

# RANCANG BANGUN PENDETEKSI MASKER DAN ALAT PEMBERSIH TANGAN OTOMATIS BERBASIS ARDUINO MENGGUNAKAN METODE CNN (CONVOLUTION NEURAL NETWORK) DI GEREJA GKI JAYAPURA PAPUA

Raimond Obet Yumame, Yosep Agus Pranoto, Joseph Dedy Irawan

Program Studi Teknik Informatika S1, Fakultas Teknologi Industri  
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia  
1818126@scholar.itn.ac.id

## ABSTRAK

Virus corona masuk ke negara Indonesia pada awal Maret 2020. Pemerintah Indonesia menerapkan kebijakan pemberlakuan pembatasan kegiatan masyarakat. diantaranya yaitu menjaga jarak, selalu menggunakan masker, mencuci tangan atau membersihkan tangan sehabis pulang berpergian. Namun, penggunaan bersama *hand sanitizer* serta masyarakat yang tidak menggunakan masker di Gereja GKI Jayapura berpotensi dalam penularan virus Corona. Berdasarkan permasalahan di atas, peneliti membuat sebuah alat otomatis pengeluaran *hand sanitizer* dari kemasan dan pendeteksi masker untuk membantu jemaat dalam membersihkan tangan tanpa perlu menekan botol *hand sanitizer* secara manual alat ini di buat menggunakan metode *Convolutional Neural Network* berbasis Arduino. Pengujian melibatkan 10 orang relawan dengan melakukan *class* berbeda menggunakan masker dan tidak menggunakan masker, uji coba sebanyak 20 kali menggunakan model dari hasil *training*. Hasil pengujian testing menghasilkan akurasi 90%.

**Kata kunci:** *Arduino, Covid-19, Convolutional Neural Network, Deep learning, Pembersih tangan, Masker.*

## 1. PENDAHULUAN

Virus corona terjadi pada akhir tahun 2019. Di Wuhan, banyak orang yang terkena virus ini sampai ada yang meninggal. Virus ini menular dengan cepat pada beberapa bulan belakangan ini di puluhan negara termasuk negara Indonesia pada awal Maret 2020 [1].

Pemerintah Indonesia menerapkan kebijakan Pemberlakuan Pembatasan Kegiatan Masyarakat. Dengan beberapa peraturan yaitu menjaga jarak, selalu menggunakan masker, mencuci tangan atau menggunakan *hand sanitizer* sehabis pulang berpergian dll. *Hand Sanitizer* merupakan cairan pembersih tangan berupa gel yang memiliki fungsi untuk membersihkan tangan dari bakteri bakteri [1]. Selain penggunaan *hand sanitizer*, penggunaan masker merupakan hal yang wajib dilaksanakan karena saat ini masih banyak orang yang tidak melaksanakan anjuran pemerintah dan WHO terkait penggunaan masker guna mengurangi dampak penyebaran COVID-19 [2].

Namun, penggunaan bersama *hand sanitizer* serta masyarakat yang tidak menggunakan masker di Gereja GKI Jayapura dapat berpotensi dalam penularan virus Corona. Potensi ini terjadi karena pengeluaran cairan *hand sanitizer* dari kemasan dengan menekan botol *hand sanitizer* serta pengecekan masker secara manual dapat menimbulkan kerumunan jemaat yang hendak mengantri untuk menggunakan pembersih tangan.

Berdasarkan permasalahan di atas, maka peneliti berinisiatif untuk membuat suatu alat otomatis pengeluaran *hand sanitizer* dari dalam kemasan dan pendeteksi masker menggunakan metode *Convolutional Neural Network* berbasis Arduino. Metode *Convolutional Neural Network* adalah pengembangan dari bentuk lain *Deep Learning* yang digunakan untuk menerima data berupa gambar dan

belajar mengenal gambar untuk mengklasifikasi suatu objek citra menjadi nilai-nilai tertentu, semakin banyak data yang di *training* dalam algoritma metode ini semakin baik nilai akurat yang di hasilkan. Tujuan dari pembuatan alat ini yaitu untuk mengurangi interaksi langsung yang dilakukan oleh pengguna pada botol *hand sanitizer* dan petugas yang melakukan pengecekan masker pada jemaat gereja. Alat ini akan digunakan di Gereja GKI Harapan Jayapura.

## 2. TINJAUAN PUSTAKA

### 2.1. Penelitian Terkait

Berdasarkan penelitian yang dilakukan oleh Purnama dan Negara pada tahun 2021 yang berjudul “Deteksi Masker Pencegahan Covid-19 Menggunakan *Convolutional Neural Network* Berbasis Android”. Pada penelitian ini metode deteksi yang digunakan adalah *convolutional neural network* (CNN), sehingga hasil yang diperoleh adalah nilai akurasi sebesar 90 % [3].

