

# Metode *Convolutional Neural Network* untuk Klasifikasi Wajah Bermasker

Farah Mufida Qotrunnada<sup>a,\*</sup>, Putranto Hadi Utomo<sup>b</sup>

<sup>a,b</sup> Universitas Sebelas Maret, Jl. Ir. Sutami No. 36, Kentingan, Jebres, Surakarta, 57126, Indonesia

\* Alamat Surel: [farahqotrunnada@student.uns.ac.id](mailto:farahqotrunnada@student.uns.ac.id)

## Abstrak

Pandemi COVID-19 banyak merubah kebiasaan masyarakat, yaitu dengan selalu menggunakan masker saat berada diluar rumah maupun di ruangan umum. Mengatasi masalah tersebut dibutuhkan sebuah sistem yang dapat membedakan antara wajah yang memakai masker dan wajah yang tidak memakai masker dengan menghasilkan tingkat akurasi yang baik, yaitu menggunakan metode *Convolutional Neural Network* (CNN). CNN merupakan salah satu model dari *deep learning* yang dirancang khusus untuk mengolah data dua dimensi. Data yang digunakan untuk penelitian ini sebanyak 3847 gambar dibagi menjadi tiga file yaitu, data *training*, *validation*, dan *testing*, data ukuran 150x150px dan epoch sebanyak 20. Proses *training* dan *validation* dilatih dengan menggunakan 3 layer dan 5 layer memperoleh tingkat akurasi dan hasil klasifikasi yang berbeda. Pelatihan dengan 3 layer memperoleh tingkat akurasi sebesar 99,20% untuk data *training* dan 70,59% untuk data *validation*, serta menunjukkan masih terdapat kelasah pengklasifikasian pada data *testing*. Sedangkan pengujian sistem dengan 5 layer memperoleh tingkat akurasi sebesar 98,20% untuk data *training* dan 82,35% untuk data *validation*, menunjukkan kecocokan klasifikasi untuk semua data *testing*. Proses dengan jumlah layer yang lebih banyak mendapatkan hasil klasifikasi yang lebih baik.

## Kata kunci:

*Convolutional Neural Network*, Klasifikasi, COVID-19, Masker

© 2022 Dipublikasikan oleh Jurusan Matematika, Universitas Negeri Semarang

## 1. Pendahuluan

Masker menjadi benda yang wajib dikenakan pada saat pandemi COVID-19. Masker merupakan suatu alat yang berfungsi sebagai pelindung diri dari penularan virus berbahaya. Pada 29 Januari 2020 Organisasi Kesehatan Dunia (WHO) mengeluarkan anjuran penggunaan masker dalam konteks COVID-19 (WHO, 2020). Masalah yang terjadi dalam proses pengenalan wajah yaitu, kondisi wajah yang menggunakan masker input ke sistem akan mempengaruhi tingkat akurasi sistem dalam menganalisis wajah. Laporan akhir Juli dari US National Institute of Standart and Technology (NIST) mengkonfirmasi bahwa penggunaan masker akan mengurangi akurasi dari pengenalan wajah. Algoritma paling akurat yang diuji lab menemukan error antara 5% sampai 50% pada hasil uji. Studi ini menguji keefektifan dari 89 algoritme face recognition komersial (NIST, 2020).

Mengatasi masalah tersebut digunakan sebuah sistem yang dapat meningkatkan keakuratan dalam mendeteksi wajah bermasker yaitu, *Convolutional Neural Network* (CNN). CNN merupakan suatu sistem dengan menerapkan model kerja *deep learning*. Penelitian ini menggunakan model CNN karena dinilai sebagai model yang paling sesuai fungsinya. Dalam artikel ilmiahnya Indra (Alam et al., 2019) mengatakan kemampuan CNN diklaim sebagai model terbaik yang digunakan untuk memecahkan permasalahan deteksi objek dan pengenalan objek karena tidak memerlukan komputasi yang besar dalam prosesnya. Menurut Yann (LeCun et al., 2015) *convolutional neural network* merupakan salah satu model dari *deep learning* yang disempurnakan untuk menyempurnakan model sebelumnya, model ini diharapkan dapat mengurangi jumlah parameter bebas serta deformasi gambar masukan seperti translasi, rotasi, dan skala dapat diatasi dengan baik.

