

Analisis Perbandingan Tingkat Performa Arsitektur MobileNetV2 dan ResNet50 untuk Klasifikasi Penyakit Paru-Paru

Bryan Immanuel¹, Joshua Albertus Samariyanto², Wirya Wonggo³, Yosua Wijaya⁴, Hans Wirjawan⁵

¹²³⁴⁵Program Studi Teknik Informatika, Universitas Surabaya, Surabaya, Jawa Timur

¹s160419015@student.ubaya.ac.id, ²s160419033@student.ubaya.ac.id, ³s160419036@student.ubaya.ac.id,
⁴s160419038@student.ubaya.ac.id, ⁵s160420108@student.ubaya.ac.id

Abstrak

Penyakit COVID-19 merupakan penyakit menular yang menyerang sistem pernapasan manusia. Orang yang terkena penyakit ini akan mengalami beberapa gejala seperti demam, batuk, dan kesulitan bernafas. Penyakit COVID-19 ini memiliki persamaan gejala dengan penyakit *viral pneumonia*. Penelitian ini memanfaatkan ilmu radiografi untuk melakukan diagnosis dalam mengetahui penyakit yang dialami oleh pasien. Perbedaan dari hasil *scan* rontgen paru-paru sangat berperan untuk membedakan penyakit *viral pneumonia* dan penyakit COVID-19. Penelitian ini dibuat untuk memprediksi apakah pasien mengalami penyakit COVID-19, *viral pneumonia* atau berada dalam kondisi normal. Prediksi ini dilakukan dengan metode *transfer learning* dengan mengembangkan arsitektur CNN yang telah ada, yaitu MobileNetV2 dan ResNet50. MobileNetV2 adalah arsitektur umum yang sangat mirip dengan MobileNet sebelumnya, sedangkan ResNet50 adalah arsitektur *deep network* yang berbasis residual dengan 50 layer. Berdasarkan percobaan yang telah dilakukan, diketahui bahwa model terbaik didapatkan menggunakan metode *transfer learning* dengan memanfaatkan arsitektur ResNet50 yang telah dimodifikasi. Pembuatan model menggunakan arsitektur ResNet50 memberikan akurasi sebesar 97,32% dengan nilai *precision* sebesar 94%, *recall* sebesar 94%, dan *F1-score* sebesar 94%. Pembuatan model menggunakan arsitektur MobileNetV2 mampu memberikan hasil akurasi yang lebih baik, yaitu sebesar 98,54%, namun dengan hasil evaluasi yang lebih rendah dengan nilai *precision* sebesar 91%, *recall* sebesar 91%, dan *F1-score* sebesar 91%.

Kata Kunci: covid-19, mobilenetv2, resnet50, penyakit, transfer learning.

Comparative Performance Analysis of MobileNetV2 and ResNet50 Architecture for Lung Disease Classification

Abstract

COVID-19 disease is a disease that infect human respiratory system. Person with COVID-19 will undergo some symptoms such as fever, cough, and difficulty to breath. COVID-19 has similar symptoms with Viral Pneumonia. This research uses radiographic knowledge to do the diagnosis to know the patient's disease. The differences in lung x-ray result has a big role in distinguish between Viral Pneumonia and COVID-19. This research was done to predict whether the patient has infected by COVID-19, Viral Pneumonia or is in normal condition. This prediction was done by using transfer learning by improving a pre-trained CNN model which is MobileNetV2 and ResNet50. MobileNetV2 is a general CNN architectural that is very similar with the older version of MobileNet, whereas the ResNet50 is a residual-based deep network architecture with 50 layers. Based on the research that have been done before, it is known that the best CNN model was achieved by using transfer learning with ResNet50 architecture that have been modified. CNN model that is improved by using pre-trained ResNet50 architecture gives 97.32% accuracy, 94% precision, 94% recall and 94% F1-score. On the other hand, CNN model that is improved by using pre-trained MobileNetV2 gives 98.54% accuracy, but with lower evaluation result which is 91% precision, 91% recall, and 91% F1-score.

Keywords: covid-19, mobilenetv2, resnet50, disease, transfer learning.