Penelitian yang dilakukan oleh Anarki ddk pada tahun 2021 yang berjudul “Penerapan Metode *Haar Cascade* Pada Aplikasi Deteksi Masker”. Pada penelitian ini metode deteksi yang digunakan yaitu *Haar Cascade*, dan hasil dari penelitian ini adalah nilai keakuratan yang tertinggi adalah 88,7% dan juga dalam proses pendeteksian dari fitur tambahan berupa audio dan memotret berjalan dengan baik [4].

Penelitian yang dilakukan oleh Prasetya ddk pada tahun 2021 yang berjudul “Rancang Bangun Pendeteksi Wajah Bermasker Dan Tidak Bermasker Dalam Absensi Di Masa Pandemi COVID-19 Menggunakan *Convolutional Neural Network*”. Hasil pada penelitian ini memiliki nilai akurasi 95% untuk proses *training* dan untuk *testing* sebesar 91% dalam

mengklasifikasikan wajah yang menggunakan masker dan tidak menggunakan masker [5].

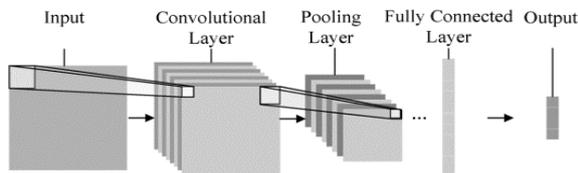
**2.2. Deep learning**

Deep learning merupakan metode yang sering kali di gunakan untuk mengklasifikasikan sebuah objek menggunakan teknik jaringan syaraf tiruan untuk permasalahan dengan dataset yang besar.

Metode deep learning bagus dalam menemukan fitur yang baik pada citra ke lapisan berikutnya untuk membentuk hipotesis nonlinier yang dapat meningkatkan kekompleksitasan sebuah model. Model yang kompleks nantinya akan memakan waktu pelatihan yang cukup lama oleh sebab itu di dunia deep learning penggunaan GPU sangatlah umum [8].

**2.3. Convolutional Neural Network**

Metode CNN sering digunakan untuk data berupa gambar atau data dengan jumlah yang banyak dimensi. Convolutional Neural Network (CNN) dapat digunakan untuk mendeteksi dan mengenali objek dalam, gambar. Convolutional Neural Network (CNN) adalah teknologi yang terinspirasi dari perilaku mamalia. Secara umum convolutional neural network menggunakan proses konvolusi dengan: memindahkan kernel (filter) konvolusi dengan ukuran tertentu ke gambar. Komputer menerima informasi representatif baru dari hasil filter yang di gunakan pada bagian gambar tersebut. Proses Convolution Neural Network (CNN) bisa di lihat pada gambar 1

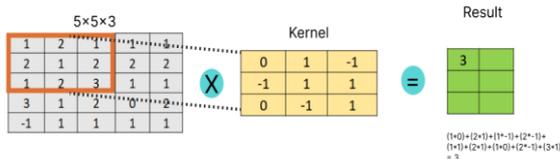


Gambar 1. Arsitektur Convolutional Neural Network

Gambar 1. Merupakan gambaran umum arsitektur metode Convolutional Neural Network yang terdiri atas input, feature extraction dan classification. Untuk feature extraction ada beberapa proses antara lain: [4]

a. Convolutional layer

Di konvolusi ini dilaksanakan proses ekstraksi fitur pada citra dengan melakukan konvolusi antara kernel dengan sebuah citra. penggunaan banyak lapis layer dan kernel yang berbeda, maka akan diperoleh fitur di level tinggi seperti edge, curve dan fitur color. Beberapa proses yang di butuhkan dalam konvolusi diantaranya sebagai berikut:



Gambar 2. Perhitungan konvolusi

Proses ini merupakan perhitungan dimensi feature map yang dimana rumusnya sebagai berikut:

$$\text{Output f. maps} = \frac{W - N + 2P}{S} + 1 \dots \dots \dots (1)$$

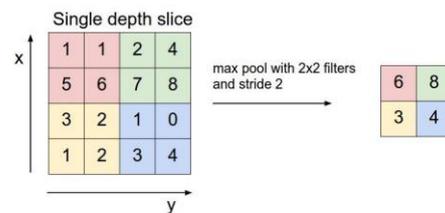
- Keterangan:  
 W = Ukuran Input  
 N = Ukuran Filter  
 P = Padding  
 S = Stride

b. Relu

(Rectification Linear Unit) ReLu yaitu proses memperkenalkan non-linearitas dan meningkatkan representasi model. Nilai output neuron bernilai 0 jika nilai input negatif. Jika neuron tersebut bernilai input positif adalah nilai input trigger itu sendiri. ReLu juga sering kali digunakan karena dapat melatih neural network lebih cepat.

c. Maxpooling layer

Maxpooling layer ini berfungsi untuk mengambil nilai rata-rata atau nilai yang tertinggi menjadi citra yang baru tujuannya yaitu mengurangi jumlah paramet dan komputasi [6].



