

# Implementasi Deteksi Kecepatan Kendaraan Menggunakan Kamera Webcam dengan Metode Frame Difference *The Implementation of Vehicle Speed Detection using Webcam with Frame Difference Method*

Nur Hilman Tsani<sup>1</sup>, Ir. Burhanuddin Dirgantoro M. T.<sup>2</sup>, Anggunmeka Luhur Prasasti S. T., M. T.<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Telkom University  
Bandung, Indonesia

<sup>1</sup>hilmantsani@students.telkomuniversity.ac.id, <sup>2</sup>burhanuddin@telkomuniversity.ac.id, <sup>3</sup>anggunmeka@telkomuniversity.ac.id

## Abstrak

Pengguna kendaraan dilingkungan kompleks perumahan terkadang tidak memperhatikan keamanan dan keselamatan lingkungan dengan cara berkendara dengan kecepatan yang sangat tinggi. Tindakan ini dapat membahayakan warga kompleks yang sedang beraktivitas di lingkungan seperti anak-anak yang sedang bermain, warga yang sedang melintas di jalan, dan pengguna kendaraan lainnya. Selama ini sanksi dan tindakan pencegahan terhadap pelanggar sulit dilakukan dikarenakan tidak adanya bukti pelanggaran dan belum adanya sistem yang dapat mencegah tindakan pelanggaran tersebut.

Dalam tugas akhir ini, penulis akan memanfaatkan teknik dari image processing untuk mendeteksi kecepatan kendaraan di lingkungan kompleks perumahan dengan menggunakan kamera webcam. Dengan kamera webcam ini, sistem dapat mendeteksi kecepatan kendaraan berdasarkan suatu urutan *frame* video dengan menggunakan kamera tunggal. Perhitungan kecepatan ini menggunakan metode *frame difference*. Metode ini bekerja dengan membandingkan antar *frame* gambar untuk mendapatkan informasi yang menentukan ada atau tidaknya suatu gerakan. Dari informasi tersebut, bisa didapatkan tingkat kecepatan suatu kendaraan yang berada pada lintasan jalan di kompleks perumahan.

Berdasarkan hasil pengujian pada tugas akhir ini, kamera diletakkan pada tiang dengan ketinggian  $\pm 5,5$  meter dan mengatur posisi sudut kamera dengan memiliki sudut sebesar  $60^\circ$ . Hasil perhitungan kecepatan kendaraan pada sistem mempunyai nilai standar deviasi sebesar 2,82 km/jam dan nilai ketidakpastian  $\pm 1$  km/jam. Sedangkan nilai MSE yang dihasilkan adalah sebesar 12,632. Ketika beberapa objek kendaraan saling berdekatan pada jarak kurang dari 1 meter dalam *frame* yang sama, maka sistem akan mengkategorikan kendaraan pada daerah ROI yang sama sehingga menghasilkan kecepatan yang sama pula.

Kata kunci: Deteksi Kecepatan Kendaraan, *Frame Difference*, *Video Processing*

## Abstract

Vehicle users in residential area sometimes not paying attention to security and environmental safety by driving at a very high speed. This action can endanger residential citizens who are active in the environment such as children who are playing, people who are passing in the street, and other vehicle users. So far, sanctions and precautions against violators are difficult because there is no evidence of violation and no system can prevent such violations.

In this final project, the author will utilize technique of image processing to detect vehicle speed in residential area by using web camera. With this web camera, the system can detect vehicle speed based on a sequence of video frames using a single camera. This speed calculation uses frame difference method. This method works by comparing the frame between images to obtain information that determines whether or not a movement exists. From the information, can be obtained the speed of a vehicle that passes in residential area trajectory.

Based on the results of testing on this final task, the camera is placed on the pole with a height of  $\pm 5.5$  meters and adjust the camera angle position with an angle of  $60^\circ$ . Vehicle speed calculation results on the system has a standard deviation of 2.82 km / hour and uncertainty  $\pm 1$  km / hour. While the value of MSE produced is 12,632. When multiple vehicle objects are close together at less than 1 meter in the same frame, the system will calibrate the vehicle in the same ROI area resulting in the same speed.

