

# Implementasi Alat Pemantauan Kepadatan Lalu Lintas

1<sup>st</sup> Moh Agung Aji Santoso  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
agungaji@student.telkomunive  
rsity.ac.id

2<sup>nd</sup> Jangkung Raharjo  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
jangkungraharjo@telkomunive  
rsity.ac.id

3<sup>rd</sup> Nur Ibrahim  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
nuribrahim@telkomuniversity.  
ac.id

**Abstrak**—Meningkatnya volume lalu lintas selama dekade terakhir menimbulkan tantangan besar pada penelitian lalu lintas dan perencanaan. Penyebab kemacetan diantaranya rambu lalu lintas yang tidak teratur serta kapasitas jalan yang tidak seimbang dengan kebutuhan masyarakat. Dengan adanya permasalahan tersebut menimbulkan tantangan pada peneliti untuk membuat sebuah sistem yang dapat melakukan perhitungan dan pencatatan pada lalu lintas. Perhitungan secara manual bisa dilakukan akan tetapi kurang efisien. Apalagi Sistem pendeteksi, hitung dan catat kepadatan lalu lintas secara *realtime* masih terbatas. Pada penelitian ini sebuah sistem pendeteksi kepadatan lalu lintas menggunakan metode *Haar Cascade* dengan menggunakan bahasa pemrograman *python*, penulis juga menggunakan komponen seperti modul kamera, *raspberry pi*, *power supply*. Komponen *raspberry pi* sebagai pengolahan data untuk memproses citra digital menjadi perhitungan jumlah kendaraan di jalan raya. Kemudian sistem akan menentukan apakah lalu lintas sedang lancar, ramai atau padat dengan parameter yang sudah penulis tentukan sebelumnya. Tugas Akhir ini menggunakan metode *Haar Cascade* dengan fitur *Haar-Like Feature*, *Internal Image*, *Adaboost* dan *Cascade Classifier*, dengan hasil pengujian deteksi objek tersebut mendapatkan tingkat akurasi tertinggi sebesar 88,02 % dengan parameter pemakaian RAM pada *Raspberry pi* sebesar 87,9MB dari 3827MB RAM yang tersedia di *Raspberry pi* dan didapatkan FPS pada sistem dari 20FPS hingga 30FPS.

**Kata Kunci**— *raspberry pi*, *haar cascade*, *traffic light*, *image processing*, *computer vision*

## I. PENDAHULUAN

Meningkatnya volume kendaraan selama satu dekade terakhir memunculkan tantangan besar pada penulis untuk membuat sistem yang berkaitan dengan lalu lintas [1]. Penyebab kemacetan ada beberapa diantaranya rambu lalu lintas yang tidak teratur, penggunaan ruang jalan yang tidak efisien dan kapasitas jalan yang tidak seimbang dengan kebutuhan masyarakat, di tambah dengan masyarakat yang lebih suka menggunakan kendaraan pribadi dibandingkan transportasi umum. Akibat dari kurang lancarnya mobilitas di jalan

mengakibatkan kecelakaan di jalan raya [2]. Dengan adanya permasalahan tersebut menimbulkan tantangan pada peneliti untuk merancang sebuah sistem yang dapat melakukan perhitungan dan pencatatan pada lalu lintas. Perhitungan secara manual bisa dilakukan akan tetapi kurang efisien [3]. Selain itu sistem pendeteksi, hitung dan catat kepadatan lalu lintas secara *realtime* masih terbatas, yang membuat inisiatif peneliti semakin besar untuk merancang teknologi ini.

Sistem pendeteksi kepadatan lalu lintas pernah diciptakan dengan menggunakan metode YOLO dan didapatkan akurasi 98,80% akan tetapi pada pengimplementasian tidak menggunakan *hardware* dan saat pengujian sistem menggunakan *video dummy* [4]. Sistem perancangan penghitungan kepadatan kendaraan di jalan tol menggunakan metode Gaussian Mixture Model dan Kalman Filter juga pernah diciptakan sebelumnya dengan hasil akurasi tertinggi 93,03% [2]. Sistem secara *realtime* yang bekerja untuk mendeteksi wajah dengan *Haar Cascade Classifiers* dengan menggunakan komponen CCTV pernah dibuat dan didapatkan akurasi tertinggi sebesar 93% [5]. Pada tugas akhir ini penulis akan melakukan perancangan pendeteksian kepadatan lalu lintas menggunakan metode *Haar cascade* dengan komponen *Raspberry pi* dan *Web Camera*.

