

Deteksi Gerakan pada Kamera CCTV dengan Algoritma *Frame Difference* dan *Frame Substraction*

Reza Arby Yuha¹, Mhd Danu Al Fiqri², Ashari³, Redo pratama⁴, *Mawaddah Harahap⁵.
Jurusan Teknik informatika, UNIVERSITAS PRIMA INDONESIA Medan
e-mail: mawaddah@unprimdn.ac.id

Abstrak

Dewasa ini, CCTV digunakan sebagai kamera pengawas untuk mengawasi pergerakan suatu objek. Kebanyakan fungsinya hanya sebatas mengawasi secara random dan informasi yang kurang detail, hal tersebut cukup menyulitkan jika membutuhkan suatu informasi detail seperti plat nomor kendaraan, pejalan kaki, dan objek lainnya. Untuk menyelesaikan permasalahan tersebut, dapat diterapkan konsep deteksi gerakan. Deteksi gerakan digunakan untuk mengidentifikasi dan memilah objek yang diinginkan. Proses pendeteksian gerakan ini menggunakan konsep pengolahan citra dengan algoritma *Frame Difference* dan *Frame Substraction*. Algoritma *Frame Difference* digunakan untuk memeriksa objek-objek pada citra. Apabila objek terdeteksi, maka diterapkan algoritma *Frame Substraction* untuk mendeteksi posisi objek dimana data posisi dikumpul menjadi gerakan. Hasil penelitian ini cukup mampu mendeteksi objek dan pergerakannya, berdasarkan hasil 25 pengujian dari video yang berbeda, presentase keberhasilannya mencapai 76,0%. Kesimpulan adalah, Metode yang digunakan cukup berhasil mendeteksi pergerakan objek dalam video berdasarkan perbedaan (*difference*) yang diperoleh dari pengurangan *frame* sebelumnya dan *frame* selanjutnya.

Kata Kunci: deteksi gerakan, pengolahan citra, *frame difference*, *frame subtraction*, Kamera CCTV

Abstract

Nowadays, CCTV is used as a surveillance camera to monitor the movement of an object. Most of its functions are limited to monitoring random and less detailed information, which is quite difficult if you need a detailed information such as vehicle license plates, pedestrians, and other objects. To solve these problems, the concept of motion detection can be applied. Motion detection is used to identify and sort the desired object. This motion detection process uses the concept of image processing with the *Frame Difference* and *Frame Substraction* algorithms. The *Frame Difference* algorithm is used to examine objects in the image. If the object is detected, then the *Frame Substraction* algorithm is applied to detect the position of the object where the position data is collected into motion. The results of this study were quite capable of detecting objects and their movements, based on the results of 25 tests from different videos, the percentage of success reached 76.0%. The conclusion is, the method used is quite successful in detecting the movement of objects in the video based on the difference (*difference*) obtained from the reduction of the previous *frame* and the next *frame*.

Keywords: motion detection, image processing, *frame difference*, *frame subtraction*, CCTV cameras

1. PENDAHULUAN

Dewasa ini, CCTV banyak digunakan sebagai kamera pengawas untuk mengawasi pergerakan setiap individu dalam suatu ruangan tertentu. Proses pengawasan secara manual akan banyak menguras tenaga dari staf pengawas. Untuk menyelesaikan permasalahan tersebut, maka dapat diterapkan konsep deteksi gerakan. Proses pendeteksian gerakan digunakan untuk mengidentifikasi dan memilah objek yang diinginkan berdasarkan citra yang tertangkap oleh kamera CCTV.

Proses pendeteksian gerakan ini dapat dilakukan dengan menggunakan konsep pengolahan citra (*image processing*) seperti algoritma *Frame Difference* dan *Frame Substraction*. Algoritma *Frame Difference* ini akan digunakan untuk mengecek apakah ada perbedaan antara dua buah *frame* yang sedang dibandingkan. Apabila terdapat perbedaan, berarti ada pergerakan objek dalam citra *frame* tersebut. Algoritma *Frame Substraction* akan digunakan untuk mendeteksi posisi dari objek yang bergerak tersebut. Beberapa tahapan yang diperlukan dalam melakukan proses pendeteksian gerakan ini yaitu proses *grayscale*, *binary image*, segmentasi, deteksi, pelacakan, pengenalan, dan perhitungan. Kamera yang digunakan untuk mengambil gambar haruslah tidak bergerak.

