BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pada bagian ini akan diuraikan konsep system secara utuh. Tujuan dari bagian analisis system ini adalah untuk mengidentifikasi masalah, hambatan dan kebutuhan. Pada penelitian ini analisis sistem akan dibagi ke dalam 4 bagian yaitu, analisis masalah, deskripsi umum sistem, analisis non fungsional dan analisis fungsional.

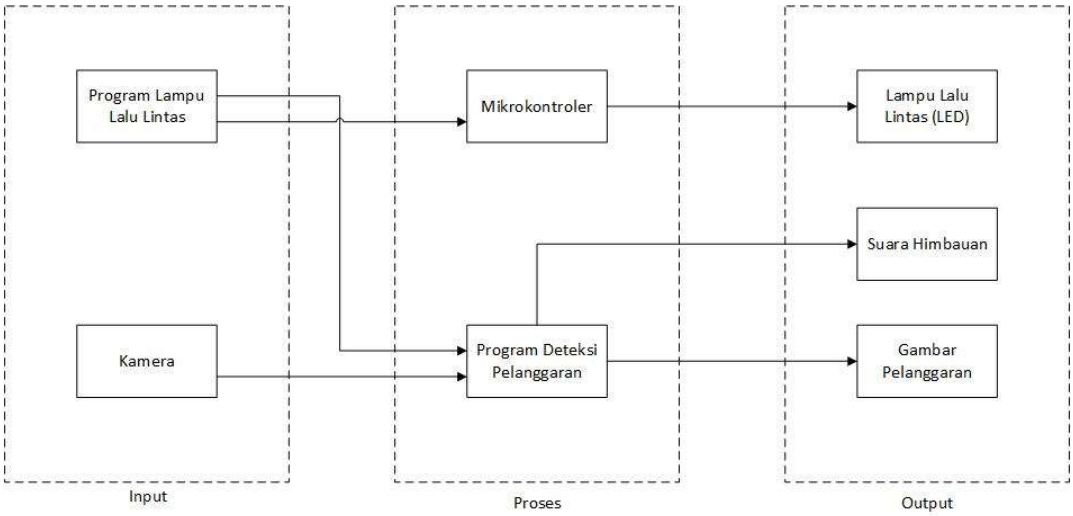
3.1.1 Analisis masalah

Setiap persimpangan jalan pada umumnya memiliki aturan yang sama, seperti dilarang berhenti di jalur belok kiri langsung dan dilarang berhenti di *zebra cross*. Saat ini hampir semua persimpangan yang ada di bandung memiliki kamera CCTV yang terus memantau keadaan persimpangan dari *control room* dishub Bandung. Namun pemantauan yang dilakukkan oleh operator di dalam *control room* masih dapat dikatakan manual. Operator memantau gambar yang ditangkap oleh cctv, ketika terlihat sebuah pelanggaran operator yang sedang bertugas langsung mengambil gambar kendaraan tersebut dan mendata pelanggaran apa yang dilanggar dengan cara mencatat nya di file *excel,* data dan gambar yang diambil oleh operator akan diberikan kepada kepolisian untuk ditinjau lebih lanjut apakah perlu ditindak atau tidak. Namun cara yang dilakukkan dishub masih kurang efisien karena masih ada pelanggaran yang tidak terpantau oleh operator dikarenakan terbatasnya perempatan yang bisa dipantau oleh operator.

Menjawab permasalahan tersebut dapat digunakan pengolahan citra dengan sistem lalu lintas yang terintegrasi dengan lampu lalu lintas yang ada di persimpangan jalan agar pelanggaran seperti pada zebra cross bisa terdeteksi dengan memantau setiap kendaraan yang ada di area perempatan yang melakukkan pelanggaran dan agar bisa mengurangi tenaga manusia yang memantau persimpangan, pada control room. Terintegrasinya lampu lalu lintas dengan deteksi pelanggaran akan dijelaskan pada poin 3.1.3 yaitu analisis solusi.

3.1.2 Deskripsi Umum Sistem

Berikut adalah gambaran alur sistem secara keseluruhan.



Gambar 3. 1 Sistem Keseluruhan

Sistem akan mendeteksi dan akan melakukan *tracking* objek kendaraan yang ada di area perempatan, dengan menggunakan pengolahan citra akan ditentukan area mana saja yang menjadi area terlarang untuk kendaraan yang melewati perempatan tersebut, dan pelanggaran ditentukan juga oleh status lampu yang ada pada program lampu lalu lintas, status lampu lalu lintas juga dikirim ke mikrokontroler untuk menyalakan lampu led sesuai dengan statusnya. Sedangkan hasil dari deteksi pelanggaran berupa himbauan suara dan gambar pelanggaran.

Berikut adalah uraian dari setiap komponen blok sistem pada Gambar 3.1.

1. Blok *Input*

Blok *input* memiliki dua komponen yaitu kamera dan program lampu lalu lintas yang berfungsi untuk mengirim status lalu lintas dan frame gambar untuk diproses.

a. Program Lampu Lalu Lintas

   Program lampu lalu lintas akan menentukan dan mengirimkan status lampu lalu lintas ke mikrokontroler dan program deteksi, guna untuk menyalakan lampu lalu lintas (led) dan menentukan pelanggaran di bagian proses deteksi.

b. Kamera

Kamera digunakan unuk mengambil frame secara terus menerus selama program deteksi berjalan.

2. Blok Proses

Terdapat dua komponen pada blok ini yaitu mikrokontroler yang dihubungkan dengan kabel serial dan program deteksi yang terhubung dengan kamera.

a. Mikrokontroler

Membaca status lampu dan mengendalikan lampu lalu lintas mana yang harus menyala sesuai dengan status.

b. Program Deteksi

Untuk mendeteksi dan *tracking* kendaraan yang melakukkan pelanggaran, akan dilakukkan beberapa langkah yaitu sebagai berikut.

- *Grayscaling*
- *Blurring*
- *Background Subtraction*
- *Thresholding*
- *Countour Detection*
- *Convex Hull dan Blob Detection*

Penjelasan deteksi atau tahapan deteksi akan dijelaskan lebih detailnya pada poin 3.1.3.2

3. Blok *Output*

Blok ini berisi tiga komponen yaitu lampu lalu lintas (LED), gambar dan suara himbauan

a. Lampu lalu lintas (LED)

Lampu yang ada adalah lampu berwarna merah, kuning dan hijau. Ketiga lampu tersebut akan menyala sesuai dengan status yang dikirimkan oleh program lampu lalu lintas, jika dibaca merah maka lampu led merah akan menyala, jika dibaca kuning maka lampu led kuning akan menyala dan jika hijau lampu led hijau akan menyala.

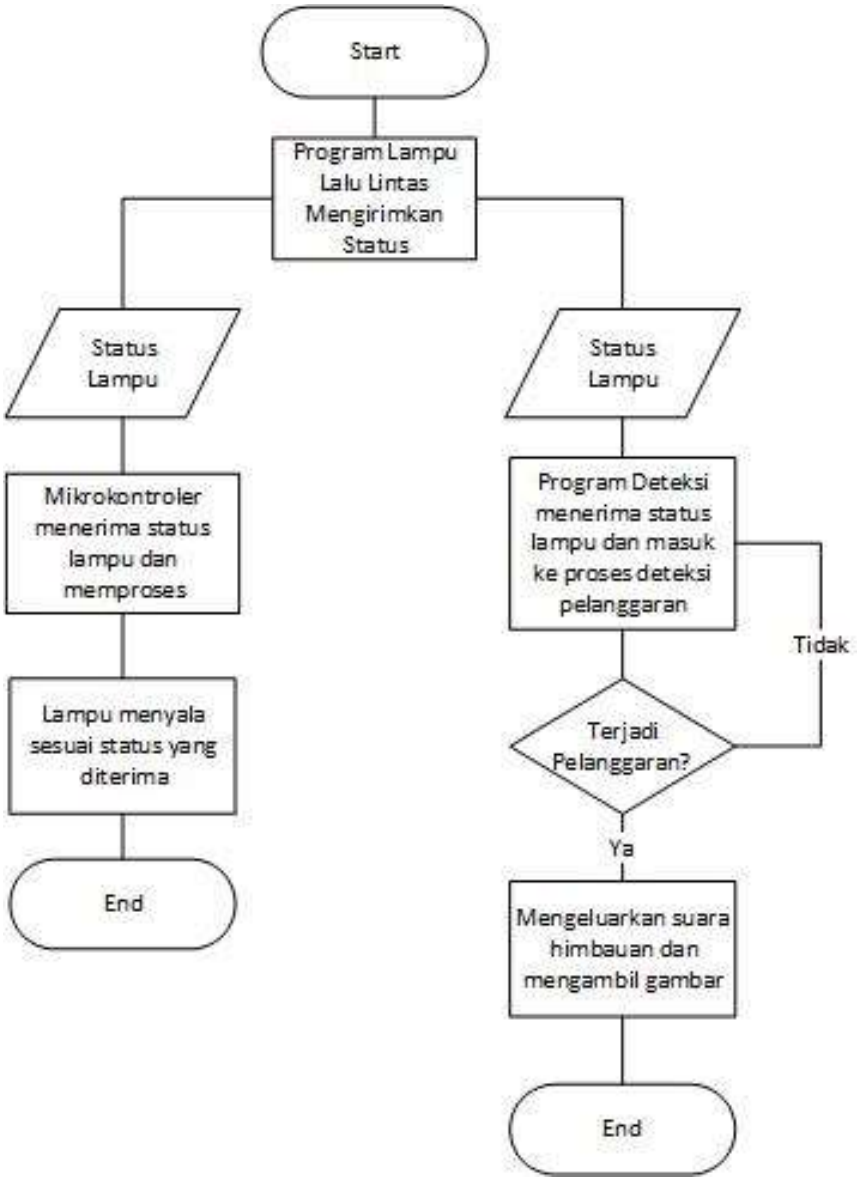
b. Suara Himbauan

Suara himbauan terdiri dari dua jenis, himbauan untuk jalur kiri dan himbauan untuk area *zebra cross*. Pada jalur kiri jika terjadi pelanggaran maka pengendara akan dihimbau untuk langsung belok ke kiri, sedangkan area *zebra cross* pengendara akan di himbau untuk belok ke kiri atau mundur.

c. Gambar Pelanggaran

Gambar pelanggaran merupakan foto kendaraan yang melanggar, pada jalur kiri foto akan diambil ketika kendaraan berhenti selama 5 detik. Sedangkan pada area *zebra cross* kendaraan akan difoto saat memasuki area tersebut pada lampu merah.

Ketika blok tersebut jika direpresentasikan dengan flowchart adalah sebagai berikut.