Sistem yang dibuat penulis akan mendeteksi kendaraan di jalan raya secara *realtime*, ketika sebuah kendaraan melewati garis virtual maka sistem akan melakukan perhitungan, dengan hasil yang didapatkan akan ditarik kesimpulan apakah keadaan lalu lintas tersebut sedang padat, ramai atau lancar. Pada perancangan kali ini penulis menggunakan metode *Haar Cascade Classifier* untuk mendeteksi objek yang bergerak. *Haar Cascade Classifier* merupakan salah satu algoritma yang dapat mendeteksi secara cepat dan *Realtime* pada sebuah objek, pada algoritma ini memiliki kelebihan yang dapat melakukan komputasi yang cepat karena hanya bergantung pada jumlah piksel

dan tidak membutuhkan spesifikasi *Hardware* yang *High End*[6].

Oleh karena itu, solusi yang ditawarkan adalah dibuatkannya Alat Pemantauan Kepadatan Lalu Lintas menggunakan metode Haar Cascade agar membantu dalam perhitungan dan pencatatan transportasi di jalan raya yang efisien.

## II. KAJIAN TEORI

### A. Object Detection

*Object Detection* merupakan bagian dari proses *Computer Vision*. *Object Detection* ialah proses melakukan pencarian sebuah objek didalam gambar atau video dengan ditandai adanya bounding box[6]. *Object Detection* bisa melakukan pemrediksian objek yang telah terdeteksi dengan class object, bounding box dan titik koordinat disetiap objek[6]. *Object Detection* berbeda dengan *Image Classification Algorithm* yang hanya dapat melakukan proses klasifikasi objek pada citra ke dalam suatu kelas saja tanpa adanya prediksi lokasi dari objek.

### B. Pengolahan Citra Digital

Pengolahan citra digital ialah pemrosesan gambar dua dimensi yang menggunakan computer yang tertuju.pada pemrosesan setiap data.pada gambar dua dimensi. Citra digital adalah sebuah array yang berisi nilai asli dan kompleks yang mempresentasikan dengan deretan bit tertentu. Citra digital dapat diwakilkan berupa matrix dua dimensi  $f(x,y)$  yang terdiri dari (M) untuk kolom dan (N) untuk baris, dimana setiap perpotongan antara baris dan kolom dapat disebut piksel.

#### 1. Citra RGB

Citra RGB adalah citra dengan yang memiliki komponen warna merah, hijau dan biru dengan memiliki piksel masing-masing 3 atau bisa disingkat menjadi RGB (red, green, blue). Setiap warna dipiksel diwakili dari kombinasi intensitas warna merah, hijau dan biru[7]. Pada setiap warna RGB mempunyai intensitas warna yang berbeda-beda dengan jumlah 8 bit, sehingga ruang pada warna di RGB mempunyai total 24 bit. dapat disimpulkan bahwa untuk menghasilkan warna yang luas maka diperlukan kombinasi dari 3 warna dasar yang setiap warnanya memiliki intensitas sendiri dengan nilai maksimum di 255 (8 bit)[8].

#### 2. Citra Grayscale

Citra yang dapat ditunjukkan oleh citra grayscale terdiri dari warna abu, yang bagian terlemah didominasi warna hitam dan pada intensitas yang kuat berwarna putih [8].

### C. Python

Python merupakan bahasa pemrograman tingkat tinggi yang bersifat *interpreter, interactive, object oriented* dan bisa dapat beroperasi di *platform Mac,*

*Linux Windows, Java Virtual Machine, OS/2, Amiga, Symbian.* Python merupakan bahasa pemrograman yang berfokus untuk memudahkan dalam memahami setiap source code, karena sintaks pada source code yang jelas dapat digabungkan dengan menggunakan modul-modul siap pakai, dan struktur data yang efisien. Bahasa pemrograman Python merupakan produk yang diciptakan secara open source dan multiplatform.

### D. OpenCV

OpenCV (Open Source Computer Vision Library) merupakan sebuah library open source yang diciptakan oleh intel yang berfokus untuk mempermudah membuat program terkait citra digital. *Open computer vision* adalah salah satu bidang pengolahan citra yang memungkinkan sebuah *computer* memiliki penglihatan seperti manusia. OpenCV memiliki berbagai fitur, diantaranya : pengenalan dan pelacakan wajah, deteksi objek, Kalman filtering serta berbagai jenis *AI (Artificial Intelligence)*. Untuk *low level* memiliki algoritma sederhana terkait *Computer Vision*. OpenCV mempunyai karakter open source komputer untuk bahasa pemrograman C/C++, dan sudah dikembangkan dengan bahasa pemrograman python, java dan matlab.

### E. Metode Viola-Jones

Haar Cascade yaitu salah satu machine learning yang biasa digunakan sebagai aplikasi *Object Detection* (diutamakan face recognition) [10]. Metode Viola-Jone merupakan penggabungan dari support vector machines, cascade classifier serta algoritma boosting. Metode viola-jones memiliki beberapa keunggulan pada hasil yang akurat dan efisien. Proses pendeteksian sebuah objek dilakukan dengan mengklasifikasikan gambar. Pada Metode Viola-Jones memiliki empat langkah dalam mengolah sebuah gambar, yaitu *Haar Like Features, Integral, AdaBoost dan Cascade Classifier*.

