PROTOTYPE AUTONOMOUS CAR
MENGGUNAKAN *HAAR-CASCADE CLASSIFIER*
BERBASIS RASPBERRY PI

Disusun oleh:

Muhammad Fachrurazi

16524055

Jurusan Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2020

LEMBAR PENGESAHAN

PROTOTYPE AUTONOMOUS CAR
MENGGUNAKAN *HAAR CASCADE CLASSIFIER*
BERBASIS RASPBERRY PI

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

Disusun oleh:

Muhammad Fachrurazi
16524055

Yogyakarta, 23-juni-2020

Menyetujui,

Pembimbing 1	Pembimbing 2
Medilla Kusriyanto, S.T., M.Eng.	**Elvira Sukma Wahyuni, S.Pd.T., M.Eng.**
015240101	**155231301**

LEMBAR PENGESAHAN

PROTOTYPE AUTONOMOUS CAR MENGGUNAKAN *HAAR CASCADE CLASSIFIER* BERBASIS RASPBERRY PI

Dipersiapkan dan disusun oleh:

Muhammad Fachrurazi

16524055

Telah dipertahankan di depan dewan penguji

Pada tanggal: 5 juli 2020

Susunan dewan penguji

Ketua Penguji : Nama lengkap dengan gelar, _____

Anggota Penguji 1: Nama lengkap dengan gelar, _____

Anggota Penguji 2: Nama lengkap dengan gelar, _____

Skripsi ini telah diterima sebagai salah satu persyaratan untuk memperoleh gelar Sarjana

Tanggal: 5 juli 2020

Ketua Program Studi Teknik Elektro

Yusuf Aziz Amrulloh, S.T, M.Eng, Ph.D.

NIK

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan Saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.

Yogyakarta, 23 juni 2020

Muhammad Fachrurazi

KATA PENGANTAR

Assalamu'alaikum Wr.Wb

Puji dan syukur penulis panjatkan Kehadirat Allah SWT yang telah melimpahkan rahmat, karunia serta hidayah-Nya, sehingga penulis dapat menyelesaikan Laporan Skripsi dengan judul "*Prototype Autonomous Car* Menggunakan *Haar-Cascade Classifier* Berbasis Raspberry Pi" dengan baik dan benar. Tugas Akhir ini wajib ditempuh oleh mahasiswa Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia, sebagai syarat untuk menempuh gelar kesarjanaan.

Kelancaran dalam mempersiapkan dan menyelesaikan Laporan Skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Yusuf Aziz Amrulloh, S.T, M.eng, Ph.D., selaku Ketua Jurusan Teknik Elektro, Universitas Islam Indonesia.
2. Bapak Medilla Kusryanto, S.T, M.Eng., selaku Sekretaris Jurusan Teknik Elektro dan Dosen Pembimbing pertama, yang telah memberikan bantuan dan pengarahan sehingga terselesaikan Laporan Skripsi ini.
3. Ibu Elvira Sukma Wahyuni, S.Pd.T., M.Eng., selaku Dosen Pembimbing kedua, yang telah memberikan bantuan dan pengarahan sehingga terselesaikan Laporan Skripsi ini.
4. Para Dosen dan Karyawan program studi Teknik Elektro, Universitas Islam Indonesia.
5. Kedua Orang Tua dan keluarga yang selalu memberikan do'a, usaha, dan kasih sayangnya.
6. Teman-teman mahasiswa Jurusan Teknik Elektro, Universitas Islam Indonesia angkatan 2016.
7. Teman-teman Ulil Albab Student Center (UASC), Universitas Islam Indonesia angkatan.
8. Semua Pihak yang telah memberikan masukan dan dorongan dalam menyelesaikan Laporan Skripsi ini.

Wassalamu'alaikum wr.wb

Yogyakarta, 23 juni 2020

Muhammad Fachrurazi

ARTI LAMBANG DAN SINGKATAN

Singkatan	Arti Singkatan
RPi	Raspberry Pi
ESC	Electronic Speed Control
Li-ion	Lithium-Ion
IDE	Integrated Development Environment
ADC	Analog to Digital Converter
USB	Universal Serial Bus
SBC	Single Board Computer
ML	Machine Learning
cm	Centimeter
YOLO	You Only Look Once
OpenCV	Open Source Computer Vision
CV	Computer Vision
AdaBoost	Adaptive boosting
SBC	Single Board Computer
PC	Personal Computer
V	Voltage
A	Ampere
CSI	Camera Serial Interface
GPIO	General Purpose Input Output
APILL	Alat Pemberi Isyarat Lalu Lintas
RC	Radio Control

ABSTRAK

Terkait dengan tingginya akan kebutuhan transportasi, maka salah satu dampaknya adalah kecelakaan lalu lintas. Berdasarkan statistik data kecelakaan dan pelanggaran lalu lintas kepolisian Republik Indonesia untuk bagian Daerah Istimewa Yogyakarta pada tahun 2017 tercatat terdapat 4011 jumlah kecelakaan dengan korban meninggal dunia hingga 442 jiwa, serta kerugian materi sebesar Rp 2.382.120.300, lalu untuk tahun 2018 tercatat 5061 kecelakaan dengan korban meninggal dunia 485 jiwa, serta kerugian sebesar Rp 406.952.975.000[1], dari data tersebut dikembangkan sebuah inovasi transportasi bernama *autonomous car*. *Autonomous Car* merupakan mobil yang dapat berjalan atau bergerak secara otomatis sesuai dengan yang diperintahkan oleh pemilik mobil, yang dikendalikan oleh sebuah sistem komputer. Sistem ini dibangun menggunakan algoritma pengenalan objek *Haar-Cascade Classifier* serta *image processing* sebagai pemroses sinyal masukan berupa gambar bergerak, dan dengan menggunakan Sharp sensor untuk mengukur jarak benda yang berada di depan mobil. Hasil dari penelitian ini sistem mampu berjalan sesuai dengan tujuan yang telah ditentukan secara *real time*, dengan akurasi rata-rata yang diperoleh oleh sistem dalam pendeteksian *traffic light* sebesar 95%, presisi 100%, sensitivitas 90%, dan *error* 5%, serta proses pembacaan marka jalan memiliki akurasi hingga 100%, dan untuk pengukuran jarak menggunakan Sharp Sensor GP2Y0A41SK0F didapatkan akurasi sebesar 98,45%. Hal ini tentu saja sangat berdampak positif bagi kehidupan manusia saat ini, dimana manusia dapat lebih produktif saat berkendara, serta dapat menjaga keselamatan dalam perjalanan.

Kata Kunci : kecelakaan, *Autonomous car, Haar-Cascade classifier, image processing*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Memasuki era digital ini, perkembangan di dunia teknologi sangat pesat serta diiringi juga dengan peningkatan akan kebutuhan beragam moda transportasi. Pesatnya pertumbuhan inovasi teknologi berbasis *Artificial Intelligence* di sektor kendaraan sangat terlihat, ditandai dengan diproduksinya mobil-mobil masal yang berteknologi tinggi. Terkait dengan tingginya akan kebutuhan transportasi, maka salah satu dampaknya adalah kecelakaan lalu lintas. Kecelakaan lalu lintas menurut UU RI NO. 22 tahun 2009 adalah suatu peristiwa di jalan raya yang tidak diduga dan tidak disengaja melibatkan kendaraan dengan atau tanpa pengguna jalan lain yang mengakibatkan korban manusia dan atau kerugian harta benda [1]. Berdasarkan statistik data kecelakaan dan pelanggaran lalu lintas kepolisian Republik Indonesia untuk bagian Daerah Istimewa Yogyakarta pada tahun 2017 tercatat terdapat 4011 jumlah kecelakaan dengan korban meninggal dunia hingga 442 jiwa, serta kerugian materi sebesar Rp 2.382.120.300, lalu untuk tahun 2018 tercatat 5061 kecelakaan dengan korban meninggal dunia 485 jiwa, serta kerugian sebesar Rp 406.952.975.000 [2], dari data tersebut dapat dikehatui bahwa terjadi peningkatan jumlah kecelakaan yang cukup pesat.

Seiring dengan bertambahnya jumlah mobil yang ada dan mengingat banyaknya angka kecelakaan yang terjadi di Indonesia, maka tentu tidaklah mudah untuk menguranginya, karena kecelekaan yang terjadi memiliki beberapa faktor, yang diantaranya ada kelalaian pengemudi atas ketertiban lalu lintas, kendaraan yang bermasalah, ataupun faktor-faktor lainnya. Dengan angka kecelakaan sebesar itu, maka seharusnya sudah ada upaya untuk menguranginya, salah satu contohnya adalahh *Autonomous Car*. *Autonomous Car* merupakan mobil yang dapat berjalan atau bergerak secara otomatis sesuai dengan yang diperintahkan oleh pemilik mobil, dengan begitu pemilik mobil bisa lebih merasa nyaman dan tenang saat berada di dalam mobil.

Autonomous car dapat mengurangi resiko kecelakaan jika persyaratannya terpenuhi, semisal marka jalan yang dibuat secara baik, kondisi jalan yang sesuai dengan peta dan sebagainya, oleh karena itu mengapa *full autonomous car* belum bisa dijalakankan di Indonesia disebabkan banyaknya persyaratan yang belum terpenuhi ditambah dengan kondisi pengendara di Indonesia yang kurang mematuhi peraturan berkendara dijalan raya, namun sudah banyak *semi-autonomous car* yang dipasarkan di Indonesia, dengan mengusung sistem *self-driving car* yang sudah memiliki sistem keamanan lebih, memungkinkan mobil memberi peringatan kepada pengemudi jika ada kendaraan yang terlalu dekat dengan mobil, mampu menjaga jarak aman mobil dengan kendaraan

didepannya, lalu ada *line keeping assist* yang berguna untuk menjaga mobil selalu pada jalurnya, terutama saat pengendara mengantuk dan terindikasi akan keluar jalur, serta sistem keamanan lainnya yang disematkan pada mobil tersebut. *Autonomous car* merupakan sebuah sistem yang dijalankan oleh komputer yang tidak memiliki perasaan mengantuk ataupun kelelahan seperti manusia, maka diharapkan mampu lebih meningkatkan keamanan saat berkendara.

Dengan begitu kecelakaan dapat di minimalkan secara bertahap, dikarenakan adanya inovasi yang terus berkembang dan diharapkan teknologi *Autonomous Car* tersebut dapat diterapkan di setiap mobil yang ada di dunia dan berfungsi secara sempurna, agar pelanggaran-pelanggaran yang sering dilakukan pengemudi dapat ditanggulangi. Penelitian terdahulu [3] yang berisikan tentang *prototype autonomous car* menggunakan *multi-layer perceptrons* (MLP) pada bagian pendeteksian jalur, serta *Haar-Cascade Classifier* yang diterapkan untuk pendeteksian lampu *traffic light* dan rambu berhenti sudah berhasil dilakukan dengan indikator mobil mampu berjalan sesuai dengan tujuan yang telah ditentukan, namun pada penelitian tersebut Raspberry Pi hanya digunakan sebagai pengirim gambar yang kemudian diolah gambarnya oleh laptop dan keputusannya dikirimkan ke Arduino yang akan mengendalikan remot dari mobil *radio control* (RC) tersebut, serta sensor ultrasonic yang belum diuji tingkat keakuratannya. Oleh karena itu pada penelitian ini peneliti ingin membuat sebuah *prototype autonomous car* yang mandiri dengan menggunakan sensor jarak yang akan diuji tingkat keakuratannya, kata ‘mandiri’ disini berarti seluruh sistem berada di mobil, baik dari pengolahan gambar hingga keputusan pergerakan mobil.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, maka rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimanan mendesain sistem *autonomous car* yang dapat memiliki kemampuan berjalan mengikuti garis marka, mendeteksi *traffic light*, dan mampu mendeteksi halangan yang ada di depan mobil secara *real-time* ?
2. Bagaimana hasil unjuk kerja sistem *autonomous car* terhadap masing-masing kemampuan yang dimilikinya ?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini diantaranya adalah:

1. Menggunakan Raspberry Pi 4 4GB RAM.
2. Menggunakan kamera Raspberry Pi V2.1.
3. Menggunakan Sharp Sensor GP2Y0A41SK0F sebagai pengukur jarak.
4. Pengambilan gambar atau pemrosesan citra dilakukan dengan cahaya yang cukup.
5. Lebar jalan 30cm ±5%.
6. Setiap tikungan terluar memiliki radius minimal 110 cm.
7. Pengambilan citra tidak boleh menghadap cahaya ataupun menggunakan karpet yang memiliki pantulan cahaya yang berlebih untuk jalur yang akan di deteksi.
8. Tinggi kamera dari permukaan jalur minimal 23cm ±5%.
9. Lebar badan mobil maksimal 17 cm.
10. Sudut kemiringan kamera dari tiang penyangga 160º ±5%.
11. Jarak tiang penyangga kamera dari bagian terdepan mobil 25cm ±5%.
12. Warna putih hanya digunakan untuk marka jalan.
13. Jalan yang datar.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang sistem *prototype autonomous car* yang mampu berjalan sesuai dengan tiga kemampuan utamanya, yakni berjalan mengikuti garis marka, mematuhi *traffic light*, dan mendeteksi halangan yang ada di depan mobil secara *real-time*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Mampu mengurangi angka kecelakaan dan pelanggaran lalu lintas yang disebabkan oleh kelalaian pengemudi.
2. Meningkatkan rasa aman bagi pengemudi dan pengguna jalan lainnya.
3. Sebagai pengemabangan di dunia otomotif.

BAB 2
TINJAUAN PUSTAKA

2.1 Studi Literatur

Penelitian serupa telah banyak dilakukan sebelumnya, salah satunya adalah *prototype autonomous car* menggunakan *neural network* yang dilakukan pada tahun 2017 lalu [4], dimana pada penelitian tersebut dibahas tentang penggunaan kamera untuk membaca lingkungan sekitar yang mana data akan diolah menggunakan algoritma *backpropagation neural network* dan *Haar-Like Classifie*r, adapun perangkat utama yang disimulasikan adalah Raspberry Pi, kamera Raspberry Pi, sensor ultrasonic, dan motor DC yang mana memiliki kegunaannya masing-masing. Hasil dari penelitian ini adalah mengetahui perbandingan penggunaan algoritma *Haar-Like Classification* dengan *backpropagation neural network,* dimana didapat kesimpulan bahwa penggunaan *Haar-Cascade Classifier* lebih tepat saat digunakan untuk mendeteksi objek yang tergolong jarang terlihat seperti rambu berhenti yang ditandai dengan keberhasilan pendeteksian rambu berhenti hingga 85%, karena algoritma ini memiliki keakuratan saat beradaptasi yang berbeda-beda. Namun untuk algoritma *backpropagation neural network* labih baik digunakan untuk deteksi objek yang sering terlihat seperti marka jalan atau bahu jalan, karena algoritma ini lebih mudah untuk mengenali pola objek, dimana kesimpulan tersebut didapat dari hasil uji coba yang menunjukan keberhasilan dalam pendeteksian pola jalan hingga 100%.

Pada tahun yang sama, yakni 2017 K.N.V.Satyanarayana dkk melakukan penelitian serupa, dimana penelitian tersebut berjudul *Based On Machine Learning Autonomous Car Using Raspberry-Pi* [3]. Pada penelitian [3] membahas tentang pembuatan *prototype autonomous rc car* dengan menggunakan algoritma *neural network* sebagai pendeteksi jalan dan *Haar-Cascade Classifier* untuk deteksi objek yang dalam hal ini adalah rambu stop dan *traffic light,* dengan hasil *testing* yang telah dilakukan didapatkan hasil berupa akurasi sebesar 85%, dengan hasil *training* didapatkan sebesar 96%. Lalu untuk rata-rata *error* dalam pengujian pengukuran jarak menggunakan kamera adalah 8.095%, hal tersebut bisa terjadi karena banyak faktor, yang diantaranya ada kesalahan dalam pengukuran nilai aktual, proses pendeteksian, kesalahan dalam kalibrasi kamera, dan lainnya. Namun seperti yang sudah diketahui bahwa semakin jauh jarak antara kamera dan objek, maka semakin besar *error*-nya.

Pada penelitian [4] telah melakukan implementasi tentang *Haar-Cascade Classifier* untuk *prototype system autonomous car* mereka, namun pada jurnal tersebut tidak ditampilkan hasil rancang bangun dari mobil *prototype* mereka, yang mana memberikan asumsi bahwa pendeteksian hanya dilakukan pada gambar statis yang sudah diambil dan dibuat sedemikian rupa agar sesuai

dengan kenyataan dijalan, seperti garis marka ataupun rambu jalan. Pada penelitian [3] mobil *prototype* telah diciptakan dengan basis dari mobil RC (*radio control*), tetapi hardware yang digunakan disini masih saling memiliki keterkaitan, hal yang dimaksud disini adalah bahwa Raspberry Pi yang terpasang pada mobil hanya digunakan sebagai pengirim gambar video yang terekam oleh kamera kepada komputer yang digunakan, serta dari komputer inilah yang memerintahkan sebuah Arduino untuk melakukan tugasnya sebagai pengendali remot dari mobil itu, jadi bisa dikatakan bahwa mobil tidak benar-benar bisa berjalan sendiri.

Dari tingginya nilai akurasi pada pendeteksian objek yang telah dicapai oleh peneliti [3] dan [4] menggunakan algoritma *Haar-Cascade Classifier*, maka mendorong peneliti untuk menggunakan algoritma yang sama pada pendeteksian *traffic light*, dan pemrosesan citra pada pendeteksian marka, serta peneliti ingin membuat sebuah sistem *prototype autonomous car* yang mampu berjalan secara mandiri, dengan tujuan-tujuan yang telah di paparkan pada 1.4.

2.2 Tinjauan Teori

Berikut adalah tinjauan teori yang digunakan untuk mendukung penelitian, dimana Pada bagian ini akan membahas tentang beberapa komponen atau piranti yang digunakan dalam membangun *system autonomous car*, lalu teori-teori dasar yang mendukung penelitian, seperti *Haar-Cascade Classifier*, OpenCV dan sebagainya.

