

Deteksi Wajah Bermasker Menggunakan Webcam dan AWS EC2 Berbasis Raspberry Pi

Maitsa Nabila[#], Rika Idmayanti[#], Indri Rahmayuni[#]

[#]Jurusan Teknologi Informasi, Politeknik Negeri Padang, Limau Manis, Padang, 25164, Indonesia
E-mail : rikaidmayanti@pnp.ac.id, indri@pnp.ac.id

ABSTRACTS

In the COVID-19 pandemic, masks are an important necessity for the community. The use of masks is very important to prevent the transmission of COVID-19. During the current pandemic, the Padang State Polytechnic campus allows students to visit the library on the terms and conditions that students must comply with health protocols. However, there are still many students who do not comply with health protocols, specially in using masks. Therefore, a tool that can detect masks is needed to protect/prevent people who do not wear masks from wearing masks when visiting the library in order to prevent the spread of COVID-19. The face detection system is masked or not using python version 3 with the OpenCV library, and Tensorflow. This system uses a webcam, Raspberry Pi 3 Model B, and Buzzer. The webcam is used to capture the image of the mask user. The captured image will be processed and classified on the Raspberry Pi 3 Model B. A buzzer will sound if the classification result is a face without a mask. The tool will be equipped with displaying the results of monitoring library visitor data in the form of a website using AWS EC2 as the infrastructure.

ABSTRAK

Pada pandemi COVID-19, masker adalah sebuah kebutuhan penting bagi masyarakat. Penggunaan masker sangat penting untuk mencegah penularan COVID-19. Pada masa pandemi seperti saat ini, kampus Politeknik Negeri Padang mengizinkan mahasiswa untuk mengunjungi perpustakaan dengan syarat dan ketentuan mahasiswa harus mematuhi protokol kesehatan. Namun masih banyak mahasiswa yang tidak mematuhi protokol kesehatan terutama dalam menggunakan masker. Maka dari itu diperlukan alat yang dapat mendeteksi masker untuk menjaga/mencegah orang-orang yang tidak menggunakan masker agar menggunakan masker saat mengunjungi perpustakaan demi menghambat penyebaran dari COVID-19. Sistem pendeteksian wajah bermasker atau tidak menggunakan python versi 3 dengan library OpenCV, dan Tensorflow. Sistem ini menggunakan webcam, Raspberry Pi 3 Model B, dan Buzzer. Webcam digunakan untuk menangkap citra pengguna masker. Citra yang telah ditangkap akan diolah dan diklasifikasikan di Raspberry Pi 3 Model B. Buzzer akan berbunyi bila hasil pengklasifikasian adalah wajah tidak bermasker. Alat akan dilengkapi dengan menampilkan hasil monitoring data pengunjung perpustakaan berupa website dengan menggunakan AWS EC2 sebagai infrastrukturnya.

KATA KUNCI

*Mask Detection,
Maskless Detection,
OpenCV,
Python,
TensorFlow,
Raspberry Pi,
Webcam,
Website Monitoring,
AWS EC2*

1. PENDAHULUAN

Sejak awal 2020, penyebaran coronavirus COVID-19 telah menyebabkan kondisi pandemik di seluruh dunia. Pandemi ini menyebabkan gangguan kehidupan masyarakat di setiap negara di dunia, dan Indonesia adalah salah satu negara yang terdampak pandemik ini. Sejak bulan Mei 2020, Kementerian Kesehatan Indonesia telah mensosialisasikan regulasi baru terkait kebiasaan hidup sehat New Normal dalam rangka usaha mencegah

penyebaran virus COVID-19 ini di dalam masyarakat [1]. Salah satu regulasi ini adalah kewajiban pemakaian masker bila seseorang harus melakukan kegiatan di luar rumah dan memasuki suatu kawasan tertentu yang memiliki lokasi tertutup dan dengan jumlah pengunjung yang cukup banyak seperti sekolah, kantor, tempat perbelanjaan dan lainnya. Walaupun terdapat pembatasan jumlah pengunjung di lokasi-lokasi yang padat tersebut, kemungkinan terjadinya penyebaran virus di tempat-tempat yang ramai masih sangat tinggi sehingga penggunaan masker saat ini telah menjadi keharusan.

Pada masa pandemi seperti saat ini, kampus Politeknik Negeri Padang mengizinkan mahasiswa untuk mengunjungi perpustakaan dengan syarat dan ketentuan mahasiswa harus mematuhi protokol kesehatan demi menghambat penyebaran dan penularan dari COVID-19. Namun masih banyak mahasiswa yang tidak mematuhi protokol kesehatan terutama dalam menggunakan masker. Banyak mahasiswa yang tidak menggunakan masker atau yang salah dalam menggunakan masker seperti, menggunakan masker di bawah dagu, menggunakan masker hanya menutupi mulut, dan menggantungkan masker di leher saja. Maka dari itu diperlukan alat yang dapat mendeteksi masker untuk menjaga/mencegah orang-orang yang tidak menggunakan masker agar menggunakan masker saat mengunjungi perpustakaan demi menghambat penyebaran dan penularan dari COVID-19.

