



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

TUGAS AKHIR

**Program Studi
S1 Teknik Komputer**

Oleh:

Abimas Setyawan Gultom

15410200059

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2020

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

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



Disusun Oleh:

Nama : Abimas Setyawan Gultom

NIM : 15410200059

Program Studi : S1 Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2020

“Setiap orang membuat alasan dan menyalahkan keadaan karna mereka tidak ingin berusaha dan berjuang.”

Abimas Setyawan Gultom



UNIVERSITAS
Dinamika

Dipersembahkan kepada Bapak, Ibu dan Keluarga saya yang selalu memotivasi dan memberikan doa yang terbaik kepada saya, angkatan 15 Teknik Komputer yang selalu solid, serta semua orang yang selalu membantu, mendukung dan memotivasi agar terus belajar dan berkarya lebih baik dari hari kemarin.



UNIVERSITAS
Dinamika

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

Dipersiapkan dan disusun oleh

ABIMAS SETYAWAN GULTOM

NIM: 15410200059

Telah diperiksa, diuji, dan disetujui oleh Dewan Penguji

Pada: 24 Agustus 2020

Susunan Dewan Penguji

Pembimbing

I. Dr. Susijanto Tri Rasmana, S.Kom., M.T.

NIDN. 0727097302

I. Weny Indah Kusumawati, S.Kom., M.MT.

NIDN. 0721047201

Pembahas

II. Hariato, S.Kom., M.Eng.

NIDN. 0722087701


Digitally signed by
Universitas Dinamika
Date: 2020.09.01
13:15:13 +07'00'


Digitally signed by
Universitas Dinamika
Date: 2020.09.01
12:55:46 +07'00'


Digitally signed by
Universitas Dinamika
Date: 2020.09.01 08:50:04
+0700'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana


Digitally signed by
Universitas
Dinamika
Date: 2020.09.06
07:53:25 +07'00'

NIDN : 0708017101

Dekan Fakultas Teknologi dan Infomatika
Universitas Dinamika

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : Abimas Setyawan Gultom
NIM : 15410200059
Program Studi : SI Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **Sistem Keamanan Rumah Berdasarkan Multiface Recognition Menggunakan Metode Convolutional Neural Network (CNN)**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (Non-Exclusive Royalti Free Right) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (database) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 24 Agustus 2020

Yang menyatakan



Abimas Setyawan Gultom
NIM: 15.41020.0059

ABSTRAK

Dewasa ini tingkat tindak kriminal semakin meningkat, walau sudah hampir sebagian besar kaum menengah keatas memasang CCTV di setiap sudut rumah, akan 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 Tugas Akhir ini terdapat sebuah sistem keamanan rumah berdasarkan *multiface recognition* berbasis *Android* menggunakan metode CNN. Dalam penerapannya sistem ini akan mampu mengidentifikasi wajah dari anggota keluarga dan yang bukan termasuk. Bilamana objek yang terdeteksi adalah anggota keluarga maka sistem akan 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 akan melakukan pengambilan foto terhadap objek dan akan mengirimkan notifikasi beserta jam dan tanggal ke *smartphone user*, hal ini tentu akan meminimalisir dan pencegahan terhadap terjadinya tindak kriminalitas. 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%, presisi sebesar 92.5%, dan *recall* sebesar 100% berdasarkan 40 kali pengambilan data. Sistem pengiriman *notifikasi* dengan berbasis *Android* dan *firebase* juga bekerja cukup baik dengan presentase keberhasilan dalam mengunggah data berupa data gambar dan *string* sebesar 100% dan penerimaan *notifikasi* yang cukup baik dengan presentase keberhasilan 90% dengan berdasarkan pada 20 kali pengambilan data.

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

KATA PENGANTAR

Atas berkat rahmat Tuhan Yang Maha Esa, maka penulis berhasil menyelesaikan dan menyusun naskah Tugas Akhir yang berjudul Sistem Keamanan Rumah Berdasarkan *Multiface Recognition* Berbasis *Android* Menggunakan Metode *Convolutional Neural Network*(CNN). Tugas Akhir ini merupakan salah satu syarat yang harus ditempuh oleh setiap mahasiswa untuk meraih gelar kesarjanaan di Jurusan Teknik Komputer, Fakultas Teknologi dan Informatika, Universitas Dinamika.

Dalam mengerjakan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

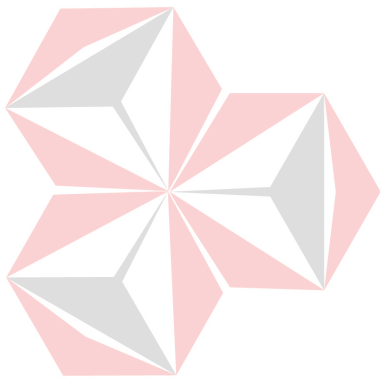
1. Bapak Prof. Dr. Budi Jatmiko, M.Pd., selaku Rektor Universitas Dinamika.
2. Bapak Dr. Jusak, selaku Dekan Fakultas Teknologi dan Informatika Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Prodi S1 Teknik Komputer Universitas Dinamika.
4. Bapak Dr. Susijanto Tri Rasmana, S.Kom., M.T., selaku dosen pembimbing pertama yang telah memberikan arahan dan bimbingan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
5. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku dosen pembimbing kedua yang telah memberikan arahan dan bimbingan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
6. Bapak Harianto, S.Kom., M.Eng., selaku dosen pembahas yang telah memberikan arahan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
7. Seluruh dosen pengajar Program Studi S1 Teknik Komputer Universitas Dinamika yang telah mendidik dan memberikan motivasi kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan baik.

8. Rekan-rekan mahasiswa S1 Teknik Komputer dan semua pihak yang telah memberikan dorongan dan bantuan baik secara langsung maupun secara tidak langsung.

Akhir kata, penulis berharap semoga segala sesuatu yang telah dihasilkan dalam pelaksanaan Tugas Akhir ini dapat bermanfaat.

Surabaya, 24 Agustus 2020

Abimas Setyawan Gultom



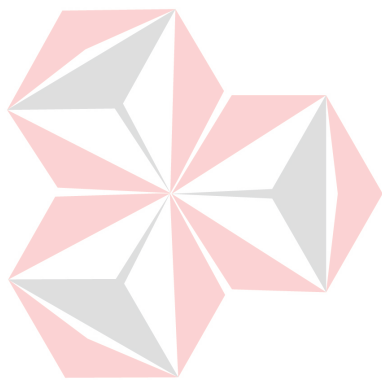
UNIVERSITAS
Dinamika

DAFTAR ISI

	Halaman
ABSTRAK	v
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	3
1.5 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 <i>Convolutional Neural Network</i>	5
2.2 Akurasi, Presisi dan <i>Recall</i> dalam <i>Machine Learning</i>	7
2.3 <i>Image Processing</i>	9
2.4 <i>Firebase</i>	10
2.5 <i>Python</i>	10
2.6 <i>OpenCV</i>	10
2.7 <i>Android Studio</i>	11
2.8 <i>Open Computer Vision (OpenCV)</i>	11
2.9 <i>Computer Vision</i>	11
2.10 Kamera.....	12
BAB III METODOLOGI PENELITIAN	13
3.1 Metode Penelitian.....	13
3.2 Perancangan Perangkat Keras	15
3.3 Perancangan Program	16
3.4 Pengenalan / <i>Recognition</i> Wajah	17
3.5 Penghitungan Objek.....	18
3.6 Pengiriman <i>Notifikasi Android</i>	20

3.6.1	Proses Pengiriman <i>Notifikasi</i> Oleh Sistem.....	21
3.6.2	Proses Penerimaan <i>Notifikasi</i> Oleh <i>Android</i>	22
BAB IV	HASIL DAN PEMBAHASAN	24
4.1	Pengujian Deteksi Terhadap Objek.....	24
4.1.1	Tujuan Pengujian Deteksi Terhadap Objek	24
4.1.2	Proses Pengujian Deteksi Terhadap Objek	24
4.1.3	Hasil Pengujian Deteksi Terhadap Objek	25
4.2	Pengujian Akurasi <i>Recognition</i> CNN Terhadap Objek	28
4.2.1	Tujuan Pengujian Akurasi <i>Recognition</i> CNN Terhadap Objek	28
4.2.2	Proses Pengujian Akurasi <i>Recognition</i> CNN Terhadap Objek	28
4.2.3	Hasil Pengujian Akurasi <i>Recognition</i> CNN Terhadap Objek	28
4.3	Pengujian Presisi <i>Recognition</i> CNN Terhadap Objek	31
4.3.1	Tujuan Pengujian Presisi <i>Recognition</i> CNN Terhadap Objek	32
4.3.2	Proses Pengujian Presisi <i>Recognition</i> CNN Terhadap Objek	32
4.3.3	Hasil Pengujian Presisi <i>Recognition</i> CNN Terhadap Objek	32
4.4	Pengujian <i>Recall Recognition</i> CNN Terhadap Objek	35
4.4.1	Tujuan Pengujian <i>Recall Recognition</i> CNN Terhadap Objek.....	35
4.4.2	Proses Pengujian <i>Recall Recognition</i> CNN Terhadap Objek	36
4.4.3	Hasil Pengujian <i>Recall Recognition</i> CNN Terhadap Objek	36
4.5	Pengujian Pengiriman/ <i>Upload</i> Data ke <i>Database</i>	39
4.5.1	Tujuan Pengujian Pengiriman/ <i>Upload</i> Data ke <i>Database</i>	39
4.5.2	Proses Pengujian Pengiriman/ <i>Upload</i> Data ke <i>Database</i>	39
4.5.3	Hasil Pengujian Pengiriman/ <i>Upload</i> Data ke <i>Database</i>	40
4.6	Pengujian Penerimaan <i>Notifikasi Android</i>	41
4.6.1	Tujuan Pengujian Penerimaan <i>Notifikasi Android</i>	41
4.6.2	Proses Pengujian Penerimaan <i>Notifikasi Android</i>	41
4.6.3	Hasil Pengujian Penerimaan <i>Notifikasi Android</i>	42
4.6.4	Hasil Pengujian Klarifikasi Peringatan Pada <i>Notifikasi</i>	44
BAB V	PENUTUP	49
5.1	Kesimpulan	49
5.2	Saran.....	49

DAFTAR PUSTAKA	50
LAMPIRAN	51
BIODATA	72



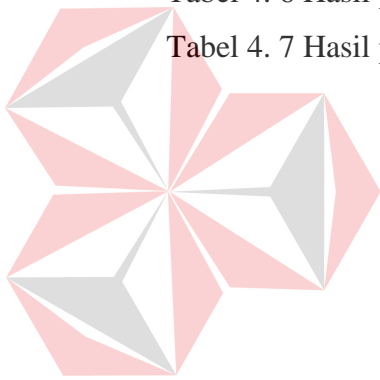
UNIVERSITAS
Dinamika

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 <i>Convolutional Neural Network</i>	5
Gambar 2. 2 <i>Feature Extraction Layer</i>	6
Gambar 2. 3 Kamera <i>Webcam</i>	12
Gambar 3. 1 Bagan Metodologi Penelitian	13
Gambar 3. 2 Arsitektur <i>Convolutional Layer</i> Untuk Pengenalan Wajah Pada Sistem Keamanan Rumah.....	14
Gambar 3. 3 Blok Diagram Sistem Keamanan Rumah.....	15
Gambar 3. 4 <i>Flowchart</i> Sistem Keamanan Rumah.....	16
Gambar 3. 5 Pengenalan Wajah	18
Gambar 3. 6 Hasil Klasifikasi Oleh Sistem	19
Gambar 3. 7 Hasil <i>Upload</i> Sistem.....	21
Gambar 3. 8 <i>Notifikasi</i> Oleh <i>Android</i>	23
Gambar 3. 9 <i>Capture</i> Dan <i>Date-Time</i>	23
Gambar 4. 1 Hasil Deteksi Objek	26
Gambar 4. 2 Hasil Akurasi <i>Recognition Cnn</i>	29
Gambar 4. 3 Hasil Presisi <i>Recognition Cnn</i>	33
Gambar 4. 4 Hasil <i>Recall Recognition</i>	36
Gambar 4. 5 Hasil Pengiriman/ <i>Upload</i> Data Ke <i>Database</i>	40
Gambar 4. 6 Hasil Penerimaan <i>Notifikasi Android</i>	42
Gambar 4. 7 Kondisi Bahaya 1	44
Gambar 4. 8 Kondisi Bahaya 2	44
Gambar 4. 9 Kondisi Bahaya 3	45
Gambar 4. 10 Peringatan <i>Notifikasi Bahaya 1</i>	45
Gambar 4. 11 Peringatan <i>Notifikasi Bahaya 2</i>	46
Gambar 4. 12 Peringatan <i>Notifikasi Bahaya 3</i>	46

