Deteksi Penggunaan Masker Pada Citra Menggunakan RetinaFace dengan MobileNetV2

1st Ghilman Muhammad Zaki
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
ghilmanzaki@student.telkomuniversity.ac.id

2nd Febryanti Sthevanie
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
sthevanie@telkomuniversity.ac.id

3rd Kurniawan Nur Ramadhani
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
kurniawannr@telkomuniversity.ac.id

***Abstrak* — Pada penelitian ini dibentuk sistem pendeteksi masker menggunakan RetinaFace sebagai face detection dan MobileNetV2 sebagai sistem klasifikasi penggunaan masker pada wajah. Alur sistem diawali dengan sistem pendeteksi wajah menggunakan RetinaFace. RetinaFace mengambil bagian wajah pada citra dan diteruskan ke model sistem klasifikasi penggunaan masker menggunakan model MobileNetV2. Model MobileNetV2 dilakukan training dengan training set yang terdiri dari 23.115 citra wajah bermasker dan tidak bermasker dan dilakukan testing dengan testing set yang terdiri dari 6.998 citra wajah bermasker dan tidak bermasker. Hasil akurasi yang didapatkan oleh model MobileNetV2 sebesar 99,3%. Namun model MobileNetV2 masih dapat memberikan hasil prediksi yang salah untuk citra wajah dari citra orang yang diambil dari jarak yang jauh.**

***Kata kunci*— COVID-19, retinaface, mobilenetv2**

I. PENDAHULUAN

A. Latar Belakang

Coronavirus 19 (COVID-19) telah mengakibatkan perubahan pada dunia secara signifikan pada aspek kehidupan manusia seperti kesehatan, pasokan makanan, pendidikan, transportasi hingga politik. Gejala yang ditimbulkan oleh orang yang terinfeksi oleh COVID-19 dapat berupa penyakit pernapasan. Tercatat pada tanggal 13 Juli 2020 bahwa COVID-19 telah menyebar dengan cepat ke 214 negara dan Negara Amerika Serikat (AS) menjadi negara dengan kasus penularan COVID-19 tertinggi dengan jumlah kasus sebesar 3,4 Juta kasus[1]. Untuk mengantisipasi terjadinya pelanggaran dalam penggunaan masker, maka diperlukan suatu sistem berupa sistem pendeteksi penggunaan masker pada setiap orang.

Berdasarkan penelitian sebelumnya, sistem pendeteksi penggunaan masker dengan menggunakan SSD sebagai face detection dan MobileNetV2 sebagai face mask classification berhasil memberikan akurasi sebesar 92.6% dan f1-score sebesar 93%[2]. Namun, jumlah false negative yang dihasilkan sebesar 163 di mana jumlah tersebut masih dapat dikurangi dengan diadakannya improvisasi.

Berdasarkan penelitian lainnya, RetinaFace mengungguli pendeteksi lainnya dengan menunjukkan akurasi deteksi wajah sebesar 91,4% di hard subsets[3]. RetinaFace memiliki keunggulan tersendiri seperti dapat mendeteksi wajah dari jarak yang jauh, wajah yang terhalangi sebagian objek dan juga wajah yang tampak dari samping[3], [4]. Sehingga RetinaFace cocok untuk dijadikan sebagai pendeteksi wajah dengan wajah yang tertutup oleh masker. Oleh karena itu, penelitian ini menggunakan model deep learning CNN dengan arsitektur MobileNetV2 dan RetinaFace sebagai pendeteksi wajah pada citra.

B. Topik dan Batasannya

Topik dan batasan pada penelitian ini adalah membangun sistem yang dapat mendeteksi penggunaan masker. Metode yang diuji pada penelitian ini adalah RetinaFace untuk mendeteksi wajah dan MobileNetV2 untuk mengklasifikasikan wajah yang menggunakan masker dan tidak menggunakan masker. Penelitian ini hanya berfokus pada face mask classification karena RetinaFace hanya sebatas pendeteksi wajah dan juga tidak adanya perubahan model yang dilakukan pada RetinaFace, sedangkan MobileNetV2 perlu dilakukan training menggunakan dataset citra wajah bermasker dan tidak bermasker.

C. Tujuan

Tujuan penelitian ini adalah untuk membangun sistem pendeteksi penggunaan masker pada wajah berbasis CNN dengan arsitektur MobileNetV2 sebagai klasifikasi penggunaan masker pada wajah dan didukung dengan RetinaFace untuk mendeteksi wajah.

D. Organisasi Tulisan

Studi terkait membahas penelitian yang telah dilakukan sebelumnya dan menjelaskan model yang digunakan berupa RetinaFace dan MobileNetV2. Sistem yang dibangun membahas kegiatan yang dilakukan dalam membangun sistem. Evaluasi membahas hasil serta analisis dari penelitian

yang telah dilakukan. Kesimpulan membahas kesimpulan yang didapat dari hasil penelitian ini.

