

**IMPLEMENTASI SISTEM PENDETEKSI KENDARAAN DAN JARAK AMAN
PADA KENDARAAN BERDASARKAN PENGOLAHAN CITRA MENGGUNAKAN
METODE *HISTOGRAM OF ORIENTED GRADIENT* PADA *EMBEDDED BOARD***

***IMPLEMENTATION OF VEHICLE DETECTOR AND SAFETY DISTANCE SYSTEM
BASED ON IMAGE PROCESSING USING HISTOGRAM OF ORIENTED GRADIENT
METHOD ON EMBEDDED BOARD***

TUGAS AKHIR

Disusun dalam rangka memenuhi salah satu persyaratan untuk menyelesaikan program Studi
Strata 1 Sistem Komputer

Disusun oleh :

Harits Arifandi Saputra

1103148192



Program Studi Sarjana Sistem Komputer

Fakultas Teknik Elektro

Bandung

2016

LEMBAR PENGESAHAN

**IMPLEMENTASI SISTEM PENDETEKSI KENDARAAN DAN JARAK AMAN
PADA KENDARAAN BERDASARKAN PENGOLAHAN CITRA MENGGUNAKAN
METODE *HISTOGRAM OF ORIENTED GRADIENT* PADA *EMBEDDED BOARD*
*IMPLEMENTATION OF VEHICLE DETECTOR AND SAFETY DISTANCE SYSTEM
BASED ON IMAGE PROCESSING USING HISTOGRAM OF ORIENTED GRADIENT
METHOD ON EMBEDDED BOARD***

HARITS ARIFANDI SAPUTRA

NIM: 1103148192

Disusun dalam rangka memenuhi persyaratan dalam mengajukan Tugas Akhir pada
Program Studi Sarjana Sistem Komputer
Fakultas Teknik Elektro
Universitas Telkom

Proposal ini disetujui untuk menyelesaikan Tugas Akhir
Bandung, 26 Oktober 2015

Calon Pembimbing I

Calon Pembimbing II

Agus Virgono .Ir.,MT

Andrew Brian Osmond.ST.,MT

NIK: **93660027**

NIK : 10860736-3

	UNIVERSITAS TELKOM	No. Dokumen	ITT-AK-FEK-PTT-FM-004/001
	Jl. Telekomunikasi No. 1 Ters. Buah Batu Bandung 40257	No. Revisi	00
	FORMULIR PERNYATAAN ORISINALITAS	Berlaku Efektif	02 Mei 2011

LEMBAR PERNYATAAN

Nama : HARITS ARIFANDI SAPUTRA
NIM : 1103148192
Alamat : Jln. Perwira No 4 Walikota Kupang – Kupang Nusa Tenggara Timur
No. Tlp/HP : +6282144223352
E-mail : harits.a.saputra@gmail.com

Menyatakan bahwa Tugas Akhir II ini merupakan karya orisinal saya sendiri, dengan judul:

IMPLEMENTASI SISTEM PENDETEKSI KENDARAAN DAN JARAK AMAN PADA KENDARAAN BERDASARKAN PENGOLAHAN CITRA MENGGUNAKAN METODE *HISTOGRAM OF ORIENTED GRADIENT* PADA *EMBEDDED BOARD*

IMPLEMENTATION OF VEHICLE DETECTOR AND SAFETY DISTANCE SYSTEM BASED ON IMAGE PROCESSING USING HISTOGRAM OF ORIENTED GRADIENT METHOD ON EMBEDDED BOARD

Atas pernyataan ini, saya siap menanggung risiko / sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.



Bandung, September 2016

Harits Arifandi Saputra

1103148192

ABSTRAK

Pada penelitian ini dibuat sebuah sistem yang dapat mendeteksi mobil pada jarak aman untuk membantu pengendara dalam meminimalisir tingkat kecelakaan. Metode yang digunakan yaitu *Histogram of Oriented Gradient* (HOG) untuk mendeteksi mobil pada video. Karakteristik *feature HOG* dari sebuah mobil ditampilkan oleh distribusi nilai gradien berupa garis vektor. Hasil dari *feature HOG* dilearning oleh *Support Vector Machine* (SVM) untuk mendapatkan acuan deteksi antar data mobil atau bukan mobil. *Learning* dilakukan dengan menggunakan 800 data positif (mobil) dan 800 data negatif (bukan mobil). Setelah itu digunakan raspberry sebagai *embedded board* untuk memproses keseluruhan sistem.

Hasil pengujian menunjukan *classifier* yang digunakan berada di tingkat akurasi 75% dengan kondisi jalan tanpa *obstacle* (pohon, tiang penunjuk arah, jembatan) dan tingkat akurasi 70% dengan kondisi jalan memiliki banyak *obstacle* (pohon, tiang penunjuk arah, jembatan). Kemudian pada pengujian raspberry menunjukan bahwa terdapat selisih 20 detik antar video yang telah diimplementasikan sistem dan yang belum diimplementasikan sistem. Dan pada tujuan akhir dari system dapat mendeteksi objek dalam batas respon dengan waktu dibawah satu detik untuk tiap-tiap kecepatan.

Kata kunci : *embedded board*, , *histogram of oriented gradient*, *support vector machine*.

ABSTRACT

In this study, a system that detects the car at a safe distance to help drivers in minimizing the accident rate. The method used is the Histogram of Oriented Gradient (HOG) to detect the car in the video. HOG feature characteristics of a car shown by the distribution of gradient value in the form of vectors. Results of HOG feature dilearning by Support Vector Machine (SVM) to get a reference between data detection or a car is not a car. Learning is done by using positive data 800 (cars) and 800 negative data (not the car). And it is used as an embedded board raspberries for processing the whole system.

The test results showed that the classifier used is in the 75% accuracy rate with the condition of the road with no obstacles (trees, poles signpost, bridges) and 70% accuracy rate with the condition of the road has a lot of obstacles (trees, poles signpost, bridges). Then in raspberry testing showed that there is a difference between the 20 seconds of video that have been implemented and the systems that have not implemented the system. And the goal of the system can detect objects within a response time below one second for each speed.

Keyword : *embedded board, , histogram of oriented gradient, support vector machine.*

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT yang telah memberikan karunia dan hidayah-Nya sehingga Tugas dapat penulis selesaikan. Tugas Akhir ini disusun sebagai persyaratan untuk menyelesaikan studi penulis pada Program Studi S1 Sistem Komputer Fakultas Elektro Universitas Telkom.

Penulis menyadari masih banyak kesalahan dalam Tugas Akhir ini penulis memohon maaf, semua ini semata dikarenakan kekurangan penulis sendiri. Kritik dan saran yang membangun sangat penulis butuhkan untuk kesempurnaan Tugas Akhir ini.

Dengan segala kerendahan hati, penulis berharap semoga Tugas Akhir yang telah dibuat ini dapat bermanfaat bagi pembaca dan penulis pada khususnya, serta bagi masyarakat pada umumnya.

Terima Kasih

Bandung, 20 Oktober 2016

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PERNYATAAN	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metode Penyelesaian	2
1.5.1 Studi Literatur	2
1.5.2 Analisa Kebutuhan	3
1.5.3 Perancangan Sistem	2
1.5.4 Pengujian Sistem	3
1.5.5 Analisis Pengujian	3
1.5.6 Penyusunan Laporan	3
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Pengolahan Citra Digital	5
2.2 Histogram Of Oriented Gradient	6
2.3 Support Vector Machine	8
2.4 Embedded System	9
BAB III PERANCANGAN SISTEM	11
3.1 Gambaran Umum Sistem	11
3.2 Pengolahan Citra	13

3.3 Inisialisasi Kamera	13
3.4 Preproses	13
3.5 Mengambil Frame Video.....	13
3.6 Grayscale.....	14
3.7 Histogram of Oriented Gradient.....	14
3.8 Support Vector Machine	15
3.9 Penentuan Jarak Aman	16
BAB IV PENGUJIAN DAN ANALISIS	17
4.1 Spesifikasi Sistem.....	17
4.1.1 Perangkat Keras, Raspberry Pi 2 Model B	17
4.1.2. Perangkat Lunak.....	17
4.2 Pengujian Sistem	17
4.3 Skenario Pengujian Sistem.....	18
4.3.1 Pengujian Classifier	18
4.4 Data Analisis Sistem	19
4.5 Pengujian sistem.....	19
4.5.1 Pengujian Classifier	20
4.5.2 Pengujian Raspberry	26
4.5.3 Pengujian deteksi HOG.....	27
4.5.4 Pengujian Jarak Aman.....	28
BAB V KESIMPULAN DAN SARAN	33
5.1 Kesimpulan.....	33
5.2 Saran.....	33
DAFTAR PUSTAKA	34

