

SISTEM DETEKSI PEMAKAIAN MASKER PADA WAJAH SECARA REAL TIME MENGUNAKAN FRAMEWORK TENSORFLOW DAN LIBRARY OPENCV

Althaf Adhari Rachman^a, Ivan Maurits^b

^aFakultas Teknologi Industri / Teknik Informatika, adharialthaf2@gmail.com, Universitas Gunadarma

^bFakultas Teknologi Industri / Teknik Informatika, ivan_maurits@staff.gunadarma.ac.id, Universitas Gunadarma

ABSTRACT

At the end of December 2019, there was a mysterious virus that attacked the city of Wuhan in China. Which at the beginning of 2020 the new virus was a new type, namely (SARS-CoV-2) and the disease was called Coronavirus disease 19 (COVID-19). The World Health Organization (WHO) provides instructions to maintain a minimum distance of 1 meter, always wash hands with soap, and always use a mask when leaving the house. The purpose of this writing is to create a program to detect wearing masks on the face, which makes it easier for users to detect who is wearing a mask or not. The method used in this writing uses the SDLC model or System Data Life Cycle by carrying out five stages namely, planning, analysis, design, implementation, and testing. The design uses flowcharts and navigation structures as well as two UML diagrams, namely use case diagrams and activity diagrams. Then for making the program using Python as a programming language, Tensorflow as a framework, Keras and Opencv as libraries, and Visual Studio Code as a text editor. From the results of trials that have been carried out using three different devices and with different specifications, this system can detect mask objects very well and this detection system can also detect more than one person wearing a mask, or not wearing a mask.

Keywords: *covid-19, machine learning, mask detection system*

ABSTRAK

Pada akhir Desember tahun 2019 lalu, terdapat salah satu virus misterius yang menyerang kota Wuhan di China. Yang mana pada awal tahun 2020 virus baru tersebut merupakan jenis baru yaitu (SARS-CoV-2) dan penyakitnya disebut Coronavirus disease 19 (COVID-19). World Health Organization (WHO) memberikan instruksi untuk menjaga jarak minimal 1 meter, selalu mencuci tangan menggunakan sabun, dan menggunakan selalu masker keluar rumah. Tujuan dari penulisan ini adalah membuat suatu program untuk mendeteksi pemakaian masker pada wajah, yang memudahkan penggunaannya dalam mendeteksi siapa saja yang menggunakan masker atau tidak. Metode yang digunakan dalam penulisan ini menggunakan model SDLC atau System Data Life Cycle dengan melakukan lima tahap yaitu, perencanaan, analisa, perancangan, implementasi, dan pengujian. Untuk perancangannya menggunakan Flowchart dan Struktur Navigasi serta dua diagram UML yaitu Use Case Diagram dan Activity Diagram. Kemudian untuk pembuatan programnya menggunakan Python sebagai Bahasa pemrograman, Tensorflow sebagai framework, Keras dan Opencv sebagai librarynya, dan Visual Studio Code sebagai teks editor. Dari hasil uji coba yang telah dilakukan dengan menggunakan tiga perangkat yang berbeda serta dengan spesifikasi yang berbeda, sistem ini dapat mendeteksi objek masker dengan sangat baik dan juga sistem deteksi ini dapat mendeteksi lebih dari satu orang yang bermasker, maupun tidak bermasker.

Kata Kunci: *covid-19, machine learning, sistem deteksi masker.*

1. PENDAHULUAN

Pada akhir Desember tahun 2019 lalu, terdapat salah satu virus misterius yang menyerang kota Wuhan di China. Yang mana pada awal tahun 2020 virus baru tersebut merupakan jenis baru yaitu (SARS-CoV-2) dan penyakitnya disebut Coronavirus disease 19 (COVID-19). Total kasus penyakit ini dari Januari 2020 hingga saat ini (Data April 2022) sudah ada 497,49 juta kasus (Our World in Data, 2022). Di Indonesia sendiri COVID-19 pertama kali terdeteksi pada tanggal 2 Maret 2020, Presiden Joko Widodo