I. PENDAHULUAN

COVID-19 adalah penyakit menular yang disebabkan oleh SARS-CoV-2, salah satu jenis coronavirus. SARS-CoV-2 merupakan salah satu virus yang menyerang sistem pernapasan manusia. Penderita COVID-19 memiliki gejala demam, batuk kering, dan kesulitan bernafas. Sejak Januari 2020 sampai Juli 2022 telah ditemukan sebanyak 6 juta kasus COVID-19 dan sebanyak 156 ribu orang telah meninggal karena COVID-19 di Indonesia [1]. Saat ini sudah banyak penelitian yang dilakukan terhadap penyakit COVID-19. Penyakit COVID-19 memiliki kemiripan gejala dengan penyakit Flu dan Pneumonia. Pneumonia adalah penyakit peradangan paru-paru yang menyebabkan kantung udara pada paru-paru terisi oleh cairan nanah. Pneumonia dapat disebabkan oleh COVID-19 yang menyerang saluran pernapasan dan menyumbat saluran pernapasan pada paru-paru sehingga menyebabkan peradangan paru-paru [2]. Penyakit ini memiliki tingkat penularan yang sangat tinggi, dan sangat berbahaya terutama terhadap orang yang memiliki kondisi tubuh yang lemah. Oleh karena itu, diperlukan adanya metode yang tepat dan cepat dalam mendeteksi penyakit COVID-19 maupun pneumonia pada seseorang agar dapat segera mendapatkan pertolongan yang tepat. Metode tersebut adalah dengan membuat model klasifikasi hasil rontgen paru-paru dengan menggunakan model *deep learning*.

Mundher Mohammed Taresh *et al.* [3] mengevaluasi efektivitas dari model Convolutional Neural Network (CNN) yang sudah dilatih sebelumnya untuk mendeteksi penyakit COVID-19 dari gambar rontgen paru-paru. Pembuatan model klasifikasi dilakukan dengan teknik *transfer learning* menggunakan model VGG16 dan MobileNet. Kedua *pre-trained model* ini berhasil melakukan klasifikasi penyakit COVID-19 dengan akurasi paling tinggi didapatkan dari VGG16 sebesar 98,28%. Kinerja VGG16 dinilai paling baik dibandingkan dengan semua *pre-trained model* dalam hal mendeteksi penyakit COVID-19 yang memiliki *F1-score* sebesar 97,59% serta nilai presisi sebesar 96,43%. Menurut peneliti, jumlah *dataset* rontgen COVID-19 yang lebih banyak dibutuhkan untuk mendapatkan akurasi dan hasil klasifikasi yang lebih baik ketika menggunakan *pre-trained model transfer learning*.

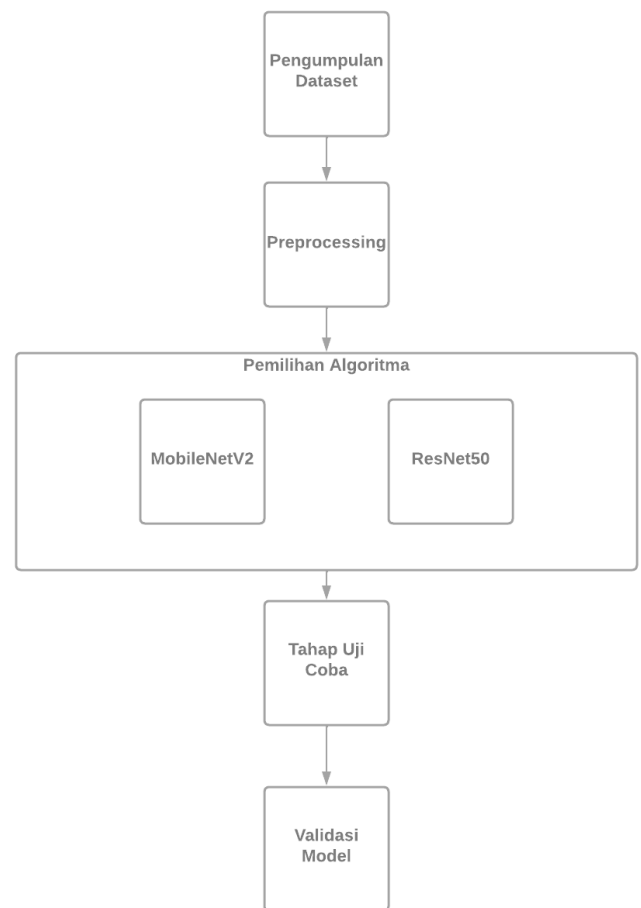
Kedong Rao *et al.* [4] melakukan eksperimen klasifikasi dari hasil tes rontgen paru-paru COVID-19. Pembuatan model klasifikasi dilakukan dengan menggunakan teknik *transfer learning* menggunakan *pre-trained model* VGG16, ResNet50, InceptionV3, Xception, SVRNet, dan SVDNet. Dari hasil penelitian yang telah dilakukan, didapatkan hasil yang paling bagus adalah SVRNet dan SVDNet dengan masing-masing akurasi 99,13% dan 99,37%. Menurut hasil peneliti, model SVRNet dapat memberikan pendeteksian hasil yang lebih cepat, tetapi jika lebih mengutamakan akurasi, maka akan lebih tinggi jika menggunakan SVDNet.