DAFTAR GAMBAR

Gambar 2.1 Algoritma Histogram of Oriented Gradient	6
Gambar 2.2 Filter pada HOG	7
Gambar 2.3 Cell pada Histogram of Oriented Gradient	7
Gambar 3.1 Skema perancangan sistem.....	12
Gambar 3.2 Blok diagram preproses.....	13
Gambar 3.3 Konversi citra RGB menjadi Grayscale.....	14
Gambar 3.4 Alur kerja metode Histogram of Oriented Gradient.....	15
Gambar 3.5 Alur kerja Support Vector Machine	15
Gambar 3.6 Penentuan jarak aman.....	16
Gambar 4.1 positif sampel	20
Gambar 4.2 Negatif sampel.....	21
Gambar 4.3 Total data sampel (training dan testing).....	22
Gambar 4.4 sample 1.....	24
Gambar 4.5 sample 2.....	25
Gambar 4.6 sample 3.....	26
Gambar 4.7 128x128p.....	27
Gambar 4.8 64x64p.....	28
Gambar 4.9 jarak aman	29
Gambar 4.10 kalibrasi jarak aman	30

DAFTAR TABEL

Tabel 4.1 Data Analisis	19
Tabel 4.2 pengujian classifier	23
Tabel 4.3 pengujian raspberry	26
Tabel 4.4 pengujian scan HOG	27
Tabel 4.5 standar jarak aman	26
Tabel 4.6 perbandingan jarak aman	30
Tabel 4.7 parameter pengujian jarak aman	31
Tabel 4.8 pengujian jarak aman	31

BAB I

PENDAHULUAN

1.1 Latar Belakang

Peningkatan jumlah kendaraan khususnya roda empat semakin memperbesar resiko kecelakaan. Hal ini tidak diimbangi dengan teknologi yang mendukung keselamatan pengendara. Masalah yang menjadi tolak ukur penyebab kecelakaan yaitu kelalaian pengemudi dalam mengambil jalur aman.

Sistem keamanan dibutuhkan dalam mendukung upaya menurunkan resiko terjadinya kecelakaan. Salah satu penerapan yang dibutuhkan yaitu pendeteksi posisi mobil yang berada didepannya agar pengendara dapat mengetahui jarak aman dalam berkendara.

Pada kasus ini penulis ingin membuat sebuah sistem yang diterapkan dalam sistem keamanan mobil dengan memanfaatkan metoda pengolahan *image processing* untuk mendeteksi jarak aman kendaraan dan akan diimplementasikan langsung pada kendaraan. Maka dari itu *embedded board* akan digunakan sebagai perangkat utama untuk melakukan pendeteksian citra dan perhitungan jarak aman antar kendaraan. Dengan begitu penulis berharap dengan adanya sistem tertanam ini maka akan mengurangi tingkat kecelakaan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dibahas diatas dan kemudian diidentifikasi, maka perumusan masalah yang didapati dipenelitian ini adalah sebagai berikut :

1. Bagaimana sistem dapat mengenali objek berupa kendaraan yang berada didepannya.
2. Bagaimana sistem dapat mendeteksi sebuah objek pada jarak aman yang telah ditentukan?
3. Apakah sistem dapat digunakan pada *embedded board*?

1.3 Tujuan

Adapun tujuan dari penyusunan tugas akhir ini adalah sebagai berikut :

1. Merancang suatu sistem yang dapat mengenali objek berupa kendaraan yang berada didepannya.
2. Merancang suatu sistem yang dapat mendeteksi sebuah objek berupa kendaraan terhadap jarak aman yang telah ditentukan
3. Merancang suatu sistem yang menggunakan *embedded board*.

1.4 Batasan Masalah

Adapun batasan-batasan yang akan dibahas pada tugas akhir ini yaitu :

1. Bahasa pemograman yang digunakan adalah *python* dan *library open CV* dalam mengolah data citra.
2. Objek yang akan diamati yaitu kendaraan pada jalan searah atau jalan yang cenderung lurus.
3. Objek kendaraan yang diamati berada didepan pengemudi dan bukan berada di belakang kendaraan pengemudi.
4. Pengolahan citra dilakukan pada siang hari.
5. Perangkat keras utama (*embedded board*) yang digunakan untuk mengolah citra dan sistem keseluruhan ialah raspberry pi.

1.5 Metode Penyelesain

1.5.1 Studi Literatur

Tahap ini bertujuan untuk mendapatkan studi literatur sebagai sumber acuan dalam penulisan tugas akhir dan pengembangannya tentang teori-teori mengenai pengolahan citra untuk mendeteksi objek menggunakan metoda *Histogram Of Oriented Gradient* dan *Support Vector Machine*.

1.5.2 Perancangan Sistem

Pada tahap ini sistem pendeteksi kendaraan dibangun pada perangkat keras *Embedded Board* dan sistem operasi berbasis Linux yang dapat mendukung analisa

pengolahan citra. Rancangan tersebut dibuat agar sistem dapat berjalan senantiasa sesuai dengan apa yang diharapkan dan ditujukan.

1.5.3 Pengujian Sistem

Pada tahap ini dilakukan pengujian sistem pendeteksi kendaraan untuk dilihat apakah sistem berjalan secara baik dan menghasilkan keluaran yang sesuai dan tepat sesuai dengan perancangannya.

1.5.4 Analisis Pengujian

Pada tahap ini dilakukan analisis dari pengujian sistem pendeteksi kendaraan yang telah dibuat dan dilakukan evaluasi berupa kendaraan yang berhasil di deteksi berupa gambar dan perhitungan dari sistem yang telah dibuat. Hal ini bertujuan agar sistem mampu berjalan dengan output sesuai yang diharapkan.

1.5.5 Penyusunan Laporan

Pada tahap ini dilakukan penyusunan laporan tugas akhir dan dokumentasi yang diperlukan untuk tugas akhir ini.

1.5.6 Analisa Kebutuhan

Dalam tahap ini dilakukan analisa kebutuhan sesuai dengan kebutuhan sistem yang akan dibangun. Kebutuhan yang dianalisa dibagi menjadi analisa data dan analisa spesifikasi kebutuhan sistem. Analisa tersebut dilakukan agar sistem yang akan dibuat dapat berjalan dengan baik.

1.6 Sistematika Penulisan

Sistematika dalam penulisan tugas akhir ini dibagi menjadi beberapa pokok pembahasan yaitu :

BAB I PENDAHULUAN

Pada bab ini dijelaskan latar belakang masalah dari pembuatan sistem pendeteksi kendaraan dan jarak aman berdasarkan pengolahan citra menggunakan metode *Histogram of Oriented Gradient*, perusmusan masalah, tujuan, batasan masalah, metode penyelesaian masalah dan sistematika penelitian.

BAB II TINJAUAN PUSTAKA

Pada bab ini dijelaskan teori dasar yang berisi uraian singkat yang berhubungan dengan materi penelitian

BAB III PERANCANGAN DAN IMPLEMENTASI

Pada bab ini dijelaskan tentang perancangan sistem secara umum, alur pengerjaan dan penyelesaian sistem, serta algoritma dan metode yang digunakan untuk mendeteksi mobil serta jarak aman antara mobil yang terdeteksi dan pengendara.

BAB IV PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang hasil pengujian, evaluasi dan implikasi dari perancangan dan implementasi sistem.

BAB V KESIMPULAN DAN SARAN

Pada bab ini menjelaskan kesimpulan dan saran dari penulis untuk pengembangan sistem.

BAB II

LANDASAN TEORI

2.1 Pengolahan Citra Digital

Pengolahan citra adalah suatu sistem yang akan melakukan proses input dan output sama-sama berupa nilai citra. Pada pengolahan citra digital informasi yang terkandung bersifat diskret dan tidak selalu merupakan hasil langsung dari video atau data rekaman. Dalam suatu citra digital diperlukan proses konversi untuk dapat diproses dalam komputasi menggunakan komputer.

Dalam pengolahan citra, terdapat noise citra yang disebabkan tingkat keberadaan derau sensor dan kamera yang kurang fokus. Sehingga perlu dilakukan proses filter citra pada citra digital.

