
FACE MASK DETECTION MENGGUNAKAN PYTHON DAN OPENCV UNTUK MENDETEKSI PELANGGARAN PROTOKOL KESEHATAN COVID-19

Riza Syahrial¹⁾, Tati Sukmawati²⁾, Elva Nofia Dewi³⁾
^{1,2,3}Prodi Sistem Informasi, Fakultas Teknologi, ITB Swadharma

Correspondence author: R. Syahrial, rzsyahrial@swadharma.ac.id, Jakarta, Indonesia

Abstract

Face Detection Technology is a computer technology for detecting human faces. The popular method used is the Convolutional Neural Networks (CNN) method. Using a combination of object detection classification, image and object tracking, for masked or non-masked face detection systems in the form of images or videos. Face detection in Python can be done using the OpenCV library and the method applied is the Convolutional Neural Networks (CNN) method. The problem in this research is how to build a computer program with the Python programming language, using Google Collab, and apply and test it in the face detection process using a mask or not to detect violations of the Covid-19 health protocol. The goal to be achieved through this research is to produce an image-processing system that uses Image processing aims to help prevent Covid-19, especially regarding calls for the use of masks in public spaces.

Keywords: face mask detection, convolutional neural network, opencv

Abstrak

Teknologi Deteksi Wajah adalah teknologi komputer untuk mendeteksi wajah manusia, metode yang populer digunakan adalah metode Convolutional Neural Networks (CNN). Menggunakan kombinasi klasifikasi deteksi objek, gambar, dan pelacakan objek, untuk sistem pendeteksian wajah bermasker atau tidak bermasker dalam bentuk gambar atau video. Pendeteksian wajah dengan Python dapat dilakukan dengan menggunakan library OpenCV dan Metode yang diterapkan adalah metode Convolutional Neural Networks (CNN). Permasalahan dalam penelitian ini bagaimana membangun program komputer dengan bahasa pemrograman Python, menggunakan Google Collab, serta menerapkan dan mengujinya dalam proses pendeteksian wajah menggunakan masker atau tidak untuk mendeteksi pelanggaran protokol kesehatan Covid-19. Tujuan yang ingin dicapai melalui penelitian ini adalah menghasilkan sistem pengolahan citra yang menggunakan image processing bertujuan untuk membantu pencegahan Covid-19 khususnya terkait himbuan penggunaan masker pada ruang publik.

Kata Kunci: deteksi wajah, convolutional neural network, opencv

A. PENDAHULUAN

Pada bulan Desember 2019 dunia dikejutkan dengan kemunculan wabah baru yaitu virus Covid-19 (Ahmadi et al., 2021). Kasus pertama virus ini ditemukan di negara China, tepatnya di kota Wuhan, kemudian menyebar hampir ke seluruh dunia, termasuk Indonesia. Pada Maret 2020 virus Covid-19 mulai muncul di Indonesia yang kemudian menyebabkan krisis, terutama pada sektor kesehatan dan ekonomi. Untuk menekan penyebaran virus tersebut pemerintah melakukan berbagai upaya, salah satunya dengan menerapkan Pembatasan Sosial Berskala Besar (PSBB) dan mewajibkan masyarakat untuk melaksanakan protokol kesehatan dalam segala bidang kegiatan, yaitu dengan menerapkan 5M yaitu memakai masker, mencuci tangan pakai sabun dan air mengalir, menjaga jarak, menjauhi kerumunan, serta membatasi mobilisasi dan interaksi (Septiana et al., 2020). Dengan adanya kewajiban menerapkan protokol kesehatan tersebut, maka dalam kegiatan sehari-hari masyarakat di tempat umum diwajibkan untuk menggunakan masker guna menekan penyebaran virus Covid-19.

