

**PENERAPAN PENGOLAHAN CITRA DENGAN METODE ADAPTIVE MOTION
DETECTION ALGORITHM PADA SISTEM KAMERA KEAMANAN DENGAN PUSH
NOTIFICATION KE SMARTPHONE ANDROID**

^[1]Alvin Antonius, ^[2]Dedi Triyanto, ^[3]Ikhwan Ruslianto

^{[1][2][3]}Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jl. Ahmad Yani, Pontianak
Telp./Fax.: (0561) 577963

e-mail:

^[1]alvin@untan.ac.id, ^[2]dedi.triyanto@siskom.untan.ac.id,

^[3]ikhwanruslianto@siskom.untan.ac.id

ABSTRAK

Kamera pengawas seperti CCTV selama ini sebagian besar digunakan sebagai bahan penyidikan ketika terjadi tindakan kriminal dan kurang dapat berperan dalam proses pencegahan tindak kriminal seperti pencurian. Penelitian ini ditujukan untuk membangun sebuah sistem kamera pengawas yang juga dapat berperan aktif dalam proses pencegahan tindak kriminal seperti pencurian. Sistem kamera pengawas yang dibuat menggunakan Raspberry Pi sebagai mesinnya dikarenakan harganya yang murah, ukurannya yang kecil serta penggunaan daya listrik yang rendah. Sistem menggunakan OpenCV untuk proses pengolahan citra dengan metode Adaptive Motion Detection untuk mendeteksi ada atau tidaknya suatu gerakan. Setiap gerakan yang terdeteksi oleh perangkat segera dikirimkan kepada penggunanya melalui push notification pada smartphone berbasis sistem operasi android. Sistem ini dapat mendeteksi gerakan dengan baik namun akan menghasilkan false positive apabila dalam pandangan kamera terdapat objek yang mudah bergerak seperti pohon, genangan air, lampu hias, televisi dan lain-lain. Beberapa hal yang dapat dilakukan untuk menambah efektivitas dari metode Adaptive Motion Detection yaitu dengan menambahkan proses blur pada gambar sebelum dilakukan proses thresholding agar dapat menghilangkan noise pada gambar. Nilai threshold yang digunakan mempengaruhi efektivitas sistem dalam mendeteksi gerakan dimana untuk menemukan nilai threshold yang tepat sangat bergantung pada kondisi cahaya pada lingkungan sekitar serta kualitas dari kamera yang digunakan.

*Kata kunci: Adaptive Motion Detection, Deteksi Gerakan, Pengolahan Citra,
Push Notification*

1. PENDAHULUAN

Kamera pengawas atau yang sering disebut CCTV (Closed Circuit Television) sudah sering ditemui di gedung-gedung perkantoran, bank, pusat perbelanjaan bahkan juga digunakan oleh toko-toko skala kecil hingga menengah maupun di rumah-rumah kalangan menengah atas. Hal tersebut secara tidak langsung menunjukkan semakin meningkatnya kekhawatiran dan kewaspadaan masyarakat terhadap tindak

pencurian. Penggunaan kamera pengawas selama ini sebagian besar digunakan sebagai bukti-bukti kejahatan ataupun sebagai referensi bagi penegak hukum untuk mengenali pelaku sehingga dapat menggali informasi lebih lanjut untuk menangkap pelaku namun kurang berperan dalam tindak pencegahan.

Oleh karena itu dilakukan penelitian untuk membangun sistem kamera pengawas yang dapat melakukan proses pengolahan

citra pada kamera dan mendeteksi adanya gerakan-gerakan dalam jarak pandang kamera dan memberitahu penggunaannya jika mendeteksi adanya gerakan sehingga dapat berperan dalam pencegahan terhadap tindak pencurian. Metode pendeteksian gerakan yang diterapkan dalam sistem didapat dari sebuah artikel dengan judul penelitian “Adaptive Motion Detection Algorithm using Frame Differences and Dynamic Template Matching Method” yang ditulis oleh Widyawan, Muhammad Ihsan Zul dan Lukito Edi Nugroho dari Universitas Gadjah Mada dan diterbitkan dalam jurnal The 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI) pada tahun 2012.

2. TINJAUAN PUSTAKA

2.1 Computer Vision

Computer Vision adalah salah satu disiplin ilmu dalam bidang komputer yang mempelajari relasi dan proses transformasi data visual seperti gambar ataupun video menjadi data yang lebih sederhana sehingga dapat diolah oleh computer untuk menghasilkan suatu keputusan atau informasi baru dari data visual yang diberikan. Berbeda dengan mata manusia yang menangkap data setiap objek secara visual dan dapat mengetahui bentuk dan jarak suatu objek dengan mudah, komputer menangkap tiap data yang ada sebagai susunan angka dalam matrix dua dimensi yang mewakili warna dari data visual yang ada [1]

2.2 OpenCV

OpenCV adalah sebuah *open source library* untuk *Computer Vision* yang boleh dipergunakan secara bebas. *OpenCV* dibuat dengan menggunakan bahasa pemrograman C dan C++ dan dapat dijalankan di berbagai sistem operasi mulai dari *Linux*, *Windows* dan *Mac OS X*. *OpenCV* memiliki banyak modul yang dapat membantu dalam

menyelesaikan bermacam ragam permasalahan terkait *Computer Vision* [2]. Arsitektur dan manajemen memori yang dimiliki oleh *OpenCV* yang memungkinkan untuk menjalankan proses dan perhitungan yang kompleks dengan sumberdaya yang terbatas sehingga memberi keleluasaan bagi penggunaannya untuk membangun algoritma pengolahan citra baik itu dengan menggunakan masukan berupa gambar ataupun video tanpa harus khawatir dengan proses alokasi dan dealokasi memori.

