

Deep Learning Untuk Deteksi Covid-19, Pneumonia, Dan Tuberculosis Pada Citra Rontgen Dada Menggunakan CNN Dengan Arsitektur Alexnet

Deep Learning For The Detection Of Covid-19, Pneumonia, And Tuberculosis In Chest X-Ray Imaging Using Cnn With Alexnet Architecture

1st Sendhy Septhyan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
sendhysepthyan@telkomuniversity
.ac.id

2nd Rita Magdalena
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ritamagdalena@telkomuniversity.a
c.id

3rd Nor Kumalasari Caecar Pratiwi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
caecarnkcp@telkomuniversity.ac.i
d

Abstrak—Seperti yang kita ketahui *Pneumonia* pada dasarnya disebabkan oleh virus dan bakteri salah satunya bernama *Mycobacterium*, yaitu bakteri (penyebab TB paru). Menurut data Organisasi Kesehatan Dunia atau juga umum dikenal sebagai *World Health Organization*, sepuluh penyebab kematian terbesar di dunia salah satunya yaitu TB atau *tuberculosis*. COVID-19 adalah varian yang dapat memicu penyakit *pneumonia*. Gejala yang dihadapi hampir sama persis seperti gejala *pneumonia* dan *tuberculosis* seperti demam, sesak nafas dan juga serangan kronik. Pendeteksian COVID19, *pneumonia*, *tuberculosis* dilakukan dengan pengamatan manual oleh dokter ahli, sehingga rentan human error. Oleh karena itu, telah dikembangkan beberapa solusi untuk mengatasi permasalahan diagnosa penyakit ini, di antaranya dengan menggunakan metode pengolahan citra rontgen dada juga bahasa pemrograman dan menghasilkan perangkat lunak yang valid untuk digunakan diagnosa dan mengklasifikasikan ketiga penyakit tersebut, salah satunya yaitu menggunakan *Convolutional Neural Network* Masukan citra dataset yang diambil dari Kaggle dengan nama *Chest X-Ray (Pneumonia, Covid-19, Tuberculosis)* dengan total dataset yang digunakan yaitu 2304 dimana 75% citra merupakan data *train* dan 25% citra data *test*. Hasil terbaik yang didapatkan untuk setiap skenario pengujian yaitu menggunakan *Optimizer Adam*, *Resize 64x64 pixel*, *Learning Rate 0,0001*, *Epoch 35*,

Batch Size 16 dengan hasil *Accuracy 95%* dan *Loss 0,161*

Kata Kunci— *Convolutional Neural Network (CNN)*, *alexnet*, *COVID-19*, *pneumonia*, *tuberculosis*.

Abstract—As we know *Pneumonia* is basically caused by viruses and bacteria, one of which is called *Mycobacterium*, namely bacteria (the cause of pulmonary TB). According to data from the World Health Organization or also commonly known as the World Health Organization, the top ten causes of death in the world are TB or tuberculosis. COVID-19 is a variant that can trigger pneumonia. The symptoms faced are almost exactly the same as the symptoms of pneumonia and tuberculosis such as fever, shortness of breath and chronic attacks. The detection of COVID19, pneumonia, tuberculosis is done by manual observation by expert doctors, so it is prone to human error. Therefore, several solutions have been developed to overcome the problem of diagnosing this disease, including using chest X-ray image processing methods as well as programming languages and producing valid software to be used to diagnose and classify the three diseases, one of which is using the *Convolutional Neural Network*. Input the dataset image taken from Kaggle with the name *Chest X-Ray (Pneumonia, Covid-19, Tuberculosis)* with a total dataset used of 2304 where 75% of the images are train data and 25% are test data images. The best results obtained for each test scenario are using *Adam's Optimizer*, *Resize 64x64*

pixels, Learning Rate 0.0001, Epoch 35, Batch Size 16 with 95% Accuracy and 0.161 Loss.

Keywords— *Convolutional Neural Network (CNN), alexnet, COVID-19, pneumonia, tuberculosis.*

I. PENDAHULUAN

COVID19, *pneumonia, tuberculosis* adalah keluarga virus yang paling berbahaya. Data yang disampaikan *World Health Organization (WHO)*, *tuberculosis* adalah salah satu dari 10 penyebab kematian teratas di seluruh dunia [1]. Munculnya COVID-19 berdampak pada semua aspek, jutaan karyawan kehilangan pekerjaan, kehilangan pendapatan di sektor jasa, turunnya devisa pariwisata, meningkatnya inflasi, dan berujung pada resiko memburuknya kondisi ekonomi. Seperti yang kita ketahui, tantangan terbesar dalam menghadapi COVID-19 yaitu perlu mengidentifikasi diagnostik cepat dan handal, dimana berfungsi untuk alternatif untuk *Reverse-Transcriptase Testing Polymerase Chain Reaction (RT-PCR)* [30].

Proses deteksi adanya COVID19, *pneumonia, tuberculosis* dilakukan dengan pengamatan visual oleh dokter ahli, sehingga rentan terjadinya human error. Kecerdasan buatan atau bisa kita sebut sebagai AI kini telah berkembang pesat untuk membantu kehidupan manusia dan telah diterapkan pada banyak aspek kehidupan, model berbasis AI pada klasifikasi gambar sering digunakan dalam teknologi Computer Aided Diagnosis (CAD) untuk membantu dokter atau para ahli medis mendiagnosis pasien dengan akurat. Dengan teknik RT-PCR, citra rontgen dada (CXR) dapat digunakan untuk melakukan deteksi dini kondisi pasien dengan akurasi hingga 80%. Dengan memanfaatkan model *Deep Learning, Convolutional Neural Network (CNN)*, citra CXR kemudian diklasifikasikan ke dalam kategori tertentu, sehingga model *deep learning* ini sangat menarik untuk dikembangkan [31][2].

Pada penelitian sebelumnya yaitu "*COVID-19 Diagnosis from Chest X-ray Images Using Deep Learning Approach*", deteksi COVID-19, *pneumonia, tuberculosis* dengan menggunakan arsitektur CNN yang terdiri dari empat Convolutional Block. Skenario dua kelas yaitu normal dan COVID-19 dengan total dataset 2159 citra mendapatkan akurasi 99,7%, skenario tiga kelas yaitu normal, COVID-19, dan *pneumonia* dengan total dataset 6432 citra mendapatkan akurasi 95,02%, dan skenario empat kelas yaitu normal, COVID-19, *pneumonia*, dan *tuberculosis* dengan total dataset 6587 mendapatkan akurasi 94,53% [3]. Pada penelitian dengan judul "*Comperative Experiment of CNN Models Based on Pneumonia X-ray Image Detection*", penelitian ini menggunakan dataset berisi 5216 gambar rontgen dada data latih (3875 *pneumonia*, 1341 normal) dan 624 gambar dalam data uji (390

pneumonia dan 234 normal) dengan tiga metode yaitu classic CNN, VGG16, dan ResNet50 dengan hasil accuracy pada masing-masing metode yaitu classic CNN 73,88%, VGG16 80%, dan ResNet50 80% [4].

