

Sistem Pengenalan Covid-19 Berdasarkan Foto X-ray Paru dengan Metode EfficientNet-B0

Jourdan Stanley¹, Chairisni Lubis², Teny Handayani³

Teknik Informatika, FTI, Universitas Tarumanagara

Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia

email : 1jourdan.535180097@stu.untar.ac.id, 2chairisnil@fti.untar.ac.id, 3tenyh@fti.untar.ac.id

ABSTRACT

Covid-19 is a viral infection disease severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), Covid-19 is a group of viruses that attack the respiratory system in humans which can cause symptoms ranging from mild symptoms to severe symptoms. Currently, to detect whether a person is infected with the Covid-19 virus or not, several tests can be carried out, one of which is the polymerase chain reaction (PCR) examination. This type of examination has a high level of accuracy but this examination requires quite expensive costs, adequate laboratories and requires a long time. So from these problems there is another alternative, namely radiological examination. From these problems, a system was built that can perform classification based on x-ray images of the lungs using the convolutional neural network (CNN) method of Efficientnet-B0 architecture. This system is expected to assist medical personnel in pre-diagnosing a patient's lung condition based on their lung x-ray without changing the role of the medical personnel. After successfully building a Covid-19 recognition system, the system will be tested using the confusion matrix method where in this test there are 2 scenarios. In the first scenario, the data trained using the CLAHE preprocessing method obtained an accuracy rate of 98%, while in the second scenario the data was trained without using the CLAHE preprocessing method, the results obtained an accuracy rate of 97%. Previous research was conducted using the resnet-18 method and obtained an accuracy rate of 92%. From the results obtained prove that Efficientnet is able to increase the level of accuracy from previous studies.

Key words

Covid-19, Confusion matrix, Convolutional neural network (CNN), efficientnet-B0, preprocessing

1. Pendahuluan

Covid-19 merupakan penyakit menular yang disebabkan oleh virus SARS-CoV-2. Orang yang

terinfeksi dengan penyakit ini akan mengalami penyakit pernapasan dari gejala ringan hingga gejala berat Kasus pertama kali ditemukan penyakit Covid-19 terjadi adalah di kota Wuhan, Cina pada tanggal 1 Desember 2019. Menurut *World Health Organization* (WHO) pada tanggal 7 Juni 2022 telah ada 529.410.287 kasus Covid-19 yang dikonfirmasi, termasuk 6.296.771 kasus kematian yang disebabkan oleh Covid-19 [1].

Saat ini ada cara untuk mengetahui apakah seseorang terjangkit virus Covid-19 atau tidak yaitu bisa dengan menjalani beberapa test seperti *rapid test antibody*, *Swab antigen (Rapid antigen test)*, dan *PCR (polymerase chain reaction)*. Test *PCR (polymerase chain reaction)* sendiri memiliki kelebihan yaitu mampu mendeteksi antigen dengan konsentrasi yang rendah dan memiliki tingkat akurasi yang tinggi berkisar antara 91-100%. Namun PCR juga memiliki kekurangan yaitu memerlukan biaya pemeriksaan yang cukup mahal, laboratorium yang memadai, serta waktu pengerjaan yang cukup lama.

Alternatif lainnya adalah pemeriksaan radiologi. Ada beberapa kasus dengan hasil tes virologi yang negatif, pemeriksaan radiologi dapat menunjukkan adanya infeksi paru-paru sehingga dapat mendeteksi infeksi virus pada tahap awal.

Dengan perkembangan teknologi informasi pada saat ini, ada sebuah cara yang bisa digunakan untuk melakukan proses pengenalan covid-19 dengan menggunakan foto *x-ray* paru-paru. Cara ini adalah dengan menggunakan metode *convolutional neural network* (CNN). Metode CNN sendiri memiliki berbagai macam arsitektur, arsitektur yang digunakan pada pembuatan sistem ini adalah *Efficientnet-B0*.

