

## PENGEMBANGAN SISTEM KAMERA KEAMANAN DENGAN PENYIMPANAN *DATABASE CLOUD* DAN NOTIFIKASI MELALUI *ANDROID*

Muhammad Rahmadi Surya<sup>1)</sup> Susijanto Tri Rasmana<sup>2)</sup> Musayyanah<sup>3)</sup>

Program Studi/Jurusan Teknik Komputer  
Institut Bisnis dan Informatika Stikom Surabaya  
Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1)[muhammadrahmadi30@gmail.com](mailto:muhammadrahmadi30@gmail.com), 2)[Susyanto@stikom.edu](mailto:Susyanto@stikom.edu), 3)[Musayyanah@stikom.edu](mailto:Musayyanah@stikom.edu)

**Abstrak:** Pada suatu sistem keamanan konvensional dibutuhkan sebuah DVR (*Digital Video Recorder*). DVR berperan sebagai media pengolah dan penyimpanan citra dari kamera CCTV (*Close Circuit Television*). Cara kerja dari kamera ini dengan merekam keadaan rumah yang diamati, dan disimpan ke DVR (*Digital Video Recorder*) sebagai video yang bersumber dari CCTV, sehingga Kualitas kamera CCTV berpengaruh terhadap hasil penyimpanan. Kapasitas penyimpanan pada perangkat DVR bergantung pada ukuran hardisk yang digunakan pada perangkat tersebut, sehingga hanya dapat menyimpan rekaman video maksimal 20 hingga 30 hari rekaman video dengan kualitas normal dengan kualitas normal dan kapasitas *hardisk 1 terabyte*. Jika lebih dari itu, maka data otomatis dihapus. Solusinya adalah membuat kamera otomatis yang hanya menyimpan ketika mendeteksi gerakan, dan tanpa menyimpan jika tidak ada pergerakan. Untuk menambah tingkat keamanan penyimpanan, digunakan *database cloud firebase* dikarenakan sistem penyimpanan memori *hardware* sering rusak/*corrupt*. Sistem yang dibuat pada penelitian ini menunjukkan bahwa sistem dapat mendeteksi pergerakan manusia dengan akurasi 100% sedangkan deteksi benda bergerak dengan akurasi 90%. Penyimpanan data pun sudah berada di *database cloud firebase* dan tersimpan dengan baik. Serta aplikasi *android* dapat memberi notifikasi jika terjadi pergerakan dan mengirim data tangkapan kamera ke *mobile android* secara *real time* dan sesuai *database*.

**Kata kunci:** Keamanan, *Background Substraction*, *Firebase*, Notifikasi

### PENDAHULUAN

Berbagai macam permasalahan terus terjadi di masyarakat, contohnya masalah pada sistem keamanan yang tidak ada habisnya. Permasalahan ini bermula dari tingkat keamanan yang tidak memenuhi standar keamanan. Berdasarkan data dari Markas besar Polri, terdapat 316.500 kasus kejahatan yang terdiri dari kasus konvensional 304.835 kasus, kasus transional 7.171 kasus, dan kasus kekayaan negara 3.844 kasus. Sehingga setiap 91 detik terjadi satu kejahatan di Indonesia pada tahun 2012 (Tempo, 2013). Maka dibutuhkan pengawasan yang lebih dalam menjaga keamanan rumah/ruko tersebut. Kurang standarnya keamanan rumah merupakan sumber-sumber permasalahan yang ada di sekeliling kehidupan. Kadang, meskipun adanya petugas keamanan masih resiko dengan keamanan. Petugas keamanan juga merupakan manusia yang tak luput dari kesalahan. Maka dari itu, dengan berkembangnya teknologi tentunya bisa sangat membantu di sistem keamanan rumah ini.

