



TUGAS AKHIR - EE184801

## **RANCANG BANGUN SISTEM PENGHITUNG JUMLAH KENDARAAN PADA JALAN RAYA BERBASIS *INTERNET OF THINGS***

Akhmad Lutfi Rusidi  
NRP 07111340000171

Dosen Pembimbing  
Dr.Ir. Hendra Kusuma, M.Eng.S.  
Dr. Eng Ir. Totok Mujiono, M.kom

DEPATERMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018







**TUGAS AKHIR - EE184801**

**RANCANG BANGUN SISTEM PENGHITUNG JUMLAH  
KENDARAAN PADA JALAN RAYA BERBASIS *INTERNET  
OF THINGS***

Akhmad Lutfi Rusidi  
NRP 07111340000171

Dosen Pembimbing  
Dr.Ir. Hendra Kusuma, M.Eng.S.  
Dr. Eng Ir. Totok Mujiono, M.kom

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





**FINAL PROJECT - EE184801**

## **DESIGN OF VEHICLE COUNTING SYSTEMS ON HIGHWAY BASED AN INTERNET OF THINGS**

**Akhmad Lutfi Rusidi**  
**NRP 07111340000171**

**Supervisor**  
**Dr.Ir. Hendra Kusuma, M.Eng.S.**  
**Dr. Eng Ir. Totok Mujiono, M.kom**

**ELECTRICAL ENGINEERING DEPARTMENT**  
**Faculty of Electrical Technology**  
**Sepuluh Nopember Institute of Technology**  
**Surabaya 2018**

## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Rancang Bangun Sistem Penghitung Jumlah Kendaraan Pada Jalan Raya Berbasis *Internet of Things*” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 12 Desember 2018

Akhmad Lutfi Rusidi  
NRP. 07111340000171





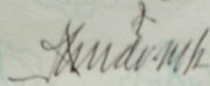
**Rancang Bangun Sistem Penghitung Jumlah Kendaraan  
pada Jalan Raya Berbasis Internet of Things**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk  
Memperoleh Gelar Sarjana Teknik  
Pada  
Bidang Studi Elektronika  
Departemen Teknik Elektro  
Institut Teknologi Sepuluh Nopember

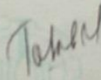
Menyetujui

Dosen Pembimbing I



Dr. Ir. Hendra Kusuma, M.Eng.Sc.  
NIP. 196409021989031003

Dosen Pembimbing II



Dr. Eng. Ir. Totok Mujiono, M.Kom  
NIP. 196504221989031001





# **Rancang Bangun Sistem Penghitung Jumlah Kendaraan Pada Jalan Raya Berbasis *Internet of Things***

Nama : Akhmad Lutfi Rusidi  
Pembimbing : Dr.Ir. Hendra Kusuma, M.Eng.Sc.  
Dr. Eng. Ir. Totok Mujiono, M.Kom

## **ABSTRAK**

Pada tugas akhir ini dikembangkan suatu sistem penghitung jumlah kendaraan kategori L dan kendaraan kategori M, dimana kendaraan kategori L adalah kendaraan yang memiliki roda kurang dari 4 dan kendaraan kategori M ialah kendaraan yang memiliki 4 roda atau lebih, Sistem ini bertujuan untuk memberi informasi kepadatan jalan raya baik kepada pemerintah maupun kepada pengguna jalan, metode yang akan digunakan pada sistem ini adalah melakukan pembaharuan obyek yang ada secara langsung, sehingga mendapatkan obyek, lalu obyek tersebut akan diidentifikasi sebagai kategori M atau kategori L, bila objek melalui garis yang diinginkan maka penghitung akan aktif, dengan cara luas objek yang melewati garis sehingga akan ditentukan jenis obyek tersebut, posisi kamera diletakan vertikal dengan tanah sehingga pada saat kemacetan kamera masih dapat melakukan proses identifikasi obyek. Dan mampu menyimpan data jumlah kendaraan di web server. Sistem ini akan direalisasikan dengan menggunakan komputer personal sebagai hardware untuk pemroses data. Hasil dari tugas akhir ini memiliki rata-rata presentase akurasi kebenaran 93,175% dari 5 menit percobaan dengan kesalahan rata-rata 0,6 kesalahan per menitnya. Metode pengujian pada tugas akhir ini dilakukan dengan cara melakukan perbandingan antara perhitungan manual dan hasil yang di dapatkan pada program.

Kata kunci: *Internet of thing(IOT)*, pengolahan gambar digital, penghitung Kendaraan



# **Traffic Vehicle Counting Design Based on Internet of Things**

*Name* : Akhmad Lutfi Rusidi  
*Supervisor* : Dr.Ir. Hendra Kusuma, M.Eng.Sc.  
Dr. Eng. Ir. Totok Mujiono, M.Kom

## ***ABSTRACT***

This undergraduate theses concerns the development of category L and M vehicle counting system, in which the L category consist of vehicles with less than 4 wheels, while M type vehicles consist of vehicles with 4 wheels or more. This system aims to informs the traffic density whether to the government or the traffic user. The method used for this system is to directly update the object, which then identifies the object as either the L or M category vehicle. When the object passes through the line, the counter will be active, with the area used as identifier of what the object is. The camera is positioned vertically above ground, so that during dense traffic, the camera may still be able to identify the objects, and capable of storing the data regarding the amount of vehicle counted in the web server. This system will be implemented using Personal Computer as data processing hardware. Final result of this project shows average percentage of accuracy for 93,175% for 5 minutes test, with the average error amounts to 0,6 per minute. The method on this final project is done by using comparison between manual counting and the result obtained from program

*Keywords: internet of thing(IOT), vehicle counting, computer vision*

.....*Halaman ini sengaja dikosongkan*.....

## KATA PENGANTAR

Puji dan syukur kepada Allah SWT atas berkat dan hikmat yang diberikan, penulis dapat menyelesaikan laporan tugas akhir dengan judul “**Rancang Bangun Sistem Penghitung Jumlah Kendaraan Pada Jalan Raya Berbasis *Internet Of Things*.**”, sebagai salah satu persyaratan dalam menyelesaikan pendidikan program sarjana di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.

Dalam penyusunan dan penyelesaian laporan Tugas Akhir ini penulis bantuan dan dukungan dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih yang sebesar – besarnya kepada :

1. Kepala Departemen Teknik Elektro ITS.
2. Kepala bidang studi elektronika Departemen Teknik Elektro ITS
3. Dosen pembimbing tugas akhir ini bapak Dr. Eng. Ir. Totok Mujiono, M.Kom dan bapak Dr.Ir. Hendra Kusuma, M.Eng.Sc.

Penulis menyadari bahwa masih banyak yang dapat dikembangkan pada tugas akhir ini. Oleh karena itu penulis menerima setiap masukan dan kritik yang diberikan. Semoga tugas akhir ini dapat memberikan manfaat bagi banyak pihak.

Surabaya, 12 Desember 2019

Akhmad Lutfi Rusidi

.....*Halaman ini sengaja dikosongkan*.....



## DAFTAR ISI

ABSTRAK.....	v
<i>ABSTRACT</i> .....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xv
<b>PENDAHULUAN</b> .....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah .....	2
1.3. Tujuan Penelitian .....	2
1.4. Batasan Masalah .....	2
1.5. Metodologi Penelitian.....	2
1.6. Sistematika Penulisan .....	3
1.7. Relevansi .....	4
<b>TINJAUAN PUSTAKA</b> .....	5
2.1. Penginderaan komputer .....	5
2.2. Dji Phantom 4.....	6
2.3. Gambar Grayscale .....	8
2.3. Gambar RGB .....	9
2.4. Median Filtering .....	11
2.5. OpenCV .....	12
2.7. FTP .....	13
2.8. Python.....	13
2.9. Pencarian Kontur .....	14
2.10 Erosi .....	14
2.11 Dilatasi .....	15
2.12 Closing Filter .....	15
2.13 Opening Filter .....	16
2.14 Threshsholding .....	17
2.15 Background Recontruction.....	17
2.16 Background Subtraction .....	18
2.17 Xlsx Writer .....	19
2.18 Server .....	19
2.19 Konutr area.....	20
2.20 Batch script.....	20
3.1. Perancangan Internet Of Things (IOT).....	21
3.2. Penghitungan Kendaraan .....	22
3.3. Pengumpulan Data.....	24

3.2.1.	Grayscale .....	24
3.2.2.	Penghapusan Latar belakang .....	25
3.3	Penghapusan noise .....	26
3.3.1	Blurring.....	26
3.3.2	Thresholding .....	26
3.4.	Segmentasi Video .....	29
3.5.	Penghitungan .....	29
3.6.	Klasifikasi objek dan perhitungan kendaraan .....	30
3.7	Prose pengiriman data dan pembuat file .....	31
4.1.	Pengambilan data .....	33
4.2.	Grayscale .....	34
4.3.	Penghapusan latar belakang.....	35
4.4.	Penghapusan noise .....	36
4.5.	Pemotongan gambar .....	40
4.6.	Aktif kontur .....	42
4.7.	Pembuatan daerah hitung .....	46
4.8.	Pelebaran titik hitung.....	47
4.9.	Proses hitung.....	47
4.10.	Hasil hitung.....	48
4.11.	Pembuatan file excel.....	54
4.10	Proses <i>Internet Of Things (IOT)</i> .....	54
PENUTUP .....		57
5.1.	Kesimpulan .....	57
5.2.	Saran .....	57
DAFTAR PUSTAKA .....		59
Lampiran 1 .....		61
Lampiran 2 .....		73
Biodata Penulis .....		74

## DAFTAR GAMBAR

Gambar 2. 1 Citra digital lena yang digunakan Sebagai gambar digital [5]	5
Gambar 2. 2 spesifikasi dji phantom 4 .....	6
Gambar 2. 3 spesifikasi kamera dji phantom 4.....	7
Gambar 2. 4 dji phantom 4 .....	8
Gambar 2. 5 Citra digital greysclae [6] .....	9
Gambar 2. 6 Citra pixel (gambar2.5) [6] .....	9
Gambar 2. 7 Citra RGB [6] .....	10
Gambar 2. 8 Nilai pixel (gambar2.7) [6] .....	10
Gambar 2. 9 Citrar biner [6] .....	11
Gambar 2. 10 Median blur window [8] .....	12
Gambar 2. 11 Closing filter .....	16
Gambar 2. 12 Opening Filter .....	16
Gambar 2. 13 Citra hasil threshold.....	17
Gambar 2. 14 hasil Background Recontruction .....	18
Gambar 3. 1 Blok diagram sistem <i>Internet of Things</i> .....	22
Gambar 3. 2 Flow chart sistem penghitungan .....	23
Gambar 3. 3 Citra dalam format RGB .....	24
Gambar 3. 4 Citra dengan format grayscale .....	25
Gambar 3. 5 Proses subraksi gambar .....	26
Gambar 3. 6 Median blur [5].....	27
Gambar 3. 7 Opening filter [5] .....	28
Gambar 3. 8 Close filter [5] .....	28
Gambar 3. 9 Dilatsi filter [5].....	29
Gambar 3. 10 Objek struktur.....	30
Gambar 4. 1 Citra pada awal video .....	33
Gambar 4. 2 Citra pada akhir video.....	34
Gambar 4. 3 Gambar setelah di grayscale .....	35
Gambar 4. 4 Hasil penghapusan latar belakang .....	36
Gambar 4. 5 Citra setelah bluring .....	37
Gambar 4. 6 Citra setelah threshold .....	38
Gambar 4. 7 Citra setelah open filter .....	39
Gambar 4. 8 Citra setelah Close filter .....	40
Gambar 4. 9 Citra baru bagian atas.....	41

Gambar 4. 10 Citra baru bagian bawah .....41

Gambar 4. 11 pemnaggilan arrary sebagai gambar baru .....42

Gambar 4. 12 Kontur citra atas .....43

Gambar 4. 13 Kontur citra bawah. ....43

Gambar 4. 14 Citra kontur atas dalam biner .....44

Gambar 4. 15 Citra kontur bawah dalam biner .....44

Gambar 4. 16 Citra sorting atas.....45

Gambar 4. 17 Citra sorting bawah .....45

Gambar 4. 18 Citra garis hitung atas ditunjukan oleh warna hijau .....46

Gambar 4. 19 Citra garis hitung bawah ditunjukan oleh warna hijau ...47

Gambar 4. 20Proses hitung yang dilakukan pada shell python .....48

Gambar 4. 21 Tampilan direktori pada server .....55

Gambar 4. 22 Tampilan web .....55

Gambar 4. 23 Tampilan realtime.....56

## DAFTAR TABEL

Tabel 4.1. Table akurasi kendaraan L atas. ....	49
Tabel 4.2. Table akurasi kendaraan M ats. ....	50
Tabel 4.3. Table akurasi kendaraan L bawah.. ....	51
Tabel 4.4. Table akurasi kendaraan M bawah. ....	51
Table 4.5. Table perhitungan kendaraan L atas. ....	52
Table 4.6. Table perhitungan kendaraan M atas.....	53
Table 4.7. Table perhitungan kendaraan L Bawah. ....	54
Table 4.8. Table perhitungan kendaraan M Bawah.....	54

.....*Halaman ini sengaja dikosongkan*.....

