

# Pendeteksi Penggunaan Masker Wajah dengan ESP32 Cam Menggunakan OpenCV dan TensorFlow

Cokorda Gde Indra Raditya<sup>1</sup>, Putu Adhitya Santika Dharma<sup>2</sup>, Komang Agus Widyatmika<sup>3</sup>,  
I Nengah Suparta<sup>4</sup>, I Made Sumerta Yasa<sup>5</sup>, Anak Agung Ngurah Gde Saptaka<sup>6</sup>

[Submission: 21-03-2022, Accepted: 14-07-2022]

**Abstract**— The government's recommendation to wear masks daily is very important to avoid the Covid-19. This article analyzes the use of the ESP32 Cam as a mask detector, where proper use must cover the nose and mouth. The captured images can be retrieved and executed by a program created using Python. The OpenCV library is used to access the camera, while the TensorFlow library with training files in the datasheet performs mask detection. The experimental results of detecting masks in bright conditions show the fastest time is 1 second with an ideal distance of 1m and a maximum distance of 2.5m. On the other hand, in dark conditions, the fastest time is 2 seconds with an ideal distance of 1m and a maximum distance of 1.5m.

**Intisari**— Anjuran pemerintah untuk memakai masker setiap hari sangat penting untuk menghindari Covid-19. Artikel ini menganalisis penggunaan ESP32 Cam sebagai pendeteksi masker, di mana penggunaan yang benar harus menutupi hidung dan mulut. Gambar yang diambil dapat diproses dan dieksekusi oleh program yang dibuat menggunakan bahasa pemrograman Python. Library OpenCV digunakan untuk mengakses kamera, sedangkan library TensorFlow dengan file training dalam datasheet digunakan untuk menjalankan deteksi masker. Hasil eksperimen pendeteksian masker dalam kondisi terang menunjukkan waktu tercepat adalah 1 detik dengan jarak ideal 1m dan jarak maksimum 2,5m. Sebaliknya, dalam kondisi gelap, waktu tercepat adalah 2 detik dengan jarak ideal 1m dan jarak maksimum 1,5m.

**Kata Kunci**— covid-19, ESP32 Cam, masker, OpenCV, TensorFlow

## I. PENDAHULUAN

Menjaga kesehatan saat ini sangat berharga dan penting bagi kehidupan manusia. Kesehatan yang buruk akan mengganggu kehidupan sehari-hari [1]. Untuk itu, kesehatan sangat penting bagi semua orang, terutama di masa pandemi Virus Covid-19 sejak Maret 2019. Virus Covid-19 atau corona ini beresiko menular melalui udara, droplet atau percikan liur, serta lewat bersin dan batuk dari orang yang terinfeksi [2]. Oleh karena itu pemerintah menerapkan Protokol Kesehatan 5M bagi

<sup>1,2,3</sup>Mahasiswa, Program Studi Teknik Otomasi, Jurusan Teknik Elektro, Politeknik Negeri Bali, Jalan Bukit Jimbaran 80361 INDONESIA (e-mail: [cokgdeindra@gmail.com](mailto:cokgdeindra@gmail.com), [adhityasantikad@gmail.com](mailto:adhityasantikad@gmail.com), [mangcupliz@gmail.com](mailto:mangcupliz@gmail.com))

<sup>4,5,6</sup>Dosen, Program Studi Teknik Otomasi, Jurusan Teknik Elektro, Politeknik Negeri Bali, Jalan Bukit Jimbaran 80361 INDONESIA (e-mail: [nengahsupartast01@pnb.ac.id](mailto:nengahsupartast01@pnb.ac.id), [sumertayasa6@pnb.ac.id](mailto:sumertayasa6@pnb.ac.id), [sapteka@pnb.ac.id](mailto:sapteka@pnb.ac.id))

masyarakat, yaitu memakai masker, mencuci tangan pakai sabun dan air mengalir, menjaga jarak, menjauhi kerumunan, dan membatasi mobilitas dan interaksi [3]. Oleh sebab itu penting adanya sebuah sistem otomasi yang dapat mendeteksi penggunaan masker untuk menjaga orang yang akan memasuki ruangan dari paparan virus COVID-19 [4].

Penelitian sebelumnya tentang metode *Haar Cascade Classifier* untuk klasifikasi pengenalan wajah telah dilakukan oleh beberapa peneliti [5], [6], [7]. Namun penelitian tersebut tidak mencantumkan kecepatan dan kondisi cahaya dari pendeteksian yang dilakukan. Adapun penelitian mengenai TensorFlow yakni library Python untuk pengenalan objek juga telah dilaporkan oleh beberapa peneliti [8], [9], [10]. Selanjutnya pada penelitian oleh Guntur Wahyono dkk. tentang modul ISD 1820 dijelaskan terkait penggunaan dari ISD 1820 setelah alat mendeteksi adanya gerakan [11].