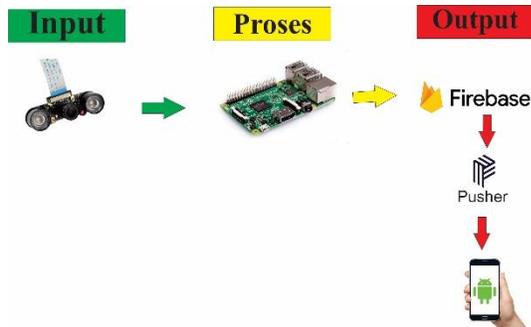
Pada penelitian sebelumnya (Widipratama, M. Z. 2017) yang berjudul "Sistem Pemantau Keamanan Menggunakan Kamera Dengan Metode *Background Substraction*" memiliki prinsip hampir sama yaitu mendeteksi gerakan menggunakan metode *Background*

*Substraction* dan *face detection* dengan penyimpanan hasil data di memori SDHC *Raspberry Pi* serta menggunakan notifikasi melalui sms. Oleh karena itu, pada penelitian selanjutnya dibuat sistem "Sistem Kamera Keamanan Dengan Penyimpanan Database Cloud Dan Notifikasi Melalui Android" dimana penyimpanan menggunakan *database firebase*, sehingga dapat meminimalisir penyimpanan internal pada *Raspberry pi*. Selain itu, sistem menggunakan metode *background subtraction* dan metode *human detection* sebagai pembatas komputasi.

*Raspberry Pi* dapat menggantikan sistem dari sebuah DVR dengan fitur yang dimilikinya. Selain ukuran *Raspberry Pi* yang lebih kecil, harganya pun relatif lebih murah dibandingkan dengan sebuah DVR. *Raspberry Pi* disini berperan sebagai pengolah data apabila terdeteksi suatu gerakan, *Raspberry Pi* menyimpan hasil citra gambar ke Memori SDHC yang tertanam di sistem *Raspberry Pi* kemudian diunggah ke *database firebase*. Apabila pengunggahan data sukses maka data yang tersimpan di memori dihapus secara otomatis. Sistem memiliki peringatan berupa notifikasi pada android sehingga pengguna dapat memantau dimanapun. Notifikasi ini menampilkan data berupa *string* dan citra gambar

## METODE PENELITIAN

Perancangan perangkat sistem meliputi unit masukan dan keluaran pada *mini pc Raspberry pi*. Perancangan blok diagram pada sistem ini dapat dilihat pada gambar 1.



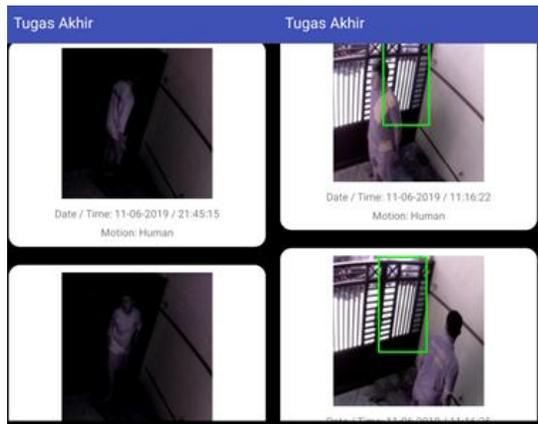
Gambar 1. Diagram blok

### Raspberry Pi Camera Module Support Night Vision

Kamera yang dipakai adalah modul kamera dengan *Adjustable-Focus* yang *universal* untuk semua tipe *Raspberry pi*. Modul kamera dilengkapi dengan dual *Infrared LED Board* yang mendukung fitur malam hari sehingga memungkinkan untuk melihat dalam kondisi cahaya rendah di malam hari ataupun cahaya yang terang di siang hari.



Gambar 2. Kamera Night Vision



Gambar 3. Hasil kamera malam(kiri) dan siang(kanan)  
**Background Subtraction**

*Background subtraction* adalah suatu proses yang digunakan untuk menemukan suatu objek pada gambar

dengan cara membandingkan gambar yang ada dengan latar belakang pada gambar tersebut sebelumnya. Prosedur *background subtraction* terdiri menjadi 3 tahap, yaitu *preprocessing*, *background subtraction*, dan *thresholding* (Ikhsan Samir, 2016).