Citra yang didapat seringkali mempunyai derau sehingga sebelum dianalisa citra perlu dihaluskan dengan tapis citra. Hal ini dapat dilakukan dengan memanipulasi piksel-piksel tetangga. Desain tapis tersebut membuat citra lebih halus dan bentuk sudut benda dalam citra tetap terjaga. Ada beberapa tapis citra yang digunakan untuk menghilangkan derau. Derau aditif (Gaussian) biasanya ditapis dengan tapis Wiener. Derau Multiplikatif (Speckle) biasanya dihilangkan dengan tapis Homomorfik. Sedangkan derau salt-and-pepper biasanya digunakan tapis median atau LPF untuk menghilangkannya.

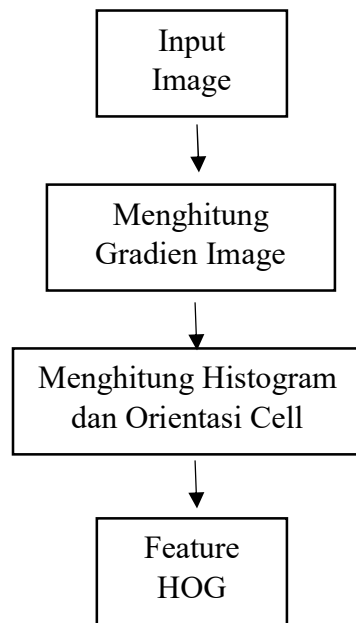
Segmentasi dalam citra digital digunakan untuk membagi citra menjadi bagian-bagian untuk kemudian dapat dianalisis. Segmentasi sendiri merupakan proses pemisahan latar depan-latar belakang.

Pada pemrosesan citra digital akan dipelajari bagaimana struktur dan topologi dalam objek-objek citra. Proses ini bertujuan untuk mengenali objek [1].

2.2 Histogram Of Oriented Gradient

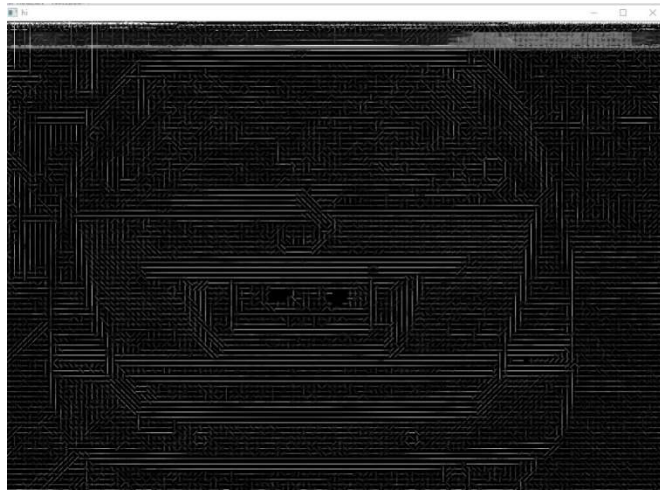
Histogram of Oriented Gradient (HOG) adalah sebuah metode yang digunakan dalam *image processing* untuk tujuan deteksi obyek. Teknik ini menghitung nilai gradien dalam daerah tertentu pada suatu gambar. Tiap gambar mempunyai karakteristik yang ditunjukkan oleh distribusi gradien. Karakteristik ini diperoleh dengan membagi gambar kedalam daerah kecil yang disebut *cell*. Tiap *cell* disusun sebuah *histogram* dari sebuah gradien. Kombinasi dari *histogram* ini dijadikan sebagai deskriptor yang mewakili sebuah obyek.

Algoritma *Histogram Of Oriented Gradient* adalah sebagai berikut;



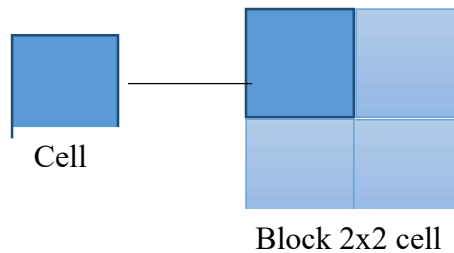
Gambar 2.1 Algoritma *Histogram of Oriented Gradient*

Dari gambar 2.1 tahap awal dari *HOG* adalah menghitung nilai gradien dari input gambar. Metode yang paling umum untuk menghitung besarnya gradien adalah dengan menggunakan *sobel filter* dalam satu atau dua arah, baik secara horisontal atau vertikal.



Gambar 2.2 Filter pada *HOG*

Selanjutnya adalah membuat bagian-bagian yang dinamakan *cell*. Setiap piksel dalam sebuah *cell* mempunyai nilai *histogram* sendiri-sendiri berdasarkan nilai yang dihasilkan dalam perhitungan gradien. *Cell* memiliki ukuran 4×4 *pixel* pada sebuah gambar sedangkan *block* memiliki ukuran 2×2 *cell* atau 8×8 *pixel*.

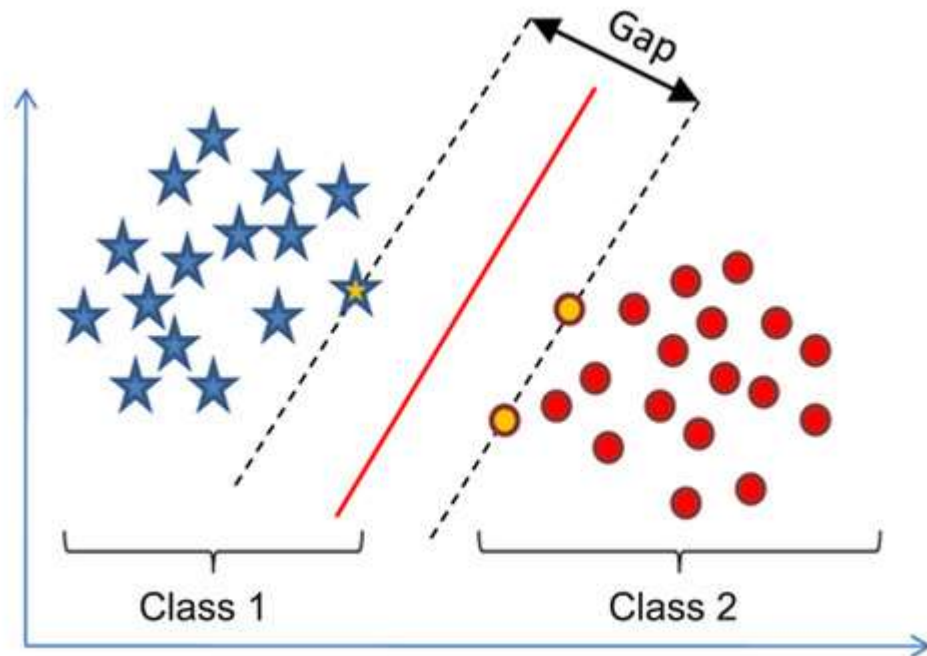


Gambar 2.3 Cell pada *Histogram of Oriented Gradient*

Hasil *feature Histogram of Oriented Gradient* dirubah menjadi *feature vector* untuk diproses kedalam *Super Vector Machine classifier*. *Super Vector Machine classifier* digunakan untuk menemukan garis pemisah (*hyperplane*) dari *class +1* yang berisi *feature vector* positif dengan *class -1* yang berisi *feature vector* negatif. Setelah dilatih model hasil *training* diuji untuk diketahui tingkat akurasi dalam mengenali obyek yang dites [2].

2.3 Support Vector Machine

Konsep *SVM* dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada *input space*.



Gambar 2.4 *support vector machine*

Gambar 2.4 memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class* : *class* 1 (dinotasikan dengan +1) dan *class* 2 (dinotasikan dengan -1). *Pattern* yang tergabung pada *class* 2 disimbolkan dengan lingkaran, sedangkan *pattern* pada *class* 1, disimbolkan dengan bintang. Proses pembelajaran dalam problem klasifikasi diterjemahkan sebagai upaya menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. *Hyperplane* pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan data terdekat dari masing-masing *class*. *Subset data training set* yang paling dekat ini disebut sebagai *support vector*. Garis solid pada gambar 2.4 menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik kotak dan lingkaran yang berada dalam lingkaran hitam adalah *support vector*. Upaya mencari lokasi *hyperplane* optimal ini merupakan inti dari proses pembelajaran pada *SVM* [3].

2.4 Embedded System

Embedded system adalah sebuah sistem komputer yang didesain untuk melakukan suatu pekerjaan khusus. Embedded system biasanya merupakan bagian dari perangkat yang lebih besar yang didalamnya terdapat hardware dan peralatan mekanik. Tidak seperti sistem komputer yang di bangun di PC, embedded system relatif lebih cepat dalam *runtime* bahkan seringkali *real time* karena *embedded system* dibangun dengan bahasa pemrograman yang lebih dekat/dikenali oleh hardware.