Teknologi deteksi wajah (*Face Detection*) adalah teknologi komputer yang digunakan untuk mendeteksi bagian wajah manusia (Baay et al., 2021; Luthfillah Ahmad et al., 2021; Septiana et al., 2020). Teknologi ini dibangun dengan menggunakan algoritma tertentu yang akan menentukan dimana bagian wajah yang muncul pada citra masukan. Salah satu metode yang populer digunakan untuk pendeteksian wajah saat ini adalah metode Convolutional Neural Networks (CNN) (Septiana et al., 2020; Setyadi et al., 2022). Penelitian ini akan menggunakan metode Convolutional Neural Networks (CNN). Algoritma pembelajaran Convolutional Neural Networks (CNN) memanfaatkan ekstraksi fitur dari citra yang nanti akan dipelajari oleh beberapa hidden layer. Dengan sistem ini menggunakan kombinasi

klasifikasi deteksi objek, gambar, dan pelacakan objek sehingga dapat mengembangkan sistem pendeteksian wajah bermasker atau tidak bermasker dalam bentuk gambar atau video (Septiana et al., 2020).

Bahasa pemrograman Python merupakan bahasa pemrograman tingkat tinggi yang dapat digunakan untuk membangun program pendeteksian wajah (Wirawan et al., 2022). Pendeteksian wajah dengan Python dapat dilakukan dengan menggunakan library OpenCV (Bagus Setiawan et al., 2021). Metode yang diterapkan untuk deteksi objek pada Library OpenCV adalah metode Convolutional Neural Networks (CNN). Saat ini, Google memberikan fasilitas bahasa pemrograman Python melalui Google Collaboratory (Google Collab). Google Collab merupakan salah satu produk yang berbasis cloud dan dapat digunakan secara gratis (Boulila et al., 2021).

Dalam penelitian ini akan dibangun program komputer untuk mendeteksi wajah dan mengklasifikasikan apakah wajah yang terdeteksi menggunakan masker atau tidak. Data yang akan digunakan untuk pengujian merupakan gambar manusia yang didapat dari lokasi penelitian pada ruang publik. Program yang dibangun akan memanfaatkan Google Collab untuk pemrograman Python, serta menggunakan library OpenCV dengan metode Convolutional Neural Networks (CNN) untuk deteksi objek wajah dan masker. Permasalahan yang akan dipelajari lebih lanjut berdasarkan latar belakang yang telah diuraikan yaitu Bagaimana membangun program komputer dengan bahasa pemrograman Python, menggunakan Google Collab, serta menerapkan dan menguji metode Convolutional Neural Networks (CNN) pada Library OpenCV dalam proses pendeteksian wajah menggunakan masker atau tidak untuk mendeteksi pelanggaran protokol kesehatan Covid-19? Tujuan yang ingin dicapai melalui penelitian ini adalah untuk mengetahui tingkat akurasi deteksi masker wajah setelah pengujian dataset

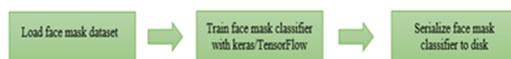
dilakukan. Secara lebih lanjut, tujuan dari penelitian ini adalah menghasilkan sistem pengolahan citra (Fadjeri et al., 2022; Setyadi et al., 2022) yang menggunakan *image processing* bertujuan untuk membantu pencegahan Covid-19 khususnya terkait himbauan penggunaan masker pada ruang publik.

B. METODE PENELITIAN

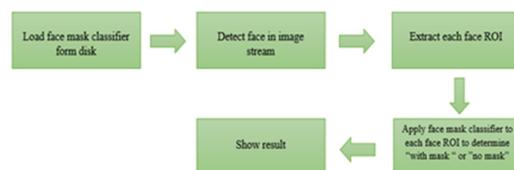
Metode yang digunakan dalam penelitian ini adalah metode kualitatif dengan metode pengumpulan data dengan observasi lapangan dan studi pustaka.

Observasi dilakukan pada berkaitan dengan *image processing* yang memanfaatkan *deep learning* untuk mempelajari ciri khusus dari setiap objek terutama dalam membaca objek wajah yang menggunakan masker dan tidak menggunakan masker.

