

Otomasi Kamera Perangkat Menggunakan Deteksi Gerak dan Komputer Papan Tunggal

Habib Dwi Cahya*¹, Agus Harjoko²

¹Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta

²Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: *Habib.dwi.c@mail.ugm.ac.id, aharjoko@ugm.ac.id

Abstrak

Kamera USB saat ini telah dipakai dalam kehidupan sehari-hari untuk berbagai keperluan. Pada perkembangannya, penggunaan kamera USB dapat digunakan untuk membuat kamera perangkat dan dapat digunakan untuk mengamati perkembangan hewan dengan sistem yang terintegrasi. Pada penelitian ini deteksi gerak digunakan untuk mengamati hewan secara online menggunakan komputer papan tunggal.

Kamera perangkat pada penelitian ini menggunakan Single Board Computer (SBC) berupa raspberry pi 3 B. Bahasa pemrograman yang digunakan adalah Python dengan library OpenCV. Metode yang digunakan untuk mendeteksi gerak adalah mixture of gaussian (MOG). Hasil dari citra yang didapat oleh deteksi gerak akan diunggah ke Dropbox API.

Pada pengujian terhadap 11 video, sistem dapat memproses citra dengan resolusi 320x240. Hasil pengujian menghasilkan nilai blur terbaik sebesar $k = 13$, nilai threshold terbaik sebesar 100 piksel dengan akurasi sebesar 80,3%, dan jarak maksimal sistem dapat mendeteksi objek hewan sejauh 6m. Waktu respon yang didapat untuk sistem memproses frame per detik rata-rata sebesar 0,098 detik, sedangkan untuk mengunggah citra ke dropbox mempunyai rata-rata sebesar 1,618 detik. Hasil pengujian menunjukkan sistem masih mempunyai ruang untuk pengembangan dan perbaikan.

Kata kunci—camera trap, kamera USB, deteksi gerak, OpenCV, SBC

Abstract

USB camera is currently used in daily life for various purposes. On its development, the use of USB camera can be used to create camera traps and can be used to observe the development of animal with integrated systems. In this research, motion detection was used to observe animals online using Single Board Computer (SBC)

Camera trap in this research using Single Board camera in form of raspberry pi 3 B. Python programming language is used with OpenCV library. The method used to detect motion is the Mixture of Gaussian (MOG). The result image gained by motion detection will be uploaded to the dropbox API.

The test performed on 11 videos, the system can process images with 320x240 resolution. The test results show the best blur value of $k = 13$, the best threshold value is 100 pixel with an accuracy of 80,3%, and the maximum distance system can detect animal objects as far as 6m. The response time gained for the system to process frame per second have average of 0,098 seconds, while for uploading image to dropbox has an average of 1,618 seconds. The test result show the system still has room for development and improvement.

Keywords—camera trap, USB camera, Motion detection, OpenCV, SBC

1. PENDAHULUAN

Pengamatan satwa liar langka menggunakan kamera perangkap memungkinkan kita untuk mengumpulkan data aktivitas satwa liar. Saat ini semakin banyak satwa liar yang dikategorikan langka, sehingga diperlukan pemantauan khusus untuk konservasi. Pemantauan satwa liar saat ini dilakukan secara langsung oleh tim konservasi. Hal ini tidak efisien ketika dilakukan pemantauan pada area yang luas. Oleh karena itu, penggunaan kamera perangkap (kamera perangkap) dalam pengamatan satwa liar langka menjadi lebih efisien dalam segi waktu, sumber daya manusia serta dalam segi finansial.

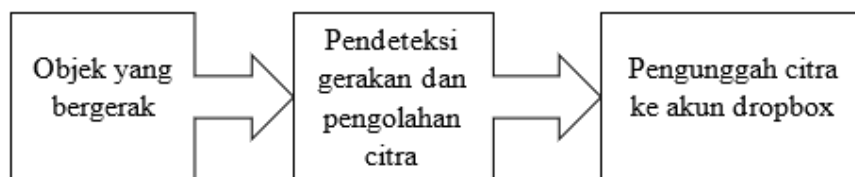
Penelitian yang berkaitan tentang kamera perangkap telah banyak dilakukan baik di dalam maupun luar negeri dari dahulu sampai sekarang. Dengan berkembangnya teknologi pula maka semakin banyak metode yang bisa di implementasikan oleh para peneliti untuk membuat ataupun menunjang sistem kamera perangkap. Penelitian [1] menjelaskan pendeteksian gerakan dan pelacakan hewan berdasarkan *background subtraction* menggunakan perbedaan *frame* dengan *multi threshold* dan morfologi matematika. Dengan metode tersebut dapat digunakan untuk menghitung objek yang bergerak.