mengkonfirmasi bahwa terdapat 2 warga kota Depok yang positif terjangkit COVID-19 setelah melakukan kontak dengan warga negara Jepang. Maka World Health Organization (WHO) memberikan instruksi untuk menjaga jarak minimal 1 meter, selalu mencuci tangan menggunakan sabun, dan menggunakan selalu masker keluar rumah. Masker sendiri adalah perlindungan pernafasan yang digunakan sebagai metode untuk melindungi individu dari menghirup zat-zat bahaya atau kontaminan yang berada di udara, perlindungan pernafasan atau masker tidak dimaksudkan untuk menggantikan metode pilihan yang dapat menghilangkan penyakit, tetapi digunakan untuk melindungi secara memadai pemakainya (Cohen & Birdner, 2012).

Beberapa orang masih tidak mau menggunakan masker ketika diluar rumah, hal ini tentu saja merupakan salah satu pelanggaran protokol kesehatan yang telah ditentukan. Untuk itu dengan adanya Machine Learning dapat membuat suatu aplikasi sistem deteksi otomatis, untuk mengetahui orang – orang mana saja yang menggunakan masker dan yang tidak menggunakan masker. Aplikasi ini bekerja secara real – time, dimana hasil dari objek – objek yang nanti terdeteksi akan langsung terlihat perbedaannya. Machine learning merupakan serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara mempresentasikan data-data tersebut dengan algoritma pembelajaran (Danukusumo, 2017).

2. METODOLOGI PENELITIAN

Metode penulisan untuk penelitian ilmiah ini menggunakan model SDLC (System Data Life Cycle), dengan melakukan dalam lima tahap yaitu:

1. Perencanaan
Mengumpulkan berbagai kebutuhan serta informasi sebagai kebutuhan dalam penulisan ini. Sumber informasi didapat dari berbagai sumber seperti jurnal, artikel, serta video yang terkait dalam pembuatan sistem.
2. Analisa
Menganalisis bagaimana sistem akan bekerja terhadap objek yang akan dideteksi secara baik, agar memenuhi ekspektasi.
3. Perancangan
Pada tahap ini perancangan menggunakan flowchart, struktur navigasi, usecase diagram, activity diagram serta menampilkan beberapa dataset yang digunakan dalam membangun sistem deteksi ini.
4. Implementasi
Melakukan pengimplementasian dengan Bahasa pemrograman Python, framework Tensor Flow, library OpenCV, Keras, serta MobileNET.
5. Pengujian
Pengujian dilakukan lebih dari satu kali dengan menjalankan sistem di beberapa versi desktop yang berbeda. Agar dapat dicek apakah terjadi error pada sistem atau tidak sebelum digunakan

3. HASIL DAN PEMBAHASAN

Analisis Kebutuhan

Dalam membuat suatu program dibutuhkan suatu analisis untuk menggambarkan bagaimana program ini untuk bisa digunakan. Analisis kebutuhan adalah suatu metode untuk menggambarkan kebutuhan yang diperlukan dalam penelitian, khususnya dalam pembuatan program sistem deteksi pemakaian masker pada wajah secara real-time dengan framework tensorflow dan library opencv.

a. Analisis Kebutuhan Perangkat Lunak

Dalam pembuatan program ini, penulis menggunakan beberapa perangkat lunak (software) diantaranya adalah:

- Windows 10 Home 64 bit
- Visual Studio Code
- Python 3.8.7
- Draw.io
- Github
- Tensorflow 2.3.4
- Opencv 4.2.0
- Keras 2.3.1

