Analisa *Motion Detection* Berbasis *Frame-to-Frame Difference* dan Segmentasi Untuk Program Rekam Video Otomatis Berdasarkan Objek Yang Bergerak

Zidan Rasyidi Lazuardani[1]*

[1]Divisi RnD zidcreative
**Korespondensi: zidan@zidcreative.com*

ABSTRAK

Sistem pemantauan berbasis kamera telah banyak digunakan dalam dunia modern di berbagai bidang dan sektor termasuk keamanan lingkungan yang banyak sekali menggunakan CCTV dan perangkat sejenis, perangkat seperti itu secara fundamental bekerja seperti mata manusia yang terus menerus mengawasi dan dan menyimpan rekaman kejadian sepanjang waktu yang dapat menghabiskan ruan penyimpanan perangkat sehingga tidak efisien. Telah banyak dikembangkan sistem rekaman otomatis yang hanya merekam jika terdapat objek yang bergerak dan beberapa diantaranya terbukti dapat meningkatkan tingkat efisiensi perekaman dan memangkas ruang penyimpanan perangkat. Untuk mengetahui bagaimana sistem tersebut bekerja dan parameter apa saja yang harus disesuaikan untuk menghasilkan sistem pendeteksian yang optimal diperlukan pengujian dan analisa secara langsung. Aplikasi / program perekam otomatis ini menggunakan perbandingan perbedaan antar *frame* dan melakukan segmentasi terhadap pixel untuk menentukan adanya gerakan atau tidak dengan mengubah 2 variabel atau parameter yaitu *Threshold* dan Min area. Program dikembangkan menggunakan pustakan OpenCV Python yang dikemas dalam bentuk program desktop .exe yang dapat dihubungkan dengan kamera internal komputer atau kamera eksternal dengan tampilan yang didesain untuk pengujian secara langsung, pengujian dilakukan dengan mengarahkan objek yang bergerak dan mengatur 2 variabel untuk mendapat nilai yang sesuai untuk pengujian gerakan pada berbagai kondisi objek. Hasilnya program berjalan dengan baik dan mampu mendeteksi dan merekam otomatis setiap pergerakan yang terdeteksi pada berbagai objek dan kondisi dengan nilai variabel yang telah disesuaikan.

Kata kunci : *Motion Detection*, *Segmentation, OpenCV*, Python

PENDAHULUAN

Banyak kejadian berlalu dalam setiap waktu, mata kita mampu mengenali lingkungan dan kejadian dengan sangat baik melalui indra visual kita dan memproses suatu kejadian menjadi suatu informasi yang berharga. Semakin banyak suatu kejadian yang bisa diambil informasinya maka semakin banyak pula informasi yang didapatkan.

Pada zaman modern ini manusia selalu ingin mendapatkan informasi dari suatu kejadian melalui visual, mata manusia tidak diracang untuk melihat suatu lingkungan seara terus menerus dan mengambil kejadian yang ada untuk itulah diciptakan CCTV untuk memudahkan proses pemantauan di segala bidang untuk mempermudah memperoleh suatu kejadian [1].

CCTV memerlukan sumber daya penyimpanan visual yang membutuhkan banyak ruang penyimpanan, penggunaan CCTV yang menyala sepanjang waktu akan menguras sumber daya penyimpanan tersebut dan menjadikan sistem pemantauan kurang efisien terutama jika CCTV hanya memantau area kosong tanpa banyak kejadian, dan oleh karena itu lebih baik

jika pengambilan dan penyimpanan visual kejadian hanya dilakukan ketika kejadian berlangsung.

Melalui pengolahan citra digital memugkinkan mengambil banyak citra sekaligus untuk dideteksi perubahan visualnya, metode *frame by frame* pada pengolahan citra video bisa digunakan untuk mendeteksi perubahan citra dalam setiap frame [2], perubahan inilah yang bisa dijadikan informasi bahwa ada suatu "kejadian" dalam visual yang diterima [3].

Dari memanfaatkan deteksi kejadian ini maka sebuah CCTV atau perangkat sejenisnya tidak perlu terus menerus merekam dan bisa merekam hanya jika ada sesuatu kejadian [4]. Kejadian sendiri didefinisikan sebagai sesuatu yang dinamis yang dalam konteks perekaman CCTV adalah sesuatu yang bergerak.