Performa *camera trap* dapat ditingkatkan menggunakan DCNN (*Deep Convolutional Neural Network*) dan fitur FV-HOG (*Fisher Vector Histogram of Oriented Network*)[2]. *Camera trap* dapat dirakit menggunakan USB kamera dengan menggunakan *motion detector*[3]. Selain menggunakan USB kamera, *camera trap* dapat menggunakan kamera pada ponsel pintar untuk mendapatkan citra gerakan[4]. *Background motion subtraction* dapat digunakan untuk mendeteksi objek yang bergerak dengan kamera yang dapat bergerak secara bebas[5].

2. METODE PENELITIAN

Sistem deteksi yang digunakan pada penelitian ini merupakan deteksi terhadap gerakan. Deteksi gerakan tersebut menggunakan metode *mixture of gaussian* (MOG). MOG dapat membedakan citra antara *background* dengan citra *foreground* dengan cara pengurangan nilai piksel antara *background* dan *foreground*[6].

2.1 Rancangan Penelitian



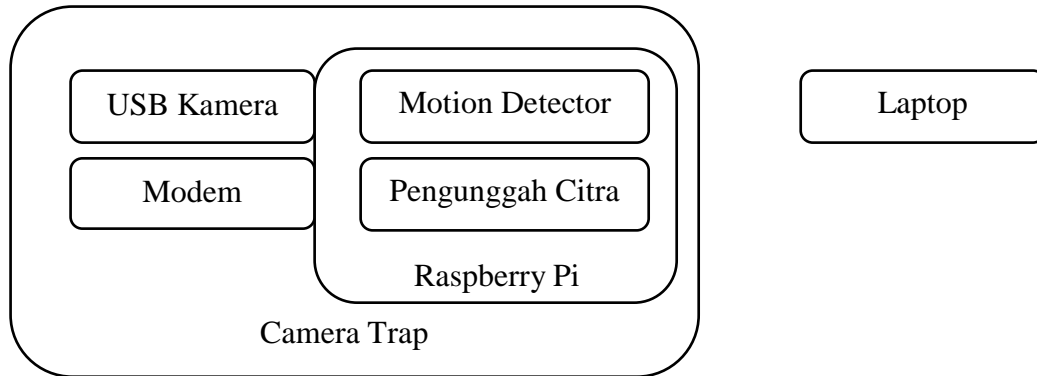
Gambar 1 Diagram blok sistem untuk pendeteksian gerak

Sinyal dari kamera USB akan diproses di Raspberry Pi 3 B, gerakan pada kamera USB akan dideteksi oleh sistem menggunakan program dengan bahasa Python. Python dilengkapi dengan *library* OpenCV (*Open Computer Vision*)[7]. Sistem yang dibangun adalah sistem pemantau lingkungan yang dapat melakukan pengambilan citra secara otomatis saat terdeteksi gerakan oleh hewan dan dapat mengirimkan citra yang diambil ke *dropbox*. Kamera yang digunakan untuk mengambil citra diletakkan pada suatu bidang dalam keadaan tidak bergerak atau diam.

Sistem bekerja secara *real time*, sehingga dapat dipantau secara langsung. Citra yang diambil oleh sistem akan disimpan sebagai citra sementara, sehingga tidak akan membuat penyimpanan pada raspberry pi 3 penuh. Citra kemudian dikirimkan ke dropbox lewat modem

yang ditancapkan pada raspberry pi. Setelah sukses mengunggah citra ke dropbox, citra yang tersimpan sementara akan dihapus secara otomatis agar citra tidak menumpuk pada penyimpanan Raspberry. Rancangan sistem dapat ditunjukkan pada diagram blok rancangan sistem pada Gambar 1.