2.2.1 *Machine Learning (ML)*

Machine Learning adalah jenis *Artificial Intelligence* (AI) yang memberikan komputer kemampuan untuk belajar tanpa diprogram secara eksplisit. Pembelajaran mesin berfokus pada pengembangan program komputer yang mampu beradaptasi atau paham saat diberikan dengan data baru, sistem ini mencari melalui data untuk menemukan pola. ML menggunakan data tersebut untuk mendeteksi pola dalam data dan menyesuaikan tindakan sesuai program. Algoritma ML sering dikategorikan menjadi 2 bagian, yakni diawasi atau tidak diawasi. Algoritma yang diawasi dapat menerapkan apa yang telah dipelajari di masa lalu ke data baru, sebagai contoh pada pendeteksian kendaraan mobil, saat sistem sudah berhasil melalui tahap pembelajaran, maka sistem sudah mampu mendeteksi mobil saat dikenakan gambar tersebut, baik gambar baru maupun lama yang berbentuk mobil. Algoritma yang tidak diawasi dapat menarik kesimpulan dari dataset, sebagai contoh pada bagian laman facebook menggunakan ML untuk memberikan rekomendasi masing-masing anggota, Jika sebuah anggota sering membaca atau "menyukai" suatu posting teman tertentu, maka rekomendasi yang akan diberikan sejenis atau serupa dengan hal yang sering dilihat tersebut [3].

2.2.2 OpenCV

OpenCV merupakan singkatan dari *open source computer vision* dimana OpenCV sendiri adalah sebuah API (*Application Programming Interface)* yang didalamnya memuat banyak struktur data serta algoritma yang diperlukan untuk melakukan *real-time computer vision* atau pengolahan citra gambar secara efisien dan cepat. Pada umumnya library ini dipergunakan oleh para pengguna linux dan windows dengan coding berbasis python, C++, maupun java untuk mengakses kemampuan dari camera itu sendiri, namun didalamnya juga berisikan sekumpulan algoritma yang nantinya akan difungsikan untuk pengolahan citra.

2.2.3 Citra Digital

Citra digital merupakan sebuah gambar dua dimensi yang secara garis besar dapat diolah oleh komputer yang terbangun dari titik-titik warna yang disebut sebagai piksel, piksel sendiri merupakan satuan terkecil dari sebuah gambar digital. Secara umum citra digital dapat digolongkan kedalam tiga jenis berdasarkan variasi warnanya, yakni:

a. Citra warna (RGB)

   Citra warna ini paling umum ditemui dalam kehidupan sehari-hari, karena memiliki imformasi yang paling lengkap untuk merepresentasikan dunia analog atau dunia sebenarnya, dimana citra ini mengusung sistem penggabungkan tiga kanal warna, yakni merah (R), hijau (G), dan biru (B) untuk membentuk warna lainnya. Setiap kanal tersebut mempunyai intensitas piksel dengan kadalaman sebesar 8bit atau 2^8 yang berarti bahwa setiap warna dapat memiliki variasi intensitas sebanyak 256 atau 0 hingga 255.

   Pada citra warna ini, setiap piksel merupakan gabungan dari ketiga kanal tersebut, dengan kata lain setiap piksel dapat mengeluarkan variasi warna sebesar 255*255*255 = 16.777.216, hal ini yang menyebabkan citra warna dapat merepresentasikan dunia analog dengan hasil paling baik, serta dari hasil ini yang menjadi asal mula standar untuk layar, atau yang biasa dituliskan dengan spesifikasi display 16 juta warna.

b. Citra *grayscale*

   Citra *grayscale* adalah citra yang hanya memiliki intensitas warna pada derajat keabuan, citra ini sering digunakan dalam proses pendeteksian objek karena hanya perlu sedikit informasi untuk menjelaskan arti pada tiap pikselnya, yang berarti akan memperkecil proses perhitungan dalam frame, dimana derajat keabuan pada citra ini ditentukan dengan rata-rata dari gabungan citra warna. Pada citra *grayscale* dengan intensitas 8bit, warna hitam sampai dengan putih dibagi kedalam 256 derajat keabuan, dimana 0 merepresentasikan warna hitam, dan 255 merepresentasikan warna putih.

$$Grayscale = \frac{\mathrm{R} + \mathrm{G} + \mathrm{B}}{3} \quad (2.1)$$

Keterangan :

R : citra warna merah

G : citra warna hijau

B : citra warna biru

c. Citra biner

Citra biner merupakan citra dengan informasi paling sedikit, karena hanya memiliki intensitas kedalaman 1bit atau 2^1 untuk merepresentasikan sebuah warna, yakni 0 untuk hitam dan 1 untuk putih, citra biner dapat dibentuk dengan mudah dari citra *grayscale* dengan metode *thresholding* atau ambang batas, dimana dalam proses ini diwajibkan untuk menentukan nilai ambang batas guna mengkonversi citra *grayscale*, semisal nilai batas adalah 100, maka piksel dengan nilai intensitas dibawah 100 akan menjadi hitam dan diatas 100 akan menjadi putih.

Metode ini umum digunakan untuk segmentasi citra, dimana dapat dengan mudah memisahkan daerah yang kita kehendaki dengan yang tidak dikehendaki untuk mendapatkan pola suatu objek.

$$b(x,y) = \begin{cases} 1 \; if \; g(x,y) > th \\ 0 \; if \; g(x,y) \leq th \end{cases} \quad (2.2)$$

Keterangan :

$b(x,y)$: piksel hasil citra biner

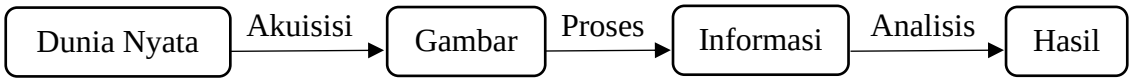
$g(x,y)$: piksel citra asli

th : nilai *threshold*

2.2.4 *Computer Vision*

Computer vision (CV), sesuai dengan namanya yang berarti pengelihatan komputer adalah salah satu bidang dari *Artificial Intelligence* (AI) yang difokuskan untuk memberikan pemahaman kepada komputer agar bisa mengerti akan objek yang diinginkan, dimana hal tersebut akan terjadi jika komputer berhasil mengambil informasi yang ada pada gambar yang dideteksi. Berdasarkan penjelasan tersebut maka dapat ditarik informasi bahwa CV bertujuan untuk meniru kerja mata pada manusia yang diimplementasikan kedalam citra digital.

Terdapat tiga langkah utama yang wajib dilakukan agar tujuan dari CV ini tercapai, adapun langkah tersebut harus berjalan secara berurutan seperti pada Gambar 2.1 berikut.



Gambar 2.1 Alur *computer vision*

Dimana langkah awal yang perlu dilakukan adalah pengambilan gambar (akuisisi), lalu pemrosesan dari data digital (matriks) yang didapatkan, dan terakhir adalah pemahaman atau analisis dari gambar tersebut.

Akuisisi gambar merupakan serangkaian proses yang dilakukan oleh komputer untuk merubah citra analog di dunia nyata menjadi sekumpulan data biner yang berisikan angka 0 dan 1 sebagai representasi dari citra digital. Perangkat keras yang berfungsi sebagai akuisisi gambar diantaranya ada kamera digital, *embedded camera, webcam*, dan sebagainya.

Alur kedua yakni pemrosesan citra, pada tahap ini pemrosesan yang dilakukan adalah tingkat rendah, dimana data yang sudah didapatkan akan diberikan algoritma untuk menggali informasi yang membangun sebuah gambar tersebut, pemrosesan tingkat rendah ini bertujuan untuk mencari garis tepi, segmen ataupun titik yang secara keseluruhan geometriknya akan membangun sebuah gambar yang utuh. Algoritma yang sering digunakan dalam proses ini diantaranya ada *edge detection dan segmentation.*

Tahap terakhir adalah analisis gambar, tahap ini merupakan pemrosesan gambar tingkat tinggi, dimana tahap satu dan dua akan digabungkan dengan sebuah algoritma yang akan memberikan pemahaman kepada komputer tentang apa yang dilihatnya, contoh implementasinya adalah *object tracking, object detection, dan object recognition* [5].

2.2.5 *Canny Edge Detection*

Canny edge detection adalah salah satu algoritma didalam OpenCV yang digunakan untuk mencari garis tepi dari sebuah citra, algoritma ini dikembangkan oleh John F. Canny pada tahun 1986 dengan menyebutkan bahwa ada tiga kriteria utama yang ingin dicapai, yakni:

a. Pendeteksian yang baik

   Memiliki tingkat kesalahan yang rendah, yang dapat diartikan bahwa pendeteksian hanya terjadi pada tepi yang ada.

b. Lokalisasi yang baik

   Jarak garis tepi yang terdeteksi dengan garis tepi sebenarnya harus memiliki jarak yang sangat kecil atau hingga bisa dikatakan sama.

c. Respon yang jelas

   Hanya terdapat satu respon untuk setiap tepi yang terdeteksi, sehingga tidak timbul kerancuan pada proses pengolahan citra selanjutnya [6].

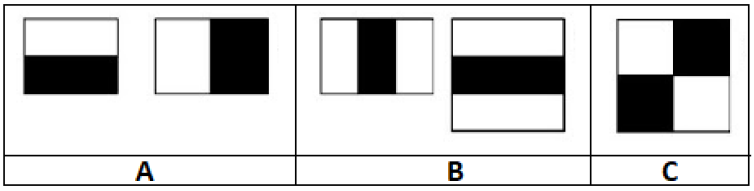
2.2.6 *Segmentation*

Segmentation atau dalam bahasa Indonesia berarti segmentasi adalah metode yang digunakan untuk memisahkan citra yang dikehendaki (*foreground*) dengan citra yang tidak dikehendaki (*background*), tujuan segmentasi sendiri untuk menyederhanakan representasi suatu citra menjadi sesuatu yang lebih bermakna dan lebih mudah untuk dianalisis. Metode segmentasi paling sederhana disebut *thresholding*, metode ini didasarkan pada variasi intensitas warna antara piksel objek dengan piksel latar belakang, untuk membedakan *foreground* dengan *background* digunakan nilai ambang batas, dimana nilai ambang adalah kunci utama dalam keberhasilan segmentasi ini, semisal segmentasi dilakukan pada citra RGB, maka dapat dilakukan pembatasan nilai atas dan bawah terhadap masing-masing kanal yang membangun citra tersebut.

2.2.7 *Haar-Cascade Classifier*

Haar-Cascade Classifier atau yang serinng disebut juga dengan *Haar-Like Feature* sangat umum digunakan untuk pendeteksian wajah atau yang dikenal sebagai *face detection*, algoritma ini dikembangkan oleh Viola dan Jones dalam jurnalnya yang berjudul *Rapid Object Detection using a Boosted Cascade of Simple Features* pada tahun 2001 silam [4]. Setidaknya dibutuhkan empat tahap untuk membangun kerangka utama algoritma ini [7], yakni:

a. *Haar-Like Feature* pada dasarnya bertujuan untuk menemukan objek dengan menggunakan beberapa jenis fitur yang ada (Gambar 2.2), bukan menggunakan seluruh piksel yang didapat, hal ini yang menyebabkan perhitungan dapat dilakukan dengan cepat, karena hanya dilakukan perhitungan jumlah piksel dalam fitur.



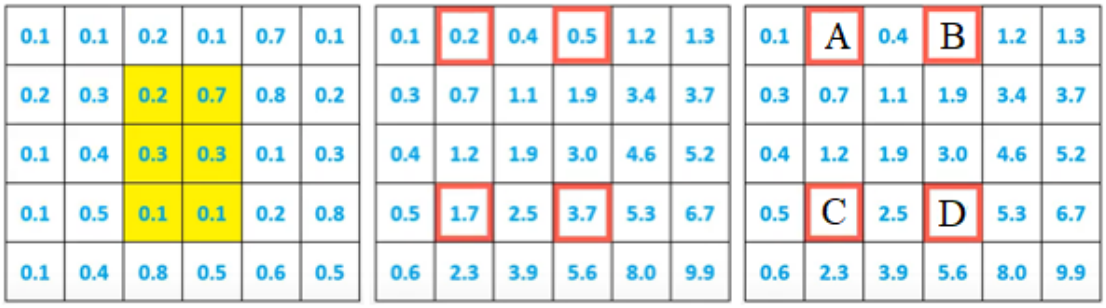
Gambar 2.2 Jenis-jenis fitur

Fitur yang digunakan pada algoritma ini ada tiga jenis, yakni *edge feature* (A), *line feature* (B), dan *four rectangle feature* (C). fitur ini menggunakan bentuk gelombang Haar yang berbentuk persegi secara dua dimensi dengan warna terang dan gelap yang saling berdampingan. Perhitungan nilai untuk setiap fitur adalah pengurangan dari total nilai daerah gelap dan terang. Pada kasus citra bergerak (video), perhitungan fitur dilakukan pada seluruh piksel secara terus-menerus yang berakibat pada lambatnya waktu pemrosesan, oleh karena itu digunakan *integral image* untuk mempercepat proses tersebut.

b. *integral image* merupakan citra yang nilai pikselnya merupakan jumlahan dari piksel kiri dan atasnya, dimana dengan menggunakan *integral image* perhitungan pada fitur Haar akan menjadi lebih cepat. Secara umum integral sendiri bermakna sebagai penambahan bobot, dengan bobot sendiri merupakan nilai dari setiap piksel yang akan ditambahkan dalam sebuah citra asli. Gambar 2.3 merupakan contoh dari perhitungan *integral image.*

Gambar 2.3 Perhitungan *integral image*

integral image digunakan untuk meningkatkan kecepatan dari perhitungan setiap fitur dalam gambar, dengan menggunakan 4 titik referensi (Gambar 2.4), maka nilai salah satu dari daerah fitur diperoleh, dengan begitu akan menghilangkan perhitungan yang berat atau besar, dan menyebabkan perhitungan menjadi konstan di seluruh sisinya, lalu untuk mendapatkan nilai dari salah satu sisi fitur dapat dengan persamaan (2.3).



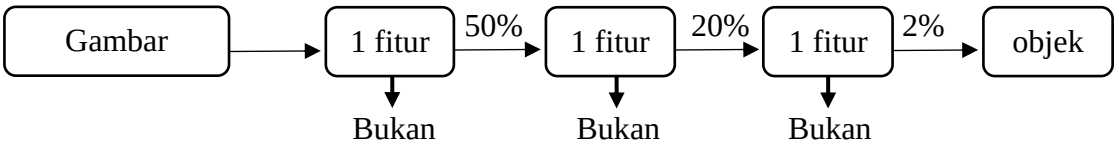
Gambar 2.4 Penjumlahan piksel pada setiap kotak

$$daerah\ kuning = D - (B + C) + A \quad (2.3)$$

Pemilihan fitur yang digunakan untuk pendeteksian objek dan penentuan ambang batas nilai (*threshold*) adalah metode dari *machine learning* yang disebut sebagai *AdaBoost (Adaptive Boosting)* [5].

c. *Adaptive Boosting* berperan dalam pemilihan fitur secara spesifik dan melakukan penentuan nilai ambang batas yang akan digunakan, *AdaBoost* akan menggabungkan beberapa *classifier* lemah untuk menjadi sebuah *classifier* kuat, dimana arti lemah disini adalah urutan dari filter pada *classifier* hanya mendapat sedikit jawaban benar, jadi jika keseluruhan *classifier* lemah digabungkan dan ditambahkan dengan sejumlah bobot, maka akan menjadi *classifier* yang kuat. Viola-Jones menerapkan beberapa *AdaBoost classifier* menjadi sebuah rangkaian filter yang cukup efisien dalam menggolongkan daerah gambar yang akan dideteksi.

d. *Casecade classifier* adalah sebuah algoritma yang akan menggabungkan beberapa klasifikasi yang telah terlatih oleh data positif dan negatif. Untuk dapat meningkatkan kecepatan pendeteksian maka pencarian akan dipusatkan pada objek yang akan di deteksi saja, hal ini tidak menutup kemungkinan untuk mengetahui letak objek, algoritma ini memiliki beberapa tahapan, dimana setiap tahapan terdiri dari pembelajaran lemah. Algoritma ini akan menghasilkan beberapa sub-citra yang bukan objek, hal ini dilakukan karena pendeteksian yang bukan termasuk objek lebih mudah dilakukan.



Gambar 2.5 Proses *Cascade classifier*

Seperti pada Gambar 2.5 tahapan pertama gambar akan diberikan salah satu fitur dari objek yang akan dideteksi, lalu jika tidak termasuk kedalam fitur tersebut maka dapat diasumsikan bahwa piksel tersebut bukan objek, dari tahap awal ini setidaknya akan menghilangkan 50% subcitra, lalu seiring proses pendeteksian maka akan bertambah juga jumlah fitur yang dipergunakan, dan akan menyisakan setidaknya 2% subcitra yang lolos dan hasil akhir pendeteksian adalah gambar yang memenuhi proses *AdaBoost.*

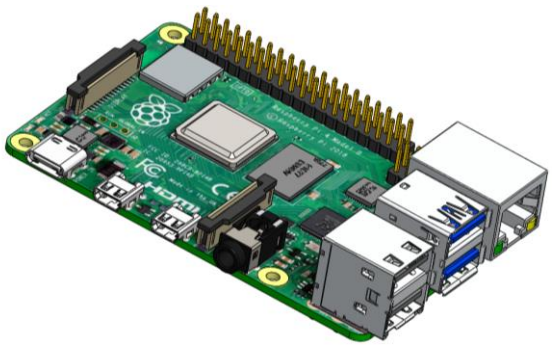
2.2.8 Raspberry Pi 4

Secara umun Raspberry Pi (RPi) merupakan *single board computer* (SBC) dengan ukuran yang sangat *compact*, berbeda dengan mini PC, Raspberry Pi sendiri sudah memiliki GPIO digital dengan tegangan kerja untuk setiap GPIO adalah 3.3V. Sampai saat ini sudah mengeluarkan 5 model dengan spesifikasi yang berbeda-beda setiap serinya yakni RPi zero, RPi A series, RPi 2 series, RPi 3 series, dan RPi 4 series, secara umum sistem operasi pada Raspberry Piditanamkan pada micro SD ataupun SDHC, sistem operasinya sendiri merupakan raspbian OS yang berbasis linux dengan segala macam kemudahan aksesnya, dan program yang digunakan berbasis python.

