Diagnosis COVID-19 Berdasarkan Citra X-ray Paru-Paru Menggunakan Metode Convolutional Neural Network

Diagnose Of COVID-19 Based On X-ray Image Of The Lungs Using Convolutional Neural Network

Gilang Trisetya Indrawan[1)], Agung Nilogiri[2)], Habibatul Azizah Al Faruq[3)]

[1]Mahasiswa Fakultas Teknik, Universitas Muhammadiyah Jember
Email: gilangtrisetya123@gmail.com
[2]Dosen Fakultas Teknik, Universitas Muhammadiyah Jember
Email: agungnilogiri@unmuhjember.ac.id
[3]Dosen Fakultas Teknik, Universitas Muhammadiyah Jember
Email: habibatulazizah@unmuhjember.ac.id

Abstrak

12 Maret 2020, WHO mengumumkan kasus misterius pneumonia di Wuhan yang diberi nama COVID-19 sebagai pandemik. Salah satu cara untuk mendiagnosis COVID-19 adalah dengan menganalisis citra X-ray paru-paru. Ahli medis, menganalisis visual citra X-ray paru-paru harus secara teliti dan tepat, guna menentukan apakah pasien benar-benar terjangkit COVID-19. Namun menganalisis citra X-ray paru-paru membutuhkan proses yang cukup memakan waktu, maka dari itu dibutuhkan teknologi yang dapat dengan cepat mendiagnosis penyakit tersebut. Convolutional Neural Network (CNN) merupakan salah satu pengembangan algoritma *Multilayer Perceptron* (MLP) yang di dirancang untuk mengidentifikasi berbagai pola gambar dari berbagai sisi. Model CNN yang dibangun pada penelitian ini memiliki 40 convolution layer dengan fungsi aktivasi ReLU, *Batch Normalization*, dan 5 *max-pooling layer*. Layer klasifikasi model CNN yang dibangun menerapkan *global average pooling* yang menghasilkan 512 neuron yang langsung terhubung ke *layer* terakhir dengan fungsi *softmax*. Akurasi dari hasil model CNN yang dibangun berhasil mencapai keseluruhan akurasi 92,14% yang diuji menggunakan 318 data citra. Kesimpulan dari penelitian ini algoritma *Convolutional Neural Network* (CNN) yang dibangun relatif mampu mendiagnosis penyakit COVID-19 berdasarkan citra X-ray paru-paru dan tingkat keefektivitas model mendiagnosis penyakit COVID-19 lebih rendah dibanding mendiagnosis penyakit yang tidak terjangkit COVID-19.

Kata Kunci : *Convolutional Neural Network*, COVID-19, diagnosis,

Abstract

March 12, 2020, WHO announced a mysterious case of pneumonia in Wuhan which was named COVID-19 as a pandemic. One of way to diagnose COVID-19 is to analyze X-ray images of the lungs. Medical experts, analyze the visual X-ray image of the lungs carefully and precisely, to determine whether the patient really has COVID-19. However, analyzing an X-ray image of the lungs is a time-consuming process, therefore technology is needed that can quickly diagnose the disease. Convolutional Neural Network (CNN) is one of the developments of the Multilayer Perceptron (MLP) algorithm which is designed to identify various image patterns from various sides. The CNN model built in this study has 40 convolution layers with ReLU activation functions, Batch Normalization, and 5 max-pooling layers. The classification layer of the CNN model that is built applies global average pooling which produces 512 neurons which are directly connected to the last layer with the softmax function. The accuracy of the CNN results that were built managed to reach overall accuracy 92.14% which was tested using 318 image data. The conclusion of this study is that the Convolutional Neural Network (CNN) algorithm that was built is relatively

capable of diagnosing COVID-19 disease based on X-ray images of the lungs and the effectiveness of the model for diagnosing COVID-19 disease is lower than diagnosing diseases that are not infected with COVID-19.

Keywords : *Convolutional Neural Network*, COVID-19, diagnosis,

1. PENDAHULUAN

Desember 2019, kasus pneumonia misterius pertama kali dilaporkan di Wuhan, Provinsi Hubei China. Sumber penularan kasus pneumonia misterius ini masih tidak diketahui secara pasti, tetapi banyak yang mengaitkannya dengan pasar ikan di Wuhan. Tanggal 18 Desember hingga 29 Desember 2019, terdapat lima pasien yang dirawat dengan perawatan *Acute Respiratory Distress Syndrome* (ARDS). Lalu sejak 31 Desember 2019 kasus ini meningkat pesat, ditandai dengan dilaporkannya sebanyak 44 kasus. Kemudian penyakit ini telah menyebar di berbagai provinsi di China, Thailand, Jepang dan Korea Selatan. Pada 12 Maret 2020, WHO mengumumkan COVID-19 sebagai pandemik [1].