Melalui problematika tersebut, penelitian ini membuat suatu analisa deteksi gerakan pada suatu citra video untuk mendapatkan parameter yang sesuai yang bisa digunakan untuk merancang suatu sistem atau program monitoring berbasis gerakan yang bisa dterapkan ke sistem tertanam ataupun sistem berbasis IoT.

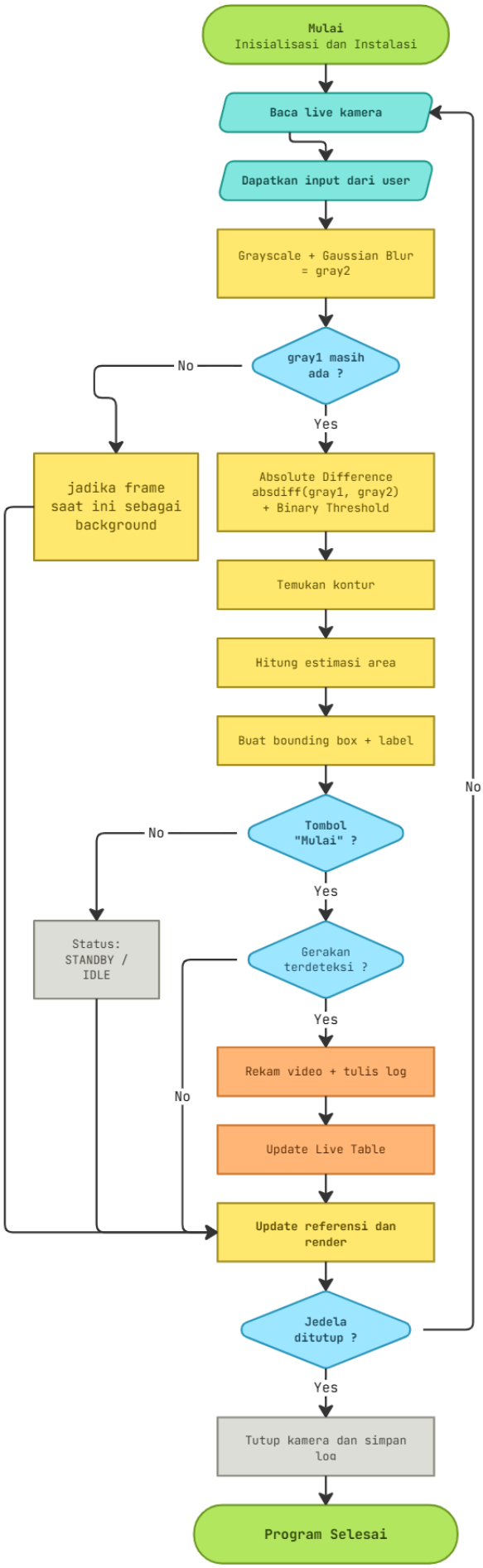
METODOLOGI

Penelitian ini menggunakan metode kuantitatif dengan berbagai data parameter yang akan diuji untuk menentukan nilai parameter yang sesuai yang dibutuhkan untuk mendeteksi objek dalam berbagai kondisi dengan menggunakan GUI Python sebagai prgram pengujiannya.

2.1. Alur proses / diagram alur

Pengujian ini menggunakan OpenCV yang ditulis menggunakan python dengan tampilan / User Interface tkinter agar proses pengujian lebih mudah dan interaktif. Gambar 1 adalah diagram alur program dari mulai pembacaan citra hingga mendeteksi objek dan menghasilkan output rekaman video.

1. Mulai, proses inisialisasi pustaka, buka jendela UI dan instalasi folder log dan rekaman yang dibutuhkan,
2. Mengambil citra gambar untuk frame pertama
3. Mengambil input parameter threshold, jarak dan min area dari user
4. Pre-processing: Konversi ke Grayscale & Gaussian Blur



***Gambar 1**, Diagram alur program*

5. Apakah 'gray1' (Frame Sebelumnya) Ada ?, jika ada maka akan ke proses berikutnya, jika tidak maka jadikan frame saat ini sebagai frame background lalu menuju ke langkah 14
6. Hitung Selisih Absolut (Absolute Difference) & Threshold
7. Cari Kontur Objek & Filter Berdasarkan Minimum Area
8. Hitung Estimasi Luas Fisik (m²) Berdasarkan Jarak Kamera (D²)
9. Gambar Kotak & Label Teks pada Kontur (Contoh: #1 | 0.45m²)
10. Apakah Tombol "▶ Mulai" Ditekan ?, jika tidak maka akan statusnya menjadi “Stand by” dan ke proses nomor 14 namun jika diklik maka akan ke langkah berikutnya
11. Apakah Terdeteksi Gerakan Valid ?, jika tidak maka lanjut ke proses nomor 14 dan jika iya maka lanjut ke proses berikutnya
12. Proses rekam video
13. Update tabel dan log
14. Update referensi dan render pada frame
15. Apakah jendela UI ditutup ?, jika tidak maka proses akan kembali berulang ke langkah 2 dan jika iya maka tutup kamera dengan aman dan log disimpan, proses selesai