Tabel 2.1 Spesifikasi RPi 4

No	Hardware
1	Quad core 64-bit ARM-Cortex A72 running at 1.5GHz
2	4 Gigabyte LPDDR4 RAM
3	Supports dual HDMI display output up to 4Kp60
4	H.264 hardware decode (up to 1080p60)
No	**Interface**
5	802.11 b/g/n/ac Wireless LAN
6	1x Raspberry Pi camera port (2-lane MIPI CSI)
7	28x user GPIO supporting various interface option
8	2x USB3 ports
9	1x Gigabit Ethernet port (supports PoE with add-on PoE HAT)

Dengan spesifikasi Tabel 2.1 sudah cukup dalam pembuatan *prototype* ini, namun tentu saja dengan spesifikasi yang lebih tinggi akan membuat komputasi menjadi lebih cepat serta bisa mengakomodasi data yang lebih besar. Raspberry Pi 4 sendiri bisa dilihat pada Gambar 2.6.



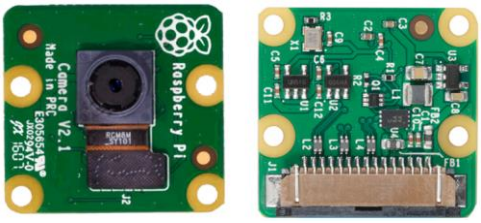
Gambar 2.6 Raspberry Pi 4

2.2.9 Kamera Raspberry Pi V2.1

Gambar 2.7 merupakan kamera yang digunakan pada penelitian ini, Pi camera merupakan salah satu jenis kamera yang *compatible* dengan seluruh model Raspberry Pi yang memiliki port *Camera Serial Interface* (CSI). Modul kamera dengan resolisi 8mp ini mampu merekam video dan gambar 1080p30 dan terhubung langsung ke Raspberry Pi, kamera ini sudah mengalami perkembangan yang jauh dari V1 sebelumnya, dimana untuk V1 masih memiliki 5mp. Kamera ini dapat dengan mudah digunakan, dimana hanya perlu memasanag kabel pita yang disertakan ke port CSI pada Raspberry, dan sedikit melakukan konfigurasi. Kamera ini dipilih karena memiliki ukuran yang tergolong kecil, yakni sekitar 25 x 20 x 9mm dan beratnya hanya di bawah 3g. Adapun spesifikasi dari kamera ini ditampilkan pada Tabel 2.2.

Tabel 2.2 Spesifikasi kamera

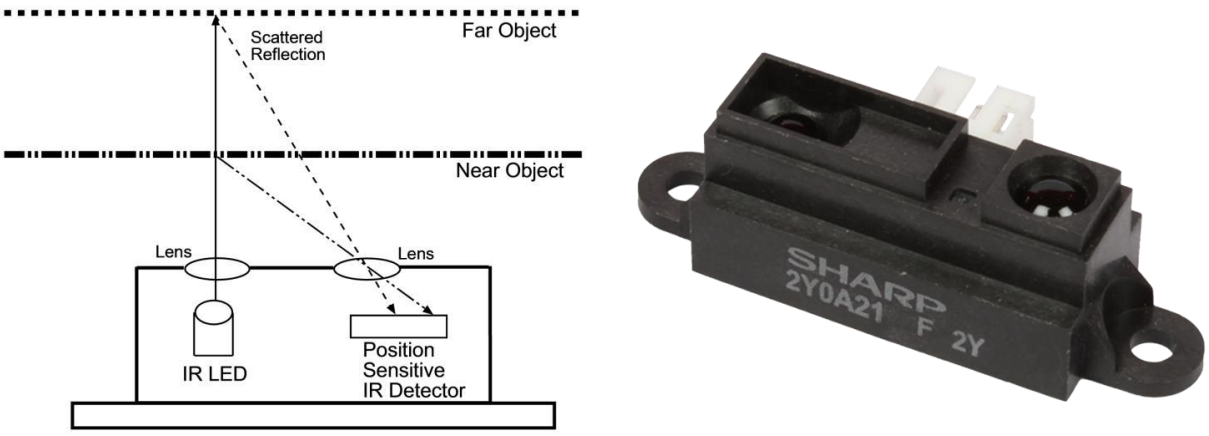
No	Parameter	Spesifikasi
1	Linux integration	V4L2 driver available
2	Sensor	Sony IMX219
3	Resolution	3280 × 2464 pixels
4	Optical size	1/4”
5	Horizontal field of view	62.2 degrees
6	Vertical field of view	48.8 degrees



Gambar 2.7 Kamera raspi V2.1

2.2.10 Sharp Sensor GP2Y0A41SK0F

Sharp Sensor GP2Y0A41SK0F merupakan salah satu sensor jarak yang mudah ditemui di Indonesia, jarak ukur dari sensor ini bisa mencapai 30cm (Tabel 2.3) tergantung jenis yang digunakan, semakin jauh jarak sensornya maka semakin besar juga minimal pembacaan sensornya, dimana yang digunakan dalam penelitian ini memiliki range 4-30cm. Prinsip kerja dari sensor ini menggunakan cahaya inframerah, seperti pada Gambar 2.8 terlihat ada sebuah lampu inframerah yang berbentuk seperti LED yang berfungsi sebagai pemancar, dan ada penerimanya persis di sebelah lampu, serta terdapat dua buah lensa kecil untuk memfokuskan lampu inframerah tersebut. Proses pengukuran jarak berdasarkan pada durasi dari pengiriman sinyal inframerah hingga kembali lagi ke penerima, wujud sensor ini bisa dilihat pada Gambar 2.8.



Gambar 2.8 Prinsip kerja dan wujud sharp sensor

Tabel 2.3 Spesifikasi sharp sensor

No	Parameter	Spesifikasi
1	Tipe sensor	Analog
2	Operating voltage	4,5 to 5,5 V
3	Measuring distance range	4 to 30 Cm
4	Output voltage	0,3 to 3,05 V
5	Supply current	12 to 22 mA
6	Operating temperature	-10ºC to 60ºC

BAB 3
METODOLOGI

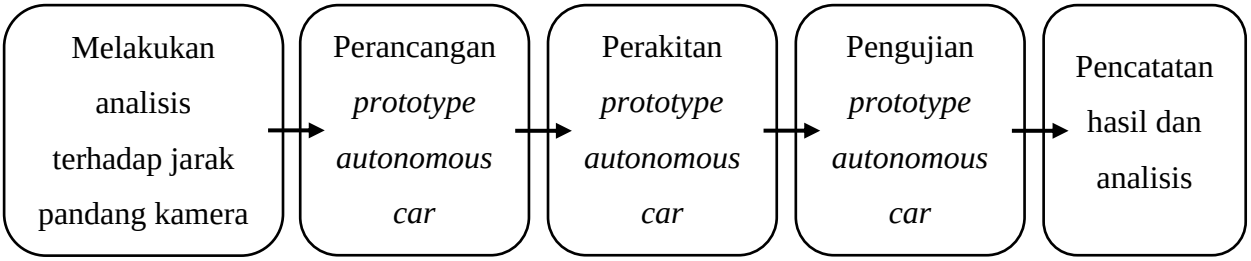
3.1 Alat dan Bahan

Alat dan bahan yang dipergunakan dalam menyelesaikan penelitian ini diantaranya adalah:

1. Raspberry Pi 4 (4GB)
2. Camera Raspberry Pi V2.1
3. microSD sandisk ultra 16GB
4. MN D90 kit version
5. Motor DC 6V
6. Mini servo 12g
7. ESC 20a
8. Dfrobot Beetle (smallest Arduino Leonardo board)
9. Sharp Sensor GP2Y0A41SK0F
10. IC ADC MCP3008
11. Wild Turbo Fan (WTF)
12. PVC 1mm
13. Voltmeter*5
14. Li-ion Battery *4
15. Module charger Li-ion *4
16. Mini switch*7
17. Diton paint black
18. Aerox paint gray
19. Kabel
20. Step down 3A*2
21. Step down 5a
22. Karpet sebagai jalur mobil
23. Lem polyacryl dan lem G (lem korea)
24. Bahasa Python 3
25. Raspbian OS
26. *Library* yang digunakan :
    a. OpenCV 3.4.1
    b. Pyfirmata
    c. Numpy

3.2 Alur Penelitian

Dalam suatu penelitian tentu saja tidak lepas dari susunan waktu, dimana susunan tersebut akan terealisasi jika memiliki alur penelitian yang baik juga. Adapun alur penelitian yang peneliti gunakan dapat dilihat pada Gambar 3.1



Gambar 3.1 Alur penelitian

Diagram blok tersebut menunjukan secara garis besar alur penelitian ini, adapun tahap pertama yang peneliti lakukan adalah melakukan analisis terhadap jarak pandang kamera, dimana salah satu komponen terpenting dari *prototype* ini adalah kamera itu sendiri, hal ini sangat penting dilakukan agar peletakan dari kamera pada mobil tidak terjadi kesalahan, dan dari hasil analisis ini kita dapat menentukan jenis mobil apa yang akan digunakan dan dimensi jalur yang sesuai dengan keadaan mobil.

Tahap kedua adalah melakukan perancangan, dimana perancangan ini berpatokan terhadap posisi kamera tersebut, hal ini dilakukan karena kamera dan beberapa komponen lainnya saling berkaitan dengan menggunakan kabel pita yang hanya 15 cm, serta perancangan terhadap sistem yang akan digunakan untuk mengakomodir tujuan dari penelitian ini.

Tahap ketiga adalah perakitan mobil itu sendiri, dimana pada tahap perancangan peneliti sudah membuat model 3D dari *protoype* ini, maka pada tahap ini peneliti merealisasikan model tersebut dengan menggunakan bahan PVC dan melakukan beberapa perubahan pada *chassis* maupun *body* mobil RC agar pemasangan komponen bisa presisi dan mudah untuk dibongkar-pasang guna keperluan lainnya.

Tahap keempat adalah melakukan pengujian terhadap hasil *prototype* ini, dimana pengujian dilakukan dengan cara menjalankan *coding* satu-persatu, lalu setelah semua dirasa sudah benar, *coding* tersebut digabungkan dan mobil dijalankan secara terus menerus hingga *battery* tidak mempu lagi menjalankan sistem.

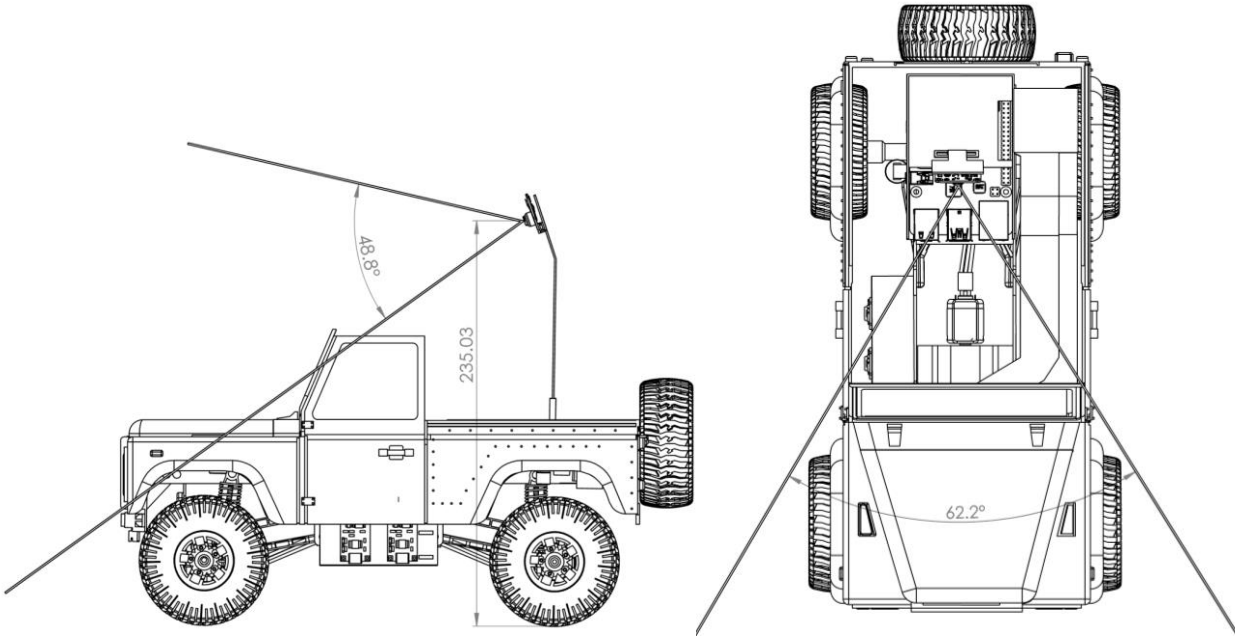
Tahap kelima adalah pencatatan hasil dan analisis, dimana peneliti mencatat hasil dari keberhasilan pendeteksian objek *traffic light*, pendeteksian marka jalan yang masing-masing dilakukan 10 kali pengulangan untuk kemudian ditentukan nilai akurasinya, mencatat hasil pembacaan jarak oleh sharp sensor yang dibandingkan dengan jarak aktual, analisis pengereman, dan terakhir melakukan analisis terhadap ketahanan pada sistem *autonomous car* ini.

3.3 Perancangan Sistem

Pada bagian ini peneliti menjelaskan tentang bagaimana sistem *prototype* autonomous car ini bisa terbentuk.

3.3.1 Penentuan Posisi Kamera

Peletakan posisi kamera sangat penting ditentukan, karena hampir seluruh sistem dari mobil ini mengandalkan kamera sebagai masukannya, bersesuaian dengan Tabel 2.2 yang mengatakan bahwa sudut kemiringan penangkapan gambar secara horizontal sebesar 62.2 derajat dan vertikal 48.8 derajat, oleh karena itu posisi kamera diletakan seperti pada Gambar 3.2 dan hasil penangkapan gambar pada Gambar 3.3.



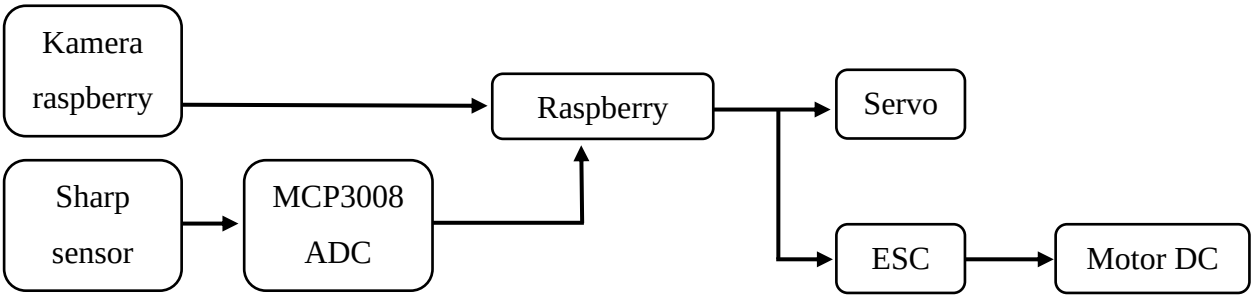
Gambar 3.2 Posisi kamera

Gambar 3.3 Hasil penangkapan gambar

Posisi demikian bukanlah yang terbaik, karena seharusnya titik ujung penangkapan kamera berada persis disebelah ban depan mobil, hal tersebut akan menguntungkan sistem karena pembacaan tikungan tidak lebih dulu dibaca dibandingkan posisi mobil, namun hal itu tidak bisa dilakukan mangingat kabel pita hanya 15cm dan mengurangi estetika mobil jika kamera dipasang terlalu tinggi, bisa juga dengan cara menundukan kamera lebih dalam, namun pengelihatan kedepan menjadi terbatas dan sangat mengganggu pendeteksian *traffic light*.

3.3.2 Alur Kerja Sistem (*Hardware*)

Secara garis besar alur kerja sistem dapat dilihat pada diagram blok Gambar 3.4 dimana sistem ini diawali dengan pembacaan dua buah sensor, yakni kamera dan inframerah, dari hasil pembacaan itulah yang berikutnya akan diproses oleh Raspberry Pi dan dijadikan sebuah perintah atau keputusan yang akan diterjemahkan oleh aktuator.



Gambar 3.4 Alur kerja sistem

Karena sensor inframerah memiliki keluaran analog sedangkan GPIO Raspberry Pi adalah digital, maka dibutuhkan ADC untuk menerjemahkan sinyal tersebut, lalu untuk menjalankan mobil dibutuhkan sebuah motor DC 6V, dimana putaran motor ditentukan berdasar tegangan dan polaritasnya, sedangkan didalam GPIO RPi terdapat pin pwm, namun bekerja pada tegangan 0 sampai +3.3V yang tentu saja kurang untuk menjalankan motor dan tidak dapat digunakan untuk memutar balik putaran motor DC, maka ditambahlah sebuah ESC *(Electronic Speed Controller)* untuk mengatasinya, tetapi untuk memerintahkan ESC dan servo sendiri membutuhkan tegangan kerja pwm 0V hingga 5V, maka digunakan Arduino Leonado untuk membuat perintah tersebut.

3.3.3 Dataset (*Sample* Gambar)

Data yang digunakan oleh peneliti dalam penelitian ini adalah citra *prototype traffic light* atau alat pemberi isyarat lalu lintas (APILL) yang diambil menggunakan kamera Raspberry Pi V.2, dimana pengambilan citra dilakukan dengan cara perekaman video yang berikutnya dipotong pada bagian yang berisikan lampu isyaratnya untuk seluruh dataset. Secara garis besar citra dalam dataset ini terbagi manjadi 2 bagian, yakni:

1. Citra positif

   Kumpulan citra positif merupakan kumpulan citra yang ingin dideteksi atau dikenali, dalam hal ini adalah *traffic light*, citra ini digunakan untuk menentukan apa yang sebenarnya ingin dicari. Ada dua cara untuk mendapatkan citra ini:

   a. Dengan cara melakukan foto satu persatu *traffic light*, cara ini mudah untuk dilakukan, namun memakan waktu yang sangat lama.
   b. Cara kedua dengan pengambilan video yang berikutnya dipotong secara manual menggunakan aplikasi, cara ini efisien karena kita hanya perlu memilih sekiranya bagian mana yang ingin dideteksi.

2. Citra negatif

   Merupakan citra yang bukan bagian dari citra positif, dalam hal ini citra negatif berfungsi sebagai latar belakang dari citra positif. Banyak cara untuk mengumpukan citra ini, bisa dengan cara seperti citra positif maupun pengambilan secara acak pada internet, namun alangkah lebih baiknya jika citra ini relevan terhadap situasi sebenarnya dari citra positif, semisal pendetaksian *traffic light* maka *background* (citra negatif) yang diambil adalah jalanan, mobil dan sebagainya yang berhubungan.