Pada penelitian *face mask detection* terdapat dua proses yaitu *Train Face Mask Detector* dan *Apply Face Mask Detector*. Gambar 1. merupakan diagram training dengan menggunakan dataset deteksi wajah, dengan dataset tersebut akan membuat serial deteksi masker wajah. Dataset tersebut nanti akan digunakan untuk membuat detektor masker, melakukan deteksi wajah, lalu mengklasifikasikan setiap wajah yang menggunakan masker dan tidak menggunakan masker yang ditunjukkan pada gambar 2.



Gambar 1. Train Face Mask Detector



Gambar 2. Apply Face Mask Detector

Pada penelitian *mask face detection* menggunakan *image processing* dengan sistem preprocessing gambar sebelum melanjutkan ke tahap berikutnya. Dalam langkah preprocessing gambar di rubah menjadi RGB karena gambar berwarna RGB mencakup banyak informasi berlebih yang tidak diperlukan untuk *face mask detection*.

Deep Learning Architecture mempelajari hal penting yaitu fitur non-linier dari sampel yang diberikan. Kemudian, mempelajari arsitektur yang digunakan untuk memprediksi sampel yang sebelumnya tidak terlihat. Oleh sebab itu, *Deep Learning Architecture* sangat tergantung pada CNN. Pada *Deep Learning Architecture* ada beberapa tahapan yang harus dilakukan yaitu, *Dataset Collection*, *Training*, dan *Deployment*.

Pada *Dataset Collection* merupakan tahapan pengumpulan data dari beberapa sumber yang berbeda, data yang dikumpulkan terdapat 2 jenis yaitu, data pelatihan dan data pengujian model. Dari data yang telah dikumpulkan diperoleh total 7553 sampel.

Training merupakan processing yang terfokus untuk membuat *dataset face mask detection* dari penyimpanan dataset, melatih model dengan menggunakan instrumen dari Keras atau TensorFlow dari dataset ini, akan membuat serial *face mask detection* pada penyimpanan dataset.

Deployment merupakan processing setelah *face mask detection* yang sudah melalui training dataset, selanjutnya akan dapat melanjutkan untuk membuat detektor wajah, melakukan deteksi wajah dengan memberikan *bounding box* yang akan menyeleksi area sekitar wajah, kemudian akan terfokus pada *bounding box* yang akan memproses sistem ROI (*Region Of Interest*), kemudian *detect facial landmark* yang akan mendeteksi titik-titik yang ada pada area wajah, setelah proses tersebut maka sistem akan mengklasifikasikan setiap wajah yang terdeteksi menggunakan masker 'Mask' atau tidak menggunakan masker 'No Mask' dan

menggambar bounding boxes di sekitar objek deteksi dengan nama kelas yang terdeteksi.

Untuk studi Pustaka penulis mengumpulkan mencari referensi yang dibutuhkan untuk mengumpulkan informasi dalam penelitian ini. Pencarian referensi didapat dari jurnal, buku, literatur sejenis, skripsi yang ada dan dari internet. Studi literatur dilakukan untuk mencari referensi yang terkait atau mempunyai kesamaan dengan penelitian ini.

C. HASIL DAN PEMBAHASAN

Pengumpulan data dilakukan dengan mengidentifikasi kebutuhan data. Data apa saja yang dibutuhkan untuk dapat mencapai sasaran/goal penelitian. Sistem yang dibangun adalah sistem face mask detection dengan menggunakan metode Convolutional Neural Networks (CNN). Pada dasarnya terbagi dalam dua tahapan, yaitu tahap training dan tahap testing.