II. KAJIAN TEORI

A. Studi Terkait

1. Studi literatur

Pada tahun 2020, banyak penelitian yang telah dilakukan untuk membangun sistem deteksi penggunaan masker. Das Arjya et al. membuat sistem deteksi masker menggunakan model sequential CNN dengan Tensorflow sebagai backend pada arsitektur Sequential CNN, keras sebagai implementasi pada seluruh layer di model CNN, dan OpenCV untuk perubahan ukuran citra dan pengkonversian warna pada citra. Hasil akurasi yang diberikan pada penelitian ini sebesar 95,77% pada dataset 1 dan 94,58% pada dataset 2. Besarnya jumlah neuron dan filter dapat mengakibatkan penurunan kinerja[5].

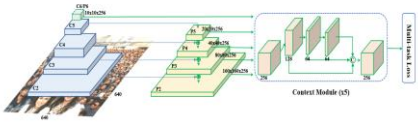
Lad AM et al. pada penelitiannya melakukan analisis terhadap arsitektur CNN seperti sequential CNN, VGG-16, dan MobileNetV2 dalam mengklasifikasikan penggunaan masker. hasil akurasi training terbesar didapatkan oleh MobileNetV2 dengan besar akurasi training sebesar 99,2%. MobileNetV2 memberikan hasil feature extraction yang sangat akurat dalam mendeteksi penggunaan masker pada citra wajah dan efektif dalam melakukan segmentasi citra[6].

Nagrath P et.al. membuat sistem deteksi masker secara real time menggunakan SSD sebagai face detection dan MobileNetV2 sebagai face mask classification. Penelitian ini melakukan training sebanyak 100 epoch dan menghasilkan akurasi sebesar 93% namun masih terdapat 163 false negative dan 2 false positive. Hasil perbandingan dengan arsitektur CNN seperti Le-Net 5, AlexNet, VGG-16, Resnet-50 menunjukkan bahwa nilai akurasi, f1-score, dan performa pada model SSDMNV2 memiliki nilai yang tinggi dibandingkan dengan model arsitektur CNN lainnya[2].

Deng J at al. membuat sistem deteksi wajah dengan pendekatan single shot multi-level localisation dengan nama RetinaFace. RetinaFace melakukan strategi multi-task learning untuk memprediksi confidence level, 5 face landmarks, bounding box, dan 3d face. RetinaFace memberikan nilai Average Precision (AP) yang paling baik di seluruh subsets pada validation set dan testing set dengan nilai 96.9% (Easy), 96.1% (Medium) dan 91.8% (Hard) untuk validation set, dan 96.3% (Easy), 95.6% (Medium) dan 91.4% (Hard) untuk test set.

2. RetinaFace

RetinaFace merupakan pendeteksi wajah *single stage* yang kuat dengan melakukan lokalisasi wajah berdasarkan piksel pada berbagai skala wajah. RetinaFace memanfaatkan *extra-supervised* dan *self-supervised multi-task learning*. RetinaFace menggunakan strategi *multi-task learning* secara bersamaan untuk memprediksi skor akurasi wajah, box wajah, lima *landmark* wajah, serta posisi 3d dan korespondensi pada setiap piksel wajah. RetinaFace terdiri dari 3 komponen utama yaitu *feature pyramid network*, *context module* dan *cascade multi-task loss*[3] seperti yang digambarkann pada Gambar 1.



GAMBAR 1.
Arsitektur RetinaFace

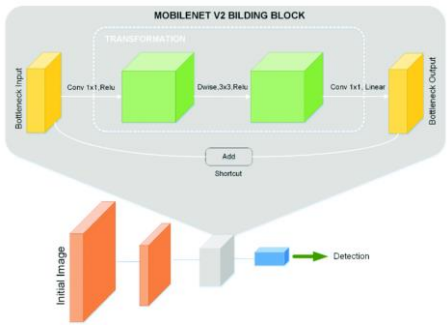
RetinaFace bekerja menggunakan *feature pyramid* level dari P2 sampai P6. P2 hingga P5 dihitung dari keluaran tahap residu ResNet yang sesuai menggunakan *top-down* dan *lateral connections* di dalamnya. P6 dihitung melalui *convolution layer* 3x3 dengan stride=2 pada C5. C1 hingga C5 merupakan jaringan klasifikasi yang terlatih dari dataset ImageNet-11k sedangkan P6 diinisialisasi secara acak menggunakan metode Xavier. *Feature Pyramid Network* menerima *input* berupa citra wajah dan memberikan menghasilkan *output* berupa lima *feature maps* dari skala yang berbeda[3].