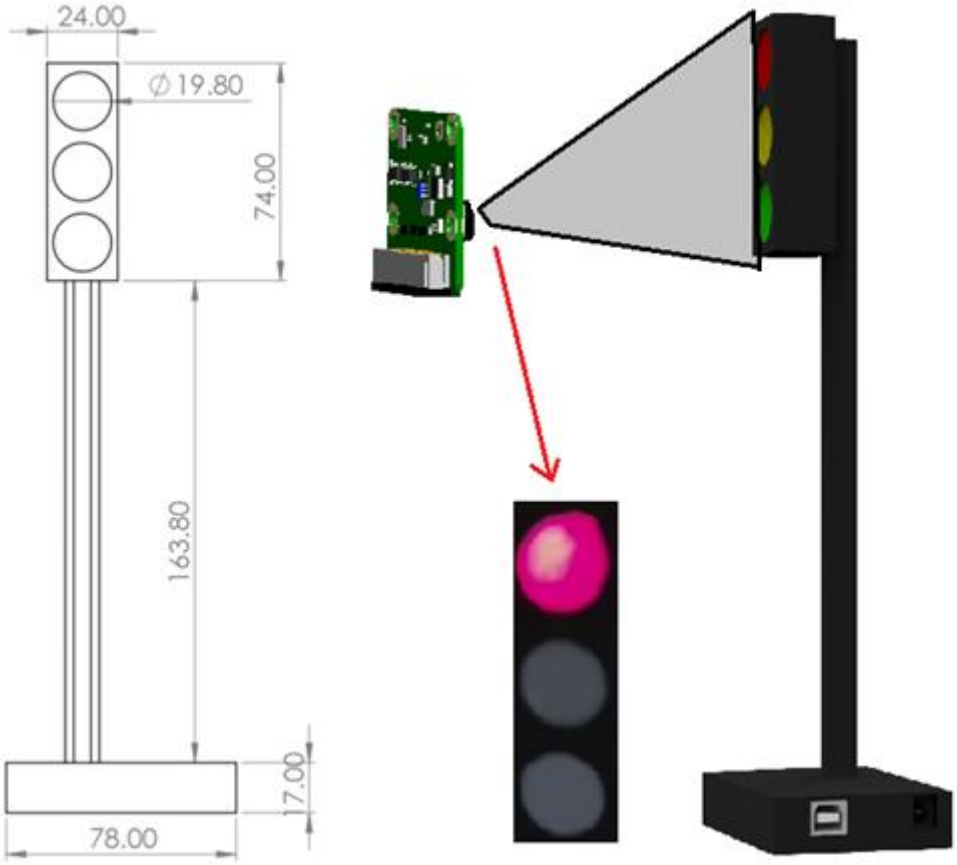
Adapun citra positif yang peneliti kumpulkan dilakukan dengan menggunakan kamera Raspberry Pi, hal ini dilakukan karena jika pengambilan gambar dari *smartphone,* maka hasilnya jauh lebih baik dibandingkan dengan kamera Raspberry Pi, oleh karena itu pengambilan dataset bertujuan agar Raspberry Pi mampu mendeteksi objek yang sesuai dengan penangkapan dari kamera Raspberry Pi sendiri. Adapun jumlah dataset dapat dilihat pada Tabel 3.1.

Tabel 3.1 Pembagian dataset

No	Dataset	Jumlah
1	Positif lampu merah	50
2	Positif lampu kuning	50
3	Positif lampu hijau	50
4	Negatif	50
5	Data uji positif untuk masing-masing warna lampu	50
6	Data uji negatif untuk masing-masing warna lampu	50

Citra yang digunakan adalah citra *traffic light* yang diambil dari sudut pandang tagak lurus dari bagian depan lampu dengan pencahayaan yang cukup untuk melihat ketiga pola dari bentuk masing-masing lampu isyarat tersebut. Untuk mengantisipasi kegagalan sistem yang dikarenakan beratnya pendeteksian gambar ini, maka dataset positif yang digunakan tidak terlalu banyak, namun berdampak pada kurangnya akurasi pendeteksian atau bisa dikatakan kurang luasnya

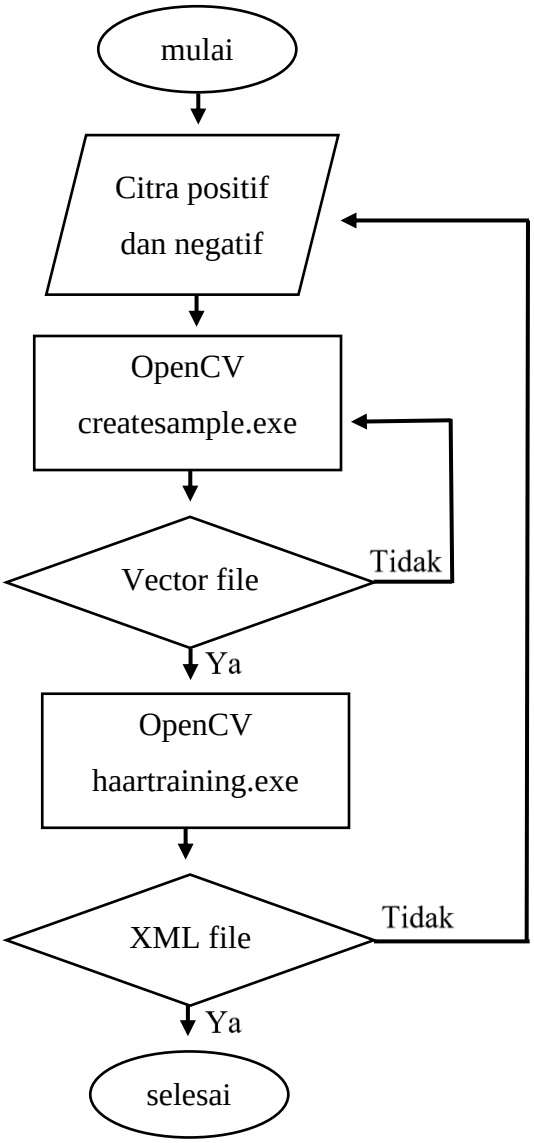
pemahaman mesin dalam penbelajaran ini, karena dataset yang digunakan hanya mampu mengakomodir bagian depan lampu secara tegak lurus dengan pencahayaan yang sangat cukup, jadi kelemahan yang didapat adalah tidak mampunya sistem dalam mendeteksi *traffic light* yang memiliki kemiringan berlebih serta minimnya cahaya penerangan yang berakibat pada mendominasinya kecerahan dari lampu APILL tersebut terhadap kamera yang digunakan, namun sisi baiknya sistem tidak bekerja terlalu keras untuk mencari objek yang dideteksi, untuk mengatasi kurangnya pemahaman sistem, maka diberlakukan cara yang akan dibahas pada 4.3. Ilustrasi pengambilan dataset serta ukuran *pototype traffic light* dapat dilihat pada Gambar 3.5.



Gambar 3.5 *Prototype traffic light*

3.3.4 Alur Proses Kerja Pembelajaran *Haar-Cascade Classifier*

Sebelum sistem bisa mengenali sebuah objek, tentu saja ada pembelajaran terlebih dahulu yang dilakukan, adapun alur pembelajarannya sendiri dapat dilihat pada Gambar 3.6, yang berikutnya dilakukan validasi dengan beberapa gambar positif dan negatif pada proses pembelajaran tersebut, hasil pembelajaran ini bisa langsung diterapkan menggunakan kamera Raspberry Pi secara *real-time*. Pada tahap ini peneliti menggunakan gambar positif (*traffic light)* sebanyak 50 buah dan negative (bukan *Traffic light*) sebanyak 50 buah untuk setiap warna lampu, sample gambar tersebut bisa berubah jumlahnya sesuai kebutuhan.

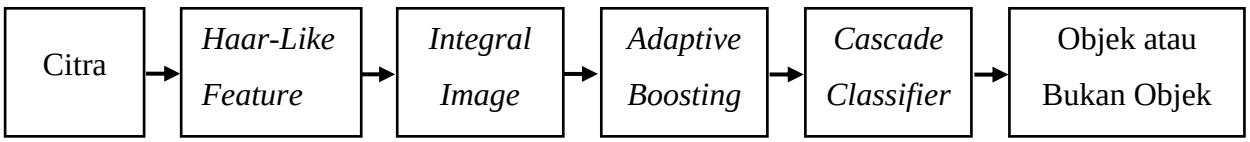


Gambar 3.6 Pembelajaran *Haar-Cascade Classifier*

Hasil dari pembelajaran ini berupa file XML yang bisa langsung diterapkan menggunakan library OpenCV [8].

3.3.5 Alur Pendeteksian *Traffic Light*

Alur pemdeteksian menggunakan algoritma *Haar-Cascade Classifier* pada *traffic light* dari awal proses hingga hasil pendeteksian ditunjukan pada Gambar 3.7.



Gambar 3.7 Alur pendeteksian *traffic light*

Pada tahapan pertama yaitu pembacaan sampel citra *traffic light*, yang berikutnya dari citra tersebut dilakukan pembacaan fitur *Haar* dengan cara mengolah citra tersebut menjadi persegi panjang utnuk mencari perbedaan nilai antara daerah hitam dan putih, dimana jika nilai tersebut lebih kecil dibandingkan dengan nilai ambang batasnya, maka dapat dikatakan bahwa fitur tersebut tidak ada.

Selanjutnya pada tahapan ketiga adalah penggunaan *integral image* untuk menentukan keberadaan dari fitur *Haar* pada sebuah citra dengan skala yang berbeda-beda secara efisien, dimana proses tersebut bekerja dengan cara menambahkan nilai integral pada nilai piksel asli, dengan nilai integral yang diperoleh dengan cara penjumlahan seluruh piksel pada bagian kiri dan atasnya hingga seluruh piksel terlewati.

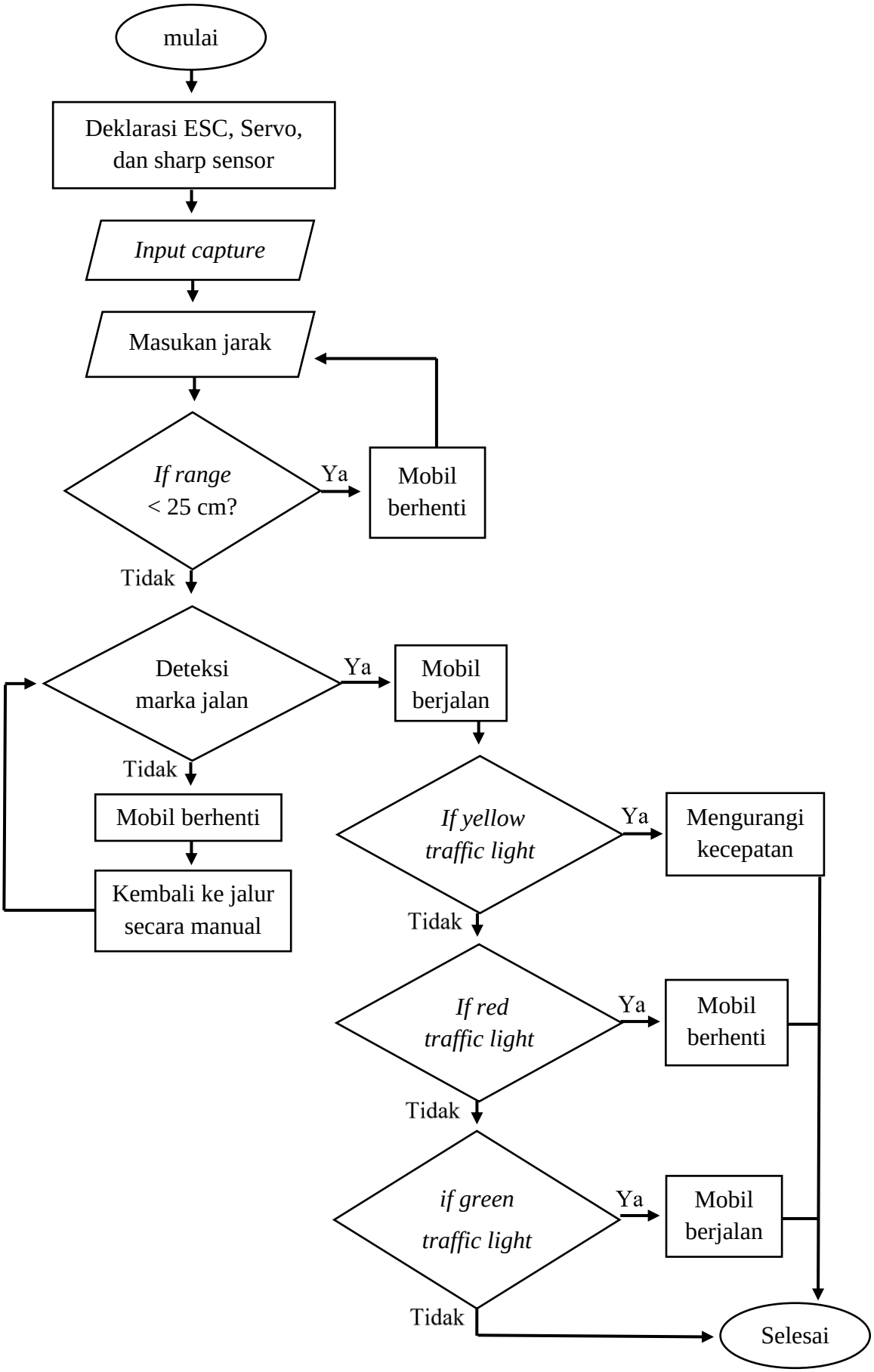
Pada tahapan keempat pemilihan fitur secara spesifik dan penentuan nilai ambang batas dilakukan dengan metode *machine learning* yang disebut sebagai *AdaBoost.* Peran *AdaBoost* disini adalah menggabungkan beberapa *classifier* lemah untuk dijadikan sebagai *classifier* kuat. Penggabungan *AdaBoost classifier* sebagai sebuah filter mampu untuk menggolongkan daerah yang ingin dideteksi secara efisien, dimana sebuah filter terbangun dari sebuah *classifier* lemah atau satu fitur *Haar*.

Tahap kelima adalah proses penentuan objek, bila terdapat filter yang gagal melewati sebuah subcitra, maka daerah tersebut bisa langsung dikatakan sebagai daerah yang tidak berisikan objek yang ingin dideteksi, namun jika seluruh filter berhasil dilewatkan, maka daerah tersebut dikatakan sebagai objek atau dalam hal ini *traffic llight.*

Tahapan selanjutnya untuk mempercepat proses pendeteksian, maka filter diurutkan berdasar bobot yang diberikan oleh *AdaBoost*, dengan filter terpenting atau bobot terendah diletakan pada bagian awal untuk menghilangkan daerah bukan objek secara cepat. Terakhir adalah menampilkan hasil pendeteksian apakah citra tersebut merupakan objek atau tidak.

3.3.6 Alur Kerja Sistem (*Software*)

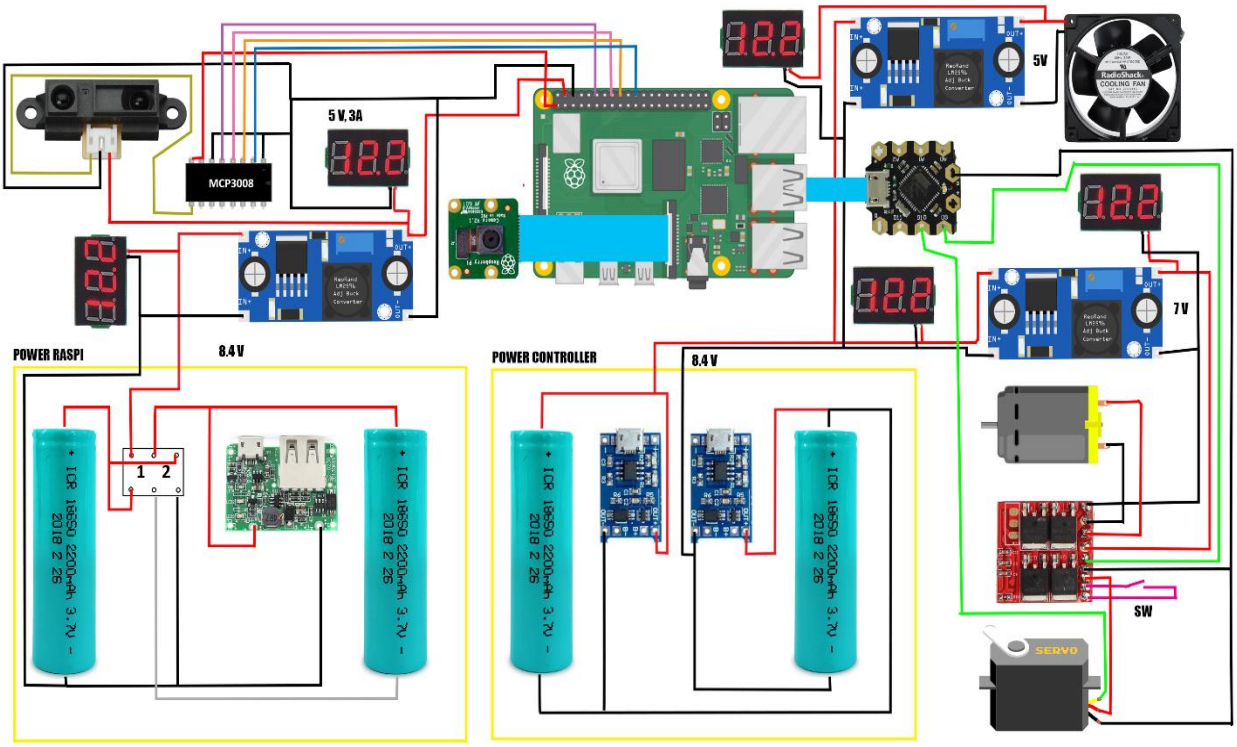
Untuk mempermudah penjelasan mengenai sistem yang peneliti gunakan untuk menyelesaikan penelitian ini dituangkan dalam bentuk *flowchart* yang dapat dilihat pada Gambar 3.8, dimana dalam tahap awal merupakan pendeklarasian yang dibutuhkan, lalu mobil akan mendeteksi halangan yang ada di depan, jika ada halangan semua sistem akan menghentikan mobil, namun jika tidak ada halangan maka pendeteksian oleh kamera dilakukan, saat mobil mendeteksi adanya jalur maka mobil akan berjalan sesuai jalur, lalu akan mengurangi kecepatan jika terdetekksi adanya lampu APILL kuning dan berhenti jika ada lampu APILL merah atau jalanan tidak sesuai kriteria, serta kembali berjalan jika terdeteksi lampu APILL hijau.