To cite this article:

Qotrunnada, F. M., & Utomo, P. H. (2022). Metode *Convolutional Neural Network* untuk Klasifikasi Wajah Bermasker. *PRISMA, Prosiding Seminar Nasional Matematika 5*, 799-807

Pada penelitian ini, dirumuskan masalah: bagaimana cara mengimplementasi deep learning untuk klasifikasi wajah bermasker menggunakan metode CNN? Tujuan dari penelitian ini adalah untuk mengkonstruksi algoritme CNN serta mengimplementasikan deep learning untuk klasifikasi wajah bermasker menggunakan metode CNN. Penelitian ini diharapkan bermanfaat untuk mengembangkan dan mengaplikasikan ilmu pengetahuan di bidang matematika, mengetahui, dan menambah wawasan tentang cara klasifikasi wajah bermasker menggunakan metode CNN, serta dapat dikembangkan sehingga sistem dapat mendeteksi wajah bermasker secara cepat.

Berdasarkan hasil pengujian beberapa penelitian yang telah dilakukan pada tahun 2020 dengan menggunakan metode CNN diantaranya adalah penelitian yang dilakukan untuk mendeteksi wajah pegawai bank. Sistem keamanan dipasang pada pintu akses pegawai bank sekaligus mengaplikasikan face recognition. Menggunakan dataset sebanyak 350 data, menghasilkan tingkat akurasi 95% (Arsal et al., 2020). Selanjutnya penelitian untuk klasifikasi penggunaan masker dengan epoch 50 serta rasio dataset 90:10 untuk pelatihan data dan pengujian data menghasilkan tingkat akurasi sebesar 96% (Rahim et al., 2020). Pada tahun 2016 dilakukan penelitian untuk mendeteksi wajah secara real-time. Dataset yang digunakan sebanyak 126 gambar memperoleh rata-rata persentase akurasi sebesar 87,48% (Zufar et al., 2016).

---

## 2. Metode

Metode yang dilakukan untuk klasifikasi wajah bermasker meliputi studi literatur, pengumpulan data, perancangan sistem, pelatihan sistem, dan pengujian sistem.

### 2.1. Studi Literatur

Studi literatur pada penelitian ini digunakan untuk menganalisa dan mencari metode yang tepat untuk digunakan, serta memenuhi tujuan penelitian untuk menyelesaikan masalah yaitu, *deep learning*, *convolutional neural network* dan *tensorflow*. Masing-masing studi literatur tersebut diuraikan sebagai berikut.

#### 2.1.1. Deep Learning

Deep learning merupakan algoritme pemodelan yang ditata mendalam dan berlapis-lapis pada sekumpulan fungsi transformasi non-linear. Deep learning memanfaatkan banyak pengolahan informasi nonlinear untuk melakukan ekstraksi fitur, pengenalan pola, dan klasifikasi (Lambacing et al., 2020).

#### 2.1.2. Convolutional Neural Network (CNN)

Convolutional Neural Network atau yang disingkat sebagai CNN merupakan salah satu algoritme yang ada pada deep learning. CNN digunakan untuk mengklasifikasi gambar ataupun video juga digunakan untuk mendeteksi objek yang ada pada gambar atau bahkan wilayah yang ada didalam gambar (Moolayil, 2019). CNN tersusun dari layer yang memiliki susunan neuron 3D yaitu, lebar, tinggi, dan kedalaman yang mana susunan lebar dan tinggi merupakan ukuran layer, dan kedalaman merupakan jumlah layer (Zufar et al., 2016).