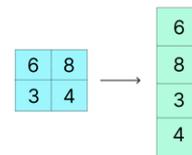
Gambar 3. Ilustrasi maxpooling

d. Classification

Classification merupakan proses untuk melakukan klasifikasi terhadap setiap neuron yang telah di ekstraksi pada proses feature learning. Dalam classification ada beberapa tahapan di antaranya:

1. Flatten

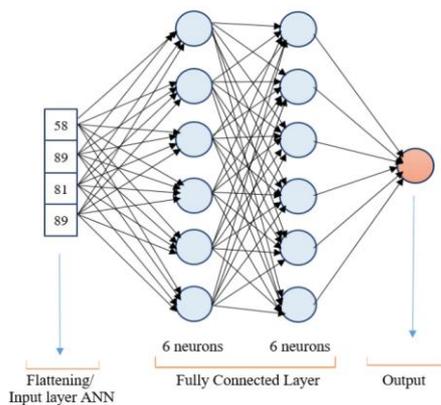
flattening berfungsi untuk mengubah matriks dua dimensi menjadi satu dimensi atau satu kolom. vektor ini yang akan menjadi layer di lapisan fully connected layer.



Gambar 4. Ilustrasi flatten

2. Full Connected Layer

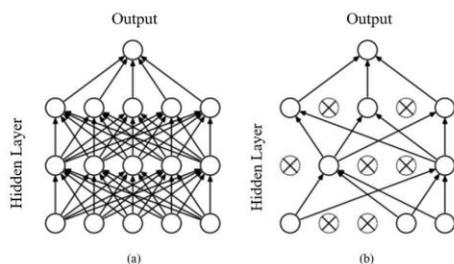
hasil dari proses flattening kemudian tahap berikutnya yaitu vector ini akan di proses dalam teknik artificial neural networks. Proses ANN di bagi beberapa bagian, yaitu input layer, hidden layer, dan output layer.



Gambar 5. Ilustrasi full connected layer

3. Dropout regularization

Dropout regularization, Merupakan proses regulasi pada sebuah jaringan syaraf tiruan dimana beberapa neuron akan dipilih secara acak atau random dan tidak digunakan pada proses training.



Gambar 6. Ilustrasi Dropout regularization

4. Softmax

Fungsi softmax dalam proses klasifikasi adalah menghitung nilai probabilitas seluruh label untuk semua kelas. Biasanya rentang probabilitas output pada softmax adalah nilai dari 0 hingga 1. Nilai softmax dapat di hitung menggunakan persamaan:

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)} \dots\dots\dots(2)$$

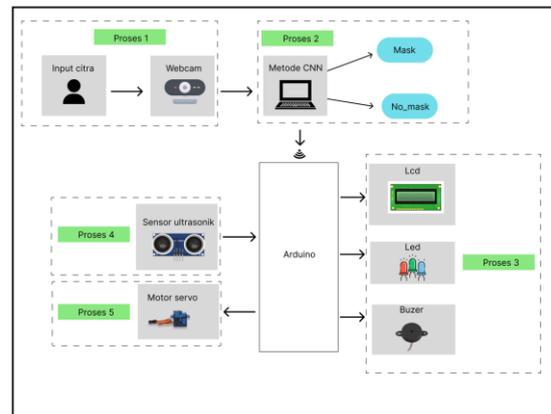
2.4. Arduino

Arduino adalah mikrokontroler / pengendali mikro papan tunggal yang bersifat sumber terbuka dan menjadi salah satu proyek Open Source Hardware yang paling populer. Dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang

3. METODE PENELITIAN

3.1. Analisis Sistem

Sistem pendeteksi masker akan melakukan penambahan data otomatis berupa citra digital sebagai dataset yang akan di gunakan untuk proses training dan akan menghasilkan model .h5 untuk di gunakan sebagai klasifikasi label dari hasil deteksi wajah yang bermasker dan tidak bermasker. Desain blog diagram system di tunjukan pada gambar 7.