Keywords: Vehicle Speed Detection, Frame Difference, Video Processing

## 1. Pendahuluan

Pada zaman ini, teknologi berkembang begitu pesat. Semakin banyak penemuan-penemuan baru dan pengembangan teknologi yang sudah ada sebelumnya dimana memberikan berbagai dampak positif dan membantu mengatasi permasalahan yang sering dialami manusia. Dalam kasus ini, pemanfaatan teknologi digunakan untuk menindak pengendara motor yang melanggar aturan dengan berkendara melebihi batas kecepatan yang telah ditentukan dilingkungan kompleks perumahan. Pengendara yang tidak disiplin dapat membahayakan warga kompleks perumahan maupun dirinya sendiri. Oleh karena itu, dibutuhkan suatu sistem yang berfungsi untuk mendeteksi kecepatan kendaraan yang melintas dilingkungan kompleks perumahan. Dengan sistem ini, kecepatan kendaraan yang melebihi batas maksimum dapat di deteksi. Para pelanggar dapat diberikan sanksi yang tegas sehingga menimbulkan efek jera. Dengan begitu, pengendara akan lebih disiplin dalam berkendara sehingga menciptakan lingkungan kompleks perumahan yang aman dan nyaman.

Dalam sistem ini dibutuhkan sebuah kamera, sehingga dapat memantau aktivitas kendaraan yang melintas dilingkungan kompleks perumahan dan deteksi gerakan yang lainnya. Teknik untuk mendeteksi objek bergerak ini dikenal dengan nama sistem deteksi gerakan. Sistem deteksi gerakan merupakan suatu sistem yang mampu mendeteksi gerakan yang terjadi di dalam video [1]. Deteksi gerakan merupakan subjek penting dalam bidang *computer vision* yang digunakan oleh banyak sistem pada aplikasi video pengawas, monitoring trafik, kompresi video, perhitungan kecepatan dsb. Objek yang bergerak dalam rekaman video dapat diketahui kecepatannya dengan menggunakan suatu teknik tertentu. Objek bergerak tersebut dapat dilakukan perhitungan kecepatannya dengan cara menggunakan suatu urutan *frame* video. Teknik ini bekerja dengan cara membandingkan antar *frame* gambar untuk mendapatkan informasi yang menentukan adanya gerakan atau tidak. Metode yang digunakan untuk pembandingan ini adalah metode *frame difference*.

Metode *frame difference* merupakan bentuk pengurangan latarbelakang yang sederhana. Proses metode ini hanya mengurangi *frame* yang sekarang dengan *frame* sebelumnya sepanjang jumlah urutan *frame* video. Jika nilai mutlak piksel dari hasil pengurangan ini lebih besar dari *threshold* yang ditentukan maka akan dipertimbangkan sebagai latar depan atau objek bergerak [2].

## 2. Dasar Teori

### 2.1 Computer Vision

Computer vision merupakan suatu ilmu di bidang komputer yang dapat membuat mesin atau robot untuk ‘melihat’. Kata ‘melihat’ disini berarti mesin atau robot dapat bekerja seperti memiliki panca indera layaknya manusia. Terdapat beberapa klasifikasi dari vision itu sendiri, yaitu Low Level Vision, Medium Level Vision, dan High Level Vision [3]. Low Level Vision meliputi Sensing, yaitu pengambilan input berupa gambar, dan Preprocessing, yaitu memperoleh suatu gambar sebelum di proses. Medium Level Vision meliputi proses Segmentation, Description, Recognition. Segmentation adalah proses pemisahan gambar digital ke dalam beberapa region. Description merupakan proses mendeskripsikan suatu gambar, sedangkan Recognition merupakan pengenalan terhadap suatu gambar. Pada level yang lebih tinggi (High Level Vision) terdapat proses Interpretation, dimana Interpretation merupakan suatu kemampuan untuk memperkirakan bentuk asli dari gambar yang didapat, hal ini dapat dilakukan cara mendapatkan berbagai informasi yang diperlukan pada gambar tersebut. Maka proses Interpretation memerlukan deteksi, identifikasi, dan pengukuran dari fitur-fitur pada suatu gambar.

### 2.2 OpenCV

*OpenCV (Open Computer Vision)* adalah perangkat lunak yang berbasis *open source*. *OpenCV* dirancang untuk efisiensi komputasi dan berfokus penuh pada aplikasi yang berjalan secara *real time*. *OpenCV* ditulis menggunakan bahasa pemrograman C, bahasa pemrograman *python* dan bahasa pemrograman lainnya yang mendukung *library* pada *OpenCV* dan dapat mengambil keuntungan dari *processor multicore*. Salah satu tujuan dari *OpenCV* adalah untuk menyediakan infrastruktur *computer vision* sederhana yang digunakan yang membantu orang membangun aplikasi *vision* yang canggih dan cepat [4].