### Tahap Preprocessing

*Preprocessing* merupakan tahap awal di metode *background subtraction*, dimana gambar awal yang masih berupa warna RGB (*RED, GREEN, BLUE*) diubah menjadi warna *grayscale* agar mudah saat perhitungan pergerakan gambar. *Grayscale* adalah *frame* yang berwarna gradasi dari hitam ke putih atau putih ke hitam. Piksel *grayscale* memiliki nilai 8 bit atau setara nilai 256. Rumus dibawah ini merupakan salah satu rumus untuk mengubah *frame* RGB menjadi *grayscale*.

$$Y=0.2*R+0,5*G+0,2*B$$

Y = nilai *graycale*

R = nilai piksel *Red*

G = nilai piksel *Green*

B = nilai piksel *Blue*

### Tahap Background Subtraction

Tahapan ini adalah tahapan inti dari metode *background subtraction*, yaitu suatu tahapan yang menjadikan atau menghitung perbedaan nilai piksel menjadi *grayscale* antar satu *frame* dengan *frame* selanjutnya, Sehingga setiap terjadi gerakan pasti nilai antar *frame* berubah. Metode *background subtraction* banyak digunakan dalam beberapa *project* pengolahan citra contohnya pendeteksian mobil, manusia, hewan dan sebagainya. Perhitungannya sangat mudah yaitu *frame* awal dikurangi *frame* selanjutnya. Ide dasar dari *Background subtraction* adalah  $|frame(n) - frame(m)| > threshold$ , (n) adalah *frame* saat ini dan (m) adalah *frame* selanjutnya. Bila *Piksel* ke (n) memenuhi persamaan tersebut, maka *Piksel* tersebut digolongkan kedalam kelompok *Piksel* objek dan selain itu adalah latar.

### Tahap Thresholding

*Thresholding* adalah tahapan terakhir dari metode *background subtraction*, yaitu penetapan nilai hasil 0 dan 1 berdasarkan penetapan nilai *thresholding* atau jumlah sisa pengurangan *background subtraction*. Nanti dapat ditentukan diatur berapakah nilai hasil pengurangan jika didapatkan suatu gerakan.

### Histogram of Oriented Gradients (HOG)

*Histogram of oriented gradients* adalah suatu metode untuk mendeteksi atau analisa bentuk tubuh manusia, sehingga di penelitian ini sangat berguna untuk membantu efisiensi data. Dikarenakan data yang diambil hanya ketika terjadi gerakan yang signifikan dan dideteksi manusia atau bukan sebagai perbandingan data. Proses metode HOG adalah mendeteksi bentuk manusia per

frame gambar di pecahkan file video. Pendeteksian manusia dengan metode HOG berdasarkan pendeteksian wajah saja atau *fullbody*, dimana deteksinya meliputi wajah, bentuk tangan dan bentuk kaki. Gambar 4 merupakan contoh hasil deteksi HOG pendeteksian *fullbody* (Kusno Suryadi, 2015).



Gambar 4. Hasil deteksi HOG

### Database Cloud Firebase

*Firebase* merupakan suatu layanan DbaaS (*Database as a Service*) dengan data *realtime*. Tidak hanya untuk menyimpan data, *Firebase* juga menyediakan API untuk penggunaan *web socket*. *Firebase* menyediakan *hosting* untuk *static file* tersebut lengkap dengan fasilitas CDN dan SSL. *Firebase* menyediakan fitur kapasitas *database* sampai 50 GB. Tetapi yang bisa didapatkan secara gratis hanya 1 GB. Di penelitian ini tidak didapatkan masalah meskipun dibatasi hanya 1 GB karena data yang disimpan di *database* hanya berupa data *string* dan data gambar. Gambar dibawah ini merupakan contoh tampilan *database cloud firebase* (Oriza Ahmad, 2014).

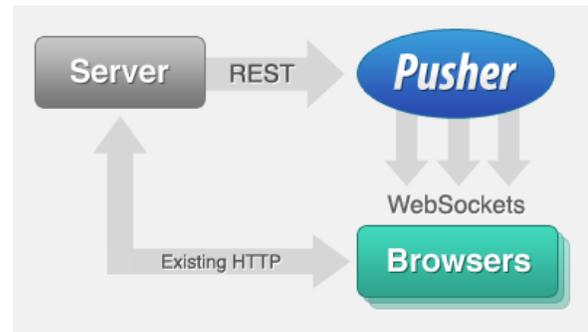


Gambar 6. Contoh tampilan data *string realtime* database *firebase*

### Pusher

*Pusher* merupakan *web server* yang termasuk layanan yang sangat dibutuhkan dan SDK khusus untuk mengelola *token push* perangkat aplikasi *android* atau *iOS*. *Pusher* masih berkaitan dengan *database cloud firebase*. Gambar 7 merupakan contoh cara kerja *pusher* yang menyediakan wadah terminal *server* ke *browser*

