

ANALISIS PERHITUNGAN KECEPATAN MOBIL PADA VIDEO TERCUPLIK DENGAN COMPRESSIVE SENSING

ANALYSIS OF CAR SPEED CALCULATIONS ON VIDEO CLIPS WITH COMPRESSIVE SENSING

Diflah Salsasanaura¹, Gelar Budiman², Indrarini Dyah Irawati³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹dsalsasanr@student.telkomuniversity.ac.id, ²gelarbudiman@telkomuniversity.ac.id,

³indrarini@telkomuniversity.ac.id

Abstrak

Frame difference adalah metode deteksi objek bergerak yang dilakukan dengan membandingkan *frame-frame* citra yang ditangkap sesuai dengan urutan waktu. Dengan *Frame Difference*, dapat dibuat sistem pendeteksi gerakan dalam video yang diaplikasikan di suatu sistem. *Compressive sensing* merupakan teknik dalam proses sampling data dan kompresi data. Tujuan suatu data dikompresi agar dapat memperkecil ukuran file data namun tetap menjaga kualitas setelah proses rekonstruksi data kompresi. Pada Tugas Akhir ini dirancang sistem perhitungan kecepatan mobil pada suatu video yang dikompresi dengan metode *compressive sensing* dengan kemudian dihitung kecepatannya metode deteksi gerak *frame difference*. Sistem yang telah dirancang menghasilkan keluaran kecepatan mobil, PSNR dan rasio kompresi. Data uji untuk sistem pada Tugas Akhir ini berupa video dengan kecepatan mobil yang berbeda. Parameter performansi sistem yaitu PSNR, rasio kompresi dan akurasi sistem. Sistem diuji dengan variabel pengujian berupa perubahan nilai *threshold* dan perubahan nilai *L* (ukuran matriks pada *compressive sensing*). Dari hasil pengujian ini, sistem dengan *compressive sensing* sudah mendapatkan hasil performansi akurasi terbaik sebesar 99.1 % dengan PSNR sebesar 33.35 dB, sedangkan akurasi rata-rata untuk sistem terbaik sebesar 98.4 % pada *threshold* sebesar 25 hingga 35 untuk sistem yang menggunakan *compressive sensing* dan 25 hingga 40 pada sistem tanpa *compressive sensing*.

Kata Kunci : Kecepatan mobil, *Compressive Sensing*, *Frame Difference*.

Abstract

Frame difference is a method of detecting moving objects which is done by comparing the captured image frames according to the time sequence. With *Frame Difference*, a motion detection system in video can be created that is applied to a system. *Compressive sensing* is a technique in the process of data sampling and data compression. The purpose of a compressed data in order to reduce the size of the data file but still maintain the quality after the compression data reconstruction process. In this final project, a car speed calculation system is designed on a compressed video using the *compressive sensing* method and then the speed is calculated using the *frame difference* motion detection method. The system that has been designed produces the output of car speed, PSNR and compression ratio. The test data for the system in this Final Project is a video with different car speeds. The system performance parameters are PSNR, compression ratio and system accuracy. The system is tested with test variables in the form of changes in the *threshold* value and changes in the value of *L* (matrix size on *compressive sensing*). From the results of this test, the system with *compressive sensing* has obtained the best accuracy performance of 99.1% with a PSNR of 33.35 dB, while the average accuracy for the best system is 98.4% at a *threshold* of 25 to 35 for systems using *compressive sensing* and 25 to 35. 40 on systems without *compressive sensing*.

Keywords: Car speed, *Compressive Sensing*, *Frame Difference*.

1. Pendahuluan

Saat ini teknologi informasi berkembang pesat di berbagai sektor, salah satunya dalam teknologi penginderaan yaitu *Intelligent Transport System* (ITS). ITS ini didasarkan pada penerapan teknologi canggih di bidang elektronika, komputer dan telekomunikasi untuk menjadikan infrastruktur transportasi lebih berbasis informasi, lancar, aman, nyaman dan ramah lingkungan [1]. Dengan ITS kita dapat meningkatkan monitoring namun kita dapat

mengetahui data berupa kecepatan kendaraan, banyak kendaraan pada lokasi tersebut, jenis kendaraan dan masih banyak lagi. Adanya monitoring diharapkan dapat mendeteksi permasalahan lalu lintas dan dapat dijadikan referensi untuk menanggulangi masalah tersebut.

Untuk mendeteksi kecepatan kendaraan bergerak banyak metode yang digunakan sebelumnya, Jinhui Lan dkk. pada penelitian yang berjudul *Vehicle Speed Measurement Based On Gray Constraint*

Optical Flow Algorithm mengusulkan bahwa penggunaan metode algoritma *optical flow* berdasarkan tiga tingkatan *frame* yang berbeda lebih untuk mengukur kecepatan kendaraan [2]. Lalu Faris Chandra Febrianto, dkk. melakukan penelitian perhitungan kecepatan kendaraan dengan metode *Frame Difference* Berbasis *Raspberry Pi* [3]. Kemudian Migliore dkk. pada penelitiannya yang berjudul *A Revaluation Of Frame Difference In Fast And Robust Motion Detection*, mengusulkan sebuah pengkajian dengan pendekatan *frame difference* dengan memanfaatkan algoritma pengurangan *foreground* yang memungkinkan untuk melakukan klasifikasi latar depan piksel tanpa perlu meninjau latar belakang sebelumnya [4].

