

SISTEM KEAMANAN RUMAH BERDASARKAN *MULTIFACE RECOGNITION* MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK (CNN)*

Abimas Setyawan Gultom¹⁾ Susijanto Tri Rasmana²⁾ Weny Indah Kusumawati³⁾

Program Studi/Jurusan Teknik Komputer

Universitas Dinamika

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1)abimas@gmail.com, 2)susyanto@dinamika.ac.id, 3)weny@dinamika.ac.id

Abstrak: Dewasa ini tingkat tindak kriminal semakin meningkat, walau sudah hampir sebagian besar kaum menengah keatas memasang CCTV di setiap sudut rumah, tetapi masih terdapat sebuah kekurangan yang cukup fatal yakni pemberitahuan secara langsung kepada pemilik rumah. Tindakan baru bisa dilakukan jika sudah dilakukan identifikasi terhadap rekaman CCTV yang juga memerlukan waktu yang lama. Berdasarkan hal tersebut, maka dalam Penelitian ini terdapat sebuah sistem keamanan rumah berdasarkan *multiface recognition* berbasis Android menggunakan metode CNN. Dalam penerapannya sistem ini mampu mengidentifikasi wajah dari anggota keluarga dan yang bukan termasuk. Bilamana objek yang terdeteksi adalah anggota keluarga, maka sistem mengirimkan Notifikasi ke *smartphone user* berupa nama, jam dan tanggal saat objek teridentifikasi. Objek dari selain anggota keluarga yang telah ditentukan oleh *user* bila teridentifikasi oleh kamera, maka sistem melakukan pengambilan foto terhadap objek dan mengirimkan Notifikasi beserta jam dan tanggal ke *smartphone user*, hal ini tentu meminimalisir dan pencegahan terhadap tindak kriminalitas yang dilakukan oleh oknum-oknum tertentu. Sistem yang dibuat pada penelitian ini menunjukkan bahwa metode *Convolutional Neural Network (CNN)* memiliki akurasi yang tinggi sebesar 100% dalam melakukan deteksi terhadap objek/wajah manusia berdasarkan 40 kali pengambilan data. Terdapat beberapa perbedaan pada tingkat akurasi, presisi dan *recall* pada metode *Convolutional Neural Network (CNN)* yakni, akurasi sebesar 87.5%, rata-rata presisi sebesar 90%, dan *recall* sebesar 100% berdasarkan 40 kali pengambilan data. Sistem pengiriman notifikasi dengan berbasis Android dan *firebase* juga bekerja cukup baik dengan persentase keberhasilan dalam mengunggah data berupa data gambar dan *string* sebesar 100% dan penerimaan notifikasi yang cukup baik dengan persentase keberhasilan 90% dengan berdasarkan pada 20 kali pengambilan data. Wajah terdeteksi dan dikenali oleh program saat berada dalam radius 3 meter dari kamera. Hal ini disebabkan karena sistem menggunakan kamera dengan resolusi 5 MP.

Kata kunci: *Convolutional Neural Network (CNN), Face Recognition, Android, Firebase.*

PENDAHULUAN

Dewasa ini tingkat tindak kriminal semakin meningkat, baik berupa pencurian dengan pemberatan (curat), pencurian kendaraan bermotor (curanmor), pencurian dengan kekerasan (curas), dan pencurian dengan senjata api, maupun perampokan terhadap rumah-rumah. Berdasarkan data dari Mabes Polri, berbagai kasus tindak kriminal semakin meningkat terutama pada kasus perampokan rumah yang tercatat dari bulan 15-30 April 2019 hanya terdapat 37 kasus meningkat menjadi 44 kasus pada 1-15 Mei 2019 (*News*,

2019). Sudah hampir sebagian besar kaum menengah keatas memasang CCTV di setiap sudut rumah, tetapi masih terdapat sebuah kekurangan yang cukup fatal yakni pemberitahuan secara langsung kepada pemilik rumah. Tindakan baru bisa dilakukan jika sudah dilakukan identifikasi terhadap rekaman CCTV yang juga memerlukan waktu yang lama.

Berdasarkan hal tersebut, maka dalam Penelitian ini terdapat sebuah sistem keamanan rumah berdasarkan *multiface recognition* berbasis Android menggunakan metode CNN. Dalam penerapannya sistem ini mampu mengidentifikasi

wajah dari anggota keluarga dan yang bukan termasuk. Bilamana objek yang terdeteksi adalah anggota keluarga, maka sistem mengirimkan Notifikasi ke *smartphone user* berupa nama, jam dan tanggal saat objek teridentifikasi. Objek dari selain anggota keluarga yang telah ditentukan oleh *user* bila teridentifikasi oleh kamera, maka sistem melakukan pengambilan foto terhadap objek dan mengirimkan Notifikasi beserta jam dan tanggal ke *smartphone user*, hal ini tentu meminimalisir dan pencegahan terhadap tindak kriminalitas yang dilakukan oleh oknum-oknum tertentu.