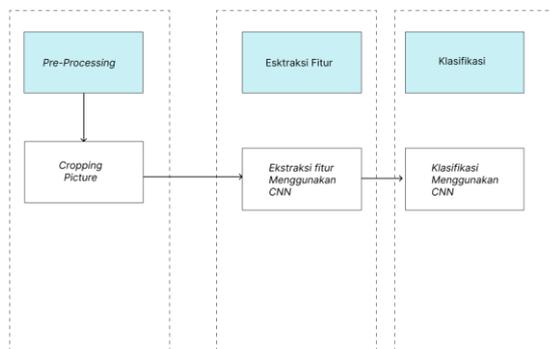


Gambar 7. Desain blog diagram sistem

1. Proses 1  
Memperlihatkan sistem rancang bangun alat pendeteksi masker dan pembersih tangan berbasis Arduino menggunakan metode Convolutional Neural Network dimana jemaat yang akan memasuki gedung gereja melakukan proses scan wajah yang bermasker atau tidak bermasker ke webcam.
2. Proses 2  
Selanjutnya di lakukan klasifikasi apakah input citra atau jemaat yang datang itu menggunakan masker atau tidak menggunakan masker. Proses ini akan di jelaskan pada desain sistem CNN.
3. Proses 3  
Pada proses ini hasil klasifikasi CNN akan mendapatkan serial data berupa nilai 0 dan 1 jika terdeteksi menggunakan masker maka nilai serial datanya adalah 1 sehingga signal yang di kirim ke Arduino melalui nodemcu akan menyalakan LED akan berwarna hijau dan LCD running text akan memberikan informasi “Silahkan Masuk”, jika terdeteksi tidak menggunakan masker maka serial datanya adalah 0 sehingga LED akan menyala berwarna merah dan LCD running text akan memberikan informasi “Di larang masuk, Gunakanlah masker”.
4. Proses 4  
Setelah jemaat gereja memenuhi persyaratan yang menggunakan masker maka langkah selanjutnya adalah jemaat akan membersihkan tangan secara otomatis dimana jemaat akan mengarahkan tangan ke alat atau yang akan di baca oleh sensor ultrasonic untuk mendeteksi objek yang kurang dari 10cm Ketika memenuhi jarak kurang dari 10cm maka sensor ultrasonic akan mengirim signal ke Arduino untuk di lanjutkan ke proses 5.
5. Proses 5  
Setelah mendapatkan signal dari sensor ultrasonic arduino akan menggerakkan servo untuk menekan atau berputar 90 derajat untuk menekan cairan pada botol handsanitizer.

### 3.2. Desain Sistem CNN

Desain sistem pengenalan ekspresi wajah dalam penelitian ditunjukkan pada gambar 8.



Gambar 8. Desain Sistem Pemindaian Wajah masker dan tidak menggunakan masker.

Pada penelitian ini akan, dilakukan pengamatan, dari proses input citra Masker diantaranya adalah, citra yang menggunakan masker berwarna-warni. Kemudian citra yang di input akan, melalui beberapa tahapan. yaitu proses *preprocessing*, ekstraksi fitur, dan klasifikasi masker menggunakan *Convolutional Neural Networks*. Di dalam proses *preprocessing* akan di lakukan proses *cropping picture* atau memotong yang di perlukan saja.

Gambar ini didapat dari *Dataset*. *Dataset* merupakan akses terbuka dari penelitian serupa sebelumnya yang sample datanya di ambil dari web resmi *Kaggle.com* terdiri dari 2 label masker yang di ambil dari *Dataset* yaitu yang menggunakan masker berisi 974 sampel, yang tidak menggunakan masker berisi 817sampel setelah proses

Kemudian di lanjutkan dengan proses Ekstraksi fitur menggunakan *cnn* dimana ada proses *Konvolusi*, *relu*, *max pooling*. Setelah mendapatkan hasil dari ekstraksi fitur proses akhir yaitu klasifikasi menggunakan *CNN*.

### 3.3. Preprocessing

*Flowchart Preprocessing* di tunjukan pada gambar 9.



Gambar 9. Flowchart Preprocessing

Tahapan ini merupakan tahapan *pre-processing* Pertama diawali dengan input citra *datasatet* yang menggunakan maskser dan tidak menggunakan masker. selanjutnya akan di lakukan *Cropping gambar* atau mengambil area citra yang bagian wajah kemudian mendapatkan hasil *cropping*.

### 3.4. Ekstraksi Fitur *Convulutinal Neural Network*

Setelah proses *preprocessing* selesai, maka tahapan selanjutnya adalah ekstraksi fitur. Ekstraksi fitur dilakukan untuk pengambilan ciri atau fitur dari citra yang nantinya akan menghasilkan sebuah-nilai berupa matriks kemudian disimpan di database lalu dianalisis untuk proses klasifikasi deteksi masker. Proses ekstraksi pada *CNN* dilakukan dengan tahapan seperti pada gambar 10.