Video digital cenderung menggunakan bandwidth saat pemrosesan, sehingga dilakukan proses kompresi untuk menghemat bandwidth tersebut, salah satu metode kompresi adalah *Compressive Sensing* atau CS. Menurut Candes dkk. [5], dan Donoho [6] sinyal dalam pengukurannya ini memiliki batas bisa dipulihkan lagi memakai metode yang diusulkannya dan dengan syarat sinyal itu harus ada pada situasi *sparse*. CS bertujuan dalam memberikan prediksi sinyal aslinya dari bagian kecil ukur linier yang tidak koheren terhadap menggunakan sparsitasnya [7] [8]. Dengan CS ini adalah kita dapat memakai lebih sedikit sampelnya terhadap pengukurannya, tapi memiliki kemampuan mendapatkan kembali sinyalnya dengan baik bahkan lebih baik lagi lalu data yang dikompres tidak membutuhkan ruang penyimpanan yang besar. Pada teori CS bukan hanya menyusun rencana mengakuisisi sinyalnya tapi dilaksanakan peluang pemulihan sinyalnya itu memakai algoritmanya berbeda-beda [9]. CS membagi pendekatan tersebut atas dua bagian yakni *convex optimization* seperti basis pursuitnya serta algoritma *greedy* seperti *matching pursuit* [10]. Saat menggunakan CS memerlukan transformasi karena transformasi sangat penting dilakukan pada CS dalam mendapatkan sinyal jarang (*sparsity*) sebagai syarat pada CS. Metode transformasi yang umumnya dikenal adalah *Discrete Cosine Transform* (DCT), *Discrete Sine Transform* (DST), *Discrete Wavelet Transform* (DWT), *Fast Fourier Transform* (FFT), dan masih banyak lagi.

Berdasarkan hal tersebut, penulis membuat Tugas Akhir mengenai perhitungan kecepatan mobil pada data berupa video dengan memanfaatkan metode *compressive sensing* sebagai metode kompresi video dan mendeteksi mobil yang bergerak menggunakan metode *frame difference*. Untuk rekonstruksi CS akan

digunakan *Orthogonal Matching Pursuit* (OMP) dan *Discrete Cosine Transform* (DCT) sebagai transformasinya. Pada tugas akhir ini bertujuan untuk menganalisis hasil perhitungan tanpa CS dan saat menggunakan CS dengan faktor perubahan nilai *threshold* dan nilai baris kompresi pada CS.

2. Studi Terkait

2.1 Citra Digital

Citra digital didefinisikan sebagai citra kontinu yang diubah dalam bentuk citra diskrit melalui proses digitalisasi berupa *sampling* dan *kuantisasi* (*quantization*). Pada proses *sampling* citra akan dibagi menjadi piksel, lalu pada proses *kuantisasi* tiap piksel akan diberi nilai berupa bilangan bulat agar dapat diketahui nilai intensitas warnanya [11]. Citra digital dituliskan sebagai fungsi intensitas cahayanya pada wujud dua dimensi yang terbentuk dari $f(x, y)$. Jika citra dua dimensi kontinu $f(x, y)$ dilakukan pembagian dengan N sebagai baris serta M sebagai kolom, sehingga baris beserta kolomnya yang berpotongan itu dikenal dengan sebutan piksel. Citra digital bisa dituliskan melalui wujud matriks seperti yang ada di Persamaan 1 berikut [11] :

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & \dots & \dots & f(1, M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix} \quad (1)$$

Dari penulisan matriks Persamaan 1 diatas dapat dilihat bahwa x dan y sebagai koordinat spasial (bidang), serta *amplitude* f terhadap koordinat (x, y) dikenal dengan atau informasi warna citra. Koordinat (x, y) menandakan tempat piksel di citra tertentu, sementara *amplitude* $f(x, y)$ menandakan angka intensitas warna citra [11]. Terdapat 3 jenis citra digital berdasarkan komponen warna penyusunnya yakni citra berwarna, citra *grayscale*, serta citra biner.

2.2 Video Digital

Video digital adalah jenis file yang didalamnya terdapat berkas digital berupa video yang dapat diputar menggunakan suatu perangkat. Setiap *frame* pada video digital sebetulnya mencakup rangkaian gambar atau citra digitalnya yang ditunjukkan secara cepat di percepatan yang stabil. Kecepatan pembacaan suatu *frame* yang dihitung disebut *frame* rata-rata (*frame rate*) dengan satuan yang ditampilkan adalah *frame per second* (fps) [12]. Setiap *frame* merupakan gambar digital yang terdiri dari raster pixel. Gambar digital akan memiliki lebar sebanyak W piksel dan tinggi sebanyak H piksel. Suatu video digital mencakup *frame-frame* yang akan dilakukan pengompresan

menuju file komputer yang bisa ditampilkan dari software multimedia player [12]. Mengacu pada wujud kompresannya itu, timbul berbagai format video digital yang tersedia pada para penggunanya yang memiliki keunggulan serta kelemahannya. Pada video digital terdapat beragam karakteristiknya yang dapat menentukan kualitas video digital, diantaranya adalah resolusi, kedalaman bit dan *frame rate*.

