

APLIKASI FACE MASK DETECTION MENGGUNAKAN NEURAL NETWORK MOBILENETV2 BERBASIS ANDROID

Azizulhaq Dharmaputra¹⁾, Margi Cahyanti²⁾, M Ridwan Dwi Septian³⁾, dan Ericks Rachmat Swedia⁴⁾

^{1,3}Teknik Informatika, Universitas Gunadarma

^{2,4}Sistem Informasi, Universitas Gunadarma

^{1,2,3,4}Jalan Margonda Raya No.100, Pondok Cina, Depok

E-mail : azizdharma18@gmail.com¹⁾, margi@staff.gunadarma.ac.id²⁾, ridwandwiseptian@staff.gunadarma.ac.id³⁾, ericks_rs@staff.gunadarma.ac.id⁴⁾

ABSTRAK

Adanya pandemi Covid-19 memberikan dampak yang luar biasa kepada hampir seluruh masyarakat di dunia, tidak terkecuali di Indonesia di mana Kementerian Kesehatan serta pemerintah yang terkait menerbitkan protokol kesehatan guna meminimalisasikan paparan yang diakibatkan oleh virus Covid-19. Salah satu protokol kesehatan tersebut yaitu menganjurkan masyarakat untuk selalu menggunakan masker saat beraktivitas baik itu di dalam maupun di luar ruangan. Dalam konteks ini, semua masyarakat dapat memainkan perannya masing-masing dengan berkontribusi untuk memerangi virus ini. Hal ini merupakan kesempatan yang bagus untuk para peneliti dalam bidang teknologi informasi untuk memanfaatkan kemajuan teknologi dalam membantu memerangi penyebaran virus corona, seperti pembuatan aplikasi untuk mendeteksi penggunaan masker. Penelitian ini membuat sebuah aplikasi untuk mendeteksi apakah objek yang terdapat pada preview image menggunakan masker atau tidak menggunakan masker. Pengumpulan data *training* dilakukan dengan pengambilan data dari situs *kaggle* berjumlah 1000 data citra yang terdiri dari 500 data citra menggunakan masker dan 500 data citra tidak menggunakan masker. Hasil yang diperoleh berdasarkan uji coba yang dilakukan terhadap 60 orang yang tidak terdapat dalam data *training*. Arsitektur *MobileNetV2* digunakan dalam penelitian ini dan diimplementasikan dalam bahasa pemrograman *Kotlin* yang dapat digunakan oleh semua *smartphone* berbasis android. Akurasi pendeteksian penggunaan masker dari 30 data uji coba yang didapatkan adalah sebesar 90%.

Kata Kunci: Covid-19, Deep Learning, Face Mask, Mobilenet, Virus

1. PENDAHULUAN

Covid-19 merupakan penyakit yang disebabkan oleh virus corona dan menimbulkan gejala berupa gangguan pernapasan (Hasan, G., & Kelly, K., 2021). Penyakit ini menjadi sorotan karena kemunculannya di akhir tahun 2019 di Wuhan, China. Dengan adanya penyebaran virus corona, penduduk di seluruh dunia diharuskan untuk mencegah tertularnya virus corona ini dengan cara rutin mencuci tangan menggunakan sabun, menggunakan masker, menjaga jarak antar satu sama lain, dan lain-lain. Pandemi virus Corona telah mendorong penduduk di seluruh dunia untuk terjun ke tantangan baru. Dalam konteks ini kita semua dapat memainkan peran kita dengan berkontribusi untuk memerangi virus ini. Hal ini merupakan kesempatan yang bagus bagi para peneliti untuk memanfaatkan kemajuan teknologi dalam membantu memerangi penyebaran virus corona (Mubah, A. S., & Anabarja, S., 2020) seperti membuat aplikasi *Face Mask Detection*.

Penelitian yang dilakukan oleh (Das, A., Ansari, M.W., & Basak, R., 2020) yang berjudul "*Covid-19 Face Mask Detection Using Tensor Flow, Keras, and OpenCV*" yang bertujuan untuk menyajikan pendekatan yang disederhanakan menggunakan beberapa paket

Machine Learning dengan metode untuk mendeteksi wajah dari gambar yang kemudian diidentifikasi apakah menggunakan masker atau tidak. Dengan menggunakan metode ini diperoleh akurasi hingga 95.77% dan 94.58 masing-masing pada dua dataset yang berbeda. Mereka mengeksplorasi nilai parameter yang dioptimalkan menggunakan model *Sequential Convolutional Neural Network* untuk mendeteksi keberadaan masker.