# **BAB I**

## **PENDAHULUAN**

Tugas akhir merupakan suatu penelitian yang dilakukan sebagai persyaratan akademik untuk mendapatkan gelar sarjana teknik di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Dengan topik yang diambil mengenai penghitung jumlah kendaraan menggunakan pengolah gambar

### **1.1. Latar Belakang**

Pada tahun 2015 kemacetan di Surabaya menduduki posisi keempat di dunia, kemacetan di Surabaya terjadi pada jam – jam tertentu, hal ini terjadi dikarenakan tingkat pertumbuhan kendaraan di jalan raya, sehingga diperlukan pemetaan waktu kemacetan. [1]

Dinas Perhubungan kota Surabaya terus berupaya berkontribusi pada pemerinkatan Peningkatan e-Government Indonesia (PeGI) dengan inisiatif memanfaatkan Teknologi Informasi dan Komunikasi dalam program-program pembangunan sistem cerdas (*intelligent*) untuk manajemen dan rekayasa Lalu-Lintas. Salah satu hasil program tersebut adalah penerapan awal *Intelligent Transportation System* (ITS), Upaya ini didukung oleh jaringan CCTV Surabaya sebagai bagian dari Traffic Management System. Intelligent Transport Sistem yaitu Sistem cerdas untuk mendukung manajemen transportasi dengan memanfaatkan teknologi (informasi, komunikasi, sensor, kontrol dan komputerisasi) untuk membangun sistem informasi dan manajemen transportasi secara otomatis [2].

Penginfoan jalan macet sangat perlu di infokan ke pengguna jalan sehingga pengguna jalan dapat memanfaatkan info tersebut sehingga tidak terjebak kemacetan, Dengan bantuan pengindraan komputer dan penerapan teknologi IOT system informasi jalan macet berdasarkan kepadatan jalan dapat membantu pengguna jalan sehingga memungkinkan pengguna jalan terhindar dari kemacetan.

Maka dari itu diciptakanlah sistem ini yang berfungsi untuk melakukan pemetaan waktu terhadap kemacetan yang ada pada daerah yang rawan macet, sistem ini akan memberikan data waktu kapan kepadatan kendaraan terjadi dan, bisa di pantau secara realtime

## **1.2. Perumusan Masalah**

Permasalahan yang dibahas dalam Tugas Akhir ini adalah :

1. Merancang sistem yang dapat mendeteksi kendaraan
2. Merancang sistem yang dapat mendeteksi saat kemacetan
3. Merancang sistem yang bisa membuat laporan
4. Merancang sistem yang dapat mengkasifikasikan kendaraan
5. Merancang sistem yang dapat data ke Server

## **1.3. Tujuan Penelitian**

Tujuan yang ingin dicapai dalam perancangan ini adalah :

1. Sistem dapat menghitung kendaraan
2. Sistem dapat mengkasifikasikan objek
3. Sistem dapat mengirimkan data berupa laporan

## **1.4. Batasan Masalah**

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Tidak dapat mengidentifikasi lawan arah
2. Di implementasikan pada siang hari
3. Hanya melakukan klasifikasi 2 jenis kendaraan

## **1.5. Metodologi Penelitian**

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut:

### **1. Studi literature**

Tahap ini meliputi pengumpulan dasar teori yang dapat menjadi acuan tugas akhir. Dasar teori akan diambil dari jurnal dari IEEE library ,hasil konsultasi dengan dosen pembimbing.

### **2. Perancangan Sistem IOT**

PC sebagai pusat proses, kamera yang sebagai input memberikan data yang akan di proses di raspberri pi, setelah data di proses raspberri pi mengirimkan data ke server, data yang diterima oleh server di ubah menjadi format exel dan siap untuk di unduh.



### **3. Perancangan Sistem Software**

Pada tahap ini software akan memakai software akan Menggunakan opencv sebagai pengolah gambar digital, dan menggunakan bahasa pemrograman python sebagai program utama dalam perhitungan mobil dan juga sebagai program yang digunakan untuk melakukan.

### **4. Tahap Pengujian**

Sistem akan diletakan pada daerah rawan macet, untuk pengujianyaprgram apakah mampu untuk mendeteksi obyek, Sistem akan diuji apakah program mapu mebuat laporan. Sistem akan di uji apakah program mapu mengirmkan data

### **5. Penyusunan Laporan Tugas Akhir**

Setelah dilakukan segala percobaan dan pengambilan data, maka akan dilakukan penulisan laporan tugas akhir yang final.

## **1.6. Sistematika Penulisan**

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

- **BAB I : Pendahuluan**

Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

- **BAB II : Tinjauan Pustaka**

Pada bab ini berisi mengenai teori yang mendasari penyusunan laporan tugas akhir secara umum khususnya yang berhubungan komponen yang akan digunakan.

- **BAB III : Perancangan Sistem**

Bab ini menjelaskan tentang perencanaan sistem yang meliputi perangkat perangkat lunak (*software*) untuk pembuatan alat ini.

- **BAB IV : Pengujian dan Analisis**

Pada bab ini menguraikan tentang pengujian yang dilakukan pada system tersebut dan melakukan analisa sistem tersebut

- **BAB V : Penutup**

Bab ini berisi tentang kesimpulan yang diperoleh dari pembuatan alat serta saran untuk pengembangan lebih lanjut.

### **1.7. Relevansi**

Dengan dibuatnya tugas akhir ini akan membantu melakukan perhitungan kendaraan sehingga di dapat jumlah kendaraan yang melintas pada satu waktu, data diambil menggunakan kamera yang dipasang 90 derajat kearah tanah.

## **BAB II**

### **TINJAUAN PUSTAKA**

Suatu penelitian memerlukan teori-teori yang digunakan untuk bahan referensi dan acuan sehingga penelitian ini menjadi lebih terarah.

Pada bab ini terdapat teori dasar yang menjadi landasan untuk merumuskan dan menyelesaikan masalah yang akan dibahas pada penelitian ini.

#### **2.1. Penginderaan komputer**

Penginderaan komputer adalah ilmu pemrograman komputer untuk memproses dan memahami gambar atau video, atau dapat dikatakan penglihatan komputer [3]. Penginderaan komputer juga bisa didefinisikan sebagai pengolahan gambar dua dimensi secara matematis yang digunakan membuat ulang bentuk tiga dimensi dan tampilan objek dalam gambar [4]

Penginderaan komputer adalah metode yang digunakan untuk melakukan pemrosesan gambar secara digital, utamanya digunakan untuk perbaikan pada gambar yang nantinya diartikan manusia, dan juga untuk penyimpanan data gambar untuk penyimpanan



**Gambar 2. 1** Citra digital lena yang digunakan Sebagai subjek pemrosesan gambar digital [5]

2.2. Dji Phantom 4

Dji phantom 4 adalah alat yang digukan pada tugas akhir ini, dji phantom 4 digunakan untuk melakukan pengambilan data video, dji phantom 4 memiliki spekifikasi sebagai berikut:

Pada kamera dji memiliki ukuran gambar maksimal 4000x3000 pixel, dan memiliki 4k sebagai video ukuran maksimal.

Pada drone dji phantom 4 memiliki maksimal waktu terbang selama 28 menit dan tahan terhadap kecepatan angin 10m/s untuk mempertahankan posisinya

AIRCRAFT	
Weight (Battery & Propellers Included)	1380 g
Diagonal Size (Propellers Excluded)	350 mm
Max Ascent Speed	S-mode: 6 m/s
Max Descent Speed	S-mode: 4 m/s
Max Speed	S-mode: 20 m/s
Max Tilt Angle	S-mode: 42° A-mode: 35° P-mode: 15°
Max Angular Speed	S-mode: 200°/s A-mode: 150°/s
Max Service Ceiling Above Sea Level	19685 feet (6000 m)
Max Wind Speed Resistance	10 m/s
Max Flight Time	Approx. 28 minutes
Operating Temperature Range	32° to 104°F (0° to 40°C)
Satellite Positioning Systems	GPS/GLONASS
Hover Accuracy Range	Vertical: ±0.1 m (with Vision Positioning) ±0.5 m (with GPS Positioning) Horizontal: ±0.3 m (with Vision Positioning) ±1.5 m (with GPS Positioning)

Gambar 2. 2 spesifikasi dji phantom 4

CAMERA	
Sensor	1/2.3" CMOS Effective pixels: 12.4 M
Lens	FOV 94° 20 mm (35 mm format equivalent) f/2.8 focus at ∞
ISO Range	<ul style="list-style-type: none"> <li>100-3200 (video)</li> <li>100-1600 (photo)</li> </ul>
Electronic Shutter Speed	8 - 1/8000 s
Image Size	4000×3000
Still Photography Modes	Single shot Burst shooting: 3/5/7 frames Auto Exposure Bracketing (AEB): 3/5 bracketed frames at 0.7 EV Bias Timelapse HDR
Video Recording Modes	UHD: 4096×2160 (4K) 24 / 25p 3840×2160 (4K) 24 / 25 / 30p 2704×1520 (2.7K) 24 / 25 / 30p FHD: 1920×1080 24 / 25 / 30 / 48 / 50 / 60 / 120p HD: 1280×720 24 / 25 / 30 / 48 / 50 / 60p
Max Video Bitrate	60 Mbps
Supported File Systems	FAT32 (≤32 GB); exFAT (>32 GB)
Photo	JPEG, DNG (RAW)
Video	MP4, MOV (MPEG-4 AVC/H.264)
Supported SD Cards	Micro SD Max capacity: 64 GB Class 10 or UHS-1 rating required
Operating Temperature Range	32° to 104°F (0° to 40°C)

**Gambar 2. 3** spesifikasi kamera dji phantom 4



**Gambar 2. 4** dji phantom 4

### **2.3. Gambar *Grayscale***

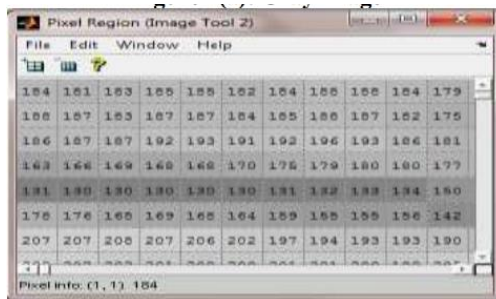
Gambar *grayscale* adalah gambar yang menggunakan intensitas skala abu-abu atau gambar dengan tingkatan abu-abu pada kelas array uint8, uint16, int16, single, atau double yang memiliki pixel, Nilai dalam array tersebut memiliki nilai intensitas tergantung pada jenis array yang digunakan [6].

Skala abu-abu adalah tingkatan yang ada pada gambar *grayscale* untuk menentukan tingkatan abu-abu yang berada pada gambar tersebut, skala abu-abu yang sering digunakan adalah penyimpanan 8-bit dimana nilai 0 menjadi warna hitam dan nilai 255 menjadi warna putih [6].

nantinya warna ini akan menjadi gambar biner dimana hanya memiliki dua nilai yaitu 0 dan 1 dimana nilai 0 mewakili warna hitam dan 1 mewakili warna putih yang akan digunakan pada proses pengolahan gambar digital.



**Gambar 2. 5** Citra digital greyscale [6]



**Gambar 2. 6** Citra pixel (gambar2.5) [6]

### 2.3. Gambar RGB

Gambar RGB juga dikenal juga sebagai gambar yang sebenarnya, dimana gambar ditentukan oleh tiga nilai dimana setiap nilainya memiliki nilai tersendiri, yang menunjukkan skala warna merah, biru, hijau itu sendiri, skala warna biasanya ditunjukkan dalam integer.

Dalam metode RGB setiap warna muncul pada spectrumnya, yang dikenal sebagai saluran warna, model warna diwakili oleh fungsi intensitas.

$$I_{RGB} = (F_R, F_G, F_B) \quad I_{RGB} = (F_R, F_G, F_B) \dots\dots\dots(1)$$

Dimana  $F_R$  adalah intensitas merah  $F_G$  adalah intensitas hijau dan  $F_B$  adalah intensitas biru [6].



**Gambar 2. 7** Citra RGB [6]

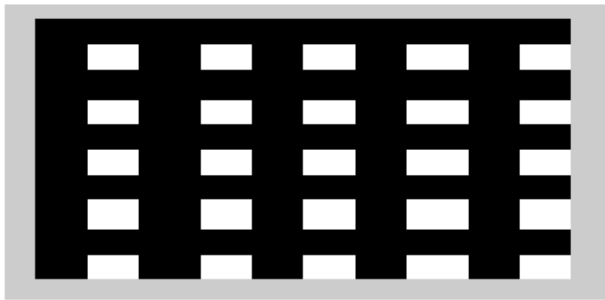
Pixel Region (Image Tool 1)					
File Edit Window Help					
R:190	R:187	R:187	R:189	R:189	R:186
G:183	G:179	G:161	G:163	G:161	G:178
B:175	B:174	B:181	B:187	B:192	B:193
R:194	R:193	R:169	R:193	R:193	R:190
G:188	G:186	G:182	G:186	G:186	G:182
B:176	B:178	B:174	B:180	B:182	B:180
R:189	R:190	R:190	R:195	R:196	R:194
G:166	G:167	G:167	G:192	G:193	G:191
B:181	B:182	B:180	B:185	B:184	B:182
R:166	R:167	R:168	R:167	R:167	R:169
G:162	G:168	G:171	G:170	G:171	G:173
B:161	B:161	B:162	B:159	B:156	B:158
Pixel info: (1, 1) [190 183 175]					

**Gambar 2. 8** Nilai pixel (gambar2.7) [6]



Gambar biner adalah gambar yang intensitasnya hanya bernilai bernilai 0 dan 1 [6], gambar biner berbeda dengan gambar grayscale yang range datanya berupa nilai 0 sampai 255, gambar biner hanya akan berwarna putih dan hitam, perubahan nilai gambar menjadi nilai gambar biner adalah untuk mendapatkan data yang spesifik dalam sebuah pemrosesan gambar digital.