2.3 Frame Difference

Frame difference adalah metode yang pada prinsipnya membandingkan nilai piksel citra [13]. *Frame Difference* bekerja melalui pendekatan deteksi benda bergerak pada video yang berasal dari kamera. Dengan pendekatan ini dapat melakukan deteksi ada tidaknya objek bergerak dari perbedaan nilai piksel antar *frame* saat ini dengan *frame* background ketika jalan kosong (tanpa kendaraan) ini dilakukan untuk mengetahui objek pada *foreground* dan objek pada *background*. Proses segmentasi gerak adalah langkah awal pendeteksian kendaraan pada rangkaian *frame* yang bergerak agar nilai piksel tiap kordinat dapat diketahui [14]. Proses dari metode ini hanya mengurangi *frame* saat ini dari *frame* sebelumnya di sepanjang nomor urut dari *frame* video. Jika nilai absolut dari piksel yang dihasilkan dari pengurangan ini lebih besar dari ambang batas (*threshold*) yang ditentukan maka hasil tersebut diperlakukan sebagai suatu latar depan yang artinya objek tersebut mengalami pergerakan [13]. Untuk perumusannya, metode *frame difference* menggunakan rumus yang ditunjukkan pada Persamaan 2 sebagai berikut:

$$D(x, y, t) = \begin{cases} 1, & |f(x, y, t) - f(x, y, t - 1)| > T \\ 0, & \text{untuk lainnya} \end{cases} \quad (2)$$

2.4 Thresholding

Thresholding adalah metode yang digunakan dalam pemisahan citra dengan cara pemisahan objek pada citra dengan latar belakang yang ada didalamnya. Metode *thresholding* citra dapat digunakan untuk memisahkan antara objek dan *background*. Jika suatu *threshold* pada algoritma *frame difference* memiliki nilai yang rendah, maka banyaknya noise yang dapat dideteksi akan semakin tinggi. Teknik ini diproses dengan cara mengolah tiap piksel suatu citra dan bisa juga dengan mengeliminasi sebagian kecil piksel serta dengan menahan sejumlah piksel agar dapat memperoleh citra hasil penyortiran piksel yang telah dilalui [15]. Dengan menentukan *threshold* maka

dapat dengan mudah mendapatkan *edge* suatu citra. Jika piksel yang terdapat pada frame video memiliki nilai dibawah *threshold*, maka akan dikatakan sebagai *background* karena menandakan objek tersebut tidak bergerak dan begitu pula sebaliknya.

2.5 Compressive Sensing

Compressive sensing merupakan suatu teknik pengelolaan sinyal dan penginderaan baru dimana pengambilan akusisi dan kompresi dilakukan secara bersamaan, sehingga memberikan dua keuntungan yaitu data yang lebih kecil dan proses yang efisien. Pada *Compressive Sensing* sinyal dapat direkonstruksi dari jumlah pencuplikan yang jauh lebih sedikit daripada yang disyaratkan oleh *Shannon-Nyquist* [16]. Proses *Compressive Sensing* dapat dilakukan dengan perangkat keras maupun perangkat lunak dengan melalui dua tahapan. Tahap pertama yaitu tahap penginderaan yang dilakukan agar jumlah cuplikan lebih sedikit sehingga ukuran data lebih kecil. Setelah melalui tahap penginderaan akan ada tahap rekonstruksi yang dilakukan untuk memperoleh sinyal yang semula [16]. Saat *minimum sparsity level* (*k*) bernilai satu, maka sinyal akan lebih efektif jika menggunakan *compressive sensing*. Pemrosesan pada *compressive sensing* ini dilakukan dengan perkalian matriks menggunakan Persamaan 3 berikut [17] :

$$y = Ax \quad (3)$$

Dimana *y* dapat dicari dengan Persamaan 4 :

$$y = (A \times \psi') \times b \quad (4)$$

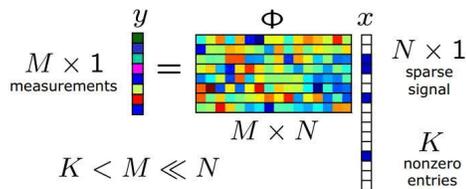
Dan untuk nilai *b* dan *x* diketahui dengan Persamaan 5 dan 6 berikut :

$$b = \psi x \quad (5)$$

dan

$$x = \psi' \times b \quad (6)$$

Disini *b* sebagai basis transformasi, *x* adalah sinyal asli, sebagai ψ metode transformasi yang digunakan dan ψ' sebagai metode invers transformasi yang digunakan. Proses pengukuran pada *compressive sensing* akan dijelaskan pada Gambar 1 berikut :



Gambar 1 Proses Pengukuran *Compressive Sensing* [17]. Pada Gambar 1 terdapat matriks vektor $M \times 1$ yang diasumsikan sebagai *y*. Matriks *y* merupakan hasil

kompresi perkalian antara matriks $A = M \times N$ dengan matriks $x = N \times 1$, dengan syarat $K < M \ll N$. Pada matriks y hanya terdapat sebagian informasi dari data aslinya, hal ini dikarenakan matriks x mengalami kompresi oleh A sehingga y hanya berisi sebagian kecil dari informasi x , namun informasi y sudah merepresentasikan informasi pada sinyal x [17]. Karena pada proses *Compressive Sensing* terdapat syarat $K < M \ll N$ artinya nilai rasio yang dihasilkan akan lebih baik saat bernilai < 1 atau $< 100\%$.

