk olehgambar lain yang telah digambar menggunakan kordinat, sehingga hasilnya gambar asli seperti dipotong.

Pada pemrogaman OpenCV telah disediakan untuk melakukan teknik ini yaitu dengan fungsi***cv2.fillypoly****dan* ***cv2.bitwise_and.*** Fungsi ***cv2.fillpoly***digunakan untuk mengambar bidang tutupnya, yang dimana gambar yang akan diproses oleh fungsi ini adalah gambar kosong yang besar resolusinya sama dengan gambar asli. Fungsi ini membutuhkan tiga masukan dalam prosesnya yaitu pertama gambar yang akan diproses,kedua kordinat bidang gambar,ketiga warna pengisi bidang gambar. Sedangkan pada fungsi ***cv2.bitwise_and*** digunakan untuk menggabungkan dua gambar asli dengan gambar hasil proses **cv2.fillpoly.** Fungsi ini hanya membutuhkan dua masukan,pertama data gambar asli,keduadatagambar yang akan digabungkan. Flowchart proses ROI dapat dilihat pada Gambar 5.19.



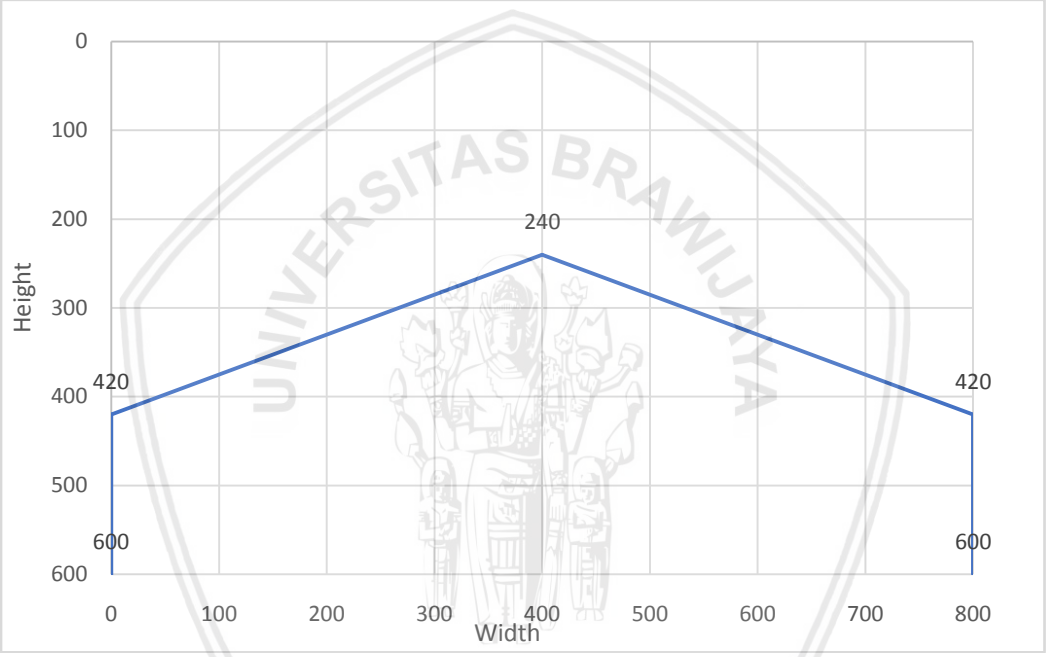
Gambar 5.19 Flowchart ROI

1. *Copy Frame Zero*

Proses ini adalah melakukan copy gambar pada *Canny Frame* tetapi gambar yang telah ditelah didalin akan dikosongkan dengan resolusi sama.

2. *Drawing*

Pada proses ini gambar yang didapat dari proses satu akan digambar bidang sesuai kordinat yang di tentukan. Fungsi yang digunakan adalah **cv2.fillpoly,** melakukan proses drawing dibutuhkan kordinat gambar yang nantinya dari kordinat tersebut dapat membentuk suatu bidang. Kordinat gambar dapat dilihat pada Gambar 5.20.



Gambar 5.20 Kordinat Gambar

Pada gambar 5.20 terdapat lima kordinat gambar yang membentuk suatu bidang,Urutan kordinat dapat dilihat dibawah:

a. Kordinat 1 (Width,Height) = (0,600)
b. Kordinat 2 (Width,Height) = (0,420)
c. Kordinat 3 (Width,Height) = (400,240)
d. Kordinat 4 (Width,Height) = (800,420)
e. Kordinat 5 (Width,Height) = (800,600)

Kordinat yang telah ditentuan diatas didapat melalui proses perbandingan dengan rangkaian kordinat lain, sehingga dari perbandingan kordinat tersebut dapat ditentukan kordinat mana yang lebih bagus digunakan oleh sistem. Pada tabel proses perbandingan terdapat tiga rangkaian kordinatsebagai berikut:

a. Pertama [(0,600),(0,420),(400,158),(800,420).(800,600)]
b. Kedua [(0,600),(0,420),(400,198),(800,420).(800,600)]

c. Ketiga [(0,600),(0,420),(400,240),(800,420),(800,600)]

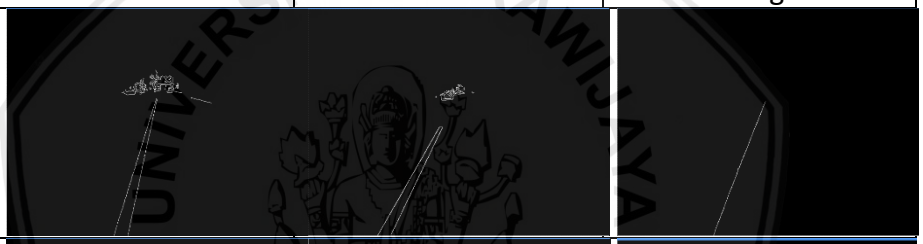
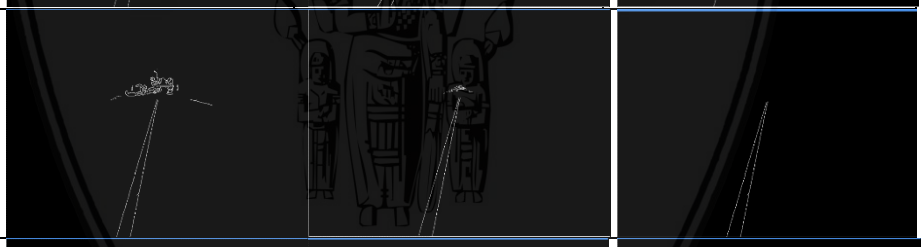
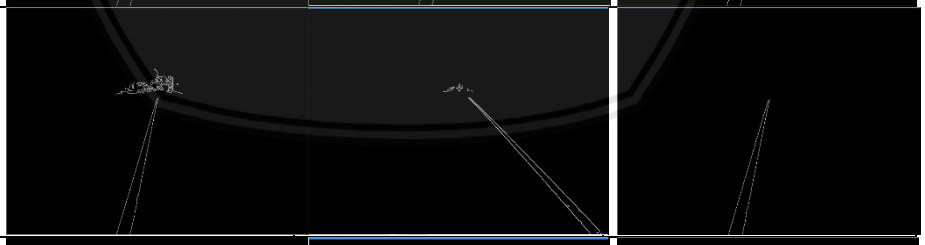
Setelah kordinat didapatkan maka langkah selanjutnya adalah memberikan warna masking pada bidang yang telah digambar dengan memberikan masukan ketiga dengan nilai 255, nilai 255 disini memiliki aarti memberikanwarnahitampadabidang gambar.

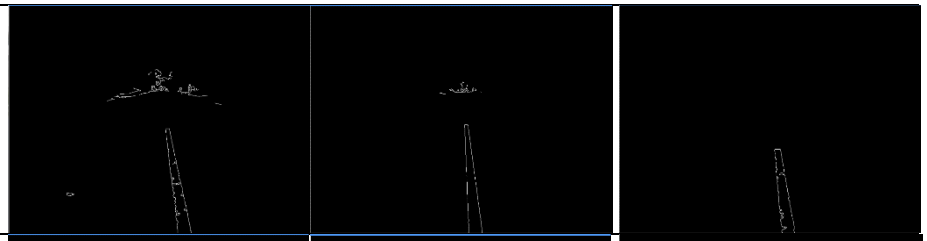
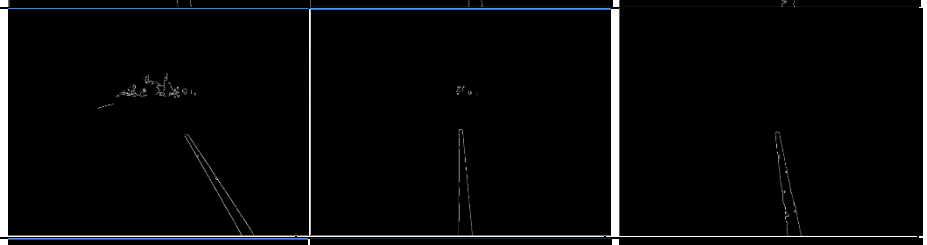
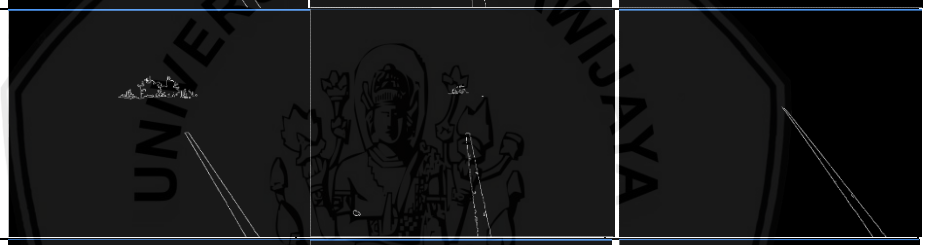
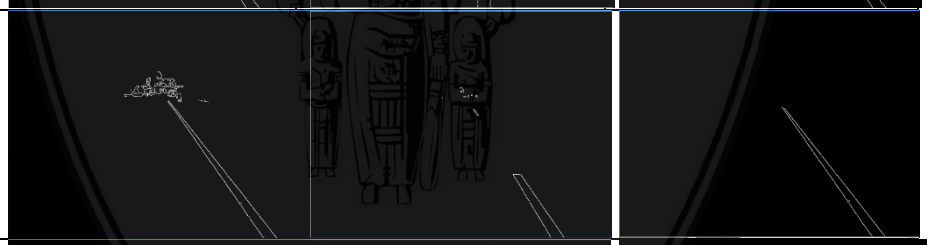
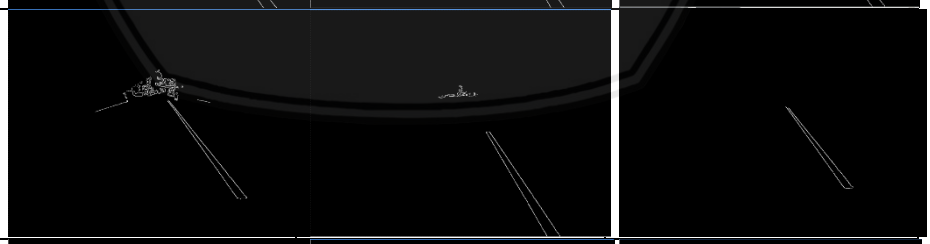
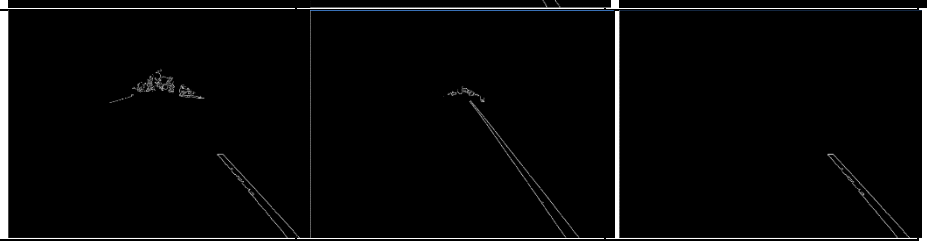
3. *Combine*

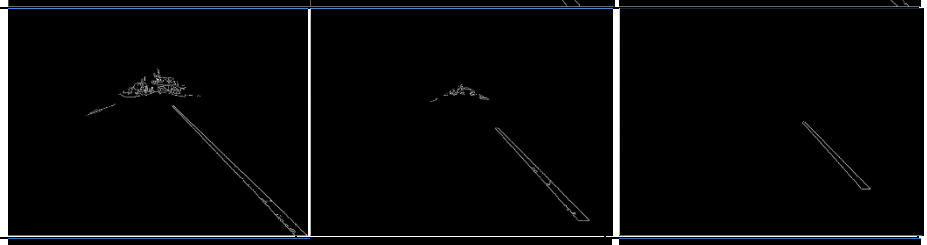
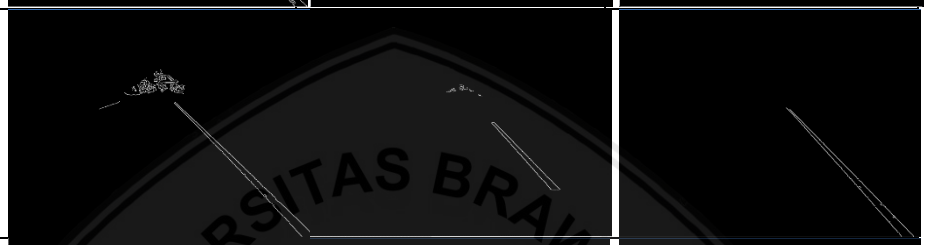
Pada proses ini gambar yang dihasilkan pada proses drawing nilaik piksel akan di kalikan dengan nilai gambar *canny frame*menggunakan fungsi***cv2.bitwise_and***.Sehingga didapatkan satu gambar yang seolah olah gambar mengalami cropping.

Hasil perbandingan kordinat gambar dapat dilihat pada Tabel 5.3

Tabel 5.3 Perbandingan Rangkaian Kordinat

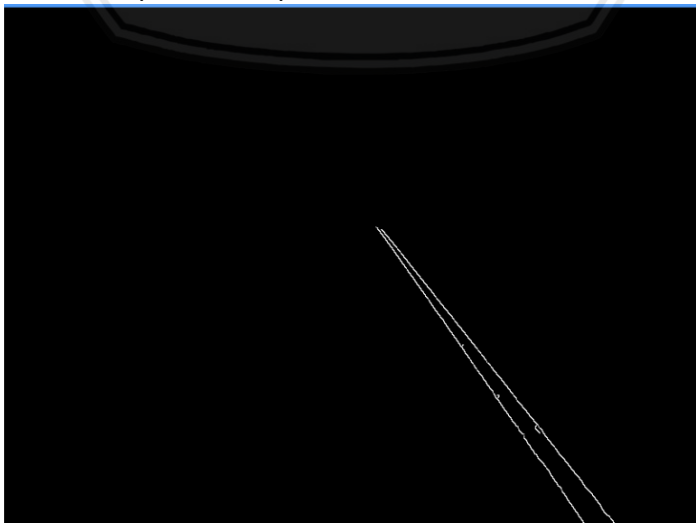
NO	Kordinat Pertama	Kordinat Kedua	Kordinat Ketiga
1.			
2.			
3.			
4.			

5.	
6.	
7.	
8.	
9.	
10	
11.	

12.	
13.	
14.	
15.	

Dari hasil perbandingan didapatkan rangkaian kordinat terbaik adalah pada Kordinat ketiga, Karena pada kordinat ketiga hasil dari gambar yang dihasilkan fokus terhadap marka jalan yang akan dideteksi sedangkan objek lain yang tidak dibutuhkan oleh sistem tertup dengan baik.

Hasil dari proses ROI dapat dilihat pada Gambar 5.21.

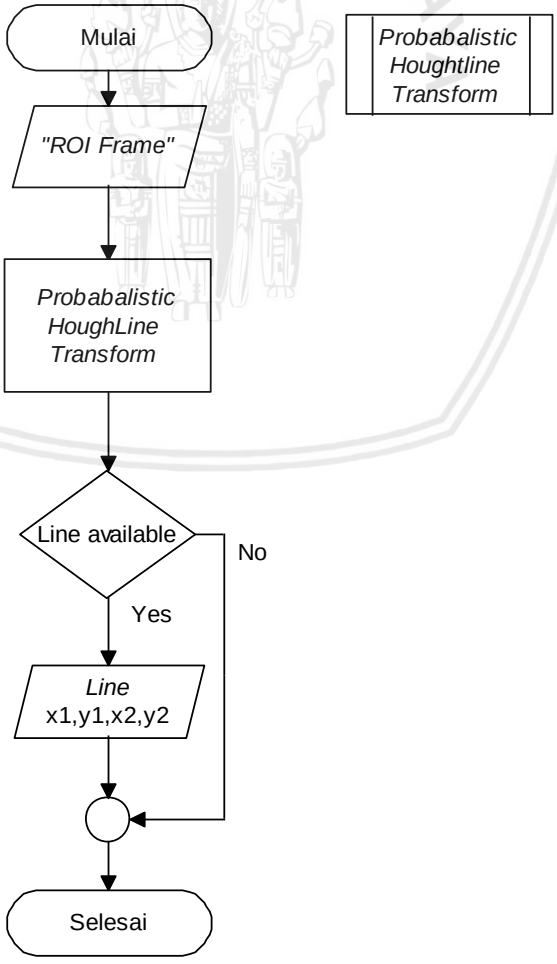


Gambar 5.21 Hasil Proses ROI

B. Probabalistic Houghline Transform

Metode ini berfungsi untuk mendeteksi hasil proses deteksi tepi *canny* yang yang menghasilkan gambar garis dari tepian objek yang terdeteksi menjadi garis yang dapat dideteksi kordinat (x,y). Pada proses ini garis yang sesuai syarat panjang minimal dan panjang gapnya saja yang akan diambil, sedangkan garis yang tidak memenuhi syarat akan dibuang. Hasil dari proses ini yang berupa kordinat garis akan diproses proses hypot dan deteksi jarak.