Penelitian ini dilakukan dengan memperoleh data dari Kaggle.com dengan jumlah data sebanyak 1500 foto *x-ray* paru yang terbagi menjadi 3 kelas yaitu, Covid-19, *Pneumonia* dan Normal. Setelah selesai mengembang sistem pengenalan Covid-19 ini akan dilakukan evaluasi untuk mengetahui performa prediksi sistem yang dikembangkan. Metode evaluasi yang digunakan yaitu *Confusion matrix* dengan parameter *precision*, *recall*, *f1-score*, dan *accuracy*.

2. Landasan Teoretik

2.1 Covid-19

Coronavirus pertama kali ditemukan pada tanggal 1 Desember 2019 di kota Wuhan, Cina. *Coronavirus* merupakan virus RNA dengan ukuran yang kecil sekitar 120-160 nm. Pada awalnya virus Covid-19 merupakan virus yang menginfeksi hewan, seperti kelelawar dan unta. Kemudian di perkirakan virus yang ada pada hewan itu menular kepada manusia. Sebuah penelitian yang dilakukan oleh virologi dengan analisis filogenetik menunjukkan bahwa virus ini termasuk ke dalam *genus betacoronavirus*[2]. *Coronavirus* merupakan kelompok virus yang menyerang sistem pernapasan pada manusia yang bisa mengakibatkan gejala mulai dari gejala ringan sampai gejala berat.

2.2 Pneumonia

Pneumonia adalah infeksi akut yang menyerang jaringan paru-paru (alveoli) yang disebabkan oleh bakteri, virus, maupun jamur. *Pneumonia* dapat juga disebabkan oleh iritasi kimia dari paru-paru atau akibat dari penyakit lainnya, seperti kanker paru-paru atau terlalu banyak minum alcohol [3]. *Pneumonia* merupakan suatu infeksi yang bisa ditularkan melalui pernafasan. Pendeteksian bisa dilakukan dengan melakukan *c-reactive protein* (CRP) serum 24-48 jam. Metode ini merupakan uji laboratorium yang telah dikenal luas untuk mendiagnosis dan memonitor berbagai proses infeksi dan inflamasi akut, termasuk *pneumonia*.

2.3. Contrast Limited Adaptive Histogram Equalization (CLAHE)

Metode *Contrast Limited Adaptive Histogram Equalization* (CLAHE) merupakan salah satu proses peningkatan kualitas citra, agar analisis visual dan komputasi dapat menghasilkan hasil yang lebih baik. Metode ini sendiri sangat berguna pada bidang medis karena berhasil diterapkan dalam USG payudara dan peningkatan citra mamografi dalam sel segmentasi gambar, dan dalam peningkatan kualitas gambar patah tulang [4].

Keunggulan dari metode CLAHE ini adalah mampu meningkatkan kualitas citra menjadi lebih baik dari citra aslinya terutama pada citra *grayscale* yang sering digunakan di bidang medis yaitu foto *x-ray* dan mampu membuat nilai *pixel* yang tersembunyi terlihat lebih jelas. Adapun algoritma dari metode CLAHE adalah sebagai berikut :

1. Citra dibagi menjadi blok-blok berukuran 8 x 8 yang tidak saling menumpang tindih.
2. Melakukan perhitungan histogram pada setiap blok yang telah dibagi pada langkah pertama.
3. Meningkatkan kontras pada setiap blok dengan mengatur batas *clip limit* histogram, Menghitung *clip limit* suatu histogram dilakukan dengan persamaan (1) sebagai berikut.

$$\beta = \frac{M}{N} \left(1 + \frac{a}{100} (s - 1) \cos \cos \frac{1}{2} (\alpha \mp \beta) \right) \quad (1)$$

Keterangan :

M = Luas *Region Size*

N = Nilai *Region Scale* (256)

A = *Clip factor* yang bernilai antara 0 sampai 100

4. Melakukan pemotongan histogram, histogram akan didistribusikan kembali.
5. Setiap histogram blok diubah dengan menggunakan persamaan (2) sebagai berikut.