Gambar 10. Ekstraksi fitur

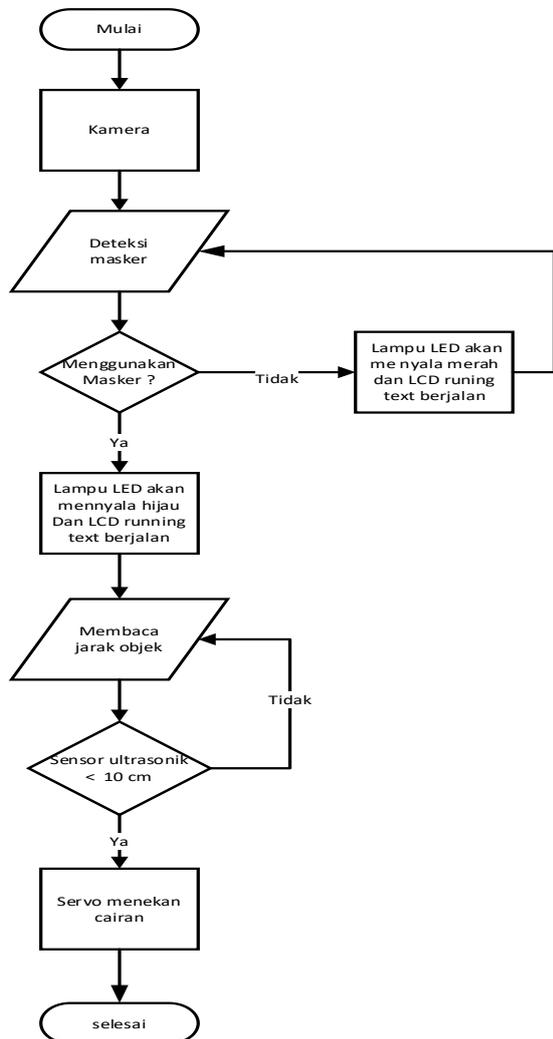
Hasil dari *preprocessing* di lanjutkan dengan proses *convolution* yang dimana nilai pixelnya akan di hitung menggunakan kernel 3x3 setelah mendapatkan hasil akan di lanjutkan dengan proses *pooling layer* dimana akan di ambil nilai terbesar. Selanjutn dikakukan untuk mengurangi *overfitting* pada citra dan *dropout layer* dilanjutkan dengan proses *dense layer* untuk klasifikasi.

### 3.5. Klasifikasi

Dalam penelitian ini, klasifikasi yang digunakan adalah metode *CNN (Convulutional Neural Network)* sebagai klasifikasi label masker, dimana hasil klasifikasi di dapatkan karena adanya pembentukan model dari proses *training*.

### 3.6. Flowchart Alur Sistem Alat

Flowchart alur sistem di tunjukan pada gambar 11.

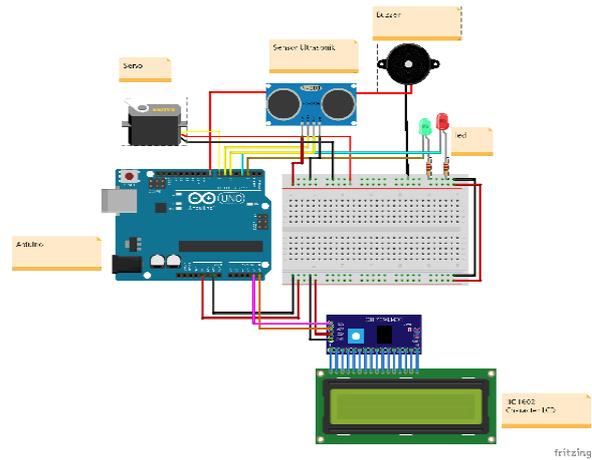


Gambar 11. Flowchart Alur sistem

Proses jalanya sistem dimulai dengan jemaat menginput wajah di kamera dan sistem akan mendeteksi wajah yang bermasker atau tidak menggunakan masker jika terdeteksi memakai masker maka lampu led akan menyala berwarna hijau dan lcd akan menampilkan text terimakasih telah menggunakan masker jika tidak menggunakan masker maka lampu led akan menyala berwarna merah dan lcd akan menampilkan text harap menggunakan masker. Kemudian proses selanjutnya jemaat akan mengulurkan tangan ke sensor *ultrasonic* untuk membaca pergerakan Ketika kurang dari 10cm maka servo akan menekan botol *handsanitizer* untuk mengeluarkan cairan secara otomatis.

### 3.7. Prototype Alat Pembersih Tangan

Desain alat pembersih tangan pada gambar 12:



Gambar 12. Desain Prototype Alat pembersih tangan

Pada gambar 12. Merupakan gambar desain *prototype* yang menggunakan Arduino sebagai mikrokontroler, 2 led berwarna hijau dan merah, 1 lcd 16x2, sensor ultrasonik sebagai pembaca suatu jarak atau gerakan, servo sebagai penekan cairan *handsanitizer*, dan satu buah *buzzer*.

