

**Klasifikasi Citra *Chest X-Ray* (CXR) Penyakit
COVID-19 dan Normal dengan Metode
Convolutional Neural Network (CNN)
menggunakan *Python***

SKRIPSI

Diajukan untuk Memenuhi Sebagian Syarat
Guna Memperoleh Gelar Sarjana Sains
dalam Ilmu Fisika



Oleh

Nur Chikmi Azizi

NIM. 1608026002

**PROGRAM STUDI FISIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG**

2022

PERNYATAAN KEASLIAN

Yang bertandatangan dibawah ini :

Nama : Nur Chikmi Azizi

NIM : 1608026002

Jurusan : Fisika

Menyatakan bahwa skripsi yang berjudul :

Klasifikasi Citra *Chest X-Ray* (CXR) Penyakit COVID-19 dan Normal dengan Metode *Convolutional Neural Network* (CNN) menggunakan *Python*.

Secara keseluruhan adalah hasil penelitian saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 12 Desember 2022
Pembuat pernyataan,

A handwritten signature in black ink is written over a 1000 Rupiah stamp. The stamp features the Garuda Pancasila emblem and the text 'REPUBLIK INDONESIA', 'SERIBU RUPIAH', 'METER', 'TELAPIL', and 'AN 666AKX168841773'.

Nur Chikmi Azizi

NIM. 1608026002

LEMBAR PENGESAHAN



KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI

Jl. Prof. Dr. H. Soekarno Kampus II UIN Walisongo Ngaliyan Semarang
Telp. (024) 7601295, Fax 7613 097

PENGESAHAN

Naskah skripsi berikut ini:

Judul : Klasifikasi Citra *Chest X-Ray* (CXR) Penyakit COVID-19 dan Normal dengan Metode *Convolutional Neural Network* (CNN) menggunakan *Python*

Nama : Nur Chikmi Azizi

NIM : 1608026002

Jurusan : Fisika

Telah diajukan dalam sidang munaqasyah oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam ilmu fisika.

Semarang, 28 Desember 2022

DEWAN PENGUJI

Ketua

Henri Sumarti, M.Si
NIP. 19871011 201903
Penguji I

Sekretaris

Fachrizal Rini Pratama, M.Sc.
NIP. 198906262019031012
Penguji II

Irman Sujit Prastyo, M.Si
NIP. 199112282019031009
Pembimbing I



Kully Anggita, S.Pd., M.Si.
NIP. 199005052019032017
Pembimbing II

Edi Daenuri Anwar, M.Si
NIP. 197907262009121002

Henri Sumarti, M.Si
NIP. 19871011 201903 2009

NOTA DINAS

Semarang, 13 Desember 2022

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Walisongo
Di Semarang

Assalamualaikum wr.wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : Klasifikasi Citra *Chest X-Ray* (CXR) Penyakit COVID-19 dan Normal dengan Metode *Convolutional Neural Network* (CNN) menggunakan *Python*

Nama : Nur Chikmi Azizi

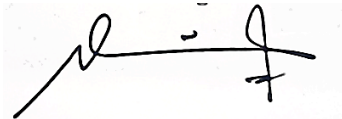
NIM : 1608026002

Jurusan : Fisika

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang untuk diajukan dalam Sidang Munaqosyah.

Wassalamu'alaikum wr.wb.

Pembimbing I



Edi Daenuri Anwar, M.Si

NIP. 19790726 200912 1 002

NOTA DINAS

Semarang, 13 Desember 2022

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Walisongo
Di Semarang

Assalamualaikum wr.wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : Klasifikasi Citra *Chest X-Ray* (CXR) Penyakit COVID-19 dan Normal dengan Metode *Convolutional Neural Network* (CNN) menggunakan *Python*

Nama : Nur Chikmi Azizi

NIM : 1608026002

Jurusan : Fisika

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang untuk diajukan dalam Sidang Munaqosyah.

Wassalamu'alaikum wr.wb.

Pembimbing II



Heni Sumarti, M.Si

NIP. 19871011 201903 2 009

ABSTRAK

Diagnosis penyakit COVID-19 yang cepat dan akurat memiliki peran penting dalam membantu tenaga medis untuk memberikan perawatan yang tepat waktu. CXR dapat menjadi alternatif yang baik dalam diagnosis COVID-19 dibanding dengan RT-PCR yang memiliki sensitivitas rendah. Namun, terkadang kualitas CXR kurang optimal sehingga waktu yang dibutuhkan cukup lama. Penelitian ini bertujuan untuk membuat model klasifikasi CXR penyakit COVID-19 dan normal yang efektif dan efisien menggunakan *python*. Metode *Convolutional Neural Network* (CNN) dengan data sebanyak 200 citra CXR digunakan dalam penelitian ini. Sebelum melatih model yang diusulkan, dilakukan *preprocessing* seperti pelabelan, *resize*, dan mengubah ke *grayscale*. Berdasarkan hasil pengujian dari model yang diusulkan diperoleh waktu rata-rata yang dibutuhkan 3 s, nilai rata-rata akurasi sebesar 96% dan nilai rata-rata *loss* sebesar 15,29%. Model yang diusulkan lebih efektif dan efisien sehingga dapat memberikan solusi penanganan kasus

Kata kunci : *Convolutional Neural Network*, klasifikasi, COVID-19, *python*

KATA PENGANTAR

Assalamu'alaikum wr. wb.

Puji syukur penulis panjatkan atas kehadiran Allah SWT, karena berkat rahmat dan karunia-Nya sehingga skripsi ini dapat terselesaikan. Sholawat serta salam senantiasa tercurah kepada Nabi Muhammad SAW, beserta keluarga, para sahabat serta para pengikutnya. Penyusunan skripsi dengan judul "**Klasifikasi Citra *Chest X-Ray* (CXR) Penyakit COVID-19 dan Normal dengan Metode *Convolutional Neural Network* (CNN) menggunakan *Python***" dimaksudkan untuk memenuhi syarat memperoleh gelar Sarjana Sains di Jurusan Fisika Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo Semarang.

Dalam kesempatan ini, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu, memberi bimbingan, nasihat, arahan, dorongan, semangat, maupun doa, sehingga skripsi ini dapat terselesaikan dengan baik. Ucapan terima kasih ini penulis sampaikan kepada :

1. Prof. Dr. Imam Taufiq, M.Ag., Rektor Universitas Islam Negeri Walisongo Semarang.
2. Dr. Ismail, M.Ag., Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo Semarang.
3. Agus Sudarmanto, M.Si., Ketua Program Studi Fisika Universitas Islam Walisogo Semarang.
4. Edi Daenuri Anwar, M.Si, Dosen Pembimbing I yang telah memberikan banyak semangat, dan dedikasinya kepada penulis sehingga mampu untuk terus berkarya dalam proses penyusunan skripsi.
5. Heni Sumarti, M.Si., Dosen Pembimbing II yang memberikan bimbingan, kritik, saran serta motivasi kepada penulis dengan penuh pengertian dalam penyusunan skripsi.
6. M. Ardhi Khalif, M.Sc, wali dosen yang memberikan banyak masukan, dukungan, dan doa kepada penulis sehingga dapat menyelesaikan proses penyusunan skripsi.
7. Dosen, Pegawai dan civitas akademik Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo Semarang yang telah memberikan banyak ilmu pengetahuan dan pemahaman.
8. Kedua Orang Tua, Ayahanda tercinta (Nur Hasyim) dan ibunda tercinta (Sri Wahyuni Faiqoh) serta Wafiq Nur Azizi, Sinta Choiriyah, Sukani selaku saudara kandung

dan seluruh keluarga besar yang selalu memberikan do'a, semangat, dukungan dan kasih sayang kepada penulis sehingga menyelesaikan studi dengan baik dan lancar.

9. Puspita, Arina, Ririn, Zulfa, Dika, Martini, Zadid Taqwa, Noto s, Susi, Agus, Ika, Hani, Ela, yessi yang selalu mendukung dan membantu penulis untuk segera menyelesaikan skripsi ini.
10. Keluarga besar kelas Fisika 2016, segenap keluarga Ikatan Mahasiswa Fisika, yang telah turut membentuk pribadi penulis.
11. Sahabat- sahabati PMII rayon sains & teknologi, Immawan immawati al-Faruqi, kanda dan dinda, serta Kesatuan Aksi Mahasiswa Muslim Indonesia (KAMMI), yang telah memberikan pengalaman dan memberikan banyak sudut pandang bagi penulis.
12. Keluarga besar bimbingan belajar cemerlang, genius school, *alternative education* yang selalu memberikan motivasi dan pengalaman selama masa perjuangan menyelesaikan studi.
13. Semua pihak yang tidak bisa disebutkan satu per satu yang telah memberikan semangat demi terselesaikanya tugas akhir ini.

Penulis menyadari skripsi ini banyak kekurangan dan jauh dari kesempurnaan. Besar harapan penulis untuk saling

berbagi wawasan satu sama lain. Segala pertanyaan, kritik, dan saran dapat disampaikan melalui surat elektronik di alamat chikmiazizi001@gmail.com semoga naskah skripsi ini dapat memberikan manfaat bagikita semua khususnya bagi perkembangan ilmu fisika medis di masa yang akan datang.

Semarang, 12 Desember 2022

Penulis

A handwritten signature in black ink, consisting of a stylized 'C' followed by 'h' and 'i' and a long horizontal line extending to the right.

Nur Chikmi Azizi

DAFTAR ISI

PERNYATAAN KEASLIAN	xii
LEMBAR PENGESAHAN	xiii
NOTA DINAS	xiv
NOTA DINAS	xv
ABSTRAK	xvi
KATA PENGANTAR.....	xvii
DAFTAR ISI.....	xxi
DAFTAR GAMBAR	xxiii
DAFTAR TABEL	xxiv
DAFTAR LAMPIRAN.....	xxv
BAB I PENDAHULUAN	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	7
C. Tujuan Penelitian.....	7
D. Manfaat Penelitian	7
E. Batasan Masalah.....	7
BAB II TINJAUAN PUSTAKA.....	9
A. Kajian Teori.....	9
1. COVID-19	9
2. <i>Chest X-Rays</i>	11
3. Pengolahan Citra Digital.....	15
4. <i>Contrast Limited Adaptive Histogram Equalization</i> (CLAHE)	17

5.	<i>Convolutional Neural Network (CNN)</i>	18
6.	<i>Confusion Matrix</i>	23
7.	<i>Python</i>	25
B.	Kajian Pustaka.....	26
C.	Rumusan Hipotesis	28
BAB III METODOLOGI PENELITIAN		29
A.	Waktu dan Tempat Penelitian.....	29
B.	Alat Penelitian.....	29
C.	Bahan Penelitian	30
D.	Prosedur Penelitian	31
E.	Analisis Data	38
BAB IV HASIL DAN PEMBAHASAN		39
A.	Dataset Citra	39
B.	Hasil <i>Preprocessing</i> citra.....	40
C.	Hasil arsitektur CNN.....	42
D.	Hasil Pengujian	49
E.	Perbandingan dengan penelitian sebelumnya	54
BAB V PENUTUP		57
DAFTAR PUSTAKA		59
LAMPIRAN- LAMPIRAN		66

DAFTAR GAMBAR

Gambar 2. 1	(a) Representasi Skema Tabung X Ray (b) Foto tabung X-Ray	12
Gambar 2. 2	Citra Chest X-rays	15
Gambar 2. 3	Representasi citra dan intensitas piksel citra	16
Gambar 2. 4	Ilustrasi arsitektur CNN secara umum	19
Gambar 2. 5	Proses Pada Tahap Klasifikasi	23
Gambar 3. 1	Skema sistem klasifikasi COVID-19	32
Gambar 3. 2	Skema prosedur preprocessing citra	33
Gambar 4. 1	Citra CXR paru-paru (a) Normal (b) COVID-19	40
Gambar 4. 2	sourcecode python preprocessing citra	41
Gambar 4. 3	Citra CXR paru-paru setelah preprocessing	42
Gambar 4. 4	Hasil sitem arsitektur CNN	43
Gambar 4. 5	Sourcecode python arsitektur CNN	43
Gambar 4. 6	sourcecode python compile arsitektur CNN	46
Gambar 4. 7	Sample data citra dalam bentuk array(a) COVID-19 (b) normal (c) X-train yang telah diubah menjadi range 0-1	47
Gambar 4. 8	Plotting grafik Hasil training Model accuracy	50
<i>Gambar 4. 9</i>	<i>Plotting grafik Hasil training Model Loss</i>	51
Gambar 4. 10	Hasil Confusion Matriks model dalam bentukHeatmap	53

DAFTAR TABEL

Tabel 3. 1	Training Dataset dan Testing dataset	365
Tabel 3. 2	Arsitektur Model CNN	356
Tabel 4. 1	Layers dan layerparameters	48
Tabel 4. 2	Rangkuman perbandingan hasil	55

DAFTAR LAMPIRAN

Lampiran 1: Citra Chest X-Ray	66
Lampiran 2: Citra Chest X-Ray setelah pre-processing	86
Lampiran 3: Algoritma sistem klasifikasi dengan metode CNN	119
Lampiran 4: Daftar Istilah	139
Lampiran 5: Hasil Training model	142
Lampiran 6: Contoh tampilan Confusion matrix 2x2	143
Lampiran 7: Contoh perhitungan Confusion matrix 2x2	144
Lampiran 8: Contoh perhitungan Confusion matrix 2x2	145
Lampiran 9: Riwayat Hidup	146

BAB I

PENDAHULUAN

A. Latar Belakang

Coronavirus Disease-2019 (COVID-19) merupakan penyakit menular akut yang terjadi karena virus corona jenis baru bernama *SARS-Cov2*, yang menginfeksi paru-paru (Ozturk et al., 2020). Pada tahun 2020, COVID-19 telah menjadi pandemi utama karena mudah menyebar melalui udara, kontak dengan benda, serta orang yang terkontaminasi (Cucinotta and Vanelli 2020). Tanda - tanda infeksi yang disebabkan oleh COVID-19 antara lain demam, batuk, kehilangan nafsu makan, dan sesak nafas. Untuk kasus yang lebih serius, Infeksi yang disebabkan oleh COVID-19 dapat menyebabkan pneumonia, gangguan pernafasan akut (ARDS), syok septik, kegagalan organ-organ dalam, atau bahkan kematian (Mahase, 2020).