$$A_t = \sum_i^t = 0p_t(A_i) \quad (2)$$

dimana $p_t(A_i)$ merupakan fungsi kepadatan probabilitas untuk nilai *grayscale* blok yang terdapat pada citra *input* pada i dan didefinisikan dengan persamaan (3) berikut.

$$p_t(A_i) = \frac{m_i}{m} \quad (3)$$

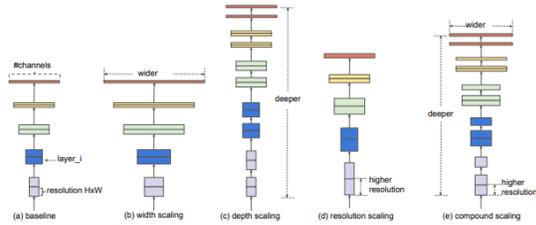
dimana m_i adalah nilai *grayscale* dari piksel yang diinput i , dan m adalah jumlah total piksel dalam satu blok.

6. Interpolasi bilinear yang berfungsi untuk menggabungkan blok-blok yang saling bersinggungan antara satu dengan yang lain.

2.4. Efficientnet-B0

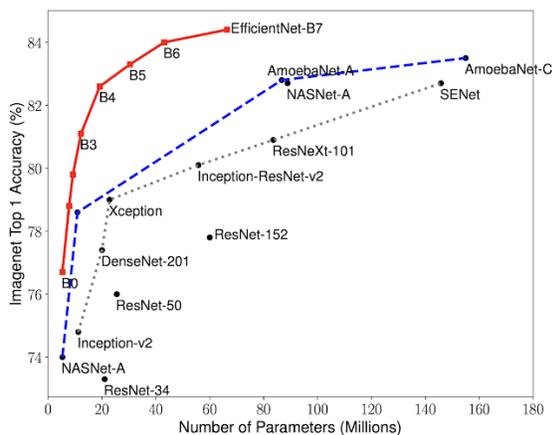
Penelitian yang diterapkan oleh Tan dan Le pada tahun 2019 adalah membuat pengembangan baru tentang *neural network*. Penelitian itu menjabarkan tentang suatu model *scalling* dan cara untuk identifikasi hubungan antara *depth*, *width*, dan *resolution* untuk mencapai performa yang lebih baik dari arsitektur CNN lainnya [5]. Dari penelitian tersebut ditemukan lah *Efficientnet*.

Efficientnet adalah suatu arsitektur CNN yang ditemukan dengan melakukan *scaling* secara teratur pada tiga komponen, yaitu lebar (*width*) mengacu pada jumlah *channel* pada lapisan manapun, kedalaman (*depth*) mengacu pada lapisan di CNN dan resolusi (*resolution*) mengacu pada ukuran resolusi gambar yang digunakan. Arsitektur dari *Efficientnet* dapat dilihat pada **Gambar 1**.



Gambar 1 Arsitektur Efficientnet

Kelebihan dari arsitektur ini yaitu mampu memberikan akurasi yang tinggi, dan mampu memperbaiki efektivitas model, dengan melakukan pengurangan parameter dan *Floating Point Operations Per Second* (FLOPS). Penambahan ketiga komponen lebar (*width*), kedalaman (*depth*) dan resolusi (*resolution*) dilakukan dengan teratur sehingga mendapatkan jumlah parameter yang lebih sedikit dan membuat waktu proses menjadi lebih cepat namun bisa mendapatkan akurasi yang baik dari model-model sebelumnya. Perbandingan antara *Efficientnet* dengan model lainnya dapat dilihat pada Gambar 2.



Gambar 2 Perbandingan Efficientnet dengan metode CNN lainnya

Untuk model B0 hingga B7, resolusi citra yang di *input* itu berbeda-beda, berikut adalah resolusi citra yang cocok untuk setiap model :

Tabel 1 Resolusi citra yang cocok untuk model *efficientnet*

Model Dasar	Resolusi
B0	224 pixels

B1	240 pixels
B2	260 pixels
B3	300 pixels
B4	380 pixels
B5	456 pixels
B6	528 pixels
B7	600 pixels

Pada penelitian ini model *Efficientnet* yang akan digunakan adalah B0. Arsitektur *Efficientnet-B0* bisa dilihat pada Gambar 3 dibawah ini.

