

# Estimasi Kecepatan Kendaraan Bergerak Menggunakan *Motion Vector*

Yudah Mulia Nanang Sukabhakti dan Dwi Ratna Sulistyaningrum  
Departemen Matematika, Fakultas Matematika, Komputasi dan Sains Data,  
Institut Teknologi Sepuluh Nopember (ITS)  
*e-mail*: dwiratna.math@gmail.com

**Abstract**—*Intelligent Transportation System (ITS)* adalah penerapan teknologi informasi dan komunikasi pada bidang transportasi. Teknologi dan aplikasi ITS memantau kondisi lalu lintas pada ruas jalan melalui video hasil rekaman kamera CCTV yang dipasang di ruas jalan. Informasi dari video lalu lintas kemudian diolah sehingga menghasilkan informasi terkait lalu lintas tersebut seperti banyak kendaraan dan kecepatan kendaraan. Informasi kecepatan kendaraan kemudian diestimasi dan dimanfaatkan nantinya untuk kebijakan pengembangan lalu lintas dan pengaturan lalu lintas. Pada Studi ini dilakukan estimasi kecepatan kendaraan bergerak menggunakan *motion vector*. Penelitian dilakukan dalam beberapa langkah yaitu akuisisi video dan pengaturan parameter, pembuatan *region of interest* pada video, ekstraksi citra *frame video*, *preprocessing* citra, menentukan *motion vector* dengan *three-step search block-matching algorithm*, penyeleksian *motion vector*, dan penghitungan kecepatan kendaraan. Dari uji coba yang sudah dilakukan, *motion vector* mampu melakukan estimasi kecepatan kendaraan bergerak dengan akurasi terburuk 92.6194 % pada kecepatan 20 Km/jam dan terbaik 96.5582 % pada kecepatan 30 Km/jam.

**Kata Kunci**—*Motion Vector*, Estimasi Kecepatan.

## I. PENDAHULUAN

TEKNOLOGI informasi dan komunikasi (TIK) berkembang dengan cepat dan luas pada banyak bidang seperti ekonomi, kesehatan, transportasi lain-lain. Contoh perkembangan TIK di bidang transportasi adalah *intelligence transportation system (ITS)*. ITS merupakan sistem di mana TIK diterapkan di bidang transportasi darat, termasuk infrastruktur, kendaraan dan pengguna, dan manajemen lalu lintas dan manajemen mobilitas, serta untuk antarmuka dengan moda transportasi lainnya. ITS dapat meningkatkan efisiensi transportasi dalam sejumlah situasi, yaitu transportasi jalan, manajemen lalu lintas, mobilitas, dan lain-lain. Teknologi penginderaan merupakan salah satu teknologi ITS dan dapat dilakukan dengan memanfaatkan rekaman hasil kamera *closed circuit television (CCTV)*. Teknologi penginderaan ini dapat memberikan data dan informasi terkait dengan kondisi lalu lintas yang diamati. Data dan informasi tersebut dapat berupa banyaknya jumlah, kepadatan, jenis-jenis hingga kecepatan kendaraan dan lain sebagainya. Data yang berasal dari rekaman hasil kamera CCTV dapat menjadi informasi yang berguna bagi ITS dalam pengembangan lalu lintas kedepannya.

Pengambilan data informasi dari rekaman hasil kamera CCTV dilakukan dengan bantuan TIK melalui komputer agar lebih cepat dan biayanya tidak besar jika dibanding dengan

pengamatan oleh manusia secara manual. Rekaman hasil kamera CCTV merupakan video digital, sehingga perlu dilakukan pengolahan video secara digital untuk mendapatkan data dan informasi yang diinginkan. Terdapat bermacam-macam pengolahan video digital yang dapat dilakukan dengan tujuan dan metodenya masing-masing. Untuk pengolahan terkait objek dalam video, dalam hal ini khususnya objek kendaraan bergerak, dapat dilakukan pengolahan video dengan tujuan pendeteksian kendaraan dilanjutkan dengan pelacakan pergerakan hingga estimasi kecepatan kendaraan dan sebagainya. Terdapat beberapa penelitian terdahulu mengenai pengolahan video untuk deteksi, pelacakan hingga estimasi kecepatan kendaraan bergerak beserta metodenya masing-masing. Metode yang pernah dilakukan contohnya *gray constraint optical flow* [1], *euclidean distance* [2], *diagonal hexadecimal pattern (DHP)* [3], *mean filter (CVS)* [4], *motion vector* [5] dan lain-lain.

