

ANALISIS DETEKSI PERGERAKAN OBJEK PADA CITRA VIDEO MENGUNAKAN ALGORITMA KALMAN FILTER

Ahmad Apandi, Rahmah Suci Hati
Universitas Gunadarma
ahmadapandi@staff.gunadarma.ac.id, rahmahsucihati@gmail.com

ABSTRAK

Pada sistem pemantauan kendaraan dibutuhkan suatu sistem pelacakan dan deteksi kendaraan guna mengetahui jumlah kendaraan, kecepatan kendaraan dan klasifikasi kendaraan pada jalan raya dengan menggunakan metode tertentu. Pengamatan objek bergerak pada video banyak digunakan untuk pemantauan kendaraan di jalan raya ataupun pemantauan keadaan suatu ruangan. Pendeteksian objek juga digunakan dalam berbagai macam aplikasi seperti: keamanan, video pengawasan (surveillance), kecelakaan lalu lintas, potensi terjadinya suatu kerusakan di suatu tempat atau bahkan kasus kriminal. Pendeteksian objek menjadi begitu penting seiring dengan meningkatnya beberapa kasus kriminal di Indonesia. Keamanan menjadi hal yang sangat penting dalam melihat banyaknya tingkat kejahatan yang terjadi dilingkungan sekitar. Pada penulisan ini, penulis akan melakukan analisis terhadap deteksi objek bergerak yang terekam kamera dalam sample video dengan objek bergerak yang menjadi fokus pada penelitian ini adalah orang. Analisis yang dilakukan yaitu analisis cara kerja algoritma Kalman Filter dalam mendeteksi objek dengan menggunakan 3 parameter, yaitu pencocokan piksel (pixel based matching), pencocokan blok piksel (block-based pixel) dan pencocokan region (region-based matching) dengan simulasi video yang tidak terkoneksi CCTV. Hasil penelitian ini menyimpulkan bahwa algoritma Kalman Filter berhasil melakukan pelacakan pergerakan objek yang tidak menentu berdasarkan 3 parameter pencocokan, yaitu pencocokan piksel, pencocokan blok piksel dan pencocokan region.

Kata Kunci: Deteksi Objek, Kalman Filter, Region of Interest, Threshold, Video.

PENDAHULUAN

Pada sistem pemantauan kendaraan dibutuhkan suatu sistem pelacakan dan deteksi kendaraan guna mengetahui jumlah kendaraan, kecepatan kendaraan dan klasifikasi kendaraan pada jalan raya dengan menggunakan metode tertentu. Pada sistem pemantauan manusia pada suatu ruangan/ suatu tempat digunakan untuk mengetahui gerakan yang dilakukannya. Pendeteksian objek juga digunakan dalam berbagai macam aplikasi seperti keamanan, video pengawasan (surveillance), kecelakaan lalu lintas, potensi terjadinya suatu kerusakan di suatu tempat atau bahkan kasus kriminal. Pendeteksian objek menjadi begitu penting seiring dengan meningkatnya beberapa kasus kriminal di Indonesia.

Keamanan menjadi hal yang sangat penting dalam melihat banyaknya tingkat kejahatan yang terjadi dilingkungan sekitar. Dalam melakukan pemantauan keamanan, suatu tempat saat ini banyak menggunakan *Closed Circuit Television (CCTV)*. Pengamatan objek bergerak juga dapat dengan mudah dilakukan dengan kamera. Perangkat pemantauan keamanan seperti CCTV hanya dapat menangkap banyak objek yang bergerak tetapi tidak dapat memberikan informasi tentang gerakan dari objek tersebut. Hal ini menjadi tidak efektif bila terlalu banyak gambar yang akan diamati dan memungkinkan terjadinya kesalahan pengamatan.