DAFTAR TABEL

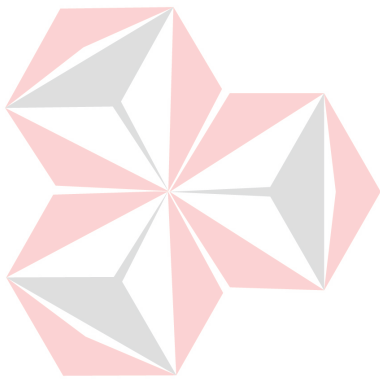
	Halaman
Tabel 2. 1 Tabel kebenaran <i>confusion matrix</i>	7
<u>Tabel 2. 2</u> Tabel prediksi mahasiswa yang berpotensi di DO.....	8
<u>Tabel 3. 1</u> Pembagian jenis <i>Notifikasi</i> sistem keamanan rumah	17
Tabel 4. 1 Hasil pengujian deteksi terhadap objek	26
Tabel 4. 2 Hasil pengujian akurasi <i>recognition</i> CNN terhadap objek	29
Tabel 4. 3 Hasil pengujian presisi <i>recognition</i> CNN terhadap objek	33
Tabel 4. 4 Hasil pengujian <i>recall recognition</i> CNN terhadap objek.....	37
Tabel 4. 5 Hasil pengujian pengiriman/ <i>upload</i> data ke <i>database</i>	40
Tabel 4. 6 Hasil pengujian penerimaan <i>notifikasi android</i>	43
Tabel 4. 7 Hasil pengujian klarifikasi peringatan pada <i>notifikasi</i>	47



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

	Halaman
LAMPIRAN 1. Hasil Pengujian	51
LAMPIRAN 2. Hasil Uji <i>Upload</i> Data.....	56
LAMPIRAN 3. Hasil Uji <i>Notifikasi</i>	59
LAMPIRAN 4. Hasil Uji Peringatan Bahaya	61
LAMPIRAN 5. <i>Listing Program</i> Sistem Keamanan Rumah.....	65



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

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, akan 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 Tugas Akhir ini terdapat sebuah sistem keamanan rumah berdasarkan *multiface recognition* berbasis *Android* menggunakan metode CNN. Dalam penerapannya sistem ini akan mampu mengidentifikasi wajah dari anggota keluarga dan yang bukan termasuk. Bilamana objek yang terdeteksi adalah anggota keluarga maka sistem akan 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 akan melakukan pengambilan foto terhadap objek dan akan mengirimkan *Notifikasi* beserta jam dan tanggal ke *smartphone user*, hal ini tentu akan meminimalisir dan pencegahan terhadap tindak kriminalitas yang akan 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 Sucusvieri dari Universitas Dinamika (Sucusvieri, 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 akan 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 (Pangestu1, 2018). Dikarenakan tugas akhir ini bertujuan untuk mengutamakan kecepatan dan keakuratan dalam melakukan pengolahan citra / indentifikasi serta pengiriman data, maka metode CNN lebih sesuai untuk diimplementasikan didalam tugas akhir ini.

1.2 Rumusan Masalah

Berdasarkan dari latar belakang diatas terdapat rumusan masalah sebagai berikut:

1. Bagaimana proses indentifikasi *multiface recognition* menggunakan metode *Convolutional Neural Network*(CNN) pada sistem keamanan rumah ?
2. Bagaimana proses pengiriman citra gambar dan *Notifikasi* ke *Android* melalui *firebase* ?

1.3 Batasan Masalah

Dalam penerapan sistem keamanan rumah berdasarkan *multiface recognition* berbasis *Android* menggunakan metode CNN terdapat beberapa batasan masalah sebagai berikut:

1. Perekaman menggunakan kamera *webcam* dengan ukuran 5MP.

2. Objek yang dideteksi adalah wajah yang utuh dan tidak terdapat penghalang secara penuh.
3. Memiliki koneksi internet yang stabil.
4. *Smartphone* sudah terpasang aplikasi *Android* yang khusus dibuat untuk menerima hasil transmit dari sistem.

1.4 Tujuan

Berdasarkan paparan rumusan masalah diatas, maka adapun tujuan dari Tugas Akhir ini ialah:

1. Proses identifikasi *multiface recognition* menggunakan metode *Convolutional Neural Network* (CNN).
2. Melakukan pengiriman citra gambar dan *Notifikasi* ke *Android* melalui *firebase*.

1.5 Sistematika Penulisan

Pembahasan Tugas Akhir ini disusun menjadi 5 (lima) garis besar bab pembahasan, yaitu sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini menyajikan pembahasan, mengenai latar belakang, rumusan masalah, batasan masalah, tujuan dari, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini disajikan teori penunjang dari permasalahan, diantaranya *Convolution Neural Network, Firebase, multiface, Computer Vision, OpenCV, Python.*

BAB III METODE PENELITIAN

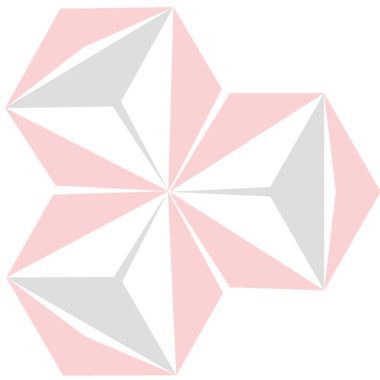
Pada bab ini disajikan mengenai tahapan perancangan perangkat lunak sistem keamanan. Dijelaskan proses pembuatan sub-sub program dan metode percobaan yang digunakan untuk menguji dan analisis sistem keamanan.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini membahas tentang meliputi pengujian deteksi terhadap objek, pengujian akurasi, presisi, dan *recall*, pengujian pengunggahan dan keberhasilan penerimaan *notifikasi* pada *Android*.

BAB V PENUTUP

Pada bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan serta saran untuk pengembangan penelitian selanjutnya.



UNIVERSITAS
Dinamika

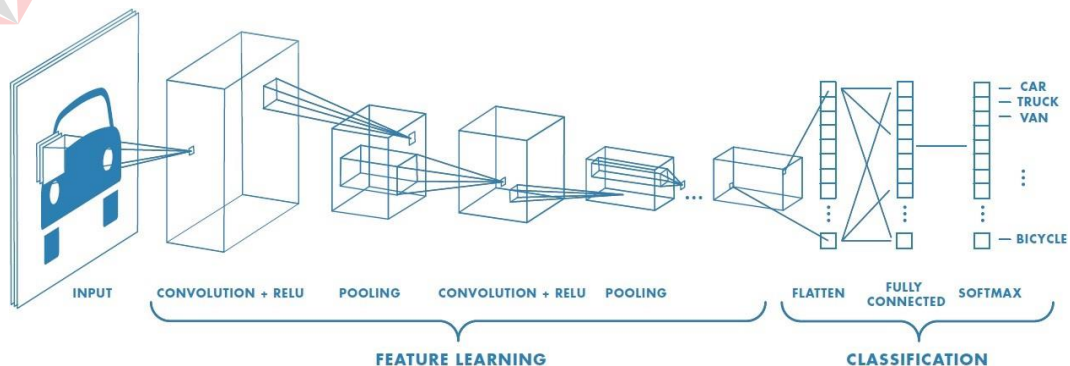
BAB II

LANDASAN TEORI

2.1 Convolutional Neural Network

Multilayer Perceptron yang juga dikenal dengan CNN atau *ConvNet* adalah pengembangan dari *Multilayer Perceptron* (MLP) untuk mengolah data dua dimensi. *Multilayer Perceptron* sendiri adalah pengembangan dari *Artificial Neural Network* (ANN) yang ditujukan untuk menutupi kekurangan *Artificial Neural Network* (ANN) dengan *Single-layer Perceptron* dalam menyelesaikan operasi logika yang kompleks. Hal tersebut dimungkinkan dengan menambahkan *hidden layer* yang membuat ANN *powerful* untuk memecahkan operasi logika yang kompleks (*universal approximation*) dan sering digunakan untuk permasalahan-permasalahan klasifikasi, *recognition*, dan prediksi (A., 2016).

Convolutional Neural Network merupakan salah satu dari berbagai jenis neural network yang sering digunakan pada data *image*. Selain itu CNN juga sering digunakan untuk melakukan deteksi dan identifikasi objek terhadap sebuah citra gambar (Sena, 2017).



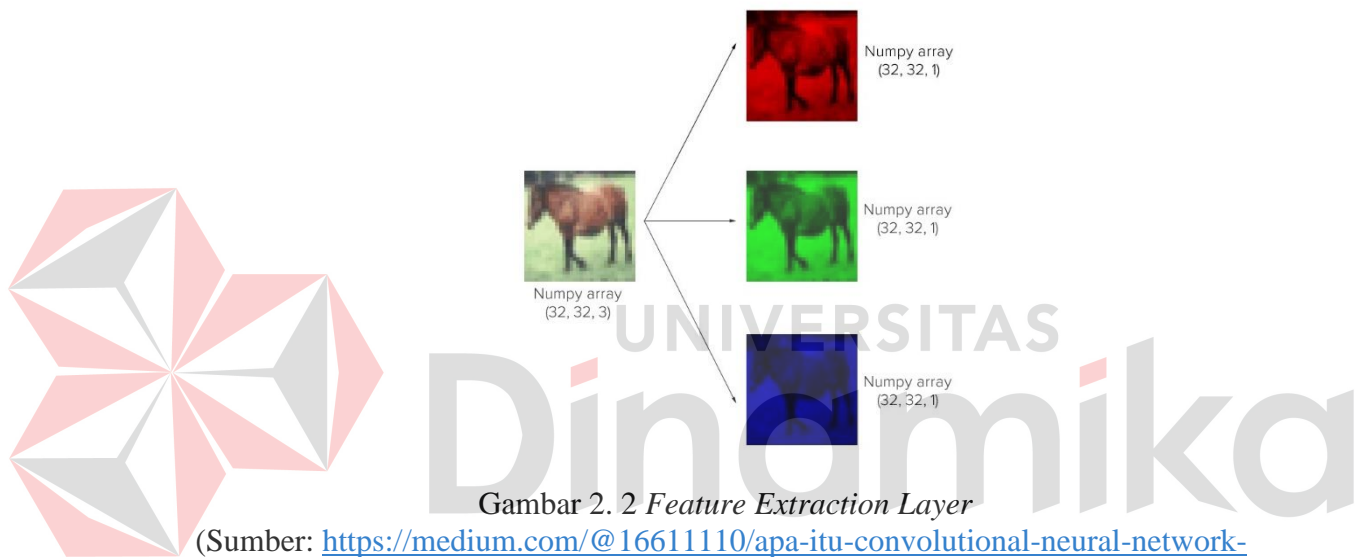
Gambar 2. 1 Convolutional Neural Network

(Sumber: <https://medium.com/@nadhifasofia/1-convolutional-neural-network-convolutional-neural-network-merupakan-salah-satu-metode-machine-28189e17335b>)

Secara garis besar CNN dapat dibagi/dibedakan menjadi 2 bagian, yakni *feature extraction layer* dan *fully-connected layer*. Adapun penjelasan kedua istilah tersebut sebagai berikut:

1. *Feature extraction layer*

Feature extraction layer merupakan sebuah istilah yang kebanyakan dipakai oleh *programmer image processing* sebagai istilah lain dalam proses *encoding* dari sebuah citra gambar menjadi sebuah *features* atau sebuah angka-angka yang mewakili atau mempresentasikan gambar tersebut.



Gambar 2. 2 *Feature Extraction Layer*

(Sumber: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>)

Feature extraction layer juga terdiri dari dua bagian, yakni *Convolutional layer* dan juga *pooling layer*. Tetapi pada umumnya terdapat beberapa riset/paper yang tidak menggunakan *pooling*.

a. *Convolutional layer*

Convolutional layer merupakan sebuah neuron yang tersusun sangat kompleks dan membentuk sebuah filter, dimana filter tersebut memiliki piksel (panjang dan tinggi). *Convolutional layer* juga bisa dianggap sebagai sebuah metode untuk melakukan komputasi dalam mempresentasikan sebuah citra *image*.

b. *Pooling layer*

Pooling layer pada umumnya terdapat setelah *Convolutional layer*, dimana pada prinsipnya *pooling layer* juga terdiri dari sebuah filter yang akan bergerak atau bergeser keseluruh area yang terdapat pada *feature map* dengan memiliki ukuran serta *stride* tertentu. *Pooling* yang biasa digunakan ialah *max pooling* dan *average pooling*. Seperti halnya namanya, *max pooling* mengambil nilai terbesar dan *average pooling* mengambil nilai rata-rata dari data representasi *image*.

2. *Fully-connected layer*

Fully-connected layer merupakan sebuah *feature map* yang dihasilkan dari proses *feature ekstraction layer* yang masih berbentuk *array* multidimensional. Dikarenakan *layer* ini masih dalam bentuk yang multidimensional *array*, maka perlu dilakukan proses *flatten* atau *reshape* terhadap *feature map* agar menjadi sebuah vektor yang bisa digunakan sebagai input dari *fully-connected layer*.

2.2 Akurasi, Presisi dan *Recall* dalam *Machine Learning*

Dalam melakukan sebuah evaluasi terhadap sebuah performa dari algoritma *machine learning* dapat menggunakan sebuah acuan *confusion matrix*. *Confusion matrix* berfungsi untuk mempresentasikan sebuah prediksi dan kondisi aktual atau kondisi yang sebenarnya. Berdasarkan *confusion matrix*, dapat diperoleh atau dapat menentukan akurasi, presisi dan juga *recall* dari sebuah hasil sistem algoritma.