Penyakit yang disebabkan virus COVID-19 memiliki gejala yang sama mirip seperti pneumonia. Pneumonia biasa menyebabkan kantung udara pada saluran pernapasan di paru-paru mengalami peradangan dan di penuhi oleh cairan. Sedangkan pada COVID-19, umumnya menyerang saluran napas bagian atas yang pada akhirnya menyebar ke seluruh paru-paru [2].

Untuk mendiagnosis pasien terduga positif penyakit COVID-19, biasanya dilakukan uji klinis melalui pemeriksaan gejala fisik. Salah satu cara untuk mendiagnosis COVID-19 adalah dengan menganalisis citra X-ray paru-paru. Ahli medis, menganalisis visual citra X-ray paru-paru harus secara teliti dan tepat, guna menentukan apakah pasien benar-benar terjangkit COVID-19 atau tidak. Namun menganalisis citra X-ray paru-paru membutuhkan proses yang cukup memakan waktu. Maka dari itu dibutuhkan sebuah teknologi *Machine Learning* yang dapat mendiagnosis COVID-19 dengan cepat berdasarkan citra X-ray paru-paru dengan cepat dan akurat.

Deep learning adalah salah satu cabang dari *Machine Learning* yang akhir-akhir ini berkembang cukup pesat dalam beberapa tahun. Salah satu algoritma dari *deep learning* yang mempunyai keunggulan dalam mengklasifikasi citra adalah *Convolutional Neural Network* (CNN). CNN adalah salah satu algoritma hasil pengembangan *MultiLayer Perceptron* (MLP) yang terbukti mampu mengekstrak citra secara rinci. Tetapi CNN dalam pelatihan modelnya membutuhkan waktu relatif lebih lama dan membutuhkan kemampuan komputasi yang besar [3].

Pada penelitian yang dilakukan oleh [4], mengajukan suatu solusi yang disebut dengan *deep Bayes-SqueezeNet.* Solusi tersebut berhasil mendapatkan akurasi sebesar 98,26% dengan jumlah data latih sebanyak 1229 untuk tiap kelas dan data validasi sebanyak 154 untuk tiap. Namun data tersebut merupakan hasil dari proses *Offline Data Augmentation*. Jumlah data sebenarnya pada saat itu COVID-19 sebanyak 66, Normal 1349, dan Pneumonia 3895.

2. TINJAUAN PUSTAKA

A. Convolutional Neural Network

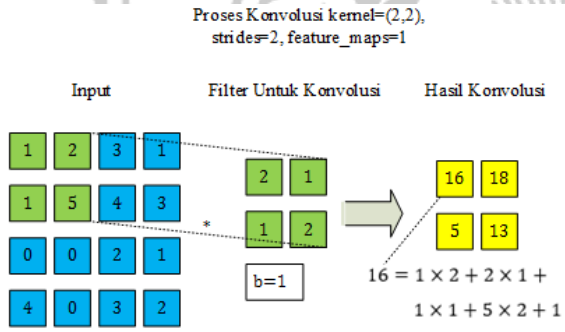
Menurut [5] *Convolutional Neural Network* (CNN) merupakan salah satu pengembangan algoritma *Multilayer Perceptron* (MLP) yang di dirancang untuk mengidentifikasi berbagai pola gambar dari berbagai sisi (translation invariance). [6] mengatakan bahwa nama "Convolutional Neural Network" mengindikasikan bahwa jaringan ini memanfaatkan operasi matematika yang bernama *convolution* atau konvolusi.

CNN dikembangkan pertama kali oleh [7] di Tokyo, Jepang, tetapi pada saat itu masih dinamakan "*Neocognition*". Kemudian dikembangkan kembali oleh [8] untuk pengenalan tulisan tangan dengan menerapkan *gradient-based learning.* Lalu Pada tahun 2012 [9] dengan menerapkan metode CNN

memenangkan kompetisi *ImageNet Large-Scale Visual Recognition Challenge* (ILSVRC). Dikarenakan prestasi tersebut Alex Krizhevsky membuktikan bahwa metode CNN terbukti mampu mengatasi pengenalan objek citra mengalahkan metode *machine learning* lainnya. Secara umum model CNN terdiri dari beberapa komponen yaitu *convolution layer*, *pooling layer*, *activation function*, dan *fully connected layer*.

B. Convolution Layer

Convolution layer merupakan layer yang pertama kali diproses ketika *input* citra memasuki arsitektur. *Convolution layer* mengaplikasikan proses operasi matematis secara berulang yang dinamakan *convolution* . Kemudian menurut [10] hasil dari proses *convolution* tersebut menghasilkan suatu informasi yang disebut dengan *feature maps*. Operasi konvolusi ditunjukkan pada Gambar 1.