Sampai saat ini sudah banyak penelitian yang mengarah pada permasalahan ini. Salah satunya adalah sebagai berikut, implementasi metode *Single Shot Detector* (SSD) untuk pengenalan wajah oleh Adrianto Sukusvieri dari Universitas Dinamika (Sukusvieri, 2019). Adapun penelitian lainnya sistem keamanan rumah menggunakan kamera CCTV dengan pengiriman Notifikasi ke Android menggunakan *thresholding* oleh Rahmadi Surya dari Universitas Dinamika (Surya, 2019).

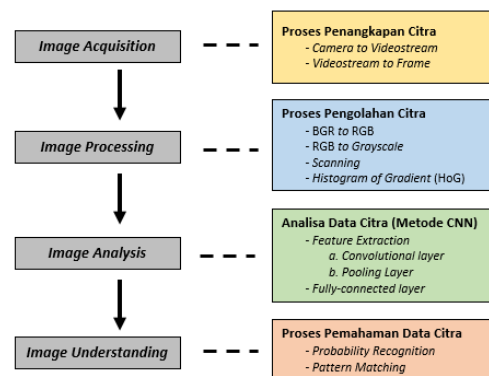
Berdasarkan permasalahan dan acuan diatas, maka penelitian selanjutnya dibuat sebuah sistem keamanan rumah berdasarkan *multiface recognition* berbasis Android menggunakan metode *Convolutional Neural Network* (CNN). Metode CNN dipilih karena berdasarkan beberapa literatur terdapat hipotesa bahwa untuk tingkat akurasi *Convolutional Neural Network* (CNN) lebih tinggi dibandingkan dengan SSD dan lebih rendah dibanding YOLO, serta untuk tingkat kompleksitas komputasi atau beban terhadap sistem CNN lebih ringan daripada YOLO dan sedikit lebih kompleks dari SSD. Dikarenakan penelitian ini bertujuan untuk mengutamakan kecepatan dan keakuratan dalam melakukan pengolahan citra / indentifikasi serta pengiriman data, maka metode CNN lebih sesuai untuk diimplementasikan didalam penelitian ini.

METODE PENELITIAN

Metode penelitian dalam melakukan penelitian ini terdapat beberapa tahap. Dimana tahapan ini juga merupakan sebuah alur sistem yang sinergi dalam metode penelitian yang dilakukan. Tahap pertama adalah *security camera* yang merupakan tahap awal sebagai media input data citra, baik berupa *image* maupun *videostream realtime*. Hasil output ini nantinya masuk ke tahap *artificial intelegent* yang merupakan tahap pengolahan citra dengan menggunakan metode CNN untuk mengidentifikasi dari objek yang masukkan oleh kamera. Setelah input

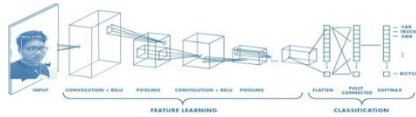
teridentifikasi, maka sistem memutuskan hasilnya dan mengirim/*Upload* data kedalam *database server*. Setelah data terkirim dan berhasil diterima oleh *database server*, maka secara otomatis sistem melakukan *forwarding* ke *Android application* dengan mengirimkan data *string* dan data *image* kepada pengguna.

Secara spesifik terkait layer yang digunakan dalam metode CNN adalah *feature extraction layer* dengan menjalankan kedua fungsi khusus dari *Convolutional layer* dan juga *pooling layer*. *Feature extraction layer* merupakan sebuah proses *encoding* dari sebuah citra gambar menjadi sebuah *features* atau sebuah angka-angka yang mewakili atau mempresentasikan gambar tersebut. Dimana untuk mempresentasikan gambar tersebut *feature extraction layer* melakukan dua tahap algoritma, yakni *convolutional layer* dan juga *pooling layer*.



Gambar 1. Bagan metodologi penelitian

Convolutional layer merupakan sebuah neuron yang tersusun sangat kompleks dan membentuk sebuah filter, dimana filter tersebut membentuk sebuah *pixel* pada citra untuk melakukan komputasi dan menentukan nilai *pixel* dari tiap gambar. Selanjutnya hasil dari proses *convolutional layer* dilanjutkan oleh *pooling layer* yang juga terdiri dari sebuah filter yang bergerak atau bergeser keseluruhan area yang terdapat pada *feature map* dengan memiliki ukuran serta stride tertentu. Dalam algoritma ini *pooling layer* menggunakan *average pooling* mengambil nilai rata-rata dari data representasi *image*. Hasil representasi data tersebut yang nantinya digunakan dalam melakukan identifikasi / kecocokan terhadap nilai citra yang terdapat didalam *dataset* yang telah di *training* dengan citra masukan yang diambil secara *realtime*.