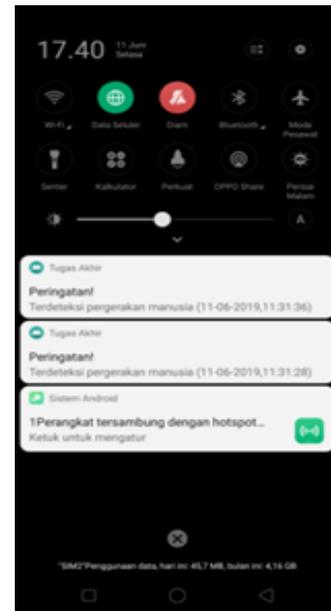
untuk memberi text *message/notifikasi*. *Browser* disini yang dimaksud adalah *Firebase Cloud*.



Gambar 7. Cara kerja *pusher*

### Notifikasi Android

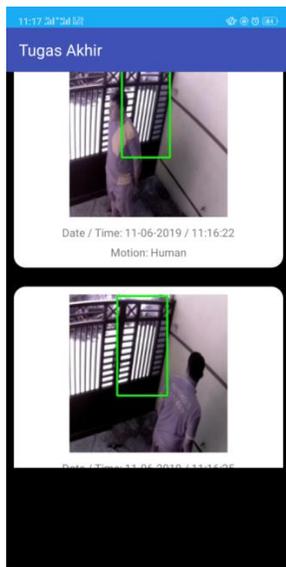
Program *Android* sederhana dibuat untuk menampilkan isi data dari *database cloud firebase*. *Pusher* sebagai pihak ketiga membantu pemberian *firebase cloud message* di notifikasi aplikasi *android* tersebut ketika ada data baru masuk *database*, sehingga para *user* nantinya bisa mengetahui jika terjadi suatu gerakan yang perlu dilihat. Adapun sebagai contoh hasil notifikasi aplikasi seperti gambar 7.



Gambar 7. Contoh hasil notifikasi

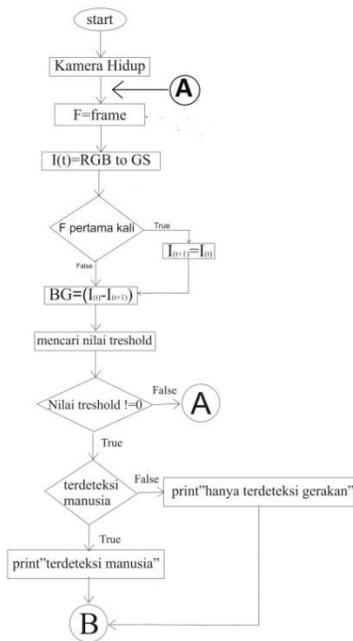
Dinotifikasi tersebut diberi jeda/*delay* agar data tidak bergandeng terus menerus dan data tidak *double*. Terdapat dua jenis notifikasi yang ditampilkan yaitu *motion/ gerakan manusia (human)* atau bukan manusia (*nonhuman*) yang dibedakan nanti jarak waktu *delay* perdata masuk. Adapun tampilan program pada gambar

dibawah ini terdapat data gambar yang disertai detail waktu beserta keterangan pergerakan (*motion*).