2.4.1 Komponen Utama Embedded System

Embedded system memiliki tiga komponen utama, yaitu:

1. Hardware

- Power Supply
- Processor
- Memory
- Timers

- Serial communication ports
- Output/Output circuits
- System application specific circuits

2. Software

Embedded system memiliki software yang ditulis dalam bahasa yang lebih dekat/dikenali oleh hardware, misalnya Assembly, C, C++, Java

3. Real-Time Operating System

RTOS membawahi software dan menyediakan mekanisme agar prosesor dapat menjalankan proses sesuai jadwal dan melakukan switching dari satu proses (tugas) ke proses lain. Contoh RTOS: VxWorks, OS9, RTLinux, Symbian.

2.4.2 *Microcontroller* dan Hubungannya dengan *Embedded System*

Microcontroller adalah sebuah komputer kecil di dalam satu sirkuit yang berisi inti prosesor, memori, dan input/output peripheral yang dapat diprogram, dan di desain khusus untuk embedded system. Microcontroller berukuran kecil dan murah, sangat berbeda dengan apa yang ada di komputer.

Embedded system tidak membutuhkan kapasitas data dan address bus yang terlalu besar, itu sebabnya *microcontroller* ini berukuran lebih kecil, dan harga harus terjangkau karena akan dipasangkan ke barang-barang elektronik yang harganya juga terjangkau.

2.4.3 Contoh-contoh Implementasi *Embedded System*

Embedded system dapat berjalan karena salah satunya ada microcontroller di dalamnya. Sudah sedikit disinggung di atas bahwa microcontroller akan dipasangkan ke barang-barang elektronik, berarti di dalam barang-barang elektronik itu terdapat embedded system. Contoh yang paling dekat dengan kita adalah barang-barang elektronik yang berhubungan dengan kebutuhan rumah-tangga, misalnya lemari pendingin, mesin cuci otomatis, kompor listrik, televisi, telepon, dan lain-lain [4].

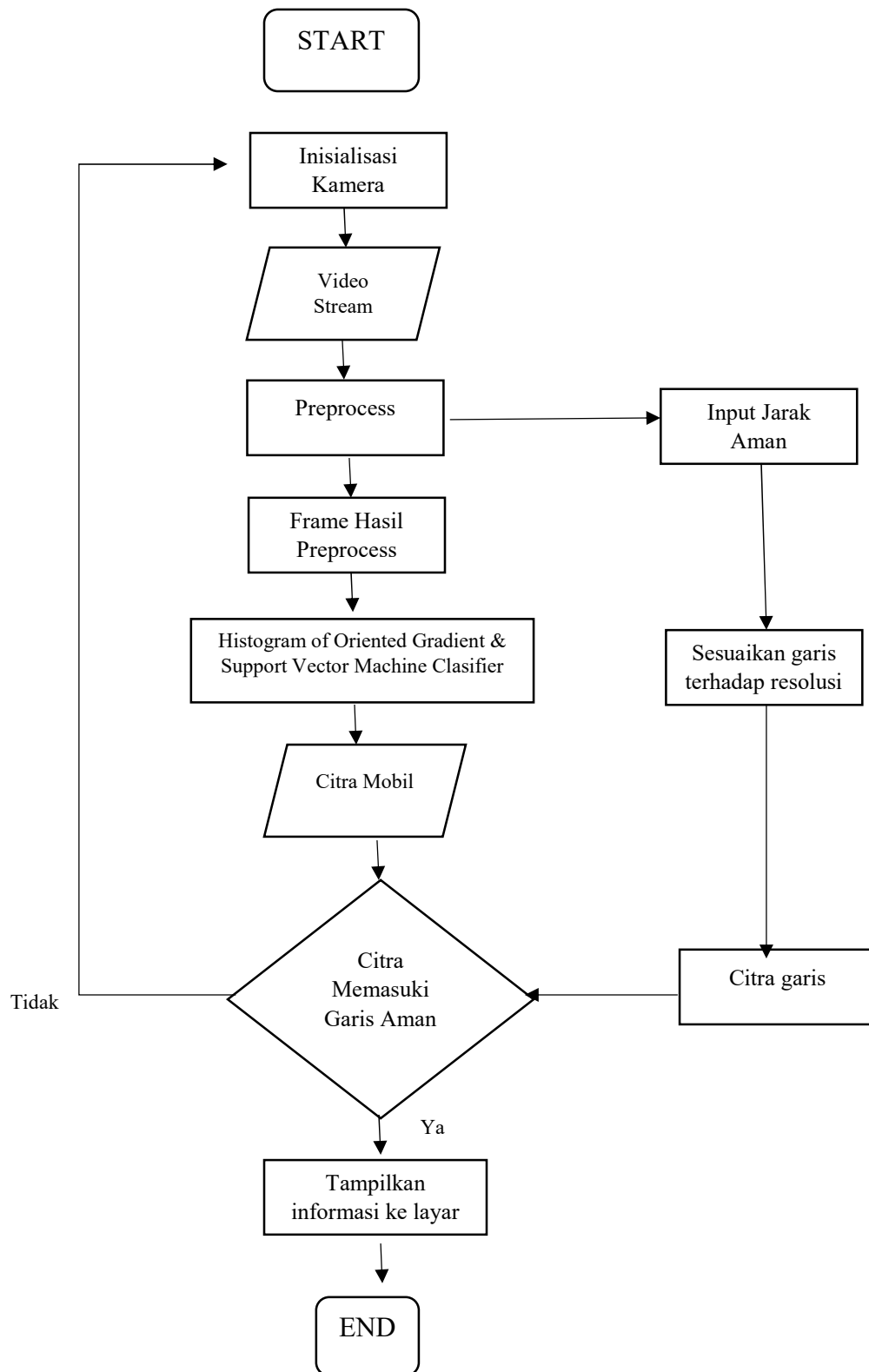
BAB III

PERANCANGAN SISTEM

3.1 Gambaran Umum Sistem

Sistem pendeteksi jarak aman pada kendaraan merupakan sistem yang menangkap citra dari kamera sebagai inputan sistem. Kemudian hasil dari video citra tersebut dianalisis dan diolah. Hasil dari analisa dan pengolahan citra tadi maka sistem dapat menghitung objek yang ada di depan lalu mengklasifikasikannya dan menampilkan hanya objek yang berupa mobil serta mengklasifikasikannya apakah mobil tersebut berada pada garis aman atau tidak.

Secara umum rancangan sistem dalam tugas akhir ini digambarkan pada diagram alir berikut ini :



Gambar 3.1 Skema perancangan sistem

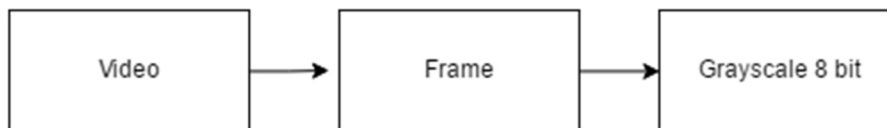
3.2 Pengolahan Citra

Citra yang diterima dari kamera raspberry akan menggunakan metode *Histogram of Oriented Gradient* untuk mendeteksi sebuah objek. Sementara *Support Vector Machine* akan berperan sebagai *learning machine* dalam mengklasifikasikan objek mobil dan objek bukan mobil.

3.3 Inisialisasi Kamera

Pada inisialisasi kamera citra yang didapat akan dianalisa. Tahapan inisialisasi yang digunakan pada python sudah terdapat *library* bernama PiCam. Hasil dari gambar akan dibuat menjadi sebuah array agar mudah dalam pemrosesannya. Pada tugas akhir ini citra akan diambil berdasarkan input melalui kamera dan kemudian akan dianalisis menggunakan video hasil rekaman kamera.

3.4 Preproses



Gambar 3.2 Blok diagram preproses

3.5 Mengambil Frame Video

Pada tahapan ini sebuah video akan diproses akan diambil *memory buffer* berupa frame dari video tersebut. Hal ini dilakukan agar pemrosesan citra lebih baik karena akan diolah berdasarkan tiap frame yang diambil menggunakan kamera.

3.6 Grayscale

Tahapan *grayscale* bertujuan untuk mengurangi beban komputasi dengan mengkonversikan nilai RGB menjadi citra dua warna atau *grayscale*. Dengan begitu maka nilai yang akan didapat dari hasil komputasi untuk mendeteksi objek akan jauh lebih singkat.