Pada Persamaan 3 terdapat matriks A yang dapat dicari pada Persamaan 7 ini :

$$A = L \times B^2 \tag{7}$$

Dengan Persamaan 7, pada Persamaan 3 x dan y dapat diubah menjadi bentuk Persamaan 8 dan 9 dibawah ini :

$$x = B^2 \times 1 \tag{8}$$

dengan

$$y = L \times 1 \tag{9}$$

Sehingga dapat disebut kompresi apabila $L < B$. Dari hal tersebut dapat diketahui bahwa untuk mengetahui rasio kompresi data berupa citra dapat dirumuskan dengan Persamaan 10 berikut :

$$\text{Rasio Kompresi} = \frac{L}{B^2} \times 100\% \tag{10}$$

Dengan L nilai dari baris kompresi dan B merupakan *block based* atau segmentasi pada citra.

Pada proses *compressive sensing* terdapat akuisisi sebagai fungsi pengambilan data *sparse* lalu data yang dikompresi. Proses kompresi dan sampling terjadi pada satu blok yang sama, yaitu dinamakan *block project*. Hasil sampling ini akan dikirim ke penerima dan setelah diterima, sinyal akan dilakukan rekonstruksi sehingga menghasilkan *output* yang akan dibandingkan dengan sinyal aslinya.

2.6 Orthogonal Matching Pursuit (OMP)

OMP merupakan algoritma rekonstruksi yang digunakan pada *compressive sensing*. Algoritma ini bersifat *greedy* dan merupakan versi pembaruan algoritma *Matching Pursuit* (MP). Besaran dengan koefisien non-zero dari x dapat diprediksi menggunakan algoritma OMP. Hal ini dilakukan dengan penyelesaian kuadrat terkecil pada proyeksi *orthogonal* dari x yang telah dikembalikan lalu dilanjut dengan menghitung vektor y [18]. *Orthogonal Matching Pursuit-Partially Known Support* merupakan *greedy algorithm* untuk rekonstruksi citra *sparse* yang dikembangkan berdasarkan algoritma

OMP konvensional. Berdasarkan Gambar 1 apabila A sebagai matriks random *gaussian* pada CS, artinya A merepresentasikan baris dari suatu matriks sensing. Metode rekonstruksinya yaitu saat nilai k mengisi hampir seluruh sinyal x yang ada pada Gambar 1 Dengan k bernilai *non-zero* saat memenuhi $K < M \ll N$. Matriks x dapat direkonstruksi menggunakan OMP jika matriks A memenuhi *Restricted Isometry Property* (RIP). RIP adalah properti fundamental dari matriks yang memungkinkan pemulihan *sparse* [19]. Matriks A dikatakan sesuai RIP apabila orde k terdapat $\delta_k \in 0.1$ seperti pada Persamaan 11 dibawah ini :

$$(1 - \delta_k) \|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k) \|x\|_2^2 \tag{11}$$

Dengan δ_k adalah *Restricted Isometry Property* (RIP), δ_k sebagai *Restricted Isometry Constant* (RIC), A sebagai matriks random *gaussian* dan x sebagai sinyal asli. Jika A memenuhi RIP dengan orde $2k$ dengan $\delta_k < \sqrt{2} - 1$, k mendistribusi vektor x sehingga x dapat direkonstruksi menjadi Persamaan 12 :

$$\hat{x} = \arg \min_x \|y - Ax\|_2 \tag{12}$$

Dimana \hat{x} adalah sinyal setelah rekonstruksi dan y adalah hasil perhitungan kompresi.

2.7 Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) merupakan transformasi pada metode kompresi yang dapat mengkonversi sinyal pada bentuk elemen frekuensi dasar. Pendekatan pada transformasi ini ditandai dengan tingginya korelasi antar piksel. DCT dilakukan dengan tujuan untuk mengkonversi citra dari kawasan spasial ke kawasan frekuensi begitu pula sebaliknya yang dapat digunakan dengan *invers* DCT sehingga dapat meminimalisir korelasi antar piksel tetangga pada proyeksi citra dalam frekuensi menjadi gelombang cosinus. Adapun DCT dapat dirumuskan dengan Persamaan 13 berikut [20] :

$$x(k) = e(k) \sum_{n=0}^{N-1} x(n) \cos \left[\frac{(2n+1)\pi k}{2N} \right] \tag{13}$$

Invers Discrete Cosine Transform (IDCT) merupakan fungsi invers dari *Discrete Cosine Transform* (DCT) yang berfungsi untuk menginvers nilai matriks setelah terjadi transformasi. Masukannya berupa matriks citra hasil transformasi dan output berupa citra dan nilai matriks yang telah direkonstruksi. Secara matematis dirumuskan dengan Persamaan 14 [20]:

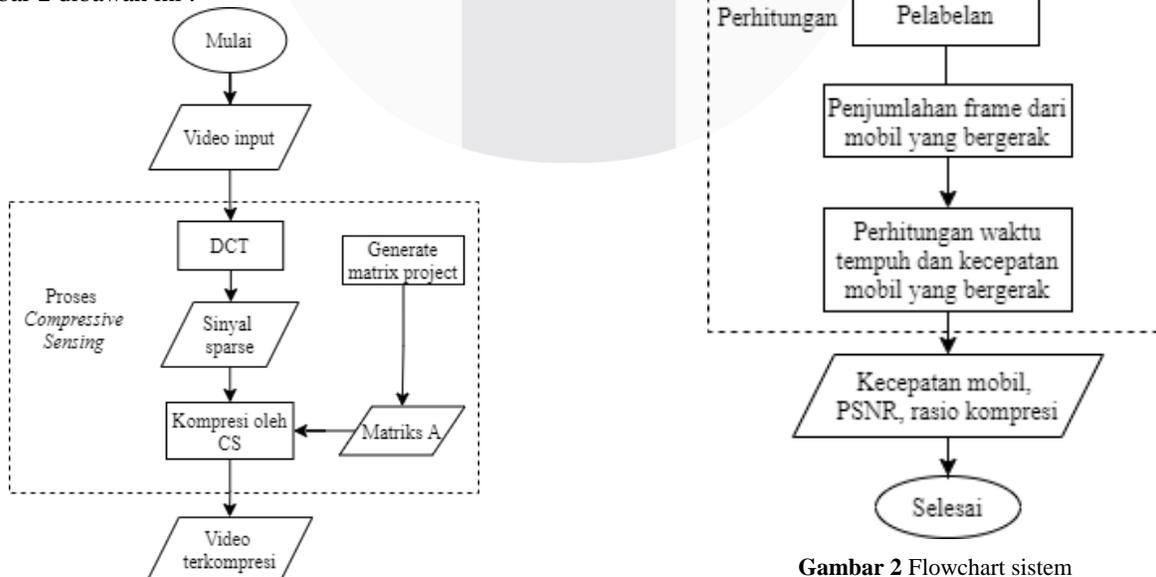
$$X(n) = \frac{2}{N} \sum_{k=0}^{N-1} e(k)x(k) \cos \left[\frac{(2n+1)\pi k}{2N} \right] \quad (14)$$

Dimana $e(k)$ didefinisikan sebagai :

$$e(k) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{jika } k = 0 \\ 1, & \text{jika } k \text{ lebih dari } 0 \end{cases}$$

3. Model dan Perancangan Sistem

Pada Tugas Akhir ini sistem dibuat untuk mendeteksi objek yang bergerak dalam urutan *frame* video yang akan dikompresi pada sistem sebelum akhirnya terdeteksi berapa kecepatan objek tersebut. Proses awal yang dilakukan dengan memberi video berformat .avi sebagai masukan pada sistem lalu akan dilakukan kompresi menggunakan metode *compressive sensing* dengan tujuan agar ukuran data dapat dikompres lebih kecil dari data aslinya. Pada tahapan *compressive sensing* ini data akan direkonstruksi menggunakan algoritma *orthogonal matching pursuit* dengan tujuan untuk merekonstruksi sinyal hasil pengukuran *compressive sensing*. Setelah video dikompresi akan dilanjut dengan proses *pre-processing* yang bertujuan untuk mengkonversi video masukan yang berupa video RGB menjadi video *grayscale* agar lebih mudah untuk mempermudah proses pada tahap selanjutnya. Hasil dari *pre-processing* selanjutnya akan menuju tahapan deteksi objek bergerak pada sistem. Metode *frame difference* akan digunakan dalam proses ini, setelah objek berhasil dideteksi sistem, akan ada keluaran berupa kecepatan objek pada video, PSNR, dan rasio kompresi. Alur sistem akan digambarkan pada Gambar 2 dibawah ini :



Gambar 2 Flowchart sistem

Desain sistem ini dirancang untuk menghitung kecepatan mobil yang bergerak pada video dengan metode deteksi objek bergerak yaitu *frame difference* dan metode *compressive sensing* sebagai metode kompresi. Pada Tugas Akhir ini ada tiga parameter kinerja yang dianalisis untuk menilai kinerja sistem diantaranya adalah *Peak Signal to Noise Ratio* (PSNR), rasio kompresi dan akurasi.

4. Hasil dan Pembahasan

Pada pengujian sistem hasil dari perhitungan kecepatan pada mobil yang bergerak akan dianalisis dengan melihat variabel pengujian dan parameter performansinya yaitu PSNR, rasio kompresi dan lamanya waktu proses per satu *framenya*. Sistem diuji menggunakan aplikasi pemrograman MATLAB R2019b dengan menggunakan metode kompresi yaitu *compressive sensing* dan metode deteksi gerak menggunakan *frame difference*. Dilakukan 3 skenario pada Tugas Akhir ini.

4.1 Skenario 1: Pengujian Terhadap Perubahan Nilai Threshold dengan CS.

Pada pengujian ini menggunakan *compressive sensing*. Berikut data hasil pengujian pada skenario 1 yang tertera pada Tabel 1, 2, 3 dan 4:

Tabel 1. Nilai *Threshold* pada Kecepatan dengan CS

Th	Kecepatan mobil pada sistem dengan CS				
	40 km/jam	55 km/jam	60 km/jam	70 km/jam	80 km/jam
5	0	41.10	30.69	31.19	0
15	0	54.08	59.46	70.71	80.89
35	41.13	54.08	59.46	70.71	80.89
35	41.13	54.08	59.46	70.71	80.89
40	57.13	54.08	59.46	70.71	80.89
45	0	57.08	59.46	70.71	80.89
50	0	0	59.46	70.71	80.89
55	0	0	95.14	70.71	80.89