#### 1. Haar Like Features

Haar like feature merupakan representasi fungsi dari persegi, yang dimana fungsi tersebut menandakan terdapat sebuah objek. Fitur ini menggunakan gelombang haar yang merupakan gelombang kotak. Pada image dua dimensi gelombang adalah pasangan persegi yang warnanya 1 terang dan 1 gelap [11].

Fitur sepasang persegi ditunjukkan pada A dan B, C menunjukkan fitur tiga persegi dan D menunjukkan fitur dua pasang persegi. Haar like features tidak menggunakan nilai piksel sebagai indikator untuk mendeteksi objek dan nilainya ditentukan oleh pengurangan rata – rata pada piksel gelap dan piksel terang.

#### 2. Integral Image

Integral ialah penambahan nilai-nilai kecil yang bersamaan, sedangkan pada Viola-Jones terdapat penambahan unit-unit kecil yang terdiri dari nilai piksel serta nilai integral ke piksel dari jumlah dari semua piksel dari atas hingga ke bawah. Diawali pada sudut atas kiri hingga sudut bawah kanan, untuk keseluruhan citra tersebut bisa ditotalkan dengan operasi bilangan bulat per piksel [11]. Haar like feature menggunakan perhitungan yang sangat banyak secara matematis untuk mengimplemensikannya di dalam source code, dengan adanya integral image maka perhitungan fitur akan jauh lebih cepat dengan cara menjumlahkan nilai pixel dari sebuah image.

### 3. AdaBoost

Viola-Jones menggunakan algoritma adaboost yang memiliki kepanjangan adaptive boosting yang bertujuan untuk meningkatkan tingkat akurasi dan membentuk template objek yang akan dideteksi. Algoritma Adaboost membangun strong Classifier dengan mengkombinasikan sejumlah simple secara linier [11].

Adaboost mengklasifikasikan pada classifier lemah yang akan digabungkan dengan classifier yang kuat, yang dimana classifier lemah merupakan classifier yang mempunyai nilai error paling sedikit. Terdapat dua data yang digunakan pada proses training adaboost classifier yaitu kelas positif dan kelas negative.

### 4. Cascade Classifier

Pada tahapan berikutnya yaitu Cascade classifier. Alur filter dari Cascade classifier ditentukan oleh total yang ada di AdaBoost. Cascade classifier merupakan proses klasifikasi bertingkat. Klasifikasi ini menggabungkan fitur yang sudah di training dari proses adaboost classifier [11]. Kemudian akan melakukan proses pengecekan apakah di dalam suatu image objek yang penulis uji terdeteksi atau tidak ada.

### F. Raspberry Pi 4

Raspberry Pi 4 adalah komponen terbaru dari jajaran komponen lainnya yang dirilis oleh Raspberry pi. Raspberry Pi 4 mempunyai keunggulan dari generasi sebelumnya seperti meningkatnya kecepatan prosesor, memori, multimedia serta pada konektivitas jaringan yang lebih baik. Pada produk Raspberry Pi 4 Model B memberikan kinerja *desktop* yang setara dengan sistem komputer *x86 entry-level*.

Fitur unggulan pada produk ini merupakan prosesor *quad-core 64-bit* dengan performa tinggi dengan dukungan layar ganda dengan resolusi hingga 4K melalui sepasang *port micro-HDMI*, decode video perangkat keras hingga 4Kp60, Raspberry pi 4 memiliki RAM dari 2 GB hingga 8 GB dengan didukung konektivitas dual-band 2.4 / 5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB

3.0, dan kemampuan PoE (melalui *add-on PoE HAT* terpisah).

Raspberry pi 4 memiliki LAN *nirkabel dual-band* serta pada Bluetooth mempunyai sertifikasi kepatuhan modular, yang memungkinkan board dibuat dijadikan sebagai produk akhir dengan pengujian kepatuhan yang berkurang sangat signifikan.

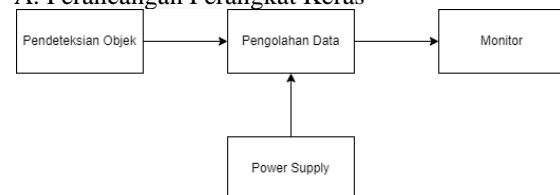
### G. Web Camera

*Web Camera* ialah singkatan dari *Web Camera*, *Web Camera* sendiri merupakan perangkat kamera digital yang dihubungkan ke komputer atau laptop. Didalam *Web Camera* tidak hanya terdapat kamera saja akan tetapi terdapat fitur lain seperti mikrofon, engsel kamera yang dapat diatur sesuai posisi yang diinginkan, sensor yang dapat mendeteksi gerakan, serta terdapat lampu indikator yang akan menyala ketika kamera sedang digunakan.