Gambar 3.8 *Flowchart* sistem kerja *prototype autonomous car*

3.3.7 Rangkaian *Hardware Prototype Autonomous Car*

Rangakain *hardware* yang peneliti gunakan dapat dilihat pada Gambar 3.9



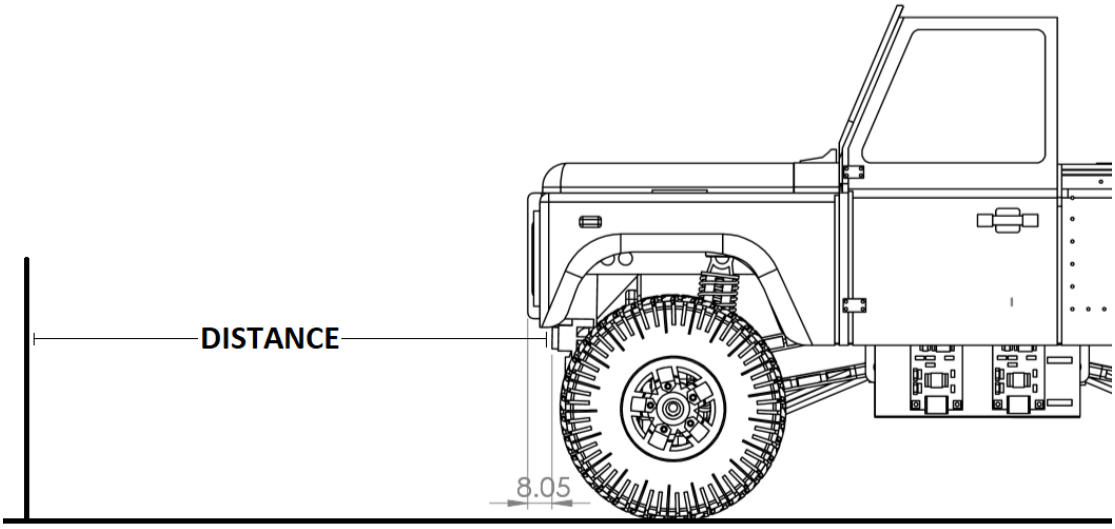
Gambar 3.9 *Wiring* diagram

3.4 Cara Analisis

Pada bagian ini peneliti sedikit menerangkan bagaimana Langkah yang dilakukan dalam menganalisis hasil dari penelitian yang telah dilakukan.

3.4.1 Analisis Pengukuran Jarak

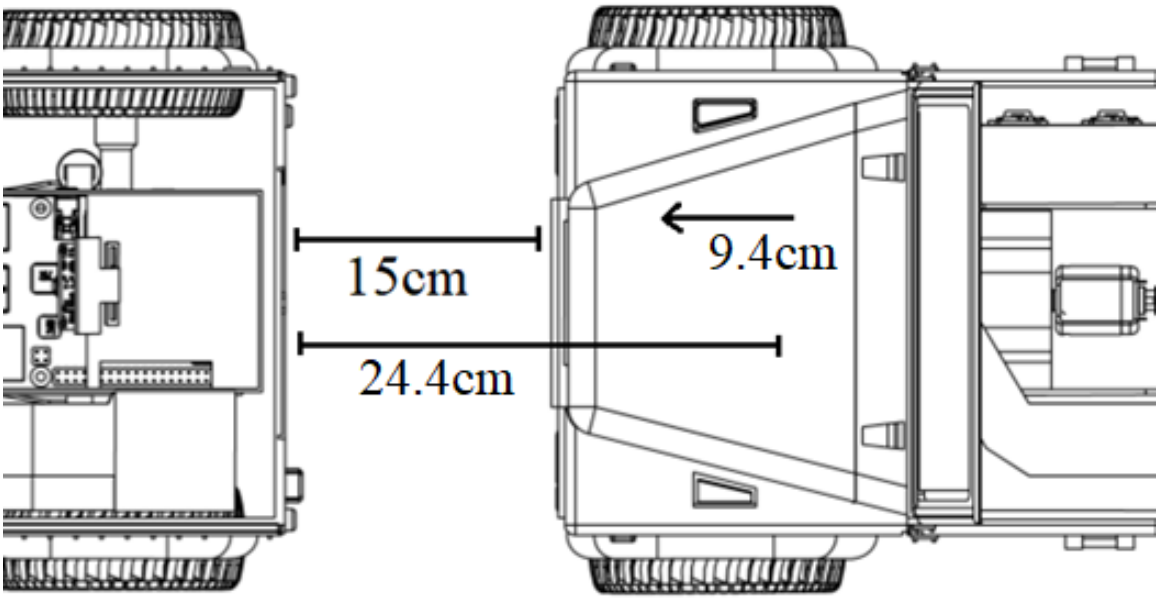
parameter untuk menentukan keberhasilan pengukuran jarak adalah dengan melakukan pengambilan data terhadap jarak aktual yang dibandingkan dengan pembacaan sensor, dimana persyaratan *error* untuk masing-masing data dibawah 5% dan *error* untuk rata-rata keseluruhan pembacaan data dibawah 3%, persyaratan ini harus terpenuhi karena pembacaan jarak sangat krusial untuk menunjang keselamatan pengemudi dan pengguna jalan lainnya. Pengambilan data yang dilakukan sesuai dengan kemampuan sensor, yakni 4 hingga 30cm yang diwakilkan dalam setiap penambahan 1cm per data. Untuk mendapatkan hasil yang terbaik, pertama dilakukan pengambilan data berupa data digital dengan range 0 hingga 1024 (10 bit) yang berikutnya akan dilakukan *curve fitting* dengan jarak aktual menggunakan *power law model* pada Microsoft excel. Adapun ilustrasi analisis pengukuran jarak dapat dilihat pada Gambar 3.10.



Gambar 3.10 Analisis pengukuran jarak

3.4.2 Analisis Pengereman Mobil

Analisis pada bagian ini sangat erat kaitannya dengan pengukuran jarak oleh Sharp sensor pada bagian 3.4.1, dimana Analisis ini penting dilakukan karena benda bergerak seperti mobil tidak akan bisa berhenti secara mendadak, dikarenakan ada gaya dorong yang tercipta saat mobil bergerak, salah satu solusi yang diambil untuk penerapan pada *prototype* ini adalah dengan menambahkan jarak semu, semisal mobil harus berhenti pada jarak 15cm, sedangkan mobil mampu berhenti dari posisi bergerak sejauh 9.4cm, maka jarak yang diatur adalah 24.4cm, hal ini relevan digunakan karena pada *prototype* ini mobil diatur dengan kecepatan yang konstan, jadi jarak pengereman mobil relatif sama. Ilustrasi pada bagian ini dapat dilihat pada Gambar 3.11.



Gambar 3.11 Analisis pengereman mobil

3.4.3 *Confusion Matrix*

Confusion matrix adalah metode yang digunakan dalam menganalisis kinerja dari *training* terhadap *traffic light*, metode ini merepresentasikan kondisi aktual dari hasil pembelajaran yang telah dilakukan, dengan menampilkan hasil berupa akurasi, *error*, sensitivitas, dan presisi. Adapun ilustrasi dalam evaluasi pembelajaran dapat dilihat pada Gambar 3.12.

Gambar 3.12 Contoh evaluasi pembelajaran

Sensitivitas dan presisi adalah sebuah nilai dari prediksi benar yang dilakukan kepada citra positif atau dalam hal ini adalah *traffic light* saja, sedangkan akurasi adalah prediksi benar terhadap citra positif maupun negatif yang dibandingkan terhadap seluruh data yang dimiliki. Tabel 3.2 menunjukan *confusion matrix* yang akan memberi prediksi kondisi aktual dari algoritma *Haar-Cascade Classifier*.

Tabel 3.2 *Confusion matrix*

Prediksi	**Aktual**	
	Positif	**Negatif**
Positif	TP	FP
Negatif	FN	TN

Keterangan:

TP (*True Positive*) : memprediksi benar dalam gambar *traffic light* atau algoritma berhasil mendeteksi *traffic light* yang memang benar *traffic light.*

FP (*False Positive*) : memprediksi salah dalam gambar bukan *traffic light* atau algoritma mendeteksi *traffic light* yang seharusnya adalah bukan *traffic light.*

FN (*False Negative*) : memprediksi salah dalam gambar *traffic light* atau algoritma mendeteksi bukan *traffic light* namun sebenarnya adalah *traffic light*

TN (*True Negative*) : memprediksi benar dalam gambar bukan *traffic light* atau algoritma mendeteksi bukan *traffic light* yang memang sebenarnya bukan *traffic light*

Dari table *confusion matrix* dapat dilakukan beberapa perhitungan untuk menjawab pertanyaan yang berkaitan tentang kinerja pada algoritma pendeteksian objek, Adapun pertanyaan yang timbul, diantaranya:

a. Berapa persen keberhasilan pendeteksian terhadap citra positif dan negatif?

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \text{ x } 100\% \quad (3.1)$$

b. Berapa persen tingkat kepresisian yang berhasil dicapai?

$$\text{Presisi} = \frac{\text{TP}}{\text{TP} + \text{FP}} \text{ x } 100\% \quad (3.2)$$

c. Berapa persen tingkat sensitivitas yang berhasil dicapai?

$$\text{Sensitivitas} = \frac{\text{TP}}{\text{TP} + \text{FN}} \text{ x } 100\% \quad (3.3)$$

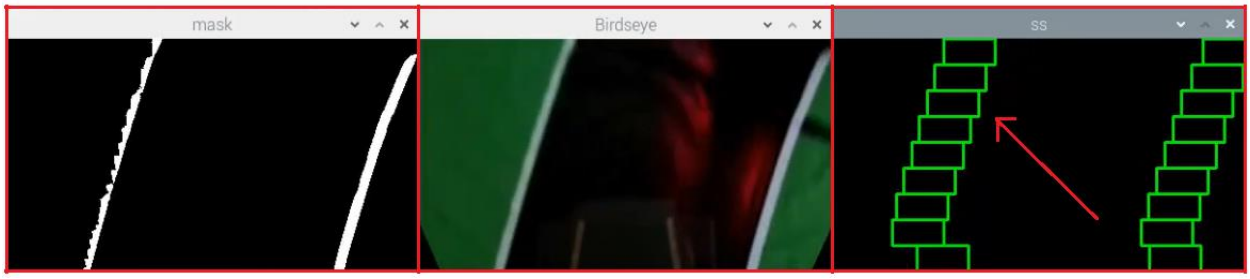
d. Berapa persen kesalahan yang terjadi dalam proses pendeteksian?

$$Error = 100\% - \text{Akurasi} \quad (3.4)$$

Adapun data yang digunakan untuk memenuhi *confusion matrix* adalah 50 citra positif dan 50 citra negatif yang serupa, dan dilakukan evaluasi terhadap ketiga warna dari *traffic light.*

3.4.4 Analisis Pembacaan Marka Jalan

Analisis ini dilakukan dengan cara pengambilan sampel terhadap pembacaan garis marka jalan oleh kamera, dimana cara kerja dari mobil *prototype* ini dalam mengikuti alur marka jalan adalah dengan melakukan pembacaan terhadap garis putih pada jalan, lalu dari hasil pembacaan tersebut dibagi menjadi beberapa persegi panjang yang bertumpu pada bagian tengah garis marka, setelah dilakukan pembagian tersebut maka didapatlah ukuran piksel setiap persegi panjangnya, lalu perbedaan posisi secara *horizontal* (sumbu x) antar persegi panjang yang digunakan untuk menentukan bentuk marka yang terbaca oleh camera. Adapun ilustrasinya dapat dilihat pada Gambar 3.13.



Gambar 3.13 Pembacaan marka jalan

BAB 4
HASIL DAN PEMBAHASAN

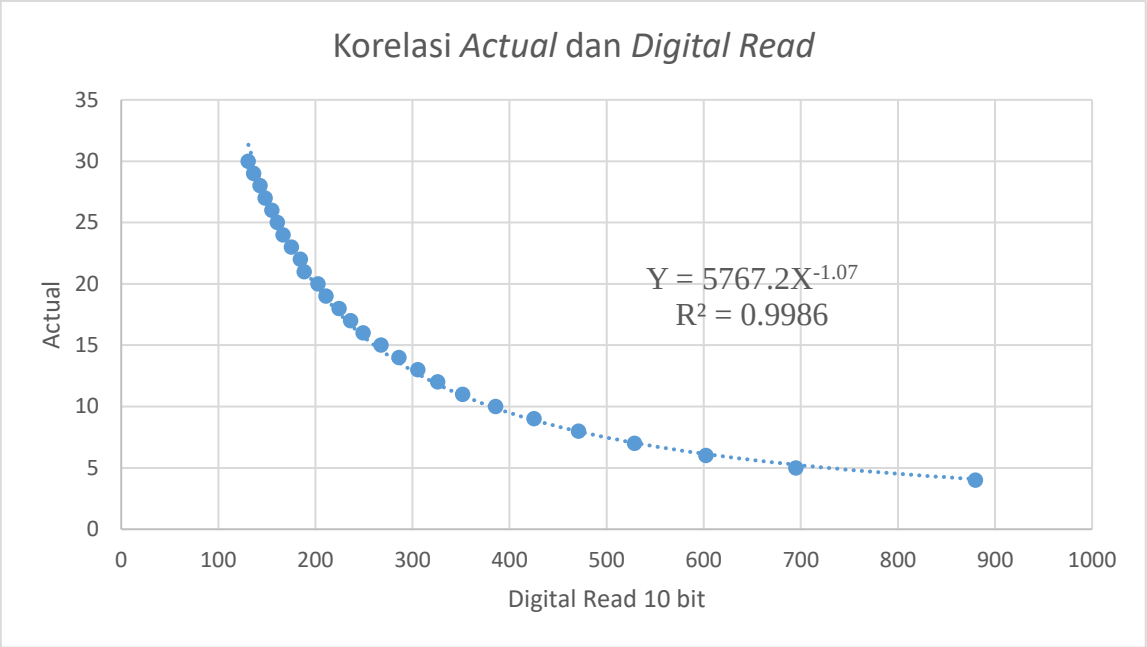
4.1 Pengujian Pembacaan Jarak

Metode yang digunakan dalam pengujian pembacaan jarak oleh Sharp Sensor GP2Y0A41SK0F sesuai dengan analisis kinerja diatas, dimana peneliti akan melakukan sampling per 1cm mulai dari 4cm hingga 30cm. hal ini dilakukan guna mendapatkan data 10 bit *resolution* dari ADC MCP3008 yang digunakan untuk mengkonversi data analog dari sensor menjadi digital agar dapat dibaca oleh Raspberry Pi. Setiap centimeter dilakukan pengambilan data sebanyak 120 kali yang berikutnya akan di rata-ratakan. Dari proses ini didapatkan hasil seperti pada Tabel 4.1.

Tabel 4.1 Hasil pembacaan jarak

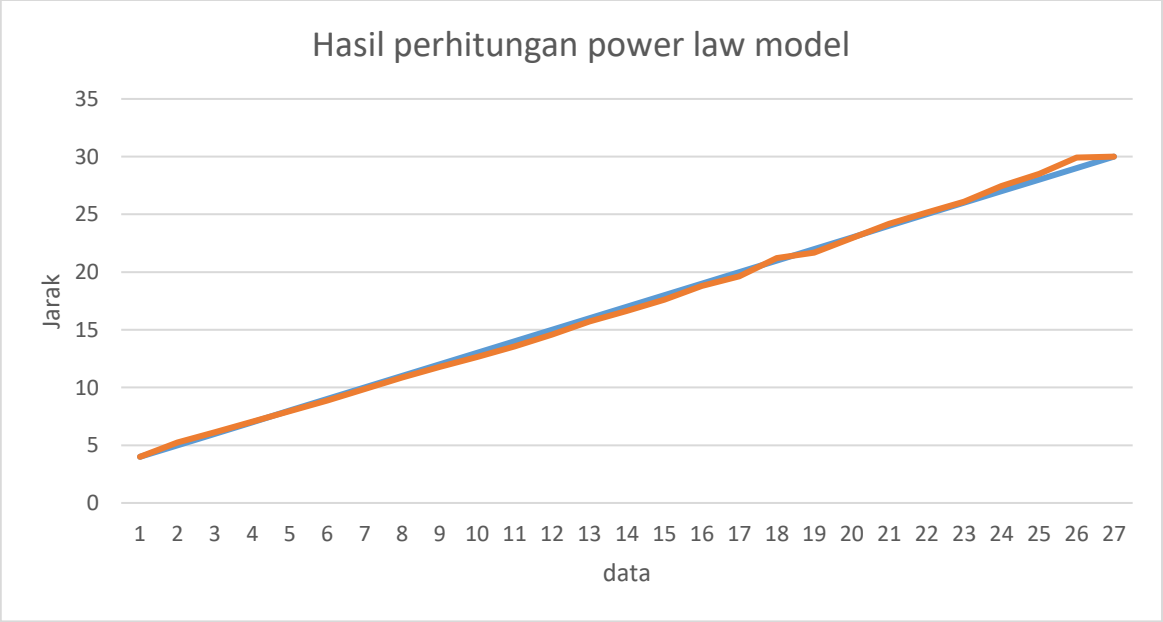
Tegangan (V)	Digital 2^10	Actual (cm)	Power in excel (cm)
2.838709677	879.9999999	4	4 (batas bawah)
2.241569824	694.8866454	5	5.249489478
1.942387217	602.1400373	6	6.119115942
1.706086466	528.8868045	7	7.030187201
1.519115525	470.9258128	8	7.95986756
1.371799035	425.2577009	9	8.877837115
1.243947277	385.6236559	10	9.857570418
1.134489424	351.6917214	11	10.87856264
1.052061288	326.1389993	12	11.79299425
0.985456872	305.4916303	13	12.64782198
0.922606257	286.0079397	14	13.57189393
0.863279054	267.6165067	15	14.57223654
0.803732374	249.1570359	16	15.73036268
0.762428781	236.3529221	17	16.64388737
0.72347319	224.2766889	18	17.60459292
0.680206198	210.8639214	19	18.80539995
0.653416655	202.5591631	20	19.63154528
0.607964754	188.4690737	21	21.20596839
0.595405647	184.5757506	22	21.68493598
0.565057946	175.1679633	23	22.93340309
0.537802297	166.7187121	24	24.17919076
0.518474192	160.7269995	25	25.14490254
0.500952449	155.2952592	26	26.08709626
0.477794267	148.1162228	27	27.44228057
0.461128527	142.9498434	28	28.50483102
0.440548226	136.5699501	29	29.93194933
0.422367381	130.9338881	30	30 (batas atas)

data tersebut akan melalui proses *curve fitting* terhadap jarak secara aktual yang diukur menggunakan penggaris, proses ini menggunakan metode *power law model* pada *Microsoft excel*. Grafik perhtungan *power law model* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Grafik *curve fitting*

Dari grafik tersebut menyatakan bahwa rumus untuk menjelaskan 10 bit *resolution* menjadi nilai aktual 4-30cm adalah $Y = 5767.2X\text{^}(-1.07)$ dengan R^2 yang berarti mampu menafsirkan hubungan antar variabel hingga 99,86%. Hasil dari perhitungan rumus tersebut dapat dilihat pada Tabel 4.1 menghasilkan grafik perbandingan antara pembacaan sensor dengan jarak aktual yang dapat dilihat pada Gambar 4.2.