*OpenCV (Open Computer Vision)* merupakan sebuah API (Application Programming Interface) library yang sudah sangat familiar pada pengolahan citra *computer vision*. *Computer vision* itu sendiri adalah salah satu cabang dari bidang ilmu pengolahan citra (*image processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa implementasi dari *Computer Vision* adalah Face Recognition, Face Detection, Face/Object Tracking,

Road Tracking, dll. OpenCV adalah library Open Source untuk Computer Vision untuk C/C++, OpenCV didesain untuk aplikasi real-time, memiliki fungsi-fungsi akuisisi yang baik untuk image/video [6].

### 2.3 Definisi Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. Menurut arti secara harfiah citra (image) adalah gambar pada bidang dua dimensi [10].

Ditinjau dari sudut pandang matematik, citra merupakan fungsi menerus (continue) dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, kemudian objek memantulkan kembali sebagian dari berkas cahaya, pantulan cahaya ini ditangkap oleh alat-alat optik, seperti mata pada manusia, kamera, pemindai (scanner), dan lain-lain sehingga bayangan objek dalam bentuk citra dapat terekam. Citra sebagai output dari suatu sistem perekaman data dapat bersifat optik berupa foto, analog berupa sinyal video seperti gambar pada monitor televisi, digital yang dapat langsung disimpan pada suatu pita magnetik.

Citra dapat dikelompokkan menjadi dua bagian yaitu citra diam (*still image*) dan citra bergerak (*moving image*). Citra diam adalah citra tunggal yang tidak bergerak sedangkan citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun (*sekuensial*) sehingga memberi kesan pada mata sebagai gambar yang bergerak. Setiap citra di dalam rangkaian itu disebut *frame*. Gambar-gambar yang tampak pada film layar lebar atau televisi pada hakikatnya terdiri dari ratusan sampai ribuan *frame* [11].

### 2.4 Segmentasi

Tahapan segmentasi dilakukan agar bagian-bagian citra dikelompokkan menjadi bagian-bagian pokok yang mengandung informasi yang dibutuhkan saat menganalisa. Segmentasi gambar pada video bertujuan untuk mendeteksi daerah-daerah yang berhubungan dengan objek bergerak dengan lingkungannya atau *background*. Diantara teknik-teknik segmentasi untuk memecah suatu citra kedalam beberapa segmen dengan suatu kriteria tertentu, maka salah satu metode yang dapat digunakan yaitu metode *frame difference*.

#### a. Frame Difference

Banyak metode yang dapat digunakan untuk mendeteksi obyek yang bergerak, salah satunya adalah metode frame difference. Metode frame difference merupakan bentuk pengurangan latar belakang yang sederhana. Proses metode ini hanya mengurangi frame yang sekarang dengan frame sebelumnya sepanjang jumlah urutan frame video. Jika nilai mutlak piksel dari hasil pengurangan ini lebih besar dari threshold yang ditentukan maka akan dipertimbangkan sebagai latar depan atau objek bergerak. Secara matematis, metode frame difference dapat dijelaskan sebagai berikut:

$$D(x, y, t+1) = \begin{cases} 1 & |f(x, y, t) - f(x, y, t+1)| > Th \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

Dimana  $f(x, y, t)$  adalah frame video pada saat waktu  $t$ . Sedangkan  $f(x, y, t+1)$  adalah frame berikutnya pada saat waktu  $t+1$ . Nilai  $Th$  merepresentasikan nilai threshold untuk pengambilan keputusan. Jika pada frame video terjadi perubahan nilai yang melebihi nilai threshold, maka dapat dikatakan sebagai foreground (obyek bergerak). Sebaliknya, jika ternyata pada frame video nilai yang berubah dibawah threshold maka dapat dikatakan sebagai background.

#### b. Segmentasi Citra Video

Segmentasi citra video merupakan suatu proses pengelompokkan citra menjadi beberapa *region* berdasarkan kriteria tertentu. Berdasarkan pengertiannya, segmentasi memiliki tujuan menemukan karakteristik khusus yang dimiliki suatu citra. Oleh karena itu, segmentasi sangat diperlukan pada proses pengenalan pola. Semakin baik kualitas segmentasi maka semakin baik pula kualitas pengenalan polanya dari suatu objek ditentukan.