Pada Sistem ini penulis menggunakan *Web Camera Logitech C270* karena memiliki Resolusi HD 720p/30fps yang mampu mendeteksi objek dengan baik, *Web Camera C270* mampu menampilkan warna yang alami dengan keadaan realtime.

## III. METODE

### A. Perancangan Perangkat Keras



GAMBAR 3.1

PERANCANGAN PERANGKAT KERAS

Pada penelitian ini merancang sebuah sistem pendeteksi kepadatan lalu lintas menggunakan metode Haar Cascade dengan menggunakan bahasa pemrograman python, penulis menggunakan beberapa komponen seperti modul kamera, raspberry pi, power supply dan monitor. Pada desain sistem diatas pendeteksian objek menggunakan *Web Camera* yang berfungsi untuk mendeteksi objek kendaraan dan menjadi masukan data yang akan diproses di Raspberry pi dengan menggunakan metode haar cascade classifier. Komponen Raspberry pi sebagai pengolahan data yang akan memproses citra digital dan mengkategorikan serta menghitung kendaraan roda empat dan roda dua di jalan raya secara realtime. Pada sistem ini membutuhkan komponen monitor yang berfungsi untuk memulai sistem pendeteksian dan mengetahui hasil perhitungan kendaraan di jalan raya, monitor juga berfungsi untuk melihat keadaan lalu lintas ketika sedang padat, ramai ataupun lancar. Penulis juga membutuhkan power supply yang berfungsi untuk menyuplai tegangan langsung ke

raspberry pi, power supply bisa menggunakan tegangan AC maupun DC.

### B. Diagram Alir



GAMBAR 3.2  
DIAGRAM ALIR

Ketika alat dijalankan Raspberry pi akan menghubungkan *Web Camera* yang sudah terintegrasi oleh sistem, *Web Camera* yang sudah terhubung akan mendeteksi ke ruas jalan untuk mengambil citra digital kendaraan, jika sebuah kendaraan terdeteksi oleh *Web Camera* maka citra akan diolah di raspberry pi dengan metode Haar Cascade. Hasil dari sistem ini ialah jumlah kendaraan roda empat dan roda dua di jalan raya, dengan hasil yang didapatkan akan ditarik kesimpulan apakah keadaan lalu lintas tersebut sedang padat, ramai atau lancar.

#### 1. Pendeteksi Objek

Pada Proses pendeteksi objek peneliti menggunakan komponen *Web Camera* Logitech C270 yang akan menangkap video secara realtime yang menjadi masukan sistem ini. Ketika sistem dinyalakan kamera akan mendeteksi objek kendaraan dalam bentuk citra digital dan akan dikirimkan ke raspberry pi untuk menjadi data masukan yang akan diproses dengan metode haar cascade dan akan ditampilkan pada monitor sebagai keluaran akhir pada sistem ini.

#### 2. Pengolahan Data

Proses selanjutnya ialah pengolahan data dimana data masukan dari kamera akan diproses di raspberry pi dan haar cascade. Penulis menggunakan komponen Raspberry pi yang berfungsi sebagai Mini Computer karena Raspberry pi dapat menjalankan Operation System(OS) sendiri. Seperti komputer pada umumnya Raspberry pi dapat melakukan pengolahan angka, pengolahan kata, memutar multimedia seperti music atau video serta dapat menyimpan file. Pada sistem yang penulis rancang, raspberry pi bisa menjadi pusat pengolahan data dari masukan *Web Camera* yang menangkap objek

kendaraan di ruas lalu lintas. Pada perancangan sistem ini penulis menggunakan raspberry pi 4 karena pada komponen tersebut terdapat port HDMI Video Output yang akan berfungsi sebagai output ke monitor untuk melihat hasil perhitungan dan juga terdapat Port CSI Camera yang berguna untuk menjadi inputan dari *Web Camera*.

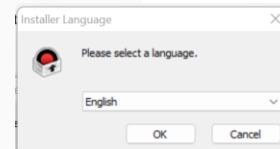
#### 3. Konfigurasi Alat

Pada rancangan sistem ini, konfigurasi alat sangat dibutuhkan karena konfigurasi alat merupakan cara instalasi sistem, pengaturan komponen dan pengoperasian sistem alat rancangan kedalam komputer. *Web Camera* yang sudah terhubung ke raspberry pi yang dijadikan sebagai mini computer. Penulis menggunakan laptop sebagai pengganti monitor untuk mengoperasionalkan sistem dan juga untuk mengetahui jumlah kendaraan di lalu lintas secara realtime, selain itu monitor juga berfungsi untuk mengetahui performansi pada Raspberry pi. Pada sistem ini konfigurasi alat menggunakan bahasa pemrograman python untuk menghubungkan antara kamera dengan raspberry pi dan juga laptop sebagai pengganti komponen monitor.