#### 2.1.3. Tensorflow

Tensorflow merupakan salah satu *library open source* yang dikembangkan oleh Tim Google Brain. Dalam *deep learning* tensorflow digunakan untuk menjalankan secara otomatis manajemen memori yang sama pada data yang digunakan. Tensorflow dapat menjalankan serta melatih jaringan untuk klasifikasi.

#### 2.1.4. Klasifikasi

Klasifikasi merupakan suatu proses untuk menemukan fitur yang dapat membedakan atau mendeskripsikan kedalam kategori kelas data. Klasifikasi bertujuan untuk memberikan perkiraan suatu objek yang belum diketahui labelnya kedalam suatu kelas.

### 2.2. Pengumpulan Data

Dataset yang digunakan pada penelitian ini adalah data gambar wajah bermasker dan tidak bermasker terbagi menjadi tiga file yaitu, data *training*, data *validation*, dan data *testing*.

Banyaknya data yang digunakan maka program *training* menjadi lebih banyak hingga memperoleh tingkat akurasi yang terbaik. Data pada penelitian ini merupakan data skunder yang didapatkan dari <https://www.kaggle.com/wobotintelligence/face-mask-detection-dataset>, terdapat 4326 gambar yang dibagi dalam dua kelas. Setelah mendapatkan dataset dilakukan pemilihan data, terdapat beberapa data yang dikurangi dan ditambahkan. Selanjutnya dilakukan pembagian data yang terbagi kedalam tiga file yaitu, *training*, *validation* dan *testing* dengan masing-masing file terbagi dua yaitu, *with mask* dan *without mask*, proses ini menghasilkan data sebanyak 3847 gambar.

### 2.3. Perancangan Sistem

Perancangan sistem klasifikasi wajah dilakukan pada tahapan ini. Tahapan ini terdiri dari *grayscale*, *thresholding*, *segmentasi*, dan *resize*.

#### 2.3.1. Grayscale

Pada Tahap ini citra berwarna *Red*, *Green*, *Blue* (RGB) diubah menjadi citra keabuan (*greyscale*). Menurut (Sinaga, 2019) Jumlah masing-masing komponen RGB adalah, R adalah 255 atau 8 bit, G adalah 255 atau 8 bit, dan B adalah 255 atau 8 bit biasa disebut sebagai *true color* atau intensitas 24 bit. Sementara untuk mendapatkan jumlah warna citra RGB masing-masing komponen dikalikan sehingga jumlah *byte* yang diperlukan untuk file citra RGB lebih dari 3 kali ukuran file *grayscale*. Perhitungan dilakukan agar mengecilkan range warna citra menjadi 0 hingga 255 agar dapat diproses dengan mudah pada tahap *thresholding*. Perhitungannya dapat dituliskan pada persamaan berikut:

$$y = (0,2989 * R) + (0,5870 * G) + (0,1141 * B) \quad (1)$$

Keterangan

y	:	<i>pixel grayscale value</i>
R	:	<i>pixel red value</i>
G	:	<i>pixel green value</i>
B	:	<i>pixel blue value</i>

#### 2.3.2. Thresholding

Pada proses ini menghasilkan citra biner yang diperoleh dari citra berwarna (*citra grayscale*) dengan memposisikan nilai *pixel* ke nilai 0 atau 1 tergantung dari ambang batas pada nilai *pixel* berada diatas atau dibawah (Solomon, 2011).

#### 2.3.3. Segmentasi

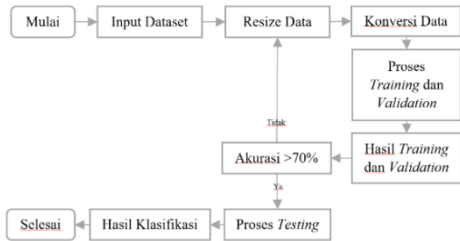
Proses segmentasi merupakan pembagian kelompok, yang pada penelitian ini hasil dari proses *thresholding* didapatkan citra biner yang kemudian diproses pemotongan pada setiap baris (vertikal) lalu pemotongan setiap kolom (horizontal).