Secara umum ada beberapa pendekatan yang banyak digunakan dalam proses segmentasi antara lain:

1. Teknik threshold, yaitu pengelompokkan citra sesuai dengan distribusi properti piksel penyusun citra.
2. Teknik region-based, yaitu pengelompokkan citra ke dalam region tertentu secara langsung berdasar persamaan karakteristik suatu areacitranya.

3. Edge-based methods, yaitu pengelompokkan citra ke dalam wilayahberbeda yang terpisahkan karena adanya perbedaan perubahan warna tepidan warna dasar citra yang mendadak.

Pendekatan pertama dan kedua merupakan contoh kategori pemisahan image berdasarkan kemiripan area citra sedangkan pendekatan ketiga merupakan salah satu contoh pemisahan daerah berdasarkan perubahan intensitas yang cepat terhadap suatu daerah.

### c. Treshold

Threshold merupakan teknik yang sederhana dan efektif untuk segmentasi citra dengan melakukan pengolahan piksel pada suatu citra atau menghilangkan beberapa piksel dan juga mempertahankan beberapa piksel sehingga menghasilkan suatu citra baru hasil sortir piksel yang telah dilakukan. Dengan dilakukannya komputasi threshold maka dapat dengan mudah mendapatkan edge atau tepi dari suatu citra.

Threshold dilakukan agar mempermudah dalam proses identifikasi ataupun perbandingan dari dua atau lebih citra. Terdapat 5 tipe threshold, yaitu binary threshold, binary inverted, truncate, threshold to zero dan threshold to zero inverted. Binary threshold merupakan citra yang telah melalui proses pemisahan piksel-piksel berdasarkan derajat keabuan yang dimiliki. Pembentukan citra biner memerlukan nilai batas keabuan yang akan digunakan sebagai nilai patokan. Piksel dengan derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 sebagai warna putih dan sebaliknya piksel dengan derajat keabuan lebih kecil dari nilai batas akan diberi nilai 0 sebagai warna hitam. Warna putih biasanya digunakan untuk warna foreground sedangkan warna hitam adalah warna background. Maka dapat dilihat dalam persamaan berikut

$$x_{bw} = \begin{cases} 1 & \text{jika } x \geq \bar{x} \\ 0 & \text{jika } x < \bar{x} \end{cases} \quad (2.2)$$

## 2.6. Feature Extraction

Feature Extraction adalah adalah proses pengambilan ciri-ciri yang unik dari data yang akan diolah, yang bertujuan untuk memperkecil jumlah data mengambil informasi yang terpenting dari data yang diolah dan mempertinggi presisi pengolahan data. Dalam metode bounding box gambar di visualisasikan melalui dimensi objek yang dihasilkan dari gambar sebelumnya yaitu dengan menggunakan hasil output frame difference. Namun, terlebih dahulu melakukan proses pengambilan ekstraksi ciri benda yang akan diolah untuk menentukan setiap koordinat dari suatu objek menggunakan feature extraction agar proses bounding box dapat dilakukan. Scanning dilakukan untuk mencari batas-batas nilai edge berupa nilai non zero disetiap frame yang dimulai dari baris atas sampai baris bawah setiapframe, kolom kiri sampai dengan kolom kanan setiap frame.

Dalam hal ini manfaat bounding digunakan untuk menetapkan objek box yang diidentifikasi agar lebih jelas, dimana feature extraction menggambarkan nilai intensitas warna piksel tersebut. Nilai-nilai intensitas terdiri dari nilai atas, bawah, kiri, dan kanan disetiap frame menggunakan dimensi nilai kotak pembatas persegi panjang yang diplot dalam batas-batas nilai yang dihasilkan. Kekuranagn dari penggunaan bounding box, yaitu penggunaan pemakaian memori yang tidak sedikit. Penggunaan bounding box pada laju frame terdapat pada gambar berikut.

## 2.7. Pengukuran Kecepatan

Untuk mendapatkan hasil kecepatan objek bergerak dapat dihitung dengan mencari total jarak yang ditempuh dari awal objek bergerak sampai akhir objek bergerak dibagi dengan waktu yang ditempuh dalam melakukan perpindahan, dapat diketahui dengan menggunakan persamaan berikut.

$$V = \frac{D}{T} \quad (2.3)$$

Dimana:

V (*velocity*) = kecepatan yang dicapai dalam melakukan perpindahan.