#### 4. Instalasi Bandicam

Bandicam adalah program perekam untuk Windows yang bisa merekam kegiatan di layar PC-mu dengan kualitas tinggi. Fungsi Bandicam pada sistem ini ialah sebagai konfigurasi untuk menampilkan UI pada mini PC *raspberry pi*. Berikut cara instalasi bandicam pada Windows:

##### a. Memilih Bahasa untuk sistem Bandicam



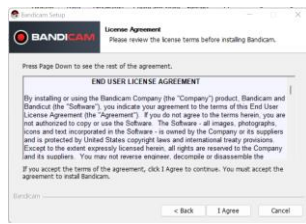
GAMBAR 3.3  
PEMILIHAN BAHASA UNTUK BANDICAM

##### b. Pilih tombol "Next"



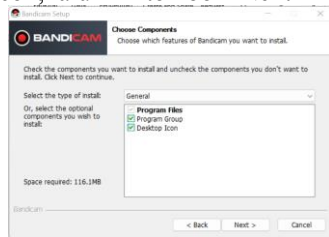
GAMBAR 3.4  
TAMPILAN AWAL INSTALASI

##### c. Pilih tombol "I Agree"



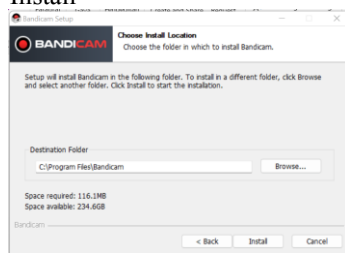
GAMBAR 3.5  
LISENSI BANDICAM

- d. Mencentang Program Group dan Desktop Icon, lalu klik tombol “Next”



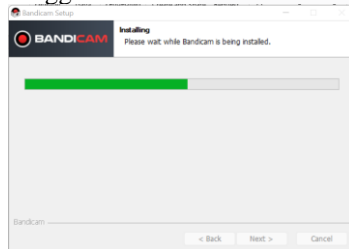
GAMBAR 3.6  
PEMILIHAN KOMPONEN INSTALASI

- e. Memilih lokasi penyimpanan di “C:\Program Files\Bandicam” lalu klik “Install”



GAMBAR 3.7  
PILIH LOKASI PENYIMPANAN BANDICAM

- f. Tunggu Proses Instalasi



GAMBAR 3.8  
PROSES INSTALASI

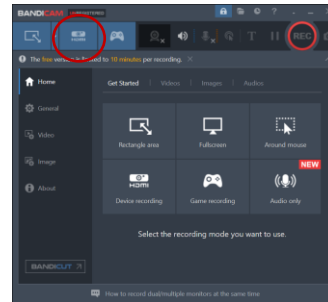
- g. Bandicam berhasil terpasang di windows.



GAMBAR 3.9  
INSTALASI SELESAI

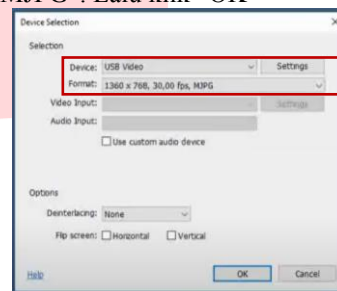
Berikut cara untuk menampilkan UI pada mini PC raspberry pi:

- a. Klik ikon HDMI



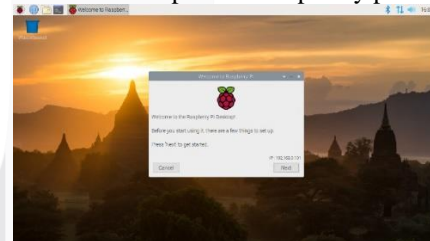
GAMBAR 3.10  
TAMPILAN BANDICAM

- b. Pada pengaturan pilih “USB Video” di kolom device dan “1360 x 768, 30,00 fps, MJPG”. Lalu klik “OK”



GAMBAR 3.11  
PENGATURAN UNTUK MENAMPILKAN UI RASPBERRY PI

- c. Berikut tampilan UI raspberry pi



GAMBAR 3.12  
UI RASPBERRY PI

#### IV. HASIL DAN PEMBAHASAN

##### A. Performansi Alat

Performansi alat bertujuan untuk mengetahui kemampuan alat pada saat mendeteksi kepadatan lalu lintas, dengan mengetahui performa alat penulis dapat mengetahui kemampuan dari Raspberry pi. Performansi alat dilakukan dengan cara melakukan pengujian RAM dan pengujian FPS.

