

# Aplikasi Pendukung Diagnosis COVID-19 Yang Menganalisis Hasil X-Ray Paru-Paru Dengan Model *EfficientNet*

Ananta Kusuma Pangkasidhi, Henry Novianus Palit, Alvin Nathaniel Tjondrowiguno  
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra  
Jl. Siwalankerto 121 – 131 Surabaya 60236  
Telp. (031) – 2983455, Fax. (031) – 8417658  
E-Mail: ananta96@gmail.com, hnpalit@petra.ac.id, alvin.nathaniel@petra.ac.id

## ABSTRAK

Pada bulan Desember tahun 2019, telah terdeteksi penyakit berjenis coronavirus yang baru yang memiliki gejala pneumonia di pasar seafood di kota Wuhan, provinsi Hubei, China. Virus ini lalu menyebar keseluruh dunia, yang pada bulan Maret 2020, WHO mengumumkan status dari epidemik menjadi pandemik. WHO akhirnya memberi nama virus pneumonia ini sebagai COVID-19. COVID-19 sudah menginfeksi lebih dari 105 juta orang diseluruh dunia dengan kematian yang sudah mencapai lebih dari 2,3 juta diseluruh dunia. Di Indonesia sendiri sudah terjadi lebih dari 1 juta kasus COVID-19 dan lebih dari 30 ribu kematian pada Februari 2021 .

Dari jumlah kasus yang memprihatinkan ini, penanganan pasien harus tanggap dilakukan. Salah satu pendukung diagnosis COVID-19 adalah *Chest X-ray*. *Chest X-ray* menjadi tahap wajib yang dilakukan pasien sebagai penegasan dan penentuan *treatment* yang sesuai untuk mengobati dengan tepat.

Dalam penelitian ini menggunakan arsitektur *Deep Learning EfficientNet* untuk melakukan klasifikasi antara orang yang terkena COVID-19, pneumonia, dan normal dari *Chest X-ray*. Hasil pengujian diukur dengan Akurasi, *F1-Score*, *Recall*, *Precision*, dan *Specificity*. Dengan adanya penelitian diharapkan dapat melakukan deteksi secepat mungkin sehingga mengurangi penyebaran COVID-19 serta lebih hemat biaya karena *Chest X-ray* lebih murah, lebih cepat, dan dengan radiasi yang lebih sedikit dari *CT-Scan*. Hasilnya adalah akurasi pada penelitian ini mencapai 96 persen, serta *F1-Score*, *Recall*, *Precision*, *Specificity* diatas 95 persen.

**Kata Kunci:** COVID-19, CNN, *EfficientNet*, *Chest X-Ray*, Diagnosis

## ABSTRACT

*In December 2019, a new type of corona virus was detected that had symptoms of pneumonia in the seafood market in Wuhan City, Hubei Province, China. The virus then spread throughout the world, which in March 2020, WHO announced the status of the epidemic as a pandemic. WHO finally named this virus as COVID-19. COVID-19 has infected more than 105 million people worldwide, and deaths that have reached more than 2.3 million worldwide. In Indonesia alone there have been more than 1 million cases of COVID-19 and more than 30 thousand deaths in February 2021 .*

*Based on number of cases, patient must be handled responsively. One of the supporting diagnosis for COVID-19 is Chest X-Ray. Chest X-Ray becomes one of the mandatory steps for patients to*

*confirm and determine the treatment(s) to medicate the patients appropriately.*

*In this study using the Deep Learning EfficientNet architecture to classify people affected by COVID-19, pneumonia, and normal from Chest X-Ray. The test results are measured by Accuracy, F1-Score, recall, precision, and specificity. With this research it is expected to be able to detect as quickly as possible so that it reduces the spread of COVID-19 and is more cost-effective because Chest X-Ray is cheaper, faster, and less radiation than CT-Scan. The result is that the accuracy in this study reaches 96 percent, and the F1-Score, Recall, Precision, Specificity is above 95 percent.*

**Keywords:** COVID-19, CNN, *EfficientNet*, *Chest X-Ray*, Diagnosis

## 1. PENDAHULUAN

Pada bulan Desember tahun 2019, telah terdeteksi penyakit saluran pernafasan atau penyakit yang disebabkan oleh coronavirus yang baru yang memiliki gejala seperti pneumonia di pasar seafood di kota Wuhan, China. Virus ini menyerang sistem pernafasan yang menyebabkan kerusakan pada paru-paru dan paling fatal adalah kematian. Pada bulan Maret 2020, WHO (*World Health Organization*) telah resmi mengumumkan kondisi dari epidemi menjadi pandemik dan memberi nama virus pneumonia ini sebagai COVID-19. Kemudian, di bulan Februari 2021 virus ini telah menginfeksi lebih dari 105 juta orang diseluruh dunia dengan jumlah kematian yang hampir mencapai 2,3 juta diseluruh dunia. Di Indonesia sendiri jumlah kasus COVID-19 telah mencapai lebih dari 1 juta kasus dan lebih dari 30 ribu diantaranya berujung kematian. Jumlah kasus COVID-19 yang terus naik diseluruh dunia menyebabkan sistem kesehatan di rumah sakit memburuk, salah satunya adalah menurunnya performa petugas kesehatan. Hal ini dikarenakan peningkatan jumlah pasien COVID-19 yang drastis sehingga rumah sakit kekurangan persediaan medis, mulai dari peralatan hingga stok obat-obatan. Penelitian mengenai COVID-19 ini sendiri terus berlanjut demi mendapatkan cara penanganan kasus dengan tanggap dan cepat. Hingga saat ini, semua orang yang mengalami gejala seperti demam, batuk, dan tidak bisa mencium bau, diharapkan untuk melakukan pengecekan kesehatan dengan metode *Polymerase Chain Reaction* (PCR). PCR ini dianggap metode pendeteksian COVID-19 paling efisien dan cepat sejauh ini. Menurut seorang dokter spesialis pulmonologi, apabila rumah sakit memiliki fasilitas yang memadai, pasien diwajibkan melakukan *Chest X-ray* sebagai “penegasan”, sekaligus sebagai pertimbangan untuk menentukan *treatment* yang harus dilakukan pasien. *EfficientNet* adalah arsitektur CNN yang dibuat oleh