D (*distance*) = jarak yang ditempuh.

T (*time*) = waktu yang ditempuh dalam melakukan perpindahan.

Selanjutnya berdasarkan persamaan kecepatan diatas, untuk mendapat hasil perhitungan kecepatan laju kendaraan dari hasil rekaman video, nilai jarak didapatkan berdasarkan jarak referensi yang ditempuh dari awal objek bergerak sampai akhir objek bergerak yang telah ditentukan dan untuk waktu diperoleh dari jumlah frame yang diperlukan saat objek melakukan perpindahan dengan nilai fps dari video tersebut. Maka dapat dilihat pada persamaan sebagai berikut.

$$Kecepatan = \frac{Jarak\ Euclidean \times Jarak\ 1\ px \times fps \times 3600}{Jumlah\ Frame \times 1000} \text{ km/jam} \tag{2.4}$$

Dimana:

Jarak Euclidean = jarak yang sudah ditentukan dalam penelitian.

Jarak 1 pixel = banyaknya frame saat perekaman video.

Jumlah frame = kemampuan kamera video menampilkan gambar dalam satuan per detik.

Proses mencari objek bergerak dalam urutan frame yang dikenal sebagai pelacakan. Pelacakan ini dapat dilakukan dengan menggunakan ekstraksi ciri benda dan mendeteksi objek atau benda bergerak diurutan frame. Dengan menggunakan nilai posisi objek disetiap frame kita dapat menghitung posisi dan kecepatan objek bergerak. Jarak yang ditempuh oleh objek ditentukan dengan menggunakan titik pusat dari bounding box tersebut. Jarak yang dihitung dengan menggunakan rumus jarak euclidean. Rumus jarak euclidean pada dua dimensi merupakan perpindahan sebuah simbol antara satu titik ke titik lainnya pada sumbu X dan sumbu Y [17].

$$D = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{2.5}$$

Dimana:

D = jarak yang ditempuh

X<sub>1</sub> = posisi titik 1 pada sumbu X

X<sub>2</sub> = posisi titik 2 pada sumbu X

Y<sub>1</sub> = posisi titik 1 pada sumbu Y

Y<sub>2</sub> = posisi titik 2 pada sumbu Y

### 3. Pembahasan

#### 3.1 Deskripsi Sistem

Penelitian ini membahas mengenai analisis dan perancangan untuk membuat aplikasi deteksi kecepatan kendaraan termasuk dari cara kerja sistem, perancangan program dan spesifikasi perangkat lunak untuk menjalankan sistem secara keseluruhan. Untuk mendapatkan kecepatan kendaraan secara umum sistem ini dibagi menjadi dua bagian utama, yaitu motion detection dan speed detection. Motion detection berfungsi untuk mendeteksi adanya pergerakan objek dalam video. Sedangkan speed detection berfungsi untuk menentukan kecepatan objek dalam hal ini kendaraan yang di tangkap oleh web kamera secara real time. Tahapan proses bagaimana sistem bekerja dapat digambarkan sebagai berikut:



Gambar - 3.1 Blok Diagram Sistem

Pada blok input, pertama kali sistem mengambil inputan dari lintasan jalan komplek perumahan menggunakan kamera webcam secara *real time* sehingga menghasilkan beberapa frame video. Setelah frame video didapatkan, dilakukan analisis pada frame tersebut untuk mendeteksi adanya pergerakan objek. Hal tersebut merupakan proses pada blok *motion detection*. Ketika terdeteksi adanya pergerakan dalam frame video dalam hal ini adalah kendaraan bermotor, maka proses *speed detection* dapat dilakaukan. Pada tahap ini, sistem melakukan perhitungan kecepatan kendaraan berdasarkan pergerakan objek kendaraan dalam frame video. *Output* dari seluruh proses adalah dapat diketahuinya kecepatan kendaraan bermotor yang melintas pada area pengamatan.

### 3.2 Pengujian

Pengujian sistem dilakukan untuk mengetahui performansi sistem sehingga dapat diketahui kelebihan dan kekurangan sistem yang telah dibangun. Pada sistem ini, pengujian dilakukan dengan menganalisa nilai ketidakpastian kecepatan kendaraan yang didapat dari program. Kecepatan pada program dibandingkan dengan perhitungan kecepatan yang dihasilkan oleh speedometer.