Berdasarkan penelitian-penelitian yang telah disebutkan sebelumnya maka akan dibuat model *deep learning* menggunakan metode Convolutional Neural Network (CNN) dengan memanfaatkan teknik *transfer learning*. Pada penelitian ini akan dicoba dua *pre-trained model*, yaitu MobileNetV2 dan Resnet50 yang mana kedua model ini akan dibandingkan performanya dalam melakukan klasifikasi hasil rontgen paru-paru. Kedua model yang didapatkan akan

digunakan untuk memprediksi kondisi paru-paru berdasarkan gambar rontgen paru-paru, apakah kondisi paru-paru tergolong normal, terkena viral pneumonia, atau terjangkit virus COVID-19. Dengan demikian, proses untuk mendeteksi penyakit paru-paru, seperti pneumonia dan juga COVID-19 dapat menjadi lebih efisien.

II. METODOLOGI PENELITIAN

Penelitian ini terdiri dari 5 tahap seperti yang ditunjukkan pada Gambar 1. Metode penelitian yang dilakukan dimulai dari tahap pengumpulan dataset, tahap *preprocess data*, tahap pemilihan model algoritma, tahap uji coba, dan tahap validasi model. Bagian ini akan menjelaskan proses-proses tersebut dengan lebih detail.



Gambar 1. Tahapan Proses Penelitian

A. Pengumpulan Dataset

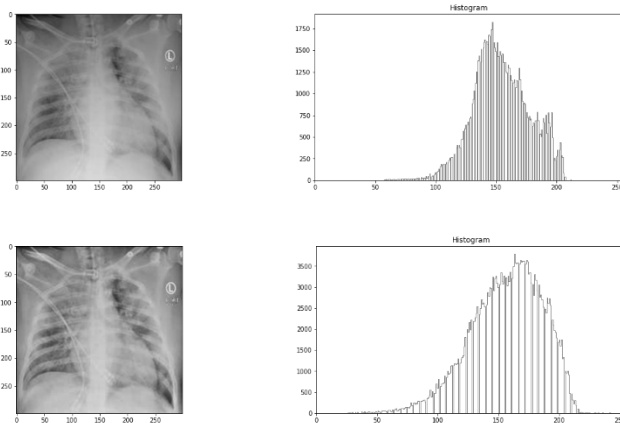
Proses pertama adalah pengumpulan *dataset*. *Dataset* yang digunakan pada penelitian ini adalah COVID-19 Radiography Database yang diambil dari *website kaggle.com* [5]. *Dataset* ini mengandung 21.173 data berjenis gambar hasil rontgen paru-paru dari penderita penyakit COVID-19,

rontgen paru-paru normal, rontgen paru-paru penderita viral pneumonia, rontgen paru-paru yang terkena penyakit *lung opacity*. Dataset ini memiliki 4 *class label*, yaitu COVID, Lung_Opacity, Normal, dan Viral Pneumonia. Masing-masing *class label* terdiri dari 3.616 jumlah gambar rontgen paru-paru yang terinfeksi COVID-19, 6.012 jumlah gambar rontgen paru-paru yang terkena penyakit *lung opacity*, 10.200 jumlah gambar rontgen paru-paru normal, dan 1.345 jumlah gambar rontgen paru-paru yang terkena penyakit viral pneumonia. Penelitian ini hanya menggunakan tiga *class label* saja, yaitu COVID, Normal, dan Viral Pneumonia dengan jumlah gambar masing-masing *class label* sama seperti yang telah disebutkan sebelumnya.

B. Preprocess data

Metode *preprocessing data* yang digunakan adalah *Contrast Limited Adaptive Histogram Equalization (CLAHE)*. Kemudian, hasil gambar yang telah diproses akan dilanjutkan dengan proses Augmentasi gambar menggunakan bantuan *library* Image Data Generator yang disediakan oleh *library* Tensorflow guna untuk meningkatkan akurasi dari model yang akan digunakan.

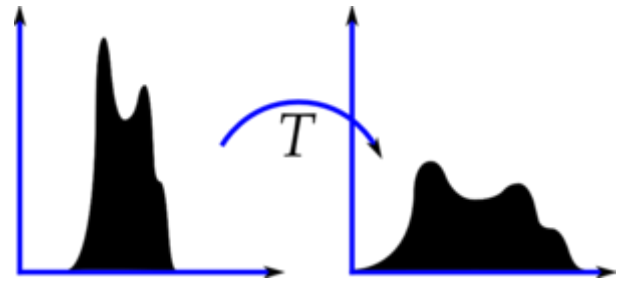
Contrast Limited Adaptive Histogram Equalization (CLAHE) merupakan metode untuk meningkatkan kontras gambar yang memiliki pixel yang mengumpul di suatu region tertentu pada histogram seperti pada gambar 2.