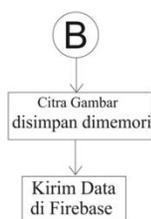


Gambar 8. Tampilan program android

**Flowchart**



Gambar 9. Flowchart program 1

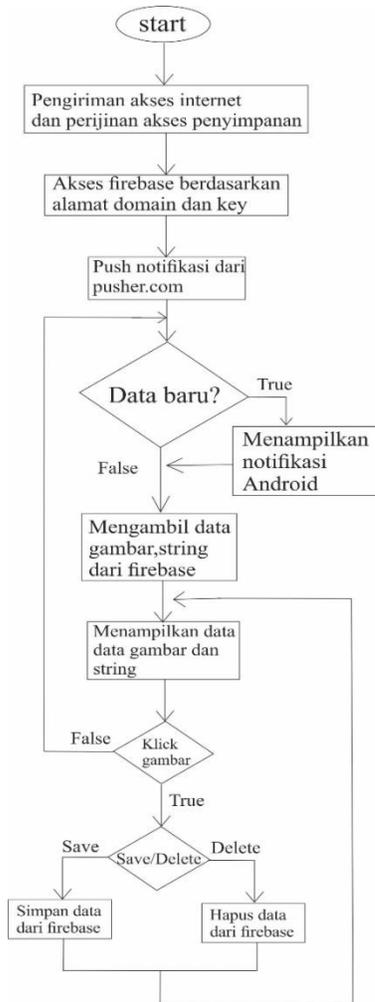


Gambar 10. Flowchart program 2

Gambar 9 dan 10 merupakan *flowchart* dari sistem Pengolahan data hasil kamera keamanan di *database cloud* berdasarkan deteksi gerakan dengan metode *background subtraction*. Dimulai dari *start* maka kamera hidup otomatis masih membuka *frame* baru dan selanjutnya *frame* diubah dari RGB ke *grayscale*. Setelah itu dilakukan percabangan ketika kondisi sudah ada atau tidak *frame* dibandingkan. Jika kondisi masih belum ada maka dilakukan inisialisasi untuk pertama kali agar ketika perhitungan *background subtraction* ada perhitungan antara 2 *frame*.  $BG (Background Substraction) = I(t) - I(t+1)$  adalah perhitungan antara hitungan *frame* lama dikurangi dengan *frame* baru.

Kemudian menuju menghitung nilai perpixel dalam satu *frame* yang dihasilkan dari perhitungan *frame* lama dengan *frame* baru/menghitung nilai *threshold*. Jika hasil perhitungan nilai *threshold* sama dengan 0 maka kembali ke inisialisasi *frame*/ (A). Tetapi jika nilai *threshold* tidak sama dengan 0 maka masuk percabangan deteksi manusia. Setelah masuk dipercabangan deteksi manusia didapatkan objek manusia maka membuat data string bahwa terjadi deteksi manusia dan dikirim ke(B). Jika tidak ada terdeteksi objek manusia maka dibuat data string bahwa hanya terjadi sebuah pergerakan dan dikirim ke (B). Masuk ke *flowchart 2 raspberry* menyimpan data ke memori. Kemudian data dikirim ke *database firebase*. Jika sudah tersimpan di *database firebase* dilakukan penghapusan otomatis data yang telah terkirim di *database*. Setelah itu sistem *android* mengambil data keterangan notifikasi beserta gambar kemudian *android* menampilkan notifikasi. Kemudian *frame* terahir kembali

lagi ke awal untuk dibandingkan lagi dengan *frame* selanjutnya/kembali ke (A).



Gambar 11. Flowchart Android

Gambar 11 merupakan *flowchart* dari program *android* yang digunakan untuk memberi notifikasi/peringatan dan menampilkan data yang diperoleh dari alat kepada *user*. Dimulai dari start program apk secara otomatis meminta persetujuan akses penyimpanan agar *user* dapat menyimpan data ke penyimpanan *android* tersebut. Selanjutnya mengakses alamat domain dan *key database firebase* agar mendapatkan data yang tersimpan bisa ditampilkan di jendela tampilan program. Berikutnya adalah mengakses website server *pusher.com* agar membantu *database firebase* memberikan notifikasi berupa *message text* agar muncul di jendela notifikasi android. Dan percabangan apakah di database *firebase* terbaca ada data baru jika *true*, maka menampilkan notifikasi di jendela notifikasi *android* jika ada baru. Setelah itu mengambil data yang tersimpan di *database firebase* untuk ditampilkan di jendela aplikasi *android*. Ketika data gambar ditekan yang tampil di jendela aplikasi maka tampil 2 pilihan menu yaitu

*save/delete*. Menu *save/delete* adalah untuk menghapus atau menyimpan data yang dipilih dari *database firebase* juga bukan hanya di aplikasi *android*.

## HASIL DAN PEMBAHASAN

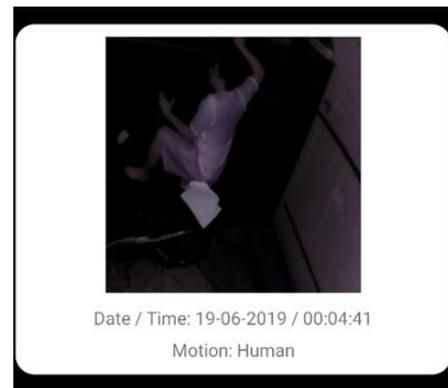
### Hasil Pengujian Pendeteksian Gerakan

Pengujian ini digunakan untuk mengetahui seberapa akurat kamera dapat mendeteksi gerakan di beberapa percobaan objek yang bergerak atau tidak dan bisa membedakan objek manusia atau bukan.