2.3 Adaptive Motion Detection Algorithm

Adaptive Motion Detection Algorithm adalah suatu algoritma dalam bidang *Computer Vision* yang berfungsi untuk mendeteksi gerakan dalam video dengan menggabungkan dua metode dalam mendeteksi gerakan pada video yaitu metode *Static Template Matching* dan *Dynamic Template Matching* [11]. Algoritma ini dibuat atas dasar untuk memperoleh metode yang lebih optimal dalam mendeteksi gerakan dimana pada metode-metode sebelumnya masih terdapat kelemahan sehingga sering kali muncul hasil berupa *False Positive* yaitu kondisi dimana sistem menyimpulkan bahwa terdapat gerakan namun sebenarnya tidak ada.

2.3.1 Frame Differences

Dalam pendeteksian gerakan pada video, algoritma ini menggunakan teknik *Frame Differences* dimana proses pendeteksian gerakan dilakukan dengan membandingkan 2 gambar atau lebih dari video dimana salah satu gambar yang digunakan disebut sebagai gambar acuan atau *template* untuk dibandingkan dengan gambar lainnya sehingga dapat dihasilkan kesimpulan mengenai ada atau tidaknya suatu gerakan [5].

Proses perbandingan gambar dengan metode *Frame Differences* dalam penelitian ini akan dilakukan perhitungan yang

mengacu pada persamaan 1, 2 dan 3 yang diambil dari artikel dalam jurnal internasional yang diterbitkan pada tahun 2012 dan ditulis oleh Widyawan, Muhammad Ihsan Zul dan Lukito Edi Nugroho dari Universitas Gadjah Mada dengan judul penelitian “*Adaptive Motion Detection Algorithm using Frame Differences and Dynamic Template Matching Method*”.

$$g_a(x, y) = \frac{g_R(x, y) + g_G(x, y) + g_B(x, y)}{3} \quad (1)$$

$$f_a(x, y) = \frac{f_R(x, y) + f_G(x, y) + f_B(x, y)}{3} \quad (2)$$

$$(g_a(x, y) - T) \leq f_a(x, y) \leq (g_a(x, y) + T) \quad (3)$$

Dimana :

$g_R(x, y)$ = nilai komponen R pada piksel x, y dari gambar yang diproses

$g_G(x, y)$ = nilai komponen G pada piksel x, y dari gambar yang diproses

$g_B(x, y)$ = nilai komponen B pada piksel x, y dari gambar yang diproses

$g_a(x, y)$ = nilai rata-rata dari jumlah nilai komponen RGB pada piksel x, y dari gambar yang diproses

$f_R(x, y)$ = nilai komponen R pada piksel x, y dari gambar referensi

$f_G(x, y)$ = nilai komponen G pada piksel x, y dari gambar referensi

$f_B(x, y)$ = nilai komponen B pada piksel x, y dari gambar referensi

$f_a(x, y)$ = nilai rata-rata dari jumlah nilai komponen RGB pada piksel x, y dari gambar referensi

T = *Threshold* atau ambang batas selisih nilai piksel dari kedua gambar

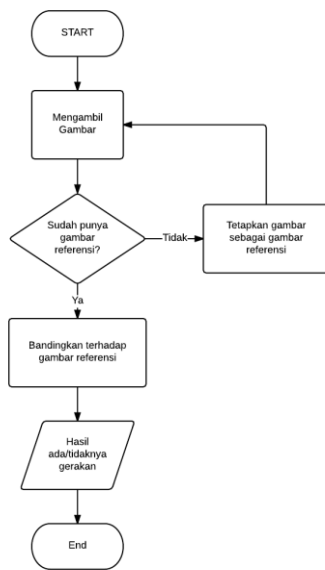
Persamaan 1 dan persamaan 2 digunakan untuk melakukan kalibrasi dengan menghitung rata-rata nilai dari komponen warna RGB pada tiap piksel dimana g_R , g_G , g_B merupakan komponen warna dari gambar yang diproses untuk

menentukan ada atau tidaknya gerakan, dan f_R , f_G , f_B adalah komponen warna RGB dari gambar referensi. Melalui persamaan 1 dan 2 diperoleh nilai g_a dan f_a yang merupakan nilai rata-rata tiap piksel untuk kedua gambar tersebut yang selanjutnya akan digunakan pada persamaan 3 untuk menentukan ada atau tidaknya perubahan piksel pada gambar berdasarkan besaran *threshold* yang telah ditentukan sehingga didapat hasil berupa data yang menyatakan piksel mana saja yang berubah melewati ambang batas (*threshold*).