Arsitektur *MobileNetV2* digunakan oleh (S.A. Sanjaya & S. Adi Rakhmawan, 2020) dengan penelitian yang berjudul "*Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic*" yang bertujuan untuk mendeteksi masker wajah dan nantinya dapat digunakan oleh pihak yang berwenang untuk melakukan mitigasi, evaluasi, pencegahan, dan perencanaan tindakan terhadap COVID-19. Model arsitektur yang dibangun ini dapat mendeteksi orang yang menggunakan masker dan tidak menggunakan masker dengan akurasi 96.85%. Setelah model diterapkan di 25 kota dari berbagai sumber citra, persentase penduduk yang menggunakan masker di kota memiliki korelasi yang kuat dengan indeks kewaspadaan COVID-19 sebesar 0.62.

Arsitektur MTCNN dilakukan oleh (A. Joshi, et.al., 2020) dengan penelitian "*Deep Learning Framework to*

Detect Face Masks from Video Footage” yang bertujuan untuk mendeteksi citra masker wajah dalam video dengan menggunakan *Deep Learning* untuk mengidentifikasi wajah dan landmark wajah yang sesuai dengan yang ada dalam *frame* video menjadi citra. Metodologi yang diusulkan menunjukkan keefektifannya dalam mendeteksi masker wajah dengan mencapai presisi, daya ingat, dan akurasi yang tinggi.

Arsitektur ResNet-50 dilakukan (Mandal, B., Okeukwu, A., & Theis, Y., 2021) dengan penelitian “*Masked Face Recognition using ResNet-50*” yang bertujuan untuk menyelidiki masalah penggunaan masker wajah dengan mengembangkan model berbasis *Deep Learning* yang mampu mengidentifikasi orang-orang dengan masker wajah secara akurat. Arsitektur berbasis ResNet-50 bekerja dengan baik dalam mengenali wajah yang menggunakan masker. Hasilnya dapat berintegrasi dengan mulus ke dalam program pengenalan wajah yang dirancang untuk mendeteksi wajah untuk tujuan verifikasi keamanan.

Arsitektur Hybrid *Deep Learning* juga dilakukan oleh (Wang, B., Zhao, Y., & Chen, C.L., 2021) dengan penelitian “*Hybrid Transfer Learning and Broad Learning System for Wearing Mask Detection in the COVID-19 Era*” yang bertujuan untuk mendeteksi orang-orang yang menggunakan masker dengan menggunakan pendekatan dua tahap *Hybrid Transfer Learning* dan *Broad Learning System*. Setelah dilakukan pengumpulan data dan dilakukan uji coba didapatkan pendekatan yang mencapai akurasi keseluruhan 97.32% untuk *scene* yang sederhana dan akurasi keseluruhan 91.13% untuk *scene* yang kompleks.

Penelitian ini dilakukan menggunakan arsitektur *Deep Learning MobileNetV2* yang merupakan arsitektur terbaru dari *MobileNet* dan diimplementasikan menggunakan bahasa pemrograman *Kotlin* yang bersumber dari bahasa pemrograman *Java* (Hansun, S., Kristanda, M. B., & Saputra, M. W., 2018), sehingga pendeteksian dapat dilakukan di mana saja dan kapan saja dengan menggunakan *smartphone* yang berbasis sistem operasi Android.

2. RUANG LINGKUP

Adapun yang menjadi batasan masalah dalam implementasi arsitektur *MobileNetV2* ini di antaranya sebagai berikut:

1. Hanya dapat mendeteksi satu orang per waktu.
2. Aplikasi ini hanya dapat berjalan pada *smartphone* yang memiliki sistem operasi Android.
3. *Output* berupa tulisan di dalam border dengan warna yang mendeteksi penggunaan masker atau tidak.