Pada Tabel 1 perubahan nilai *threshold* dapat menyebabkan berubahnya kecepatan pada sistem di beberapa nilai *threshold*. Kecepatan terkecil didapat saat *threshold* bernilai 5, sedangkan ketika *threshold* bernilai diatas 35 terjadi beberapa perubahan kecepatan yang semakin besar di beberapa sampel. Tabel 2 merupakan hasil dari perubahan *threshold* terhadap PSNR :

Tabel 2 Perubahan *Threshold* pada PSNR dengan CS

Th	PSNR (dB)				
	40 km/jam	55 km/jam	60 km/jam	70 km/jam	80 km/jam
5	0	32.75	33.30	34.49	0
15	30.38	32.45	32.91	34.62	35.67
25	30.31	32.51	33.06	34.19	35.66

Th	PSNR (db)				
	40 km/jam	55 km/jam	60 km/jam	70 km/jam	80 km/jam
35	30.75	32.70	33.35	34.76	35.51
40	30.48	32.23	33.43	33.96	35.59
45	0	32.95	33.61	34.45	35.70
50	0	0	33.25	34.11	35.65
55	0	0	32.91	34.76	35.18

Pada Tabel 2 menunjukkan bahwa perubahan *threshold* berpengaruh terhadap PSNR pada sistem yang menggunakan CS. Saat nilai *threshold* kurang dan lebih dari 35 maka nilai PSNR akan lebih kecil daripada saat *threshold* bernilai 35. Pada PSNR apabila nilai yang dihasilkan lebih besar maka lebih baik untuk digunakan.

Tabel 3 Threshold pada Akurasi Sistem dengan CS

Th	Akurasi (%)				
	40 km/jam	55 km/jam	60 km/jam	70 km/jam	80 km/jam
5	0	74.7	51.1	44.5	0
15	0	98.3	99.1	98.9	98.8
35	97.1	98.3	99.1	98.9	98.8
35	97.1	98.3	99.1	98.9	98.8
40	57.1	98.3	99.1	98.9	98.8
45	0	96.2	99.1	98.9	98.8
50	0	0	99.1	98.9	98.8
55	0	0	41.43	98.9	98.8

Tabel 4 Akurasi Rata-rata Sistem dengan CS

Th	Akurasi rata-rata dengan CS (%)
5	34.0
15	79.0
35	98.4
35	98.4
40	90.4
45	78.6
50	59.3
55	47.8

Pada Tugas Akhir ini, digunakan *threshold* pada rentang 5 hingga 55, dan akurasi terbaik jika dilihat dari hasil kecepatan, PSNR, dan akurasi sistem terdapat pada *threshold* yang bernilai 35 karena hasil perhitungannya lebih tinggi dibandingkan nilai yang lain. Akurasi tertinggi didapat pada kecepatan 60 km/jam dengan akurasi sebesar 99.1 % dan PSNR sebesar 33.35. Sedangkan untuk akurasi rata-rata tertinggi terdapat pada *threshold* 25-35 dengan akurasi rata-rata sebesar 98.4 %. Sehingga pada sistem akan ditentukan bahwa 35 sebagai nilai *threshold* sistem karena PSNR dan akurasinya lebih tinggi.

4.2 Skenario 2: Pengujian Terhadap Perubahan Nilai Baris Kompresi (L)

Pada skenario kedua dilakukan pengujian sistem dengan menggunakan *compressive sensing* yang

dilakukan dengan memasukan nilai L yang berbeda untuk mendapatkan hasil psnr, rasio kompresi dan akurasi. Berikut pengujian perubahan L pada lima sampel kecepatan yang berbeda. Hasilnya akan diperlihatkan pada Tabel 5:

Tabel 5 Perubahan nilai L pada Sistem dengan CS

V (km/jam)	B	L	V_s (km/jam)	PSNR (dB)	Rk (%)	Akurasi (%)
40	8	4	79.10	27.05	6.14	2.2
	8	8	79.10	28.40	12.29	2.2
	8	16	41.13	30.75	24.58	97.1
	8	20	41.13	30.71	30.72	97.1
	8	25	41.13	31.11	38.41	97.1
	8	30	41.13	31.75	46.09	97.1
	8	32	41.13	31.67	49.13	97.1
	55	8	4	102.75	27.79	6.13
8		8	85.62	30.17	12.26	44.3
8		16	54.08	32.70	24.52	98.3
8		20	54.08	33.54	30.66	98.3
8		25	54.08	33.85	38.32	98.3
8		30	54.08	33.81	45.99	98.3
8		32	54.08	34.15	49.05	98.3
60		8	4	95.14	28.62	6.12
	8	8	79.13	29.03	12.27	68.1
	8	16	59.46	33.35	24.5	99.1
	8	20	59.46	34.05	30.62	99.1
	8	25	59.46	34.35	38.28	99.1
	8	30	59.46	34.68	45.93	99.1
	8	32	59.46	34.69	49	99.1
	70	8	4	115.14	30.42	6.13
8		8	81.59	31.39	12.26	83.4
8		16	70.71	34.76	24.53	98.9
8		20	70.71	34.86	30.67	98.9
8		25	70.71	35.32	38.33	98.9
8		30	70.71	35.78	46.00	98.9
8		32	70.71	35.79	49.07	98.9
80		8	4	151.78	31.97	6.13
	8	8	97.07	32.83	12.27	78.6
	8	16	80.89	35.51	24.54	98.9
	8	20	80.89	35.97	30.68	98.9
	8	25	80.89	36.66	38.39	98.9
	8	30	80.89	36.95	46.02	98.9
	8	32	80.89	36.80	49.09	98.9