$$D = \frac{\sum f_i^R(x, y) + \sum f_i^G(x, y) + \sum f_i^B(x, y)}{\sum f_R + \sum f_G + \sum f_B} \times 100 \quad (4)$$

Dengan menggunakan persamaan 4 dilakukan perhitungan persentase jumlah piksel yang berubah untuk menentukan ada atau tidaknya suatu gerakan dari gambar yang diproses. Dimana $\sum f_i^R(x, y)$, $\sum f_i^G(x, y)$, $\sum f_i^B(x, y)$ adalah jumlah piksel yang terdeteksi memiliki perbedaan dalam komponen warna RGB dan $\sum f_R$, $\sum f_G$, $\sum f_B$ adalah jumlah total dari seluruh piksel yang ada dalam gambar yang diambil dari tiap komponen warna RGB.

2.3.2 Static Template Matching

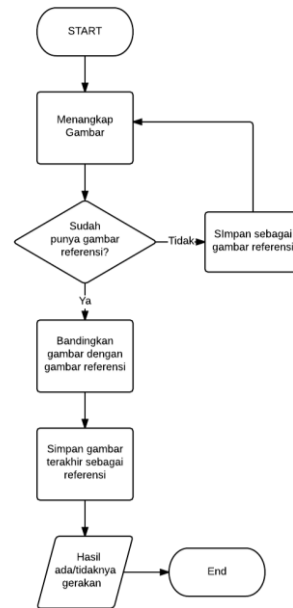


Gambar 1 Diagram Alir Algoritma *Static Template Matching*

Static template matching adalah suatu metode pendeteksian gerakan pada video dimana sistem menetapkan satu gambar sebagai referensi sebelum memulai perhitungan untuk menentukan ada atau tidaknya suatu gerakan [11]. Gambar tersebut nantinya akan dibandingkan dengan masukan dari video untuk menentukan ada atau tidaknya gerakan dalam video seperti yang dijelaskan dalam diagram alir pada Gambar 1.

Algoritma *Static template matching* masih memiliki beberapa kekurangan dalam mendeteksi suatu gerakan. Salah satu kelemahan dari algoritma ini adalah ketika ada objek yang masuk dalam sudut pandang kamera dan menetap dalam waktu yang cukup lama maka sistem akan terus menerus mendeteksi objek tersebut sebagai gerakan dalam video walaupun objek tersebut sudah tidak bergerak lagi

2.3.3 Dynamic Template Matching



Gambar 2 Diagram Alir Algoritma *Dynamic Template Matching*

Dynamic Template Matching adalah suatu metode yang digunakan dalam proses pengolahan citra untuk mendeteksi adanya atau tidaknya gerakan dalam urutan gambar atau video yang diterima dengan cara membandingkan gambar pada saat pemrosesan dengan gambar sebelumnya sebagai gambar acuan [11]. Proses perbandingan gambar tersebut dilakukan secara terus-menerus terhadap gambar acuan dimana setelah proses perbandingan selesai, gambar yang telah diproses selanjutnya digunakan sebagai gambar acuan untuk iterasi berikutnya seperti yang dijelaskan diagram alir.pada Gambar 2.

Metode *Dynamic Template Matching* masih memiliki kelemahan dalam hal pendeteksian gerakan objek yang cepat. Apabila terdapat objek yang masuk dalam sudut pandang kamera dengan cepat dan kemudian keluar dari sudut pandang kamera, maka sesaat setelah objek tersebut keluar dari pantauan kamera sistem akan menghasilkan keluaran yang menyatakan terdapat objek yang bergerak dalam gambar walaupun sebenarnya objek yang bergerak