Secara sederhana dapat dibuat sebuah ringkasan terkait pemahaman akurasi, presisi, dan *recall* kedalam sebuah tabel kebenaran. Hal ini untuk membantu memudahkan pemahaman terkait sub topik yang akan dibahas.

Tabel 2. 1 Tabel kebenaran *confusion matrix*

Prediksi / Aktual	Positif	Negatif
Positif	TP	FP
Negatif	FN	TN

Ket. TP : *True Positive*

FP : *False Posistive*

FN : *False Negative*

TN : *True Negative*

Terdapat juga sebuah tabel untuk mempermudah pemahaman dengan pemberian studi kasus secara langsung.

Tabel 2. 2 Tabel prediksi mahasiswa yang berpotensi di DO

NIM	Status Sebenarnya	Hasil Prediksi
001	Tidak DO	Tidak DO
002	Tidak DO	Tidak DO
003	Tidak DO	Tidak DO
004	Tidak DO	DO
005	Tidak DO	DO
006	DO	Tidak DO
007	DO	DO
008	DO	DO
009	DO	DO
010	Tidak DO	Tidak DO

Dari Tabel 2.2 dapat diketahui bahwa :

- TP : dimana dalam kasusnya mahasiswa yang diprediksi (positif) DO ternyata tidak jadi di DO (*true*). berdasarkan tabel terdapat 3 mahasiswa, yaitu dengan nim 007, 008, dan 009.
- TN : dimana dalam kasusnya mahasiswa yang diprediksi (negatif) DO memang benar tidak di DO (*true*). berdasarkan tabel terdapat 3 mahasiswa, yaitu dengan nim 001, 002, dan 003.
- FP : dimana dalam kasusnya mahasiswa yang diprediksi (positif) DO ternyata tidak jadi di DO (*false*) berdasarkan tabel terdapat 2 mahasiswa, yaitu dengan nim 004 dan 005.
- FN : dimana dalam kasusnya mahasiswa yang diprediksi (negatif) DO ternyata benar tidak di DO (*true*). berdasarkan tabel terdapat 1 mahasiswa, yaitu dengan nim 010.

Dalam pengukuran performa, akurasi merupakan sebuah rasio dari prediksi *true* (*positive* dan *negative*) dari keseluruhan data. sedangkan presisi merupakan sebuah rasio prediksi dimana *true positive* dibandingkan dengan seluruh data yang *true*

positive. dan *recall* merupakan rasio prediksi *true positive* dibandingkan dengan seluruh data yang *true positive*.

Rumus :

- a. Akurasi = $(TP + TN) / (TP+FP+FN+TN)$
- b. Presisi = $(TP) / (TP+FP)$
- c. *Recall* = $(TP) / (TP + FN)$

2.3 Image Processing

Image Processing atau pemrosesan gambar digital adalah penggunaan komputer digital untuk memproses gambar digital melalui suatu algoritma, sebagai sub-kategori atau bidang pemrosesan sinyal digital. Pemrosesan citra digital memiliki banyak keunggulan dibandingkan pemrosesan citra analog. Hal ini memungkinkan jangkauan algoritma yang jauh lebih luas untuk diterapkan pada data input dan dapat menghindari masalah seperti penumpukan data dan distorsi selama pemrosesan. Hal itu terjadi karena gambar didefinisikan lebih dari dua dimensi atau lebih (Raut, 2018).

Pemrosesan gambar digital dapat dimodelkan dalam bentuk sistem multidimensi. Generasi dan pengembangan pemrosesan gambar digital terutama dipengaruhi oleh tiga faktor: pertama, pengembangan komputer; kedua, pengembangan matematika (terutama penciptaan dan peningkatan teori matematika diskrit); ketiga, permintaan untuk berbagai aplikasi di lingkungan, pertanian, militer, industri dan ilmu kedokteran telah meningkat (Belharbi S. , 2015). Selain itu terdapat juga filter digital yang dapat digunakan untuk mengaburkan dan mempertajam gambar digital (Pangestu1, 2018). Penyaringan dapat dilakukan dengan konvolusi dengan kernel yang dirancang khusus (*array filter*) dalam domain spasial menutupi wilayah frekuensi tertentu dalam domain frekuensi (*Fourier*).

2.4 *Firestore*

Selama beberapa tahun terakhir *Firestore* telah berhasil membuat sebuah pencapaian besar, yakni *Realtime Database* yang mana merupakan sebuah alat yang memungkinkan pengembang untuk mengatur dan menyediakan basis data *back-end* sebagai layanan yang lebih mudah. Para *developer* terus memilih Basis Data *Realtime* karena hambatan masuk yang rendah, biaya perawatan yang rendah, dan permintaan yang cepat. Sistem tentu bukan tanpa kelemahannya, namun, mengarah ke yang lain mencari alternatif untuk *Firestore* (J., 2009).

2.5 *Python*

Python adalah bahasa pemrograman dengan tujuan umum tingkat tinggi yang ditafsirkan. Diciptakan oleh Guido van Rossum dan dirilis pertama kali pada tahun 1991, filosofi desain *Python* menekankan keterbacaan kode dengan penggunaan spasi spasi yang signifikan. Konstruksi bahasanya dan pendekatan berorientasi objek yang bertujuan untuk membantu programmer menulis kode yang jelas dan logis untuk proyek skala kecil maupun besar.

2.6 *OpenCV*

OpenCV (Open source Computer Vision) adalah pustaka/library dari fungsi pemrograman yang terutama ditujukan untuk *Computer Vision* yang *realtime*. Pada awalnya *OpenCV* dikembangkan oleh Intel, dan kemudian didukung oleh Willow Garage lalu Itseez (yang kemudian diakuisisi oleh Intel). Perpustakaan/library dari *OpenCV* adalah lintas-platform dan gratis untuk digunakan di bawah lisensi BSD *open-source*. *OpenCV* juga mendukung kerangka pembelajaran yang terdapat pada *TensorFlow*, *Torch / PyTorch* dan *Caffe*.

2.7 *Android Studio*

Android Studio adalah lingkungan pengembangan yang terintegrasi resmi (IDE) untuk sistem operasi *Android* Google, yang dibangun di atas perangkat lunak *IntelliJ IDEA JetBrains* dan dirancang khusus untuk pengembangan *Android*. *Android Studio* tersedia untuk diunduh pada sistem operasi berbasis *Windows*, *macOS* dan *Linux*. *Android Studio* merupakan pengganti dari *Eclipse Android Development Tools* (ADT) sebagai IDE utama untuk pengembangan aplikasi *Android* asli.

2.8 *Open Computer Vision (OpenCV)*

OpenCV adalah sebuah *library* yang berisikan banyak *function* pemrograman untuk teknologi *Computer Vision* secara *realtime* yang bersifat *open source* (bebas digunakan untuk hal yang bersifat akademis maupun komersial). *Library* ini ditulis dalam *C* dan *C++* dan berjalan di bawah *Linux*, *Windows* dan *Mac OS X*. *OpenCV* ditulis dalam *C* yang dioptimalkan dan dapat memanfaatkan prosesor *multicore*.

2.9 *Computer Vision*

Computer Vision adalah bidang ilmiah interdisipliner yang membahas bagaimana komputer dapat dibuat untuk memperoleh pemahaman tingkat tinggi dari gambar atau video digital. Dari perspektif teknik, *Computer Vision* berupaya mengotomatiskan tugas-tugas yang dapat dilakukan oleh sistem visual manusia.

Tugas visual komputer meliputi metode untuk memperoleh, memproses, menganalisis dan memahami gambar digital, dan ekstraksi data dimensi tinggi dari dunia nyata untuk menghasilkan informasi numerik atau simbolik, mis. dalam bentuk keputusan. Pemahaman dalam konteks ini berarti transformasi gambar visual (input retina) menjadi deskripsi dunia yang dapat berinteraksi dengan proses pemikiran lain dan memperoleh tindakan yang sesuai. Pemahaman gambar ini dapat dilihat sebagai penguraian informasi simbolik dari data gambar yang menggunakan model yang

dibangun dengan bantuan geometri, fisika, statistik, dan teori pembelajaran (Febriana, 2015).

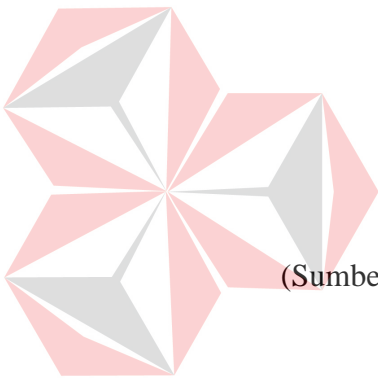
2.10 Kamera

Kamera *webcam* pada sistem ini diperlukan sebagai piranti pengambil data input yang akan diproses oleh sistem nantinya.



Gambar 2. 3 Kamera *Webcam*

(Sumber : <https://www.amazon.fr/Logitech-C930e-Webcam-vision-degr%C3%A9s/dp/B00CES5A60>)

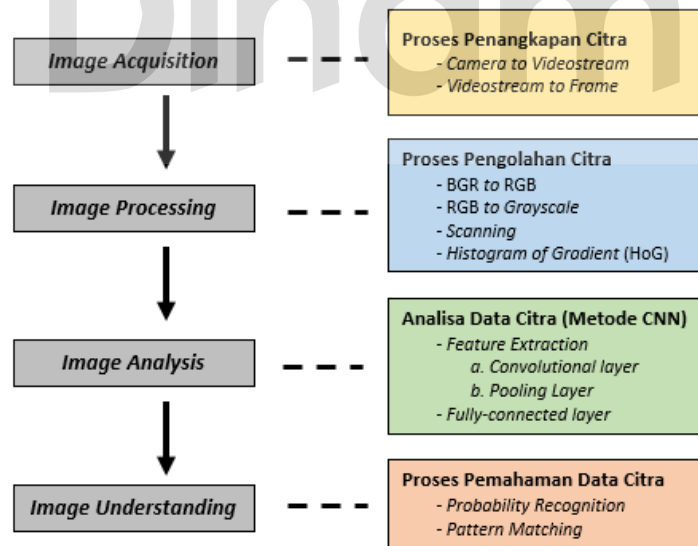


UNIVERSITAS
Dinamika

BAB III METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode penelitian dalam melakukan Tugas Akhir ini terdapat beberapa tahap. Dimana tahapan ini juga merupakan sebuah alur sistem yang sinergi dalam metode penelitian yang akan 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 akan 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 akan memutuskan hasilnya dan akan mengirim/*upload* data kedalam *database server*. Setelah data terkirim dan berhasil diterima oleh *database server*, maka secara otomatis sistem akan melakukan *forwarding* ke *Android application* dengan mengirimkan data *string* dan data *image* kepada pengguna.

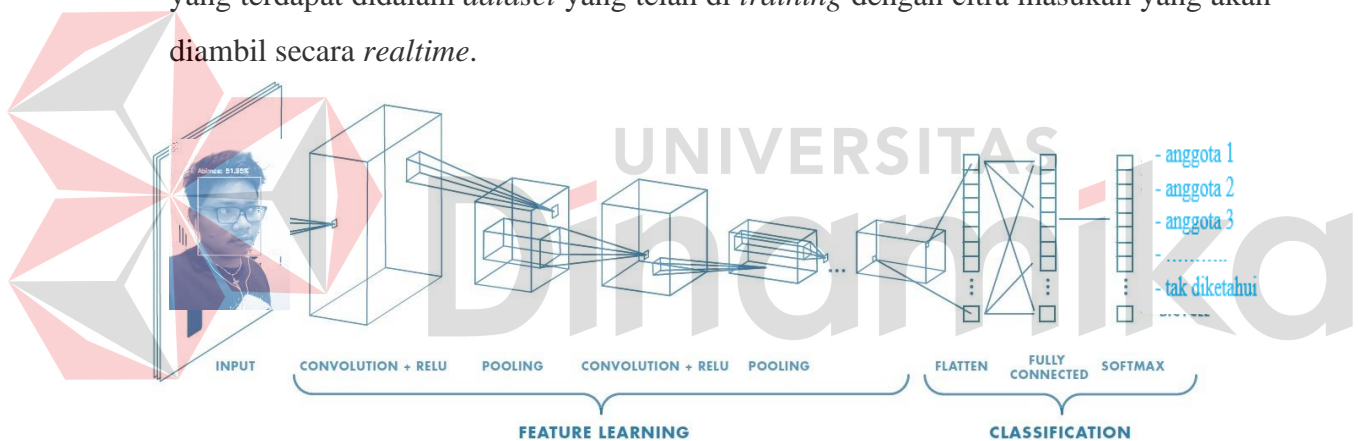


Gambar 3. 1 Bagan Metodologi Penelitian

Secara spesifik terkait *layer* yang digunakan dalam metode CNN adalah *feature extraction layer* dengan dijalankan 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* (Zisserman, 2015).

Convolutional layer merupakan sebuah neuron yang tersusun sangat kompleks dan membentuk sebuah filter, dimana filter tersebut akan membentuk sebuah pixel pada citra untuk melakukan komputasi dan menentukan nilai pixel dari tiap gambar. Dalam algoritma ini *pooling layer* akan menggunakan *average pooling* mengambil nilai rata-rata dari data representasi *image*. Hasil representasi data tersebut yang nantinya akan digunakan dalam melakukan identifikasi / kecocokan terhadap nilai citra yang terdapat didalam *dataset* yang telah di *training* dengan citra masukan yang akan diambil secara *realtime*.