peneliti di Google dengan *Neural Architecture Search* [20]. Arsitektur *EfficientNet* memiliki kelebihan dibandingkan dengan arsitektur CNN (*Convolutional Neural Network*) seperti ResNet, VGG, Gpipe yaitu memiliki akurasi yang tinggi tetapi dengan parameter yang sedikit sehingga selain mendapatkan akurasi yang tinggi, waktu inferensi yang cepat karena biaya komputasi yang rendah dan hemat dalam penggunaan *memory*. Dalam kata lain, meskipun model *EfficientNet* tergolong kecil, akurasi tidak terkorbankan dalam melakukan *task* klasifikasi, sehingga performa *EfficientNet* sangat baik [5].

## 2. DASAR TEORI

### 2.1 COVID-19

COVID-19 adalah penyakit yang menyerang saluran pernafasan oleh jenis coronavirus yang termasuk kedalam satu keluarga dengan MERS dan SARS. COVID-19 pertama kali ditemukan di kota Wuhan Seafood Market provinsi Hubei pada Desember 2019 dengan jenis coronavirus yang tidak diketahui. Gejala yang selalu dialami oleh orang yang terkena COVID-19 yaitu panas, batuk kering, dan cepat lelah. Yang jarang terjadi adalah seperti kehilangan alat indra penciuman, pusing, hidung tersumbat, diare, dan perubahan warna pada jari tangan dan kaki. Pada tahap yang parah, pasien yang terkena COVID-19 mengalami kesulitan bernafas hingga paling fatal kematian. Penyakit COVID-19 bisa menyebar melalui udara melalui hidung atau mulut ketika orang tersebut batuk, bersin, dan saat berbicara.[11]

Pada saat ini orang yang terkena virus COVID-19 di Indonesia pada bulan Februari telah mencapai lebih dari 1 juta, sekitar 30 ribu orang telah meninggal, dan hampir 900 ribu orang sembuh. Orang yang terkena COVID-19 terus meningkat dari hari ke hari. Kasus COVID-19 sekarang lebih dari 10.000 kasus harian, serta kematian lebih 300 orang.[12]

### 2.2 CNN (*Convolutional Neural Network*)

Merupakan pengembangan dari bentuk lain *Deep Neural Network* yang biasanya digunakan untuk data berupa Gambar atau data yang bersifat dimensinya banyak seperti Gambar. Dari *input* Gambar mentah yang akan menghasilkan *output* berupa nilai sebuah kelas yang akan di prediksi. Pada Gambar 1, arsitektur CNN terdiri dari [7]:

#### a) *Convolutional Layer*

Pada layer terjadi proses konvolusi antara kernel (bisa *filter*) dengan *input* yang outputnya adalah *feature map*. *Input* yang akan diberikan pada saat konvolusi yaitu *multidimensional array*. Di layer ini terjadi perkalian *scalar* antara kernel dimana ukurannya lebih kecil dari pada spasial *input*nya tetapi tersebar pada setiap input dengan input vektornya. Dalam *Convolutional Layer* ada 3 hyperparameter yaitu *depth*, *stride*, dan *zero padding*. *Depth* adalah *output* volume yang akan dihasilkan oleh *Convolutional Layer*. *Zero padding*, yaitu proses menambahkan nilai nol untuk mengontrol dimensi *output* volume dari hasil konvolusi, dan *stride* untuk seberapa dalam dimensi spasial atau banyaknya pixel yang bergeser pada setiap *input*.

