

# ANALISIS PERBANDINGAN PELACAKAN OBJEK MENGUNAKAN ALGORITMA HORN-SCHUNCK DAN LUCAS- KANADE

Wahyu Supriyatin

Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma, Depok, Indonesia

Corresponding Author: [ayu\\_ws@staff.gunadarma.ac.id](mailto:ayu_ws@staff.gunadarma.ac.id)

## Abstrak

Pelacakan objek merupakan pengembangan dari computer vision. Computer vision mirip dengan mata manusia memiliki fungsi sebagai penglihatan. Kesulitan untuk mendeteksi keberadaan suatu objek sehingga dibuat aplikasi pelacakan objek. Pelacakan objek biasanya dimanfaatkan dalam gps track pada pesawat, gps track pada mobil, alat pendeteksi tubuh manusia di bandara, alat pengatur jumlah kendaraan yang melintas dan alat navigasi pada robot. Penelitian ini bertujuan untuk mengidentifikasi objek yang melintas dalam suatu frame. Penelitian ini juga dapat menghitung jumlah objek yang melintas dalam satu frame. Pelacakan objek dilakukan dengan membandingkan dua algoritma yaitu Horn-Schunck dan Lucas-Kanade. Kedua algoritma menggunakan pengujian Source Block Parameter dan Function Block Parameter. Pengujian dilakukan dengan resolusi video 120x160 dan posisi kamera pengambilan pada jarak 2-4 m. Pengujian pelacakan objek dilakukan dalam durasi 110-120 detik. Tahapan pelacakan objek thresholding, filtering dan region berhasil memperoleh video biner objek. Pengujian algoritma Lucas-Kanade memiliki waktu yang cepat dalam mengidentifikasi objek dibandingkan algoritma Horn-Schunck.

**Kata kunci:** Horn-Schunck, Lucas-Kanade, Optical Flow, Pelacakan Objek

## Abstract

Object tracking one of computer vision. Computer vision similar to human eye function. The difficulty is to detect presence an object and object tracking application made. Object tracking used in aircraft, track cars, human body detectors at airports, a regulator the number of vehicles pass and navigation tools on robots. This study is to identify objects that pass in frame. This research also count the number of objects that pass in one frame. Object tracking done by comparing two algorithms namely Horn-Schunck and Lucas-Kanade. Both algorithms tested using the Source Block Parameter and Function Block Parameter. The test carried out with video resolution 120x160 and the position camera is 2-4 m. The object tracking test is conducted in the duration of 110-120 seconds. Stages tracking object was thresholding, filtering and region successfully obtain object binary video. The Lucas-Kanade has faster in identifying objects compared to the Horn-Schunck algorithm.

**Keywords:** Horn-Schunck, Lucas-Kanade, Object Tracking, Optical Flow

## 1. Pendahuluan

Citra digital didapat dari sekumpulan gambar ataupun video [1]. Gambar atau video merupakan hasil rekaman kamera seperti IP Camera, CCTV, digital camera dan handphone camera [1]. Gambar atau video yang dihasilkan tidak memiliki kualitas yang baik karena pengaruh spesifikasi kamera yang rendah, letak kamera yang tidak tepat, pencahayaan yang kurang serta terdapat noise atau gangguan saat merekam [1]. Berkembangnya teknologi membuat teknik pengolahan citra digital berkembang menjadi computer vision [1].

Computer vision merupakan proses otomatis menginterpretasikan sejumlah besar proses untuk persepsi visual seperti akuisisi, pengolahan citra, klasifikasi, pengenalan (recognition) dan membuat keputusan [2]. Computer vision digunakan untuk mengolah gambar dan video untuk memperoleh hasil yang dapat dianalisa [1]. Computer vision memiliki fungsi yang sama dengan mata manusia dalam hal penglihatan [1]. Pelacakan objek merupakan salah satu penerapan dari

aplikasi *computer vision* untuk mendeteksi pergerakan. Banyak sekali algoritma yang dapat digunakan untuk melakukan pelacakan suatu objek, antara lain algoritma *Optical Flow*.

Algoritma *Optical Flow* merupakan salah satu algoritma pendekatan berbasis *computer vision* yang efisien digunakan untuk melacak pergerakan objek dengan tepat [3]. *Optical Flow* adalah aliran pergerakan dari sebuah objek yang bergerak berdasarkan turunan intensitas cahayanya [1]. *Optical Flow* mempelajari gerakan relatif benda dari berbagai urutan *frame* dengan berdasarkan pergerakan kecepatan objek dan perubahan iluminasi [3]. Teknik *Optical Flow* dapat diklasifikasikan menjadi dua tipe yaitu *Space Optical Flow* dan *Dense Optical Flow* [3]. Algoritma yang termasuk dalam *Space Optical Flow* adalah *Lucas-Kanade*, *Horn-Schunck* dan *Buxton-Buxton* [3]. Algoritma *Space Optical Flow* merupakan algoritma pelacakan dengan melihat bagian tertentu yang akan diteliti seperti sudut objek, tepi objek ataupun bagian lain dari objek yang ingin diamati [3].