Stage i	Operator \hat{F}_i	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels \hat{C}_i	#Layers \hat{L}_i
1	Conv3x3	224×224	32	1
2	MBCConv1, k3x3	112×112	16	1
3	MBCConv6, k3x3	112×112	24	2
4	MBCConv6, k5x5	56×56	40	2
5	MBCConv6, k3x3	28×28	80	3
6	MBCConv6, k5x5	14×14	112	3
7	MBCConv6, k5x5	14×14	192	4
8	MBCConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Gambar 3 Arsitektur Efficientnet-B0

Gambar 3 diatas menggambarkan arsitektur pada model *Efficientnet-B0* yang terdiri dari beberapa tahapan ketika mengolah data citra. Berikut adalah tahapan-tahapan dalam melakukan klasifikasi citra :

- Dataset foto x-ray paru-paru di *resize* menjadi ukuran 224×224 sebagai data *input*.
- Stage 1
 - Tahap ini merupakan langkah pertama proses mengolah citra, *convolutional* dengan filter 3×3 dan dengan jumlah *channel* sebanyak 32.
 - Aktifkan fungsi *batch normalization*.
 - Aktifkan fungsi aktivasi ReLu.
- Stage 2
 - Tahap kedua melakukan *convolutional* dengan filter 3×3 dengan jumlah *channel* sebanyak 16.
 - Aktifkan fungsi *batch normalization*.
 - Aktifkan fungsi aktivasi ReLu.
 - Melakukan *global average*.

- e. Melakukan *reshape* menjadi 112 x 112
 - f. Melakukan 2 kali *convolution* dengan filter 3 x 3 dengan jumlah *channel* sebanyak 16.
 - g. Melakukan *multiply*.
 - h. Aktifkan fungsi *batch normalization*.
 - i. Melakukan proses MBConv1 dengan filter 3 x 3.
4. Stage 3
- a. Tahap ketiga melakukan *convolutional* dengan filter 3 x 3 dengan jumlah *channel* sebanyak 24.
 - b. Aktifkan fungsi *batch normalization*.
 - c. Aktifkan fungsi aktivasi ReLu.
 - d. Melakukan proses MBConv6 dengan filter 3 x 3.
5. Stage 4
- a. Tahap keempat melakukan *convolutional* dengan filter 3 x 3 dengan jumlah *channel* sebanyak 40.
 - b. Aktifkan fungsi *batch normalization*.
 - c. Aktifkan fungsi aktivasi ReLu.
 - d. Melakukan proses MBConv6 dengan filter 5 x 5.
6. Stage 5
- a. Tahap kelima melakukan *convolutional* dengan filter 3 x 3 dengan jumlah *channel* sebanyak 80.
 - b. Aktifkan fungsi *batch normalization*.
 - c. Aktifkan fungsi aktivasi ReLu.
 - d. Melakukan proses MBConv6 dengan filter 3 x 3.
7. Stage 6
- a. Tahap keenam melakukan *convolutional* dengan filter 3 x 3 dengan jumlah *channel* sebanyak 112.
 - b. Aktifkan fungsi *batch normalization*.
 - c. Aktifkan fungsi aktivasi ReLu.
 - d. Melakukan proses MBConv6 dengan filter 5 x 5.
8. Stage 7
- a. Tahap ketujuh melakukan *convolutional* dengan filter 3 x 3 dengan jumlah *channel* sebanyak 192.
 - b. Aktifkan fungsi *batch normalization*.
 - c. Aktifkan fungsi aktivasi ReLu.
 - d. Melakukan proses MBConv6 dengan filter 5 x 5
9. Stage 8
- a. Tahap kedelapan melakukan *convolutional* dengan filter 3 x 3 dengan jumlah *channel* sebanyak 320.
 - b. Aktifkan fungsi *batch normalization*.
 - c. Aktifkan fungsi aktivasi ReLu.
 - d. Melakukan proses MBConv6 dengan filter 3 x 3.
10. Stage 9