Tabel 1. Data uji gerakan manusia

Data Ke -	Human		Jam	Detect	
	Berhasil	Gagal		Menit	Detik
23	v		09	29	18
24	v		09	29	35
25	v		09	29	40
26	v		09	29	43
27	v		09	29	49
28	v		00	3	53
29	v		00	3	57
30	v		00	4	0
31	v		00	4	6
32	v		00	4	9
Total	10	0			

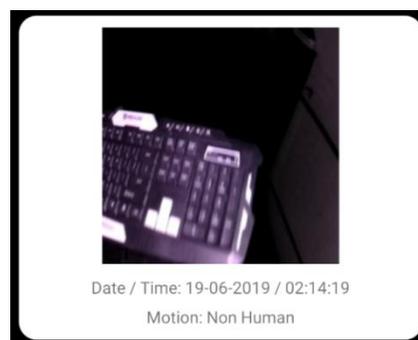
Dari hasil diagram diatas, dapat dijelaskan bahwa dari 10 data sampel uji coba terhadap siang dan malam hari sistem dapat mendeteksi gerak dengan baik dan akurat. Sehingga persentase data yang berhasil untuk uji deteksi gerakan manusia sebesar 100%.



Gambar 12. Hasil deteksi berhasil *human* ketika malam hari



Gambar 13. Hasil deteksi berhasil *human* ketika pagi hari



Gambar 15. Deteksi berhasil gerakan benda keyboard pada malam hari

Tabel 2. Data uji gerakan bukan manusia

Data Ke	Non-Human		Detect		
	Berhasil	Gagal	Jam	Menit	Detik
2	v		9	31	35
3	v		9	31	53
5	v		9	32	9
6	v		9	34	20
7	v		9	37	55
8	v		2	4	54
9	v		2	6	54
10		v	2	6	57
11	v		2	12	38
12	v		2	14	19
Total	9	1			

Dari hasil diagram diatas, dapat dijelaskan bahwa dari 10 data sampel uji coba terhadap jenis gerakan peralatan di siang dan malam hari dapat mendeteksi gerak benda dengan baik, sehingga persentase data yang berhasil untuk uji deteksi gerakan bukan manusia sebesar 90%. Terdapat kegagalan 1 data ke-10 dengan gerakan orang berpayung, sehingga sistem gagal mendeteksi bahwa gambar yang didapat adalah termasuk ada gerakan manusia.



Gambar 14. Deteksi gagal gerakan orang berpayung



Gambar 16. Deteksi berhasil gerakan benda sapu pada pagi hari

Dari hasil dua tabel (tabel 1 dan 2) diatas yaitu 10 data pergerakan manusia dan 10 data pergerakan benda. Didapatkan data human tingkat keberhasilan sebesar 100% dan *nonhuman* tingkat keberhasilan 90%. Dapat disimpulkan bahwa sistem ini lebih akurat digunakan jika mendeteksi pergerakan manusia. Tetapi hasil ini tetap bisa dikatakan berhasil karena di sistem keamanan justru yang lebih penting adalah mengawasi pergerakan manusia, dan untuk gerakan bukan manusia hanya dibuat dilakukan pembandingan deteksi.

### Hasil Pengujian Aplikasi Notifikasi *Android*

Pengujian ini untuk mengetahui apakah aplikasi berbasis *android* dapat memberi notifikasi jika terjadi pergerakan dan mengirim data tangkapan kamera kemobile *android* secara *real time* atau terdapat masalah ataupun terjadi *delay*. Dari pengujian bawah diambil 20 data *sample*, dan didapatkan data *realtime* waktu data masuk. Data dibandingkan dengan data yang lain dan diambil rata-rata jarak waktu notifikasi masuk, sehingga bisa didapatkan nilai jarak waktu masuk perdata. Data dibuat 2 tabel,yaitu human dan nonhuman. Dibedakannya data karena dari settingan program ada perbedaan juga *delay* notif human dengan *nonhuman*. Jarak waktu notifikasi *nonhuman* lebih lama daripada *human*.