Penyakit menular meningkat diseluruh dunia. Sejak 2019, terdapat lebih dari 640 juta kasus COVID-19 dan lebih dari 6,6 juta dikonfirmasi meninggal (Rahimi and Talebi Bezmin Abadi 2022). Hal tersebut menunjukkan bahwa penyakit COVID-19 dapat memberikan dampak yang besar bagi kehidupan manusia. Dalam perspektif islam, kematian

atau ajal manusia di tetapkan oleh Allah swt, tetapi pola atau kondisi kematian manusia yang berbeda, satu diantaranya adalah kematian karena penyakit. Sesuai firman Allah SWT pada QS. Al-A'raf:34 tentang kematian manusia.

وَلِكُلِّ أُمَّةٍ أَجَلٌ فَإِذَا جَاءَ أَجْلُهُمْ لَا يَسْتَأْجِرُونَ سَاعَةً وَلَا يَسْتَقْدِمُونَ

Artinya : Dan setiap umat mempunyai ajal (batas waktu). Apabila ajalnya tiba, mereka tidak dapat meminta penundaan atau percepatan sesaatpun. (QS. Al-A'raf:34)

Quraish Shihab berpendapat bahwa ayat di atas menunjukkan upaya untuk memperlambat atau mempercepat datangnya kematian, tidak akan berhasil bahkan meskipun dilakukan dengan sungguh-sungguh. Namun, bukan berarti bahwa upaya peningkatan harapan hidup tidak dilakukan. Manusia dapat berusaha, usahanya dapat berhasil, tetapi keberhasilannya tidak akan mengubah kematian yang telah ditetapkan oleh Allah SWT. Demikian semua sunnatullah atau hukum yang ditetapkan oleh Allah SWT (Shihab, 2010).

Pemerintah telah melakukan berbagai upaya guna menekan laju pertambahan penderita COVID-19 yang relatif sangat cepat. Upaya ini termasuk melacak sumber pasien, mengisolasi kota/negara, serta melakukan tes massal. Sesuai *Standard Operating Procedure* (SOP), saat ini diagnosis COVID-19 harus dilakukan dan dipastikan melalui pengurutan

gen dari sampel pernafasan atau darah, dengan melakukan *Reverse Transcription- Polymerase Chain Reaction* (RT-PCR) menggunakan sampel yang berasal dari swab nasofaring dan orofaring (Approach et al., 2020).

RT-PCR adalah metode standar emas untuk diagnosis COVID-19, namun untuk mendapatkan hasil tes memerlukan waktu lama dan memiliki sensitivitas rendah sekitar 60-70%, sehingga sekarang diagnosis lebih cenderung ke pendekatan orang yang mempunyai pola pneumonia COVID-19 melalui tindakan non-invasif salah satunya *Chest X-Ray* (CXR) (Ozturk et al., 2020). Pencitraan radiologis seperti *Computed Tomography scan* (CT-Scan) dan *Chest X-Ray* (CXR) memiliki peran penting dalam diagnosis dini dan pengobatan penyakit ini karena dapat menggambarkan manifestasi karakteristik tertentu di paru-paru terkait dengan COVID-19 (Apostolopoulos and Mpesiana, 2020).

Karakteristik COVID-19 pada citra radiologi yaitu distribusi bilateral *patchy shadows* dan *Ground Glass Opacity* (GGO) pada tahap awal, serta konsolidasi paru pada tahap akhir (F. Wang et al. 2020). Pada kasus ini, CXR lebih mudah diakses karena tersedia disebagian besar rumah sakit di seluruh dunia dan lebih murah daripada CT-Scan (El-Kenawy et al., 2020). Permasalahan dari penggunaan CXR yaitu untuk

menganalisis citra membutuhkan ahli radiologi serta memerlukan waktu yang signifikan, dimana sangat berharga ketika orang sakit di seluruh dunia. Oleh karena itu, diperlukan sebuah teknologi untuk mengklasifikasi COVID-19 melalui data citra CXR dengan cepat dan tepat.

Klasifikasi COVID-19 menggunakan citra CXR dapat dilakukan dengan *Deep Learning*. *Deep Learning* merupakan metode pembelajaran oleh mesin (*machine learning*) yang terinspirasi oleh struktur otak manusia. Dalam penelitian sebelumnya, para peneliti telah mengembangkan sistem klasifikasi kasus COVID-19 dan SARS berbasis ciri tekstur menggunakan metode *Multi Layer Perceptron* (MLP). Dataset COVID-19 dan SARS yang digunakan masing-masing terdiri dari 12 citra CXR. Sistem yang diusulkan memberikan akurasi yang baik sebesar 91,67% (Sumarti, Kusuma and Azzahra, 2022). Namun teknik klasifikasi MLP memiliki kelemahan ketika input berupa citra. Citra yang di input harus di lakukan *preprocessing*, tersegmentasi, dan di ekstrak untuk memperoleh performa yang optimal. Sehingga untuk dapat mengatasi masalah ini dikembangkan *Convolutional Neural Network* (CNN) (Nurhikmat 2018).

CNN adalah salah satu metode *deep learning* yang dapat digunakan untuk mendeteksi dan mengenali sebuah

objek pada citra digital.. CNN memiliki susunan neuron 3D (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran layer sedangkan kedalaman mengacu pada jumlah layer (S. Wang et al. 2020). Penelitian ini didasarkan pada studi sebelumnya, bahwa metode CNN unggul dalam melakukan pengenalan citra dengan akurasi yang menyaingi manusia pada dataset tertentu sehingga di klaim sebagai algoritma terbaik untuk menyelesaikan masalah *object recognition* dan *detection* (Coates, Lee, and Ng 2011).

Penelitian lain dilakukan oleh Maysanjaya dengan memanfaatkan citra CXR untuk mengklasifikasikan pneumonia menggunakan metode *Convolutional Neural Networks* (CNN) mendapatkan akurasi sebesar 89,58% (Maysanjaya 2020). Selanjutnya, penelitian oleh Linda Wang dengan judul "*COVID-Net: A tailored DEEP Convolutional Neural Network Design for Detection of COVID-19 cases from Chest X-Ray Images*". Dataset yang digunakan berjumlah 13.870 citra. Metode yang digunakan adalah CNN dengan arsitektur COVID-Net. Hasil penelitian didapatkan akurasi sebesar 93,3%, sensitivitas 91,0% serta presisi 98,9% (L. Wang, Lin, and Wong 2020).

Berdasarkan eksplorasi dari penelitian para ahli pada jurnal internasional maupun jurnal nasional, para ahli sudah

banyak yang memanfaatkan CNN. Tetapi penulis belum menemukan penelitian yang berbicara mengenai klasifikasi citra CXR penyakit COVID-19 dan normal dengan metode CNN menggunakan *python*. *Python* merupakan bahasa pemrograman populer yang digunakan di seluruh dunia untuk menyederhanakan situs web, algoritme, dan proses otomatisasi. Melalui bahasa pemrograman *python*, setiap program menjadi lebih ringkas dibandingkan dengan Bahasa pemrograman lainnya. Pemanfaatan bahasa pemrograman *python* dipilih karena mudah dipahami, telah terdapat *package library*, dan dapat diakses dengan gratis.

Penelitian ini bertujuan untuk membuat model *machine learning* "**Klasifikasi Citra *Chest X-Ray* (CXR) Penyakit COVID-19 dan Normal dengan Metode *Convolutional Neural Network* (CNN) menggunakan *Python***". Penelitian tentang COVID-19 masih tergolong baru, sehingga masih diperlukan penyempurnaan arsitektur dengan data yang terus berkembang. Perbaikan pada citra input diharapkan mampu meningkatkan akurasi proses CNN. Hasil penelitian ini diharapkan dapat memberikan bantuan dalam skrining yang lebih baik sebagai upaya solusi penanganan kasus yang efektif dan efisien dengan mengklasifikasikan citra CXR penyakit COVID-19 dan normal, serta sebagai acuan pengembangan dan aplikasi CNN.

B. Rumusan Masalah

1. Bagaimana proses klasifikasi citra CXR penyakit COVID-19 dan normal dengan metode CNN menggunakan *python*?
2. Berapa akurasi yang diperoleh pada model CNN untuk mengklasifikasi citra CXR penyakit COVID-19 dan normal?

C. Tujuan Penelitian

1. Untuk mengetahui proses klasifikasi citra CXR penyakit COVID-19 dan normal dengan metode CNN menggunakan *python*.
2. Untuk mengetahui akurasi yang diperoleh pada model CNN yang mampu mengklasifikasi citra CXR penyakit COVID-19 dan normal.

D. Manfaat Penelitian

1. Sebagai alternatif lain untuk mengidentifikasi suatu citra CXR.
2. Sebagai perkembangan IPTEK tentang citra CXR COVID-19 dan normal.
3. Sebagai perkembangan IPTEK tentang metode CNN.
4. Sebagai bahan pertimbangan bagi dokter radiologi diagnostik maupun klinisi untuk memprediksi serta mengklasifikasi COVID-19 berdasarkan citra CXR.

E. Batasan Masalah

1. Objek penelitian hanya dibatasi pada citra *Chest X-Ray* (CXR) dengan jumlah data keseluruhan sebanyak 200 citra.
2. Citra CXR diklasifikasikan menjadi penyakit COVID-19 dan normal dengan metode CNN menggunakan *software python*.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. COVID-19

COVID-19 (*Coronavirus Disease-2019*) merupakan peradangan pada parenkim paru yang bersifat menular. Penyakit tersebut terjadi karena adanya virus corona jenis baru bernama SARS-CoV-2 (*Severe acute respiratory syndrome coronavirus 2*), yang menginfeksi paru-paru dan sistem pernapasan (Ozturk et al. 2020). Saat ini *Coronavirus* yang dapat menginfeksi manusia yaitu dua *alphacoronavirus* (229E dan NL63) serta empat *betacoronavirus* (OC43, HKU1, MERS-CoV dan SARSCoV) (Huang et al. 2020; Z. Wang, Qiang, and Ke 2020). SARS-CoV-2 dikelompokkan dalam genus *betaCoronavirus* (Li et al. 2020).

Sekuens genom SARS-CoV-2 pertama diidentifikasi dengan lima subsekuens dari sekuens genom virus yang dipublikasikan pada 10 Januari 2020,. Sekuens genom novel coronavirus (SARS-CoV-2) diketahui mirip dengan MERS-CoV dan SARS-CoV.

Pohon evolusi serupa dengan MERS-CoV dan SARS-CoV tetapi tidak terlalu identik (Burhan et al. 2020).

Epidemi di Wuhan mirip dengan wabah SARS di Guangdong pada tahun 2002. Keduanya terjadi di musim dingin. Pneumonia COVID-19 cenderung memiliki tingkat kematian yang lebih rendah dibandingkan dengan SARS. Angka kematian akibat SARS mencapai 10% dan MERS sebesar 37%. Tetapi, saat ini diketahui bahwa tingkat infektivitas virus pneumonia COVID-19 setidaknya sama atau lebih besar daripada SARS-CoV. Hal ini direpresentasikan dengan R_0 (tingkat penularan). Menurut penelitian terbaru, R_0 virus pneumonia SARS-CoV-2 ini adalah 4,08. Sebagai perbandingan, SARS-CoV memiliki R_0 2,0 (Burhan et al. 2020). COVID-19 bersifat mematikan akan tetapi tingkat kematian masih belum pasti, dan saat ini masih dapat dikendalikan dan dicegah (Adu and Mas'amah 2021).