Salah satu cara untuk mempermudah dalam mengamati banyak objek bergerak adalah melakukan analisis objek bergerak dengan metode pendeteksian gerakan pada suatu objek. Sistem ini akan mendeteksi gerakan yang tertangkap oleh perangkat

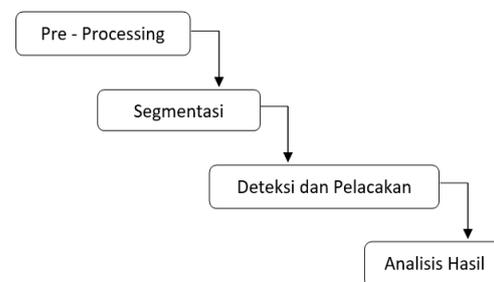
kamera dan memberikan informasi dari pergerakannya yang berupa deteksi tepi. Deteksi tepi ini nantinya memiliki keuntungan yaitu gerakan yang terjadi pada suatu objek dapat terdeteksi lebih efektif. Pendeteksian objek dapat digunakan untuk mengetahui perbedaan antara objek yang bergerak dan tidak bergerak untuk keperluan analisis, seperti analisis perilaku pejalan kaki, analisis dan prediksi terjadinya kerusakan dan analisis terjadinya kecelakaan lalu lintas (Babu, Sekhar, 2015).

Beberapa metode untuk pelacakan dan deteksi objek telah dilakukan beberapa peneliti terdahulu. Penelitian untuk mendeteksi objek bergerak pada video pengawasan (*surveillance*) dilakukan dengan metode *Kalman Filter*. Deteksi objek bergerak dilakukan untuk melacak dan mendeteksi kendaraan di jalan raya dengan menggunakan video statik. Penelitian dilakukan untuk memecahkan masalah akurasi yang dialami pada penelitian sebelumnya dengan mengimplementasikan *Frame Difference* dengan menggunakan *K-Means* secara *online* (Rao, Darwin, 2012). Penelitian dalam mendeteksi objek dilakukan peneliti lain menggunakan metode *global contour shape* yang terbukti lebih sesuai terhadap beragam skala dan rotasi. Proses pendeteksian diawali dengan segmentasi pada citra, *boundary* pada setiap region diambil sebagai input untuk proses selanjutnya. Proses segmentasi dilakukan menggunakan metode *statistical region merging* (Wang, Xiong, Conghuan Ye, Zhang, 2013). Penelitian pendeteksian objek dilakukan untuk melacak dan deteksi kendaraan dengan menggunakan algoritma *Optical Flow* dan *Background Subtraction* dilakukan untuk pemantauan lalu lintas di jalan raya. Pada penelitian ini *Optical Flow* dan *Background Subtraction* digunakan untuk pelacakan dan deteksi kendaraan dari urutan video yang diperoleh melalui kamera, dengan mengasumsikan jalanan dalam keadaan datar dan lurus (Paygude, Vibna, Manisha, 2013).

Pada penelitian ini, peneliti akan melakukan analisis deteksi pergerakan objek pada citra video (studi kasus: objek bergerak yang terekam kamera) menggunakan algoritma *Kalman Filter* berdasarkan 3 parameter antara lain : pencocokan piksel (*pixel based matching*), pencocokan blok piksel (*block-based pixel*) dan pencocokan region (*region-based matching*). Hasil analisis ini diharapkan dapat mengetahui cara kerja algoritma dalam membedakan objek bergerak dan tidak bergerak berdasarkan parameter *pixel based matching*, *block-based pixel* dan *region-based matching* algoritma *Kalman Filter*, untuk menghindari salah pengamatan dalam sebuah video yang digunakan untuk pengawasan keamanan di suatu ruangan atau suatu tempat.

METODE PENELITIAN

Secara umum, metode penelitian pada tesis ini terdiri dari empat tahapan, yaitu : *pre-processing*, segmentasi, deteksi dan pelacakan objek, dan analisis hasil.



Gambar 1. Bagan Umum Penelitian

1. *Pre-Processing* dengan latar belakang *Substraction*

Tahapan *pre-processing* merupakan tahapan yang pertama dilakukan dalam pendeteksian dan pelacakan objek. Tahapan ini akan menjadi dasar untuk tahapan selanjutnya. Pada tahapan ini ada beberapa langkah yang akan dilakukan oleh masing-masing sub langkah, diantaranya yaitu membuat urutan *frame* dari video

input. Kemudian dilanjutkan dengan proses pengambilan gambar saat ini dan sebelumnya dan dilanjutkan dengan mengambil perbedaan diantara keduanya. Proses selanjutnya yaitu memilih *threshold* dan dilanjutkan dengan melakukan penyaringan atau *filtering* terhadap *noise* dari objek.