b. Analisis Kebutuhan Perangkat Keras

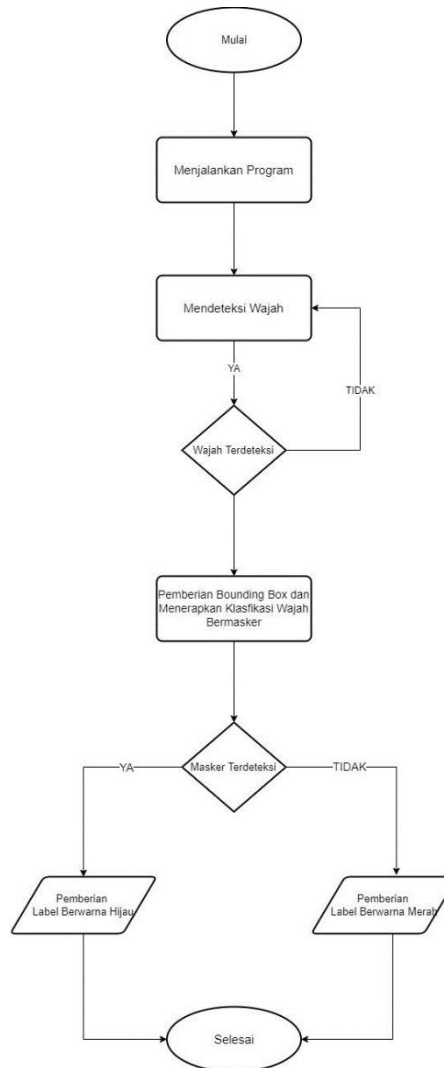
Dalam merancang program ini, penulis menggunakan laptop Lenovo Legion 5

sebagai wadah dalam uji coba program dengan spesifikasi sebagai berikut:

- AMD Ryzen 5 5600H
- 16GB RAM
- 512GB SSD
- NVIDIA GeForce RTX 3050
- 720p Camera

Flowchart

Flowchart digunakan untuk menggambarkan alur program yang dibuat, dalam hal ini adalah aplikasi sistem deteksi pemakaian masker pada wajah secara real-time dengan framework tensorflow dan library opencv.



Gambar 1. Flowchart Aplikasi

Struktur Navigasi

Struktur navigasi berfungsi untuk menggambarkan alur proses terhadap aplikasi yang dibuat. Struktur navigasi yang digunakan pada pembuatan aplikasi sistem deteksi pemakaian masker pada wajah secara real-time dengan framework tensorflow dan library opencv, adalah jenis struktur navigasi linier, sebagai berikut.



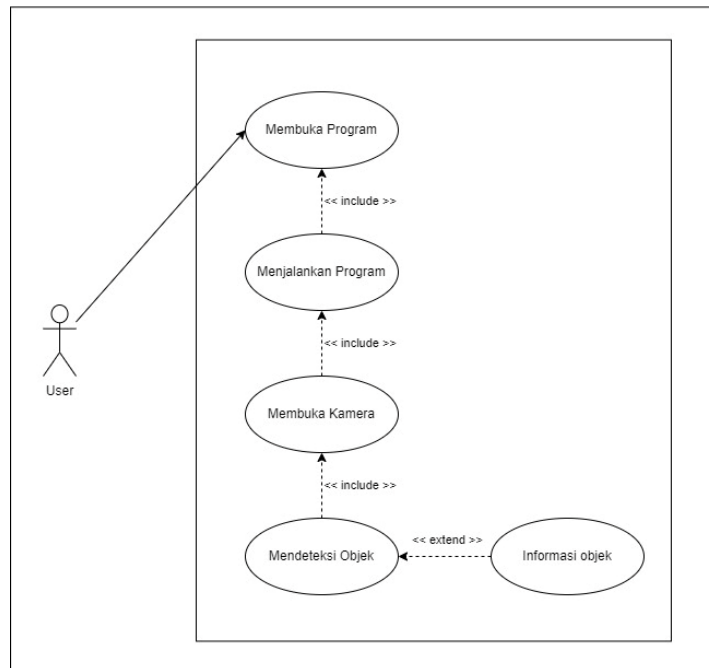
Gambar 2. Struktur Navigasi Program

Perancangan Diagram UML

Dalam pembuatan aplikasi ini, akan menampilkan dua diagram UML diantaranya, use case diagram dan activity diagram.