Metode *motion vector (MV)* merupakan metode yang menggunakan vektor arah gerakan sekelompok piksel antar *frame* pada video digital. *MV* dikenal luas dalam bidang kompresi, atau pengecilan ukuran, pada video digital, terkait dengan salah satu metode kompresi yang handal, yaitu *motion compensation (MC)*, yang menggunakan *MV* sebagai dasarnya. Dikarenakan kehandalan metode kompresi *MC*, yang berdasarkan *MV*, mengecilkan ukuran video digital dengan tetap mempertahankan kualitas video, maka metode *MV* layak dipertimbangkan sebagai metode estimasi kecepatan kendaraan bergerak yang handal. Dengan pertimbangan tersebut maka pada Studi ini penulis melakukan penelitian tentang estimasi kecepatan kendaraan bergerak menggunakan *MV*.

## II. URAIAN PENELITIAN

### A. Penelitian Terdahulu

Terdapat beberapa penelitian sebelumnya yang berkaitan dengan estimasi kecepatan pada citra kendaraan bergerak dengan *motion vector*.

Danang Wahyu Wicaksono dan Budi Setyono dalam jurnalnya yang berjudul "*Speed Estimation On Moving Vehicle Based On Digital Image Processing*" [2] menyimpulkan bahwa estimasi kecepatan adalah komponen penting ITS yang dapat dilakukan dengan menggunakan teknik pengolahan citra. Menggunakan teknik pengolahan citra lebih efisien dan ekonomis daripada teknik konvensional seperti radar kecepatan atau pemeriksaan manual. Terdapat pengaruh antara estimasi kecepatan, sudut kamera dan pemilihan *region of interest (ROI)* saat menggunakan *euclidian distance*. Penelitian yang dilakukan mampu melakukan estimasi kecepatan kendaraan bergerak dengan akurasi terendah 87.01 % dan akurasi tertinggi 99.38 %.

Jinhui Lan dkk dalam jurnalnya yang berjudul “*Vehicle speed Measurement Based On Gray Constraint Optical Flow*” [1] menyimpulkan metode *improved three-frame difference* dan *gray constraint optical flow algorithm* dapat mendeteksi kontur kendaraan bergerak. Nilai *optical flow* kontur kendaraan, yang merupakan kecepatan kendaraan pada citra, dengan satuan piksel per detik, dihitung dengan rasio antara piksel citra dan panjang jalan, untuk menghitung kecepatan dari target bergerak, dengan satuan Km per jam. Pada video digital, daerah yang diminati akan dipilih dan hanya kecepatan kontur target bergerak di wilayah ini yang dihitung sehingga dapat mengurangi komputasi lebih lanjut. Hasil penelitiannya menunjukkan tingkat deteksi kecepatan mencapai 99 % namun dapat hasil dapat salah jika terdapat dua kendaraan yang berdekatan.

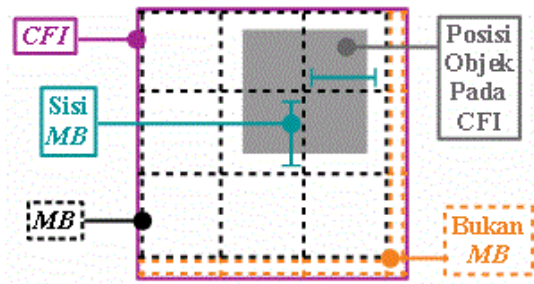
Arash G. R. dkk dalam jurnalnya yang berjudul “*Vehicle speed Detection in Video Image Sequences Using CVS Method*” [4] menyimpulkan ada tiga langkah untuk melakukan deteksi kendaraan dan pengenalan kecepatan, yaitu *background subtraction*, *object extraction* dan *speed recognition*. Pada langkah pertama, metode *mean filter* dilakukan untuk pembuatan latar belakang. Pada langkah kedua, dilakukan algoritme dengan memanfaatkan karakteristik dua warna dan menggabungkannya untuk ekstraksi objek. Pendekatan ini lebih kuat terhadap kesalahan dan masalah penggabungan atau pemisahan kendaraan. Pada langkah ketiga, kecepatan kendaraan ditentukan. Hasil penelitiannya menunjukkan rata-rata *error* estimasi kecepatan adalah 7 Km/jam.