a. Pembentukan *Frame*

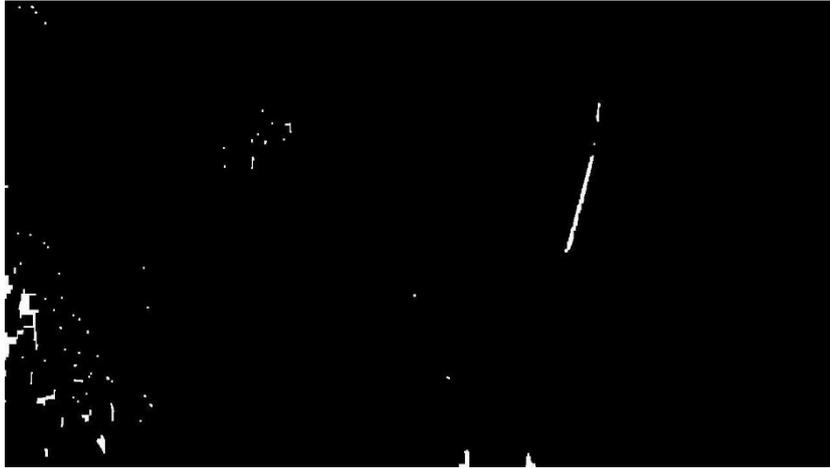
Pembentukan *frame* adalah tahapan dimana video input (video yang akan dijadikan sampel untuk dideteksi) diubah menjadi urutan *frame*. Karena pada dasarnya video merupakan kumpulan dari urutan *frame*. Setelah pembuatan urutan *frame-frame* dari video, maka dilakukan tahapan pembuatan gambar (*frame*) saat ini dan *frame* sebelumnya dan kemudian mengambil perbedaan dari kedua *frame* tersebut. Setelah mengambil perbedaan dari kedua *frame*, maka dilakukan pemilihan *threshold* dan langkah terakhirnya adalah menghapus *noise* yang terdapat pada *frame*.

b. Pembuatan Array dari *Track* dan Inisialisasi Variabel

Tahapan pembuatan array dari *track* dan inisialisasi variabel merupakan tahapan yang dilakukan setelah tahapan pemisah video menjadi gambar (urutan *frame*). *Array of track* akan digunakan untuk menyimpan perjalanan sebuah objek dari suatu *frame* ke *frame* lainnya, dimulai dari objek itu pertama kali terlihat hingga objek tersebut tidak terlihat. *Array of track* akan menyimpan semua *track* dari objek yang terdeteksi pada suatu video. Setiap *track* memiliki struktur yang bertujuan untuk menjaga state dari objek yang terdeteksi kamera. State terdiri dari beberapa atribut yang digunakan untuk pendeteksian objek, pelacakan objek, pemberhentian pelacakan dan menampilkan objek.

Sebuah *noise* biasanya menghasilkan *track* pendek, contohnya objek hanya terlihat pada beberapa *frame* saja, kemudian objek tersebut tidak terlihat lagi karena sudah melebihi batasan dari

total *visible count*. Ketika tidak ada pendeteksian keberadaan objek yang terkait dengan sebuah *track* secara berturut-turut maka dapat diasumsikan bahwa objek telah meninggalkan pandangan kamera dan menghapus *track* tersebut. Hal ini terjadi karena nilai dari *Consecutive Invisible Count* sudah melebihi batasan yang telah ada dan *track* juga bisa dihapus dikarenakan terdeteksi sebagai *noise* seperti dapat dilihat pada Gambar 2.

