

DETEKSI MASKER WAJAH SECARA REALTIME DENGAN ARDUINO UNTUK MENCEGAH PENYEBARAN COVID-19

¹ Arafat

² Silvia Ratna

³ Wagino

^a Program Studi Teknik Informatika, Fakultas Teknologi Informasi, UNISKA

^b Program Studi Teknik Informatika, Fakultas Teknologi Informasi, UNISKA

^c Program Studi Teknik Informatika, Fakultas Teknologi Informasi, UNISKA

email: aaruniska@gmail.com

email: via.borneo@gmail.com

email : ginouniska@gmail.com

Abstrak

The emergence of the COVID-19 pandemic has had a lasting impact on countries around the world since 2019. Face mask detection has been a significant advance in the field of image processing and deep learning studies. Many face detection models have been designed using different algorithms and techniques. The approach proposed in this proposal is developed to prevent people without masks from entering public places (i.e. Malls, Universities, Offices, ...etc) by detecting face masks using deep learning, TensorFlow, Keras, and OpenCV methods and sending a signal to the connected Arduino device. to the door to be opened. To detect a person's face in real-time and identify whether the person is wearing a mask or not. This method uses datasets collected from various sources.

Keywords: *Mask detection, deep learning, TensorFlow, Keras, OpenCV, Arduino*

1. PENDAHULUAN

Sejak Desember 2019, dunia digemparkan dengan munculnya COVID-19 yang merupakan penyakit pernafasan akut. Penyakit ini disebabkan oleh virus corona yang bernama severe acute respiratory syndrome coronavirus 2 atau disingkat SARS CoV-2 (Susilo et al., 2020).

Pemerintah selaku pembuat kebijakan, memberlakukan peraturan bagi setiap orang yang akan keluar rumah wajib menggunakan masker dan juga tetap melakukan physical distancing agar setiap orang dapat keluar rumah dan melakukan pekerjaan tanpa harus khawatir terpapar virus COVID-19 ini. Dengan adanya himbauan ini, diharapkan setiap masyarakat mampu untuk mematuhi penggunaan masker (Lambacing & Ferdiansyah, 2020).

Penularan virus COVID-19 ini sendiri dapat terjadi melalui percikan saat bersin atau batuk antar manusia. Oleh karena itu penggunaan masker dan physical distancing telah beroperasi pada lingkungan perusahaan yang mewajibkan

karyawannya menggunakan masker sebelum masuk ke kantor. Untuk mencegah persebaran COVID-19 yang tidak terkendali ini, maka dibuatlah penelitian sistem deteksi masker dengan metode deep learning pada era new normal COVID-19. Sistem deteksi masker ini menggunakan tools anaconda dan python 3.8. Sistem ini menggunakan beberapa library dengan meng-upgrade pip seperti scipy, pillow, matplotlib dan opencv contrib python. Metode yang digunakan pada proposal ini adalah deep learning dimana algoritma ini merupakan salah satu model machine learning yang kerap kali digunakan sebagai pondasi aplikasi object detection (terutama face recognition) dalam sebuah gambar maupun video. Metode ini sering digunakan untuk mengenali area mana saja terdapat wajah manusia (*face recognition*) pada sebuah gambar menggunakan OpenCV. Data yang digunakan adalah data dari orang yang menggunakan masker dan tidak menggunakan masker.

Pada 31 Maret 2020, Surat Keputusan Presiden No. 11 Tahun 2020 dikeluarkan karena adanya Corona Virus Disease 2019 (COVID-19). Berdasarkan keputusan tersebut, terdapat kebijakan untuk memakai masker guna meminimalisir penyebaran COVID-19 khususnya pada tempat dan fasilitas umum (Kurniawan, 2021)s. Untuk membantu pengawasan terhadap kebijakan tersebut, dibutuhkanlah teknologi yang terintegrasi pada sistem yang mengimplementasikan penggunaan deep learning. Penerapan sistem tersebut ditujukan untuk mendeteksi penggunaan masker pada masyarakat secara real-time.