Tabel 3. Uji data *delay human*

Data Ke -	Human		Detect			Report			Delay (detik)
	Berhasil	Gagal	Jam	Menit	Detik	Jam	Menit	Detik	
23	v		00	3	27	00	3	27	0
24	v		00	3	34	00	3	34	7
25	v		00	3	42	00	3	42	8
26	v		00	3	46	00	3	46	4
27	v		00	3	49	00	3	49	3
28	v		00	3	53	00	3	53	4
29	v		00	3	57	00	3	57	4
30	v		00	4	0	00	3	0	3
31	v		00	4	6	00	3	6	6
32	v		00	4	9	00	3	9	3
Total	10	0							
Rata-rata									4.2

Tabel 4. Uji data delay nonhuman

Data Ke -	Non-Human		Detect			Report			Delay (detik)
	Berhasil	Gagal	Jam	Menit	Detik	Jam	Menit	Detik	
2	v		1	48	53	1	48	53	0
3	v		1	49	9	1	49	9	9
5	v		1	52	9	1	52	9	180
6	v		1	53	11	1	53	11	62
7	v		1	55	56	1	55	56	165
8	v		2	4	54	2	4	54	534
9	v		2	6	54	2	6	54	118
10		v	2	6	57	2	6	57	3
11	v		2	12	38	2	12	38	398
12	v		2	14	19	2	14	19	101
Total	10	0							
Rata-rata									157

Dari dua tabel diatas didapatkan bahwa rata-rata jarak notifikasi *human* sebesar 4.2 detik dan *nonhuman* 157 detik atau 2 menit 37 detik. Hasil ini sudah sesuai sistem yang lebih mengutamakan notifikasi *human* daripada *nonhuman*. Dikarenakan didalam sistem keamanan lebih baik jika sistem tersebut mengawasi hanya pergerakan manusia. Tetapi, gerakan *nonhuman* hanya sebagai perbandingan. Di sistem juga memang diatur bahwa jeda *delay* perbandingan *human delay* sebesar 60 *frame* dan *nonhuman* 300 *frame*.

#### Pengujian Aplikasi Database Cloud Firebase

Pengujian ini digunakan untuk mengetahui apakah aplikasi *database cloud firebase* dapat membantu pengguna sistem keamanan untuk menyimpan data gambar dengan baik dan dapat dengan mudah digunakan

oleh pengguna ketika ingin melihat hasil gambar. Dari hasil pengambilan 10 sampel data pergerakan manusia dan 10 data bukan pergerakan manusia, didapatkan data seperti berikut:

Tabel 5. Uji data database human

Data Ke -	Human		Detect			database	
	Berhasil	Gagal	Jam	Menit	Detik	Berhasil	Gagal
23	v		00	3	27	v	
24	v		00	3	34	v	
25	v		00	3	42	v	
26	v		00	3	46	v	
27	v		00	3	49	v	
28	v		00	3	53	v	
29	v		00	3	57	v	
30	v		00	4	0	v	
31	v		00	4	6	v	
32	v		00	4	9	v	
Total	10					10	

Dari hasil data uji diatas, dapat dijelaskan bahwa dari 10 data sampel uji coba terhadap 10 Jenis data *human* dapat mengirim ke *database firebase* dengan berjalan baik dan akurat. Sehingga persentase data yang berhasil untuk uji database gerakan manusia sebesar 100%.

Tabel 6. Uji data database nonhuman

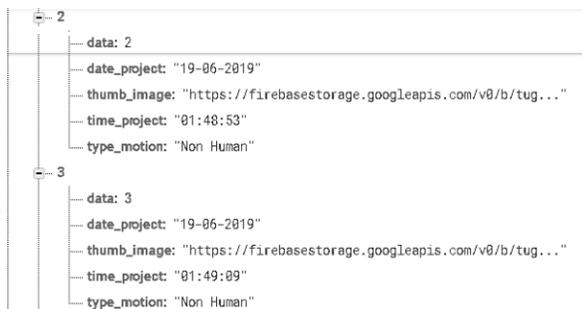
Data Ke -	Non-Human		Detect			Database	
	Berhasil	Gagal	Jam	Menit	Detik	berhasil	gagal
2	v		1	48	53	v	
3	v		1	49	9	v	
5	v		1	52	9	v	
6	v		1	53	11	v	
7	v		1	55	56	v	
8	v		2	4	54	v	
9	v		2	6	54	v	
10		v	2	6	57	v	
11	v		2	12	38	v	
12	v		2	14	19	v	
Total	9	1				10	0

Dari hasil data uji diatas, dapat dijelaskan bahwa dari 10 data sampel uji coba terhadap 10 Jenis data nonhuman dapat mengirim ke *database firebase* dengan berjalan baik dan akurat, sehingga persentase data yang berhasil untuk uji database gerakan bukan manusia sebesar 100%.