a. Use Case Diagram

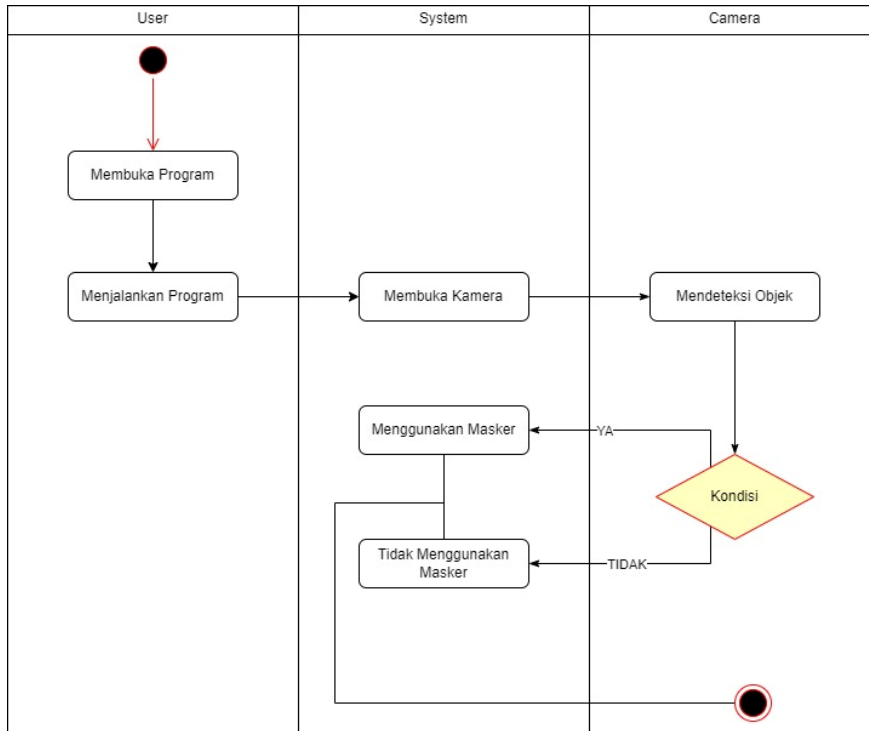
Penggunaan use case diagram, menunjukkan bagaimana suatu user/pengguna menjalankan aplikasi. Setiap user yang ingin menggunakan aplikasi ini, hal yang pertama yang dilakukan adalah membuka aplikasi serta menjalankannya, kemudian kamera akan terbuka sembari menampilkan framenya yang digunakan untuk mendeteksi objek serta memberikan informasi, apakah objek yang dideteksi memakai masker atau tidak.



Gambar 3. Use Case Diagram

b. Activity Diagram

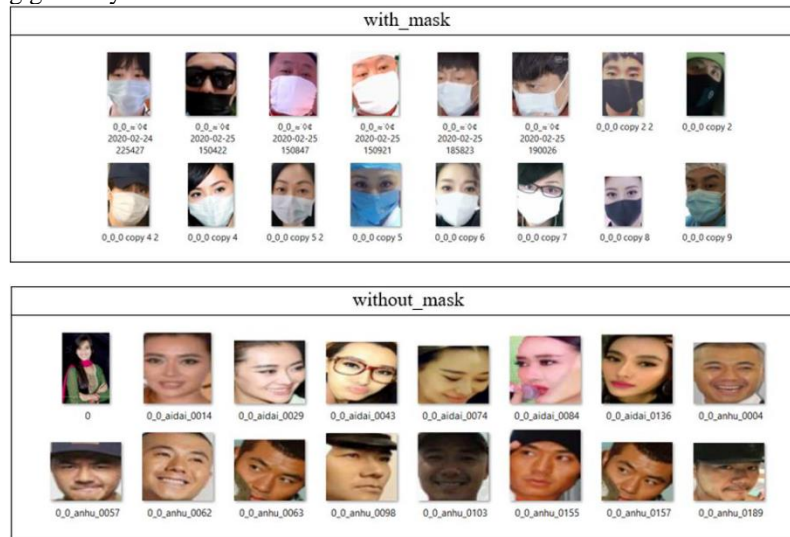
Activity diagram, menunjukkan cara kerja sistem dalam menjalankan aplikasi sistem deteksi pemakaian masker pada wajah. Dimulai ketika user membuka program, kemudian menjalankan programnya ketika program dijalankan, otomatis akan membuka kamera pada desktop yang digunakan, lalu secara otomatis kamera akan mendeteksi objek apakah objek/wajah tersebut menggunakan masker atau tidak.



