

Penggunaan Deteksi Gerak untuk Pengurangan Ukuran Data Rekaman Video Kamera CCTV

Jockie P. Sagala*¹, Ika Candradewi², Agus Harjoko³

¹Prodi Elektronika dan Instrumentasi, DIKE, FMIPA UGM, Yogyakarta, Indonesia

^{2,3}Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *¹jockysagala@gmail.com, ²aharjoko@ugm.ac.id ³ika.candradewi@ugm.ac.id

Abstrak

Dalam kasus tertentu data rekaman dari Closed Circuit Television (CCTV) di simpan untuk keperluan di masa mendatang. Dalam jangka waktu yang lama, maka diperlukan media penyimpanan data yang besar karena ukuran data citra yang besar. Pada beberapa kasus, data citra terekam citra dengan objek yang diinginkan tapi juga latar citra yang mungkin terekam dalam jangka waktu yang lama sewaktu perekaman citra. Kasus seperti ini membuat penggunaan media penyimpanan data kurang efektif. Pada penelitian ini dirancang sebuah sistem pada perangkat CCTV yang mampu memilih citra untuk mengurangi ukuran berkas data citra yang disimpan dengan pengolahan citra.

Pemilihan data dilakukan berbasis pendeteksian objek pada citra dengan metode adaptive median pada ruangan tertutup dengan penerangan yang cukup. Pada saat ada objek terdeteksi, maka yang akan disimpan adalah citra masukan sistem. Sebaliknya, jika tidak ada objek terdeteksi maka yang data akan di simpan ke berkas keluaran sistem adalah citra model latar.

Hasil yang didapat dari penelitian ini berupa sistem dengan kamera CCTV yang mampu memilih data perekaman yang disimpan dengan pendeteksian objek dengan pemrosesan citra. Model latar terus diperbaharui untuk dapat beradaptasi dengan perubahan kondisi data visual latar citra.

Kata kunci— OpenCV, Seleksi objek, CCTV

Abstract

Some cases the recording data of Closed Circuit Television (CCTV) is stored for future use. In the long term usage, the files size will grow larger and requiring large storage devices. In some cases, the recorded data not only image with the desired object but also the background images that may be recorded for long periods of time. This cases make data storage device usage to be less effective. So this research will design a system of CCTV devices that capable to select images to reduce the size of stored images data by image processing.

The images selection of this system is based on based on adaptive median algorithm. When any object get detected, the images data to be saved is current input frame. Otherwise, the data to be saved is background model image. Background model on this system is adapted with the change visual data of background image.

The results obtained from this research in the form of a CCTV system that are able to select recording data to be stored with image processing. The background model will be kept adapting with background visual data changes.

Keywords— OpenCV, Objects selection, CCTV

1. PENDAHULUAN

CCTV merupakan perangkat pengawasan yang cukup dikenal masyarakat. Salah satu penggunaannya adalah untuk membantu penyidik mengungkap terjadinya tindak pidana [1]. Dalam kasus tertentu data rekaman perangkat *Closed Circuit Television* (CCTV) di simpan untuk keperluan di masa mendatang. Dalam jangka waktu yang lama, diperlukan media penyimpanan data yang besar karena ukuran data citra yang besar. Pada beberapa kasus, data citra terekam citra dengan objek yang diinginkan tapi juga latar citra yang mungkin terekam dalam jangka waktu yang lama sewaktu perekaman citra. Kasus seperti ini membuat penggunaan media penyimpanan data kurang efektif.