### 3.8. Model Arsitektur CNN

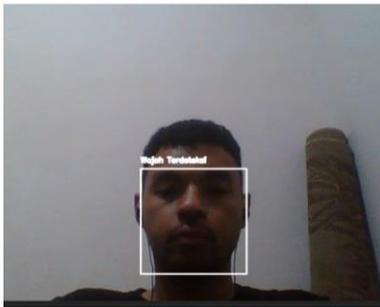
Tabel 1. Tabel Arsitektur metode CNN

No	Proses layer	Output shape	Keterangan
1	conv2d_92 (Conv2D)	(None, 98, 98, 64)	Kernel_size = 3x3 ; Stride = 1 ; Act = ReLu
2	max_pooling2d_69 (MaxPooling2D)	(None, 49, 49, 64)	Pool_size = 2x2
3	conv2d_93 (Conv2D)	(None, 47, 47, 256)	kernel_size = 3x3 ; Stride = 1 ; Act = ReLu
4	max_pooling2d_70 (MaxPooling2D)	(None, 23, 23, 256)	Pool_size = 2x2
5	conv2d_94 (Conv2D)	(None, 21, 21, 128)	kernel_size = 3x3 ; Stride = 1 ; Act = ReLu
6	conv2d_95 (Conv2D)	(None, 19, 19, 32)	kernel_size = 3x3 ; Stride = 1 ; Act = ReLu
7	max_pooling2d_71 (MaxPooling2D)	(None, 9, 9, 32)	Pool_size = 2x2
8	Flatten	(None, 2592)	" "
9	dense_69 (Dense)	(None, 100)	Act = Relu
10	dense_70 (Dense)	(None, 16)	Act = Relu
11	dense_71 (Dense)	(None, 2)	Act = Softmax

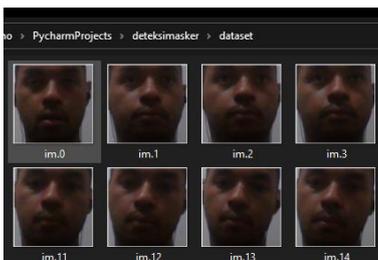
4. HASIL DAN PEMBAHASAN

4.1. Implementasi pre-processing

Implementasi ini merupakan tahap *preprocessing* yang terdiri dari tahap konversi video menjadi *frame*. kemudian akan lakukan proses *cropping* otomatis sebanyak jumlah yang di tentukan yaitu 10 pada daerah wajah. Hasil akhir *preprocessing* akan disimpan pada penyimpanan internal yang sudah di atur. *Sample* data yang telah menerapkan *preprocessing*, berikut hasil citra *preprocessing*:



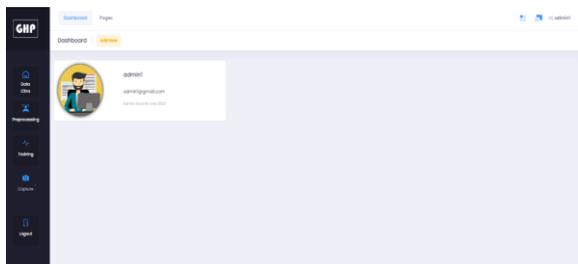
Gambar 13 sample hasil preprocessing



Gambar 14. Login form

4.2. Halaman Dashboard

Pada halaman *dashboard* menampilkan informasi mengenai *Admin* yang login dalam web. Selain itu pada admin terdapat menu *preprocessing*, *traing*, *capture* manual dan *capture* otomatis serta tombol *log out*.



Gambar 15. Dashboard

4.3. Halaman Informasi citra

Pada halaman data citra menampilkan informasi mengenai data citra warga yang sudah mengambil gambar secara otomatis.



Gambar 16. Data History

4.4. Halaman Capture Otomatis

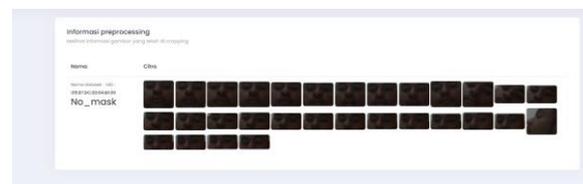
Pada halaman ini akan menampilkan satu buah *button* yang Ketika di klik maka *webcam* akan mengcapture gambar kira secara otomatis sebanyak 10 kali.



Gambar 17. Halaman capture otomatis

4.5. Halaman Preprocessing

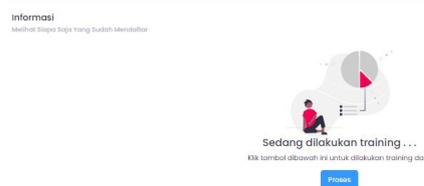
Pada halaman *preprocessing* di gunakan untuk memotong gambar di sekitara wajah yang telah di dapat dari proses *capture* otomatis.



Gambar 18. Preprocessing

4.6. Halaman Training

Pada halaman *training* Ketika di klik akan mentraining dataset yang telah di proses melalui *preprocessing* dan akan menghasilkan model *.h5* dalam penyimpanan sistem seperti gambar di bawah ini.

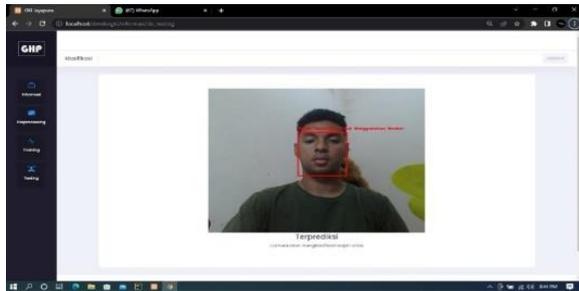


Gambar 19. Training

4.7. Halaman Testing

Pada halaman *training* Ketika di klik maka proses klasifikasi wajah akan di proses untuk di beri

tau label yang tertangkap oleh *webcam* hasilnya menggunakan masket atau tidak menggunakan masket.