Gambar 2. Contoh Implementasi Metode *CLAHE* terhadap Rontgen Paru-Paru

Cara kerja metode ini adalah mendistribusikan jumlah *pixel* tertentu dari suatu gambar agar tersebar secara merata guna meningkatkan kontras dari suatu gambar. Metode ini merupakan pengembangan dari penerapan *histogram equalization*. Bedanya, *histogram equalization* hanya sebatas meratakan *pixel* secara menyeluruh pada gambar, sedangkan *CLAHE* meratakan *pixel* dengan cara membagi gambar menjadi beberapa bagian. Setelah itu, setiap bagian *pixel* akan dikenai metode *histogram equalization*. Gambar 3 merupakan gambaran aplikasi metode *histogram equalization* terhadap sebuah gambar dengan merentangkan histogram *pixel* dari suatu gambar ke arah dua sisi yang berbeda [6]. Pada penelitian ini, setiap gambar pada *dataset* dibagi menjadi

bagian-bagian kecil sebesar 8×8 *pixel* yang merupakan *default parameter* dari *library* cv2. Selain itu, pembatasan pemotongan gambar juga diatur sebesar 0,2 jika pada saat proses CLAHE bertemu dengan *noise*.



Gambar 3. Implementasi *Histogram Equalization*

Augmentasi gambar adalah sebuah metode yang digunakan untuk mengembangkan jumlah dataset buatan melalui *library* Image Data Generator yang disediakan oleh *library* Tensorflow. Image Data Generator berguna untuk meningkatkan variasi *dataset* dengan cara mentransformasikan gambar seperti merotasi, menggeser, memutar, dan lain-lain [7]. Pada penelitian ini dataset akan dibagi menjadi 80% *training dataset* dan 20% *validation dataset*. Augmentasi gambar hanya digunakan pada *dataset training* agar model yang nantinya digunakan dapat mempelajari data *training* saja serta mampu memberikan hasil akurasi yang sebenarnya terhadap *validation dataset*. Metode Image Data Generator yang digunakan pada penelitian ini adalah melakukan perbesaran gambar sebesar 0,2, rotasi gambar sebesar 10° , serta menggantikan gambar daerah yang kosong dengan piksel terdekatnya.

Tabel 1. Augmentasi Gambar yang Digunakan

Parameter	Value
Zoom Range	0,2
Rotation Range	10°
Fill Mode	nearest
Preprocessing Function	clahe
Validation Split	0,2

C. Pemilihan Model Algoritma

Pada penelitian ini, teknik yang akan digunakan adalah *transfer learning* dengan menggunakan *pre-trained model* MobileNetV2 dan ResNet50 yang disediakan oleh Keras Tensorflow. Implementasi MobileNetV2 dan ResNet50 yang digunakan sama-sama membutuhkan parameter *input shape*

dengan ukuran gambar 160 x 160 *pixel* yang memiliki 3 *channel*. Selain itu, parameter 'include_top' diberi nilai *false* dikarenakan penelitian ini tidak menggunakan *layer output* dari MobileNetV2 tetapi menggunakan *layer output* buatan sendiri yang disesuaikan dengan jumlah kelas pada dataset. *Weights node* pada tiap layer menggunakan *weight* dari pelatihan data *imagenet* dikarenakan secara *defacto*, *imagenet* merupakan standar dari klasifikasi gambar. *Imagenet* bekerja dengan cara membandingkan gambar yang satu dengan gambar yang lain sehingga model memiliki kemampuan untuk menilai kecocokan model untuk klasifikasi gambar. Untuk *optimizer* yang digunakan pada kedua model ini adalah *adam optimizer* dengan nilai *learning rate* sebesar 0,001.