Eliminasi dalam gambar biner ditentukan dari nilai gambar sebelumnya yaitu nilai gambar RGB atau nilai grayscale, gambar biner pada umumnya digunakan untuk melakukan fitur sebuah objek, gambar biner didapat dari operasi threshold yang dilakukan pada gambar sebelumnya



**Gambar 2. 9** Citrar biner [6]

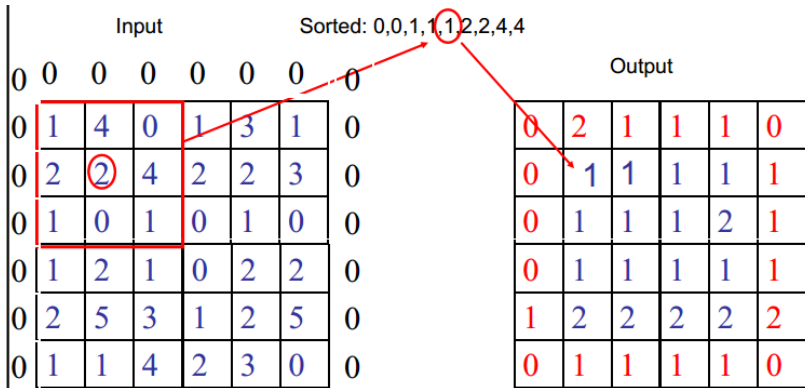
## 2.4. Median Filtering

Gambar digital kadang-kadang bisa rusak oleh kebisingan. Salah satu jenis kebisingan biasanya terkait dengan gambar digital adalah suara impuls. Kebisingan impuls bias dimodelkan secara matematis oleh beberapa persamaan [7].

Filter standar median (SMF), atau juga dikenal sebagai median halus, telah diperkenalkan oleh Tukey pada tahun 1971 [6].

Salah satu cabang median filter adalah *weighted median filter* (WMF). WMF pertama kali diperkenalkan oleh Justusson di 1981, dan selanjutnya diuraikan oleh Brownrigg. Operasinya yang ada dalam WMF mirip dengan SMF, kecuali bahwa WMF memiliki berat badan yang terkait

dengan masing-masing elemen filternya. Ini bobot sesuai dengan jumlah duplikasi sampel untuk perhitungan nilai median [6]



**Gambar 2. 10** Median blur window [8]

## 2.5. OpenCV

*Open Source Computer Vision Library* (OpenCV) merupakan librari komputer visual dan *machine learning*. OpenCV dbuat untuk menyediakan infrastruktur umum untuk *computer vision* dan mempercepat penggunaan mesin persepsi pada produk komersial. Menjadi produk berlisensi BSD, OpenCV mempermudah pengguna *mechine learning* yang *open source*. [5]

Librari ini memiliki lebih dari 2500 algoritma yang dioptimalkan, yang mencakup satu set lengkap *computer vision* dan algoritma *mechine learning*. Algoritma ini dapat digunakan untuk mendeteksi wajah dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia dalam video, melacak gerakan dengan kamera, melacak objek yang bergerak, mengekstrak objek model 3D, menggabungkan gambar untuk menghasilkan gambar dengan resolusi tinggi, menemukan gambar yang sama pada database gambar, mengikuti gerakan mata, dan lain-lain. OpenCV memiliki lebih dari 47 ribu pengguna dan perkiraan jumlah yang mengunduh lebih dari 14 juta [5].

## 2.7 FTP

*File transfer protocol*(FTP) adalah operasi yang digunakan untuk mengunggah file ke server FTP dan operasi FTP digunakan untuk mengunduh file dari server FTP ke node klien. Di OPNET, aplikasi FTP memodelkan proses mentransfer file dalam satu waktu. Secara default, FTP berjalan di atas TCP. Ketika menggunakan protokol transport default, ia membuka koneksi TCP terpisah untuk setiap file. Tidak seperti di jaringan nyata, selama prosedur operasi transfer satu file, model FTP menggunakan koneksi TCP yang sama untuk mengirim pesan kontrol dan pemodelan data [9]

## 2.8 Python

Python adalah bahasa pemrograman tingkat tinggi yang difafsirkan, berorientasi objek, dengan semantik dinamis. Tingkatannya yang tinggi dalam struktur data, dikombinasikan dengan pengetikan dinamis dan pengikatan dinamis, membuatnya sangat menarik untuk Pengembangan Aplikasi Cepat, serta untuk digunakan sebagai bahasa scripting untuk menghubungkan komponen yang ada secara bersamaan. Python, mudah dipelajari sintaks menekankan keterbacaan dan karenanya mengurangi biaya pemeliharaan program. Python mendukung modul dan paket, yang mendorong modularitas program dan penggunaan kembali kode. Interpreter Python dan pustaka standar yang luas tersedia dalam bentuk sumber atau biner tanpa biaya untuk semua platform utama, dan dapat didistribusikan secara bebas [10].

Seringkali, programmer menyukai Python karena peningkatan produktivitas yang diberikannya. Karena tidak ada langkah kompilasi, siklus edit-uji-debug sangat cepat. Debugging Python program mudah: bug atau input yang buruk tidak akan pernah menyebabkan kesalahan segmentasi. Sebaliknya, ketika penerjemah menemukan kesalahan, itu menimbulkan pengecualian. Ketika program tidak menangkap pengecualian, interpreter mencetak jejak stack. Sebuah debugger tingkat sumber memungkinkan pemeriksaan variabel lokal dan global, evaluasi ekspresi sembarang, pengaturan breakpoint, melangkah melalui kode baris pada suatu waktu, dan seterusnya. Debugger ditulis dengan Python itu sendiri, memberi kesaksian

tentang kekuatan introspektif Python. Di sisi lain, seringkali cara tercepat untuk men-debug sebuah program adalah dengan menambahkan beberapa pernyataan cetak ke sumbernya: siklus edit-test-debug yang cepat membuat pendekatan sederhana ini sangat efektif [10].

## 2.9 Pencarian Kontur

Pencarian kontur dilakukan dengan melakukan kode rantai( *chain code*) digunakan untuk mempresentasikan batas berdasarkan rangkaian yang terkoneksi dari segmen garis lurus dengan arah dan panjang tertentu . representasi yang digunakan berdasarkan 8 koentivitas yang menyerupai arah mata angin.

Kode rantai dapat dibentuk dengan menelusuri batasan objek searah jarum jam dan menentukan arah pada setiap pasang filter yang terkoneksi. Arah tersebut kemudian dikodekan berdasarkan skema penomoran. [11]

## 2.10 Erosi

*Erode* atau erosi merupakan metode *Morphological Operation* dasar pada gambar digital yang berfungsi mengurangi nilai piksel dengan membandingkan nilai piksel terhadap nilai piksel tetangganya. Metode erosi sering digunakan untuk menghilangkan piksel noise pada sebuah gambar digital. Erosi dimodelkan seperti pada gambar 2.4 dimana *E* merupakan matriks luaran dari proses erosi, *A* adalah matriks *input*, dan *S* adalah matriks *Kernel* atau *Structure Element*. [12]

Ditentukan poros matriks *S* berada ditengah kemudian poros diletakkan pada posisi awal matriks *A* (0,0). Setelah itu dilakukan pengecekan apakah seluruh piksel matriks *S* yang *overlap* dengan matriks *A* terlingkupi oleh matriks *A*. Jika iya maka nilai matriks pada poros akan dipertahankan, selainnya akan dihapus. [11]

$$E(A,S) = A \ominus S \dots\dots (2)$$

## 2.11 Dilatasi

*Dilation* atau dilasi merupakan metode *Morphological Operation* dasar pada gambar digital yang berfungsi menambahkan nilai piksel dengan membandingkan nilai piksel terhadap nilai piksel tetangganya. Dilasi sering digunakan untuk menutup piksel yang "bolong" pada sebuah gambar digital. Dilasi dimodelkan seperti pada gambar 2.6 dimana  $D$  merupakan matriks luaran dari proses dilasi,  $A$  adalah matriks *input*, dan  $S$  adalah matriks *Kernel* atau *Structure Element*. Matriks *Kernel* atau *Structure* [12]

Ditentukan poros matriks  $S$  berada ditengah kemudian poros diletakkan pada posisi awal matriks  $A$  (0,0). Setelah itu dilakukan pengecekan apakah piksel matriks  $S$  yang *overlap* dengan matriks  $A$  terdapat piksel yang bernilai bukan 0. Jika iya maka nilai matriks pada poros akan ditambahkan, selainnya akan diabaikan. [11]

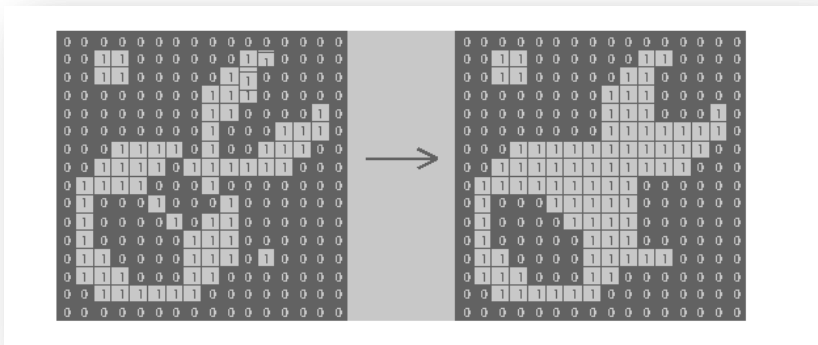
$$E(A, S) = A \oplus S \dots (3)$$

## 2.12 Closing Filter

Closing filter adalah morfologi filter yang digunakan pada gambar biner, filter ini digunakan untuk menutupi kontur yang berlubang, filter ini adalah filter yang menggunakan dua filter yaitu filter dilatasi dan filter erosi.

Cara kerja filter ini adalah dimana Gambar akan dilakukan proses dilatasi filter lalu selanjutnya akan dilakukan proses erosi

Hal ini menyebabkan gambar berlubang dapat ditutup

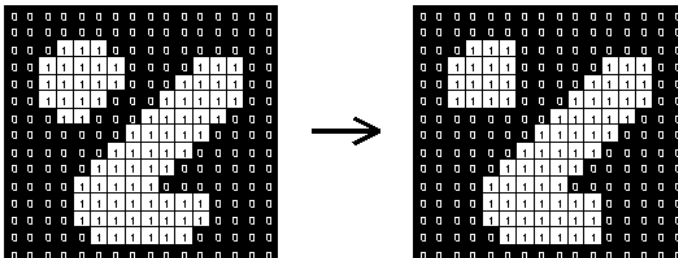


**Gambar 2. 11** Closing filter [13]

### 2.13 Opening Filter

Opening filter berguna untuk menghapus noise yang ada pada gambar filter ini adalah salah satu morfologi filter yang menggunakan dua filter morfologi, filter morfologi yang digunakan adalah, dilatasi filter dan erosi filter, dimana proses pada filter ini adalah gambar yang ada dimasukkan ke dalam erosi filter dan kemudian dilakukan proses dilatasi.

Hal ini menyebabkan noise pada gambar berkurang, proses ini dilakukan untuk mengurangi noise pada gambar.



**Gambar 2. 12** Opening Filter [13]

## 2.14 Threshholding

Pada pemrosesan gambar digital, *Thresholding* adalah suatu teknik yang terkenal untuk segmentasi gambar digital. Metode *Threshold* yang paling simpel adalah dengan mengganti setiap piksel dalam sebuah gambar dengan piksel hitam jika nilai intensitas gambar kurang dari nilai yang diinginkan, jika nilai intensitas melebihi dari nilai yang diinginkan maka pixel tersebut akan diganti menjadi 255, teknik *thresholding* paling sederhana digunakan untuk membagi pixel menjadi hitam dan putih.