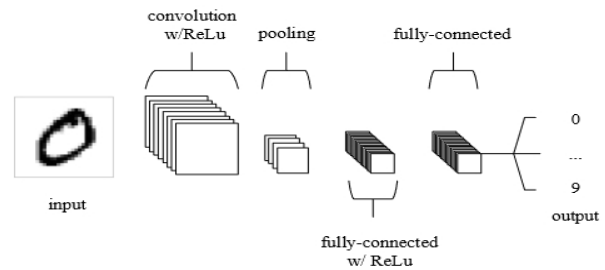
#### b) *Pooling Layer*

Pada layer ini, setiap dimensi yang sudah di konvolusi akan di kurangi dimensinya lebih kecil lagi. Setelah mengurangi dimensi maka dilakukan pengurangan parameter serta kompleksitas dalam model untuk mempercepat proses training. Ada beberapa macam *pooling* yaitu *max pooling* yaitu mengambil nilai maksimum pada *feature map*, *general pooling* dengan menggunakan normalisasi L1/L2, dan *average pooling* yaitu menggunakan nilai rata dari

*feature map*. Tetapi paling banyak adalah menggunakan *max pooling*. Pada Gambar 1 menggunakan operasi *max* pada saat melakukan *pooling*. Terdapat tulisan *max pooling* pada Gambar 1 dengan ukuran kernel 2x2 setelah melakukan operasi konvolusi.

#### c) *Fully Connected Layer*

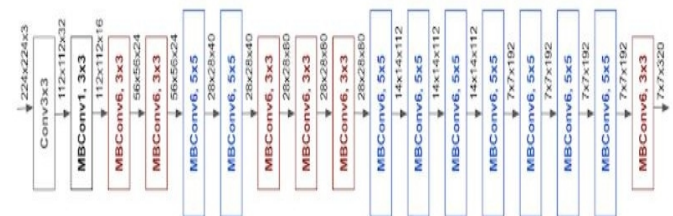
Pada layer ini terdapat neuron yang terhubung pada setiap layer. Pada layer ini menggunakan bentuk tradisional *Neural Network*. Layer ini akan menghasilkan *output* klasifikasi dari sebuah *class* yang bisa berbentuk probabilitas jika fungsi aktivasinya *softmax* atau bernilai antara 0 hingga 1 dengan menggunakan fungsi aktivasi *sigmoid*. Pada Gambar 1 terdapat tulisan *fully connected* yang menggunakan fungsi aktivasi *ReLU* (*Rectified Linear Unit*) yang mengeluarkan *output* sesuai nilai *feature map* menghasilkan nilai 0 jika hasil *feature map* kurang dari 0 atau nilai diatas 0 jika nilai *feature map* diatas 0. Didalam *library Tensorflow* dengan Keras API, *fully connected layer* disebut dengan *Dense Layer*.



Gambar 1. Proses CNN

### 2.3 Arsitektur *EfficientNet*

*EfficientNet* adalah arsitektur *Convolutional Neural Network* (CNN) yang dibuat melalui *Neural Architecture Search* untuk mendesain CNN dasar oleh Principal Scientist Quoc V. Le dan Mingxing Tan, Software Engineer Staff dari Google [10]. Berawal dari pemikiran dasar dari CNN adalah arsitektur *Neural Network* dengan *cost resource* yang *fixed* lalu melakukan *scaling up* untuk mendapatkan akurasi yang tinggi. Dalam praktik pembuatan arsitektur CNN khususnya dalam *scaling*, orang akan melakukan percobaan dengan mengubah dari lebar CNN, hingga resolusi Gambar untuk meningkatkan akurasi secara manual sampai mendapatkan nilai yang optimum. Sehingga mereka mengusulkan dengan metode *Neural Architecture Search / AutoML* (framework yang dipakai adalah *MNAS AutoML* yang dipakai oleh Google juga untuk membuat model *Mobile NASNet*) dalam melakukan otomatis mencari pengaturan *scaling* yang optimum sehingga mendapatkan akurasi yang tinggi tetapi dengan parameter lebih sedikit sehingga secara kecepatan inferensi bisa lebih cepat. Karena kalkulasi tidak memakan resource computing yang banyak, maka dari itu mereka mengeluarkan keluarga model CNN dengan nama *EfficientNet*. *Scaling* yang dilakukan oleh peneliti, yaitu *compound scaling* yaitu dimana dilakukan *scaling up* pada *width*, *height*, dan resolusi.[10] Pada Gambar 2, diilustrasikan arsitektur *EfficientNet* B-0.



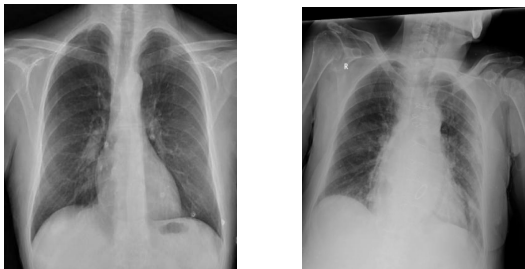
Gambar 2. Arsitektur *EfficientNet*

### 3. DESAIN SISTEM

#### 3.1 Dataset

Dataset yang akan digunakan adalah *COVID-19 Radiography Database*. Dataset ini dibuat oleh tim riset dari universitas di Qatar, Bangladesh, Pakistan, dan Malaysia. *Dataset* ini terbuka secara publik melalui Kaggle. *Dataset* ini berisi Gambar dengan format PNG dengan ukuran 1024 x 1024 pixel, serta metadata yang memiliki format CSV untuk memetakan antara Gambar dengan label. Pada penyimpanan data sudah terorganisir berdasarkan label sehingga metadata berformat CSV untuk memperjelas saja. Label pada dataset ini ada 3 yaitu COVID-19, Normal, dan *Pneumonia*. Pada *dataset* terdapat 1200 data untuk orang yang terkena COVID-19, 1341 data untuk orang normal, dan 1345 data untuk orang yang terkena *pneumonia* [1].