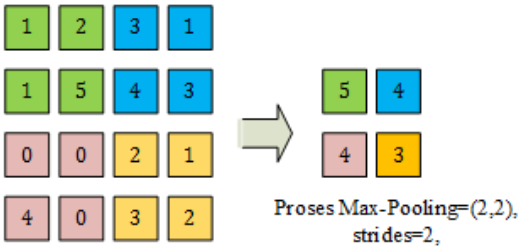


Gambar 1 Ilustrasi Proses Konvolusi

Sumber: Hasil Perhitungan

C. Pooling Layer

Setelah *convolution layer* biasanya ditambahkan *layer* untuk melakukan *pooling*. Menurut [10] apa yang dilakukan *pooling layer* adalah menyederhanakan *feature maps* yang dihasilkan oleh *convolution layer*. Proses penyederhanaan tersebut mengambil informasi yang penting dari *feature maps* dan menghilangkan detail yang tidak diperlukan, sehingga meringankan proses komputasi *layer* selanjutnya. Salah satu prosedur *pooling* yang sering digunakan pada CNN dinamakan *max-pooling*. Prosedur *max-pooling* mengambil nilai maksimal dari setiap grid untuk menyusun *feature maps* yang telah disederhanakan seperti pada Gambar 2

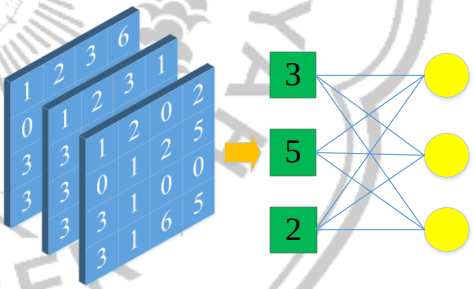


Gambar 2 Ilustrasi Proses *Max-Pooling*

Sumber: Hasil Perhitungan

D. Global Average Pooling

Global average pooling merupakan salah satu alternatif untuk menggantikan tradisional CNN *fully connected layer*, dimana setiap *feature maps* diambil nilai rata-ratanya dan memasukkan nilai rata-rata *feature maps* tersebut ke *output layer*. Salah satu keuntungan jika menerapkan proses *global average pooling*, menurut [11] adalah tidak ada proses memperbarui parameter sehingga *overfitting* dapat dihindari di layer ini.



Gambar 3 lustrasi *Global Average Pooling*

Sumber: Hasil Perhitungan

E. Softmax

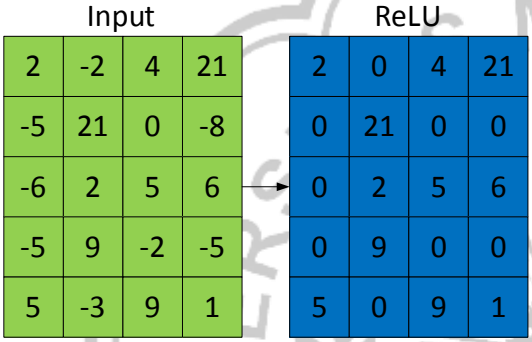
Menurut [12] *softmax* merupakan fungsi aktivasi yang digunakan untuk menghasilkan probabilitas dari suatu nilai vektor. Fungsi *softmax* menghasilkan nilai probabilitas tiap kelas dengan rentang nilai 0 sampai 1. Jika di jumlahkan seluruh nilai probabilitas tiap kelas akan memiliki nilai sama dengan 1. Fungsi *softmax* biasanya bertujuan untuk mengklasifikasi lebih dari 2 dan digunakan di

akhir jaringan . Persamaan dari fungsi *softmax* adalah sebagai berikut.

$$f(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} \quad \#(1)$$

F. ReLU

ReLU (Rectified Liner Unit) merupakan fungsi aktivasi yang mengaplikasikan fungsi max(0,x). Nilai *output* dari neuron dinyatakan sebagai 0 jika *input* nya adalah negatif. Jika *input*nya adalah positif, maka *output*nya adalah nilai *input* itu sendiri.



Gambar 4 Ilustrasi Proses ReLU

Sumber: Hasil Perhitungan

G. Batch Normalization

Batch Normalization pertama kali diperkenalkan oleh [13] dengan cara menormalisasi fungsi pada layer jaringan syaraf tiruan yang bertujuan meningkatkan kinerja model dan mempercepat proses pelatihan model. Model BN biasanya dilatih dengan nilai *learning* yang tinggi dimana menyebabkan model mencapai keadaan konvergen dan generalisasi yang baik. Lalu BN juga mengurangi penggunaan teknik *regularization* seperti *Dropout* [14] tanpa meningkatkan keadaan *overfitting*.