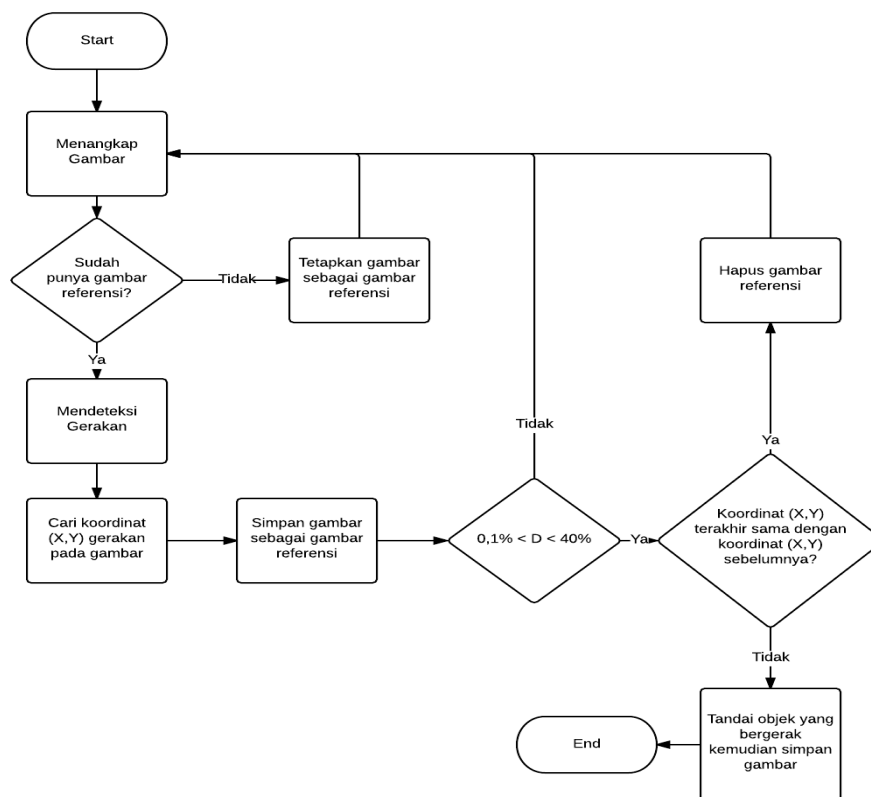
tersebut baru saja keluar dari pandangan kamera.

2.3.4 Dynamic and Adaptive Template Matching

Dynamic and Adaptive Template Matching adalah metode yang dikembangkan berdasarkan *Dynamic Template Matching* dengan tujuan untuk menghasilkan suatu metode yang lebih akurat dalam pendeteksian gerakan terhadap video [11]. Tahap pendeteksian gerakan yang dilakukan dalam metode ini adalah mengambil gambar referensi kemudian membandingkan gambar referensi tersebut dengan gambar berikutnya untuk mencari apakah ada perbedaan piksel pada kedua gambar kemudian sistem mengambil koordinat piksel-piksel yang memiliki perbedaan warna tersebut untuk selanjutnya dibandingkan dengan gerakan yang telah terdeteksi pada iterasi sebelumnya apakah

koordinat gerakan yang didapat sama seperti koordinat pada gerakan sebelumnya. Apabila kedua koordinat tersebut sama, maka sistem tidak akan memproses gambar tersebut lebih lanjut dikarenakan hasil yang sama telah didapat pada iterasi sebelumnya sehingga meminimalisir jumlah kesalahan pendeteksian gerakan. Keseluruhan tahap dari metode *Dynamic and Adaptive Template Matching* dapat dilihat dalam diagram alir pada Gambar 3.

Melihat kelebihan metode *Dynamic and Adaptive Template Matching* yang dibandingkan dengan metode *Static Template Matching* serta *Dynamic Template Matching*, maka dalam penelitian ini digunakan metode pendeteksian gerakan tersebut dalam sistem yang akan dibuat agar sistem tersebut lebih akurat dalam mendeteksi gerakan sekaligus menguji kehandalan metode tersebut dalam penerapannya.



Gambar 3 Diagram Alir Algoritma *Dynamic and Adaptive Motion Detection*

2.4 Raspberry Pi



Gambar 4 Raspberry Pi model B

Raspberry Pi adalah komputer seukuran kartu kredit yang dikembangkan di UK oleh Raspberry Pi Foundation dengan latar belakang untuk mempromosikan pembelajaran ilmu komputer di sekolah-sekolah [9]. Raspberry Pi menggunakan SoC (System on Chip) Broadcom BCM2835 dengan prosesor ARM1176JZF-S 700 MHz, GPU VideoCore IV dan memiliki kapasitas RAM sebesar 256 MB pada model A dan sebesar 512 MB untuk model B. Dalam unit Raspberry Pi tidak dibekali dengan storage unit seperti harddisk ataupun SSD melainkan menggunakan SD card sebagai unit penyimpanannya untuk melakukan booting dan untuk menyimpan data. Bentuk fisik dari Raspberry Pi dapat dilihat pada Gambar 4.

Raspberry Pi bersifat *open source* dan bisa dimodifikasi sesuai kebutuhan penggunaannya. Sistem operasi utama Raspberry Pi menggunakan *Debian GNU/Linux* dan bahasa pemrograman Python.

2.5 Google Cloud Messaging

Google Cloud Messaging adalah suatu layanan yang memungkinkan pengembang aplikasi untuk mengirim data dari servernya ke aplikasi *Android* secara *realtime*. Layanan ini menyediakan sebuah mekanisme sederhana yang tidak memakan

banyak sumber daya CPU yang memungkinkan server untuk melakukan kontak terhadap aplikasi *mobile* pada sistem operasi *Android* agar aplikasi tersebut segera mengambil data terbaru pada server.