Pada komponen selanjutnya, RetinaFace bekerja menggunakan *independent context modules* pada lima *feature pyramid levels* untuk meningkatkan bidang reseptif dan kekuatan pemodelan konteks yang kaku. Seluruh *convolution layer* 3x3 dalam *lateral connections* dan *context module* diganti dengan *deformable convolution network* (DCN) yang memperkuat kapasitas pemodelan konteks nonrigid[[3].

RetinaFace bekerja menggunakan *cascade regression* dengan *multi-task loss* untuk meningkatkan performa lokalisasi wajah[3].

3. MobileNetV2

Saat ini sudah banyak model yang difungsikan untuk kebutuhan klasifikasi objek. Salah satu model yang dipakai untuk proses klasifikasi objek adalah MobileNetV2. MobileNetV2 merupakan arsitektur model CNN yang ringan dan dapat diimplementasikan pada *mobile*. MobileNetV2 adalah hasil perkembangan dari model sebelumnya yaitu MobileNet. Tujuan dibentuknya MobileNet adalah untuk mengatasi penggunan *resource* yang besar. MobileNet menghadirkan *Depthwise Separable Convolutions* yang memecah atau membagi konvolusi menjadi dua *layer* yaitu *depthwise convolution* dan *pointwise convolution*. Seiring perkembangannya, MobileNetV2 menghadirkan *Linear Bottlenecks* dan *Inverted Residual* yang dapat meningkatkan performa MobileNet itu sendiri[7]. Berikut arsitektur MobileNetV2 yang digambarkan pada Gambar 2.



GAMBAR 2.
Arsitektur MobileNetV2

Komponen pertama yang dimiliki oleh MobileNetV2 adalah *Depthwise Separable Convolutions.* Pada komponen ini, terjadi proses filterisasi yang ringan dengan menerapkan filter konvolusi tunggal per *input* channel. *Layer* yang kedua disebut sebagai *pointwise convolution*. Pada *layer* ini terjadi proses membangun *feature* baru melalui komputasi kombinasi linear dari *input* channel. Hal ini dapat mengurangi komputasi dibandingkan dengan dengan *layer* tradisional. MobileNetV2 menggunakan 3 × 3 *depthwise separable convolutions* sehingga *computational cost* menjadi 8 atau 9 kali lebih kecil dibandingkan dengan *standard convolutions*[7].

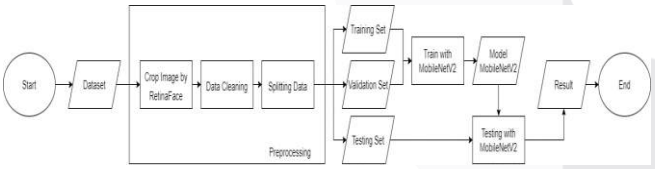
Komponen selanjutnya yaitu *Linear Bottleneck* yang merupakan *bottleneck* tanpa *last activation*. Menggunakan *non linear* pada *bottleneck* dapat merusak beberapa persen performa dan hilangnya beberapa informasi. Untuk mengatasi hal ini, konsep *linear bottleneck* diimplemenntasikan dengan menambahkan *linear bottleneck* ke dalam *convolutional blocks*[7].

Komponen terakhir yang diunggulkan oleh MobileNetV2 adalah *Inverted Residual*. *Inverted Residual* memiliki pendekatan *low, high, low*. Pada *layer* pertama operator yang digunakan adalah 1x1 *Convolution* dengan ReLU6. Hal ini bertujuan memperluas peta fitur *input* dengan dimensi yang rendah ke ruang berdimensi lebih tinggi yang cocok untuk aktivasi *non-linear*. Di *layer* kedua menggunakan 3x3 *depthwise* dengan ReLU6, dan *layer* yang terakhir menggunakan 1x1 *linear convolution*[7].

III. METODE

A. Sistem yang Dibangun

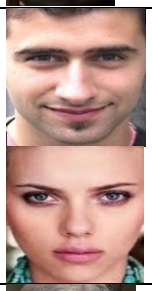
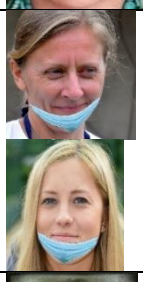
Penelitian ini bertujuan untuk membangun sistem deteksi masker menggunakan RetinaFace sebagai *face detection* dan MobileNetV2 untuk klasifikasi penggunaan masker. Berikut pada Gambar 3 yang menggambarkan alur pembangunan model klasifikasi penggunaan masker pada wajah.