1. Tahap training akan menggunakan data input yang berasal dari dataset train yang disimpan pada google cloud (<https://drive.google.com/drive/folders/1kOfpmBs8a4WDS6uHWnimr3XQ8w0Pl4-1?usp=sharing>). Data train terdiri dari dua dataset yaitu dataset with_mask (dengan masker) berisi 100 foto wajah orang yang menggunakan masker dan dataset without_mask (tanpa masker) berisi 100 foto wajah orang yang tanpa menggunakan masker dengan ukuran foto yang beragam dan foto berekstensi .jpg dengan 3 channel (RGB).
2. Tahap testing akan menggunakan data gambar dari proses pengumpulan data yang dilakukan oleh peneliti yang berasal dari dataset testing yang disimpan pada google cloud (https://drive.google.com/drive/folders/1G6jQDQ9jy8hkUckTXj9YrVIuh0_U6ROG?usp=sharing). Data testing terdiri dari dua dataset yaitu dataset with_mask (dengan masker) berisi 100 foto wajah orang yang menggunakan

masker dan dataset without_mask (tanpa masker) berisi 100 foto wajah orang yang tanpa menggunakan masker.

Proses pengambilan data training dilakukan menggunakan hasil gambar yang didapat dari lingkungan kerja dan kampus. Pada dataset yang dikumpulkan harus memiliki komposisi yang seimbang antara memakai masker dan tidak memakai masker dengan satu objek orang yang sama.



Gambar 3. Dataset Training

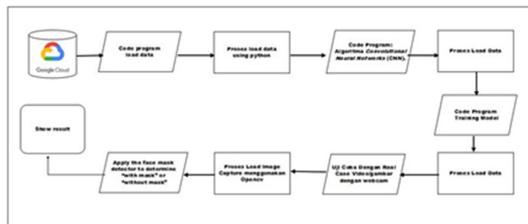
Proses pengambilan data testing dilakukan secara langsung menggunakan hasil gambar yang didapat dari teman di lingkungan kerja dan kampus. Kemudian dataset testing disimpan di google drive sebagai data objek penelitian.



Gambar 4. Dataset Testing

Metode Pemrosesan Data

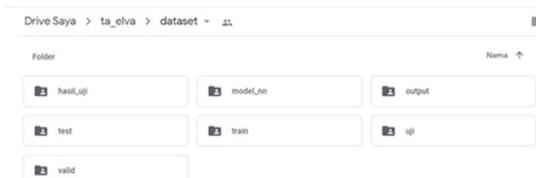
Data preparation dan proses model dilakukan pada google cloud platform dengan skema seperti berikut:



Gambar 5. Google Cloud Platform

Data Preparation

Data preparation adalah suatu proses yang dilakukan untuk menyiapkan data. Diantaranya yaitu menempatkan data pada environment cloud. Dan juga menyiapkan library yang digunakan untuk pemrosesan data. Menyiapkan koneksi untuk menjalankan pemrosesan data. Pertama-tama dataset di simpan di google cloud sehingga terlihat seperti ini:



Gambar 6. Dataset Google Drive

Project Workflow (Alur Kerja Proyek)

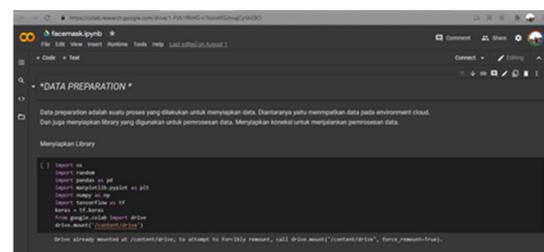
1. Load Dataset

Perlu mengimpor semua library yang diperlukan. Tentukan fungsi loading_data() untuk beralih melalui direktori untuk memasukkan semua direktori file ke dalam daftar python.

- Library os : berfungsi untuk menampilkan letak direktori program python yang akan disimpan dalam sistem operasi tersebut.
- Library Random : digunakan untuk melakukan operasi-operasi yang berkaitan dengan bilangan acak.
- Library pandas : library di Python yang berlisensi BSD dan open source yang

menyediakan struktur data dan analisis data yang mudah digunakan. Pandas biasa digunakan untuk membuat tabel, mengubah dimensi data, mengecek data, dan lain sebagainya .