H. Catergorical Cross Entropy

Salah satu fungsi objektif untuk tugas *multi class classification*, yaitu *Categorical Cross Entropy* atau di kenal dengan sebutan *softmax loss*. Nilai *error* atau *loss* tersebut nantinya akan digunakan memperbarui parameter (bobot dan bias). Nilai *loss* mengindikasikan seberapa buruk model melakukan prediksi untuk 1 sampel. Jika prediksinya sempurna nilai *lossnya* 0, sebaliknya *lossnya* akan besar.

Persamaan dari fungsi *Categorical Cross Entropy* sebagai berikut.

$$CE = -\sum_{i} y_i \log(\hat{y}_i) \quad \#(2)$$

Dimana (y_i) merupakan nilai kebenaran tiap kelas i, dan $(\hat{y}_i)$ nilai probabilitas tiap kelas i.

I. Pelatihan CNN

Untuk melatih CNN agar mampu memprediksi dengan baik digunakanlah algoritma yang disebut dengan *Backpropagation*. Menurut [5] algoritma tersebut mempropagasikan nilai *error* dari *output* dan memperbarui parameter (bobot dan bias) berdasarkan nilai *error* dengan metode *gradient descent*, secara bertahap mulai dari *output layer* sampai *ke input layer*.

Gradient descent merupakan algoritma optimasi yang bertujuan untuk meminimalisir nilai *error* (fungsi objektif) $\nabla_\theta J(\theta)$ ke titik yang optimal dengan cara memperbarui parameter (bobot dan bias) yang berlawanan arah dengan *gradient* fungsi objektif [15]. Jika nilai *error* kecil model memberikan prediksi yang benar, sebaliknya model akan memberikan prediksi yang salah. Salah satu fungsi objektif, yaitu *Mean Square Error*, *Binary Cross Entropy*, dan *Categorical Cross Entropy*. Nilai *learning rate* pada *gradient descent* menentukan seberapa cepat mencapai titik optimal [15]. Beberapa varian algoritma gradient descent terdiri dari Stochastic Gradient Descent (SGD), Mini Batch Gradient Descent dan SGD dengan Momentum.

a. Stochastic gradient descent (SGD)

Stochastic gradient descent (SGD) merupakan salah satu varian algoritma *gradient descent* untuk melakukan pembaruan

parameter menggunakan tiap data kelas $x^{(i)}$ dan kelas $y^{(i)}$.

$$\theta = \theta - \eta \nabla_\theta J\left(\theta; x^{(i)}; y^{(i)}\right) \quad \#(3)$$

b. Mini-batch gradient descent

Varian *Gradient descent* ini memperbarui jaringan syaraf tiruan setiap n atau sampel dari *mini batch* yang ditentukan.

$$\theta = \theta - \eta \nabla_\theta J\left(\theta; x^{(i:i+n)}; y^{(i:i+n)}\right) \quad \#(4)$$

Varian ini mengurangi pembaruan parameter, di mana menyebabkan fungsi objektif relatif lebih stabil dan proses komputasi menjadi lebih efisien. Secara umum jumlah *mini-batch* berkisar antara 50 sampai 256, tetapi bisa saja berbeda untuk setiap kasus.

c. SGD dengan Momentum

SGD dengan Momentum merupakan salah satu peningkatan algoritma SGD Polos, di mana SGD polos pada gambar (a) ragu-ragu untuk mencapai titik optimal. Dengan menambahkan γ momentum memberikan sifat akselerasi pada proses parameter seperti pada gambar (b).

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta) \quad \#(5)$$
$$\theta = \theta - v_t$$

Ketika menggunakan momentum, bola menuruni bukit. Lalu bola mengumpulkan momentum saat menggelinding menuruni bukit, bola menjadi semakin cepat di jalan. hasilnya model cepat mencapai titik optimal dan jumlah iterasi berkurang.

(a) SGD tanpa momentum (b) SGD dengan momentum

Gambar 5 Perbandingan SGD

Sumber: (Ruder, 2017)

3. METODE PENELITIAN

Pada penelitian ini akan dirancang tahapan-tahapan yang meliputi *Preprocessing*, pembangunan model, pelatihan model dan pengujian model.

1. *Preprocessing* Data: Mengolah data asli sehingga memudahkan model CNN dalam proses pelatihan dan pengujian.
2. Pembangunan model CNN: Membangun model CNN untuk kasus klasifikasi citra X-ray paru-paru.
3. Pelatihan model CNN: Melatih model CNN yang telah dibuat terhadap *dataset COVIDx*.
4. Pengujian model CNN: Mengetahui kinerja model CNN dengan menerapkan metode *K-Fold Cross validation,* lalu mengujinya dengan data uji (unseen data).