**Gambar 2. 13** Citra hasil threshold

## 2.15 Background Recontruction

*Background Reconstruction* atau Rekonstruksi gambar latar adalah metode pengolahan gambar digital untuk merekonstruksi sebuah gambar latar dari sebuah rangkaian frame atau video. Proses ini digunakan untuk mendapatkan gambar latar belakang yang diinginkan

Metode yang digunakan untuk melakukan proses rekonstruksi

latar belakang adalah dengan menghitung jumlah bobot dari gambar lalu melakukan jumlah rata-rata dalam gambar yang ada

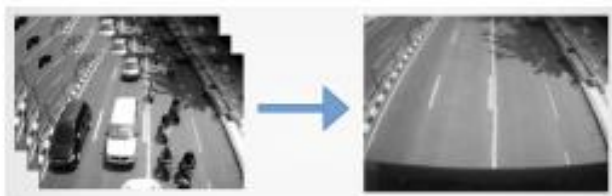
$$\text{dst}(x,y) \leftarrow (1 - \alpha) \cdot \text{dst}(x,y) + \alpha \cdot \text{src}(x,y) \text{ if } \text{mask}(x,y) \neq 0 \dots (4)$$

dst = akumulasi gambar dengan saluran yang sama

alpha = berat gambar input

src = gambar inputan sebagai chanel atau 3

mask = optional operasi



**Gambar 2. 14** hasil Background Recontruction [12]

## 2.16 Background Subtraction

Pengurangan latar belakang merupakan langkah preprocessing utama dalam banyak aplikasi berbasis visi. Misalnya, pertimbangan kasus seperti penghitung pengunjung di mana kamera statis mengambil jumlah pengunjung yang masuk atau keluar ruangan, atau kamera lalu lintas yang mengekstraksi informasi tentang kendaraan dll. Dalam semua kasus ini, pertama-tama Anda perlu mengekstrak orang atau kendaraan saja. Secara teknis, Anda perlu mengekstrak latar depan yang bergerak dari latar belakang statis [14].

untuk melakukan analisa pada pengindraan komputer maka dilakukan penghapusan latar belakang, menggunakan persamaan rekursif yang digunakan untuk terus-menerus [14].

Ini adalah Gaussian Campuran berbasis / Algoritma Segmentasi Foreground. Ini diperkenalkan di paper "An improved adaptive background mixture model for real-time tracking with shadow detection" oleh P.



KadewTraKuPong dan R. Bowden pada tahun 2001. Ini menggunakan metode untuk memodelkan setiap piksel latar belakang dengan campuran distribusi K Gaussian (  $K = 3$  hingga 5). Bobot campuran menunjukkan proporsi waktu warna-warna tersebut tetap berada di lokasi. Warna latar belakang yang mungkin adalah yang tinggal lebih lama dan lebih statis [5].

*Background Subtractor* adalah suatu metode gambar digital untuk melakukan pencarian latar depan dari sebuah video atau dari rangkaian frame untuk mendapatkan latar depan, hasil dari *Background Subtractor* adalah gambar sebelumnya.

frame dalam video melakukan penghapusan latar belakang dengan mencari rata-rata nilai latar belakang dan dilanjutkan dengan penghapusan gambar latar belakang dengan frame selanjutnya, dan setelah proses yang ada pada gambar diatas pada library opencv gambar yang dihasilkan adalah gambar yang sebelumnya digunakan, pada library opencv metode yang digunakan adalah menggunakan fungsi `backgroundsubtractor`

## 2.17 Xlsx Writer

XlsxWriter adalah modul Python untuk menulis file dalam format file Excel 2007+ XLSX.

Ini dapat digunakan untuk menulis teks, angka, dan formula ke beberapa lembar kerja dan mendukung fitur seperti pemformatan, gambar, bagan, pengaturan halaman, autofilters, pemformatan bersyarat dan banyak lainnya.

XlsxWriter memiliki beberapa kelebihan dan kekurangan dibandingkan modul Python alternatif untuk menulis file Excel. [15]

## 2.18 Server

Server adalah seperangkat komputer yang berisi program-program yang mampu menghasilkan informasi dan informasi tersebut didistribusikan kepada komputer client yang mengaksesnya.

Server secara sederhana dapat berupa satu buah komputer untuk beberapa layanan aplikasi, atau jika jaringannya lebih kompleks dan rumit, maka server dapat disetting hanya untuk memberikan satu atau beberapa layanan saja, sementara layanan yang lain diserahkan kepada server yang lain, jadi disin terjadi kolaborasi dan kerjasama dari beberapa

server untuk memberikan layanan dan informasi kepada beberapa client. Biasanya konfigurasi server yang kompleks dan rumit ini diterapkan pada organisasi yang besar seperti perusahaan-perusahaan kelas atas, sedangkan server yang terdiri dari satu buah komputer yang melayani beberapa layanan biasanya hanya digunakan untuk lingkungan yang lebih kecil. [16]

## **2.19 Kontur area**

Kontur area menghitung area kontur. Serupa dengan momen, area dihitung menggunakan rumus Hijau. Dengan demikian, area yang dikembalikan dan jumlah piksel yang bukan nol, jika Anda menggambar kontur menggunakan drawContour atau fillPoly, bisa berbeda. Juga, fungsi itu pasti akan memberikan hasil yang salah untuk kontur dengan persimpangan-sendiri [5]

Dalam matematika, teorema Green memberikan hubungan antara garis integral di sekitar kurva tertutup sederhana  $C$  dan integral ganda di atas bidang pesawat  $D$  yang dibatasi oleh  $C$ . Dinamai berdasarkan George Green, meskipun bukti pertamanya adalah karena Bernhard Riemann [1] dan merupakan kasus khusus dua dimensi dari teorema Kelvin – Stokes yang lebih umum.

## **2.20 Batch script**

Berkas instruksi atau batch file adalah sebuah berkas teks yang berisi beberapa seri perintah untuk dijalankan oleh interpreter perintah. Atau dapat didefinisikan juga batch merupakan kumpulan perintah-perintah command line yang dapat dijalankan pada Microsoft Windows. Berkas instruksi biasanya berekstensi `.bat` atau `.cmd`. Pemrograman Batch sangat sederhana, cukup ketik source code program pada notepad atau text editor lainnya lalu simpan dengan ekstensi `.bat`. [17]

## **BAB III**

### **PERANCANGAN SISTEM**

Penyelesaian tugas akhir dilakukan melalui beberapa tahap bab 3 akan menjelaskan setiap prosedur secara detail guna menyelesaikan Rancang Bangun Sistem Penghitung Jumlah Kendaraan Pada Jalan Raya Berbasis *Internet Of Things*.

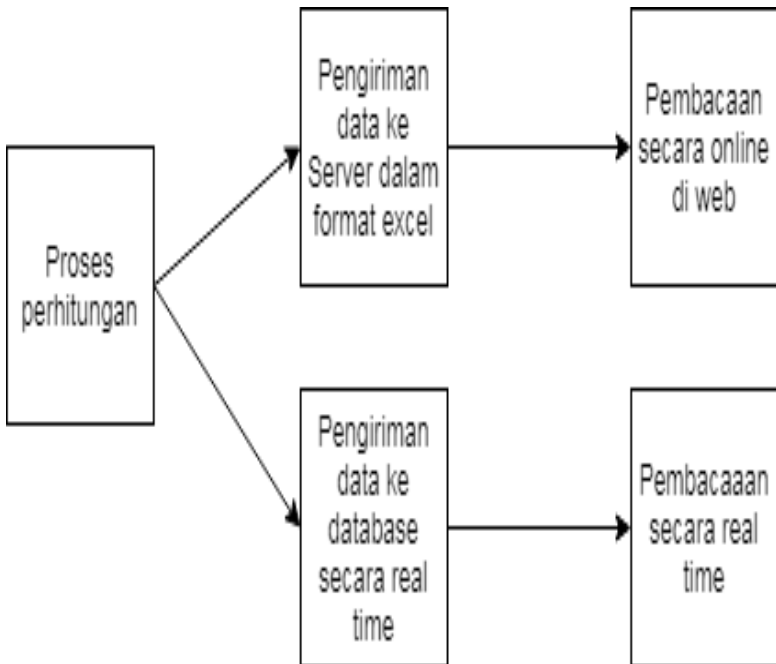
Pada tugas akhir ini seluruh system dirancang menggunakan bahasa pemrograman python 3, digunakan python 3 untuk melakukan proses pemrograman dikarenakan python 3 memiliki siklus mesin yang lebih ringan dari python 2 dan jika terjadi *bug* lebih mudah di tanggani.

Pada bab ini dijelaskan juga perancangan sistem secara keseluruhan. Metode yang digunakan bertujuan untuk melakukan perhitungan kendaraan dengan error seminimal mungkin dan melakukan pembuatan data perhitungan. Dengan menggunakan skema berikut:

#### **3.1. Perancangan *Internet Of Things* (IOT)**

Sistem IOT pada tugas akhir ini dilakukan dengan cara melakukan pengiriman data secara ke server dan database, dimana data yang dihasilkan oleh perhitungan dikirimkan secara realtime ke realtime database dimana nantinya data tersebut dapat dilihat secara realtime pada website dan juga pada perangkat smartphone, sehingga pengguna lain dapat mendapatkan data secara realtime,

Sistem ini juga pada sistem ini hasil yang diterima setiap menitnya dikirimkan ke server berupa format excel sehingga pengguna mendapatkan pemetaan jalan macet yang ada



**Gambar 3. 1** Blok diagram sistem *Internet of Things*

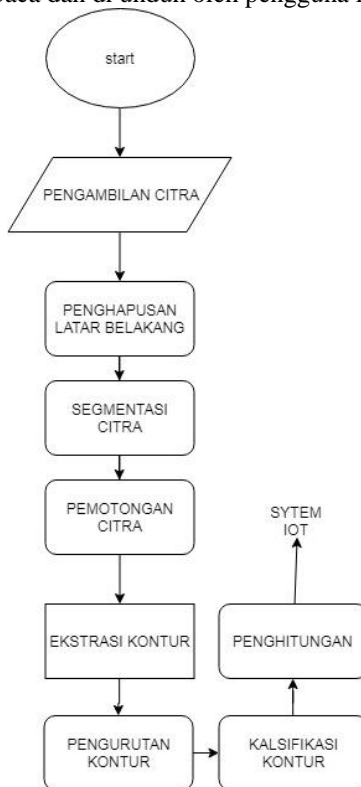
### **3.2. Penghitungan Kendaraan**

Blok diagram sistem penghitungan ditunjukkan pada Gambar 3.1, pada flow chart tersebut gambar diambil dari kamera atau video yang didapatkan, setelah gambar diambil gambar lalu diolah dengan cara melakukan pengubahan gambar ke gambar grayscale, setelah gambar grayscale didapat gambar lalu dilakukan penghapusan latar belakang untuk mendapatkan latar depan, dikarenakan gambar latar depan memiliki banyak noise, gambar latar depan dilakukan proses filtering dengan cara melakukan morfologi filter hal ini dilakukan pada proses morfologi filter.

Setelah gambar yang diproses dari morfologi filter didapat gambar lalu dilakukan pemotongan, pada gambar yang sudah dipotong dilakukan proses pencarian kontur, dikarenakan kontur yang dicari tidak sesuai urutan yang diharapkan maka kontur dilakukan pengurutan dengan cara

melakukan pengurutan kontur dengan cara melakukan pengurutan berdasarkan posisi sumbu x pada kontur tersebut.

Pada proses perhitungan kontur yang didapat diberikan titik hitung dan juga pada gambar utama diberikan garis hitung, bila titik hitung melewati garis hitung maka nilai pada counter akan bertambah. Setelah dilakukan perhitungan nilai yang didapatkan pada kontur dibuatkan format excelnya, setelah dibuatkan format excelnya data tersebut lalu dikirimkan ke server agar dapat dibaca dan di unduh oleh pengguna lain.



**Gambar 3. 2** Flow chart sistem penghitungan

Pada tugas akhir ini, library utam yang digunakan adalah OpenCV dengan menggunakan python sebagai bahasa program utamanya.

Gambar atau video yang digunakan berasal dari video drone DJI phantom 4 yang diambil di jalan Dharma husada permai dengan kamera drone menghadap secara vertikal ke tanah untuk dengan durasi video sekitar 3 menit, jumlah tersebut sudah memenuhi syarat untuk melakukan perhitungan dan pembuatan data pada tugas akhir ini.

### **3.3. Pengumpulan Data**

Proses pengambilan gambar digunakan untuk objek analisis metode yang akan digunakan pada tugas akhir ini, lokasi pengambilan dilakukan di jalan dharma husada permai menggunakan drone DJI Phantom, diambil di berbagai lokasi dan berbagai durasi untuk mendapatkan video yang ideal.

Cara pengambilan dilakukan dengan Drone DJI phantom 4 yang diambil secara vertikal menghadap ke bawah, dengan resolusi 4k pada saat pengambilan dan diolah kembali di komputer sehingga menghasilkan resolusi 1024x768 dan dengan effect kamera standar.



**Gambar 3. 3** Citra dalam format RGB

#### **3.2.1. Grayscale**

Tujuan dilakukan pengubahan warna gambar RGB menjadi grayscale agar proses penghapusan latar belakang dan proses segmentasi menjadi lebih

mudah.

Pada grayscale dilakukan perubahan warna pada gambar menjadi intensitas warna abu-pada gambar, grayscale melakukannya dengan cara melakukan penghapusan chanel warna pada gambar RGB dan dimasukkan dalam satu chanel grayscale.

Hasil gambar yang di dapatkan pada proses ini adalah warna abu-abu dengan intensitas warna hitam yang bervariasi dengan demikian hasil gambar lebih muda untuk dilakukan penghapusan pada proses selanjutnya.