Berdasarkan sebuah penelitian yang telah dilakukan, terdapat tiga metode yang bisa digunakan untuk dapat mendeteksi pergerakan. Yaitu metode *temporal differences*, *background subtraction* dan *optical flow*. Metode *Background subtraction* sendiri dapat membandingkan citra tertentu dengan sebuah citra yang dijadikan referensi awal.(Mishra, 2011)[1]

Penelitian juga pernah dilakukan Murali, bahwa metode *temporal differences* membutuhkan sebuah proses komputerasi yang mendetail dan hal tersebut sangat peka terhadap sebuah gangguan. Metode *temporal differences* juga biasa dikenal dengan sebutan *frame differences*. Sebuah metode pembanding antara sebuah *frame* citra yang didapat dan sesuai dengan urutan waktunya.(Murali & Girisha, 2009)[2]

Penelitian lain juga dilakukan Migliore dengan metode yaitu, *frame differences*. Hasilnya terdapat beberapa perbedaan, dalam penelitian ini juga digunakan sebuah teknik yaitu teknik penentuan citra awal, dengan cara menggabungkan dua buah metode yaitu metode *frame differences* dengan metode *background subtraction*. Hasilnya dikenal dengan sebutan metode *hybrid*. Perbandingan itu sendiri dapat dilakukan setelah melakukan penggabungan antara perbedaan objek yang dapat terdeteksi antara kumpulan *frame*.(Migliore, Matteucci, & Naccari, 2006)[3]

2. METODE PENELITIAN

Langkah - langkah yang dilakukan dalam penelitian ini dijabarkan sebagai berikut:

1. Studi Pustaka
Mengumpulkan dan mempelajari berbagai karya ilmiah seperti jurnal nasional dan internasional yang berhubungan dengan proses pendeteksian objek khususnya mengenai cara penerapan metode *Frame Difference* dan *Frame Substraction*.
2. Proses pembangunan aplikasi pendeteksian pergerakan objek mencakup:
 - a. Analisis masalah
Analisis sistem yang dilakukan mencakup proses analisis masalah, analisis kebutuhan fungsional dengan menggunakan *use case diagram*, analisis kebutuhan non fungsional dengan menggunakan kerangka PIECES.
 - b. Perancangan aplikasi
Pada tahap ini dilakukan perancangan tampilan antarmuka pada aplikasi (*user interface*) dengan aplikasi *balsamiq Mockup*.
 - c. Pembuatan Aplikasi

Pada tahap ini dilakukan proses pengkodean (*coding*) sistem dengan metode *Frame Difference* dan *Frame Substraction* menggunakan bahasa pemrograman Visual Studio 2013.

d. Pengujian Aplikasi

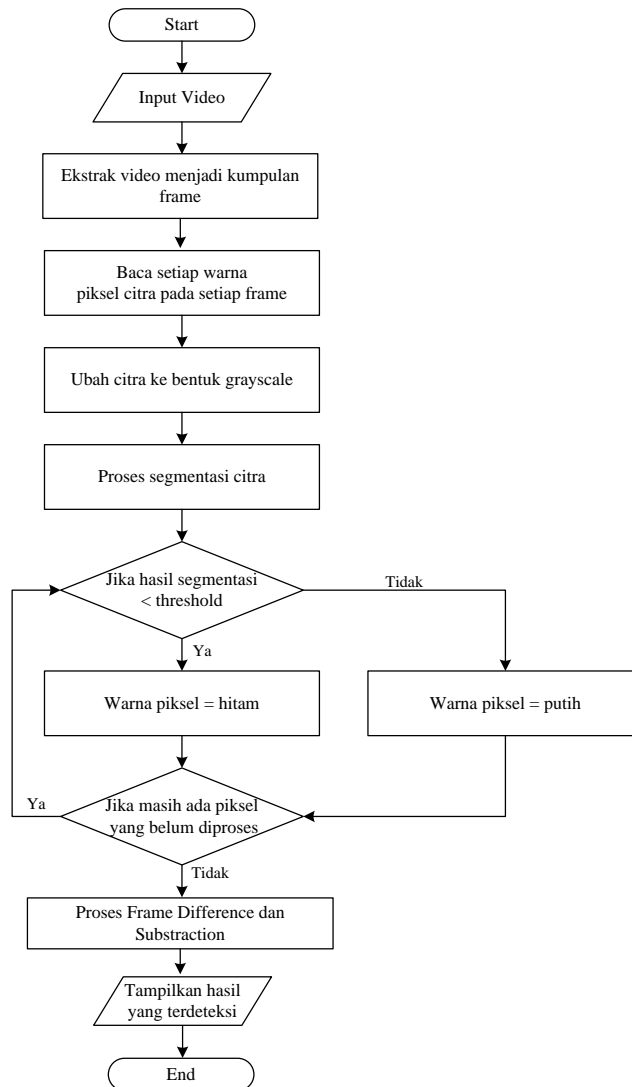
Aplikasi yang telah selesai dibuat akan diuji dengan tujuan untuk mengetahui bagaimana cara kerja algoritma *Frame Difference* dan *Frame Substraction* yang telah diimplementasikan pada aplikasi. Pengujian akan dilakukan untuk mengukur akurasi dari hasil pendeteksian objek. Proses pengujian akan menggunakan metode *Confusion Matrix*.