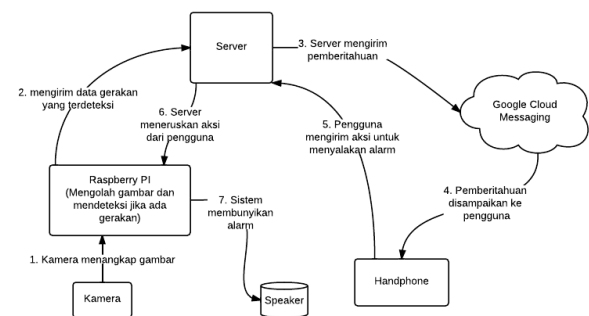
3. METODOLOGI PENELITIAN

Penelitian ini menggunakan metodologi penelitian yang dimulai dari studi pustaka. Metode studi pustaka yang dilakukan, yakni dengan cara mengkaji buku-buku yang berkaitan dengan pengolahan citra dan Raspberry Pi, literatur, halaman *web*, makalah hasil penelitian, serta jurnal-jurnal yang membahas algoritma-algoritma yang digunakan dalam pendeteksian gerakan pada video.

Tahap selanjutnya adalah perancangan yang diawali dengan analisis kebutuhan sistem dari perangkat lunak maupun perangkat keras. Tahap yang terakhir adalah pengujian. Pengujian dilakukan pada setiap sub sistem.

4. PERANCANGAN SISTEM

4.1 Alur Kerja Sistem



Gambar 5 Alur Kerja Sistem

Dalam sistem ini terdapat beberapa komponen penyusun dan dalam operasinya terdapat alur yang digambarkan pada Gambar 5. Berikut ini adalah penjelasan dari alur kerja sistem :

1. Modul kamera yang terpasang pada Raspberry Pi akan melakukan penangkapan gambar secara terus menerus dalam interval waktu tertentu kemudian diproses oleh Raspberry Pi

- untuk menemukan ada atau tidaknya suatu gerakan dari sudut pandang kamera
2. Apabila hasil pengolahan citra pada *Raspberry Pi* menunjukkan bahwa pada sudut pandang kamera terdapat gerakan, maka *Raspberry Pi* akan mengirimkan data berupa gambar berisi objek yang bergerak tersebut kepada server.
 3. Server meneruskan data yang dikirimkan oleh *Raspberry Pi* kepada *Google Cloud Messaging*.
 4. *Google Cloud Messaging* meneruskan data yang dikirim oleh server kepada ponsel pengguna dalam bentuk notifikasi yang juga turut memicu aplikasi pada ponsel pengguna untuk mengunduh gambar terbaru pada server.
 5. Setiap kali pengguna menerima gambar dari server, pengguna dapat mengirimkan perintah untuk menyalakan alarm pada sekitar kamera
 6. Perintah yang dikirimkan oleh pengguna untuk menyalakan alarm tidak langsung dikirimkan dari ponsel pengguna ke *Raspberry Pi*, melainkan melalui bantuan server.
 7. Setelah perintah untuk menyalakan alarm diterima oleh *Raspberry Pi*, maka sistem akan menyalakan alarm menggunakan speaker yang terpasang pada *Raspberry Pi*.

4.2 Perancangan Sistem Pendeteksi Gerakan pada *Raspberry Pi*

Berikut ini adalah kebutuhan *hardware* dalam perancangan sistem pendeteksi gerakan dengan *Raspberry Pi*:

1. *Raspberry Pi*
2. Kabel HDMI
3. TV atau monitor dengan port HDMI
4. Keyboard
5. *Micro SD* dengan *SD Card* adapter

6. Laptop
7. Kabel *Ethernet* dengan RJ45
8. *Webcam*
9. *Charger* dengan port mikro USB
10. Speaker

Sedangkan kebutuhan *software* untuk membangun sistem pendeteksi gerakan dengan *Raspberry Pi* adalah sebagai berikut:

1. File Sistem operasi *Raspbian*
2. Python
3. Pyaudio
4. OpenCV
5. Python Numpy

Pada sistem ini akan digunakan *Webcam Logitech QC Pro9000*. Alasan dipilihnya kamera tersebut dikarenakan kamera tersebut termasuk salah satu kamera yang sudah kompatibel dengan *Raspberry Pi* tanpa perlu melakukan instalasi driver dan memiliki kebutuhan daya yang rendah sehingga dapat dijalankan secara langsung dari USB port milik *Raspberry Pi*.

Sistem ini menerapkan metode *Adaptive Motion Detection* dalam mendeteksi gerakan serta akan mengirimkan gambar dari gerakan yang terdeteksi ke sebuah server dimana gambar tersebut nantinya akan diteruskan oleh server kepada *smartphone* penggunanya.

4.3 Perancangan Server

Server dalam sistem ini berperan sebagai penerima gambar hasil deteksi gerakan yang dilakukan oleh perangkat *Raspberry Pi*. Apabila perangkat *Raspberry Pi* menangkap adanya pergerakan, maka gambar dari gerakan yang tertangkap akan dikirimkan kepada server untuk kemudian diteruskan kepada pengguna melalui *push notification* menggunakan layanan *Google Cloud Messaging*. Tidak hanya itu, server dalam sistem ini juga berperan untuk mengirimkan perintah dari pengguna kepada perangkat *Raspberry Pi* untuk membunyikan alarm.