Nilai ketidakpastian didapatkan dari perbedaan hasil perhitungan kecepatan pada program dengan hasil perhitungan kecepatan yang berada pada speedometer. Perbedaan hasil kecepatan tersebut dikurangi dengan rata-rata perbedaan kecepatan yang dihasilkan, lalu dibagi dengan jumlah banyaknya hasil percobaan dikurangi satu. Persamaan untuk menentukan standar deviasi adalah sebagai berikut:

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \tilde{x})^2}{n - 1}} \quad (3.1)$$

Dimana:

$s$  = standar deviasi

$x_i$  = perbedaan nilai hasil perhitungan kecepatan antara program dan speedometer

$\tilde{x}$  = rata-rata dari perbedaan nilai perhitungan kecepatan antara program dan speedometer

$n$  = jumlah data

Setelah nilai standar deviasi diperoleh, langkah selanjutnya adalah melakukan perhitungan nilai ketidakpastian.

Persamaan untuk menghitung nilai ketidakpastian adalah sebagai berikut:

$$u(\tilde{x}) = \frac{s}{\sqrt{n}} \quad (3.2)$$

Dimana:

$u(\tilde{x})$  = Nilai ketidakpastian

$s$  = standar deviasi

$n$  = jumlah data

Untuk pengujian performansi sistem, digunakan juga Mean Square Error(MSE). MSE dalam statistik merupakan sebuah nilai yang diharapkan dari kuadrat error. *Error* yang ada menunjukkan seberapa besar perbedaan hasil estimasi dengan nilai yang akan diestimasi. Perbedaan itu terjadi karena adanya keacakan pada data atau karena estimator tidak mengandung informasi yang dapat menghasilkan estimasi yang lebih akurat.

$$MSE = \frac{1}{N} \sum_{t=h}^N (y_t - \hat{y}_t)^2 \quad (3.3)$$

Dimana:

MSE = *Mean Squared Error*

$N$  = Jumlah Sampel

$y_t$  = Nilai Aktual Indeks

$\hat{y}_t$  = Nilai Prediksi Indeks

#### 3.2.1 Pengujian terhadap pengaturan posisi sudut kamera

Pengujian terhadap penempatan posisi sudut kamera diperlukan untuk menemukan posisi sudut kamera yang tepat untuk sistem sehingga proses perhitungan kecepatan kendaraan lebih akurat. Hal ini dikarenakan setiap posisi sudut kamera yang berbeda menghasilkan perhitungan kecepatan yang berbeda pula pada sistem. Pengujian dilakukan dengan mengatur sudut tegak lurus antara tiang dengan tanah dan dilakukan beberapa kali sampai menemukan posisi sudut kamera yang tepat sehingga pengukuran kecepatan kendaraan pada sistem mendekati dengan pengukuran kecepatan yang terlihat pada speedometer kendaraan.

**Tabel 3.1** Pengujian terhadap posisi sudut kamera

Sudut	Percobaan	Kecepatan pada program (km/jam)	Kecepatan pada speedometer (km/jam)	Perbedaan	MSE
60°	1	25	23,119	-1,881	3,435
	2	25	24,940	-0,060	
	3	30	33,078	3,078	
	4	30	33,218	3,218	
	5	20	20,961	0,961	
	6	25	27,811	2,811	
	7	30	33,299	3,299	
	8	35	36,242	1,242	
	9	30	31,890	1,890	
	10	25	24,742	-0,258	
	11	35	36,094	1,094	
	12	25	26,287	1,287	
	13	20	20,142	0,142	
	14	35	36,291	1,291	
	15	40	39,883	-0,117	
	16	35	37,241	2,241	
	17	40	41,028	1,028	
	18	40	40,914	0,914	
	19	20	22,248	2,248	
	20	25	26,981	1,981	

Berdasarkan hasil pengujian yang telah dilakukan, sudut kamera yang paling akurat untuk perhitungan kecepatan kendaraan adalah menggunakan sudut sebesar 60°. Hal ini dibuktikan dengan nilai MSE pada sudut 60° adalah nilai terkecil diantara nilai MSE pengaturan sudut yang lainnya. Ketika sudut kamera diatur dengan posisi sudut sebesar 60°, *counting line* pada sistem sejajar dengan tanah. Berbeda dengan pengaturan posisi sudut kamera selain pada sudut 60°, *counting line* pada frame video berada pada posisi mengambang tidak sejajar dengan tanah. Hal tersebut mempengaruhi kesalahan perhitungan panjang lintasan, sehingga mempengaruhi perhitungan keakuratan kecepatan kendaraan yang dideteksi oleh sistem.