Alat ini akan dibuatkan sistem deteksi masker (Face Mask Detection) untuk mengenali wajah bermasker atau tidak. Untuk deteksi wajah bermasker menggunakan konsep Computer Vision yang merupakan ilmu yang mempelajari kemampuan mesin/komputer dalam melihat hingga mampu meng-ekstrak informasi dari sebuah gambar. Salah satu open source dari computer vision adalah OpenCV.

OpenCV adalah sebuah library untuk pemrograman yang digunakan untuk mengolah gambar dan video hingga kita mampu meng-ekstrak informasi didalamnya. Pada sistem pendeteksian wajah bermasker atau tidak menggunakan python versi 3 dengan library OpenCV, Tensorflow, serta Keras.

Alat dapat mendeteksi masker secara realtime yang terdiri atas beberapa komponen yaitu Raspberry Pi 3 model B untuk membangun sistem, webcam sebagai mata/sensor, dan Buzzer sebagai peringatan. Raspberry Pi merupakan mini komputer yang berukuran sebesar kartu kredit, Raspberry Pi biasa digunakan untuk berbagai keperluan, seperti security, server, dan game [3]. Alat akan dibuat dengan menampilkan hasil monitoring data pengunjung perpustakaan berupa web dengan menggunakan AWS EC2 sebagai infrastrukturnya

2. METODOLOGI PENELITIAN

Adapun penelitian terdahulu berguna sebagai acuan dalam proses pengembangan sistem yang akan dibuat dan sebagai referensi-referensi terkait sistem yang akan dibuat. Seperti Penelitian ini dilakukan oleh Muhammad Abdul Rahman Irham Harfi dan Dedi Ari Prasetya dengan judul “Prototipe Pendeteksi Masker Pada Ruang Wajah Masker untuk Kendali Pintu Otomatis Berbasis Deep Learning Sebagai Pencegahan Penularan Covid-19. [4] Dan juga penelitian ini dilakukan oleh Friendly, Zakaria Sembiring, Habibi Ramdani Safitri dengan judul “Deteksi Wajah Bermasker Berbasis Tensorflow-Keras Untuk Pengendalian Gerbang Akses Masuk Menggunakan Raspberry Pi4”. Penelitian ini bertujuan untuk merancang sebuah pengendalian gerbang akses masuk otomatis yang dapat membedakan orang atau wajah dengan bermasker dan wajah tidak bermasker. Sistem ini akan diimplementasikan pada Raspberry [5].

2.1. Deep Learning

Deep learning merupakan bagian (subset) dan pengembangan dari machine learning. Deep learning terdiri dari beberapa lapisan tersembunyi (hidden layers) yang juga merupakan bagian dari kecerdasan buatan (artificial intelligence). Metodologi deep learning menerapkan transformasi non-linier dan abstraksi model tingkat tinggi dalam database yang besar. Perkembangan yang pesat pada arsitektur deep learning di berbagai bidang telah memberikan kontribusi signifikan di dalam kecerdasan buatan (artificial intelligence). Hal tersebut ditandai dengan banyaknya algoritma-algoritma pada deep learning yang digunakan di berbagai aplikasi.[9]

Konsep dari deep learning banyak memberikan keuntungan dan semakin berkembang dalam beberapa tahun terakhir ini. Deep learning banyak digunakan dalam berbagai aplikasi kehidupan nyata. Sebagai contoh, dalam pemrosesan gambar digital (digital image processing), grayscale image coloring, biometric dan lain sebagainya. Penerapan algoritma deep learning seperti pewarnaan (coloring) dapat dilakukan secara otomatis oleh komputer. Library-library di dalam bahasa pemrograman python yang bisa digunakan untuk aplikasi deep learning yaitu library dlib (deep learning library), tensorflow, keras, theano, pytorch dan lain-lain.

2.2. Computer Vision

Computer vision menjadikan sebuah komputer “act like human sight”, sehingga komputer mempunyai kemampuan mendekati manusia dalam mendapatkan informasi secara visual. Kelebihan yang dimiliki dalam computer vision antara lain adalah:

1. 3D Interface: Menerjemahkan aksi-aksi 2D dari 3D yang di lihat.
2. Interpreting: Menerjemahkan gerakan-gerakan.

3. Recognition: Menempatkan label pada objek.
4. Object Detection : Mengenali suatu benda yang terdapat pada scane untuk melihat batasannya.
5. Description: Menugaskan properti pada objek.

2.3. Deteksi Wajah

Deteksi wajah merupakan bagian dari teknik biometrik yang sangat membantu dalam mengidentifikasi wajah seseorang. Deteksi wajah banyak digunakan untuk sistem keamanan, pengidentifikasian tindakan kriminal dan terdapat pada perangkat - perangkat mobile. Tujuan dari sistem deteksi wajah adalah untuk mengurangi terjadinya kesalahan ketika sistem dilakukan secara manual. Deteksi wajah dilakukan dengan menggunakan library OpenCV.