Infeksi COVID-19 dapat menyebabkan gejala ringan, sedang, atau berat. Gejala utama yang muncul antara lain adalah batuk, demam, dan dispnea. Gejala tambahan termasuk sakit kepala, nyeri otot, lemas, diare dan batuk darah. Untuk kasus yang lebih serius, infeksi yang disebabkan oleh COVID-19 dapat

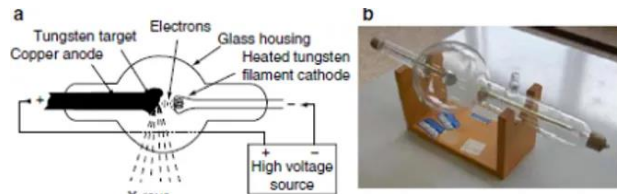
menyebabkan pneumonia, syok septik, kegagalan organ dalam, gangguan pernafasan akut (*Severe Acute Respiratory Infection-SARI*), atau bahkan kematian. Definisi SARI yaitu infeksi saluran pernapasan akut yang memiliki riwayat demam (suhu $\geq 38\text{ }^{\circ}\text{C}$) dan batuk yang terjadi dalam 14 hari terakhir dan memerlukan rawat inap. Pada beberapa pasien, gejala yang muncul ringan, bahkan tidak disertai dengan demam. (Perhimpunan Dokter Paru Indonesia 2020).

2. *Chest X-Rays*

Chest X-Rays (CXR) adalah proyeksi radiografi dengan memanfaatkan radiasi *X-Ray* untuk mengetahui kondisi anatomi di dalam *thoraks* (dada) termasuk organ jantung dan paru-paru (Risnawati, Ali, and Tubagus 2015). *X-Ray* pertama kali ditemukan oleh Wilhem Conrad Roentgen tahun 1895 pada percobaan tabung yang dibungkus dengan suatu kertas hitam untuk mencegah kebocoran fotoluminesensi dari dalam tabung ke luar, kemudian ia membuat ruang penelitian menjadi gelap. Saat menghasilkan sinar katoda, dia mengamati sesuatu yang tidak terduga. Pelat fotoluminesensi di atas meja mulai berpendar didalam kegelapan. Walaupun

berjarak 1 meter dari tabung, pelat masih tetap berpendar (Susilo, Pratiwi 2013).

X-Ray merupakan gelombang elektro magnetik energi yang dimiliki antara 200 eV hingga 1 MeV dengan panjang gelombang berkisar 0,5 Å sampai 2,5 Å. *X-Ray* dihasilkan dari tumbukan elektron berkecepatan tinggi dengan logam target. Oleh karena itu, tabung *X-Ray* harus memiliki suatu sumber elektron, voltase tinggi, dan target logam. Selanjutnya, elektron-elektron yang bertumbukan melambat dengan cepat. Energi tersebut diubah menjadi panas dan juga foton (Ramdhan, 2015).



Gambar 2. 1 (a) Representasi Skema Tabung *X Ray* (b) Foto tabung *X-Ray*

(Maula 2021)

Berkas *X-Ray* yang menembus materi, dalam hal ini tubuh, akan teratenuasi sehingga mengurangi intensitas *X-Ray* yang keluar. Intensitas gelombang yang masuk berbanding secara logaritmik terhadap

atenuasi gelombang *X-Ray* (Sumarni and Mart 2000).
Hal ini ditunjukkan dari persamaan :

$$I = I_0.e^{-\mu x} \quad (2.1)$$

Keterangan:

I = intensitas foton yang diteruskan (W/m^2)

I_0 = intensitas foton yang datang (W/m^2)

e = epsilon = 1 inv ln = 2,718281828

μ = koefisien atenuasi linier (W/m)

x = tebal materi yakni badan. (m)

Bidang radiologi diagnostik telah berkembang pesat sejak ditemukan *X-Ray* pada tanggal 9 November tahun 1895. Penggunaannya sebagai alat bantu diagnostik meningkat sekitar 5%-10% setiap tahun. Sekitar 80% dari aktivitas radiologi adalah pemeriksaan *thorax* (Kamaruddin 2010). Kemajuan yang pesat dalam teknik pemeriksaan radiologi *thoraks* dan pengetahuan untuk menilai suatu rontgenogram *thoraks* dengan *X-Ray* selama dasawarsa terakhir menjadi suatu keharusan rutin yang terus diminati.

Pemeriksaan paru-paru tanpa CXR saat ini dianggap tidak memadai. Suatu penyakit belum dapat disingkirkan secara pasti sebelum pasien melakukan pemeriksaan radiologik (Risnawati, Ali, and Tubagus 2015). Hal tersebut menunjukkan betapa pentingnya pemeriksaan ini. Pemeriksaan paru-paru umumnya menggunakan CXR, karena pemeriksaan ini mudah dilakukan, relatif cepat, serta lebih murah dibandingkan pemeriksaan MRI maupun *CT-Scan*.

CXR normal memiliki bentuk citra yang cukup jelas sehingga terlihat mana yang merupakan paru-paru, jantung dan diafragma, paru-paru tidak memiliki bercak, tekstur citra halus, serta bentuk dan ukuran paru-paru tetap. Sementara itu jika pada citra terdapat bercak putih karena terdapatnya cairan pada paru, sudut carina berada diluar batas normal, sinus prenicocostalis yang tidak tajam maka citra paru-paru dikatakan abnormal. Bentuk dan ukuran citra paru berubah atau tidak terlihat dari intensitas piksel pada objek (paru-paru) (Winangun, Widyantara, and Hartati 2020). Contoh hasil citra CXR yang digunakan dalam penelitian ini ditunjukkan pada Gambar 2.1.



Gambar 2. 2 Citra Chest X-rays

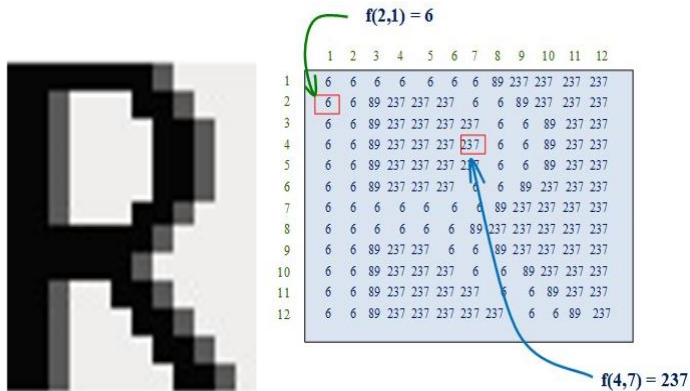
(<https://github.com/ieee8023/covid-chestxray-dataset>)

3. Pengolahan Citra Digital

Secara umum, pengolahan citra digital mengacu pada pengolahan citra dua dimensi oleh komputer (Putra, 2009). Pengolahan citra digital dapat diaplikasikan secara luas. Salah satu penerapannya yakni dalam adalah bidang kedokteran. Penggunaan olah citra digital digunakan untuk mengurangi derau, pengaturan kontras, serta klasifikasi citra.

Sebuah citra digital dapat direpresentasikan sebagai fungsi dari intensitas $f(y, x)$, dimana x (baris) dan y (kolom) adalah koordinat posisi dan $f(y, x)$ adalah nilai fungsi (y, x) yang menunjukkan besarnya

intensitas citra atau tingkat keabuan atau warna dari piksel titik tersebut (Andono, T.Sutojo, & Muljono, 2017).



Gambar 2. 3 Representasi citra dan intensitas piksel citra
(Ermawati 2020)

Gambar 2.3 menunjukkan contoh penotasian $f(y, x)$. Berdasarkan gambar tersebut maka $f(4,7)$ bernilai 237 dan $f(2,1)$ bernilai 6. Pada citra *grayscale* bernilai 237 atau 6 disebut sebagai aras intensitas yang nilainya antara 0 adalah hitam sampai dengan 255 adalah putih. Jenis citra dibagi menjadi tiga, yaitu citra citra biner, citra *grayscale*, dan citra RGB.

Citra biner, setiap piksel hanya memiliki dua kemungkinan nilai intensitas yaitu 1 (putih) dan 0 (hitam), sedangkan citra *grayscale* adalah citra yang

memiliki jumlah warna 256 berada pada rentang antara 0 sampai 255. Proses konversi secara matematis dapat dilihat dalam persamaan 2.2 :

$$I = \frac{R + G + B}{3} \quad (2.2)$$

Dimana I merupakan tingkat keabuan, R merupakan tingkat intensitas merah, G merupakan tingkat intensitas hijau, dan B merupakan tingkat intensitas biru. Selanjutnya citra warna yang tiap piksel mempunyai 3 komponen warna yaitu RGB (*Red, Green, Blue*) dengan variasi intensitas 0 sampai 255 pada masing-masing komponen. Jumlah warna RGB adalah sejumlah 16.777.216 variasi warna (Purnomo, 2010)

4. *Contrast Limited Adaptive Histogram Equalization* (CLAHE)

Contrast Limited Adaptive Histogram Equalization (CLAHE) termasuk teknik perbaikan kualitas citra yang digunakan untuk meningkatkan kontras citra. CLAHE adalah generalisasi dari AHE (*Adaptive Histogram Equalization*) yang bekerja di

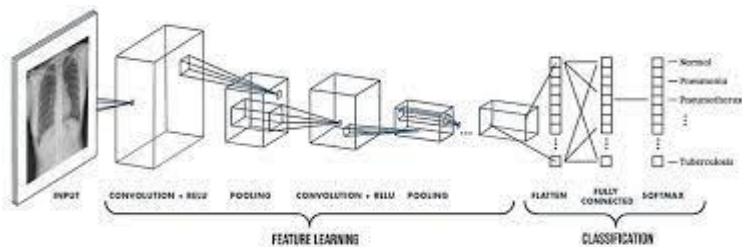
area kecil dari citra *grayscale* yang disebut dengan *tile*. Kontras pada setiap file disesuaikan sehingga histogram yang dihasilkan dari wilayah tersebut kira-kira cocok dengan bentuk histogram yang ditentukan. *Tile* piksel yang saling berdekatan dihubungkan oleh interpolasi bilinear. Hal ini dilakukan agar kombinasi *tile* yang dihasilkan halus (Matondang 2018). Kontras, terutama di area yang homogen, dapat dibatasi untuk menghindari amplifikasi *noise* pada citra.

5. *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah evolusi dari *Multi Layer Perceptron* (MLP) yang dirancang untuk memproses data berbentuk citra dimana cara kerjanya terinspirasi oleh kerja sistem syaraf pada manusia (Eka Putra 2016). CNN merupakan algoritma yang mempelajari suatu citra *input*-an dengan memperhitungkan nilai bobot dan bias sehingga dapat dikelompokkan. Secara umum, tahapan pada proses CNN terbagi menjadi dua yaitu tahap *feature learning* (pembelajaran fitur) dan tahap klasifikasi terdapat pada gambar 2.4.

Arsitektur CNN sangat sederhana, terdiri atas satu lapis masukan (*input layer*), satu lapis keluaran

(*Output layer*), dan sejumlah lapis tersembunyi (*hidden layer*). Lapis tersembunyi umumnya berisi 4 layer utama (Fachrozy 2020), yaitu seperti pada gambar 2.4:



Gambar 2. 4 Ilustrasi arsitektur CNN secara umum

(Gunawan and Setiawan 2022)

Convolutional Layer merupakan sebuah fitur terpenting dari CNN, layer ini berisi *script filtering* yang dapat digunakan untuk mempelajari citra input. Pada lapisan ini, operasi konvolusi dapat dilakukan baik dari input ataupun output dari layer sebelumnya. Konvolusi adalah istilah matematika yang berarti menerapkan fungsi ke output fungsi lain berulang-ulang. Konvolusi terhadap citra memiliki tujuan untuk mengekstraksi fitur dari citra input. Hasil dari konvolusi yaitu transformasi linear dari data input berdasarkan informasi spasial pada data. Bobot pada

layer tersebut menjelaskan dengan detail kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN (Eka Putra 2016).

a. *Max Pooling*

Max Pooling merupakan proses peningkatan invariansi posisi dari fitur (*grid*) menggunakan operasi Max (Fachrozy 2020). *Max Pooling* membagi output dari *Convolutional Layer* menjadi beberapa *grid* kecil kemudian mengambil nilai maksimum dari masing-masing *grid* untuk membangun matriks citra yang direduksi seperti pada gambar 2.5. *Grid* berwarna merah, hijau, kuning dan biru adalah kelompok *grid* yang nilai maksimumnya akan dipilih. Sehingga hasil dari proses *max pooling* dapat dilihat pada kelompok *grid* di sebelah kanan. Proses ini memastikan bahwa fitur yang akan diperoleh tetap sama meskipun objek citra mengalami pergeseran (translasi). Hal ini bertujuan untuk membantu mengurangi jumlah parameter dan waktu komputasi yang dibutuhkan saat melatih sistem.

b. *ReLU*

Rectified Linear Units (ReLU) merupakan fungsi aktivasi yang paling umum digunakan dalam *deep learning*. Sehingga, dapat ditulis dengan rumus sebagai berikut (Dios Kurniawan, 2020).

$$f(x) = \max(0, x) \quad (2.4)$$

Fungsi tersebut mengembalikan nilai 0 jika menerima input negatif, tetapi setiap nilai positif nilai x akan dikembalikan ke nilai aktivasi itu sendiri. Selain itu, pada penelitian ini, penulis juga menggunakan fungsi aktivasi *softmax* dimana lapisan ini menghubungkan antara *fully connected* dengan *dense connection*.