Untuk penelitian dengan judul "*Deep Learning for Mycoplasma Pneumonia Discrimination from Pneumonias like COVID-19*", penelitian ini menggunakan data latih 1180 dan data uji 150 citra dengan tujuh metode yaitu ResNet-18, ResNet-50, MobileNet-V2, VGG, SqueezeNet, AlexNet, DenseNet-121, pada penelitian ini menggunakan dua pembeda yaitu membedakan *pneumonia mycoplasma* dari *pneumonia typical pneumonia virus*, dan membedakan *pneumonia mycoplasma* dari COVID-19. Hasil accuracy dalam penelitian membedakan *pneumonia mycoplasma* dari *pneumonia typical pneumonia virus* dari masing-masing metode yaitu ResNet-50 76%, ResNet-18 74%, MobileNet-V2 72%, VGG 73%, SqueezeNet 76%, AlexNet 69%, DenseNet-121 67%, dan hasil accuracy dalam penelitian membedakan *pneumonia mycoplasma* dari COVID-19 dari masing-masing metode yaitu ResNet-50 89%, ResNet-18 89%, MobileNet-V2 88%, VGG 88%, SqueezeNet 88%, AlexNet 90%, DenseNet-121 85% [5].

Implementasi Deep Learning menggunakan metode CNN memiliki keunggulan dalam pengenalan gambar visual yang lebih baik. Berdasarkan keunggulan tersebut, pada penelitian ini dibuat suatu sistem deteksi COVID-19, *pneumonia*, dan *tuberculosis* dengan menggunakan dataset rontgen dada yang berasal dari kaggle [6], sistem akan dirancang menggunakan metode CNN dengan arsitektur alexnet. Arsitektur ini dipilih karena memiliki kinerja yang baik dan dapat memperkecil tingkat kesalahan dari pemenang yang lainnya, alexnet dapat menurunkan tingkat kesalahan dari 26% hingga 15,3% hanya dengan menggunakan layer yang sedikit. Selain jumlah layer yang sedikit, alexnet juga menggunakan relu yang dapat membuat proses pelatihan lebih cepat dari model CNN biasa [19].

II. KAJIAN TEORI

A. Coronavirus

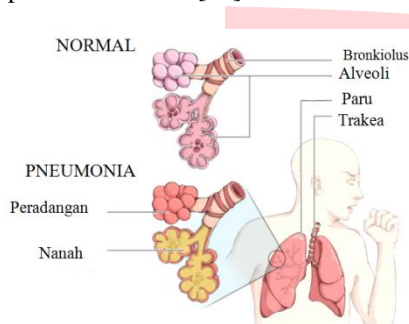
COVID19 adalah virus jenis baru yang ditemukan manusia sejak muncul di Wuhan, China pada Desember 2019, dan diberi nama *Severe A* disebut dengan *Coronavirus Disease-2019 (COVID-19)* (*World Health Organization/ WHO*, 2020) [8].

Dipercaya COVID-19 dapat ditularkan dari hewan ke manusia, dan juga antar manusia, cara penularannya antara lain tidak sengaja menghirup percikan ludah (droplet) yang keluar saat penderita COVID-19 bersin atau batuk, Memegang mulut, hidung atau mata tanpa mencuci tangan terlebih dulu, setelah menyentuh benda yang

terkena droplet penderita COVID-19, misalnya uang atau gagang pintu, kontak jarak dekat (kurang dari 2 meter) dengan penderita COVID-19 tanpa mengenakan masker [9].

B. Pneumonia

Pneumonia adalah peradangan paru-paru yang disebabkan oleh infeksi. Pneumonia juga dikenal dengan istilah paru-paru basah. Pada kondisi ini, infeksi menyebabkan peradangan pada kantong-kantong udara (alveoli) di salah satu atau kedua paru-paru. Akibatnya, alveoli bisa dipenuhi dengan cairan atau nanah sehingga menyebabkan penderitanya sulit bernapas [10]. Untuk melihat kondisi saat terjangkit *pneumonia* atau tidak, dapat dilihat pada Gambar 2.1 [10]



Gambar 2.1 Pneumonia dan Normal

C. Tuberculosis

Tuberculosis (TBC) yang juga dikenal dengan TB adalah penyakit paru-paru akibat kuman *Mycobacterium tuberculosis*. TBC akan menimbulkan gejala berupa batuk yang berlangsung lama (lebih dari tiga minggu), biasanya berdahak, dan terkadang mengeluarkan darah [11].

Kuman TBC tidak hanya menyerang paru-paru, tetapi juga bisa menyerang tulang, usus, atau kelenjar. Penyakit ini ditularkan dari percikan ludah yang keluar dari penderita TBC, ketika berbicara, batuk, atau bersin. Penyakit ini lebih rentan terkena pada seseorang yang kekebalan tubuhnya rendah, misalkan penderita HIV [12].

D. Teori Dasar Citra

Sebuah citra (gambar/image) sebagai fungsi dua dimensi kontinu dari intensitas cahaya (x, y). dimana variabel x dan y mewakili suatu koordinat, dan f pada setiap titik (x, y) menyatakan intensitas atau derajat keabuan (brightness/grayscale) [13].

1. Citra Digital

Citra digital merupakan citra berkelanjutan yang diganti ke dalam wujud diskrit, baik koordinat ataupun intensitas cahaya dari citra tersebut. Dengan kata lain, citra digital terbuat dengan metode mencuplik sesuatu citra kontinu dengan jarak seragam. Dengan demikian citra digital bisa disajikan selaku matriks berukuran $M \times N$, dengan M

merupakan tinggi serta N menyatakan lebar dari citra. Intensitas sinar citra dapat dinyatakan dari kumpulan elemen dengan nilai tertentu yang terdapat dalam Matriks tersebut. Pixel merupakan nama yang digunakan untuk elemen tersebut [13].

2. Citra Grayscale

Grayscale adalah data matriks yang nilainya mewakili bayangan abu-abu. Apabila citra digital memiliki skala keabuan 8-bit atau 16-bit, maka nilai intensitas dari masing-masing skala-nya adalah (0-255) atau (0-65535) dimana hitam untuk nilai minimal dan putih untuk nilai maksimal [26].