2. METODE PENELITIAN

Penelitian ini dilakukan berdasarkan tahapan-tahapan penting yang dikerjakan dengan berorientasikan kepada indikator keberhasilan dalam menghubungkan arduino dengan deep learning sehingga dapat digunakan untuk menyelesaikan permasalahan multi objektif. Untuk dapat mencapai, indikator tersebut, maka tahapan-tahapan penelitian ini adalah sebagai berikut :

1. Analisa masalah, dalam hal ini kebutuhan menganalisa permasalahan yang akan diteliti mengenai masker wajah.
2. Analisa kebutuhan, dalam hal ini segala kebutuhan dalam meneliti baik dari jurnal, buku, literatur-literatur, alat dan bahan.
3. Mendesain alat yang akan dibangun dengan menggunakan arduino dengan algoritma deep learning.
4. Membuat program dengan menggunakan arduino IDE
5. Menguji alat dengan kode program yang dibuat.
6. Membuat laporan dan menyimpulkan hasil penelitian

Alat dan Bahan

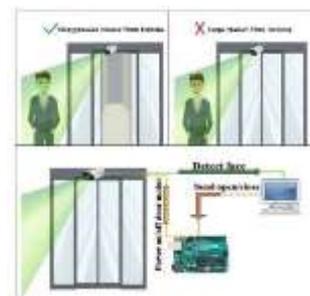
Dalam perencanaan dan pembuatan Deteksi Masker Wajah Secara Realtime Dengan Arduino Untuk Mencegah Penyebaran Covid-19 menggunakan arduino sebagai *controller* dan *deep*

learning sebagai algoritma untuk mendeksi masker wajahnya. Perancangan perangkat keras dan perangkat lunak meliputi pembuatan model pemasangan alat pembuka pintu. Bahan yang digunakan meliputi

- a. Perangkat Keras
 1. Camera
 2. Raspberry Pi 4
 3. Monitor 7” touch screen
 4. Arduino uno
 5. Relay
 6. Slide Door
- b. Software yang digunakan untuk pembuatan sistem:
 1. Python: 3.9 atau terbaru
 2. PySerial
 3. Tensorflow
 4. Keras
 5. Imutil
 6. Numpy
 7. Opencv-python
 8. Matplotlib
 9. Scipy

Perancangan Sistem

Sistem ini dirancang untuk mendeteksi wajah dalam bentuk video secara *real-time* dan untuk menentukan apakah orang tersebut memakai masker wajah atau tidak. Dengan menggunakan data yang terdeteksi, sistem dapat memutuskan apakah orang yang bersangkutan boleh masuk ke tempat-tempat umum seperti (pasar, atau rumah sakit) atau tidak. Sistem ini dapat digunakan di rumah sakit, pasar, terminal bus, restoran, dan tempat-tempat umum lainnya di mana pemantauan harus dilakukan.



Gambar 1.1 Alur Kerja Sistem

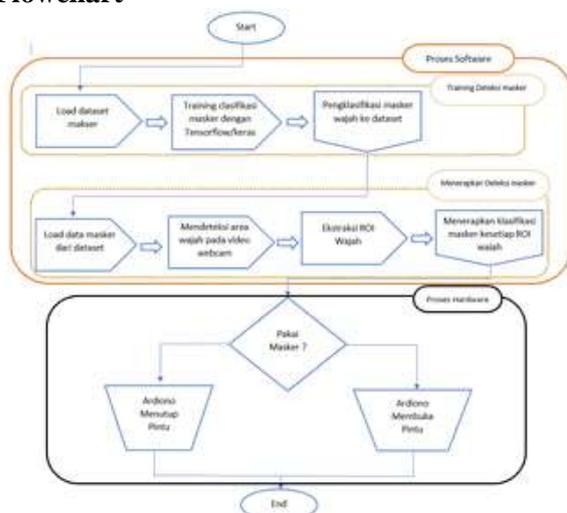
Sistem ini pada dasarnya terdiri dari, kamera yang ditempatkan di depan pintu, kamera mengirimkan frame video ke OpenCV yang berjalan di raspberry pi 4. Jika OpenCV mendeteksi wajah maka akan mengambil gambar orang yang masuk ke tempat umum dengan kamera, kemudian gambar wajah yang terdeteksi dikirim ke sistem untuk dibandingkan (diuji) apakah orang tersebut memakai masker wajah atau tidak dengan menggunakan *tenserflow*. Setelah pemrosesan data, *TensorFlow* akan mengirimkan hasilnya ke Arduino apakah akan membuka pintu atau tidak seperti yang ditunjukkan pada gambar 1.1.