Tabel 2. Parameter Model MobileNetV2 dan ResNet50

Parameter	Value
Input Shape	160,160,3
Include Top	False
Weights	Imagenet
Optimizer	Adam

Tabel 3. Layer Model MobileNetV2 yang Dimodifikasi

Layer	Value	Deskripsi
Input	(160, 160, 3)	-
Input (MobileNetV2)	(160, 160, 3)	-
Pre Process Input	Input	-
GlobalAveragePooling2D	-	-
Dropout	0,4	-
Dense	(3)	Softmax

Tabel 3 di atas menunjukkan penggunaan *layer* pada model MobileNetV2 yang dimodifikasi. Pengembangan MobileNetV2 dilakukan dengan menambahkan *input-layer* dengan ukuran yang sama dengan ukuran pembacaan gambar dari *dataset* yaitu 160x160x3 (3 *channel/RGB*) dan beberapa layer karena *top-layer* dari model MobileNetV2 tidak dipakai. Setelah *input-layer*, terdapat Pre Process Input yaitu *layer* yang bertugas untuk melakukan *rescale* pada *pixel* tiap gambar yang ada agar rentangan nilainya berubah yang mulanya adalah 0 hingga 255 menjadi -1 hingga 1 karena arsitektur MobileNetV2 hanya menerima *input* gambar dengan rentangan *pixel* dengan nilai tersebut. GlobalAveragePooling2D bertugas untuk mengganti *fully-connected layers* menjadi *average pooling*. Dropout bertugas untuk me-nonaktifkan beberapa persen *node layer* yang dirasa tidak memberikan pengaruh yang signifikan pada setiap

epoch. Terakhir, *layer Dense* dengan *activation function* yang digunakan adalah *softmax* dengan jumlah *node* sejumlah 3 yang berfungsi sebagai *output-layer*. Pada implementasinya, *layer* yang digunakan pada model MobileNetV2 hanya layer 1 hingga layer 60.

Tabel 4. Layer Model ResNet50 yang Dimodifikasi

Layer	Value	Deskripsi
Input	(160, 160, 3)	-
Input (MobileNetV2)	(160, 160, 3)	-
GlobalAveragePooling2D	-	-
Dropout	0,4	-
Dense	(3)	Softmax

Tabel 4 di atas menunjukkan penggunaan *layer* pada model ResNet50 yang dimodifikasi. Pengembangan ResNet50 dilakukan dengan menambahkan *input-layer* dengan ukuran yang sama dengan ukuran pembacaan gambar dari *dataset* yaitu 160x160x3 (3 *channel/RGB*) dan beberapa layer karena *top-layer* dari model ResNet50 tidak dipakai. Tabel 4 menunjukkan layer apa saja yang digunakan dari model ResNet50 yang dimodifikasi. *Top-layer* tersebut terdiri dari GlobalAveragePooling2D yang bertugas untuk mengganti *fully-connected layers* menjadi *average pooling*. Dropout bertugas untuk me-nonaktifkan beberapa persen *node layer* yang dirasa tidak memberikan pengaruh yang signifikan pada setiap *epoch*. Terakhir, *layer Dense* dengan *activation function* yang digunakan adalah *softmax* dengan jumlah *node* sejumlah 3 yang berfungsi sebagai *output-layer*. Pada implementasinya, *layer* yang digunakan pada model ResNet50 hanya layer 1 hingga layer 80.

D. Tahapan Uji Coba

Tahapan uji coba dilakukan dengan melakukan proses *training* terhadap *training dataset* COVID-19 Radiography menggunakan *pre-trained model* MobileNetV2 dan ResNet50 yang sudah dimodifikasi sebelumnya. Kedua model yang sudah didapatkan akan digunakan untuk melakukan proses *testing* terhadap *validation dataset* dengan melihat beberapa parameter, seperti *accuracy*, *validation loss*, dan juga *validation accuracy*.

MobileNetV2 adalah arsitektur umum yang sangat mirip dengan MobileNet sebelumnya, hanya saja pada MobileNetV2 menggunakan *residual block* terbalik dengan fitur *bottleneck*. Hal ini menyebabkan jumlah parameter jauh lebih rendah daripada MobileNet yang sebelumnya. Selain itu, MobileNetV2 juga mendukung jumlah input yang lebih besar dari 32 x 32, sehingga gambar yang lebih besar dapat meningkatkan performa [8].

ResNet adalah jenis *deep network* berbasis pembelajaran residual. Pembelajaran semacam ini dapat memfasilitasi pelatihan jaringan dengan mempertimbangkan *input layer* sebagai referensi (He, 2016). ResNet-50 adalah salah satu varian ResNet yang memiliki 50 layer. ResNet-50 melewati 3 layer dan terdapat 1x1 convolution layer [9].