Pada pemrogaman OpenCV telah tersedia *library* untuk menggunakan metode ini, yaitu ***cv2.HoughLinesP***. Fungsi ***cv2.HoughLinesP***ini membutuhkan 6 variabel data untuk melakukan prosesnya, variabel data pertama yaitu data gambar yang akan di cari garisnya,kedua nilai rho, ,ketiga nilai theta, keempat nilai *thereshold* nilai ini yang menentukan piksel yang dideteksi dimunculkan atau dihilangkan, kelima nilai panjang minimal, nilai ini adalah syarat garis minimal yang terdeteksi, jika garis memiliki panjang dibawah nilai ini maka akan dihapus, keenam nilai panjang gap, nilai ini adalah syarat kedua garis yang akan diambil, jika garis memiliki panjang melebihan nilai panjang gap maka garis tersebut juga akan dihapus. Flowchart Probabalistic Houghline Transform dapat dilihat pada Gambar 5.22.



Gambar 5.22*Flowchart Probabalistic Houghline Transform*

1. *Probabalistic HoughtLine Transform*

Para proses ini untuk menjalankan fungsi pada OpenCV membutuhkan data parameter input, seperti rho,theta,threshold,minlinelenght,maxlinegap Proses menentukan nilai dari parameter diatas akan dijelaskan dibawah ini:

a. *Rho dan Theta*

Pada proses ini adalah menentukan nilai dari *Rho*dan *ThetaI* . *Rho* adalah resolusi jarak antar piksel dari titik (0,0).Theta adalah derajat jangkauan garis dalam bentuk radian, karena dalam bentuk radian kita perlu melakukan perhitungan dengan rumus *radian*berikut:

$$\text{Radian}={}^{np.pi}/_{derajat} \qquad (5.2)$$

Ket :

Radian	: satuan sudut dalam bidang
np.pi	: nilai maksimal
derajat	: nilai yang di masukan

Nilai radian didapat dari pembagian *np.pi* yang memiliki nillai 180 derajat dengan nilai derajat yang dimasukan yaitu 60, sehingga didapatka nilai radian sebesar 3. Proses perhitungan dapat dilihat berikut

$$\text{Radian} = {}^{180°}/_{60°} = 3 \qquad (5.3)$$

b. *Threshold*

Proses ini adalah menentukan nilai dari threshold yang digunakan dalamsistem. Threshold berguna untuk memunculkan titik garis yang di perlukan oleh sistem dan menghilangkan titik garis yang tidak perlu. Cara kerja*threshold* yaitumembandingkan nilai piksel gambar dengan nilai *threshold* yang digunakan sebagai berikut.

Cara kerja dari proses Threshold sebagai berikut:

1. Jika nilai piksel lebih rendah dari 90 maka nilai piksel tersebutakan dianggap 0 dan nilai 0 ini adalah warna hitam.

2. jika nilai piksel lebih besar sama dengan 90,maka nilai piksel ini dianggap 1, yang dimana nilai 1 sama dengan nilai piksel 255 yang dianggap warna putih.

c. *Minlenght* dan *Maxlinegap*

Setelah garis ditemukan langkah selanjutnya adalah melakukan penyaringangaris sesuai panjangnya,karena jika tidak dilakukanproses ini,maka akan banyak garis kecil kecil yang sebenarnya dibutuhkan oleh sistem. Terdapat dua parameter penyaring yaitu *minlinelenghtdan maxLineGap.*Nilai kedua parameter yang digunakan sebagai berikut

Cara kerja proses ini sebagai berikut:

1. Jika panjang garis dibawah parameter *minlenght* maka garis tersebut tidak diloloskan.

2. Jika panjang garis melebihi parameter *maxlinegap* maka garis tersebut tidak diloloskan.

3. Jika panjang garis diatas parameter minlenght dan dibawah parameter maxlinegap maka garis tersebut akan diloloskan.

Untuk mendapatkan kombinasi nilai variabel diatas maka, didapat melalui proses perbandingandengan berbagai kombinasi nilai, sehingga dari perbandingan nilai tersebut dapat ditentukan kombinasi nilai mana yang lebih cocok digunakan oleh sistem. Pada tabel proses perbandingan terdapat tiga kombinasi nilai sebagai berikut:

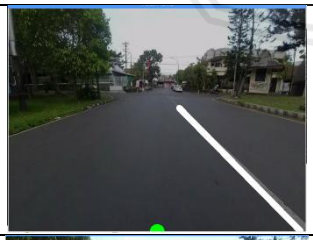
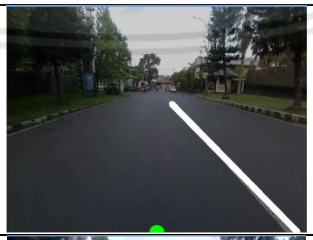
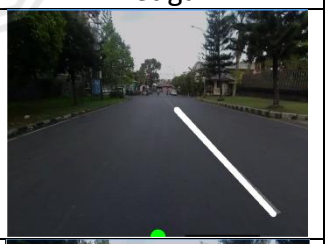
1. Pertama
   (rho:3,theta:1,threshold:80,minlinelenght:10,maxlinegap:300)

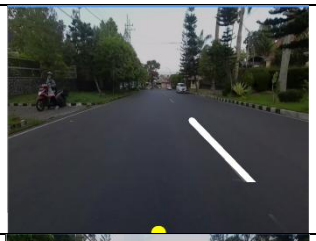
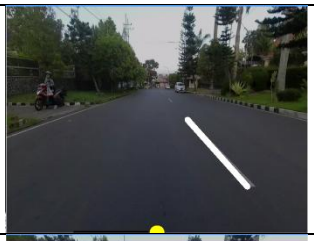
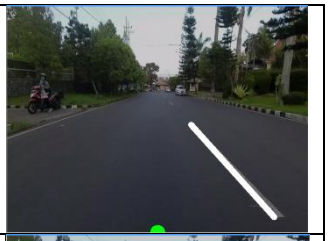
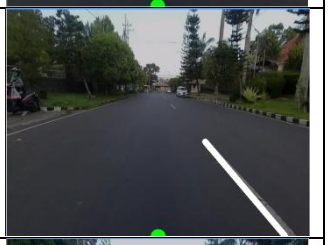
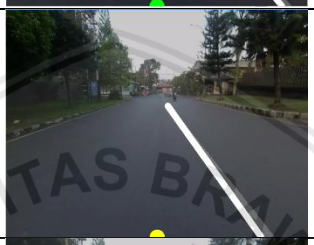
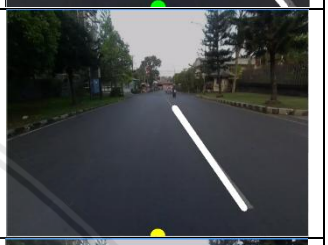
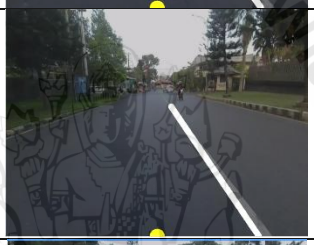
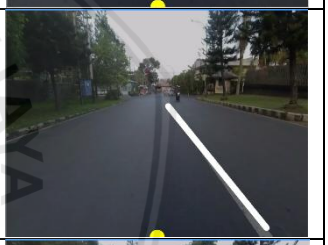
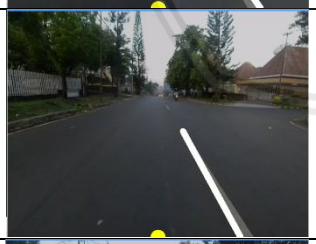
2. Kedua
   (rho: 4,theta:2,threshold:100,minlinelenght:20,maxlinegap:600)

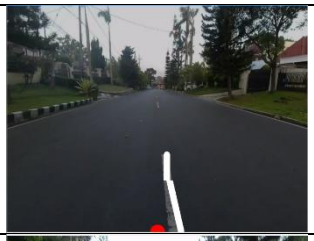
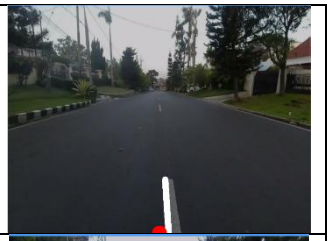
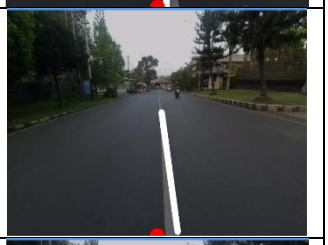
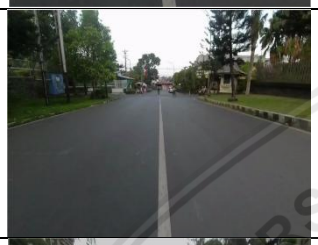
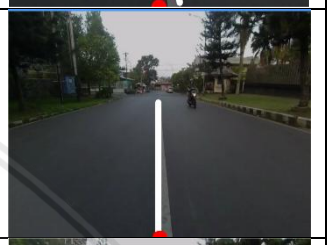
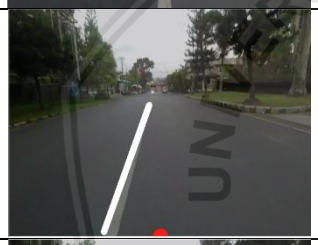
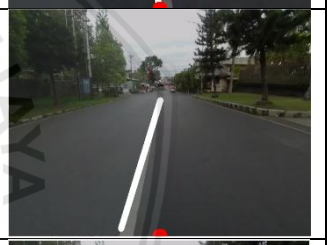
3. Ketiga
   (rho:5,theta:3,threshold:90,minlinelenght:30,maxlinegap:900)

Hasil perbandingan dapat dilihat pada Tabel 5.4

Tabel 5.4 Perbandingan Kombinasi Nilai

NO	Nilai Pertama	Nilai Kedua	Nilai Ketiga
1.			
2.			

3.			
4.			
5.			
6.			
7.			
8.			
9.			

10.			
11.			
12.			
13.			
14.			
15.			

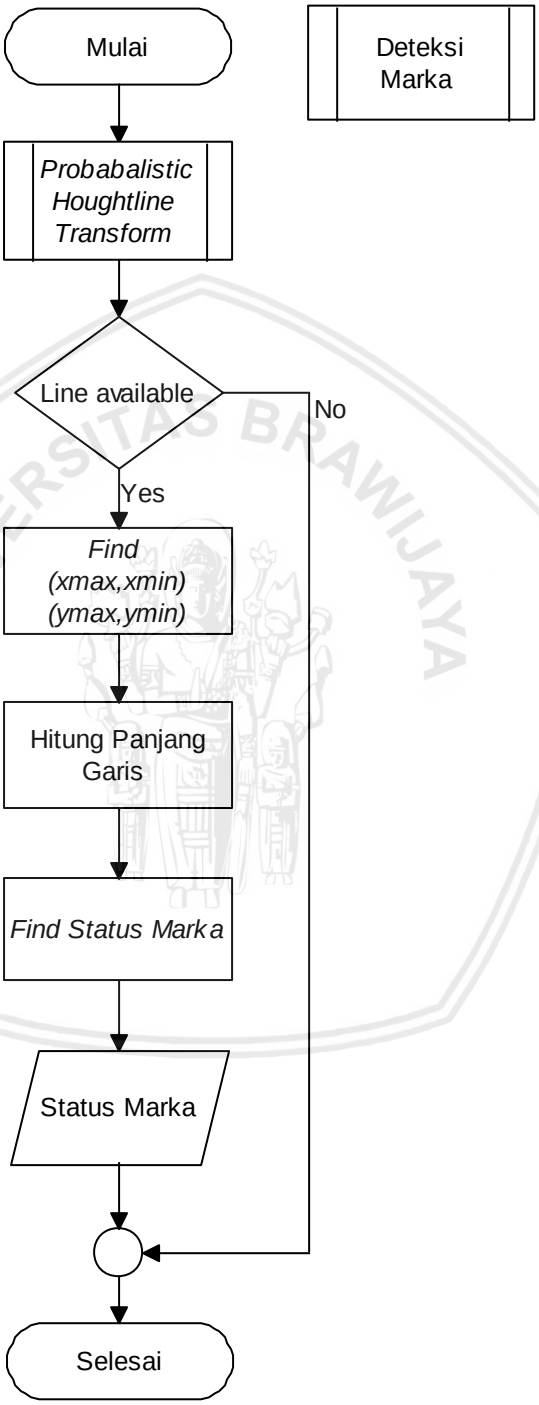
Dari hasil perbandingan pada Tabel 5.4 maka didapatkan rangkaian nilai terbaik adalah pada nilai **ketiga**, Karena pada rangkaian nilai tersebut dapat mendeteksi garis dengan baik. Sehingga hasil dari proses ini menghasilkan data garis yang dibutuhkan oleh sistem untuk melakukan prosesselanjutnya. Hasil dari proses *probabalistic hought transform* ini berupa kordinat garis dalam bentuk(x1,y1,x2,y2).

Hasil dari proses Probabalistic HouhLine Transformdapat dilihat pada Gambar 5.21.

Gambar 5.23 Hasil Proses Probabalistic HoughLine Transform

C. Deteksi Marka

Pada proses deteksi marka menggunakan fungsi Hypot, fungsi matematika ini digunakan untuk mencari panjang sisi miring dari segitiga. Fungsi matematika ini digunakan untuk mencari panjang diagonal dari garis marka jalan yang dideteksi dari proses sebelumya.Hal ini dilakukan karena posisi garis yang terdeteksi oleh kamera tidak semuanya lurus sejajar, tetapi ada kalanya garis marka yang terdeteksi oleh sistem mempunyai posisi agak miring kekiri maupun kekanan karena efek lensa kamera yang dipakai memiliki lensa wide angle sebesar 150 derajat sehingga objek di samping kiri dan kanan nampak ujung garisnya tertarik ke tengah ujung. Untuk menggunakan fungsi ini dibutuhkan dua buah data yaitu nilai panjang dasar dan nilai tinggi, kedua nilai tersebut didapat dari nilai xmax,xmin,ymax,ymin dari garis .Flowchart proses hypot dapat dilihat pada Gambar 5.24.



Gambar 5.24*FlowchartDeteksi Marka*

1. *Find* (xmax,xmin)(ymax,ymin)

Pada proses ini akan ditentukan dari keempat kordinat pembentuk garis mana yang akan masuk mewakili variabel max dan min. Untuk menentukan hal tersebut dibuatlah sebuah persamaa sebagai berikut:

1. nilai awal dari xmax,xmin,ymax,ymin sebagai berikut:
xMIn = 10000, xMax = -10000
yMIn = 10000, yMax = -10000

2. Selanjutnya dari nilai tersebut akan dicari nilai max dan min sesuai persamaan berikut:

```
if (y2 <(yMIn)):
    yMIn = y2
    xMIn = x2

if (y1 <(yMIn)):
    yMIn = y1
    xMIn = x1

if (y1>( yMax)):
    yMax = y1
    xMax = x1

if (y2>( yMax)):
    yMax = y2
    xMax = x2
```

Setelah kodinat garis dari xMax,xMin,yMax,yMin ditemukan maka tahap berikutnya adalah tahap menghitung panjang garis.

2. Hitung Panjang Garis

Pada proses ini menggunakan fungsi matematika yaitu hypot.Hypot adalah fungsi matematika yang digunakan untuk menghitung panjang garis dari dua kordinat garis. Rumus dari hypot dapat dilihat dibawah:

$$PanjangGaris = \sqrt{(xMax - xMin)^2 + (yMax - yMin)^2} \qquad (5.4)$$

Hasil nilai panjang garis ini akan di proses pada langkah selanjutnya.