Pada proses grayscale yang ada pada tugas akhir ini dilakukan perumusan nomer 4:

Rumusan diatas adalah rumusa unuk merubah 3 chanel warna Yaitu RGB menjadi 1 chanel warna yaitu Grey.

$$\text{RGB to Gray : } Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \dots\dots(5)$$



**Gambar 3. 4** Citra dengan format grayscale

### **3.2.2. Penghapusan Latar belakang**

Penghapusan latar belakang berfungsi sebagai pembeda antara latar belakang dengan latar depan, dimana nantinya akan didapatkan objek bergerak yang dikehendaki.

Penghapusan latar belakang dalam tugas akhir ini mengunakan metode `backgroundsubtractormog2` dimana setiap frame akan ditumpuk dan lalu akan diakumulasikan sehingga didapatkan objek yang bergerak dengan hasil gambar biner yang ditunjukan pada gambar 3.5.



**Gambar 3. 5** Proses subraksi gambar

### **3.3 Penghapusan noise**

Penghapusan noise dilakukan agar noise yang terdapat pada gambar berkurang atau hilang supaya gambar yang ada dapat dilakukan proses penhitungan.

Cara untuk melakukan penghapusan noise adalah dengan beberapa cara yang ada pada proses berikut.

#### **3.3.1 Blurring**

Proses bluring dilakukan agar noise pada gambar berkurang dan tidak menggagu data saat dilakukan proses analisis

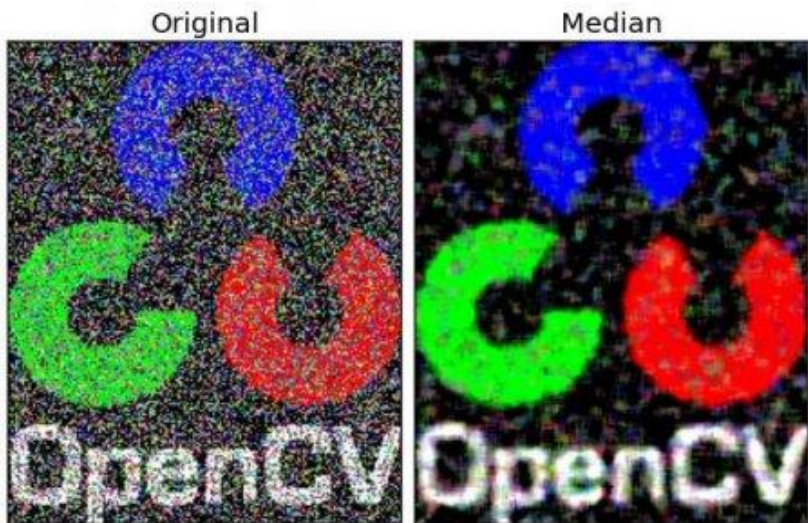
Proses segmentasi dilakukan bluring pada gambar dan dilanjutkan segmentasi proses bluring, dalam proses bluring dilakukan dengan cara membuat suatu kernel, lalu nilai di dalam kernel tersebut di urutkan dan dicari nilai tengahnya.

#### **3.3.2 thresholding**

Proses thresholding dilakukan agar noise pada gambar berkurang dan tidak menggagu data saat dilakukan proses analisis

Proses segmentasi dilakukan thresholding, dalam proses thresholding dilakukan dengan cara membuat suatu kernel, lalu nilai di dalam kernel tersebut di urutkan dan dilakukan threshold pada gambar.





**Gambar 3. 6** Median blur [5]

proses bluring dilakukan untuk menghilangkan noise pada gambar gambar digital yang telah dilakukan proses bluring menghasilkan gambar dengan noise yang lebih sedikit atau tidak ada sama sekali

### **3.4 Morfologi filter**

Pada morfology filter gambar diperhalus lagi dengan open filter dan close filter, morfology filter digunakan untuk menghindari gambar yang berhimpitan pada objek sehingga objek sulit dihitung.

Filter morfologi yang digunakan adalah filter morfologi open dan filter morfologi close dimana dan dilatasi filter.



**Gambar 3. 7** Opening filter [5]



**Gambar 3. 8** Close filter [5]



**Gambar 3. 9** Dilatsi filter [5]

### **3.4 Segmentasi Video**

Segmentasi dilakukan untuk mengurangi proses kerja pada program sehingga program lebih cepat dan prosessor pada kompute tidak bekerja lebih, objek yang di segmentasi adalah bagian yang ada pada video.

Proses segmentasi dilakukan dengan cara melakukan pemanggilan array yang ada pada video sehingga didapatkan bentuk video baru dari video yang sudah ada.

### **3.5 Penghitungan**

Hasil gambar akan dilakukan pencarian kontur sehingga objek pada gambar dapat di deteksi atau di ketahui, metode dapat mendeteksi kontur sebagai sebuah vector titik dalam gambar, metode yang digunakan adalah dengan cara mencari batas struktur objek sebuah objek.

	10	20	30	40
1	22:	333-	4444444444444444+	555= 66#
2	21:	31-	417*	771+ 51= 61#
3	2:	3-	447*	774+ 5= 6#
4	1:	3-	41*	8888888888? 71+ 5= #
5	:	31-	41*	88199999999918? 71+ 51= #
6	31-	41*	819?	991? 71+ 51=
7	3-	4*	81?	91? 7+ 5=
8	31-	4*	81?	AAAAAAA? 91? 7+ 51=
9	3-	41*	81?	A1BBBBBBB1? 91? 71+ 5=
10	3-	4*	81?	AAB BBA? 91? 7+ 5=
11	3-	4*	8?	A* B? 9? 7+ 5=
12	3-	4*	81?	A* CCCCC* B? 91? 7+ 5=
13	3-	4*	8?	A1* C1DDDD1* B1? 9? 7+ 5=
14	3-	4*	8?	A* C1 D1 B? 9? 7+ 5=
15	3-	4*	8?	A* C* D B? 9? 7+ 5=
16	3-	4*	8?	A* C* EE* D* B? 9? 7+ 5=
17	3-	4*	8?	A* C* E1* D* B? 9? 7+ 5=
18	3-	4*	8?	A* C* EE* D* B? 9? 7+ 5=
19	3-	4*	8?	A* C* D* B? 9? 7+ 5=
20	3-	4*	8?	A* C* D1* B? 9? 7+ 5=
21	3-	4*	8?	A1* C1DDDD1* B1? 9? 7+ 5=
22	3-	4*	81?	A* CCCCC* B? 91? 7+ 5=
23	3-	4*	8?	A* B? 9? 7+ 5=
24	3-	4*	81?	AAB* BBA? 91? 7+ 5=
25	3-	41*	81?	A1BBBBBBB1? 91? 71+ 5=
26	31-	4*	81?	AAAAAAA? 91? 7+ 51=
27	3-	4*	81?	91? 7+ 5=
28	31-	41*	819?	991? 71+ 51=
29	*	31-	41*	88199999999918? 71+ 51= *
30	*	3-	41*	8888888888? 71+ 5= *
31	F*	3-	447*	774+ 5= G*
32	F1*	31-	417*	771+ 51= G1*
33	FF*	333-	4444444444444444+	555= GG*
34				
35				

Gambar 3. 10 Objek struktur.

Setelah dilakukan pencarian kontur lalu dilakukan pensortingran kontur hal ini dilakukan untuk mempermudah proses perhitungan.

### 3.6 Klasifikasi objek dan perhitungan kendaraan

Pada proses klasifikasi objek, objek di kalsifikasikan berdasarkan

besaran area kontur yang dimiliki objek, pertama-tama kontur objek akan melewati area deteksi yang pasang pad video sehingga objek tersebut dapat dihitung, lalu objek akan dicari berapa besar nilai kontur yang terdapat pada objek.

Objek di sini dibedakan menjadi dua objek, objek pertama adalah objek kendaraan lebih dari 2 roda dimana objek tersebut memiliki nilai yang lebih besar dari pada objek kedua yaitu objek kendaraan 2 roda , pada objek 2 roda jumlah area akan diklasifikasikan lebih kecil dari pada objek lebih dari 2 roda.

### **3.7 Prose pengiriman data dan pembuat file**

Pada proses pembuatan file hasil perhitungan yang diterima lalu dibuat file excelnya dan file tersebut akan dikirimkan data menuju serve mengunakan FTP(File Tranfer Protocol), dengan menggunakan FTP data yang dikirimkan sudah bias berupa format file jadi sehingga langsung bias dikirimkan.

Data yang dikirmkan akan di update setiap menitnya dan akan selalu dikirimkan file baru untuk mengganti file lama yang ada pada server.

.....*Halaman ini sengaja dikosongkan*.....

## **BAB IV**

### **PENGUJIAN DAN ANALISIS**

Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang pada bab sebelumnya. Bab ini bertujuan untuk mendapatkan data yang kemudian dilakukan analisis pengujian pada proses yang dilakukan oleh setiap fungsi dalam algoritma .

#### **4.1. Pengambilan data**

Proses Pengambilan dilakukan dengan drone dji phantom 4 dan juga diambil dengan kamera drone dji phantom 4 dan dengan resolusi 1024 x 768 dan dengan efek standar.

Pada awal gambar hasil gambar yang diambil memiliki kemiringan sudut yang berbeda hal ini membuat noise yang dihasilkan pada proses berikutnya menjadi lebih banyak sehingga pada proses berikutnya harus dilakukan penghapusan noise.



**Gambar 4. 1** Citra pada awal video



**Gambar 4. 2** Citra pada akhir video

#### **4.2. Grayscale**

pada prose grayscale hasil yang terdapat pada gambar yang ada diubah dari 3 chanel warna menjadi 1 chanel warna sehingga, didapatkan intensitas tingkat keabuan.

Hasil yang didapat sangat pada proses grayscale dibutuhkan pada proses berikutnya untuk mempermudah proses selanjutnya hasil.





**Gambar 4. 3** Gambar setelah di grayscale

### **4.3. Penghapusan latar belakang**

Prose penghapusan latar belakang bertujuan untuk menentukan objek pada proses selanjutnya, proses ini sangat penting dalam tugas akhir ini, pada prose penghapusan latar belakang hasil gambar dimasukan pada fungsi yang ada.

Proses penghapusan latar belakang dilakukan dengan cara menumpuk beberapa frame yang ada, dimana banyaknya frame yang digunakan pada tugas akhir ini adalah sebanyak 500 frame, setelah itu frame – frame tersebut di tumpuk dan di lakukan proses pencarian rata- rata frame, sehingga nilai yang sama akan terhapus dan nilai yang berbeda akan tetap ada dan di lanjutkan pada proses berikutnya.

Hasil penghapusan latar belakang di dapatkan noise yang cukup banyak sehingga pada proses berikutnya dilakukan prose penghapusan noise, dapat dilihat pada gambar 4.5 dimana noise yang terdapat



**Gambar 4. 4** Hasil penghapusan latar belakang

#### **4.4. Penghapusan noise**

Pada proses penghapusan noise proses yang dilakukan dengan cara melakukan penghapusan noise yang ada pada proses sebelumnya dimana proses ini sendiri dilakukan dengan beberapa tahapan

##### **4.4.1 bluring**

Pada proses bluring gambar dari hasil sebelumnya yang berupa gambar biner dilakukan proses smotting dimana proses tersebut menggunakan medianblur, hasil dari median blur ialah biner yang ada dilakukan penyebaran kebeberapa sisi di sekitarnya dan sehingga gambar yang di hasilkan menjadi *smooth* dan hasil dari proses ini belum memenuhi kriteria pada proses berikutnya sehingga perlu dilakukan penghapusan noise lagi, pada proses bluring gambar yang berlubang mulai tertutup oleh bit-bit yang ada di sampingnya.



**Gambar 4. 5 Citra setelah bluring**

#### **4.4.1 Thresholding**

Pada proses thresholding gambar yang sebelumnya bergambar grayscale yang telah di blur dilakukan thresholding pada gambar yang nanti gambar tersebut akan bergambar biner, dimana hasil pada gambar yang telah di thresholding sudah memiliki noise yang lebih sedikit dari gambar sebelumnya, setelah gambar di thresholding gambar akan dilanjutkan dengan morfologi filter



**Gambar 4. 6** Citra setelah threshold

#### **4.4.2 Open filter**

Pada proses sebelumnya telah dilakukan penghapusan noise tetapi masih belum sempurna sehingga dilakukan kembali pada proses ini, pada proses ini dilakukan penghapusan filter dengan cara open filter, pada proses ini noise dihapus dengan cara mengkonvolusikan gambar dengan kernel yang ada dimana setiap hit akan di konvolusi dengan kernel, hasil ini belum juga sesuai untuk proses berikutnya

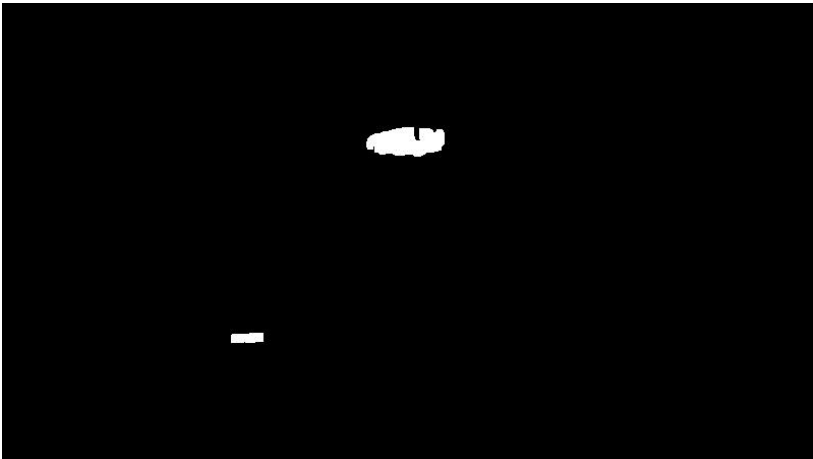
Pada proses ini hasil yang dihasilkan belum cukup bagus sehingga harus dilakukan proses fitering lagi untuk memperbaiki gambar.