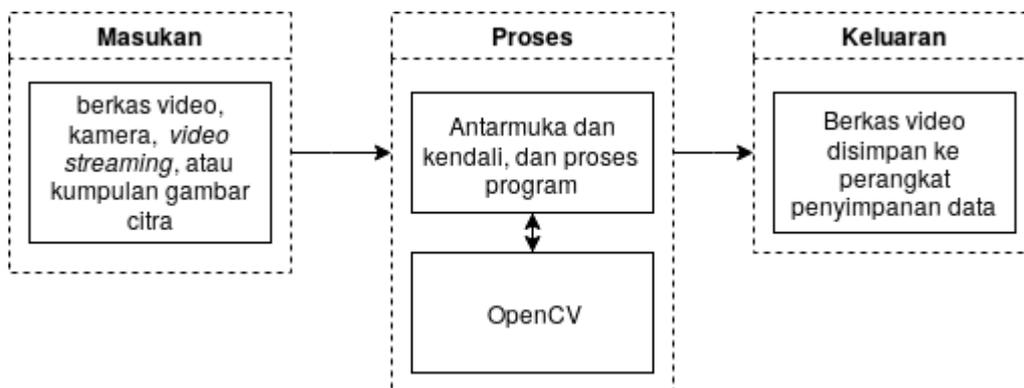
Perlengkapan perangkat lunak saat ini sudah semakin mudah didapat dan digunakan. Pustaka (*library*) OpenCV (*Open Source Computer Vision Library*) adalah salah satu pustaka untuk pemrosesan citra digital yang dirilis dengan lisensi BSD. Beberapa penelitian telah dilakukan terkait metode pendeteksian objek dan substraksi latar pada citra antara lain dengan perbedaan frame [2], pendekatan statistik, dan lain-lain. Implementasi pendeteksian objek pada citra juga diimplementasikan untuk berbagai hal, antara lain Purwarupa Pengoperasian TV Dengan Mendeteksi Isyarat Jari Berbasis Pengolahan Citra Digital [3], Pengolahan Citra Digital untuk Keyboard Virtual Sebagai Antarmuka pada Aplikasi Berbasis Web [4] dan lain-lain.

Penelitian ini dilakukan untuk merancang sistem perekaman kamera CCTV yang mampu memilih objek untuk mengurangi ukuran berkas hasil perekaman deteksi objek menggunakan metode pendeteksian objek efisien sesuai kondisi lingkungan yang diuji. *Frame* yang akan disimpan pada saat tidak ada objek terdeteksi adalah model citra sistem dan sebaliknya, jika ada objek yang terdeteksi maka akan disimpan frame yang didapat dari kamera.