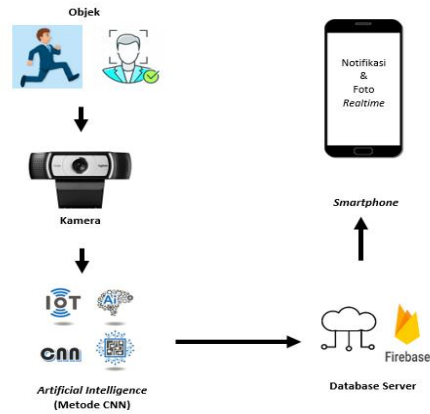


Gambar 2. Convolutional neural network

Arsitektur yang digunakan dalam metode CNN dapat diamati pada gambar 2. Dalam gambar tersebut menjelaskan terkait proses looping algoritma *convolutional layer* dan juga *pooling layer* dalam melakukan komputasi untuk mendapatkan sebuah nilai piksel dari tiap citra dan menyimpan setiap nilai piksel hasil representasi citra tersebut kedalam sebuah array. Array tersebut berfungsi untuk menyimpan dan melakukan klasifikasi terhadap tiap data citra yang telah di *training*. Selanjutnya hasil nilai yang telah diolah oleh *pooling layer* dengan menggunakan *average pooling* dicocokkan dengan data masukan citra yang didapatkan secara *realtime* untuk menentukan apakah masukan yang ada termasuk citra yang dikenal sebagai anggota keluarga atau citra yang tak dikenal.

Perancangan Perangkat Keras

Perancangan program pada penelitian bisa diamati pada gambar 3. Sistem mulai melakukan identifikasi ketika ditemukan sebuah objek berupa human/manusia. Objek ditangkap oleh kamera dan diproses dengan artificial intelligent/kecerdasan buatan dengan menggunakan bahasa pemrograman Python yang terintegrasi dengan OpenCV menggunakan metode *Convolutional neural network* (CNN). Hasil dari proses program menghasilkan data output berupa data *string* dan data *image*, dimana nantinya data ini dikirim/di-Upload kedalam *database server* ke *firebase*. Ketika data belum berhasil di-Upload, maka program melakukan re-Upload hingga berhasil di unggah kedalam *database server*. Setelah berhasil di-Upload, maka sistem melakukan transmit/pengiriman data dari *firebase* ke aplikasi Android secara *realtime*. Aplikasi Android menerima data berupa *string* dan citra gambar secara *realtime* serta memberikan sebuah Notifikasi peringatan kepada *user*.

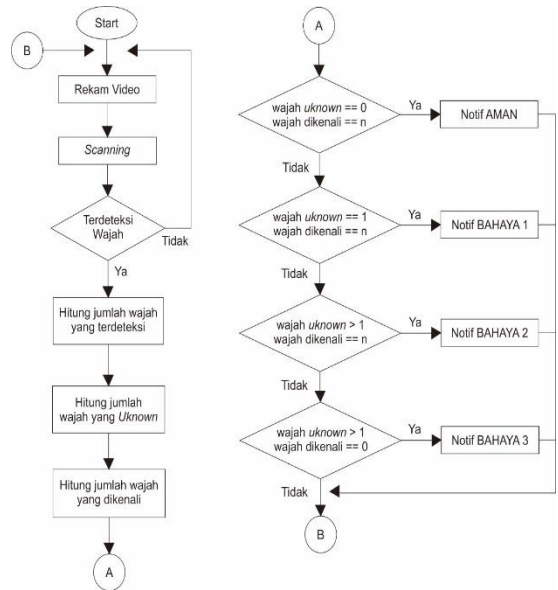


Gambar 3. Blok Diagram sistem keamanan rumah

Perancangan Sistem Program

Flowchart Sistem Keamanan Rumah

Proses dari pengolahan data atau flowchart program dengan menggunakan artificial intelligent dapat diamati pada gambar 4 yang mana pada gambar tersebut terdapat tahapan dari jalannya sistem program.



Gambar 4. Flowchart sistem keamanan rumah

Tahapan pertama, sistem melakukan *image acquisition*/proses penangkapan citra yang dilakukan oleh kamera. Kamera menjalankan status videostream ketika program dijalankan. Pada proses ini kamera yang melakukan videostream menghasilkan sebuah *realtime* video, dimana hasil dari citra video ini diproses untuk diambil citra gambar berupa sebuah frame. Setelah mendapatkan sebuah citra frame, maka program melakukan

scanning dengan menggunakan *similarity transform* untuk menemukan sebuah citra wajah dari *image*.