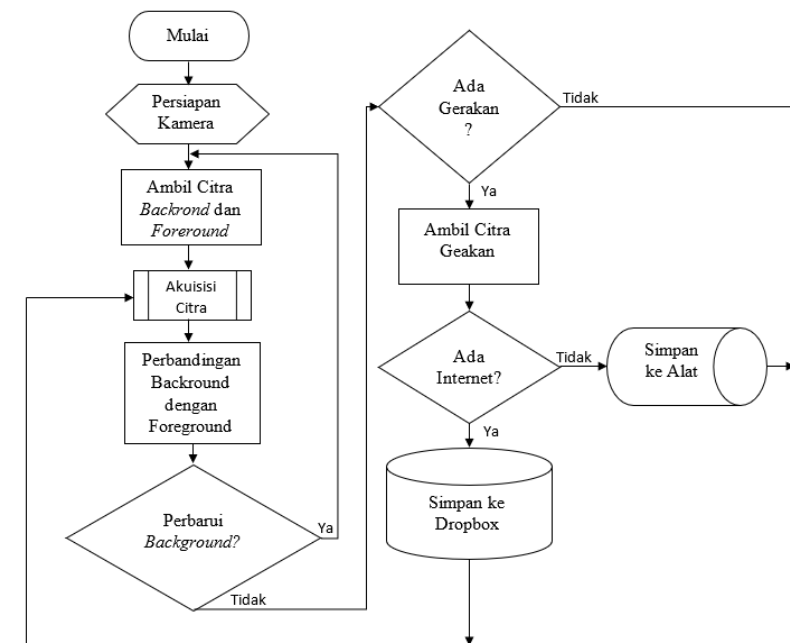
Perangkat keras yang dipakai dalam sistem ini berupa kamera USB dan *single board computer* raspberry pi 3 B. modem dan 1 unit PC sebagai pengakses akun *dropbox*. Kamera USB yang digunakan adalah webcam M-Tech WB 100. Citra di unggah dari raspberry ke *dropbox* menggunakan modem yang tertancap pada raspberry pi. Skema perangkat keras tersebut ditunjukkan pada Gambar 2.



Gambar 2 Rancangan perangkat keras

2.2 Implementasi Perangkat Lunak

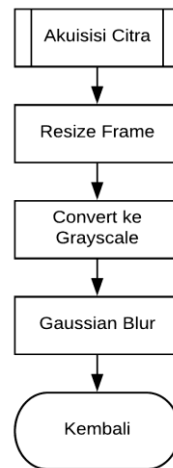
Rancangan perangkat lunak pada penelitian ini dibagi menjadi tiga bagian, yang pertama adalah akuisisi citra, disini citra hasil dari kamera USB akan diubah ukurannya menjadi 320x240. Citra diubah ke citra *grayscale* dan dihaluskan dengan *gaussian blur* untuk mengurangi *noise*[8]. Kedua adalah pendeteksi gerak dengan metode deteksi gerak *mixture of gaussian*. Ketiga adalah sistem untuk mengunggah citra ke *dropbox*. Gambar 3 menunjukkan diagram alir sistem keseluruhan.



Gambar 3 Flowchart sistem keseluruhan

2. 2.1 Implementasi Akuisisi Citra

Tahapan pertama yang dilakukan adalah pengambilan citra video menggunakan kamera USB. Pada proses ini, hewan dianggap sebagai objek utama yang ditangkap kamera dalam sebuah *frame*. Kemudian dilakukan *preprocessing* yaitu *frame* ditangkap berulang-ulang sehingga menghasilkan *video*. Resolusi *video* dalam sistem ditampilkan pada ukuran 480 x 320 piksel dengan pertimbangan agar proses komputasi pengolahan citra dapat berjalan dengan optimal. Citra yang telah diubah ukurannya kemudian diubah dari citra RGB ke citra *grayscale*. Setelah menjadi citra *grayscale* citra dihaluskan menggunakan *gaussian blur*. *Gaussian blur* dapat mengurangi *noise* citra yang tidak diperlukan. Ketiga *preprocessing* di atas dapat dilihat pada Gambar 4.



Gambar 4 *flowchart* akuisisi citra

2. 2.2 Implementasi Pendeteksi Gerak

Deteksi gerak pada penelitian ini menggunakan metode *background subtraction Mixture of Gaussian (MOG)*. Metode *background subtraction MOG* digunakan karena merupakan metode yang dapat memisahkan objek yang bergerak (*foreground*) dan latar belakang (*background*) secara halus dan adaptif terhadap kondisi *frame*. Perbedaan nilai piksel dihitung dengan membandingkan nilai intensitas kedua citra pada koordinat yang sama pada citra skala keabuan.