Algoritma *Horn-Schunck* merupakan algoritma pelacakan objek bergerak didalam video dengan memberikan urutan pergerakan [4]. Algoritma *Horn-Schunck* dapat mengasumsikan nilai penghalusan di sekitaran gambar, sehingga dapat menghasilkan kepadatan nilai vektor yang tinggi [4]. Faktor kepadatan vektor membuat adanya informasi yang hilang sekitar benda homogen dengan pergerakannya dan lebih sensitif terhadap suara bising [4].

Algoritma *Lucas-Kanade* merupakan perpindahan konten gambar antara dua *instance* terdekat dan konstan dalam lingkungan titik yang dipertimbangkan [5]. Algoritma *Lucas-Kanade* mengasumsikan titik yang dilacak dan titik tetangga dari titik yang dilacak memiliki nilai intensitas yang sama [6]. *Lucas-Kanade* memiliki perpindahan posisi titik yang dilacak dari *frame* ke *frame* adalah kecil [6]. Posisi, arah gerak, besar pergerakan yang dilakukan dan kecepatan gerak dari titik yang dilacak pada *frame* dapat diukur [6].

Pelacakan objek diperlukan untuk mengenali dan mengidentifikasi objek yang melintas dalam satu *frame*. Pelacakan objek melihat aspek kecepatan durasi waktu dalam mengenali objek, ketepatan dalam mengenali bentuk objek, ukuran resolusi objek yang digunakan, letak kamera saat pengambilan objek serta kemampuan algoritma *filtering* yang digunakan untuk menghilangkan *noise* yang ada disekitaran objek baik *background* maupun *foreground*.

Penelitian ini bertujuan untuk melakukan perbandingan pelacakan objek antara dua algoritma pelacakan pada algoritma *Optical Flow*. Algoritma yang digunakan sebagai perbandingan adalah algoritma *Horn-Schunck* dan algoritma *Lucas-Kanade*. Algoritma digunakan untuk mendeteksi objek bergerak (video) yang diambil dengan menggunakan kamera diam pada suatu posisi. Kecepatan waktu pelacakan akan menjadi nilai yang diamati dalam penelitian ini selain ketepatan dalam mengidentifikasi bentuk objek biner yang diperoleh.

Penelitian sebelumnya tentang pelacakan objek adalah Supriyatin tahun 2019 [7], dalam penelitian pelacakan objek dengan menggunakan perbandingan antara algoritma *Optical Flow* dan *Background Estimation* pada kamera bergerak diperoleh bahwa algoritma *Optical Flow* lebih baik dalam mengidentifikasi objek yang melintas dibandingkan dengan algoritma *Background Estimation*. Dengan menggunakan kamera bergerak objek berhasil dideteksi dan dikenali dengan menggunakan algoritma *Optical Flow*.

Penelitian tentang perbandingan pelacakan objek menggunakan algoritma *Optical Flow* dan algoritma *Background Estimation* pada kamera diam yang diletakkan dalam suatu posisi pernah dilakukan oleh Supriyatin, 2018 [8]. Dalam penelitian yang dilakukan ternyata algoritma *Optical Flow* dapat mengidentifikasi objek lebih jelas dan lebih cepat dibandingkan dengan algoritma *Background Estimation*. Tampak bahwa waktu yang dibutuhkan dalam mengenali objek untuk *Optical Flow* kurang dari 100 detik, jauh berbeda dengan algoritma *Background Estimation* yang selalu lebih dari 100 detik dalam mengenali objek.

Tahun 2016, Supriyatin [9] dalam penelitiannya tentang analisis perbandingan pelacakan objek dengan menggunakan *Optical Flow* pada dua kamera yang berbeda yaitu kamera diam yang diletakkan pada satu posisi dan kamera bergerak yang mengikuti pergerakan objek. Dengan menggunakan algoritma *Optical Flow* dengan kondisi perbedaan posisi kamera ternyata untuk posisi kamera diam objek lebih berhasil dideteksi dan dikenali. Sedangkan dengan menggunakan kamera bergerak, objek sulit dideteksi dan tidak terkenali.

Sutanty tahun 2016, [10] dalam penelitian tentang perbandingan algoritma *Optical Flow* dan *Background Estimation* untuk mendeteksi objek pada video ternyata algoritma *Optical Flow* berhasil akurat dalam melakukan perhitungan objek dibandingkan dengan *Background*

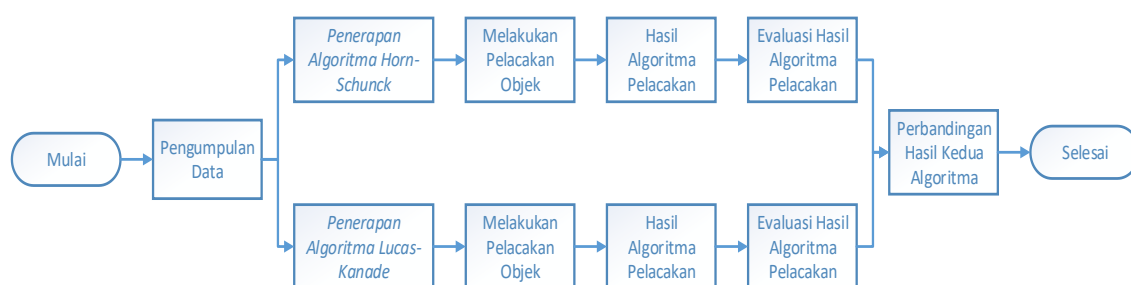
*Estimation*. Proses penangkapan objek bergerak lebih cepat dilakukan *Optical Flow* dengan jumlah *frame* 14 citra *frame/detik* dibandingkan dengan *Background Estimation*.