Pendeteksi penggunaan masker pada peneliti ini dilakukan melalui kamera sehingga input data pengujian bersifat *real-time*. Dari rekaman data tersebut akan diuji apakah objek menggunakan masker atau tidak menggunakan masker. Prototipe ini menggunakan *computer vision* yang merupakan cabang ilmu dengan tujuan utama komputer dapat mengenali suatu objek yang sedang diamati atau observasi. Dasar utama dalam penelitian ini menggunakan sistem OpenCV untuk melakukan pendeteksian masker dan menggunakan library TensorFlow untuk melakukan training dengan kumpulan gambar yang terdiri dari dua *class* yaitu *mask* dan *no-mask*. Alat ini mendeteksi masker secara *real-time* yang terdiri dari ESP32 Cam sebagai sensor yang menangkap citra gambar, Arduino Uno sebagai pemberi *input* kepada servo, dan modul ISD1820. Berdasarkan studi literatur, belum ada artikel yang menuliskan penggunaan algoritma TensorFlow yang dipadukan dengan ESP32 Cam untuk pendeteksian masker pada wajah manusia.

## II. TINJAUAN PUSTAKA

### A. Mikrokontroler

Mikrokontroler merupakan gabungan dari keseluruhan sistem komputer yang dicetak menjadi sebuah *chip* yang di dalamnya terdapat mikroprosesor, I/O, memori serta ADC [12].

### B. Python

Python adalah salah satu bahasa pemrograman yang banyak digunakan untuk pengembangan kendali perangkat elektronik.



Python mendukung berbagai pengembangan teknologi saat ini, dalam teknologi pengolahan citra tersedia *library* seperti OpenCV, Numpy dan Matplotlib diimplementasikan secara internal pada pemrograman bahasa C sehingga analisis matematis dan rutinitas manipulasi data dilakukan secara efisien. Python digunakan untuk memproses citra gambar yang akan diproses dalam ESP32 Cam [13], [14].

### C. OpenCV

Teknik pengolahan citra mampu melakukan deteksi, pengenalan, klasifikasi terhadap suatu obyek yang ada di dalam ini. OpenCV merupakan salah satu *library* dalam pemrograman yang sering digunakan dalam Teknik pengolahan citra digital. *Library* tersebut bersifat terbuka dapat diunduh secara gratis pada situs resmi OpenCV.org. OpenCV juga mendukung ke beberapa bahasa pemrograman yang paling sering digunakan oleh *programmer* seperti bahasa C, Java, Python [15]. Dukungan ke beberapa bahasa pemrograman tersebut menjadikan OpenCV lebih mudah dan banyak digunakan dalam pengembangan beberapa perangkat lunak saat ini [8].

### D. Haar Cascade Classifier

*Haar Like Feature* yang dikenal sebagai *Haar Cascade Classifier*, di mana input dari setiap tahapan klasifikasi adalah merupakan output dari klasifikasi sebelumnya. Setiap citra yang melewati klasifikasi pertama akan menuju klasifikasi kedua, demikian seterusnya [16], [17]. Jika suatu citra dapat melewati setiap tingkat klasifikasi, maka dinyatakan sebagai citra wajah. Sedangkan citra yang tidak berhasil lolos dari klasifikasi akan langsung dieliminasi, dinyatakan sebagai bukan wajah dan tidak akan diproses kembali. Kelebihan metode ini yaitu komputasinya sangat cepat karena tidak bergantung pada setiap nilai piksel dari sebuah gambar, melainkan pada jumlah piksel dalam persegi.

### E. TensorFlow

TensorFlow merupakan sebuah *library* dari Bahasa pemrograman Python. TensorFlow adalah sebuah antarmuka yang digunakan untuk ekspresi algoritma pembelajaran mesin dan untuk eksekusi perintah dengan menggunakan informasi yang dimiliki tentang objek tersebut atau target yang akan dikenali, serta dapat membedakan objek satu dengan objek yang lainnya [9], [13]. *Library* TensorFlow ini menyatukan komputasi aljabar beserta teknik optimasi kompilasi, yang mendukung perhitungan [8]. TensorFlow menggunakan *graphics processing unit* (GPU) dengan efisien, juga menata dan memaksimalkan memori yang digunakan dengan data yang dipakai. *Library* TensorFlow sendiri memiliki kemampuan untuk menuliskan kesamaan kode serta mampu menjalankan *central processing unit* (CPU) dan GPU secara maksimal.