**B. Motion Vector (MV)**

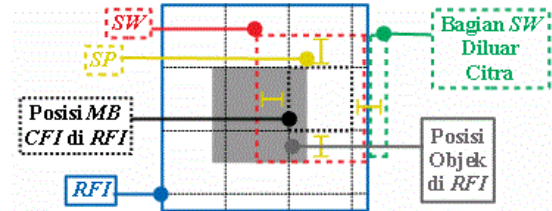
*Motion vector (MV)* adalah vektor dua dimensi yang digunakan untuk prediksi antar *frame* yang menunjukkan perpindahan, dari koordinat di citra *frame* yang dikodekan, ke koordinat di citra *frame* referensi [6]. Citra *frame* yang dikodekan adalah citra *frame* yang sedang dipilih untuk diprediksi dan dapat disebut *current frame image (CFI)*, sedangkan citra *frame* referensi adalah citra *frame* yang digunakan dalam proses prediksi dan dapat disebut *reference frame image (RFI)*.

Dengan menunjukkan perpindahan, *MV* mampu menggambarkan perpindahan objek pada *frame*. Performa *MV* yang handal diaplikasikan menjadi teknik kompresi video yaitu *motion compensation (MC)* yang, populer dan digunakan hampir pada semua standar pengkodean video seperti pada seri-seri *MPEG*. *MC* menghasilkan transformasi antar *frame* menggunakan *MV* yang mengurangi redundansi antar *frame* dan meningkatkan efisiensi kompresi video. *MV* dapat diperoleh melalui estimasi menggunakan *block-matching algorithm (BMA)*.

*Block-matching algorithm (BMA)* algoritme adalah pencarian lokasi suatu *macroblock (MB)* dengan ukuran tertentu yang paling sesuai pada *frame* lainnya berdasarkan suatu kriteria jarak pencarian [6]. *MB* adalah daerah *frame* yang dikodekan sebagai sebuah unit. *MB* biasanya adalah sekelompok piksel yang membentuk persegi dengan ukuran sisi persegi tertentu. *BMA* dapat menghasilkan *MV* pada rangkaian *frame* video dengan membagi *CFI* kedalam sejumlah *MB* seperti pada Error! Reference source not found. dan kemudian mencari daerah pada *RFI* yang memiliki kesesuaian terbaik dengan *MB CFI*.



Gambar 1. Pembagian CFI kedalam bentuk MB.

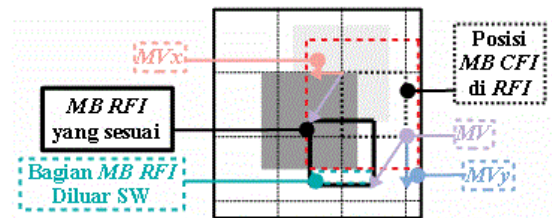


Gambar 2. SP dan SW pada RFI.

Pencarian daerah dengan kesesuaian terbaik dilakukan seperti pada **Error! Reference source not found.**, yaitu pada koordinat *MB CFI* di *RFI* dengan jarak pencarian maksimal dari koordinat *MB CFI* yang dapat disebut parameter pencarian atau *search parameter (SP)*. *SP* memiliki nilai dalam satuan piksel dan digunakan untuk menentukan besar jendela pencarian atau *search window (SW)*

$$SW = (2SP + 1)^2$$

dengan *SW* merupakan bidang dua dimensi dengan nilai koordinat vertikal dan horizontal dalam rentang  $-SP$  hingga  $SP$ .



Gambar 3. MB RFI yang sesuai dengan MB CFI dan MV-nya.