GAMBAR 3.
Alur pembangunan model klasifikasi penggunaan masker pada wajah

B. Dataset

Penelitian ini menggunakan *dataset* yang didapatkan dari tiga sumber yaitu *dataset* 1 yang didapatkan dari situs *github*[8], *dataset* 2 didapatkan dari situs *kaggle*[9] dan dataset 3 yang didapatkan dari situs *kaggle*[10]. *Dataset* 1 digunakan untuk *testing set* dan *dataset* 2 dengan *dataset* 3 digunakan untuk *training set. Dataset* yang digunakan terbagi menjadi dua kelas seperti pada Gambar 4 yang terdiri dari kelas "*with_mask"* dan "*without_mask*"

With_Mask	Wajah dengan masker	
Without_mask	Wajah tanpa masker	
	Wajah dengan penggunaan masker yang salah	
	Wajah dengan kondisi hidung dan mulut tertutupi benda selain masker	

GAMBAR 4.
Citra wajah bermasker dan tidak bermasker pada *dataset*

Dataset 1 memiliki dua kelas yaitu kelas dengan nama "*with_mask*" dengan jumlah citra sebanyak citra 5.521 citra wajah yang menggunakan masker dan kelas dengan nama "*without_mas*k" dengan jumlah citra sebanyak 5.521 citra wajah tanpa menggunakan masker wajah tanpa menggunakan masker. *Dataset* 2 memiliki dua kelas juga yaitu kelas dengan nama "*with_mask*" dengan jumlah citra sebanyak 5.883 citra wajah yang menggunakan masker dan kelas dengan nama "*without_mask*" dengan jumlah citra sebanyak 5.908. *Dataset* 3 memiliki dua kelas yaitu kelas "*with_mask*" dengan jumlah citra sebanyak dan "*without_mask*" dengan jumlah citra sebanyak 5.747 citra wajah 4.789 citra wajah.

C. Prapemrosesan

Prapemrosesan data diawali dengan melakukan *cropping image* pada *dataset* 1 menggunakan RetinaFace untuk mendapatkan citra wajah. Lalu dilakukan proses *data cleaning* dengan memindahkan citra wajah yang seharusnya ada pada kelas "*with_mask*" ke dalam kelas "*without_mask*" serta meniadakan citra yang memiliki kualitas rendah dan citra yang memiliki ukuran dibawah 85x64 *pixel.* Setelah melalui proses *data cleaning*, dilakukan proses *resizing image* menjadi ukuran 224x224 *pixel* dan proses *splitting data* dengan membagi *dataset* menjadi *training set* dan

testing set dengan *ratio* 8:2. *Dataset* 1 menjadi *testing set* dan *dataset* 2 dengan *dataset* 3 menjadi *training set*. Sebagian citra pada *testing set* dipindahkan ke *training set* agar sesuai dengan *ratio* 8:2. Jumlah *testing set* menjadi 6.998 citra dan jumlah *training set* menjadi 23.115 citra.

D. Pembangunan Model

Pada penelitian ini, dilakukan proses *transfer learning* dengan menggunakan model MobileNetV2. Penelitian ini menggunakan *pretrained* MobileNetV2 dikarenakan untuk menghindari *computational cost* yang tidak perlu dan memanfaatkan *weight* yang telah dilakukan *training* dengan *dataset* ImageNet. Penelitian ini melakukan proses *transfer learning* pada model MobileNetV2 yang hanya dapat mengklasifikasikan penggunaan masker pada wajah untuk dua kelas saja yaitu "*with_mask*" dan "*without_mask*". Hal ini bertujuan untuk mengukur dengan hasil penelitian acuan yang memanfaatkan SSDMNV2 sebagai pendeteksi penggunaan masker pada wajah. Model MobileNetV2 didapatkan dengan melakukan *loading* model dari tensorflow tanpa mengambil *fully-connected layer*. Kemudian, seluruh *layer* yang telah melalui proses *loading* dibekukan agar *weight* pada *layer* MobileNetV2 tidak berubah ketika proses *training model.* Selanjutnya, dibentuk *fully-connected layer* yang baru dengan *output* untuk dua kelas saja yaitu "*with_mask*" dan "*without_mask*". layer tersebut dilakukan *training* menggunakan *training set* sehingga dapat menghasilkan hasil klasifikasi untuk penggunaan masker pada wajah.

E. Training

Proses *training* dijalankan dengan jumlah *epoch* sebanyak 20. Data yang digunakan merupakan *training set* yang telah melalui tahap prapemrosesan. 20% *training set* digunakan untuk *validation set* sehingga jumlah citra pada *training set* sebesar 18.514 citra dan jumlah *validation set* sebesar 4.629 citra. Pada model ini, *optimizer* yang digunakan adalah Adam Optimizer[11], [12].