### F. ESP32 Cam

ESP32 Cam merupakan modul yang banyak digunakan dalam berbagai proyek, khususnya dengan Arduino. Modul ini adalah sebuah mikrokontroler lengkap yang terintegrasi serta dapat membuatnya bekerja secara mandiri [18]-[20]. Gambar modul ESP32 Cam ditunjukkan pada Gambar 1.



Gambar 1: ESP32 Cam

### G. Modul ISD1820

Modul ini merupakan *IC* yang dapat merekam suara dengan batas maksimal hanya 20 detik dengan kapasitas memori 3,2K yang langsung dihubungkan dengan *output speaker* 8ohm atau *speaker* aktif dan memerlukan tegangan 3,3V [11]. Gambar modul ISD1820 ditunjukkan pada Gambar 2.



Gambar 2: Modul ISD1820

### H. Arduino Uno

Arduino Uno merupakan mikrokontroler *embedded* dan *opensource* dan merupakan kombinasi dari perangkat keras, bahasa pemrograman dan *Integrated Development Environment*. Pada Arduino Uno terdapat ATMEGA328-8 bit sebagai komponen utama [19], [21], [22]. Gambar Arduino Uno ditunjukkan pada Gambar 3.



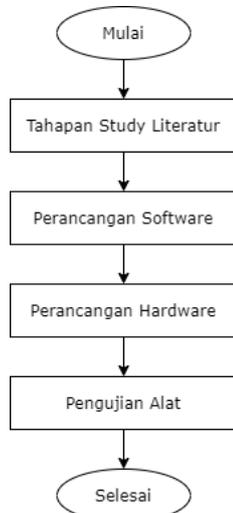
Gambar 3: Arduino Uno

### I. Motor Servo

Motor servo adalah aktuator putar dengan kontrol umpan balik loop tertutup. Motor ini dapat disesuaikan untuk menentukan dan mengamankan posisi sudut poros keluaran. Motor servo terdiri dari motor DC, serangkaian *gearbox* yang terpasang pada poros motor DC yang meningkatkan torsi motor servo. Potensiometer, yang resistansinya berubah saat motor bergerak, membantu untuk menentukan batas posisi rotasi poros motor servo [23].

### III. METODOLOGI

Penelitian ini dilakukan dengan metode eksperimental untuk menguji sistem yang dirancang, baik *software* maupun *hardware*. Gambar 4 menampilkan diagram alir tahapan penelitian pembuatan alat pendeteksi penggunaan masker wajah.

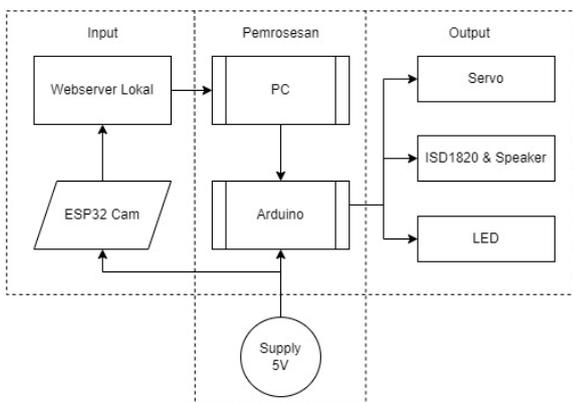


Gambar 4: Metodologi Penelitian

Tahap penelitian dimulai dari tahapan studi literatur, perancangan dan penyusunan *software*, perancangan dan pembuatan *hardware*, serta dilanjutkan dengan analisis dan kesimpulan. Tahap perancangan alat dilakukan penyusunan diagram blok dan *flowchart* sebagai acuan dalam proses pembuatan. Sedangkan pada tahap analisis dilakukan kajian terhadap jarak, dan waktu respon sensor serta output berupa pesan suara, nyala *LED* serta gerakan servo.

#### A. Diagram Blok

Sebelum dilakukannya perancangan pada *hardware* dan *software*, terlebih dulu ditentukan diagram blok kerja alat yang akan dibuat untuk mempermudah perancangan tersebut. Diagram blok dibagi menjadi 3 bagian yaitu *input* (ESP32Cam), pemrosesan (PC, Arduino Uno), dan *output* (Speaker, LED, Servo). Diagram blok tersebut dapat dilihat pada Gambar 5.