2. METODE PENELITIAN

2.1 Analisis Sistem

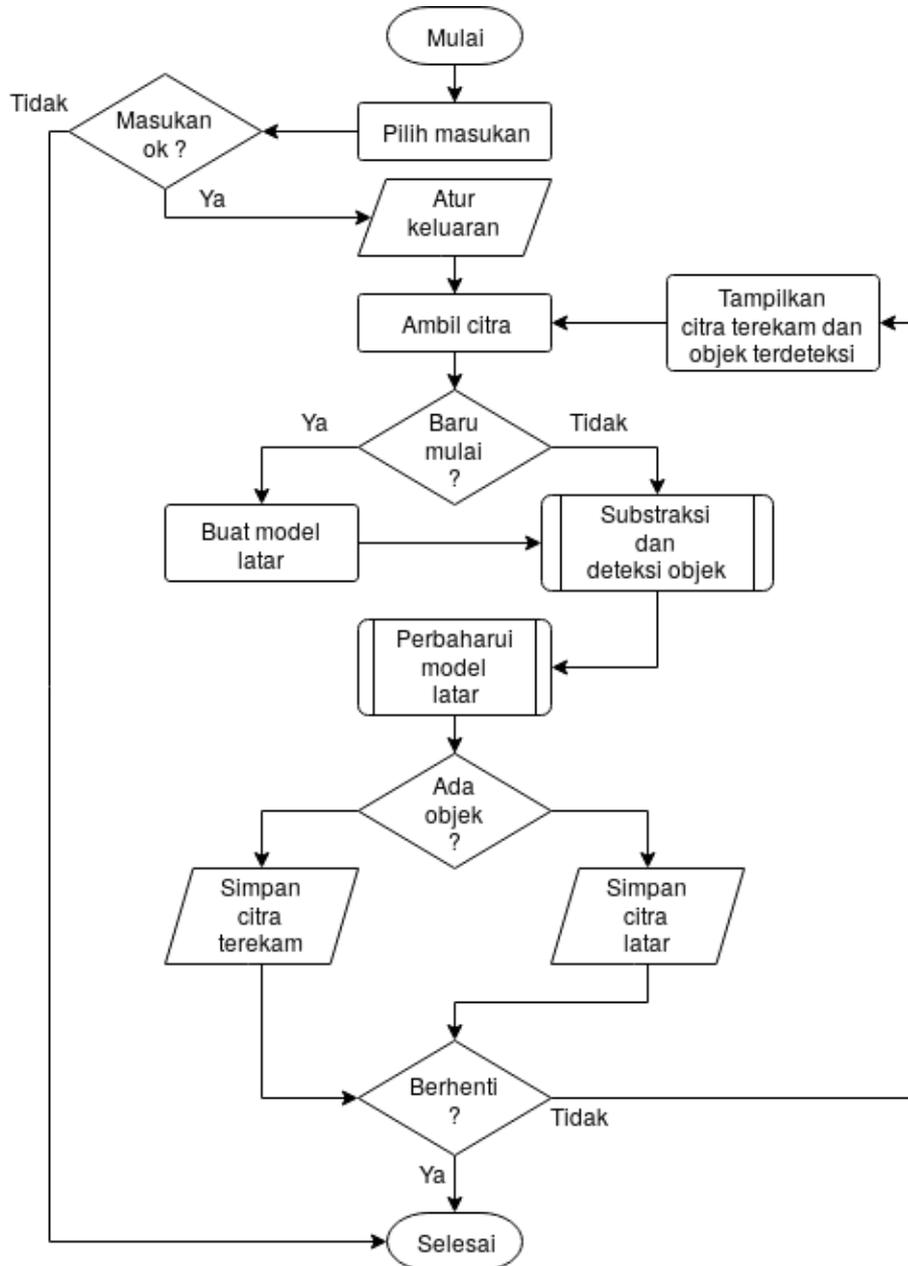
Sistem pemilihan data pada perangkat kamera CCTV dengan pendeteksian objek dilakukan dengan menggunakan pengolahan citra digital. Pengolahan citra digital pada sistem dilakukan merancang perangkat lunak yang menggunakan pustaka (*library*) OpenCV versi 3.4 dengan menggunakan bahasa pemrograman C++. Antarmuka program dirancang untuk memudahkan penggunaan program. Diagram sistem secara keseluruhan dimuat pada Gambar 1. Proses dimulai dengan masukan data visual dikirim ke laptop untuk melakukan pemrosesan dengan perangkat lunak sistem. Saat ada objek terdeteksi, citra yang akan disimpan adalah citra masukan, sedangkan saat tidak ada objek yang terdeteksi maka citra yang disimpan adalah model latar citra pada sistem. *Video codec* mengompres video untuk menghasilkan berkas video yang lebih dipadatkan dengan ukuran data yang lebih kecil [5].



Gambar 1 Diagram sistem keseluruhan

2. 2 Rancangan Perangkat Lunak

Dengan menggunakan masukan data visual yang diterima secara terus-menerus dilakukan pendeteksian objek dan adaptasi model latar sistem. Rancangan perangkat lunak sistem dimuat pada Gambar 2.



Gambar 2 Diagram rancangan perangkat lunak

2. 2.1 Rancangan Antarmuka

Antarmuka perangkat lunak sistem dirancang sedemikian rupa sehingga dapat mempermudah penggunaannya. Pada antarmuka pengguna dapat menentukan masukan memulai dan menghentikan program. Antarmuka juga menyediakan masukan untuk masukan berkas video, direktori untuk kumpulan gambar citra dan *video streaming*. Untuk berkas keluaran perangkat lunak keluaran, pengguna dapat menentukan direktorinya.

2. 2.2 Deteksi Objek

Beberapa komponen yang diperlukan untuk proses pendeteksian disiapkan sebelum pada saat program dijalankan pertama kali. Komponen ini antara lain, nilai *threshold* tinggi dan rendah, model latar citra, dan masukan citra. Program dapat memproses masukan citra dapat berupa *video streaming* dari jaringan tertutup, kumpulan gambar citra berurut, berkas video dan webcam pada laptop. Masukan ini akan dibuka dan diperiksa pada saat program baru berjalan.