#### 2.3.4. Resize

Setelah dari proses segmentasi, selanjutnya citra dilakukan proses *resize*. *Resize* merupakan suatu proses untuk mengubah ukuran citra mejadi lebih kecil maupun lebih besar dari ukuran sebelumnya. Pada penelitan ini, citra wajah di-*resize* menjadi citra berukuran 150x150 *px*.

### 2.4. Pelatihan Sistem

Pelatihan sistem dilakukan sebagai tahapan awal dari pengolahan sistem yang bertujuan untuk mengolah dataset yang telah disiapkan. Pada tahap ini dataset citra input diproses *training* dan *validation* menggunakan arsitektur CNN. Tahapan pada arsitektur CNN sendiri terdiri dari inialisasi, *feedforward*, *backpropagation*, dan *update* bobot. Proses ini akan mengubah data input yaitu citra biner menjadi sebuah *output* yang berupa klasifikasi wajah bermasker.



**Gambar 1.** Flowcart proses training sistem

2.4.1. Inisialisasi

Tahapan ini berisi inisialisasi parameter yaitu terdiri dari penentuan jumlah epoch, nilai *learning rate*, dan minimum *error*. Serta inisialisasi bobot yang terdiri dari penentuan nilai awal bobot dan bias pada tahap *convolution layer*, *hidden layer*, dan *fully connected layer*.

2.4.2. Feedforward

Tahapan ini akan memasuki proses arsitektur CNN, proses ini melewati *convolution layer*, *activation layer*, *pooling layer*, dan *fully connected layer* sehingga menghasilkan keluaran sebuah vektor.

a. Convolution Layer

Tahapan ini merupakan lapisan yang melakukan proses konvolusi antara matriks citra *input* dengan matriks *filter* sehingga menghasilkan keluaran berupa *feature map*. Persamaan rumus dari operasi konvolusi dapat dituliskan sebagai berikut:

$$FM[i]_{j,k} = (\sum_m \sum_n N_{[j-m,k-n]} F_{[m,n]}) + bF \tag{2}$$

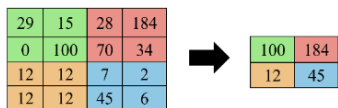
Keterangan

- FM[i] : Matriks *feature map* ke-i
- j,k : Posisi *pixel* pada matriks citra *input*
- m,n : Posisi *pixel* pada matriks *filter* konvolusi
- N : Matriks citra masukan
- F : Matriks *filter* konvolusi
- bF : Nilai bias pada *filter*

Selanjutnya dengan memasukkan fungsi aktivasi *Rectified Linear Unit* (ReLU). Seluruh *pixel* yang ada di *feature map* dimasukkan kedalam fungsi ReLU, dimana fungsi aktivasi ini berkerja memilah *pixel* yang memiliki nilai kurang dari 0 lalu mengubah nilainya menjadi 0, dengan rumus  $f(x) = \max(0,x)$ .

b. Pooling Layer

Pada lapisan ini dilakukan proses *pooling*. Jenis yang digunakan pada penelitian ini adalah max-pooling. Proses ini bekerja untuk mengurangi nilai dari *feature map* dengan cara memilah nilai maksimum di area tertentu. Hasil dari proses ini berupa matriks *feature map* yang berisi nilai maksimum yang telah dipilih.



**Gambar 2.** Contoh proses max pooling.

c. Fully Connected Layer

Lapisan ini terdiri dari *input layer*, *hidden layer* dan *output layer*.

d. Input Layer

Pada lapisan ini dilakukan proses penggabungan dari keseluruhan matriks *feature map* yang diperoleh pada proses *pooling*. Kemudian semua *pixel* tersebut diubah menjadi suatu vektor dengan panjang sejumlah *pixel* dari matriks yang diperoleh saat proses *pooling*. Nilai yang diperoleh pada proses ini digunakan untuk perhitungan ditahap selanjutnya.