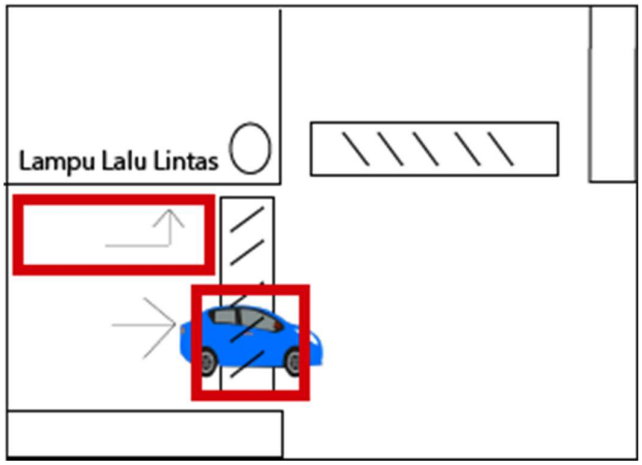


Gambar 3. 2 *Flow* Sistem Lampu Lalu Lintas Secara Umum

3.1.3 Analisis Peraturan

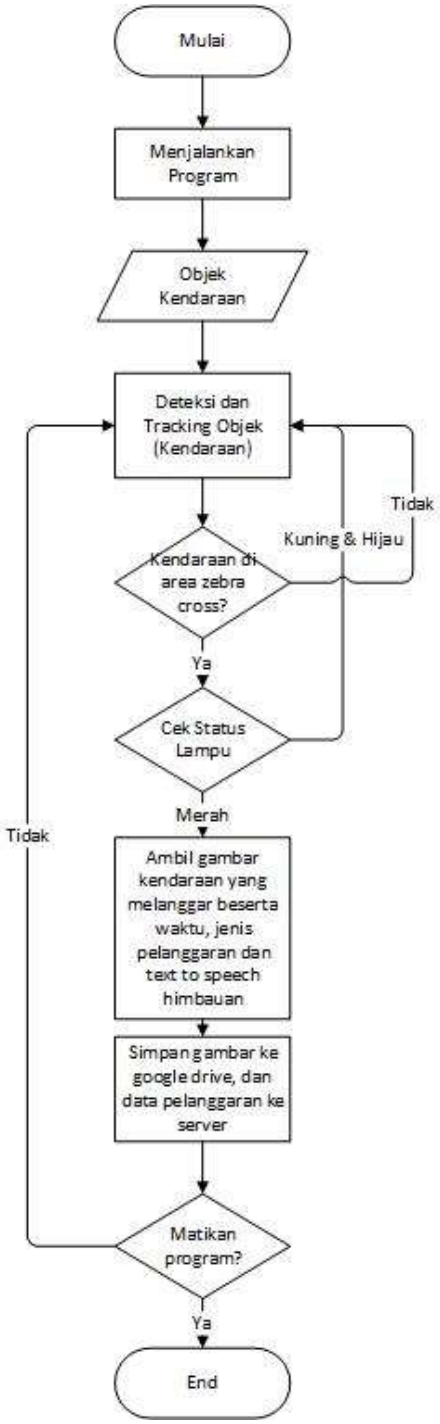
Pada bagian ini akan dibahas peraturan yang ada dan bagaimana pelanggaran tersebut akan dideteksi. Peraturan yang ada mengacu kepada UU 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, yang sudah dibahas pada BAB 2 poin 2.1.

1. Peraturan pada area zebra cross



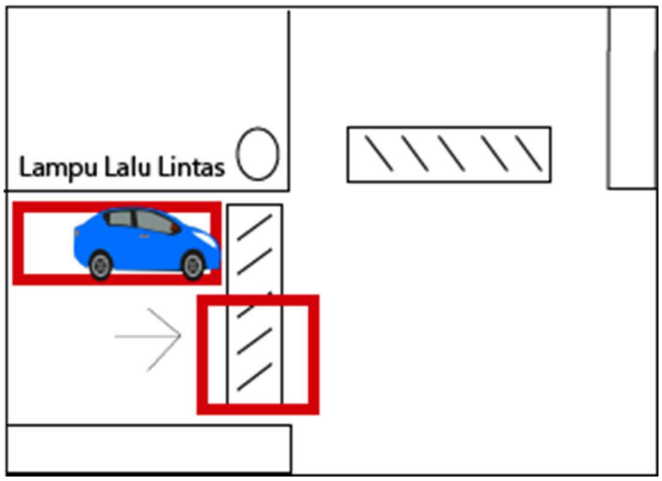
Gambar 3. 3 Pelanggaran 1 Zebra Cross

Saat status lampu menunjukan merah, maka area zebra cross menjadi area terlarang bagi kendaraan untuk berhenti. Jika ada kendaraan yang terdeteksi berhenti di area tersebut maka kamera akan mendeteksi, memperingatkan dengan speaker dan mengambil gambar kendaraan tersebut untuk disimpan beserta informasi lainnya. Berikut adalah *flowchart* deteksi pelanggaran pada *zebra cross* dapat dilihat pada gambar 3.4.



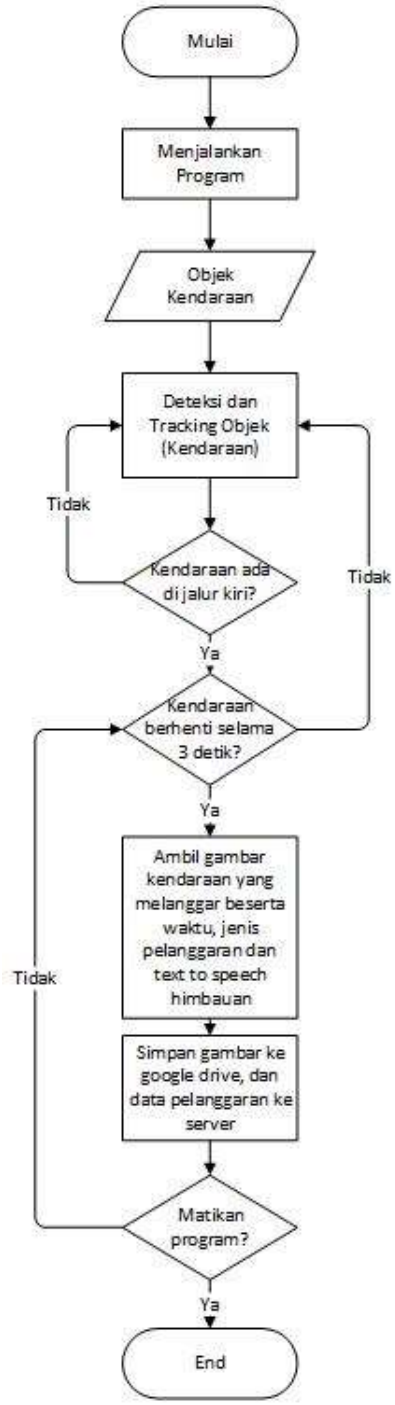
Gambar 3. 4 *Flowchart* Deteksi Pelanggaran pada *Zebra Cross*

2. Peraturan pada jalur belok kiri langsung



Gambar 3. 5 Pelanggaran 2 Jalur Belok Kiri Langsung

Jalur belok kiri langsung adalah jalur dimana seharusnya tidak ada satupun kendaraan yang berhenti, dengan tujuan untuk menghindari kemacetan bagi kendaraan yang bertujuan untuk belok ke kiri. Jika kendaraan masuk ke dalam area jalur kiri yang telah ditentukan maka program akan menghitung sampai beberapa detik yang telah ditentukan, dan jika kendaraan tersebut masih berada di dalam area terlarang, maka akan dianggap sebagai kendaraan yang berhenti dan melanggar. Maka kendaraan tersebut akan diperingati, dideteksi dan diambil gambarnya untuk disimpan beserta informasi lainnya. Pelanggaran ini tidak tergantung kepada lampu lalu lintas yang ada, dikarenakan contoh yang diambil pada penelitian ini adalah perempatan yang memiliki rambu untuk langsung belok ke kiri tanpa berhenti. Berikut adalah *flowchart* deteksi pelanggaran pada jalur kiri, dapat dilihat pada gambar 3.5.



Gambar 3. 6 *Flowchart* deteksi pelanggaran jalur kiri

3.1.4 Analisis Solusi

Pada poin 3.1.1 telah dijelaskan bahwa solusi dari permasalahanya adalah membuat sistem lampu lalu lintas yang dilengkapi dengan teknologi *computer vision* serta dapat terintegrasi dengan lampu lalu lintas yang ada di persimpangan jalan untuk memantau mobil – mobil yang melanggar di area perempatan, dan agar bisa mengetahui kondisi yang sebenarnya serta mengurangi tenaga manusia untuk memantau persimpangan.

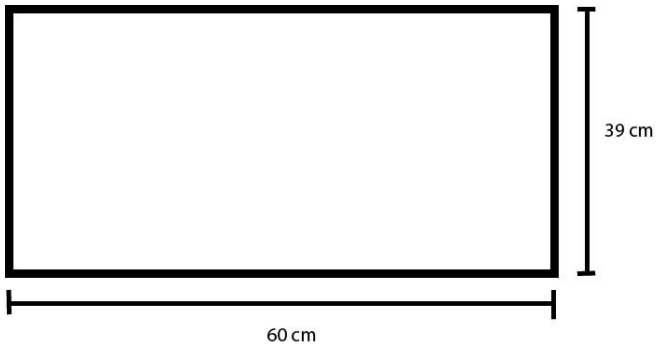
Maksud dari terintegrasinya dengan sistem lampu adalah bahwa ketika lampu lalu lintas menunjukan berwarna merah maka area zebra cross akan dijadikan sebagai area pelanggaran, ketika lampu lalu lintas berwarna hijau maka area zebra cross tidak lagi menjadi area pelanggaran. Sedangkan untuk jalur belok kiri langsung tidak terintergrasi dengan sistem lampu karena pada jalur tersebut, pengendara diharuskan untuk terus jalan. Namun jika ada pengendara yang berhenti pada area tersebut, sistem akan menghitung sampai 5 detik, lalu pengendara tersebut akan difoto oleh sistem dan diberikan himbauan.

3.1.4.1 Pengambilan Footage

Dalam penelitian ini, sangat tidak mungkin untuk melakukkan percobaan langsung di lapangan, oleh karena itu untuk kebutuhan percobaan akan dibuat maket jalan sederhana berukuran 49,5 x 38,5 cm dan mobil berukuran 7x3 cm untuk mengilustrasikan mobil yang ada di perempatan jalan.

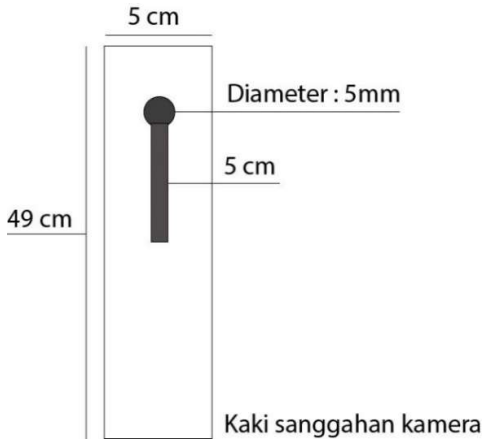
Berikut adalah penjelasan perancangan dan pembuatan maket jalan untuk percobaan deteksi pelanggaran.