c. *Full Connected Layer*

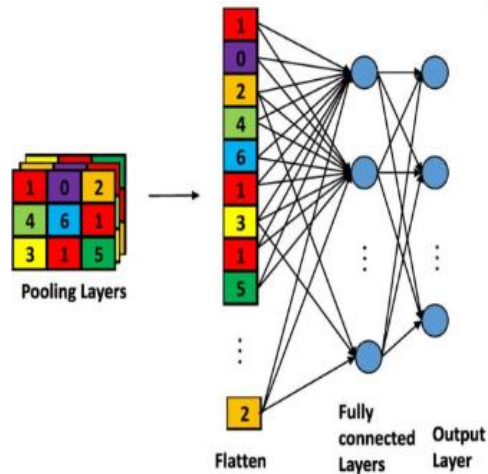
Fully Connected Layer merupakan *layer* yang biasa digunakan dalam aplikasi MLP yang bertujuan untuk keperluan transformasi dimensi data sehingga data dapat diklasifikasikan secara linear (Eka Putra 2016). Pada *Fully Connected Layer* terdapat proses *flattened* atau proses dimana *output* dari *layer* sebelumnya akan diubah menjadi 1 dimensi atau

vektor(Azizah *et al.*, 2021) Seperti pada gambar 2.6. Vektor tersebut akan terhubung dengan persamaan sebagai berikut:

$$f_{cj} = b_j + \sum_i x_i \omega_{ij}, j = 1,2,3, \dots, t \quad (2.5)$$

Di mana **b** merupakan nilai vector bias, ω adalah bobot, t merupakan jumlah kelas, dan x adalah *vector input*. Optimasi pembaruan bobot dari CNN menggunakan *Adaptive Moment Estimation* atau ADAM. Metode ini dipilih karena waktu yang digunakan dalam proses komputasi relatif efisien. Berikut formulasi untuk menghasilkan pembaharuan bobot dapat dilihat pada persamaan 2.5: dengan $\omega^{(t+1)}$ merupakan nilai bobot baru, $\omega^{(t)}$ nilai bobot lama, η *batch size*, ϵ epsilon untuk menghindari kondisi *division by zero*.

$$\omega^{(t+1)} \leftarrow \omega^{(t)} - \frac{\eta \hat{m}_\omega}{\sqrt{\hat{v}_\omega} + \epsilon} \quad (2.6)$$



Gambar 2. 5 Proses Pada Tahap Klasifikasi

(Azizah et al. 2021)

6. Confusion Matrix

Confusion matrix adalah matriks yang berisi informasi tentang hasil prediksi klasifikasi dan data aktual yang dilakukan oleh sistem klasifikasi (Andono, T.Sutojo, & Muljono, 2017). Terdapat empat istilah pada pengukuran performa menggunakan *confussion matrix* antara lain:

- True Positive* (TP) adalah jumlah hasil klasifikasi benar dari kelas aktual positif.
- True Negative* (TN) adalah jumlah hasil klasifikasi benar dari kelas aktual negatif.

- c. *False positive* (FP) adalah jumlah hasil klasifikasi salah dari kelas aktual positif.
- d. *False Negative* (FN) jumlah hasil klasifikasi salah dari kelas aktual negative

Pengukuran performa sistem klasifikasi merupakan hal yang penting untuk menjelaskan seberapa bagus sistem yang diusulkan dalam mengklasifikasi data. Penentuan baik atau tidaknya performa suatu model klasifikasi dapat diamati dari parameter pengukuran performanya, yaitu Presisi, Recall dan akurasi.

Presisi merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *Recall* atau sensitivitas merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Akurasi merupakan tingkat kedekatan antara nilai prediksi nilai aktual, yang kemudian dihitung nilai-nilai evaluasi menggunakan persamaan berikut:

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2.7)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.8)$$

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.9)$$

7. Python

Python merupakan bahasa nomor satu paling populer yang digunakan untuk *Machine Learning* dan *Artificial Intelligence* (Dios Kurniawan, 2020). Popularitas *python* ini salah satunya adalah karena tersedianya banyak library (modul-modul siap pakai) yang mendukung kekuatan *Data Analytic* dan *Machine Learning*, misalnya *Matplotlib*, *Numpy*, *pandas*, *Sickit-learn*, dan library tambahan *python* lainnya. Selain itu *software python* memiliki lisensi yang bebas biaya. Itu pula alasan utama mengapa penelitian ini dibuat agar semua orang dapat mempelajari model CNN menggunakan *python* tanpa hambatan biaya

Source code (kode program) yang ditulis pada *python* akan dijalankan baris-per-baris oleh *python Interpreter* (Az-zahra, 2019). Ada pula *python Interpreter* yang dibuat oleh *Ipython project*, yang bertujuan mempermudah pekerjaan programmer dibanding dengan menggunakan *python interpreter* standar. *Ipython project* menyediakan fasilitas Jupyter Notebook. Fasilitas ini berbasis web, jadi peneliti bisa melakukan semuanya melalui web browser. *Source code* bisa disimpan di server, tidak disimpan di disk computer.

Python digunakan untuk pengembangan perangkat lunak di perusahaan dan organisasi seperti Google, Yahoo, CERN, Industrial Light and Magic, dan NASA (Hellmann, 2011). Tim yang bekerja di bagian riset atau analisis bisa menggunakan bahasa yang sama dengan pengembang dan *programmer*. Hal ini tentunya membuat peningkatan produktivitas karena hasil kerja bisa segera diterapkan untuk keuntungan organisasi atau perusahaan.

B. Kajian Pustaka

Para peneliti berupaya mengurangi jumlah penularan COVID-19 dengan berfokus pada pengembangan vaksin, penggunaan PCR, dan sistem yang memanfaatkan citra digital. Penelitian tersebut diantaranya dilakukan oleh Ai *et al.*, (2020) CNN berbasis model ResNet50, InceptionV3, dan Inception-ResNetV2 untuk klasifikasi citra *Chest X-Ray* untuk mendeteksi kelas normal atau COVID-19. Hasil penelitian menunjukkan korelasi yang baik antara *CT-scan* dan pendekatan PCR.

Penelitian berikutnya oleh Gozes *et al.*, (2020) mengusulkan alat analisis citra *CT-scan* otomatis berbasis AI untuk mendeteksi COVID-19. Pada studi mereka mengembangkan sebuah sistem dengan model *deep*

learning untuk segmentasi citra otomatis dan mendeteksi citra yang terinfeksi. Hasil klasifikasi yang telah dilakukan mencapai tingkat akurasi 95%, sensitivitas 98,2%, dan spesifisitas 92,2%. Penelitian ini menyatakan bahwa analisis citra berbasis AI memiliki performa bagus meskipun citra yang digunakan dalam penelitian masih sedikit.

Penelitian mengenai sistem yang memanfaatkan citra digital juga dilakukan oleh Sumarti (2022). Mereka menggunakan 24 citra CXR untuk mengklasifikasikan COVID-19 dan SARS berdasarkan ciri tekstur. Model yang digunakan *Multi Layer Perceptron* (MLP). Penelitian ini menghasilkan tingkat akurasi 91,67% dengan *specifity* 91,67% dan *sensitivity* 91,67%.

Penelitian mengenai klasifikasi citra pneumonia pada CXR paru-paru telah dilakukan oleh Maysanjaya (2020). Dalam penelitian tersebut menyatakan bahwa metode CNN mampu dimanfaatkan untuk mengklasifikasikan pneumonia pada citra CXR paru-paru. Hasil penelitian menunjukkan bahwa metode yang dilakukan mampu mencapai akurasi sebesar 89,58%.

Studi lain pada Linda Wang (2020) dilakukan riset terhadap *dataset* citra radiography *open source* CXR

yang disebut COVIDx, dan Arsitektur NN yang disebut COVID-Net dirancang untuk mendeteksi COVID-19. Dataset dikategorikan menjadi tiga kelas yaitu *X-Ray* normal (tanpa infeksi), Infeksi non-COVID-19, dan *X-Ray* COVID-19. Hasil penelitian menunjukkan bahwa COVID-net mencapai akurasi pengujian yang baik sebesar 93,3%, *sensitivity* 91,0% dan presisi 98,9 %.

C. Rumusan Hipotesis

Berdasarkan deskripsi teori dan kajian Pustaka, maka dengan *software python* pada proses klasifikasi menggunakan metode CNN dapat mengklasifikasikan citra CXR menjadi kategori COVID-19 dan normal, dengan semakin banyaknya *epoch* yang digunakan maka akurasi yang dihasilkan dari proses klasifikasi semakin tinggi.

BAB III

METODOLOGI PENELITIAN

A. Waktu dan Tempat Penelitian

Penelitian yang dilakukan penulis dilaksanakan di Semarang, dengan kurun waktu antara bulan desember 2020 sampai desember 2021 untuk proses pengambilan dan pengolahan data.

B. Alat Penelitian

Alat yang digunakan dalam penelitian ini meliputi:

1. Laptop dengan spesifikasi *processor intel(R) Core (TM) i5-5300U CPU @2.30GHz 2.29 GHz, RAM 8GB, Disk: 256 GB, dan sistem operasi windows 10 Pro 64 bit.*
2. Bahasa Pemrograman python versi 3.8.
3. *platform anaconda navigator* untuk menginstal *package* yang dibutuhkan.
4. *Jupyter notebook* untuk membuat kode..
5. Microsoft word tahun 2010 untuk menyusun lapran penelitian..

C. Bahan Penelitian

Bahan yang digunakan dalam penelitian ini berupa citra *Chest X-Ray* (CXR) dengan total 200 citra yang bersumber dari dataset baru (Pham 2021). Penentuan jumlah data berdasarkan rumus:

$$n = \frac{N}{N(d^2) + 1}$$

Keterangan :

n = Sampel

N = Populasi

d = Derajat Kebebasan (1%, 5%, atau 10%)

Citra COVID-19 sebanyak 100 data dan citra normal sebanyak 100 data. Citra CXR gabungan diambil dari situs terpercaya dibawah lisensi CC BY-NC-SA 4.0 yakni <https://github.com/ieee8023/covid-chestxray-dataset> dan <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>. Hasil diagnosa berdasarkan hasil tes RT-PCR. Semua data dikategorikan citra RGB dengan format jpg.

D. Prosedur Penelitian

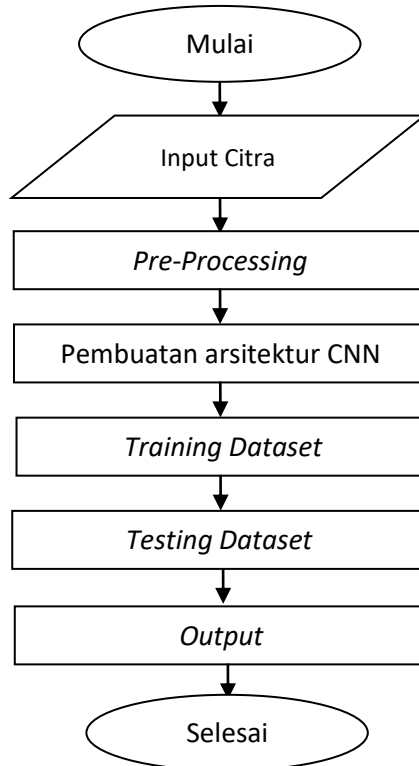
Penelitian dilakukan melalui tiga tahap utama yaitu, persiapan, pengolahan citra, serta penulisan hasil penelitian dengan menggunakan Microsoft word.

1. Tahap I : Tahap Persiapan

Tahap Persiapan dilakukan untuk memahami permasalahan dan metode yang akan digunakan. Pada tahap ini, dilakukan pengumpulan data citra paru-paru COVID-19, dan citra paru-paru normal yang kemudian data diformulasikan dengan menggabungkan dua kelas yang masing-masing berjumlah 100 citra. Setelah dataset terkumpul, langkah selanjutnya menyiapkan perprograman *python* untuk pengolahan citra.

2. Pengolahan Citra

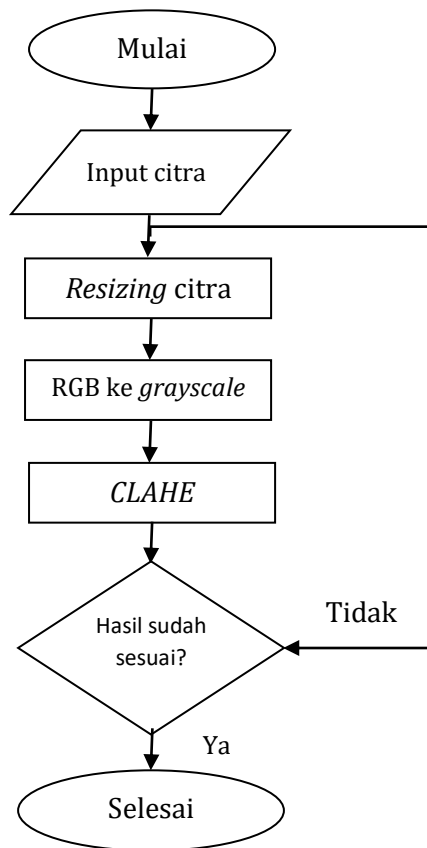
Tahapan pengolahan data ini dilakukan melalui beberapa pemrosesan yang dapat dilihat pada skema pengolahan citra.