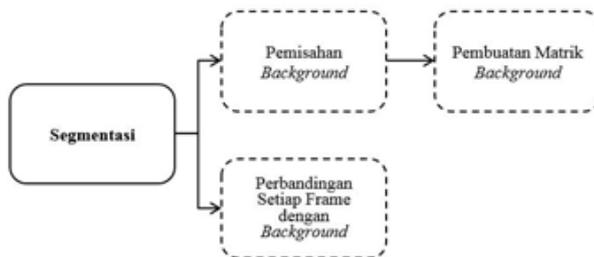


Gambar 2. Contoh Noise Pada *Frame*

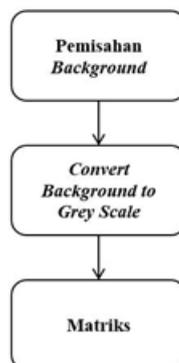
2. Segmentasi

Tahapan ini akan dilakukan dengan beberapa langkah umum diantaranya menghitung titik pusat dari objek yang terdeteksi dan membuat *bounding box* untuk objek yang terdeteksi serta

mengklasifikasikan *foreground* dari *background*. Tahapan segmentasi yang dilakukan pada penelitian ini yaitu menggunakan *background subtraction* dengan langkah-langkahnya seperti pada Gambar 3.



Gambar 3. Bagan Tahapan Segmentasi



Gambar 4. Bagan Tahapan Pemisahan *Background*

Gambar 4 adalah bagan tahapan pemisahan *background* hingga pembentukan matriks.

Pemisahan *background* dilakukan dengan menggunakan teknik *background*

substraction. Gambar 5 merupakan gambar dari pemisahan *background* atau *background* yang dihasilkan pada salah satu video yang digunakan pada penelitian ini.



Gambar 5. Background

Setelah *background* didapatkan maka dilanjutkan ketahapan selanjutnya yaitu tahapan pengubahan *background* kedalam bentuk *grayscale* dengan menggunakan

convert to grayscale pada matlab. Gambar 6 merupakan gambar *background* yang sudah diubah ke dalam bentuk *grayscale*.



Gambar 6. Pengubahan Background menjadi Citra Grayscale

Tahapan selanjutnya yang dilakukan adalah pembuatan matrik dari *background* yang sudah diubah ke dalam bentuk *grayscale*.

Untuk matriknya adalah seperti yang terlihat dibawah ini:

$$\begin{bmatrix} 74 & 79 & 47 & 10 & 0 \\ 82 & 67 & 30 & 0 & 2 \\ 18 & 0 & 0 & 0 & 6 \\ 0 & 11 & 54 & 71 & 0 \\ 14 & 86 & 147 & 153 & 109 \end{bmatrix}$$

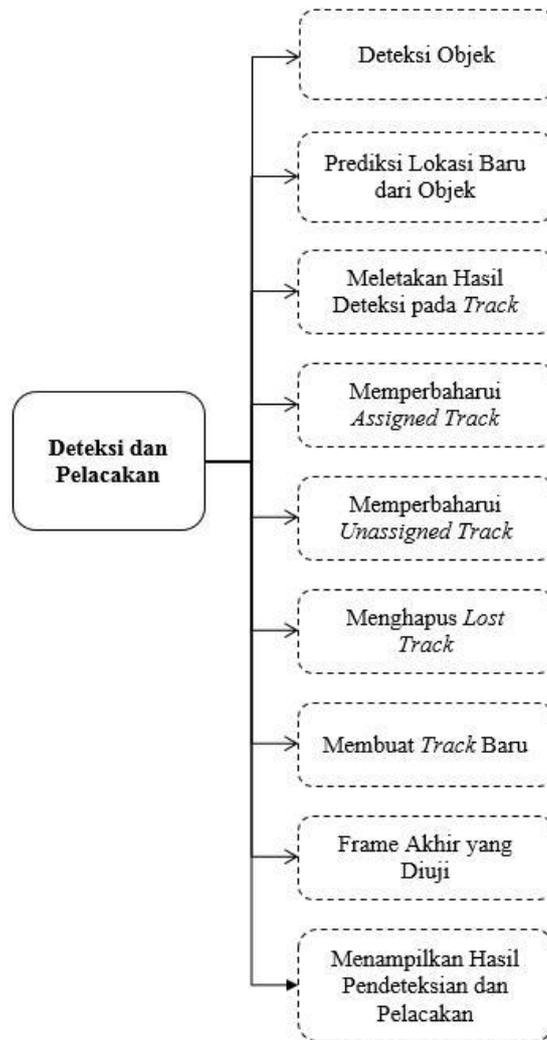
Matrik di atas merupakan bagian dari matrik *background* yang sudah diubah kedalam bentuk *grayscale*. Untuk matriks secara keseluruhan tidak dapat di tampilkan karena ukuran matriks yang sangat besar

yaitu 720 x 1280 sesuai dengan ukuran dimensi gambar.