##### 1. Pengujian RAM

Pengujian RAM (Random Access Memory) bertujuan untuk mengetahui pemakaian RAM pada Raspberry pi. Pengujian dilakukan pada saat menjalankan pendeteksian kepadatan lalu lintas. Pengujian RAM dapat dilakukan dengan melihat pada Task Manager di Raspberry pi pada saat sistem mulai mendeteksi. Pengujian RAM dilakukan pada

sistem sedang melakukan pendeteksian kepadatan lalu lintas. Berikut adalah hasil pengujiannya:

- a. Pengujian RAM pada jam 08:30

TABEL 4. 1  
PENGUJIAN RAM SKENARIO 1

No	Waktu	Pemakaian RAM
1	08:35	87.2 MB
2	08:40	87.9 MB

- b. Pengujian RAM pada jam 13:30

TABEL 4. 2  
PENGUJIAN RAM SKENARIO 2

No	Waktu	Pemakaian RAM
1	13:35	87.5 MB
2	13:40	88.2 MB

- c. Pengujian RAM pada jam 16:00

TABEL 4. 3  
PENGUJIAN RAM SKENARIO 3

No	Waktu	Pemakaian RAM
1	16:05	87.3 MB
2	16:10	87.7 MB

Berdasarkan hasil pengujian ini, penggunaan RAM pada pengujian pagi, pengujian siang, dan pengujian sore hari cenderung menghasilkan hasil yang sama, yaitu menghabiskan sekitar 87.93MB yang dimana setiap skenario pengujian deteksi dilakukan selama 5 menit. Hal ini menyebabkan kenaikan penggunaan RAM tidak terlalu signifikan. Raspberry pi masih berfungsi dengan baik dalam menjalankan proses deteksi, artinya sistem dapat berfungsi dengan sempurna selama kamera dapat memperoleh visual objek yang baik.

## 2. Pengujian FPS

Pengujian FPS (*Frame per Second*) bertujuan untuk mengetahui di FPS berapa sistem dapat mendeteksi objek yang stabil dan mengolah citra dengan baik serta *delay* yang sangat sedikit. Pengujian FPS dilakukan dengan cara menambahkan *Source Code* pada program yang dibuat agar bisa menampilkan FPS secara langsung. Pengujian FPS dilakukan 3 kali pada saat pengambilan data di jalan Cihampelas. FPS yang didapatkan pada ujicoba alat ialah 20 hingga 30FPS karena pada *Web Camera* yang digunakan memiliki maksimal FPS sebesar 30. Penulis menyebarkan kuisisioner ke 10 responden dengan latar belakang yang berbeda-beda dengan tujuan untuk mengetahui keterangan dari berbagai pihak setelah melihat video pendeteksian dari FPS 20 hingga 30. Responden melihat video hasil pendeteksian dan kemudian responden akan menjawab 5 pertanyaan yang sudah tersedia di *Google Form*. Hasil dari kuisisioner yang dibagikan terhadap 10 responden dapat dilihat sebagai berikut.

TABEL 4. 4  
HASIL KUISISIONER FPS

Frame per Second	Keterangan		
	Tidak Lancar	Cukup Lancar	Lancar
20	6 Orang	3 Orang	1 Orang
22	4 Orang	5 Orang	1 Orang
24	3 Orang	5 Orang	2 Orang
26		6 Orang	4 Orang
28		2 Orang	8 Orang
30			10 Orang

Dari hasil survei yang didapatkan responden memberikan keterangan semakin besar fps yang didapatkan maka semakin lancar video yang berjalan.

## B. Intensitas Cahaya

Pengujian intensitas cahaya bertujuan untuk mengetahui nilai intensitas cahaya dalam sistem pendeteksian lalu lintas. Intensitas cahaya sangat mempengaruhi dalam pendeteksian kendaraan di jalan raya. Pengukuran intensitas cahaya diambil dibawah sinar matahari secara langsung. Intensitas cahaya diukur melalui Lux Meter yang tersedia di ios. Pengujian ini dilakukan 3 kali di tempat yang sama pada pemasangan *Web Camera*. Waktu pengujian dilakukan pada jam 08:30, 13:30, dan 16:00 dibawah sinar matahari langsung. Berikut data hasil pengujian.