#### **4.4.3 Erosi filter**

Pada proses sebelumnya telah dilakukan penghapusan noise tetapi masih belum sempurna sehingga dilakukan kembali pada proses ini, pada proses ini dilakukan penghapusan filter dengan cara erosi

filter, pada proses ini noise dihapus dengan cara mengkonvolusikan gambar dengan kernel yang ada dimana setiap hit akan di konvolusi dengan kernel, hasil ini belum juga sesuai untuk prose berikutnya

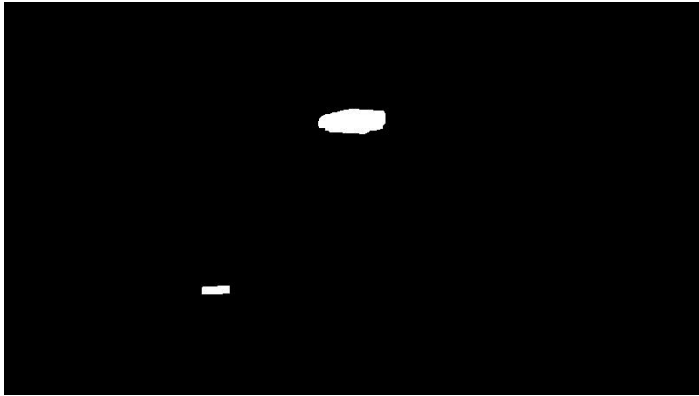
Pada proses ini hasil yang dihasilkan belum cukup bagus sehingga harus dilakukan prose fitering lagi untuk memeperbaiki gambar.



**Gambar 4. 7** Citra setelah open filter

#### **4.4.4 Close filter**

Pada prose close filter hasil yang di dapat masih memiliki noise tetapi hasil ini sudah cukup untuk dimasukan ke proses selanjutnya pada proses selanjutnya, prose ini dapat dilakukan dengan cara dilakukan dilatasi dan lalu dilakuka proses erosi



**Gambar 4. 8** Citra setelah Close filter

#### **4.4.5 Dilatasi filter**

Pada prose dilatasi filter hasil yang di dapat masih memiliki noise tetapi hasil ini sudah cukup untuk dimasukan ke proses selanjutnya pada proses selanjutnya, prose ini dapat dilakukan dengan tujuan engmbalikan jumlah pixel yang hilang dari proses erosi

#### **4.5. Pemotongan gambar**

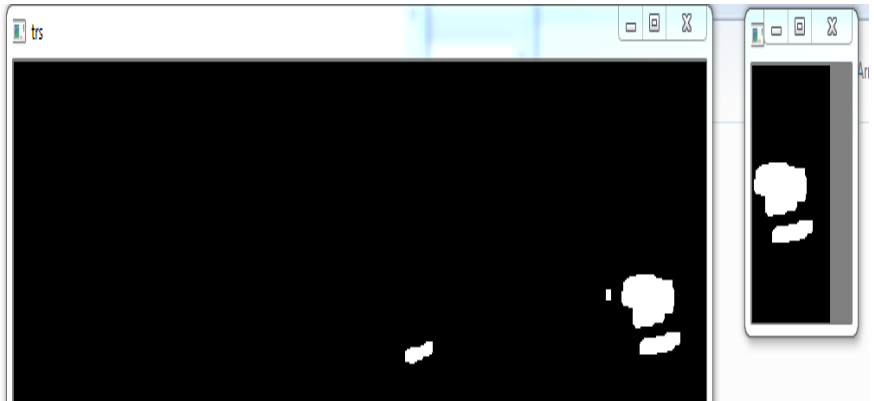
Pada prose segmentasi gambar gambar yang sudah ada secara utuh diambil ptongan gambar dimana kendaraan itu akan keluar, hal ini dilakukan agar proses yang ada pada prosesor tidak terlalu banyak dan terbebani, hasil dari pemotongan gambar membuat objek yang mau dideteksi lebih terarah dan lebih muda.

Pada proses ini ada dua lokasi gambar yang di potong sehingga prose ini diapat dilakukan, proses ini dilakukan dengan memanggil alamat array yang ada pada gambar dan dan disimpan pada variable array baru.

Array baru yang ada ditampilkan kembali menjadi gambar biner baru, gambar baru ini sudah memenuhi syarat untuk proses perhitungan, gambar dibagi menjadi 2 dapat dilihat pada Gambar 4.9 dan 4.10 dimana gambar yang di potong dan gambar sebelum peomotongan di perlihatkan

Proses tersebut dapat dilakukan dengan cara pemanggilan array

yang dapat dilihat pada gambar 4.11 dimana array tersebut hanya dipanggil dan dimasukkan ke variable baru untuk di ubah menjadi gambar kembali.



**Gambar 4. 9** Citra baru bagian atas



**Gambar 4. 10** Citra baru bagian bawah

```
bresn = res[250:400,0:100]  
aresn = res[62:200,700:790]
```

**Gambar 4. 11** pemnaggilan arrary sebagai gambar baru

#### **4.6. Aktif kontur**

Pada proses aktif kontur proses yang dilakukan dengan cara mencari kontur biner dan melakukan inisialisasi pada kontur yang ditemukan.

Dikarenakan hasil dari pencarian kontur bersifat acak maka diperlukan cara untuk menanggulangi hal tersebut, cara tersebut adalah dengan melakukan sorting pada kontur.

Kontur yang ditemukan ditandai dengan angka, angka tersebut adalah nilai yang menandakan urutan kontur, urutan kontur di urutkan berdasarkan jarak antara kontur dengan jarak menuju keluar gambar, hal ini terjadi pada gambar yang sudah di potong pada proses sebelumnya.

Hasil dari proses ini cukup memuaskan dikarenakan hasil kontur cukup bagus walaupun memiliki beberapa noise yang mengganggu sehingga proses berikutnya terdapat sedikit gangguan.

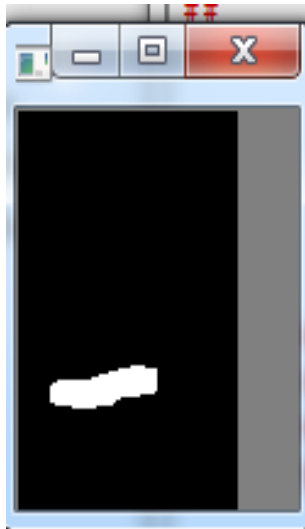




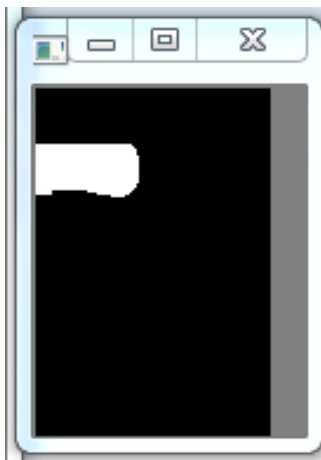
**Gambar 4. 12** Kontur citra atas



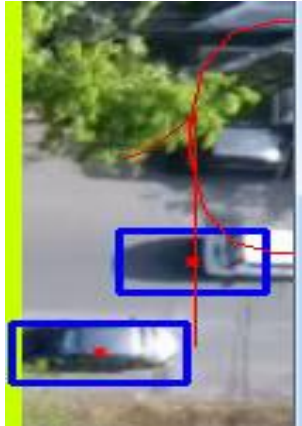
**Gambar 4. 13** Kontur citra bawah.



**Gambar 4. 14** Citra kontur atas dalam biner



**Gambar 4. 15** Citra kontur bawah dalam biner



**Gambar 4. 16** Citra sorting atas



**Gambar 4. 17** Citra sorting bawah

Dari hasil sorting dapat dilihat bahwa kontur yang mendekati titik keluar atau menuju keluar gambar maka akan mendapatkan nilai terkecil dan kontur yang lebih jauh akan mendapatkan nilai terbesar oleh karena itu nilai kontur bias digunakan untuk melakukan perhitungan kendaraan

asalkan tidak ada noise yang mengganggu.

#### **4.7. Pembuatan daerah hitung**

Pada proses perhitungan kendaraan dilakukan penghitungan kendaraan saat kontur melewati daerah poenghitung, pada awalnya daerah penghitung adalah sebuah garis yang nantinya akan dilalui tetapi karena ada lompatan pada kontur sehingga tidak memungkinkan melakukan perhitungan maka dilakukan pemanjangan daerah perhitungan, diamana daerah ini diperpanjang mencapai 4 pixel, begitu juga dengan daerah penghitungan bawah yang juga diperpanjang hingga 4 pixel



**Gambar 4. 18** Citra garis hitung atas ditunjukkan oleh warna hijau



**Gambar 4. 19** Citra garis hitung bawah ditunjukkan oleh warna hijau

#### **4.8. Pelebaran titik hitung**

Pada proses sebelumnya daerah hitung belum efektif untuk melakukan perhitungan karena titik hitung pada kontur mengalami loncatan maka dari pada itu dilakukan pelebaran titik hitung titik hitung dilebarkan sebanyak 7 pixel, begitu juga dengan titik hitung pada kontur bawah, titik hitung pada kontur bawah juga dilebarkan sampai 7 pixel.

Hal ini cukup memudahkan pada proses selanjutnya yaitu proses hitung,

#### **4.9. Proses hitung**

Pada proses perhitungan jumlah kendaraan dihitung dan diklasifikasi berdasarkan jumlah luas pixel yang ada pada kontur tersebut, kontur akan di klasifikasi berdasarkan variable yang sudah ditentukan.

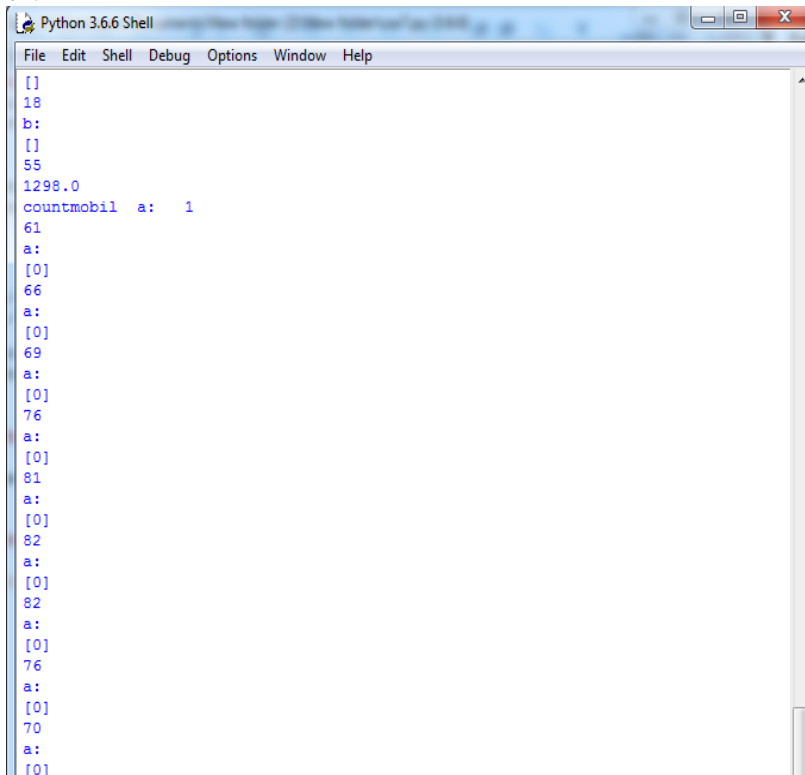
Proses ini dilakukan dengan cara melakukan perhitungan saat daerah hitung dan titik hitung bertemu, saat daerah hitung dan titik hitung bertemu nilai counter pada perhitungan akan ditambahkan dan akan disimpan data kontur yang di inisialisasi sehingga tidak terdapat perhitungan dua kali

Pada proses ini biasanya komputer yang saya gunakan mengalami

pelambatan proses atau lag sehingga gambar terasa lambat.

Pada proses ini komputer atau alat yang digunakan mengalami upaya lebih saat melakukan proses, sehingga proses penampilan gambar menjadi agak lambat pada komputer yang saya gunakan, hal ini dikarenakan proses perhitungan dilakukan seluruhnya oleh prosesor.

Pada gambar 4.17 diperlihatkan proses hitung yang ada pada shell python.