Gambar 4. Activity Diagram

Pengumpulan Dataset Gambar

Dataset ini terdiri dari dua buah folder yaitu `with_mask` yang terdiri dari 1.915 gambar orang/objek menggunakan masker, dan `without_mask` yang terdiri dari 1.918 gambar orang/objek yang tidak menggunakan masker. Setiap gambar yang ada pada dataset ini memiliki keadaan dan kondisi yang berbeda, seperti latar belakang gambarnya.

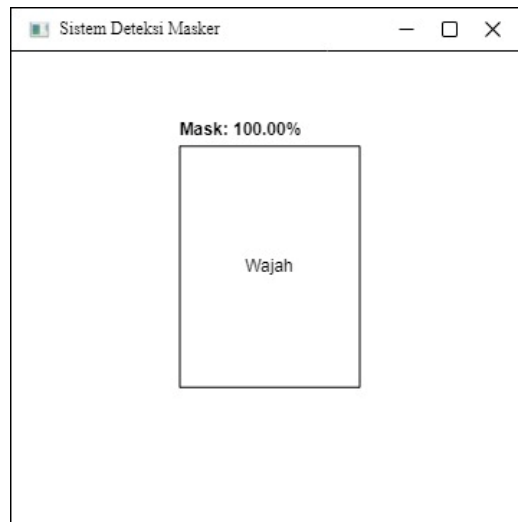


Gambar 5. Dataset with_mask dan without_mask

Perancangan Sistem Deteksi Pemakaian Pada Wajah

a) Tampilan Frame Menggunakan Masker

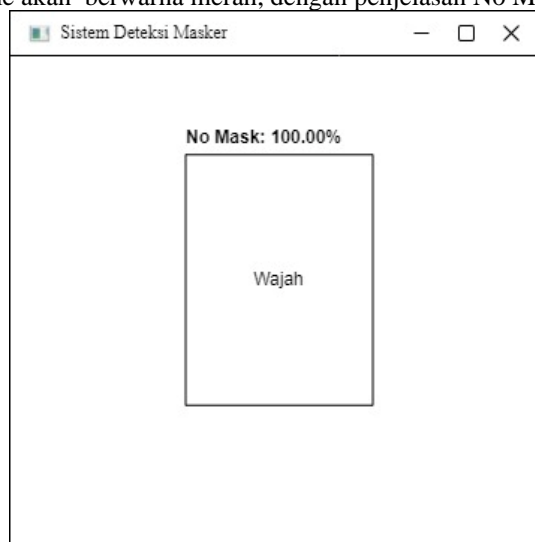
Karena menggunakan metode real time video maka tampilan frame ini akan otomatis terbuka, ketika user menjalankan programnya. Tampilan ini menyesuaikan kondisi terhadap objek yang dideteksi, jika objek menggunakan masker pada wajah maka tampilan label pada frame akan berwarna hijau, dengan penjelasan Mask: 100%.



Gambar 6. Rancangan Tampilan Bermasker

b) Tampilan Frame Tidak Menggunakan Masker

Tampilan frame ini akan otomatis terbuka, ketika user menjalankan programnya. Tampilan ini juga menyesuaikan kondisi terhadap objek yang dideteksi, jika objek tidak menggunakan masker pada wajah maka tampilan label pada frame akan berwarna merah, dengan penjelasan No Mask: 100%.