3. Analisis Hasil Pengujian

Dari pengujian data sample berdasarkan dataset dapat disimpulkan apakah metode yang digunakan cukup mampu mendeteksi pergerakan objek dalam video berdasarkan perbedaan (*difference*) yang diperoleh dari pengurangan frame sebelumnya dan frame selanjutnya. Presentase keberhasilan pendeteksian juga menunjukkan nilai 76,0% dari 100% data yang diuji

4. Menyusun Laporan Penelitian.

Proses kerja dari sistem yang dibuat dapat dirincikan sebagai berikut:



Gambar 1. Analisis Proses Kerja

2.1 Frame Difference dan Substraction

Mendeteksi pergerakan objek dari sederetan *frame* yang diambil dari sebuah kamera statis dapat dilakukan dengan menggunakan metode *frame difference*. Sasaran dari pendetektan ini adalah untuk dapat mendeteksi suatu objek bergerak dari perbedaan antara satu frame dengan frame awal. Metode *frame difference* itu sendiri adalah sebuah metode yang sering digunakan untuk mendeteksi sebuah pergerakan. Metode ini mengadopsi perbedaan berbasis pixel untuk menemukan sebuah objek yang bergerak.(P, 2014)[4]

Penelitian Kenchannavar, menjelaskan tentang algoritma apa yang diterapkan pada metode *background subtraction* dan *frame differences*, penerapan sebuah konsep SAD atau (*Sum of Absolute Difference*), sering pula digunakan untuk membuktikan ada atau tidak sebuah pergerakan dalam sebuah kumpulan citra.(Kenchannavar, Patkar, & Kulakarni, 2010)[5]

Sebuah nilai *threshold* dinyatakan berbanding terbalik terhadap jumlah *noise* yang dapat dideteksi setelah dibandingkan, hal tersebut menjelaskan bahwa jika nilai *threshold* kecil, maka akan semakin besar pula jumlah *noise* yang dapat terdeteksi.(Yudhistiro, 2018)[6]

Untuk bisa menghitung sebuah *frame difference* dapat dilakukan dengan melakukan *substraction* antara dua buah *frame*. Citra hasil *substraction* akan digunakan untuk menentukan dimana posisi objek yang bergerak.(Gonzalez, Woods, & Masters, 2009)[7]

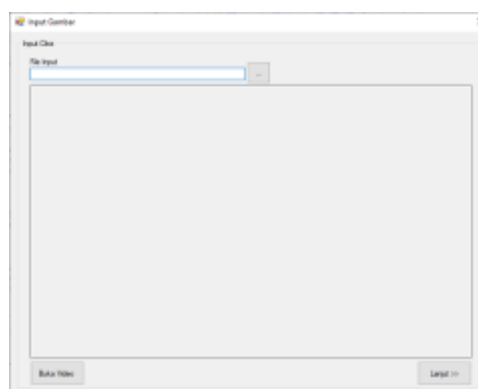
Agar dapat membandingkan sebuah citra dengan waktu t , dengan citra $t-1$, dapat pula dilakukan dengan algoritma *Double Differences*, selanjutnya akan dilakukan perbandingan antara kedua citra $t-1$ dengan citra $t-2$.(Hatta & Susrama, 2017)[8]