3. Citra RGB

Model warna RGB didasarkan pada mata manusia, yang memahami cahaya dan mengubahnya menjadi gelombang di otak. RGB merupakan pemodelan warna yang memungkinkan untuk menciptakan banyak warna melalui warna primer. Disini artinya *R (Red)*, *G (Green)* dan *B (Blue)*, yang apabila dikombinasikan dengan suatu nilai takaran tertentu dapat membentuk suatu susunan warna yang banyak. Setiap warna dasar, misalnya hijau, memiliki rentang nilai. Layar komputer biasanya memiliki nilai rentang minimum 0 dan nilai rentang maksimum 255. Pemilihan pada skala 256 didasarkan pada tampilan angka biner 8 digit yang digunakan pada komputer [14].

Dengan cara ini, maka didapatkan warna campuran sebanyak $256 \times 256 \times 256 = 16.777.216$ jenis warna. Jenis warna dapat dianggap sebagai vektor dalam ruang tiga dimensi yang biasa digunakan dalam matematika, yang berkoordinat tiga angka yaitu komponen x, y, z . Komponen *R (red)*, *G (green)*, *B (blue)* tersebut menggantikan sebuah vektor dituliskan sebagai $r = (x, y, z)$. Jadi, sebuah warna dapat dituliskan sebagai berikut: warna = RGB (70, 70, 255). Putih = RGB (255, 255, 255), sedangkan untuk hitam = RGB (0, 0, 0). Representasi warna RGB seperti ditunjukkan dalam Gambar 2.2 [15]

E. Deep Learning

Definisi Deep Learning yang biasa berlaku untuk banyak orang agak susah didefinisikan sebab sudah berganti wujud di beberapa tahun terakhir ini. Salah satu definisi yang simpel yaitu: "Deep Learning merupakan tiap jaringan saraf dengan lebih dari 2 susunan" [16].

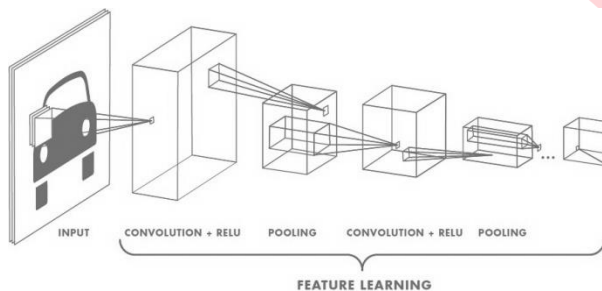
Selain itu, para ahli sedang membangun model yang lebih andal yang sebenarnya memperbarui konsep dan ide lama, seperti halnya jaringan saraf kompleks (CNN) hingga jaringan Kapsul atau Capsnet, hingga Pembelajaran Penguatan Jaringan Dalam dan Jaringan Dalam (DRL). Banyak dari model *deep learning* baru ini telah terbukti sangat efektif dalam memproses kumpulan data yang sangat besar [16]. Oleh karena itu, dalam mengelola big data, *deep learning*

merupakan solusi pembaharuan dari *machine learning*.

F. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah jenis jaringan saraf yang biasa digunakan dalam data gambar. CNN dapat digunakan untuk mendeteksi dan mengenali objek pada citra. Bisa dikatakan bahwasannya persepsi visual manusia yaitu bagaimana cara kerja otak manusia mendeteksi dan mengenali suatu objek adalah hal yang menginspirasi terciptanya CNN [17].

Arsitektur CNN sangat sederhana yaitu terdiri atas satu layer input, output layer, dan hidden layer [18]. Lapisan tersembunyi biasanya mencakup kedalam tiga layer utama yaitu lapisan *convolutional*, lapisan *pooling*, dan lapisan *fully connected* [16]. Oleh karena itu, CNN memiliki keuntungan utama dari pada pendahulunya yaitu, dapat mengotomatisasi deteksi fitur penting dengan mempelajari, menerjemahkan serta mengekstrak data berupa gambar. Berikut gambar arsitektur CNN pada Gambar 2.3 [18].



Gambar 2. 2 Arsitektur CNN

1. Convolutional Layer

Convolutional Layer adalah layer yang ada pada *Feature Extraction Layer* yang mana merupakan proses utama pada CNN. Pada layer ini terdiri dari neuron yang berdampingan membentuk suatu *filter* dua dimensi dengan panjang dan tinggi (piksel) [33]. Untuk menghitung dimensi dari *feature map* dapat dihitung menggunakan rumus sebagai berikut [33].

$$output = \frac{W - N + 2P}{S} + 1 \quad (2.1)$$

dimana:

W : Panjang/Tinggi *Input*

N : Panjang/Tinggi *Filter*
P : *Zero Padding*
S : *Stride*

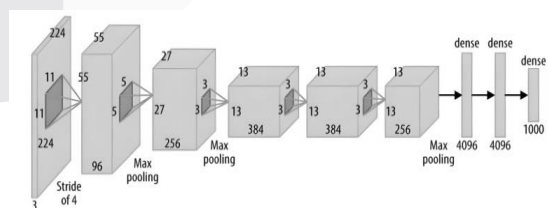
2. Pooling Layer

Pooling Layer terdiri dari sebuah *filter* dan *stride* dengan ukuran tertentu. Tujuan menggunakan *pooling layer* ini untuk mengurangi dimensi *feature map* (*downsampling*), sehingga proses komputasi jadi lebih cepat karena parameter yang di update semakin sedikit dan untuk mengatasi *overfitting*. Layer ini juga digunakan untuk mengurangi data menggunakan *Max Pooling* atau *Average Pooling*. *Max Pooling* akan mengambil nilai maksimum pada area *filter* saat pergeseran, sedangkan *Average Pooling* akan mengambil nilai rata-rata pada *filter* [33].

G. AlexNet

AlexNet Menjadi suatu terobosan baru pada deep learning dengan menerapkan *ConvNet* yang dipadukan dengan teknik *Dropout Regularization*, pemanfaatan *ReLU* sebagai fungsi aktivasi dan data augmentation. Alexnet dirancang untuk dapat melakukan klasifikasi dengan seribu kategori [19].