3. Deteksi dan Pelacakan Objek

Tahapan deteksi dan pelacakan objek merupakan tahapan terakhir sebelum

tahapan analisis hasil yang mana tahapan ini akan terdiri dari beberapa proses hingga pada akhir proses akan menampilkan hasil dari deteksi dan pelacakan objek. Bagan tahapan deteksi dan pelacakan objek adalah seperti yang terlihat pada Gambar 7.



Gambar 7. Bagan Tahapan Deteksi dan pelacakan

Proses-proses yang terdapat pada tahapan ini diantaranya adalah tahapan deteksi objek yang kemudian dilanjutkan dengan proses memprediksi lokasi baru dari objek. Selanjutnya yaitu proses meletakkan hasil deteksi ke dalam *track* dan dilanjutkan dengan memperbaharui *assigned track* dan *unassigned track*. Setelah proses tersebut maka akan dilanjutkan dengan proses menghapus *lost track* dan membuat *track* yang baru hingga

menghasilkan *frame* yang akan diuji dan pada akhirnya akan menampilkan hasil pendeteksian dan pelacakan objek. Ada kondisi dimana proses langsung masuk ke proses memperbaharui *assigned track* jika objek yang muncul sebelumnya tetap muncul atau tidak hilang. Namun jika objek yang sebelumnya muncul menjadi hilang, maka sebelum masuk ke proses memperbaharui *assigned track* harus melewati proses memprediksi lokasi yang

baru dan baru dilanjutkan ke proses selanjutnya.

4. Prediksi Lokasi Baru dari Objek

Tahapan prediksi lokasi baru dari objek akan diproses jika objek yang sebelumnya muncul namun kemudian objek tersebut hilang dan tidak akan di proses jika objek tidak hilang. Fase ini merupakan fase yang memprediksi kemungkinan munculnya *track* baru. Pada fase inilah digunakan algoritma *kalman filter*. Penggunaan *kalman filter* digunakan untuk memprediksi lokasi objek yang baru dan mengurangi *noise* dari pendeteksian objek atau membantu menghubungkan beberapa objek dalam suatu *track*. Dalam pendeteksian objek, syarat penggunaan *kalman filter* adalah objek harus bergerak konstan (tetap) atau percepatan konstan (tetap). *Kalman filter* terdiri dari 2 tahapan yaitu:

Prediction

Pada tahapan *prediction* saat pendeteksian objek untuk memprediksi *state* atau *track* saat ini dengan menggunakan *state* atau *track* yang sebelumnya.

Correction

Pada tahapan *correction*, bahan pengoreksian yang digunakan adalah hasil pengukuran pada tahapan sebelumnya seperti pendeteksian lokasi objek.

Nama : atrium.avi (dengan ekstensi video adalah .avi)

Durasi : 20 detik

Width : 640

Height : 360

c. Terdapat objek *background*, yaitu objek yang menjadi latar belakang pada pendeteksian objek. Objek ini merupakan objek yang paling sering muncul pada *frame-frame* yang

Jika pada *frame* sebelumnya (t-1) objek terdeteksi, namun pada *frame* saat ini (t) objek tiba-tiba tidak terdeteksi maka prediksi dilakukan dengan menggunakan *frame* sebelumnya.

HASIL DAN PEMBAHASAN

1. PreProcessing

Tahapan *pre-processing* merupakan suatu tahapan yang dilakukan pertama kali pada pendeteksian dan pelacakan objek. Tahapan ini berkaitan dengan video yang digunakan untuk penelitian serta proses perubahan video kedalam bentuk *frame-frame*. Penelitian ini menggunakan video sebagai bahan studi kasus. Hasil analisa mengenai video yang dijadikan sebagai video *input* pada penelitian adalah video yang telah disediakan oleh *library* Matlab atau *default* video.