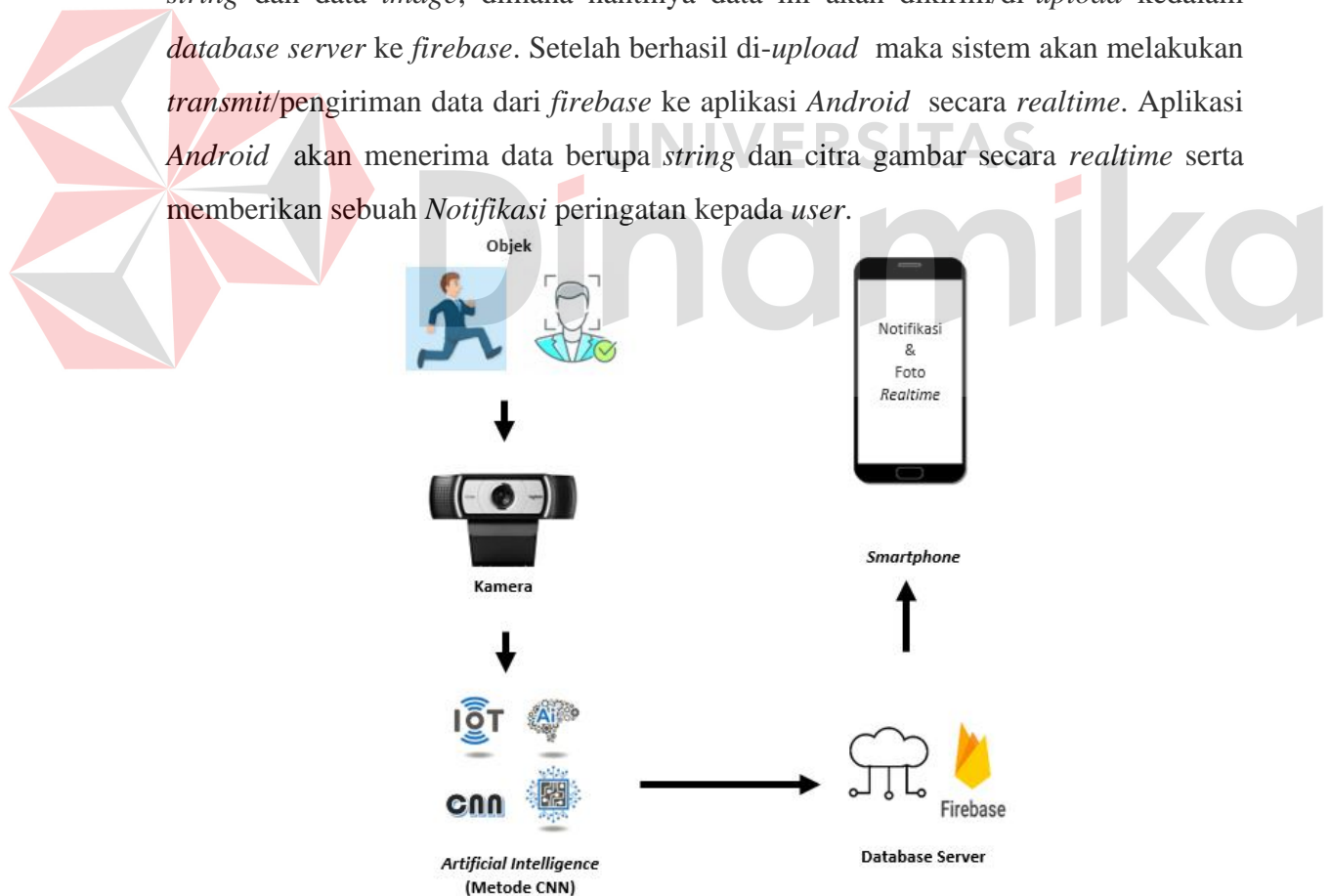
Gambar 3. 2 Arsitektur *Convolutional layer* Untuk Pengenalan Wajah Pada Sistem Keamanan Rumah

Arsitektur yang digunakan dalam metode CNN dapat diamati pada Gambar 3.2. 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 akan 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 *polling layer* dengan menggunakan *average polling* akan 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.

3.2 Perancangan Perangkat Keras

Perancangan program pada tugas akhir bisa diamati pada Gambar 3.3. sistem akan mulai melakukan identifikasi ketika ditemukan sebuah objek berupa *human/manusia*. Objek akan ditangkap oleh kamera dan akan 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 akan menghasilkan data *output* berupa data *string* dan data *image*, dimana nantinya data ini akan dikirim/di-*upload* kedalam *database server* ke *firebase*. Setelah berhasil di-*upload* maka sistem akan melakukan *transmit*/pengiriman data dari *firebase* ke aplikasi *Android* secara *realtime*. Aplikasi *Android* akan menerima data berupa *string* dan citra gambar secara *realtime* serta memberikan sebuah *Notifikasi* peringatan kepada *user*.

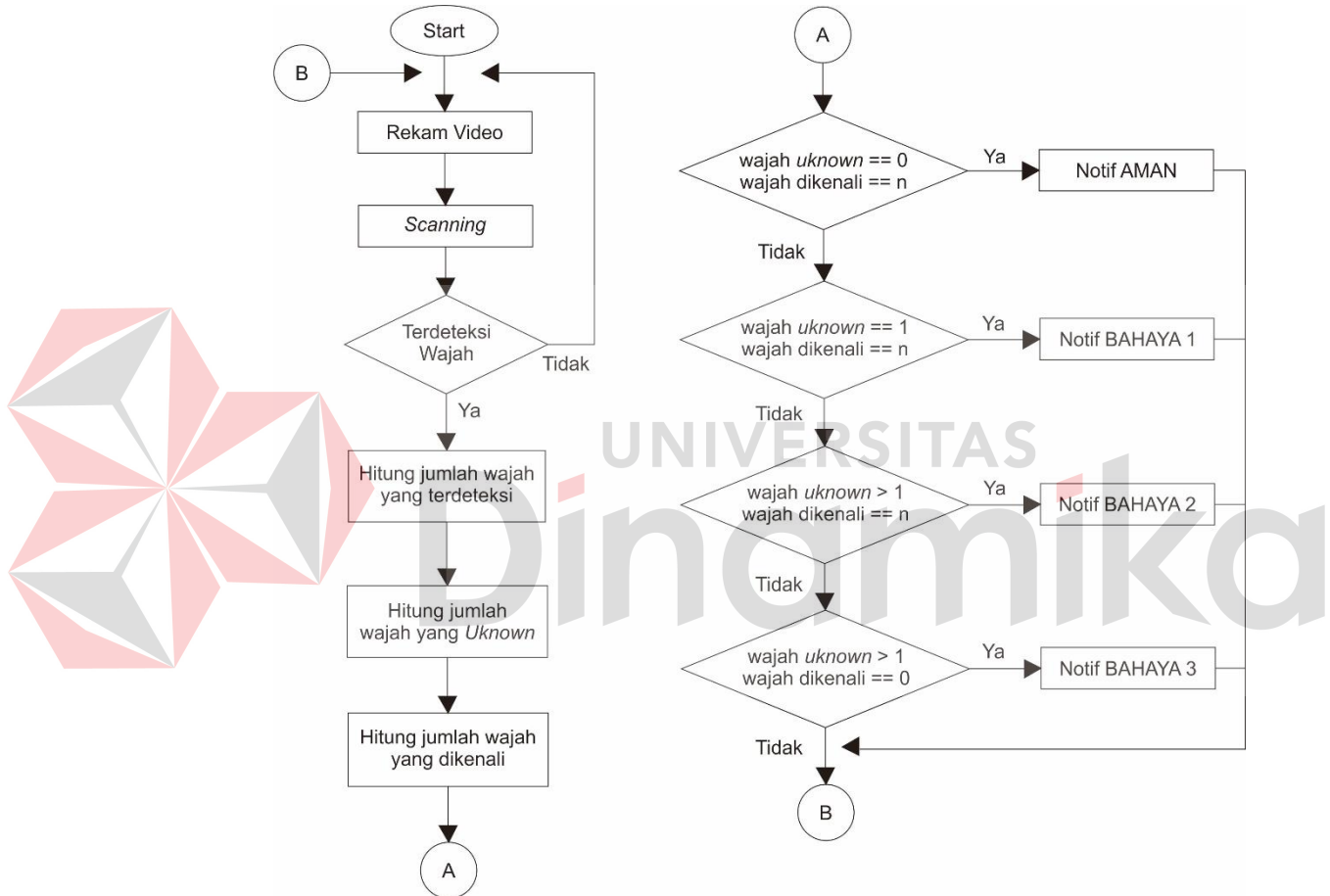


Gambar 3. 3 Blok Diagram Sistem Keamanan Rumah

3.3 Perancangan Program

Flowchart Sistem Keamanan Rumah

Proses dari pengolahan data atau *flowchart* program dengan menggunakan *artificial intelligent* dapat diamati pada gambar 3.4. yang mana pada gambar tersebut terdapat tahapan dari jalannya sistem program.



Gambar 3. 4 Flowchart Sistem Keamanan Rumah

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

mendapatkan sebuah citra *frame* maka program akan melakukan *scanning* dengan menggunakan *similarity transform* untuk menemukan sebuah citra wajah dari *image*.

Tabel 3.1 Pembagian jenis *Notifikasi* sistem keamanan rumah

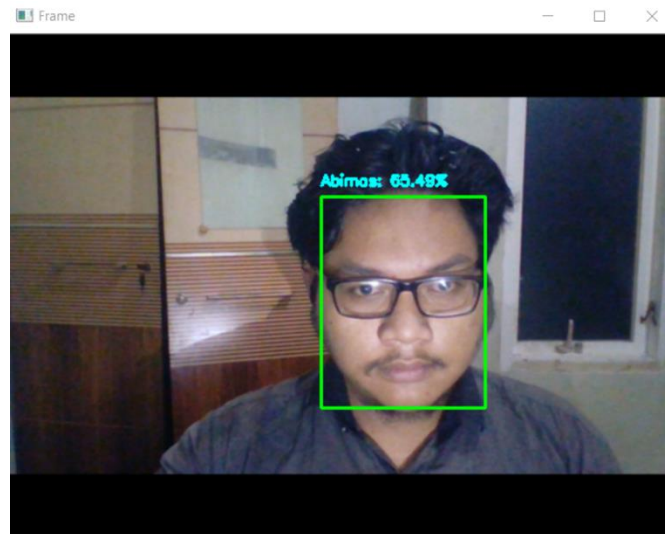
No	Wajah dikenali	Wajah tidak dikenali	Jenis <i>Notifikasi</i>
1	n	0	Aman
2	n	1	BAHAYA 1
3	n	>1	BAHAYA 2
4	0	n	BAHAYA 3

Ket : $n = 1$ atau lebih

Ketika sebuah wajah terdeteksi, maka sistem akan melakukan perhitungan terhadap jumlah wajah yang terdeteksi. Setelah itu sistem akan melakukan sebuah proses identifikasi wajah (*multiface recognition*) untuk menentukan apakah wajah objek yang terdeteksi dikenali atau tidak. Dalam tahapan selanjutnya, kondisi tersebut akan 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.

3.4 Pengenalan / *Recognition* Wajah

Ketika program mulai dijalankan, maka program akan melakukan sebuah proses untuk identifikasi terhadap objek yang tertangkap oleh kamera. Didalam prosesnya, sistem akan melakukan sebuah deteksi wajah terlebih dahulu. Pada saat wajah telah terdeteksi maka sistem akan 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 3.5.



Gambar 3. 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 akan melakukan proses *recognition* berdasarkan dari daaset yang telah dibuat seblumnya untuk menentukan objek mana saja yang 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]
```

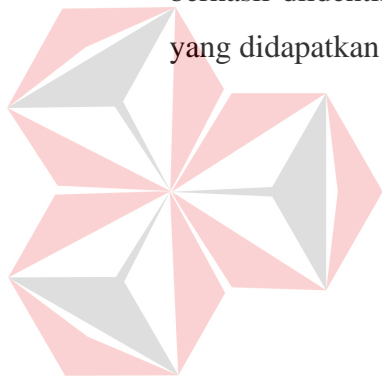
3.5 Penghitungan Objek

Penghitungan objek merupakan tahapan yang selanjutnya dalam proses untuk menentukan tingkat keamanan yang akan 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 akan diolah kembali menjadi sebuah step untuk menentukan tingkat keamanan berdasarkan dari jumlah objek yang terdeteksi dengan sebuah klasifikasi khusus yang sebelumnya telah ditentukan oleh *user*. 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 1 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 akan dilakukan dalam menangani bahaya berdasarkan hasil yang berhasil diidentifikasi oleh sistem. Pada Gambar 3.6. dapat diamati hasil klasifikasi yang didapatkan oleh sistem.