- Library Matplotlib : pustaka visualisasi data multiplatform yang dibangun di atas array NumPy. Matplotlib juga dapat digunakan untuk memvisualisasikan data secara 2D maupun 3D di dalam Python dan menghasilkan gambar berkualitas dalam berbagai format..
- Library Numpy: berfungsi memudahkan operasi komputasi tipe data numerik seperti penjumlahan, pengurangan, perkalian, pangkat, dan operasi lainnya yang bisa diterapkan pada vektor atau matriks.
- Library Tensorflow memungkinkan developer untuk membuat grafik aliran data (data flow graph), yaitu struktur yang mendeskripsikan bagaimana data yang digunakan bergerak melalui tampilan grafis atau serangkaian pemrosesan node. Node dan Tensor di dalam Tensorflow merupakan suatu object pada Python.



Gambar 7. Library

```
def loading_data():
    dirname = '/content/drive/MyDrive/ta_elva/dataset/'
    dirtype = ['train', 'valid']
    dirclass = ['with_mask/', 'without_mask/']

    x_train = []
    y_train = []
    x_test = []
    y_test = []

    for typ in dirtype:
        for cls in dirclass:
            for i in os.listdir(dirname+typ+cls):
                if typ == 'train/':
                    x_train.append(dirname+'train/'+cls+i)
                    y_train.append(cls[:-1])
                else:
                    x_test.append(dirname+'valid/'+cls+i)
                    y_test.append(cls[:-1])
    return np.array(x_train), np.array(y_train), np.array(x_test), np.array(y_test)

# Run loading_data() function
x_train, y_train, x_test, y_test = loading_data()
```

Gambar 8. Load Data

Untuk memproses data koneksi data gambar atau data image yang ada di cloud ke python. Dan juga membagi menjadi data training dan testing untuk proses model.

2. Data Processing

Mempersiapkan Image untuk Image Processing (Data Training dan Data Testing). Dan juga mempersiapkan image dalam bentuk array. Di sini peneliti tidak menggunakan Keras flow_from_directory untuk memuat dataset, tetapi menggunakan input output Tensorflow tf.io.read_file(path) karena peneliti ingin membuat batch data sebelum melatih model, sehingga proses pelatihan dapat berjalan lebih cepat

```
def process_label(label):
    label = [i == unique_label for i in label]
    label = np.array(label).astype(int)
    return label

def processImage(path):
    image = tf.io.read_file(path)
    image = tf.image.decode_jpeg(image, channels=3)
    image = tf.image.convert_image_dtype(image, tf.float32)
    image = tf.image.resize(image, size=[224, 224])
    return image

# Create batch data of numpy array
def pairData(image, label):
    return processImage(image), label

def batchData(image, label=None, for_valid=False, for_test=False):
    if for_test:
        data = tf.data.Dataset.from_tensor_slices((image))
        batch = data.map(processImage).batch(32)
        return batch
    elif for_valid:
        data = tf.data.Dataset.from_tensor_slices((tf.constant(image), tf.constant(label)))
        batch = data.map(pairData).batch(32)
        return batch
    else:
        data = tf.data.Dataset.from_tensor_slices((tf.constant(image), tf.constant(label)))
        data = data.shuffle(buffer_size=len(image))
        batch = data.map(pairData).batch(32)
        return batch

unique_label = np.unique(y_test)
y_test = process_label(y_test)
y_train = process_label(y_train)

train_data = batchData(x_train, y_train)
valid_data = batchData(x_test, y_test, for_valid=True)
```