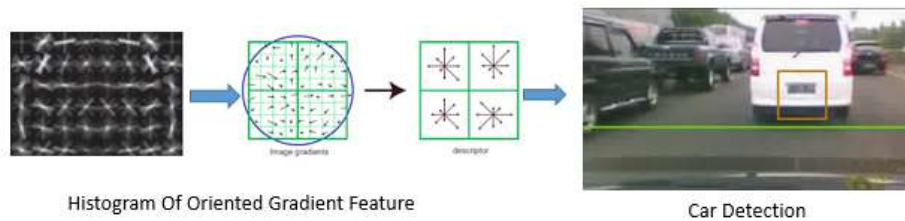


Gambar 3.3 Konversi citra RGB menjadi Grayscale

3.7 Histogram of Oriented Gradient

Pada sistem ini akan digunakan metode *Histogram of Oriented Gradient (HOG)* untuk tujuan deteksi obyek. Metode ini menghitung nilai gradien dalam daerah tertentu pada suatu *image*. Tiap *image* mempunyai karakteristik yang ditunjukkan oleh distribusi gradien. Karakteristik ini diperoleh dengan membagi *image* kedalam daerah kecil yang disebut *cell*. Tiap *cell* disusun sebuah *histogram* dari sebuah gradien. Kombinasi dari *histogram* ini dijadikan sebagai deskriptor yang mewakili sebuah obyek.

Untuk metode pendeteksian menggunakan *Histogram Of Oriented Gradient* adalah sebagai berikut,

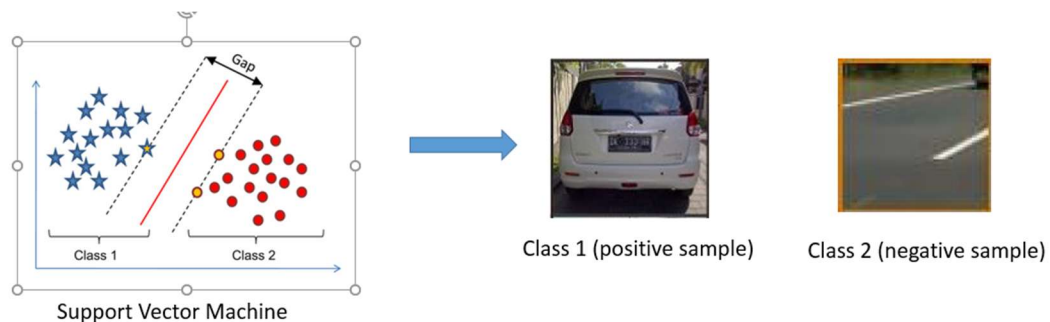


Gambar 3.4 Alur kerja metode *Histogram of Oriented Gradient*

3.8 Support Vector Machine

Setelah mendapatkan nilai *descriptor* dari *histogram of oriented gradient*. Maka tahapan selanjutnya yaitu pengklasifikasian dengan membagi dua *class* (*class* positif dan *class* negatif) dengan membuat *hyperplane* atau garis pisah. Pada sistem ini *support vector machine* akan berperan dalam melakukan *training and learning data* agar objek yang dideteksi sesuai dengan tujuan dari sistem.

Alur sistem pada algoritma *classifier* menggunakan *Support Vector Machine*,



Gambar 3.5 Alur kerja *Support Vector Machine*

3.9 Penentuan Jarak Aman

Jarak aman berdasarkan inputan dari pengguna. Jarak aman direpresentasikan kedalam bentuk *pixel* pada layar. Jarak aman ditentukan berdasarkan pengukuran yang telah dilakukan sebelumnya dan dikalibrasi oleh kamera dalam menentukan jarak jauhnya masing masing inputan terhadap jarak aman. Contohnya, pada jarak 17 meter maka ukuran jarak aman terhadap pixel yaitu sebesar 150 pixel.



Gambar 3.6 Penentuan jarak aman

BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini akan membahas mengenai pengujian dan analisis dari metode *Histogram of Oriented Gradient* dan *Support Vector Machine Classifier*, waktu deteksi dan akurasi deteksi kendaraan serta jarak aman pada kendaraan. Hasil pengujian dimaksudkan untuk mendapatkan pengukuran performansi sistem yang telah dibuat.

4.1 Spesifikasi Sistem

4.1.1 Perangkat Keras, Raspberry Pi 2 Model B

1. Broadcom BCM837 ARM7 Quad Core Processor
2. 1GB RAM
3. 40 PIN extended GPIO
4. Full Size HDMI
5. CSI camera port for connecting Camera

4.1.2. Perangkat Lunak

1. Operating System, Raspbian Jessie
2. Python 2.7
3. Library OpenCV 4.9

4.2 Pengujian Sistem

Pengujian sistem dimaksudkan untuk mengetahui performansi sistem yang telah dibuat. Berikut adalah tujuan dari pengujian sistem yang telah dibuat dalam Tugas Akhir ini:

1. Mengetahui performansi sistem dari metode yang digunakan terhadap kemampuan *embedded board*.
2. Menganalisis hasil kerja sistem agar dapat diambil kesimpulan mengenai performansi sistem.

4.3 Skenario Pengujian Sistem

Pada pengujian system akan dilakukan menggunakan input berupa video. Hal ini ditujukan agar data pengujian dapat diolah sebanyak mungkin dengan berbagai macam sumber video. Berikut adalah skenario pengujiannya:

4.3.1 Pengujian Classifier

Pengujian ini dilakukan dengan membandingkan data yang telah didapat setelah proses deteksi oleh sitem terhadap classifier positif dan classifier negatif. Data yang dihasilkan akan menentukan apakah jumlah classifier sudah memenuhi untuk diimplementasikan pada berbagai medan jalan.

4.3.2 Pengujian Raspberry

Pengujian pada raspberry adalah untuk mencari kualitas video agar sistem dapat berjalan dengan lancar dan membandingkan waktu deteksi sebelum dan sesudah metode diimplementasikan. Data yang akan diuji yaitu berupa nilai *frame rate* dan resolusi pada video.

4.3.3 Pengujian Metode *Scan*

Pengujian metode *scan* yaitu pada “*window slide approach*” akan dilakukan dengan membandingkan tiap nilai *pixel* yang akan digunakan sistem untuk melakukan *scanning* terhadap objek.

4.3.4 Pengujian Jarak aman

Pengujian jarak aman ditunjukan untuk menentukan apakah sistem dapat bekerja sesuai dengan standar jarak aman yang ditentukan.

4.4 Data Analisis Sistem

Pada system akan diuji pengaruh dari classifier, raspberry, metode *HOG*, dan jarak aman dalam kemampuan sistem untuk mendeteksi objek berupa mobil. Data analisis yang akan diambil adalah sebagai berikut;

Tabel 4.1 Data Analisis

Posisi Mobil	Kecepatan	Jarak Aman	jumlah (1 sample/3menit)			Total Mobil	Total Mobil Terdeteksi	Waktu Deteksi Mobil	Akurasi
			Sample 1	Sample 2	Sample 3				
Normal	20km/h	10	3	1	0	4	3	0,41 sec	75%
Sangat dekat	10km/h	5	3	2	1	6	5	0,43 sec	75%
Sangat jauh	40km/h	26	2	2	1	5	3	0,68 sec	60%

Pada table diatas akan dibagi tiga klasifikasi analisis yaitu depan normal, depan sangat dekat dan depan sangat jauh. Pada tiga klasifikasi tersebut memiliki masing-masing nilai parameter yang berbeda. Setelah itu data klasifikasi tersebut diuji menggunakan tiga sampel terhadap metode deteksi. Hasil yang akan diukur adalah total mobil terdeteksi, waktu deteksi, dan akurasi. Untuk mobil yang dihitung adalah ketika melewati garis jarak aman.

4.5 Pengujian sistem

Pengujian system akan dilakukan beberapa tahap sehingga nantinya tiap nilai hasil pengujian akan digunakan untuk menganalisis kemampuan sistem mendeteksi mobil dalam jarak aman yang ditentukan. Hasil akhir dari pengujian sistem adalah kecepatan sistem dapat mendeteksi mobil ketika telah melewati garis aman.