TABEL 4. 5  
HASIL PENGUJIAN INTENSITAS CAHAYA

No	Waktu Pengujian	Intensitas Cahaya	Akurasi Alat
1	08:30	150 Lux	88,02 %
2	13:30	175 Lux	63,04 %
3	16:00	168 Lux	75,98 %

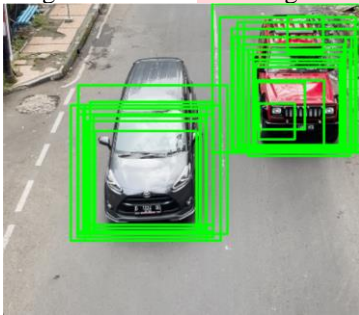
## 1. Analisis Pengujian

Dari hasil pengujian pada jam 08:30 didapatkan intensitas cahaya sebesar 150 Lux dengan tingkat akurasi alat sebesar 88,02 %. Dengan intensitas 150 Lux *Web Camera* dapat melakukan pendeteksian secara maksimal karena dapat menangkap citra yang sesuai dengan data uji. Pada pengujian jam 13:30 didapatkan intensitas cahaya sebesar 175 Lux dengan tingkat akurasi alat 63,04 %. Pada siang hari cahaya matahari yang begitu cerah mengakibatkan warna kendaraan yang lewat menjadi buram dan perhitungan kendaraan menjadi tidak bekerja secara maksimal. Hasil pengujian pada jam 16:30 didapatkan akurasi sebesar 75,98 % dengan intensitas cahaya 168 Lux. Dengan begitu alat dapat bekerja secara maksimal karena dapat mendeteksi kendaraan sesuai warna di data uji dan dapat menghitung kendaraan yang lewat.

Dapat disimpulkan dari hasil pengujian diatas intensitas cahaya sangat berpengaruh pada kinerja *Web Camera*. Alat dapat bekerja secara maksimal ketika cahaya matahari tidak begitu terik seperti pada pagi dan sore hari. Sementara ketika cahaya matahari sangat terik mengakibatkan pendeteksian tidak begitu akurat, seperti pada siang hari intensitas cahaya sangat tinggi yang berakibat pada warna mobil menjadi buram saat tertangkap pada *Web Camera*.

C. Pengujian Minimum Neighbors

Pengujian ini dilakukan untuk mengetahui penggunaan perubahan nilai minimum neighbors terhadap akurasi pendeteksian objek kendaraan. Minimum neighbors merupakan argument yang menentukan nilai minimum jumlah bounding box atau *feature* objek tetangga yang harus ada untuk mengindikasi *bounding box* dengan tepat pada area objek kendaraan. Misalkan pada contoh Gambar 4.1 dibawah dengan nilai minimum neighbors 0:

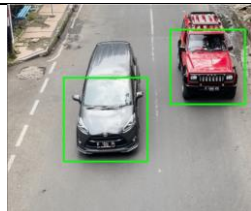


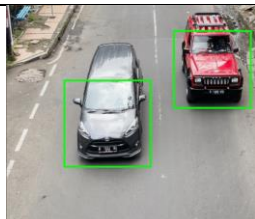
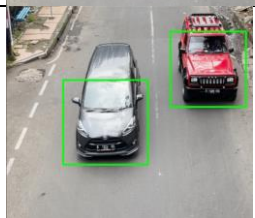
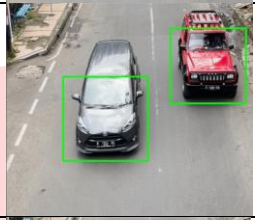
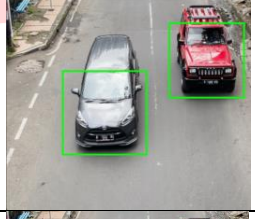
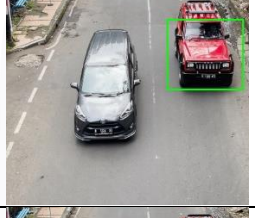
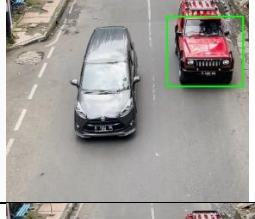
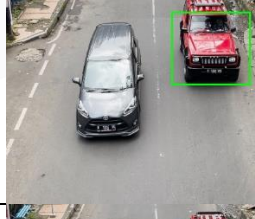

GAMBAR 4. 1  
PENGUJIAN MINIMUM NEIGHBORS 0


Terlihat pada gambar diatas terdapat banyak *bounding box* yang mengindikasikan objek dikarenakan banyak *feature* yang mirip dengan kendaraan pada proses pengujian, sehingga menghasilkan *false positive* yang banyak. Oleh karnanya dibutuhkan nilai minimum *neighbors* yang tepat untuk menghilangkan banyaknya false positive pada proses deteksi. Berikut beberapa nilai minimum *neighbors* yang diuji pada penelitian ini:

TABEL 4. 6

PARAMETER PENGUJIAN MINIMUM NEIGHBORS

No	Hasil Gambar	Minimum Neighbors	Akurasi
1		1	100%

2		2	100%
3		3	100%
4		4	100%
5		5	100%
6		6	50%
7		7	50%
8		8	50%
9		9	50%

10		10	0%
----	---	----	----

Dari nilai *minimum neighbors* diatas dapat disimpulkan bahwa nilai yang memberikan akurasi yang baik adalah antara 1 sampai 5 sedangkan untuk nilai yang menghasilkan miss adalah antara 6 sampai 10. Namun sesungguhnya dalam proses pengujian data secara *real time* minimum neighbors dengan nilai 1 sampai 5 juga terdapat error deteksi, namun memiliki akurasi yang jauh lebih baik dengan minimum neighbors dengan nilai 6 sampai 10. Sehingga nilai yang digunakan untuk minimum neighbors pada pengujian ini adalah 5.