**3.2.2 Pengujian terhadap akurasi perhitungan kecepatan kendaraan**

Setelah menemukan posisi sudut kamera yang tepat berdasarkan pengujian dari berbagai sudut, diperlukan pengujian lebih lanjut untuk membuktikan keakuratan perhitungan kecepatan pada sistem. Pengujian ini dilakukan dengan membandingkan perhitungan kecepatan yang dihasilkan oleh sistem dengan perhitungan kecepatan yang terdapat pada speedometer kendaraan. Pengujian kecepatan dilakukan pada saat kecepatan kendaraan berada pada kecepatan dibawah 30 km/jam dan diatas 30 km/jam.

Berdasarkan hasil percobaan, didapatkan kecepatan kendaraan yang berasal dari program maupun dari speedometer sebagai perbandingan. Hasil dari percobaan tersebut dapat di analisis menggunakan standar deviasi dan nilai ketidakpastian. Nilai standar deviasi yang didapat dari percobaan tersebut adalah 2,82 km/jam. Sedangkan nilai ketidakpastian yang didapat adalah ± 1 km/jam. Disamping itu, untuk membandingkan nilai error hasil perhitungan kecepatan digunakan MSE dan nilai MSE yang didapat adalah sebesar 12,632.

**Tabel 3.2** Pengujian terhadap kecepatan dibawah 30 km/jam

Percobaan	Kecepatan pada program (km/jam)	Kecepatan pada speedometer (km/jam)	Perbedaan	MSE
1	25,641	25	0,641	4,416
2	22,727	24	-1,273	
3	17,857	20	-2,143	
4	19,389	22	-2,611	
5	19,403	22	-2,597	
6	20,979	21	-0,021	
7	25,690	24	1,690	
8	15,513	16	-0,487	
9	18,592	20	-1,408	
10	20,253	25	-4,747	
11	11,880	13	-1,120	
12	20,629	22	-1,371	
13	23,059	25	-1,941	
14	22,497	20	2,497	
15	27,173	25	2,173	
16	20,600	25	-4,400	
17	18,907	18	0,907	
18	19,522	21	-1,478	
19	25,237	25	0,237	
20	26,483	26	0,483	

**Tabel 3.3** Pengujian terhadap kecepatan diatas 30 km/jam

Percobaan	Kecepatan pada program (km/jam)	Kecepatan pada speedometer (km/jam)	Perbedaan	MSE
1	40,294	44	3,706	20,848
2	36,645	40	-3,355	
3	29,786	32	-2,214	
4	36,857	40	-3,143	
5	43,694	40	3,694	
6	36,951	42	-5,049	
7	37,006	40	-2,994	
8	32,062	40	-7,938	
9	32,872	37	-4,128	
10	34,662	40	-5,338	
11	39,861	36	3,861	
12	41,315	44	-2,685	
13	32,462	35	-2,538	
14	29,062	35	-5,938	
15	35,256	39	-3,744	
16	33,807	40	-6,193	
17	30,016	35	-4,984	
18	32,483	38	-5,517	
19	38,994	45	-6,006	
20	36,395	40	-3,605	

Deteksi kecepatan kendaraan pada sistem lebih akurat pada saat kecepatan kendaraan dibawah 30 km/jam dibandingkan pada saat kecepatan kendaraan diatas 30 km/jam. Hal ini bisa terjadi dikarenakan deteksi kecepatan kendaraan ketika berada diatas 30 km/jam jumlah sample frame video yang diambil oleh sistem lebih sedikit dibandingkan jumlah sample frame video pada saat kecepatan kendaraan dibawah 30 km/jam. Dengan pengambilan jumlah sample frame video yang cukup, perhitungan kecepatan kendaraan akan semakin akurat.