4.5.1 Pengujian Classifier

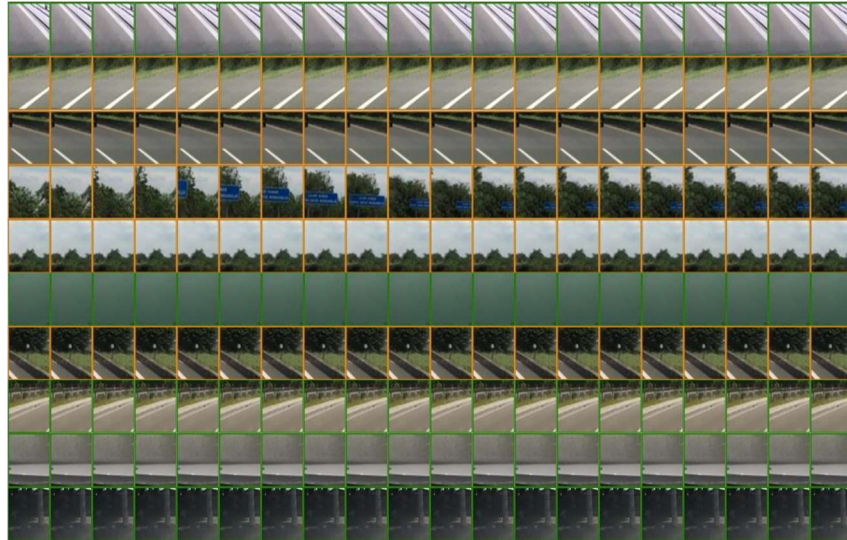
Pada pengujian kali ini bertujuan untuk menentukan apakah data sampel *training* (positif dan negatif) dapat digunakan untuk mendeteksi mobil atau tidak. Classifier positif dan negatif berpengaruh terhadap objek yang akan dideteksi. Dikarenakan terbatasnya data training yang bisa diolah raspy maka data maksimal adalah 400 sampel untuk sampel positif dan 400 sampel untuk sampel negatif. Classifier akan diuji untuk tiap lokasi berbeda untuk menentukan apakah data *training* tersebut dapat digunakan atau tidak. Dalam pemograman akan dilabelkan nilai 0 sebagai data positif (mobil) dan nilai 1-11 sebagai data negatif (bukan mobil).

Untuk sample positif akan diambil gambar mobil. Total sample adalah 400 sample. Dengan membuat label dari 0 untuk tiap baris 128x128 pixel. Gambar dibawah adalah data sampel positif yang digunakan.



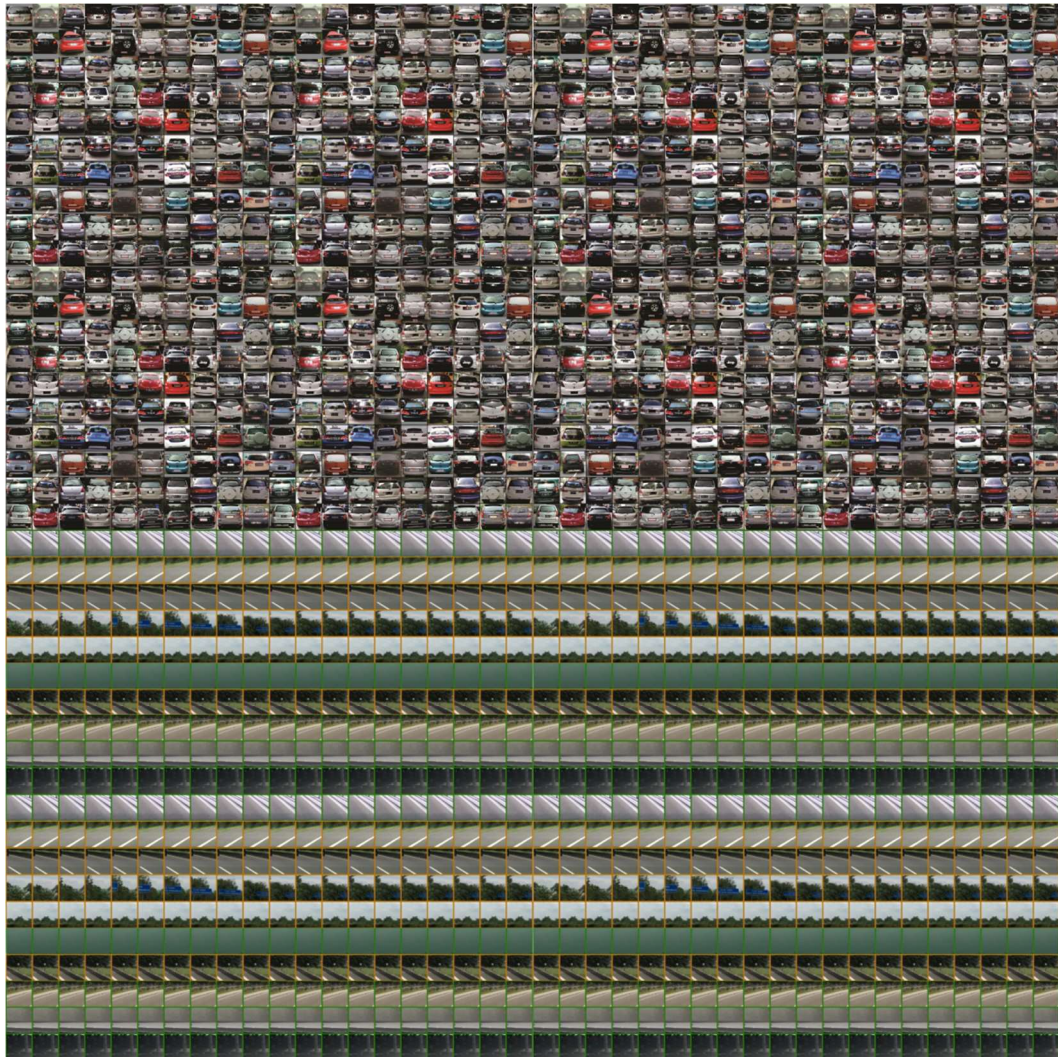
Gambar 4.1 positif sampel

Untuk sampel negatif akan diambil gambar bukan mobil. Total sampel adalah 400 sampel. Dengan membuat label dari 1-11 untuk tiap baris 128x128 pixel. Gambar dibawah adalah data sampel negatif yang digunakan.



Gambar 4.2 Negatif sampel

Pada sistem akan digabungkan seluruh data positif dan data negatif untuk dilakukan *training* dan *testing classifier*. Hasil sampel gabungan dapat dilihat pada gambar dibawah ini.



Gambar 4.3 Total data sampel (*training* dan *testing*)

Total data sampel adalah 1600 dengan 800 sebagai *data training* dan 800 sebagai *data testing*.