3. HASIL DAN PEMBAHASAN

Prinsip Kerja Alat

Sistem ini terdiri dari fase *software* dan *hardware*. Fase *software* bertanggung jawab untuk melakukan training gambar pada dataset dan kemudian menerapkan proses pengambilan gambar wajah dari video secara real-time kemudian mengidentifikasi, apakah menggunakan masker atau tidak menggunakan masker, sedangkan pada fase *hardware* adalah untuk melakukan proses membuka/menutup pintu geser sesuai dengan hasil fase software seperti yang ditunjukkan pada gambar. Jika menggunakan masker maka pintu akan terbuka dan jika tidak menggunakan masker pintu akan tertutup yang akan di perintah oleh arduino untuk melakukan proses buka dan tutup pintu

Flowchart



Gambar 1.2 Flowchart proses deteksi masker dan buka tutup pintu

Pengumpulan Data

Proses deteksi masker wajah secara *real-time* akan melibatkan dua tahap yaitu proses training dan deteksi masker.

Training

a. Dataset

Untuk training model, penulis memiliki kumpulan data yang di dapatkan dari Kaggle (Chakraborty, 2020) yang berisi sekitar total 3833 gambar, yaitu, 1918 gambar dengan menggunakan masker dan 1915 gambar tanpa masker seperti yang ditunjukkan pada gambar 1.3 dan gambar 1.4. Penulis telah menggunakan kumpulan data ini seperti yang ada di dalamnya yang berupa gambar horizontal dan vertikal.



Gambar 1.3 Dataset (dengan masker)



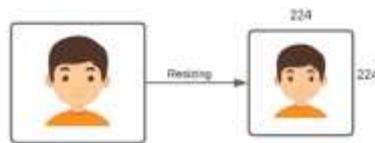
Gambar 1.4 Dataset (tanpa masker)

b. Pra pemrosesan gambar

Untuk akurasi yang lebih baik dalam pra-pemrosesan gambar adalah sebuah langkah penting sebelum melakukan training yang sebenarnya. Seperti dalam penelitian ini, penulis menggunakan *Convolutional Neural Networks*, untuk merubah ukuran gambar *dataset* menjadi resolusi 224 x 224 piksel seperti yang ditunjukkan pada Gambar 1.5.

Setelah merubah ukuran gambar, gambar tersebut telah menjadi *pixelized* (yaitu, dirubah menjadi

sebuah array) dengan library OpenCV. Setelah proses *pixelized*, gambar diberi label sebagai ‘Menggunakan Masker’ atau ‘Tanpa Masker’ dengan fungsi library dari *TensorFlow* dan *Keras*. *Array master* yang berisi semua gambar berpiksel dengan label yang telah dibuat untuk dilakukan training.



Gambar 1.5 Merubah ukuran resolusi gambar menjadi 224 x 224 piksel

c. Training dan validasi dataset

Dalam melakukan training dan validasi dataset akan menggunakan bantuan *python* dengan menggunakan metode *Deep Neural Network*, untuk tahapannya sebagai berikut:

1. Mempersiapkan *variable global*

Variable global merupakan sebuah parameter dalam training data untuk menghitung nilai koreksi bobot pada waktu. Semakin besar learning rate maka rentang untuk menentukan perubahan bobot dalam neural network semakin besar dan berpengaruh dengan kedekatan fitur kepada kelas yang lain. Saat learning rate kecil maka perubahan bobot semakin kecil dan kedekatan pola ke kelas lain dari inisial target juga semakin jauh.