- a. Tahap kesembilan melakukan *convolutional* dengan filter 3 x 3 dengan jumlah *channel* sebanyak 1280.
 - b. Aktifkan fungsi *batch normalization*.
 - c. Aktifkan fungsi aktivasi ReLu.
11. Hasil yang diperoleh dari *stage 1 – stage 9* akan melalui proses *pooling*, setelah itu di *flatten layer* akan menghasilkan *output* berupa *array* satu dimensi dan dengan fungsi aktivasi *softmax* akan melakukan klasifikasi jenis kelas dari citra yang diproses tersebut.

2.5. Confusion Matrix

Langkah terakhir yang harus dilakukan adalah mengevaluasi hasil prediksi yang dihasilkan oleh sistem yang dikembangkan. Metode evaluasi yang bisa digunakan adalah metode *Confusion matrix*. Berikut merupakan contoh *Confusion matrix* yang dapat dilihat pada **Gambar 4**.

	P	N
P	True Positive	False Positive
N	False Negative	True Negative

Gambar 4 *confusion matrix*

Untuk perhitungan *True Positive* (TP) dan *False Positive* (FP) bisa dilihat dari persamaan (4) dan (5).

$$tp\ rate = \frac{Positives\ correctly\ classified}{Total\ positives} \quad (4)$$

$$fp\ rate = \frac{Negatives\ correctly\ classified}{Total\ negatives} \quad (5)$$

Pada metode ini terdapat beberapa parameter untuk mengukur performanya antara lain sebagai berikut :

1. Akurasi
Digunakan untuk menghitung rasio kelas prediksi yang benar dengan jumlah total sampel dievaluasi yang bisa dilihat pada persamaan (6).

$$Akurasi = \frac{TP+FP}{TP+FP+TP+FN} \quad (6)$$

2. Presisi

Digunakan untuk menghitung pola positif yang diprediksi dengan benar oleh semua pola prediksi di kelas positif. Berikut adalah persamaan (7) dari presisi.

$$Presisi = \frac{TP}{TP+FP} \quad (7)$$

3. Recall

Digunakan untuk menghitung pecahan dari pola positif yang diklasifikasikan dengan benar yang bisa diperhatikan pada persamaan (8).

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

4. F1-Score

Menghitung rata-rata harmonik antara tingkat recall dan presisi yang bisa dilihat pada persamaan (9).

$$F - 1Score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (9)$$

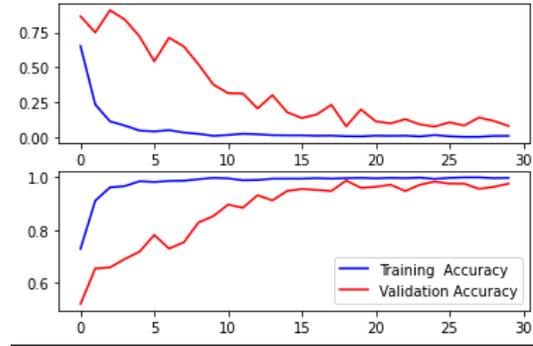
3. Hasil Percobaan

Pelatihan model *Convolutional Neural Network* dengan arsitektur *Efficientnet-B0* dilakukan dengan dataset yang diperoleh dari Kaggle.com. Dataset tersebut berjumlah 1392 yang terdiri dari 3 kelas yaitu foto *x-ray* paru Covid-19, foto *x-ray* paru *pneumonia* dan foto *x-ray* paru normal. Kemudian dataset akan dibagi menjadi data latih dan data uji secara manual dengan ratio 80 : 20. Parameter yang digunakan pada model ini adalah *Stochastic Gradient Decent* (SGD) sebagai *optimizer* nya dengan *learning rate* 0.001 dan momentum sebesar 0.9 beserta *Loss* yang digunakan pada model ini adalah *categorical crossentropy*. Pelatihan dilakukan dengan proses iterasi sebanyak 30 *epochs* dan ukuran *batch* 32.