Gambar 3. 1 Skema sistem klasifikasi COVID-19

Tahap ini dimulai dengan *Preprocessing*, yang terdiri dari proses *scaling* yaitu mengubah ukuran piksel citra. Kemudian dilakukan penyeragaman citra ke aras keabuan (*grayscale*). Selanjutnya, dilakukan proses perbaikan citra dengan menghapus *noise* dari citra CXR agar tidak terjadi kesalah pahaman dalam

mendiagnosa sebuah penyakit. Teknik yang digunakan pada proses ini adalah *Contrast Limited Adaptive Histogram Equalization (CLAHE)* bertujuan untuk menaikkan kontras sebuah citra agar fitur-fitur semakin jelas.



Gambar 3. 2 Skema prosedur preprocessing citra

Tahap kedua adalah pembuatan arsitektur CNN. Arsitektur CNN yang akan digunakan menggunakan 2 proses konvolusi “Conv” dan 2 proses pooling layer “Maxpool” seperti pada tabel 3.2. Setiap proses konvolusi digunakan aktivasi fungsi ReLU. Selanjutnya dilakukan proses *flattening*, yakni hasil akhir dari *max pooling* di transformasikan menjadi *array* satu dimensi. Tujuan dilakukannya proses ini adalah untuk memastikan bahwa data tidak kehilangan informasi spasial dan *reversible*. Proses selanjutnya adalah “FC” *fully connected layer*.

Jumlah neuron yang digunakan 128 dengan menggunakan aktivasi fungsi ReLU. *Output* berjumlah 2 node tergantung jumlah kelas yang diharapkan. Aktivasi yang digunakan pada output *Softmax* yaitu memilih probabilitas terbesar.

Penentuan model terbaik dilakukan dengan mencari nilai tertinggi untuk parameter dalam model CNN. Parameter yang dimaksud seperti ukuran kernel, *learning rate* dan Jumlah *epoch*. Tujuan penentuan parameter model ini adalah membandingkan model mana yang terbaik dengan memperhatikan nilai parameternya.

Tahap keempat *training* model CNN. Setelah model melewati fase parameterisasi, model akan dilatih dengan citra CXR yang telah dikelompokkan kedalam *training* dataset. Selanjutnya, metode optimasi yang digunakan adalah *adaptive moment estimation* atau ADAM yang berfungsi untuk update bobot, sehingga didapatkan bobot yang dapat meminimumkan *loss function*. *Loss function* yang dipilih kategori *Cross-entropy*.

Tabel 3. 1 Arsitektur Model CNN

Layers	Deskripsi	Nilai
Input Layer	Input Citra	150x150x1
Hidden 1	Conv1	32 kernel berukuran 3x3
	ReLu	Dengan Stride 2
	Maxpool1	Ukuran kernel 2x2
Hidden 2	Conv2	32 kernel berukuran 3x3
	ReLu	Dengan Stride 2
	Maxpool2	Ukuran kernel 2x2
Classsificatin Layer	FC	2 Node

Tahap ketiga yaitu pembagian input citra yang terbagi menjadi dua jenis. Dua jenis tersebut terdiri dari 75% citra yang telah dikumpulkan sebagai *training* dataset dan 25 % dari citra yang telah dikumpulkan sebagai *testing* dataset. Dataset yang

telah dibagi menjadi *training* dataset dan *testing* dataset untuk klasifikasi penyakit ditunjukkan pada Tabel 3.1

Tabel 3. 2 Training Dataset dan Testing dataset

<i>Dataset</i>	<i>Training Dataset</i>	<i>Testing Dataset</i>	Jumlah Data
COVID-19	75	25	100
Normal	75	25	100
Total	150	50	200

Tahap ke lima, yakni tahap *Testing*. Pada tahap ini sistem diuji dengan parameter *epoch* yang berbeda-beda tetapi menggunakan parameter *learning rate* yang sama. Jumlah *Epoch* yang digunakan dalam penelitian ini adalah 25 *epoch*. *Dataset* dibagi per *batch* dengan batch size tertentu dengan melihat jumlah sampel yang ada untuk menyederhanakan dan mempercepat proses *training*. *Epoch* yang diuji mulai dari *epoch* 1 hingga *epoch* 25. Eksperimen ini dilakukan untuk

menemukan jumlah *epoch* terbaik, sehingga dapat memperoleh nilai akurasi yang seimbang antara proses pelatihan dan pengujian. Kemudian, model CNN dengan nilai akurasi terbaik pada *testing dataset* akan dipilih untuk di evaluasi guna melihat performa setiap model dalam memprediksi masing-masing data.

Tahapan terakhir pada prosedur penelitian adalah Output. Output dari proses klasifikasi berupa *confusion matrix* yang dapat digunakan untuk melakukan perhitungan indeks pengukuran pada proses klasifikasi.

E. Analisis Data

Analisis data dilakukan pada proses klasifikasi. Analisis data bertujuan untuk evaluasi pengukuran keakuratan hasil klasifikasi suatu *classifier*. Penentuan performa baik atau tidaknya suatu model klasifikasi dengan melihat dari perhitungan indeks pengukurannya yaitu *accuracy* (persamaan 2.7), *recall* (persamaan 2.8), dan presisi (persamaan 2.9). Pengukuran performa dari klasifikator berdasarkan hasil *Confusion matrix*.

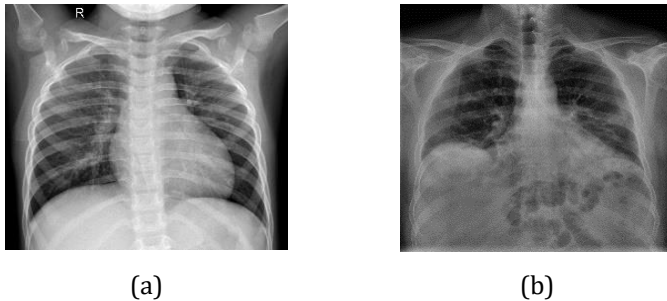
BAB IV

HASIL DAN PEMBAHASAN

Penelitian ini merupakan penelitian yang bertujuan untuk melakukan pengujian klasifikasi hasil citra *Chest X-Ray* (CXR) menjadi dua kategori yaitu citra CXR COVID-19 dan normal, dengan metode *Convolutional Neural Network* (CNN) menggunakan *software python*. Selanjutnya, dalam bab ini akan membahas keseluruhan dari kerja sistem dan pengujian sistem sesuai dengan analisis data dan perancangan yang telah dibahas pada bab 3.

A. Dataset Citra

Tahapan awal yaitu pengumpulan citra yang digunakan sebagai input dalam proses klasifikasi pada sistem. Jumlah keseluruhan data yang digunakan sebanyak 200 citra. Dimana terdiri dari masing-masing 100 citra CXR normal dan citra COVID-19 yang terlampir pada lampiran 1. Citra CXR yang digunakan dalam penelitian ini adalah RGB dalam format (.jpg) dan (.png). Sampel dari citra CXR normal dan COVID-19 ditunjukkan pada gambar 4.1.



Gambar 4. 1 Citra CXR paru-paru (a) Normal (b) COVID-19

B. Hasil *Preprocessing* citra

Preprocessing citra bertujuan untuk memperbaiki kualitas citra sebelum proses konvolusi. Langkah pertama dalam *preprocessing* adalah *resize* agar keseluruhan isi citra konsisten yakni 150x150 piksel. Fokus utama *resize* adalah membaca data karakteristik piksel pada citra, sehingga mendeteksi perbedaan karakter pada setiap citra.

Tahap selanjutnya, citra yang semula bertipe RGB (warna) diubah menjadi citra keabuan (*grayscale*). Hal ini dilakukan karena gambar berwarna memiliki variasi warna yang lebih banyak dan membutuhkan waktu lebih lama untuk sistem bekerja. Oleh karena itu, diperlukan citra dengan aras keabuan untuk menyederhanakan sistem. Meskipun secara kasat mata tidak ada perbedaan, hal tersebut dikarenakan citra dengan aras keabuan

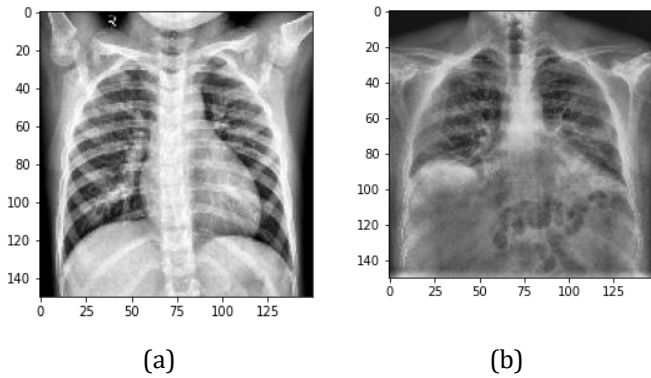
hanya memiliki satu kanal warna untuk setiap pikselnya. Setelah dilakukan *grayscale*, teknik yang digunakan berikutnya adalah *Contrast Limited Adaptive Histogram Equalization (CLAHE)*. Cara bekerja CLAHE dengan menaikkan kontras sebuah citra agar fitur-fitur semakin jelas.

```
# Panggil dataset
datadir= 'C:/Users/ThinkPad X250/Documents/Python Skripsi/Citra untuk skripsi'
file= 'C:/Users/ThinkPad X250/Documents/Python Skripsi/hasil'

label = 0
im_arr = []
lb_arr = []
X = []
y = []
for i in os.listdir(datadir): #Loop all directory
    count = 0
    for pic in glob.glob('C:/Users/ThinkPad X250/Documents/Python Skripsi/Citra untuk skripsi/'+i+'/*'):
        #for pic in glob.glob('C:/Users/ThinkPad X250/Documents/dataset/'+i+'/*'):
            img = cv2.imread(pic)
            img = cv2.resize(img, (150,150))
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
            clahe= clahe.apply(gray)
            count = count + 1
            X.append(clahe)
            y.append(label)
            if(count <= 10):
                im_arr.append({str(i):clahe})
    print("Jumlah "+str(i)+" : "+str(count))
    label = label + 1
    lb_arr.append(i)
X = np.array(X)
y = np.array(y);
```

Gambar 4. 2 sourcecode python preprocessing citra

Proses *preprocessing* citra dituliskan dalam program *python* seperti pada gambar 4.2. Kemudian dari perintah pada *sourcecode* tersebut, menghasilkan keluaran seperti pada gambar 4.3.

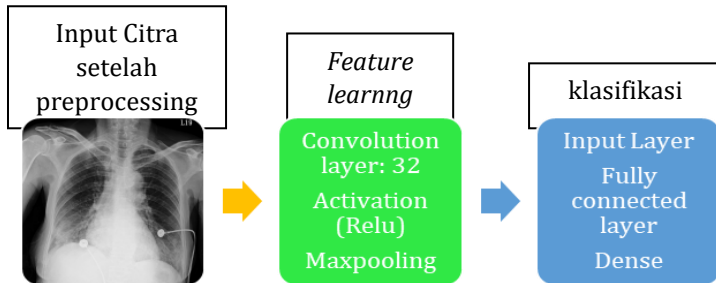


Gambar 4. 3 Citral CXR paru-paru setelah *preprocessing*

(a) normal (b) COVID-19

C. Hasil arsitektur CNN

Teknik klasifikasi yang digunakan dalam penelitian ini adalah CNN. Citra yang sudah di *preprocessing* akan dimasukkan ke dalam mesin klasifikasi CNN. Ilustrasi arsitektur CNN yang digunakan dalam klasifikasi ini dapat dilihat pada gambar 4.3.



Gambar 4. 4 Hasil sitem arsitektur CNN

Berdasarkan gambar 4.4 arsitektur CNN dibagi menjadi 2 bagian besar yaitu *feature learning* dan klasifikasi. *feature learning* terdiri dari konvolusi, ReLu, dan *pooling layer*. Arsitektur yang telah direncanakan dituangkan kedalam Bahasa pemrograman *python* seperti pada gambar 4.5. CNN merupakan *supervised neural network* karena metode ini membutuhkan proses *training* agar mampu melakukan klasifikasi.

```
#arsitektur
model = Sequential() #model = Sequential
model.add(Conv2D(32 , kernel_size=(3,3), activation = 'relu' , input_shape = (150,150,1))) #Layer convolutional 2D
model.add(MaxPooling2D(pool_size=(2,2))) #max pooling with stride (2,2)
model.add(Conv2D(32 , (3,3), activation = 'relu')) #Layer convolutional 2D
model.add(MaxPooling2D(pool_size=(2,2))) #max pooling with stride (2,2)
model.add(Dropout(0.25)) # delete neuron randomly while training and remain 75%
model.add(Flatten()) # make Layer flatten
model.add(Dense(128 , activation = 'relu'))#fully connected layer
model.add(Dropout(0.5)) #delete neuron randomly and remain 50%
model.add(Dense(2 , activation = 'softmax'))
```

Gambar 4. 5 Sourcecode python arsitektur CNN

Gambar 4.5 menunjukkan *sourcecode* Arsitektur CNN yang diusulkan. Penelitian ini menggunakan dua layer konvolusi. Layer konvolusi merupakan bagian hal terpenting dalam CNN. Tujuan digunakannya layer konvolusi untuk proses ekstraksi fitur pada citra. Pada proses Konvolusi pertama digunakan kernel berukuran 3x3 dan jumlah filter sebanyak 32 filter, proses konvolusi ini adalah proses kombinasi antara dua buah matriks yang berbeda untuk menghasilkan suatu nilai matriks yang baru. Output proses konvolusi yaitu menghasilkan transformasi linear dari data input sesuai dengan transformasi spasial pada data.