2.2 Segmentasi Citra

Citra merupakan suatu cerminan, imitasi ataupun kemiripan dari sebuah objek, sebagai tempat keluaran dalam sebuah sistem perekaman. data yang didapat biasanya bersifat optik baik berbentuk foto dan dapat pula bersifat analog berupa video seperti yang biasa ditampilkan pada layar televisi dan lain sebaainya, atau bersifat digital yang bisa langsung disimpan pada media penyimpan seperti harddisk. Segmentasi citra adalah sebuah proses pengolahan citra yang tujuannya untuk memisahkan suatu kumpulan wilayah objek dengan kumpulan wilayah pada latar belakang yang berguna agar sebuah objek mudah dianalisis.(Pereira, Barbeiro, Lemos, Morgado, & Silva, 2012)[9]

3. HASIL DAN PEMBAHASAN

Tampilan awal dari aplikasi Implementation *Motion Capture* pada Kamera CCTV dengan Algoritma *Frame Difference* dan *Frame Substraction* dapat dilihat pada gambar berikut:



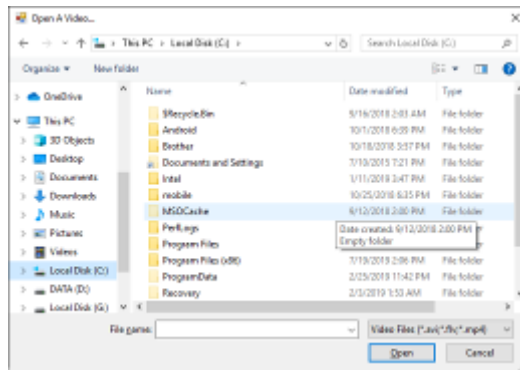
Gambar 2. Tampilan Awal

Untuk melakukan proses ekstraksi *file video* menjadi kumpulan *frame*, maka pemakai dapat mengklik tombol *Buka Video*, sehingga sistem akan menampilkan *form Video* seperti terlihat pada gambar berikut:



Gambar 3. Tampilan *Form Dataset*

User dapat memilih *file* video yang akan diproses dengan mengklik tombol Buka Video, sehingga sistem akan menampilkan *form Browse* seperti terlihat pada gambar berikut:



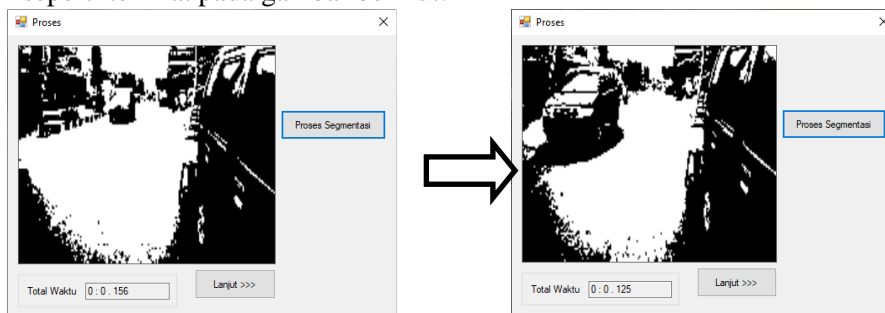
Gambar 4. Tampilan *Form Browse*

Setelah selesai melakukan proses pemilihan *file* yang diinginkan, maka *user* dapat mengklik tombol Open, sehingga sistem akan menampilkan *form Ekstrak File Video* seperti terlihat pada gambar berikut:

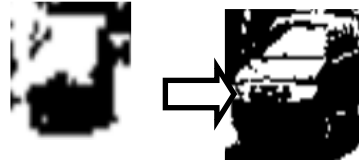


Gambar 5. Tampilan *Form Ekstrak Video*

Setelah berhasil mengekstrak video, maka *user* dapat memulai proses pendeteksian pergerakan seperti terlihat pada gambar berikut:

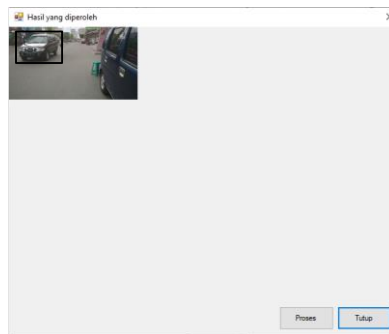


Gambar 6. Tampilan *Form* Segmentasi Terhadap Gambar Sebelum dan Sesudah Hasil *difference* dan *substraction* dari kedua *frame* diatas dapat dirincikan sebagai berikut:



Gambar 7. Tampilan Hasil *Difference* dan *Substraction*































Hasil proses pendeteksian pergerakan yang diperoleh dapat dilihat pada gambar berikut:



Gambar 8. Tampilan Hasil Proses Pendeteksian Pergerakan

Pada beberapa tes uji, proses pendeteksian gerak dilakukan terhadap 25 data untuk diuji, data tersebut memiliki background dan kondisi cahaya yang berbeda. Hasil Proses pendeteksian pergerakan dengan *frame difference* dan *frame subtraction* dapat dilihat pada tabel berikut:

No	Data Uji	Video awal	Video Segmentasi	Nilai Threshold	Nilai Deteksi RGB	Keterangan
1	Videouji1			104	87%	Berhasil Terdeteksi
2	Videouji2			108	72%	Berhasil Terdeteksi
3	Videouji3			197	22%	Gagal, Objek Tidak Terdeteksi
4	Videouji4			119	68%	Berhasil Terdeteksi
5	Videouji5			194	32%	Gagal, Objek Tidak Terdeteksi
6	Videouji6			137	63%	Berhasil Terdeteksi
7	Videouji7			107	38%	Gagal, Objek Tidak Terdeteksi
8	Videouji8			112	56%	Berhasil Terdeteksi

9	Videouji9			129	59%	Berhasil Terdeteksi
12	Videouji12			112	55%	Berhasil Terdeteksi
13	Videouji13			104	53%	Berhasil Terdeteksi
14	Videouji14			117	61%	Berhasil Terdeteksi
15	Videouji15			104	22%	Gagal, Objek Tidak Terdeteksi
16	Videouji16			124	62%	Berhasil Terdeteksi
17	Videouji17			194	31%	Gagal, Objek Tidak Terdeteksi
18	Videouji18			119	55%	Berhasil Terdeteksi
19	Videouji19			112	65%	Berhasil Terdeteksi
20	Videouji20			148	67%	Berhasil Terdeteksi
21	Videouji21			178	72%	Berhasil Terdeteksi
22	Videouji22			138	56%	Berhasil Terdeteksi
23	Videouji23			139	29%	Gagal, Objek Tidak Terdeteksi
24	Videouji24			132	54%	Berhasil Terdeteksi
25	Videouji25			112	52%	Berhasil Terdeteksi

Gambar 9. Hasil test uji

Berdasarkan hasil pengujian dari 25 data diuji pada video diatas, maka dapat diperoleh bahwa pengaruh warna background, warna objek dan kondisi cahaya akan menentukan berhasil atau tidaknya pendeteksian gerak pada video,

Dari tabel diatas dapat dianalisis dan diketahui berapa tingkat keberhasilan dari program pendeteksian gerakan dengan kamera cctv ini. program ini mampu melakukan proses pendeteksian gerak sebanyak 19 data dari 25 data yang diuji, sehingga presentase keberhasilan dari program ini adalah sebagai berikut :

$$\text{Akurasi} = \frac{19}{25} * 100 \% = 76,0 \%$$

$$\text{Error Rate} = \frac{6}{25} * 100 \% = 24,0 \%$$

Terdapat beberapa hal yang sudah dihasilkan dan dapat dianalisis dari pengunggahan citra adalah waktu sistem untuk memproses suatu frame guna mengetahui adanya gerakan, Belum dapat diketahui standar batasan yang baku untuk menentukan waktu respon ideal pada sebuah sistem pengunggahan citra, waktu ideal dapat ditentukan dengan besaran delay waktu dalam pemrosesanya.(Nurhopipah & Harjoko, 2018)[10]