OpenCV (Open Source Computer Vision Library) adalah visi komputer open- source dan perpustakaan perangkat lunak pembelajaran mesin. OpenCV dibangun untuk menyediakan infrastruktur umum untuk aplikasi visi komputer dan untuk mempercepat penggunaan persepsi mesin dalam produk komersial.[15]

Algoritma ini dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia dalam video, melacak pergerakan kamera, melacak objek bergerak, mengekstrak model objek 3D, menghasilkan point cloud 3D dari kamera webcam stereo, menyatukan gambar untuk menghasilkan resolusi tinggi gambarseluruh pemandangan, temukan gambar serupa dari database gambar, hapus mata merah dari gambar yang diambil menggunakan flash, ikuti gerakan mata, kenali pemandangan dan buat penanda untuk melapisinya dengan augmented reality, dll

Bahasa Pemrograman yang digunakan pada umumnya untuk deteksi wajah adalah python. Python merupakan bahasa pemrograman tingkat tinggi yang ditafsirkan, berorientasi objek, dengan semantik dinamis. Struktur data bawaan tingkat tinggi, dikombinasikan dengan pengetikan dinamis dan pengikatan dinamis, membuatnya sangat menarik untuk Pengembangan Aplikasi Cepat, serta untuk digunakan sebagai bahasa skrip atau lem untuk menghubungkan komponen yang ada bersama-sama

2.4. TensorFlow

TensorFlow adalah sebuah platform end-to-end untuk melakukan machine learning. TensorFlow memiliki library yang luas dan fleksibel. Salah satu fitur library pada TensorFlow adalah CNN. TensorFlow dapat dijalankan diberbagai macam platform termasuk Raspberry Pi tanpa mengorbankan banyak performa dan kecepatan.[6]

TensorFlow adalah pustaka perangkat lunak aliran data open source gratis dan pustaka pemrograman yang dapat dibedakan yang dapat mencakup sejumlah tugas. TensorFlow adalah kerangka kerja generasi kedua Google Brain, untuk analisis dan pengembangan pada Google. Pada 11 Februari, versi 1.0.0 dirilis, TensorFlow dapat beroperasi di beberapa CPU dan GPU (dengan ekstensi CUDA dan SYCL opsional pada unit grafis untuk komputasi tujuan umum), sedangkan Kode Referensi berjalan di komputer individu.[5]

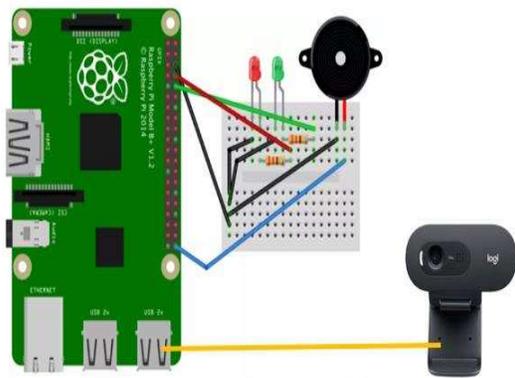
2.5. AWS

AWS (Amazon Web Service) menyediakan layanan komputasi awan, dimana setiap fungsi yang ada di dalamnya bisa diakses dengan panggilan Web Service. Sedangkan Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang memberikan kapasitas komputasi yang aman dan berukuran fleksibel di cloud. Amazon EC2 dirancang untuk membuat komputasi cloud berskala web lebih mudah bagi pengembang. Antarmuka layanan web sederhana Amazon EC2 memungkinkan pengguna mendapatkan dan mengonfigurasi kapasitas dengan friksi minimal.

Sistem yang dibangun merupakan sistem pendeteksi wajah bermasker di Perpustakaan Politeknik Negeri Padang. Sistem ini berfungsi melakukan pendeteksian wajah bermasker atau tidak bermasker. Apabila tidak menggunakan masker maka sistem akan memberikan peringatan. Sistem akan dibangun menggunakan raspberry pi yang digunakan untuk pengenalan wajah bermasker dengan menggunakan kamera sebagai penginputan wajah dan buzzer sebagai peringatan(warning). Untuk mengenali wajah bermasker Raspberry Pi yang merupakan mini computer diisi dengan program Python dengan library OpenCV dan Tensorflow yang dibuat untuk menjalankan sistem dan melakukan pemrosesan baik training, prediction, dan detection objek. Setelah itu, data pengujung yang menggunakan masker atau tidak akan disimpan di database dan outputnya akan terlihat pada web. Amazon Web Service EC2 sebagai database dan web server berbasis cloud computing untuk menampung data akan ditampilkan melalui web sederhana.

2.6 Perancangan Alat

Perancangan rangkaian alat menggunakan konfigurasi pin dari Raspberry Pi yang digunakan dapat dilihat pada gambar berikut:

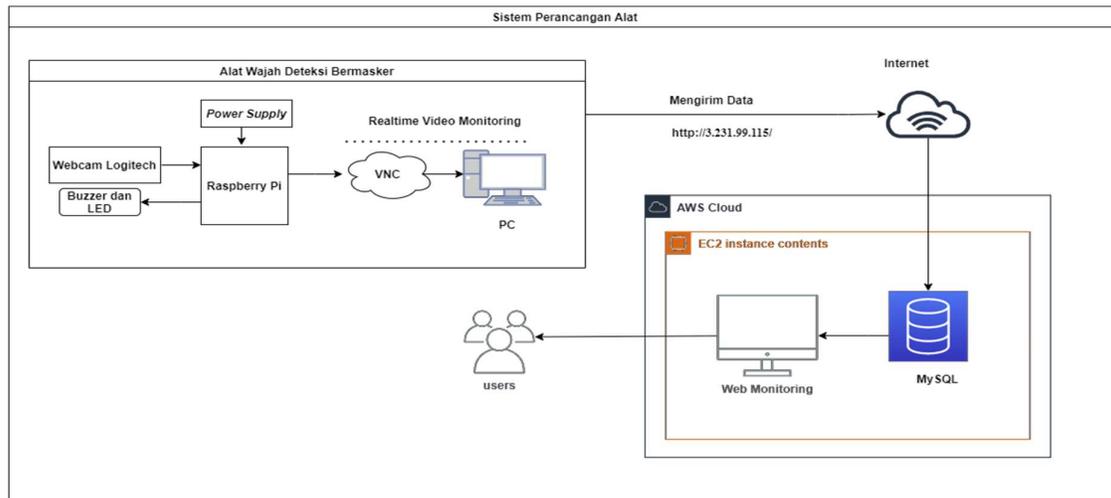


GAMBAR 1. Rangkaian Alat Deteksi Wajah Bermasker

TABEL 1. Penjelasan Rangkaian Alat

| No | Komponen | Keterangan |
|----|------------------------|--|
| 1 | LED | - Tambahkan resistor 330 ohm antara pin (+) LED. - Pin (+) LED merah dihubungkan dengan GPIO14 dan pin (+) LED hijau dihubungkan dengan GPIO15. - Pin (-) LED akan dihubungkan dengan GND. |
| 2 | Buzzer | - Pin (+) Buzzer dihubungkan dengan GPIO 21. - Pin (-) Buzzer akan dihubungkan dengan GND. |
| 3 | Webcam | Dihubungkan dengan port USB. |
| 4 | Laptop/ komputer | Dihubungkan dengan port LAN |
| 5 | Raspberry Pi 3 Model B | Dihubungkan ke LED, Buzzer, Webcam, dan komputer |