Gambar 4.2 Perbandingan pembacaan sensor terhadap jarak aktual

Berdasarkan persamaan (4.1) berikut, maka didapatkan hasil seperti pada Tabel 4.2.

$$Error = \frac{\text{Aktual} - \text{Pembacaan sensor}}{\text{Aktual}} \quad (4.1)$$

Tabel 4.2 Perhitungan *error*

Aktual (Y)	**Power in excel (cm)**	***Error* (%)**
4	4	0
5	5.24949	4.98979
6	6.11912	1.985266
7	7.03019	0.431246
8	7.95987	0.501655
9	8.87784	1.357365
10	9.85757	1.424295
11	10.8786	1.103975
12	11.793	1.725047
13	12.6478	2.709061
14	13.5719	3.057900
15	14.5722	2.851756
16	15.7304	1.685233
17	16.6439	2.094780
18	17.6046	2.196705
19	18.8054	1.024210
20	19.6315	1.842273
21	21.206	0.980802
22	21.6849	1.432109
23	22.9334	0.289551
24	24.1792	0.746628
25	25.1449	0.579610
26	26.0871	0.334986
27	27.4423	1.638076
28	28.5048	1.802968
29	29.9319	3.213618
30	30	0
Rata-Rata		1.5555153

Pengujian ini dilakukan guna mengetahui kinerja dari sensor yang digunakan, dimana dapat dilihat pada Tabel 4.2 bahwa rata-rata *error* masih masuk dalam parameter yang telah ditentukan, yakni dibawah 3%, yang artinya bahwa sensor bekerja dengan baik sesuai dengan target.

4.2 Pengujian Pengereman

Pengujian ini dilakukan untuk mengevaluasi kinerja dari sharp sensor GP2Y0A41SK0F pada mobil *prototype* dalam menghindari penghalang yang berada di depan mobil, seperti pada umumnya dimana benda bergerak tidak bisa langsung berhenti begitu saja karena disebabkan masih adanya gaya dorong yang tercipta saat mobil bergerak. Semakin kencang suatu benda bergerak maka akan semakin jauh juga jarak benda tersebut berhenti saat mulai dikakukan pengereman, untuk menghindari hal tersebut, maka peneliti memprogram mobil agar selalu memiliki kecepatan yang konstan.

Pada tahapan ini peneliti menggunakan beberapa skenario, diantaranya dengan memprogram mobil agar berhenti pada jarak 15cm, 20cm, dan 25cm dari penghalang yang ada didepannya. Adapun teknis yang digunakan adalah dengan mengatur *controller motor* untuk berada pada kondisi 0 atau bisa dikatakan tidak ada tegangan yang keluar menuju ke motor untuk menggerakan mobil, hal ini aman dilakukan karena tidak menimbulkan pengereman mendadak yang dapat membahayakan pengguna jalan lain, maupun pengendara itu sendiri. Pengujian ini dilakukan dengan cara pengambilan 5 data pengereman mobil saat seluruh sistem dijalankan, disini peneliti mendapatkan data dari 3 buah skenario seperti yang dapat dilihat pada Tabel 4.3.

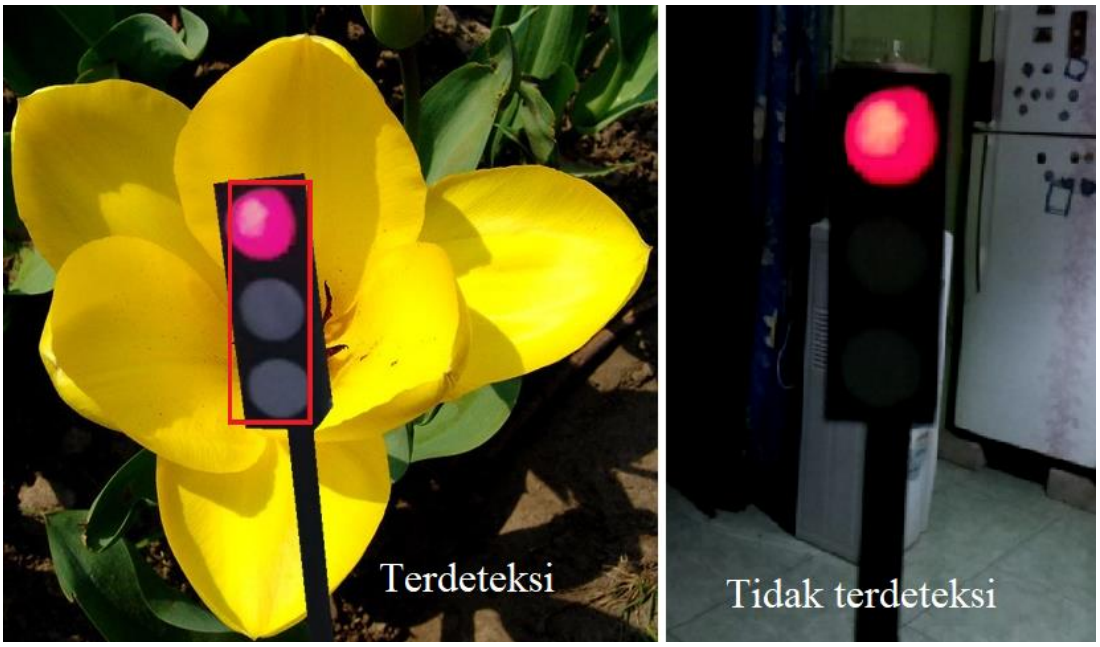
Tabel 4.3 Data pengujian pengereman mobil *prototype*

no	Jarak yang ditentukan	Mobil berhenti	selisih
1	15 cm	6	9
2	15 cm	6.5	8.5
3	15 cm	6	9
4	15 cm	6.5	8.5
5	15 cm	6	9
6	20 cm	11	9
7	20 cm	9.5	10.5
8	20 cm	11	9
9	20 cm	10	10
10	20 cm	11	9
11	25 cm	15	10
12	25 cm	14.5	10.5
13	25 cm	14.5	10.5
14	25 cm	16	9
15	25 cm	16	9
Rata-rata			9.36667

Dari Tabel 4.3 tersebut dapat dilihat bahwa rata-rata mobil mampu menghentikan laju kendaraannya dalam jarak 9,4cm (jarak semu), oleh karena itu ditambahkan jarak semu untuk mengatasi hal ini, dimana sistem diprogram agar mobil berhenti sejauh jarak yang diinginkan ditambah jarak semu tersebut.

4.3 Pengujian Pendeteksian *Traffic light*

Bersesuaian dengan pembagian dataset pada Tabel 3.1 terlihat bahwa data uji terdapat 100 citra, hal ini dilakukan guna mempermudah pencarian akurasi, dimana untuk tiap citra mewakili 1% akurasi, lalu penentuan citra itu sendiri memiliki kriteria yang sama saat dilakukannya proses pembelajaran, yakni tegak lurus serta memiliki pencahayaan yang cukup, karena jika kedua hal itu tidak terpenuhi maka hasilnya dapat dilihat pada Gambar 4.3. untuk memperkuat hasil pembelajaran maka didalam 100 citra tersebut diberikan citra yang tidak seharusnya terdeteksi namun memiliki kemiripan, yakni dengan citra kedua warna lampu *traffic light* lainnya berjumlah 50 buah, dengan persamaan (3.1), (3.2), (3.3), dan (3.4), maka hasil dari pengujian dapat dilihat pada Tabel 4.5.



Gambar 4.3 Pendeteksian diluar kriteria

Tabel 4.4 Data pengujian pendeteksian *traffic light*

Klasifikasi	TP	TN	FP	FN
Lampu merah	46	50	0	4
Lampu hijau	47	50	0	3
Lampu kuning	42	50	0	8

Dengan scale factor 1,05, dan minimum nighbors 1 didapatkan hasil:

a. Unjuk kerja *traffic light* merah:

1. Akurasi $= \frac{46+50}{46+0+4+50}$ x 100%
2. Presisi $= \frac{46}{46+0}$ x 100%
3. Sensitivitas $= \frac{46}{46+4}$ x 100%
4. $Error = 100\% - 96\%$

b. Unjuk kerja *traffic light* hijau:

1. Akurasi $= \frac{47+50}{47+0+3+50}$ x 100%
2. Presisi $= \frac{47}{47+0}$ x 100%
3. Sensitivitas $= \frac{47}{47+3}$ x 100%
4. $Error = 100\% - 97\%$

c. Unjuk kerja *traffic light* kuning:

1. Akurasi $= \frac{42+50}{42+0+8+50}$ x 100%
2. Presisi $= \frac{42}{42+0}$ x 100%
3. Sensitivitas $= \frac{42}{42+8}$ x 100%
4. $Error = 100\% - 92\%$

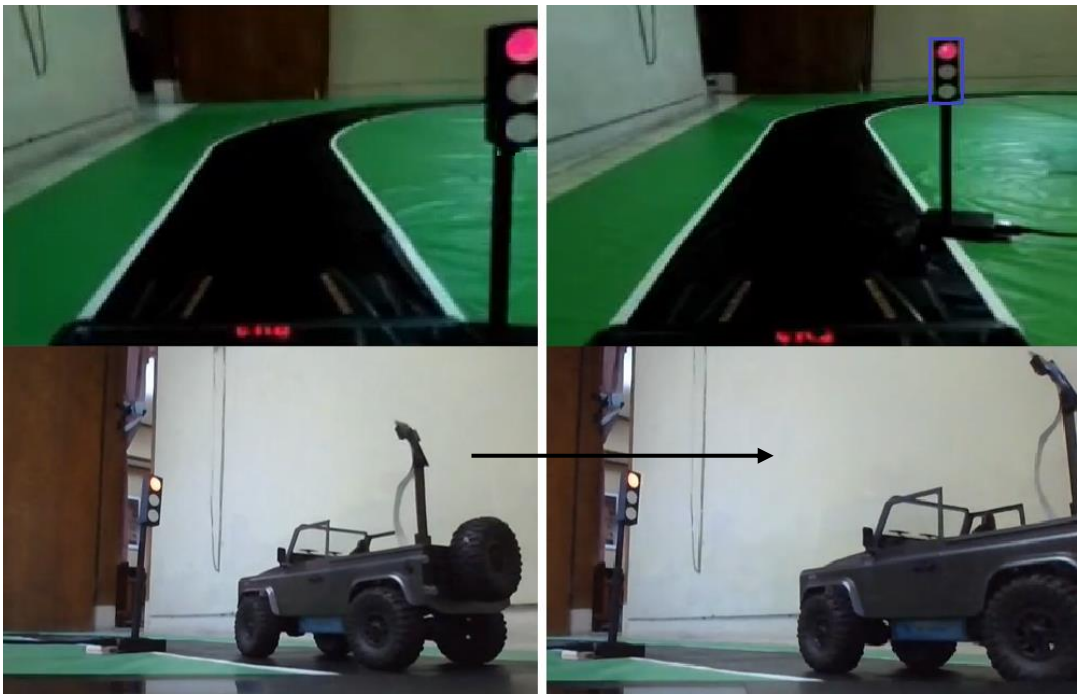
Tabel 4.5 Hasil pembelajaran *Haar-Cascade Classifier*

klasifikasi	**Akurasi**	**Presisi**	**Sensitivitas**	***Error***
Lampu merah	96%	100%	92%	4%
Lampu hijau	97%	100%	94%	3%
Lampu kuning	92%	100%	84%	8%

Dari Tabel 4.5 terllihat hasil yang baik untuk pendeteksian objek itu sendiri, namun pada saat pengaplikasian pada *prototype* masalah baru muncul, yakni sistem tidak dapat selalu mendeteksi objek tersebut secara terus menerus, namun mendeteksi seperti pengambilan sampel, jadi dalam kurun waktu beberapa detik bisa beberapa kali berhasil dan tidak, hal ini terjadi dikarenakan mobil yang bergerak akan mengakibatkan getarnya kamera yang menyebabkan gambar yang ditangkap menjadi *blur*, dalam kasus ini peneliti mengambil jalan keluar dengan menambahkan program jika sistem mendeteksi satu kali saja terlihat lampu APILL merah dengan ukuran yang sesuai kriteria maka mobil akan berhenti dan kembali berjalan saat melihat satu kali saja lampu hijau, jadi hal ini bisa mengatasi ketidak konsistenan pendeteksian, lalu untuk lampu APILL kuning tidak

diberlakukan hal serupa, namun sepenuhnya mengandalkan pendeteksian secara terus, karena sistem hanya akan sedikit menurunkan kecepatan, dan tidak terlalu krusial dalam penelitian ini, serta jika mobil melewati lampu APILL saat kondisi kuning, maka tidak ada lampu yang bisa memicunya kembali ke posisi normal, hingga bertemu lampu APILL berikutnya dengan warna selain kuning.

Karena resolusi dikecilkan hingga 240*180 piksel maka pendeteksian dari jarak yang cukup jauh dari objek tidak bisa dilakukan, maka ditambahkan program untuk memerintahkan mobil mundur sedikit jika terlalu dekat dengan lampu merah yang berfungsi untuk terhindar dari frame yang terpotong agar pendeteksian selanjutnya bisa dilakukan. Ilustrasi ini dapat dilihat pada Gambar 4.4.



Gambar 4.4 Posisi *prototype* harus mundur

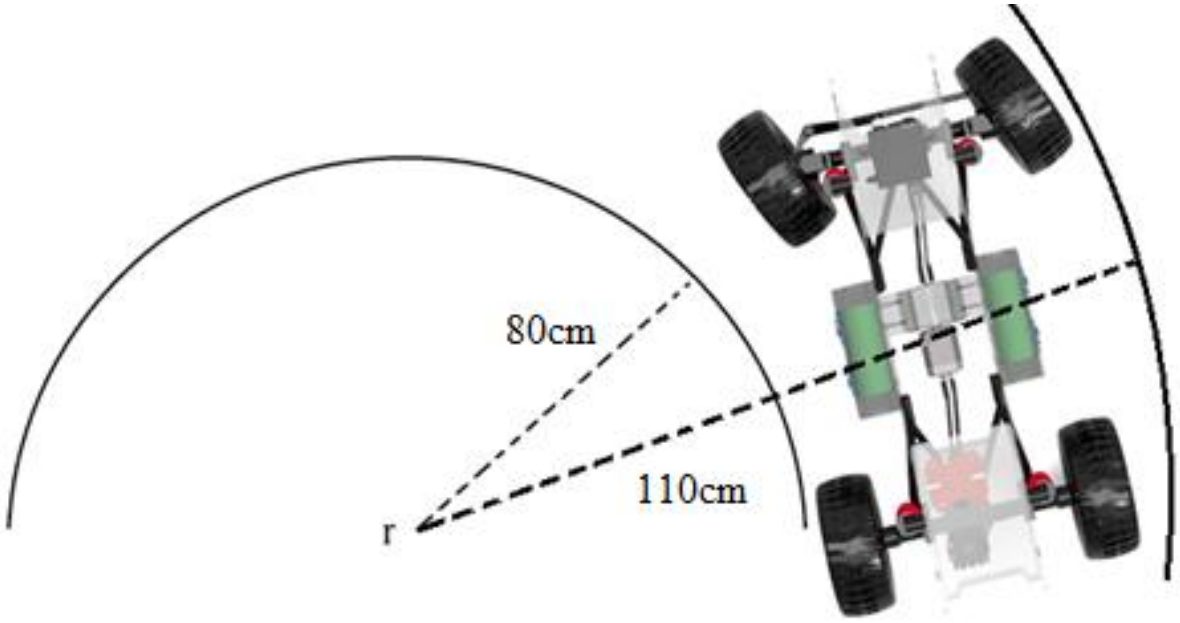
Walau demikian keadaan tersebut tidak selalu terjadi, ada kalanya kamera mampu mendeteksi sesuai dengan tinggi piksel yang telah ditentukan.

4.4 Pengujian Pembacaan Marka Jalan

Pengujian terhadap pembacaan marka jalan harus sebisa mungkin tidak ada kesalahan, karena erat kaitannya dengan pergerakan mobil, pada pengujian kali ini peneliti menemukan keterbatasan kemampuan dalam mengikuti marka, yang disebabkan oleh pantulan cahaya yang berakibat kesalahan pembacaan, karena pantulan tersebut jalur terlihat menjadi bewarna putih seperti pada Gambar 4.5. Oleh karena itu peneliti menggunakan kertas agar mampu menyerap cahaya dengan baik saat mobil melakukan pembacaan marka.