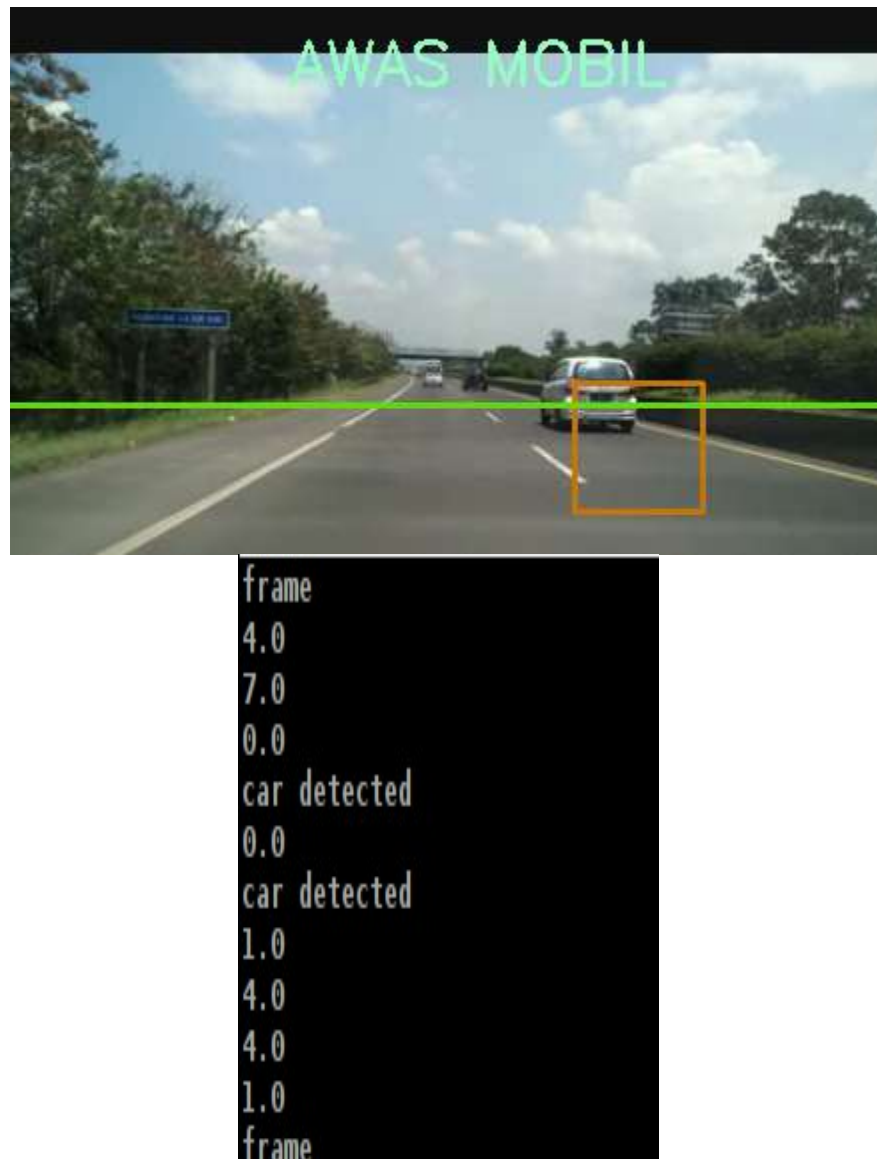
Tabel 4.2 pengujian classifier

Sample	sampel positif		sampel negatif	
	jumlah mobil	Jumlah mobil terdeteksi	jumlah bukan mobil terdeteksi	nilai label sampel negatif
1	5	5	1	1.0,4.0,6.0,7.0
2	4	4	3	1.0, 4.0, 7.0
3	7	7	6	1.0,2.0,4.0,6.0,7.0



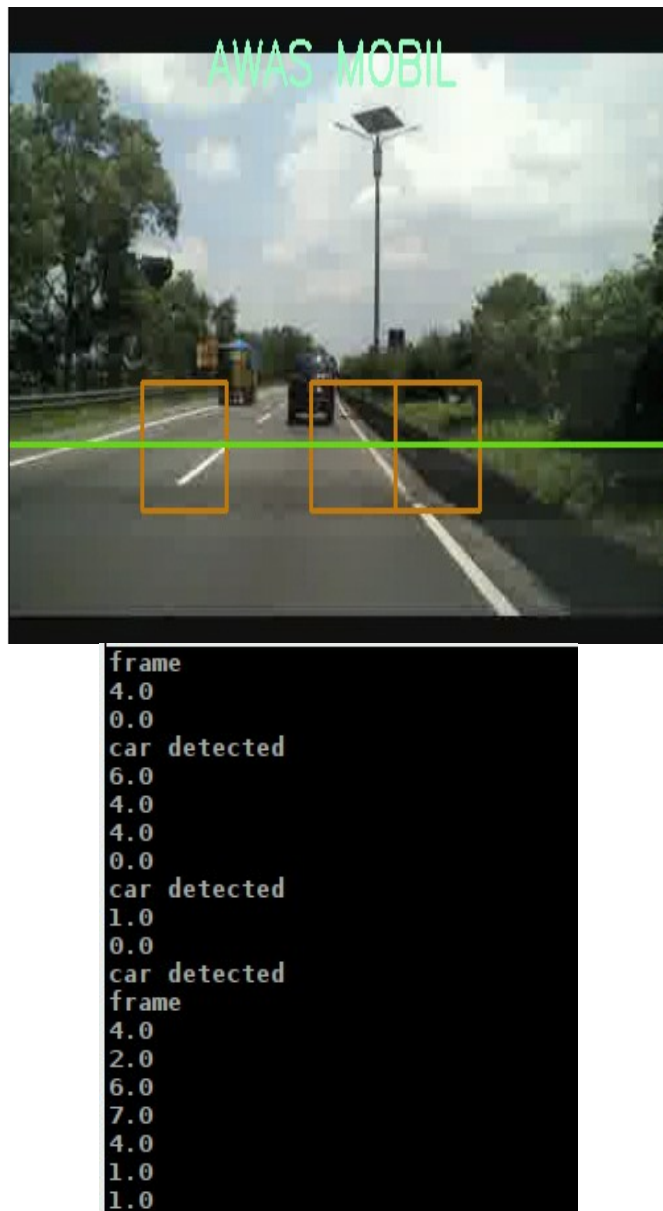
Gambar 4.4 sampel 1

Pada gambar 4.1 dapat dilihat *classifier negative* dapat digunakan pada sistem dengan nilai 1.0, 4.0, 6.0, dan 7.0 sebagai data *sample negative* terdeteksi.



Gambar 4.5 sampel 2

Pada gambar 4.2 *classifier* juga dapat digunakan dalam mendeteksi mobil. Dengan nilai classifier negatif 1.0, 4.0, dan 7.0. Kemudian ketika nilai label mencapai 0.0 maka akan muncul informasi berupa “*car detected*” pada sistem.



Gambar 4.6 sampel 3

Pada gambar 4.6 dapat dilihat bahwa system dapat membaca *classifier negative* dengan nilai label 1.0, 2.0, 4.0, 6.0 dan 7.0 namun ada terjadi kesalahan yaitu objek bukan mobil juga terdeteksi sebagai nilai 0 atau label data mobil. Hal tersebut bisa dilihat dari hasil informasi “*car detected*” berjumlah tiga namun hanya satu bagian objek mobil yang terdeteksi.

Dari hasil pengujian dapat disimpulkan bahwa classifier yang digunakan masih belum cukup dengan adanya data berupa bukan mobil yang masih terdeteksi. Namun dalam mendeteksi mobil atau objek dengan data *classifier* positif sudah berjalan dengan baik. Masalah pada deteksi objek bukan mobil disebabkan kurangnya data sampel negatif sehingga *SVM* sebagai *learning machine* tidak memiliki data yang cukup untuk membentuk klasifikasi yang tepat. Sehingga sebagian objek tidak terdeteksi dengan sempurna.

4.5.2 Pengujian Raspberry

Pengujian raspberry dilakukan untuk mencari pengaruh dari video yang akan diproses terhadap hasil deteksi dari sistem dan perbandingannya terhadap waktu deteksi keadaan nyata. Dalam pengujian ini akan dilakukan perubahan *frame rate* dan resolusi tiap sampel agar dapat menentukan nilai *frame rate* dan resolusi yang memiliki nilai paling cepat dalam mendeteksi objek. Pada percobaan raspberry akan dilihat nilai waktu deteksi dari video yang telah diimplementasikan sistem dengan video asli. Hal ini untuk menentukan apakah raspberry dapat digunakan pada keadaan *real* di jalan.

Tabel 4.3 pengujian raspberry

Percobaan	fps	Resolusi	waktu deteksi(video implementasi sistem)	Waktu deteksi (video asli)	Selisih antara sistem dan video asli
1	30	720p	46 sec	10 sec	36
	15		43 sec		33
	5		41 sec		31
2	30	480p	36 sec		26
	15		33 sec		23
	5		31 sec		21
3	30	320p	35 sec		25
	15		33 sec		23
	5		30 sec		20

Pada hasil dari tabel pengujian diatas nilai fps dan resolusi video yang paling cepat dalam mendeteksi adalah 5fps dan 320p dengan waktu selisih deteksi 10 detik.

4.5.3 Pengujian deteksi HOG

Window slide approach atau metode scan berbasis slide window sangat menentukan objek yang akan dideteksi. Begitupun dengan performansi suatu prosesor apakah dapat mengolah scan tiap pixel atau tidak. Besar kecil nilai pixel yang akan *discan* akan mempengaruhi dari objek yang akan dideteksi. Mulai dari waktu proses deteksi hingga ketepatan objek yang dideteksi sesuai dengan classifier yang dibuat. Parameter yang digunakan adalah ketika objek melewati garis aman.

Tabel 4.3 pengujian scan

Percobaan	Nilai pixel	Objek terdeteksi	Objek terdeteksi (garis aman)	Waktu deteksi(rata-rata)	Akurasi
1	128p	5	2	0,32 detik	40%
2	64p	5	5	0,45 detik	100%

Pada tabel diatas akan dilakukan dua percobaan dengan masing-masing parameter yaitu nilai pixel. Untuk waktu deteksi akan dihitung pada mobil pertama yang terlihat pada video. Sebagai informasi tambahan, nilai dari frame rate dan resolution dari video yang digunakan yaitu 5 fps dan 320p.



Gambar 4.7 128x128 px

Pada gambar diatas adalah nilai scan per 128x128 px. Nilai pixel tersebut dapat mendeteksi objek dengan waktu 0,32 detik lebih cepat dibandingkan nilai pixel 64px yang memiliki waktu 0.45 detik. Namun hasil scan akan terlalu besar sehingga objek yang belum melewati garis aman tetap dihitung dengan akurasi 40%.