Penelitian tentang deteksi dan pelacakan perpindahan objek dengan menggunakan *Hybrid* model yaitu kombinasi GMM dan *Optical Flow (Horn-Schunck)* dilakukan oleh Suzan tahun 2017 [4]. *Background subtraction* merupakan cara yang paling cepat untuk mendeteksi pergerakan objek dengan *foreground subtraction* dari *background*. *Background subtraction* dilakukan dengan menggunakan GMM dan pelacakan objek dideteksi dengan menggunakan *Optical Flow*. *Optical Flow* yang digunakan dalam penelitian untuk melacak objek adalah *Horn-Schunck*. Dengan penggabungan kedua model *hybrid* ini maka dapat menghasilkan pelacakan objek yang baik dan tepat.

## 2. Metode Penelitian

Metode yang digunakan dalam penelitian adalah metode pengumpulan data dengan melihat pada *library research*. Mempelajari *tools* yang digunakan dalam pelacakan objek pada Simulink Matlab. Mencari video pelacakan objek dengan mengambil secara langsung dan melalui *library* Matlab.

Analisis perbandingan pelacakan objek dalam penelitian ini membandingkan dua buah algoritma *Optical Flow*, yaitu algoritma *Horn-Schunck* dan algoritma *Lucas-Kanade*. Perbandingan pelacakan dilakukan dengan melihat durasi waktu yang dihasilkan kedua algoritma tersebut dalam mengenali objek. Gambar 1 merupakan tahapan dari penelitian perbandingan pelacakan objek kedua algoritma.



Gambar 1. Tahapan Penelitian

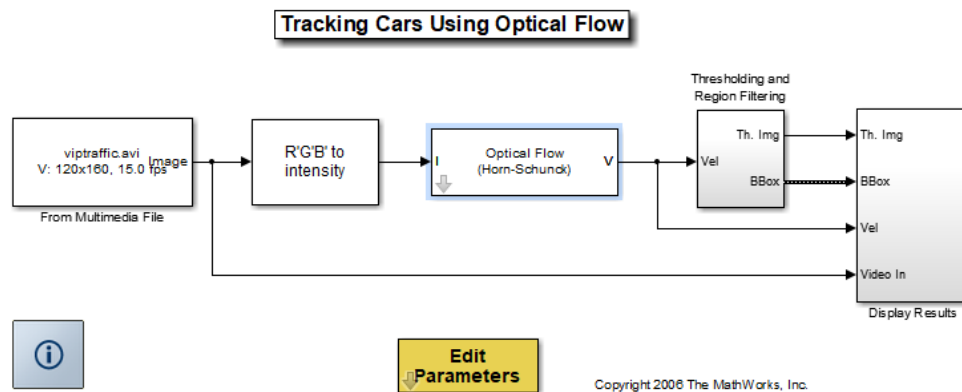
Algoritma *Optical Flow* adalah metode yang digunakan untuk memperkirakan pergerakan *sector* pada setiap *frame* dalam urutan video [7]. *Optical Flow* akan merepresentasikan gerakan dalam aliran *sector* kemudian menghitung nilai koordinat *centroid* untuk menghitung kecepatan kendaraan yang sedang bergerak [7].

Algoritma *Horn-Schunck* adalah algoritma klasik dalam melakukan estimasi algoritma *Optical Flow* [11]. Algoritma ini mengasumsikan suatu sistem kelancaraan dalam *frame* yang ada, melakukan minimalisasi distorsi dalam *flow* dan memilih solusi yang lebih lancar [11] [12]. Gambar 2 adalah algoritma *Horn-Schunck* yang digunakan dalam pelacakan objek.

Tahapan algoritma *Horn-Schunck* dalam mengenali objek pada Gambar 2 adalah :

1. Mulai.
2. Membaca objek : Membanggil video yang akan diolah dengan resolusi 120x160 dan format warna RGB (*Red, Green, Blue*).
3. Melakukan konversi parameter video ke *Intensity* : Mengubah format video dari video RGB ke video *Intensity*.
4. Melakukan Algoritma *Optical Flow (Horn-Schunck)* : Menentukan nilai *Function Block Parameter*, membuat nilai *N frame* dalam kondisi 1, membuat nilai *optical flow* antara *current frame* dan *N frame back*, menentukan nilai *smoothness factor* dalam nilai 1 dan menentukan nilai *iterations* dalam algoritma yaitu 10.
5. *Thresholding* dan *Region Filtering* :
  - a. *Thresholding* : Melakukan segmentasi video untuk mengenali bentuk video. Melihat nilai intensitas piksel dan derajat keabuan sehingga membuat video menjadi bentuk biner.