2.7. Blok Diagram Sistem

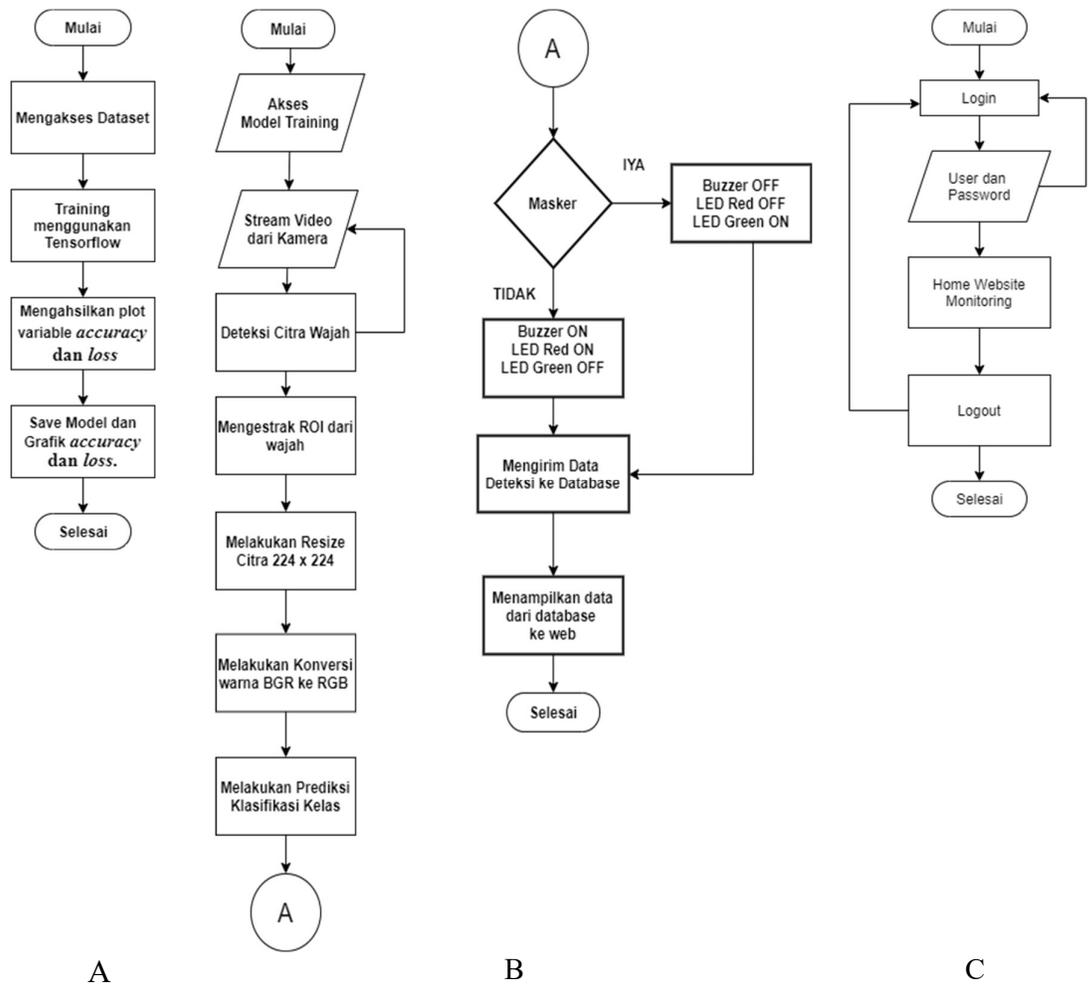


GAMBAR 2. Block Diagram Sistem

Perancangan sistem pada pendeteksi wajah bermasker di perpustakaan Politeknik Negeri Padang sesuai dengan gambar 2 yaitu menggunakan Raspberry Pi yang akan diisi dengan program Python dengan Library OpenCV dan Tensorflow yang dibuat untuk menjalankan sistem dan melakukan pemrosesan baik training, prediction, dan detection objek. Kemudian webcam sebagai inputan wajah akan memproses lagi pada Raspberry Pi untuk memprediksi apakah menggunakan masker atau tidak. Apabila dari hasil prediksi yang dibandingkan dengan dataset yang telah dilakukan training maka Raspberry Pi akan memberi perintah pada Buzzer dan LED. Lalu untuk monitoring video secara realtime dapat menggunakan VNC Viewer yang terinstall di Laptop/PC. Setelah itu, data pengujung yang menggunakan masker atau tidak akan disimpan di database dan outputnya akan terlihat pada web monitoring. Untuk website monitoring webservice dan database menggunakan Amazon Web Service EC2 berbasis cloud computing

2.8. Flowchart

Sebelum sistem mengenali wajah bermasker atau tidak, sistem akan melakukan training terlebih dahulu. Training dataset berfungsi untuk membuat data model klasifikasi wajah bermasker atau tidak pada dataset yang berisi data gambar wajah bermasker dan gambar wajah tidak bermasker. Pertama sistem akan mengakses dataset kemudian dataset akan ditraining menggunakan TensorFlow. Training terhadap dataset akan menghasilkan sebuah diagram yaitu Loss dan Accuracy. Hasil train akan disimpan pada disk/memory dalam bentuk file model dan grafik accuracy dan loss.



GAMBAR 3. A. Flowchart Training Dataset B. Flowchart Rancangan Sistem C. Flowchart Website Monitoring

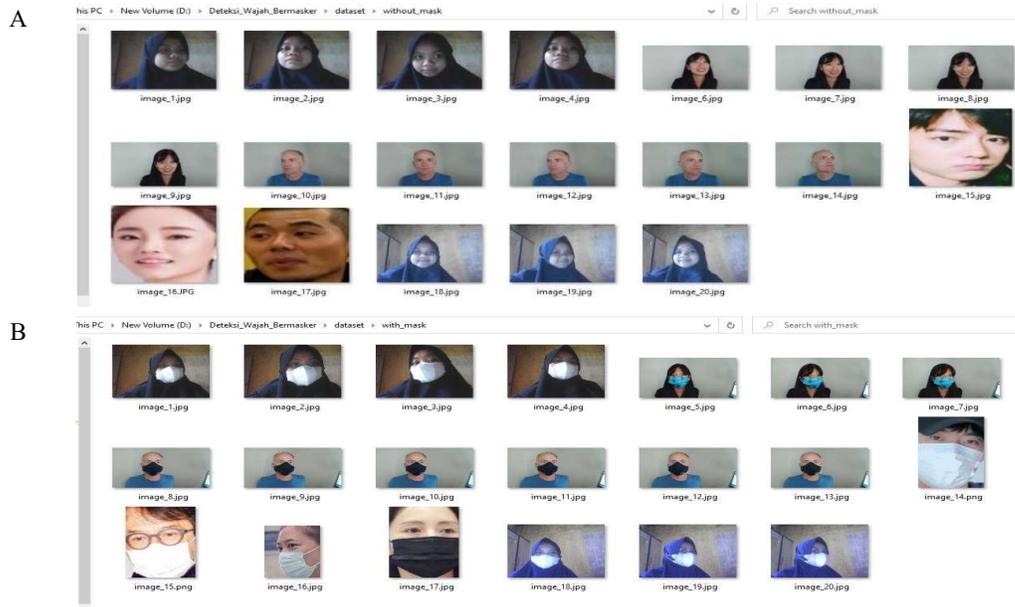
Setelah webcam aktif, webcam akan menangkap citra wajah kemudian mengekstrak ROI wajah. Secara default OpenCV membaca gambar dalam BGR. Maka objek wajah akan dikonversi dari BGR ke RGB, fungsinya untuk mendapatkan citra yang sebenarnya ketika dibaca oleh OpenCV. Setelah dikonversi objek wajah dilakukan resize menjadi ukuran 224 x 224. Setelah itu sistem akan memprediksi wajah apakah menggunakan masker atau tidak, untuk klasifikasi menggunakan masker atau tidak dilakukan dengan menggunakan library OpenCV dan Tensorflow.

Apabila hasil pendeteksian adalah tidak menggunakan masker maka sistem akan memberi perintah mengaktifkan Buzzer dan LED Merah. Dan sebaliknya jika menggunakan masker maka sistem akan memberi perintah mengaktifkan LED Hijau, mematikan Buzzer dan LED Merah. Setelah itu, data pengujung yang menggunakan masker atau tidak akan disimpan di database dan outputnya akan terlihat pada web monitoring. Basis data dibutuhkan untuk memenuhi kebutuhan sistem yang akan dirancang, maka diperlukan sebuah database yang berisikan kumpulan data dan informasi yang tersimpan dan tersusun di dalam sebuah table, diantaranya tabel data dari Raspberry dan user.