Jarak kamera dari objek perlu dimasukkan secara manual dalam program agar dapat menentukan ukuran obek dan gerakan yang terdeteksi. Kemudian dalam penelitian ini ada 2 variabel yang nilai dapat diubah selama pemantauan bijalankan yaitu threshold dan min area yang akan menetukan tingkat akurasi deteksi.

2.2. Variabel

Segmentasi menggunakan *Threshold* [5] digunakan untuk mengukur perbedaan tingkat keabuan (grayscale intensity) antara frame latar belakang (gray1) dengan frame saat ini (gray2), semakin rendah nilainya maka semakin sensitif program dalam meneteksi perubahan grayscale yang menjadi memicu utama pendeteksian gerakan.

$$D(x,y) = |I(x,y) - B(x,y)|$$

$$\mathrm{T_(binary)(x,y)} = \{ (255, \text{if } \mathrm{D(x,y)} \geq \mathrm{T_(val)}),$$

$$(0, \text{if } \mathrm{D(x,y)} < \mathrm{T_(val)}) :\}$$

Dengan :

- $I(x,y)$: Nilai intensitas keabuan (grayscale) piksel pada koordinat (x,y) dari frame saat ini.
- $B(x,y)$: Nilai intensitas keabuan piksel pada koordinat (x,y) dari frame latar belakang (background).
- $D(x,y)$: Nilai selisih absolut intensitas piksel.
- $T_\{val\}$: Nilai ambang batas (threshold value) yang ditentukan pengguna (skala 0–255).
- $T_\{binary\}(x,y)$: Hasil piksel biner (255 = putih/bergerak, 0 = hitam/diam).

Sedangkan variabel Minimum Area adalah filter ukuran objek berdasarkan jumlah total piksel yang saling terhubung membentuk suatu kontur (contour area), Meskipun suatu area terdeteksi bergerak oleh threshold, jika luas total piksel yang bergerak tersebut di bawah angka minimum area yang diset, maka sistem akan mengabaikannya. Semakin kecil nilai maka program akan mendeteksi perubahan kecil sebagai gerakan. Penjumlahan area pixel $A(C_k)$ didefinisikan sebagai

$$1/2\ sum_(i = 1)\hat{}(N)\ (x_i\ y_(i + 1)\ -\ x_(i + 1)\ y_i)$$

Dengan fungsi filter validasi gerakan $M(C_k)$ berdasarkan batas minimum area $A_\{min\}$ didefinisikan sebagai :

$$\mathrm{M(C_k)} = \{ (1\ (\text{Valid / Merekam}),$$

$$\text{if } \mathrm{A(C_k)} >= \mathrm{A_(min)}),$$

$$(0\ (\text{Diabaikan}), \text{if } \mathrm{A(C_k)} < \mathrm{A_(min)}) :\}$$

Dengan :

- C_k : Kontur objek bergerak ke- k yang terdeteksi.
- $A(C_k)$: Total luas area kontur C_k (dalam satuan piksel ^2).
- (x_i, y_i) : Koordinat titik sudut batas (boundary points) ke- i dari kontur C_k .
- N : Jumlah total titik batas kontur.
- $A_\{min\}$: Batas minimum area objek yang ditentukan pengguna via slider (dalam piksel).
- $M(C_k)$: Keputusan deteksi (1 = gerakan sah yang memicu pencatatan/perekaman, 0 = noise/diabaikan)

Kedua variabel tersebut digunakan bersamaan sebagai filter untuk mendeteksi pergerakan objek. Variabel pertama yang akan bekerja adalah *Threshold* yang akan memeriksa perubahan piksel antar frame berdasarkan nilai kesensitifan yang ditentukan, variabel tersebut akan menentukan seberapa pixel yang harus dideteksi sebagai pixel gerakan dan menampilkan hasilnya sebagai pixel putih pada frame grayscale. Meskipun pixel berubah