Selanjutnya, ditambahkan sebuah aktivasi fungsi yaitu ReLU (*Retrified Linear Unit*). Fungsi aktivasi ini bertujuan untuk mengubah nilai negatif menjadi nol (menghilangkan nilai negatif dalam sebuah matriks hasil konvolusi). Operasi ReLU mempercepat proses konvergensi yang dilakukan dengan *stochastic gradient descent*. Jika dibandingkan dengan sigmoid / tanh yang memiliki operasi operasi yang “*expensive*” (*exponentials*, dll), Hasil konvolusi ini memiliki ukuran yang sama yakni 148x148 karena pada saat proses konvolusi digunakan nilai padding 0.

Setelah melalui proses konvolusi hasil akhir dari *maxpooling* akan diubah ke dalam bentuk vector dua dimensi. Penelitian ini menggunakan *maxpooling* untuk mendapatkan nilai matriks yang baru dari proses pooling. Berdasarkan hasil pooling menghasilkan matriks baru berukuran 74×74 dengan menggunakan kernel pooling 2×2 . Cara kerja *Maxpool* mengecilkan input dengan mengambil maksimum wilayah yang ditentukan oleh filternya.

Proses konvolusi kedua yaitu meneruskan hasil dari proses pooling pertama yakni dengan input matriks gambar sebesar 72×72 dengan jumlah filter sebanyak 32 filter dan dengan ukuran kernel 3×3 . Setelah proses konvolusi selesai, hasil akhir dari *maxpooling* akan dikonversi kedalam bentuk *vector* dua dimensi. Perbedaan dengan *pooling* yang pertama yaitu pada *output* akhir dari matriksnya yaitu 36×36 .

Proses selanjutnya, terdapat layer *dense* dengan jumlah vector 128 menggunakan aktivasi fungsi RELU. Setelah itu, digunakan nilai *Dropout* sebesar 0.5. *Dropout* adalah sebuah teknik regulasi jaringan syaraf dengan tujuan memilih beberapa neuron secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain

neuron-neuron tersebut dibuang secara acak. Tujuan dari proses ini yaitu mengurangi *overfitting* pada saat proses training. Selanjutnya dilakukan proses *Flatten* atau *fully connected*. Pada tahap ini digunakan hanya satu *hidden layer* pada jaringan CNN. *Flatten* disini mengubah *output pooling* layer menjadi sebuah vektor. Aktivasi fungsi *softmax* digunakan dengan tujuan untuk mengklasifikasikan input terhadap targetnya, yaitu kedalam dua kelas. Kemudian arsitektur CNN di *compile* seperti pada gambar 4.5.

```
#compile
from keras.optimizers import SGD
epochs=25
lr=0.01
decay=lr/epochs
model.compile(optimizer = "ADAM", loss = 'categorical_crossentropy' , metrics = ['accuracy'])
model.summary()
```

Gambar 4. 6 sourcecode python compile arsitektur CNN

Adam optimizer digunakan dalam proses kompilasi model untuk pembaruan bobot, *loss* untuk mendefinisikan *loss function* dengan memanfaatkan *library* yang ada pada *python* yaitu keras *binary_crossentropy*, dan *metrics* untuk menentukan performa sistem. Seluruh *source code* dalam penelitian ini ditulis dalam *software python* dapat dilihat pada lampiran 3.

Proses terakhir dalam penelitian ini adalah klasifikasi, *Feature map* yang dihasilkan dari *feature*

Perbedaan antara *fully connected layer* dan lapisan konvolusi biasa adalah neuron pada lapisan konvolusi hanya terhubung ke wilayah tertentu pada *input*. Sedangkan *fully connected layer* memiliki neuron yang terhubung satu sama lain. Namun, kedua lapisan tersebut masih menggunakan produk dot, sehingga fungsinya tidak jauh berbeda. Pada tahap ini hasil klasifikasi terbagi menjadi dua yaitu sebagai COVID-19 dan normal. Adapun detail layer dan parameter layer model diberikan pada Tabel 4.1. Arsitektur CNN yang dikembangkan terdiri dari 5.318.370 parameter.

Tabel 4. 1 Layers dan layerparameters

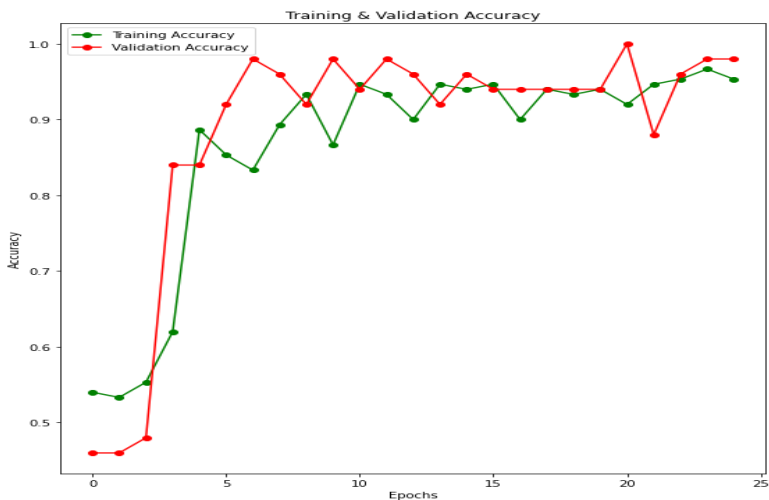
<i>Layer type</i>	<i>Output Shape</i>	<i>Number of Trainable Parameters</i>
<i>Conv2d</i>	[148,148,32]	320
<i>Max Pooling</i>	[74,74,32]	0
<i>Conv2D</i>	[72,72,32]	9248
<i>Max Pooling</i>	[36,36,32]	0
<i>Dropout</i>	[36,36,32]	0
<i>Flatten</i>	[41472]	0
<i>Dense</i>	[128]	5308544
<i>Dropout</i>	[128]	0
<i>Dense</i>	[2]	258

D. Hasil Pengujian

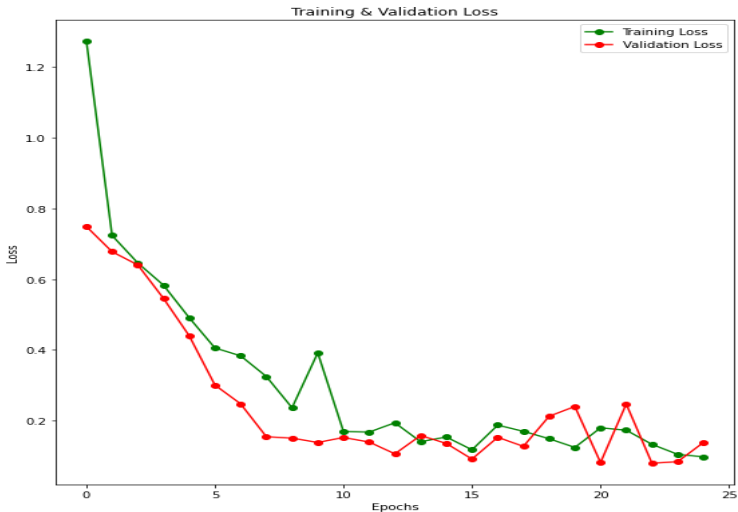
Proses klasifikasi citra *Chest X-Ray* (CXR) dalam penelitian ini menggunakan software *python* dengan menggunakan metode *deep learning Convolutional Neural Network* (CNN). Dengan menggunakan metode ini dapat klasifikasi citra CXR COVID-19 dengan baik dan akurat. Penentuan model terbaik, diperoleh dari pencarian nilai terbaik parameter-parameter dalam model CNN. Parameter yang dimaksud dalam penelitian ini adalah pengaruh jumlah *epoch* dengan *learning rate*. Tujuan dari penentuan parameter model ini membandingkan model mana yang paling baik dengan memperhatikan parameternya.

Epoch adalah ketika seluruh dataset sudah melalui proses *training* pada CNN sampai dikembalikan ke awal dalam satu putaran. Dalam CNN satu *epoch* itu terlalu besar dalam proses pelatihan, karena seluruh data diikutkan ke dalam proses *training* sehingga akan membutuhkan waktu cukup lama. Untuk mempermudah dan mempercepat proses *training* dataset, dilakukan pembagian per *batch* dengan nilai *batch size* yang digunakan dalam penelitian ini yakni 32.

Hasil pengujian model klasifikasi COVID-19 berdasarkan citra CXR bekerja dengan baik dapat dilihat pada lampiran 3. Agar lebih jelas, hasil di visualisasikan pada Gambar 4.3 yang mana pada gambar tersebut menjelaskan perjalanan penggunaan *epoch* pada CNN yang mana hasil akurasi dari *training* terbukti lebih unggul dibandingkan hasil akurasi *testing*. *Loss* pada *neural network* digunakan sebagai penanda dimana model yang sedang dijalankan sudah tidak ada perubahan lagi dalam segi nilai akurasi. *Loss* akan terus diperbarui setiap iterasi *epoch*, sampai *loss* sudah tidak ada perbaikan (lebih rendah) dari sebelumnya.



Gambar 4. 8 Plotting grafik Hasil training Model accuracy

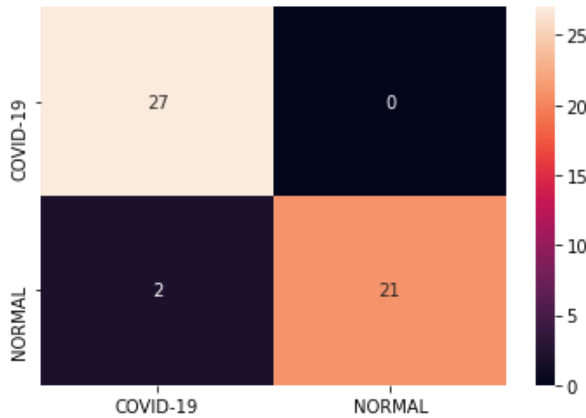


Gambar 4. 9 Plotting grafik Hasil training Model Loss

Berdasarkan gambar 4.4 dengan menggunakan *epoch* ke 25 didapatkan akurasi yang cukup tinggi yakni mencapai 0,98. Jika dilihat dari grafik tersebut disimpulkan bahwa semakin tinggi nilai *epoch* yang digunakan maka akurasi yang didapatkan dari hasil *testing* ikut cenderung naik, Sedangkan pada gambar 4.5 nilai *loss* akan mengalami penurunan. Naik turun nilai akurasi dan *loss* pada grafik dapat disebabkan oleh jumlah *epoch* yang terlalu banyak atau bisa juga dipengaruhi oleh banyaknya dataset.

Langkah terakhir yakni menguji model CNN yang telah dibangun menggunakan data *testing*. Proses *testing* menggunakan data uji sebanyak 50, untuk setiap kelas sebanyak 25 citra. Model CNN pada penelitian ini menggunakan *input shape* berukuran 150x150, nilai *learning rate* 0,01, ukuran filter 3x3, *epoch* 25 dan data *training* 150. Menghasilkan tingkat akurasi *training* dan *testing* dalam melakukan klasifikasi COVID-19 dan normal pada citra CXR sebesar 95,999 %.

Selanjutnya, Gambar 4.6 merupakan *output* dari metode evaluasi yang dimanfaatkan dalam penelitian ini. Metode evaluasi klasifikasi yang digunakan berupa *confusion matrix*. *Confusion matrix* inilah yang akan digunakan dalam perhitungan nilai akurasi, presisi, *recall* atau sensitivitas, dan *f1-score*. Contoh tampilan *confusion matrix* dapat dilihat pada lampiran 4.



Gambar 4. 10 Hasil Confusion Matriks model dalam bentuk Heatmap

Prediksi dari model terhadap data testing yang ditampilkan gambar 4.4 menunjukkan hasil yang baik. Prediksi pada COVID-19 diklasifikasikan benar sebanyak sebanyak 27 dan *missing* data dari input COVID-19 diklasifikasikan sebagai normal sebanyak 0 data. Kemudian prediksi pada normal diklasifikasikan benar sebagai normal sebanyak 21 dan missing data dari input normal diklasifikasikan sebagai COVID-19 sebanyak 2 data. Perhitungan performa sistem (akurasi) , *recall* , *F1 score* berdasarkan pada gambar 4.4 terdapat di lampiran 5, dan berdasarkan perhitungan tersebut maka sistem klasifikasi *Citra Chest X-Ray* (CXR) Penyakit COVID-19 dan Normal dengan Metode *Convolutional Neural*

Network (CNN) menggunakan *Python* mendapatkan akurasi sebesar 96 %, dan precision untuk COVID-19 sebesar 93%, dan precision Normal 100%. Recall COVID-19 100%, sedangkan recall untuk normal 0,91% dan f1-score COVID-19 sebesar 96% sedangkan f1-score Normal sebesar 95%.