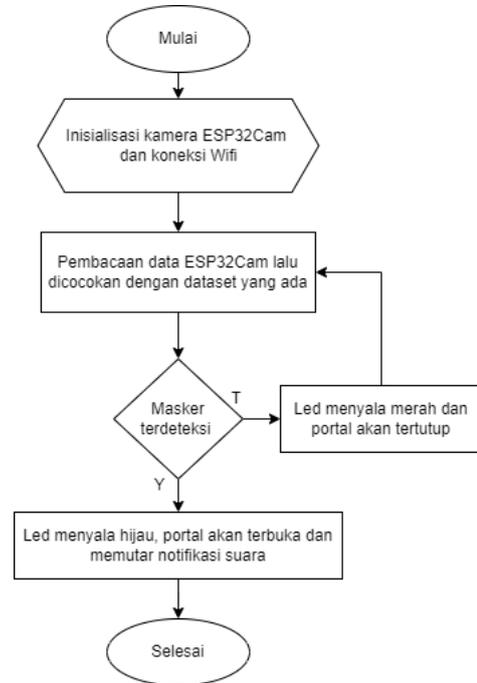


Gambar 5: Diagram Blok Rangkaian

Cokorda Gde Indra Raditya: Pendeteksi Penggunaan Masker ...

#### B. Flowchart

*Flowchart* atau diagram alir berfungsi untuk menggambarkan proses kerja dari alat pendeteksi penggunaan masker wajah dengan ESP32Cam menggunakan OpenCV dan Tensorflow. *Flowchart* dari cara kerja alat ini dapat dilihat pada Gambar 6.



Gambar 6: Flowchart

Dapat dilihat pada Gambar 6, proses pertama yang dilakukan adalah inisialisasi *port* dari sensor dan sistem. Selanjutnya dilakukan proses menghubungkan ESP32 Cam dengan internet. Ketika alat dapat terhubung dengan internet, maka alat akan mengirimkan pesan tanda terkoneksi pada serial monitor. Selanjutnya alat melakukan pembacaan data pada ESP32 Cam serta menampilkan data berikut perubahannya pada layar PC. Ketika alat mendeteksi adanya wajah tanpa masker, LED merah akan menyala tanda akses ditolak dan servo akan bergerak menutup. Demikian juga jika alat mendeteksi wajah dengan masker, maka LED akan menyala hijau menandakan akses diberikan dan servo akan bergerak membuka, lalu akan ada notifikasi suara yang mengajak untuk terus memakai masker.

#### C. Perancangan Software

Perancangan *software* bertujuan untuk membuat sistem pengenalan wajah yang memakai masker dengan citra ESP32 Cam. Langkah pertama yaitu membuat *dataset image* menggunakan *library* Tensorflow yang berisi kumpulan orang yang memakai masker dan yang tidak, selanjutnya foto tersebut digunakan untuk pencocokan data yang ditangkap oleh ESP32 Cam. Adapun *software* yang digunakan adalah Spyder (Anaconda3) dan Arduino IDE. Kode program yang dibuat sesuai dengan rancangan *flowchart* sebelumnya. Setelah kode



program selesai, lalu program Arduino diunggah ke board ESP32 Cam dan Arduino Uno. Sedangkan untuk program Python akan dijalankan langsung oleh PC/Laptop. Berikut adalah rancangan *software* dari alat yang dibuat. Tampilan program ditunjukkan pada Gambar 7 dan Gambar 8.

```

WifiCam | Arduino 1.8.13
File Edit Sketch Tools Help

WifiCam
1 #include <WebServer.h>
2 #include <WiFi.h>
3 #include <esp32cam.h>
4
5 const char* WIFI_SSID = "cuplis custom";
6 const char* WIFI_PASS = "cuplicustom13";
7
8 WebServer server(80);
9
10 static auto loRes = esp32cam::Resolution::find(640, 480);
11 static auto hiRes = esp32cam::Resolution::find(480, 320);
12
13 void
14 handleBmp()
15 {
16     if (!esp32cam::Camera.changeResolution(loRes)) {
17         Serial.println("SET-LO-RES FAIL");
18     }
19
20     auto frame = esp32cam::capture();
21     if (frame == nullptr) {
22         Serial.println("CAPTURE FAIL");
23         server.send(503, "", "");