#### e. Hidden Layer

Lapisan ini melakukan proses perhitungan dengan mengalikan nilai dari *input layer* dengan bobot yang telah diinisialisasi dan ditambahkan dengan nilai bias. Rumus dari perhitungan ditulis sebagai berikut:

$$z\_in_i = \sum_{j=1}^n X_j * V_{j,i} + V_{o,i} \quad (3)$$

Keterangan

$z\_in_i$	:	Input untuk <i>node hidden layer</i> ke-i dengan jumlah <i>node</i> n
$X_j$	:	Node X ke-j
$V_{j,i}$	:	Bobot V untuk $X_j$ dengan <i>node</i> $Z_i$
$V_{o,i}$	:	Bias V untuk $z\_in_i$

Selanjutnya dengan memasukkan fungsi aktivasi ReLU pada seluruh hasil perhitungan sehingga didapatkan nilai *output* Z. Kemudian digunakan pada proses perhitungan di *output layer*.

#### f. Output Layer

Pada lapisan ini dilakukan perhitungan nilai hasil dari *hidden layer* dikalikan dengan bobot yang sudah diinisialisasi, kemudian nilai tersebut ditambahkan dengan nilai bias. Rumus dari perhitungan dapat ditulis sebagai berikut:

$$y\_in_i = \sum_{j=1}^m Z_j * W_{j,i} + W_{o,i} \quad (4)$$

Keterangan

$y\_in_i$	:	Masukan untuk <i>node hidden layer</i> Z ke-i dengan jumlah <i>node</i> m
$W_{j,i}$	:	Bobot W untuk $Z_j$ dengan <i>node</i> $Y_i$
$W_{o,i}$	:	Bias W untuk $y\_in_i$
$Z_j$	:	Node Z ke-j

Selanjutnya dengan memasukkan fungsi aktivasi, pada penelitian ini digunakan *softmax* untuk semua hasil dari perhitungan, kemudian diperoleh nilai hasil perhitungan berupa nilai *output* Y. Rumus dari fungsi dapat ditulis sebagai berikut:

$$Y_i = \frac{e^{y\_in_i}}{\sum_{m=1}^m e^m} \quad (5)$$

Keterangan

M	:	Semua masukan untuk <i>output layer</i> sejumlah m
$y\_in_i$	:	Input untuk <i>node layer</i> ke-i
$Y_i$	:	Output untuk <i>output layer</i> ke-i

#### 2.4.3. Backpropagation

Tahapan ini dilakukan agar dapat menyesuaikan bias dan seluruh bobot berdasarkan *error* yang diperoleh saat proses *feedforward*. Langkah-langkah pada proses *backpropagation* yaitu, (1) menghitung gradien kesalahan terhadap bobot  $W_{ij}$ , (2) menghitung gradien kesalahan terhadap bobot  $V_{kj}$ , (3) menghitung gradien terhadap *filter*.

Setelah nilai gradien pada *input layer* didapatkan, lalu nilai dikembalikan dalam bentuk *feature map*, lalu mengacu pada nilai maksimum *feature map* sesuai hasil proses *max-pooling* pada proses *feedforward* dilakukan proses perhitungan nilai gradien terhadap *pooling layer*. Hasil akhir tahap ini merupakan gradien yang digunakan untuk *filter*.

#### 2.4.4. Update Parameter

Pada tahap ini metode yang digunakan pada penelitian adalah metode adam optimizer. Langkah-langkah pada proses *update parameter* yaitu, (1) hitung nilai momentum pertama ( $m_t$ ), (2) hitung nilai momentum kedua ( $v_t$ ), (3) hitung nilai momentum pertama dengan pembaruan bias ( $\hat{m}_t$ ), (4) *update* bobot dengan persamaan berikut:

$$\theta_t = \theta_{t-1} - \alpha \left( \frac{\check{m}_t}{\sqrt{\check{v}_t + \epsilon}} \right) \quad (6)$$