Dataset ini merupakan gabungan dari COVID-19 *Image Data Collection* [3], *Italian Society of Medical and Interventional Radiology (SIRM) COVID-19 DATABASE, RSNA-Pneumonia-Detection-Challenge Chest X-Ray Images (pneumonia)* [4], tersedia di Kaggle secara publik dan COVID-19 *Chest imaging at thread reader*. *Dataset* ini diambil pada versi Januari 2021 dengan penambahan *dataset* COVID-19 dari COVID-19 *Image Repository* lalu *dataset* lain. Peneliti juga merilis paper juga melakukan deteksi virus COVID-19 dengan judul paper “*Can AI help in screening Viral and COVID-19 pneumonia?*” [1]. Gambar 3 menunjukkan salah satu contoh *dataset* COVID-19 *Radiography Database*.



Gambar 3. X-Ray Pasien Penderita COVID-19 (Kiri) dan Pasien Normal (Kanan)

#### 3.2 Preprocessing

Dataset COVID-19 *Radiography Database* tidak dibagi dalam data *training*, *testing*, maupun validasi. Dataset akan dilakukan *split* dengan formasi 60 persen *training*, 20 persen validasi, dan 20 persen untuk *testing*.

Sebelum masuk dalam proses *training*, akan dirubah ukuran Gambar agar bisa dipakai pada arsitektur *EfficientNet*. *EfficientNet* memiliki 8 arsitektur yang memiliki input ukuran Gambar yang berbeda juga. Dalam penelitian ini akan menggunakan arsitektur B2 dengan ukuran Gambar 260 kali 260 pixel dan B1 dengan ukuran Gambar 240 kali 240 pixel dan B0 dengan ukuran Gambar 224 kali 224 pixel. Untuk B3 sampai B7 tidak dipakai karena memory GPU yang tidak mampu untuk melakukan proses *training* karena membutuhkan GPU VRAM yang lebih banyak (model semakin besar maka parameter pada *Neural Network* semakin besar sehingga keperluan GPU VRAM juga semakin besar. Lalu dilakukan juga untuk mengubah *input channel* setiap Gambar, yaitu RGB. Hal ini dilakukan karena seluruh arsitektur *EfficientNet* hanya bisa menerima *input channel* Gambar sebesar 3.

Setelah melakukan perubahan Gambar maka dilakukan proses augmentasi. Augmentasi adalah proses menambah dataset Gambar dengan melakukan transformasi Gambar. Transformasi yang akan digunakan adalah transformasi geometri yaitu rotasi, translasi dan, *zoom* karena paling mudah digunakan dimana seluruh *library deep learning* sudah menyiapkan augmentasi Gambar geometris [8]. Penelitian ini akan dilakukan augmentasi rotasi dengan parameter *factor* 0.15, dan translasi Gambar dengan *factor height* dan *factor width* sebesar 0.1, serta melakukan random *zooming* dengan *width factor* 0.5 dan *height factor* 0.2.

#### 3.3 Load Model

Sebelum melakukan proses *training*, akan dilakukan load *pretrained weight*. Ini dilakukan karena dataset yang sedikit meskipun dataset sudah dilakukan augmentasi Gambar. *Deep Learning* memiliki kelemahan yaitu ketergantungan pada dataset sangat tinggi [14]. Sehingga dataset untuk model *Deep Learning* haruslah besar. Masalah pada dunia nyata seperti data klinikal pada kasus ini memiliki jumlah dataset yang sedikit, salah satunya adalah data medis [9]. *Transfer learning* adalah menggunakan *weight* yang sudah *training* pada dataset yang besar lalu memodifikasi pada layer *classifier* pada *Neural Network* untuk dilakukan *training* pada dataset lain untuk menghasilkan model yang baru [6]. *Transfer learning* dapat mengurangi generalisasi *error* [14]. Maka pada kasus ini cocok menggunakan metode *transfer learning* dari *pretrained weight* yang sudah disediakan oleh *library*. Peneliti *weight* yang digunakan adalah *noisy student* dimana hasil dari *training semi supervised learning* antara model *teacher* dengan *student*, dimana *top 1 accuracy* lebih tinggi 2 persen dengan *EfficientNet* dengan *pretrained EfficientNet* daripada *weight imagenet* biasa. [13]

#### 3.4 Proses Training

Pada tahap ini, akan dilakukan perubahan pada bagian *classifier* dengan konfigurasi yang sesuai pada penelitiannya. Untuk bagian *classifier* baru pada *EfficientNet* akan menghasilkan 3 output yaitu berbentuk probabilitas pada setiap class yaitu normal, COVID-19, dan *pneumonia* dengan fungsi aktivasi *softmax*.

Pada proses *training* akan dilakukan penambahan *callback*, yaitu *checkpoint* untuk menyimpan model terbaik. *Checkpoint* ini akan menyimpan berdasarkan dengan akurasi terbaik pada saat melakukan validasi untuk mendapatkan akurasi terbaik pada saat memprediksi data terbaru. Sehingga yang disimpan adalah model yang terbaik saja jika ada penurunan pada akurasi validasi tidak akan disimpan.