Tabel 1. Pembagian jenis Notifikasi sistem keamanan rumah

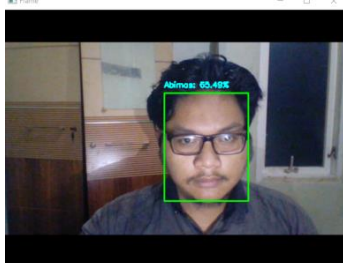
No	Wajah dikenali	Wajah tidak dikenali	Jenis Notifikasi
1	n	0	Aman
2	n	1	BAHAYA 1
3	n	>1	BAHAYA 2
4	0	n	BAHAYA 3

Keterangan: $n = 1$ atau lebih

Ketika sebuah wajah terdeteksi, maka sistem melakukan perhitungan terhadap jumlah wajah yang terdeteksi. Setelah itu sistem melakukan sebuah proses identifikasi wajah (*multiface recognition*) untuk menentukan apakah wajah objek yang terdeteksi dikenali atau tidak. Dalam tahapan selanjutnya, kondisi tersebut mempengaruhi sistem dalam melakukan pengambilan keputusan dalam memberikan sebuah Notifikasi atau pemberitahuan kepada *user*. Dimana jika semua objek wajah dikenali, maka tidak ada Notifikasi atau aman, bila terdapat satu wajah yang tidak dikenali dan sebagian besar dikenali, maka tampil Notifikasi bahaya 1, bila terdapat satu wajah yang dikenali dan sebagian besar tidak dikenali, maka tampil Notifikasi bahaya 2, dan bila semua objek wajah yang terdeteksi tidak dikenali, maka tampil Notifikasi bahaya 3.

Pengenalan / Recognition Wajah

Ketika program mulai dijalankan, maka program melakukan sebuah proses untuk identifikasi terhadap objek yang tertangkap oleh kamera. Didalam prosesnya, sistem melakukan sebuah deteksi wajah terlebih dahulu. Pada saat wajah telah terdeteksi, maka sistem melanjutkan kedalam tahap *recognition* untuk menentukan objek atau pengenalan terhadap objek yang telah berhasil tertangkap oleh kamera menggunakan metode CNN. Pengenalan wajah tersebut dapat diamati pada gambar 5.



Gambar 5. Pengenalan wajah

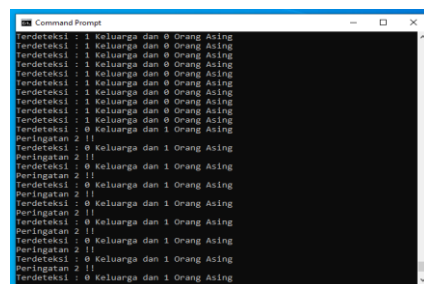
Proses pengenalan atau *recognition* wajah dengan menggunakan metode CNN telah dijelaskan pada sub bab yang sebelumnya. Secara ringkas dapat dijelaskan bahwa sistem melakukan proses *recognition* berdasarkan dari daaset yang telah dibuat seblumnya untuk menentukan objek mana saja yng termasuk anggota keluarga dan mana objek yang tidak dikenal. Untuk *syntax* program dalam proses pengenalan wajah dapat diamati pada cuplikan teks berikut:

```
faceBlob = cv2.dnn.blobFromImage(face, 1.0 / 255,
                                (96, 96), (0, 0, 0),
                                swapRB=True, crop=False)
embedder.setInput(faceBlob)
vec = embedder.forward()

preds = recognizer.predict_proba(vec)[0]
j = np.argmax(preds)
proba = preds[j]
name = le.classes_[j]
```

Penghitungan Objek

Penghitungan objek merupakan tahapan yang selanjutnya dalam proses untuk menentukan tingkat keamanan yang dilakukan oleh sistem dalam melakukan proses pengiriman *Notifikasi* terhadap *user* terkait tingkat bahaya yang telah terdeteksi dan diolah oleh sistem. Dalam tahap ini hasil identifikasi yang telah dilakukan oleh algoritma CNN diolah kembali menjadi sbuah step untuk menentukan tingkat keamanan berdasarkan dari jumlah objek yang terdeteksi dengan sebuah klasifikasi khusus yang sebelumnya telah ditentukan oleh *user*.



Gambar 6. Hasil klasifikasi oleh sistem

Dimana dalam pengklasifikasian ini dilakukan dengan pembagian beberapa tingkat keamanan/bahaya berdasarkan jumlah dan jenis objek yang telah berhasil teridentifikasi sebagai berikut:

1. Peringatan 1: Terdeteksi satu manusia tidak dikenal.
2. Peringatan 2: Terdeteksi banyak manusia tidak dikenal dan beberapa dikenal.

3. Peringatan 3: Terdeteksi banyak manusia tidak dikenal.

Klasifikasi ini dilakukan untuk lebih menspesifikasikan terkait objek yang telah berhasil diidentifikasi, sehingga mempermudah *user* dalam mengambil keputusan terkait tindakan yang dilakukan dalam menangani bahaya berdasarkan hasil yang berhasil diidentifikasi oleh sistem. Pada gambar 6 dapat diamati hasil klasifikasi yang didapatkan oleh sistem.