F. Evaluasi

Penelitian ini melakukan pengujian model untuk mengukur performa yang dihasilkan apabila dilakukan *testing* dengan *testing set*. Performa model diukur menggunakan *confusion matrix*. *Confusion Matrix* merupakan sekumpulan informasi klasifikasi yang aktual dan hasil prediksi yang dikerjakan pada sistem tertentu[13]. *Confusion Matrix* mengandung nilai positif dan negatif seperti yang digambarkan pada Tabel 1 yang dapat dievaluasi untuk mendapatkan nilai akurasi, *precision*, *recall* dan *f1-score*[14].

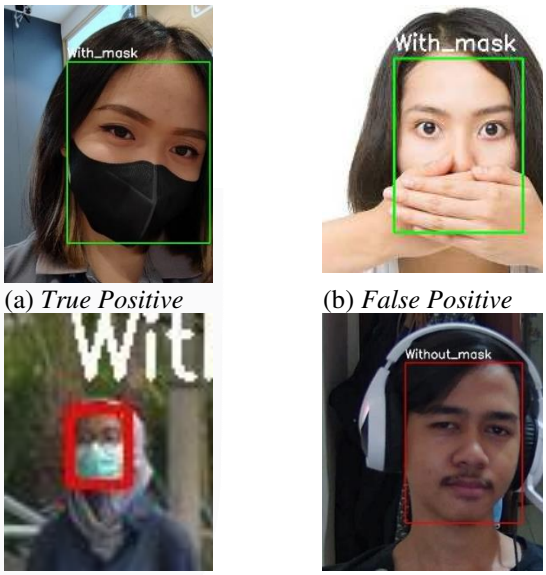
TABEL 1.
Confusion Matrix

		Actual	
		Positive	*Negative*
Predicted	*Positive*	*True Positive (TP)*	*False Positive (FP)*
	Negative	*False Negative (FN)*	*True Negative (TN)*

Berikut penjelasan dari Tabel 1 confusion matrix :

1. *True Positive* (TP) menunjukkan bahwa model MobileNetV2 memberikan hasil prediksi "*with_mask*" untuk citra wajah bermasker.
2. *False Positive* (FP) menunjukkan bahwa model MobileNetV2 memberikan hasil prediksi "*with_mask*" untuk citra wajah tidak bermasker dan citra wajah dengan penggunaan masker yang salah.
3. *False Negative* (FN) menunjukkan bahwa model MobileNetV2 memberikan hasil prediksi "*without_mask*" untuk citra wajah bermasker.
4. *True Negative* (TN) menunjukkan bahwa model MobileNetV2 memberikan hasil prediksi "*without_mask*" untuk citra wajah tidak bermasker atau citra wajah dengan penggunaan masker yang salah

Berikut pada Gambar 5 merupakan contoh hasil deteksi citra berdasarkan *true positive, false positive, false negative,* dan *true negative*.



(a) *True Positive* (b) *False Positive*

(c) *False Negative* (d) *True Negative*

GAMBAR 5.
Contoh citra hasil deteksi berdasarkan dari *confusion matrix*

Confusion matrix dapat menghasilkan hasil evaluasi berupa akurasi, *precision*, *recall*, dan *f1-score* yang dapat digunakan untuk evaluasi kinerja dari model MobileNetV2 sebagai model klasifikasi penggunaan masker. Akurasi ditujukan untuk mengukur keakuratan prediksi model MobileNetV2. *Precision* merupakan rasio dari *true positive* yang dibandingkan dengan keseluruhan hasil positif yang diprediksi oleh model. *Precision* menunjukkan ketetapan model MobileNetV2 dalam mendeteksi citra wajah bermasker dari seluruh hasil prediksi citra dengan kelas "*with_mask*". *Recall* merupakan rasio dari *true positive* yang dibandingkan dengan keseluruhan data aktual positif. *Recall* menunjukkan besaran citra wajah yang diprediksi menggunakan masker dari seluruh data aktual citra wajah bermasker. *F1-score* ditujukan untuk mengevaluasi keakuratan pada hasil prediksi model MobileNetV2 berdasarkan *precision* dan *recall*[14].

Cara perhitungan nilai akurasi, *precision*, *recall* dan *f1-score* terdapat pada rumus (1), (2), (3), dan (4).

$$Akurasi = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1-Score = 2 \cdot \frac{precision*recall}{precision+recall} \quad (4)$$

IV. HASIL DAN PEMBAHASAN

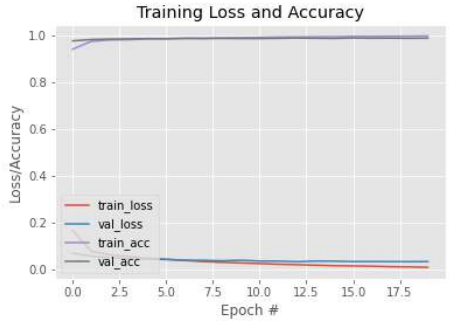
A. Evaluasi

Penelitian ini melakukan tiga skenario pengujian yang berbeda menggunakan data testing yang telah disiapkan.