```

Gambar 7: Tampilan Program Arduino IDE

```

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help

...oad\Face_Mask_Detection-main-20220117\003442-001\Face_Mask_Detection-main\test error coba espcam.py
test error coba espcam.py* controller.py test error.py controller.py

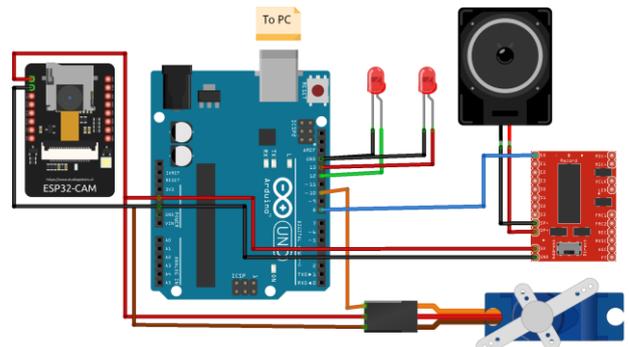
1 import warnings
2 warnings.filterwarnings('ignore')
3 import numpy as np
4 import cv2
5 import urllib.request
6 from keras.models import load_model
7 from controller import doorAutonate
8
9 facedetect = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
10 threshold=0.90
11
12 #cap=cv2.VideoCapture(0)
13 #cap.set(3, 640)
14 #cap.set(4, 480)
15
16 url='http://192.168.18.235/cam-1a.jpg'
17 cv2.namedWindow('gotcha', cv2.WINDOW_AUTOSIZE)
18
19 font=cv2.FONT_HERSHEY_COMPLEX
20 model = load_model('myTrainingModel.h5')
21
22 def preprocessing(img):
23     img=img.astype("uint8")
24     img=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
25     img=cv2.equalizeHist(img)
26     img = img/255
27     return img
28
29
30 def get_className(classNo):
31     if classNo==0:
32         return "Mask"
33     elif classNo==1:
34         return "No Mask"
35
36
37 while True:

```

Gambar 8: Tampilan Program Spyder (Anaconda3)

**D. Perancangan Hardware**

Rancangan *hardware* dari alat pendeteksi masker yang dibuat berupa *wiring diagram* dapat dilihat pada Gambar 9. Desain rancangan alat dibuat menggunakan aplikasi Fritzing.



Gambar 9: Rancangan Hardware

Konektivitas *pin* antar komponen pada *wiring diagram* rangkaian alat di atas dapat dilihat pada Tabel I.

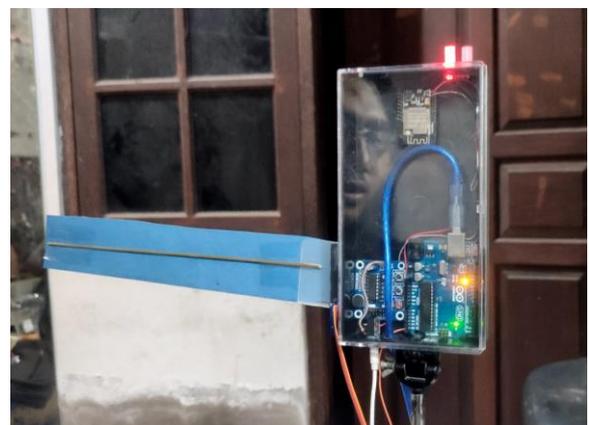
TABEL I  
KONEKSI PIN PADA RANGKAIAN

No	Nama Komponen	Pin Arduino Uno
1	ISD 1820 P-E	8
2	Servo	10
3	LED hijau	11
4	LED merah	12
5	VCC +5V	5V
6	GND	GND