1) *Learning rate* merupakan sebuah parameter dalam training data untuk menghitung nilai koreksi bobot pada waktu proses training. Semakin besar *learning rate* maka rentang untuk menentukan perubahan bobot dalam *neural network* semakin besar dan berpengaruh dengan kedekatan fitur kepada kelas yang lain. Saat learning rate kecil maka perubahan bobot semakin kecil dan kedekatan pola ke kelas lain

dari inisial target juga semakin jauh.

2) Epoch

Epoch adalah *hyperparameter* yang menentukan berapa jumlah algoritma pembelajaran akan bekerja mengolah seluruh dataset *training*.

3) Batch size

Merupakan jumlah kelompok dari sampel data. Contoh: jika mempunyai 100 dataset dan batch size kita adalah 5 maka dataset akan dibagi menjadi 5 bagian dan disetiap bagian ada 20 dataset. Untuk jumlah nilai *variable* seperti yang ditampilkan pada pseudocode berikut:

```
INIT_LR = 1e-4
EPOCHS = 5
BS = 32
```

2. Partisi *dataset*

Partisi yang dimaksudkan ada kapasitas dari dataset yang akan dibagi menjadi 2 yaitu 80% dataset digunakan untuk training dan sisanya digunakan untuk validasi dan ini dilakukan pada setiap epoch atau setiap sekali training dataset yang menggunakan semua data dari dataset. Seperti yang terlihat pada pseudocode berikut:

```
trainX, testX, trainY, testY) =
train_test_split(data, labels
, test_size=0.20, stratify=lab
els, random_state=20)
```

3. *Data Augmenatation*

Data Augmentation adalah sebuah teknik memanipulasi sebuah data tanpa kehilangan inti atau esensi dari data tersebut. Untuk data berupa Image, kita bisa lakukan rotate, flip, crop, dll.

```
aug = ImageDataGenerator(
rotation_range=20,
zoom_range=0.15,
width_shift_range=0.2,height
_shift_range=0.2,shear_range
=0.15,horizontal_flip=True,f
ill_mode="nearest")
```

Dari pseudocode diatas terdiri dari:

- *Rotation range*, merupakan derajat rotasi gambar
 - *Zoom range*, merupakan tingkat perbesar dari gambar
 - *Width shift range*, untuk mengatur lebar gambar
 - *Hight shift range*, untuk mengatur tinggi gambar
 - *Shear range*, gambar akan didistorsi sesuai dengan sumbu.
 - *Horizontal flip*, untuk mengatur gambar berbentuk horizontal
 - *Fill mode*, untuk mengatur *layout* gambar.
4. Pembuatan struktur *Deep Neural Network*

Dalam menerapkan *Deep Neural Network* disini dijelaskan memiliki 2 *hidden layer* yang dijelaskan pada *pseudocode* berikut:

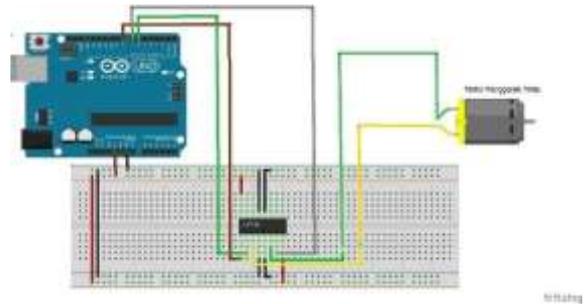
```
baseModel =
MobileNetV2(weights="imagenet", include_top=False,
input_tensor=Input(shape=(224, 224, 3)))
headModel = baseModel.output
headModel =
AveragePooling2D(pool_size=(7, 7))(headModel)
headModel =
Flatten(name="flatten")(headModel)
headModel = Dense(128,
activation="relu")(headModel)
headModel =
Dropout(0.5)(headModel)
headModel = Dense(2,
activation="softmax")(headModel)
```

Dari *pseudocode* diatas dijelaskan bahwa untuk input menggunakan gambar dengan ukuran 224 x 224 pixel dan menghasilkan 2 output yaitu *with mask* dan *without mask*.