- Threshold* dilakukan dengan menggunakan *Mean Velocity Threshold* dengan melihat nilai per *frame*.
- b. *Median Filter* : Memisahkan atau mengelompokkan hasil segmentasi video (video biner) dengan *noise*. Menyaring untuk menghilangkan *noise* yang ada pada video.
  - c. *Region Filtering* : Melakukan ekstraksi analisis untuk matrik dan vektor yang diperoleh. Mengelompokkan objek dengan berdasarkan bentuk dan membuang *noise* yang ada.
6. Tampilkan hasil pengenalan objek :
- a. *Video Original* : Video awal dalam bentuk RGB
  - b. *Video Motion Vector* : Menentukan nilai piksel matrik dan vektor yang diperoleh dari *region filtering*.
  - c. *Video Threshold* : Video hasil dari *thresholding* dan *median filter*. Video yang sudah di segmentasi menjadi video biner dan dipisahkan dari *noise* yang ada dalam video.
  - d. *Video Result* : Video akhir dalam bentuk RGB yang sudah di segmentasi dengan *thresholding*, sudah dibersihkan dari *noise* yang ada disekitaran objek dengan *median filter* dan pengenalan bentuk objek dengan menggunakan nilai matrik dan vektor dari objek dengan kondisi *n frame* yang diberikan.
7. Selesai.



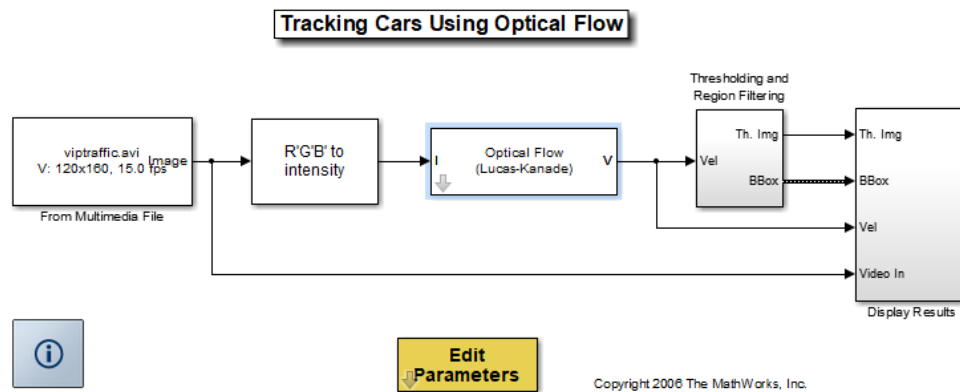
Gambar 2. Algoritma *Horn-Schunck*

Algoritma *Lucas-Kanade* adalah metode diferensial yang banyak digunakan untuk estimasi aliran *optic* [5]. Algoritma *Lucas-Kanade* digunakan untuk menyelesaikan *ambiguitas inheren* dari persamaan aliran *optic* [5]. Algoritma *Lucas-Kanade* diasumsikan bahwa titik yang dilacak dan titik-titik tetangga dari titik yang dilacak memiliki nilai intensitas yang sama dari *frame* ke *frame*, serta perpindahan posisi titik yang dilacak dari *frame* ke *frame* adalah kecil [6]. Gambar 3 adalah algoritma *Lucas-Kanade* yang digunakan dalam pengenalan objek.

Tahapan algoritma *Lucas-Kanade* dalam mengenali objek pada Gambar 3 adalah :

1. Mulai.
2. Membaca objek : Membanggil video yang akan diolah dengan resolusi 120x160 dan format warna RGB (*Red, Green, Blue*).
3. Melakukan konversi parameter video ke *Intensity* : Mengubah format video dari video RGB ke video *Intensity*.
4. Melakukan Algoritma *Optical Flow (Lucas-Kanade)* : Menentukan nilai *Function Block Parameter*, membuat nilai *N frame* dalam kondisi 1, membuat nilai *optical flow* antara *current frame* dan *N frame back*, menentukan nilai *thresholding* untuk *noise* kurang dari 0,0050.
5. *Thresholding dan Region Filtering* :
  - a. *Thresholding* : Melakukan segmentasi video untuk mengenali bentuk video. Melihat nilai intensitas piksel dan derajat keabuan sehingga membuat video menjadi bentuk biner.

- Threshold* dilakukan dengan menggunakan *Mean Velocity Threshold* dengan melihat nilai per *frame*.
- b. *Median Filter* : Memisahkan atau mengelompokkan hasil segmentasi video (video biner) dengan *noise*. Menyaring untuk menghilangkan *noise* yang ada pada video.
  - c. *Region Filtering* : Melakukan ekstraksi analisis untuk matrik dan vektor yang diperoleh. Mengelompokkan objek dengan berdasarkan bentuk dan membuang *noise* yang ada.
6. Tampilkan hasil pengenalan objek :
- a. *Video Original* : Video awal dalam bentuk RGB
  - b. *Video Motion Vector* : Menentukan nilai piksel matrik dan vektor yang diperoleh dari *region filtering*.
  - c. *Video Threshold* : Video hasil dari *thresholding* dan *median filter*. Video yang sudah di segmentasi menjadi video biner dan dipisahkan dari *noise* yang ada dalam video.
  - d. *Video Result* : Video akhir dalam bentuk RGB yang sudah di segmentasi dengan *thresholding*, sudah dibersihkan dari *noise* yang ada disekitaran objek dengan *median filter* dan pengenalan bentuk objek dengan menggunakan nilai matrik dan vektor dari objek dengan kondisi *n frame* yang diberikan.
7. Selesai.