```

Command Prompt
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 0 Keluarga dan 1 Orang Asing
Peringatan 2 !!
Terdeteksi : 0 Keluarga dan 1 Orang Asing
Peringatan 2 !!
Terdeteksi : 0 Keluarga dan 1 Orang Asing
Peringatan 2 !!
Terdeteksi : 0 Keluarga dan 1 Orang Asing
Peringatan 2 !!
Terdeteksi : 0 Keluarga dan 1 Orang Asing
Peringatan 2 !!
Terdeteksi : 0 Keluarga dan 1 Orang Asing
Peringatan 2 !!
Terdeteksi : 0 Keluarga dan 1 Orang Asing
Peringatan 2 !!
Terdeteksi : 0 Keluarga dan 1 Orang Asing
Peringatan 2 !!
Terdeteksi : 0 Keluarga dan 1 Orang Asing
Peringatan 2 !!
Terdeteksi : 0 Keluarga dan 1 Orang Asing
Peringatan 2 !!
Terdeteksi : 0 Keluarga dan 1 Orang Asing

```

Gambar 3. 6 Hasil Klasifikasi 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)).s
    tream(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)).s
    tream(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)).s
    tream(stream_handler3,None)
    dataint = dataint + 1
    sending = False
```



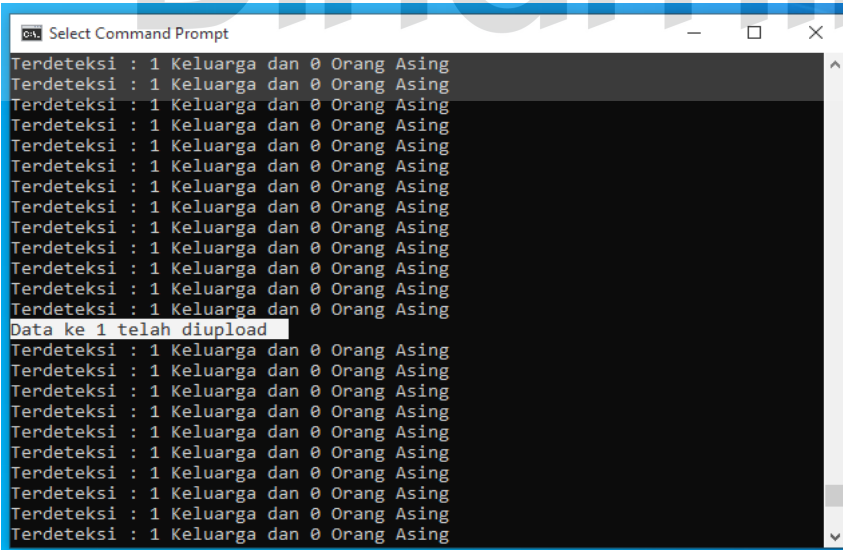
UNIVERSITAS
Diponegoro
Diponegoro

3.6 Pengiriman *Notifikasi Android*

Pada pelaksanaannya sistem ini juga akan melakukan sebuah pengiriman *Notifikasi* kedalam perangkat *Android user*. Pada dasarnya tahap ini juga merupakan sebuah tahap pengolahan sistem yang cukup kompleks. Sehingga untuk selanjutnya akan 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*.

3.6.1 Proses Pengiriman *Notifikasi* Oleh Sistem

Pengiriman *Notifikasi* oleh sistem ini akan dilakukan oleh program dengan memanfaatkan *library* dari *firebase* sebagai *server* yang dimana *server* ini telah bersifat *opensource* dan terintegrasi atau kompatibel dengan bahasa pemrograman *Python*. Secara umum dapat dijelaskan bahwa setelah sistem melakukan identifikasi objek dengan menggunakan metode CNN serta dilakukan klasifikasi berdasarkan kondisi yang telah dibuat oleh *user*, maka sistem akan melakukan eksekusi program sesuai dengan kondisi yang telah terpenuhi. Jika salah satu dari ketiga kondisi telah terpenuhi maka secara otomatis sistem akan melakukan *capture* frame/gambar terhadap object yang telah teridentifikasi dan selanjutnya sistem akan melakukan proses *uploading* data kedalam *firebase*. Dimana *firebase* ini merupakan sebuah *server* untuk menghubungkan sistem dengan *Android*. Proses pengiriman data ini berupa data *string* yang terdiri atas jenis identifikas (nama keluarga / tak dikenal), waktu dan tanggal pengambilan / pengiriman data, serta data *images* / gambar dari hasil *capture* sistem berdasarkan hasil identifikasi. Pada Gambar 3.7. dapat diamati proses CMD hasil *capture* dari data *image* yang telah didapatkan oleh sistem dan akan di *upload* kedalam *server firebase* melalui koneksi internet.



```
cmd Select Command Prompt
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Data ke 1 telah diupload
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
```

Gambar 3. 7 Hasil *Upload* Sistem

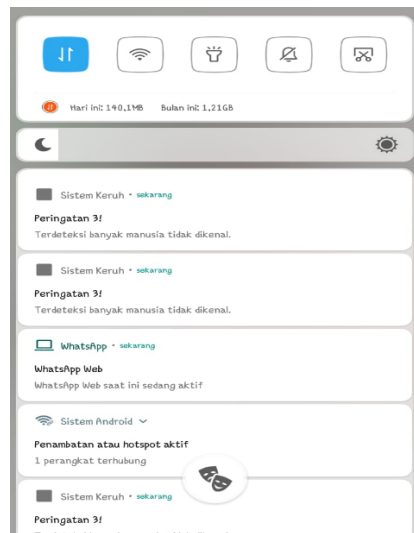
Hasil dari pengiriman data oleh sistem ini nantinya akan disimpan kedalam database yang terdapat pada *firebase*. *Syntax* untuk melakukan *uploading* data dan proses pemberian *Notifikasi* terhadap aplikasi *Android* dapat diamati pada teks berikut:

```
def stream_handler(message):
    rumah.".format(message))
    print("Peringatan 1 !!")
    response = pn_client.publish( interests=['hello'],
        publish_body={'apns': { 'aps': {'alert': 'Hello!'}, }, },
        'fcm': {'notification': {'title': 'Peringatan 1!',
            'body': 'Terdeteksi 1 manusia tidak dikenal.', }, }, },
```

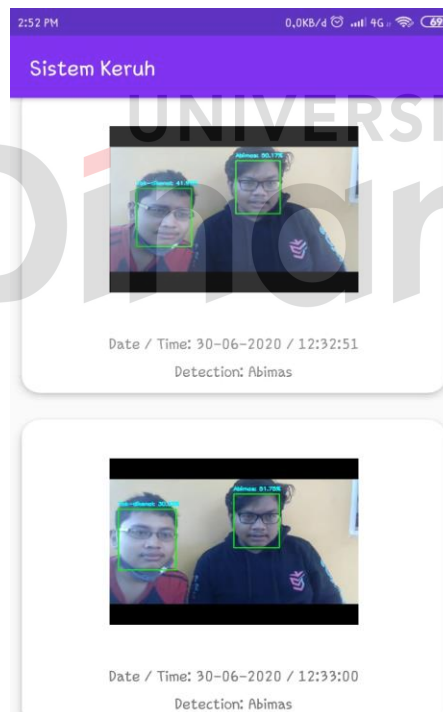
3.6.2 Proses Penerimaan *Notifikasi* Oleh *Android*

Pada proses selanjutnya setelah data berhasil dikirim oleh sistem kedalam *firebase*, maka secara otomatis aplikasi yang telah dibuat dan diinstall kedalam *smartphone Android* akan segera melakukan pengunduhan / *downloading* data dan akan memunculkan notifikasi kepada *user*. Dalam tahapannya ketika sistem telah berhasil melakukan pengiriman data kedalam *firebase*, maka sistem juga akan 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 akan melakukan *scanning* atau pengecekan data yang terdapat pada *firebase*. Apabila terdapat perubahan data pada *firebase* maupun pengurangan data, maka secara *realtime* aplikasi *Android* juga akan 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 dari proses penerimaan *notifikasi* oleh *Android* dapat diamati pada Gambar 3.8 serta hasil dari penerimaan data berupa hasil *capture* dan *date-time* dapat diamati pada Gambar 3.9.



Gambar 3. 8 Notifikasi oleh Android



Gambar 3. 9 Capture Dan Date-Time

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan terkait dari hasil dan pembahasan terhadap proses pengujian yang telah dilakukan oleh penulis. Adapun beberapa pengujian yang telah dilakukan oleh penulis adalah pengujian pengiriman/*upload* data ke *database*, pengujian penerimaan *notifikasi Android*, pengujian deteksi terhadap objek, pengujian akurasi, presisi, dan *recall* terhadap objek.

4.1 Pengujian Deteksi Terhadap Objek

Dalam pengujian deteksi terhadap objek terdapat beberapa hal yang harus kita ketahui dan pahami. Adapun beberapa hal tersebut yakni sebagai berikut.

4.1.1 Tujuan Pengujian Deteksi Terhadap Objek

Pengujian deteksi terhadap objek ini dilakukan untuk melihat tingkat akurasi deteksi orang yang akan dilakukan oleh sistem. Dimana pada tahapan ini kita dapat mengetahui apakah sistem dapat mengenali antara manusia dengan objek benda mati.

4.1.2 Proses Pengujian Deteksi Terhadap Objek

Dalam proses pengujian terdapat beberapa prosedur yang perlu dilakukan untuk melakukan pengambilan data. Adapun beberapa prosedur yang perlu dilakukan dalam proses pengujian deteksi terhadap objek adalah sebagai berikut:

1. Menjalankan program yang telah dibuat dengan urutan alur sistem, adapun urutan *running* program serta penjelasan singkatnya sebagai berikut.

- a. *buat_dataset.py*

Program ini digunakan untuk melakukan pengolahan citra terhadap *dataset* yang telah dibuat agar nantinya bisa terorganisir oleh sistem.

b. *pelatihan_dataset.py*

Program ini berfungsi untuk melakukan *rescaning* serta *labeling* terhadap *dataset* yang telah dibuat pada program poin (a).

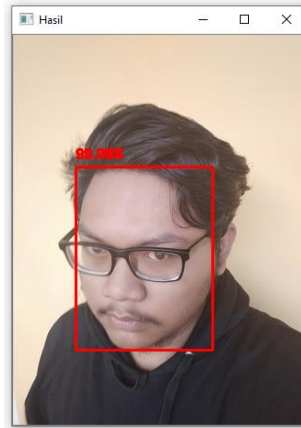
c. *keamanan_rumah.py*

Program ini merupakan program inti yang berfungsi untuk melakukan pengenalan terhadap objek yang muncul pada kamera menggunakan metode CNN dengan berdasarkan pada *dataset* yang telah dibuat seperti yang dijelaskan pada poin (a) dan (b).

2. Pengambilan data dari setiap objek yang akan diklasifikasikan, dimana objek yang ditentukan pada pembahasan ini adalah tiga anggota keluarga dan satu orang tak dikenal. Dimana pengambilan data dari masing-masing objek sebanyak 10 kali.
3. Setiap pengambilan data dilakukan dengan melakukan 10 posisi yang berbeda oleh setiap objek.
4. Untuk mempermudah proses pengolahan data maupun analisis, maka akan dilakukan *sreenshoot frame* setiap 1 menit. Kemudian melakukan *rename* dari hasil *sreenshoot frame* dengan nomor sesuai urutan pengambilan.
5. Kemudian memasukkan nomor kedalam tabel pada tab nomer *sreenshoot*.
6. Isi tabel yang telah dibuat sesuai dengan hasil yang diperoleh selama pengujian dengan memberi centang (v) pada hasil pengujian.

4.1.3 Hasil Pengujian 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 4.1 dibawah ini.



Gambar 4. 1 Hasil Deteksi Objek

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

Tabel 4. 1 Hasil pengujian deteksi terhadap objek

Uji ke-	Objek	Hasil Uji Deteksi		Nomer <i>Screenshot</i>
		Berhasil	Gagal	
1	Anggota 1	V	-	a1
	Anggota 2	V	-	b1
	Anggota 3	V	-	c1
	Tak dikenal	V	-	d1
2	Anggota 1	V	-	a2
	Anggota 2	V	-	b2
	Anggota 3	V	-	c2
	Tak dikenal	V	-	d2
3	Anggota 1	V	-	a3
	Anggota 2	V	-	b3
	Anggota 3	V	-	c3
	Tak dikenal	V	-	d3
4	Anggota 1	V	-	a4
	Anggota 2	V	-	b4
	Anggota 3	V	-	c4
	Tak dikenal	V	-	d4
5	Anggota 1	V	-	a5
	Anggota 2	V	-	b5
	Anggota 3	V	-	c5
	Tak dikenal	V	-	d5
6	Anggota 1	V	-	a6
	Anggota 2	V	-	b6
	Anggota 3	V	-	c6

Uji ke-	Objek	Hasil Uji Deteksi		Nomer <i>Screenshot</i>
		Berhasil	Gagal	
7	Tak dikenal	V	-	d6
	Anggota 1	V	-	a7
	Anggota 2	V	-	b7
8	Anggota 3	V	-	c7
	Tak dikenal	V	-	d7
	Anggota 1	V	-	a8
9	Anggota 2	V	-	b8
	Anggota 3	V	-	c8
	Tak dikenal	V	-	d8
10	Anggota 1	V	-	a9
	Anggota 2	V	-	b9
	Anggota 3	V	-	c9
	Tak dikenal	V	-	d9
	Anggota 1	V	-	a10
	Anggota 2	V	-	b10
	Anggota 3	V	-	c10
	Tak dikenal	V	-	d10

Dari hasil Tabel 4.1 diatas, dapat dijelaskan bahwa dari 40 data sampel uji coba terhadap 4 responden memiliki presisi recognition terhadap objek dengan baik. Dimana sistem mendeteksi sebagai anggota x dan secara aktual memang benar anggota x disimbolkan dengan TP. Jika sistem mendeteksi sebagai tak dikenal / anggota y / z dan memang aktualnya tak dikenal / anggota y / z disimbolkan dengan TN. Dan sistem mendeteksi anggota x sedangkan aktualnya adalah tak dikenal / anggota y / z disimbolkan dengan FP. Sedangkan sistem mendeteksi tak dikenal / anggota y / z tetapi aktualnya adalah anggota x disimbolkan dengan FN sehingga persentase data dari setiap anggota maupun tak dikenal dapat dihitung menggunakan persamaan 4.1 sebagai berikut:

$$\begin{aligned}
 \text{Akurasi Deteksi} &= \frac{(TP + TN)}{TP + FP + FN + TN} \times 100\% & (4.1) \\
 &= \frac{(10 + 30)}{10 + 0 + 0 + 30} \times 100\% \\
 &= \frac{40}{40} \times 100\% \\
 &= 100\%
 \end{aligned}$$

4.2 Pengujian Akurasi *Recognition* CNN Terhadap Objek

Dalam akurasi *recognition* CNN terhadap objek terdapat beberapa hal yang harus kita ketahui dan pahami. Adapun beberapa hal tersebut yakni sebagai berikut.