- Triplek berukuran 60 x 39 cm sebagai alas untuk gambar jalan



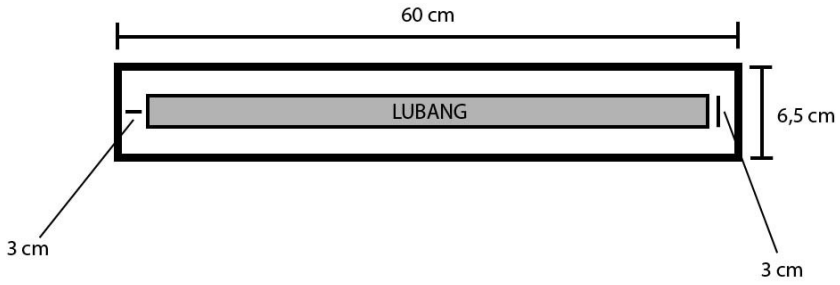
Gambar 3. 7 Gambaran Triplek Untuk Alas Jalan

- Bentuk triplek untuk kaki sanggahan kamera



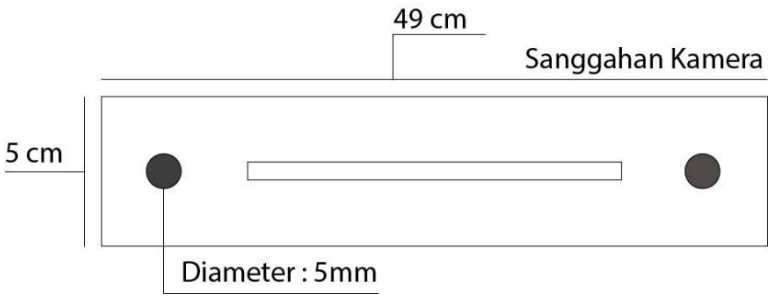
Gambar 3. 8 Rancangan Kaki Sanggahan kamera

- Triplek untuk kaki maket



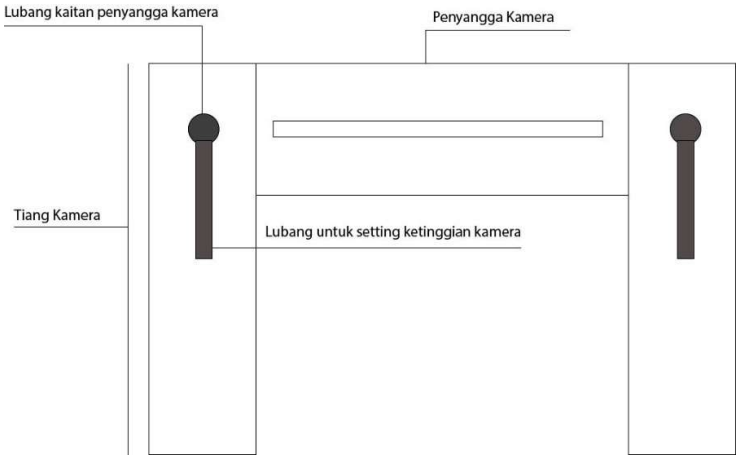
Gambar 3. 9 Rancangan Triplek Untuk Kaki Maket

- Sanggahan kamera



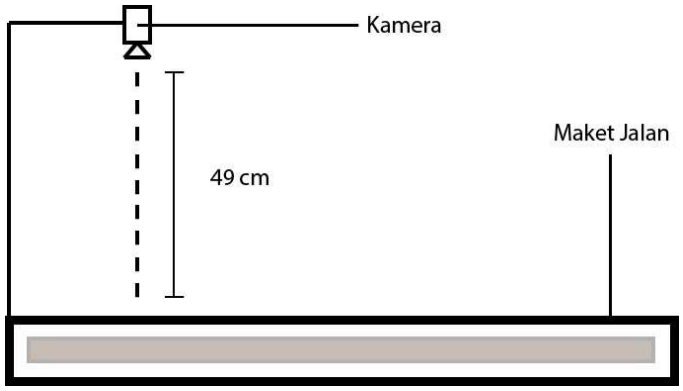
Gambar 3. 10 Rancangan Sanggahan Kamaera

- Rancangan Sanggahan Kamera Keseluruhan



Gambar 3. 11 Rancangan Sanggahan Kamera Keseluruhan

- Kamera yang ditempatkan 48cm diatas permukaan maket jalan



Gambar 3. 12 Penempatan Kamera

- Mobil berukuran 7 x 3 cm yang akan digerakan dengan magnet dari bawah maket jalan.



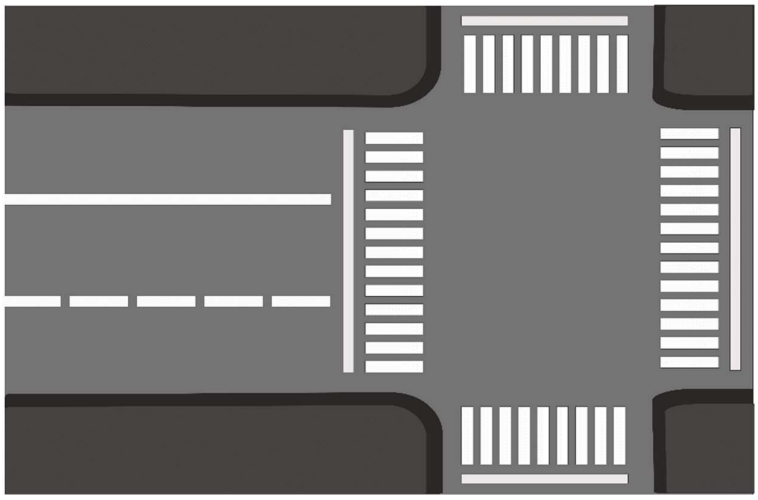
Gambar 3. 13 Mobil yang dipakai untuk pengujian deteksi

- Kamera yang digunakkan untuk pengambilan footage, minimal mempunyai resolusi 640 x 480.



Gambar 3. 14 Kamera yang digunakan

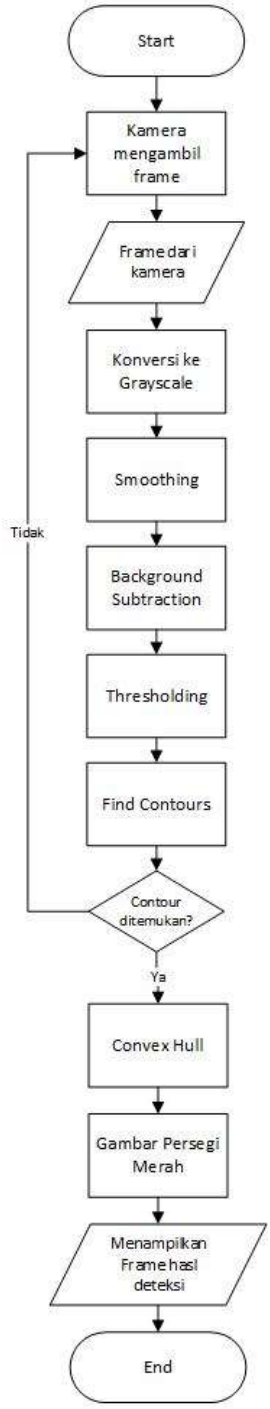
- Berikut adalah rancangan gambar jalan yang akan dicetak dan ditempel pada maket jalan.



Gambar 3. 15 Rancangan Gambar Jalan

3.1.4.2 Proses Deteksi

Pada poin ini akan dijelaskan bagaimana proses pendeteksian kendaraan,yang terdiri dari proses *Grayscaling, Smoothing, Background Subtraction, Thresholding, Contour Detection, dan Convex Hull.* Berikut adalah *flowchart* proses deteksi. Agar lebih mudah di pahami proses deteksi akan direpresentasikan dengan *flowchart* sebagai berikut dapat dilihat pada gambar 3.16.



Gambar 3. 16 Flowchart Proses Deteksi

3.1.4.2.1 Grayscaling

Langkah ini digunakan untuk meminimalisir iluminasi cahaya yang akan berakibat kepada berkurangnya persentase keberhasilan deteksi kendaraan. Berikut adalah fungsi yang digunakan.

```
cv::cvtColor(imgFrame1Copy, imgFrame1Copy, CV_BGR2GRAY);
```

Tabel 3. 1 Tabel Penjelasan Fungsi *Grayscaling*

Komponen	Deskripsi
cvtColor	Nama Fungsi
imgFrame1Copy	Image / matriks yang akan dijadikan grayscale
imgFrame1Copy	Image / matriks tempat menampung hasil grayscale
CV_BGR2GRAY	Metode untuk mengkonversi *img* RGB menjadi grayscale

3.1.4.2.2 Blurring

Langkah ini bertujuan untuk meminimalisir adanya noise pada citra/frame, yang akan berakibat kepada berkurangnya persentase keberhasilan deteksi kendaraan. Fungsi yang digunakkan adalah GaussianBlur, berikut adalah penjelasanya.

```
cv::GaussianBlur(imgFrame1Copy, imgFrame1Copy, cv::Size(5, 5), 0);
```

Tabel 3. 2 Tabel Penjelasan Fungsi *Gaussian Blur*

Komponen	Deskripsi
GaussianBlur	Nama Fungsi
imgFrame1Copy	Image / matriks yang akan ditambahkan efek blur
imgFrame1Copy	Image / matriks tempat menampung hasil blur
Size	Nilai intensitas blur

3.1.4.2.3 Background Subtraction

Langkah ini digunakan untuk memisahkan latar belakang yang statis dan objek yang akan dideteksi. Langkah pertama yang dilakukkan adalah mengambil foto dari suatu background yang statis, dalam kasus ini gambar jalan. Posisi kamera yang digunakan tidak boleh berubah jika ingin melakukkan background subtraction, posisi kamera saat mengambil foto jalan yang diambil harus sama dengan posisi kamera saat akan mendeteksi pelanggaran di perempatan. Berikut adalah fungsi yang digunakkan.

```
cv::subtract(img1,img2,img3)
```

Tabel 3. 3 Tabel Penjelasan Fungsi *Subtract*

Komponen	Deskripsi
subtract	Nama Fungsi
img1	Image / matriks yang akan dibandingkan atau di subtraksi (statis)
img2	Image / matriks yang akan dibandingkan atau di subtraksi (frame kamera)
img3	Image / matriks yang akan menampung hasil nya

3.1.4.2.4 Thresholding

Dengan thresholding didapatkan warna hitam dan putih yang membedakan latar belakang statis (hitam) dan objek yang akan dideteksi (putih). Berikut adalah hasil thresholding pada video percobaan dan fungsi yang digunakkan.

```
cv::threshold(img1, img2, threshold value, max binary value, threshold
type);
```