Jika *MB CFI* menemukan daerah dengan kesesuaian terbaik di *RFI*, maka dihasilkan *MV* yang akan memodelkan gerakan dari koordinat *MB CFI* relatif pada citra, ke koordinat daerah kesesuaiannya di *RFI* relatif pada citra, dengan nilai *MV* sebesar jarak antara kedua koordinat seperti ditampilkan pada **Error! Reference source not found.**

Penilaian tentang kesesuaian *MB* dan pasangan kesesuaiannya dilakukan berdasarkan sebuah fungsi biaya. Pada Studi ini digunakan fungsi biaya *mean absolut difference (MAD)*

$$MAD = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}|$$

dengan parameter *N* adalah ukuran sisi *MB*, dan  $C_{ij}$  dan  $R_{ij}$  adalah besar nilai keabuan piksel di koordinat *MB* (*i, j*) pada *CFI C* dan *RFI R*.

C. Three-Step Search (TSS)

MV dapat diperoleh melalui estimasi menggunakan BMA. BMA memiliki beberapa variasi berdasarkan cara mencari kesesuaian MB, seperti algoritme exhaustive search (ES) [5] dan algoritme three-step search (TSS) [7]. Pada awalnya, BMA menggunakan algoritme ES untuk mencari kesesuaian MB, yang membandingkan MB CFI dengan semua kemungkinan MB pada SW di RFI. Kemudian, TSS dikembangkan untuk meningkatkan performa ES. TSS mampu mengurangi komputasi algoritme ES dengan hanya mencari kesesuaian MB CFI pada beberapa posisi MB pada SW di RFI dibanding mencari ke seluruh bagian SW.

Algoritme TSS memiliki langkah-langkah sebagai berikut:

1. Mulai dengan koordinat pencarian di pusat bidang SW (0,0).
2. Tetapkan nilai parameter step size (SS). Sebagai contoh, SS biasanya diberi nilai 4 piksel untuk SP bernilai 7 piksel.
3. Kemudian lakukan penyesuaian di 8 lokasi disekitar koordinat (0,0) sejauh ±SS piksel vertikal dan horizontal, dan juga di lokasi (0,0).
4. Pilih lokasi dengan kesesuaian terbaik dari 8 lokasi penyesuaian sebelumnya dan juga lokasi awal (0,0).
5. Masukkan koordinat lokasi yang memiliki kesesuaian terbaik tersebut menggantikan nilai koordinat pencarian.
6. Ganti nilai SS dengan setengah dari nilainya. Ulang pencarian kesesuaian sampai nilai SS = 1.

D. Region of Interest (ROI)

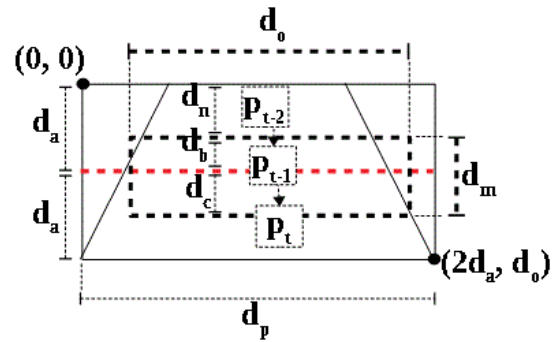
Region of Interest adalah bagian dari keseluruhan yang dipilih untuk dilakukan suatu operasi. Terdapat beragam tujuan penerapan ROI seperti mengurangi gangguan pada operasi dengan tidak melibatkan bagian yang terdapat gangguan pada pemilihan ROI, memperkecil ukuran data sehingga suatu operasi berjalan lebih cepat, dan lainnya. Pada Studi ini, ROI adalah satu bagian berbentuk segi empat dari keseluruhan luas bidang dari citra frame video untuk kemudian dilakukan operasi cropping atau pemotongan.