3. BAHAN DAN METODE

Pada bagian ini menjelaskan bahan-bahan dan metode apa yang akan digunakan dalam penelitian ini, diantaranya:

3.1 Neural Network

Neural Network merupakan sebuah teknologi komputasi, tidak memberikan suatu keajaiban tetapi jika digunakan secara tepat akan menghasilkan suatu hasil yang luar biasa. Kemampuan *Neural Network* dalam menyelesaikan masalah yang rumit telah dibuktikan dalam berbagai macam penelitian, seperti analisis data, meteorologi, pengenalan pola, sistem kontrol, pendeteksian, prediksi, dan sebagainya.

Neural Network atau yang sering disebut juga dengan jaringan saraf tiruan merupakan bagian dari *Machine Learning* dan merupakan inti dari algoritma *Deep Learning*. Struktur *Neural Network* ini meniru cara kerja otak manusia dimana neuron biologis saling memberi sinyal (Chen, T., Peng, L., Yin, X., Rong, J., Yang, J., & Cong, G., 2020).

3.2 Masker

Masker merupakan alat bantu perlindungan pernafasan yang digunakan untuk melindungi individu dari zat-zat berbahaya yang berada di udara. Perlindungan pernafasan atau masker tidak dimaksudkan untuk menggantikan metode pilihan yang dapat menghilangkan penyakit, tetapi digunakan untuk melindungi secara memadai kepada penggunanya (Untari, S., & Kodiyah, N., 2020).

Masker mempunyai beberapa tipe yang salah satunya adalah *Air Purifying Respirators* (APR). Masker ini menggunakan *filter* atau *cartridge* yang dapat mencegah zat-zat berbahaya yang berada di udara. Masker mempunyai jenis yang dapat melindungi bergantung dengan tingkat bahaya dari paparan aerosol atau partikel yang berada di udara. (Rosyanti, L., & Hadi, I., 2020) masker memiliki beberapa jenis di antaranya sebagai berikut (Atmojo, J. T., Iswahyuni, S., Rejo, R., Setyorini, C., Puspitasary, K., Ernawati, H., & Mubarak, A. S., 2020):

1. Masker Untuk Pekerja Kesehatan

1) Masker N95

Masker N95 merupakan masker yang terpasang erat di sekitar wajah. Masker N95 dapat menyaring 95% atau lebih partikel terkecil di udara.

2) Masker Bedah

Masker bedah seringkali berwarna biru dengan pinggiran putih. Masker ini melindungi dari tetesan besar yang berasal dari batuk atau bersin orang sakit, tetapi terlalu longgar untuk melindungi dari semua kuman dan tidak dapat memblokir partikel terkecil yang mungkin membawa virus corona.

2. Masker Untuk Pekerja Non-Kesehatan

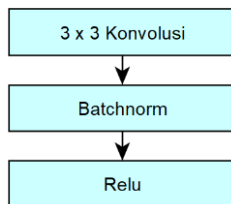
Masker kain merupakan masker paling baik untuk orang yang tidak bekerja di bidang kesehatan. Masker kain pun dapat dibuat sendiri dengan memperhatikan beberapa hal di antaranya sebagai berikut.

- 1) Kain dapat dijahit dan diikat di sekitar wajah atau melipatnya di sekitar ikat rambut untuk membuat simpul di telinga.
- 2) Menggunakan setidaknya material dua lapis.
- 3) Dapat menambahkan kantong untuk filter.
- 4) Menambahkan pita tembaga atau kawat pada bagian hidung masker agar lebih pas.

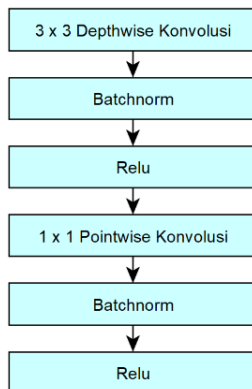
3.3 MobileNetV1

MobileNet merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang dapat digunakan untuk mengatasi kebutuhan komputasi pada sumberdaya yang berlimpah (Qin, Z., Zhang, Z., Chen, X., Wang, C., & Peng, Y., 2018), sehingga *MobileNet* cocok digunakan untuk melakukan *training* pada *dataset* yang berjumlah besar.