- a. Video direkam dengan menggunakan kamera dengan posisi kamera statis atau tetap dengan lokasi pengambilan video pada suatu atrium. Cahaya saat pengambilan video adalah terang.
- b. Spesifikasi video yang digunakan dalam penelitian:

dihasilkan dari video atau dapat disebut juga sebagai benda diam. Contoh objek *background* diantaranya yaitu: kursi, meja, pohon, pagar dan taman.



Gambar 9. Background

d. Terdapat objek foreground, yaitu objek bergerak yang terdapat pada video. Objek ini merupakan objek yang akan dideteksi. Objek bergerak atau objek foreground pada

video ini adalah orang yang berjalan atau melintasi area cakupan kamera yang digunakan untuk merekam. Contoh objek foreground dapat dilihat pada Gambar 10.



Gambar 10. Background dan Foreground

Pada tahapan pre-processing selain mendapatkan analisa mengenai video input yang digunakan untuk penelitian, terdapat juga proses perubahan video menjadi *frame-frame* atau gambar. Jumlah *frame* yang dihasilkan akan sesuai dengan durasi video dikali dengan banyaknya *frame* yang dihasilkan perdetik (*frame rate*). Informasi durasi dan *frame* yang dihasilkan video

perdetik dapat dilihat pada Gambar 11 serta contoh *frame* yang dihasilkan dapat dilihat pada Gambar 12. Pada Gambar 11 dapat dilihat bahwa video yang digunakan untuk penelitian ini memiliki informasi sebagai berikut:

Durasi : 20 detik
 Frame rate : 30 frames/ detik

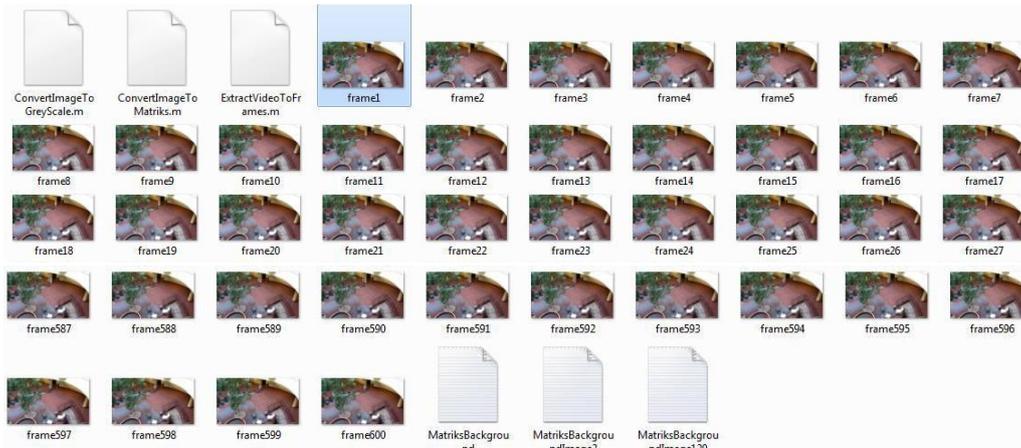
**Maka, total *frame* yang harusnya dihasilkan oleh video input ini adalah:
 20 detik X 30 *frames* = 600 *frames*.**

Dapat dilihat pada Gambar 12 bahwa jumlah *frame* yang dihasilkan adalah 600 *frames*. Maka dapat disimpulkan bahwa

jumlah *frame* yang dihasilkan telah sesuai dengan jumlah *frame* yang seharusnya dihasilkan.



Gambar 11. Informasi Video



Gambar 12. Contoh Urutan *Frame* Video

2. Segmentasi

Tahapan segmentasi yaitu tahapan pemisahan *background* dan juga tahapan pembuatan matrik *background*. Hasil dari tahapan segmentasi ini atau tahapan pencarian *background* ini dilakukan secara

otomatis dengan mencari nilai modus pada *frame-frame* yang dihasilkan. Gambar 13 merupakan *background* yang didapatkan dari pencarian nilai modus dari *frame-frame* yang dihasilkan



Gambar 13. *Background* Video

Setelah didapatkan *background* dengan cara mencari nilai modus dari *frame-frame* yang dihasilkan video, maka selanjutnya dapat dihasilkan matrik *background* dari penelitian ini. Namun sebelum membuat matriks *background*, dilakukan terlebih dahulu perubahan *background* dari citra RGB menjadi citra

greyscale. Hasil pengubahan *background* menjadi citra greyscale dapat dilihat pada Gambar 14. Matrik *background* ini dihasilkan dengan cara mengkonversi citra greyscale ke dalam bentuk matriks. Matriks yang dihasilkan sesuai dengan ukuran height dan width *frame* yaitu 360 x 640.