Peranan server dalam keseluruhan sistem ini adalah sebagai penghubung antara *Raspberry Pi* dengan aplikasi yang ada pada *smartphone* pengguna mulai dari menerima dan menyimpan gambar yang dikirimkan oleh *Raspberry Pi*, mengirim *push notification* kepada *smartphone* pengguna, serta meneruskan perintah dari pengguna untuk menyalakan atau mematikan alarm pada *Raspberri Pi*.

4.4 Perancangan Aplikasi Mobile

Aplikasi *mobile* pada sistem ini berfungsi untuk menerima *push notification* yang dikirimkan oleh server sebagai peringatan adanya gerakan yang terdeteksi oleh kamera pada perangkat *Raspberry Pi*. Selain menerima *push notification* aplikasi ini juga berguna untuk mengunduh gambar-gambar gerakan yang tertangkap yang dikirim oleh perangkat *Raspberry Pi*.

5. HASIL, IMPLEMENTASI DAN PEMBAHASAN

5.1 Hasil

Setelah melalui proses perancangan sistem dimana ketiga komponen utama dalam sistem yaitu server, aplikasi dan perangkat *Raspberry Pi* telah selesai dibuat dan telah dapat menjalankan fungsinya, maka ketiga komponen tersebut sudah siap untuk dilakukan implementasi guna mengetahui kemampuan dan manfaat dari sistem yang telah dibangun. Pada *Raspberry Pi* program pendeteksi gerakan telah dapat berjalan dengan baik sesuai dengan metode yang digunakan yaitu metode *Adaptive Motion Detection* dimana pertama-tama program melakukan proses *frame differencing* untuk mencari selisih dari dua *frame* gambar terakhir.



Gambar 6 Dua Gambar Yang Akan Diproses Dengan *Frame Differencing*

Terhadap kedua Gambar 6a dan 6b tersebut kemudian dilakukan proses *frame differencing* untuk menghasilkan sebuah gambar baru yang menunjukkan piksel mana saja pada gambar yang terdapat perubahan nilai atau dapat dikatakan bergerak.



Gambar 7 Gambar Hasil Proses *Frame Differencing*

Setelah proses *frame differencing* dilakukan maka selanjutnya hasil proses *frame differencing* pada Gambar 7 dari proses tersebut diolah dengan proses *thresholding* untuk mencari piksel-piksel mana saja yang nilainya melewati batas *threshold*. Pada gambar hasil proses *thresholding*, piksel yang nilainya melewati

batas *threshold* akan bernilai 1 dan berwarna putih sedangkan piksel yang nilainya dibawah *threshold* akan bernilai 0 dan berwarna hitam seperti yang terlihat pada Gambar 8.



Gambar 8 Gambar Hasil Proses *Thresholding*

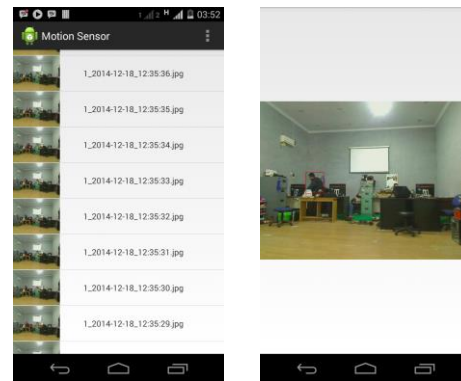
Apabila jumlah piksel yang berwarna putih pada gambar hasil *thresholding* berjumlah 0.1% hingga 40% dari gambar, maka sistem akan menyatakan bahwa dalam gambar tersebut terdapat pergerakan dan kemudian mencari koordinat terluar dan jumlah piksel yang berwarna putih pada gambar biner tersebut untuk menentukan daerah terjadinya gerakan. Dari koordinat yang didapat, maka akan digambar sebuah garis merah berbentuk persegi seperti pada Gambar 9 yang mewakili daerah terjadinya gerakan.



Gambar 9 Contoh Gambar Hasil Akhir Dari Sistem

Setiap gambar yang didapat dari hasil pendeteksian gerakan akan langsung segera dikirim oleh *Raspberry Pi* ke server dengan

menggunakan *request* HTTP POST. Gambar yang dikirim tersebut akan disimpan oleh server dan kemudian server akan mengirim nama file dari gambar tersebut kepada aplikasi dalam ponsel pengguna melalui layanan *Google Cloud Messaging*. Pengguna akan menerima pemberitahuan pada ponselnya jika sistem mendeteksi adanya gerakan dan pengguna dapat meng-klik pemberitahuan yang diterima untuk melihat tangkapan gambar dari gerakan yang terdeteksi seperti yang terlihat pada Gambar 10.



Gambar 10 Tampilan Halaman Pada Aplikasi *Android*

5.2 Implementasi

Untuk mengetahui seberapa efektif sistem yang dibuat dalam mendeteksi gerakan, maka dilakukan uji coba implementasi sistem dimana sistem tersebut akan dipasang pada beberapa lokasi berbeda baik itu pada dalam ruangan maupun luar ruangan selama kurang lebih 12 hingga 14 jam pada setiap lokasi percobaan.