Perbedaan mendasar antara arsitektur *MobileNet* dan arsitektur CNN pada umumnya adalah penggunaan lapisan konvolusi dengan ketebalan filter yang sesuai dengan ketebalan dari *input image* (Hendriyana, H., & Maulana, Y. H., 2020). Perbedaan arsitektur CNN dan *MobileNet* yang membagi konvolusi menjadi *depth wise* dan *point wise* pada gambar 1 dan gambar 2.



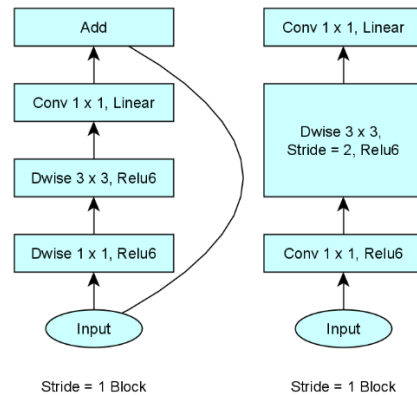
Gambar 1. Arsitektur Lapisan Konvolusi CNN



Gambar 2. Arsitektur Lapisan Konvolusi MobileNet

3.4 MobileNetV2

MobileNetV2 yang dirilis pada tahun 2018 (Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018)) masih menggunakan *depth wise* dan *point wise* konvolusi seperti pada versi sebelumnya, namun *MobileNetV2* menambahkan dua fitur baru yaitu *linear bottleneck* dan *shortcut connections* antar layer. Struktur dasar dari arsitektur *MobileNetV2* dapat di lihat pada Gambar 3.



Gambar 3. Arsitektur MobileNetV2
 (sumber : Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018))

Berdasarkan Gambar 3 pada *MobileNetV2* terdapat dua tipe blok. Yang pertama *blok residual* dengan *stride of 1* dan blok kedua dengan *stride of 2* untuk *downsizing* citra. Terdapat tiga *layers* untuk tiap-tiap blok. *Layer* pertama adalah konvolusi 1x1 dengan ReLU6, *layer* kedua adalah *depth* konvolusi, dan *layer* ketiga adalah konvolusi 1x1.

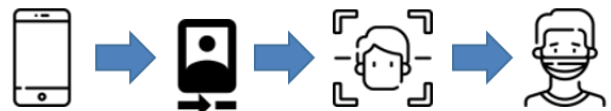
3.5 Kotlin

Kotlin biasanya dikaitkan dengan pengembangan Android. *Kotlin* adalah bahasa *Java Virtual Machine* (JVM), sehingga kompiler akan menjalankan *byte code* Java. Karena itu, tentu saja kode *Kotlin* dapat memanggil kode Java, dan sebaliknya (Samuel, S., & Bocutiu, S., 2017).

4. PEMBAHASAN

Tahapan perancangan dari aplikasi yang akan dibuat, berupa rancangan tampilan dan diagram alur. Pada Gambar 4 merupakan diagram alur dengan keterangan.

1. Rancangan Tampilan *Capture Kamera* dan *Preview Image*
2. *Pseudo code*
3. *Output*
4. Hasil Uji Coba



Gambar 4. Diagram Alur

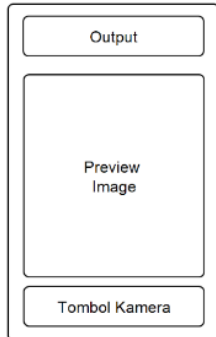
2.1. Rancangan Tampilan *Capture Kamera* dan *Preview Image*

Rancangan tampilan *capture* kamera pada aplikasi terlihat seperti pada Gambar 5. Terdapat *icon* kamera yang dapat digunakan oleh user untuk memilih apakah ingin menggunakan kamera depan atau kamera belakang.



Gambar 5. Rancangan Tampilan Capture Kamera

Rancangan tampilan *preview image* pada aplikasi terlihat seperti pada Gambar 6. Terlihat *preview* dari *image* yang di ambil dari kamera lalu di bagian *output* akan menampilkan tulisan *with mask* atau *without mask*.