Gambar 9. Data Processing

3. Membangun Model menggunakan Convolutional Neural Network (CNN)

Peneliti membuat Convolutional Neural Network (CNN) dengan Keras. Disini peneliti menggunakan unit Dense layer 2 dengan aktivasi Softmax sebagai output node karena ingin 2 output array agar lebih mudah menerima informasi prediksi.

```
model = keras.Sequential([
    # Input Layer
    keras.layers.Conv2D(input_shape=(224, 224, 3), filters=32, kernel_size=(3, 3), activation='relu'),
    keras.layers.MaxPooling2D(),
    # Hidden Layer
    keras.layers.Conv2D(input_shape=(224, 224, 3), filters=64, kernel_size=(3, 3), activation='relu'),
    keras.layers.MaxPooling2D(),
    keras.layers.Conv2D(input_shape=(224, 224, 3), filters=128, kernel_size=(3, 3), activation='relu'),
    keras.layers.MaxPooling2D(),
    keras.layers.Flatten(),
    keras.layers.Dense(128),
    keras.layers.Activation('relu'),
    # Output Layer
    keras.layers.Dense(2),
    keras.layers.Activation('softmax')
])

# Compile model
model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.Adam(), metrics=['acc'])
```

Gambar 10. Building Neural Network

Pengujian yang dilakukan untuk model CNN terhadap model yang sudah dilatih dengan jumlah epoch sebanyak 50 epoch. Pelatihan dilakukan menggunakan jumlah data latih sebanyak 200 gambar yang dibagi menjadi 100 gambar menggunakan masker dan 100 tidak menggunakan masker. Sedangkan untuk validasi sebanyak 50 gambar menggunakan masker dan 50 gambar tidak menggunakan masker sehingga total data validasi sebanyak 100 data. Setelah dilakukan proses

Model: "sequential"

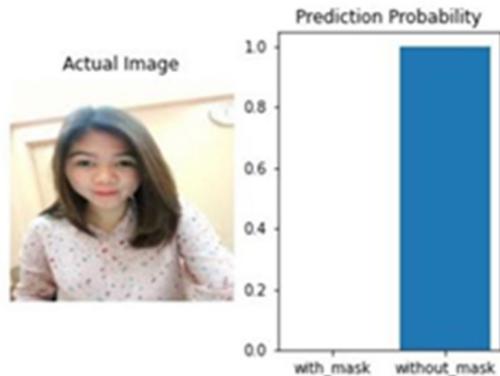
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 128)	11075712
activation (Activation)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258
activation_1 (Activation)	(None, 2)	0

Total params: 11,169,218
 Trainable params: 11,169,218
 Non-trainable params: 0

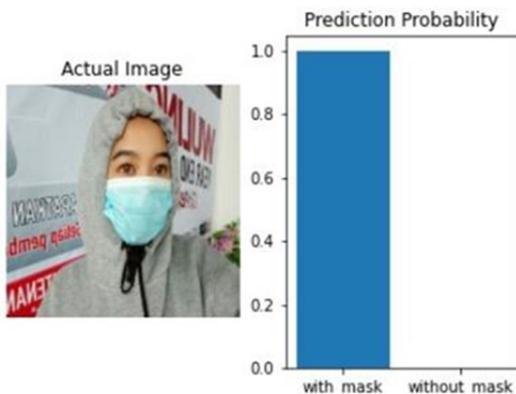
Gambar 11. Model Summary

4. Training Model

Proses training model terdiri dari 50 epoch dan hanya berjalan selama 17.5 epoch. Karena menggunakan early stop agar proses training bisa dihentikan secara otomatis jika model tidak membaik selama 4 epoch. Ini juga digunakan untuk menghindari overfitting.



Gambar 14. Without Mask



Gambar 15. With Mask

Pendekatan yang diusulkan untuk face mask detection dinyatakan dengan berbagai langkah implementasi yang terlibat dalam pendekatan ini. Diagram alir untuk ini disajikan di bawah yang menyatakan aliran keseluruhan dari pendekatan tersebut.

Dimulai dari penyimpanan dataset yang disimpan pada google cloud kemudian Load dataset untuk memproses data koneksi data gambar atau data image yang ada di cloud ke python dan juga membagi menjadi data training dan testing untuk proses model. Kemudian mempersiapkan image processing (data training dan data testing) dan mempersiapkan image dalam bentuk array.