Jumlah *epoch* yang akan dicoba adalah 10, dan 100 dengan *batch size* 16 agar data yang akan diproses melalui GPU dengan jumlah VRAM yang cukup untuk diproses, karena semakin besar ukuran *batch* maka membutuhkan jumlah GPU VRAM yang lebih besar. Jika semakin besar *batch* dapat menimbulkan resiko GPU *out of memory*. *Optimizer* yang digunakan adalah Adam dan SGD dengan 3 konfigurasi *learning rate*, yaitu 1e-3, 1e-4, dan 1e-5. Pada konfigurasi *optimizer* SGD ada tambahan parameter, yaitu momentum sebesar 0.9.

#### 3.5 Proses Inference

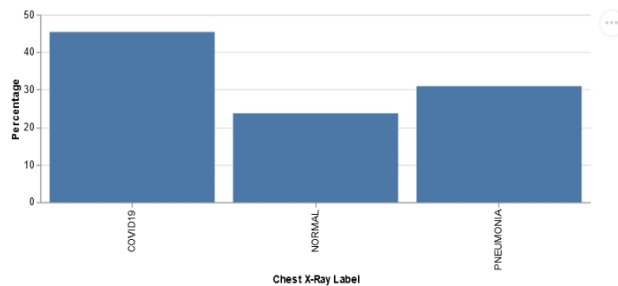
Pada tahap ini akan dilakukan proses pengujian aplikasi. Aplikasi ini berbasis *website* yang dibuat dengan *framework streamlit*. *User* akan mengunggah Gambar dalam format JPEG, PNG, atau JPG. Kemudian model yang sudah dilakukan *training* pada *dataset* COVID-19 akan melakukan perubahan ukuran Gambar agar bisa diterima oleh model *EfficientNet*. Setelah itu akan

melakukan prediksi dengan fungsi *predict* dari *pretrained weight* yang sudah *training* pada dataset *COVID-19*. Hasil *output* ini adalah dalam bentuk probabilitas setiap *class* yang dihasilkan dari *fully connected layer* (didalam keras disebut *dense layer*) melalui fungsi aktivasi *softmax*. Pada aplikasi akan ditunjukkan label hasil prediksi *EfficientNet* berserta hasil probabilitas dalam bentuk *bar chart*. Tampilan aplikasi ditunjukkan pada Gambar 4, kemudian Gambar 5 menunjukkan hasil statistik dari prediksi gambar.



Gambar 4. Tampilan Program hasil Prediksi dari Chest X-Ray

### Pneumonia Statistics:



Gambar 5. Tampilan Probabilitas Setiap Class hasil Prediksi

### 3.6 Model *EfficientNet*

Untuk arsitektur *EfficientNet* yang digunakan adalah *EfficientNet* B-0, *EfficientNet* B-1, dan *EfficientNet* B-2. Hal ini dilakukan untuk mencegah GPU *Out Of Memory* karena semakin besar model *EfficientNet*, maka *memory* pada kartu grafis juga besar. Berikut adalah Tabel 1 yang berisi jenis dan jumlah *layer* pada *EfficientNet*. Pada Tabel 1 dan 2 menggunakan singkatan FC yang merujuk pada *Fully Connected Layer*.

Tabel 1. Jenis dan Jumlah *Layer EfficientNet*

Jenis Layer	Jumlah Layer
Conv 3x3	1
MBCConv1 dengan Kernel 3x3	1
MBCConv6 dengan Kernel 3x3	2
MBCConv6 dengan Kernel 5x5	2
MBCConv6 dengan Kernel 3x3	3
MBCConv6 dengan Kernel 5x5	3
MBCConv6 dengan Kernel 5x5	4

Tabel 1. Jenis dan Jumlah *Layer EfficientNet* (lanjutan)

Jenis Layer	Jumlah Layer
MBCConv6 dengan Kernel 3x3	1
Conv1x1 => Pooling => FC	1

Jumlah layer berdasarkan Tabel 1 berlaku untuk semua bentuk *EfficientNet* termasuk *EfficientNet* yang akan dipakai, yaitu *EfficientNet* B-0, *EfficientNet* B-1, dan *EfficientNet* B-2. Perbedaan yang ada adalah *input* ukuran Gambar, pada *EfficientNet* B-0 memiliki *input* ukuran Gambar 224x224 pixel, *EfficientNet* B-1 memiliki *input* ukuran Gambar 240x240, dan *EfficientNet* B-2 memiliki *input* ukuran Gambar 260x260. *EfficientNet* B-0 dan *EfficientNet* B-1 memiliki *channel* pada setiap *block*, sedangkan untuk jumlah *channel* pada *EfficientNet* B-2 memiliki jumlah *channel* yang lebih banyak untuk meningkatkan jumlah parameter. Berikut adalah Tabel 2 berisi jumlah *channel* pada setiap arsitektur.