3. *Find Statuss*

Pada tahap ini nilai panjangaris yang didapat pada prosessebelumnya akan dimasukan dalam persamaan yang memiliki nilai penentu, nilai penentu didapatkan dari rekap nilai panjanggaris yang terdeteksi pada gambar marka membujur utuh dan putus,rekap nilai dapat dilihat pada tabel 5.5

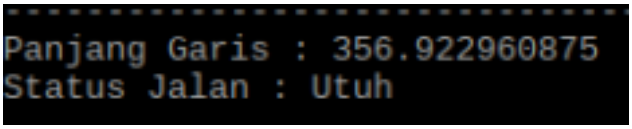
Tabel 5.5Rekap Nilai Panjang Garis pada Marka Membujur Utuh

NO	Nilai Panjang Garis pada Marka Membujur Utuh (piksel)	NO	Nilai Panjang Garis pada Marka Membujur Utuh (piksel)	NO	Nilai Panjang Garis pada Marka Membujur Utuh (piksel)
1.	353.8	16.	345,5	31.	469,1
2.	372,6	17.	366,9	32.	475,9
3.	356,7	18.	437,4	33.	446,4
4.	349,6	19.	426,0	34.	438,4
5.	356,9	20.	440,2	35.	421,4
6.	356,9	21.	444,8	36.	439,7
7.	343,8	22.	454,3	37.	426,0
8.	418,9	23.	466,7	38.	357,0
9.	424,4	24.	479,4	39.	359,0
10	427,8	25.	469,5	40.	345,5
11.	345.8	26.	356,9	41.	356,0
12.	469,5	27.	357,0	42.	356,7
13.	455.1	28.	355,5	43.	359,2
14.	454.5	29.	357,0	44.	372,4
15.	480.1	30.	373,8	45.	342,7
Nilai Ninimal Panjang Garis pada Video Marka Lurus Bernilai					
342,7					

Dari Tabel 5.5 didapatkan bahwa nilai minimal panjang garis pada marka membujur garis utuh adalah 342,7 untuk menghindari adanya drop nilai pada kondisi marka membujur garis utuh maka nilai penentu tersebut akan dibulatkan menjadi 340, sehingga didapatkan persamaan sebagai berikut:

$$Statuss = \text{Putus-Putus} \quad (5.5)$$
$$If\ (PanjangGaris > 340)$$
$$Statuss = "Utuh"$$

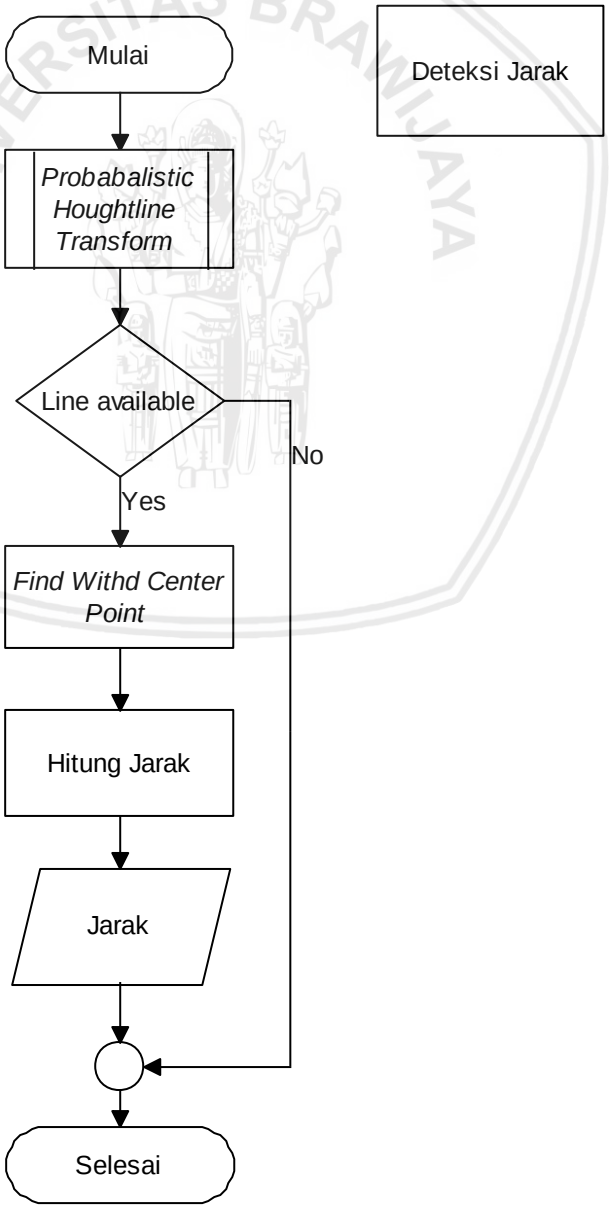
Dari hasil proses deteksi marka adalah sebuah status yang diatampilkan pada terminal pada Gambar 5.25.



Gambar 5.25 Hasil Proses Deteksi Marka

D. Deteksi Jarak

Proses yang digunkan deteksi jalan ini adalah dengan teknik pengurangan dari nilai x1 pada garis yang tedeteksi dengan nilai*width center point*. Width center pointadalah nilai Setengah dari width frame. Nilai ini menjadi acuan bahwa pada nilai tersebut adalah posisi tengah dari roda depan kendaraan bermotor. Selanjutnya adalah proses *find status* yaitu proses pengurangan piksel x1 yang berasal dari garis dengan nilai*width center point*. Flowchart Deteksi jarak dapat dilihat pada Gambar 5.26.



Gambar 5.26*Flowchart Deteksi Jarak*

1. *Find Width Center Point*

Teknik ini berfungsi untuk mencari titik tengah dari lebar frame, hal ini dilakukan untuk menjadi kan titik tengah tersebut adalah posisi tengah dari roda depan motor. Proses mencari titik tengah ini dengan cara pembagian sesuai rumus berikut:

$$\text{Center} = {}^{width\ frame}/_{2} \quad (5.6)$$

Ket:

Width frame : lebar piksel frame

Sehingga didapatkan hasil perhitungan sebagai berikut:

$$\text{center} = {}^{800}/_{2} = 400 \quad (5.7)$$

Dari perhitungan diatas maka nilai width center poin dimasukan dalam variabel *center* yang bernilai 400 piksel, serta ditentukan juga bahwa nilai tersebut adalah titik tengah frameserta titik tingah posisi roda depan kendaraan.

2. Hitung jarak

Proses ini adalah proses lanjutan dari proses find width center point. pada proses ini, dilakukan penghitungan jarak antara nilai with center point dengan garis kordinat x1, dengan teknik pengurangan.Karena nilai x1 garis dengan nilai *widthcenter point*berada pada nilai*width*maka perhitungan dapat langsung dilakukan sesuai rumus 5...

$$jarak = x1 - center \quad (5.8)$$

Hasil dari perhitungan ini akan menjadi data jarak yang nantinya akan menjadi data masukan pada proses penentuan status.

5.2.1.4 Penentuan Status Marka dan Status Jarak

Tipe jalan yang telah ditentukan statusnya akan dikombinasikan dengan status jarak untuk mendapatkan output yang sesuai, yang dimana status jarak didapatkan dari hasil perbandingan nilai panjnag jarak dengan nilai batas status jarak yaitu status AWAS, HATI-HATI, AMAN, dan BAHAYA. Untuk menentukan nilai batas tersebut maka perlu dilakukan menampilkan nilai jarak yang terdeteksi dengan berbagai nilai batasnya dan dicari nilai batas terbaik pada tiap statusnya. Nilai jarak diambil dari video yang sesuai dengan statusnya.

A. Batas Status Awas

Pada batas status awas dilakukan perbandingan dengan menggunakan tiga nilai, nilai pertama adalah 40, dan nilai selanjutnya diambil dari nilai jarak dari garis maksimal tiap tabel. nilai tersebut nantinya akan dipasang pada proses, jika nilai jarak lebih kecil sama dengan dari nilai tersebut maka lolos jika tidak maka tidak lolos. Nilai yang memiliki banyak lolos maksimal akan di jadikan batasnya. Proses penentuan batas awas dapat dilihat pada Tabel 5.6 sampai Tabel 5.8.

Tabel 5.6Penentu Batas Awas Satu

Nilai jarak dari Garis<=40		
No.	Jarak dari Garis (piksel)	Status
1.	77	Tidak Lolos
2.	65	Tidak Lolos
3.	77	Tidak Lolos
4.	81	Tidak Lolos
5.	38	Lolos
6.	53	Tidak Lolos
7.	52	Tidak Lolos
8.	53	Lolos
9.	34	Tidak Lolos
10.	44	Lolos
11.	16	Lolos
12.	38	Tidak Lolos
13.	65	Tidak Lolos
14.	62	Tidak Lolos
15.	38	Lolos
Jarak dari garis max: 81		

Pada Tabel 5.6 nilai batas awal yang digunakan adalah 40, dan setelah dilakukan cek dengan 15 frame maka nilai tersebut masih belum cocok digunakan sistem ,sehingga dicari nilai piksel dari jarak ke garis terbesar yang nantinya kan digunakan sebagai batas pada Tabel 5.7.

Tabel 5.7Penentu Batas Awas Dua

Nilai jarak dari Garis<=81		
No	Jarak dari Garis (piksel)	Status
1.	51	Lolos
2.	55	Lolos
3.	53	Lolos
4.	58	Lolos
5.	62	Lolos
6.	65	Lolos
7.	90	Tidak Lolos

8.	85	Tidak Lolos
9.	100	Tidak Lolos
10.	44	Lolos
11.	45	Lolos
12.	54	Lolos
13.	58	Lolos
14.	38	Lolos
15.	98	Tidak Lolos
Jarak dari garis max: 100		

Pada Tabel 5.7 batas yang digunakan adalah nilai piksel jarak dari garis terbesar pada Tabel 5.6 dan akan dicoba pada 15 frame berbeda, dan didapatkan adanya ketidak cocokan nilai batas untuk digunakan sistem, sehingga dicari nilai piksel jarak dari garis terbesar yang nantinya akan menjadi batas pada Tabel 5.8.

Tabel 5.8Penentu Batas Awas Tiga

Nilai jarak dari Garis<=100		
No	Jarak dari Garis (piksel)	Status
1.	23	Lolos
2.	28	Lolos
3.	24	Lolos
4.	53	Lolos
5.	24	Lolos
6.	16	Lolos
7.	44	Lolos
8.	45	Lolos
9.	81	Lolos
10.	90	Lolos
11.	85	Lolos
12.	73	Lolos
13.	54	Lolos
14.	24	Lolos
15.	62	Lolos
Jarak dari garis max: 95		

Pada Tabel 5.8 batas yang digunakan adalah nilai piksel jarak dari garis terbesar pada Tabel 5.7, dan didapatkan kecocokan nilai batas untuk digunakan pada sistem.Dari perbandinagan ketiga tabel diatas maka didapatkan bahwa nilai batas paling baik sebesar 100.

B. Batas Status Hati - hati

Pada batas status hati - hati dilakukan perbandingan dengan menggunakan tiga nilai, nilai pertama adalah 230, dan nilai selanjutnya diambil dari nilai jarak dari garis maksimal tiap tabel. nilai tersebut nantinya akan dipasang pada proses, jika nilai jarak lebih kecil sama dengan dari nilai tersebut maka lolos jika tidak maka tidak lolos. Nilai yang memiliki banyak lolos maksimal akan dijadikan batasnya. Proses penentuan batas awas dapat dilihat pada Tabel 5.9 sampai Tabel 5.11.

Tabel 5.9Penentu Batas Hati - hati Satu

Nilai jarak dari Garis<=230		
No	Jarak dari Garis (piksel)	Status
1.	220	Lolos
2.	211	Lolos
3.	222	Lolos
4.	232	Tidak Lolos
5.	171	Lolos
6.	169	Lolos
7.	235	Tidak Lolos
8.	224	Lolos
9.	220	Lolos
10.	248	Tidak Lolos
11.	219	Lolos
12.	225	Lolos
13.	236	Tidak Lolos
14.	230	Lolos
15.	215	Lolos
Jarak dari garis max: 248		

Pada Tabel 5.9 nilai batas awal yang digunakan adalah 230, dan setelah dilakukan cek dengan 15 frame maka nilai tersebut masih belum cocok digunakan sistem ,sehingga dicari nilai piksel dari jarak ke garis terbesar yang nantinya kan digunakan sebagai batas pada Tabel 5.10.

Tabel 5.10Penentu Batas Hati - hati Dua

Nilai jarak dari Garis<=248		
No	Jarak dari Garis (piksel)	Status
1.	251	Lolos
2.	230	Lolos
3.	285	Tidak Lolos
4.	251	Lolos
5.	246	Lolos
6.	251	Lolos

7.	258	Lolos
8.	289	Tidak Lolos
9.	288	Tidak Lolos
10.	260	Lolos
11.	188	Lolos
12.	274	Tidak Lolos
13.	290	Tidak Lolos
14.	276	Tidak Lolos
15.	205	Lolos
Jarak dari garis max: 290		

Pada Tabel 5.10 batas yang digunakan adalah nilai piksel jarak dari garis terbesar pada Tabel 5.9 dan akan dicoba pada 15 frame berbeda, dan didapatkan adanya ketidak cocokan nilai batas untuk digunakan sistem, sehingga dicari nilai piksel jarak dari garis terbesar yang nantinya akan menjadi batas pada Tabel 5.11.

Tabel 5.11Penentu Batas Hati - hati Tiga

Nilai jarak dari Garis<=290		
No	Jarak dari Garis (piksel)	Status
1.	263	Lolos
2.	278	Lolos
3.	282	Lolos
4.	282	Lolos
5.	277	Lolos
6.	285	Lolos
7.	275	Lolos
8.	270	Lolos
9.	260	Lolos
10.	254	Lolos
11.	285	Lolos
12.	281	Lolos
13.	284	Lolos
14.	264	Lolos
15.	290	Lolos
Jarak dari garis max: 290		

Pada Tabel 5.11 batas yang digunakan adalah nilai piksel jarak dari garis terbesar pada Tabel 5.10, dan didapatkan kecocokan nilai batas untuk digunakan pada sistem. Dari perbandinagan ketiga tabel diatas maka didapatkan bahwa nilai batas paling baik sebesar 290.

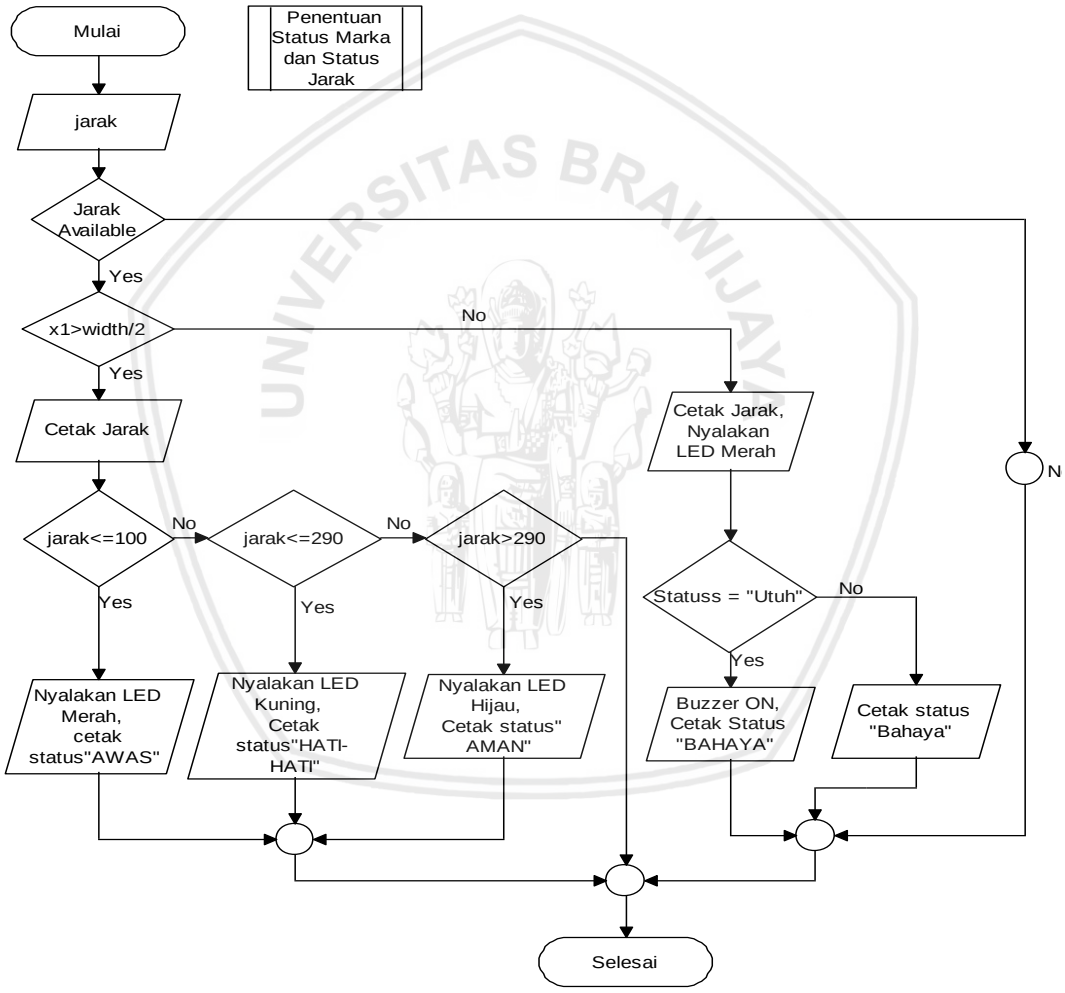
C.Batas Status Aman

Pada batas status aman adalah batas paling jauh dari marka sehingga nilai batas status aman diambil dari nilai batas hati-hati keatas yaitu lebih besar dari 290.

D.Batas Status Bahaya

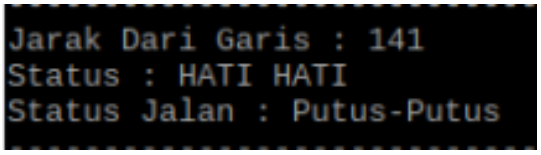
Pada batas status bahaya selalu memiliki nilai negatif, sehingga nilai batas status bahaya adalah harus dibawah nol

Dari hasil perbandingan diatas maka didapatkan sebuah rancangan diagram alir pada Gambar 5.27.