Tabel 3. 4 Tabel Penjelasan Fungsi *Thresholding*

Komponen	Deskripsi
threshold	Nama Fungsi
img1	Image / matriks yang akan diproses (hasil *grayscaling*)

img2	Image / matriks yang akan dibandingkan atau di subtraksi
threshold value	Nilai yang digunakan sebagai ambang batas pada proses threshold.
max binary value	Nilai yang digunakan untuk threshold biner (untuk piksel yang sudah dipilih)



Gambar 3. 17 *Thresholding*

3.1.4.2.5 Contour Detection

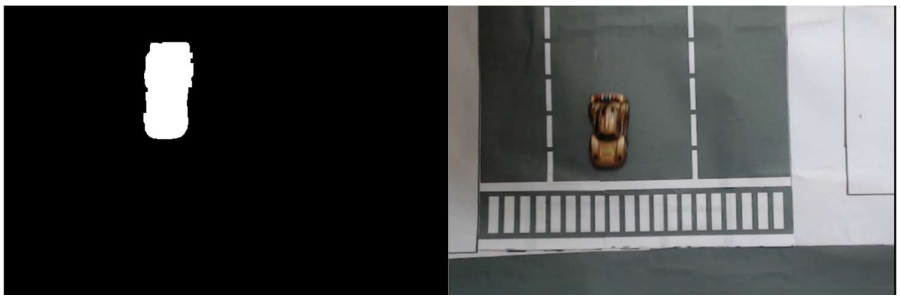
Hasil dari thresholding menunjukan bahwa piksel yang merupakan objek tujuan deteksi tidak terdeteksi secara optimal, masih ada garis tepi yang terputus, maka dilakukkan Contour Detection untuk menyempurnakan garis tepi pada objek yang telah terdeteksi.

```
cv::findContours(img1, vectorPoints, cv::RETR_EXTERNAL,
                 cv::CHAIN_APPROX_SIMPLE);
```

Tabel 3. 5 Tabel Penjelasan Fungsi *Find Contours*

Komponen	Deskripsi
img1	Matriks gambar yang akan dideteksi (hasil thresholding)
findCountours	Nama fungsi yang digunakan
vectorPoints	Variable untuk menampung poin – poin kontur yang didapat
RETR EXTERNAL	Perintah untuk mendapatkan kontur terluar

CHAIN APPROX SIMPLE	Perintah untuk mengurangi *vector poin* yang redundan



Gambar 3. 18 *Contour Detection*

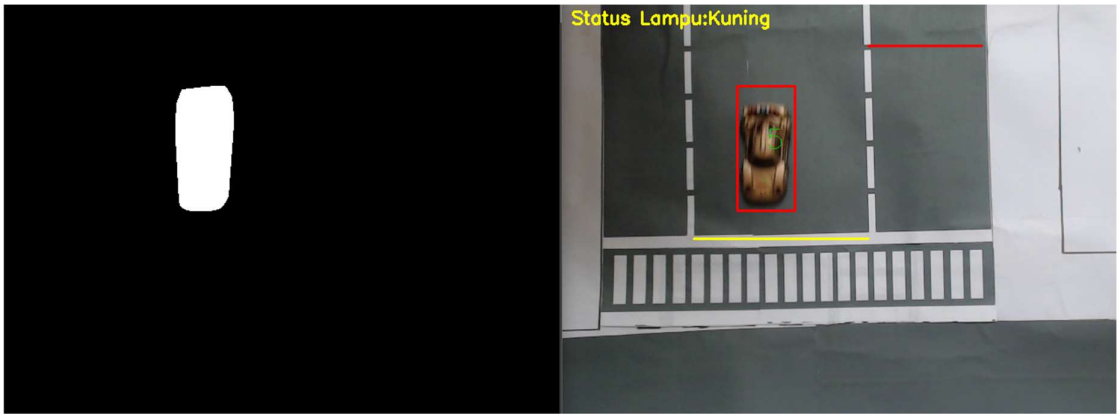
3.1.4.2.6 Convex Hull dan Blob Detection

Hasil dari *Countour Detection* ditambahkan dengan metode *Convex Hull* yang menghasilkan sebuah blob, blob tersebut merepresentasikan kendaraan pada perempatan. Pada citra normal nya, persegi merah akan digambarkan pada blob yang sudah terdeteksi untuk menunjukan bahwa objek telah berhasil di deteksi. Selanjutanya akan digambarkan garis yang akan digunakan sebagai "sensor" untuk uji coba apakah program bisa mendeteksi mobil yang melewati garis tersebut atau tidak.

```
cv::convexHull(contours[i], convexHulls[i]);
```

Tabel 3. 6 Tabel Penjelasan Fungsi Convex Hull

Komponen	Deskripsi
convexHull	Nama fungsi yang digunakan
contours	Variable poin poin kontur yang didapat dari deteksi kontur
convexHulls	Variable poin yang akan menampung poin – poin convexHull



Gambar 3. 19 Hasil *Convex Hull* dan *Blob Detection*

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

3.1.5.1 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras pada penelitian ini, merupakan perangkat keras yang digunakan dalam pembangunan perangkat lunak. Adapun spesifikasi perangkat keras yang digunakan tertera pada tabel sebagai berikut.

Tabel 3. 7 Tabel Analisis Kebutuhan Perangkat Keras

No	Perangkat Keras	Keterangan
1	*Processor*	Intel Core i5-6198DU 2.4 GHz
2	RAM	8 GB
3	VGA	2 GB
4	Resolusi Layar	1366x768
5	Kamera	Logitech HD 1080p

3.1.5.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak pada penelitian ini merupakan perangkat lunak yang digunakkan dalam pembangunan perangkat lunak. Adapun spesifikasi perangkat lunak tertera pada tabel sebagai berikut.

Tabel 3. 8 Tabel Analisis Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows 10 Pro
2	Integrated Development Environment (IDE)	Visual Studio 2015
		Netbeans IDE 8.2
3	Pemodelan	Visio 2016
		Star UML
4	Database	MySQL

3.1.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dibutuhkan untuk mendefinisikan pemodelan system dan basis data perangkat lunak. Pada penelitian ini, pemodelan sistem dibagi menjadi dua jenis yaitu pemodelan dengan *Unified Modelling Language* (UML) untuk memodelkan program lampu lalu lintas dan program deteksi pelanggaran. Data Flow Diagram untuk memodelkan website yang menampilkan data pelanggaran dan pemodelan database akan dimodelkan menggunakan *Entity Relationship Diagram* (ERD).

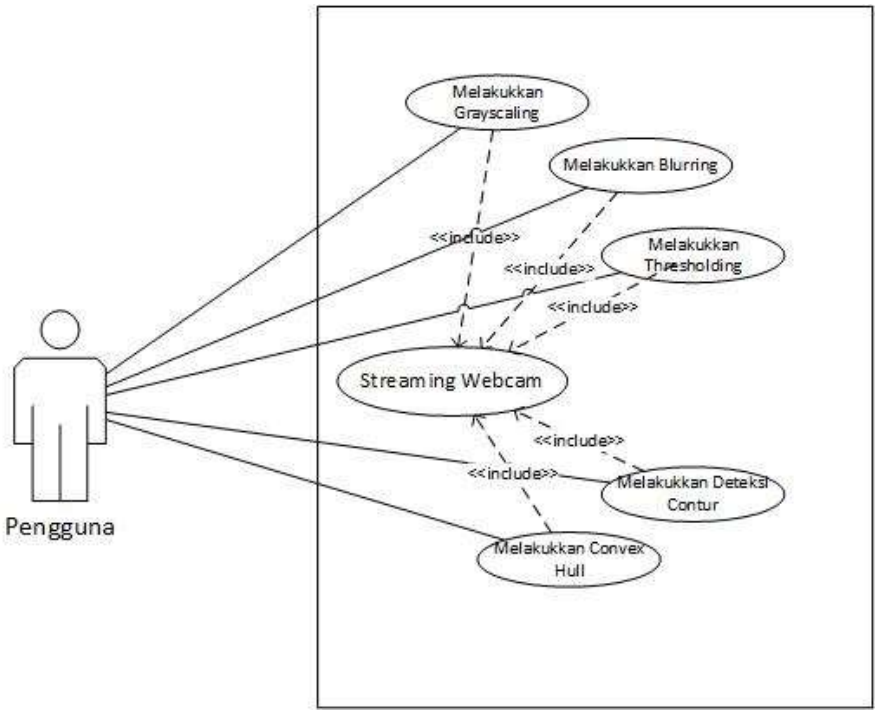
3.1.6.1 Analisis Basis Data

Secara umum analisis basis data direpresentasikan dengan Entity Relationship Diagram, namun di dalam penelitian ini basis data yang digunakan adalah sebagai pelengkap untuk menyimpan data, sehingga tidak dibutuhkan adanya relasi antar tabel. Detail setiap tabelnya dapat dilihat pada poin 3.2.1.2 Struktur Tabel..

3.1.6.2 UML Program Deteksi Pelanggaran

3.1.6.2.1 Use Case Diagram

Aktor yang berinteraksi dengan program / sistem deteksi pelanggaran hanya satu orang, program ini hanya sebatas untuk menampilkan footage perempatan dan mendeteksi mobil yang melakukkan pelanggran (berada pada area terlarang).



Gambar 3. 20 Use Case Program Deteksi

3.1.6.2.2 Use Case Skenario

Pada bagian ini akan dijelaskan scenario dari setiap usecase yang merupakan urutan interaksi aktor dan program deteksi tersebut. Berikut usecase scenario yang terdapat pada tabel dibawah ini.