Tabel 2. Jenis dan Jumlah *Layer EfficientNet*

Jenis Layer	B-0	B-1	B-2
Conv 3x3	32	32	32
MBCConv1 dengan Kernel 3x3	16	16	16
MBCConv6 dengan Kernel 3x3	24	24	24
MBCConv6 dengan Kernel 5x5	40	40	48
MBCConv6 dengan Kernel 3x3	80	80	88
MBCConv6 dengan Kernel 5x5	112	112	120
MBCConv6 dengan Kernel 5x5	192	192	208
MBCConv6 dengan Kernel 3x3	320	320	352
Conv1x1 => Pooling => FC	1280	1280	1408

## 4. PENGUJIAN SISTEM

Pengujian ini dimaksudkan untuk mendapatkan konfigurasi *EfficientNet* terbaik dari segi metrik akurasi, *F1-Score*, *recall*, *precision*, dan *specificity* pada dataset *COVID-19 Radiography Database*. Arsitektur *EfficientNet* sendiri memiliki 8 model, namun penelitian ini hanya mencoba 3 model diantaranya, yaitu tipe B-0, B-1, dan B-2. *Hardware* yang dipakai saat ini memiliki spesifikasi processor Intel Core-i7 dengan tipe 8750H, RAM sebesar 16 GB serta kartu grafis Nvidia GeForce GTX 1060 sebesar 6 GB. Pengujian antar arsitektur *EfficientNet* hingga *learning rate* akan menggunakan jumlah validasi sebesar 20 persen. Berikut adalah Tabel 3 untuk menjelaskan jumlah *dataset* yang sudah dilakukan dilakukan *split* lebih detail setiap per *class* dengan formasi 10 persen *training*, 20 persen validasi, dan 20 persen untuk *testing*.

Tabel 3. Jumlah Data Point Setiap *Class*.

Jenis Data	COVID-19	Normal	Pneumonia
Train	768	860	857
Validation	192	216	215
Testing	240	269	269

### 4.1 Pengujian Arsitektur *EfficientNet*

Pada bagian ini akan dilakukan eksperimen untuk menguji arsitektur mana yang paling optimal dengan pengukuran metrik yang sudah dibahas pada bab 2. Pengujian akan dilakukan pada arsitektur *EfficientNet* B-0, *EfficientNet* B-1, *EfficientNet* B-2. Pada pengujian menggunakan konfigurasi *paper* dari ECOVNet [2] dimana dengan konfigurasi menambahkan *dropout* sebesar 0.3, dan menambahkan regularisasi L1 dan L2 sebagai penalti pada *weight* dengan sebesar 1e-5 untuk L1 dan 1e-3 untuk L2 untuk mencegah *overfitting*. Pada pengujian ini juga menggunakan

optimizer Adam dengan *learning rate*  $1e-4$ . Untuk ukuran batch sebesar 16, berbeda dengan ECOVNet dengan ukuran batch 8 [2]. Serta epoch pada penelitian ini adalah 10 jika dibandingkan dengan ECOVNet adalah 25 kali [2]. Berikut adalah Tabel 4 perbandingan hasil pengujian setiap arsitektur dengan acuan data *testing*.

**Tabel 4. Metrik Hasil Pengujian Arsitektur *EfficientNet***

Arsitektur	Accuracy	F1-Score	Precision	Recall	Specificity
B-0	0.9254	0.9277	0.9303	0.9251	0.9621
B-1	0.9319	0.9353	0.9377	0.9330	0.9653
B-2	0.9203	0.9284	0.9356	0.9214	0.9594

Dari hasil eksperimen yang dilakukan dari Tabel 4 didapatkan bahwa arsitektur *EfficientNet B-1* mendapatkan akurasi, *F1-Score*, *precision*, dan *recall* tertinggi. Namun jika dibandingkan dengan arsitektur *EfficientNet B-0* dan *EfficientNet B-2*, ketiganya memiliki nilai yang tidak selisih jauh. Untuk *EfficientNet B-0* dan *EfficientNet B-2* memiliki akurasi yang hampir sama, yaitu sekitar 92 persen, tetapi *EfficientNet B-0* memiliki nilai akurasi yang lebih tinggi. Dimana dari keseluruhan metrik, akurasi *EfficientNet B-1* memiliki nilai tertinggi yang mencapai 93 persen, kemudian baru diikuti oleh dengan *EfficientNet B-0* dan terakhir diikuti *EfficientNet B-0*.

**Tabel 5. Metrik Hasil Eksperimen Perbandingan Epoch 10 dan 100 Pada *EfficientNet B-1***

Epoch	Accuracy	F1-Score	Precision	Recall	Specificity
10	0.9319	0.9353	0.9377	0.9330	0.9653
100	0.9679	0.9672	0.9678	0.9666	0.9836

Pada Tabel 5 menunjukkan bahwa penggunaan *epoch* 100 dapat menyebabkan kenaikan nilai *accuracy* sebesar 3 persen dari *epoch* 10. Pada eksperimen ini menunjukkan bahwa dengan meningkatkan jumlah *epoch* pada *EfficientNet B-1* dapat menghasilkan titik optimum yang lebih baik. Maka dari itu untuk eksperimen seterusnya akan menggunakan jumlah *epoch* sebesar 100 kali.