4.2.1 Tujuan Pengujian Akurasi *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 (Latasha, 2018).

4.2.2 Proses Pengujian Akurasi *Recognition* CNN Terhadap Objek

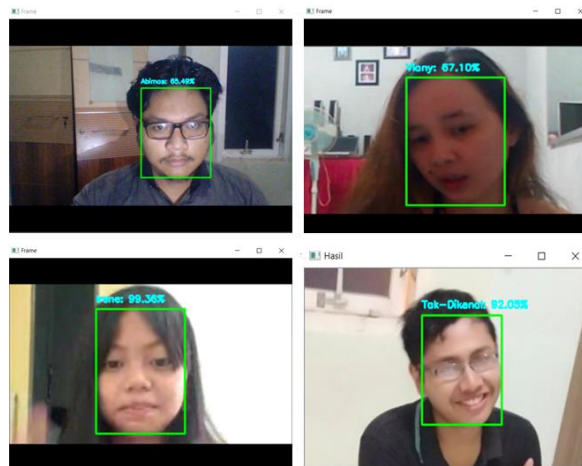
Dalam proses pengujian terdapat beberapa prosedur yang perlu dilakukan untuk melakukan pengambilan data. Adapun beberapa prosedur yang perlu dilakukan dalam proses pengujian akurasi *recognition* CNN terhadap objek adalah sebagai berikut:

1. Menjalankan program yang telah dibuat dengan urutan alur sistem seperti yang telah dijelaskan pada sub bab 4.1.2 terkait proses pengujian. Dimana alur dari poin 1 sampai dengan 5 tetap sama dan yang membedakan adalah tahapan sebagai berikut:
 - a. Mengisi tabel yang telah dibuat sesuai dengan hasil yang diperoleh selama pengujian dengan memberi centang (v) pada hasil pengujian apakah sesuai antara objek yang tampil pada kamera dengan nama yang muncul pada sistem dari hasil *recognition* CNN.
 - b. Jika terdapat ketidaksesuaian antara objek dengan hasil dari *recognition* program, maka isi nama yang tidak sesuai tersebut kedalam tabel keterangan.

4.2.3 Hasil Pengujian Akurasi *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 4.2 dibawah ini.



Gambar 4. 2 Hasil Akurasi *Recognition* CNN

Adapun proses pengujian akurasi *recognition* CNN terhadap objek ini dapat dilakukan dengan menggunakan tabel berikut :

Tabel 4. 2 Hasil pengujian akurasi *recognition* CNN terhadap objek

Uji ke-	Objek	Hasil Uji Deteksi		Keterangan	Nomer Screenshoot
		Berhasil	Gagal		
1	Anggota 1	V	-		a1
	Anggota 2	-	V	Terdapat 2 Viany	b1
	Anggota 3	-	V	Deteksi Abimas	c1
	Tak dikenal	V	-		d1
2	Anggota 1	V	-		a2
	Anggota 2	-	V	Deteksi Abimas	b2
	Anggota 3	-	V	Terdapat 2 Irene	c2
	Tak dikenal	V	-		d2
3	Anggota 1	V	-		a3
	Anggota 2	-	V	Terdapat 2 Viany	b3
	Anggota 3	V	-		c3
	Tak dikenal	V	-		d3
4	Anggota 1	V	-		a4
	Anggota 2	V	-		b4
	Anggota 3	V	-		c4
	Tak dikenal	V	-		d4
5	Anggota 1	V	-		a5
	Anggota 2	V	-		b5
	Anggota 3	V	-		c5
	Tak dikenal	V	-		d5

Uji ke-	Objek	Hasil Uji Deteksi		Keterangan	Nomer <i>Screenshoot</i>
		Berhasil	Gagal		
6	Anggota 1	V	-	Terdapat 2 Viany	a6
	Anggota 2	V	-		b6
	Anggota 3	V	-		c6
	Tak dikenal	V	-		d6
7	Anggota 1	V	-		a7
	Anggota 2	-	V		b7
	Anggota 3	V	-		c7
	Tak dikenal	V	-		d7
8	Anggota 1	V	-		a8
	Anggota 2	V	-		b8
	Anggota 3	V	-	c8	
	Tak dikenal	V	-	d8	
9	Anggota 1	V	-	a9	
	Anggota 2	V	-	b9	
	Anggota 3	V	-	c9	
	Tak dikenal	V	-	d9	
10	Anggota 1	V	-	a10	
	Anggota 2	V	-	b10	
	Anggota 3	V	-	c10	
	Tak dikenal	V	-	d10	

Dari hasil Tabel 4.2 diatas, dapat dijelaskan bahwa dari 40 data *sampel* uji coba terhadap 4 responden memiliki akurasi *recognition* terhadap objek dengan baik. Dimana sistem mendeteksi sebagai anggota x dan secara aktual memang benar anggota x disimbolkan dengan TP. Jika sistem mendeteksi sebagai tak dikenal / anggota y / z dan memang aktualnya tak dikenal / anggota y / z disimbolkan dengan TN. Dan sistem mendeteksi anggota x sedangkan aktualnya adalah tak dikenal / anggota y / z disimbolkan dengan FP. Sedangkan sistem mendeteksi tak dikenal / anggota y / z tetapi aktualnya adalah anggota x disimbolkan dengan FN. Sehingga dengan menggunakan rumus pengukuran performa dapat diperoleh sebuah presentase akurasi sebagai berikut:

Anggota 1 :

$$\begin{aligned}
 \text{Akurasi } recognition &= \frac{(TP + TN)}{TP + FP + FN + TN} \times 100\% & (4.2) \\
 &= \frac{(10 + 24)}{10 + 0 + 6 + 24} \times 100\% \\
 &= \frac{34}{40} \times 100\% \\
 &= 85\%
 \end{aligned}$$

Anggota 2 :

$$\begin{aligned}
 \text{Akurasi } recognition &= \frac{(TP + TN)}{TP + FP + FN + TN} \times 100\% & (4.3) \\
 &= \frac{(5 + 24)}{6 + 4 + 6 + 24} \times 100\% \\
 &= \frac{29}{40} \times 100\% \\
 &= 72.5\%
 \end{aligned}$$

Anggota 3 :

$$\begin{aligned}
 \text{Akurasi } recognition &= \frac{(TP + TN)}{TP + FP + FN + TN} \times 100\% & (4.4) \\
 &= \frac{(9 + 24)}{9 + 1 + 6 + 24} \times 100\% \\
 &= \frac{33}{40} \times 100\% \\
 &= 82.5\%
 \end{aligned}$$

Tak dikenal :

$$\begin{aligned}
 \text{Akurasi } recognition &= \frac{(TP + TN)}{TP + FP + FN + TN} \times 100\% & (4.5) \\
 &= \frac{(10 + 24)}{10 + 0 + 6 + 24} \times 100\% \\
 &= \frac{34}{40} \times 100\% \\
 &= 85\%
 \end{aligned}$$

4.3 Pengujian Presisi *Recognition* CNN Terhadap Objek

Dalam pengujian presisi *recognition* CNN terhadap objek terdapat beberapa hal yang harus kita ketahui dan pahami. Adapun beberapa hal tersebut yakni sebagai berikut.

4.3.1 Tujuan Pengujian Presisi *Recognition* CNN Terhadap Objek

Pada 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 (Latasha, 2018).

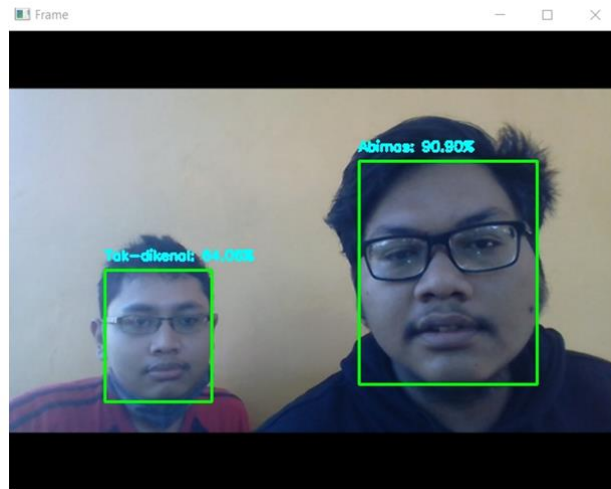
4.3.2 Proses Pengujian Presisi *Recognition* CNN Terhadap Objek

Dalam proses pengujian terdapat beberapa prosedur yang perlu dilakukan untuk melakukan pengambilan data. Adapun beberapa prosedur yang perlu dilakukan dalam proses pengujian deteksi terhadap objek adalah sebagai berikut:

1. Menjalankan program yang telah dibuat dengan urutan alur sistem seperti yang telah dijelaskan pada sub bab 4.1.2 terkait proses pengujian. Dimana alur dari poin 1 sampai dengan 5 tetap sama dan yang membedakan adalah tahapan sebagai berikut:
 - a. Mengisi tabel yang telah dibuat sesuai dengan hasil yang diperoleh selama pengujian dengan memberi centang (v) pada hasil pengujian. Dimana pada pengujian ini indikasi untuk keberhasilan program apabila dalam durasi 1 menit hasil sesuai antara objek yang diujikan dengan nama yang muncul pada sistem.
 - b. Mengisi tabel keterangan dengan memasukkan nama yang muncul dari sistem memiliki tingkat probabilitas/tingkat kemunculan nama yang berbeda dengan objek tinggi.

4.3.3 Hasil Pengujian Presisi *Recognition* CNN Terhadap Objek

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 4.3 dibawah ini.



Gambar 4. 3 Hasil Presisi *Recognition* CNN

Adapun proses pengujian presisi *recognition* CNN terhadap objek ini dapat dilakukan dengan menggunakan tabel berikut:

Tabel 4. 3 Hasil pengujian presisi *recognition* CNN terhadap objek

Uji ke-	Objek	Hasil Uji Deteksi		Keterangan	Nomer Screenshot
		Berhasil	Gagal		
1	Anggota 1	V	-		a1
	Anggota 2	V	-		b1
	Anggota 3	-	V	Deteksi Abimas	c1
	Tak dikenal	V	-		d1
2	Anggota 1	V	-		a2
	Anggota 2	-	V	Deteksi Abimas	b2
	Anggota 3	V	-		c2
	Tak dikenal	V	-		d2
3	Anggota 1	V	-		a3
	Anggota 2	V	-		b3
	Anggota 3	V	-		c3
	Tak dikenal	V	-		d3
4	Anggota 1	V	-		a4
	Anggota 2	V	-		b4
	Anggota 3	V	-		c4
	Tak dikenal	V	-		d4
5	Anggota 1	V	-		a5
	Anggota 2	V	-		b5
	Anggota 3	V	-		c5
	Tak dikenal	V	-		d5
6	Anggota 1	V	-		a6

Uji ke-	Objek	Hasil Uji Deteksi		Keterangan	Nomer Screenshot
		Berhasil	Gagal		
	Anggota 2	V	-		b6
	Anggota 3	V	-		c6
	Tak dikenal	V	-		d6
	Anggota 1	V	-		a7
7	Anggota 2	V	-		b7
	Anggota 3	V	-		c7
	Tak dikenal	V	-		d7
	Anggota 1	V	-		a8
8	Anggota 2	V	-		b8
	Anggota 3	V	-		c8
	Tak dikenal	V	-		d8
	Anggota 1	V	-		a9
9	Anggota 2	V	-		b9
	Anggota 3	V	-		c9
	Tak dikenal	V	-		d9
	Anggota 1	V	-		a10
10	Anggota 2	V	-		b10
	Anggota 3	V	-		c10
	Tak dikenal	V	-		d10