Pada Tabel 5 menunjukkan perubahan L berpengaruh terhadap perubahan rasio kompresi, sehingga saat rasio kompresi berubah, kecepatan sistem, PSNR dan akurasi sistem terpengaruh. Pada L yang bernilai rendah, rasio kompresi akan semakin rendah mengakibatkan hasil kecepatan sistem lebih tinggi daripada kecepatan aslinya, sehingga PSNR dan akurasi pada sistem semakin rendah. Jika dilihat dari hasil PSNR dan kecepatan yang didapat dari sistem, dapat dilihat bahwa saat L bernilai 16 memiliki hasil yang lebih bagus dibandingkan nilai lainnya. Karena

saat rasio kompresi lebih kecil, informasi yang tersimpan pada citra yang terkompresi masih terjaga dengan baik. Namun pada Tugas Akhir ini kita harus melihat lagi bagaimana hasil pengujian apabila menggunakan rasio kompresi yang lebih rendah. Jika menghasilkan kecepatan, PSNR dan akurasi yang lebih baik dibandingkan nilai L yang lebih tinggi maka L yang terkecil layak digunakan, namun apabila menghasilkan kecepatan, PSNR dan akurasi yang rendah maka akan dipilih L yang lain dengan hasil pengujian kecepatan yang mendekati kecepatan sebenarnya dan PSNR serta akurasi yang paling baik.

4.3 Skenario 3: Pengujian Terhadap Perubahan Threshold Tanpa Menggunakan Compressive Sensing.

Berikut hasil pengujian pada Tabel 6:

Tabel 6 Nilai Threshold pada Kecepatan tanpa CS

Th	Kecepatan mobil pada sistem dengan CS				
	40 km/jam	55 km/jam	60 km/jam	70 km/jam	80 km/jam
5	0	33.14	31.71	28.66	151.78
15	64.27	54.08	59.46	70.71	80.89
35	41.13	54.08	59.46	70.71	80.89
35	41.13	54.08	59.46	70.71	80.89
40	41.13	54.08	59.46	70.71	80.89
45	0	57.08	59.46	70.71	80.89
50	0	57.08	59.46	70.71	80.89
55	0	57.08	59.46	70.71	80.89

Pada Tabel 6 menunjukkan bahwa pada sistem tanpa menggunakan CS, *threshold* dapat berpengaruh terhadap kecepatan yang dihasilkan oleh sistem. Semakin kecil *threshold* maka semakin jauh ketepatan deteksi kecepatan pada sistem apabila dibandingkan dengan kecepatan sebenarnya. Namun, apabila *threshold* dinaikan pada beberapa sampel nilai kecepatan sistem semakin besar menjauhi kecepatan sebenarnya. Hasil pengujian data ditunjukkan pada Tabel 7:

Tabel 7. Nilai Threshold pada Akurasi tanpa CS

Th	Akurasi (%)				
	40 km/jam	55 km/jam	60 km/jam	70 km/jam	80 km/jam
5	0	60.2	47.1	40.9	10.2
15	39.3	98.3	99.1	98.9	98.8
35	97.1	98.3	99.1	98.9	98.8
35	97.1	98.3	99.1	98.9	98.8
40	97.1	98.3	99.1	98.9	98.8
45	0	96.2	99.1	98.9	98.8
50	0	96.2	99.1	98.9	98.8
55	0	96.2	99.1	98.9	98.8

Dari Tabel 7 dapat diketahui akurasi rata-rata pada tiap threshold. Untuk akurasi rata-rata akan ditunjukkan di

Tabel 8 berikut :

Tabel 4 Akurasi Rata-rata Sistem tanpa CS.

Th	Akurasi rata-rata tanpa CS (%)
5	31.6
15	86.8
35	98.4
35	98.4
40	98.4
45	78.6
50	78.6
55	78.6

Pada Tabel 7 dapat dilihat jika *threshold* berpengaruh pada akurasi sistem walaupun sistem tidak menggunakan CS. Saat *threshold* bernilai terlalu rendah atau terlalu tinggi maka hasil akurasi akan semakin kecil. Dan akurasi terbaik rata-rata saat *threshold* pada rentang nilai 25 hingga 40.

5. Kesimpulan

Berdasarkan hasil analisis terhadap pengujian sistem yang dilakukan untuk perhitungan kecepatan mobil menggunakan metode deteksi *frame difference* dan metode kompresi menggunakan *compressive sensing*, maka dapat disimpulkan bahwa pengujian sistem tanpa CS menghasilkan akurasi lebih optimal dibanding sistem saat menggunakan CS, hal ini karena saat menggunakan CS sinyal terkompresi sehingga dapat menghilangkan sedikit informasi. Saat *threshold* diubah terhadap parameter akurasi dengan, sistem mendapatkan akurasi tertinggi yaitu sebesar 99.1%, saat PSNR sebesar 33.35 dB, dan akurasi rata-rata tertinggi mencapai 98.4%. Berdasarkan pengujian terhadap nilai L yang diubah berpengaruh terhadap besar kecilnya rasio kompresi. Semakin kecil nilai L maka rasio kompresi akan semakin kecil yang artinya semakin baik, namun untuk melihat akurasi berdasarkan nilai L dapat dilihat dari rasio kompresi, PSNR dan akurasi kecepatan yang paling bagus. Pada tugas akhir ini akurasi terbaik yang didapat berdasarkan perubahan nilai L yaitu saat B bernilai 8 dan L bernilai 16.