Tabel 1 Data Hasil Implementasi

Lokasi	Objek Pengganggu	Gambar False Positive	Total Gambar	Persentase False Positive
Dalam Ruangan	Tidak ada	16	2217	0.72%
Luar Ruangan	Tidak ada	56	1654	3.38%
Dalam Ruangan	Televisi, Kipas Angin	5762	9808	58.75%

Luar Ruang	Tidak Ada	105	3114	3.37%
Luar Ruang	Genangan Air, pohon	4862	8880	54.75%

Kinerja sebuah sistem pendeteksian gerakan dapat ditentukan dari jumlah *false positive* yang dihasilkannya dimana semakin kecil jumlah *false positive* yang didapat maka semakin baik sistem tersebut dalam mendeteksi gerakan. Jumlah *false positive* yang didapat dalam implementasi tidak begitu dipengaruhi lokasi percobaan baik itu di dalam maupun di luar ruangan seperti yang dinyatakan pada Tabel 1 dimana persentase *false positive* pada dalam ruangan dan luar ruangan bisa saja bernilai kecil ataupun besar. Hal yang mempengaruhi besar kecilnya jumlah *false positive* yang didapat pada implementasi adalah sudut pandang kamera dan objek-objek yang ada dalam pandangan kamera. Hasil *false positive* yang tinggi akan didapat bila dalam pandangan kamera terdapat objek yang mudah bergerak atau objek lain yang bergerak secara konstan dimana dalam percobaan ini disebut sebagai objek pengganggu yang mengakibatkan tingginya nilai *false positive* yang didapat.

5.3 Pembahasan

Metode pengolahan citra dapat diterapkan untuk melakukan pencegahan terhadap tindak pencurian pada rumah-rumah kosong maupun ruangan lain yang ditinggal oleh penghuninya dengan cara melakukan pendeteksian gerakan terhadap setiap gambar yang ditangkap menggunakan kamera untuk mengawasi ruangan tersebut. Gerakan yang terjadi dalam jarak pandang kamera dapat diketahui oleh sistem dan membantu dalam mencegah terjadinya tindak pencurian dikarenakan setiap gerakan yang didapat oleh sistem tersebut akan disampaikan kepada penghuninya.

Dalam implementasi pengolahan citra dengan metode *Adaptive Motion Detection* pada perangkat *Raspberry Pi* digunakan bantuan pustaka *OpenCV* yang sudah menyediakan fungsi-fungsi dasar dalam proses pengolahan citra seperti *frame differencing* dan *thresholding* yang digunakan dalam metode yang digunakan. Dengan bantuan pustaka *OpenCV* tersebut, maka untuk menerapkan metode *Adaptive Motion Detection* pada perangkat *Raspberry Pi* cukup dibuat sebuah program yang berisi serangkaian fungsi-fungsi dari *OpenCV* yang disusun sesuai dengan algoritma dari metode yang digunakan.

Metode pendeteksian gerakan yang digunakan sudah efektif dalam mendeteksi setiap gerakan yang terjadi di dalam ruangan maupun di luar ruangan. Akan tetapi, apabila dalam pandangan kamera terdapat objek-objek yang mudah bergerak seperti kipas angin, televisi, pohon, air hujan, serta gerakan air pada kolam atau sungai maka sistem akan menghasilkan banyak *false positive*.

Tindakan yang dapat dilakukan untuk meningkatkan efektifitas pendeteksian gerakan dengan metode tersebut adalah dengan menambahkan proses *blur* ringan dengan radius 3 piksel pada hasil selisih *frame* sebelum dilakukan *thresholding*. Proses *blur* tersebut berfungsi untuk menghilangkan noise yang ditangkap oleh kamera dimana noise tersebut akan sangat berpengaruh terhadap hasil koordinat gerakan yang didapat. Kemudian, menghilangkan batasan jumlah piksel yang berubah untuk dikatakan sebagai gerakan yang dalam metode tersebut disarankan sekitar 0.1% sampai 40%. Hal itu dikarenakan jumlah piksel yang berubah dalam suatu gambar sangat dipengaruhi oleh ukuran dan jauh atau dekatnya sebuah objek yang bergerak.

Untuk dapat menyampaikan informasi dari hasil pengolahan citra pada *Raspberry*

Pi kepada penggunanya, dibutuhkan bantuan sebuah server sebagai penghubung antara pengguna dengan *Raspberry Pi*. Kepada pengguna diberikan sebuah aplikasi yang dipasang pada *smartphone* dengan sistem operasi *Android* dimana aplikasi tersebut akan menerima *push notification* yang dikirimkan dari server setiap kali terdapat gerakan yang ditangkap oleh kamera pada *Raspberry Pi*. Pada sisi server digunakan *NodeJS* sebagai komponen dasar server yang menghubungkan perangkat *Raspberry Pi* dengan aplikasi.