Dalam proses klasifikasi dari penghitung objek ini dilakukan dengan cara melakukan perulangan terhadap loop yang terdapat pada tahap *recognition* kemudian membuat sebuah variabel khusus untuk membuat sebuah logika kondisi dengan berdasarkan pembagian klasifikasi sesuai dengan yang telah dijelaskan sebelumnya. *Syntax* dalam klasifikasi tingkat keamanan berdasarkan jumlah dan jenis objek dapat diamati pada teks berikut:

```
if sending:
print("Terdeteksi : {} Keluarga dan {}
Orang Asing".format(known, unknown))

if known > 0 and unknown == 1:
ms = known + unknown
print("dataint:" + str(dataint))
my_stream =db.child("datagambar").child
(str(dataint)).stream(stream_handler,None)
dataint = dataint + 1
sending = False

elif known > 0 and unknown > 1:
ms = known + unknown

print("dataint:" + str(dataint))
my_stream
=db.child("datagambar").child(str
(dataint)).stream(stream_handler2,None)
dataint = dataint + 1
sending = False

elif known == 0 and unknown > 0:
ms = known + unknown
print("dataint:" + str(dataint))
my_stream =
db.child("datagambar").child(str
(dataint)).stream(stream_handler3,None)
dataint = dataint + 1
sending = False
```

Pengiriman Notifikasi Android

Pada pelaksanaannya sistem ini juga melakukan sebuah pengiriman Notifikasi kedalam perangkat Android *user*. Pada dasarnya tahap ini juga merupakan sebuah tahap pengolahan sistem yang cukup kompleks, sehingga untuk selanjutnya penulis jelaskan secara garis besar atau proses secara global dalam pengiriman Notifikasi Android ini. Dalam tahapannya terdapat dua proses utama dalam pengiriman notifikasi Android, yakni proses

pengiriman Notifikasi oleh sistem dan proses penerimaan Notifikasi oleh Android.

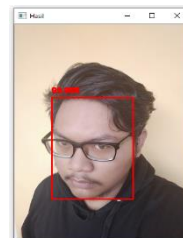
Proses Penerimaan Notifikasi Oleh Android

Pada proses selanjutnya setelah data berhasil dikirim oleh sistem kedalam *firebase*, maka secara otomatis aplikasi yang telah dibuat dan diinstall kedalam *smartphone* android segera melakukan pengunduhan / *downloading* data dan memunculkan notifikasi kepada *user*. Dalam tahapannya ketika sistem telah berhasil melakukan pengiriman data kedalam *firebase*, maka sistem juga melakukan pengiriman *ack* atau sebuah *trigger* kepada aplikasi android, selain itu pada aplikasi android juga sudah terdapat kode akses maupun API dari *firebase*, sehingga aplikasi android juga melakukan *scanning* atau pengecekan data yang terdapat pada *firebase*. Apabila terdapat perubahan data pada *firebase* maupun pengurangan data, maka secara *realtime* aplikasi android juga melakukan sinkronisasi. Hal ini tentu saja sangat berpengaruh terhadap sistem keamanan karena bersifat *realtime*, sehingga *user* dapat segera mengetahui kondisi pada tempat yang dipantau secara *realtime*.

HASIL DAN PEMBAHASAN

Hasil Uji Deteksi Terhadap Objek

Pengujian deteksi terhadap objek ini dilakukan untuk melihat tingkat akurasi deteksi orang yang dilakukan oleh sistem. Untuk hasil dari pengujian akurasi deteksi terhadap objek dapat diamati pada gambar 7 dibawah ini.



Gambar 7. Hasil deteksi objek

Adapun proses pengujian deteksi terhadap objek ini dapat dilakukan dengan menggunakan tabel berikut:

Tabel 2. Uji tingkat akurasi deteksi objek

Uji ke-	Objek	Hasil Uji	
		Berhasil	Gagal
1	Anggota 1	V	-
	Anggota 2	V	-
	Anggota 3	V	-
	Tak dikenal	V	-
2	Anggota 1	V	-
	Anggota 2	V	-

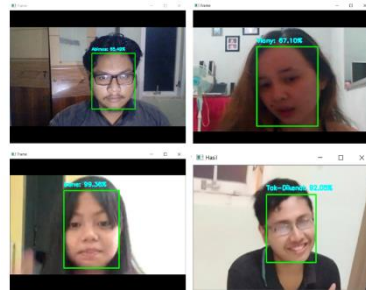
Uji ke-	Objek	Hasil Uji	
		Berhasil	Gagal
3	Anggota 3	V	-
	Tak dikenal	V	-
	Anggota 1	V	-
	Anggota 2	V	-
	Anggota 3	V	-
	Tak dikenal	V	-
4	Anggota 1	V	-
	Anggota 2	V	-
	Anggota 3	V	-
	Tak dikenal	V	-
5	Anggota 1	V	-
	Anggota 2	V	-
	Anggota 3	V	-
	Tak dikenal	V	-
	Anggota 1	V	-
6	Anggota 2	V	-
	Anggota 3	V	-
	Tak dikenal	V	-
	Anggota 1	V	-
7	Anggota 2	V	-
	Anggota 3	V	-
	Tak dikenal	V	-
	Anggota 1	V	-
8	Anggota 2	V	-
	Anggota 3	V	-
	Tak dikenal	V	-
	Anggota 1	V	-
9	Anggota 2	V	-
	Anggota 3	V	-
	Tak dikenal	V	-
	Anggota 1	V	-
10	Anggota 2	V	-
	Anggota 3	V	-
	Tak dikenal	V	-