Proses *training* pada model MobileNetV2 dan ResNet50 sama-sama menggunakan iterasi (*epoch*) sebanyak 20 kali serta menggunakan bantuan metode *early stopping* guna untuk menghentikan proses training jika *validation loss* tidak berkurang hingga batas *patience* yang telah ditentukan. Batas *patience* yang digunakan pada penelitian ini adalah sebanyak 8 *epoch*. Pada penelitian ini, *resource* yang digunakan untuk melakukan proses *training* dan *testing* model adalah Graphics Processing Unit (GPU) yang disediakan oleh Google Colab

E. Validasi Model

Berdasarkan hasil yang telah didapatkan dari model sebelumnya, akan dilakukan proses validasi model menggunakan penghitungan dari *confusion matrix* yang mencakup nilai *precision*, *recall*, dan *F1-score*. Selain itu, pada tahapan ini juga dilakukan perbandingan dari lama waktu *training model* serta performa *accuracy* dari setiap algoritma model. Hal ini dilakukan untuk mengetahui algoritma mana yang paling cocok dan efisien pada dataset COVID-19 Radiography Database. Rumus untuk menghitung *accuracy* model dapat dilihat pada persamaan (1), *precision* pada persamaan (2), *recall* pada persamaan (3), dan *F1-score* pada persamaan (4).

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

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

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

$$F1 \text{ Score} = 2 * \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

III. HASIL DAN PEMBAHASAN

Percobaan dilakukan terhadap *dataset* COVID-19 Radiography Database menggunakan dua arsitektur model CNN (*Convolutional Neural Network*), yaitu MobileNetV2 dan ResNet50 yang menggunakan konsep *transfer learning* yang telah dimodifikasi sebelumnya, yaitu menggunakan *pre-trained model* atau model yang telah di latih menggunakan *dataset* tertentu. Pada penelitian ini *dataset* yang digunakan untuk pelatihan model sebelumnya adalah ImageNet. Berdasarkan percobaan yang telah dilakukan, didapatkan beberapa hasil berupa akurasi model terhadap *train data*,

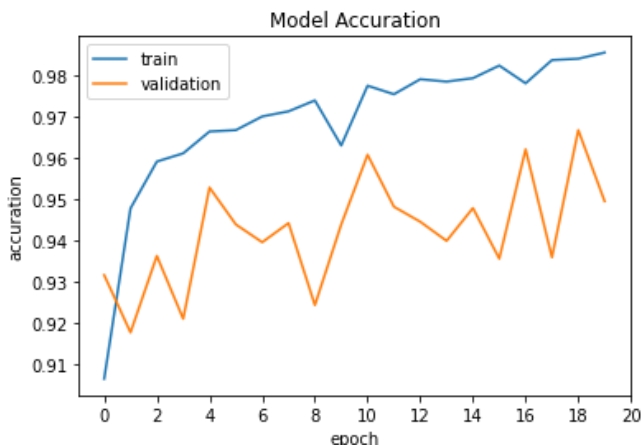
akurasi terhadap *test data*, nilai *F1-score*, *precision*, dan *recall*.

Pengembangan model menggunakan arsitektur *pre-trained model* MobileNetV2 memberikan beberapa hasil untuk setiap jumlah *epoch* yang dijalankan. Hasil *training* untuk setiap *epoch* pada model ini dapat dilihat pada Tabel 5. Tabel 5 menunjukkan *training accuracy* dan *validation accuracy* dari model yang dibuat dengan mengembangkan arsitektur MobileNetV2 dengan proses *training* selama 20 *epoch*. Dengan bantuan pengamatan *validation loss* dari *early stopping* yang sudah diterapkan pada proses *training*, diketahui bahwa *validation loss* pada model ini tidak mengalami kenaikan terus menerus selama 8 *epoch*, sehingga proses *training* tetap dapat berjalan sampai *epoch* ke dua puluh. Proses *training* model ini memerlukan waktu selama 35 menit 33 detik dengan menggunakan GPU milik Google Colab. Model dengan arsitektur MobileNetV2 hasil *training* ini mampu menghasilkan *train accuracy* sebesar 98,54% dan *validation accuracy* sebesar 94,95%. Grafik yang menunjukkan perkembangan *training accuracy* model dengan arsitektur MobileNetV2 dari *epoch* 1 hingga 20 dapat dilihat pada gambar 4.

Tabel 5. Hasil Training Model MobileNetV2

Epoch	Train Accuracy (%)	Validation Accuracy (%)
1	90,66	93,17
2	94,78	91,78
3	95,91	93,63
4	96,11	92,11
5	96,63	95,28
6	96,67	94,39
7	97,00	93,96
8	97,12	94,42
9	97,39	92,44
10	96,30	94,39
11	97,74	96,07
12	97,53	94,82
13	97,90	94,46
14	97,84	93,99
15	97,92	94,79
16	98,23	93,56

Epoch	Train Accuracy (%)	Validation Accuracy (%)
17	97,80	96,20
18	98,36	93,60
19	98,39	96,67
20	98,54	94,95