Gambar 7. Rancangan Tampilan Tidak Bermasker

Kemudian objek/wajah yang sudah terdeteksi, akan diubah ke dalam bentuk vector yaitu jarak antar titik dengan sudut diantara citra. Kemudian vector akan ditempatkan ke dalam bentuk klasifikasi, hal ini dilakukan agar dapat menghitung nilai vector dengan dataset yang sudah tersimpan.

- (1, 1, 200, 7)
- (1, 1, 200, 7)
- (1, 1, 200, 7)
- (1, 1, 200, 7)

Objek yang sudah terdeteksi dan lalu sudah mendapatkan nilai vector, akan melakukan pencocokan terhadap gambar/citra yang ada di dataset. Vector objek/wajah yang tidak menggunakan masker, akan diprediksi sesuai dengan data train yang sebelumnya telah diuji.

Secara otomatis sistem akan memberi informasi kepada user sebuah label dengan bentuk rectangle atau persegi panjang, sesuai dengan hasil deteksi wajah berdasarkan hasil koordinat (x, y) dan (w, h). Koordinat (x, y) digunakan sebagai pusat rectangle yang di gambar di dalam frame, sedangkan koordinat (w, h) merupakan nilai lebar dan tinggi dari rectangle.

Untuk wajah yang menggunakan masker, juga akan disesuaikan dengan data train yang sebelumnya telah diuji. Sistem akan memasukkan kembali output ke dalam frame untuk ditampilkan kembali. Hal tersebut akan menghasilkan nilai lebar dan tinggi sehingga mampu mendeteksi penggunaan masker pada wajah.

Uji Coba Program

Uji coba program bertujuan untuk melihat hasil program di perangkat laptop yang berbeda beda, dengan maksud untuk mengamati bagaimana program ini bisa berjalan di tempat yang berbeda. Berikut merupakan hasil dari pengujian di tiga device laptop yang berbeda.


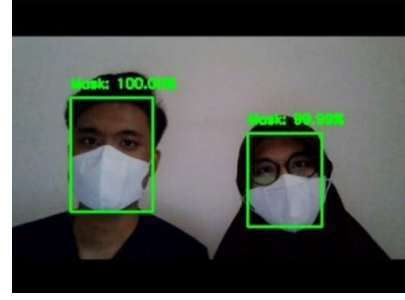
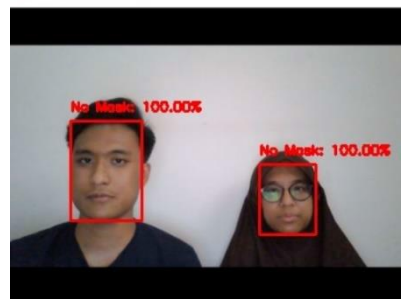

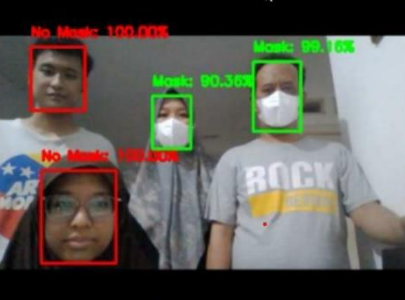
Tabel 1. Hasil Uji Coba Program

Device	Spesifikasi	Hasil
Lenovo Legion 5	AMD Ryzen 5 5600H 16GB RAM 512GB SSD NVIDIA GeForce RTX 3050 720p Camera	Pada perangkat laptop Lenovo Legion 5, program berjalan dengan baik dan mendeteksi dengan baik.
Lenovo Ideapad Flex 5	Intel Core i5-1135G7 8GB RAM 512GB SSD 720p Camera	Pada perangkat laptop Lenovo Ideapad Flex 5, program berjalan dengan baik dan mendeteksi dengan baik.
Asus X540Y	AMD E1-7010 Dual-core 2GB RAM 256GB SSD HD Camera	Pada perangkat laptop Asus X540Y, program berjalan kurang baik atau terpatah – patah pada saat dijalankan, tetapi tetap dapat mendeteksi dengan baik meskipun dengan spesifikasi yang cukup rendah.