Gambar 5.27*FlowchartPenentuan Status Marka dan Status Jarak*

Hasil dari proses penentuan status marka dan Jalan dapat dilihat pada Gambar5.28



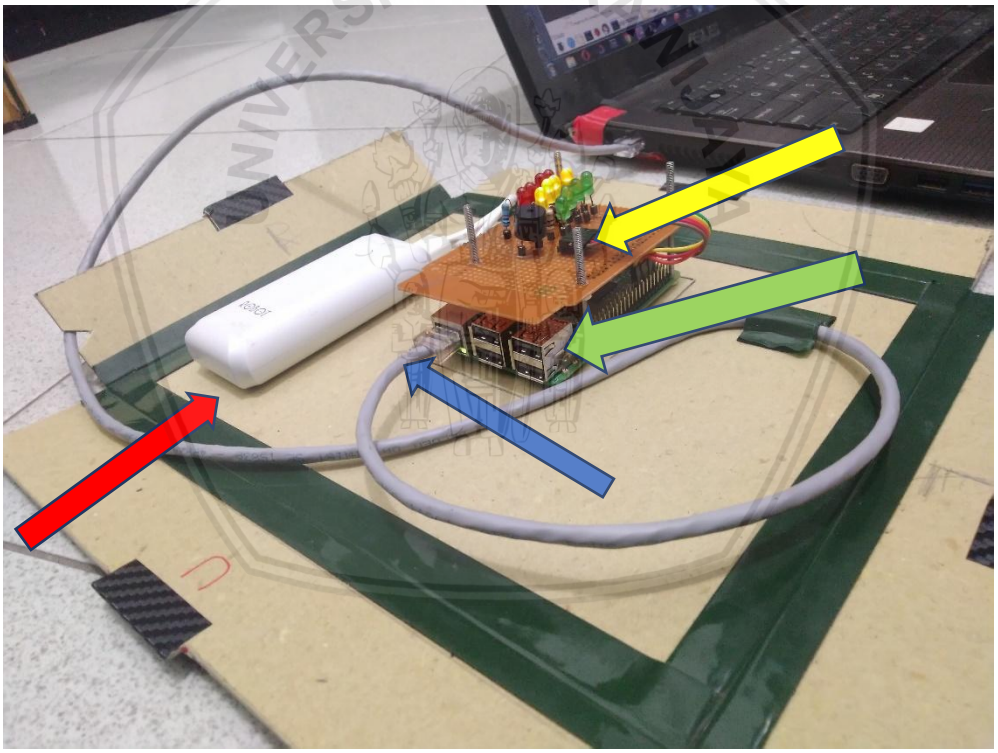
Gambar 5.28 Hasil Proses Penentuan Status Marka dan Status Jarak

5.3 Implementasi Sistem

Pada sub bab ini akan ditampilkanhasil dari perancangan perangkat keras dan, perangcangan perangkat lunak yang diaplikasikan secara nyata.

5.3.1 Implementasi Perangkat Keras

Implementasi perangkat keras meliputi rancangan catu daya, perancangan papan notifikasi, perancangan interface, perancangan purwarupa, penempatan kamera akan dirangkai menjadi satu yang akan ditampilkan pada Gambar 5.29.



Gambar 5.29 Perancangan Sistem dari Dalam

Keterangan:

: Raspberry Pi 3 (Pemroses)

: Papan Notifikasi (*Output)*

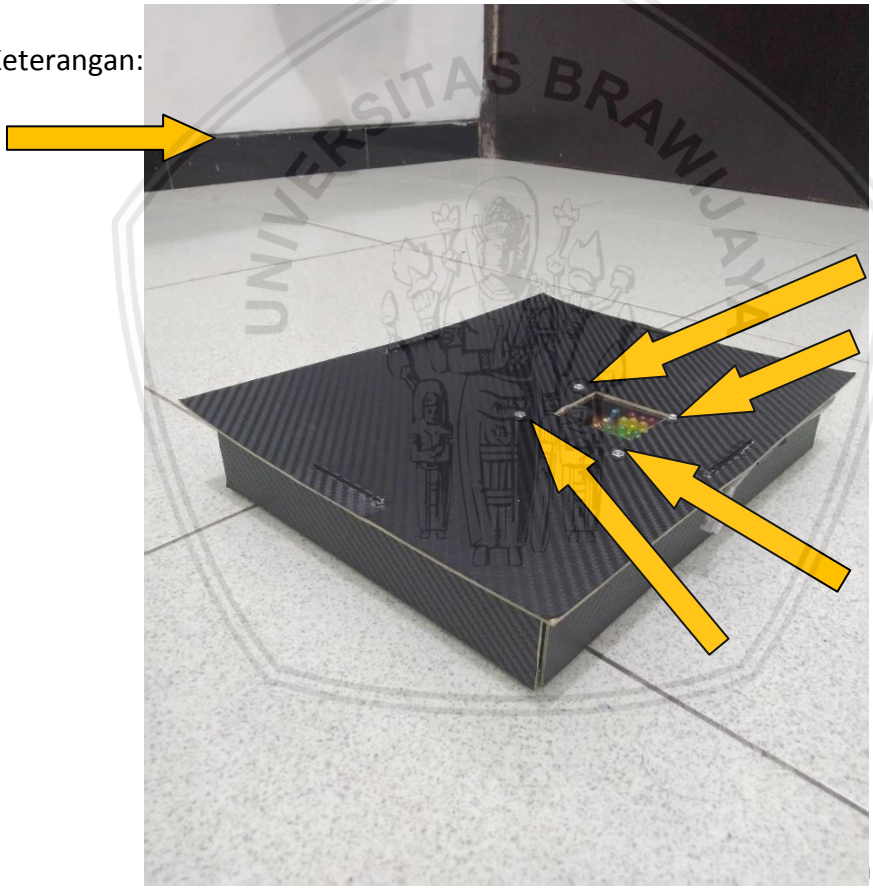
: Power Bank (*Power Supply*)

: Kabel RJ45 Crozz (konektor *Interface* dengan laptop)

Dari keempat perangkat tersebut berada pada satu balok yang terbuat dari kertas karton tebal, Agar terlihat menarik maka sisi luar balok karton di berikan lapisan stiker dengan motif karbonfiber. Pada bagian penutup atas terdapat lubang berbentuk persegi panjang berguna sebagi output dari papan notifikasi. Lubang tersebut di beri lapisan yang terbuat dari mika sehingga papan penutup bagian lebih rapi.

Dalam pembuatan purwarupan ini,penutup bagian atas mempunyai 4 kuncian yang menggunakan baut dan mur , mur tersebutdipasangkan pada tiap baut penyangga bagian atas dari rasberry dengan papan notifikasi, hal ini dilakukan agar penutup bagian atas dapat tertutup dengan sempurna, dan bentuk dari purwarupa dapat tetap berdiri kokoh. Bentuk purwarupa dapat dilihat pada Gambar 5.30.

Keterangan:



pat baut penyangga.

Sedangkan pada penempatan Kamera dibutuhkan karena objek yang diambil adalah jalan raya sehingga saat melakukan proses pengambilan video bisa berjalan dengan lancar dan aman. Dikarenakan motor yang digunakan adalah tipe automatis maka headlamp berada pada dada depan motor,bukan berada pada stir motor. Penempatan kamera di rekatkan pada posisi yang telah ditentukan dengan menggunakan perekat dua sisi yang kuat, hal ini digunakan

agar saat pengambilan video getaran pada kamera akan direduksi, untuk lebih detail penempatan kamera bisa dilihat pada Gambar 5.31.

Gambar 5.31Penempatan Kamera Tampak Depan dan Samping

Penempatan kamera berada diatas lampu dan dibawah windshield, hal ini dilakukan karenamenurut penelitian yang dilakukan semakin tinggi penempatan kamera maka semakin baik pula objek yang ditangkap, tetapi karena keterbatasan penempatan kamera maka pada posisi di Gambar 5.31 adalah posisi paling ideal untuk menangkap objek. Jika penempatan kamera berada di windshield maka efek getaran dari mesin kendaraan akan banyak diserap oleh windshield,karena bahan pembuat windshield adalah plastik ABS yang dimana memiliki sifat lentur sehingga getaran mesin akan banyak diserap oleh windshield, sehingga video yang ditangkap oleh kamera akan banyak memiliki noise getaran yang berasal dari getaran mesin. Ketinggian penempatan kamera adalah 37 inci yang dimana diukur tegak lurus dari tanah serta sudut kamera dari bawah ke atas 110 derajat seperti pada Gambar 5.31.

5.3.2 Implementasi Perangkat Lunak

Implementasi perangkat luanak meliputi ektraksi video, pengolahan citra, deteksi marka garis lurus dan jarak,dan klasifikasi. Hasil dari perancangan perangkat lunak yaitu dalam bentuk tabelkode progam beserta penjelasanya.

5.3.2.1 Ektraksi Video

Tabel 5.12 Kode Progam Ekstraksi Video

No	Kode Progam

```
1.  def readVideo(path):
2.      cap = cv2.VideoCapture(path)
3.      if (cap.isOpened() == False):
4.          print("Error opening video stream or
    file")
5.      while (cap.isOpened()):
            ret, frame = cap.read()
            if ret == True:
                height, width, channels = frame.shape
6.              print ("----------------------------
    ----------------------------------------")
```

Penjelasan dari Tabel 5.12 dapat dilihat pada Tabel 5.13.

Tabel 5.13Penjelasan Kode Progam Ekstraksi Video

No	Kode Progam
1.	deklarasi fungsi pemutaran video
2.	variabel Cap dilakukan proses pemutaran video dengan fungsi cv2.VideoCapture
3.	jika variabel cap dibuka kosong maka
4.	print sesuai dalam kurung
5.	ketika variabel cap dibuka dan terdapat file video maka akan dilakukan grap frame yang dimasukan dalam variabel frame dengan resolusi sama seperti video yang diputar
6.	print sesuai dalam kurungs

5.3.2.2 Pengolahan Citra

A. Resize

Tabel 5.14 Kode Progam *Resize*

No	Kode Progam
1.	`def resizeForce(image, width=None, height=None, inter=cv2.INTER_AREA):`
2.	`    (h, w) = image.shape[:2]`
3.	`    if width is None and height is None:`
4.	`        return image`
5.	`    resized = cv2.resize(image, (width, height), interpolation=inter)`
6.	`    return resized`
	`resized_image = resizeForce(frame, width=800, height=600, inter=cv2.INTER_AREA);`

Penjelasan dari Tabel 5.14 dapat dilihat pada Tabel 5.15.

Tabel 5.15 Penjelasan Kode Progam Resize

No	Kode Progam

1.	Deklarasi fungsi Resize beserta parameternya
2.	Jika nilai width dan height kosong maka
3.	Ambil gambar selanjutnya
4.	Deklarasi variabel resized yag berisikan fungsi cv2.resize beserta parameternya
5.	Kembali pada fungsi
6.	Pemangginlan fungsi yang telah dbuat pada variabel resize_image dengan mengisika parameternya nilai width 800 height 600 serta memasukan teknik nya dengan nama cv2.INTER_AREA.

B. Average Filter

Tabel 5.16 Kode Progam *Average Filter*

No	Kode Progam
1.	`blur = cv2.blur(gray_image, (3, 3))`

Penjelasan dari Tabel 5.16 dapat dilihat pada Tabel 5.17.

Tabel 5.17 Penjelasan Kode Progam Average Filter

No	Kode Progam
1.	Deklarasi variabel blur yang diisikan oleh fungsi filter dengan nama cv2.blur dengan parameter nya yang berisikan kernel dengan ukuran 3x3

c. Konversi RGB ke *Grayscale*

Tabel 5.18 Kode Progam Konversi RGBke *Grayscale*

No	Kode Progam
1.	`gray_image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)`

Penjelasan dari Tabel 5.18 dapat dilihat pada Tabel 5.19.

Tabel 5.19Penjelasan Kode Progam Konversi RGB ke *Grayscale*

No	Kode Progam
1.	Deklarasi variabel gray_image dengan fungsi cv2.cvtColor beserta parameternya dengan teknik konversi cv2.COLOR_RGB2GRAY

d. Canny

Tabel 5.20 Kode Progam *Canny*

No	Kode Progam
1.	`cannyed_image = cv2.Canny(gray_image, 50, 200)`

Penjelasan dari Tabel 5.220 dapat dilihat pada Tabel 5.21.

Tabel 5.21Penjelasan Kode Progam Canny

No	Kode Progam
1.	Deklarasi variabel cannyed_image dengan fungsi deteksi tepi canny yaitu cv2.Canny beserta variabel min ambang sebesar 50 dan nilai max ambang 200.

5.3.2.3 Deteksi Garis Lurus dan Deteksi Jarak

A. ROI(*Region of Interest*)

Tabel 5.22 Kode Progam *ROI*

No	Kode Progam
1.	`def region_of_interest(img, vertices):`
2.	`    mask = np.zeros_like(img)`
3.	`    match_mask_color = 255`
4.	`    cv2.fillPoly(mask, vertices, match_mask_color)`
5.	`    masked_image = cv2.bitwise_and(img, mask)`
6.	`    return masked_image`
7.	`region_of_interest_vertices = [`
8.	`(0, height),`
9.	`(0, height * 0.7),`
10	`(width * 0.5, height * 0.4),`
11.	`(width, height * 0.7),`
12.	`(width, height),`
13.	`cropped_image = region_of_interest( cannyed_image, np.array([region_of_interest_vertices], np.int32),`

Penjelasan dari Tabel 5.22 dapat dilihat pada Tabel 5.23.

Tabel 5.23 Penjelasan Kode Progam ROI

No	Kode Progam
1.	deklarasi fungsi region_of_interest beserta parameternya yaitu data gambar yang akan diolah dan kordinatnya
2.	deklarasi variabel mask yang berisikan gambar kosong dengan resolusi sama dengan data input melalui proses np.zero_like
3.	memberika warna mask dengan nilai 255 yang ber arti berwarna hitam
4.	fungsi untuk ngambar bidang penutup dengan parameter titik kordinat gambar, serta warna masknya
5.	fungsi untuk menumpuk gambar yang telah digambar pada fungsi
6.	sebelumnya dengan gambar asli. Dengan teknik gambar asli berada dibawah gambar bidang penutup

7.	Deklarasi fungs yang berisikan kordinat gambar penutup
8.	Hasil konversi Kordinat 1 (Width,Height) = (0,600) ke dalam code progam
9.	Hasil konversi Kordinat 2 (Width,Height) = (0,420) ke dalam code progam
10.	Hasil konversi Kordinat 3 (Width,Height) = (400,240) ke dalam code progam
11.	Hasil konversi Kordinat 4 (Width,Height) = (800,420) ke dalam kode progam
12.	Hasil konversi Kordinat 5 (Width,Height) = (800,600) ke dalam kode progam
13.	Fungsi yang telah deklarasikan di panggil ke dalam variabel cropped_image beserta parameter yang dibutuhkan.

B. Probabalistic Houghline Transform

Tabel 5.24 Kode Progam *Probabalistic Houghline Transform*

No	Kode Progam
1.	`lines = cv2.HoughLinesP(`
2.	`cropped_image,`
3.	`rho=5,`
4.	`theta=np.pi / 60,`
5.	`threshold=90,`
6.	`lines=np.array([]),`
7.	`minLineLength=30,`
8.	`maxLineGap=900)`
9.	`if lines is None:`
10.	`cv2.imshow('Video Frame', resized_image)`

Penjelasan dari Tabel 5.24 dapat dilihat pada Tabel 5.25.

Tabel 5.25 Penjelasan Kode Progam *Probabalistic Houghline Transform*

No	Kode Progam
1.	deklarasi variabel lines dengan fungsi deteksi garis cv2.HouglinesP
2.	data gambar yang akan di proses
3.	nilai *rho*, nilai dari jarak piksel ke (0,0) di berisi nilai 5
4.	nilai *theta*, nilai dari derajat bukaan garis dengan nilai 3 derajat hasil dari pembagian np.i/60
5.	Nilai Threshold berisi 90
6.	variabel Line memiliki data dalam bentuk *array*
7.	panjang minimum garis yang dimunculkan berisi nilai 30
8.	panjang maksimal gap garis yang dimungculkan berisi nilai 900
9.	jika garis tidak terdeteksi
10.	tampilkan data resize_image dengan nama Video Frame

C. Deteksi Marka

Tabel 5.26 Kode Progam Deteksi Marka

No	Kode Progam
1.	`def draw_linesv2(img, lines, color=[255, 0, 0], thickness=3):`
	`    # If there are no lines to draw, exit.`
2.	`    if lines is None:`
3.	`        return`
4.	`    img = np.copy(img)`
5.	`    line_img = np.zeros(`
	`        (img.shape[0],img.shape[1],3), dtype=np.uint8,)`
6.	`for line in lines:`
7.	`        for x1, y1, x2, y2 in line:`
8.	`            cv2.line(line_img, (x1, y1), (x2, y2), color, thickness)`
9.	`    img = cv2.addWeighted(img, 0.8, line_img, 1.0, 0.0)`
10.	`    return img`
11.	`line_image = draw_linesv2(image, lines)`
12.	`            x_ =[]`
13.	`            y_ =[]`
14.	`            idex = 0;`
15.	`            xMIn = 10000;`
16.	`            yMIn = 10000;`
17.	`            xMax = -10000;`
18.	`            yMax = -10000;`
19.	`            for line in lines:`
20.	`                for x1, y1, x2, y2 in line:`
21.	`                    if (y2 <(yMIn)):`
22.	`                        yMIn = y2`
23.	`                        xMIn = x2`
24.	`                    if (y1 <(yMIn)):`
25.	`                        yMIn = y1`
26.	`                        xMIn = x1`
27.	`                    if (y1>( yMax)):`
28.	`                        yMax = y1`
29.	`                        xMax = x1`
30.	`                    if (y2>( yMax)):`
31.	`                        yMax = y2`
32.	`                        xMax = x2`
33.	`                    slope = (y2 - y1) / (x2 - x1)`
34.	`                    x_.append(x1)`
35.	`                    x_.append(x2)`
36.	`                    y_.append(y1)`
37.	`                    y_.append(y2)`

```
                        idex = idex+1
                        # cv2.line(line_image, (x1,
y1), (x2, y2), [255, 255, 255], 3)
                        cv2.circle(line_image, (x1,
y1), 3, [255, 0, 0], -1)
                        cv2.circle(line_image, (x2,
y2), 3, [0, 255, 0], -1)
                panjangGaris = math.hypot(xMax -
xMIn, yMax - yMIn)
                print ("Panjang Garis :
{}".format(panjangGaris)  )
status = "Putus-Putus"
                if (panjangGaris > 340):
                    status = "Utuh"
```

Penjelasan dari Tabel 5.26 dapat dilihat pada Tabel 5.27.