Tabel 3. 9 Use Case Skenario Program Deteksi

No	1
Usecase	Melakukkan Stream Webcam
Tujuan	Untuk mendapatkan footage pada perempatan
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Program menampilkan footage yang diambil
Skenario Utama	
1.	Pengguna menjalankan program deteksi
2.	Sistem menampilkan live footage yang diambil kamera webcam

No	2
Usecase	Melakukkan Grayscaling
Tujuan	Untuk merubah warna frame menjadi hitam putih
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Frame hasil grayscaling diambil untuk dilakukkan blurring
Skenario Utama	
1.	Pengguna menjalankan program deteksi
2.	Sistem melakukkan proses grayscaling dibelakang layar
No	3
Usecase	Melakukkan Blurring
Tujuan	Untuk menambahkan efek blur pada frame hitam putih
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Frame hasil blurring diambil untuk dilakukkan thresholding
Skenario Utama	
1.	Pengguna menjalankan program deteksi
2.	Sistem melakukkan proses blurring dibelakang layar
No	4
Usecase	Melakukkan Thresholding
Tujuan	Untuk menambahkan nilai ambang batas pada frame
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Frame hasil thresholding diambil untuk dilakukkan deteksi kontur
Skenario Utama	
1.	Pengguna menjalankan program deteksi
2.	Sistem melakukkan proses thresholding di belakang layar
No	5
Usecase	Melakukkan Deteksi Kontur

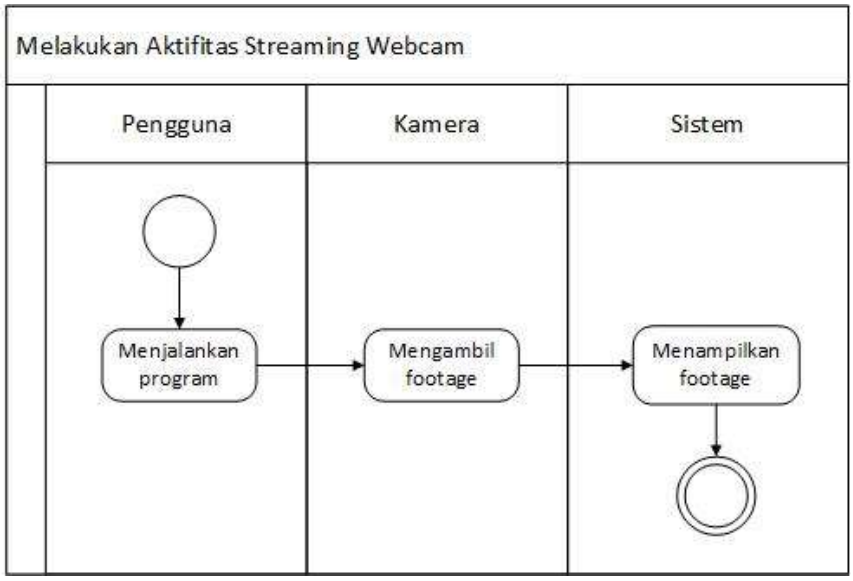
Tujuan	Untuk mendeteksi kontur objek yang ada pada frame dari hasil thresholding
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Frame hasil deteksi kontur diambil untuk dilakukkan penelusuran convex hull
Skenarion Utama	
1.	Pengguna menjalankan program deteksi
2.	Sistem melakukkan proses deteksi kontur dibelakang layar
No	6
Usecase	Melakukkan Convex Hull
Tujuan	Untuk menelusuri convex hull dari hasil deteksi kontur
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Menampilkan hasil convex hull
Skenario Utama	
1.	Pengguna menjalankan program deteksi
2.	Sistem menampilkan hasil convex hull

3.1.6.2.3 Activity Diagram

Activity diagram menggambarkan alur aktifitas sistem, diagram ini digunakan untuk melihat aktifitas yang dilakukkan oleh sistem dalam poin ini sistem / program deteksi, dapat dilihat pada gambar – gambar berikut.

3.1.6.2.3.1 Activity Streaming Webcam

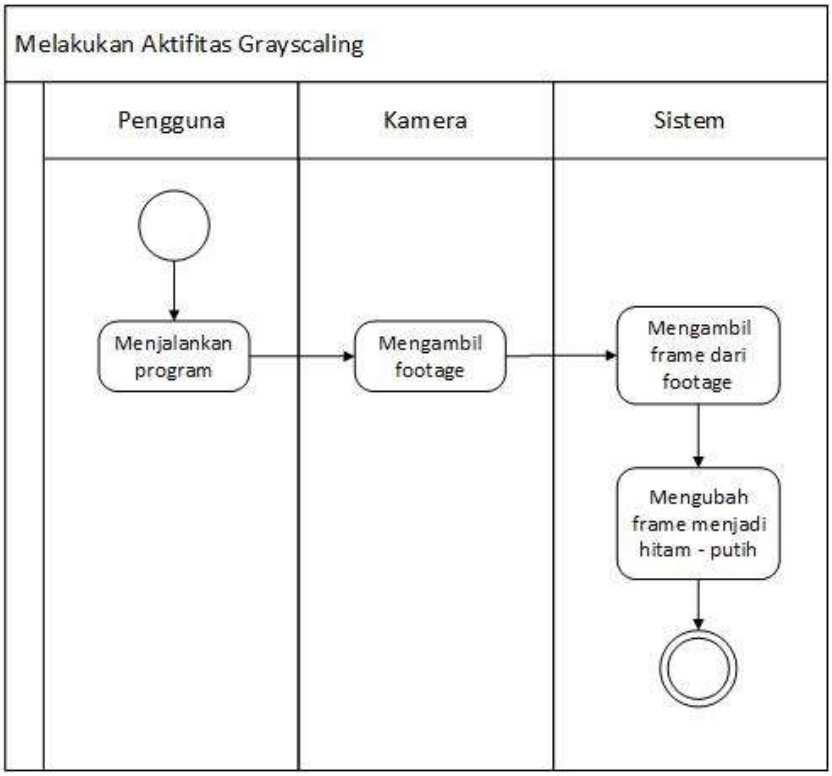
Berikut ini merupakan gambaran alur aktifitas yang dilakukkan user saat melakukkan streaming webcam.



Gambar 3. 21 Activity Streaming Webcam

3.1.6.2.3.2 Activity Grayscaling

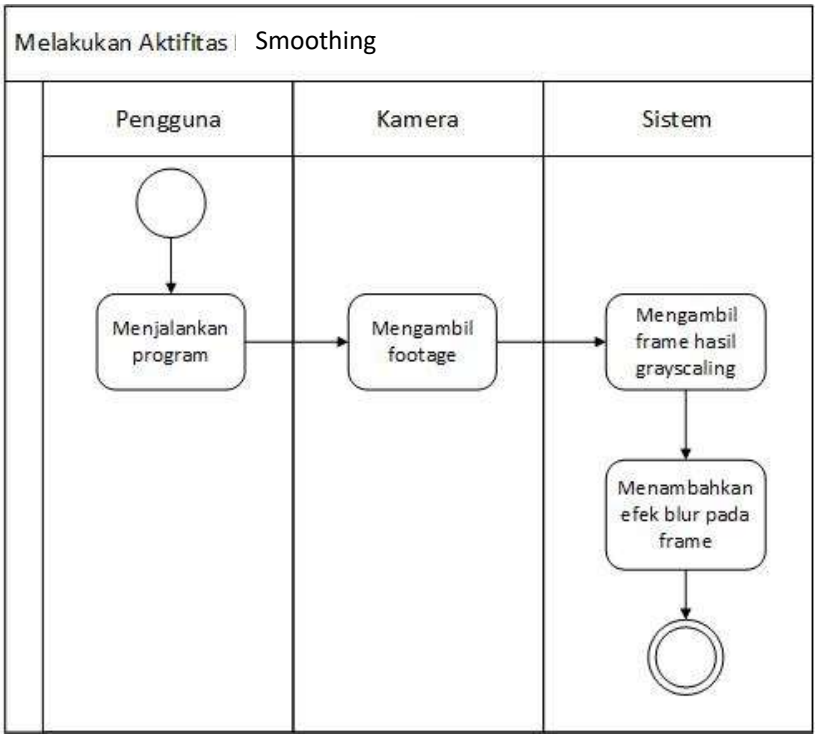
Berikut ini merupakkan gambaran alur aktifitas grayscaling, yang dilakukkan oleh pengguna saat menyalakan program. Aktifitas berjalan di belakang layar. Dapat dilihat pada gambar berikut ini.



Gambar 3. 22 Activity Grayscaling

3.1.6.2.3.3 Activity Smoothing

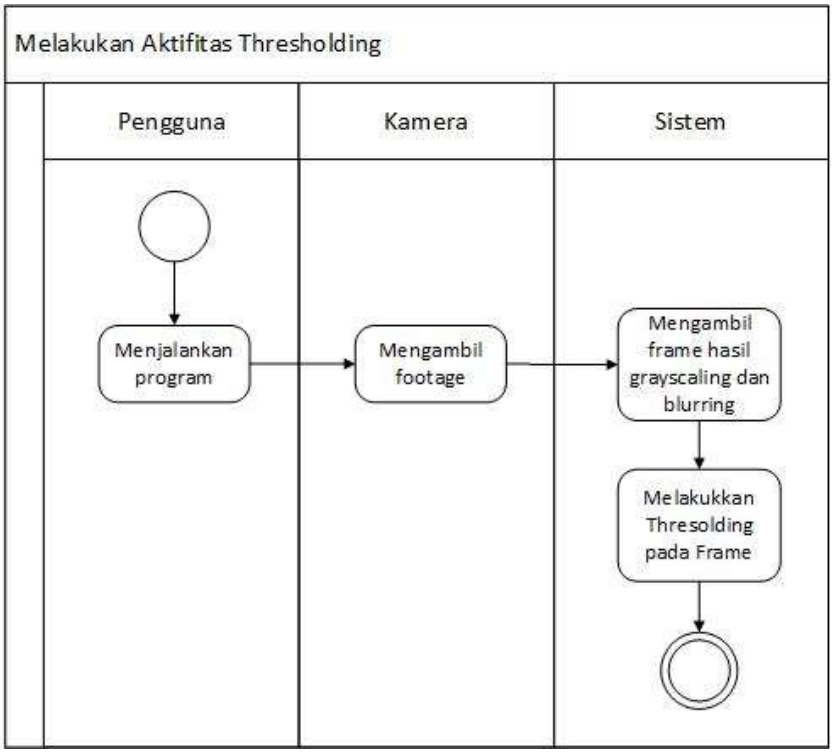
Berikut ini merupakkan gambaran alur aktifitas smoothing, yang dilakukkan oleh pengguna saat menyalakan program. Pada aktifitas ini hasil dari grayscaling diambil untuk digunakkan dan ditambahkan efek blur. Aktifitas berjalan di belakang layar. Dapat dilihat pada gambar berikut ini.



Gambar 3. 23 Acitivity Smoothing

3.1.6.2.3.4 Activity Thresholding

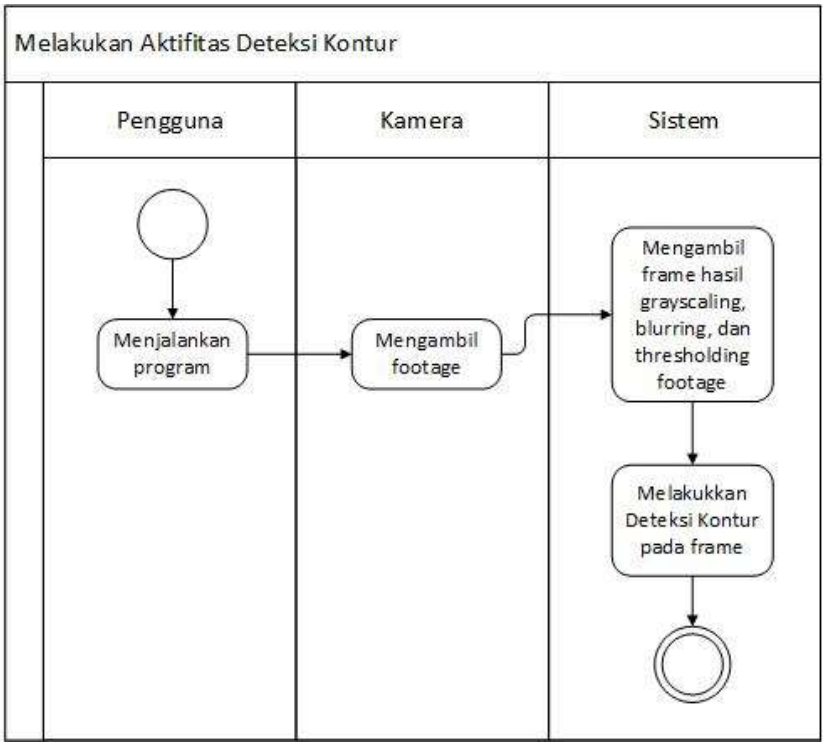
Berikut ini merupakkan gambaran alur aktifitas blurring, yang dilakukkan oleh pengguna saat menyalakan program. Pada aktifitas ini hasil dari grayscaling dan blurring diambil untuk ditambahkan nilai ambag batas. Aktifitas berjalan di belakang layar. Dapat dilihat pada gambar berikut.