Dari hasil tabel 2 diatas, dapat dijelaskan bahwa dari 40 data sampel uji coba terhadap 4 responden dapat melakukan deteksi terhadap objek dengan baik dan akurat dengan persentase akurasi 100%.

Hasil Pengujian Akurasi, Presisi, dan Recall Recognition CNN Terhadap Objek

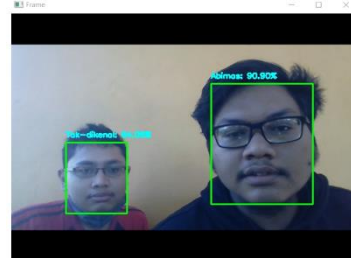
Pengujian akurasi *recognition* CNN terhadap objek ini dilakukan untuk melihat tingkat kesesuaian antara objek manusia yang dideteksi dengan hasil dari proses identifikasi yang dilakukan oleh CNN. Untuk hasil dari pengujian akurasi

recognition CNN terhadap objek dapat diamati pada gambar 8 berikut ini.



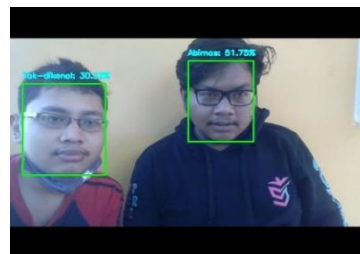
Gambar 8. Hasil akurasi *recognition* CNN

Pengujian presisi *recognition* CNN terhadap objek ini dilakukan untuk melihat tingkat ketepatan dari informasi yang di request oleh *user* dengan hasil identifikasi yang diberikan oleh CNN. Untuk hasil dari pengujian presisi *recognition* CNN terhadap objek dapat diamati pada gambar 9 dibawah ini.



Gambar 9. Hasil presisi *recognition* CNN

Pengujian deteksi terhadap objek ini dilakukan untuk melihat tingkat akurasi deteksi orang yang dilakukan oleh sistem. Untuk hasil dari pengujian *recall recognition* CNN terhadap objek dapat diamati pada gambar 10 dibawah ini.



Gambar 10. Hasil *recall recognition*

Adapun proses pengujian akurasi, presisi, dan *recall* menggunakan *recognition* CNN dapat dilakukan dengan menggunakan tabel berikut:

Tabel 3. Hasil pengujian akurasi, presisi, dan Recall Recognition CNN

Uji ke-	Objek	Hasil Uji			Ket.
		Akurasi	Presisi	Recall	
1	Anggota 1	V	V	V	-
	Anggota 2	-	V	V	-
	Anggota 3	V	V	V	-
2	Tak dikenal	V	V	V	-
	Anggota 1	V	V	V	-
	Anggota 2	-	V	V	-
3	Anggota 3	-	V	V	-
	Tak dikenal	V	V	V	-
	Anggota 1	V	V	V	-
4	Anggota 2	-	V	V	-
	Anggota 3	V	-	V	-
	Tak dikenal	V	V	V	-
5	Anggota 1	V	V	V	-
	Anggota 2	V	V	V	-
	Anggota 3	V	V	V	-
6	Tak dikenal	V	V	V	-
	Anggota 1	V	V	V	-
	Anggota 2	V	-	V	-
7	Anggota 3	V	V	V	-
	Tak dikenal	V	V	V	-
	Anggota 1	V	V	V	-
8	Anggota 2	-	V	V	-
	Anggota 3	V	V	V	-
	Tak dikenal	V	V	V	-
9	Anggota 1	V	-	V	-
	Anggota 2	V	V	V	-
	Anggota 3	V	V	V	-
10	Tak dikenal	V	V	V	-
	Anggota 1	V	V	V	-
	Anggota 2	V	V	V	-
	Anggota 3	V	V	V	-
	Tak dikenal	V	V	V	-

Hasil Uji Upload & Notifikasi Data ke Database

Pengujian pengiriman/Upload data ke database dilakukan untuk mengetahui tingkat keberhasilan dari program untuk mengirimkan sebuah data dari program kedalam database pada firebase. Adapun proses pengujian Upload dan notifikasi data ke database ini dapat dilakukan dengan menggunakan tabel 4 berikut:

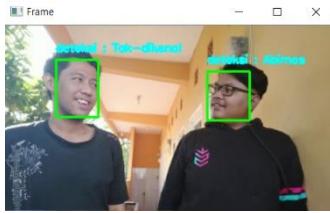
Tabel 4. Hasil uji upload dan notifikasi data ke database

Uji Ke-	Jenis Data	Hasil Uji		Ket.
		Upload	Notifikasi	
1	String	V	V	-
	Image	V	V	-
2	String	V	V	-
	Image	V	V	-
3	String	V	V	-
	Image	V	V	-
4	String	V	V	-
	Image	V	V	-
5	String	V	V	-
	Image	V	V	-
6	String	V	-	-
	Image	V	-	-
7	String	V	V	-
	Image	V	V	-
8	String	V	V	-
	Image	V	V	-
9	String	V	V	-
	Image	V	V	-
10	String	V	V	-
	Image	V	V	-

Pengujian penerimaan notifikasi Android dilakukan untuk mengetahui tingkat keberhasilan sistem pada saat melakukan pengiriman data dari firebase menuju ke aplikasi Android yang telah dibuat. Proses pengujian ini tidak dapat dilakukan bila tingkat keberhasilan pada proses sebelumnya (uji Upload) belum terselesaikan.

Hasil Pengujian Klarifikasi Peringatan Pada Notifikasi

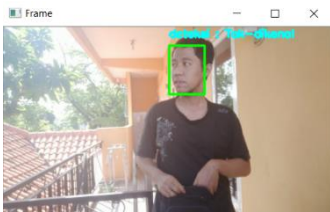
Pengujian ini bertujuan untuk melakukan uji peringatan pada notifikasi android. Dimana pada sistem notifikasi terdapat 3 jenis peringatan yang muncul didalam bar notifikasi android yakni, bahaya 1 jika terdapat satu tak dikenali dan satu atau lebih anggota yang dikenali, dan bahaya 2 jika terdapat tak dikenali lebih dari satu dan anggota yang dikenali hanya satu atau lebih, serta bahaya 3 jika terdapat tak dikenal satu atau lebih tetapi tidak ada anggota yang dikenal.



Gambar 11. Kondisi bahaya 1

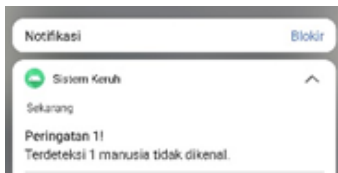


Gambar 12. Kondisi bahaya 2

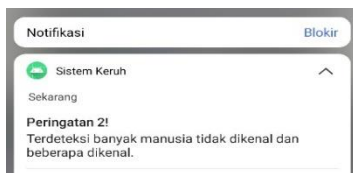


Gambar 13. Kondisi bahaya 3

Adapun tampilan dari sistem keamanan rumah saat melakukan pengenalan wajah dapat diamati pada gambar 11 yang merupakan sebuah kondisi dari bahaya 1 dimana terdapat satu anggota dikenal dan satu orang tak dikenal. Dapat diamati pula pada gambar 12 yang merupakan sebuah kondisi dari bahaya 2 dimana terdapat satu anggota dikenal dan dua orang tak dikenal, serta pada gambar 13 yang merupakan kondisi bahaya 3 dimana terdapat satu orang tak dikenal dan tidak terdapat anggota yang dikenal.



Gambar 14. Peringatan notifikasi bahaya 1



Gambar 15. Peringatan notifikasi bahaya 2



Gambar 16. Peringatan notifikasi bahaya 3

Ketika sebuah kondisi masuk kedalam range pengklarifikasian tingkat bahaya seperti halnya yang sudah dijelaskan sebelumnya, maka secara otomatis sistem mengirimkan sebuah peringatan berupa notifikasi sesuai dengan tingkat bahayanya. Dapat diamati pada gambar 14 yang merupakan peringatan untuk bahaya 1. Pada gambar 15 sebuah peringatan untuk bahaya 2, dan pada gambar 16 untuk peringatan bahaya 3. Dengan adanya pengklarifikasian tersebut diharapkan *user* nantinya dapat mengambil sebuah tindakan dalam melakukan pengamanan terhadap rumah sesuai dengan tingkat bahaya yang terdeteksi. Adapun tabel pengujian terhadap pengklarifikasian tingkat bahaya tersebut dapat diamati pada tabel 5 berikut.