Gambar 6. Rancangan Tampilan Preview Image

2.2. Pseudo Code

Berikut merupakan algoritma dari cara kerja aplikasi *Face Mask Detection*.

```

Facemask Detection
If tekan tombol front camera then
Preview front camera
  Deteksi figur pada camera
  If figur menggunakan masker then
    Tampilkan teks "with mask"
  Else
    Tampilkan teks "without mask"
Else
  Preview rear camera
  Deteksi figur pada camera
  If figur menggunakan masker then
    Tampilkan teks "with mask"
  Else
    Tampilkan teks "without mask"
Finish
  
```

2.3. Proses Learning

Dataset diunduh dari *Kaggle* dengan url (<https://www.kaggle.com/vijaykumar1799/face-mask-detection>) yang berjumlah 1000 data citra, 500 citra adalah objek yang menggunakan masker dan 500 citra merupakan objek yang tidak menggunakan masker. Langkah awal dari proses melatih *dataset* ini adalah menginisialisasikan parameter *learning rate*, dan *number*

of epochs. Hasil *training* pada gambar 7 merupakan hasil dari kode program dengan *learning rate* sebesar 0.0001, *epochs* sebesar 20, dan *batch size* sebesar 32, serta akan mengambil file citra dari *dataset*, dan akan mengategorikannya menjadi dua yaitu *with mask* atau *without mask*.

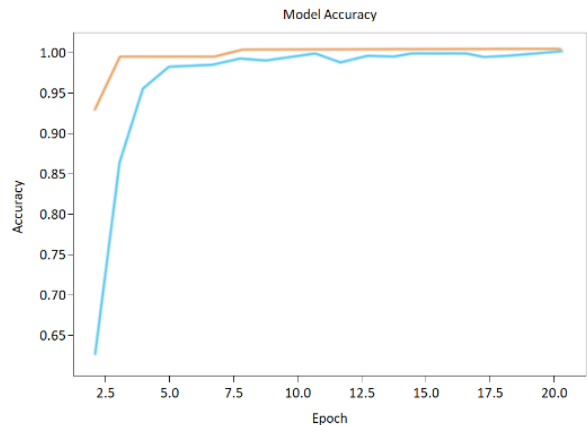
Langkah selanjutnya melakukan *pre-processing image* yang diambil dari direktori citra tersebut berada dan berdasarkan kategorinya. Lalu citra tersebut nantinya akan berubah menjadi ukuran 224 x 224 pixel dan akan diberikan label *with mask* dan label *without mask* berdasarkan kategori folder citra tersebut disimpan. *Training dataset* sesuai dengan ketentuan inisialisasi yang sudah di deskripsi pada blok program sebelumnya.

```

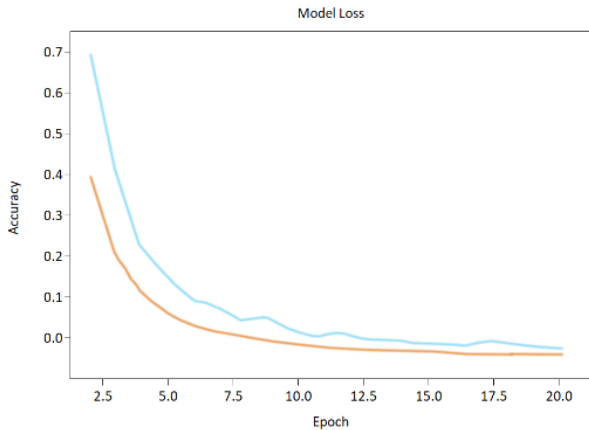
[INFO] training head...
Epoch 1/20
15/15 [=====] - 22s 1s/step - loss: 0.7683 - accuracy: 0.5491 - val_loss: 0.4078 - val_accuracy: 0.9274
Epoch 2/20
15/15 [=====] - 17s 1s/step - loss: 0.4432 - accuracy: 0.8360 - val_loss: 0.2342 - val_accuracy: 0.9919
.....
Epoch 19/20
15/15 [=====] - 17s 1s/step - loss: 0.0249 - accuracy: 0.9949 - val_loss: 0.0150 - val_accuracy: 1.0000
Epoch 20/20
15/15 [=====] - 18s 1s/step - loss: 0.0272 - accuracy: 0.9969 - val_loss: 0.0145 - val_accuracy: 1.0000
15/15 [=====] - 18s 1s/step - loss: 0.0686 - a
  
```

Gambar 7. Training Dataset

Hasil dari *training* dapat dilihat dalam dua model, model akurasi dan model *loss*. Akurasi menunjukkan hasil *training* dengan nilai maksimum 100%, yang didapatkan dari perbandingan antara data validasi dengan data *training*, *Loss* menunjukkan *training* yang telah mencapai batas *learning-rate* yang diinginkan sebesar 0.0001. Aplikasi selanjutnya akan menampilkan grafik hasil *training dataset* terhadap model akurasi dan *loss* yang ditunjukkan pada gambar 8 dan 9.