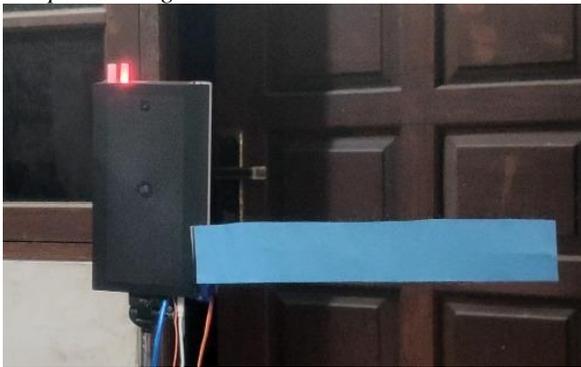
**IV. HASIL DAN PEMBAHASAN**

**A. Realisasi Hasil Perancangan Pendeteksi Penggunaan Masker Wajah**

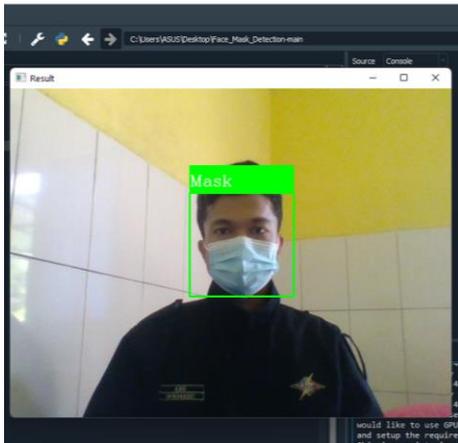
Realisasi dari pendeteksi penggunaan masker wajah ditunjukkan pada Gambar 10 dan Gambar 11. Tampilan layar pada monitor untuk kondisi ini dapat dilihat pada Gambar 12 yang menampilkan pendeteksian objek menggunakan masker dengan ditandai kotak fokus objek berwarna hijau. Sementara itu pada Gambar 13 ditampilkan pendeteksian objek tidak menggunakan masker dengan ditandai kotak fokus objek berwarna merah.



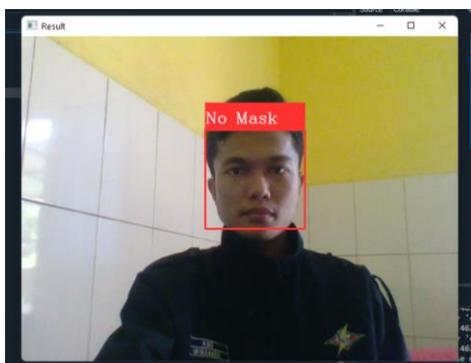
Gambar 10: Alat Pendeteksi Penggunaan Masker Wajah (Tampak Belakang)



Gambar 11: Alat Pendeteksi Penggunaan Masker Wajah (Tampak Depan)



Gambar 12: Tampilan Kotak Fokus Objek Berwarna Hijau



Gambar 13: Tampilan Kotak Fokus Objek Berwarna Merah

Pada alat ini terpasang komponen elektronika sesuai dengan yang dirancang yaitu ESP32 Cam, Arduino Uno, motor servo dan modul ISD 1820.

### B. Pengujian ESP32 Cam

Pengujian ESP32 Cam bertujuan untuk mengetahui apakah sudah terhubung dengan *WiFi* atau tidak dengan *username* dan *password* yang sudah diatur pada kode program sebelumnya. Jika sudah terhubung, maka pada *serial monitor* akan memunculkan teks “CAMERA OK” yang berarti ESP32 Cam sudah terkoneksi dengan *WiFi*. Hasil dari pengujian ESP32 Cam dapat dilihat pada Gambar 14.

```
COM7
19:55:16.741 -> ;$d|d<□▲↑□1□#<□□□□{?c□#□n'□1
19:55:17.819 -> Connecting.....
19:55:21.319 -> Successfully connected to : R3P
19:55:21.319 -> IP address: 192.168.43.136
19:55:21.319 -> CAMERA OK
```

Gambar 14: ESP32 Cam Berhasil Terhubung Dengan *WiFi*

### C. Pengujian Tensorflow

Pengujian sistem TensorFlow bertujuan untuk menganalisis keakuratan dalam mendeteksi sebuah objek secara langsung dengan jarak yang berbeda dan dalam kondisi cahaya yang berbeda. Penempatan ESP32 Cam sebagai kamera pendeteksi objek harus menghadap depan agar dapat lebih mudah mendeteksi objek di depannya. Metode pengujian sistem TensorFlow dilakukan dalam jarak uji yang berbeda dan juga kondisi cahaya yang berbeda. Hasil pengujian sistem TensorFlow dapat dilihat pada Tabel II.

TABEL II  
 PENGUJIAN SISTEM TENSORFLOW

Jarak (m)	Cahaya	Waktu	LED	Kondisi sistem	Motor servo
1	Terang	1 detik	Hijau	Terdeteksi	Terbuka
1,5	Terang	2 detik	Hijau	Terdeteksi	Terbuka
2	Terang	3 detik	Hijau	Terdeteksi	Terbuka
2,5	Terang	3 detik	Hijau	Terdeteksi	Terbuka
3	Terang	-	Merah	Tidak terdeteksi	Tertutup
1	Gelap	2 detik	Hijau	Terdeteksi	Terbuka
1,5	Gelap	3 detik	Hijau	Terdeteksi	Terbuka
2	Gelap	-	Merah	Tidak terdeteksi	Tertutup
2,5	Gelap	-	Merah	Tidak terdeteksi	Tertutup
3	Gelap	-	Merah	Tidak terdeteksi	Tertutup