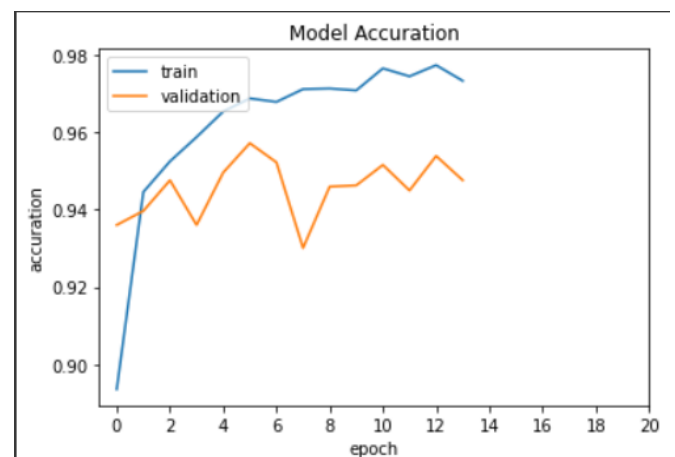
Gambar 4. Perkembangan Akurasi Setiap Epoch Pelatihan Model MobileNetV2

Selanjutnya, pengembangan model menggunakan arsitektur ResNet50 dilakukan dengan melakukan proses *training* terhadap *dataset* yang sama, yaitu *dataset* COVID-19 Radiography Database. Hasil *training* untuk setiap *epoch* dengan menggunakan ResNet50 dapat dilihat pada Tabel 6.

Tabel 6. Hasil *Training* Model ResNet50

Epoch	Train Accuracy (%)	Validation Accuracy (%)
1	89,36	93,60
2	94,45	93,96
3	95,24	94,75
4	95,87	93,60
5	96,53	94,95
6	96,87	95,71
7	96,77	95,21
8	97,10	93,00
9	97,12	94,59
10	97,07	94,62
11	97,64	95,15
12	97,43	94,49
13	97,72	95,38
14	97,32	94,75

Berdasarkan hasil *training* tersebut, dapat diketahui bahwa model yang dibuat dengan mengembangkan arsitektur ResNet50 mampu memberikan akurasi yang cukup baik setelah proses *training* selama 14 *epochs*. Iterasi *epochs* berhenti pada *epoch* ke-14 dari 20 karena penerapan metode *early stopping*. Proses *training* berhenti pada *epoch* ke-14 karena selama 8 *epoch*, *validation loss* terendah ada di *epoch* ke-6 (dengan parameter *patience* sebesar delapan). Proses *training* model ini memerlukan waktu selama 27 menit 12 detik dengan menggunakan GPU milik Google Colab. Model dengan arsitektur ResNet50 hasil *training* ini mampu menghasilkan hasil *train accuracy* sebesar 97,32% dan *validation accuracy* sebesar 94,75%. Grafik yang menunjukkan perkembangan *training accuracy* model dengan arsitektur ResNet50 dari *epoch* 1 hingga 14 dapat dilihat pada gambar 5.



Gambar 5. Perkembangan Akurasi Setiap Epoch Pelatihan Model ResNet50

Selain memperhatikan nilai akurasi yang dihasilkan, ada beberapa nilai juga yang diukur dalam mengevaluasi model yang didapatkan, seperti nilai *precision*, *recall*, dan *F1-score*. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *Recall* adalah tingkat keberhasilan dalam menemukan kembali sebuah informasi [10]. *F1-score* adalah perbandingan rata-rata dari *precision* dan *recall* yang dibobotkan [11]. Evaluasi dilakukan terhadap dua model dari arsitektur MobileNetV2 serta ResNet50 dengan melakukan penghitungan nilai *precision*, *recall*, *F1-score* terhadap 3030 gambar rontgen paru-paru yang berasal dari *validation dataset*. Hasil evaluasi kedua model dapat dilihat pada tabel 7.

Tabel 7. Hasil Evaluasi Kedua Model

Epoch	Precision (%)	Recall (%)	F1-score (%)
MobileNetV2	91	91	91
ResNet50	94	94	94

Berdasarkan hasil dari pengembangan model dengan menggunakan arsitektur *pre-trained model* MobileNetV2 dan ResNet50 yang dapat dilihat pada tabel 7. Lama waktu *training* MobileNetV2 selama 35 menit 33 detik, sedangkan ResNet50 selama 27 menit 12 detik. Dengan membandingkan antara lama waktu *training* dan hasil evaluasi antara MobileNetV2 dengan ResNet50, dapat dilihat bahwa ResNet50 lebih unggul dalam kedua hal tersebut dengan nilai *precision* sebesar 94%, *recall* sebesar 94%, dan *F1-score* sebesar 94%. Proses *training* model juga hanya dilakukan selama 27 menit 12 detik serta hanya memerlukan 14 *epochs* untuk mencapai hasil tersebut.