Gambar 8. Grafik Model Akurasi

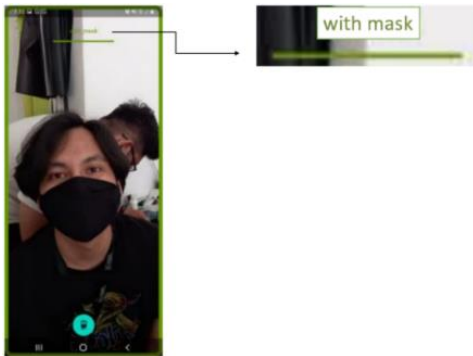


Gambar 9. Grafik Model Loss

Dari gambar 8 dan 9, terlihat bahwa proses melatih data dari 1000 data citra, telah mencapai akurasi maksimum sebesar 100% atau *error loss* di angka 0.0001 pada *epoch* ke-20.

2.4. Output

Pada tahapan akhir membuktikan apakah aplikasi yang dirancang telah sesuai dengan rancangan pada sistem alur, di mana akan menampilkan keluaran dari objek yang menggunakan master atau tidak menggunakan masker, gambar 10 dan gambar 11 dengan *output* yang menampilkan menggunakan masker dan tidak menggunakan masker.



Gambar 10. Tampilan Objek Menggunakan Master















Gambar 11. Tampilan Objek Tidak Menggunakan Masker

2.5. Hasil Uji Coba

Uji coba aplikasi dilakukan pada 60 objek yang berbeda yang tidak terdapat dalam *dataset training*. Dari 60 objek ini dibagi menjadi dua, 30 objek menggunakan master dan 30 objek yang tidak menggunakan master, tabel 2 dan tabel 3 di merupakan hasil dari uji coba yang dilakukan.

Tabel 2. Hasil Uji Coba Menggunakan Masker

No	Hasil Uji Coba	Penjelasan
1		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
2		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
3		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
4		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
5		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
6		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
7		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
8		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
9		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
10		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%

11		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
12		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
13		Dapat dilihat bahwa <i>progress bar</i> tidak terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya kurang lebih sebesar 90%
14		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya kurang lebih sebesar 50%
15		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
16		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
17		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
18		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
19		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
20		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
21		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
22		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
23		Dapat dilihat bahwa <i>progress bar</i> tidak terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya kurang lebih sebesar 90%

24		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
25		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
26		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
27		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
28		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
29		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
30		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%

Tabel 3. Hasil Uji Coba Tidak Menggunakan Masker

No	Hasil Uji Coba	Penjelasan
1		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
2		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
3		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
4		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
5		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
6		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%

7		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
8		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
9		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
10		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
11		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
12		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
13		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
14		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
15		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
16		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
17		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
18		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
19		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
20		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%

21		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
22		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
23		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
24		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
25		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
26		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
27		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
28		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
29		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%
30		Dapat dilihat bahwa <i>progress bar</i> terisi penuh yang berarti aplikasi mendeteksi tingkat keakuratannya sebesar 100%

Dari hasil uji coba, untuk objek yang tidak menggunakan masker terlihat bahwa aplikasi dapat mendeteksi keseluruhan data uji coba. Untuk objek yang menggunakan masker, aplikasi dapat mendeteksi penggunaan masker di 27 data dari 30 data yang diuji coba. Tiga data objek, data bernomor 13, 14 dan 23 aplikasi tidak dapat menampilkan pendeteksian penggunaan masker yang sempurna. Objek nomor 13 dan 14 menggunakan aksesoris kepala (topi) yang hampir menutupi keseluruhan citra, sedangkan pada objek nomor 23, aplikasi tidak dapat mendeteksi dengan sempurna karena warna masker yang mirip dengan warna pada background citra.