## 4.2 Pengujian Dengan Arsitektur Lain

Pada bagian ini mencoba membandingkan dengan arsitektur lain sesuai dengan *paper* pembuat dataset *COVID-19 Radiography Database* [2]. Dalam *paper* pembuat dataset ini ada dua kali pengujian yaitu pada kasus pertama yaitu *binary classification* dan *multiclass classification*. Pada kasus ini akan diambil pada bagian *multiclass* karena pada eksperimen yang dilakukan adalah *multiclass classification*. Di dalam *paper* juga dibagi menjadi dua yaitu hasil eksperimen dengan data telah teraugmentasi dan data tidak teraugmentasi. Pada kali ini hanya akan menggunakan hasil eksperimen *paper* pada *dataset* yang telah diaugmentasi oleh pembuat dataset. Berikut adalah Tabel 6 perbandingan dengan *paper* pembuat *dataset* dengan eksperimen ini dalam kasus ini menggunakan arsitektur *EfficientNet B-1*.

**Tabel 6. Perbandingan Hasil Pengukuran Metrik Eksperimen ini dengan Paper Pembuat Dataset COVID-19 Radiography Database.**

Arsitektur	Accuracy	F1-Score	Precision	Recall	Specificity
<i>EfficientNet B-1</i>	0.9679	0.9672	0.9678	0.9666	0.9836
SqueezeNet	0.9510	0.9514	0.9518	0.9510	0.9717

**Tabel 6. Perbandingan Hasil Pengukuran Metrik Eksperimen ini dengan Paper Pembuat Dataset COVID-19 Radiography Database (lanjutan)**

Arsitektur	Accuracy	F1-Score	Precision	Recall	Specificity
InceptionV3	0.9620	0.9660	0.9700	0.9640	0.9750
ResNet101	0.9622	0.9623	0.9624	0.9622	0.9780
DenseNet201	0.9794	0.9794	0.9795	0.9794	0.9780
VGG19	0.9600	0.9637	0.9650	0.9625	0.9752
ResNet18	0.9644	0.9646	0.9648	0.9644	0.9791
MobileNet V2	0.9622	0.9623	0.9624	0.9623	0.9780

Pada hasil eksperimen pada Tabel 6, menunjukkan bahwa arsitektur DenseNet memiliki nilai metrik paling tinggi, kemudian disusul oleh *EfficientNet B-1* dan selisih dari kedua arsitektur ini hanya 0.01 poin pada nilai metrik. Pada eksperimen ini, dapat dilihat bahwa adanya perbedaan nilai metrik yang tidak signifikan ditunjukkan oleh arsitektur *EfficientNet B-1* dengan arsitektur perbandingan lainnya yang memiliki nilai metrik dibawah *EfficientNet B-1*. Berdasarkan informasi yang ada pada Tabel, arsitektur *EfficientNet B-1* kalah pada nilai *precision* jika dibandingkan dengan *InceptionV3* dan *DenseNet201*. Namun jika melihat nilai metrik lainnya, *EfficientNet B-1* masih memiliki nilai metrik yang lebih baik diantara lainnya.

Demi analisa lebih dalam, maka eksperimen akan dilanjutkan dengan melihat perbedaan waktu. Arsitektur yang dibandingkan adalah *EfficientNet B-1* dan *DenseNet*. Konfigurasi *DenseNet* akan disesuaikan dengan jurnal pembuat *dataset* COVID-19.

**Tabel 7. Perbandingan Ukuran Model, Jumlah Parameter, dan Kecepatan antara *EfficientNet B-1* dengan *DenseNet201***

Arsitektur	Ukuran Model	Parameter	Kecepatan
<i>DenseNet201</i>	145 MB	18 Juta	9.9891 seconds
<i>EfficientNet B-1</i>	78 MB	6 juta	5.4704 seconds

Tabel 7 menampilkan perbandingan waktu hasil inferensi pada data *testing* dari DenseNet dengan *EfficientNet*. Arsitektur *DenseNet* memiliki jumlah parameter yang lebih banyak dan ukuran model yang besar. Apabila jumlah parameter semakin besar maka ukuran model juga akan semakin besar dan menyebabkan waktu inferensi yang lama karena komputasi pada *matrix weight* yang besar. Hal ini hasil eksperimen yang dapat dilihat dari Tabel 7 bahwa *DenseNet201* memiliki ukuran model 3 kali lipat lebih besar dari *EfficientNet B-1*. Selain itu, waktu inferensi *DenseNet201* memiliki waktu yang lebih lama dibandingkan dengan *EfficientNet B-2* dengan selisih 4 detik. Demikian juga untuk jumlah parameter, dimana *DenseNet201* memiliki jumlah 2 kali lipat lebih banyak dari *EfficientNet B-2*.

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Setelah dilakukan pengujian dengan menggunakan arsitektur *EfficientNet* dalam melakukan klasifikasi *Chest X-Ray* COVID-19, *pneumonia*, dan normal, maka dapat disimpulkan sebagai berikut:

- Dari ketiga arsitektur yang diuji, arsitektur *EfficientNet B-1* mendapatkan akurasi paling baik. Namun seluruh arsitektur *EfficientNet B-0*, *B-1*, maupun *B-2* masih memiliki akurasi diatas 90 persen.