A. PreProcessing Data

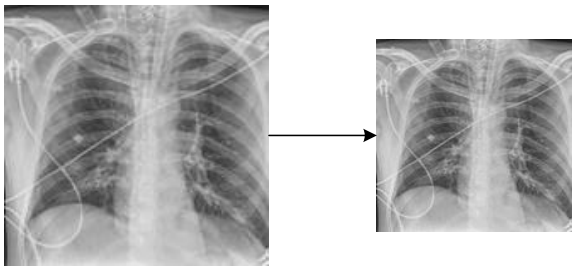
Tahapan *PreProcessing* data merupakan tahapan untuk mempersiapkan *datasets COVIDx* [16] sebelum dilakukan proses lain. Karena tidak semua data langsung di mengerti oleh model CNN.

a. Split Data

Datasets COVIDx yang sudah dikumpulkan lalu di pisah menjadi 2 kategori untuk data pelatihan model dan data uji (unseen data). Data pelatihan berjumlah 4977 dan data uji (unseen data) berjumlah 318.

b. Resize

Semua citra dari *datasets COVIDx* [16] mempunyai ukuran resolusi yang berbeda-beda dan model CNN hanya menerima *input* dengan ukuran resolusi yang sama, maka dari itu diterapkanlah proses *Resize*. Proses tersebut mengubah ukuran asli yang berbeda-beda menjadi ukuran yang di inginkan. Pada penelitian ini ukuran citra yang ditetapkan adalah (224×224), karena jika ukuran citra makin besar banyak informasi yang bisa didapatkan tetapi akan memakan banyak waktu untuk memprosesnya ketika proses pelatihan.

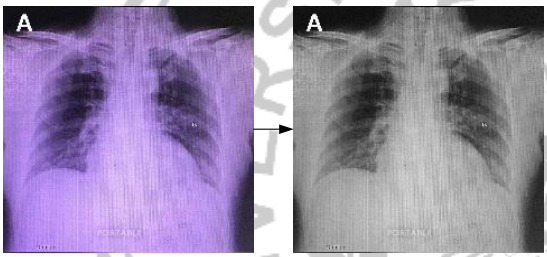


Gambar 6 Ilustrasi *Resize*

Sumber: Hasil Pengamatan

c. RGB Ke Grayscale

Semua citra dari *datasets COVIDx* [16], mempunyai kedalaman 3 warna (RGB) maka dari itu untuk menyederhanakan dan mempercepat proses komputasi ketika proses pelatihan. Maka dilakukan proses mengubah Citra RGB ke Grayscale.

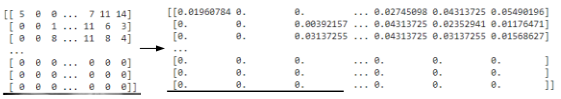


Gambar 7 Ilustrasi konversi RGB ke *Grayscale*

Sumber: Hasil Pengamatan

d. Normalization

Rentang *pixel* yang dimiliki citra berada di antara 0 sampai 255. Maka dari itu untuk mempercepat model CNN mencapai konvergen dan generalisasi, dilakukanlah proses *normalization*. Proses *normalization* yang dilakukan penelitian ini mengubah nilai *pixel* dengan rentang nilai 0 sampai 1, dengan cara membagi *pixel* dengan nilai 255, ketika nilai *pixelnya* berada di rentang nilai 0 dan 1 akan mempercepat proses pelatihan.



Gambar 8 Ilustrasi *Normalization*

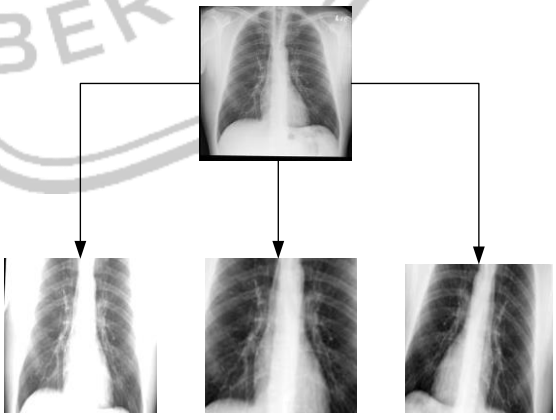
Sumber: Hasil Pengamatan

e. Data Augmentation

Data Augmentation (augmentasi data) merupakan salah satu teknik untuk menambah keberagaman data citra dengan melakukan transformasi dasar seperti, rotasi , refleksi , kecerahan dinaikkan dll. Menurut [17] dengan melakukan teknik ini model dapat mengatasi masalah *overfitting* dan meningkatkan akurasi model CNN. Augmentasi data dapat dilakukan dengan 2 cara yaitu, *Offline Augmentation* dan *RealTime Augmentation*. *Offline Augmentation* berarti menambah jumlah citra sebelum proses pelatihan dimulai, sedangkan *Realtime Augmentation* menambah keberagaman citra ketika pelatihan dimulai. Jadi setiap *epoch*, model menerima citra dengan transformasi yang berbeda-beda. Pada penelitian ini menerapkan augmentasi data secara *RealTime*.