1. Skenario yang pertama dilakukan *training* menggunakan *testing set* dengan jumlah citra sebesar 23.315 citra dan *testing* model menggunakan 6.998 citra pada *testing set* yang telah disiapkan dan tidak diikutsertakan dalam proses *training* model.
2. Skenario yang kedua dilakukan *training* dan *testing* model dengan tiga kondisi model MobileNetV2 yang berbeda yaitu seluruh *layer* MobileNetV2 dibekukan, setengah dari seluruh *layer* MobileNetV2 dibekukan, dan seluruh *layer* MobileNetV2 tidak dibekukan.
3. Skenario yang ketiga dilakukan untuk menguji kinerja RetinaFace dan MobileNetV2 dalam mendapatkan wajah dan melakukan klasifikasi pada citra wajah dengan penggunaan masker yang tidak sesuai. Model klasifikasi yang dilakukan *testing* adalah model MobileNetV2 yang dilakukan *training* ulang (*fine-tuning model*) dan yang tidak dilakukan *training* ulang (*pretrained model*).

1. Pengujian Skenario Pertama

Pada pengujian skenario pertama dilakukan proses *training* menggunakan *training set* dan pengujian model menggunakan *testing set*. Pengujian ini bertujuan untuk mengukur hasil pelatihan model serta performa yang dihasilkan oleh model MobileNetV2 dalam melakukan klasifikasi penggunaan masker pada citra wajah. Pada Gambar 6 Dapat dilihat bahwa nilai akurasi dan *loss* pada proses *training* sudah stabil pada *epoch* kedua.



GAMBAR 6.
Nilai akurasi dan *loss* pada proses *training* di setiap *epoch*

Berikut ini adalah hasil proses *training* model dan *confusion matrix* dari hasil pengujian skenario satu pada Tabel 2 dan Tabel 3.

TABEL 2.
Hasil proses *training* MobileNetV2

MobileNetV2	
Training Accuracy	0.9973
Validation Accuracy	0.9881
Training Loss	0.0102
Validation Loss	0.0348

TABEL 3.
Confusion Matrix untuk hasil *testing* model

MobileNetV2	
True Positive	3442
False Positive	3458
False Negative	45
True Negative	73

Berdasarkan Tabel 3 dapat dihitung nilai akurasi, *precision*, *recall*, dan *f1-score*. Berikut adalah hasil perhitungan nilai akurasi, *precision*, *recall*, dan *f1-score* pada Tabel 4

TABEL 4.
Hasil perhitungan evaluasi berdasarkan *confusion matrix*

MobileNetV2	
Accuracy	0.9832
Precision	0.9871
Recall	0.9792
F1-Score	0.9831

Berdasarkan hasil pengujian pada Tabel 4 untuk skenario pertama, model MobileNetV2 menghasilkan akurasi sebesar 98,3% dengan nilai *f1-score* sebesar 98,3%.

2. Pengujian Skenario Kedua

Pada pengujian skenario kedua dilakukan pengujian dengan tiga model MobileNetV2 dengan kondisi *layer* yang berbeda. Kondisi pertama yaitu seluruh *layer* MobileNetV2 dibekukan. Kondisi kedua yaitu setengah dari keseluruhan *layer* MobileNetV2 dibekukan. Kondisi terakhir yaitu seluruh *layer* MobileNetV2 tidak dibekukan atau dilakukan *training* ulang. Hal ini bertujuan untuk mengukur kinerja dari ketiga model yang dapat menghasilkan akurasi terbaik. Berikut ini adalah hasil proses *training* dari ketiga model dan *confusion matrix* dari hasil *testing* menggunakan *testing set* Pada Tabel 5 dan Tabel 6.