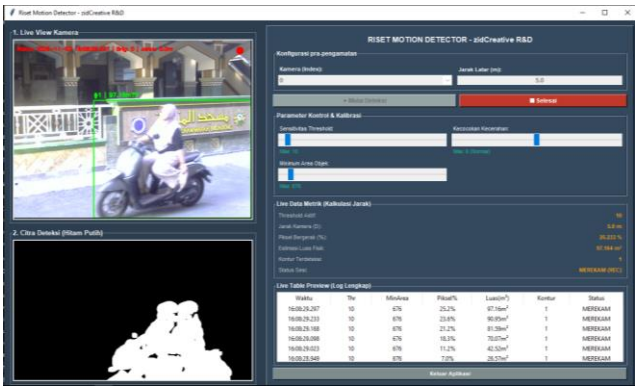
tidak berarti otomatis terdeteksi sebagai gerakan, butuh variabel berikutnya untuk menentukan apakah itu sebuah gerakan atau tidak.

Variabel min area digunakan untuk menghitung jumlah pixel yang telah diproses oleh *Threshold* untuk menentukan apakah jumlah pixel sudah memenuhi syarat untuk dijadikan sebuah gerakan. Apabila jumlah pixel yang telah diproses mencapai nilai yang ditentukan maka program / aplikasi akan memutuskan bahwa pixel pada bagian frame tersebut adalah sebuah gerakan.

PENGUJIAN DAN ANALISA

3.1. Setup pengujian

Program ini berupa file aplikasi / software .exe yang bisa dibuka di komputer maupun laptop, terdapat beberapa setup untuk mengakomodasi kebutuhan pengujian yaitu input kamera, jarak dan kecerahan. Ketika program aplikasi dibuka maka program akan mengidientifikasi kamera secara otomatis. Dalam program ini terdapat 3 parameter yang tidak terpengaruh ketika nilai ketiga parameter tersebut diubah yaitu kamera, jarak dan kecerahan serta terdapat 2 variabel yang jika diubah akan mempengaruhi hasil pengujian yaitu *Threshold* dan Min Area.



***Gambar 2**, tampilan UI program*

Ketika aplikasi dijalankan pengguna dapat mengatur ketiga parameter, pengguna dapat memilih salah satu dari 3 pilihan kamera yang tersedia berdasarkan perangkat, secara default aplikasi akan membuka kamera yang tersedia. Input jarak kamera bersifat opsional yang digunakan untuk mendukung program mengukur ukuran objek yang terdeteksi bergerak dalam meter. Sedangkan kecerahan bisa digunakan untuk mengatur kecerlangan citra agar tidak terlalu terang dan terlalu gelap. Ketiga parameter ini tidak mempengaruhi hasil pengujian.

Sedangkankan variabel yang diuji secara default untuk *Threshold* nilainya adalah 25 sedangkan untuk min area nilainya dalah 1000. Kedua variabel tersebut dapat diubah melalui slider pada aplikasi dan begitu nilai diubah maka akan otomatis mempengaruhi hasil pengujian.

3.2. Proses pengujian

Program dijalankan secara *offline* yang dihubungkan dengan kamera eksternal yaitu kamera webcam Logitech 720p. Kamera diletakkan di beberapa tempat pengujian untuk memantau pergerakan berbagai objek dalam berbagai waktu dan kondisi. Selama proses pengujian, nilai variabel Threshold dan min area terus diubah untuk mengetahui detail perubahan dan untuk mendapatkan nilai terbaik untuk berbagai kondisi tertentu.

Tantangan utama dalam proses pengujian adalah pencahayaan, jika citra yang didapatkan terlalu gelap maka program / aplikasi tidak bisa membedakan perubahan grayscale pada pixel antar frame sehingga peluang untuk mendeteksinya sebagai perubahan pixel menjadi berkurang. Hal ini bisa diatur melalui sensifitas *Threshold* dengan menurunkan sensifitasnya, namun jika terlalu sensitif maka perubahan sedikit saja seperti perubahan cahaya atau bayangan matahari dapat memicu terjadinya deteksi gerakan.