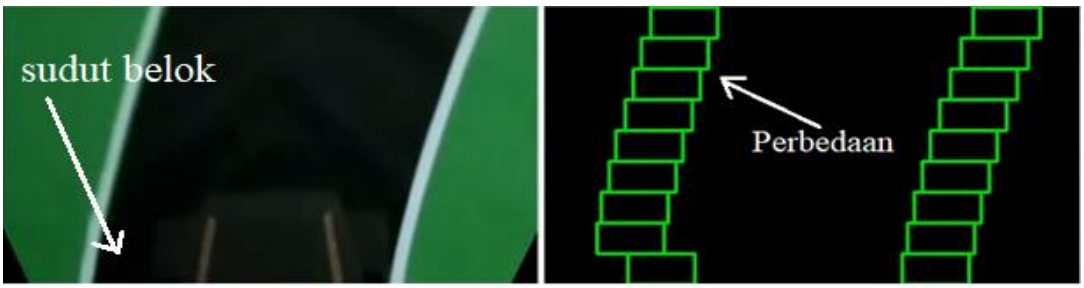
Gambar 4.5 Pantulan cahaya pada jalur

Permasalahan berikutnya yang harus dipertimbangkan agar mobil mampu berjalan mengikuti marka jalan adalah radius putar dari chassis yang digunakan, dalam penelitian ini *chassis* yang digunakan adalah MN D90 yang memiliki radius putar maksimal 110cm seperti ilustrasi pada Gambar 4.6, dimana hal ini wajib diikuti karena saat kamera mampu menerjemahkan jalan tetapi mobil tidak mempu mengikutinya, maka mobil akan tetap keluar dari jalur dan berhenti.



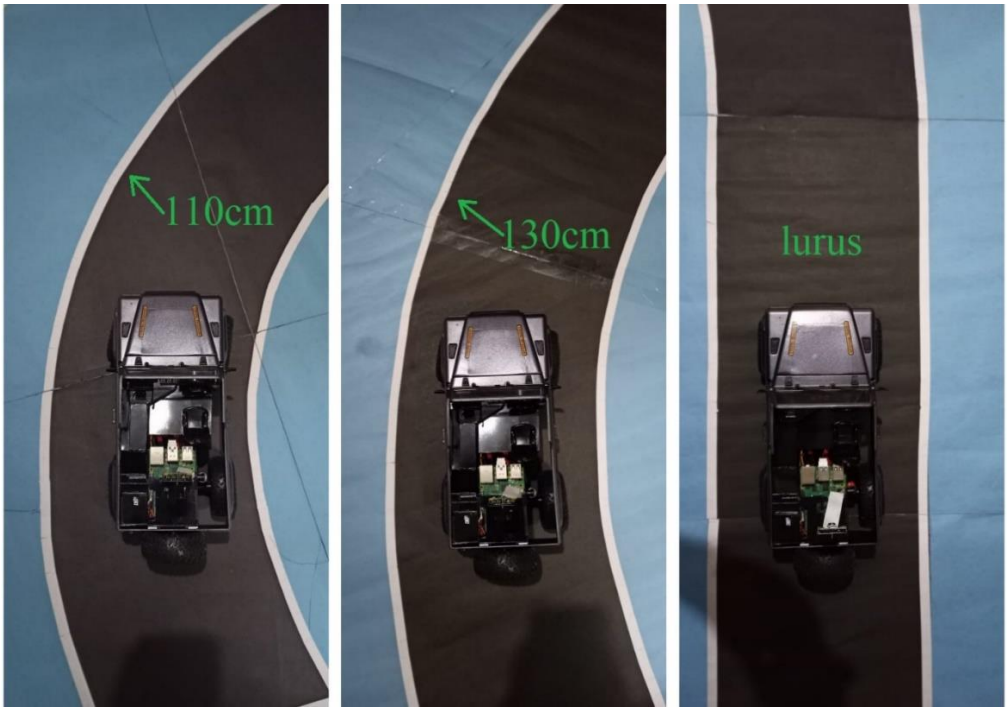
Gambar 4.6 Radius putar *prototype*

Berikutnya merupakan metode yang digunakan untuk pendeteksian jalan ini adalah dengan cara melakukan perhitungan piksel pada marka jalan, dimana pada tahap awal peneliti membagi gambar menjadi 2 bagian, yakni bagian kiri dan kanan sebagai deklarasi awal untuk membedakan marka kiri dan kanan, lalu dilanjutkan pembuatan bingkai sedemikian rupa yang bertumpu pada garis marka, dan pada tahap akhir menghitung perbedaan posisi antar bingkai yang digunakan untuk melihat bentuk marka seperti yang diilustrasikan pada Gambar 4.7.



Gambar 4.7 Pendeteksian marka jalan

Untuk mengevaluasi hasil unjuk kerja dari pembacaan marka ini, maka pengujian dilakukan dengan cara menjalankan *prototype autonomous car* pada jalur yang sesuai dengan kriteria 1.3 sebanyak 10 kali pengulangan, dengan jalur yang dibuat sedemikian rupa untuk meliputi belok kiri, kanan, dan lurus seperti pada Gambar 4.8.



Gambar 4.8 Ilustrasi pengambilan data

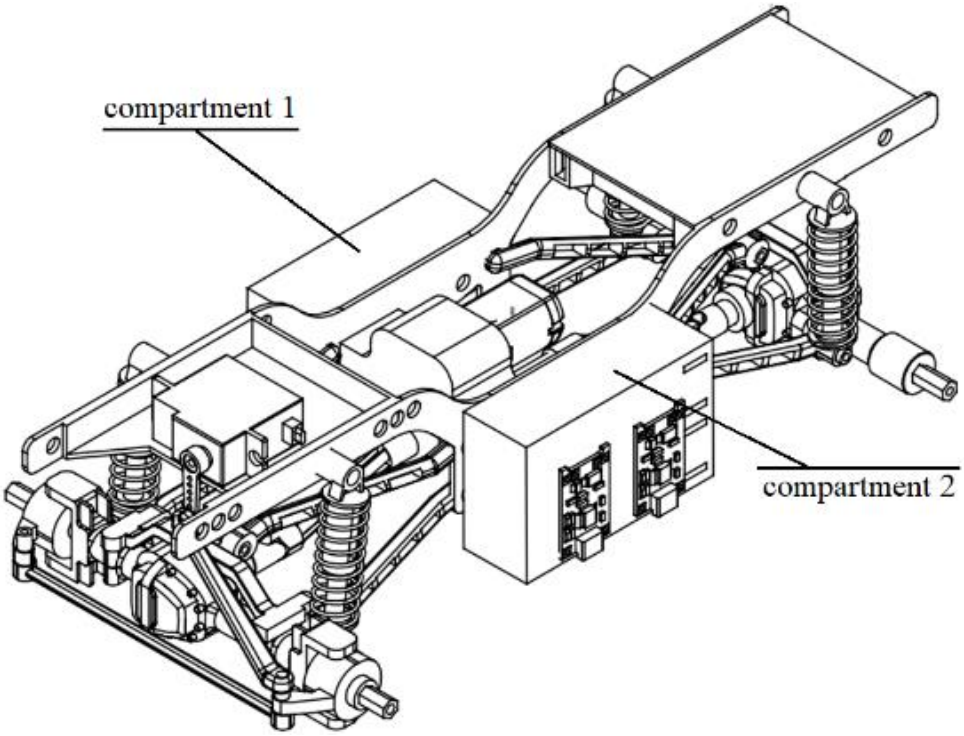
Tabel 4.6 Unjuk kerja pembacaan marka jalan

No	Kategori	Pengulangan	Keberhasilan	Persentase
1	Belok kiri (radius 110 cm)	10	10	100%
2	Belok kanan (radius 110 cm)	10	10	100%
3	Belok kiri (radius 130 cm)	10	10	100%
4	Belok kanan (radius 130 cm)	10	10	100%
5	Lurus	10	10	100%

Dari Tabel 4.6 dapat dilihat bahwa hasil dari pengujian ini menunjukan bahwa sistem mampu menbaca marka jalan dengan tepat, dimana berhasil mencapai nilai akurasi hingga 100%.

4.5 Pengujian Ketahanan Mobil

Pengujian kali ini erat kaitannya dengan ketahanan *battery* dan sistem pendinginan yang digunakan. Disini peneliti menggunakan 4 buah *battery* li-ion dengan kapasitas 2000mah 3.7V, yang terbagi menjadi 2 bagian utama, dengan masing-masing *compartment* memiliki 2 buah battery secara seri, dengan konfigurasi *compartment* 1 untuk Raspberry Pi, camera, dan sharp sensor, lalu untuk *compartment* 2 digunakan untuk aktuator pada sistem, Peletakan *battery* dapat dilihat pada Gambar 4.9.

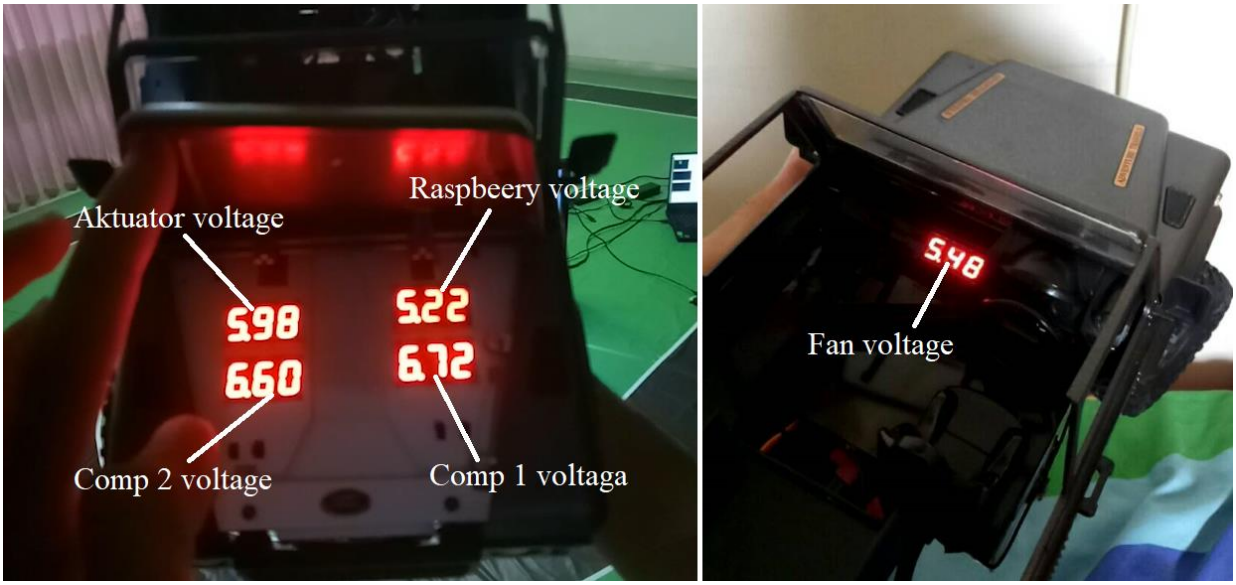


Gambar 4.9 Posisi *battery*

Karena *battery* dikonfigurasi seri maka menghasilkan tegangan 7.4V dengan *full charge* menjadi 8.4V, sedangkan raspbeery hanya membutuhkan maksimal tegangan 5.5V dengan arus 3A, oleh karena itu disematkan *stepdown dc-dc* 5A untuk merubah tegangan menjadi 5.25V, disini pemilihan 5.25V untuk mengantisipasi terjadinya *throttling* karena kekurangan tegangan dan untuk mencegah terjadinya kerusakan hardware karena kelebihan tegnagan. Pada bagian *compartment* 2 digunakan *stepdown dc-dc 3A* sebanyak 2 buah, dimana salah satu digunakan untuk kipas raspbeery, dan *stepdown* kedua untuk aktuator (motor, ESC, servo, Dfrobot Arduino Leonardo). Pada bagian aktuator seluruh sistem memiliki kriteria yang linear terhadap kapasitas battery, maka saat tegangan turun motor dan servo akan ikut melemah, begitu juga pada bagian kipas, oleh karena itu *stepdown* aktuator diatur menjadi 6V agar kecepatan maupun kekuatan servo menjadi konstan, lalu pada bagian kipas diatur menjadi 5.5V agar putaran kipas tidak terlalu kencang dan terlalu banyak mengonsumsi daya.

Dari konfigurasi yang telah dipaparkan diatas, maka *prototype* ini mampu berjalan tanpa sesuai kemampuan dalam kurun waktu 25 menit tanpa berhenti. *Prototype* berhenti karena kekurangan daya pada bagian aktuator, maka dapat diartikan bahwa penambahan kapasitas *battery* dapat memperpanjang waktu jalan. Dengan ini seluruh sistem dapat bekerja dengan sempurna sesuai parameter, baik dalam sektor pendingin, penggerak, pengkondisi, maupun pendukung lainnya. Hasil dari pengujian dapat dilihat pada Gambar 4.10 dengan posisi peletakan indikator pada Gambar 4.11 dan *wiring* diagram pada Gambar 3.9.

Gambar 4.10 Hasil pengujian ketahanan



Gambar 4.11 Posisi indikator

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian dengan judul "*prototype autonomous car* menggunakan *Haar-Cascade Classifier* berbasis Raspberry Pi" diantaranya sebagai berikut:

1. Sistem mampu berjalan sesuai dengan tujuan yang telah ditentukan, walaupun memiliki beberapa keterbatasan pada hardware yang dipergunakan.
2. Penelitian ini merupakan salah satu sarana untuk mengetahui sedikit cara kerja *system autonomous car* yang mulai marak dikembangkan oleh produsen mobil-mobil ternama.
3. Peletakan posisi kamera dibagian belakang mobil ditujukan untuk akomodasi dalam kemampuan pendeteksian marka jalan, karena semakin dekat titik pendeteksian terhadap ban depan mobil maka akan semakin baik dalam proses pengambilan keputusan, namun kelemahannya akan mengurangi sudut pandang ke bagian depan mobil yang dipergunakan untuk pendeteksian *traffic light*.
4. Memperkecil ukuran piksel sama halnya seperti memperkecil perhitungan untuk pendeteksian objek, yang berarti akan meningkatkan FPS (*frame per second*) saat dilakukannya *scanning*, namun akan berakibat pada berkurangnya kemampuan dalam pendeteksian serta dipemgaruhi oleh point no.3.
5. Akurasi rata-rata yang diperoleh oleh sistem dalam pendeteksian *traffic light* sebesar 95%, presisi 100%, sensitivitas 90%, dan *error* 5%, yang berarti pembelajaran sudah sangat baik dalam mengenali objek.
6. Proses pembacaan marka jalan dengan algoritma perhitungan piksel sesuai pergeseran dari setiap kotak yang dibuat mengikuti jalur dapat bekerja dengan sangat baik jika memenuhi kriteria yang telah ditentukan, dengan akurasi hingga 100%.
7. Dengan menggunakan power law model untuk fitting curve data dari Sharp Sensor GP2Y0A41SK0F didapatkan akurasi sebesar 98,45%, dari data ini dapat disimpulkan bahwa sensor dapat bekerja dengan baik dikarenakan *error* dibawah 3% sesuai dengan target yang ingin dicapai.
8. Penggunaan 4 buah *battery* li-ion sudah mampu mensupply kebutuhan listrik pada sistem hingga 25 menit dijalankan tanpa berhenti, serta sistem pendingin yang mampu menjaga temperatur Raspberry Pi tetap pada keadaan normal (±50ºC).

5.2 Saran

Saran yang diberikan oleh peneliti untuk pengembangan kedepannya guna memaksimalkan kinerja serta penggunaannya adalah sebagai berikut:

1. Diharapkan sistem dapat dikembangkan kembali guna pengaplikasian pada mobil sesungguhnya ataupun pada mobil listrik karya mahasiswa.
2. Karena keterbatasan SBC saat ini, maka diharapkan bagi peneliti selanjutnya untuk menggunakan hardware yang lebih handal untuk meningkatkan kinerja sistem, seandainya tetap ingin menggunakan Raspberry Pi tambahkanlah Coral USB Accelerator guna meningkatkan komputasi pada sistem.
3. Untuk kedepannya perancangan *prototype* yang lebih handal maka pertimbangkan *chassis* yang digunakan, disarankan menggunakan konfigurasi 2WD penggerak belakang agar memiliki radius putar yang kecil.
4. Peningkatan akurasi pengukuran jarak serta pembacaan halang rintang di sekitar mobil untuk kedepannya bisa menggunakan lidar sensor, namun harus diiringi pengembangan hardware lainnya agar bisa bekerja secara maksimal.
5. Untuk mengatasi pendeteksian yang kurang baik, bisa digunakan kamera USB yang banyak dijual pasaran dengan resolusi yang tinggi dengan sundut pandang yang lebih luas, serta penambahan *stabilizer* disarankan agar hasil gambar yang diperoleh menjadi stabil.
6. Saat ini sudah banyak algoritma untuk pendeteksian objek dengan kecerdasan serta komputasi yang lebih baik, hal ini disarankan jika hardware yang dimiliki sudah mendukung.
7. Perlu ditambahkannya sistem pengereman agar mobil lebih sigap dalam menghindari halangan yang ada di depan.

DAFTAR PUSTAKA

[1] "Undang-Undang Republik Indonesia Nomor 22 Tahun 2009 Tentang Lalu Lintas dan Angkutan Jalan," 2009. [Online]. Available: http://hubdat.dephub.go.id/uu/288-uu-nomor-22-tahun-2009-tentang-lalu-lintas-dan-angkutan-jalan.

[2] "Data Kecelakaan dan Pelanggaran Lalu Lintas." [Online]. Available: http://bappeda.jogjaprov.go.id/dataku/data_dasar/index/548-data-kecelakaan-dan-pelanggaran-lalu-lintas.

[3] K. N. V Satyanarayana, B. Tapasvi, P. Kanakaraju, and G. Rameshbabu, "Based on Machine Learning Autonomous Car Using Raspberry-Pi .," vol. 7, no. 12, pp. 76–82, 2017.

[4] K. Massidik, E. Susanto, P. Pangaribuan, F. T. Elektro, U. Telkom, and N. Network, "Prototype Autonomous Car Menggunakan Image Processing dan Kontrol Neural Network," *e-Proceeding Eng.*, vol. 4, no. 2, pp. 1490–1496, 2017.

[5] Y. Azzakirot, "Perhitungan Pohon Kelapa Sawit dengan Mengidentifikasi Pohon Menggunakan Algoritma Haar-Cascade Classifier," 2018.

[6] I. D. Kurniawati and I. A. Kusumawardhani, "Implementasi Algoritma Canny Dalam Pengenalan Wajah Menggunakan Antarmuka GUI Matlab," *J. Inst. Teknol. Sepuluh Nop.*, no. December, pp. 3–8, 2017.

[7] R. E. Putri, T. Matulatan, and N. Hayaty, "Sistem Deteksi Wajah Pada Camera Real Time dengan menggunakan Metode Viola - Jones," *J. Sustain. J. Has. Penelit. dan Ind. Terap.*, vol. 08, no. 01, 2019.