Berdasarkan data hasil uji coba tangkapan gambar oleh ESP32 Cam serta respon dari output indikator LED dan motor servo yang ditunjukkan pada Tabel II, dinyatakan bahwa jarak tangkap objek yaitu < 2,5m pada kondisi cahaya terang dengan waktu respon tangkapan gambar minimal 1 detik dan maksimal 3 detik. Pada kondisi cahaya gelap dengan jarak < 1,5m diperoleh waktu pendeteksian minimal 2 detik dan maksimal 3 detik. Kondisi output indikator LED dan motor servo dipengaruhi oleh respon dari pendeteksian objek tersebut. TensorFlow merupakan sistem yang relatif cepat dalam pendeteksian objek menggunakan masker dengan baik dan benar. Contoh akses ditolak dengan ditandai indikator LED berwarna merah karena tidak menggunakan masker ditunjukkan pada Gambar 15, sedangkan contoh pendeteksian penggunaan masker dengan baik dan benar ditandai dengan indikator LED berwarna hijau serta motor servo menunjuk posisi 90° membuka portal ditunjukkan pada Gambar 16.





Gambar 15: Pemakaian Masker Yang Salah



Gambar 16: Pemakaian Masker Yang Benar

## V. KESIMPULAN

Pendeteksi penggunaan masker wajah dengan ESP32 Cam menggunakan OpenCV dan TensorFlow memiliki kecepatan saat mendeteksi objek dengan baik, yaitu tercepat sampai 1 detik dengan jarak ideal 1m, dan 3 detik dengan jarak maksimum 2,5m untuk mendeteksi objek yang menggunakan masker ataupun tidak menggunakan masker dalam kondisi cahaya terang. Sementara untuk kondisi gelap, waktu tercepat selama 2 detik dengan jarak 1m, dan 3 detik dengan jarak maksimum 1,5m. Jika orang tersebut tidak menggunakan masker dengan baik dan benar, maka indikator LED merah akan menyala serta motor servo tidak bergerak sehingga portal tidak terbuka. Jika objek menggunakan masker dengan baik dan benar, maka indikator LED hijau akan menyala dan motor servo bergerak 90° untuk membuka portal. Pemasangan alat ini pada suatu area dapat memudahkan pendeteksian orang yang tidak menerapkan protokol kesehatan agar tidak dapat masuk sehingga menjadi steril. Dengan adanya alat ini diharapkan resiko penularan penyakit khususnya Covid-19 dapat diminimalisir.