3. HASIL DAN PEMBAHASAN

3.1. Training Model Wajah Bermasker

Pada tahap training ini menggunakan sekelompok data yang kemudian dibagi menjadi 2 variabel, yaitu `with_mask` sebanyak 20 gambar, dan `without_mask` sebanyak 20 gambar. Kumpulan dataset tersebut ditunjukkan pada Gambar 4.A dan Gambar 4.B



GAMBAR 4. A. Dataset Tanpa Masker B. Dataset Menggunakan Masker

Hal yang penting pertama kali adalah harus dilakukan dalam program adalah melakukan import library yang dibutuhkan.

Import Library

```
# import the necessary packages from tensorflow.keras.preprocessing.image
import ImageDataGenerator from tensorflow.keras.applications
import MobileNetV2 from tensorflow.keras.layers
import AveragePooling2D from tensorflow.keras.layers
import Dropout from tensorflow.keras.layers import Flatten from tensorflow.keras.layers
import Dense from tensorflow.keras.layers import Input from tensorflow.keras.models
import Model from tensorflow.keras.optimizers import Adam from tensorflow.keras.applications.mobilenet_v2
import preprocess_input from tensorflow.keras.preprocessing.image
import img_to_array from tensorflow.keras.preprocessing.image
import load_img from tensorflow.keras.utils
import to_categorical from sklearn.preprocessing
import LabelBinarizer from sklearn.model_selection
import train_test_split from sklearn.metrics
import classification_report from imutils
import paths
import matplotlib.pyplot as plt
import numpy as np import argparse import os
```

Pada tahap ini, Dataset tersebut dilakukan pelatihan dengan menggunakan Tensorflow telah menyediakan fungsi MobileNetV2 yang merupakan arsitektur yang menerapkan CNN yang merupakan versi terbaru dari MobileNet dalam arsitektur CNN(Convolutional Neural Network) di mana memiliki memory yang kecil sehingga dapat di aplikasikan dalam Raspberry Pi. Ukuran yang digunakan adalah 224 pixel. Arsitektur ini akan digunakan sebagai input pada CNN.

Input

```
# load the MobileNetV2 network, ensuring the head FC layer sets are
# left off
baseModel = MobileNetV2(weights="imagenet", include_top=False, input_tensor=Input(shape=(224, 224,
3)))
```

Untuk output, kita akan menggunakan model yang sama dengan arsitektur MobileNetV2 diatas. Terdapat 2 aktivasi untuk networknya yakni “relu” dengan kerapatan 128 layer dan “softmax” dengan kerapatan 2 layer

Output

```
# construct the head of the model that will be placed on top of the
# the base model
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)
```

Sistem ini menggunakan learning rate 10-4 , pemberian nilai learning rate tidak boleh melebihi 10-4 karena akan terjadi overfitting. Fungsi dari learning rate untuk meningkatkan efektifitas dari parameter tingkat pembelajaran. Dimana tingkat pembelajaran merupakan parameter yang berfungsi meningkatkan kecepatan belajar dari backpropagation sebagai fungsi training.

Arsitektur dilatih selama 20 epoch (iterations) dan batch size sebesar 32. Batch size adalah jumlah sampel data yang akan disebar ke neural networks. Epoch adalah parameter untuk seluruh dataset yang sudah melalui proses pelatihan pada model untuk sekali putaran penuh, dan jika mengatur dengan 20 epoch artinya seluruh dataset mengalami proses pelatihan sebanyak 20 kali. Setelah dilakukan training model sistem akan menyimpan dalam bentuk file model dan grafik (training loss dan accuracy) dari hasil proses latih dataset masker detector.

3.2. Proses Deteksi Wajah Bermasker

Proses deteksi wajah bermasker mempunyai beberapa tahap yang dilakukan sebelum dapat melakukan pengenalan wajah bermasker atau tidak. Yang pertama dilakukan adalah mengambil library tensorflow, opencv dan library pendukung.

Import Library

```
# import the necessary packages from gpiozero
import Buzzer, LED from tensorflow.keras.applications.mobilenet_v2
import preprocess_input from tensorflow.keras.preprocessing.image
import img_to_array from tensorflow.keras.models import load_model from imutils.video import VideoStream
import numpy as np
import imutils import time import cv2 import os
import mysql.connector import datetime
import time
```

Selanjutnya memuat model pengenalan wajah bermasker dan model pengenalan wajah. Berikut penjelasan dari source code dari deteksi wajah bermasker :

a. Deteksi Wajah bermasker dan Tidak bermasker

Berikut adalah source code yang akan mendeteksi dan mengkategorikan apakah wajah yang ditangkap kamera merupakan wajah bermasker ataupun wajah tanpa masker. Program ini akan menggunakan data model yang dihasilkan pada program pelatihan sebelumnya

Import Library

```
# import the necessary packages from gpiozero
import Buzzer, LED from tensorflow.keras.applications.mobilenet_v2
import preprocess_input from tensorflow.keras.preprocessing.image
import img_to_array from tensorflow.keras.models import load_model from imutils.video import VideoStream
import numpy as np
import imutils import time import cv2 import os
import mysql.connector import datetime
import time
```

Pada source code face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB) fungsinya wajah yang terdeteksi akan dikonversi dari BGR ke RGB, fungsinya untuk mendapatkan citra wajah yang sebenarnya ketika dibaca oleh OpenCV. Setelah dikonversi wajah dilakukan resize menjadi ukuran 224 x224. Pada source code

terdapat `preds = maskNet.predict(faces, batch_size=32)` untuk mengenali wajah dan memprediksi wajah apakah wajah yang ditangkap kamera menggunakan masker atau tidak.