Tabel 5.27 Penjelasan Kode Progam Deteksi Marka

No	Kode Progam
1.	deklarasi fungsi draw_linesv2 beserta parameternya
2.	jika garis tidak ada maka
3.	Kembali ke fungsi
4.	deklarasi variabel yang diberi data dari hail copy gambar asli
5.	deklarasi variabel yang diberi gambar kosong dengan resolusi sama dengan asli
6.	perulangan garis pada data lines
7.	jika di garis memiliki data yang disebutkan maka
8.	tungsi cv2.line akan dijalankan dengan parameter yang telah di isikan
9.	tahap selanjutnya menjalankan fungsi cv2.addWeighted beserta parameter yang telah diisikan
10.	Kembali ke variabel img
11.	fungsi draw_linev2 dipanggil pada variabel line_image besera parameternya
12.	variabel x_= null
13.	variabel y_= null
14.	nilai index = 0
15.	nilai xMin =10000
16.	nilai yMin = 10000
17.	nilai xMax = -10000
18.	nilai yMax = -10000
19.	Perulangan garis pada data lines
20.	Jika dalam garis terdapat data yang dalam fungsi perulanagan maka
21.	Jika y2 < yMin, maka
22.	yMin = y2
23.	xMin = x2
24.	Jika y1 < yMin, maka

25.	yMin = y1
26.	xMin = x1
27.	Jika y1> yMax, maka
28.	yMax = y1
29.	xMax = x1
30.	Jika y2 > yMax, maka
31.	yMax = y2
32.	xMax = x2
33.	nilai slope didapat dari pembagian (y2-y1) dan (x2-x1)
34.	x1= x1
35.	x2=x2
36.	y1=y1
37.	y2=y2
38.	idex bernilai idex+1
39.	perintah mengambar garis dengan fungsi cv2.line sesuai parameter yang diisikan
40.	perintah mengambar bidang bulat dengan fungsi cv2.cirle sesuai parameter yang diisikan
41.	perintah mengambar bidang bulat dengan fungsi cv2.cirle sesuai parameter yang diisikan
42.	deklarasi variabel panjangGaris dengan berisikan fungsi matematika *math.hypot* dengan nilai dari (xMax – Xmin) sebagai nilai dasar dan (yMax- yMin) sebagai nilai tinggi
43.	perintah mencetak nilai dari Panjang garis dengan nama Panjang Garis
44.	set statuss dengan “Putus-Putus”
45.	jika panjangGaris lebih besar dari 340, maka
46.	statuss menjadi “Utuh”

D. Deteksi Jarak

Tabel 5.28 Kode Progam *Deteksi Jarak*

No	Kode Progam
1.	`def draw_lines(height, width, img, lines, color=[255, 255, 255], thickness=5, statuss=""):`
2.	`    if lines is None:`
3.	`        return`
4.	`    img = np.copy(img)`
	`    line_imgnp.zeros(img.shape[0],img.shape[1], 3),dtype=np.uint8,)`
5.	`    cek = 0`
6.	`    for line in lines:`
7.	`        for x1, y1, x2, y2 in line:`
8.	`            cek += x1;`
9.	`    img = cv2.addWeighted(img, 0.8, line_img, 1.0, 0.0)`

```
10.     lights.off()
11.     buzzer.off()
12. jarak =  x1 - (width/2)
```

Penjelasan dari Tabel 5.28 dapat dilihat pada Tabel 5.29.

Tabel 5.29 Penjelasan Kode Progam Deteksi Jarak

No	Kode Progam
1.	Deklarasi fungsi draw_lines beserta parameter yang dibutuhkan
2.	Jika garis tidak ditemukan
3.	Kembali pada fungsi
4.	Variabel img deiberikan gambar kosong dengan resolusi sama seperti asli
5.	Cek diberi nilai 0
6.	Fungsi perulangan berdasr pad garis
7.	Jika garis memiliki parameter yang disebutkan maka
8.	Cek diberi nilai + x1
9.	Proses lajutan dengan menjalankan fungsicv2.addWeighted beserta parameter yang telah diisikan
10.	Perintah off pada parangkat *LED*
11.	Perintah off pada perangkat *buzzer*
12.	Variabel jarak memiliki nilai dari pegurangan antara x1 posisi garis dan width/2 width center point.

5.3.2.4 Penentuan Status Marka dan Status Jarak

Tabel 5.30 Kode Progam Penentu Status Marka dan Status Jarak

No	Kode Progam
1.	`If(x1 > width/2) :`
2.	`print ("{}{}".format("Jarak Dari Garis : ",jarak))`
3.	`if jarak<=100:`
4.	`lights.red.on()`
5.	`sleep(1)`
6.	`print ("{}{}".format("Status : ", "AWAS"))`
7.	`elif jarak<=290:`
8.	`lights.amber.on()`
9.	`sleep(1)`
10.	`print ("{}{}".format("Status : ", "HATI HATI"))`
11.	`else:`
12.	`lights.green.on()`
13.	`sleep(1)`
14.	`print ("{}{}".format("Status : ", "AMAN"))`
15.	`else:`

```
16. print ("{}{}".format("Jarak Dari Garis :
17. ",jarak))
        lights.red.on()
18.     if statuss=="Utuh":eria
19.   buzzer.on()
20. sleep(1)
21. print ("{}{}".format("Status : ",
    "BAHAYA"))
```

Penjelasan dari Tabel 5.30 dapat dilihat pada Tabel 5.31.

Tabel 5.31 Penjelasan Kode Progam Penentu Status Marka dan Status Jarak

No	Kode Progam
1.	Jika nilai x1 lebih besar dari width/2. Maka
2.	cetak nilai jarak pada setelah " Jarak dari Garis :"
3.	Serta jika jarak kurang dari sama dengan 100, maka
4.	lampu LED Merah ON
5.	set sleep(1ms)
6.	cetak status "AWAS"
7.	jika jarak kurang dari sama dengan 290, maka
8.	lampu LED Kuning ON
9.	set sleep(1ms)
10.	cetak status"HATI – HATI"
11.	jika jarak lebih dari 290, maka
12.	lampu LED Hijau ON
13.	set sleep(1ms)
14.	cetak status"AMAN"
15.	jika tidak, maka
16.	cetak nilai jarak pada setelah " Jarak dari Garis :"
17.	lampu LED Merah ON
18.	set sleep(1ms)
19.	jika statuss sama dengan"Utuh", maka
20.	nyalakan *buzzer*
21.	cetak status "BAHAYA"

BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian Fungsional Sistem

6.1.1 Tujuan Pengujian

Pengujian fungsional sistem bertujuan untuk menguji bagaimana performa dari sistem yang dibuat untuk menampilkan informasi kepada pengguna. Sehingga nantinya sistem dapat produktif jika digunakan pada situasi yang sebenarnya.

6.1.2 Pelaksanaan Pengujian

Papan notifikasi yang terdiri dari rangkaian *LED* serta *buzzer* tersambung dengan pin BCM GPIO Raspberry pi 3. Dengan konfigurasi LED merah tersambung pada pin BCM(17),LED kuning tesambung pada pin BCM(27),LED hijau tersambung pada pin BCM(22),dan *buzzer* tersambung pada pin BCM(4).

6.1.3 Prosedur Pengujian

1. Menghubungkan Raspberry dengan Power bank, LED power Raspberry pi menyala.
2. Menunggu kedipan LED warna kuning yang berada di sebelah led warna merah benar-benar padam.
3. Menghubungkan Raspberry Pi dengan laptop menggunakan kabel LAN (LED yang berada pada konektor LAN berkedip).
4. Mempersiapkan aplikasi VNC dan sudah tersambung dengan raspberry pi dengan media kabel LAN.
5. Menyambungkan Pin papan notifikasi dengan GPIO *Raspberry Pi* dengan menggunakan konfigurasi pin BCM.
6. Buka terminal pada UI Raspberry Pi dan masuk pada repositori tempat progam.
7. Isi foldel Uji_Acak_2 dengan video yang akan dijalankan.
8. Jalankan progam dengan cara memasukan printah pada terminal “Sudo python deteksifix.py”.
9. Sistem akan mulai mendeteksi Marka garis lurus dan deteksi jarak.
10. Memperhatikan output yang di informasikanoleh LED dan buzzer dan bandingkandengan perancangan klasifikasi serta ouput dari terminal

6.1.4 Hasil Pengujian

Tandajika LED benar benar berfungsi dapat dilihat pada Gambar 6.1

Gambar 6.1 Pengujian Fungsional Sistem

Output dari LED dan suara*buzzer* bekerja dengan baik dan sesuai dengan klasifikasi yan dirancang serta sesuai dengan keluaran dari terminal.Setelah melakukan percobaan sebanyak sepuluh kali maka didapat hasil sesuai Tabel 6.10

Tabel 6.1 Tabel Pengujian Funsional Sistem

Perco-baan	Status	Status Jalan	LED Hijau	LED Kuning	LED Merah	Buzzer	BENAR/ SALAH
1.	AMAN	Utuh	ON	OFF	OFF	OFF	BENAR
2.	AMAN	Putus-putus	ON	OFF	OFF	OFF	BENAR
3.	HATI-HATI	Utuh	OFF	ON	OFF	OFF	BENAR
4.	HATI-HATI	Putus-putus	OFF	ON	OFF	OFF	BENAR
5.	AWAS	Utuh	OFF	OFF	ON	OFF	BENAR
6.	AWAS	Putus-putus	OFF	OFF	ON	OFF	BENAR
7.	BAHAYA	Utuh	OFF	OFF	ON	ON	BENAR
8.	BAHAYA	Putus-putus	OFF	OFF	ON	OFF	BENAR
9.	AMAN	Utuh	ON	OFF	OFF	OFF	BENAR
10.	AMAN	Putus-putus	ON	OFF	OFF	OFF	BENAR

6.1.5 Analisis Hasil Pengujian

Hasil dari percobaan fungsional ini dipastikan bahwa semua komponen berkerja dengan baik. Jika dilihat pada tabel semua fungsi sistem berjalan dengan benar disemua kondisi.

6.2 Pengujian Akurasi Deteksi Marka Membujur Garis Utuh danPutus

6.2.1 Tujuan Pengujian

Pengujian ini memiliki fungsi untuk menguji tingkat akurasi performa sistem untuk mendeteksi marka garis lurus utuh dan marka garis lurus putus-putus.

6.2.2 Pelaksanaan Pengujian

Pengujian dilakukan dengan cara menjalankan progam deteksifix.py yang sudah tersimpan Raspberry Pi3 . *Output*dari sistem berupa frame berjalan yang dari video yang diputarserta tampilan dari status terminal.

6.2.3 Prosedur Pengujian

1. Menghubungkan Raspberry dengan Power bank, LED power Raspberry pi menyala.
2. Menunggu kedipan LED warna kuning yang berada di sebelah led warna merah benar-benar padam.
3. Menghubungkan Raspberry Pi dengan laptop menggunakan kabel LAN (LED yang berada pada konektor LAN berkedip).
4. Mempersiapkan aplikasi VNC dan sudah tersambung dengan raspberry pi
5. Buka terminal pada UI Raspberry Pi dan masuk pada repositori tempat progam.
6. Isi foldel Uji_Acak_2 dengan video yang akan dijalankan.
7. Jalankan progam dengan cara memasukan perintah pada terminal “Sudo python deteksifix.py”..
8. Mengamati dan mencocokan keluaran status terminal dengan frame.

6.2.4 Hasil Pengujian

Marka Garis Lurus Utuh dan Marka Garis Lurus putus-putus dinyatakanterdeteksi, jikatampilann pada frame memiliki hasil yang sama dengan status terminaldari output, serta sesuai dengan nilai panjang garis yang telah ditentukan saat perancangan perangkat lunak. Setiap jenis marka lurus akan diambil data latih sebanyak 20 kali dengan data uji 20 kali.

6.2.4.1 Marka Membujur Garis Utuh

Pengujian ini hanya terfokus pada deteksi marka garis membujur lurus utuh. Nilai Panjang Garis harus sesuai dengan rumus berikut:

$$Panjang\ Garis > 340 \quad (6.1)$$

Jika panjang garis yang muncul pada terminal tidak sesuai dengan rumus 6.1 maka diaggap gagal. Tabel 6.1 hanya menampilkan Frame , Status Terminal dan hasil.

Tabel 6.2 Deteksi Marka MembujurGaris Utuh

Perco-baan	Frame	Status Terminal	Hasil
1.		Panjang Garis : 353.861554849 Status Jalan : Utuh	B
2.		Panjang Garis : 371.57098918 Status Jalan : Utuh	B
3.		Panjang Garis : 375.362491467 Status Jalan : Utuh	B
4.		Panjang Garis : 370.645383082 Status Jalan : Utuh	B
5		Panjang Garis : 329.370611925 Status Jalan : Putus-Putus	G

Keterangan:

B = Berhasil = 100%

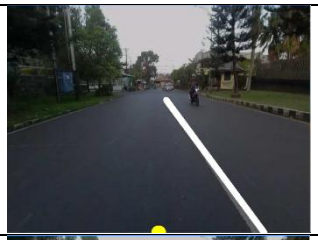
G = Gagal = 0%

Perco-baan	Frame	Status Terminal	Hasil
6.		Panjang Garis : 303.710388364 Status Jalan : Putus-Putus	G
7.		Panjang Garis : 364.444783198 Status Jalan : Utuh	B
8.		Panjang Garis : 373.807437058 Status Jalan : Utuh	B
9.		Panjang Garis : 361.44847489 Status Jalan : Utuh	B
10.		Panjang Garis : 365.0 Status Jalan : Utuh	B
11.		Panjang Garis : 365.978141424 Status Jalan : Utuh	B

Keterangan:

B = Berhasil = 100%

G = Gagal = 0%

Perco -baan	Frame	Status Terminal	Hasil
12.		Panjang Garis : 362.630666105 Status Jalan : Utuh	B
13.		Panjang Garis : 359.288185166 Status Jalan : Utuh	B
14.		Panjang Garis : 366.946862638 Status Jalan : Utuh	B
15.		Panjang Garis : 282.078003396 Status Jalan : Putus-Putus	G
16.		Panjang Garis : 372.404350136 Status Jalan : Utuh	B
17.		Panjang Garis : 372.690219888 Status Jalan : Utuh	B

Keterangan:

B = Berhasil= 100%

G = Gagal= 0%

Perco-baan	Frame	Status Terminal	Hasil
18.		Panjang Garis : 362.836051131 Status Jalan : Utuh	B
19.		Panjang Garis : 355.550277739 Status Jalan : Utuh	B
20.		Panjang Garis : 356.736597506 Status Jalan : Utuh	B

Keterangan:

B = Berhasil = 100%

G = Gagal = 0%

6.2.4.2 Marka Membujur Garis Putus

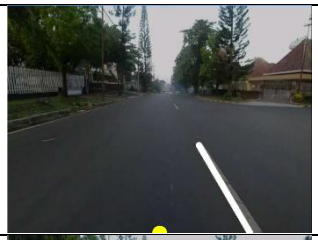
Pengujian ini hanya terfokus pada deteksi marka membujur garisputus-putusNilai Panjang Garis harus sesuai dengan rumus berikut:

$$Panjang\ Garis \leq 340 \quad (6.2)$$

Jika panjang garis yang muncul pada terminal tidak sesuai dengan rumus 6.2 maka diaggap gagal. Tabel 6.2 menampilkan Frame , Status Terminal dan hasil.