Gambar 3. 24 Activity Thresholding

3.1.6.2.3.5 Activity Deteksi Kontur

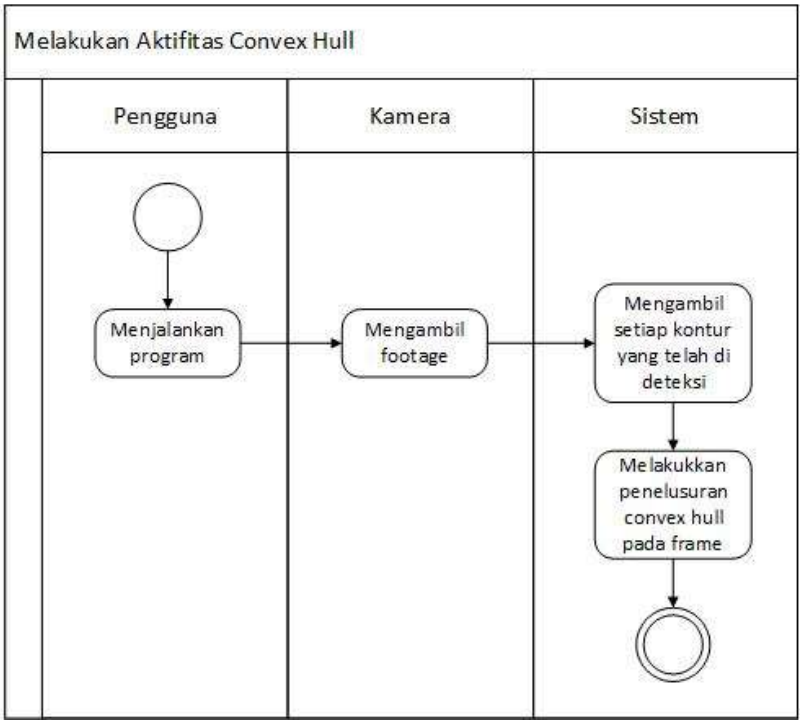
Berikut ini merupakkan gambaran alur aktifitas deteksi kontur, yang dilakukkan oleh pengguna saat menyalakan program. Pada aktifitas ini hasil dari thresholding digunakkan untuk dicari kontur dari objek yang terdeteksi. Aktifitas berjalan di belakang layar. Dapat dilihat pada gambar berikut ini.



Gambar 3. 25 Activity Deteksi Kontur

3.1.6.2.3.6 Activity Convex Hull

Berikut ini merupakkan gambaran alur aktifitas penelusuran convex hull, yang dilakukkan oleh pengguna saat menyalakan program. Pada aktifitas ini hasil dari deteksi kontur diambil untuk ditelusuri convex hull. Aktifitas berjalan di belakang layer, dan hasilnya akan ditampilkan. Dapat dilihat pada gambar berikut ini.



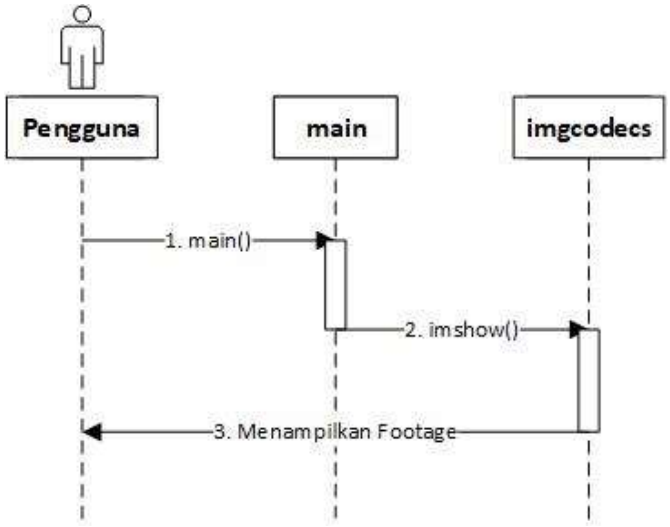
Gambar 3. 26 Activity Convex Hull

3.1.6.2.4 Sequence Diagram

Sequence diagram merupakan diagram aliran pesan, yang menggambarkan pesan yang mengalir dari objek ke objek lainya. Berikut adalah diagram aliran pesan pada sistem / program deteksi.

3.1.6.2.4.1 Sequence Diagram Streaming Webcam

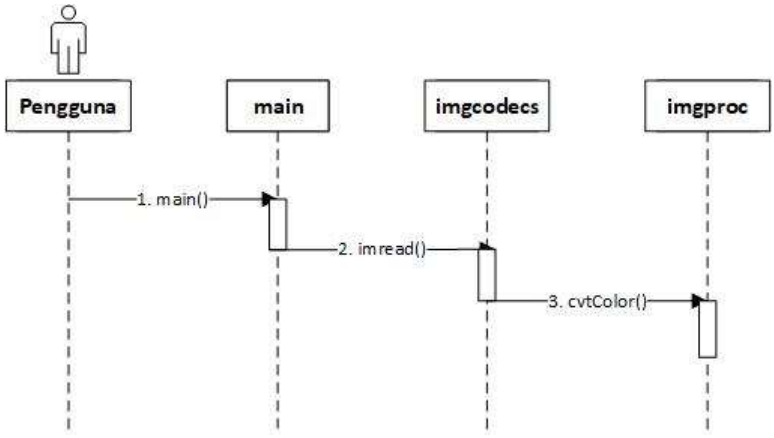
Berikut adalah sequence diagram untuk proses streaming webcam pada program deteksi.



Gambar 3. 27 Sequence Diagram Streaming Webcam

3.1.6.2.4.2 Sequence Diagram Grayscaling

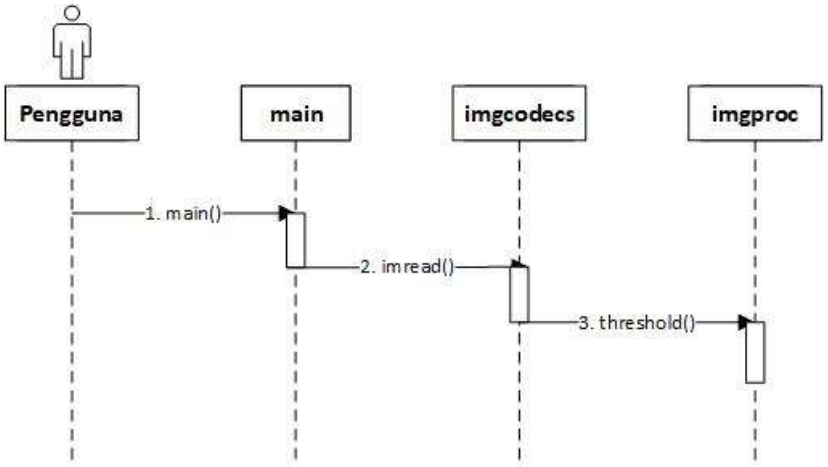
Berikut adalah sequence diagram untuk proses grayscaling pada program deteksi.



Gambar 3. 28 Sequence Diagram Grayscaling

3.1.6.2.4.3 Sequence Diagram Thresholding

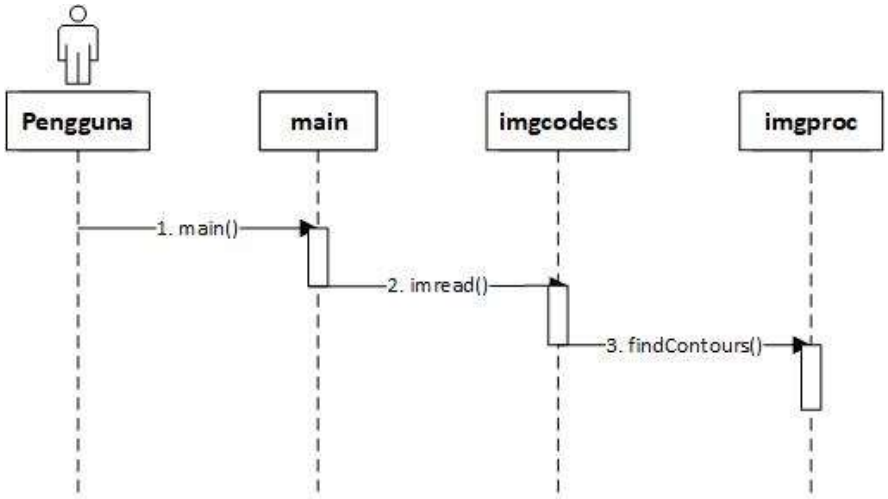
Berikut ini adalah sequence diagram untuk proses thresholding pada program deteksi.



Gambar 3. 29 Sequence Diagram Thresholding

3.1.6.2.4.4 Sequence Diagram Deteksi Kontur

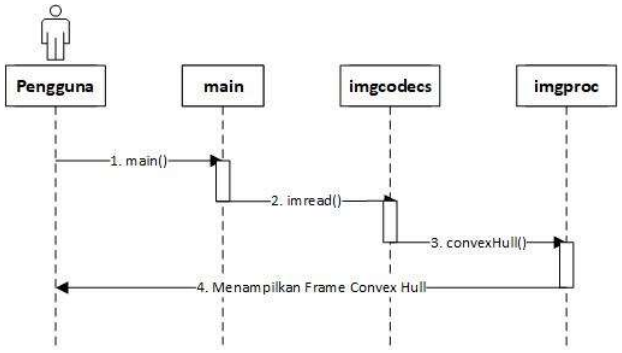
Berikut ini adalah sequence diagram untuk proses deteksi kontur pada program deteksi.