- b. Treaming Webcam untuk deteksi wajah bermasker
 Berikut source code untuk streaming dan prediksi wajah secara berulang yang ditangkap kamera merupakan wajah bermasker ataupun wajah tanpa masker.
 Pada source code terdapat `vs = VideoStream(src=0).start()` fungsinya untuk mengaktifkan webcam yang terhubung dengan raspberry pi dan streaming video. Kemudian setelah kamera aktif webcam akan mendeteksi citra wajah kemudian memanggil fungsi `detect_and_predict_mask(frame,faceNet,maskNet)` untuk memprediksi wajah bermasker atau tidak.
 Pada source code `cv2.putText(frame, label, (startX-50, startY-10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, color, 2)` `cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)` memiliki fungsi apabila yang diprediksi wajah bermasker maka wajah akan diberi kotak berwarna hijau diiringi pesan teks "Masker Terdeteksi :)" atau kotak berwarna merah diiringi pesan teks "No Mask, Gunakan Masker Anda!!!" apabila yang diprediksi wajah tidak bermasker. Setelah itu akan dipanggil fungsi `BuzzerLed(status)` untuk memberi perintah pada buzzer dan LED. Data dari prediksi akan disimpan dalam database dan dikirim ke server dengan memanggil fungsi `insert_data(status,jam,tanggal)` dengan informasi berupa status, jam, dan tanggal prediksi .
- c. Buzzer dan LED
 Berikut source code Buzzer dan LED pada deteksi wajah bermasker.
 Pada source code `buzzer = Buzzer(21)` `red = LED(14)` `green = LED(15)` fungsinya untuk mengkonfigurasi GPIO pin pada raspberry pi seperti Buzzer yang dihubungkan dengan GPIO pin 21 dan LED yang dihubungkan dengan GPIO pin 14 dan 15.
 Apabila status sama dengan mask maka Buzzer dan LED Red akan aktif serta LED Green akan tidak aktif. Sebaliknya jika status sama dengan no maka Buzzer dan LED Red akan tidak aktif serta LED Green akan aktif. Pada bagian elif `status=="mask" and status=="no"` memiliki fungsi apabila ada dua objek wajah, yang mana satu menggunakan mask dan satu lagi tidak menggunakan mask. Maka Buzzer dan LED Red akan tetap aktif begitupun LED Green akan tetap tidak aktif.
- d. Simpan data ke Database
 Berikut source code untuk simpan data ke Database yang ada pada EC2. Pada source code `mysql.connector.connect (host='xx.xx.xx.xx',database='dbFaceMask',user='**',password='**')` digunakan untuk koneksi database dan source code `mySql_insert_query` digunakan untuk memasukan data kedalam tabel pada database.

3.3. Hasil Pengujian



GAMBAR 5. Grafik Training Loss dan Accuracy Dataset

Pengujian Training Model Wajah Bermasker adalah pengujian dari source code yang telah dibuat. Hasil model akan di simpan di dalam file `Deteksi_wajah_Bermasker` dengan nama file `mask_detectorNew.model` dan gambar grafik disimpan dengan nama `mymodelplot.png`. Grafik dari hasil proses latih dataset masker detektor ditunjukkan pada gambar 5.



A



B

GAMBAR 6. A Realtime Video Monitoring Wajah Bermasker B. Realtime Video Monitoring Wajah Tidak Bermasker

4. KESIMPULAN

Hasil Penelitian disimpulkan bahwa Sistem deteksi wajah bermasker atau tidak di Perpustakaan Politeknik Negeri Padang menggunakan Webcam dan Raspberry Pi dapat di realisasikan. Dengan adanya sistem deteksi wajah bermasker atau tidak di perpustakaan Politeknik Negeri Padang memudahkan dalam memonitoring penggunaan masker agar mengurangi penyebaran COVID-19. Sistem dapat mengenali wajah bermasker atau tidak bermasker secara realtime. Alat dapat mengaktifkan Buzzer dan LED Red ketika wajah tidak bermasker terdeteksi. Jika yang terdeteksi wajah bermasker alat dapat mengaktifkan LED Green. Alat dapat mengirim data dan ditampilkan melalui website menggunakan server AWS EC2. Tampilan data dari alat dapat ditampilkan baik secara tabel maupun secara grafik .