Keterangan

$\theta_t$	:	Nilai dari bobot baru
$\theta_{t-1}$	:	Nilai dari bobot lama
$\alpha$	:	<i>Learning rate</i>
$\check{v}_t$	:	Nilai momentum kedua dengan bias baru
$\check{m}_t$	:	Nilai momentum pertama dengan bias baru

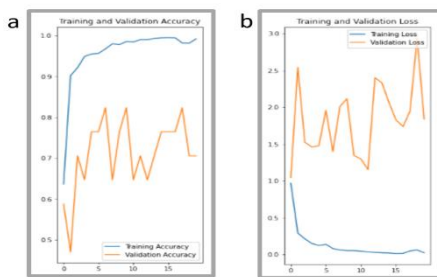
### 2.5. Pengujian Sistem

Setelah melewati proses pelatihan sistem, selanjutnya akan dilakukan pengujian sistem yang mana data akan langsung terklasifikasi. Pengujian sistem memberikan hasil akhir berupa kecocokan antara data masukan dan database dengan tingkat akurasi yang baik.

## 3. Hasil dan Pembahasan

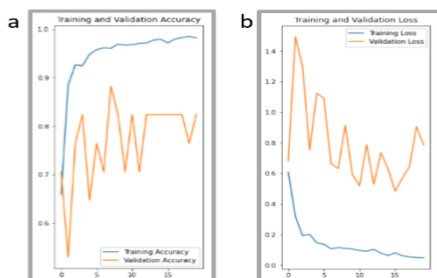
### 3.1. Pengujian Jumlah Layer

Skenario ini diuji terhadap data *training* dan *validation* dengan dua cara yaitu, dengan 3 layer dan 5 layer masing-masing epoch 20 dan gambar berukuran 150x150 px. Pengujian ini dilakukan untuk melihat performa sistem dalam pengaruh kedalaman layer.



**Gambar 3.** (a) *Training dan validation accuracy* dengan 3 layer; (b) *Training dan validation loss* dengan 3 layer.

Berdasarkan Gambar 3, dapat diketahui tingkat akurasi data mencapai optimal pada 99,20%, sementara tingkat akurasi data *validation* mencapai optimal pada 70,59% dengan menggunakan 3 layer



dan epoch 20.

**Gambar 4.** (a) *Training dan validation accuracy* sistem dengan 5 layer; (b) *Training dan validation loss* sistem dengan 5 layer.

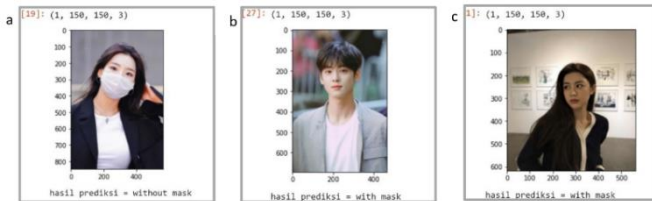
Berdasarkan Gambar 4, dapat diketahui tingkat akurasi data mencapai optimal pada 98,20%, sementara tingkat akurasi data *validation* mencapai optimal pada 82,35% dengan menggunakan 5 layer dan epoch 20.

Berdasarkan Gambar 3 dan Gambar 4, proses *training* dan *validation* sistem menggunakan 3 layer mendapatkan tingkat akurasi dengan jarak yang cukup jauh dengan nilai akurasi *validation* 70%,

dibandingkan dengan 5 layer. Tetapi proses dengan layer yang sedikit tidak membutuhkan waktu yang lama.

### 3.2. Hasil Pengujian dan Analisis

Pengujian sistem dilakukan dengan 3 layer dan 5 layer, gambar berukuran 150x150 px dan epoch 20 menguji 17 gambar yang terdapat di file *testing* menghasilkan pengklasifian 3 gambar dengan hasil yang berbeda.