E. Perbandingan dengan penelitian sebelumnya

Beberapa penelitian sebelumnya yang menjadi dasar dan perbandingan penelitian ini dirangkum pada tabel 4.2. Rangkuman meliputi data yang digunakan, sebaran data, metode dan hasil penelitian yang telah dilakukan.

Tabel 4. 2 Rangkuman perbandingan hasil

Peneliti	Dataset	Distribusi data	Metode	Akurasi
(Sumarti, Kusuma and Azzahra, 2022)	Citra CXR berjumlah 24 (12 COVID-19 dan 12 SARS)	-	Multi layer Perceptron (MLP)	91,67%
(Maysanjaya, 2020)	Citra CXR Pneumonia dan normal 5840 (Data Train, 3875 pneumonia & 1341 Normal) (Data Test, 390 pneumonia dan 234 Normal)	Data Train=5.216 Data Test=624	CNN	89,58%
(Wang, Lin and Wong, 2020)	Citra CXR=13.870 (Normal, Pneumonia dan COVID-19)	Data train=13.570 Data test=300	CNN COVID-Net	93,33%
Skenario	Citra CXR=200 (100 COVID-19 dan 100 Normal)	Data train=150 Data test=50	CNN (Model usulan)	95,99%

Penelitian ini mengklasifikasikan menjadi dua kelas sama seperti penelitian yang dilakukan oleh Maysanjaya (Maysanjaya 2020). Perbedaan penelitian ini dengan penelitian yang dilakukan Maysanjaya terletak pada distribusi data dan metodologi yang digunakan. Hasil studi Maysanjaya memiliki akurasi 89,58%, sedangkan penelitian ini memiliki akurasi sebesar 95,99%. Studi serupa lainnya juga dilakukan oleh Sumarti dengan metode MLP yang jumlah datasetnya lebih sedikit. Akurasi yang didapatkan 91,67%. Berbeda dengan penelitian Maysanjaya, penelitian oleh Wang mendapatkan akurasi sebesar 93,33% dengan mengklasifikasikan citra menjadi tiga kelas. Dari perbandingan tersebut model penelitian CNN yang diusulkan ternyata lebih efektif dan lebih unggul sehingga dapat dijadikan sebagai alternatif untuk mengklasifikasikan COVID-19 dan Normal.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian, dapat diambil kesimpulan sebagai berikut:

1. Proses klasifikasi citra CXR dengan metode CNN menggunakan bahasa pemrograman *python* dimulai dengan proses *resize* untuk mengubah ukuran piksel citra. Kemudian dilakukan penyeragaman citra ke *grayscale*. Selanjutnya dilakukan proses *CLAHE* untuk memperbaiki dan menambah kontras citra. Tahap selanjutnya adalah ekstraksi ciri pada layer konvolusi. Citra yang awalnya berbentuk multidimensional array diubah menjadi sebuah vector. Citra COVID-19 direpresentasikan dengan nilai matriks yang lebih tinggi yaitu mendekati 1 dari pada citra normal. Hal tersebut dikarenakan pada citra COVID-19 terdapat GGO yang ditandai area berwarna putih atau abu-abu. Tahap terakhir yaitu mengklasifikasikan citra CXR berdasarkan dengan klasifikasi kelasnya yaitu normal dan COVID-19.
2. Penggunaan parameter *epoch* sangat berpengaruh terhadap akurasi yang dihasilkan. Akurasi terbaik didapatkan pada *epoch* ke-25 dengan akurasi

tertinggi pada *training test* 98% dengan waktu 3s. Hasil kinerja sistem klasifikasi citra CXR penyakit COVID-19 dan normal dengan metode CNN menggunakan *python* cukup efektif dan efisien, akurasi pengujian yang diperoleh mencapai 96%.

B. Saran

Berdasarkan hasil yang diperoleh dari penelitian ini, saran yang dapat diberikan untuk perbaikan sistem maupun untuk penelitian selanjutnya, sebagai berikut :

1. Memperbanyak data sehingga sistem dapat berkembang tidak hanya untuk COVID-19 dan normal saja selain itu agar memperoleh sistem yang lebih baik.
2. Algoritma bisa lebih ditingkatkan lagi jika data diperbanyak.

DAFTAR PUSTAKA

- Adu, Apris, and Mas'amah. 2021. "Pemetaan Kematian Akibat Covid-19 Berdasarkan Variabel Epidemiologi." *Jurnal Kajian Veteriner* 9 (3): 194–202. <https://doi.org/10.35508/jkv.v9i3.5261>.
- Ai, Tao, Zhenlu Yang, Hongyan Hou, Chenao Zhan, Chong Chen, Wenzhi Lv, Qian Tao, Ziyong Sun, and Liming Xia. 2020. "Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases." *Radiology* 296 (2): E32–40. <https://doi.org/10.1148/radiol.2020200642>.
- Az-zahra, Mien Fatimah. 2019. *Implementasi Deep Learning Dalam Digital Image Processing Retina Mata Untuk Deteksi Kelainan*. Semarang: Universitas Negeri Semarang.
- Azizah, Anisa N U R, Program Studi Matematika, Fakultas Sains, D A N Teknologi, Universitas Islam, and Negeri Sunan. 2021. *Klasifikasi Penyakit Diabetic Retinopathy Menggunakan Metode Convolutional Neural Network (Cnn) Model Deep Residual Network (Resnet)*.
- Burhan, Erlina, Fathiyah Isbaniyah, Agus dwi Susanto, and dkk. 2020. *PNEUMONIA COVID-19 Diagnosis & Penatalaksanaan Di Indonesia*. Jakarta: Perhimpunan Dokter Paru Indonesia.

file:///C:/Users/ThinkPad X250/Downloads/Documents/per september/buku_pneumonia_covid19.pdf.

Coates, Adam, Honglak Lee, and Andrew Y Ng. 2011. "An Analysis of Single-Layer Networks in Unsupervised Feature Learning." *Qiche Gongcheng/Automotive Engineering* 24 (2): 130.

Cucinotta, Domenico, and Maurizio Vanelli. 2020. "WHO Declares COVID-19 a Pandemic." *Acta Biomedica* 91 (1): 157–60. <https://doi.org/10.23750/abm.v91i1.9397>.

Eka Putra, Wayan Suartika. 2016. "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101." *Jurnal Teknik ITS* 5 (1). <https://doi.org/10.12962/j23373539.v5i1.15696>.

Ermawati, Eli. 2020. "Klasifikasi Nodul Payudara Berdasarkan Ciri Tekstur Pada Citra Ultrasonografi Menggunakan Scilab."

Fachrozy, Ryan. 2020. *Identifikasi Efusi Pleura Berdasarkan Citra Chest X-Ray Dengan Menggunakan Convolutional Neural Network. Universitas Sumatera Utara.*

Gozes, Ophir, Ma Frid, Hayit Greenspan, and D Patrick. 2020. "Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic : Initial Results for Automated Detection & Patient Monitoring Using Deep Learning CT Image Analysis

Article Type : Authors : Summary Statement : Key Results :
 List of Abbreviati.” *ArXiv:2003.05037*.
<https://arxiv.org/ftp/arxiv/papers/2003/2003.05037.pdf>.

Gunawan, Danny, and Hendry Setiawan. 2022. “Convolutional Neural Network Dalam Citra Medis.” *KONSTELASI: Konvergensi Teknologi Dan Sistem Informasi 2 (2)*: 376–90.
<https://doi.org/10.24002/konstelasi.v2i2.5367>.

Hellmann, Doug. 2011. *The Python Standard Library by Example*. Pearson Education. Boston. [https://doi.org/10.1016/S0079-6123\(08\)64305-8](https://doi.org/10.1016/S0079-6123(08)64305-8).

Huang, Chaolin, Yeming Wang, Xingwang Li, Lili Ren, Jianping Zhao, Yi Hu, Li Zhang, et al. 2020. “Clinical Features of Patients Infected with 2019 Novel Coronavirus in Wuhan, China.” *Www.TheLancet.Com* 395: 497.
[https://doi.org/10.1016/S0140-6736\(20\)30183-5](https://doi.org/10.1016/S0140-6736(20)30183-5).

Kamaruddin, Shazeem. 2010. *Pengetahuan Dokter Muda (Co-Ass) Tentang Pemeriksaan Diagnostik Radiologi Foto Toraks Di RSUP Haji Adam Malik Medan 2010*. Medan: Fakultas Kedokteran Universitas Sumatera Utara.
<http://repository.usu.ac.id/handle/123456789/25853>.

Li, Qun, Xuhua Guan, Peng Wu, Xiaoye Wang, Lei Zhou, Yeqing Tong, Ruiqi Ren, et al. 2020. “Early Transmission Dynamics

in Wuhan, China, of Novel Coronavirus-Infected Pneumonia.” *New England Journal of Medicine*. Massachusetts Medical Society.
<https://doi.org/10.1056/NEJMoa2001316>.

Matondang, Zekson Arizona. 2018. “Penerapan Metode Contrast Limited Adaptive Histogram Equalization (CLAHE) Pada Citra Digital Untuk Memperbaiki Gambar X-Ray.” *Publikasi Ilmiah Teknologi Informasi Neumann* 3: 24–29.

Maula, Zulfa. 2021. “KLASIFIKASI PNEUMONIA AKIBAT VIRUS CORONA PADA CITRA RONTGEN TORAKS BERBASIS FITUR STATISTIK MENGGUNAKAN SCILAB DAN WEKA.”

Maysanjaya, I Md. Dendi. 2020. “Klasifikasi Pneumonia Pada Citra X-Rays Paru-Paru Dengan Convolutional Neural Network.” *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi* 9 (2): 190–95.
<https://doi.org/10.22146/jnteti.v9i2.66>.

Nurhikmat, Triano. 2018. “IMPLEMENTASI DEEP LEARNING UNTUK IMAGE CLASSIFICATION MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) PADA CITRA WAYANG GOLEK” 2 (1): 1–13.
<http://link.springer.com/10.1007/978-3-319-76887-8%0Ahttp://link.springer.com/10.1007/978-3-319-93594-2%0Ahttp://dx.doi.org/10.1016/B978-0-12-409517->

5.00007-

3%0Ahttp://dx.doi.org/10.1016/j.jff.2015.06.018%0Ahttp://
/dx.doi.org/10.1038/s41559-019-0877-3%0Aht.

Ozturk, Tulin, Muhammed Talo, Eylül Azra, Ulas Baran, Ozal Yildirim, and U Rajendra Acharya. 2020. "Automated Detection of COVID-19 Cases Using Deep Neural Networks with X-Ray Images." *Computers in Biology and Medicine* 121 (April): 103792.
<https://doi.org/10.1016/j.combiomed.2020.103792>.

Perhimpunan Dokter Paru Indonesia. 2020. "Panduan Praktik Klinis: Pneumonia COVID-19." *Journal of the American Pharmacists Association* 55 (5): 1–67.

Pham, Tuan D. 2021. "Classification of COVID-19 Chest X-Rays with Deep Learning: New Models or Fine Tuning?" *Health Information Science and Systems* 9 (1): 1–11.
<https://doi.org/10.1007/s13755-020-00135-3>.

Rahimi, Farid, and Amin Talebi Bezmin Abadi. 2022. "Emergence of the Omicron SARS-CoV-2 Subvariants during the COVID-19 Pandemic – Correspondence." *International Journal of Surgery* 108 (November): 106994.
<https://doi.org/10.1016/j.ijssu.2022.106994>.

Risnawati, Dewi, Ramli Haji Ali, and Vonny Tubagus. 2015.

“Gambaran Hasil Pemeriksaan Foto Toraks Pada Pasien Baru Yang Melakukan Pemeriksaan Toraks Foto Di Bagian/Smf Radiologi Blu Rsup Prof. Dr. R. D. Kandou Manado Periode Juni-Oktober 2014.” *E-CliniC* 3 (1). <https://doi.org/10.35790/ecl.3.1.2015.6514>.

Sumarni, and Terry Mart. 2000. “Penentuan Energi Efektif Sinar-X Radiodiagnostik.” *Proceeding Ilmiah Keslamatan Radiasi Dan Lingkungan*, 23–24.

Sumarti, Heni, Hadi Kusuma, and Jannatul Firdausa Azzahra. 2022. “Klasifikasi Kasus COVID-19 Dan SARS Berbasis Ciri Tekstur Menggunakan Metode Multi Layer Perceptron.” *Unnes Physics Journal* 12 (1): 16–27.