Arsitektur AlexNet terdiri atas 5 *convolution layer*, 3 *pooling layer*, 2 *dropout layer*, dan 3 *fully connected layer* [20]. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang dan menghasilkan *feature map*. Fungsi dari konvolusi sendiri adalah untuk ekstraksi fitur citra [21]. *Pooling* atau *subsampling* adalah pengurangan ukuran matriks. Terdapat dua macam *pooling* yang sering digunakan yaitu *average pooling* dan *max pooling* [22]. *Dropout* adalah cara yang dipakai untuk mengurangi *overfitting* pada *neural network*, hal ini dilakukan dengan mencegah adaptasi berlebihan pada data pelatihan. *Fully connected layer* merupakan kumpulan proses konvolusi [23]. Arsitektur Alexnet dapat dilihat pada Gambar 2.4 [20] dan detail layer Alexnet dapat dilihat pada Tabel 2.1 [20]



Gambar 2. 3 Arsitektur Alexne

Tabel 2. 1 Detail Arsitektur Alexnet

Layer	Feature Map	Size	Karnel Size	Stride	Activation
Input	1	$32 \times 32 \times 1$	-	-	-
Convolution 1	96	$16 \times 16 \times 96$	11×11	4	relu
Pooling 1	96	$8 \times 8 \times 96$	3×3	2	relu
Convolution 2	256	$8 \times 8 \times 256$	5×5	1	relu
Pooling 2	256	$4 \times 4 \times 256$	3×3	2	relu

Convolution 3	384	$4 \times 4 \times 384$	3×3	1	relu
Convolution 4	384	$4 \times 4 \times 384$	3×3	1	relu
Convolution 5	256	$4 \times 4 \times 256$	3×3	1	relu
Pooling 3	256	$2 \times 2 \times 256$	3×3	2	relu
Flatten	-	1024	-	-	relu
Fully connected 1	-	4096	-	-	relu
Fully connected 2	-	4096	-	-	relu
Fully connected 3	-	1000	-	-	Softmax

H. Confusion Matrix

Confusion matrix pada dasarnya memberikan informasi dari hasil perbandingan antara hasil klasifikasi sistem yang kita buat dengan hasil yang sebenarnya dalam bentuk tabel matriks [24]. Tabel matriks tersebut menggambarkan kinerja dari sistem klasifikasi pada serangkaian data *test* yang nilai sebenarnya diketahui. 4 Kombinasi nilai prediksi dan nilai aktual yang berbeda bisa menjadi gambaran dari *confusion matrix* dengan seperti pada Gambar 2.5 *Confusion Matrix* [25]

Untuk kasus pengklasifikasian dengan kelas lebih dari dua, maka digunakan *confusion matrix multiclass*. Maka dari itu apabila akan dilakukan pengklasifikasian dengan menggunakan N kelas maka *confusion matrix* akan menjadi $N \times N$ dimensi seperti dicontohkan pada Gambar 2.6 [29]

Tabel 2. 2 *Confusion Matrix Multiclass*

		True Class			
		→			
Predicted Class	Class Label	A	B	C	D
	A	X_{AA}	X_{AB}	X_{AC}	X_{AD}
	B	X_{BA}	X_{BB}	X_{BC}	X_{BD}
	C	X_{CA}	X_{CB}	X_{CC}	X_{CD}
	D	X_{DA}	X_{DB}	X_{DC}	X_{DD}

Pada Tabel 2.2 kita juga bisa menentukan nilai *True Positive* (TP), *True Negative* (TN), *False positive* (FP), dan *False Negative* (FN). Misalkan untuk menghitung nilai TP, TN, FP, FN untuk *Class A* keterangannya sebagai berikut

$$TP = X_{AA}$$

$$TN = (X_{BB} + X_{BC} + X_{BD} + X_{CB} + X_{CC} + X_{CD} + X_{DB} + X_{DC} + X_{DD})$$

$$FP = (X_{BA} + X_{CA} + X_{DA})$$

$$FN = (X_{AB} + X_{AC} + X_{AD})$$

Cara itu juga berlaku untuk *class* yang lainnya, hingga semua nilai untuk setiap *class* didapatkan.

Untuk mengukur performansi dari sistem yang telah dirancang. Perhitungan *performance metrics* dari *confusion matrix multiclass* adalah sebagai berikut:

Akurasi memberitahukan seberapa akurat sistem yang telah kita rancang dalam pengklasifikasian. *Accuracy* adalah rasio prediksi yang benar (positif dan negatif) terhadap keseluruhan data. Atau dapat dikatakan bahwa *accuracy* itu tingkat kedekatan antara nilai yang di prediksikan dengan nilai sebenarnya. Persamaan yang digunakan untuk menghitung *accuracy* pada *confusion matrix multiclass* adalah seperti pada Persamaan (2.1) [27] [29]

$$\text{Akurasi} = \frac{TP(A) + TP(B) + TP(C) + TP(D)}{TP + TN + FP + FN} \quad (2.2)$$

Presisi memberitahukan keakuratan antara data yang diminta dengan hasil yang di prediksikan oleh sistem. *Precision* adalah rasio prediksi perbandingan antara benar (positif) dengan seluruh hasil prediksi positif lainnya (benar dan salah). Persamaan yang digunakan untuk mendapatkan nilai *precision* adalah seperti pada Persamaan (2.2) [27]

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2.3)$$

Selanjutnya untuk menghitung nilai *average precision* dapat digunakan persamaan (2.3) [29]

$$\text{Presisi rata-rata} = \frac{P(A) + P(B) + P(C) + P(D)}{\text{Total kelas}} \quad (2.4)$$

Recall memberitahukan keberhasilan sistem ketika informasi ditemukan kembali. *Recall* itu adalah rasio perbandingan antara prediksi benar (positif) dengan benar (positif) dan salah (negative). Persamaan yang digunakan untuk mendapatkan nilai *recall* adalah seperti pada Persamaan (2.4) [27]

$$\text{recall} = \frac{TP}{TP + FN} \quad (2.5)$$

Selanjutnya untuk menghitung nilai *average recall* dapat digunakan Persamaan (2.5)

$$\text{Recall rata-rata} = \frac{R(A) + R(B) + R(C) + R(D)}{\text{Total kelas}} \quad (2.6)$$

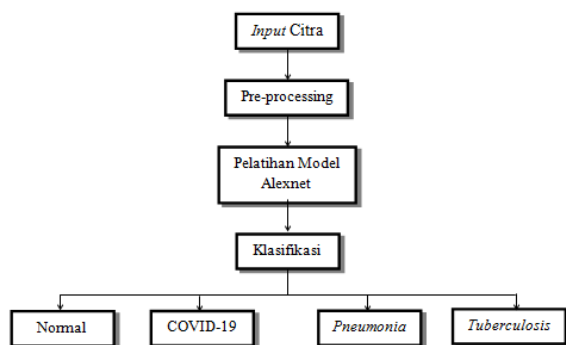
F1-score digunakan untuk membuat nilai *precision* yang rendah dan nilai *recall* yang tinggi atau sebaliknya menjadi seimbang. *F1-score* membantu perhitungan *recall* dan *precision* secara bersamaan. Persamaan yang digunakan

untuk mendapatkan nilai *F1-score* adalah seperti pada Persamaan (2.5)[28]

$$F1\text{-score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (2.7)$$

III. METODE

Pembuatan sistem untuk Tugas Akhir ini akan menggunakan sistem CNN dengan arsitektur Alexnet untuk mengklasifikasikan penyakit yang menjadi empat kelas. Gambaran umum dari desain sistem yang dibuat terbagi menjadi 4 tahapan, yaitu *input* citra, *pre-processing*, pelatihan model, dan klasifikasi.