Tabel 6.3 Deteksi Marka Membujur Garis Putus

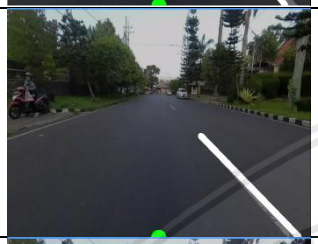
Perco-baan	Frame	Status Terminal	Hasil
1.		Panjang Garis : 220.911747085 Status Jalan : Putus-Putus	B

Perco-baan	Frame	Status Terminal	Hasil
2.		Panjang Garis : 230.846269192 Status Jalan : Putus-Putus	B
3.		Panjang Garis : 339.330222645 Status Jalan : Putus-Putus	B
4.		Panjang Garis : 223.396508478 Status Jalan : Putus-Putus	B
5.		Panjang Garis : 199.729817504 Status Jalan : Putus-Putus	B
6.		Panjang Garis : 337.249166048 Status Jalan : Putus-Putus	B
7.		Panjang Garis : 329.948480827 Status Jalan : Putus-Putus	B

Keterangan:

B = Berhasil = 100%

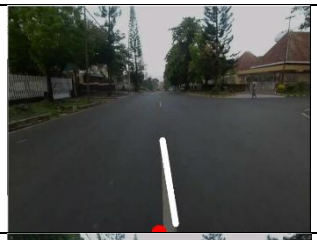
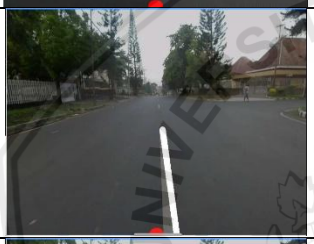
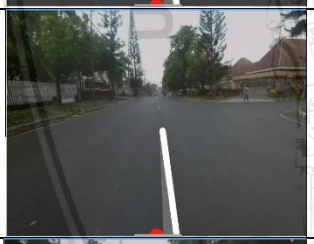
G = Gagal = 0%

Perco-baan	Frame	Status Terminal	Hasil
8.		Panjang Garis : 172.072659072 Status Jalan : Putus-Putus	B
9.		Panjang Garis : 274.052914599 Status Jalan : Putus-Putus	B
10.		Panjang Garis : 360.649414252 Status Jalan : Lurus	G
11.		Panjang Garis : 233.308808235 Status Jalan : Putus-Putus	B
12.		Panjang Garis : 279.427987145 Status Jalan : Putus-Putus	B
13.		Panjang Garis : 378.359617296 Status Jalan : Lurus	G

Keterangan:

B = Berhasil = 100%

G = Gagal = 0%

Perco-baan	Frame	Status Terminal	Hasil
14.		Panjang Garis : 265.3469427 Status Jalan : Putus-Putus	B
15.		Panjang Garis : 233.308808235 Status Jalan : Putus-Putus	B
16.		Panjang Garis : 254.332459588 Status Jalan : Putus-Putus	B
17.		Panjang Garis : 283.213700234 Status Jalan : Putus-Putus	B
18.		Panjang Garis : 264.425792993 Status Jalan : Putus-Putus	B
19.		Panjang Garis : 272.530732212 Status Jalan : Putus-Putus	B

Keterangan:

B = Berhasil = 100%

G = Gagal = 0%

Perco-baan	Frame	Status Terminal	Hasil
20.		Panjang Garis : 238.365265926 Status Jalan : Putus-Putus	B

Keterangan:

B = Berhasil = 100%

G = Gagal = 0%

6.2.4.3 Akurasi Deteksi Marka Membujur Garis Utuh dan Marka Membujur Garis Putus

Dalam mengukur tingkat akurasi dari deteksi garis membutuhkan persamaan rata-rata persentasi dari sistem dapat mendeteksi marka jalan yang dituju. Fungsi persamaan dapat dapat dilihat dibawah.

$$\frac{\text{Jumlah Total Persentasi Pemgujian}}{\text{Total Pengujian}} \quad (6.3)$$

Banyak data uji yang digunakan untuk menentukan tingkat akurasi pengujian sebanyak sepuluh data,untuk lebih detail bisa dilihat pada Tabel 6.3.

Tabel 6.4Akurasi Deteksi Marka Membujur Garis

Perco-baan	Marka Membujur Garis Utuh(%)	Marka Membujur Garis Putus-Putus (%)
1.	100	100
2.	100	100
3.	100	100
4.	100	100
5.	0	100
6.	0	100
7.	100	100
8.	100	100
9.	100	100
10.	100	0
11.	100	100
12.	100	100
13.	100	0
14.	100	100
15.	0	100

16.	100	100
17.	100	100
18.	100	100
19.	100	100
20	100	100
Rata-rata AKURASI	**85**	**90**
AKURASI TOTAL %	$\frac{85+90}{2} = 87,5$	

6.2.5 Analisis Hasil Pengujian

Setelah melakukan pengujian deteksi marka membujurgaris utuh dan marka membujur garis putus-putus maka di peroleh rata rata akurasi deteksi marka membujur garis seperti pada Tabel 6.3. Maka didapatkan sebuah hasil bahwa tingkat akurasi deteksi marka membujur garis utuh dan marka membujur garis putus putus memiliki tingkat akurasi total sebesar**87,5%**. berikut analisis penyebab keberhasilan dan kegagalan sistem dalam mendeteksi marka jalan.

a. Keberhasilan

Saat marka membujur garis terdeteksi maka pada saat itu cahaya yang di tangkap oleh kamera merata di seluruh piksel sehingga sistem tidak mendeteksi adanya simpangan yang jauh antar piksel,sehingga sistem hanya mendapati marka membujur garisterlihat yang terang.

b. Kegagalan

Faktor yang membuat gagalnya sistem dalam mendeteksi marka membujur garis utuh dan marka membujur garis putus-putus adalah adanya bayangan yang timbul dari kontur aspal yang tidak rata, sehingga menimbulkan titik-titik kecil yang dapat mempengaruhi pengukuran panjang marka jalan yang terdeteksi,itulah mengapa pada saatmarka membujur garis putus putus sistem seolah olah menggabungkan dua garis putus-putus menjadi satu garis sehingga sistem dapat mengalami kesalahan pembacaan.

6.3 Pengujian Akurasi deteksi Jarak Antara Kendaraan dengan Marka Membujur garis

6.3.1 Tujuan Pengujian

Tujuan dar pengujian ini adalah untuk mengukur tingkat akurasi sistem untuk mendeteksi jarak kendaraan dengan marka membujur garis jalan yang terdeteksi oleh sistem.

6.3.2 Pelaksanaan Pengujian

Pengujian ini dilakukan dalam empat kategori pengujian, berikut adalah kategorI yang diujikan.Yang pertama KondisiAman,Kondisi Hati-hati, Kondisi

Awas, Kondisi Bahaya.Setiap kategori akan diambil data latih sebanyak 20 kali dan data uji 20 kali.

6.3.3 Prosedur Pengambilan Video

1. Siapkan sepeda motor untuk mengambil video
2. Pasang action cam pada dada motor yang telah ditetapkan
3. Nyalakan acation cam dan lakukan *pairing* dengan smartphone
4. Kendarai kendaraan sesuai jarak kendaraan dengan marka membujur garis menutur keempat kondisi
5. Setelah dirasa tepat nyalakan perekam kamera melalui smartphone
6. Matikan perekaman jika marka membujur garis yang telah habis

6.3.4 Prosedur Pengujian

1. Menghubungkan *raspberry pi* dengan Power bank, LED power Raspberry pi menyala.
2. Menunggu kedipan LED warna kuning yang berada di sebelah led warna merah benar-benar padam.
3. Menghubungkan Raspberry Pi dengan laptop menggunakan kabel LAN (LED yang berada pada konektor LAN berkedip).
4. Hubungkan pinpapan notifikasi dengan *raspberry pi*
5. Mempersiapkan aplikasi VNC dan sudah tersambung dengan raspberry pi
6. Buka terminal pada UI Raspberry Pi dan masuk pada repositori tempat progam.
7. Isi foldel Uji_Acak_2 dengan video yang akan dijalankan.
8. Jalankan progam dengan cara memasukan printah pada terminal “Sudo python deteksifix.py”..
9. Mengamati dan mencocokan keluaran status terminal,frame, serta papan notifikasi.

6.3.5 Hasil Pengujian

Terdeteksinya Jarak antara kendaraan dengan marka jalan yang terdeteksi dinyatakan berhasil, jika apa yang ditampilkan pada frame memiliki kesamaan dengan status terminal serta output dari papan notifikasi yang di berikan oleh sistem, dengan cara memperhatikan ikon gelembung pada frame, hijau menandakan status Aman,Kuning Menandakan pada status Hati-hati, Merah menandakan status Awas dan Bahaya serta nyala lampu yang diberikan oleh papan notifikasi.

6.3.5.1 Status Aman

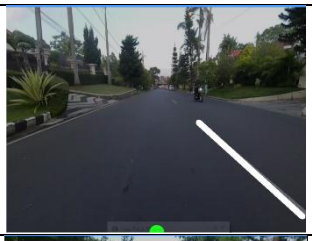
Pengujian ini hanya terfokus pada jarak kendaraan pada Status aman saja. Jarak dari garis harus sesuai dengan rumus berikut

$$Jarak\ Dari\ Garis > 290 \quad (6.4)$$

Jika Jarak dari garis yang muncul pada terminal tidak sesuai dengan rumus 6.4 maka dianggap gagal. Tabel 6.4 menampilkan Frame , Status Terminal dan hasil.

.

Tabel 6.5Status Aman

Perco-baan	Frame	Status Terminal	Hasil
1.		Jarak Dari Garis : 398 Status : AMAN	B
2.		Jarak Dari Garis : 353 Status : AMAN	B
3.		Jarak Dari Garis : 374 Status : AMAN	B
4.		Jarak Dari Garis : 337 Status : AMAN	B
5.		Jarak Dari Garis : 397 Status : AMAN	B
6.		Jarak Dari Garis : 382 Status : AMAN	B

Keterangan:

B = Berhasil = 100%

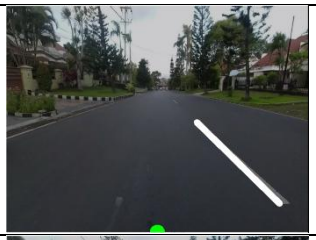
G = Gagal = 0%

Perco-baan	Frame	Status Terminal	Hasil
7.		Jarak Dari Garis : 394 Status : AMAN	B
8.		Jarak Dari Garis : 396 Status : AMAN	B
9.		Jarak Dari Garis : 285 Status : HATI HATI	G
10.		Jarak Dari Garis : 397 Status : AMAN	B
11.		Jarak Dari Garis : 312 Status : AMAN	B
12.		Jarak Dari Garis : 393 Status : AMAN	B

Keterangan:

B = Berhasil = 100%

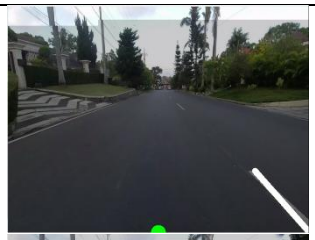
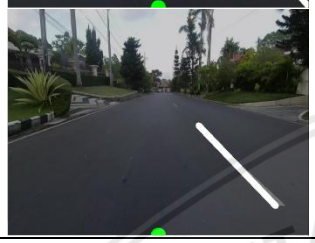
G = Gagal = 0%

Perco-baan	Frame	Status Terminal	Hasil
13.		Jarak Dari Garis : 399 Status : AMAN	B
14.		Jarak Dari Garis : 398 Status : AMAN	B
15.		Jarak Dari Garis : 324 Status : AMAN	B
16.		Jarak Dari Garis : 389 Status : AMAN	B
17.		Jarak Dari Garis : 248 Status : HATI HATI	G
18.		Jarak Dari Garis : 398 Status : AMAN	B

Keterangan:

B = Berhasil = 100%

G = Gagal = 0%

Perco-baan	Frame	Status Terminal	Hasil
19.		Jarak Dari Garis : 396 Status : AMAN	B
20.		Jarak Dari Garis : 396 Status : AMAN	B

Keterangan:

B = Berhasil = 100%

G = Gagal = 0%

6.3.5.2 Status Hati Hati

Pengujian ini hanya terfokus pada jarak kendaraan pada status hati -hati saja.

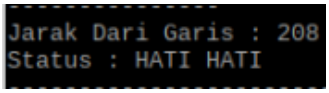
Jarak dari garis harus sesuai dengan rumus berikut

$$Jarak\ Dari\ Garis \leq 290 \quad (6.5)$$

Jika Jarak dari garis yang muncul pada terminal tidak sesuai dengan rumus 6.5 maka dianggap gagal. Tabel 6.5 menampilkan Frame , Status Terminal dan hasil.

Tabel 6.6StatusHati-Hati

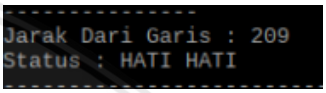
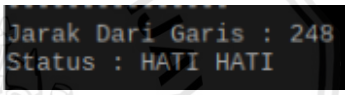
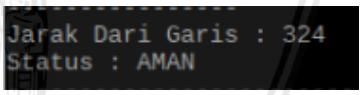
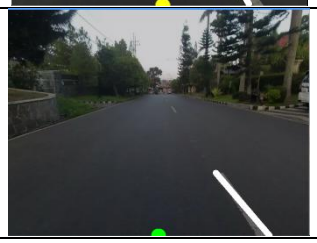
Perco-baan	Frame	Status Terminal	Hasil
1.		Jarak Dari Garis : 242 Status : HATI HATI	B

2.			B

Keterangan:

B = Berhasil= 100%

G = Gagal= 0%

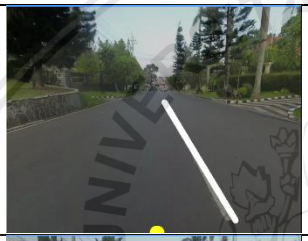
Perco-baan	Frame	Status Terminal	Hasil
3.			B
4.			B
5.			G
6.			B
7.			G

8.		---------- Jarak Dari Garis : 213 Status : HATI HATI	B

Keterangan:

B = Berhasil= 100%

G = Gagal= 0%

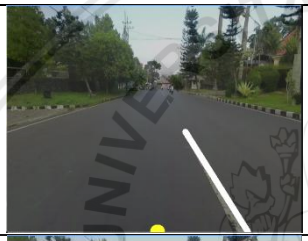
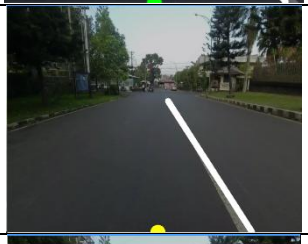
Perco-baan	Frame	Status Terminal	Hasil
9.		---------- Jarak Dari Garis : 250 Status : HATI HATI	B
10.		---------- Jarak Dari Garis : 214 Status : HATI HATI	B
11.		---------- Jarak Dari Garis : 244 Status : HATI HATI	B
12.		---------- Jarak Dari Garis : 246 Status : HATI HATI	B
13.		---------- Jarak Dari Garis : 222 Status : HATI HATI	B

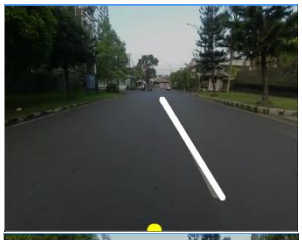
14.		---------- Jarak Dari Garis : 258 Status : HATI HATI	B

Keterangan:

B = Berhasil= 100%

G = Gagal= 0%

Perco-baan	Frame	Status Terminal	Hasil
15.		---------- Jarak Dari Garis : 219 Status : HATI HATI ----------	B
16.		---------- Jarak Dari Garis : 252 Status : HATI HATI ----------	B
17.		---------- Jarak Dari Garis : 312 Status : AMAN ----------	G
18.		---------- Jarak Dari Garis : 220 Status : HATI HATI ----------	B
18.		---------- Jarak Dari Garis : 217 Status : HATI HATI	B

19.		---------- Jarak Dari Garis : 252 Status : HATI HATI ----------	B
20.		---------- Jarak Dari Garis : 223 Status : HATI HATI ----------	B

Keterangan:

B = Berhasil= 100%

G = Gagal= 0%

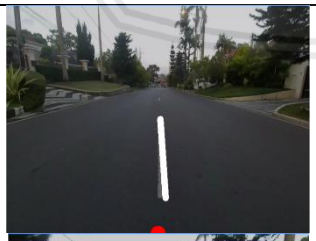
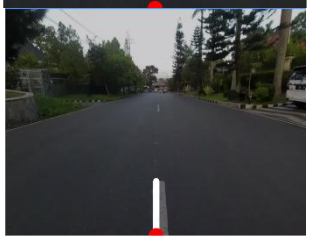
6.3.5.3 Status Awas

Pengujian ini hanya terfokus pada jarak kendaraan pada status awas saja. Jarak dari garis harus sesuai dengan rumus berikut

$$Jarak\ Dari\ Garis \leq 100 \quad (6.6)$$

Jika Jarak dari garis yang muncul pada terminal tidak sesuai dengan rumus 6.6 maka dianggap gagal. Tabel 6.6 menampilkan Frame , Status Terminal dan hasil.