Gambar 3. 30 Sequence Diagram Deteksi Kontur

3.1.6.2.4.5 Sequence Diagram Convex Hull

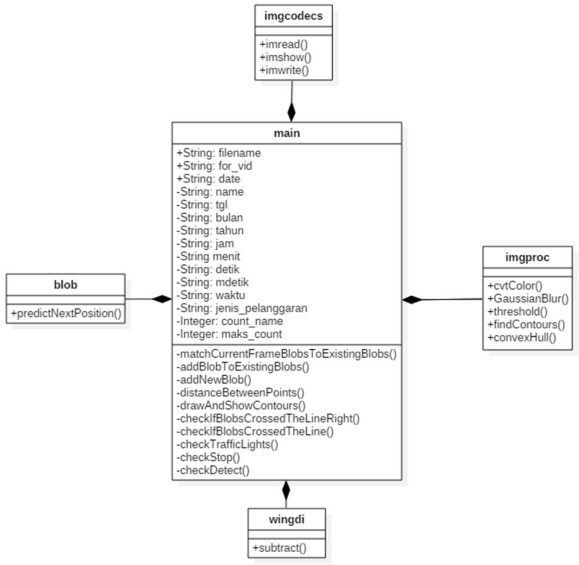
Berikut ini adalah sequence diagram untuk proses penelusuran convex hull pada program deteksi.



Gambar 3. 31 Sequence Diagram Convex Hull

3.1.6.2.5 Class Diagram

Class diagram menggambarkan objek yang ada dalam system dalam kasus ini sistem / program deteksi, class diagram merupakan yang paling sering digunakkan dalam pemodelan object – oriented, dapat dilihat pada gambar berikut.

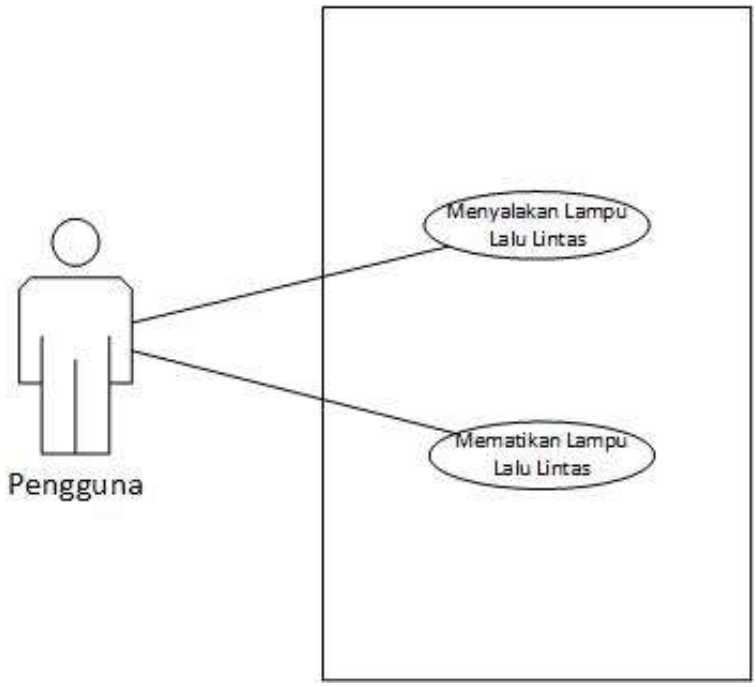


Gambar 3. 32 Class Diagram

3.1.6.3 UML Sistem Lampu

3.1.6.3.1 Use Case Diagram

Hanya terdapat satu aktor yang akan berinteraksi dengan sistem lampu lalu lintas, yaitu pengguna. Interaksi antara pengguna dengan sistem hanya sebatas menjalankan / menyalakan lampu lalu lintas dan mematikan lampu lalu lintas, digambarkan pada use case diagram sebagai berikut.



Gambar 3. 33 Usecase Diagram

3.1.6.3.2 Use Case Skenario

Pada bagian ini akan dijelaskan scenario dari setiap usecase yang merupakan urutan interaksi aktor dan sistem tersebut. Berikut usecase scenario yang terdapat pada tabel tersebut.

Tabel 3. 10 Tabel *Usecase* Skenario

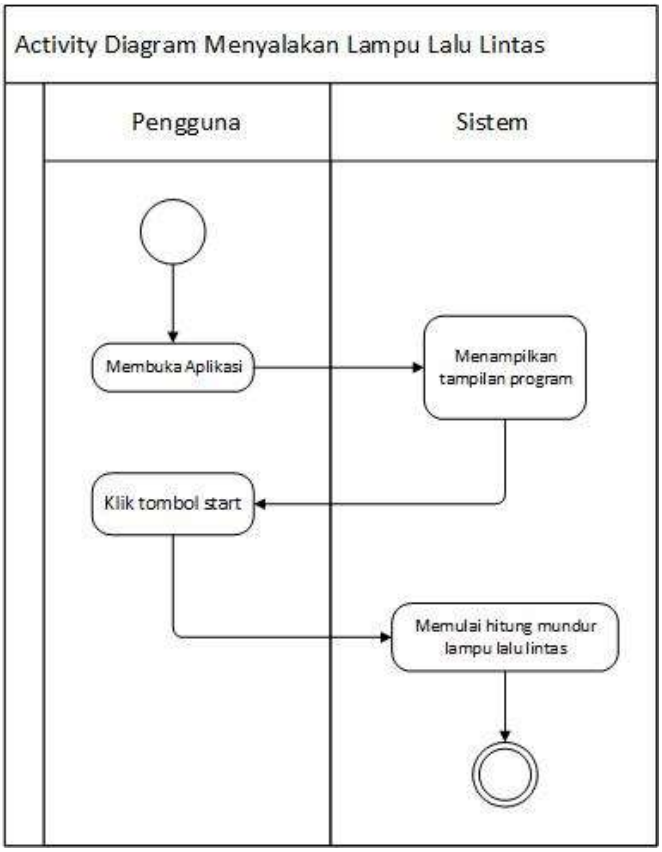
No	1
Usecase	Menyalakan Lampu Lalu Lintas
Tujuan	Agar program bisa mengirimkan status lampu lalu lintas ke program deteksi.
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Lampu lalu lintas menyala, ditandai dengan berjalanya hitungan mundur pada tampilan program.
Skenario Utama	
3.	Pengguna membuka aplikasi lampu lalu lintas
4.	Sistem menampilkan tampilan utama aplikasi
5.	Pengguna menekan tombol START
6.	Sistem menyalakan lampu lalu lintas
No	2
Usecase	Mematikan Lampu Lalu Lintas
Tujuan	Untuk memberhentikan program lampu lalu lintas
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan utama program
Kondisi Akhir	Jika user memilih untuk tidak menutup program, maka program akan terus berjalan, jika memilih untuk menutup program sistem menutup program lampu lalu lintas.
Skenario Utama	
3.	Pengguna menekan tombol stop
4.	Sistem menampilkan dialog konfirmasi untuk menutup program
5.	Jika memilih untuk ditutup, system akan menutup program. Jika tidak program akan terus berjalan.

3.1.6.3.3 Activity Diagram

Activity diagram menggambarkan alur aktifitas sistem, diagram ini digunakan untuk melihat aktifitas yang dilakukkan oleh sistem, dapat dilihat pada gambar – gambar berikut.

3.1.6.3.3.1 Activity Menyalakan Lampu Lalu Lintas

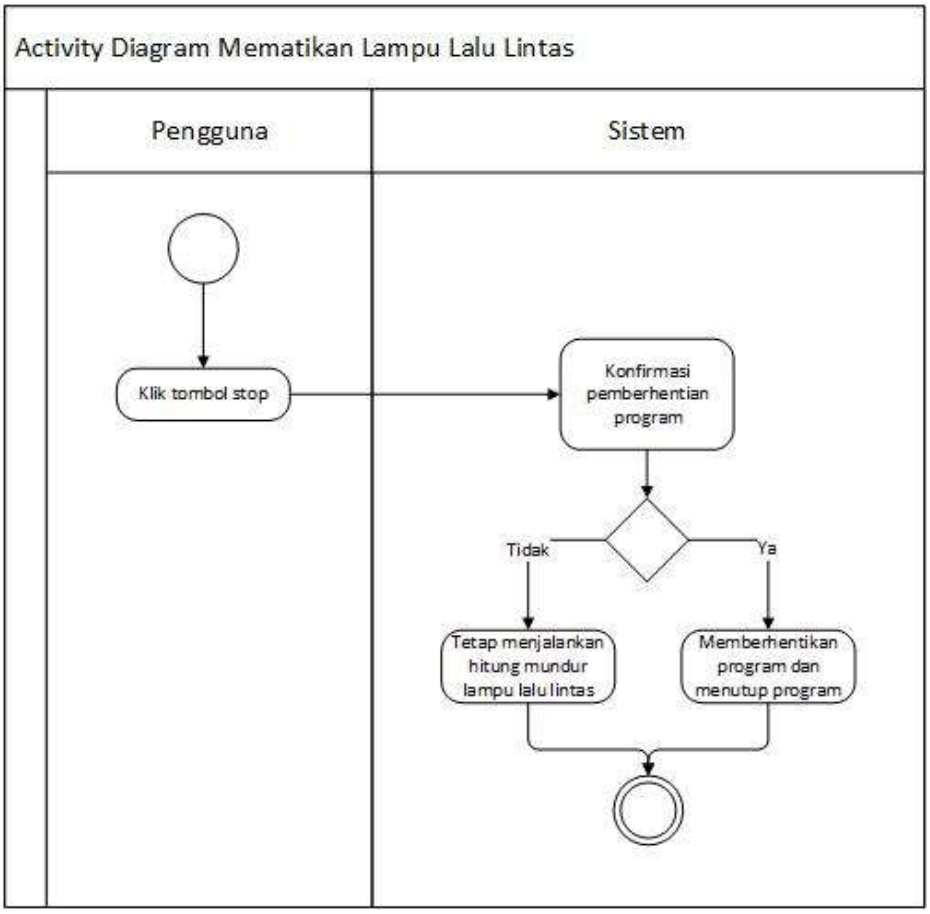
Berikut ini merupakan aktifitas untuk menyalakan lampu lalu lintas, dapat dilihat pada gambar sebagai berikut.



Gambar 3. 34 *Activity* Menyalakan Lampu Lalu Lintas

3.1.6.3.3.2 Activity Mematikan Lampu Lalu Lintas

Berikut ini merupakan aktifitas untuk mematikan lampu lalu lintas, dapat dilihat pada gambar berikut.



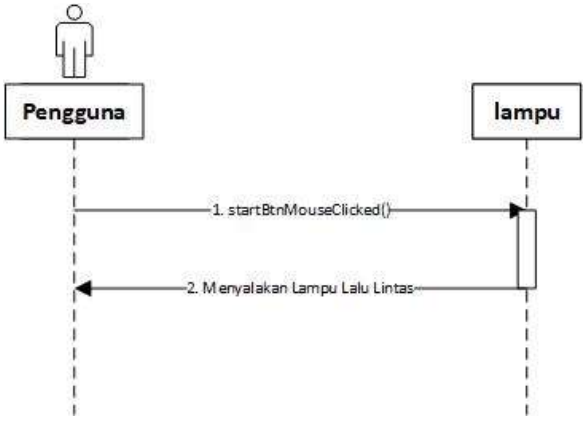
Gambar 3. 35 *Activity* Mematikan Lampu Lalu Lintas

3.1.6.3.4 Sequence Diagram

Sequence diagram merupakan diagram aliran pesan, yang menggambarkan pesan yang mengalir dari objek ke objek lainya. Akan digambarkan pada sequence diagram berikut ini.