Gambar 4.8 64x64 pixel

Pada gambar 4.8 nilai scan per 64x64 px. Nilai pixel tersebut memiliki waktu deteksi 0,45 detik lebih lambat dibandingkan nilai pixel 128px. Tetapi dalam mendeteksi mobil dalam jarak aman lebih tepat dengan akurasi 100%.

Dari pengujian tersebut dapat disimpulkan bahwa nilai scan untuk deteksi mobil yang dapat digunakan yaitu 64x64px.

4.5.4 Pengujian Jarak Aman

Pada pengujian jarak aman akan menentukan waktu deteksi dari sistem. Untuk waktu deteksi akan dihitung berdasarkan waktu respon pengemudi untuk mengerem dan jarak pengereman. Untuk standar jarak aman [5] dan perhitungan yang akan di uji adalah sebagai berikut,



Gambar 4.9 jarak aman

Dari data pada gambar 4.9 ditampilkan bahwa setiap masing – masing kecepatan memiliki jarak maksimal reaksi dan pengereman yang berbeda. Jarak response kendaraan merupakan jarak aman yang dideteksi oleh sistem.

Table 4.5 Standar jarak aman

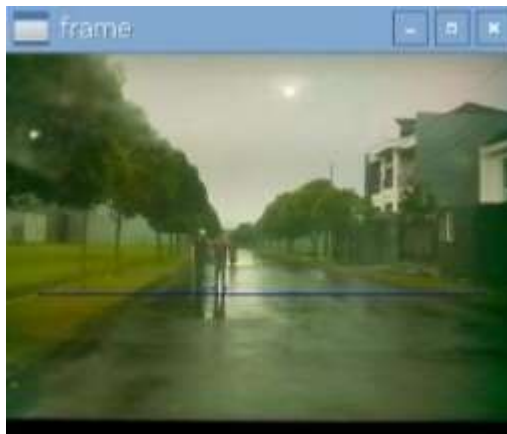
kecepatan mobil	jarak respon/waktu	jarak pengereman	total jarak aman
10km/h	3 meter/1 sec	2m	5m
20km/h	5 meter/1 sec	5m	10m
40km/h	17 meter/1 sec	9m	26m

Tabel diatas merupakan perhitungan berapa jarak aman yang harus dibuat oleh sistem. Sesuai dengan data diatas, maka waktu minimal sistem untuk mendeteksi mobil adalah satu detik untuk tiap jarak respon.

Untuk perbandingan jarak aman pada keadaan nyata dan jarak aman pada gambar adalah sebagai berikut;

Table 4.6 Perbandingan jarak aman

Jarak Nyata(meter)	Jarak Gambar (pixel)
5	210
10	160
17	150



Gambar 4.10 kalibrasi jarak aman

Kalibrasi dilakukan dengan cara mengambil gambar dan mengukur jarak dengan kamera serta dibuatlah suatu garis imajiner sebagai penanda jarak aman dari kendaraan ke sistem.

Hasil pengujian *classifier*, raspberry dan metode *scan* akan digunakan dalam pengujian jarak aman. Hal ini agar dalam mengetahui seberapa cepat system dapat merespon mobil dalam jarak aman berada pada kondisi terbaik. Kondisi terbaik dimaksudkan yaitu nilai *classifier* dapat digunakan pada sistem, waktu deteksi

tercepat dengan kualitas video yang diproses raspberry, dan nilai *scanning* yang memiliki nilai *error* terkecil.

Table 4.7 parameter pengujian jarak aman

Jumlah Classifier	Kualitas Video	Nilai scan
400 (positif) & 400 (negative)	5 fps & 320p	64x64 pixel

Setelah mengetahui informasi jarak aman, batas waktu deteksi dan parameter nilai pengujian, maka akan dilakukan pengujian terhadap sistem. Untuk hasil pengujiannya adalah sebagai berikut:

Tabel 4.8 Pengujian Jarak Aman

Sampel	kecepatan	jarak aman	waktu rata-rata deteksi
1	40km/h	26m	0,46 sec
2	20km/h	10m	0.41 sec
3	10km/h	5m	0,43 sec

Sesuai dengan tabel diatas maka perhitungan jarak respon sistem adalah sebagai berikut:

X = jarak respon maksimal terhadap kecepatan pengendara (meter)

Y = waktu deteksi

Z = Jarak respon

P = sisa jarak respon

- Sampel 1

$$X=17m$$

$$Y=0.46 \text{ detik}$$

$$Z= X*Y =17*0.46=7.82 \text{ meter}$$

$$P=17-7.82=9.18 \text{ meter}$$

Pada sampel 1 dengan kecepatan 40km/h pengemudi mempunyai batas jarak respon sejauh 7.82m untuk melakukan pengereman. Dan jarak sisa dari jarak respon

yaitu 9.18 meter. Jadi dapat dikatakan pada kecepatan 40km/h sistem mampu untuk merespon dalam mendeteksi mobil dengan baik yaitu dengan batas jarak respon sejauh 17 m masih tersisa 9.18 m untuk mencapai batas maksimal respon.

- Sampel 2

$$X=5\text{m}$$

$$Y=0.41 \text{ detik}$$

$$Z= X*Y =5*0.41=2.05 \text{ meter}$$

$$P=5-2.05=2.95 \text{ meter}$$

Pada sampel 2 dengan kecepatan 20km/h pengemudi mempunyai batas jarak respon sejauh 5 untuk melakukan pengereman. Dan jarak sisa dari jarak respon yaitu 2.05 meter. Jadi dapat dikatakan pada kecepatan 30km/h sistem mampu untuk merespon dalam mendeteksi mobil dengan baik yaitu dengan batas jarak respon sejauh 5 m masih tersisa 2.95 m untuk mencapai batas maksimal respon.

- Sample 3

$$X=3\text{m}$$

$$Y=0.43 \text{ detik}$$

$$Z= X*Y =3*0.43=1.29 \text{ meter}$$

$$P=3-1.29=1.71 \text{ meter}$$

Pada sampel 3 dengan kecepatan 10km/h pengemudi mempunyai batas jarak respon sejauh 3m untuk melakukan pengereman. Dan jarak sisa dari jarak respon yaitu 1.29m. Jadi dapat dikatakan pada kecepatan 10km/h sistem mampu untuk merespon dalam mendeteksi mobil dengan baik yaitu dengan batas jarak respon sejauh 3m masih tersisa 1.71m untuk mencapai batas maksimal respon.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Sistem yang dirancang mampu untuk mendeteksi mobil dalam jarak aman dengan tingkat akurasi rata-rata 70%. Namun masih ada kelemahan dalam mendeteksi objek bukan mobil.
2. Sistem mampu untuk mendeteksi mobil dalam jarak aman dengan waktu deteksi yang memenuhi standar yaitu satu detik. Dengan waktu deteksi tercepat adalah 0.41 detik dan waktu deteksi terlambat adalah 0,46 detik.
3. *Embedded board* yang digunakan masih belum cukup untuk diimplementasikan metode *Histogram Of Oriented Gradient* dalam mendeteksi objek. Hal ini dapat dilihat dari selisih waktu deteksi antara video yang telah diimplementasikan metode deteksi dengan video asli yaitu 20 detik dengan frame rate 5 fps dan resolusi video 320p.

5.2 Saran

1. Untuk dapat menerapkan metode *Histogram Of Oriented Gradient* pada keadaan dijalan raya dibutuhkan *embedded board* dengan cpu dan gpu yang lebih baik dari raspberry untuk meminimalisir kesalahan waktu deteksi.
2. Mengembangkan program agar dapat menyesuaikan secara otomatis jarak aman sesuai dengan kecepatan mobil pengendara.

DAFTAR PUSTAKA

- [1] "<http://elektronika-dasar.web.id/>," 23 july 2012. [Online]. Available: <http://elektronika-dasar.web.id/definisi-dan-pengolahancitra-digital/>. [Accessed 22 july 2016].
- [2] K. E. Purnama and M. C. Pratama, "Deteksi Mobil Menggunakan Histogram of Oriented Gradient," 2012.
- [3] S. R. Gunn, "Support Vector Machines for Classification and Regression," p. 1998.
- [4] I. P. A. R, "<http://www.purwadhikapress.com/>," 2 March 2013. [Online]. Available: <http://www.purwadhikapress.com/embedded-system.html>. [Accessed 21 July 2016].
- [5] Q. Government, "<http://tmr.qld.gov.au/>," Department of Transport and Main Roads, 20 september 2013. [Online]. Available: <http://tmr.qld.gov.au/Safety/Driver-guide/Speeding/Stopping-distances.aspx>. [Accessed 26 july 2016].