Tabel 5. Hasil pengujian klarifikasi peringatan pada notifikasi

Uji Ke	Kondisi Aktual	Kondisi Hasil Deteksi		Peringatan Yang Tampil			Keberhasilan	
		Dikenal	Tak Dikenal	Bahaya			Ya	Tidak
				1	2	3		
1	Bahaya 1	1	1	V	-	-	V	-
2	Bahaya 1	1	1	V	-	-	V	-
3	Bahaya 1	1	1	V	-	-	V	-
4	Bahaya 1	1	1	V	-	-	V	-
5	Bahaya 1	2	0	-	-	-	-	V
6	Bahaya 1	1	1	V	-	-	V	-
7	Bahaya 1	1	1	V	-	-	V	-
8	Bahaya 1	1	1	V	-	-	V	-
9	Bahaya 1	1	1	V	-	-	V	-
10	Bahaya 1	1	1	V	-	-	V	-
11	Bahaya 2	1	2	-	V	-	V	-
12	Bahaya 2	1	2	-	V	-	V	-
13	Bahaya 2	2	1	V	-	-	-	V

14	Bahaya 2	1	2	-	V	-	V	-
15	Bahaya 2	2	1	V	-	-	-	V
16	Bahaya 2	1	2	-	V	-	V	-
17	Bahaya 2	1	2	-	V	-	V	-
18	Bahaya 2	1	2	-	V	-	V	-
19	Bahaya 2	1	2	-	V	-	V	-
20	Bahaya 2	1	2	-	V	-	V	-
21	Bahaya 3	0	1	-	-	V	V	-
22	Bahaya 3	0	1	-	-	V	V	-
23	Bahaya 3	0	1	-	-	V	V	-
24	Bahaya 3	0	1	-	-	V	V	-
25	Bahaya 3	0	1	-	-	V	V	-
26	Bahaya 3	0	1	-	-	V	V	-
27	Bahaya 3	0	1	-	-	V	V	-
28	Bahaya 3	0	1	-	-	V	V	-
29	Bahaya 3	0	1	-	-	V	V	-
30	Bahaya 3	0	1	-	-	V	V	-

Berdasarkan tabel 5 dapat diketahui bahwa pemberian peringatan bahaya berupa notifikasi ke android yang dilakukan oleh sistem selalu sesuai dengan hasil deteksi dari sistem. Dengan demikian dapat disimpulkan bahwa tingkat akurasi maupun kesesuaian dari klarifikasi peringatan selalu linier dengan hasil deteksi atau pengenalan yang dilakukan oleh sistem.

KESIMPULAN

Berdasarkan dari beberapa tahap pengujian terhadap sistem yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Metode *Convolutional Neural Network* (CNN) memiliki akurasi yang tinggi sebesar 100% dalam melakukan deteksi terhadap objek/wajah manusia berdasarkan 40 kali pengambilan data.
2. Terdapat beberapa perbedaan pada tingkat akurasi, presisi dan *recall* pada metode *Convolutional Neural Network* (CNN) yakni, akurasi sebesar 87.5%, rata-rata presisi sebesar 90%, dan *recall* sebesar 100% berdasarkan 40 kali pengambilan data.
3. Sistem pengiriman notifikasi dengan berbasis Android dan *firebase* juga bekerja cukup baik dengan persentase keberhasilan dalam mengunggah data berupa data gambar dan *string* sebesar 100% dan penerimaan notifikasi yang cukup baik dengan persentase keberhasilan 90% dengan berdasarkan pada 20 kali pengambilan data.

4. Wajah terdeteksi dan dikenali oleh program saat berada dalam radius 3 meter dari kamera. Hal ini disebabkan karena sistem menggunakan kamera dengan resolusi 5 MP.

Saran

Pada pengembangan yang selanjutnya sistem ini juga dapat diintegrasikan dengan IoT pada rumah, bilamana terdeteksi anggota keluarga yang masuk ke rumah dan telah diklarifikasi oleh *user*, maka pagar rumah dapat terbuka secara otomatis serta listrik dapat menyala jika tidak terdapat orang sama sekali dirumah.

DAFTAR PUSTAKA

- Belharbi, S. (2015). Input/Output Deep Architecture for Structured. *Arxis*, 17.
- Febriana, R. (2015). Analysis And Implementation Of Lip Movement Translator Application To Text Using. *e-Proceeding of Engineering*, 6.
- J., D. (2009). A Large-Scale Hierarchical *Image Database*. *ImageNet*, 8.
- Latasha, K. (2018, April 23). *Perbedaan: precision, recall & accuracy*. Retrieved from Data's Base: <https://dataq.wordpress.com/2013/06/16/perbedaan-precision-recall-accuracy/>
- News, C. (2019, Mei 8). *Peningkatan Kriminalitas*. Retrieved from CNN Indonesia: www.cnn/news/peningkatan-kriminalitas-09.co.id
- Pangestu1, M. A. (2018). Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre Trained CNN Model. *Jurnal Teknik Informatika dan Sistem Informasi*, 8.
- Raut, N. (2018). Facial Emotion *Recognition* Using Machine Learning . *Masters Project*.
- Zisserman, K. S. (2015). Very Deep Convolutional Networks for Large-Scale *Image Recognition*. *ICLR*, 23.