255	255	255	254	255
244	250	253	255	255
222	229	237	242	255
227	232	230	225	237
208	212	227	240	230

Matriks *Background Video*



Gambar 14. Hasil Pengubahan *Background* menjadi Citra Greyscale

3. Deteksi dan Pelacakan Objek

Saat melakukan pendeteksian dan pelacakan objek ada beberapa proses yang akan terjadi, seperti deteksi objek yang merupakan proses pembacaan objek saat pertama kali objek masuk kedalam area

cakupan kamera dan kemudian dilanjutkan dengan proses pelacakan lainnya sesuai dengan yang telah dijelaskan pada penjelasan sebelumnya. Hasil tahapan deteksi objek dapat dilihat pada Tabel 1.

Tabel 1. Cara Kerja Pendeteksian Objek

No	Proses	Frame yang diuji	Background
1	Membaca <i>frame</i> citra		
2	Mengubah <i>frame</i> citra ke dalam bentuk <i>greyscale</i>		

3	Mengubah hasil <i>greyscale</i> ke dalam bentuk matrik (contoh bagian matriks)	<table border="1" data-bbox="644 208 874 387"> <tr><td>255</td><td>255</td><td>255</td><td>254</td><td>255</td></tr> <tr><td>241</td><td>247</td><td>251</td><td>253</td><td>255</td></tr> <tr><td>218</td><td>224</td><td>233</td><td>240</td><td>255</td></tr> <tr><td>226</td><td>231</td><td>229</td><td>224</td><td>229</td></tr> <tr><td>208</td><td>214</td><td>227</td><td>237</td><td>239</td></tr> </table> <p data-bbox="676 398 842 421">Matriks_Citra Uji_Video</p>	255	255	255	254	255	241	247	251	253	255	218	224	233	240	255	226	231	229	224	229	208	214	227	237	239	<table border="1" data-bbox="1056 208 1286 387"> <tr><td>255</td><td>255</td><td>255</td><td>254</td><td>255</td></tr> <tr><td>244</td><td>250</td><td>253</td><td>255</td><td>255</td></tr> <tr><td>222</td><td>229</td><td>237</td><td>242</td><td>255</td></tr> <tr><td>227</td><td>232</td><td>230</td><td>225</td><td>237</td></tr> <tr><td>208</td><td>212</td><td>227</td><td>240</td><td>230</td></tr> </table> <p data-bbox="1088 398 1254 421">Matriks_Background_Video</p>	255	255	255	254	255	244	250	253	255	255	222	229	237	242	255	227	232	230	225	237	208	212	227	240	230
255	255	255	254	255																																																	
241	247	251	253	255																																																	
218	224	233	240	255																																																	
226	231	229	224	229																																																	
208	214	227	237	239																																																	
255	255	255	254	255																																																	
244	250	253	255	255																																																	
222	229	237	242	255																																																	
227	232	230	225	237																																																	
208	212	227	240	230																																																	
4	Hasil perbandingan matrik <i>frame</i> citra																																																				
5	Mengubah hasil perbandingan matriks																																																				
6	Membuat Bounding Box																																																				
7	Menvisualisasikan hasil deteksi ke dalam frame yang diuji																																																				

Berikut beberapa keterangan mengenai Tabel 1. :

- a. Hasil dari tahapan pendeteksian yang ditampilkan pada Tabel 1. hanya dengan mengambil satu contoh frame saja yaitu frame 101.
- b. Bentuk matriks yang di tampilkan diatas hanya merupakan contoh

bagian dari bagian matriks frame 101. Hal ini dikarenakan matrik yang dihasilkan oleh citra mempunyai dimensi yang sangat besar yaitu 360X640.