IV. KESIMPULAN

Proses penelitian dimulai dengan pengambilan *dataset* COVID-19 Radiography Database yang diambil dari *website* kaggle.com. *Dataset* ini mengandung 21.173 data yang terdiri dari 4 *class* label, yaitu COVID, Lung_Opacity, Normal, dan Viral Pneumonia, tetapi Lung_Opacity tidak digunakan. Selanjutnya, dilakukan *preprocessing data* dengan menggunakan metode *Contrast Limited Adaptive Histogram Equalization (CLAHE)* untuk memperjelas gambar dan juga menggunakan *ImageDataGenerator* untuk augmentasi gambar. Proses selanjutnya adalah mengembangkan model menggunakan bahasa pemrograman Python pada dua arsitektur CNN *transfer learning*, yaitu MobileNetV2 dan ResNet50. Kemudian, akan dilakukan *training* terhadap model yang sudah dibuat, diikuti dengan prediksi terhadap 3030 gambar rontgen paru-paru yang berasal dari *validation dataset* untuk melakukan evaluasi terhadap kedua model yang telah dibuat.

Didapati bahwa arsitektur yang dikembangkan dengan menggunakan ResNet50 memberikan hasil yang lebih bagus berdasarkan evaluasi model yang dilakukan dengan nilai *precision* sebesar 94%, *recall* sebesar 94%, dan *F1-score* sebesar 94% tetapi *train accuracy* yang lebih kecil dibandingkan milik MobileNetV2, yaitu sebesar 97,32%. Untuk MobileNetV2 memiliki hasil *train accuracy* yang lebih baik, yaitu sebesar 98,54% tetapi memiliki hasil evaluasi yang lebih rendah dengan *precision* sebesar 91%, *recall* sebesar 91%, dan *F1-score* sebesar 91%.

Berdasarkan hasil temuan penelitian ini, maka saran untuk penelitian selanjutnya adalah dengan melakukan *tuning* parameter pada model dengan lebih baik. Selain itu, disarankan juga mencoba menggunakan *preprocessing data* yang lainnya, seperti melakukan segmentasi gambar untuk memisahkan objek dengan *background*-nya agar gambar yang digunakan menjadi lebih bagus untuk membantu melakukan diagnosis penyakit paru-paru.

REFERENSI

- [1] Indonesia: WHO Coronavirus Disease (COVID-19) Dashboard With Vaccination Data | WHO Coronavirus (COVID-19) Dashboard With Vaccination Data. (n.d.). Retrieved June 24, 2022, from <https://covid19.who.int/region/searo/country/id>
- [2] Hardi, R. A. (2021, November 23), "Pneumonia dengan Covid-19, Apa Hubungannya?", <https://www.herminahospitals.com/id/articles/pneumonia-dengan-covid-19-apa-hubungannya>
- [3] Taresh, M. M., Zhu, N., Ali, T. A. A., Hameed, A. S., & Mutar, M. L. (2021). Transfer learning to detect covid-19 automatically from x-ray images using convolutional neural networks. *International Journal of Biomedical Imaging*, 2021.
- [4] Rao, K., Xie, K., Hu, Z., Guo, X., Wen, C., & He, J. (2021). COVID-19 detection method based on SVRNet and SVDNet in lung x-rays. *Journal of Medical Imaging*, 8(S1). doi:10.1117/1.JMI.8.S1.017504
- [5] Rahman, T., Chowdhury, M., & Khandakar, A. (2022, Maret 19), "COVID-19 Radiography Database", <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>
- [6] Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- [7] Bhandari, A. (2020, Agustus 11), "Image Augmentation on the Fly using Keras ImageDataGenerator!", <https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/>
- [8] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, ... Xiaoqiang Zheng. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, <https://www.tensorflow.org/>
- [9] Hendra, S. B. H., Azizah, A. S. N., & Akbar, F. (2020). Perbandingan Kinerja Arsitektur Inception-V4 dan ResNet-50 Dalam Mengklasifikasikan Citra Paru-Paru Terinfeksi Covid-19. *Digilib Unhas*.
- [10] DATAQ (2013, Juni 16), "Perbedaan: precision, recall & accuracy", <https://dataq.wordpress.com/2013/06/16/perbedaan-precision-recall-accuracy/>
- [11] Arthana, R. (2019, April 05), "Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning.", <https://rey1024.medium.com/mengenal-accuracy-precision-recall-dan-specificity-seerta-yang-diprioritaskan-b79ff4d77de8>