Dari hasil Tabel 4.3 diatas, dapat dijelaskan bahwa dari 40 data *sampel* uji coba terhadap 4 responden memiliki presisi *recognition* terhadap objek dengan baik. Dimana sistem mendeteksi sebagai anggota x dan secara aktual memang benar anggota x disimbolkan dengan TP. Jika sistem mendeteksi sebagai tak dikenal / anggota y / z dan memang aktualnya tak dikenal / anggota y / z disimbolkan dengan TN. Dan sistem mendeteksi anggota x sedangkan aktualnya adalah tak dikenal / anggota y / z disimbolkan dengan FP. Sedangkan sistem mendeteksi tak dikenal / anggota y / z tetapi aktualnya adalah anggota x disimbolkan dengan FN. Sehingga dengan menggunakan rumus pengukuran performa dapat diperoleh sebuah presentase presisi sebagai berikut:

Anggota 1:

$$\begin{aligned}
 \text{Presisi } \textit{recognition} &= \frac{TP}{TP+FP} \times 100\% && (4.6) \\
 &= \frac{10}{10+0} \times 100\% \\
 &= \frac{10}{10} \times 100\% \\
 &= 100\%
 \end{aligned}$$

Anggota 2:

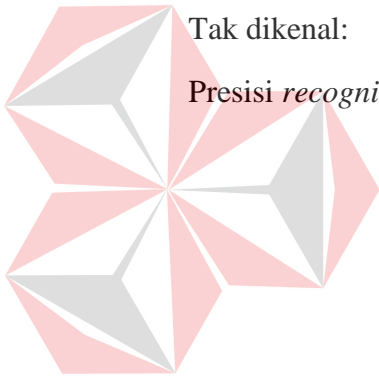
$$\begin{aligned}
 \text{Presisi } recognition &= \frac{TP}{TP+FP} \times 100\% & (4.7) \\
 &= \frac{9}{3+1} \times 100\% \\
 &= \frac{10}{10} \times 100\% \\
 &= 90\%
 \end{aligned}$$

Anggota 3:

$$\begin{aligned}
 \text{Presisi } recognition &= \frac{TP}{TP+FP} \times 100\% & (4.8) \\
 &= \frac{9}{9+1} \times 100\% \\
 &= \frac{10}{10} \times 100\% \\
 &= 90\%
 \end{aligned}$$

Tak dikenal:

$$\begin{aligned}
 \text{Presisi } recognition &= \frac{TP}{TP+FP} \times 100\% & (4.9) \\
 &= \frac{10}{10+0} \times 100\% \\
 &= \frac{10}{10} \times 100\% \\
 &= 100\%
 \end{aligned}$$



UNIVERSITAS
Dinamika

4.4 Pengujian *Recall Recognition* CNN Terhadap Objek

Dalam pengujian *recall recognition* CNN terhadap objek terdapat beberapa hal yang harus kita ketahui dan pahami. Adapun beberapa hal tersebut yakni sebagai berikut.

4.4.1 Tujuan Pengujian *Recall Recognition* CNN Terhadap Objek

Pengujian *recall recognition* CNN terhadap objek ini dilakukan untuk melihat tingkat keberhasilan dari CNN dalam menemukan kembali sebuah informasi terhadap data yang tidak berhasil teridentifikasi (Latasha, 2018).

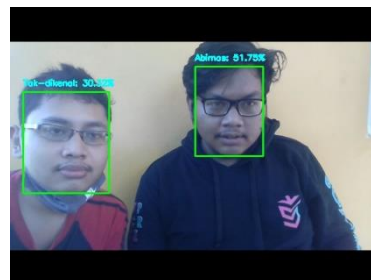
4.4.2 Proses Pengujian *Recall Recognition* CNN Terhadap Objek

Dalam proses pengujian terdapat beberapa prosedur yang perlu dilakukan untuk melakukan pengambilan data. Adapun beberapa prosedur yang perlu dilakukan dalam proses pengujian deteksi terhadap objek adalah sebagai berikut:

1. Menjalankan program yang telah dibuat dengan urutan alur sistem seperti yang telah dijelaskan pada sub bab 4.1.2 terkait proses pengujian. Dimana alur dari poin 1 sampai dengan 5 tetap sama dan yang membedakan adalah tahapan sebagai berikut:
 - a. Mencentang (v) pada hasil uji *Auto Recall*, jika pada saat terdapat kesalahan pengenalan dapat langsung kembali benar/mengenal kembali antara objek dengan nama yang muncul pada sistem saat dilakukan sedikit pembenaran posisi wajah.
 - b. Lalu pada tabel keterangan mengisi dengan nama yang tidak sesuai dengan objek yang dideteksi, jika hasil dari pengujian adalah *No Recall*.

4.4.3 Hasil Pengujian *Recall Recognition* CNN Terhadap Objek

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 4.4 dibawah ini.



Gambar 4. 4 Hasil *recall recognition*

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

Tabel 4. 4 Hasil pengujian *recall recognition* CNN terhadap objek

Uji ke-	Objek	Hasil Uji Deteksi		Keterangan	Nomer Screenshot
		Berhasil	Gagal		
1	Anggota 1	V	-		a1
	Anggota 2	V	-		b1
	Anggota 3	V	-		c1
	Tak dikenal	V	-		d1
2	Anggota 1	V	-		a2
	Anggota 2	V	-		b2
	Anggota 3	V	-		c2
	Tak dikenal	V	-		d2
3	Anggota 1	V	-		a3
	Anggota 2	V	-		b3
	Anggota 3	V	-		c3
	Tak dikenal	V	-		d3
4	Anggota 1	V	-		a4
	Anggota 2	V	-		b4
	Anggota 3	V	-		c4
	Tak dikenal	V	-		d4
5	Anggota 1	V	-		a5
	Anggota 2	V	-		b5
	Anggota 3	V	-		c5
	Tak dikenal	V	-		d5
6	Anggota 1	V	-		a6
	Anggota 2	V	-		b6
	Anggota 3	V	-		c6
	Tak dikenal	V	-		d6
7	Anggota 1	V	-		a7
	Anggota 2	V	-		b7
	Anggota 3	V	-		c7
	Tak dikenal	V	-		d7
8	Anggota 1	V	-		a8
	Anggota 2	V	-		b8
	Anggota 3	V	-		c8
	Tak dikenal	V	-		d8
9	Anggota 1	V	-		a9
	Anggota 2	V	-		b9
	Anggota 3	V	-		c9
	Tak dikenal	V	-		d9
10	Anggota 1	V	-		a10
	Anggota 2	V	-		b10
	Anggota 3	V	-		c10
	Tak dikenal	V	-		d10

Dari hasil Tabel 4.4 diatas, dapat dijelaskan bahwa dari 40 data *sampel* uji coba terhadap 4 responden memiliki *recall recognition* terhadap objek dengan baik. Dimana sistem mendeteksi sebagai anggota x dan secara aktual memang benar anggota x disimbolkan dengan TP. Jika sistem mendeteksi sebagai tak dikenal / anggota y / z dan memang aktualnya tak dikenal / anggota y / z disimbolkan dengan TN. Dan sistem mendeteksi anggota x sedangkan aktualnya adalah tak dikenal / anggota y / z disimbolkan dengan FP. Sedangkan sistem mendeteksi tak dikenal / anggota y / z tetapi aktualnya adalah anggota x disimbolkan dengan FN. Sehingga dengan menggunakan rumus pengukuran performa dapat diperoleh sebuah presentase *recall* sebagai berikut:

Anggota 1:

$$\begin{aligned} \text{Recall recognition} &= \frac{TP}{TP+FP} \times 100\% && (4.10) \\ &= \frac{10}{10+0} \times 100\% \\ &= 100\% \end{aligned}$$

Anggota 2:

$$\begin{aligned} \text{Recall recognition} &= \frac{TP}{TP+FP} \times 100\% && (4.11) \\ &= \frac{10}{10+0} \times 100\% \\ &= 100\% \end{aligned}$$

Anggota 3:

$$\begin{aligned} \text{Recall recognition} &= \frac{TP}{TP+FP} \times 100\% && (4.12) \\ &= \frac{10}{10+0} \times 100\% \\ &= 100\% \end{aligned}$$

Tak dikenal:

$$\begin{aligned} \text{Recall recognition} &= \frac{TP}{TP+FP} \times 100\% && (4.13) \\ &= \frac{10}{10+0} \times 100\% \\ &= 100\% \end{aligned}$$

4.5 Pengujian Pengiriman/*Upload* Data ke *Database*

Dalam Pengujian Pengiriman/*Upload* Data ke *Database* terdapat beberapa hal yang harus kita ketahui dan pahami. Adapun beberapa hal tersebut yakni sebagai berikut.

4.5.1 Tujuan Pengujian Pengiriman/*Upload* 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*.

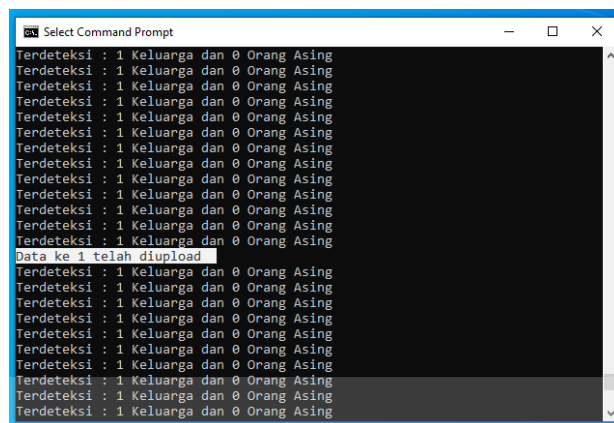
4.5.2 Proses Pengujian Pengiriman/*Upload* Data ke *Database*

Dalam proses pengujian terdapat beberapa prosedur yang perlu dilakukan untuk melakukan pengambilan data. Adapun beberapa prosedur yang perlu dilakukan dalam proses pengujian deteksi terhadap objek adalah sebagai berikut:

1. Menjalankan program yang telah dibuat dengan urutan alur sistem seperti yang telah dijelaskan pada sub bab 4.1.2 terkait proses pengujian. Dimana alur dari poin 1 sampai dengan 5 tetap sama dan yang membedakan adalah tahapan sebagai berikut:
 - a. Untuk mempermudah proses pengolahan data maupun analisis, maka akan dilakukan *sreenshoot CMD* setiap 1 menit. Kemudian melakukan *rename* dari hasil *sreenshoot CMD* dengan nomor sesuai urutan pengambilan.
 - b. Kemudian memasukkan nomor kedalam tabel pada tab nomer *sreenshoot*.
 - c. Mengisi tabel yang telah dibuat sesuai dengan hasil yang diperoleh selama pengujian dengan memberi centang (v) pada hasil pengujian. Apakah data berhasil di *upload* atau tidak dengan melihat pada *notifikasi* running program CMD.

4.5.3 Hasil Pengujian Pengiriman/Upload 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. Untuk hasil dari pengujian pengiriman/upload data ke database dapat diamati pada Gambar 4.5 dibawah ini.



```

Select Command Prompt
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Data ke 1 telah diupload
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing
Terdeteksi : 1 Keluarga dan 0 Orang Asing

```

Gambar 4. 5 Hasil Pengiriman/Upload Data Ke Database

Adapun proses pengujian pengiriman/upload data ke database ini dapat dilakukan dengan menggunakan tabel berikut:

Tabel 4. 5 Hasil pengujian pengiriman/upload data ke database

Upload Data Ke-	Jenis Data	Hasil Upload		Screenshoot CMD
		Berhasil	Gagal	
1	<i>String</i>	V	-	cm1
	<i>Image</i>	V	-	cm1
2	<i>String</i>	V	-	cm2
	<i>Image</i>	V	-	cm2
3	<i>String</i>	V	-	cm3
	<i>Image</i>	V	-	cm3
4	<i>String</i>	V	-	cm4
	<i>Image</i>	V	-	cm4
5	<i>String</i>	V	-	cm5
	<i>Image</i>	V	-	cm5
6	<i>String</i>	V	-	cm6
	<i>Image</i>	V	-	cm6
7	<i>String</i>	V	-	cm7
	<i>Image</i>	V	-	cm7
8	<i>String</i>	V	-	cm8
	<i>Image</i>	V	-	cm8

Upload Data Ke-	Jenis Data	Hasil Upload		Screenshoot CMD
		Berhasil	Gagal	
9	<i>String</i>	V	-	cm9
	<i>Image</i>	V	-	cm9
10	<i>String</i>	V	-	cm10
	<i>Image</i>	V	-	cm10

Dari hasil Tabel 4.5 diatas, dapat dijelaskan bahwa dari 20 data *sampel* uji coba terhadap 2 data yang dikirim dapat di *upload* dengan baik, sehingga persentase data dapat dihitung sebagai berikut:

$$\begin{aligned}
 \text{Akurasi pengiriman data} &= \frac{(\text{banyak pengujian} - \text{jumlah gagal uji})}{\text{jumlah seluruh data}} \times 100\% \quad (4.14) \\
 &= \frac{(20-0)}{20} \times 100\% \\
 &= \frac{20}{20} \times 100\% \\
 &= 100\%
 \end{aligned}$$

4.6 Pengujian Penerimaan *Notifikasi Android*

Dalam pengujian deteksi terhadap objek terdapat beberapa hal yang harus kita ketahui dan pahami. Adapun beberapa hal tersebut yakni sebagai berikut.