Proses pendeteksian dimulai dengan menghitung selisih citra masukan terhadap model citra. Nilai selisih ini dibandingkan dengan nilai *threshold* tinggi. Dari perbandingan ini didapati objek merupakan sebuah atau kumpulan *pixel* (elemen citra) yang nilai absolut ketiga komponen warnanya yaitu nilai citra warna merah, biru dan hijau melebihi nilai *threshold* tinggi. Pada saat ada objek yang terdeteksi maka citra yang disimpan adalah citra yang diterima dari masukan sistem. Sedangkan saat tidak ada objek yang terdeteksi maka citra yang akan disimpan adalah model citra sistem yang dideklarasikan pada awal proses dan terus diperbaharui selama program tetap berjalan.

Adaptasi model latar citra berlangsung setelah proses pendeteksian. Nilai selisih citra pada proses sebelumnya yang kemudian dipilih dengan dibandingkan dengan nilai *threshold* rendah. Elemen citra yang nilainya akan diperbaharui adalah elemen citra yang nilai ketiga elemen warnanya kurang dari atau sama dengan nilai *threshold* rendah. Elemen citra yang terpilih digunakan sebagai *mask* citra untuk memilih elemen citra pada model latar yang akan diperbaharui. Komponen dari elemen citra ini dibandingkan dengan citra yang diterima dari masukan citra. Nilai komponen pada model citra yang lebih besar terhadap komponen pada elemen citra masukan akan ditambah satu. Sebaliknya untuk nilai komponen pada model citra yang lebih kecil terhadap komponen pada elemen citra masukan akan dikurangi satu. Proses ini dijalankan pada rasio yang ditentukan dari jumlah citra yang diproses.

2. 2.3 Penyimpanan Hasil Perekaman

Hasil perekaman akan disimpan sebagai berkas video pada direktori yang dipilih pada masukan *Hasil* pada antarmuka program. Penamaan berkas dibuat secara otomatis pada program dengan menggunakan waktu yang berjalan pada sistem operasi dengan ekstensi file *.avi*. *Video codec* yang digunakan pada program ini adalah *x264* yang merupakan implementasi dari *H.264*. Dengan menyimpan model latar pada saat tidak ada objek terdeteksi, model latar mengurangi perbedaan piksel antar citra. Dengan berkurangnya perbedaan piksel antar citra, ukuran berkas keluaran yang sudah dikompres menjadi semakin kecil.

2. 3 Perbandingan Metode Sistem dengan Metode Lain

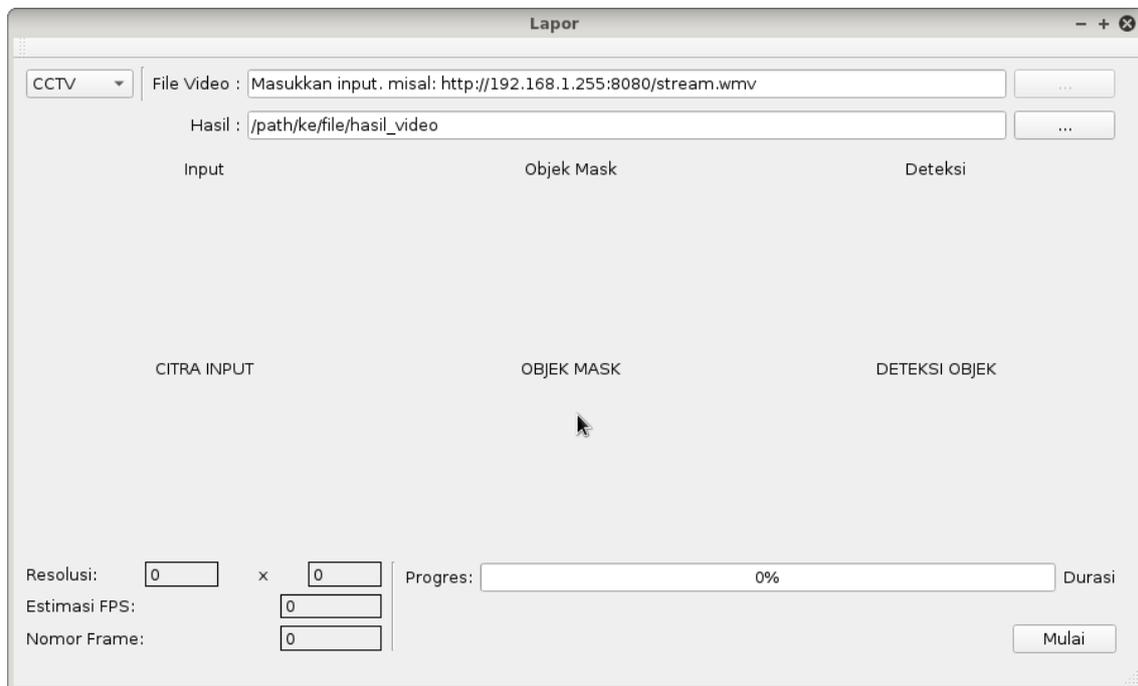
Kondisi lingkungan untuk sistem ini dibatasi yaitu pada ruangan tertutup, penerangan yang cukup dan keadaan kamera yang tetap. Dengan kondisi ini metode pendeteksian objek yang digunakan sistem akan dibandingkan dengan metode lain dengan menggunakan dataset substraksi latar citra [6] yang dipilih yaitu *CopyMachine*, *Sofa* dan *PETS2006*. Dataset yang digunakan menyediakan kumpulan citra dari data visual terekam dan citra abu-abu yang merupakan citra *mask* dari objek. Metode yang dibandingkan dengan metode pendeteksian objek pada sistem adalah 5 algoritma terbaik global [7].