Referensi

- [1] Pamudi, "Penerapan Sistem Dinamik dalam Sistem Transportasi Cerdas untuk Mengurangi Kemacetan, Polusi dan Meningkatkan Keselamatan Berjalan Lintas (Studi Kasus Dinas Perhubungan Kota Surabaya)," *Thesis*, pp. 17–18, 2018.
- [2] J. Lan, J. Li, G. Hu, B. Ran, and L. Wang, "Vehicle speed measurement based on gray

constraint optical flow algorithm," *Optik (Stuttg.)*, vol. 125, no. 1, pp. 289–295, 2014, doi: 10.1016/j.ijleo.2013.06.036.

- [3] F. C. Febrianto and F. Utamingrum, "Perhitungan Kecepatan Kendaraan Secara Otomatis Menggunakan Metode Frame Difference Berbasis Raspberry Pi," *J. Fak. Ilmu Komput.*, vol. 3, no. 12, pp. 10968–10974, 2019.
- [4] D. A. Migliore, M. Matteucci, and M. Naccari, "A revaluation of frame difference in fast and robust motion detection," *Proc. ACM Int. Multimed. Conf. Exhib.*, no. October, pp. 215–218, 2006, doi: 10.1145/1178782.1178815.
- [5] M. B. W. Candes, Emmanuel J., "An Introduction to Compressed Sensing," *Appl. Numer. Harmon. Anal.*, pp. 1–65, 2019, doi: 10.1007/978-3-319-73074-5_1.
- [6] David L. Donoho, "Compressed Sensing," *IEEE Trans. Inf. THEORY*, vol. 52, no. 1, 2006, doi: 10.1109/TAES.2017.2649698.
- [7] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressed Sensing*. 2013.
- [8] Y. C. Eldar and G. Kutyniok, "Compressed sensing: Theory and applications," *Compress. Sens. Theory Appl.*, no. October 2018, pp. 1–544, 2009, doi: 10.1017/CBO9780511794308.
- [9] Y. Zhang, Y. Huang, H. Li, P. Li, and X. Fan, "Conjugate gradient hard thresholding pursuit algorithm for sparse signal recovery," *Algorithms*, vol. 12, no. 2, pp. 1–15, 2019, doi: 10.3390/a12020036.
- [10] S. G. Mallat and Z. Zhang, "Matching Pursuits With Time-Frequency Dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993, doi: 10.1109/78.258082.
- [11] R. C. Gonzalez and R. E. Woods, *Digital Image Processing, 4e*, 4th ed. New York, 2018.
- [12] A. Murat Tekalp, *Digital Video Processing Second Edition*, Second Edi. 2015.
- [13] R. A. Yuha, M. Danu, A. Fiqri, and M. Harahap, "Deteksi Gerakan pada Kamera CCTV dengan Algoritma *Frame Difference* dan *Frame Substraction*," 2019.
- [14] S. A. Meshram, T. Engineering, M. E. Scholar, and M. E. Teacher, "Vehicle Detection and Tracking Techniques Used in Moving Vehicle," vol. 2, no. 7, pp. 48–59, 2015.

- [15] M. I. Zul, Widyawan, and L. E. Nugroho, "Deteksi Gerak dengan Menggunakan Metode *Frame Differences* pada IP Camera," *Proceeding CITEE 2012*, no. July 2015, pp. 52–56, 2012.
- [16] D. W. Wicaksono and B. Setiyono, "Speed Estimation On Moving Vehicle Based On Digital Image Processing," *Int. J. Comput. Sci. Appl. Math.*, vol. 3, no. 1, p. 21, 2017, doi: 10.12962/j24775401.v3i1.2117.
- [17] Kuku Yudhistiro, "Restorasi Citra Optical Character Recognition Dengan Algoritma Recurrent Hopfield," *Smatika Jurnal, April 2018*, vol. 8, 2018.
- [18] P. Rosin and T. Ellis, "Image Difference Threshold Strategies and Shadow Detection.," pp. 35.1-35.10, 2013, doi: 10.5244/c.9.35.J. Novakovic, A. Veljovi, S. Iiic, Z. Papic, and M. Tomovic, "Evaluation of Classification Models in Machine Learning," *Theory Appl. Math. Comput. Sci.*, vol. 7, no. 1, pp. 39–46, 2017.
- [19] M. Fornasier and H. Rauhut, "Compressive sensing," *Handb. Math. Methods Imaging Vol. 1, Second Ed.*, no. July, pp. 205–256, 2015, doi: 10.1007/978-1-4939-0790-8_6.
- [20] A. A. H. Karah Bash and S. K. Kayhan, "Watermarked Compressive Sensing Measurements Reconstructed by the Greedy Algorithms," *Int. J. Comput. Theory Eng.*, vol. 7, no. 3, pp. 219–222, 2015, doi: 10.7763/ijcte.2015.v7.960.