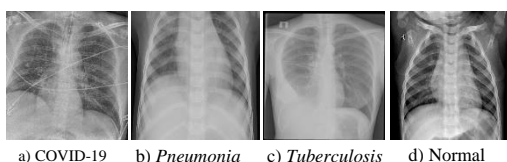


Gambar 3. 1 Blok Diagram Sistem

Seperti terlihat Gambar 3.1 yang merupakan blok diagram sistem secara keseluruhan. Tahap ini cukup menentukan untuk keberhasilan sistem, citra yang sebelumnya telah dimasukan kemudian diproses dalam beberapa tahapan dengan tujuan mendapatkan citra yang terbaik untuk dapat diproses dengan baik pada tahap selanjutnya.

A. Dataset

Pada Tugas Akhir ini *dataset* yang digunakan berasal dari *kaggle* sebanyak 2304 citra terbagi ke dalam empat kelas, diantaranya COVID-19, *pneumonia*, *tuberculosis*, dan normal. Dari keseluruhan data 2304 citra dengan format JPEG dan jumlah dataset yang sama pada masing-masing kelas yaitu 576 citra. *Dataset* tersebut dibagi menjadi data train dan data test, dari total data yang ada dibagi menjadi 75% untuk data train dan 25% untuk data test. Untuk contoh *dataset* dari beberapa kelas yang, dapat dilihat pada Gambar 3.2



Gambar 3. 2 Dataset Rontgen Dada

B. Pre-processing

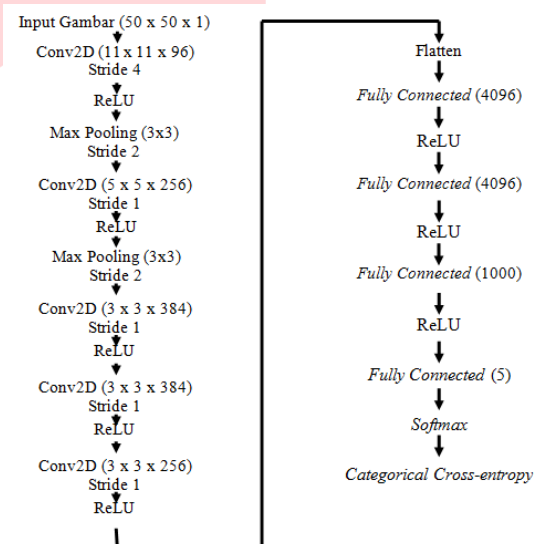
Pre-processing dilakukan dengan tujuan memperbaiki atau meningkatkan kualitas dan menyeragamkan citra agar dapat meningkatkan

performansi sistem, jika tidak seragam, model akan lebih sulit untuk mengolah data sehingga performansi model tidak maksimal. Adapun langkah yang dilakukan pada *pre-processing* yaitu: *Resizing* Citra.

Resizing citra dilakukan dengan tujuan menyeragamkan ukuran citra dalam satuan pixel. Selain itu, ukuran citra yang terlalu besar dapat membuat pemrosesan pada sistem menjadi semakin lama, dan juga ukuran citra yang terlalu kecil dapat membuat kualitas citra yang kurang baik. *Pre-processing resizing* citra dilakukan pada semua model pengujian.

C. Pelatihan Model

Tahapan dari proses pelatihan model Alexnet dimulai dengan input citra dari *dataset* hingga diperoleh klasifikasi COVID-19, *pneumonia*, *tuberculosis* yang diinginkan, untuk lebih jelasnya dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Diagram Pelatihan Model Alexnet

Pelatihan model dilakukan agar model dapat mengklasifikasikan citra dengan nilai akurasi tinggi dan kesalahan yang rendah. Proses pelatihan model dilakukan dengan memasukkan beberapa citra penyakit yang sudah memiliki label dari masing-masing kelas, untuk melewati model yang sudah dibuat. Model yang digunakan adalah CNN dengan arsitektur Alexnet. Proses pelatihan dilakukan terus berulang hingga model mendapatkan parameter terbaik, sehingga model dapat melakukan klasifikasi secara akurat.

IV. HASIL DAN PEMBAHASAN

Pengukuran hasil performansi sistem klasifikasi ini menggunakan nilai akurasi, presisi, *recall* dan *f1-score* yang diperoleh dari hasil pengujian sistem. Proses pelatihan dan pengujian sistem dilakukan menggunakan *Google Colaboratory*. Adapun pengujian sistem yang

dilakukan melalui beberapa skenario pengujian, dan menunjukkan hasil sebagai berikut.

A. Hasil Pengujian Sistem

1. Pengujian *Resize* Citra

Citra dataset yang didapat awalnya memiliki ukuran yang berbeda. Untuk klasifikasi pada citra dibutuhkan harus memiliki semua ukuran citra yang sama. Oleh karena itu dilakukan skenario pengujian sistem untuk mencari ukuran terbaik yang dapat digunakan untuk merubah ukuran citra dataset menjadi sama antara masing-masing citra. Pengujian dilakukan menggunakan parameter yang telah ditentukan, yaitu *optimizer* Adam, *learning rate* 0,0001, *epoch* 35, *batch size* 32.