A screenshot of a Python 3.6.6 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main area shows a list of numbers from 18 to 70, with a variable 'countmobil' assigned the value 1. The numbers are displayed in a light blue font, and the variable assignment is in a darker blue font. The window has a standard Windows-style title bar and window controls (minimize, maximize, close) in the top right corner.

```
Python 3.6.6 Shell
File Edit Shell Debug Options Window Help
[ ]
18
b:
[ ]
55
1298.0
countmobil a: 1
61
a:
[0]
66
a:
[0]
69
a:
[0]
76
a:
[0]
81
a:
[0]
82
a:
[0]
82
a:
[0]
76
a:
[0]
70
a:
[0]
```

**Gambar 4. 20**Proses hitung yang dilakukan pada shell python

#### 4.10. Hasil hitung

pada hasil perhitungan yang ditunjukkan oleh table 4.1 ditunjukan nilai akurasi pada program mencapai 92.04% dan pada table 4.2 ditunjukan hasil

92.26% pada table 4.3 ditunjukkan hasil akurasi 91.28% pada table 4.4 ditunjukkan hasil 93.12%

hasil perhitungan menggunakan rumusan presisi dengan rumusan sebagai berikut

$$akurasi(\%) = \frac{Tp+Tn}{Tp+Tn+Fp+Fn} \times 100 \dots \dots (5)$$

- Tp = kendaraan lewat terhitung
- Tn = bukan kendaraan tidak terhitung
- Fp = bukan kendaraan terhitung
- Fn = kendaraan tidak terhitung

Dan hasil perhitungan error dengan rumusan sebagai berikut:

$$|Error| = \text{nilai hitung manual} - \text{nilai hasil program} \dots \dots (6)$$

Data perhitungan didapatkan dengan cara melakukan perhitungan secara manual terhadap hasil video yang telah diproses oleh program sebelumnya dan di bandingkan dengan hasil perhitungan program.

Maka didapatkan hasil sebagai berikut:

**Table 4.1** Table akurasi kendaraan L atas

	Menit 1	Menit 2	Menit 3	Menit 4	Menit 5	Rata-Rata akurasi
Tp	5	5	11	8	9	
Tn	5	11	10	3	6	

Fp	0	4	2	0	0	
Fn	0	2	1	0	0	
Akurasi	100 %	72.72%	87.5%	100%	100%	92.04%

**Table 4.2** Table akurasi kendaraan M atas

	Menit 1	Menit 2	Menit 3	Menit 4	Menit 5	Rata-Rata akurasi
Tp	5	12	10	3	6	
Tn	5	6	12	8	9	
Fp	0	2	1	0	0	
Fn	0	1	0	0	0	
akurasi	100%	85.7%	95.6%	100%	100%	96.26%



**Table 4.3** Table akurasi kendaraan L bawah

	Menit 1	Menit 2	Menit 3	Menit 4	Menit 5	Rata-Rata akurasi
Tp	7	11	5	8	8	
Tn	7	16	5	5	13	
Fp	0	1	1	0	5	
Fn	0	1	0	0	3	
akurasi	100%	93.1%	90.9%	100%	72.4%	91.28%

**Table 4.4** Table akurasi kendaraan M bawah

	Menit 1	Menit 2	Menit 3	Menit 4	Menit 5	Rata-Rata akurasi
Tp	2	14	5	5	13	

Tn	2	11	5	8	11	
Fp	0	1	0	0	5	
Fn	0	2	0	0	0	
akurasi	100%	89.2%	100%	100%	82.7%	93.12%

**Table 4.5** Table perhitungan kendaraan L atas

	Menit 1	Menit 2	Menit 3	Menit 4	Menit 5	Jumlah
Kendaraan L atas( manual)	5	5	11	9	9	39
Kendaraan L (hasil program)	5	9	12	8	9	43
Error	0	4	1	1	0	6

**Table 4.6** Table perhitungan kendaraan M atas

	Menit 1	Menit 2	Menit 3	Menit 4	Menit 5	Jumlah
Kendaraan M atas ( manual)	5	13	11	2	6	37
Kendaraan M atas( hasil program)	5	14	11	3	6	39
Error	0	1	0	1	0	2

**Table 4.7** Table perhitungan kendaraan L bawah

	Menit 1	Menit 2	Menit 3	Menit 4	Menit 5	Jumlah
Kendaraan L bawah( manual)	7	12	5	7	11	42
Kendaraan L bawah( hasil program)	7	12	5	8	13	45
Error	0	0	0	1	2	3

**Table 4.8** Table perhitungan M bawah

	Menit 1	Menit 2	Menit 3	Menit 4	Menit 5	Jumlah
Kendaraan M bawah( manual)	2	15	5	5	14	41
Kendaraan M bawah( hasil program)	2	15	5	5	13	40
Error	0	0	0	0	1	1

#### **4.11. Pembuatan file excel**

Pada proses pembuatan file excel pada tugas akhir ini dilakukan dengan cara menyimpan semua hasil pada list yang ada sehingga data dapat disimpan, setelah penyimpanan data hasil yang di dapat diubah ke format excel menggunakan library yang ada pada python.

Pada pembuatan file file diberi nama dengan waktu yang ada pada hari pembuatan.

Pada proses pembuatan langkah yang harus dilalui adalah dengan membuat file dilanjutkan dengan membuat workbook dan mengisi data yang ada pada list yang sudah disimpan ke table yang ada pada excel

#### **4.10 Proses *Internet Of Things (IOT)***

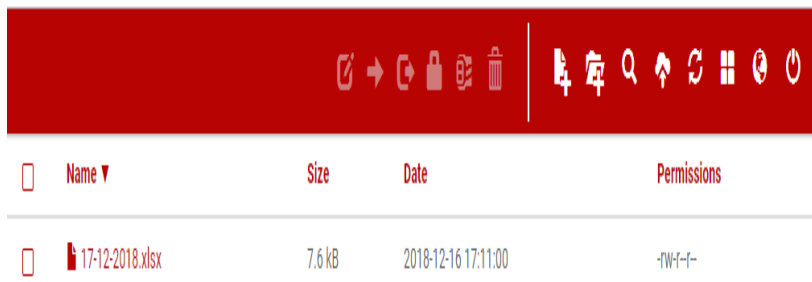
Pada saat pengiriman file keserver di tugas akhir ini, digunakan metode ftp yang dimana saat dilakukan pengiriman file dengan ftp, ftp melakukan handshake ke server, lalu dilakukan pencarian direktori atau lokasi pengugahan file ke server setelah itu data dikirim dan didapatkan hasil yang sesuai dengan rencana.

Setelah data suda ada pada server data dilakukan update dengan cara

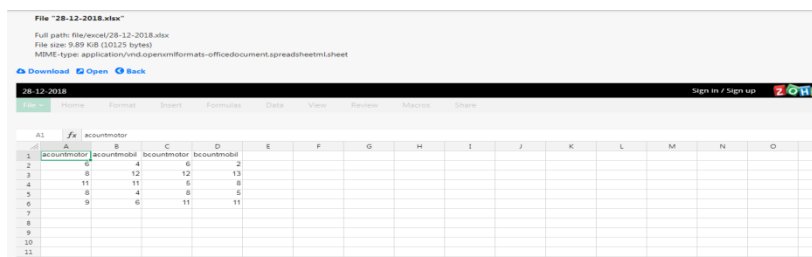
melakukan listing data pada direktori, jika ada data yang sama data yang lama akan dihapus dan data yang baru akan di unggah.

Hasil dari proses ini sudah mencapai hasil yang bagus dan dimana data dapat dapat terkirim.





Pada saat yang bersamaan juga dilakukan pengiriman data ke database, lalu pada perangkat pengguna lain data yang ada pada databse dapat langsung dilihat, data tersebut berupa hasil perhitungan yang dilakukan secara realtime, proses pengiriman data tersebut dilakukan dengan cara *client-to-server*, pada perangkat pengguna, pengguna juga dapat melihat hasil web atau hasil dari laporan yang berformat excel pada perangkat pengguna.



**Gambar 4. 21** Tampilan direktori pada server



**Gambar 4. 22** Tampilan web

Realtime		Web	
atas			
	0		
	0		
bawah			
	0		
	0		

**Gambar 4. 23** Tampilan realtime

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan percobaan yang telah dilakukan pada pelaksanaan tugas akhir ini didapat beberapa kesimpulan sebagai berikut:

- a. Noise yang ada pada proses penghapusan noise tidak seluruhnya dapat terhapus, dikarenakan pada saat pengambilan citra drone yang digunakan tidak stabil sehingga menimbulkan banyak bentuk noise yang beraneka ragam.
- b. Proses klasifikasi kendaraan menggunakan metode luas pixel pada tugas akhir ini ternyata mampu melakukan proses klasifikasi.

#### **5.2. Saran**

Sebagai sarana pengembangan Rancang Bangun Sistem Penghitung Jumlah Kendaraan Pada Jalan Raya Berbasis Internet Of Things, maka terdapat beberapa saran dari penulis berdasarkan hasil yang diperoleh saat percobaan, yaitu sebagai berikut:

- a. Karena pemrosesan data gambar membutuhkan waktu yang lama maka seharusnya digunakan komputer dengan spesifikasi yang cukup tinggi
- b. Menambahkan algoritma penghapusan noise sebelum dilakukan proses morfologi filter

.....*Halaman ini sengaja dikosongkan*.....



## DAFTAR PUSTAKA

- [1] tribunnews surabaya. (2015, June) tribunnews surabaya. [Online].  
<http://surabaya.tribunnews.com>
- [2] dishub surabaya. (2019, Jan.) dishub surabaya. [Online].  
<http://sits.dishub.surabaya.go.id>
- [3] D. Abram I. Culjak, *A brief introduction to OpenCV.*: Proceedings of the 35th International Convention MIPRO, 2012.
- [4] R. Szeliski, *Computer Vision: Algorithms and Applications.*: Springer Science & Business Media, 2010.
- [5] OpenCV Team. (2018, November) opencv.org. [Online].  
[www.opencv.org](http://www.opencv.org)
- [6] K. V. Tarun Kumar, "A Theory Based on Conversion of RGB image to Gray image," *International Journal of Computer Applications* , 2018.
- [7] Sin Hoong Teoh, "Median Filtering Frameworks for Reducing Impulse Noise," *International Journal of Future Computer and Communication*, 2012.
- [8] University Of Auckland. (2018, desember)  
<https://www.cs.auckland.ac.nz>. [Online].  
<https://www.cs.auckland.ac.nz>
- [9] Zhiguo Hong,Minyong Shi P. Wei, Z. Hong, "Performance Analysis of HTTP and FTP Based on," *IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 2016.
- [10] Python Developer Team. (2018, november) python.org. [Online].  
[www.python.org](http://www.python.org)
- [11] S.T.,M.Sc. Priyanto hidayatullah, *pengolahan citra digital teori dan aplikasi.*: penerbit informatika, 2017.
- [12] Andi Muhammad Ali Mahdi Akbar, Ed., *Traffic IP Camera untuk Menghitung*. Surabaya: ITS Journal, 2016.
- [13] Robert Fisher A. Walker and E. Wolfart. (2018, Dec.) HIPR2.  
[Online]. <http://homepages.inf.ed.ac.uk>
- [14] Z. Zivkovic, "Efficient adaptive density estimation per image pixel for task of background subtraction," *Pattern Recognit. Lett*, 2006.
- [15] John McNamara. (2018, Dec.) xlsxwriter. [Online].  
[xlsxwriter.readthedocs.io](http://xlsxwriter.readthedocs.io)

- [16] Oya Suryana, "Server dan Web Server," *research gate*, agustus 2018.
- [17] (2018) wikipedia. [Online]. [https://id.wikipedia.org/wiki/Batch\\_file](https://id.wikipedia.org/wiki/Batch_file)
- [18] Ahmed Elgammal, *Background Subtraction: Theory and Practice.*: Morgan & Claypool, 2014.
- [19] Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, 2014.