Uji Coba Keakuratan Sistem Deteksi

Uji coba keakuratan bertujuan untuk melihat seberapa akurat sistem dalam mendeteksi pemakaian masker pada wajah. Pada uji coba ini, menggunakan perangkat Lenovo Legion 5 yang mana merupakan perangkat utama, dalam pembuatan program.

Output	Kesimpulan
	<p>Nilai keakuratan sistem dalam mendeteksi objek memakai masker adalah 100.00%, yang mana dapat diartikan sistem berjalan sangat baik dalam mendeteksi masker.</p>
	<p>Nilai keakuratan sistem dalam mendeteksi objek tidak memakai masker adalah 100.00%, yang mana dapat diartikan sistem berjalan sangat baik dalam mendeteksi masker.</p>
	<p>Pada tahap ini, menggunakan satu tangan sebagai pengganti masker, hasilnya masker tetap tidak terdeteksi dengan nilai keakuratan yang didapat adalah 100.00%.</p>
	<p>Pada tahap selanjutnya, menggunakan dua tangan sebagai pengganti masker, hasilnya masker tetap tidak terdeteksi dengan nilai keakuratan yang didapat adalah 98.40%.</p>

	<p>Di tahap ini, menggunakan baju sebagai pengganti masker, hasilnya masker tetap tidak terdeteksi dengan nilai keakuratan yang didapat adalah 55.41%.</p>
	<p>Pada tahap ini, menggunakan dua objek atau orang yang sama – sama menggunakan masker, hasilnya masker tetap terdeteksi dengan masing – masing nilai keakuratan adalah 100.00% dan 99.99%.</p>
	<p>Di tahap ini, menggunakan dua objek atau orang yang sama – sama tidak menggunakan masker, hasilnya masker tetap tidak terdeteksi dengan nilai keakuratannya sama yaitu 100.00%.</p>
	<p>Pada tahap ini objek memiliki keadaan yang berbeda, dimana yang satu memakai masker dan yang satu tidak memakai masker, hasilnya sistem tetap dapat mendeteksi objek dengan baik, nilai keakuratan yang didapat sama yaitu 100.00%.</p>
	<p>Pada tahap ini dilakukan pendeteksian empat objek sekaligus, dimana dua orang memakai masker dan dua orang lainnya tidak menggunakan masker. Hasilnya sistem tetap dapat mendeteksi ke-empat objek tersebut dengan baik.</p>

Uji Coba Jarak Jangkauan Sistem

Output	Kesimpulan
	<p>Minimal jarak yang dibutuhkan untuk mendeteksi adalah 30 cm dari kamera. Pada jarak tersebut sistem dapat mendeteksi objek dengan sangat baik.</p>
	<p>Maksimal jarak yang dibutuhkan untuk mendeteksi objek adalah 1 Meter, jika jarak objek dengan kamera lebih dari 1 Meter maka sistem tidak akan bisa mendeteksi objek masker.</p>

4. KESIMPULAN DAN SARAN

Berdasarkan hasil penulisan serta uji coba yang dilakukan terhadap program sistem deteksi pemakaian masker pada wajah secara real time menggunakan framework tensorflow dan library opencv, didapatkan beberapa kesimpulan yaitu sebagai berikut:

1. Aplikasi ini dapat dimanfaatkan sebagai salah satu program pembantu dalam mendeteksi seseorang atau lebih dalam penggunaan masker.
2. Hasil pelatihan dataset yang didapat sangat baik, dengan tingkat akurasi yang menunjukkan angka 1.0 dan loss menunjukkan angka 0.0, disimpulkan bahwa dataset memiliki nilai keakurasian yang sangat tinggi dalam mendeteksi penggunaan masker.
3. Aplikasi ini dapat mendeteksi penggunaan masker secara real time menghasilkan output yang baik, dan juga dapat mendeteksi penggunaan masker dan tidak menggunakan masker.
4. Sistem pendeteksian ini bisa mendeteksi objek atau orang lebih dari satu orang. Sistem deteksi pemakaian masker pada wajah secara real time menggunakan framework tensorflow dan library opencv ini masih mengalami beberapa kekurangan, seperti penggunaan kerah baju sebagai masker terkadang masih terdeteksi menggunakan masker, disarankan ketika ingin mengembangkan program ini dilakukan penambahan atau perubahan parameter agar masalah tersebut bisa teratasi, dan juga penggunaan sistem ini bisa lebih maksimal dalam mendeteksi objek masker.