Perancangan Pengendali Pintu Masuk

Untuk sistem yang digunakan pada pengendali pintu akses masuk, diterapkan pada Raspberry Pi 4. Untuk pengendalian pintu masuk, digunakan motor DC yang dapat digerakkan untuk membuka dan menutup akses pintu masuk. Motor DC akan bergeser ke kiri untuk membuka pintu dan bergeser kekanan untuk menutup Motor DC membuka bila program

mendeteksi wajah sebagai wajah yang menggunakan masker. Sistem pengendali pintu akses masuk digunakan untuk mendeteksi apakah seseorang menggunakan masker sehingga pendeteksian yang diharapkan dapat memiliki akurasi tinggi adalah pendeteksian wajah bermasker. Algoritma program yang digunakan dalam sistem ini berjalan secara terus menerus tanpa terminasi.



Gambar 1.6 Rancangan hardware untuk membuka dan menutup pintu

Hasil Implementasi Pengendali Pintu

Pada gambar 1.7 terdapat layar monitor dan web cam yang terhubung dengan raspberry 4 dan arduino Uno sebagai kendali akses pintu masuk yang di gerakan oleh motor DC.



Gambar 1.7 Pemasangan hardware pada pintu

Pada kondisi pintu tertutup, ini berarti tidak ada orang yang bergerak di depan pintu atau tidak menggunakan masker, maka pintu tetap akan tertutup dan jika ada orang tanpa masker maka alarm berbunyi.



Gambar 1.8 Kondisi pintu tertutup

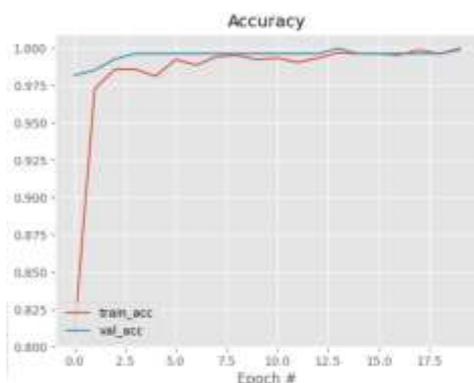
Pintu akan terbuka jika ada orang yang menghadap ke web cam dengan menggunakan masker, tetapi jika pemasangan masker pada wajah tidak menutupi mulut atau memasang secara sembarangan, maka orang tersebut dianggap tidak menggunakan masker.



Gambar 1.9 Kondisi pintu terbuka

Pengujian Sistem

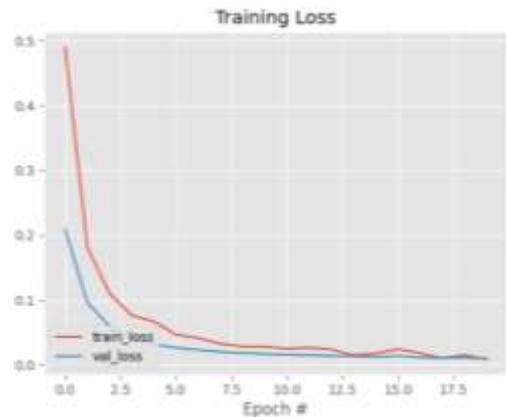
Tingkat *accuracy* (akurasi) dan *loss* (kegagalan) dalam melakukan *training* serta validasi Pada gambar 2.0 dijelaskan dari 20 epoch didapatkan tingkat akurasi yang tinggi diatas 0.9, baik dalam training maupun validasi.



Gambar 2.0 Tingkat Akurasi

Pada gambar 2.0 dijelaskan dari 20 epoch didapatkan tingkat kegagalan yang rendah

dibawah 0.1, baik dalam training maupun validasi.