Gambar 3. Algoritma *Lucas-Kanade*

Perbedaan proses dan hasil antara kedua algoritma yang digunakan adalah :

1. Algoritma *Horn-Schunck*
  - a. Proses : nilai *N frame* dalam kondisi 1, nilai *optical flow* antara *current frame* dan *N frame back*, nilai *smoothness factor* = 1 dan nilai *interactions* sebanyak 10.
  - b. Hasil : video biner hasil *thresholding* sebagai bentuk pelacakan objek dan waktu mengenali bentuk objek yang lebih lama dari algoritma *Lucas-Kanade*.
2. Algoritma *Lucas-Kanade*
  - a. Proses : nilai *N frame* dalam kondisi 1, nilai *optical flow* antara *current frame* dan *N frame back* dan nilai *thresholding* < 0,0050.
  - b. Hasil : video biner hasil *thresholding* sebagai bentuk pelacakan objek dan waktu mengenali bentuk objek yang lebih cepat dari algoritma *Horn-Schunck*.

### 3. Hasil dan Pembahasan

Penelitian pelacakan objek ini membandingkan dua algoritma dalam *Optical Flow* yaitu algoritma *Horn-Schunck* dan algoritma *Lucas-Kanade*. Pelacakan objek dilakukan dengan menggunakan objek bergerak yang digunakan untuk melacak dan mengenali bentuk dari objek yang diminta. Video yang digunakan dalam penelitian berjumlah dua buah dengan ketentuan ukuran sebagai berikut :

1. Video *Viptraffic.avi* (resolusi = 120x160; frame = 15.0 fps)
2. Video *Vipmen.avi* (resolusi = 120x160; frame = 30.0 fps)



Kedua video yang digunakan dalam penelitian diambil dengan menggunakan kamera diam yang diletakkan pada satu posisi. Sehingga yang mengalami pergerakan hanya objek yang diamati dalam penelitian. Video pertama berupa video dengan objek kendaraan dan video kedua adalah video dengan objek manusia. Video yang digunakan dalam penelitian diperoleh dari *library* Matlab. Perbandingan kedua algoritma *Optical Flow* dilakukan pada parameter yang sama. Parameter yang digunakan sebagai pembandingan dalam penelitian ini adalah :

1. *Source Block Parameter*
2. *Function Blok Parameter*

Hasil yang diperoleh pada penelitian pelacakan objek dengan menggunakan kamera diam berupa empat buah keluaran yaitu :

1. *Video Original*
2. *Video Motion Vector*
3. *Video Threshold*
4. *Video Results*

Kondisi yang digunakan dalam penelitian untuk kedua objek yang digunakan dalam pelacakan adalah :

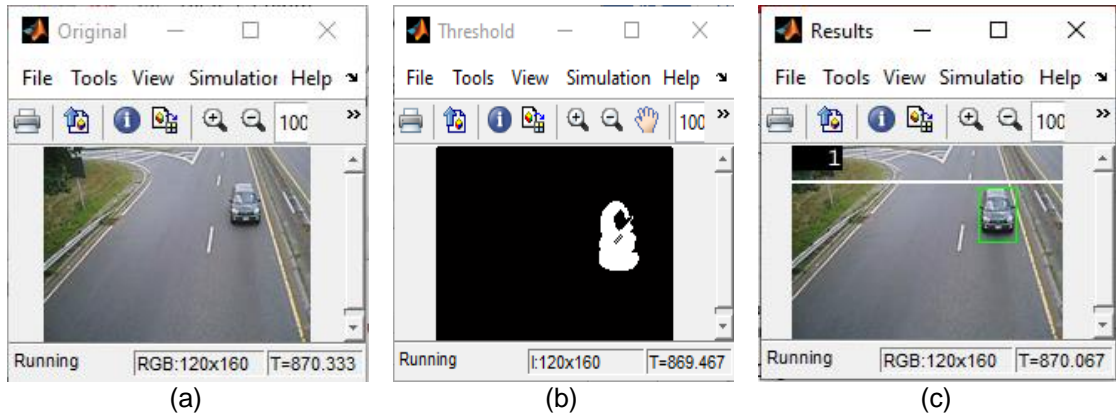
1. Video *Viptraffic.avi* (kamera yang digunakan untuk mengambil objek memiliki jarak 4 m; kecepatan pergerakan objek 60-80 m/s)
2. Video *Vipmen.avi* (kamera yang digunakan untuk mengambil objek memiliki jarak 2 m; kecepatan pergerakan objek 1-2 km/jam)

Tabel perbandingan parameter yang digunakan dalam penelitian dan simulasi untuk uji coba dapat dilihat pada Tabel 1. Parameter yang digunakan untuk algoritma *Horn-Schunck* dan *Lucas-Kanade* memiliki simulasi parameter yang sama.