Penelitian ini dilakukan dengan 2 skenario berbeda dimana skenario pertama data dilatih dengan menggunakan metode *preprocessing* CLAHE sedangkan untuk skenario kedua data dilatih tanpa menggunakan metode *preprocessing* CLAHE.

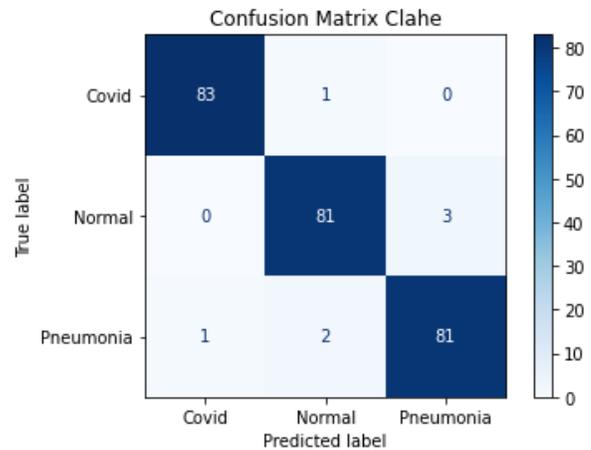
3.1. CLAHE

Pelatihan data yang dilakukan dengan skenario pertama dapat dilihat pada **Gambar 5** dibawah ini:



Gambar 5 Grafik loss dan accuracy data training dan validation menggunakan preprocessing CLAHE

Pada **Gambar 5** diatas sumbu x menunjukkan pada *epoch* ke berapa *loss* dan *validation* itu terjadi sedangkan sumbu y menunjukkan tingkat akurasi dari *loss* dan *validation*. Garis warna biru pada grafik bagian atas menandakan *loss train* dan bagian bawah menandakan *training accuracy*, sedangkan garis warna merah pada grafik bagian atas menandakan *loss validate* dan bagian bawah menandakan *validation accuracy*. Dari hasil yang telah diperoleh dengan menggunakan metode *preprocessing* CLAHE diatas dapat diketahui bahwa tingkat *accuracy training* tertinggi mencapai 1.000 pada *epoch* 27 dan 28 sedangkan untuk *validation accuracy* tertinggi mencapai 0.9881 pada *epoch* 19. Berikutnya melakukan proses pengujian dengan menggunakan metode *Confusion matrix* yang bisa dilihat pada **Gambar 6** dibawah ini.



Gambar 6 Confusion Matrix Skenario Menggunakan Metode *preprocessing* CLAHE

Dari hasil *Confusion matrix* pada **Gambar 6** diatas memperoleh hasil yang cukup memuaskan dimana pada kelas Covid, Pneumonia dan normal memperoleh

true positive diatas 80. Hasil *precision*, *recall*, *f-1 score*, dan akurasi pada **Gambar 7** dibawah ini.

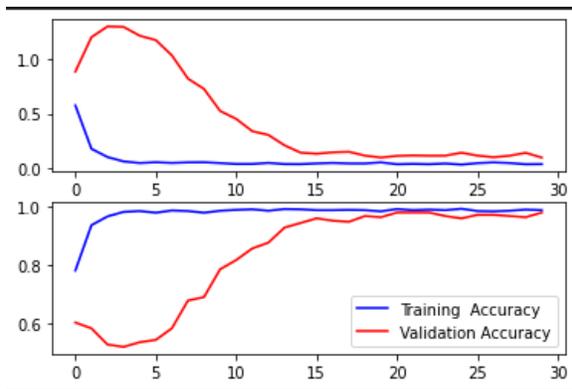
	Precision	Recall	f1-score
COVID-19	0.99	0.99	0.99
PNEUMONIA	0.96	0.96	0.96
NORMAL	0.96	0.96	0.96
ACCURACY	0.98		

Gambar 7 Hasil Skenario Pertama Data Training Diproses Dengan Metode Preprocessing Contrast Limited Adaptive Histogram Equalization (CLAHE)

Berdasarkan **Gambar 7** diatas nilai akurasi prediksi yang dilakukan oleh sistem pengenalan Covid-19 sebesar 98%.