Gambar 2.1 Tingkat Kegagalan

Pengujian sistem dibuat untuk mengetahui apakah hasil pelatihan menggunakan CNN dapat menghasilkan akurasi yang baik yang dapat digunakan untuk mengendalikan pintu akses masuk. Proses pengujian dilakukan dengan membuat 2 buah program terpisah. Program utama adalah program yang digunakan untuk mendeteksi dan mengkategorikan apakah gambar wajah yang ditangkap kamera merupakan gambar wajah bermasker ataupun gambar wajah tanpa masker. Program ini akan menggunakan data model jaringan CNN yang dihasilkan pada program pelatihan sebelumnya. Program ke-2 adalah program yang akan menampilkan gambar wajah tertentu dalam periode waktu +/- 7 detik.



Gambar 2.2 Menggunakan masker terdeteksi



Gambar 2.3 Tanpa menggunakan masker



Gambar 2.4 Penggunaan masker yang salah terdeteksi tidak pakai masker

Pada saat program kedua dijalankan, maka gambar pada monitor akan menampilkan gambar sesuai kategori yang diuji. Kategori yang diuji ada 3 yakni:

1. Gambar Wajah dengan masker standard, memiliki warna yang konstan, tidak bermotif dan tidak bergambar.
2. Gambar wajah tidak bermasker
3. Gambar wajah dengan masker bermotif, memiliki bentuk dan warna yang bervariasi, bergambar wajah, mulut ataupun gambar bentuk lainnya.

Pengujian Alat

Dari hasil pengujian didapatkan hasil seperti pada tabel 1.1, diketahui dari semua kondisi ada 2 kondisi yang tidak lolos, yaitu ketika ada sebuah gambar wajah aplikasi tetap mendeteksi bahwa itu adalah wajah, dan yang kedua adalah apabila masker hanya menutupi hidung saja aplikasi tetap menganggap penggunaan masker sudah benar, kedua kondisi ini tidak bisa memenuhi ekspektasi sehingga hasilnya *false*.

Tabel 1.1 Test case scenario prototype Mask Detector

Test case	Pre condition	Expented result	Real Result
Bunyi Alarm	Aplikasi mendeteksi user tidak menggunakan masker	Alarm berbunyi	True
	Aplikasi mendeteksi user menggunakan masker	Alarm tidak berbunyi	True
Deteksi wajah dan masker	Tidak ada user didepan kamera	Tidak mendeteksi wajah	True
	1 user didepan kamera	Mendeteksi 1 wajah	True
	2 user atau lebih didepan kamera	Tidak mendeteksi wajah	True
	Ada gambar atau pola berbentuk wajah didepan kamera	Tidak mendeteksi wajah	False
	Masker menutupi hidung dan mulut	Aplikasi mendeksi penggunaan masker dari user dan alarm tidak berbunyi	True
	Masker tidak menutupi hidung dan mulut	Aplikasi tidak mendeksi user menggunakan masker	True

		serta alarm berbunyi	
	Masker menutupi hidung tetapi tidak menutupi mulut	Aplikasi mendeksi penggunaan masker yang salah dan alarm berbunyi	False
	Masker menutupi mulut tetapi tidak menutupi hidung	Aplikasi mendeksi penggunaan masker yang salah dan alarm berbunyi	True
	User memakai masker berwarna seperti kulit dan menutupi hidung dan mulut	Aplikasi mendeksi penggunaan masker dari user dan alarm tidak berbunyi	True
	User memakai masker berwarna dan menutupi hidung dan mulut	Aplikasi mendeksi penggunaan masker dari	True

3. REFERENSI

Ariyanto, Mochammad & Haryanto, Ismoyo & Setiawan, Joga & Muna, Munadi & Radityo, M.. (2019). Real-Time Image Processing Method Using Raspberry Pi for a Car Model. 46-51.