Konfigurasi augmentasi data penelitian ini berupa:

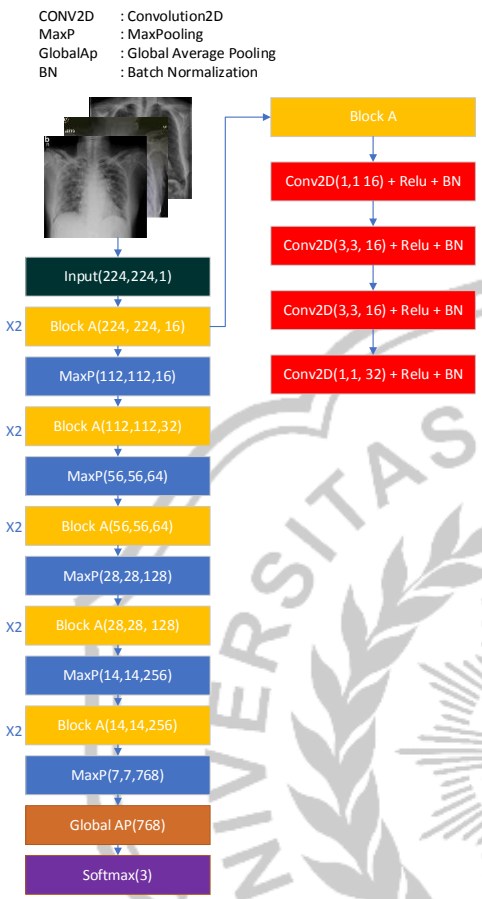
1. *Brightness Range* = [0.8-1.2], menaikkan dan menurunkan kecerahan secara acak.
2. *Rotation Range* = 15, memutar citra sampai maksimal sudut 15 derajat secara acak.
3. *Horizontal Flip* = *True*, membalik citra secara horizontal.
4. *Zoom Range* = [0.5-1.0], memperbesar citra secara acak, nilai 0.5 berarti memperbesar citra hingga 50% sedangkan 1.0 ukuran citra normal.



Gambar 9 Ilustrasi Data Augmentation

Sumber: Hasil Pengamatan

B. Pembangunan Model



Gambar 10 Arsitektur CNN yang akan dibangun

Sumber: Hasil Desain

Tahapan pembangunan model CNN merupakan tahapan membangun model *machine learning* CNN yang terdiri dari beberapa komponen seperti *convolution layer*, *max-pooling layer*, *activation function ReLU*, *Softmax*, *batch normalization layer* dan *global average pooling layer*.

Pada Gambar 10, Block A merupakan modifikasi desain *bottleneck layer* [18]. Block A mempunyai 4 *convolution layer* yang mempunyai ukuran *kernel* (1×1), (3×3), (3×3) dan (1×1). Dimana layer (1×1) bertanggung jawab mereduksi dan menambah (mengembalikan) dimensi [18].

Semua *convolution layer* menerapkan konfigurasi *zero-padding*, dan *stride* 1. *Zero-padding* merupakan konfigurasi agar *output* dari hasil *convolution* mempunyai panjang dan lebar yang sama seperti *input* nya. Diakhir *convolution layer*, menerapkan *ReLU* lalu dinormalisasi dengan *Batch-Normalization*. Jumlah *filters* yang dihasilkan *convolution layer* berjumlah 16, tetapi secara bertahap akan dikalikan 2 setelah melewati *Max-Pooling layer*. *Max-Pooling layer* menerapkan konfigurasi *poolsize* (2,2) dan *stride* 2. Kemudian di akhir jaringan *Max-Pooling layer*, diterapkan *Global Average Pooling Layer* yang langsung terhubung ke 3 neuron dengan fungsi aktivasi *softmax*.

C. Pelatihan Model

Tahapan pelatihan merupakan tahapan model *machine learning* mengenali suatu kelas dengan cara mempelajari ciri atau karakteristik terhadap data yang di latih, sehingga model tersebut memiliki kemampuan prediktif terhadap data yang serupa.

Pada penelitian ini digunakanlah metode SGD + Momentum untuk proses pelatihan model dengan konfigurasi *learning rate* 0.003 dan momentum 0.9. Menerapkan *mini batch size* dengan nilai 8, karena semakin besar nilai *mini batch size* menyebabkan proses pelatihan menjadi cepat tetapi pada akhirnya akurasi model tidak *sebagus mini batch size* dengan nilai yang lebih kecil [19] dan batasan *epoch* berjumlah 600.