Tabel 6.7Status Awas

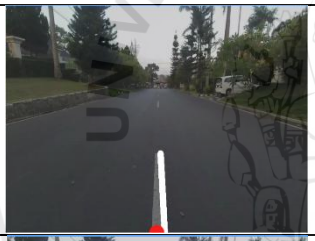
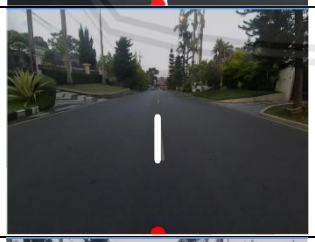
Perco-baan	Frame	Status Terminal	Hasil
1.		---------- Jarak Dari Garis : 42 Status : AWAS ----------	B
2.		---------- Jarak Dari Garis : 81 Status : AWAS ----------	B

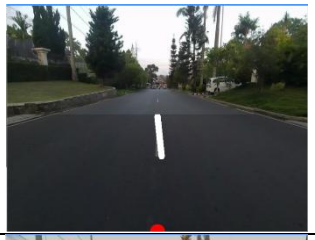
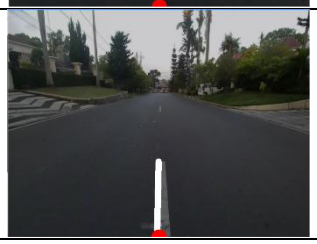
3.		Jarak Dari Garis : 55 Status : AWAS	B

Keterangan:

B = Berhasil = 100%

G = Gagal= 0%

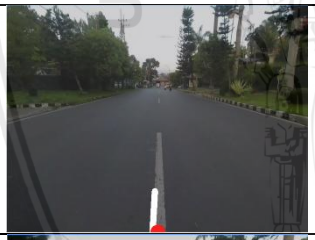
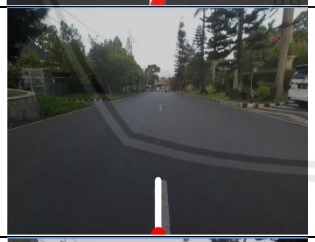
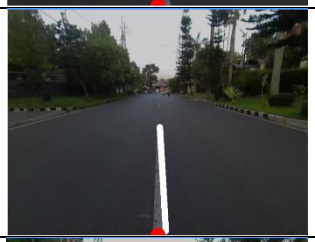
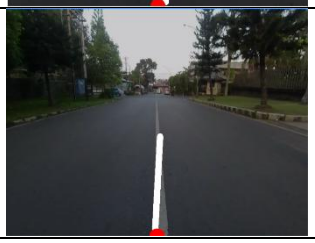
Perco-baan	Frame	Status Terminal	Hasil
4.		Jarak Dari Garis : 65 Status : AWAS	B
5.		Jarak Dari Garis : 32 Status : AWAS	B
6.		Jarak Dari Garis : -2 Status : BAHAYA	G
7.		Jarak Dari Garis : 72 Status : AWAS	B

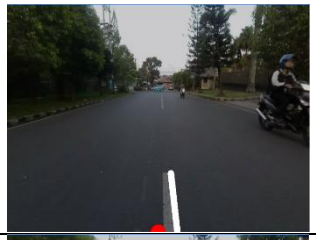
8.		Jarak Dari Garis : 74 Status : AWAS	B
9.		Jarak Dari Garis : 76 Status : AWAS	B

Keterangan:

B = Berhasil = 100%

G = Gagal= 0%

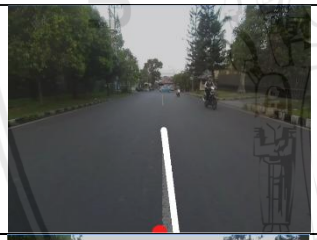
Perco-baan	Frame	Status Terminal	Hasil
10.		Jarak Dari Garis : 23 Status : AWAS	B
11.		Jarak Dari Garis : 74 Status : AWAS	B
12.		Jarak Dari Garis : 17 Status : AWAS	B
13.		Jarak Dari Garis : 14 Status : AWAS	B

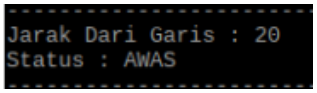
14.		Jarak Dari Garis : 11 Status : AWAS	B
15.		Jarak Dari Garis : 25 Status : AWAS	B

Keterangan:

B = Berhasil = 100%

G = Gagal= 0%

Perco-baan	Frame	Status Terminal	Hasil
16.		Jarak Dari Garis : 26 Status : AWAS	B
17.		Jarak Dari Garis : 59 Status : AWAS	B
18.		Jarak Dari Garis : 63 Status : AWAS	B
19.		Jarak Dari Garis : 65 Status : AWAS	B

20.			B

Keterangan:

B = Berhasil = 100%

G = Gagal= 0%

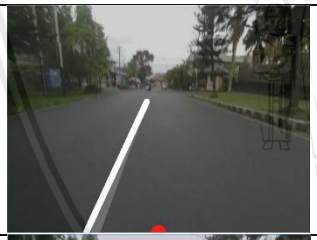
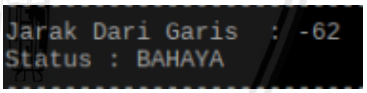
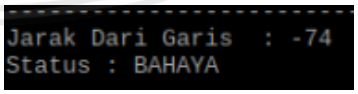
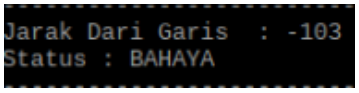
6.3.5.4 Status Bahaya

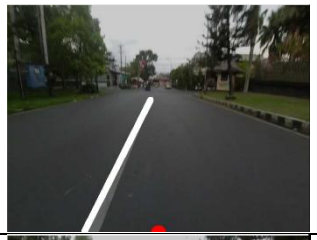
Pengujian ini hanya terfokus pada jarak kendaraan pada status Bahaya saja. dari garis harus sesuai dengan rumus berikut:

$$Jarak\ Dari\ Garis < 0 \quad (6.7)$$

Jika Jarak dari garis yang muncul pada terminal tidak sesuai dengan rumus 6.7 maka dianggap gagal. Tabel 6.7 menampilkan Frame , Status Terminal dan hasil.

Tabel 6.8Status Bahaya

Perco-baan	Frame	Status Terminal	Hasil
1.			B
2.			B
3.			B

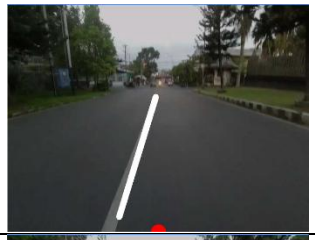
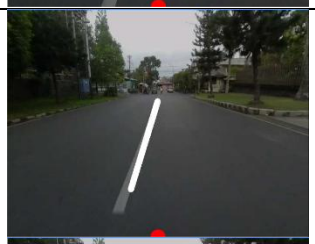
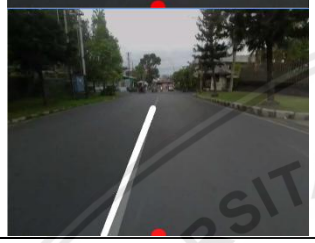
4.		Jarak Dari Garis : -60 Status : BAHAYA	B
5.		Jarak Dari Garis : -175 Status : BAHAYA	B
6.		Jarak Dari Garis : -131 Status : BAHAYA	B

Keterangan:

B = Berhasil = 100%

G = Gagal= 0%

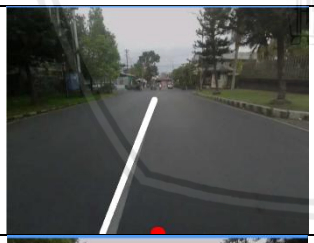
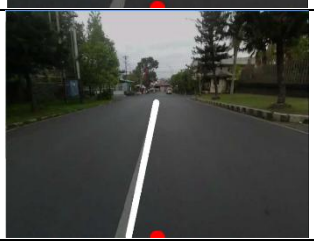
Perco-baan	Frame	Status Terminal	Hasil
7.		Jarak Dari Garis : -78 Status : BAHAYA	B
8.		Jarak Dari Garis : -124 Status : BAHAYA	B
9.		Jarak Dari Garis : -84 Status : BAHAYA	B

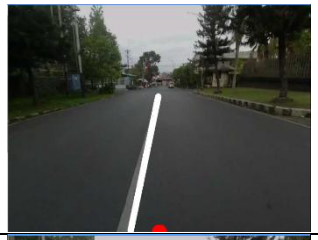
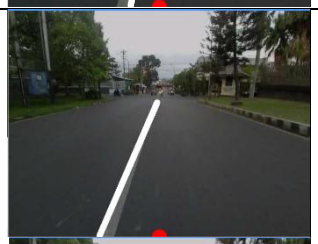
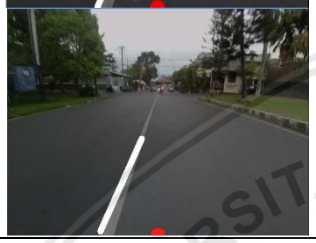
10.		Jarak Dari Garis : -81 Status : BAHAYA	B
11.		Jarak Dari Garis : -79 Status : BAHAYA	B
12.		Jarak Dari Garis : -76 Status : BAHAYA	B

Keterangan:

B = Berhasil = 100%

G = Gagal= 0%

Perco-baan	Frame	Status Terminal	Hasil
13.		Jarak Dari Garis : -75 Status : BAHAYA	B
14.		Jarak Dari Garis : -109 Status : BAHAYA	B
15.		Jarak Dari Garis : -108 Status : BAHAYA	B

16.		Jarak Dari Garis : -62 Status : BAHAYA	B
17.		Jarak Dari Garis : -74 Status : BAHAYA	B
18.		Jarak Dari Garis : -103 Status : BAHAYA	B

Keterangan:

B = Berhasil = 100%

G = Gagal= 0%

Perco-baan	Frame	Status Terminal	Hasil
19.		Jarak Dari Garis : -73 Status : BAHAYA	B
20.		Jarak Dari Garis : -51 Status : BAHAYA	B

Keterangan:

B = Berhasil = 100%

G = Gagal= 0%

6.3.5.5 Akurasi Deteksi Jarak Kendaraan dengan Marka Membujur Garis

Dalam mengukur tingkat akurasi dari deteksi garis membutuhkan persamaan rata-rata persentasi dari sistem dapat mendeteksi marka jalan yang dituju. Fungsi persamaan dapat dapat dilihat dibawah.

$$\frac{\text{Jumlah Total Persentasi Pemgujian}}{\text{Total Pengujian}} \quad (6.8)$$

Banyak data uji yang digunakan untuk menentukan tingkat akurasi pengujian sebanyak sepuluh data,untuk lebih detail bisa dilihat pada Tabel 6.8

Tabel 6.9Akurasi Deteksi Jarak dari Marka Membujur Garis

Perco-baan	Status Aman (%)	Status Hati-hati (%)	Status Awas (%)	Status Bahaya (%)
1.	100	100	100	100
2.	100	100	100	100
3.	100	100	100	100
4.	100	100	100	100
5.	100	0	0	100
6.	100	100	100	100
7.	100	0	100	100
8.	100	100	100	100
9.	0	100	100	100
10.	100	100	100	100
11	100	100	100	100
12.	100	100	100	100
13.	100	100	100	100
14.	100	100	100	100
15.	100	100	100	100
16.	100	100	100	100
17.	100	0	100	100
18.	100	100	100	100
19.	100	100	100	100
20.	100	100	100	100
Rata-rata Akurasi%	**95**	**85**	**95**	**100**
Akurasi Total %	$\frac{95+85+95+100}{4} = 93,75$			

6.3.6 Analisis Hasil Pengujian

Setelah didapat data tingkat akurasi deteksi jarak dengan marka membujur garis seperti pada Tabel 6.5. Maka didapatkan sebuah hasil bahwa tingkat akurasi deteksi jarak dengan marka membujur garis memiliki tingkat akurasi total sebesar **93,75%**. berikut analisis penyebab keberhasilan dan kegagalan sistem dalam mendeteksi jarak dengan marka membujur garis.

a. Keberhasilan
   Saat melakukan pengambilan video, kendaraan pengambil video berada pada posisi konstan dan stabi, hal ini dikarenakan saat pengambilan video tidak ada terpaan angin maupun goncangan dari kontur jalan yang membuat pengambilan video stabil, sehingga sangat mempengaruhi tingkat deteksi jarak dengan marka garis lurus.

b. Kegagalan
   Beberapa faktor yang menyebabkan sistem gagal mendeteksi jarak kendaraan dengan marka garis lurus salah satunya yaitu adanya terpaan angin yang mendadak saat pengambilan video, serta faktor kontur jalan juga berpengaruh dalam mendeteksi jarak.

6.4 Pengujian Waktu Deteksi

6.4.1 Tujuan Pengujian

Perhitungan waktu proses deteksi bertujuan untuk mengetahui rata-rata waktu yang dibutuhkan untuk mendeteksi marka garis lurus beserta deteksi jarak. Jumlah waktu dapat dijadikan acuan seberapa baik performas sistem dipandang dari segi waktu

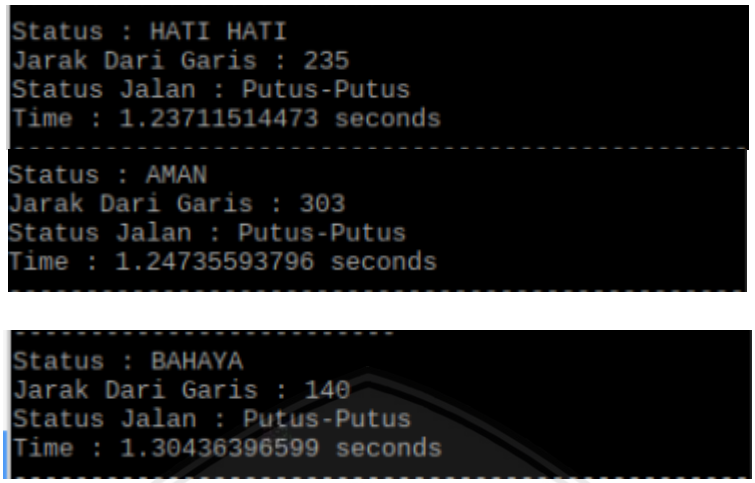
6.4.2 Pelaksanaan Pengujian

6.4.3 Prosedur Pengujian

1. Menghubungkan *raspberry pi* dengan Power bank, LED power Raspberry pi menyala.
2. Menunggu kedipan LED warna kuning yang berada di sebelah led warna merah benar-benar padam.
3. Menghubungkan Raspberry Pi dengan laptop menggunakan kabel LAN (LED yang berada pada konektor LAN berkedip).
4. Mempersiapkan aplikasi VNC dan sudah tersambung dengan raspberry pi
5. Buka terminal pada UI Raspberry Pi dan masuk pada repositori tempat progam.
6. Tambahkan baris progam diawal proses “start_time=time.time”
7. Tambahkan baris progam diakhir pencetak status “print("Time : %s seconds" % (time.time() - start_time))”
8. Jalankan progam dengan cara memasukan printah pada terminal “Sudo python deteksifix.py”..
9.

6.4.4 Hasil Pengujian

Berikut hasil perhitungan waktu yang dibutuhkan sistem dakam mendeteksi marka garis lurus beserta deteksi jaraknya seperti pada Gambar 6.1.



Gambar 6.2 Waktu Deteksi Sistem

6.4.5 Analisis Hasil Pengujian

Setelah mendapatkan waktu deteksi dari keempat status yaitu aman,hati-hati,awas,bahaya maka dapat diperoleh waktu deteksi dari tiap tiap status seperti apada Tabel 6.9.

Tabel 6.10 Waktu Deteksi Sistem

Perco-baan	Status Aman beserta Status Marka (detik)	Status Hati - Hati beserta Status Marka (detik)	Status Awas beserta Status Marka (detik)	Status Bahaya beserta Status Marka (detik)
1.	1,31	1,27	1,28	1,27
2.	1,27	1,25	1,25	1,30
3.	1,24	1,30	1,29	1,28
4.	1,30	1,26	1.30	1,29
5.	1,23	1,25	1,29	1,30
6.	1,25	1,28	1,27	1,25
7.	1,29	1,27	1,26	1,28
8.	1,21	1,25	1,25	1,25
9.	1,23	1,28	1,28	1,29
10.	1,25	1,31	1,29	1.30
Rata-Rata Waktu	**1,25**	**1,27**	**1,27**	**1,28**

Rata – Rata Waktu Total (detik)	$\frac{1,25 + 1,27 + 1,27 + 1,28}{4} = 1,26$

Setelah didapat data waktu deteksi sistem seperti pada Tabel 6.9. Maka didapatkan sebuah hasil bahwa waktu yang dibutuhkan sistem dalam mendetksi marka membujur garis dan jarak dari marka membujur garis rata rata berkisar **1,26 detik** tiap kali melakukan proses.

BAB 7 PENUTUP

Pada bab ini berisikan kesimpulan yang ditarik sesuai dengan hasil pada penelitian dan saran perkembangan yang bisa dikembangkan dari penelitian ini.

7.1 Kesimpulan

Berdasarkan hasil dari penelitian, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Dalam merancang sistem dibutuhkan kebutuhan perangkat keras, kebutuhan perangkat lunak, serta perancangan perangkat keras dan perancangan perangkat lunak yang matang dari awal, dengan cara memastianya pada batasan masalah, karena apa yang kita batasi pada batasan masalah akan membuat kita fokus pada objek penelitian yang kita kerjakan serta penelitian kita tidak akan meluas mencakup bidang lain.

2. Akurasi dari sistem dalam mendeteksi marka membujur garis utuh dan marka membujur garis putuh-putuh dilakukan secara sendiri-sendiri setelah didapatkan nilai persentase tiap tipe selanjutnya adalah di cari rata rata dari

kedua jenis marka membujur garis sehingga didapatkan akursi total. Proses akurasi dilakukan dengan cara mengambil datalatih dan adata uji sebanyak 20 kali. Hasil yang diadapatkan dari deteksi markamembujur garis utuh mendapatkan persentasi keberhasilan sebesar 85 % dengan data latih dan data uji yang gagal sebanyak 3 tiga gambar, sedang pada deteksi marka membujur garis putus-putus mendapatkan persentasikeberhasilan sebesar 90% dengan data latih dan data uji yang gagal sebanyak 2 dua gambar.Hasil rata rata total dari kedua tie marka membujur garis sebesar 87,5%.