Komponen evaluasi perbandingan ini diambil dari nilai *TP*, *TN*, *FP* dan *FN*. *TP* (*True Positive*) yaitu klasifikasi piksel (elemen citra) objek pada metode sesuai dataset, *TN* (*True Negative*) merupakan klasifikasi piksel latar pada metode dan sesuai dengan dataset. *FP* (*False Positive*) merupakan kesalahan klasifikasi pada metode yang dimana metode mendeteksi piksel sebagai objek. *FN* (*False Negative*) adalah kesalahan klasifikasi objek dimana metode mendeteksi piksel sebagai latar sedangkan pada dataset piksel merupakan piksel objek [8]. Dari nilai-nilai ini dihitung komponen evaluasi antara lain:

1. *Recall (Re)*: $TP/(TP+ FN)$
2. Spesifitas (*SP*): $TN/(TN + FP)$
3. Rasio *False Positive (FPR)*: $FP/(FP+TN)$
4. Rasio *False Negative (FNR)*: $FP/(FP+TN)$
5. Persentase kesalahan klasifikasi (*PWC*): $100(FP+TN)/(TP+TN + FP+FN)$
6. Presisi (*Pr*): $TP/(TP+ FP)$
7. F-measure (skor *FI*): $2((Pr.Re)/(Pr+ Re))$

3. HASIL DAN PEMBAHASAN

Antarmuka program dimuat pada Gambar 3. Pada antarmuka pengguna dapat menentukan masukan program. Selain masukan dengan menggunakan *webcam*, pengguna perlu memasukkan string tambahan terkait pilihan masukan program. Untuk *video streaming*, akan disediakan masukan untuk URL dari *video streaming*, Saat pengguna memilih kumpulan gambar pengguna dapat memilih direktori yang berisi kumpulan gambar. Untuk masukan berkas video, pengguna dapat memilih berkas. Program dapat dijalankan setelah menentukan masukan dengan menekan (mengklik) tombol *Mulai* pada program dan tombol *Berhenti* untuk menghentikan program. Pengguna dapat menentukan direktori untuk menyimpan hasil perekaman pada masukan *Hasil*.



Gambar 3 Rancangan antarmuka program

Pengujian pertama kali dilakukan untuk mendapati perbandingan ukuran berkas video pada 5 kondisi yang berbeda yang ditentukan dengan 3 sampel video untuk setiap kondisi. Kondisi pertama yaitu sampel video tanpa objek. Hasil dari percobaan ini termuat pada Tabel 1. Sistem menghasilkan pengurangan ukuran berkas paling baik pada kondisi ini, dimana semua citra yang disimpan adalah model latar sehingga menghasilkan sedikit perubahan nilai piksel antar citra berurutan. Hal ini membuat *video codec* mengompres berkas video lebih padat.