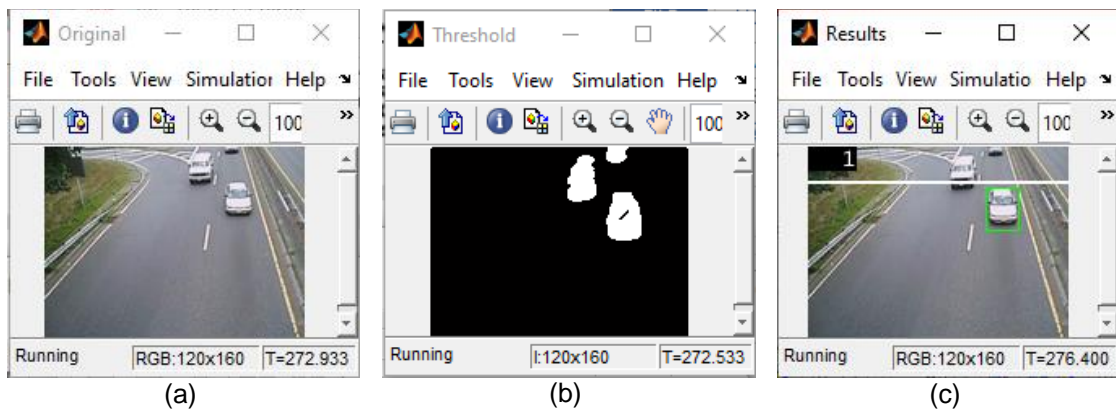
Tabel 1. Parameter Simulasi

Jenis Parameter	Algoritma <i>Horn-Schunck</i>	Algoritma <i>Lucas-Kanade</i>
<b>Source Block Parameter (Multimedia File)</b>	<ul style="list-style-type: none"> <li>• <i>Number of times to play file : inf</i></li> <li>• <i>Output color format : RGB</i></li> <li>• <i>Image signal : One multidimensional signal</i></li> <li>• <i>Video output data type : single</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Number of times to play file : inf</i></li> <li>• <i>Output color format : RGB</i></li> <li>• <i>Image signal : One multidimensional signal</i></li> <li>• <i>Video output data type : single</i></li> </ul>
<b>Function Block Parameter (Color Space Conversion)</b>	<ul style="list-style-type: none"> <li>• <i>Conversion video : RGB to intensity</i></li> <li>• <i>Image signal : One multidimensional signal</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Conversion video : RGB to intensity</i></li> <li>• <i>Image signal : One multidimensional signal</i></li> </ul>
<b>Function Block Parameter (Algorithm)</b>	<ul style="list-style-type: none"> <li>• <i>Computer optical flow between : Current frame and N-th frame back</i></li> <li>• <i>N frame : 1</i></li> <li>• <i>Velocity output : Horizontal and vertical components in complex form</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Computer optical flow between : Current frame and N-th frame back</i></li> <li>• <i>N frame : 1</i></li> <li>• <i>Velocity output : Horizontal and vertical components in complex form</i></li> </ul>

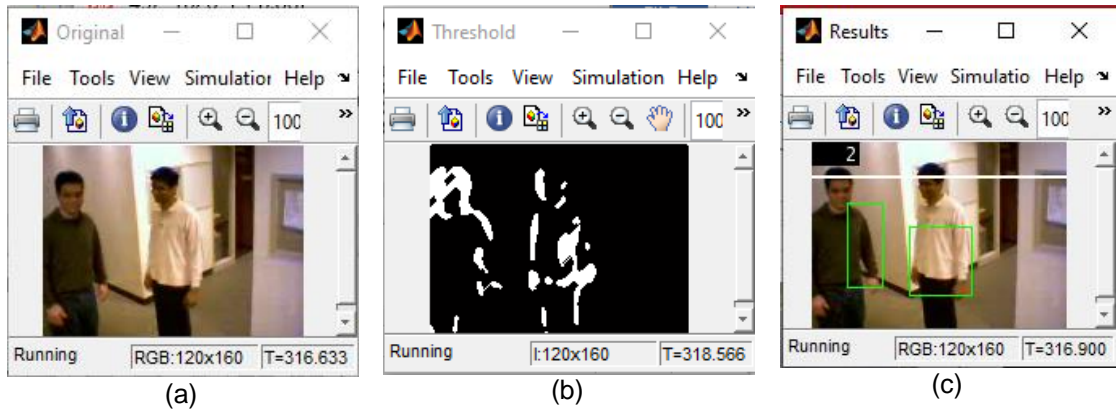
Gambar 4 sampai dengan gambar 7 merupakan hasil pelacakan dari kedua algoritma dengan simulasi parameter yang diberikan pada Tabel 1. Pengujian yang dilakukan menghasilkan video *threshold* dan video *result* seperti terlihat pada gambar 4 sampai gambar 7. Jumlah objek yang berhasil terdeteksi dalam *frame* dihitung mulai dari berapa banyak objek yang melintas dalam garis batas pada hasil video *result* di point c.