Pada pengujian pertama ini kita merubah ukuran pada citra yang akan digunakan yaitu 32×32 piksel, 64×64 piksel, dan yang terakhir 128×128 piksel. Hasil dari pengujian pertama ditunjukkan pada Tabel 4.1 sebagai berikut

Tabel 4. 1 Perbandingan Hasil Pengujian *Resize*

Parameter	Hasil Pengujian <i>Resize</i>		
	32×32 px	64×64 px	128×128 px
Akurasi	0,92	0,96	0,95
Presisi	0,927	0,96	0,96
Recall	0,927	0,96	0,96
F1-Score	0,925	0,96	0,96
Loss	0,38	0,32	0,17

Hasil pengujian *resize* terbaik ini menggunakan *resize* 64×64 piksel dengan nilai akurasi 96%, presisi 96%, *recall* 96%, *f1-score* 96%, dan nilai loss sebesar 0,32 yang dapat dilihat pada Tabel 4.1. Hal ini dikarenakan dengan menggunakan dataset dan model ini, penggunaan *resize* 32×32 piksel dan 128×128 piksel menyebabkan beberapa informasi yang hilang karena ukurannya yang kecil ataupun terlalu besar bagi dataset yang digunakan pada penelitian ini sehingga nilai akurasi yang didapat kurang optimal. Adapun kurun waktu komputasi dalam pengujian *resize* ini bisa kita lihat pada Tabel 4.2

Tabel 4. 2 Waktu Komputasi pada Pengujian *Resize*

<i>Resize</i> (px)	Waktu (s)
32×32	42
64×64	53
128×128	72

Berdasarkan Tabel 4.2 dapat kita ketahui bahwa penggunaan *resize* 128×128 piksel itu membutuhkan waktu paling lama, yaitu 72 detik, dikarenakan semakin diperbesar ukuran citra maka akan semakin lama waktu yang dibutuhkan. Sebaliknya, semakin diperkecil ukuran citra maka akan semakin sedikit waktu yang dibutuhkan.

2. Pengujian Nilai *Learning Rate*

Learning rate adalah skenario kedua pada parameter pengujian yang akan digunakan untuk mengukur kinerja sistem. Parameter pengujian terbaik yang telah didapatkan dari skenario sebelumnya akan digunakan pada skenario pengujian *learning rate* ini, dengan parameter lainnya yaitu *optimizer* Adam, *epoch* 35, *batch size* 32. Pengujian *learning rate* dilakukan dengan menggunakan nilai 0,01, 0,001, 0,0001. Berikut hasil dari pengujian *learning rate* ditunjukkan pada Tabel 4.3

Tabel 4. 3 Hasil Pengujian *Learning Rate*

Parameter	Hasil Pengujian		
	0,01	0,001	0,0001
Akurasi	0,23	0,91	0,95
Presisi	0,057	0,915	0,96
Recall	0,25	0,9	0,96
F1-Score	0,095	0,91	0,96
Loss	1,38	0,34	0,32

Berdasarkan Tabel 4.3, pengujian pada *learning rate* 0,0001 mendapatkan nilai yang terbaik yaitu dengan akurasi 95%, *loss* 0,32, dan nilai presisi, *recall*, *f1-score* yang sama yaitu mencapai nilai 96%. Hal ini dikarenakan semakin kecil nilai *learning rate* maka sistem akan bekerja lebih keras dalam melakukan identifikasi, sehingga nilai akurasi yang dihasilkan akan semakin optimal, begitupun sebaliknya.

3. Pengujian Nilai *Epoch*

Epoch merupakan jumlah parameter pengujian yang menentukan berapa kali algoritma pembelajaran akan bekerja mengolah seluruh dataset training. Penggunaan *epoch* juga mempengaruhi hasil pada tingkat akurasi. Nilai *epoch* yang dipakai dalam pengujian ini adalah 5, 10, 25, 35. Dari parameter-parameter terbaik yang sebelumnya sudah didapatkan yaitu *resize* 64×64 piksel, *learning rate* 0,0001, dan parameter lainnya yang telah ditentukan yaitu *optimizer* Adam, dan *batch size* 32. Hasil dari pengujian dapat dilihat pada Tabel 4.4 berikut

Tabel 4. 4 Hasil Pengujian *Epoch*

Parameter	Hasil Pengujian <i>Epoch</i>			
	5	10	25	35
Akurasi	0,9	0,9	0,95	0,95
Presisi	0,87	0,91	0,95	0,94
Recall	0,86	0,9	0,95	0,94
F1-Score	0,86	0,9	0,95	0,94
Loss	0,315	0,355	0,178	0,343

Berdasarkan Tabel 4.4 kita bisa ketahui bahwa pada pengujian nilai *epoch* ini, nilai terbaik yang didapat adalah nilai *epoch* 25 yang memiliki tingkat akurasi, presisi, *recall*, *f1-score*, *loss* yang sama yaitu sebesar 95% dengan nilai *loss* paling

kecil diantara pengujian nilai *epoch* yang lainnya sebesar 0,178. Kita bisa simpulkan bahwa pada model Alexnet dan dataset yang digunakan pada penelitian ini, untuk pengujian nilai *epoch* hanya cocok dengan menggunakan nilai 25. Meskipun nilai akurasi pada nilai *epoch* 25 dan 35 itu sama 95%, akan tetapi nilai *loss* pada nilai *epoch* 25 lebih rendah yaitu sebesar 0,178.

4. Pengujian Nilai Batch Size

Dari skenario pengujian sebelumnya, kita telah mendapatkan nilai terbaik *resize* 64×64 piksel, *learning rate* 0,0001, *epoch* 25. Dalam pengujian ini kita akan menggunakan nilai terbaik dari skenario pengujian sebelumnya, dan untuk *optimizer* yang telah ditentukan yaitu menggunakan *optimizer* Adam. Adapun parameter nilai *batch size* yang akan digunakan dalam pengujian kali ini yaitu menggunakan nilai 8, 16, 32. Tabel 4.5 berikut ini adalah hasil dari pengujian *batch size*

Tabel 4. 5 Hasil Pengujian Nilai Batch Size

Parameter	Hasil Pengujian Batch Size		
	8	16	32
Akurasi	0,944	0,942	0,95
Presisi	0,942	0,94	0,95
Recall	0,942	0,942	0,95
F1-Score	0,942	0,942	0,95
Loss	0,434	0,224	0,161

Bisa kita ketahui dari Tabel 4.5 Hasil Pengujian Nilai *Batch Size* terbaik terdapat pada nilai *batch size* 32 dengan akurasi, presisi, *recall*, *f1-score*, dan *loss* yang sama yaitu sebesar 95%. Dapat kita simpulkan untuk pengujian *batch size* dengan dataset dan model yang digunakan, *batch size* 32 yang cocok.