3.2. NON CLAHE

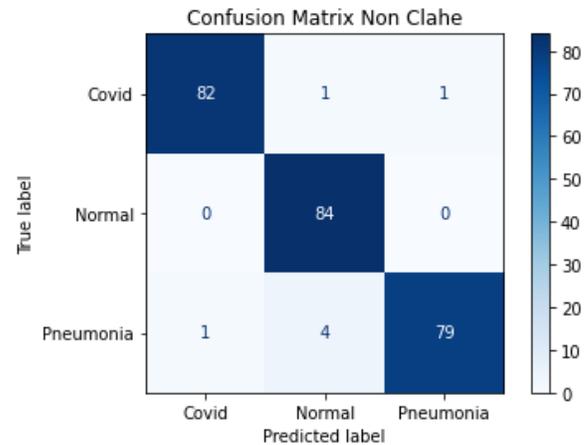
Pelatihan data yang dilakukan dengan skenario pertama dapat dilihat pada **Gambar 8** dibawah ini:



Gambar 8 Grafik loss dan accuracy data training dan validation tidak menggunakan preprocessing CLAHE

Pada **Gambar 8** tersebut sumbu x menunjukkan pada *epoch* ke berapa *loss* dan *validation* itu terjadi sedangkan sumbu y menunjukkan tingkat akurasi dari *loss* dan *validation*. Garis warna biru pada grafik bagian atas menandakan *loss train* dan bagian bawah menandakan *training accuracy*, sedangkan garis warna merah pada grafik bagian atas menandakan *loss validate* dan bagian bawah menandakan *validation accuracy*. Dari hasil yang telah diperoleh dengan tanpa menggunakan metode *preprocessing* CLAHE diatas dapat diketahui bahwa tingkat *accuracy training* tertinggi mencapai 0.9932 pada *epoch* 25 sedangkan untuk *validation accuracy* tertinggi mencapai 0.9802 pada *epoch* 21, 22, 23 dan 30. Berikutnya melakukan proses pengujian dengan menggunakan metode

confusion matrix yang bisa dilihat pada **Gambar 9** dibawah ini.



Gambar 9 Confusion Matrix Skenario Tidak Menggunakan Metode *preprocessing* CLAHE

Dari hasil *confusion matrix* pada **Gambar 9** diatas memperoleh hasil yang cukup memuaskan juga dimana pada kelas normal dan Covid memperoleh *true positive* diatas 80 sedangkan untuk *pneumonia* memperoleh *true positive* sebanyak 79. Hasil *precision*, *recall*, *f-1 score*, dan akurasi pada **Gambar 10** dibawah ini.

	Precision	Recall	f1-score
COVID-19	0.99	0.98	0.98
PNEUMONIA	0.99	0.94	0.96
NORMAL	0.94	1.00	0.97
ACCURACY	0.97		

Gambar 10 Hasil Skenario Kedua Data Training Tidak Diproses Dengan Metode Preprocessing Contrast Limited Adaptive Histogram Equalization (CLAHE)

Dari hasil data *training* **Gambar 10** diatas yang diuji ke data validasi dengan skenario kedua yaitu tidak menggunakan metode *preprocessing* Contrast Limited Adaptive Histogram Equalization (CLAHE) mendapatkan hasil yang hampir sama dengan skenario pertama dan untuk tingkat akurasi yang didapatkan menurun sebanya 0.01% namun tidak signifikan.