Langkah pertama pembuatan ROI yaitu membuat bentuk segi empat pada bidang citra frame video sebagai bakal ROI seperti pada **Error! Reference source not found.**. Nilai yang dapat dipakai dari segi empat ROI adalah jarak sisi atas ROI dengan sisi atas citra,  $d_n$ , tinggi ROI,  $d_m$ , jarak sisi kiri ROI dengan sisi kiri citra,  $d_q$ , dan lebar ROI,  $d_o$ . Nilai yang dapat dipakai dari bidang citra frame video adalah tinggi citra,  $2d_a$ , lebar citra,  $d_p$ , dan posisi tengah tinggi citra,  $d_a$ . Langkah kedua, menerapkan posisi ROI pada seluruh citra frame video. Langkah terakhir yaitu melakukan pemotongan atau cropping pada seluruh citra frame video pada posisi yang sudah diterapkan.

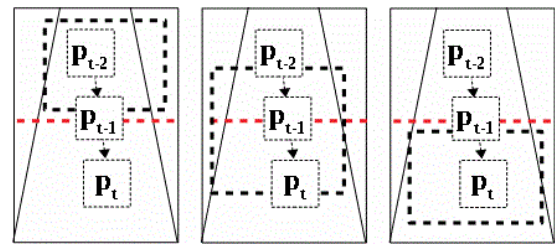
Karena efek perspektifitas tidak diabaikan, maka perekaman akan mendapati perbedaan ukuran gerakan sehingga gerakan kendaraan ketika lebih jauh akan lebih kecil daripada ketika kendaraan lebih dekat dengan perekam, padahal kendaraan diatur untuk berjalan dengan kecepatan konstan, sehingga diperlukan penyesuaian. Untuk itu, maka ROI dikembangkan dengan membagi menjadi tiga jenis berdasarkan posisi pada bidang citra seperti pada **Error! Reference source not found.** yaitu:

1. ROI dengan posisi di atas yang didefinisikan dengan jarak antara sisi citra bagian atas dengan sisi ROI bagian atas,  $d_n$ , dan jarak antara sisi citra bagian atas dengan sisi

- ROI bagian bawah,  $(d_n + d_m)$ , keduanya lebih kecil dari setengah panjang tinggi citra,  $d_a$ ,
- ROI dengan posisi di tengah yang didefinisikan dengan jarak antara sisi citra bagian atas dengan sisi ROI bagian atas,  $d_n$ , lebih kecil dari setengah panjang tinggi citra,  $d_a$ , dan jarak antara sisi citra bagian atas dengan sisi ROI bagian bawah,  $(d_n + d_m)$ , lebih besar dari setengah panjang tinggi citra,  $d_a$ , dan
- ROI dengan posisi di bawah yang didefinisikan dengan jarak antara sisi citra bagian atas dengan sisi ROI bagian atas,  $d_a$ , dan jarak antara sisi citra bagian atas dengan sisi ROI bagian bawah,  $(d_n + d_m)$ , keduanya lebih kecil dari setengah panjang tinggi citra,  $d_a$ .



Gambar 4. Ilustrasi Region of Interest.



Gambar 5. Ilustrasi Jenis ROI.

E. Konversi Panjang

Konversi panjang adalah nilai perbandingan antara panjang nyata dalam satuan meter panjang digital dalam satuan piksel pada video. Dari video tentang kendaraan tunggal bergerak, akan diperoleh gerakan pada bidang citra dalam satuan piksel yang mewakili dan dapat diubah menjadi gerakan pada bidang nyata dalam meter menggunakan nilai konversi panjang.

Untuk mendapatkan nilai konversi panjang, diperlukan pengaturan pada kamera dan kondisi perekaman seperti pada **Error! Reference source not found.** dan **Error! Reference source not found.**. Pengaturan kamera terdiri dari variabel sudut vertikal FoV,  $\theta_a$ , yang dapat diketahui dari spesifikasi kamera. Nilai sudut vertikal FoV,  $\theta_a$ , bisa didapatkan juga menggunakan nilai panjang fokus kamera,  $d_e$ , dan dimensi vertikal sensor citra kamera,  $2d_a$ , menggunakan persamaan