**Gambar 5.** (a) Hasil klasifikasi wajah bermasker dengan 3 layer; (b) Hasil klasifikasi wajah tidak bermasker dengan 3 layer; (c) Hasil klasifikasi wajah tidak bermasker dengan 3 layer.

Gambar 5 memperlihatkan hasil klasifikasi oleh sistem menggunakan 3 layer menunjukkan masih terdapat kesalahan pengklasifikasian, dari total 17 gambar yang diuji mendapatkan 3 gambar yang salah diklasifikasi.

#### 3.2.1. Hasil “a”

Hasil a merupakan hasil dari proses klasifikasi wajah dengan 3 layer. Dapat dilihat bahwa hasil a merupakan gambar wajah bermasker tetapi diklasifikasi ke wajah tidak bermasker dengan hasil prediksi = *without mask*, hasil a menunjukkan hasil klasifikasi yang salah.

#### 3.2.2. Hasil “b”

Hasil b merupakan hasil dari proses klasifikasi wajah dengan 3 layer. Dapat dilihat bahwa hasil b merupakan gambar wajah tidak bermasker tetapi diklasifikasi ke wajah bermasker dengan hasil prediksi = *with mask*, hasil b menunjukkan hasil klasifikasi yang salah.

#### 3.2.3. Hasil “c”

Hasil c merupakan hasil dari proses klasifikasi wajah dengan 3 layer. Dapat dilihat bahwa hasil c merupakan gambar wajah tidak bermasker tetapi diklasifikasi ke wajah bermasker dengan hasil prediksi = *with mask*, hasil c menunjukkan hasil klasifikasi yang salah.



**Gambar 6.** (a) Hasil klasifikasi wajah bermasker dengan 5 layer; (b) Hasil klasifikasi wajah tidak bermasker dengan 5 layer; (c) Hasil klasifikasi wajah tidak bermasker dengan 5 layer.

Gambar 6 memperlihatkan hasil oleh sistem menggunakan 5 layer menunjukkan dari total 17 gambar yang diuji mendapatkan hasil klasifikasi yang sesuai.

#### 3.2.4. Hasil “a”

Hasil a merupakan hasil dari proses klasifikasi wajah dengan 5 layer. Dapat dilihat bahwa hasil a merupakan gambar wajah bermasker yang diklasifikasi ke wajah bermasker dengan hasil prediksi = *with mask*, hasil a menunjukkan hasil klasifikasi yang benar.

#### 3.2.5. Hasil “b”

Hasil b merupakan hasil dari proses klasifikasi wajah dengan 5 layer. Dapat dilihat bahwa hasil b merupakan gambar wajah tidak bermasker yang diklasifikasi ke wajah tidak bermasker dengan hasil prediksi = *without mask*, hasil b menunjukkan hasil klasifikasi yang benar.

### 3.2.6. Hasil "c"

Hasil c merupakan hasil dari proses klasifikasi wajah dengan 5 layer. Dapat dilihat bahwa hasil c merupakan gambar wajah tidak bermasker yang diklasifikasi ke wajah tidak bermasker dengan hasil prediksi = *without mask*, hasil c menunjukkan hasil klasifikasi yang benar.

Berdasarkan Gambar 5 dan Gambar 6, diketahui bahwa dengan menggunakan 3 layer hasil sistem masih terdapat kesalahan dalam klasifikasi wajah bermasker. Terdapat data gambar wajah bermasker yang diklasifikasi ke wajah tidak bermasker dan gambar wajah tidak bermasker yang diklasifikasi ke wajah bermasker. Hasil klasifikasi wajah dengan 5 layer lebih baik daripada dengan 3 layer.