3.2. Perbandingan Arsitektur yang Digunakan Dengan Penelitian Sebelumnya

Dibawah ini adalah penelitian yang dilakukan sebelumnya yang memiliki topik hampir sama namun menggunakan arsitektur lainnya yang bisa dilihat pada **Tabel 2** dibawah ini

Tabel 2 Perbandingan dengan penelitian sebelumnya

Referensi	Metode	Akurasi
Yuli Sun Hariyani, 2020	Resnet-50	95%
Rizky Ardiawan, 2019	Resnet-18	94%
Lolita Bestari, 2021	InceptionNet V3	95%
Md Mamunur Rahaman, 2019	VGG19	89.3%
EfficientNet-B0		98%

Dari hasil penelitian yang telah dikumpulkan terbukti bahwa *Efficientnet-B0* mampu meningkatkan tingkat akurasi prediksi dibandingkan arsitektur lainnya. Walaupun dataset yang digunakan untuk pelatihan sistem tidak berasal dari sumber yang sama.

4. Kesimpulan

Kesimpulan yang bisa diambil setelah melakukan pengujian dengan menggunakan 2 skenario yang berbeda adalah sebagai berikut :

1. Penggunaan arsitektur *efficientNet-B0* pada sistem pengenalan Covid-19 ini menghasilkan tingkat akurasi sebesar 98%, dimana terbukti arsitektur *efficientNet-B0* mampu meningkatkan tingkat akurasi prediksi dibandingkan menggunakan arsitektur *resnet-18* pada penelitian sebelumnya.
2. Penggunaan metode *preprocessing* CLAHE dan tidak menggunakan *preprocessing* CLAHE terbukti tidak ada perubahan signifikan diantara kedua skenario tersebut dimana tingkat akurasi ketika menggunakan *preprocessing* CLAHE mencapai 98% sedangkan tidak menggunakan metode *preprocessing* CLAHE mencapai 97%. Perbedaan 1% diantara kedua skenario tersebut dianggap tidak ada perubahan karena tidak terlalu signifikan.

Adapun saran yang bisa diberikan kepada penelitian lebih lanjut dari sistem pengenalan Covid-19 ini adalah sebagai berikut :

1. Untuk saat ini sistem pengenalan Covid-19 ini masih belum bisa menentukan tingkat keparahan

dari pasien yang di klasifikasi terjangkit Covid-19 ataupun *pneumonia*, diharapkan bahwa sistem ini bisa dikembangkan agar bisa menentukan tingkat keparahan yang dialami oleh pasien dengan mengumpulkan *dataset* yang memiliki tingkat keparahan Covid-19 ataupun *pneumonia*.

2. Metode *preprocessing* CLAHE terbukti tidak terlalu mempengaruhi hasil prediksi yang dilakukan oleh sistem pengenalan Covid-19. Diharapkan penelitian berikutnya bisa menemukan metode *preprocessing* lainnya yang bisa membantu sistem untuk meningkatkan tingkat akurasi prediksinya.

REFERENSI

- [1] World Health Organization (WHO). Angka kasus dan kematian Covid-19, <https://covid19.who.int/>. Tanggal akses 24 Februari 2022
- [2] Susilo, Adityo. "Coronavirus Disease 2019; Tinjauan Literatur Terkini". Jurnal Penyakit Dalam Indonesia. Vol. 7, No.1. 2020
- [3] Budihardjo, Susan Natalja; dan Suryawan, I. Wayan. "Faktor-Faktor Resiko Kejadian pneumonia pada pasien pneumonia USIA 12-59 Bulan Di Rsud Wangaya". Intisari Sains Medis. Vol. 11, No. 1. 2020
- [4] Campos, Gabriel Fillipe; Mastelini, Saulo Martiello; Aguiar, Gabriel Jonas; Mantovani, Rafael Gomes; Melo, Leonimer Flavio; dan Barbon, Sylvio. "Machine learning hyperparameter selection for contrast limited adaptive histogram equalization". EURASIP Journal on Image and Video Processing. Vol. 2, No. 1. 2019.
- [5] Wardana, Bima Kusuma; Rachmawati, Ema; dan Wirayuda, Tjokorda Agung Budi. "Pengenalan Gestur Tangan Statis Menggunakan CNN dengan Arsitektur Efficient-Net B4". Jurnal Tugas Akhir Fakultas Informatika. Vol. 8, No. 2. 2021.