## REFERENSI

- [1] E. Riyanto, Perancangan Pengukur Detak Jantung dan Suhu Tubuh Berbasis Arduino Serta Smartphone Android, Surakarta: Universitas Muhammadiyah Surakarta, 2016.
- [2] K. Tiwikrama, A. Rabi, and R. Arifuddin, "Implementasi Palang Pintu Otomatis dengan Pendeteksi Masker Berbasis Raspberry Pi 3B +," *Semin. Nas. Inov. Teknol. UN PGRI Kediri*, pp. 1–6, 2021.
- [3] M. Ulum, M. Imaduddin, H. Sukri, and A. F. Ibadillah, "Deteksi Suhu Tubuh Dan Masker Otomatis Dengan Metode Haar Cascade Sebagai Solusi Pencegahan Penularan Covid-19," *Jurnal Riset Rekayasa Elektro*, vol. 3, no. 2, pp. 119–126, 2021.
- [4] M. Abdul, R. Irham, and D. A. Prasetya, "Prototipe Pendeteksi Masker Pada Ruang Wajib Masker Untuk Kendali Pintu Otomatis Berbasis Deep Learning Sebagai Pencegahan Penularan Covid-19," *Simposium Nasional RAPI XIX Tahun 2020*, pp. 47–55, 2020.
- [5] A. Thariq and R. Y. Bakti, "Sistem Deteksi Masker dengan Metode Haar Cascade pada Era New Normal COVID-19," *J. Sist. dan Teknol. Inf.*, vol. 9, no. 2, p. 241, 2021.
- [6] M. Imaduddin and M. Ulum, "Deteksi Suhu Tubuh Dan Masker Otomatis Dengan Metode Haar Cascade Sebagai Solusi Pencegahan Penularan Covid-19," *J. Ris. Rekayasa Elektro*, vol. 3, no. 2, pp. 119–126, 2021.
- [7] Supria and M. Nasir, "Monitoring of Body Temperature Non Contact Using AMG8833 Thermal Camera And Face Detection," *Semin. Nas. Terap. Ris. Inov.*, vol. 6, no. 1, pp. 396–403, 2020.
- [8] M. Malik, "Deteksi Suhu Tubuh dan Masker Wajah dengan MLX90614, Opencv, Keras / Tensorflow, dan Deep Learning," *Jurnal Engine: Energi, Manufaktur dan Material*, vol. 6, no. 1, pp. 19–25, 2022.
- [9] H. G. Ghifari, D. Darlis, and A. Hartaman, "Pendeteksi Golongan Darah Manusia Berbasis Tensorflow Menggunakan ESP32-CAM," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 2, p. 359, 2021.
- [10] J. Arifianto and I. Muhimmah, "Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Algoritma Deep Learning Dengan TensorFlow," *Automata*, vol. 2, no. 2, 2021.
- [11] G. Wahyono, W. D. Susanto, and A. Tafrikhatin, "Peringatan Menggunakan Sensor PIR Dengan Keluaran ISD 1820 Sebagai Pengganti Keberfungsian Garis Pengaman," *JASATEC*, vol. 1, no. 2, pp. 74–81, 2021.
- [12] A. S. Handayani, "Aplikasi Teknologi GSM/GPRS Pada Sistem Deteksi Kebakaran Berbasis Mikrokontroler Atmega 8535," *J. Nas. Pendidik. Tek. Inform.*, vol. 1, no. 1, p. 29, 2012.
- [13] R. D. Nurfitra and G. Ariyanto, "Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari," *Emit. J. Tek. Elektro*, vol. 18, no. 01, pp. 22–27, 2018.
- [14] R. M. R. Clinton and S. Sengkey, "Purwarupa Sistem Daftar Pelanggaran Lalulintas," *J. Tek. Elektro dan Komput. Vol.8*, vol. 8, no. 3, pp. 181–192, 2019.
- [15] F. Hermawanto, "Monitoring Deteksi Obyek Manusia Pada Sistem Keamanan Rumah Tinggal Berbasis Internet of Things," *SemanTECH (Seminar Nas. Teknol.)*, pp. 81–85, 2020.
- [16] B. Budiman, "Pendeteksian Penggunaan Masker Wajah Dengan Metode Convolutional Neural Network," *J. Ilmu Komput. dan Sist. Inf.*, vol. Vol.9 No.1, 2021.
- [17] H. Santoso and A. Harjoko, "Haar Cascade Classifier dan Algoritma Adaboost untuk Deteksi Banyak Wajah dalam Ruang Kelas," *Jurnal Teknologi AKPRIND*, vol. 6, no. 2, pp. 108–115, 2013.
- [18] A. Prafanto, E. Budiman, P. P. Widagdo, G. M. Putra, and R. Wardhana, "Pendeteksi Kehadiran menggunakan ESP32 untuk Sistem Pengunci Pintu Otomatis," *JIT (Jurnal Teknol. Ter.)*, vol. 7, no. 1, p. 37, 2021.
- [19] M. F. Wicaksono and M. D. Rahmatya, "Implementasi Arduino dan ESP32 CAM untuk Smart Home," *J. Teknol. dan Inf.*, vol. 10, no. 1, pp. 40–51, 2020.
- [20] M. K. Difa, "Implementasi Sistem Pengenalan Wajah Sebagai Automatic Door Lock Menggunakan Modul ESP32 CAM," *Patjou: Patria Arta Teknologikal Journal*, vol. 5, no. 2 October 2021.
- [21] Z. R. S. Elsi et al., "Perancangan Alat Deteksi Suhu Tubuh Dengan Sensor Contactless Berbasis Arduino Uno," *Jusikom: Jurnal Sistem Komputer Musirawas*, vol. 6, no. 1, pp. 50–59, 2021.
- [22] R. I. Putri, M. Rifa, and A. N. Anjarsari, "Datalogger Suhu Dan Tekanan Pada Smart Biogas Sampah Rumah Tangga Menggunakan MPX5500DP," *Majalah Ilmiah Teknologi Elektro*, vol. 19, no. 1, pp. 61–66, 2020.
- [23] I G. H. Putrawan, P. Rahardjo, and I G. A. P. R. Agung, "Sistem Monitoring Tingkat Kekeuhan Air dan Pemberi Pakan Otomatis pada Kolam Budidaya Ikan Koi Berbasis NodeMCU," *Majalah Ilmiah Teknologi Elektro*, vol. 19, no. 1, pp. 1–10, 2020.