Dalam aplikasi yang dipasang pada ponsel pengguna disertakan fitur untuk menyalakan alarm pada ruangan yang diawasi sehingga pengguna dapat mengirimkan aksi secara langsung apabila pada tangkapan gambar yang diterimanya dari perangkat *Raspberry Pi* terdapat gerakan yang mengindikasikan adanya tindak pencurian.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Setelah melalui proses perancangan, implementasi sistem serta penelitian yang telah dilakukan, didapat beberapa kesimpulan sebagai berikut:

1. Sistem akan sangat membantu dalam pengawasan terhadap rumah kosong yang ditinggal penghuninya ataupun untuk pengawasan terhadap gudang maupun brankas dari tindak pencurian.
2. Sistem ini tidak dapat menggantikan fungsi sepenuhnya dari kamera pengawas atau CCTV dikarenakan kamera CCTV merekam secara terus menerus baik itu ada atau tidaknya gerakan sedangkan kamera pada sistem yang dibangun hanya menangkap potongan gambar dari gerakan-gerakan yang terjadi dalam jangkauan pandangannya.
3. Pengolahan citra dengan Metode Adaptive Motion Detection dapat diterapkan dan dijalankan dengan lancar oleh *Raspberry Pi*. Dengan

bantuan sebuah webcam, *Raspberry Pi* dapat menangkap gambar gerakan yang terjadi dalam jarak pandang kamera tersebut kemudian memproses hasil tangkapan gambar dari gerakan yang terjadi dan mengirimkan gerakan yang terdeteksi kepada penggunanya.

4. Implementasi metode Adaptive Motion Detection ke dalam suatu sistem yang berfungsi untuk mendeteksi ada atau tidaknya suatu gerakan merupakan salah satu penerapan metode tersebut dalam kehidupan sehari-hari.
5. Metode Adaptive Motion Detection mampu mendeteksi gerakan yang terjadi dalam jarak pandang kamera dengan baik pada saat diletakkan didalam maupun diluar ruangan dengan syarat dalam jarak pandang kamera tidak terdapat objek yang bergerak secara konstan ataupun objek yang mudah bergerak seperti genangan air, pohon, televisi, lampu hias dan lain-lain.
6. Dalam metode Adaptive Motion Detection tidak ada proses pengurangan noise pada tangkapan kamera. Padahal dengan adanya noise dalam gambar akan mengurangi efektivitas pendeteksian gerakan dikarenakan sistem akan menganggap setiap piksel noise sebagai gerakan.

6.2 Saran

1. Hasil penelitian diharapkan dapat menjadi referensi ataupun perbandingan untuk membangun sebuah sistem yang mendeteksi gerakan.
2. Untuk penelitian selanjutnya, dapat dibuat agar sistem tidak lagi menangkap gerakan yang terjadi dalam bentuk gambar, melainkan dalam bentuk video.
3. Selain menggunakan metode Adaptive Motion Detection diharapkan untuk pengembangan selanjutnya metode yang digunakan dapat bekerja efektif dalam mendeteksi gerakan yang terjadi diluar ruangan tanpa menghasilkan banyak false positive.

DAFTAR PUSTAKA

- [1] Bradski, Gary dan Adrian Kehler. 2008. *Learning OpenCV*. Sebastopol: O'Reilly Media, Inc.
- [2] Brahmbhatt, Samarth. 2013. *Practical OpenCV*. California: Apress Media
- [3] Howse, Joseph. 2013. *OpenCV Computer Vision with Python*. Birmingham: Packt Publishing Ltd.
- [4] Isa, Sani M dan Manatap Dolok Lauro. 2006. *Aplikasi Pendeteksi Gerakan Menggunakan Metode Spatial Domain dengan Pelaporan Otomatis ke Telepon Genggam*. Fakultas Teknologi Informasi, Universitas Tarumanegara, Jakarta.
- [5] Krig, Scott. 2014. *Computer Vision Metrics*. California: Apress Media
- [6] Membrey, Peter dan David Hows. 2013. *Learn Raspberry Pi with Linux*. California: Apress Media
- [7] Migliore, Davide A; Matteucci, Mateo dan Naccari, Mateo. 2006. *A Revaluation of Frame Difference in Fast and Robust Motion Detection*. Politecnico di Milano, Italy.
- [8] Monk, Simon. 2014. *Raspberry Pi Cookbook*. Sebastopol: O'Reilly Media, Inc.
- [9] Richardson, Matt dan Shawn Wallace. 2013. *Getting Started with Raspberry Pi*. Sebastopol: O'Reilly Media, Inc.
- [10] Santoso, Yonatan; Setiyawan, Iwan dan N. Papilaya, Victor. 2009. *Penerapan Kamera Web Sebagai Pendeteksi Gerakan dengan Antar Muka Directshow*. Makara, Teknologi, Vol. 13 (I). Hlm 15-18.
- [11] Widyawan, Muhammad Ihsan Zul dan Lukito Edi Nugroho. 2012. *Adaptive Motion Detection Algorithm using Frame Differences and Dynamic Template Matching Method*. Department of Electrical Engineering and Information Technology, Gadjah Mada University, Yogyakarta, Indonesia.