4. KESIMPULAN

Berdasarkan hasil uji coba dari dataset menggunakan aplikasi, penulis mendapatkan beberapa kesimpulan adalah sebagai berikut:

1. Proses pendeteksian gerakan objek akan lebih mudah dilakukan dengan menggunakan *input* berupa citra biner, sehingga citra input harus disegmentasi menjadi bentuk citra biner terlebih dahulu.
2. Proses pendeteksian gerakan objek akan gagal dilakukan apabila warna latar (*background*) memiliki warna yang hampir mirip dengan objek yang akan dideteksi.
3. Metode yang digunakan mampu mendeteksi pergerakan objek dalam video dengan akurasi 76,0 % dengan Error 24,0% berdasarkan perbedaan (*difference*) yang diperoleh dari pengurangan gambar frame sebelumnya dan gambar frame selanjutnya.

5. SARAN

Dalam penulisan ini juga terdapat beberapa saran yang penulis rangkum dan ingin sampaikan adalah sebagai berikut:

1. Agar hasil deteksi gerakan objek menjadi lebih akurat, maka dapat dikombinasikan dengan algoritma lainnya.
2. Mengoptimalkan waktu eksekusi dengan menggunakan algoritma lainnya.
3. Dapat mendeteksi objek pada kamera CCTV secara langsung, dan memberitahukan informasi yang sifatnya rahasia, seperti : pencuri dan sebagainya.
4. Agar dapat dilakukan pengujian dalam keadaan cahaya yang kurang cerah

DAFTAR PUSTAKA

- [1] Mishra, S. (2011). A Novel comprehensive method for real time Video Motion Detection Surveillance. *ArXiv.Org*.
- [2] Murali, S., & Girisha, R. (2009). Segmentation of motion objects from surveillance video sequences using temporal differencing combined with multiple correlation. *6th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009*.
<https://doi.org/10.1109/AVSS.2009.15>
- [3] Migliore, D. A., Matteucci, M., & Naccari, M. (2006). A revaluation of frame difference in fast and robust motion detection. *Proceedings of the ACM International Multimedia Conference and Exhibition*. <https://doi.org/10.1145/1178782.1178815>
- [4] P, L. M. (2014). Real Time Motion Detection Using Background Subtraction Method and Frame Difference. *International Journal of Science and Research (IJSR)*, 3(6), 1857–1861. Retrieved from <https://www.ijsr.net/archive/v3i6/MDIwMTQ2NjE=.pdf>
- [5] Kenchannavar, H., Patkar, G. S., & Kulakarni, U. P. (2010). Modeling of coverage preserving algorithm using object oriented methodology with UML. *ICWET 2010 - International Conference and Workshop on Emerging Trends in Technology 2010, Conference Proceedings*.
<https://doi.org/10.1145/1741906.1742071>

- [6] Yudhistiro, K. (2018). Restorasi Citra Optical Character Recognition Dengan Algoritma Recurrent Hopfield. *SMATIKA JURNAL*, 8(01), 18–22. <https://doi.org/10.32664/smatika.v8i01.195>
- [7] Gonzalez, R. C., Woods, R. E., & Masters, B. R. (2009). Digital Image Processing, Third Edition. *Journal of Biomedical Optics*. <https://doi.org/10.1117/1.3115362>
- [8] Hatta, M., & Susrama, I. G. (2017). COUNTING SPERMA AKTIF MENGGUNAKAN METODE OTSU THRESHOLD DAN LOCAL ADAPTIVE THRESHOLD. *Teknika : Engineering and Sains Journal*. <https://doi.org/10.5281/zenodo.1067713>
- [9] Pereira, T., Barbeiro, P., Lemos, J., Morgado, M., & Silva, E. (2012). Digital image acquisition for ophthalmoscope. *2012 IEEE 2nd Portuguese Meeting in Bioengineering, ENBENG 2012*. <https://doi.org/10.1109/ENBENG.2012.6331383>
- [10] Nurhopipah, A., & Harjoko, A. (2018). Motion Detection and Face Recognition for CCTV Surveillance System. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*. <https://doi.org/10.22146/ijccs.18198>