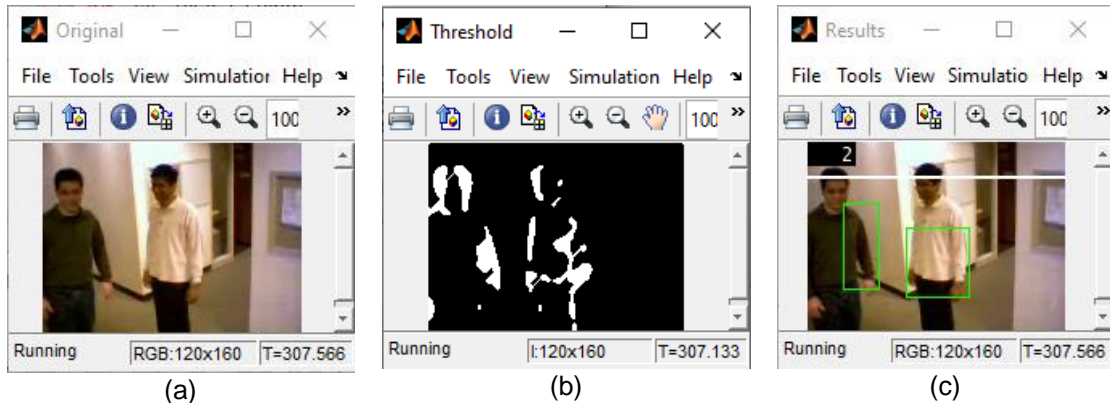
Gambar 4. Video 1 : Viptraffic.avi dengan algoritma *Horn-Schunck* (a). Video original (b). Video *threshold* (c). Video *results*



Gambar 5. Video 1 : Viptraffic.avi dengan algoritma *Lucas-Kanade* (a). Video original (b). Video *threshold* (c). Video *results*



Gambar 6. Video 2 : Vipmen.avi dengan algoritma *Horn-Schunck* (a). Video original (b). Video *threshold* (c). Video *results*



Gambar 7. Video 2 : Vipmen.avi dengan algoritma *Lucas-Kanade* (a). Video original (b). Video *threshold* (c). Video results

Tabel analisa pengujian untuk kedua algoritma dengan menggunakan simulai parameter pada tabel 1 dapat dilihat pada tabel 2. Kedua algoritma dari sisi pelacakan objek, tampak kedua algoritma sama-sama dapat mendeteksi dan mengenali objek, serta dapat mengetahui jumlah objek yang ada.

Tabel 2. Analisa Pengujian

Kriteria Analisa	Algoritma <i>Horn-Schunck</i>	Algoritma <i>Lucas-Kanade</i>
<b>Threshold</b>	Algoritma berhasil melakukan <i>thresholding</i> terhadap objek bergerak yang digunakan dalam penelitian. Objek yang digunakan dalam penelitian berhasil dikenali bentuknya.	Algoritma berhasil melakukan <i>thresholding</i> dari objek yang digunakan. Dari pengujian berhasil mengenali bentuk objek, sehingga pada tahap <i>result</i> dapat dikenali berapa jumlah objek yang melintas atau terdeteksi.
<b>Results</b>	Algoritma berhasil melakukan pelacakan objek dengan mengidentifikasi berapa jumlah objek yang melintas dalam suatu <i>frame</i> .	Algoritma berhasil mengidentifikasi objek yang melintas bahkan dapat menentukan berapa jumlah objek yang melintas dalam <i>frame</i> .

Tabel 3 merupakan perbandingan *Profile Report* antara algoritma *Horn-Schunck* dan algoritma *Lucas-Kanade*. Dari pengujian yang dilakukan terhadap simulasi waktu untuk masing-masing algoritma dengan dua video yang digunakan menggunakan durasi waktu 110 sampai 120. Dimana dari semua pengujian terhadap durasi waktu yang dilakukan memiliki hasil yang sama, hanya saja yang membedakan pada nilai total waktu *record* yang diperoleh. Berdasarkan analisa di Tabel 2 nampak kedua algoritma berhasil melakukan identifikasi objek dan mendeteksi berapa jumlah objek yang melintas. Tetapi dari perbandingan Tabel 3 tampak algoritma *Lucas-Kanade* memiliki total waktu *record* lebih cepat dalam mendeteksi objek dibandingkan dengan algoritma *Horn-Schunck*. Sebagai contoh video Vipmen, dengan waktu simulasi 116 untuk *Lucas-Kanade* menghasilkan total waktu pendeteksian dan pengenalan objek lebih cepat yaitu sebesar 25.14 detik sedangkan *Horn-Schunck* menghasilkan total waktu pendeteksian dan pengenalan objek sebesar 27.58 detik.