Tabel 1 Perbandingan ukuran berkas video pada kondisi sampel pertama

Sampel Ke	Ukuran Video Hasil Pengujian		Rata-rata <i>FPS</i>
	Dengan Sistem (<i>bytes</i>)	Tanpa Sistem (<i>bytes</i>)	
1	958.454	5.489.442	23
2	985.156	2.772.350	23
3	772.252	2.293.700	24

Kondisi pada sampel video selanjutnya adalah dengan durasi objek yang dideteksi pada video kurang dari 50 persen dari durasi keseluruhan. Hasil percobaan ini dimuat pada Tabel 2.

Tabel 2 Perbandingan ukuran berkas video pada kondisi sampel kedua

Sampel Ke	Ukuran Video Hasil Pengujian		Rata-rata <i>FPS</i>	Jumlah Frame dengan Objek
	Dengan Sistem (<i>bytes</i>)	Tanpa Sistem (<i>bytes</i>)		
1	1.806.198	5.629.936	21	173
2	2.208.912	3.478.302	23	370
3	1.563.430	3.485.852	23	247

Kondisi pada sampel video selanjutnya adalah dengan durasi objek yang dideteksi pada video lebih dari 50 persen dan kurang dari 100 persen dari durasi keseluruhan. Hasil dari percobaan ini dimuat pada Tabel 3.

Tabel 3 Perbandingan ukuran berkas video pada kondisi sampel ketiga

Sampel Ke	Ukuran Video Hasil Pengujian		Rata-rata <i>FPS</i>	Jumlah Frame dengan Objek
	Dengan Sistem (<i>bytes</i>)	Tanpa Sistem (<i>bytes</i>)		
1	3.481.316	5.115.318	22	554
2	3.890.960	4.124.976	23	890
3	2.822.412	3.480.584	23	716

Sampel video selanjutnya yaitu dengan kondisi objek baru yang akan diperbaharui menjadi model latar. Hasil dari percobaan ini dimuat pada Tabel 4. Dengan durasi waktu yang cukup objek yang diam akan menjadi bagian dari data visual model latar sistem. Perubahan model latar pada percobaan ini pada awal dan akhir perekaman dimuat pada Tabel 5.

Tabel 4 Perbandingan ukuran berkas video dengan pembaharuan model latar

Sampel Ke	Ukuran Video Hasil Pengujian		Rata-rata <i>FPS</i>	Jumlah Frame dengan Objek
	Dengan Sistem (<i>bytes</i>)	Tanpa Sistem (<i>bytes</i>)		
1	3.188.744	5.332.756	22	509
2	2.351.238	2.822.562	23	743
3	2.729.810	3.026.286	23	986

Tabel 5 Perubahan model latar citra

Sampel No.	Model Latar Awal	Model Latar Akhir
1		
2		
3		

Kondisi pada sampel video selanjutnya adalah dengan keadaan objek yang diam. Hasil dari percobaan ini dimuat pada Tabel 6.

Tabel 6 Perbandingan ukuran berkas video dengan keadaan objek diam

Sampel Ke	Ukuran Video Hasil Pengujian		Rata-rata FPS	Jumlah Frame dengan Objek
	Dengan Sistem (<i>bytes</i>)	Tanpa Sistem (<i>bytes</i>)		
1	3.903.930	4.580.760	22	845

Sampel Ke	Ukuran Video Hasil Pengujian		Rata-rata FPS	Jumlah Frame dengan Objek
	Dengan Sistem (<i>bytes</i>)	Tanpa Sistem (<i>bytes</i>)		
2	2.787.892	3.237.886	23	789
3	3.342.410	3.368.646	23	858

Pengujian selanjutnya dilakukan untuk membandingkan nilai komponen evaluasi substraksi latar citra terhadap dataset yang ditentukan. Komponen evaluasi ini antara lain *Recall (Re)*, Spesifitas (*Sp*), *False Negative Rate (FNR)*, *False Positive Rate (FPR)*, *Percentage of Wrong Classification (PWC)*, Presisi (*Pr*) dan *F-measure*. Untuk mendapatkan nilai-nilai evaluasi ini dihitung nilai FPS (Banyak frame yang diproses per detik), TP, TN, FP dan FN yang dapat dilihat pada Tabel 7.