REFERENSI

- [1] Kementerian Kesehatan Republik Indonesia, "Keputusan Menteri Kesehatan Republik Indonesia Panduan Protokol Kesehatan Bagi Masyarakat di Tempat dan Fasilitas Umum Dalam Rangka Pencegahan dan Pengendalian Corona Virus Disease2019 (Covid-19)," No. Hk.01.07/Menkes/382/2020, 2021.
- [2] Anonim. 2021. <https://www.kemdikbud.go.id/main/blog/2020/12/perkuliah-an-dapat-dilakukan-secara-tatap-muka-dan-dalam-jaringan-tahun-2021>. (Diakses tanggal 30 Agustus 2021)
- [3] Anonim. 2021. <https://www.raspberrypi.org/>. (Diakses Tanggal 30 Agustus 2021)
- [4] Harfi, Muhammad Abdul Rahman Irham, and Dedi Ari Prasetya. "Prototipe Pendeteksi Masker pada Ruang Wajah Masker untuk Kendali Pintu Otomatis berbasis Deep Learning sebagai Pencegahan Penularan COVID-19." (2020).
- [5] Friendly, Sembiring, Z., & Safitri, H. R. (2020). DETEKSI WAJAH BERMASKER BERBASIS TENSORFLOW-KERAS UNTUK PENGENDALIAN GERBANG AKSES MASUK MENGGUNAKAN RASBERRY Pi4. JIKSTRA Vol. 2, 45-55.
- [6] Nugraraga, B. D., Fitriyah, H., & Syauby, D. (2021). Deteksi Orang Bermasker Medis Menggunakan Metode Convolutional Neural . Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2600-2609.
- [7] Puspita Putri, Tri Septiana Nadia. FACE MASK DETECTION COVID-19 USING CONVOLUTIONAL NEURAL NETWORK (CNN). Diss. Universitas Muhammadiyah Malang, 2021.
- [8] Anonim. 2021. <https://www.kemkes.go.id/folder/view/full-content/structure-faq.html>. (Diakses Tanggal 30 Agustus 2021)
- [9] Jurjawi, Imam. "Implementasi Pengenalan Wajah Secara Real Time untuk Sistem Absensi Menggunakan Metode Pembelajaran Deep Learning dengan Pustaka Open CV (Computer Vision)." (2020).
- [10] Anggraini, Wulan. Deep Learning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network (CNN) Dengan Tensorflow. Diss. UIN Ar-Raniry Banda Aceh, 2020.
- [11] Anton, Kurniawan. PERANCANGAN ALAT PENGAWAS RUANGAN MENGGUNAKAN RASPBERRY PI. Diss. Universitas Buddhi Dharma, 2018.
- [12] Christian, Frendy. "Modul Pembelajaran Raspberry Pi." Universitas Sanata Dharma: Yogyakarta (2017).
- [13] Aldi, Triputranda. MONITORING SUHU RUANGAN SERVER BERBASIS RASPBERRY PI MENGGUNAKAN SENSOR SUHU DHT11. Diss. Politeknik Negeri Sriwijaya, 2016.
- [14] Python (2021). Python. URL <https://www.python.org/doc/essays/blurb/> (accessed 15 Agustus 2021).
- [15] OpenCV Team (2021). OpenCV. URL <https://opencv.org/about/> (accessed 15 Agustus 2021).
- [16] D. S. Amelia, H. Amnur, and H. A. M. Mooduto, "Monitoring Suhu dan Kelembaban Tanah Serta Penyiraman Otomatis Buah Naga Berbasis AWS", jitsi, vol. 2, no. 3, pp. 90 - 96, Dec. 2021.
- [17] Anonim. 2021. <https://indonesiancloud.com/mengenal-cloud-computing/> (Diakses Tanggal 13 September)
- [18] Anonim. 2021. <https://aws.amazon.com/id/what-is-aws/> (Diakses Tanggal 1 September 2021)

- [19] Nugroho, Adi, and Techn Khabib Mustofa. "Implementasi Komputasi Awan Menggunakan Teknologi Google App Engine (GAE) dan Amazon Web Services (AWS)." *Jurnal Teknik Informatika* 1.1 (2012).
- [20] Anonim. 2021. <https://aws.amazon.com/id/ec2/?ec2-whats-new.sort-by=item.additionalFields.postDateTime&ec2-whats-new.sort-order=desc> (Diakses Tanggal 1 September 2021)
- [21] Arjuni, Sandy. "Perancangan dan Implementasi Proxy server dan Manajemen Bandwidth Menggunakan Linux Ubuntu Server." *Studi Kasus di Kantor Manajemen PT. Wisma Bumiputera Bandung*. Tugas Akhir. Tidak diterbitkan. Institut Teknologi Bandung: Bandung (2010).
- [22] Sitinjak, Daniel Dido Jantce TJ, and Jaka Suwita. "ANALISA DAN PERANCANGAN SISTEM INFORMASI ADMINISTRASI KURSUS BAHASA INGGRIS PADA INTENSIVE ENGLISH COURSE DI CILEDUG TANGERANG." *Insan Pembangunan Sistem Informasi dan Komputer (IPSIKOM)* 8.1 (2020).
- [23] Arunawati, Anindya Putri. OPTIMASI APACHE WEB SERVER MENGGUNAKAN VARNISH WEB CACHE DAN REVERSE PROXY NGINX. Diss. Universitas Negeri Semarang, 2020.
- [24] HAFSARI, R. T., & ISNANI, S. R. (2021). RANCANG BANGUN ALAT PENDETEKSI WAJAH DAN PENDETEKSI SUHU TUBUH OTOMATIS GUNA MEMINIMALISIR PENYEBARAN COVID-19. Makassar.