Pengurangan citra (*background subtraction*) dilakukan dengan cara melakukan pengurangan piksel pada citra dengan objek dengan citra latar belakang. Citra hasil *preprocessing* akan diubah menjadi citra biner dengan cara *thresholding biner*. Citra *background* diubah menjadi citra biner dan akan bernilai 0 dengan warna hitam. Objek pada *foreground* akan bernilai biner 1 ditandai dengan warna putih. Objek yang terdeteksi diproses untuk mencari kontur[9]. Kontur akan menunjukkan besar perubahan piksel antara *background* dan *foreground*. Objek dianggap sebagai gerakan apabila pengurangan dari *background* terhadap *foreground* lebih dari nilai *threshold* yang diatur. Objek dengan perubahan piksel lebih dari nilai *threshold* akan ditandai dengan memberi label berwarna hijau pada area tersebut.

Sistem ini dapat memperbaharui latar belakang citra yang sedang diamati. Objek yang masuk ke citra latar belakang lebih dari 100 *frame* maka akan dianggap sebagai latar belakang baru.

2. 2.3 Implementasi Pengunggah Citra

Saat terdeteksi gerakan sistem akan mengambil citra gerakan tersebut. Setelah citra didapat, citra akan ditempatkan pada penyimpanan alat pada *temporary folder*. Sistem akan mengecek apakah sistem mempunyai akses internet. Jika sistem memiliki akses internet sistem akan otomatis mengunggah citra ke akun *user* pada *dropbox*. Akun *user* diidentifikasi

menggunakan *access token* yang telah diatur pada pengaturan *dropbox*. Penamaan citra disesuaikan dengan waktu pengambilan citra. Setelah selesai diunggah citra pada direktori sementara akan dihapus. Hal ini dilakukan agar penyimpanan pada sistem tidak terlalu besar. Apabila sistem tidak mempunyai akses internet citra yang tersimpan pada direktori sementara akan dipindah ke direktori baru untuk disimpan pada penyimpanan sistem.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Sistem Secara Keseluruhan

Pengujian sistem ini dilakukan untuk menjawab salah satu tujuan penelitian yaitu, Merancang dan membuat sistem pemantauan satwa liar menggunakan Raspberry pi dan metode motion detector. Sistem dapat mendeteksi gerakan hewan dan mengunggah citra hasil gerakan ke *dropbox*. Sistem dapat menjalankan program *camera trap* dan dapat menggkoneksikan internet secara otomatis saat pertama kali dinyalakan. Gambar 5 menunjukkan citra hasil deteksi gerak menggunakan MOG.



Gambar 5 Pengujian deteksi gerak pada sistem

3.2 Hasil Akuisisi Citra

Proses akuisisi citra menghasilkan citra secara individual yang akan menjadi masukan proses selanjutnya. Proses akan mengubah resolusi panjang dari kamera USB menjadi 320. Perubahan tinggi pada frame diubah berdasarkan perbandingan 4 : 3. Semakin rendah resolusi citra yang didapat akan mempercepat proses dalam sistem.

Proses pengolahan warna dengan *grayscale* dilakukan dengan merubah tiga channel warna, yaitu merah, hijau dan biru ke dalam satu *channel* warna berskala keabuan. Penerapan *grayscale* pada penelitian ini memungkinkan proses dilakukan secara cepat, karena channel yang diolah berkurang dari tiga channel menjadi satu channel. Selain itu citra dalam skala keabuan memungkinkan pencarian objek dilakukan lebih sederhana dan mudah.

Proses *gaussian blur* dilakukan untuk mengurangi *noise* pada citra. Penerapan blur dapat mengurangi pergerakan yang tidak diperlukan seperti daun dan ranting yang terkena angin. Pada penelitian ini dilakukan simulasi terhadap 5 buah video untuk mengetahui perbandingan antara nilai kernel gaussian blur terhadap pendeteksian gerak. Tabel 1 menunjukkan simulasi nilai kernel gaussian blur.