Tabel 7 Nilai FPS, TP, TN, FP dan FN tiap algoritma pada ketiga dataset

Algoritma	Dataset	FPS	TP	TN	FP	FN
Algoritma pada sistem	CopyMachine	22	9.032	268.681	2.832	11.268
	PETS2006	18	2.037	305.541	53	1.986
	Sofa	76	688	58.767	79	1.998
<i>Rata-rata</i>		38,67	3.919	210.996,33	988	5.084
Adaptive Background Learning	CopyMachine	56	9.708	255.630	15.883	10.593
	PETS200	40	2.774	300.310	5.284	1.248
	Sofa	192	967	57.282	1.564	1.719
<i>Rata-rata</i>		96	4.483	204.407,33	7.577	4.520
DPWrenGA	CopyMachine	11	11.076	264.875	6.638	9.224
	PETS2006	8	2.697	305.118	476	1.325
	Sofa	44	923	58.646	200	1.763
<i>Rata-rata</i>		21	4.898,67	209.546,33	2.438	4.104
LBAdaptive SOM	CopyMachine	3	18.633	264.298	7.215	1.668
	PETS2006	2	3.700	304.372	1.222	322
	Sofa	16	2.183	58.533	313	503
<i>Rata-rata</i>		7	8.172	209.067,67	2.916,67	831
MultiLayer	CopyMachine	1	7.655	270.902	611	12.645
	PETS2006	1	2.929	305.435	158	1.093
	Sofa	5	909	58.800	46	1.777
<i>Rata-rata</i>		2,33	3.831	211.712,33	271,67	5.171,67
PixelBased Adaptive Segmenter	CopyMachine	0	9.703	269.072	2.441	10.598
	PETS2006	3	2.750	304.693	901	1.272
	Sofa	0	1.011	58.697	149	1.674
<i>Rata-rata</i>		1	4.488	210.820,67	1.163,67	4.514,67

Pada Tabel 7 dievaluasi perbandingan kecepatan pengolahan citra per detik (*FPS*) pada setiap algoritma. Dapat dilihat bahwa algoritma *LBAdaptiveSOM*, *MultiLayer* dan *Pixel Based Adaptive Segmenter* memproses gambar sangat lambat. Pada sistem kamera CCTV yang membutuhkan kecepatan pendeteksian metode pendeteksian yang digunakan pada sistem lebih unggul algoritma tersebut. Perhitungan nilai-nilai komponen evaluasi dimuat pada Tabel 8.

Tabel 8 Hasil perhitungan komponen evaluasi

Algoritma	Re (%)	Sp (%)	FPR (%)	FNR (%)	PWC (%)	Pr (%)	F-measure (%)
Algoritma sistem	43,53	99,53	0,47	2,40	2,7477	79,87	56,35
<i>Adaptive Background Learning</i>	49,79	96,43	3,57	2,13	5,4741	37,17	42,57
<i>DPWrenGA</i>	54,41	98,85	1,15	1,94	2,9604	66,77	59,96
<i>LBAdaptiveSOM</i> [9]	90,77	98,62	1,38	0,39	1,6959	73,70	81,35
<i>MultiLayer</i> [10]	42,55	99,87	0,13	2,44	2,4632	93,38	58,46
<i>PixelBased Adaptive Segmenter</i>	49,85	99,45	0,55	2,13	2,5695	79,41	61,25

Dari hasil pengujian didapati algoritma pada sistem memiliki nilai spesifitas tertinggi dengan nilai 99,53% dan nilai rasio kesalahan pendeteksian objek terendah dengan nilai 0,47%. Pada perbandingan dengan algoritma *Adaptive Background Learning* dan *DPWrenGA*, algoritma sistem unggul pada dengan nilai rasio *False Positive* 0,47%, nilai presisi tertinggi yaitu 79,87% dan nilai Persentase kesalahan klasifikasi terendah yaitu 2.74.

4. KESIMPULAN

Dari penelitian yang dilakukan didapati bahwa sistem pendeteksian objek pada data visual mampu mendeteksi objek tingkat presisi mencapai 79,87% dengan kecepatan pemrosesan citra yang cepat dengan rata-rata sekitar 38,67 citra per-detik. Sistem juga mampu beradaptasi dengan perubahan pada latar citra. Dengan menggunakan sistem sebagai perangkat kamera CCTV, semakin sedikit objek yang terdeteksi oleh sistem, semakin efektif penggunaan sistem karena semakin kecil ukuran berkas video yang dihasilkan.