Tabel 3. Perbandingan *Profile Report*

Nama Video	Kriteria Penilaian	<i>Profile Report</i> Algoritma <i>Horn-Schunck</i>	<i>Profile Report</i> Algoritma <i>Lucas-Kanade</i>
Viptraffic.avi	Simulation Times	T = 112.333	T = 110.333
	Total Recorded Time	15.63 detik	13.92 detik
	Number of Block Methods	105	105
	Number of Internal Methods	5	5
	Number of Model Methods	9	9
	Number of Nonvirtual Subsystem Methods	15	15
	Clock Precision	0.00000005 detik	0.00000005 detik
	Clock Speed	2000 MHz	2000 MHz
Vipmen.avi	Simulation Times	T =116.233	T = 116.100
	Total Recorded Time	27.58 detik	25.14 detik
	Number of Block Methods	105	105
	Number of Internal Methods	5	5
	Number of Model Methods	9	9
	Number of Nonvirtual Subsystem Methods	15	15
	Clock Precision	0.00000005 detik	0.00000005 detik
	Clock Speed	2000 MHz	2000 MHz

#### 4. Kesimpulan

Pelacakan objek dengan menggunakan algoritma *Horn-Schunck* dan algoritma *Lucas-Kanade* berhasil mendeteksi objek bergerak dari dua video yang digunakan dalam penelitian. Kedua algoritma berhasil mengetahui berapa jumlah objek yang melintas dalam *frame*. Perhitungan jumlah objek yang melintas dilihat dari objek yang melewati garis batas pada hasil uji coba *result*. Dalam segi waktu pelacakan dengan durasi waktu yang sama 110 sampai 120 ternyata algoritma *Lucas-Kanade* lebih cepat dalam mengenali objek dibandingkan dengan algoritma *Horn-Schunck*. Penelitian dilakukan menggunakan data objektif dari dua video. Penelitian dengan menggunakan Simulink Matlab untuk kedua algoritma ini hanya dapat dilakukan pada video dengan resolusi dibawah 200x200 dan *frame* antara 10-30 fps.

Pengembangan selanjutnya untuk penelitian pelacakan objek adalah dapat menggunakan video dengan ukuran resolusi dan *frame* yang lebih besar serta dapat menggunakan kamera yang diletakkan pada posisi dan jarak kamera yang lebih baik. Dapat menggunakan algoritma *Optical Flow* lainnya untuk dibandingkan dengan algoritma *Lucas-Kanade* dalam pengenalan objek.

#### References

- [1] K. M. Kaloh, V. C. Poekoel and M. D. Putro, "Perbandingan Algoritma Background Substraction dan Optical Flow Untuk Deteksi Manusia," *E-Journal Teknik Informatika*, vol. 13, no. 1, pp. 1-9, 2018.
- [2] W. Supriyatin, Y. Rafsyam and Jonifan, "Analisis Pelacakan Objek Menggunakan Background Estimation Pada Kamera Diam dan Bergerak," *ORBITH*, vol. 13, no. 2, pp. 124-130, 2017.
- [3] A. Balasundaram, S. A. Kumar and S. M. Kumar, "Optical Flow Based Object Movement Tracking," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 1, pp. 3913-3916, 2019.
- [4] A. M. Suzan and G. Prathibha, "Detection And Tracking of Moving Objects Using Hybrid Model (GMM & Horn-Schunck)," *International Journal of Electrical and Electronic Engineering & Telecommunications (IJEETC)*, vol. 6, no. 2, pp. 37-42, 2017.
- [5] Wikipedia, "Lucas-Kanade Method," Wikipedia, 14 December 2019. [Online]. Available: [https://en.m.wikipedia.org/wiki/Lucas%E2%80%93Kanade\\_method](https://en.m.wikipedia.org/wiki/Lucas%E2%80%93Kanade_method). [Accessed 08 April 2020].



- [6] Y. Syarif, R. R. Isnanto and K. I. Satoto, "Perangkat Lunak Pengendali Pointer Menggunakan Pelacakan Mata (Eye Tracking) Dengan Algoritma Lucas Kanade," *TRANSIENT*, vol. 2, no. 3, pp. 714-719, 2013.
- [7] W. Supriyatin, "Analisis Perbandingan Pelacakan Objek Menggunakan Optical Flow dan Background Estimation Pada Kamera Bergerak," *ILKOM Jurnal Ilmiah*, vol. 11, no. 3, pp. 191-199, 2019.
- [8] W. A. W. W. Supriyatin and I. Astuti, "Comparative Analysis of Tracking Objects Using Optical Flow and Background Estimation on Silent Camera," *KINETIK*, vol. 3, no. 2, pp. 91-100, 2018.
- [9] W. Supriyatin and W. W. Ariestya, "Analisis Pelacakan Objek Mobil Dengan Optical Flow Pada Kamera Diam dan Bergerak," *Seminar Riset Teknologi Informasi (SRITI)*, pp. 48-56, 2016.
- [10] E. Sutanty and A. Rosemala, "Analisis Perbandingan Algoritma Optical Flow dan Background Estimation untuk Pendeteksian Objek pada Video," *Jurnal Ilmiah KOMPUTASI*, vol. 15, no. 1, pp. 15-21, 2016.
- [11] Z. Fan, "Optical Flow Estimation With Horn-Schunck Method," 25 March 2015. [Online]. Available: <https://fzheng.me/2015/03/25/optical-flow/>. [Accessed 04 May 2020].
- [12] Wikipedia, "Horn-Schunck method," Wikipedia, 05 February 2020. [Online]. Available: [https://en.m.wikipedia.org/wiki/Horn-Schunck\\_method](https://en.m.wikipedia.org/wiki/Horn-Schunck_method). [Accessed 04 May 2020].