## V. KESIMPULAN

### A. Kesimpulan

Berdasarkan hasil pengujian data dan analisis pada alat yang dibuat dapat disimpulkan sebagai berikut:

1. Pendeteksian Objek dengan metode Haar Cascade menggunakan fitur Haar-Like Feature, Internal Image, Adaboost dan Cascade Classifier, dengan hasil pengujian deteksi objek tersebut mendapatkan tingkat akurasi tertinggi sebesar 88,02 %.
2. Sistem dengan tingkat akurasi 88,02% didapatkan dengan parameter pemakaian RAM pada Raspberry pi sebesar 87,9MB dari 3827MB RAM yang tersedia di Raspberry pi dan didapatkan FPS pada sistem dari 20FPS hingga 30FPS.
3. Intensitas cahaya sangat mempengaruhi performansi alat saat mendeteksi karena warna pada mobil menjadi samar pada saat sinar matahari terkena langsung pada mobil seperti yang terjadi disiang hari.

## REFERENSI

- [1] A. K. Dewantoro, I. Iwut, and E. Susatio, "Simulasi dan Analisis Sistem Penghitung Kepadatan Lalu Lintas dan Klasifikasi Kendaraan Berbasis *Web Camera* dengan Metode Background Subtraction," *Signal Processing*, vol. 1, p. 2, 2015.
- [2] M. J. Yuwono, B. Purnama, and F. Sthevanie, "Penghitungan Kepadatan Kendaraan Di Jalan Tol Menggunakan Metode Gaussian Mixture Model Dan Kalman Filter," *e-Proceedings Appl. Sci.*, vol. 3, no. 2, pp. 1–7, 2015, doi: 10.21108/indosc.2015.5.
- [3] E. H. Mujahid Habibullah, Dedy Rahman Wijaya, "Aplikasi Pemantau Kepadatan Lalu Lintas Berbasis Web dengan Video Analisis Menggunakan Camlytics ( Dinas Kominfo Gunungkidul )," *e-Proceedings Appl. Sci.*, vol. 5, no. 2, pp. 1–10, 2019.
- [4] M. I. HermaTelkomwan et al., "Pengaturan Lampu Lalu Lintas Berdasarkan Kepadatan Kendaraan Menggunakan Metode Yolo," *e-Proceeding Eng.*, vol. 8, no. 1, pp. 198–205, 2021.
- [5] M. R. A. Pangestu, S. Sumaryo, and C. Setianingsih, "Real Time Cctv Deteksi Wajah Dengan Haar Cascade Classifiers Opencv," *e-Proceeding Eng.*, vol. 6, no. 2, pp. 1–8, 2019.
- [6] S. Abidin, "Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis *Web Camera* Pada Matlab," *J. Teknol. Elekterika*, vol. 15, no. 1, p. 21, 2018, doi: 10.31963/elekterika.v15i1.2102.
- [7] E. C. Rahmad, S. T. M. Kom, D. Rawansyah, M. Pd, and T. K. Rochastu, "Sistem Objek Deteksi Sebagai Alat Bantu Mendeteksi Objek Sekitar untuk Penyandang Tunanetra.," *e-Proceeding Eng.*, vol. 4, no. 1, pp. 81–88, 2018.
- [8] M. R. Pratama, Rizal, and S. Sumaryo, "Desain Sistem Deteksi Objek Real Time Dengan Metode Haar Cascade Classifier," *e-Proceeding Eng.*, pp. 26–34, 2020.
- [9] Dicoding Space, "Memulai Pemrograman Dengan Python," *Dicoding*, 2020. <https://www.dicoding.com/academies/86> (accessed Nov. 11, 2021).
- [10] A. D. Yusyahnur, A. Virgono, and U. A. Ahmad, "Pengukuran Jarak Kendaraan Dengan Metode Haar Cascade Menggunakan OpenCV," *J. Teknol. Elekterika*, vol. 1, pp. 2–12, 2022.
- [11] I. P. Hadi et al., "Perancangan Sistem Deteksi Dan Pengenalan Wajah Menggunakan Metode Haar Cascade Classifier Dan Local Binary Pattern Histogram," *J. Teknol. Elekterika*, vol. 3, no. 1, pp. 1–8, 2021.