D. Pengujian Model

Metode pengujian yang digunakan untuk mengukur kinerja model CNN pada penelitian ini adalah *K-Fold Cross Validation*. *K-Fold Cross Validation* merupakan salah satu metode untuk mengukur suatu kinerja model algoritma dengan menghitung varians kinerja dari K model. Kemudian nilai rata-rata kinerja model adalah sebuah perkiraan kinerja model terhadap data uji [5]. Metode *K-Fold Cross Validation* penerapannya dengan membagi-bagi data yang dilatih menjadi beberapa K

bagian. Kemudian salah satu bagian tersebut dijadikan data validasi dan sisanya dijadikan sebagai data latih. Untuk distribusi kelas tiap bagian haruslah sama (*Stratified Sampling*).

Pada penelitian ini nilai K yang di gunakan = 7. Dengan total jumlah data citra sebanyak 4977, jadi setiap blok memiliki data citra sebanyak 711. lalu dikategorikan menjadi 3 kelas yaitu COVID-19, Normal, dan Pneumonia.

Setelah melakukan *K-Fold Cross Validation*, langkah selanjutnya mengujinya dengan data uji (unseen data).

4. HASIL DAN PEMBAHASAN

A. Hasil Uji Coba Model *K-Fold Cross Validation*

Pada pengujian *K-FOLD Cross Validation*, *fold* dibagi menjadi 7. Setiap 1 *fold* memiliki data latih sebanyak 4266 dan data validasi sebanyak 711. Hasil pengujian *K-Fold Cross Validation* dapat dilihat pada Tabel 1.

Tabel 1 Pengujian *K-Fold Cross Validation*

K-FOLD	**Keseluruhan Akurasi (%)**
FOLD 1	92,12
FOLD 2	92,55
FOLD 3	93,39
FOLD 4	92,55
FOLD 5	92,12
FOLD 6	92,12
FOLD 7	92,26
Rata-Rata	92,44

Sumber: Hasil Penelitian

Berdasarkan Tabel 1, *fold* 3 mendapatkan nilai akurasi tertinggi sebesar 93,39% dan yang paling rendah ialah *fold* 1, *fold* 5, dan *fold* 6. Rata-rata keseluruhan akurasi dari ketujuh fold tersebut mencapai 92,44%

B. Hasil Uji Coba Model Terhadap *Unseen Data*

Pengujian kedua digunakanlah data uji (unseen data). Kemudian mengujinya dengan model dari *K-Fold Cross Validation*. Hasil Pengujian *K-FOLD Cross Validation* terhadap *Unseen Data* dapat dilihat pada Tabel 2

Tabel 2 Pengujian Terhadap *Unseen Data*

K-FOLD	**Keseluruhan Akurasi (%)**
FOLD 1	88,69
FOLD 2	89,62
FOLD 3	90,25
FOLD 4	89,31
FOLD 5	89,62
FOLD 6	92,14
FOLD 7	88,37

Sumber: Hasil Penelitian

Berdasarkan Tabel 2, *fold* 6 mendapatkan nilai akurasi paling tinggi sebesar 92,14% dan *fold* 7 mendapatkan nilai akurasi terendah sebesar 88,37%. *Fold* 3 pada pengujian *K-FOLD Cross Validation* memiliki akurasi yang paling tinggi tetapi ketika di uji menggunakan data uji (unseen data), akurasi tidak sebagus *fold* 6, karena *fold* 3 mengalami keadaan yang disebut *overfitting* ketika proses pelatihan.

Tabel 3 *Confusion Matrix fold 6*

		Actual		
		COVID-19	Normal	Pneumonia
Predicted	COVID-19	96	0	2
	Normal	3	102	9
	Pneumonia	7	4	95

Sumber: Hasil Penelitian

Model *fold* 6 jika dilihat kinerjanya secara rinci, dari sisi *class* COVID-19, 96 data COVID-19 diprediksi benar (TP), 2 data diprediksi sebagai (FP), dan 10 data diprediksi sebagai (FN). Dari sisi *class* normal, 102 data normal diprediksi benar (TP), 12 data sebagai (FP), dan 4 data diprediksi sebagai Pneumonia (FN). Dari sisi *class* pneumonia, 95 data pneumonia diprediksi benar (TP), 11 data diprediksi sebagai (FP), dan 11 data diprediksi sebagai (FN).

Tabel 4 Akurasi, Sensitivitas, dan Spesifisitas model *fold* 6

	Akurasi (%)	Sensitivitas (%)	Spesifisitas (%)
COVID-19	96,23	90,57	99,06
Normal	94,97	96,23	94,34
Pneumonia	93,08	89,62	94,81

Sumber: Hasil Penelitian

Berdasarkan Tabel 4 *Class* COVID-19 mencapai nilai akurasi tertinggi sebesar 96,23%, sensitivitas 90,57%, dan spesifisitas 99,06%. Untuk *class* normal mencapai nilai akurasi sebesar 94,97%, sensitivitas 96,23%, dan spesifisitas 94,34%. dan Untuk *class* pneumonia mencapai nilai akurasi sebesar 93,08%, sensitivitas 89,62%, dan spesifisitas 94,81%.