$$\theta_a = \tan^{-1} \left( \frac{d_a}{d_e} \right)$$

Jika nilai panjang fokus kamera dan dimensi vertikal sensor citra kamera tidak diketahui, secara geometri, nilai sudut vertikal FoV,  $\theta_a$ , bisa didapatkan juga dengan menggunakan nilai tinggi objek nyata yang tingginya terpantau penuh pada kamera,  $2d_g$ , dan nilai panjang jarak antara kamera dengan objek,  $d_f$ , menggunakan persamaan

$$\theta_a = \tan^{-1} \left( \frac{d_g}{d_f} \right)$$

Pengaturan kamera juga terdiri dari dimensi tinggi video hasil rekaman,  $2d_a$ . Pengaturan kondisi perekaman terdiri dari tinggi kamera,  $d_h$ , dan kemiringan kamera,  $\theta_a$ . Tinggi kamera,  $d_h$ , dihitung dari posisi lensa kamera ke posisi di dasar bawah kamera pada jalanan. Kemiringan kamera,  $\theta_a$ , dihitung dengan sudut  $0^\circ$  kamera menghadap vertikal ke bawah.

Cara kerja estimasi kecepatan dengan posisi kamera di atas adalah dengan mengukur gerakan objek kendaraan pada video,  $d_d$ , relatif terhadap ukuran salah satu bidang video, lalu mengkonversikan panjangnya menjadi panjang nyata,  $d_j$ , relatif terhadap ukuran bidang nyata teramati yang sesuai. Pertama, satuan nilai konversi dicari dengan menghitung perbandingan bidang pada video dengan bidang nyata yang bersesuaian dan teramati pada video. Jinhui Lan dkk [1] menggunakan perbandingan sisi horizontal video,  $d_p$ , terhadap lebar jalan yang teramati di sisi horizontal bawah video. Arash G. R. dkk [4] menggunakan perbandingan sisi vertikal video,  $2d_a$ , terhadap panjang jalan yang teramati,  $d_l$ , yang didapat dari

$$d_l = d_h \times (\tan(\theta_d + \theta_a) - \tan(\theta_d - \theta_a))$$

Pada Studi ini, nilai konversi panjang dikembangkan dengan menyesuaikan terhadap *ROI* agar menghasilkan nilai yang lebih akurat dibanding nilai konversi panjang yang umum untuk seluruh bagian citra video [1][4]. Nilai konversi panjang menggunakan perbandingan sisi vertikal *ROI* video,  $d_b + d_c$ , terhadap panjang jalan yang teramati *ROI*,  $d_k$ , yang didapat dari

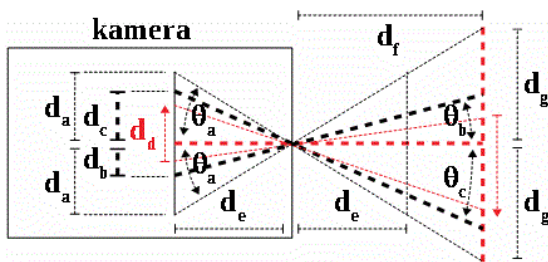
$$d_k = d_h \times (\tan(\theta_d + \theta_b) - \tan(\theta_d - \theta_c))$$

dengan

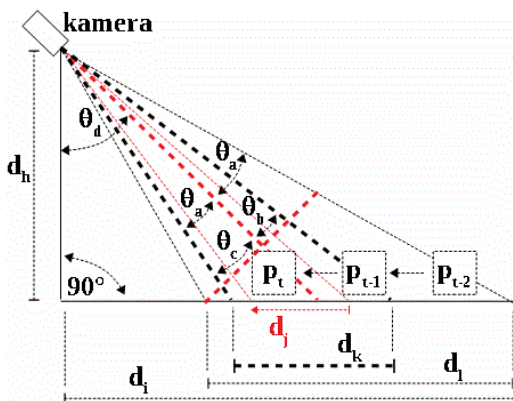
$$\theta_b = \tan^{-1} \left( \frac{d_b}{d_e} \right)$$

$$\theta_c = \tan^{-1} \left( \frac{d_c}{d_e} \right)$$

sementara  $d_b$  dan  $d_c$  dapat diperoleh dari posisi *ROI*.



Gambar 6. Pengaturan Kondisi Kamera.