- b) *Epoch* yang baik untuk *training* adalah 100 kali, dimana *loss* akan menurun dan akan berdampak perubahan nilai metrik. Namun pengaturan ini memiliki konsekuensi, yaitu waktu yang dibutuhkan untuk melakukan *training* akan jauh lebih lama.
- c) Dari hasil eksperimen, *EfficientNet* B-1 menghasilkan nilai metrik yang tinggi dengan waktu inferensi yang lebih cepat karena jumlah parameter yang kecil tetapi memiliki akurasi yang baik. Jika dibanding *DenseNet201*, meski nilai metrik pada *DenseNet201* lebih tinggi daripada *EfficientNet* B-1, tetapi *EfficientNet* B-1 memiliki waktu inferensi yang lebih cepat daripada *DenseNet201*.

## 5.2 Saran

Karena banyaknya parameter yang bisa diuji tetapi dalam pengerjaan waktu terbatas, saran yang akan diberikan dalam melakukan pengembangan lebih lanjut adalah :

- a) Mencoba meningkatkan jumlah *epoch* diatas 100
- b) Mencoba meningkatkan persentase validasi dari 20 persen ke 30 persen.
- c) Jika ingin menggunakan regularisasi, maka perlu melakukan eksperimen regularisasi pada L1 dan L2 dengan angka yang berbeda dari konfigurasi ECOVNet, seperti memberikan nilai *learning rate*  $1e-4$  pada salah satu tipe regularisasi.
- d) Mencoba *optimizer* lain seperti *adadelta* atau Adam *weight decay* untuk membuktikan apakah *optimizer* tersebut lebih *convergence* model untuk mencapai titik optimum.
- e) Menggunakan algoritma optimasi seperti *genetic algorithm* atau *bayesian optimization* untuk melakukan kombinasi *parameter tuning* secara otomatis. Hal ini untuk menghindari waktu eksperimen yang lebih banyak akibat banyaknya kombinasi parameter yang bisa digunakan pada *EfficientNet*.
- f) Mencoba arsitektur *EfficientNet* versi kedua dengan jumlah parameter yang lebih kecil dari eksperimen ini. Selain itu, bisa mencoba arsitektur lainnya seperti ViT yang memiliki akurasi yang jauh lebih tinggi dari *EfficientNet*.

## 6. DAFTAR REFERENSI

- [1] Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir M. A., Mahbub, Z. Bin, Islam, K. R., Khan, M. S., Iqbal, A., Al-Emadi, N., & Reaz, M. B. I. 2020. *Can AI help in screening Viral and COVID-19 pneumonia?* URI=<http://arxiv.org/abs/2003.13145>
- [2] Chowdhury, N. K., Kabir, M. A., Rahman, M. M., & Rezoana, N. 2020. *ECOVNet: an ensemble of Deep Convolutional Neural Networks based on EfficientNet to detect covid-19 from chest x-rays a preprint.* URI=<https://arxiv.org/abs/2003.13145>
- [3] Cohen, J. P., Morrison, P., & Dao, L. 2020. *COVID-19 Image Data Collection.* URI=<http://arxiv.org/abs/2003.11597>
- [4] Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., ... Zhang, K. 2018. *Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning.* *Cell*, 172(5), 1122-1131.e9. DOI=<https://doi.org/10.1016/j.cell.2018.02.010>
- [5] Luz, E., Silva, P. L., Silva, R., Silva, L., Moreira, G., & Menotti, D. 2020. *Towards an Effective and Efficient Deep Learning Model for COVID-19 Patterns Detection in X-ray Images.* URI=<http://arxiv.org/abs/2004.0571>
- [6] Maghdid, H. S., Asaad, A. T., Ghafoor, K. Z., Sadiq, A. S., & Khan, M. K. 2020. *Diagnosing COVID-19 Pneumonia from X-Ray and CT Images using Deep Learning and Transfer Learning Algorithms.* 1-8. URI=<http://arxiv.org/abs/2004.00038>
- [7] O'Shea, K., & Nash, R. 2015. *An Introduction to Convolutional Neural Networks.* November. URI=<http://arxiv.org/abs/1511.08458>
- [8] Shorten, C., & Khoshgoftaar, T. M. 2019. *A survey on Image Data Augmentation for Deep Learning.* *Journal of Big Data*, 6(1). DOI = <https://doi.org/10.1186/s40537-019-0197-0>
- [9] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. 2018. A survey on deep transfer learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11141 LNCS, 270-279. DOI = [https://doi.org/10.1007/978-3-030-01424-7\\_27](https://doi.org/10.1007/978-3-030-01424-7_27)
- [10] Tan, M., & Le, Q. V. 2019. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.* URI=<https://arxiv.org/abs/1905.11946>
- [11] World Health Organization. *Q&A on coronaviruses (COVID-19).* URI = <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/q-a-coronaviruses#:~:text=symptoms>
- [12] Worldometer. Indonesia Coronavirus. URI=<https://www.worldometers.info/coronavirus/country/indonesia/>
- [13] Xie, Q., Luong, M.-T., Hovy, E., & Le, Q. V. 2020. *Self-Training With Noisy Student Improves ImageNet Classification.* 10684-10695. URI=<https://arxiv.org/abs/1911.04252>
- [14] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. 2014. *How transferable are features in Deep Neural Networks?* *Advances in Neural Information Processing Systems.* URI=<https://arxiv.org/abs/1411.1792>