3. Akurasi dari sistem dalam mendeteksi jarak dengan marka membujur garis dilakukan tiap statusnya yaitu status aman status hati-hati,status awas ,dan status bahaya, setelah semua status didapatkan persentase keberhasilan, maka keempat status tersebut dicari rata rata dari semua status sehingga didapatkan akurasi totalnya. Proses akurasi dilakukan dengan cara mengambil data latih dan data uji sebanyak 20 kali. Hasil yang didapatkan dari deteksi jarak dengan marka membujur garis pada status aman didapatkan persentase keberhasilan sebesar 95%, dengan data latih dan data uji gagal sebanyak 1 satu gambar,pada status hati-hati didapatkan persentase keberhasilan sebesar 85%, dengan data latih dan data uji yang gagal sebanyak 3 tiga gambar,serta pada status awas didapatkan persentase keberhasilan sebesar 95%,dengan data latih dan data uji yang gagal sebanyak 1 satu gambar,dan pada status bahaya hasil yang yang didapatkan adalah maksimal dengan persentase keberhasilan sebesar 100%, dengan data latih dan data uji yang gagal sebesar nol.Hasil rata rata total dari keempat status tersebuat didapatkan persentasi keberhasilan sebesar 93,73%.

4. Waktu yang dibutuhkan untuk melakukan sebuah proses bergantung pada alur progam yang dibuat semakin efisien alur progam maka semakin cepat waktu proses. Pada penelitian ini pengukuran rata-rata waktu proses dilakukan pada keempat status yaitu status aman beserta status garis,status hari-hati beserta status garis, status awas beserta status garis,stats bahaya berserta status garis secara independen, setalah didapat waktu rata rata tiap status maka keempat status akan di cari waktu rata rata total sistem. Hasil yang didapatkan pada status aman beserta status marka didapatkan waktu **1,25 detik** ,didapat dari sepuluh kali pengujian.Pada status hati-hati beserta status marka didapatkan waktu rata-rata sebesar **1,27 detik**,didapat dari sepuluh kali pengujian.Selanjutnyapada status awas beserta status marka didapatkan waktu rata-rata **1,27 detik**,didapat dari sepuluh kali pengujian. Sertapada status bahaya beserta status marka didapatkan waktu rata-rata sebesar **1,28 detik**,didapat dari sepuluh kali pengujian. Hasil rata rata total dari keempat status tersebut didapatkan waktu rata-rata sebesar**1,26 detik**.

7.1Saran

Dalam penelitian ini terdapat beberapa potensi yang dapat dikembangkan menjadi penelitian berikutnya, saran dari penulis adalah sebagai berikut:

1. Penelitian ini dapat dikembangkan lagi pada tipe kendaraan lain yang banyak digunakan serta memperbaiki algoritma deteksi agar akurasinya dapat lebih baik , contohnya adalah kendaraan tipe mobil, karena marka membujur garis yang ada pada jalan raya berlaku juga dengan kendaraan mobil, maka penelitian menggunakan tipe kendaraan mobil dapat dilakukan, yang perlu dilakukan adalah menentukan ulang penempatan video, center width point, serta nilai deteksi untuk menentukan status jarak yang dicari sesuai dengan kondisi serta melakukan perombakan algoritma agar akurasi sistem dapat lebih baik..

2. Untuk pengembangan lebih kompleks penelitian ini dapat dikembangkan untuk mendukung sistem *autonomous* pada kendaraan. Sistem *autonomus* pada kendaraan tentunya membutuhkan pembacaan untuk mendeteksi marka jalan salah satunya marka membujur garis beserta jarak kendaraan dengan marka jalan tersebut, sehingga sangat dimungkinkan penelitian tersebut dapat dilakukan.

DAFTAR REFERENSI

Ahmad, Usman., 2005. Pengolahan Citra Digital dan Teknik Pemrogramannya. Edisi Pertama Yogyakarta: Graha ilmu

Andre Wedianto, 2016. Analisa Perbandingan Metode Filter Gausian,Mean,dan Medianterhadap Reduksi Noise Vol 12

Average Filter, 2013. Averaging . [Online] Available at : https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html [diakses 2 februari 2019]

Basuki, Achmad, 2005. *Metode Numerik dan Algoritma Komputasi*, Yogyakarta

Binary, 2018. Transform Binary. [Online] Available at : https://www.mathworks.com/help/images/distance-transform-of-a-binary-image.html?s_tid=srchtitle [diakses 25 januari 2018]

Chandra,E. A., Jufry, H., Tan,S., 2013. Deteksi Marka Jalan dan Estimasi Posisi Menggunakan Muktiresolution Hough Transform. Vol.21, No. 1, Jakarta Barat: Universitas Binus

Fauzah,Abd. 2014. Tiga Metode Algoritma Mengubah Nilai RGB Menjadi Grayscale. [Online] Available at: http://www.charisfauzan.net/2014/06/tiga-metode-algoritma-mengubah-nilai-r.html [diakses 25 Januari 2018]

Fitradian,2018. Waspada Melanggar Marka Lurus Diancam Denda Rp 500 ribu. [Online] Available at :https://www.otosia.com/berita/waspada-melanggar-markah-lurus-diancam-denda-rp-500-ribu.html [diakses 26 januari 2018]

Hougline, 2015.Tutorial Hougline. [Online] Available at : https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html [diakses 25 januari 2018]

Keputusan menteri perhubungan, 2014. PERATURAN MENTERI PERHUBUNGAN REPUBLIK INDONESIA NOMOR KM 34 TAHUN 2014

Korlantas, 2013. Kecelakaan di Indonesia Selama Triwulan Terakhir. [Online] Available at :http://www.korlantas-irsms.info/graph/accidentData?lang=id[diakses 20 Januari 2018]

Korlantas polri,2017.10 Penyebab Utama Kecelakaan Lalu Lintas Menurut Korlantas Polri. [Online] Available at : https://otomotif.tempo.co/read/1022850/10-penyebab-utama-kecelakaan-lalu-lintas-menurut-korlantas-polri [diakses 20 Januari 2018]

Mathwork, 2018.ImageThresholding Procesing. [Online] Available at : https://www.mathworks.com/discovery/imagethresholding.html[diakses pada 19 September 2018]

Opencv. 2017. Introduction to OpenCV. [Online] Available at : http://binus.ac.id/malang/2017/10/introduction-to-open-cv/[diakses 2 Februari 2019]

Patrick, 2017. Simple Line detection using Raspberry Pi Camera and Open CV, Python. [Online] Available at: https://www.openmaker.co.kr/single-post/2017/03/17/Simple-Line-Detection-using-Raspberry-Pi-camera-and-OpenCV-Python-for-drone-applications [diakses 05 September 2018]

Pollo, D., E., D., G. & Aduatae, D., 2013. Deteksi dan Klasifikasi Marka Pembatas Jalan Menggunakan Metode Voting dan Sampling. Vol 1, ISSN 9772252 – 669007.

Pratama,Christoper Johan., 2015. *Implementasi Metode Thresholding dan Metode Regionprops untuk Mendeteksi Marka Jalan Secara Live Video,* Jakarta Barat: Universitas Binus

Rahma,Sayuti.2013.,Thresholding. [Online] Available at : http://sayutirahman.stthmedan.ac.id/2013/12/thresholding.html [diakses 20 Januari 2018]

Raspberry Pi3 model B, 2016. Review Raspberry Pi 3 Model B.[Online] Available at: http://www.pcmag.com/review/342705/raspberry-pi-3-model-b [diakses 15 April 2018]

Rege, S., Memane, R., Phatak M.& Agarwal, P., 2013. 2D Geometric Shape and Color Recognition Using Digital Image Processing. Vol 2, pp. 2320 – 3765.

ROI, Ichwanti, Nurul, 2013. Implementasi Pemrogaman Citra Metode Croping Menggunakan C.SHARP dan MATLAB. [Online] Available at: https://nenkk.wordpress.com/2013/07/19/tugas-artikel-ilmiah/#more-368 [dikses 10 Februari 2019]

Simanjuntak, T., dkk. 2011. Deteksi Pelanggaran Marka Jalan Raya berbasis Korelasi Citra.

Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, O., D.& Wijanarto.2009. "Teori Pengolahan Citra Digital", Yogyakarta : Penerbit Andi.

repository.ub.ac.id

LAMPIRAN A KODE PROGAM UTUH

NO.	Source Code
1.	import os
2.	import cv2
3.	import numpy as np
4.	import math
5.	import time
6.	from gpiozero import Buzzer, LED, TrafficLights
7.	from time import sleep
8.	buzzer = Buzzer(4)
9.	lights = TrafficLights(17, 27, 22)
10.	def region_of_interest_v1(img, vertices):
11.	mask = np.zeros_like(img)
12.	channel_count = img.shape[2]

UNIVERSITAS BRAWIJAYA

```
13.     match_mask_color = (255,) * channel_count
14.     cv2.fillPoly(mask, vertices,
    match_mask_color)
15.     masked_image = cv2.bitwise_and(img, mask)
16.     return masked_image
17. def region_of_interest(img, vertices):
18.     mask = np.zeros_like(img)
19.     match_mask_color = 255
20.     cv2.fillPoly(mask, vertices,
    match_mask_color)
21.     masked_image = cv2.bitwise_and(img, mask)
22.     return masked_image
23. dirData = "/home/pi/Project/UJI_ACAK_2/"
    arr = os.listdir(dirData)
24. def resizeForce(image, width=None, height=None,
25. inter=cv2.INTER_AREA):
        (h, w) = image.shape[:2]
26.     if width is None and height is None:
27.         return image
28.     resized = cv2.resize(image, (width, height),
    interpolation=inter)
29.     return resized
30. def draw_linesv2(img, lines, color=[255, 0, 0],
    thickness=3):
        # If there are no lines to draw, exit.
31.     if lines is None:
32.         return
33.     img = np.copy(img)
34.     line_img = np.zeros(
            (
35.             img.shape[0],
36.             img.shape[1],

                3
            ),
37.         dtype=np.uint8,
        )
38.     for line in lines:
39.         for x1, y1, x2, y2 in line:
40.             cv2.line(line_img, (x1, y1), (x2,
    y2), color, thickness)
41.     img = cv2.addWeighted(img, 0.8, line_img,
    1.0, 0.0)
42.     return img
43. def draw_lines_v1(height, width, img, lines,
    color=[255, 0, 0], thickness=5):
        line_img = np.zeros(
            (
44.             img.shape[0],
```


repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
45.                 img.shape[1],
                    3
                ),
47.             dtype=np.uint8
            )
48.         img = np.copy(img)
49.         if lines is None:
50.             return
51.         for line in lines:
52.             for x1, y1, x2, y2 in line:
53.                 cv2.line(line_img, (x1, y1), (x2,
    y2), color, thickness)
54.         img = cv2.addWeighted(img, 0.8, line_img,
    1.0, 0.0)
55.         return img
56.     def draw_lines(height, width, img, lines,
    color=[255, 255, 255], thickness=5, statuss=""):
57.         if lines is None:
58.             return
59.         img = np.copy(img)
60.         line_img = np.zeros(
                (
61.                 img.shape[0],
62.                 img.shape[1],
                    3
                ),
63.             dtype=np.uint8,
            )
64.         cek = 0
65.         for line in lines:
66.             for x1, y1, x2, y2 in line:
67.                 cek += x1;
68.         img = cv2.addWeighted(img, 0.8, line_img,
    1.0, 0.0)
69.         lights.off()
70.         buzzer.off()
71.         jarak =  x1 - (width/2)
72.         if (x1 > width/2 ) :
                color2 = [0, 0, 255]
73.             print ("{}{}".format("Jarak Dari Garis :
    ",jarak))
74.             if jarak<=100:
75.                 color2 = [0, 0, 255]
76.                 cek = 1
77.                 lights.red.on()
78.                 sleep(1)
79.                 print ("{}{}".format("Status : ",
    "AWAS"))
80.             elif jarak<=290:
```

```
            color2 = [0, 255, 255]
            cek = 1
            lights.amber.on()
            sleep(1)
            print ("{}{}".format("Status : ",
"HATI HATI"))
        else:
            color2 = [0, 255, 0]
            lights.green.on()
            sleep(1)
            print ("{}{}".format("Status : ",
"AMAN"))
    else:
       jarak =  abs(jarak)
       print ("{}{}".format("Jarak Dari Garis :
",jarak))
       color2 = [0, 0, 255]
       lights.red.on()
       if statuss=="Utuh":
           buzzer.on()
       sleep(1)
       print ("{}{}".format("Status : ",
"BAHAYA"))
    for line in lines:
        for x1, y1, x2, y2 in line:
            cv2.line(img, (x1, y1), (x2, y2),
color, thickness)
    cv2.circle(img, (width / 2, height), 20,
color2, -1)
    cv2.imshow('Video Frame', img)
    return img
def readVideo(path):
    cap = cv2.VideoCapture(path)
    if (cap.isOpened() == False):
        print("Error opening video stream or
file")
    while (cap.isOpened()):
        ret, frame = cap.read()
        if ret == True:
            height, width, channels = frame.shape
            print ("----------------------------
----------------------------------------")
            start_time = time.time()
            resized_image = resizeForce(frame,
width=800, height=600, inter=cv2.INTER_AREA);
            height, width, channels =
resized_image.shape
            region_of_interest_vertices = [
                (0, height),
```

```
                    (0, height * 0.7),
                    (width * 0.5, height * 0.4),
                    (width, height * 0.7),
                    (width, height),
                ]
                image = resized_image
                blur = cv2.blur(image, (3, 3))
                gray_image = cv2.cvtColor(blur,
cv2.COLOR_RGB2GRAY)
                cropped_image = region_of_interest(
                    gray_image,
np.array([region_of_interest_vertices],
np.int32),
                )
cv2.imshow('cropped_image_ori',cropped_image)
                cannyed_image = cv2.Canny(gray_image,
50, 200)
                cropped_image = region_of_interest(
                    cannyed_image,
np.array([region_of_interest_vertices],
np.int32),
                )
                lines = cv2.HoughLinesP(
                    cropped_image,
                    rho=5,
                    theta=np.pi / 60,
                    threshold=90,
                    lines=np.array([]),
                    minLineLength=30,
                    maxLineGap=900
                )
                left_line_x = []
                left_line_y = []
                if lines is None:
                    cv2.imshow('Video Frame',
resized_image)
                    if cv2.waitKey(1000) == 27:
                        break
                    continue
                #
                line_image = draw_linesv2(image,
lines)
                x_ =[]
                y_ =[]
                idex = 0;

                xMIn = 10000;
                yMIn = 10000;
                xMax = -10000;
```

```
151.                yMax = -10000;
152.                for line in lines:
153.                    for x1, y1, x2, y2 in line:
154.                        if (y2 <(yMIn)):
155.                            yMIn = y2
156.                            xMIn = x2
157,                        if (y1 <(yMIn)):
158.                            yMIn = y1
159.                            xMIn = x1
160.                        if (y1>( yMax)):
161.                            yMax = y1
162.                            xMax = x1
163.                        if (y2>( yMax)):
164.                            yMax = y2
165.                            xMax = x2
166.                        slope = (y2 - y1) / (x2 - x1)
167.                        x_.append(x1)
                            x_.append(x2)
168.                        y_.append(y1)
169.                        y_.append(y2)
170.                        idex = idex+1
171.                        cv2.circle(line_image, (x1,
172. y1), 3, [255, 0, 0], -1)
173.                        cv2.circle(line_image, (x2,
    y2), 3, [0, 255, 0], -1)
174.                panjangGaris = math.hypot(xMax -
    xMIn, yMax - yMIn)
175.                print ("Panjang Garis :
    {}".format(panjangGaris)  )
176.                status = "Putus"
177.                if (panjangGaris > 340):
178.                    status = "Utuh"
179.                cv2.circle(line_image, (xMIn, yMIn),
180. 3, [0, 0, 255], -1)
181.                cv2.circle(line_image, (xMax, yMax),
    3, [0, 255, 255], -1)
182.                cv2.imshow('cropped_image',lines)
                    if math.fabs(slope) < 0.1:
182.                    cv2.imshow('Video Frame', image)
183.                    if cv2.waitKey(1000) == 27:
184.                        break
185.                    continue
186.                left_line_x.extend([xMIn, xMax])
187.                left_line_y.extend([yMIn, yMax])
188.
189.                min_y = yMIn
                    max_y = yMax
190.                poly_left = np.poly1d(np.polyfit(
191.                    left_line_y,
```


repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
                    left_line_x,
                    deg=1
                ))
                left_x_start = int(poly_left(max_y))
                left_x_end = int(poly_left(min_y))
                line_image = draw_lines(height,
width,
                    image,
                    [[
                        [left_x_start, max_y,
left_x_end, min_y],
                    ]],
                    thickness=15, statuss=status   )
                print ("Status Jalan :
{}".format(status)  )
                print("Time : %s seconds" %
(time.time() - start_time))
                print ("-----------------------------
-----------------------------------------")
                if cv2.waitKey(1000) == 27:
                    break
            else:
                break
        cap.release()
        cv2.destroyAllWindows()
for fi in arr:
        path = ("".join([dirData, fi]))
        print (path)
        lights.off()
        buzzer.off()
        readVideo(path)
```