**Tabel 1.** Hasil pengujian sistem

Jumlah Dataset	Jumlah Layer	Jumlah Data Testing	Jumlah Kecocokan Data	Hasil (%)	
				Training	Validation
3847	3 Layer	17	14	99,20	70,59
	5 Layer		17	98,20	82,35

Berdasarkan hasil pengujian sistem pada Table 1, diketahui bahwa dengan menggunakan data berjumlah 3847 gambar, ukuran 150x150 px, epoch sebanyak 20, menggunakan 3 layer untuk proses training dan validation dengan menggunakan convolutional neural network menghasilkan tingkat akurasi optimal sebesar 99,20% dan 70,59%, sedangkan pada penggunaan 5 layer menghasilkan tingkat akurasi optimal 98,20% dan 82,35%. Pengujian pada data yang menggunakan 3 layer menghasilkan kecocokan 14 data dari 17 data testing, sedangkan pengujian data dengan 5 layer menghasilkan kecocokan 17 data dari 17 data testing.

## 4. Simpulan

Kesimpulan yang diperoleh dari penelitian ini, penelitian berhasil mengimplementasikan *deep learning* dengan menggunakan metode *convolutional neural network*, serta menggunakan *library tensorflow* untuk klasifikasi gambar wajah bermasker yang terbagi dalam kelas *with mask* dan *without mask*. Dari hasil pengujian dataset sebanyak 3847 gambar ukuran 150x150 px dan epoch sebanyak 20 memperoleh tingkat akurasi data sebesar 98,20% pada 5 layer. Pencocokan 17 data testing menghasilkan semua data memiliki kecocokan dengan database yang ada. Penggunaan jumlah layer mempengaruhi waktu dan tingkat akurasi pada pengujian data. Proses dengan jumlah layer yang sedikit tidak membutuhkan waktu yang lama, sementara proses dengan jumlah layer yang lebih banyak membutuhkan waktu yang cukup lama dan memperoleh hasil klasifikasi yang lebih baik.

## Daftar Pustaka

- Alam, I. F., Sarita, M. I., & Sajiah, M. A. (2019). Implementasi Deep Learning dengan Metode Convolutional Neural Network (CNN) untuk Identifikasi Objek Secara Real Time Berbasis Sistem Android. *semanTIK*, 237-244.
- Arsal, M., Wardijono, B. A., & Anggraini, D. (2020). Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN. *Jurnal Nasional Teknologi dan Sistem Informasi*, 06, 055-063.
- Lambacing, M. M., & Ferdiansyah. (2020). Rancang Bangun New Normal COVID-19 Masker Detector Dengan Notifikasi Telegram Berbasis Internet of Things. *Jurnal Dinamik*, 25, 77-84.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 436-444.



- Moolayil, J. J. (2019). *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python*. British Columbia: Apress.
- NIST. (2020). *Algorithms created before the pandemic generally perform less accurately with digitally masked faces*. (Online). (<https://www.nist.gov/news-events/news/2020/07/nist-launches-studies-masks-effect-face-recognition-software>, diakses 20 September 2021).
- Rahim, A., Kusriani, & Luthfi, E. (2020). Convolutional Neural Network untuk Klasifikasi Pengguna Masker. *Jurnal Teknologi Informasi dan Komunikasi*, 01, 109-115.
- Sinaga, A. S. (2019). Color-based Segmentation of Batik Using the L\*a\*b Color Space. *Journal Publications & Informatics Engineering Research (SinKron)*, 175-179.
- Solomon, C., & Breckon, T. (2011). *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. New Jersey: Wiley.
- WHO. (2020). Advice on the use of masks in the community, during home care and in health settings in the context of the novel coronavirus (2019-nCov) outbreak. (Online). ([https://www.who.int/publications/i/item/advice-on-the-use-of-masks-in-the-community-during-home-care-and-in-healthcare-settings-in-the-context-of-the-novel-coronavirus-\(2019-ncov\)-outbreak](https://www.who.int/publications/i/item/advice-on-the-use-of-masks-in-the-community-during-home-care-and-in-healthcare-settings-in-the-context-of-the-novel-coronavirus-(2019-ncov)-outbreak), diakses 20 September 2021).
- Zufar, M., & Setiyono, B. (2016). Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time. *Jurnal Sains dan Seni ITS*, 2337-3520.