5. SARAN

Penelitian ini dapat dikembangkan dengan merumuskan metode untuk pendeteksian objek dengan latar yang dinamis yang banyak dijumpai pada ruang terbuka. Untuk mendapatkan hasil yang lebih baik pengembangan juga dapat dilakukan untuk meminimalisir pembaharuan model latar pada objek yang ditentukan.

DAFTAR PUSTAKA

- [1] M. Basori, dan I. Navianto dan Zakaria, 2013, "Penggunaan Data yang Diperoleh dari Rekaman *Closed Circuit Television* (CCTV) dalam Membantu Penyidik Mengungkap Terjadinya Tindak Pidana", Fakultas Hukum, Universitas Brawijaya, Malang [Online]. Available:

- <http://hukum.studentjournal.ub.ac.id/index.php/hukum/article/download/334/328>
[Accessed: 29 May 2018].
- [2] N. Singla, 2014, "Motion Detection Based on Frame Difference Method, *International Journal of Information and Computation Technology*", vol. 15, no. 4, pp. 1559-1565 [Online]. Available: https://www.ripublication.com/irph/ijict_spl/ijictv4n15spl_10.pdf [Accessed: 3 May 2018].
- [3] H. O. Wibawa, A. Harjoko, 2015, "Purwarupa Pengoperasian TV Dengan Mendeteksi Isyarat Jari Berbasis Pengolahan Citra Digital", *IJEIS*, vol. 5, no. 1, pp. 31-42 [Online]. Available: <https://doi.org/10.22146/ijeis.7151> [Accessed: 30 May 2018].
- [4] R. I. Alhaqq, A. Harjoko, 2015, "Pengolahan Citra Digital untuk Keyboard Virtual Sebagai Antarmuka pada Aplikasi Berbasis Web", *IJEIS*, vol. 5, no. 2, pp. 111-122 [Online]. Available: <http://dx.doi.org/10.22146/ijeis.7634> [Accessed: 30 May 2018].
- [5] X. Niu and J. Fan, "System-on-a-Chip (SoC) Based Hardware Acceleration for Video Codec", *Optics and Photonics Journal*, vol. 3 no. 2B, 2013, pp. 112-117 [Online]. Available: <http://dx.doi.org/10.4236/opj.2013.32B028> [Accessed: 30 May 2018]
- [6] W. Yi, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, 2014, "CDnet 2014: An Expanded Change Detection Benchmark Dataset", in Proc. IEEE Workshop on Change Detection (CDW-2014) at CVPR-2014, pp. 387-394 [Online]. Available: http://www.cv-foundation.org/openaccess/content_cvpr_workshops_2014/W12/papers/Wang_CDnet_2014_An_2014_CVPR_paper.pdf [Accessed: 16-May-2018]
- [7] A. Sobral dan A. Vacavant, 2014, "A Comprehensive Review of Background Subtraction Algorithms Evaluated with Synthetic and Real Videos". [Online]. Available: https://www.researchgate.net/publication/259340906_A_comprehensive_review_of_background_subtraction_algorithms_evaluated_with_synthetic_and_real_videos. [Accessed: 16-May-2018]
- [8] D. M. Saidou, 2017, "Tracking a Moving Objects Using Foreground Detector and Improved Morphological Filter", *Open Access Library Journal*, vol. 4, no. 12, pp. 1-7 [Online]. Available: <https://doi.org/10.4236/oalib.1104152> [Accessed: 30 May 2018]
- [9] L. Maddalena and A. Petrosino. "The SOBS algorithm: what are the limits?" In IEEE Workshop on Change Detection, vol. 17, no. 7, July 2008, pp 1168-1177 [Online]. Available: <https://dl.acm.org/citation.cfm?id=2321600> [Accessed: 12-May-2018]
- [10] J. Yao dan J.-M. Odobez, "Multi-Layer Background Subtraction Based on Color and Texture", 2007, IDIAP Research Institute, Rue du Simplon 4, 1920 Martigny, Switzerland [Online]. Available: <http://www.idiap.ch/~odobez/human-detection/doc/YaoOdobezCVPR-VS2007.pdf>. [Accessed: 12-May-2018]