DAFTAR PUSTAKA

[1] Allaam, M.R.R. and Wibowo, A.T. (2021) ‘Klasifikasi Genus Tanaman Anggrek Menggunakan Metode Convolutional Neural Network’, e-Proceeding of Engineering, 8(2), pp. 3180–3196. Available at: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/14709/14486>.

[2] Alwanda, M.R., Ramadhan, R.P.K. and Alamsyah, D. (2020) ‘Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle’, Jurnal Algoritme, 1(1), pp. 45–56. Available at: <https://doi.org/10.35957/algoritme.v1i1.434>.

[3] Hikmatia A.E, N. and Ihsan Zul, M. (2021) ‘Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow’, Jurnal Komputer Terapan, 7(Vol. 7 No. 1 (2021)), pp. 74–83. Available at: <https://doi.org/10.35143/jkt.v7i1.4629>.

- [4] Kamil, H. (2013) 'Pengembangan Aplikasi Distribusi Surat Di Fakultas Teknologi Informasi Dengan Notifikasi SMS Menggunakan Framework Yii dan Gammu', *Jurnal Nasional Teknik Elektro*, 2(2), pp. 39–45. Available at: <https://doi.org/10.20449/jnte.v2i2.73>.
- [5] M Teguh Prihandoyo (2018) 'Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web', *Jurnal Informatika: Jurnal Pengembangan IT*, 3(1), pp. 126–129.
- [6] Muhammad Romzi and Kurniawan, B. (2020) 'Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma', *JTIM: Jurnal Teknik Informatika Mahakarya*, 03(2), pp. 37–44.
- [7] Prasetya, S.Y. et al. (2020) 'Sistem Deteksi Hama Pada Kolam Budidaya Ikan Berbasis Audio dan Video', *Jurnal Bumigora Information Technology (BITe)*, 2(2), pp. 119–129. Available at: <https://doi.org/10.30812/bite.v2i2.911>.
- [8] Rochmawati, N. et al. (2021) 'Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam', *Journal of Information Engineering and Educational Technology*, 5(2), pp. 44–48. Available at: <https://doi.org/10.26740/jieet.v5n2.p44-48>.
- [9] Syahrudin, A.N. and Kurniawan, T. (2018) 'Input Dan Output Pada Bahasa Pemrograman Python', *Jurnal Dasar Pemrograman Python STMIK*, (January), pp. 1–7.
- [10] Syarifah (2018) 'Deep Learning Object Detection Pada Video', *Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network [Preprint]*.
- [11] Victoria, O. and Solihin, I.P. (2018) 'Pendeteksi Wajah Secara Realtime Menggunakan Metode Eigenface', *SEINASI-KESI (Seminar Nasional Informatika, Sistem Informasi Dan Keamanan Siber)*, pp. 126–131.
- [12] Wahyono, T. (2018) 'Fundamental of Python for Machine Learning: Dasar-Dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan', *Gava Media*, (September 2018), p. 49.
- [13] Warsito, A.B., Yusup, M. and Yulianto, Y. (2014) 'Kajian Yii Framework Dalam Pengembangan Website Perguruan Tinggi', *CCIT Journal*, 7(3), pp. 437–451. Available at: <https://doi.org/10.33050/ccit.v7i3.265>.
- [14] Wasil, M., Harianto, H. and Fathurrahman, F. (2022) 'Pengaruh Epoch pada Akurasi menggunakan Convolutional Neural Network untuk Klasifikasi fashion dan Furniture', *Infotek : Jurnal Informatika dan Teknologi*, 5(1), pp. 53–61. Available at: <https://doi.org/10.29408/jit.v5i1.4393>.