Gambar 7. Pengaturan Kondisi Perekaman.

F. Estimasi Kecepatan

Estimasi kecepatan pada Studi ini dilakukan pada video kendaraan bergerak yang direkam dari sisi atas dengan langkah-langkah melakukan estimasi kecepatan adalah sebagai berikut:

1. Menentukan nilai konversi panjang, yang menggambarkan skala panjang nyata dalam meter untuk tiap panjang satu piksel pada video. Penentuan nilai konversi panjang dibantu menggunakan *ROI*.
2. Karena *MV MB* menggambarkan gerakan objek pada *MB* dalam sekali pergantian *frame*, dengan membatasi hanya merekam objek tunggal, *MV MB* yang dihasilkan untuk tiap *MB CFI* dapat dirata-ratakan dengan semua *MV MB* lainnya sehingga didapat nilai *MV frame*, yang menggambarkan gerakan objek tunggal pada *frame* dalam sekali pergantian *frame*.
3. Dengan mengasumsikan kecepatan kendaraan dan mengatur kecepatan kendaraan uji coba adalah konstan, *MV frame* dapat dirata-ratakan dengan semua *MV frame* lainnya, sehingga menghasilkan *MV video* yang menggambarkan gerakan objek tunggal pada *frame*, yang konstan untuk tiap *frame*, dalam sekali pergantian *frame*.
4. *MV video* dikalikan dengan nilai konversi panjang dan frekuensi *frame* video sehingga menghasilkan nilai estimasi kecepatan dalam satuan meter per detik yang dapat diubah kedalam kilometer per jam.

III. HASIL DAN PEMBAHASAN

A. Data, Parameter, dan Skema Uji Coba

Data uji coba program penelitian ini adalah beberapa video digital rekaman kendaraan bergerak. Kendaraan yang direkam bergerak dengan kecepatan dan arah yang konstan dan sudah ditentukan. Video yang digunakan memiliki variasi berdasarkan kecepatan kendaraannya seperti ditampilkan pada Tabel 1.

Tabel 1.

Data Video yang Digunakan.

No.	Nama Video (*.mp4)	Kecepatan Sebenarnya	Resolusi (Piksel × Piksel)
1	Pmd-20-60-4	20 Km/jam	540 × 960
2	Pmd-30-60-4	30 Km/jam	540 × 960
3	Pmd-40-60-4	40 Km/jam	540 × 960
4	Pmd-50-60-4	50 Km/jam	540 × 960
5	Pmd-60-60-4	60 Km/jam	540 × 960

Parameter uji coba program penelitian ini terdiri dari posisi *ROI*, ukuran sisi *MB*, sudut Seleksi, dan penggunaan *MF MV*. Variasi nilai parameter ditampilkan dalam Tabel 2.

Tabel 2.

Nilai Parameter yang Digunakan.

No.	Posisi <i>ROI</i>	Ukuran Sisi <i>MB</i>	Sudut Seleksi	<i>MF MV</i>
1	Atas	15 piksel	30°	Tidak
2	Bawah	25 piksel	60°	Ya
3	Tengah	35 piksel	90°	-
4	-	-	180°	-

Skema uji coba program penelitian ini adalah dengan melakukan estimasi kecepatan pada semua data dengan menggunakan seluruh kombinasi nilai parameter-parameter.



Hasil uji coba berupa nilai estimasi kecepatan dan akurasi estimasi kecepatan. Akurasi dihitung menggunakan

$$AE = \frac{KS - |KS - HE|}{KS} \times 100\%$$

dengan *AE* adalah Akurasi Estimasi, *KS* adalah Kecepatan Sebenarnya, dan *HE* adalah Hasil Estimasi.

Setelah dilakukan uji coba, diperoleh nilai parameter ukuran sisi *MB* terbaik adalah 35 piksel, sehingga untuk hasil uji coba akan dipakai nilai parameter ukuran sisi *MB* 35 piksel.

**B. Hasil dan Pembahasan Uji Coba**

Hasil uji coba estimasi kecepatan dirata-ratakan untuk tiap kombinasi nilai parameter dan didapatkan rata-rata akurasi hasil uji coba terbaik.