Gambar 20. Testing

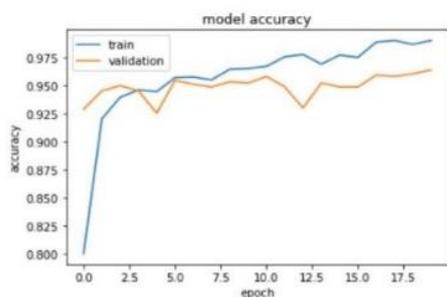
**4.8. Pengujian dataset**

Dataset yang telah ambil akan menerapkan *preprocessing* dan proses *training* untuk mendapatkan model .h5 sebagai pengenalan wajah masket yang menggunakan masket dan tidak menggunakan masket. Dataset pada setiap kelas abjad berjumlah sekitar 2 kelas dan totalnya sebanyak 3570 data citra digital RGB. *Training* dilakukan dengan nilai parameter *batchsize* sebanyak 32 serta dilakukan pembelajaran (*training epoch*) sebanyak 20 kali dengan durasi waktu 20 menit

Tabel 2. Hasil *training*, *validation loss* dan *accuracy* tiap *epoch*

Epoch	Training loss	Training accuracy	Validation Loss	Validation Accuracy
15	0.0294	0.9809	0.0907	0.9771
15	0.0656	0.9785	0.1716	0.9557
20	0.0250	0.9935	0.1654	0.9510
20	0.0335	0.9900	0.1885	0.9638

Berikut adalah hasil dari uji coba *training* model berupa *grafik accuracy model* dari proses *training* data dengan jumlah epoch sebanyak 20, ditunjukkan pada gambar 21



Gambar 21. Grafik model accuracy

**4.9. Hasil Uji Coba**

Uji coba aplikasi dilakukan pada 8 objek yang berbeda. Dari 8 objek ini dibagi menjadi dua, 4 objek menggunakan masket dan 2 objek yang tidak menggunakan masket, tabel 2 dan 3 merupakan hasil dari uji coba yang dilakukan.

maka langkah-langkah yang harus dilakukan adalah sebagai berikut:

Tabel 3. Hasil uji coba menggunakan masket

NO	Uji Coba	Penjelasan	Benar	Salah
1		webcam berhasil mendeteksi objek yang menggunakan masket	1	0
2		webcam berhasil mendeteksi objek yang menggunakan masket	1	0
3		webcam berhasil mendeteksi objek yang menggunakan masket	1	0
4		webcam berhasil mendeteksi objek yang menggunakan masket	1	0
5		webcam berhasil mendeteksi objek yang menggunakan masket	1	0
6		webcam gagal mendeteksi objek yang menggunakan masket	0	1
7		webcam berhasil mendeteksi objek yang menggunakan masket	1	0
8		webcam berhasil mendeteksi objek yang menggunakan masket	1	0
9		webcam berhasil mendeteksi objek yang menggunakan masket	1	0
10		webcam berhasil mendeteksi objek yang menggunakan masket	1	0
TOTAL			9	1

Tabel 4. Hasil uji coba tidak menggunakan masket

NO	Uji Coba	Penjelasan	Benar	Salah
1		webcam berhasil mendeteksi objek yang tidak menggunakan masket	1	0
2		webcam berhasil mendeteksi objek yang tidak menggunakan masket	1	0
3		webcam berhasil mendeteksi objek yang tidak menggunakan masket	1	0
4		webcam berhasil mendeteksi objek yang tidak menggunakan masket	1	0
5		webcam berhasil mendeteksi objek yang tidak menggunakan masket	1	0
6		webcam berhasil mendeteksi objek yang tidak menggunakan masket	1	0

NO	Uji Coba	Penjelasan	Benar	Salah
7		webcam gagal mendeteksi objek yang tidak menggunakan masker	0	1
8		webcam berhasil mendeteksi objek yang tidak menggunakan masker	1	0
9		webcam berhasil mendeteksi objek yang tidak menggunakan masker	1	0
10		webcam berhasil mendeteksi objek yang tidak menggunakan masker	1	0
TOTAL			9	0

Sehingga jumlah pengujian yang benar menggunakan masker dan tidak menggunakan masker total yaitu 18 dari jumlah 20 objek pengujian. Sehingga dapat di simpulkan bahwa tingkat akurasi yang di dapat yaitu 90 dengan perhitungan sebagai berikut:

$$\frac{18}{20} \times 100 = 90\%$$

#### 4.10. Pengujian black box

Pengujian *Black Box* dilakukan dengan cara menguji fitur yang terdapat pada sistem yang telah dibuat. Tujuan pengujian *Black Box* adalah untuk mengetahui apakah sistem yang telah dibuat dapat berjalan dengan baik atau masih terdapat kesalahan fungsi pada sistem.