5. Pengujian Optimizer

Beberapa parameter pengujian *optimizer* yang akan digunakan adalah Adam, Nadam, dan RMSprop. Pengujian dilakukan dengan menggunakan parameter terbaik yang telah didapat pada skenario pengujian sebelumnya, yaitu *resize* 64×64 piksel, *learning rate* 0,0001, *epoch* 25, *batch size* 32, dikarenakan pengujian ini adalah pengujian yang terakhir, sehingga tidak ada lagi nilai yang ditentukan. Berikut adalah hasil dari skenario pengujian *optimizer* dapat dilihat pada Tabel 4.6

Tabel 4. 6 Hasil Pengujian Optimizer

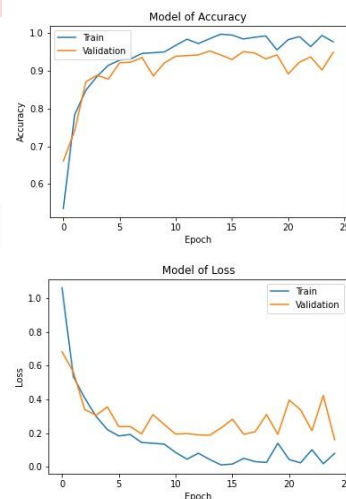
Parameter	Hasil Optimizer		
	Adam	Nadam	RMSprop
Akurasi	0,95	0,947	0,93
Presisi	0,95	0,947	0,94
Recall	0,95	0,947	0,94
F1-Score	0,95	0,947	0,94

Loss	0,161	0,287	0,251
------	-------	-------	-------

Berdasarkan hasil pengujian *optimizer* pada Tabel 4.6 menunjukan penggunaan *optimizer* terbaik pada model dan dataset penelitian ini adalah menggunakan *optimizer* Adam dengan hasil akurasi, presisi, *recall*, *f1-score* sebesar 95% dan nilai *loss* sebesar 0,161.

B. Analisa Pengujian Terbaik

Dari lima skenario parameter pengujian, telah dilakukan *resizing* citra, *learning rate*, *epoch*, *batch size* dan *optimizer*, didapatkan beberapa parameter terbaik dari masing-masing skenario pengujian. Nilai terbaik yang didapatkan dari masing-masing skenario pengujian antara lain untuk *resizing* 64x64 piksel, *learning rate* 0,0001, *epoch* 25, *batch size* 32 dan *optimizer* Adam. Untuk grafik model akurasi dan *loss* dari skenario terbaik dapat dilihat pada gambar 4.1 berikut ini.

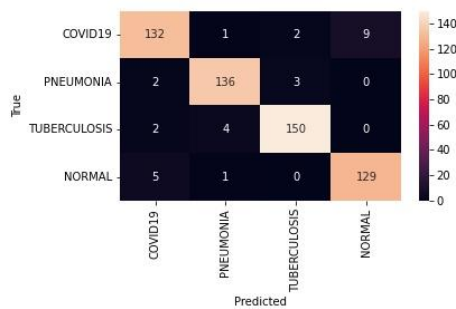


Gambar 4. 1 Grafik Model Akurasi dan Loss terbaik

Tabel 4. 7 Hasil Performansi dari Parameter Pengujian Terbaik

Parameter		Performansi	
Resize	64×64 piksel	Akurasi	0,95
Learning Rate	0,0001	Presisi	0,95
Epoch	25	Recall	0,95
Batch Size	32	F1-Score	0,95
Optimizer	Adam	Loss	0,161

Selain itu hasil pelatihan dapat dilihat dari *confusion matrix* yang dihasilkan, untuk mengetahui jumlah data yang berhasil dideteksi dengan benar ataupun salah oleh sistem. Berikut Gambar 4.2 menunjukan *confusion matrix* dari hasil kinerja terbaik,



Gambar 4. 2 Hasil *Confusion Matrix* dari skenario terbaik

Hasil *confusion matrix* yang didapatkan lihat pada Gambar 4.2 menunjukkan bahwa dari 576 citra yang digunakan untuk data validasi, sebanyak 547 citra berhasil dideteksi dengan benar sesuai dengan kelasnya masing-masing, sedangkan 29 citra lainnya terdeteksi salah.

V. KESIMPULAN

A. Kesimpulan

Untuk klasifikasi empat kelas yaitu Normal, Covid19, *Pneumonia*, *Tuberculosis* menggunakan CNN dengan arsitektur Alexnet. Hasil yang didapat sudah cukup baik dan optimal dalam melakukan pengklasifikasian, dibuktikan dengan akurasi yang diperoleh sistem yaitu sebesar 95%. Selain itu, nilai performansi yang lain seperti presisi, *recall*, *f1-score* mendapatkan nilai yang sama yaitu sebesar 95% dan *loss* sebesar 0,161.

Untuk pengaruh *Resize*, *Learning Rate*, *Epoch*, *Batch Size*, dan *Optimizer* pada sistem telah mendapatkan nilai terbaik yaitu dengan *Resize* 64×64 piksel, *learning rate* 0,0001, *epoch* 25, *batch size* 32, dan yang terakhir dengan *Optimizer* Adam.

Bisa kita simpulkan bahwa aspek dari setiap parameter pengujian itu sangat berpengaruh terhadap hasil performansi sistem yang didapatkan, jika nilai parameter terlalu besar atau terlalu kecil dapat menghasilkan tingkat error yang berbeda-beda. Jadi untuk penelitian kali ini dengan dataset dan model Alexnet yang dipakai nilai *Resize* 64×64 piksel, *learning rate* 0,0001, *epoch* 25, *batch size* 32, dan *optimizer* Adam adalah yang paling cocok.

B. Saran

Dikesempatan ini penulis ingin menjelaskan bahwa penulis sadar masih terdapat banyak kesalahan atau kurang dalam Tugas Akhir ini, dengan harapan penelitian ini bisa ditindak lebih lanjut juga lebih dalam lagi agar bisa berguna untuk siapapun. Jumlah dataset yang digunakan dalam penelitian ini masih jauh dari kata cukup, untuk penambahan jumlah dataset mungkin akan menghasilkan tingkat akurasi dan parameter lainnya yang lebih tinggi. Mencoba untuk

mengklasifikasikan dengan lebih banyak model arsitektur lain dan membandingkan hasilnya.