## 4. Kesimpulan dan Saran

### 4.1. Kesimpulan

Berdasarkan hasil uji coba sistem pada penelitian ini, didapatkan beberapa kesimpulan sebagai berikut:

1. Dari penelitian yang sudah dilakukan, posisi sudut kamera mempengaruhi tingkat akurasi perhitungan kecepatan. Sudut paling optimal untuk mendeteksi kecepatan kendaraan adalah sebesar  $60^\circ$ .
2. Berdasarkan hasil pengujian, nilai standar deviasi yang didapat adalah sebesar 2,82 km/jam dan nilai ketidakpastian  $\pm 1$  km/jam. Sedangkan untuk nilai MSE, nilai yang didapat adalah sebesar 12,632.
3. Perhitungan kecepatan dibawah batas kecepatan maksimum (30 km/jam) lebih baik dibandingkan perhitungan kecepatan diatas batas kecepatan maksimum.
4. Keadaan lokasi pengamatan mempengaruhi kinerja sistem. Seperti adanya pohon, bangunan, orang yang sedang berjalan, mobil yang sedang parkir, dan lain-lain. Semakin banyak objek yang diam pada frame video, kinerja sistem akan semakin baik.

### 4.2. Saran

Saran untuk penelitian selanjutnya:

1. Untuk penelitian selanjutnya, pengujian sistem dapat dilakukan pada malam hari dengan penambahan sumber cahaya atau menggunakan teknologi kamera infra merah.
2. Pengembangan algoritma pada sistem agar dapat mendeteksi kecepatan kendaraan pada setiap kendaraan walaupun terdapat beberapa kendaraan dalam frame yang sama.
3. Penambahan konsep komputasi paralel pada sistem agar kinerja sistem untuk pengolahan citra lebih cepat dan lebih baik.

## DAFTAR PUSTAKA

- [1] M. Ihsan Zul, Widyawan dan L. Edi Nugroho, "Deteksi Gerak dengan Menggunakan Metode Frame Differences pada IP Camera," *International Conference on Information Technology and Electrical Engineering*, 2012.
- [2] N. Singla, "Motion Detection Based on Frame Difference Method," *International Journal of Information & Computation Technology*, vol. 4, no. 15, pp. 1559-1565, 2014.
- [3] A. B. Tucker, *Computer Science Handbook*, Massachusetts: CRC Press, 2004.
- [4] G. R. Bradski, *Learning OpenCV*, New Delhi: Shroff Publishers & Distributors, 2008.
- [5] A. Bovik, *Handbook of Image and Video Processing*, Texas: Elsevier Academic Press, 2005.
- [6] A. Murat Tekalp, *Digital Video Processing*, New Jersey: Prentice Hall Press, 2010.
- [7] T. Sutoyo dan dkk, *Teori Pengolahan Citra Digital*, vol. 5 (17), Yogyakarta: Penerbit Andi, 2009.
- [8] K. Dwi Irianto, "Pendeteksi Gerak Berbasis Kamera Menggunakan OpenCV pada Ruangan," vol. 2, no. 1, 2010.
- [9] A. Setiawan dan M. Hariadi, "Vehicle Detection Based Lane Masking and Background Elimination using K-means Clustering," 2012.
- [10] D. Putra, *Pengolahan Citra Digital*, Yogyakarta: Penerbit Andi, 2010.
- [11] T. Sutoyo dan Mulyanto, *Teori Pengolahan Citra Digital*, Yogyakarta: Penerbit Andi, 2009.
- [12] H. Pratomo, *Pengantar Model Warna RGB*, Surabaya: STIKOM Surabaya, 2012.
- [13] A. Wikan, *Pengolahan Citra Digital*, Yogyakarta: Universitas Kristen Duta Wacana, 2007.
- [14] D. Stalin Alex dan D. Amitabh Wahi, "Background Subtraction Frame Difference Algorithm for Moving Object Detection and Extraction," *Journal of Theoretical and Applied Information Technology*, vol. 60, no. 3, pp. 1992-8645, 2014.
- [15] R. Gonzales dan R. Woods, *Digital Image Processing Second Edition*, New Jersey: Prentice Hall Press, 2002.
- [16] J. J. Physics Through Applications, Oxford University Press, 1989.
- [17] L. Peddireddi, "Object Tracking and Velocity Determination using TMS320C6416T DSK," 2008.
- [18] P. Hartoto, "Sistem Deteksi Kecepatan Kendaraan Bermotor pada Real Time Traffic Information," 2011.
- [19] AIPCR, *Road Network Operations Handbook*, 2003.