Tabel 3.  
Rata-rata Akurasi Hasil Uji Coba

No.	Posisi ROI	Sudut Seleksi	Rata-rata Akurasi Tanpa MF	Rata-rata Akurasi Dengan MF
1	Atas	30°	93.466 %	93.9621 %
2	Atas	60°	92.4297 %	92.5397 %
3	Atas	90°	92.1279 %	92.6187 %
4	Atas	180°	90.7844 %	89.0145 %
5	Bawah	30°	87.8193 %	86.171 %
6	Bawah	60°	86.4469 %	89.5207 %
7	Bawah	90°	84.7701 %	87.8294 %
8	Bawah	180°	78.6275 %	83.1176 %
9	Tengah	30°	84.2835 %	87.8705 %
10	Tengah	60°	87.1629 %	88.9939 %
11	Tengah	90°	86.9937 %	88.717 %
12	Tengah	180°	83.9141 %	82.8959 %

Tabel 4.  
Hasil Uji Coba Menggunakan Nilai Parameter Terbaik Secara Umum

Kecepatan	Estimasi	Akurasi
20 Km/jam	21.4761 Km/jam	92.6194 %
30 Km/jam	29.8617 Km/jam	96.5582 %
40 Km/jam	39.3041 Km/jam	93.9621 %
50 Km/jam	48.8663 Km/jam	94.3056 %
60 Km/jam	58.722 Km/jam	95.104 %

Rata-rata akurasi hasil uji coba pada Tabel 3 menunjukkan jika rata-rata akurasi terbaik yaitu 93.9621 % diperoleh dengan menggunakan nilai parameter terbaik secara umum yaitu posisi *ROI* di atas, sudut seleksi 30°,

ukuran sisi *MB* 35 piksel, dan menggunakan *MF MV*. Dengan menggunakan nilai parameter terbaik secara umum, dari Tabel 4 diperoleh akurasi terbaik dari variasi kecepatan yang digunakan yaitu 96.5582 %.

**IV. KESIMPULAN**

Dari uji coba yang sudah dilakukan didapatkan hasil bahwa estimasi kecepatan kendaraan bergerak mampu dilakukan menggunakan *motion vector*. Proses metode *motion vector* meliputi pemasukkan data dan parameter, termasuk penghitungan nilai konversi panjang, penentuan *motion vectors* serta seleksi arah dan implementasi *median filter* pada *motion vector*, dan penghitungan kecepatan untuk menghasilkan nilai estimasi kecepatan kendaraan dalam satuan Km/jam dan akurasinya.

Penilaian kinerja metode *motion vector* ditunjukkan oleh nilai akurasi estimasi kecepatan. Nilai rata-rata terbaik adalah 96.5582 % yang dicapai ketika dengan menggunakan posisi *ROI* di atas, sudut seleksi 30°, ukuran sisi *MB* 35 piksel, dan dengan menggunakan *MF MV*.

Penelitian kedepan diharapkan mengembangkan batasan seperti deteksi objek jamak, variasi cahaya dan variasi sudut perekaman, sistem program, model estimasi kecepatan, atau perbandingan dengan metode estimasi kecepatan lainnya.

**DAFTAR ISI**

[1] 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, Jan. 2014.

[2] 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, Mar. 2017.

[3] D. Jeyabharathi and D. D. Dejeey, "Vehicle Tracking and Speed Measurement system (VTSM) based on novel feature descriptor: Diagonal Hexadecimal Pattern (DHP)," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 816–830, Oct. 2016.

[4] A. Gholami Rad, A. Dehghani, and M. Rehan Karim, "Vehicle speed detection in video image sequences using CVS method," 2010.

[5] S. S. S. Ranjit, S. A. Anas, S. K. Subramaniam, K. C. Lim, A. F. I. Fayeez, and A. R. Amirah, "Real-Time Vehicle Speed Detection Algorithm using Motion Vector Technique," 2012.

[6] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*. Chichester, UK: John Wiley & Sons, Ltd, 2003.

[7] T. Koga *et al.*, "Motion-compensated interframe coding for video conferencing."