3.1.6.3.4.1 Sequence Diagram Menyalakan Lampu Lalu Lintas

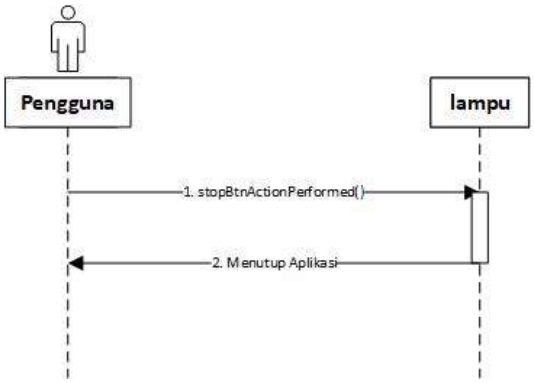
Berikut ini adalah sequence diagram untuk menyalakan lampu lalu lintas dapat dilihat pada gambar berikut.



Gambar 3. 36 *Sequence* Menyalakan Lampu Lalu Lintas

3.1.6.3.4.2 Sequence Diagram Mematikan Lampu lalu lintas

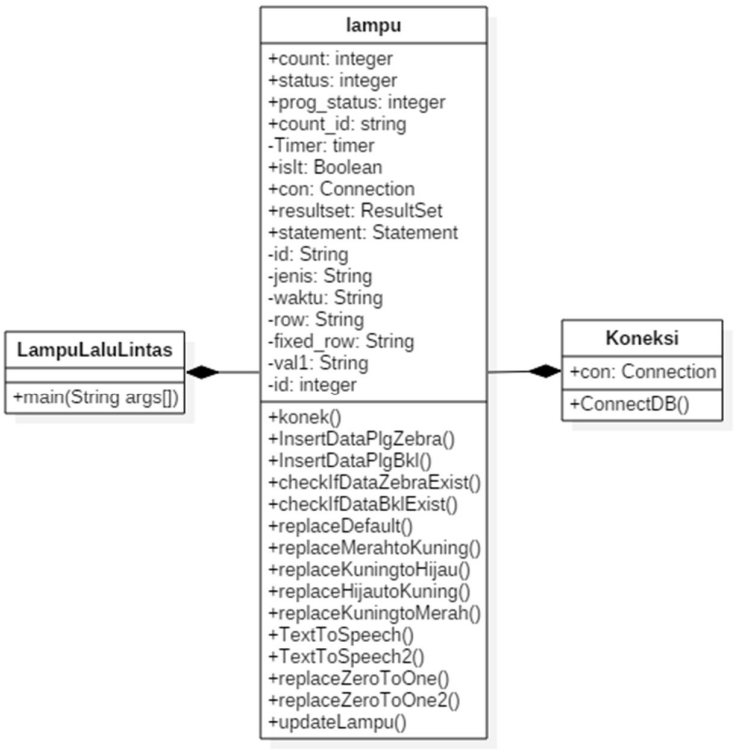
Berikut adalah sequence diagram untuk mematikan lampu lalu lintas dapat dilihat pada gambar berikut.



Gambar 3. 37 *Sequence* Mematikan Lampu Lalu Lintas

3.1.6.3.5 Class Diagram

Class diagram menggambarkan objek yang ada dalam system, class diagram merupakan yang paling sering digunakkan dalam pemodelan object – oriented, dapat dilihat pada gambar berikut.



Gambar 3. 38 Class Diagram

3.2 Perancangan Sistem

Perancangan sistem merupakan penggambaran, perencanaan, dan pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi.

3.2.1 Perancangan Basis Data

Perancangan basis data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Namun pada poin analisis basis data telah dijelaskan bahwa pada panelitian ini, basis data digunakan sebagai

pelengkap untuk menyimpan data, sehingga tidak dibutuhkanya relasi antar tabel. Namun detail dari setiap tabelnya dapat dilihat pada tabel sebagai berikut.

3.2.1.1 Struktur Tabel

Tabel 3. 11 Struktur Tabel tbl_pelanggaran

Nama Field	Tipe Data	Ukuran	Tipe *Key*
id	Varchar	25	Primary Key
jenis	Varchar	80	
waktu	Varchar	200	

Tabel 3. 12 Struktur Tabel status_lampu

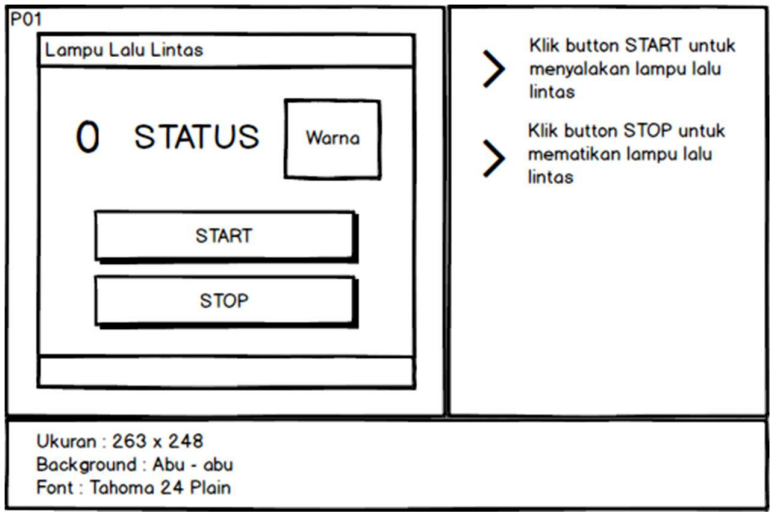
Nama Field	Tipe Data	Ukuran	Tipe *Key*
id	Varchar	1	Primary Key
status_lampu	Varchar	8	

3.2.2 Perancangan Antar Muka

Perancangan antar muka merupakan suatu bentuk rancangan tampilan dari perangkat lunak yang akan dibangun, untuk kebutuhan saat akan membangun bagian tampilan perangkat lunak. Berikut adalah rancangan antar muka system.

3.2.2.1 Perancangan Antar Muka Program Lampu Lalu Lintas

1. Perancangan antarmuka tampilan utama

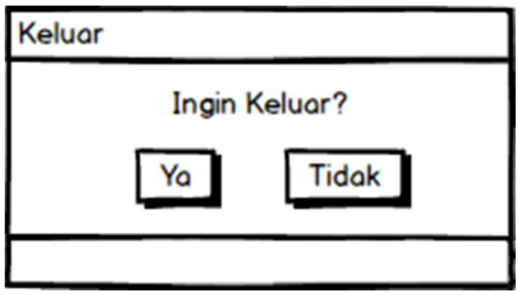


Gambar 3. 39 *Mock Up* Tampilan Utama

3.2.2.1.1 Perancangan Pesan Program Lampu Lalu Lintas

Perancangan pesan berisi rancangan pesan yang akan disampaikan oleh system kepada pengguna pada beberapa keadaan tertenu. Dapat dilihat pada gambar berikut.

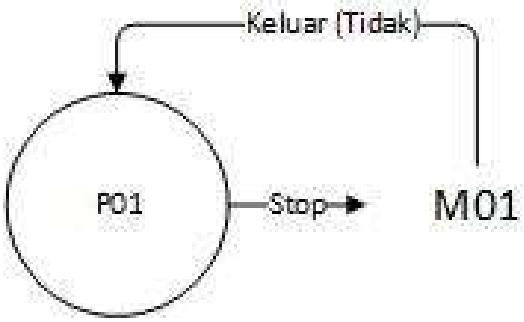
1. Perancangan pesan kofirmasi untuk keluar dari program (M01)



Gambar 3. 40 Pesan Konfirmasi Keluar

3.2.2.1.2 Jaringan Semantik Sistem Lampu

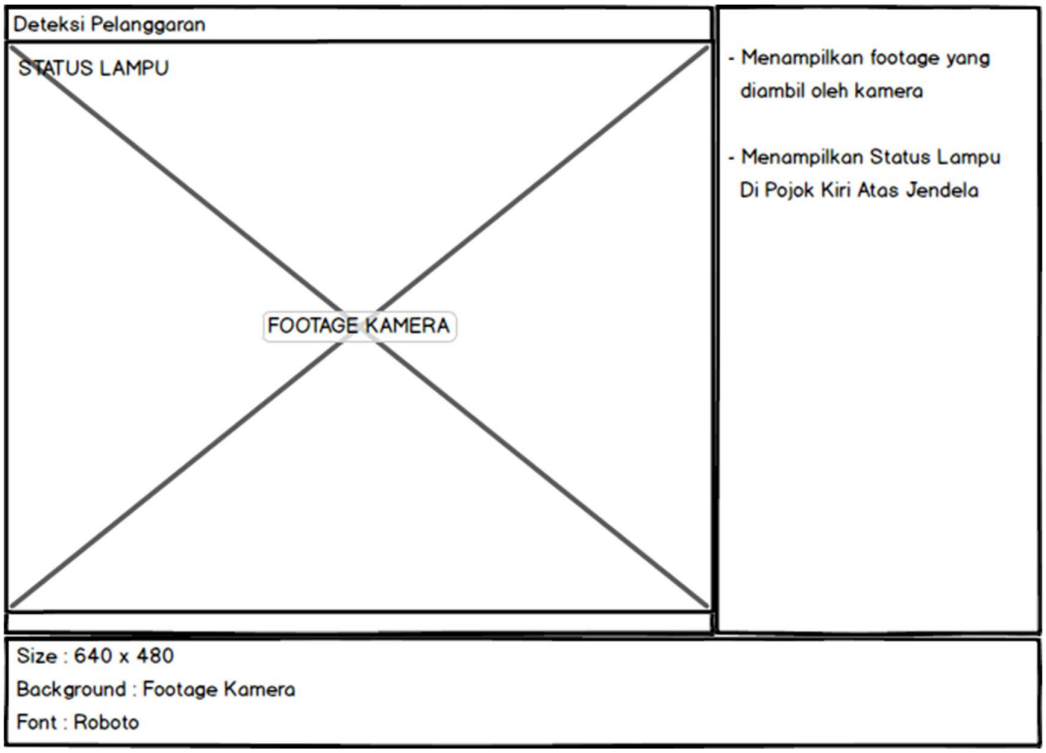
Jaringan semantic merupakan gambaran hubungan antarmuka satu dengan antar muka yang lainya, dapat dilihat pada gambar berikut.



Gambar 3. 41 Jaringan Semantik

3.2.2.2 Perancangan Antarmuka Sistem Deteksi

1. Tampilan Utama (D01)



Gambar 3. 42 Mockup Sistem Deteksi