4.6.1 Tujuan Pengujian Penerimaan *Notifikasi Android*

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 belum terselesaikan.

4.6.2 Proses Pengujian Penerimaan *Notifikasi Android*

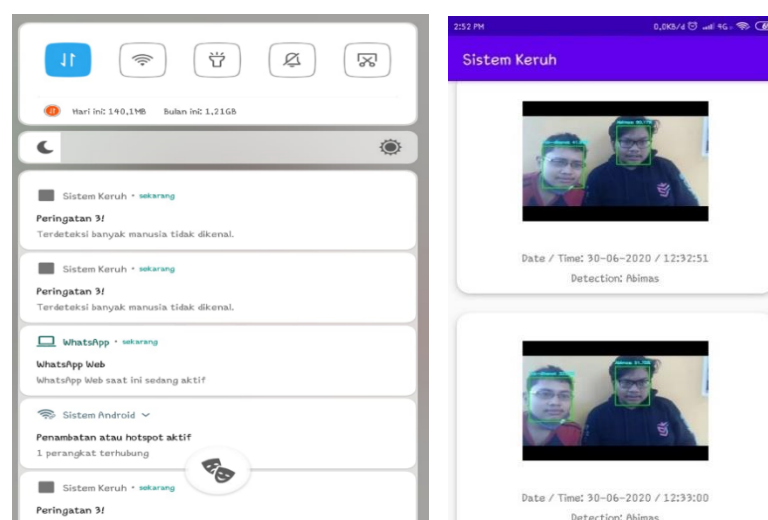
Dalam proses pengujian terdapat beberapa prosedur yang perlu dilakukan untuk melakukan pengambilan data. Adapun beberapa prosedur yang perlu dilakukan dalam proses pengujian deteksi terhadap objek adalah sebagai berikut:

1. Menjalankan program yang telah dibuat dengan urutan alur sistem seperti yang telah dijelaskan pada sub bab 4.1.2 terkait proses pengujian. Dimana alur dari poin 1 sampai dengan 5 tetap sama dan yang membedakan adalah tahapan sebagai berikut:
 - a. Untuk mempermudah proses pengolahan data maupun analisis, maka akan dilakukan *sreenshoot notifikasi* yang muncul pada *Android*. Kemudian melakukan *rename* dari hasil *sreenshoot notifikasi* dengan nomor sesuai urutan pengambilan.
 - b. Kemudian memasukkan nomor kedalam tabel pada tab nomer *sreenshoot*.
 - c. Mengisi tabel yang telah dibuat sesuai dengan hasil yang diperoleh selama pengujian dengan memberi centang (v) pada hasil pengujian. Dengan identifikasi keberhasilan adalah apabila *notifikasi* muncul pada perangkat *Android*.

4.6.3 Hasil Pengujian Penerimaan Notifikasi Android

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 belum terselesaikan.

Untuk hasil dari pengujian penerimaan *notifikasi Android* dapat diamati pada Gambar 4.6 dibawah ini.

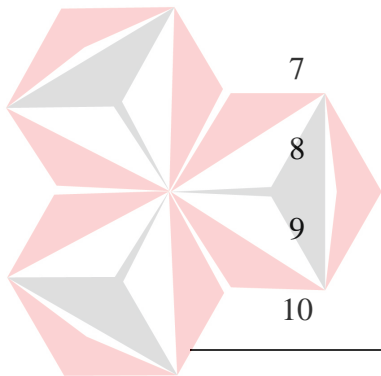


Gambar 4. 6 Hasil Penerimaan *Notifikasi Android*

Adapun proses pengujian penerimaan *notifikasi Android* ini dapat dilakukan dengan menggunakan tabel berikut:

Tabel 4. 6 Hasil pengujian penerimaan *notifikasi android*

Notifikasi Data Ke-	Jenis Data	Hasil Notifikasi		Screenshoot Notifikasi
		Masuk	Gagal	
1	String	V	-	nots1
	Image	V	-	nots1
2	String	V	-	nots2
	Image	V	-	nots2
3	String	V	-	nots3
	Image	V	-	nots3
4	String	V	-	nots4
	Image	V	-	nots4
5	String	V	-	nots5
	Image	V	-	nots5
6	String	-	V	
	Image	-	V	
7	String	V	-	nots7
	Image	V	-	nots7
8	String	V	-	nots8
	Image	V	-	nots8
9	String	V	-	nots9
	Image	V	-	nots9
10	String	V	-	nots10
	Image	V	-	nots10

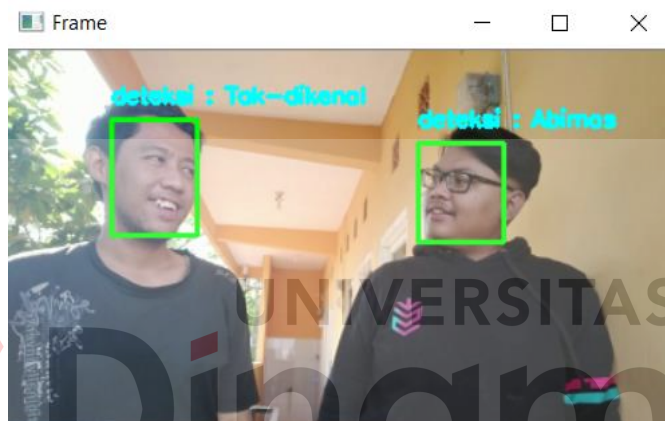


Dari hasil Tabel 4.6 diatas, dapat dijelaskan bahwa dari 20 data *sampel uji* coba terhadap 2 data dapat menerima *notifikasi* dengan baik, sehingga persentase data dapat dihitung sebagai berikut:

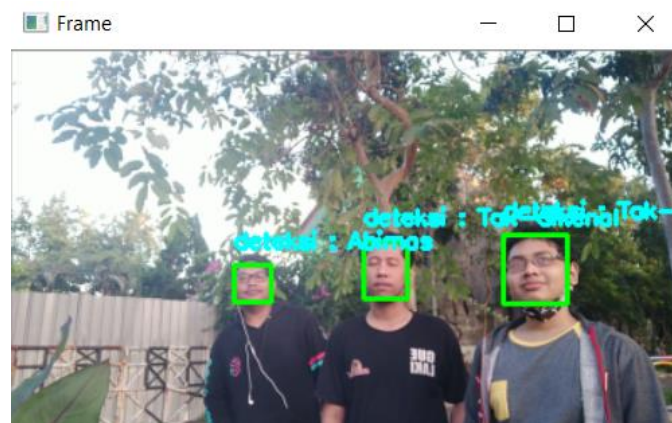
$$\begin{aligned}
 \text{Akurasi penerimaan data} &= \frac{(\text{banyak pengujian} - \text{jumlah gagal uji})}{\text{jumlah seluruh data}} \times 100\% \quad (4.15) \\
 &= \frac{(20-2)}{20} \times 100\% \\
 &= \frac{18}{20} \times 100\% \\
 &= 90\%
 \end{aligned}$$

4.6.4 Hasil Pengujian Klarifikasi Peringatan Pada *Notifikasi*

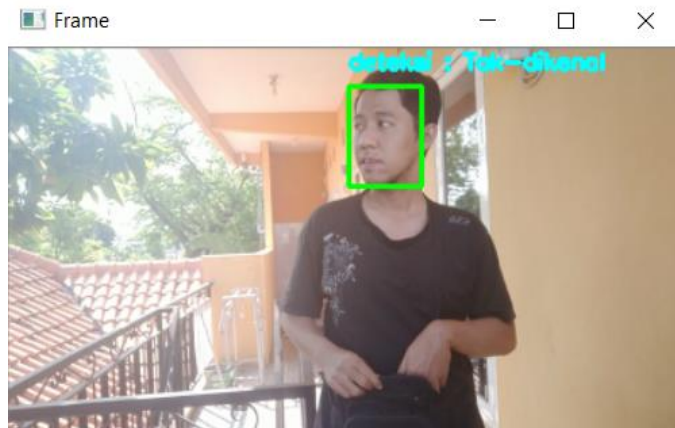
Pengujian ini bertujuan untuk melakukan uji peringatan pada notifikasi android. Dimana pada sistem notifikasi terdapat 3 jenis peringatan yang akan muncul didalam bar notifikasi android yakni, bahaya 1 jika terdapat satu tak dikenali dan satu atau lebih anggota yang dikenali, dan bahaya 2 jika terdapat 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 4. 7 Kondisi Bahaya 1



Gambar 4. 8 Kondisi Bahaya 2

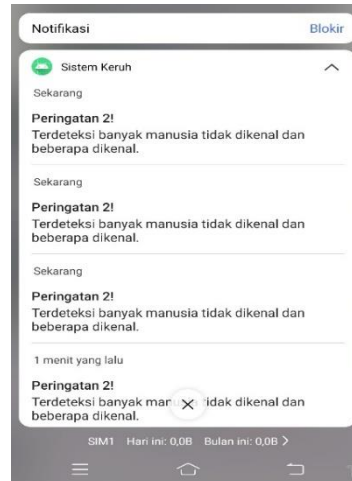


Gambar 4. 9 Kondisi Bahaya 3

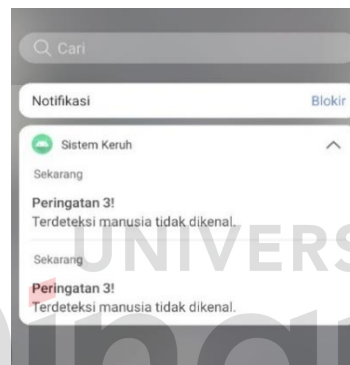
Adapun tampilan dari sistem keamanan rumah saat melakukan pengenalan wajah dapat diamati pada Gambar 4.7 yang merupakan sebuah kondisi dari bahaya 1 dimana terdapat satu anggota dikenal dan satu orang tak dikenal. Dapat diamati pula pada Gambar 4.8 yang merupakan sebuah kondisi dari bahaya 2 dimana terdapat satu anggota dikenal dan dua orang tak dikenal. Serta pada Gambar 4.9 yang merupakan kondisi bahaya 3 dimana terdapat satu orang tak dikenal dan tidak terdapat anggota yang dikenal.



Gambar 4. 10 Peringatan Notifikasi Bahaya 1



Gambar 4. 11 Peringatan Notifikasi Bahaya 2



Gambar 4. 12 Peringatan Notifikasi Bahaya 3

Ketika sebuah kondisi masuk kedalam range pengklarifikasian tingkat bahaya seperti halnya yang sudah dijelaskan sebelumnya, maka secara otomatis sistem akan mengirimkan sebuah peringatan berupa notifikasi sesuai dengan tingkat bahayanya. Dapat diamati pada Gambar 4.10 yang merupakan peringatan untuk bahaya 1. Pada Gambar 4.11 sebuah peringatan untuk bahaya 2, dan pada Gambar 4.12 untuk peringatan bahaya 3. Dengan adanya pengklarifikasian tersebut diharapkan *user* nantinya dapat mengambil sebuah tindakan dalam melakukan pengaman terhadap rumah sesuai dengan tingkat bahaya yang terdeteksi. Adapun tabel pengujian terhadap pengklarifikasian tingkat bahaya tersebut dapat diamati pada Tabel 4.7 berikut.

Tabel 4. 7 Hasil pengujian klarifikasi peringatan pada *notifikasi*

Uji Ke	Kondisi Aktual	Kondisi Hasil Deteksi		Peringatan Yang Tampil			Keberhasilan	
		Dikenal	Tak Dikenal	Bahaya 1	Bahaya 2	Bahaya 3	Sesuai	Gagal
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 4.7 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.

Dari hasil Tabel 4.7 diatas juga dapat dijelaskan bahwa dari 30 data pengujian dapat melakukan klarifikasi pemberian peringatan dengan baik, sehingga persentase data dapat dihitung sebagai berikut:

$$\begin{aligned}\text{Kesesuaian Peringatan} &= \frac{\text{Jumlah Keberhasilan pemberitahuan}}{\text{jumlah seluruh data}} \times 100\% && \text{(4.16)} \\ &= \frac{27}{30} \times 100\% \\ &= 90\%\end{aligned}$$



UNIVERSITAS
Dinamika

BAB V PENUTUP

5.1 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 presentase keberhasilan dalam mengunggah data berupa data gambar dan *string* sebesar 100% dan penerimaan *notifikasi* yang cukup baik dengan presentase keberhasilan 90% dengan berdasarkan pada 20 kali pengambilan data.
4. Wajah akan terdeteksi dan dikenali oleh program saat berada dalam radius 3 meter dari kamera. Hal ini disebabkan karena sistem menggunakan kamera dengan resolusi 5 MP.

5.2 Saran

Pada pengembangan yang selanjutnya sistem ini juga dapat diintegrasikan dengan IoT pada rumah, bilamana terdeteksi anggota keluarga yang akan 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

- A., D. (2016, Juli 20). *A Beginner's Guide To Understanding Convolutional Neural Networks*. Retrieved from Artificial Net: <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>
- 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.