5. KESIMPULAN

Berdasarkan hasil analis selama perancangan, implementasi dan pengujian. Diambil beberapa kesimpulan sebagai berikut.

1. Berdasarkan uji coba model fold 6 menggunakan data testing (unseen data) tingkat akurasi untuk mendiagnosis COVID-19 mencapai akurasi sebesar 96,23%, dengan nilai tersebut, model relatif mampu mendiagnosis COVID-19 berdasarkan citra x-ray paru-paru.
2. Berdasarkan uji coba model fold 6 menggunakan data testing (unseen data) memiliki keseluruhan akurasi 92,14%. Class COVID-19 mencapai akurasi sebesar 96,23%, sensitivitas 90,57%, dan spesifisitas 99,06%. sensitivitas lebih kecil dibanding spesifisitas artinya tingkat keefektivitas model mendiagnosis pasien terjangkit COVID-19 lebih rendah dibandingkan keefektivitas model mendiagnosis pasien tidak terjangkit COVID-19.

6. SARAN

Penelitian mengenai diagnosis COVID-19 berdasarkan citra x-ray paru-paru menggunakan metode CNN masih memiliki kekurangan, sehingga perlu adanya perubahan di kemudian hari. Beberapa hal yang disarankan oleh peneliti selanjutnya ke depan ialah:

1. Mengumpulkan lebih banyak data sehingga lebih banyak ragam pola yang dikenali.
2. Menerapkan modul tambahan seperti *shortcut connection*, karena berdasarkan penelitian [18] akurasi model bertambah ketika jumlah layernya ditambahkan hingga 152.

DAFTAR PUSTAKA

[1] Susilo, A. *dkk.*, Coronavirus Disease 2019: Tinjauan Literatur Terkini, *Jurnal Penyakit Dalam Indonesia*, vol. 7, no. 1, hal. 45–67, maret 2020.

[2] Handayani, V.V., Gejalanya Mirip, Ini Bedanya Pneumonia dengan COVID-19, *Gejalanya Mirip, Ini Bedanya Pneumonia dengan COVID-19*, 09-Apr-2020. [Daring]. Tersedia di: https://www.halodoc.com/artikel/gejalanya-mirip-ini-bedanya-pneumonia-dengan-covid-19. [Diakses: 21-Mei-2021].

[3] Lai, Y., A Comparison of Traditional Machine Learning and Deep Learning in Image Recognition, *Journal of Physics: Conference Series*, vol. 1314, hal. 012148, oktober 2019.

[4] Ucar, F. dan Korkmaz, D., COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images, *Medical Hypotheses*, vol. 140, hal. 109761, juli 2020.

[5] Putra, J.W.G., *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*. 2020.

[6] Goodfellow, I., Bengio, Y. dan Courville, A., *Deep Learning*. MIT Press, 2016.

[7] Fukushima, K., Neocognitron: A self-organizing neural network model for a

mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics*, vol. 36, no. 4, hal. 193–202, april 1980.

[8] Lecun, Y., Bottou, L., Bengio, Y. dan Haffner, P., Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, vol. 86, no. 11, hal. 2278–2324, 1998.

[9] Krizhevsky, A., Sutskever, I. dan Hinton, G.E., ImageNet Classification with Deep Convolutional Neural Networks, dalam: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, Red Hook, NY, USA, hal. 1097–1105, 2012.

[10] Nielsen, M., *Neural Networks and Deep Learning*. Determination Press, 2015.

[11] Lin, M., Chen, Q. dan Yan, S., *Network In Network*. 2014.

[12] Nwankpa, C., Ijomah, W., Gachagan, A. dan Marshall, S., *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 2018.

[13] Ioffe, S. dan Szegedy, C., *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015.

[14] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. dan Salakhutdinov, R., Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *J. Mach. Learn. Res.*, vol. 15, no. 1, hal. 1929–1958, januari 2014.

[15] Ruder, S., *An overview of gradient descent optimization algorithms*. 2017.

[16] Wang, L., Lin, Z.Q. dan Wong, A., COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images, *Scientific Reports*, vol. 10, no. 1, hal. 19549, november 2020.

[17] Perez, L. dan Wang, J., *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. 2017.

[18] He, K., Zhang, X., Ren, S. dan Sun, J., *Deep Residual Learning for Image Recognition*. 2015.

[19] Masters, D. dan Luschi, C., *Revisiting Small Batch Training for Deep Neural Networks*. 2018.