3.3. Hasil pengujian

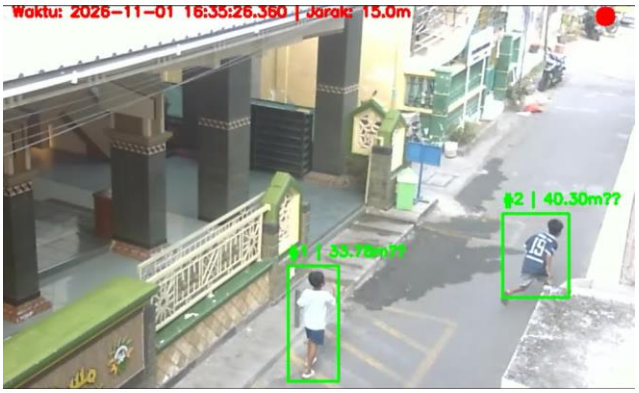
Terdapat 2 output yang dihasilkan oleh aplikasi / program secara otomatis yaitu log data selama sesi dimulai dan video gerakan yang otomatis terekam ketika terdeteksi gerakan. Berikut ini adalah poin-poin yang didapat selama pengujian :

1. Jarak kamera harus disesuaikan dari objek yang ingin dipantau agar memperoleh jarak pendeteksian yang akurat.
2. Jika kontras antara objek dan latar belakang kecil atau objek dan latar belakang sama-sama redup sehingga keduanya sulit dibedakan maka dibutuhkan pengaturan nilai *Threshold* yang kecil agar cukup sensitif mendeteksi perubahan pada pixel



***Gambar 3**, objek dan latar belakang jauh / kecil dengan kontras kecil / redup*

3. Namun jika kontras atau objek dan latar belakang cukup jelas untuk dibedakan maka nilai *Threshold* bisa diperbesar agar tidak menimbulkan *noise* / gangguan terhadap perubahan pencahayaan kecil.
4. Jika objek gerak jauh atau kecil maka dibutuhkan pengaturan nilai min area yang kecil agar jumlah sedikit pixel yang berubah tersebut dapat memenuhi kriteria sebagai sebuah gerakan.



***Gambar 4**, Ukuran objek ideal dan kontras objek dan latar belakang sangat jelas*

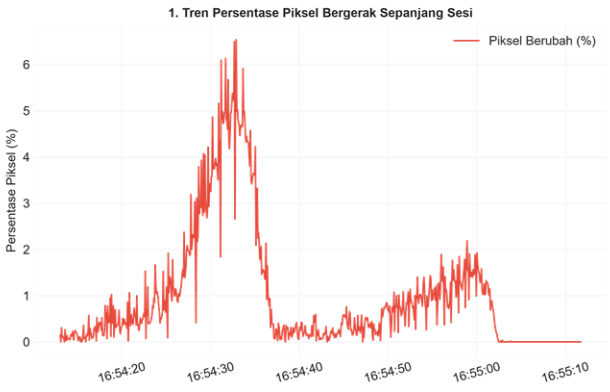
5. Sebaliknya jika objek gerak dekat atau besar maka dibutuhkan min area yang besar agar gerakan objek-objek kecil / *noise* tidak ikut terdeteksi sebagai gerakan

***Table 1,** Tabel data log pengujian*

T	Th.	Ar.	Dst.	Br.	Pix.	Es.	Con	St.
49:25	25	1000	5	-30	0	0	0	St
49:26	25	1000	5	-30	0	0	0	St
49:27	25	1000	5	-30	0	0	0	St
49:28	25	1000	5	-30	0	0	0	St
49:29	25	1000	5	-30	0	0	0	St

Adapun parameter yang terekam selama proses pengujian otomatis tersimpan dalam format .csv dengan kolom meliputi :

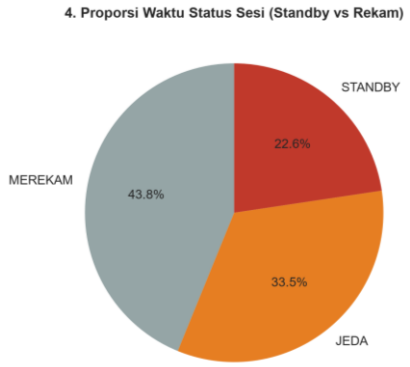
- T *(Time),* adalah waktu *(timestamp)* data diambil. Satu baris data mewakili data 1 frame
- Th *(Threshold),* adalah nilai variabel *Threshold* sensifitas pixel yang telah diatur
- Ar (Min Area), adalah nilai batas minimal jumlah pixel yang memenuhi syarat untuk menjadi sebuah gerakan
- Dst (Camera Distance), adalah jarak kamera dari objek
- Br (Brightness), adalah nilai kecerlangan citra
- Pix (PixelChangedPct), adalah jumlah pixel yang terdeteksi sebagai gray 2 atau pixel yang berubah dari frame sebelumnya. Nilai ini dipengaruhi oleh sensifitas *Threshold*
- Est (Estimated), adalah presentase perubahan pixel pada citra yang berubah menjadi gray 2
- Con (Contour), adalah kontur yang merupakan jumlah kontur yang terdeteksi sebagai gerakan. Jumlah kontur dipengaruhi oleh banyaknya pixel yang menjadi gray 2 dan nilai min area yang ditentukan
- St (Status), status kamera saat ini. Jika tidak ada gerakan sama sekali maka statusnya “Stand by”, jika terdeteksi kontur gerakan maka akan otomatis merekam video lalu statusnya menjadi “Merekam” dan jeda antar kontur dalam waktu yang berdekatan maka statusnya akan “Jeda”.