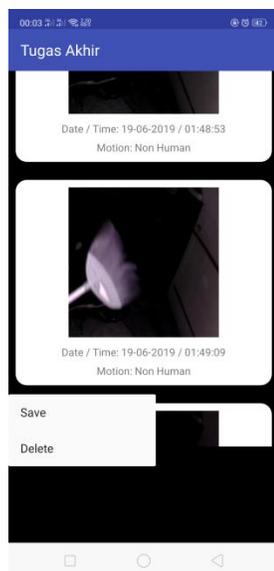
Tabel 7. Uji hapus data dari database maupun android mobile

Delete Data Ke -	Firebase		Android	
	Berhasil	Gagal	Berhasil	Gagal
2	v		v	
3	v		v	
5	v		v	
6	v		v	
7	v		v	
23	v		v	
24	v		v	
25	v		v	
26	v		v	
27	v		v	
Total	10	0	10	0

Dari pengamatan tabel 5 dan 6 didapatkan bahwa program *database* dapat berjalan dengan akurat dari pengunggahan data sampai penghapusan data di tabel 7 berjalan sangat baik. Sehingga dapat disimpulkan bahwa database cloud firebase terhadap program sistem keamanan dapat sangat membantu untuk penyimpanan data. Gambar 17 merupakan contoh isi data *string* ke-2 dan 3 di *database*. Gambar 18 merupakan contoh tampilan *mobile android* untuk hapus data.



Gambar 17. Contoh *database string* data ke 2 dan 3



Gambar 18. Contoh tampilan *mobile android* untuk hapus data

## KESIMPULAN

Dari perancangan alat ataupun program sampai dengan pengujian didapatkan kesimpulan sebagai berikut:

1. Kamera yang digunakan menggunakan infrared sehingga keadaan lingkungan terutama intensitas pencahayaan tidak mempengaruhi hasil deteksi secara signifikan. Deteksi gerakan manusia (*human*) bekerja dengan akurat dengan akurasi 100% sedangkan deteksi benda (*non-human*) memiliki akurasi 90%.

2. Storage database firebase yang hanya berkisar 1 GB tidak terkendala penuhnya penyimpanan data dari *raspberry pi*. Dikarenakan data yang dikirimkan hanya data string dan gambar. Dan di pengujian proses menghapus data dari *mobile android* maupun di *database firebase* tidak didapatkan masalah.
3. Aplikasi berbasis *android* dapat memberi notifikasi jika terjadi pergerakan dan mengirim data tangkapan kamera *kemobile android* secara *real time* dan sesuai database. Smartphone dapat selalu *update* data dengan menghidupkan aplikasi *android* terlebih dahulu.

## Saran

Dalam pengembangan selanjutnya dapat dilakukan pengiriman data tidak hanya berupa data image tetapi juga data video *streaming* secara *realtime* dengan meningkatkan kapasitas perangkat pengirim. Hal tersebut dikarenakan kapasitas operating *raspberry pi* terbatas dalam pengolahan *transfer* data file yang besar ke *firebase*.

## DAFTAR PUSTAKA

- Ikhsan Samir, Zuraiyah A. (2016). "Penerapan Algoritma Background Substraction Untuk Tracking dan Klasifikasi Kendaraan". Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Pakuan
- Oriza Ahmad, (2014). Firebase Membantu Kita Membuat Aplikasi Realtime. Retrieved from <https://www.codepolitan.com>.
- Widipratama, M. Z. (2017). Sistem Pemantau Keamanan Menggunakan Kamera dengan Metode Background Subtraction. Surabaya: Tugas Akhir Institut Bisnis dan Informatika Stikom Surabaya
- Kusno Suryadi, Supriyanto Sikumbang. (2015). Human Detection Menggunakan Metode Histogram of Oriented Gradients. Malang: Universitas Gajayana Malang.