Coronavirus Study Group,” Nat. Microbiol., pp. 1–15, Feb. 2020. WHO, “WHO Coronavirus Disease (COVID-19) Dashboard

Febtriko, A. (2017). Sistem Kontrol Perternakan Ikan Dengan Menggunakan Mikrokontroller Berbasis Android. *Rabit: Jurnal Teknologi Dan Sistem Informasi Univrab*, 2(1), 140–149. <https://doi.org/10.36341/rabit.v2i1.148>

Kurniawan, M. B. (2021). POLITIK HUKUM PEMERINTAH DALAM PENANGANAN PANDEMI COVID-19 (Government Legal

L. Wisesa, “OpenCV Face Recognition Berbasis Algoritma Haar Cascade,” Purwadhika Connect, 2019. [Online].

		user dan alarm tidak berbunyi	
--	--	-------------------------------	--

4. KESIMPULAN

1. Sistem kecerdasan buatan modern dan algoritma machine learning telah merevolusi pendekatan terhadap tantangan ilmiah dan teknologi di berbagai bidang. Saat ini, teknik Deep Learning (DL) dan Machine Learning (ML) telah menjadi algoritma pada berbagai alat yang berguna dalam memecahkan masalah.
2. Dalam model deteksi masker wajah secara *real-time* yang dihasilkan, metode dengan *Deep Learning* digunakan untuk mendeteksi wajah dan menentukan apakah orang tersebut memakai masker atau tidak. Dari tabel hasil percobaan, terlihat bahwa sistem pendeteksi wajah secara real time memiliki akurasi yang tinggi dalam mendeteksi masker, hal ini membantu untuk mengendalikan penyebaran COVID-19 di tempat-tempat umum dengan cara mencegah orang masuk tanpa menggunakan masker.

Available:

<https://medium.com/purwadhikaconnect/opencv-face-recognitionberbasis-algoritma-haar-cascade-1a5e23d9e8cb>. [Accessed: 26-Jun-2022].

Lambacing, M. M., & Ferdiansyah, F. (2020). Rancang Bangun New Normal Covid-19 Masker Detektor Dengan Notifikasi Telegram Berbasis Internet of Things. *Dinamik*, 25(2), 77–84. <https://doi.org/10.35315/dinamik.v25i2.8070>

M. Kim, D. Lee, and K.-Y. Kim, “System architecture for realtime face detection on analog video camera,” *Int. J. Distrib. Sens.Networks*, vol. 11, no. 5, p. 251386, 2015.

- Najiyah, I., & Topiq, S. (2021). Klasifikasi Jenis Kendaraan Roda Empat, 3(2), 199–206.
- Putri, N. I. (2020). Deep Learning Dan Teknologi Big Data Untuk Keamanan IOT. *COMPUTING/ Jurnal Informatika*, 7(1), 48–73.
- R. K. Sinuraya, D. P. Destiani, I. M. Puspitasari, and A. Diantini, “Pengukuran Tingkat Kepatuhan Pengobatan Pasien Hipertensi di Fasilitas Kesehatan Tingkat Pertama di Kota Bandung,” *Indones. J. Clin. Pharm.*, vol. 7, no. 2, pp. 124–133, 2018.
- S. Susanto, F. A. Putra, R. Analia and I. K. L. N. Suciningtyas, (2020) "The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam," 2020 3rd International Conference on Applied Engineering (ICAIE), Batam, Indonesia, pp. 1-5.
- Susilo, A., Rumende, C. M., Pitoyo, C. W., Santoso, W. D., Yulianti, M., Herikurniawan, H., Yunihastuti, E. (2020). Coronavirus Disease 2019: Tinjauan Literatur Terkini. *Jurnal Penyakit Dalam Indonesia*, 7(1), 45. <https://doi.org/10.7454/jpdi.v7i1.415>
- V. K. Gudipati, O. R. Barman, M. Gaffoor, and A. Abuzneid, “Efficient facial expression recognition using adaboost and haar cascade classifiers,” in 2016 Annual Connecticut Conference on Industrial Electronics, Technology & Automation (CT-IETA), 2016, pp. 1–4