Tabel 2. Tabel Deteksi Objek

Item	Jumlah
Total objek bergerak (orang) yang terekam video	7
Jumlah objek bergerak (orang) yang terdeteksi	7
Jumlah objek bergerak (orang) yang tidak terdeteksi	0
Presentase keberhasilan objek bergerak terdeteksi	100%
Presentase kegagalan objek bergerak tidak terdeteksi	0%

Keterangan:

Presentase keberhasilan objek bergerak terdeteksi:

$$\frac{\text{Total objek terdeteksi}}{\text{Total objek}} \times 100\%$$

Presentase kegagalan objek bergerak tidak terdeteksi:

$$\frac{\text{Total objek tidak terdeteksi}}{\text{Total objek}} \times 100\%$$

Dari hasil diatas dapat dilihat bahwa untuk tahapan pendeteksian objek, untuk semua objek *foreground* yang muncul terdeteksi dengan baik tanpa ada kesalahan. Jadi, dapat disimpulkan bahwa untuk pendeteksian objek menggunakan algoritma *Kalman Filter* bekerja dengan baik. Namun, hasil pendeteksian untuk objek bergerak (*foreground*) yang berdekatan algoritma bekerja kurang optimal. Hal ini disebabkan karena jika 2 atau lebih objek bergerak berdekatan maka akan terdeteksi sebagai 1 objek.

KESIMPULAN DAN SARAN

Pendeteksian dan pelacakan objek bergerak dengan menggunakan suatu metode tertentu merupakan salah satu cara untuk mempermudah dalam mengamati objek bergerak. Sistem dengan menggunakan metode tersebut akan

memberikan informasi dari pergerakan objek yang berupa deteksi tepi. Salah satu algoritma yang digunakan untuk pendeteksian dan pelacakan objek bergerak adalah algoritma Kalman Filter. Berdasarkan analisis cara kerja algoritma Kalman Filter untuk pendeteksian dan pelacakan objek bergerak dengan menggunakan contoh video dapat diambil beberapa kesimpulan bahwa Algoritma kalman filter berhasil melakukan pelacakan pergerakan masing-masing objek berdasarkan posisi dan jumlah objek yang terlihat tidak menentu. Adapun saran yang dapat penulis berikan untuk mengembangkan penelitian ini adalah video yang digunakan sebagai bahan studi kasus penelitian dapat dikembangkan dengan menggunakan video yang berasal dari rekaman CCTV dengan kondisi cahaya yang berbeda-beda. Bagi peneliti lainnya, pengembangan dapat dilakukan dengan

menghitung nilai korelasi frame berdasarkan karakteristik tertentu dan jarak objek / area dari titik utama citra untuk dicocokkan, dan pencarian posisi objek dalam menentukan algoritma deteksi objek bergerak dengan tepat.

Unscented Kalman Filtering for Additive Noise Case: Augmented vs. Non-augmented, American Control Conference, Portland, OR, USA.

DAFTAR PUSTAKA

Aldila, Antoni. (2012). *Desain Filter Kalman untuk Mengestimasi Variabel Keadaan yang Tidak Terukur pada Sistem Tata Udara Presisi*, Universitas Indonesia.

Arioputra, Dimas. (2012). *Analisa Perbandingan Akurasi Metode Optical Floe dan Gaussian Mixture Model untuk Sistem Pemantauan Lalu Lintas Berbasis Computer Vision*, Universitas Indonesia.

Babu K, Maruthi. and Sekhar K, Sri Chandra. (2015). *Pedestrian Head Impact Analysis*. International Journal of Mechanical Engineering and Technology (IJMET). Volume 6 Issue 3, pp. 08-13.

Burger, Wilhelm. and Mark, J. Burge. (2009). *Principles of Digital Image Processing Core Algorithms*. Verlag London: Springer.

Cyganek, Boguslaw. (2013). *Object Detection and Recognition in Digital Images*, United Kingdom, Wiley.

Gonzales, Rafael. C. and E. Woods, Richard. (2008) *Digital Image Processing Third Edition*, Prentice Hall

Kusumanto. and Tompunu, Alan Novi. (2011). *Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB*, ISBN 979-26-0255-0, pp. 1-7.

Yuanxin, Wu., Dewen, Hu., Meiping, Wu., and Xiaoping, Hu. (2005).