## Lampiran 1

```
import cv2
import numpy as np
import time
import xlswriter
import os
import pyrebase
#from firebasedata import LiveData
#from multiprocessing import Process
import threading
#####
#####
config = {
    "apiKey": "AIzaSyDjZCpc8mwU_P2nmXRMx5p9MU",
    "authDomain": "cacon-154541941.firebaseio.com",
    "databaseURL": "https://carcon-1546941.firebaseio.com",
    "storageBucket": "carcon-1413641.appspot.com"
}
db = pyrebase.initialize_app(config).database()
#live = LiveData(firebase, 'hasil/atas2')

#####
#####
def lapor(tam,tac,tbm,tbc,ntam,ntac,ntbm,ntbc,t_f):

    workbook = xlswriter.Workbook(ex)
    worksheet = workbook.add_worksheet()
    worksheet2 = workbook.add_worksheet()
    worksheet.write('A1',      str(ntam))
    worksheet.write('B1',      str(ntac))
    worksheet.write('C1',      str(ntbm))
    worksheet.write('D1',      str(ntbc))
    xam = len(tam)+1
    xac = len(tac)+1
    xbm = len(tbm)+1
    xbc = len(tbc)+1
    # menulis ket
```

```

worksheet.write_column('A2', tam)
worksheet.write_column('B2', tac)
worksheet.write_column('C2', tbm)
worksheet.write_column('D2', tbc)

```

**# membuat diagram**

```

chart1 = workbook.add_chart({'type': 'line'})
chart1.set_x_axis({
    'name': 'roda 2 atas',
    'name_font': {'size': 14, 'bold': True},
    'num_font': {'italic': True },})
chart2 = workbook.add_chart({'type': 'line'})
chart2.set_x_axis({
    'name': 'roda lebih dari 2 atas',
    'name_font': {'size': 14, 'bold': True},
    'num_font': {'italic': True },})
chart3 = workbook.add_chart({'type': 'line'})
chart3.set_x_axis({
    'name': 'roda 2 bawah ',
    'name_font': {'size': 14, 'bold': True},
    'num_font': {'italic': True },})
chart4 = workbook.add_chart({'type': 'line'})
chart4.set_x_axis({
    'name': 'roda lebih dari 2 bawah',
    'name_font': {'size': 14, 'bold': True},
    'num_font': {'italic': True },})

```

**# memasukkan data diagram**

```

chart1.add_series({'values': '=Sheet1!$A$2:$A$' + str(xam) })
chart2.add_series({'values': '=Sheet1!$B$2:$B$' + str(xac)})
chart3.add_series({'values': '=Sheet1!$C$2:$C$' + str(xbm)})
chart4.add_series({'values': '=Sheet1!$D$2:$D$' + str(xbc)})

```

**# memasukkan diagram ke worksheet**

```

worksheet2.insert_chart('A1', chart1)
worksheet2.insert_chart('A16', chart2)
worksheet2.insert_chart('A31', chart3)
worksheet2.insert_chart('A45', chart4)

```

```
workbook.close()
```

```
#####  
#####
```

```
import threading
```

```
##def gfg():
```

```
##    tx = open("log.txt","r")  
##    dae = tx.read().splitlines()  
##    ll = dae[-1]  
##    tx.close()  
##    return ll
```

```
def fire(atas1,atas2,bawah1,bawah2):
```

```
    db.child("/hasil/atas2").set(atas1)  
    db.child("/hasil/atas2l").set(atas2)  
    db.child("/hasil/bawah2").set(bawah1)  
    db.child("/hasil/bawah2l").set(bawah2)
```

```
#####  
#####
```

```
bs = cv2.createBackgroundSubtractorMOG2( 100 )
```

```
img = cv2.VideoCapture("zx6.mp4")
```

```
opf = np.ones((7,7),np.uint8)
```

```
clf = np.ones((1,11),np.uint8)
```

```
kernelerode = np.ones((5,3),np.uint8)
```

```
kerneldil = np.ones((3,9),np.uint8)
```

```
acountmotor = 0
```

```
bcountmotor = 0
```

```
acountmobil = 0
```

```
bcountmobil = 0
```

```
motora=[]
```

```
motorb=[]
```

```
mobila=[]
```

```

mobilb=[]
motoras=0
motorbs=0
mobilas=0
mobilbs=0
line = np.array(range(48,52))
line0 = np.array(range(48,52))
xs0=[]
xs1 =[]
xs2=[]
vline = np.zeros(128,np.uint8)
a = []
b = []
tmp_w = 0
tmp_w2 = 0
tmp_w3 = 0
tmp_w4 = 0
tmp_hari = 0
hari = 0

```

**while 1:**

```

    ret, frame = img.read()
    hari = time.localtime()[2]
    res1 = cv2.resize(frame, (800,449))

```

```

    hsv = cv2.cvtColor(res1, cv2.COLOR_BGR2HSV)
    # pengubhan gambar bgr menjadi gray
    grey = cv2.cvtColor(res1, cv2.COLOR_BGR2GRAY)
    #melakukan pengecilan gambar

```

```

    #menjalankan subtraction background
    bsub = bs.apply(grey)
    #median gaussian blur
    blur = cv2.medianBlur(bsub,3)
    ret2, th = cv2.threshold(blur, 125, 255, cv2.THRESH_BINARY)
    # filter morphology

```

```

opening = cv2.morphologyEx(th, cv2.MORPH_OPEN, opf)
erode = cv2.erode(opening,kernelerode,iterations = 1)
closing = cv2.morphologyEx(erode, cv2.MORPH_CLOSE, clf)
dil = cv2.dilate(closing,kerneldil,iterations = 1)

#res = cv2.dilate(res, kernel)
#gambar garis
line1 = 50
cv2.line(res1, (line1, 255), (line1, 372), (255, 255, 0), 5)

#gambar garis2
line2 = 750
cv2.line(res1, (line2, 60), (line2, 184), (5, 255, 200), 5)

bresn = dil[250:400,0:100]
aresn = dil[62:200,700:790]

##      cv2.putText(res1, gfg() , (28, 32),
cv2.FONT_HERSHEY_SIMPLEX,
##                                0.8, (237, 280, 63), 3)
      cv2.putText(res1, 'roda 2 :' + str(acountmotor) , (340, 130),
cv2.FONT_HERSHEY_SIMPLEX,
                                0.8, (200, 72, 63), 2)
      cv2.putText(res1, 'roda>2 :' + str(acountmobil) , (340, 170),
cv2.FONT_HERSHEY_SIMPLEX,
                                0.8, (200, 72, 63), 2)

      cv2.putText(res1, 'roda 2 :' + str(bcountmotor) , (340, 270),
cv2.FONT_HERSHEY_SIMPLEX,
                                0.8, (0, 0, 200), 2)
      cv2.putText(res1, 'roda>2 :' + str(bcountmobil) , (340, 310),
cv2.FONT_HERSHEY_SIMPLEX,
                                0.8, (0, 0, 243), 2)

_, contours,hierarchy1 = cv2.findContours(aresn,
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
_, bcontours,hierarchy2 = cv2.findContours(bresn,

```

```
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
```

```
    contours.sort(key=lambda contours:  
(cv2.boundingRect(contours)[0]+ cv2.boundingRect(contours)[2]) ,  
reverse = True) #melakukan sorting dari kanan ke kiri  
    bcontours.sort(key=lambda contours:  
(cv2.boundingRect(contours)[0]) , reverse = False) #melakukan sorting  
dari kiri ke kanan
```

```
#print(len(contours))
```

```
maxarea = 50000
```

```
#####  
#####-----ATAS-----#####  
#####
```

```
for i in range(len(contours)):
```

```
    area = cv2.contourArea(contours[i])
```

```
    if 100 < area < 100000:
```

```
        x, y, w, h = cv2.boundingRect(contours[i])
```

```
        cv2.rectangle(res1, (x+700, y+62), (x + 700+w, y +  
62+h), (255, 0, 0), 2)
```

```
        cv2.putText(res1, str(i) , (x + 710, y + 72),
```



```

cv2.FONT_HERSHEY_SIMPLEX,
                    5, (0, 0, 255), 1)

cv2.drawMarker(res1, (x+700+int(w/2),
y+62+int(h/2)), (0, 0, 255), cv2.MARKER_STAR, markerSize=5,
thickness=1,

                    line_type=cv2.LINE_AA)
x = np.array(range(x+w-8,x+w))

z = any(elem in x for elem in line)

print(w)
if i not in a:
    if z:

        zs = any(elem in xs0 for elem in x)
        xs0 = x

        if zs == False :

            print(area)
            a.append(i)
            if area > 70 and area < 500:
                accountmotor +=1
                motoras +=1
                print("countmotor   a:  " + "
"+ str(accountmotor))

            elif area >= 450:
                accountmobil +=1
                mobilas +=1
                print("countmobil   a:  " + " "+
str(accountmobil))

```

```

        continue
    else:
        if w < 20:
            imax = max(a)
            a.remove(imax)
        xs0=[]
        print("a:")
        print(a)

#####
###-----BAWAH-----#####
#####

    for j in range(len(bcontours)):

        area1 = cv2.contourArea(bcontours[j])

        if 100 < area1 < 100000:

            x1, y1, w1, h1 = cv2.boundingRect(bcontours[j])

            cv2.rectangle(res1, (x1, y1+250), (x1+w1, y1+ 250 +
h1), (255, 0, 0), 2)

            cv2.putText(res1, str(j) , (x1 +10, y1+250  ),
cv2.FONT_HERSHEY_SIMPLEX,
                    5, (0, 0, 255), 1)

            cv2.drawMarker(res1, (x1+int(w1/2),
y1+250+int(h1/2)), (0, 0, 255), cv2.MARKER_STAR, markerSize=5,
thickness=1,

                        line_type=cv2.LINE_AA)

```

```

x1 = np.array(range(x1,x1+7))

z = any(elem in x1 for elem in line0)
#print(hierarchy)
print(w1)
#print("area1"+" "+str(area1))
if j not in b:
    if z:

        zs = any(elem in xs1 for elem in x1)
        xs1 = x1

    if zs == False :

        print(area1)
        b.append(j)
        if area1 > 70 and area1 < 500:
            bcountmotor +=1
            motorbs +=1
            print("countmotor   b:  " + "
" + str(bcountmotor))

            elif area1 >= 450:
                bcountmobil +=1
                mobilbs +=1
                print("countmobil   b:  " + " "+
str(bcountmobil))

        continue
    else:
        if w1 < 20:
            imax = max(b)
            b.remove(imax)
xs1=[]

```

```

        print("b:")
        print(b)

    if len(contours) == 0:
        a.clear()
    if len(bcontours) == 0:
        b.clear()

    #cv2.imshow("frame", res)
    cv2.imshow("frae", res1)
    ##    cv2.imshow("trs", bsub)
    ##    cv2.imshow("res a", opening)
    ##    cv2.imshow("res b", bresn)

    mi = img.get(cv2.cv2.CAP_PROP_POS_MSEC) # pengambilan
waktu video
    tmp_w = int(mi/60000)
    tmp_w3 = int(mi/3000)
    #Process(target=,
args=(motoras,mobilas,motorbs,mobilbs)).start()
    timer = threading.Timer(1.0,
fire,args=(motoras,mobilas,motorbs,mobilbs))
    timer.start()

    #if tmp_w3 != tmp_w4 :
        #fire(motoras,mobilas,motorbs,mobilbs)
        #live.set_data('', motoras)
    if tmp_w != tmp_w2:

        tmp_time = time.localtime()
        times = time.strftime("%d-%m-%Y", time.gmtime())
        ex = str(times)+''.xlsx'

```

```

    motora.append(acountmotor)

    mobila.append(acountmobil)

    motorb.append(bcountmotor)

    mobilb.append(bcountmobil)

    lapor(motora,mobila,motorb,mobilb,'aRODA 2','aRODA 2
LEBIH','bRODA 2','bRODA 2 LEBIH',times)

    os.startfile('ftp.bat')

    bcountmobil = 0
    bcountmotor = 0
    acountmobil = 0
    acountmotor = 0

    if tmp_hari != hari:
        motora = []
        mototb = []
        mobila = []
        mobilb = []
    tmp_w4 = int(tmp_w3)
    tmp_w2 = int(tmp_w)
    tmp_hari = time.localtime()[2]

    if cv2.waitKey(1) & 0xFF == ord('q'):
        cv2.imwrite("gambar.jpg", res1)
        cv2.imwrite("bsub.jpg", bsub)
        cv2.imwrite("blur.jpg",blur)
        cv2.imwrite("th.jpg",th)
        cv2.imwrite("erode.jpg",erode)
        cv2.imwrite("dil.jpg",dil)
        cv2.imwrite("close.jpg",closing)
        cv2.imwrite("open.jpg",opening)
        cv2.imwrite("grey.jpg",grey)
        cv2.imwrite("grey.jpg",grey)

```

```
        cv2.imwrite("potong.jpg",aresn)
    break
aresn
img.release()
cv2.destroyAllWindows()
```

## Lampiran 2

```
from ftplib import FTP
import time
tx = open("log.txt", "a")
tmp_time = time.localtime()
times = time.strftime("%d-%m-%Y", time.gmtime())
error = "\n" + "error" + " " +
str(time.strftime("%H:%M %d-%m-%Y", time.localtime()))
success = "\n" + "success" + " " +
str(time.strftime("%H:%M %d-%m-%Y", time.localtime()))
tf = times+'.xlsx'
#login dan direktori

# pengecekan dan pengiriman
try:
    ftp = FTP('files.000webhost.com', 'carconlutf',
'0000xxxxx')
    ftp.cwd('public_html/file/excel/')
    file = tf
    ftp.storbinary('STOR '+file, open(file, 'rb'))
    tx.write(success)
    tx.close()
    print(success)
except :
    tx.write(error)
    tx.close()
    print(error)

#penutup
ftp.quit()
```

## **Biodata Penulis**



Akhmad Lutfi Rusidi lahir di Lamongan pada tanggal 27 Juli 1994. Penulis menyelesaikan pendidikan dasar di MI Murni Sunan Drajat, dilanjutkan pendidikan pendidikan tingkat menengah di SMP Negeri 1 Lamongan dan sekolah tingkat atas di SMA Negeri 1 Lamongan. Penulis memulai kehidupan perkuliahan pada tahun 2013 di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Email : lutfirusidi@gmail.com  
Hp/WA : 082140341979  
Facebook : lutfi rusidi  
Line : lutfirusidi