Kemudian Model dimulai dengan pembuatan dataset untuk face mask detection dan preprocessing data dilakukan dengan algoritma Convolutional Neural Networks (CNN). Kemudian setelah dilakukan

pemodelan data dilakukan training model dengan dataset 50 epoch yaitu 1/50 epoch dan yang terdeteksi menghasilkan training history. Kemudian setelah serialisasi dilakukan pengklasifikasi masker wajah. Fase pertama dari model ini adalah seperti yang dinyatakan dalam pernyataan di atas dan proses pelatihan dilakukan dengan menggunakan framework python dan opencv.

Setelah mendapatkan serialisasi masker wajah akan memuat pengklasifikasi masker wajah ini, dan kemudian wajah dimuat dari aliran gambar / video yang tersedia dengan menggunakan webcam. Tahap selanjutnya dengan preprocessing transformasi opencv. Di sini, di ujung detektor masker wajah mendeteksi orang-orang dengan masker dan tanpa masker.

Analisis Kelayakan Sistem

Untuk menjalankan sistem yang telah dirancang, dibutuhkan beberapa faktor pendukung sebagai berikut:

Perangkat keras yang dibutuhkan pada saat pengembangan sistem pelayanan ini adalah Komputer PC sebagai Server dengan spesifikasi sebagai berikut: Core i3 3.40 GHz, RAM 8 GB, Hard disk 500 GB, VGA 1 GB, Jaringan Komputer.

Perangkat lunak yang dibutuhkan pada saat pengembangan aplikasi ini adalah sebagai berikut; Sistem operasi Windows 10 / LINUX / MAC-OS, Web browser google chrome, Penyimpanan data menggunakan Google Cloud Platform (Google Drive & Google Collaboratory / Collab), Pemrosesan di google cloud platform, Webcam (5MP). Perangkat lunak dan perangkat keras yang diusulkan dengan spesifikasi tersebut, sehingga pengembangan sistem yang diusulkan layak secara teknologi.

Kelayakan operasional dinilai dari segi penggunaan mudah karena memakai alat-alat yang standar yaitu web browser, akun gmail, webcam, google Collab yang biasa kebanyakan orang menggunakannya juga

sehingga dapat dikembangkan dan dioperasikan dengan baik secara operasional sudah layak memenuhi standar.

Kelayakan hukum adalah kelayakan yang berkaitan dengan legalitas atau kekuatan hukum. Aplikasi ini menggunakan dibuat environment google cloud platform yang disediakan secara gratis / free dibuat untuk para developer, dan codingan yang dibuat tidak mengandung unsur-unsur yang membahayakan pada perangkat keras maupun perangkat lunak, dan tidak melanggar hukum yang berlaku, baik hukum yang ditetapkan oleh pemerintah maupun hukum yang ditetapkan berdasarkan peraturan- peraturan organisasi. Proyek aplikasi yang akan dikembangkan secara hukum dinilai layak secara hukum.

D. PENUTUP

Pendeteksian wajah menggunakan metode Convolutional Neural Networks (CNN) memiliki kemampuan dan performance yang sangat baik, hal ini terlihat berdasarkan pengujian yang dilakukan terhadap identifikasi masker wajah dalam kondisi yang memungkinkan dapat mempengaruhi hasil pendeteksian masker wajah. Keakuratan pendekatan face mask detection yang diusulkan dengan membagi dataset menjadi dua yaitu dataset Training dan dataset Testing dengan data with_mask dan without_mask. Prediksi ini menghasilkan angka akurasi sebanyak 95% terlihat pada garis akurasi dan training loss menghasilkan angka sebanyak 5%.

Kinerja aplikasi face mask detection ini dipengaruhi oleh spesifikasi laptop, menggunakan spesifikasi yang rendah akan menyebabkan “Not Responding” pada aplikasi atau slow motion pada hasil tampilan kamera. Spesifikasi yang disarankan adalah menggunakan processor dengan kecepatan diatas 3.0 Ghz, dan kamera webcam dengan kemampuan resolusi hingga 5 MP.

Mengatur hyperparameter yang terbaik untuk pengenalan masker wajah yang lebih

baik, sehingga dapat memiliki akurasi yang tinggi dan lebih baik.