Tabel 1 Simulasi nilai kernel gaussian blur











Nilai k	Citra Deteksi Gerak			Jumlah	Error
	TP	FP	FN		
1	123	55	48	226	0,456
5	135	48	51	234	0,423
9	150	47	45	242	0,380
13	135	38	44	217	0,378
17	142	40	66	248	0,427

Berdasarkan pengujian didapat hasil simulasi menggunakan 5 buah *video* dengan menerapkan threshold sebesar 100 piksel. Simulasi menghasilkan nilai error minimal sebesar 0,378 pada nilai $k = 13$. Nilai k yang terlalu kecil akan menyebabkan sistem terlalu sensitif, sehingga walaupun ada sedikit gerakan sistem mendeteksi gerakan dan menghasilkan *false positif* yang tinggi. Sebaliknya nilai k yang terlalu tinggi menyebabkan sistem susah mendeteksi gerakan sehingga menghasilkan *false negative* yang tinggi. Tabel 2 menunjukkan persentase hasil deteksi gerak dengan variasi k pada akuisisi citra menggunakan *noise reduction* pada gaussian *blur*. Tabel 3 menunjukkan perbedaan hasil nilai *blur* pada *video* no 2 dengan tingkat kesalahan yang tinggi.

Tabel 2 Pengaruh hasil deteksi gerak terhadap variasi nilai k

Video ke	k				
	1	5	9	13	17
1	100,00%	100%	100,00%	100,00%	100,00%
2	38,28%	38,33%	38,39%	45,05%	43,24%
3	71,79%	72,50%	80,00%	76,74%	74,42%
4	100,00%	100,00%	100,00%	100,00%	100,00%
5	51,85%	53,33%	44,83%	57,69%	57,69%
Rata-rata	72,38%	72,83%	72,64%	75,90%	75,07%

Tabel 3 Hasil perbedaan nilai *blur*

No	Nilai k	Hasil Deteksi	Hasil Blur
1	1		
2	5		
3	9		
4	13		
5	17		

Berdasarkan pengujian, sistem berhasil mendeteksi gerakan dengan persentase 75,90% pada nilai $k = 13$ yang merupakan nilai tertinggi dan mempunyai error terkecil = 0,378 dari semua sampel yang diuji. Umumnya kesalahan deteksi terjadi karena pergerakan bayangan pada pohon. Nilai *blur* terlalu kecil menyebabkan barisan citra sangat sensitif dinyatakan memiliki gerakan, walaupun kenyataannya hanya perubahan bayangan atau hanya perubahan daun yang bergerak. Nilai *blur* terlalu besar menyebabkan gerakan objek hewan menjadi sulit terdeteksi.

3. 3 Hasil Thresholding Citra

Besar perubahan piksel pada penelitian ini menjelaskan keluaran informasi berupa jumlah citra yang berhasil ditangkap oleh sistem menggunakan 11 *video* dengan variasi nilai perubahan piksel. Proses pengujian dilakukan secara manual dengan mengganti nilai *threshold* perubahan piksel. *Threshold* yang digunakan adalah 50 sampai 500. Pada Tabel 4 ditunjukkan jumlah citra yang berhasil di dapat oleh sistem dengan variasi pengujian *threshold* perubahan piksel.

Tabel 4 Pengaruh Threshold terhadap akurasi

No Video	Threshold (piksel)										
	50	100	150	200	250	300	350	400	450	500	
1	93,75 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 0%
2	37,18 %	36,36 %	38,71 %	36,21 %	42,55 %	44,19 %	50,00 %	51,02 %	51,28 %	55,2 6%	
3	72,00 %	74,36 %	69,44 %	77,42 %	74,19 %	75,86 %	75,00 %	80,95 %	73,91 %	80,9 5%	
4	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 0%	
5	52,00 %	51,85 %	47,83 %	68,57 %	56,00 %	55,00 %	50,00 %	33,33 %	37,50 %	35,7 1%	
6	33,33 %	100,0 %	0,00% %	0,00% %	0,00% %	0,00% %	0,00% %	0,00% %	0,00% %	0,00 %	
7	100,0 %	100,0 %	100,0 %	0,00% %	0,00% %	0,00% %	0,00% %	0,00% %	0,00% %	0,00 %	
8	25,00 %	20,00 %	20,00 %	33,33 %	50,00 %	50,00 %	50,00 %	50,00 %	50,00 %	50,0 0%	
9	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 0%	
10	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 0%	
11	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 %	100,0 0%	
Rata-rata	73,93 %	80,23 %	70,54 %	65,05 %	65,70 %	65,91 %	65,91 %	65,03 %	64,79 %	65,6 3%	