TABEL 5.
Hasil proses training untuk ketiga model

MobileNetV2			
	100 % *Layer*	50 % *Layer*	0 % *Layer*
Training Accuracy	0.9973	**0,9994**	0,9992

Validation Accuracy	0.9881	**0,9937**	0,9933
Training Loss	0.0102	**0,0026**	0.0028
Validation Loss	0.0348	**0,0468**	0.0477

TABEL 6.
Confusion matrix untuk ketiga model MobileNetV2

MobileNetV2			
	100 % *Layer*	50 % *Layer*	0 % *Layer*
True Positive	3442	3446	**3470**
True Negative	3458	**3493**	3475
False Positive	45	**10**	23
False Negative	73	49	**25**

Berdasarkan Tabel 6 dapat dihitung nilai akurasi, *precision*, *recall*, dan f1-score. Berikut adalah hasil perhitungan nilai akurasi, *precision*, *recall*, dan f1-score pada Tabel 7 untuk ketiga model

TABEL 7.
Hasil evaluasi berdasarkan *confusion matrix*

MobileNetV2			
	100 % *Layer*	50 % *Layer*	0 % *Layer*
Accuracy	0.9832	0,9916	**0,9931**
Precision	0.9871	**0,9971**	0,9934
Recall	0.9792	0,9860	**0,9928**
F1-Score	0.9831	0,9915	**0,9931**

Dari hasil pengujian pada Tabel 7 untuk skenario dua, model MobileNetV2 dengan seluruh *layer* tidak dibekukan atau dilakukan *training* ulang menggunakan *testing set* menghasilkan *false negative* yang sangat rendah sebesar 25 citra namun untuk model dengan setengah dari keseluruhan *layer* MobileNetV2 yang dibekukan memberikan nilai *false positive* terendah sebesar 10 citra. Hasil akurasi dan nilai *f1-score* tertinggi didapatkan oleh model MobileNetV2 dengan seluruh *layer* tidak dibekukan dengan akurasi sebesar 99,3% dan nilai *f1-score* sebesar 99,3%.

3. Pengujian Skenario Ketiga

Pengujian skenario ketiga dilakukan untuk menguji model MobileNetV2 dalam mengklasifikasikan citra wajah bermasker dan tidak bermasker. Pengujian dilakukan menggunakan *testing set* kualitatif yang bersumber di luar *dataset* penelitian. Data tersebut berisi 10 citra yang berisi lebih dari satu orang. Berikut adalah hasil pengujian ini pada Gambar 7.




(a) (b)




(c) (d)

GAMBAR 7.
Hasil sistem deteksi masker oleh RetinaFace dengan MobileNetV2 (*pretrained*)

Pada Gambar 7, RetinaFace berhasil mendeteksi wajah pada citra dan model MobileNetV2 dapat memberikan hasil klasifikasi yang benar pada Gambar 7 bagian (a). Sedangkan pada Gambar 7 pada bagian (b), (c), dan (d) masih terdapat hasil prediksi dari model MobileNetV2 yang salah.

Selanjutnya dilakukan pengujian yang sama menggunakan RetinaFace dan model MobileNetV2 yang telah dilakukan *training* ulang menggunakan *training set.* Model ini digunakan untuk pengujian karena memberikan hasil akurasi terbaik dari hasil pengujian skenario kedua. Berikut adalah hasil pengujian untuk RetinaFace dan MobileNetV2 yang telah dilakukan *training* ulang pada Gambar 8.




(a) (b)




(c) (d)

GAMBAR 8.
Hasil sistem deteksi masker oleh RetinaFace dengan MobileNetV2 (*fine-tuning*)

Pada Gambar 8, RetinaFace berhasil mendeteksi wajah pada citra dan model MobileNetV2 dapat memberikan hasil prediksi yang benar pada Gambar 8 bagian (a) dan (d). Namun, pada Gambar 8 bagian (b) dan (c) masih terdapat hasil prediksi yang salah.

4. Analisis Hasil Pengujian

Berdasarkan hasil skenario pengujian model MobileNetV2 menunjukkan hasil yang baik dalam mengklasifikasikan citra wajah yang menggunakan masker dan tidak menggunakan masker. Hasil akurasi yang dihasilkan sangat tinggi yaitu 98,3% dan *f1-score* sebesar 98,3%. Pada skenario kedua model MobileNetV2 yang memiliki hasil akurasi dan *f1-score* yang tinggi adalah model MobileNetV2 dengan seluruh *layer* MobileNetV2 tidak dibekukan atau dilakukan *training* ulang menggunakan *training set* citra wajah bermakser dan tidak bermasker. Hasil akurasi yang didapatkan sebesar 99,3% dan *f1-score* 99,3%. Terjadi peningkatan pada nilai *f1-score* dari model MobileNetV2 dengan seluruh *layer* yang dibekukan hingga seluruh *layer* tidak dibekukan. Hal ini dikarenakan seluruh *layer* MobileNetV2 dilakukan *training* ulang menggunakan *dataset* citra wajah bermakser dan tidak bermasker yang

sedikit berbeda dengan *dataset* ImageNet sehingga *weight* yang dihasilkan menyesuaikan dengan *dataset* citra wajah bermasker dan tidak bermasker. Berdasarkan tabel 6, dapat dilihat terjadi penurunan nilai pada *false positive* dan *false negative* dari model MobileNetV2 dengan seluruh *layer* yang dibekukan hingga model MobileNetV2 dengan seluruh *layer* dilakukan *training* ulang. Hal ini menunjukkan bahwa dengan melatih *layer* ulang MobileNetV2 dengan *dataset* dapat meningkatkan akurasi.