Tabel 5. Pengujian halaman dashboard

No	Pengujian	Hasil yang di inginkan	Hasil Sistem	Kesimpulan
1	Fitur pada dashboard	Menampilkan informasi admin dengan jelas	Menampilkan informasi admin dengan jelas	Sesuai
2	Menu training data	Menampilkan waktu training dan model .h5	Menampilkan waktu training dan model .h5	Sesuai
3	Menu Preprocessing	Berhasil menampilkan waktu training	Berhasil menampilkan waktu training	Sesuai
4	Menu capture otomatis	Berhasil meng capture gambar dalam jumlah yang telah di tentukan	Berhasil meng capture gambar dalam jumlah yang telah di tentukan	Sesuai
5	Menu capture manual	Button di klik dan menyimpan gambar pada directory penyimpanan dan database	Button di klik dan menyimpan gambar pada directory penyimpanan dan database	Sesuai

#### 4.11. Pengujian Aplikasi

Pengujian fungsional aplikasi dilakukan dengan cara mengakses aplikasi pada *web browser* yang berbeda, bertujuan untuk mengetahui apakah aplikasi dapat menampilkan hasil yang sesuai dengan perancangan. Hasil dari pengujian fungsional dapat dilihat pada tabel 6.

Tabel 6. Pengujian Aplikasi

No	Pengujian	Chrome	Firefox	Microsoft edge
1	Dashboard	√	√	√
2	Data History	√	√	√
3	Preprocessing	√	√	√
4	Training	√	√	√
5	Pengujian kamera	√	√	√

Keterangan: √ Berhasil, X tidak berhasil

### 5. KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan pengujian dari perancangan alat pendeteksi masker Menggunakan Metode *Convolutional Neural Network* (CNN) dan pembersih tangan otomatis berbasis arduino, dapat ditarik kesimpulan dengan menggunakan *dataset* sebanyak 5.000 *dataset*. Maka hasil *accuracy model* yang di hasilkan sebesar 98% lebih baik dari pada *dataset* yang berjumlah 1.751 hanya menghasilkan *accuracy* sebesar 95%. Sensor *ultrasonik* berhasil mengenali jarak kurang dari 10 cm dan akan mengirimkan signal ke Arduino untuk menggerakkan servo menekan botol *handsanitizer* dengan akurasi sebesar 100%. Pada proses pengujian deteksi wajah yang memakai masker dan tidak bermasker. Sistem belum berhasil mengenali secara sempurna Ketika warna masker sama dengan baju dan kurangnya cahaya Berdasarkan pengujian *blackbox* dapat di simpulkan bahwa fitur menu telah sesuai seperti yang di harapkan dengan nilai berhasil 100% dari total pengujian 5 fitur. Adapun beberapa saran yang diberikan penulis untuk penelitian selanjutnya disarankan untuk memakai kamera yang memadai dengan resolusi full HD ataupun diatasnya, dengan syarat dapat mendeteksi wajah seseorang dengan jelas pada jarak 1 meter. Ketika ingin melakukan proses *testing* atau *monitoring* di depan kamera, sebaiknya level instensitas cahaya yang baik berkisar 50-500 lux. Agar proses *training* berjalan dengan baik spesifikasi pada komputer atau laptop minimal dengan processor intel i5gen 7, Ram 16 GB, dan VGA 1050 TI.

#### DAFTAR PUSTAKA

- [1] WHO. (2020). *Coronavirus disease (COVID-19): Schools*. Retrieved from WHO: <https://www.who.int/news-room/questions-and-answers/item/coronavirus-disease-covid-19-schools>
- [2] Gunawan, K. R. (2020). Dampak Covid-19 Terhadap Penjualan Masker dan *Handsanitizer* di

- Kabupaten Sumenep. *Eco-Entrepreneur*, 6(1), 25-33.
- [3] M, L. d. (2021, April). A technical review of face mask wearing in. *Curr. Opin. Colloid Interface Sci*, 52. doi:10.1016/j.cocis.2021.101417
- [4] Purnama, Dkk. (2021). Deteksi Masker Pencegahan Covid19 Menggunakan Convolutional Neural Network Berbasis Android. *Jurnal Resti*, 5 ( 3 ), 576 - 583. doi:2580-0760
- [5] Anarki, Dkk (2021). Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker.5(1), 179-186.
- [6] O'Shea, K., & Nash, R. (2015, Desember 2). An Introduction to Convolutional Neural Networks. 1-11.
- [7] Feri, D. (2011). *Pengenalan Arduino*. Retrieved from [www.tokobuku.com](http://www.tokobuku.com)
- [8] Arasada, B., & Suprianto, B. (2017). Aplikasi Sensor Ultrasonik Untuk Deteksi Posisi Jarak Pada Ruang Menggunakan Arduino Uno. 137-145.