***Gambar 5**, Grafik presentase piperubahan pixel terhadap waktu*

Gambar 5 menujukkan analisa data terhadap suatu pengujian. Garis merah menunjukkan jumlah pixel yang berhasil terdeteksi sebagai gray 2, semakin tinggi garis menunjukkan semakin banyak jumlah pixel yang terdeteksi dan bervariasi terhadap waktu, jika kita baca dari kiri ke kanan maka gtafik tersebut menunjukkan adanya 2 objek atau kelompok pixel yang bergerak mulanya kecil pada frame lalu

perlahan membesar lalu menghilang dan disusul objek berikutnya. Garis landai 0 berarti tidak pixel yang terdeteksi artinya sama sekali tidak ada gerakan.



***Gambar 6**, presentase status kamera selama sesi pengujian*

Gambar 6 menunjukkan presentase kamera dalam satu sesi pengujian. Ketika sesi uji diaktifkan / tombol “mulai” ditekan maka program tidak otomatis merekam namun memantau dan mencatat aktifitas citra dengan status ‘Stand by” dan hanya merekam jika terdeteksi gerakan.

KESIMPULAN DAN EVALUASI

Program aplikasi rekam otomatis berdasarkan deteksi gerakan ini telah dirancang tidak hanya untuk membuat sistem perekeman otomatis ketika terdapat objek bergerak namun juga didesain untuk mencari parameter yang sesuai membuat sistem berjalan efisien di berbagai kondisi yang nantinya bisa diterapkan ke berbagai aplikasi. Aplikasi berjalan dengan baik dan parameter bisa disesuaikan untuk mencari nilai yang sesuai untuk beragam kondisi.

Tujuan penggunaan pendeteksian menjadi penting untuk apa, objek apa dan kondisi seperti apa yang akan dilakukan perekaman otomatis berdasarkan deteksi objek yang bergerak. Perlu mengatur nilai min area jika yang menjadi acuan adalah besar-kecil atau jauh-dekatnya objek yang dideteksi, namun *Threshold* sensifitas segmentasi pixelnya lah yang perlu diatur jika yang menjadi acuan adalah gelap atau terangnya citra yang ditangkap. Keduanya memainkan peranan penting agar program bekerja dengan optimal dan sesuai dengan kebutuhan.

Perlu diingat bahwa penelitian ini masih banyak keterbatasan namun juga memiliki peluang besar untuk diujikan dengan berbagai kejadian yang lebih beragam.

REFERENSI

[1] B. Susilo, G. D. Mardianto, and D. Aldilaga, “Implementasi Closed Circuit Television (CCTV) Sebagai Sistem Keamanan di Lingkungan Fakultas Ilmu Pendidikan Universitas Negeri Malang,” vol. 9, no. 2, pp. 84–90, 2024, doi: 10.17977/um027v9i22024p84-90.

[2] I. Desember-, V. N. Page, and S. Saluky, “Moving Object Detection on CCTV Surveillance Using the Frame Difference Method,” vol. 4, pp. 114–122, 2019.

[3] R. A. Yuha, M. Danu, A. Fiqri, and M. Harahap, “Deteksi Gerakan pada Kamera CCTV dengan Algoritma Frame Difference dan Frame Substraction,” 2019.

[4] R. Akash, L. S. Anderson, M. F. Removal, R. Akash, and L. S. Anderson, “E cient Storage and Analysis of Videos through Motion-Based Frame Removal Efficient Storage and Analysis of Videos through,” 2023.

[5] R. T. Wahyuningrum, “SEGMENTASI OBYEK PADA CITRA DIGITAL MENGGUNAKAN,” vol. 13, no. 1, pp. 1–8, 2015, doi: 10.9744/informatika.13.1.1-8.