Pada skenario ketiga, sistem deteksi penggunaan masker menggunakan RetinaFace dan MobileNetV2 dengan seluruh layer dilakukan *training* ulang (*fine-tuning*) menghasilkan hasil yang lebih baik seperti yang digambarkan pada Gambar 8 bagian (a) dan (d). Namun, sistem deteksi penggunaan masker ini masih memberikan hasil prediksi yang salah pada citra orang yang diambil dari jarak yang jauh seperti yang digambarkan pada Gambar 8 bagian (b) dan (c).

V. KESIMPULAN

A. Kesimpulan

Pada penelitian ini telah dibentuk sistem deteksi masker menggunakan RetinaFace sebagai *face detection* dan MobileNetV2 sebagai *face mask classification* yang telah dikustomisasi untuk bagian *fully-conneccted layer* yang hanya dapat menghasilkan *output* untuk dua kelas saja yaitu "*with_mask*" dan "*without_mask*". Hasil akurasi MobileNetV2 mendapatkan nilai sebesar 99,3% dan nilai *f1-score* sebesar 99,3% setelah dilakukan proses *fine-tuning.* Namun model MobileNetV2 masih dapat memberikan hasil prediksi yang salah untuk citra wajah dari citra orang yang diambil dari jarak yang jauh.

REFERENSI

[1] Q.-V. Pham, D. C. Nguyen, T. Huynh-The, W.-J. Hwang, and P. N. Pathirana, "Artificial Intelligence (AI) and Big Data for Coronavirus (COVID-19) Pandemic: A Survey on the State-of-the-Arts," *IEEE Access*, vol. 8, pp. 130820–130839, 2020, doi: 10.1109/ACCESS.2020.3009328.

[2] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustain. Cities Soc.*, vol. 66, p. 102692, Mar. 2021, doi: 10.1016/j.scs.2020.102692.

[3] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-stage Dense Face Localisation in the Wild." arXiv, May 04, 2019. Accessed: Jan. 29, 2023. [Online]. Available: http://arxiv.org/abs/1905.00641

[4] A. Chavda, J. Dsouza, S. Badgujar, and A. Damani, "Multi-Stage CNN Architecture for Face Mask Detection." arXiv, Sep. 17, 2020. doi: 10.48550/arXiv.2009.07627.

[5] A. Das, M. Wasif Ansari, and R. Basak, "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV," in *2020 IEEE 17th India Council International Conference (INDICON)*, Dec. 2020, pp. 1–5. doi: 10.1109/INDICON49873.2020.9342585.

[6] A. M. Lad, A. Mishra, and A. Rajagopalan, "Comparative Analysis of Convolutional Neural Network Architectures for Real Time COVID-19 Facial Mask Detection," *J. Phys. Conf. Ser.*, vol. 1969, no. 1, p. 012037, Jul. 2021, doi: 10.1088/1742-6596/1969/1/012037.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks." arXiv, Mar. 21, 2019. doi: 10.48550/arXiv.1801.04381.

[8] TheSSJ2612, "Real-Time-Medical-Mask-Detection." Jun. 14, 2021. Accessed: Jan. 29, 2023. [Online]. Available: https://github.com/TheSSJ2612/Real-Time-Medical-Mask-Detection

[9] "Face Mask Detection ~12K Images Dataset." https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset (accessed Jan. 29, 2023).

[10] "Face Mask Dataset." https://www.kaggle.com/datasets/shiekhburhan/face-mask-dataset (accessed Feb. 14, 2023).

[11] M. Yaqub *et al.*, "State-of-the-Art CNN Optimizer for Brain Tumor Segmentation in Magnetic Resonance Images," *Brain Sci.*, vol. 10, no. 7, Art. no. 7, Jul. 2020, doi: 10.3390/brainsci10070427.

[12] P. Poudel, S. H. Bae, and B. Jang, "Comparison of Different Deep Learning Optimizers for Modeling Photovoltaic Power," *J. Chosun Nat. Sci.*, vol. 11, no. 4, pp. 204–208, 2018, doi: 10.13160/ricns.2018.11.4.204.

[13] G. I. Webb, "Model Evaluation," in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds. Springer, 2017, pp. 844–845. doi: 10.1007/978-1-4899-7687-1_555.

[14] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. Springer Science & Business Media, 2011.