Daftar Pustaka:

- [1] WHO "Tuberculosis and COVID-19" online tersedia <https://www.who.int/teams/global-tuberculosis-programme/covid-19>.
- [2] Pebri Andhi Herry Pratama, Rony Teguh, Abertun Sagit Sahay, Valencia Wilentine. 2021. DETEKSI COVID-19 BERDASARKAN HASIL RONTGEN DADA (CHEST X-RAY) MENGGUNAKAN PYTHON. Jurusan Teknik Informatika, Fakultas Teknik Universitas Palangkaraya, Kalimantan Tengah, Indonesia.
- [3] Noor N. Qaqos, Omar S. Kareem. 2020. COVID-19 DIAGNOSIS FROM CHEST X-RAY IMAGES USING DEEP LEARNING APPROACH. IT Department Duhok Polytechnic University.
- [4] Xiaochun Wei, Zihao Zhang. 2020. Comparative Experiment of CNN Models Based on Pneumonia X-ray Image Detection. IEEE
- [5] Ali Serener, Sertan Serte. 2020. Deep Learning for Mycoplasma Pneumonia Discrimination from Pneumonias like COVID-19. IEEE
- [6] Kaggle "Chest X-Ray (Pneumonia, COVID-19, Tuberculosis)" <https://www.kaggle.com/jtptj/chest-xray-pneumoniacovid19tuberculosis>
- [7] Imagenet Large Scale Visual Recognition Challenge (ILSVRC). Diakses 10 Oktober 2020.
- [8] Nurul Hidayah Nasution, Arinil Hidayah, Khoirunnisa Mardiah sari, Wirda Cahyati, Mar'atun Khoiriyah, Riska Putriani Hasibuan, Ahmad Afandi Lubis, Andi Yahya Siregar. 2021. Gambaran Pengetahuan Masyarakat tentang Pencegahan COVID-19 di Kecamatan Padangsidimpuan Batun Adua, Kota Padangsidimpuan. *Indonesian Health Scientific Journal*
- [9] Doremalen, Et Al. (2020). Aerosol And Surface Stability Of SARS-Cov-2 As Compared With SARS-Cov-1. *The New England Journal Of Medicine*, 382(16), Pp. 1564-7.
- [10] Htun, et al. (2019). Clinical Features for Diagnosis of Pneumonia among Adults in Primary Care Setting: A Systematic and Meta-review. *Scientific Reports*, 9(1), pp. 7600.
- [11] Campbell, I. Bah-Sow, O. (2006). Pulmonary Tuberculosis: Diagnosis and Treatment. *BMJ*. 332(7551), pp.1194–1197.
- [12] Kementerian Kesehatan Republik Indonesia (2003). Artikel. Perlu Keterpaduan untuk Pemberantasan Tuberkulosis.
- [13] Rinaldi Munir, 2018. PENGANTAR PENGOLAHAN CITRA (BAGIAN1) 47 IF4073 INTERPRETASI DAN PENGOLAHAN CITRA, Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, Indonesia.

- [14] Samuel, Arthur. MACHINE VISION. McGraw-Hill. 1959.
- [15] Wong, Yue-Ling (2009). Digital Media Primer - Digital, Audio, Video, Imaging, and Multimedia Programming (8th ed). Pearson International Education. ISBN 978-0-13-815582-7.
- [16] Suryanto, Nur Ramadhani Kurniawan, Satria Mandala. 2019. 'DEEP LEARNING MODERNISASI MACHINE LEARNING UNTUK BIG DATA'. Bandung: Informatika Bandung.
- [17] Hanung Adi Nugroho, Julisa Bana Abraham, Aina Hubby Azira, Eka Legya Franita, Rizki Nur Fauzi, Maula Azif, E. Elsa Herdiana Murhandarwati. 2019. PERFORMANCE OF CONVOLUTIONAL NEURAL NETWORK IN DETECTING PLASMODIUM PARASITES. Yogyakarta: Universitas Gajah Mada.
- [18] H. Darmanto, D. Learning, T. Learning, dan G. Descent, "PENGENALAN SPESIES IKAN BERDASARKAN KONTUR OTOLITH MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK," *juornal informatics Educ.*, vol. 2, 2019.
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *NeurIPS Proceedings*, 2012.
- [20] M. Alban and T. Gilligan, "Automated Detection of Diabetic Retinopathy using Fluorescein Angiography Photograph," Stanford University, 2016.
- [21] I. W. Suartika, A. Y. Wijaya, dan R. Soelaiman, "Klasifikasi Citra Menggunakan *Convolutional Neural Network* (CNN) pada Caltech 101," *Jurnal Teknik ITS* Vol. 5, No. 1, , 2016.
- [22] M. B. Bejiga, A. Zeggada, A. Nouffidj and F. Melgani, "A *Convolutional Neural Network* Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery". *Remote Sens.* pp 5, 2016.
- [23] S. Hijazi, R. Kumar and C. Rowen, "Using Convolutional Neural Networks for Image Recognition," *Remote Sens.* pp 7, 2015.
- [24] Burchard, 2006; Sudoyo et al., 2009
- [25] "Medical Image Analysis Group (MIAG)," [Online]. Available: <http://medimrg.webs.ull.es/>. [Diakses 29 November 2020].
- [26] B. I. Taweh, Introduction to Deep Learning Using R, San Francisco, California, USA: Apress, 2017.
- [27] S. N. Kuncahyo, "Confusion Matrix untuk Evaluasi Model pada Supervised Learning," *medium.com*, 13 November 2019. [Online]. Available: <https://medium.com/@ksnugroho/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f>. [Accessed 5 Desember 2020].
- [28] N. Sarang, "Understanding Confusion Matrix," towards data science, 9 Mei 2018. [Online]. Available: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>. [Accessed 5 Desember 2020].
- [29] s. shubhanshi, "Confusion Matrix in Machine Learning," *medium.com*, 14 Desember 2018. [Online]. Available: <https://medium.com/@shubhanshi.shubh860/confusion-matrix-in-machine-learning-cd7333d72f5d>. [Accessed 5 Desember 2020].
- [30] NK Caecar Pratiwi, Nur Ibrahim, Yunendah Nur Fu'adah, Kholidiyah Masykuroh. "Computer-Aided Detection (CAD) for COVID-19 based on Chest X-Ray Images using Convolutional Neural Network". Electrical Engineering faculty – Telkom University, Bandung, Indonesia. ICETIR 2020, IOP Publishing.
- [31] Elbert Alfredo Tjoa, I Putu Yowan Nugraha Suparta, Rita Magdalena, Nor Kumalasari CP. "The Use of CLAHE for improving an accuracy of CNN architecture for detecting pneumonia". ETLTC 202, SHS Web of Conferences 139.
- [32] Dian Ayu N. "ANALISIS PERFORMANSI SISTEM KLASIFIKASI KANKER KULIT MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*" Telkom University, 2022
- [33] Eka P. W. S., Wijaya, A. Y., Soelaiman, R., "Klasifikasi Citra Menggunakan *Convolutional Neural Network* (Cnn) pada Caltech 101," *J. Tek. ITS*, vol. 5, ISSN: 2337-3539, 2016.