Pengujian deteksi gerak dengan menggunakan variasi *threshold* menunjukkan tingkat akurasi yang berbeda. Nilai akurasi tertinggi adalah 80,23% pada *threshold* 100 piksel. Akurasi didapat dengan perhitungan antara *true positif*, *false positif*, dan *false negatif*. *Threshold* kecil dapat mendeteksi gerakan lebih banyak tetapi, karena terlalu sensitif sistem mendeteksi gerakan yang tidak diperlukan seperti, bayangan dan pergerakan daun. *Threshold* besar menyebabkan sistem sulit untuk mendeteksi objek yang bergerak bila objek berukuran kecil.

3. 4 Hasil Pengunggah Citra

Hasil pengunggah citra dilakukan pada citra yang terdeteksi gerakan oleh sistem. Sistem dapat mengunggah citra apabila waktu terakhir mengunggah citra dikurangi waktu awal citra terdeteksi lebih dari samadengan 3 detik dan terdapat lebih dari 3 *frame* bergerak yang terdeteksi. Pada penelitian ini, pengunggahan akan dihitung waktu eksekusi pada 50 citra yang terunggah. Rangkuman waktu eksekusi program ditunjukkan pada Tabel 5.

Tabel 5 Rangkuman waktu eksekusi program

Waktu	Proses / Frame (detik)	Waktu Upload (detik)	Waktu Sistem (detik)
Minimum	0,024	1,001	1,001
Maksimum	0,210	3,727	3,743
Rata-rata	0,098	1,618	1,633

Beberapa hal yang dihasilkan dan dapat dianalisis dari pengunggah citra adalah waktu sistem untuk memproses *frame* untuk mengetahui adanya gerakan mempunyai rata-rata 0,098 detik antara 0,024 detik sampai 0,210 detik. Sedangkan waktu untuk mengunggah citra yang terdeteksi gerakan memerlukan waktu rata-rata 1,633 detik berkisar antara 3,737 detik sampai 1,001 detik.

Belum diketahui standar batasan yang baku untuk menentukan waktu respon ideal pada sistem pengunggahan citra. Namun demikian, waktu pengunggahan citra rata-rata sebesar 1,611 detik merupakan waktu yang cukup kecil bagi pengguna sistem untuk mengetahui adanya gerakan hewan sedini mungkin.

Waktu yang ideal dapat ditentukan dengan besarnya delay waktu pemrosesan. *Video* yang diuji mempunyai 10 *frame* per detik. Ini artinya waktu ideal untuk memproses *video* tersebut adalah dibawah $1/10=0,1$ detik per *frame*[10]. Pada penelitian ini, waktu yang dibutuhkan untuk mendeteksi gerak rata-rata adalah 0,98 detik. Hal ini dapat dikatakan sistem dapat mendeteksi gerakan secara ideal dibawah 0,1 detik.

3. 5 Hasil Pengujian Jarak

Penelitian ini, diuji jarak maksimal sistem dapat mendeteksi gerakan. Objek yang digunakan adalah hewan kucing sebagai acuan batas bawah. Pada pengujian, diambil 5 sampel *video* per jarak uji. Tabel 6 menunjukkan hasil sistem dapat mendeteksi objek pada jarak tertentu.

Tabel 6 Pengujian jarak maksimum sistem

No Video	Jumlah Terdeteksi						
	2m	3m	4m	5m	6m	7m	8m
1	1	1	1	1	0	0	0
2	1	1	1	1	1	0	0
3	1	1	1	1	1	0	0
4	1	1	0	0	0	0	0
5	1	1	1	1	1	0	0

Beberapa hal yang dihasilkan dan dapat dianalisis dari pengujian jarak maksimal adalah Sistem dapat medeteksi objek pada jarak 2m dan 3m dengan akurasi 100%, sedangkan untuk jarak maksimal sistem dapat mendeteksi objek hewan kucing adalah pada jarak 6m.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, telah dibuat sebuah sistem pemantauan hewan diluar ruang menggunakan raspberry pi 3 sebagai pemroses deteksi gerak. Citra yang terdeteksi gerak akan diunggah ke *dropbox* API. Sistem dapat memproses per *frame* dengan waktu respon sebesar 0,098 detik pada resolusi 320 x240, nilai blur sebesar 13 x13 dan *threshold* sebesar 100 piksel. Sistem dapat memperbarui *background* setiap perubahan 100 *frame*, serta dapat mendeteksi hewan kucing sebagai batas minimal pada jarak 6m.