Dataset yang digunakan untuk data training perlu diperhatikan pada kualitas image, disarankan tidak memiliki kualitas yang pecah saat dilakukan zoom face, hal ini berpengaruh juga pada hasil training datanya. Posisi kamera webcam pada sistem deteksi pemakaian masker ini diletakkan pada tempat yang dapat melihat keseluruhan wajah. Mengambil dataset untuk masker wajah dengan jenis masker yang beragam, sehingga proses mendeteksi dan mengenali pemakaian masker wajah lebih baik, akurat dan tepat. Untuk melakukan face mask detection secara multiface dapat ditentukan posisi wajah yang sama, sehingga pada proses deteksi pemakaian masker wajah dapat dikenali semua wajah lebih dari satu.

E. DAFTAR PUSTAKA

- Ahmadi, Saputra, A. C., & Lestari, A. (2021). Rancang Bangun Aplikasi New Normal Covid- 19 Deteksi Penggunaan Masker. *Jurnal Teknologi Informasi*, 15(2), 199–209.
- Baay, M. N., Irfansyah, A. N., & Attamimi, M. (2021). Sistem Otomatis Pendeteksi Wajah Bermasker Menggunakan Deep Learning. *Jurnal Teknik ITS*, 10(1). <https://doi.org/10.12962/j23373539.v10i1.59790>
- Bagus Setiawan, A., Rachmawati, W., Taufiq Arrahman, A., Natasyah, N., & Fadil, F. N. S. (2021). Aplikasi Monitoring Stok Barang Berbasis Web Pada PT. Intermetal Indo Mekanika. *ADI Bisnis Digital Interdisiplin Jurnal*, 2(2), 1–6. <https://doi.org/10.34306/abdi.v2i2.254>
- Boulila, W., Alzahem, A., Almoudi, A., Afifi, M., Alturki, I., & Driss, M. (2021). A Deep Learning-based Approach for Real-time Facemask Detection. *Proceedings - 20th IEEE International Conference on Machine Learning and Applications, ICMLA 2021, DL*, 1478–

1481.
<https://doi.org/10.1109/ICMLA52953.2021.00238>
- Fadjeri, A., Saputra, B. A., Adri Ariyanto, D. K., & Kurniatin, L. (2022). Karakteristik Morfologi Tanaman Selada Menggunakan Pengolahan Citra Digital. *Jurnal Ilmiah SINUS*, 20(2), 1.
<https://doi.org/10.30646/sinus.v20i2.601>
- Luthfillah Ahmad, F., Nugroho, A., & Alfa Faridh Suni. (2021). Deteksi Pemakai Masker Menggunakan Metode Haar Cascade Sebagai Pencegahan COVID 19. *Edu Elekrika Journal*, 10(1), 13–18.
- Septiana, T., Puspita, N., Fikih, M. Al, & Setyawan, N. (2020). Face Mask Detection Covid-19 Using Convolutional Neural Network (Cnn). *Seminar Nasional Teknologi Dan Rekayasa (SENTRA) 2020*, 3, 27–32.
- Setyadi, A., Kallista, M., & Setianingsih, C. (2022). *Skripsi : Deteksi Social Distancing Dan Penggunaan Masker Di Restoran Dengan Algoritma Convolutional Neural Network (CNN)*. Bandung : Universitas Telkom.
- Sugiyono. (2017). *Metode Penelitian Bisnis: Pendekatan Kuantitatif, Kualitatif, Kombinasi dan R&D*. Bandung : Alfabeta.
- Wirawan, I. G. B., I Gede Putu Wirarama Wedashwara, & Ahmad Zafrullah Mardiansyah. (2022). Sistem IoT Protokol Kesehatan Depan Toko Menggunakan Raspberry Pi Camera dan Haar Cascade Classifier. *Journal of Computer Science and Informatics Engineering (J-Cosine)*, 6(1), 30–38.
<https://doi.org/10.29303/jcosine.v6i1.411>