[8] R. A. Firmansyah and E. Alfianto, "Pembuatan Haar-Cascade Dan Local Binary Pattern Sebagai Sistem Pendeteksi Halangan Pada Automatic Guided Vehicle," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 2, pp. 1073–1082, 2018.

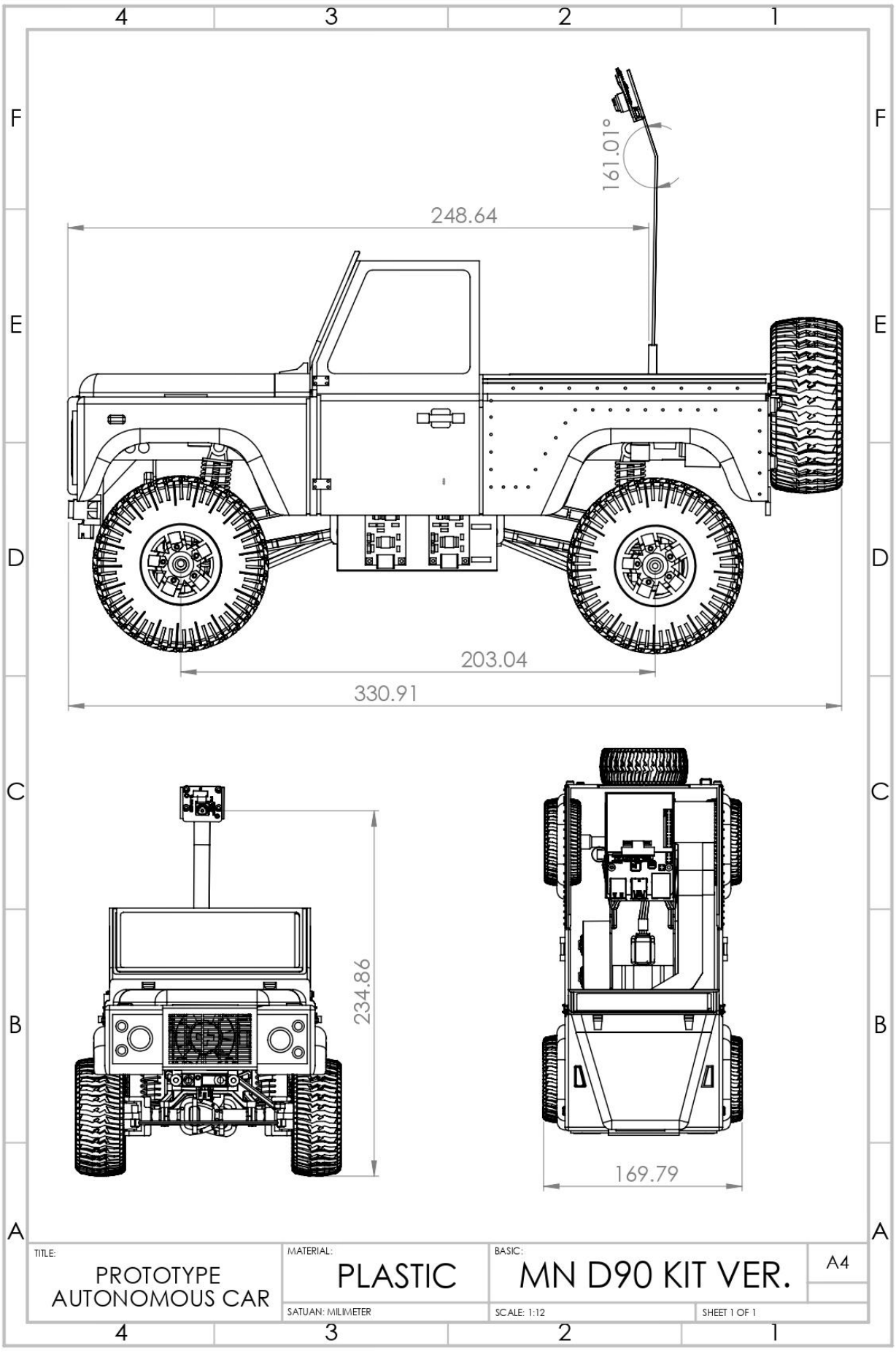
LAMPIRAN

Lampiran 1 – Rincian Biaya Skripsi

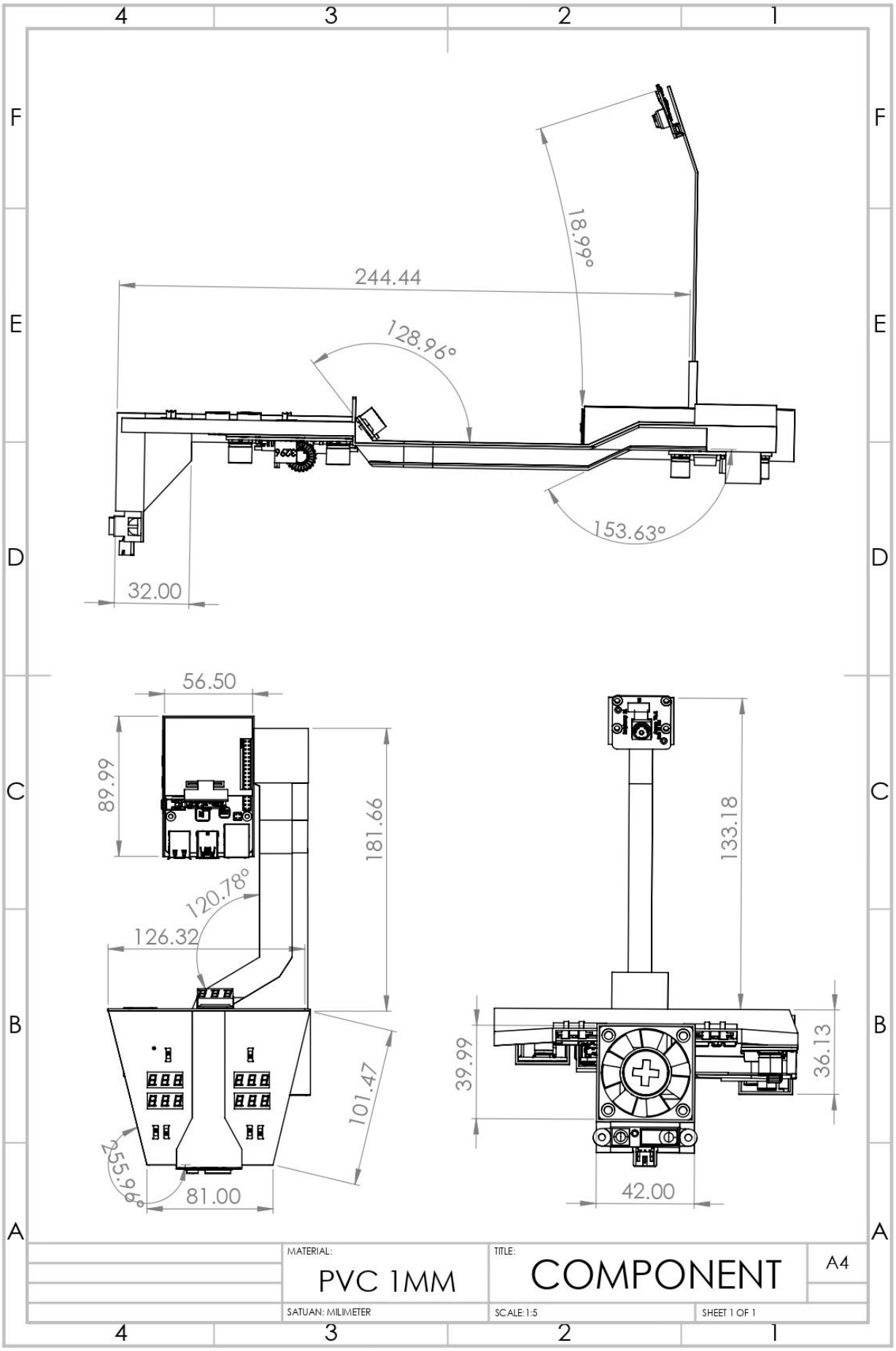
No	Rincian	Frekuensi (Kali)	Volume (Unit)	Satuan (Rp)	Jumlah (Rp)
(1)	(2)	(3)	(4)	(5)	(6) = (3) × (4) × (5)
1	Raspberry Pi 4 4GB	1	1	Rp1,250,000	Rp1,250,000
2	Camera raspi V2.1	1	1	Rp450,000	Rp450,000
3	li-ion battery	1	4	Rp70,000	Rp280,000
4	Modul charger li-ion	1	4	Rp10,000	Rp40,000
5	Voltmeter	1	5	Rp20,000	Rp100,000
6	Sharp sensor	1	1	Rp120,000	Rp120,000
7	IC MCP3008	1	1	Rp50,000	Rp50,000
8	WTF fan 8.4v	1	1	Rp275,000	Rp275,000
9	Arduino Leonardo	1	1	Rp220,000	Rp220,000
10	Arduino UNO	1	1	Rp70,000	Rp70,000
11	LED	1	3	Rp500	Rp1,500
12	ESC 20a	1	1	Rp80,000	Rp80,000
13	MN D90 kit version	1	1	Rp650,000	Rp650,000
14	PVC 1mm 60*100cm	1	1	Rp65,000	Rp65,000
15	Diton paint black	1	2	Rp22,000	Rp44,000
16	Aerox paint gray	1	1	Rp50,000	Rp50,000
17	Mini switch	1	7	Rp1,500	Rp10,500
18	Energizer PD *	1	1	Rp650,000	Rp650,000
19	Keyboard *	1	1	Rp50,000	Rp50,000
20	Logitech G102 *	1	1	Rp270,000	Rp270,000
21	Sandisk ultra 16GB	1	1	Rp80,000	Rp80,000
22	Cable / m	1	4	Rp2,000	Rp8,000
23	Step down 3A	1	2	Rp35,000	Rp70,000
24	Step down 5A	1	1	Rp50,000	Rp50,000
25	Karpet / 120*100cm	1	23	Rp5,000	Rp115,000
26	Polyacril glue	1	1	Rp50,000	Rp50,000
27	Monitor *	1	1	Rp400,000	Rp400,000
29	Kit Tamiya cc01 *	1	1	Rp2,000,000	Rp2,000,000
30	Tx-rx FS GT3b *	1	1	Rp600,000	Rp600,000
Jumlah					Rp8,099,000.00

Note: *(optional) karena hanya sekali pakai dan akan dijual kembali.

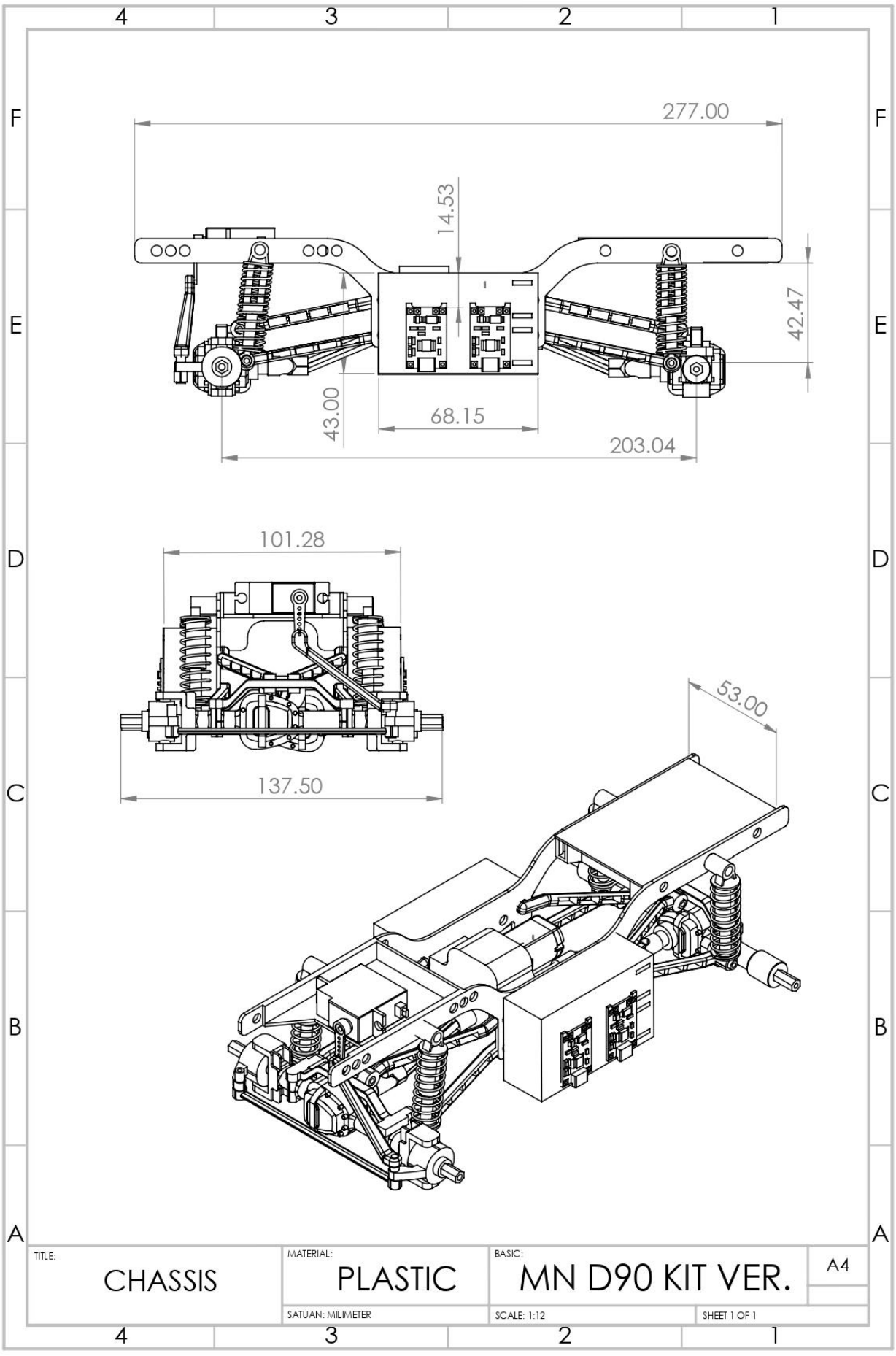
Lampiran 2 – Desain *prototype autonomous car*



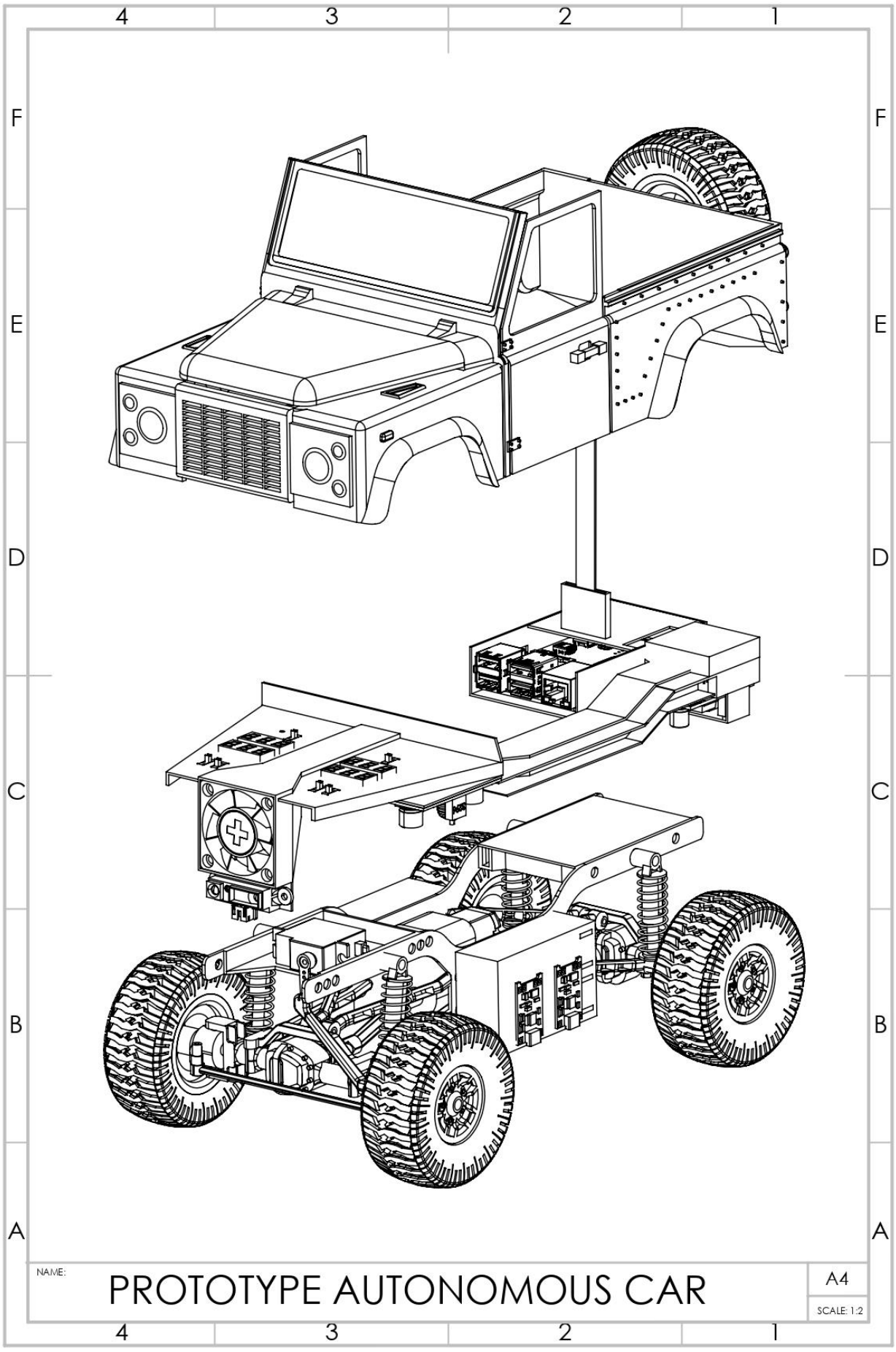
Lampiran 3 – Desain *prototype autonomous car*



Lampiran 4 – Desain *prototype autonomous car*



Lampiran 5 – Desain *prototype autonomous car*



Lampiran 6 – Datasheet Sharp Sensor GP2Y0A41SK0F

■Supplements

●Example of output distance characteristics