5. SARAN

Pada penelitian ini terdapat beberapa hal yang perlu disempurnakan. Saran-saran yang dapat diunakan untuk menyempurnakan penelitian selanjutnya seperti membuat sistem *camera trap* menggunakan metode lainnya sehingga bisa menjadi pembanding. Kemudian diperlukan sistem yang dapat memproses citra dengan resolusi lebih besar dengan waktu lebih cepat untuk mendapat hasil citra lebih bagus.

DAFTAR PUSTAKA

- [1] E. Surendar, V. M. Thomas, and A. M. Psonia, 2016, Animal tracking using background subtraction on multi threshold segmentation, *Proc. IEEE Int. Conf. Circuit, Power Comput. Technol. ICCPCT 2016*, no. Ccd. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7530223> [Accessed: 30-Mei-2017].
- [2] Z. ZHANG, Z. He, G. Cao, and W. Cao, 2016, Animal Detection from Highly Cluttered Natural Scenes Using Spatiotemporal Object Region Proposals and Patch Verification, *IEEE Trans. Multimed.*, vol. 9210, no. c, pp. 1–1. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7523423> [Accessed: 30-Mei-2017]
- [3] O. Krejcar, 2013, Motion Detection Using a USB Camera, pp. 281–286. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6632827> [Accessed: 30-Mei-2017].
- [4] A. Mahabalagiri, K. Ozcan, and S. Velipasalar, 2014, Camera motion detection for mobile smart cameras using segmented edge-based optical flow, *11th IEEE Int. Conf. Adv. Video Signal-Based Surveillance, AVSS 2014*, pp. 271–276. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6918680> [Accessed: 30-Mei-2017].
- [5] Y. Wu, X. He, and T. Nguyen, 2015, Moving Objects Detection with Freely Moving Camera via Background Motion Subtraction, *IEEE Trans. Circuits Syst. Video Technol.*, vol. PP, no. 99, pp. 1–1. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7303924> [Accessed: 30-Mei-2017].
- [6] F. Amaluddin, M. A. Muslim, dan A. Naba, "Klasifikasi Kendaraan Menggunakan Gaussian Mixture Model (GMM) dan Fuzzy Cluster", *Jurnal EECCIS* , Vol. 9, no. 1,p. 19–24, 2015 [online]. Available: <http://jurnaleeccis.ub.ac.id/index.php/eccis/article/view/269>. [Accessed: 30-Mei-2017]
- [7] S. Brahmhatt, 2013 Embedded Computer Vision: Running OpenCV Programs on the Raspberry Pi, *Pract. OpenCV*, no. 9, vol. 53, 201–218. [online]. Available: <http://www.bookmetrix.com/detail/chapter/6e671f63-fb9f-4fed-ae7d-dcb14a5970da#downloads> [Accessed: 30-Mei-2017]

-
- [8] S. Manchanda, Analysis of Computer Vision based Techniques for Motion Detection, *International Conference - Cloud System and Big Data Engineering (Confluence)*, No 6, 445–450, 2016 [Online]. Available:<http://ieeexplore.ieee.org/document/7508161/>. [Accessed : 30-mei-2017]
- [9] G. Braedski, dan A. Kaehler, 2008, Learning OpenCV , O'Reilly Media Inc, Gravenstein Highway North [Online]. Available: <http://www.bogotobogo.com/cplusplus/files/OReilly%20Learning%20OpenCV.pdf> [Accessed: 30-Mei-2017]
- [10] A. Nurhopipah, 2017, Deteksi Gerak dan Pengenalan Wajah untuk Sistem Pengawasan Melalui Video CCTV, *Tesis*, Program Studi S2 Ilmu Komputer, Univ. Gadjah Mada, Yogyakarta.