Akurasi aplikasi dapat dihitung menggunakan formula, data yang dapat dideteksi dibagi dengan

keseluruhan data uji coba, sehingga tingkat akurasi yang didapatkan adalah 27 dibagi 30, sebesar 90%.

5. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, aplikasi yang dibangun dapat menangkap wajah objek yang menggunakan masker dan tidak menggunakan masker pada aplikasi *Face Mask Detection*. Pengujian *real-time* yang telah dilakukan, aplikasi menghasilkan tingkat akurasi yang tinggi yaitu sebesar 90%.

6. SARAN

Adapun saran yang dianjurkan agar dapat menjadi masukan dalam pengembangan bagaimana menangkap lebih banyak wajah dalam layar, dapat menggunakan kamera *night vision* karena pencahayaan juga menjadi salah satu faktor dalam akurasi pendeteksian terutama di malam hari, dan meningkatkan akurasi untuk warna masker yang mirip dengan *background* warna citra.

7. DAFTAR PUSTAKA

- Atmojo, J. T., Iswahyuni, S., Rejo, R., Setyorini, C., Puspitasary, K., Ernawati, H., & Mubarak, A. S. (2020). Penggunaan Masker Dalam Pencegahan Dan Penanganan Covid-19: Rasionalitas, Efektivitas, Dan Isu Terkini. *Avicenna: Journal of Health Research*, 3(2).
- Chen, T., Peng, L., Yin, X., Rong, J., Yang, J., & Cong, G. (2020, September). Analysis of user satisfaction with online education platforms in China during the COVID-19 pandemic. In *Healthcare* (Vol. 8, No. 3, p. 200). Multidisciplinary Digital Publishing Institute.
- Das, A., Ansari, M. W., & Basak, R. (2020, December). Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. In *2020 IEEE 17th India Council International Conference (INDICON)* (pp. 1-5). IEEE.
- Hansun, S., Kristanda, M. B., & Saputra, M. W. (2018). *Pemrograman Android dengan Android Studio IDE*. Yogyakarta: Andi.
- Hasan, G., & Kelly, K. (2021, May). Perancangan Promotion Mix pada PT. Rambun Pratama Metalindo. In *ConCEPT-Conference on Community Engagement Project* (Vol. 1, No. 1, pp. 2590-2595).
- Hendriyana, H., & Maulana, Y. H. (2020). Identification of Types of Wood using Convolutional Neural

Network with Mobilenet Architecture. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(1), 70-76.

- Joshi, A. S., Joshi, S. S., Kanahasabai, G., Kapil, R., & Gupta, S. (2020, September). Deep learning framework to detect face masks from video footage. In *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)* (pp. 435-440). IEEE.
- Mandal, B., Okeukwu, A., & Theis, Y. (2021). Masked Face Recognition using ResNet-50. *arXiv preprint arXiv:2104.08997*.
- Mubah, A. S., & Anabarja, S. (2020). From Big Data to Mask Diplomacy: Taiwan's Effort to Strengthen International Role. *Jurnal Global & Strategis*, 14(2), 15-28.
- Qin, Z., Zhang, Z., Chen, X., Wang, C., & Peng, Y. (2018, October). Fd-mobilenet: Improved mobilenet with a fast downsampling strategy. In *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 1363-1367). IEEE.
- Rosyanti, L., & Hadi, I. (2020). Dampak psikologis dalam memberikan perawatan dan layanan kesehatan pasien COVID-19 pada tenaga profesional kesehatan. *Health Information: Jurnal Penelitian*, 12(1), 107-130.
- Samuel, S., & Bocutiu, S. (2017). *Programming kotlin*. Packt Publishing Ltd.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).
- Sanjaya, S. A., & Rakhmawan, S. A. (2020, October). Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic. In *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)* (pp. 1-5). IEEE.
- Untari, S., & Kodiyah, N. (2020). Gambaran Penggunaan Masker Di Masa Pandemi Covid-19 Pada Masyarakat Di Kabupaten Grobogan. *The Shine Cahaya Dunia Kebidanan*, 5(2).
- Wang, B., Zhao, Y., & Chen, C. P. (2021). Hybrid Transfer Learning and Broad Learning System for Wearing Mask Detection in the COVID-19 Era. *IEEE Transactions on Instrumentation and Measurement*, 70, 1-12.