Susilo, Pratiwi, Rudi. 2013. “PENGUKURAN PAPARAN RADIASI PESAWAT SINAR X DI INSTALASI RADIODIAGNOSTIK UNTUK PROTEKSI RADIASI.” *Unnes Physics Journal* 1 (2252). <http://journal.unnes.ac.id/sju/index.php/upj>.

Wang, Fan, Jiayan Nie, Haizhou Wang, Qiu Zhao, Yong Xiong, Liping Deng, Shihui Song, Zhiyong Ma, Pingzheng Mo, and Yongxi Zhang. 2020. “Characteristics of Peripheral Lymphocyte Subset Alteration in COVID-19 Pneumonia” 221 (1). <https://doi.org/10.1093/infdis/jiaa150>.

Wang, Linda., Zhong Qiu. Lin, and Alexander. Wong. 2020.

“COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images.” *ArXiv*, 1–12.








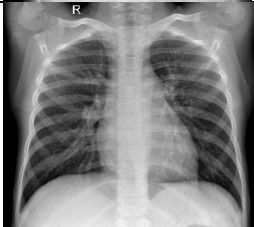


Wang, Shuai, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai, et al. 2020. “A Deep Learning Algorithm Using CT Images to Screen for Corona Virus Disease (COVID-19).” *MedRxiv*, April, 2020.02.14.20023028. <https://doi.org/10.1101/2020.02.14.20023028>.








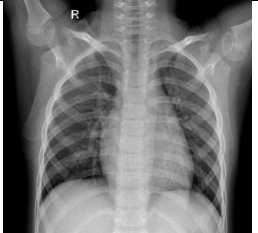


Wang, Zhou, Wang Qiang, and Hu Ke. 2020. “A Handbook of 2019-NCoV Pneumonia Control and Prevention.” *Hubei Science and Technology Press*, 1–108.



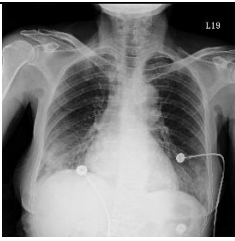





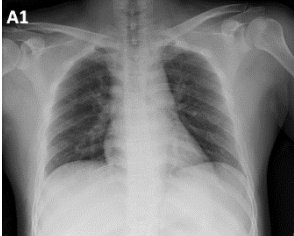

Winangun, Putu Prima, I Made Oka Widyantara, and Rukmi Sari Hartati. 2020. “Pendekatan Diagnostik Berbasis Extreme Learning Machine Dengan Kernel Linear Untuk Mengklasifikasi Kelainan Paru-Paru.” *Majalah Ilmiah Teknologi Elektro* 19 (1): 83. <https://doi.org/10.24843/mite.2020.v19i01.p12>.

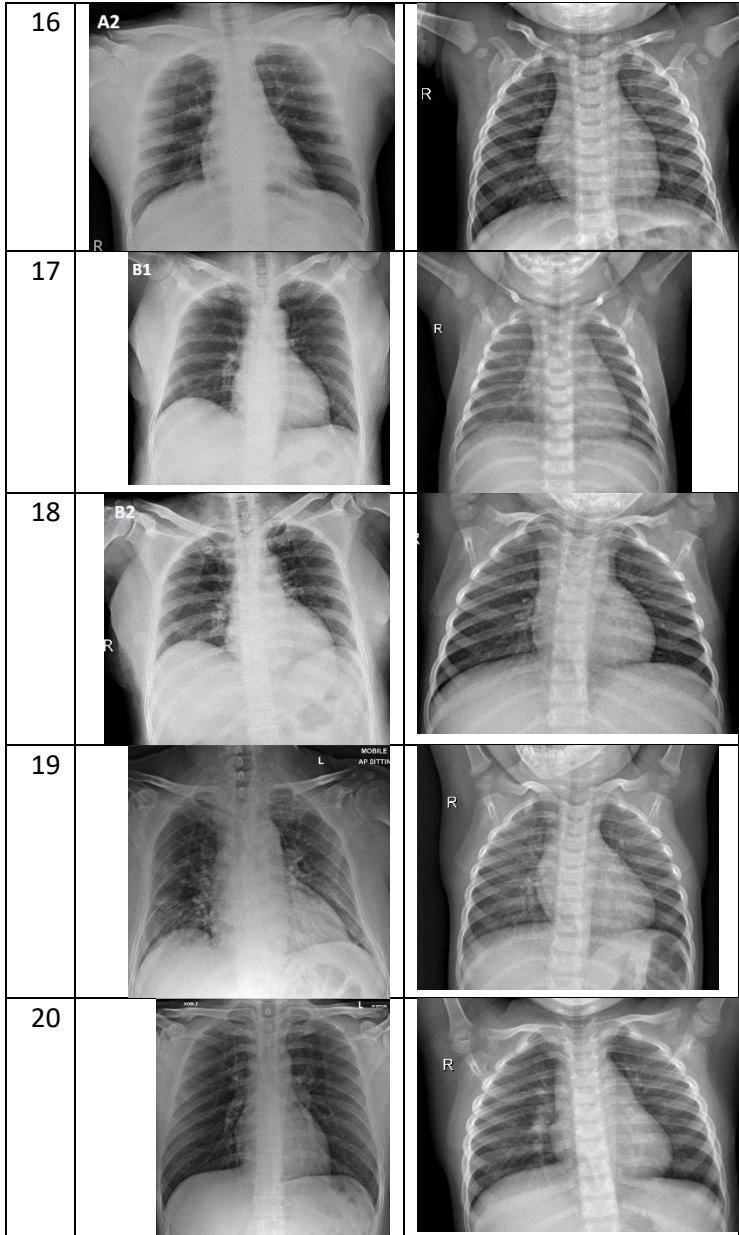
LAMPIRAN- LAMPIRAN

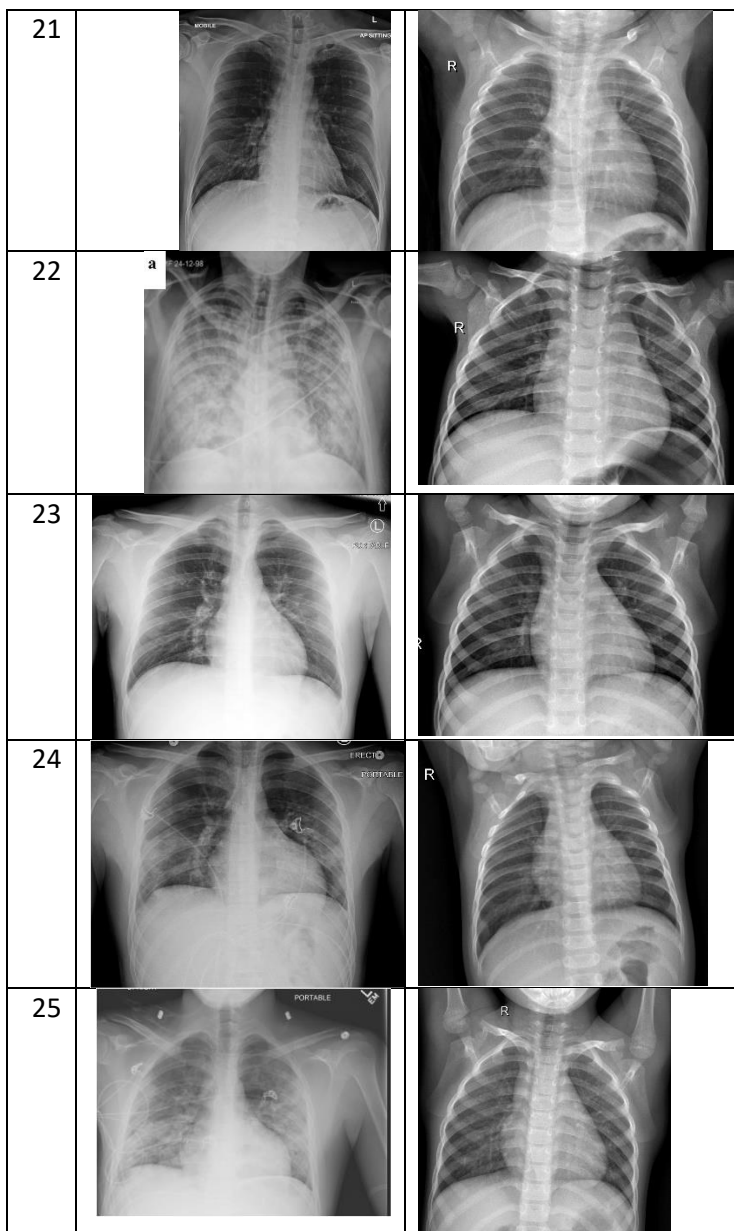
Lampiran 1: Citra Chest X-Ray

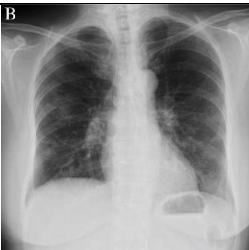
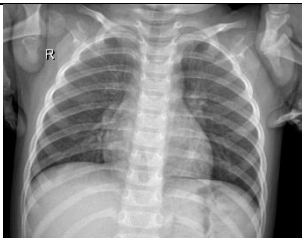
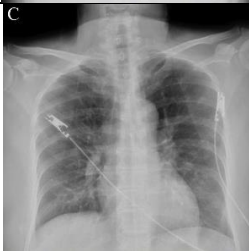

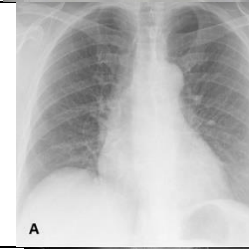





No	COVID-19	Normal
1		
2		
3		
4		
5		

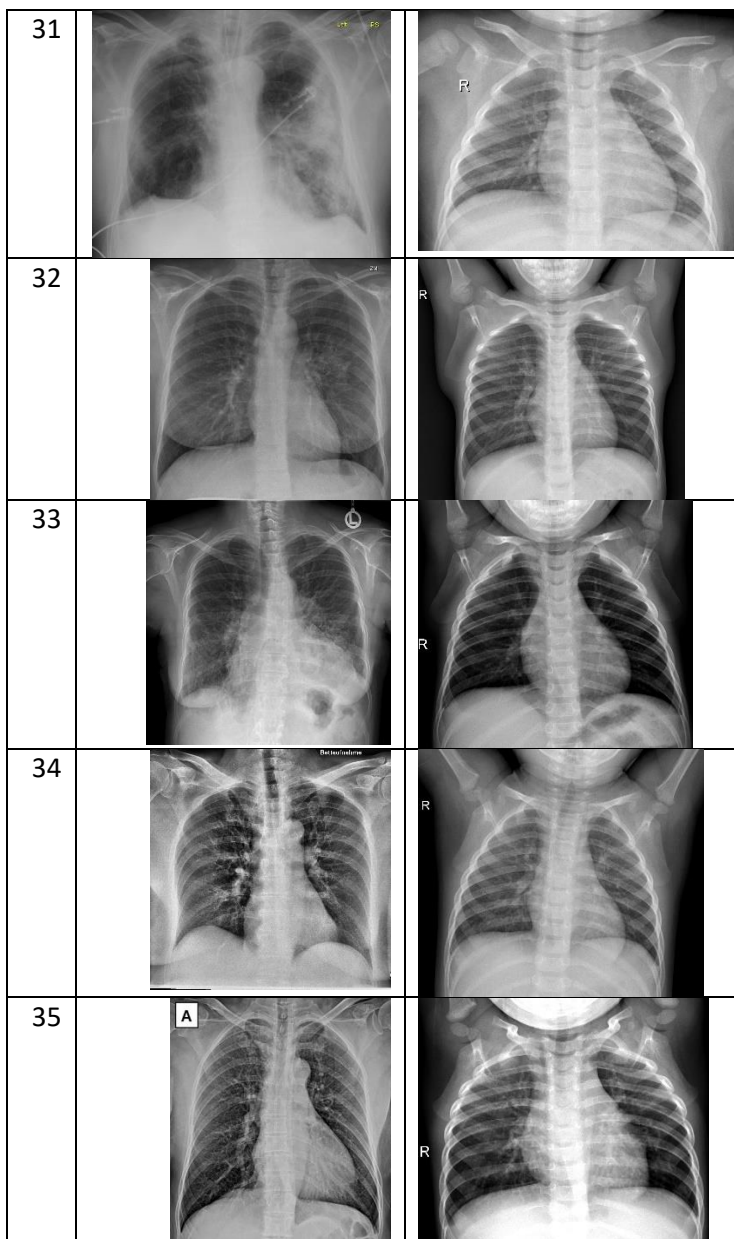
6		
7		
8		
9		
10		




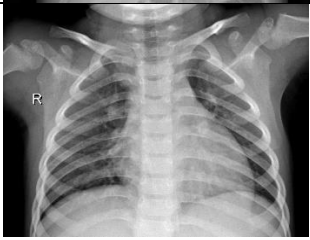
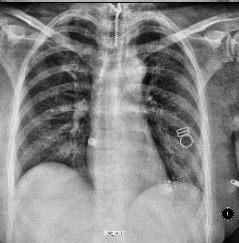





11		
12		
13		
14		
15		













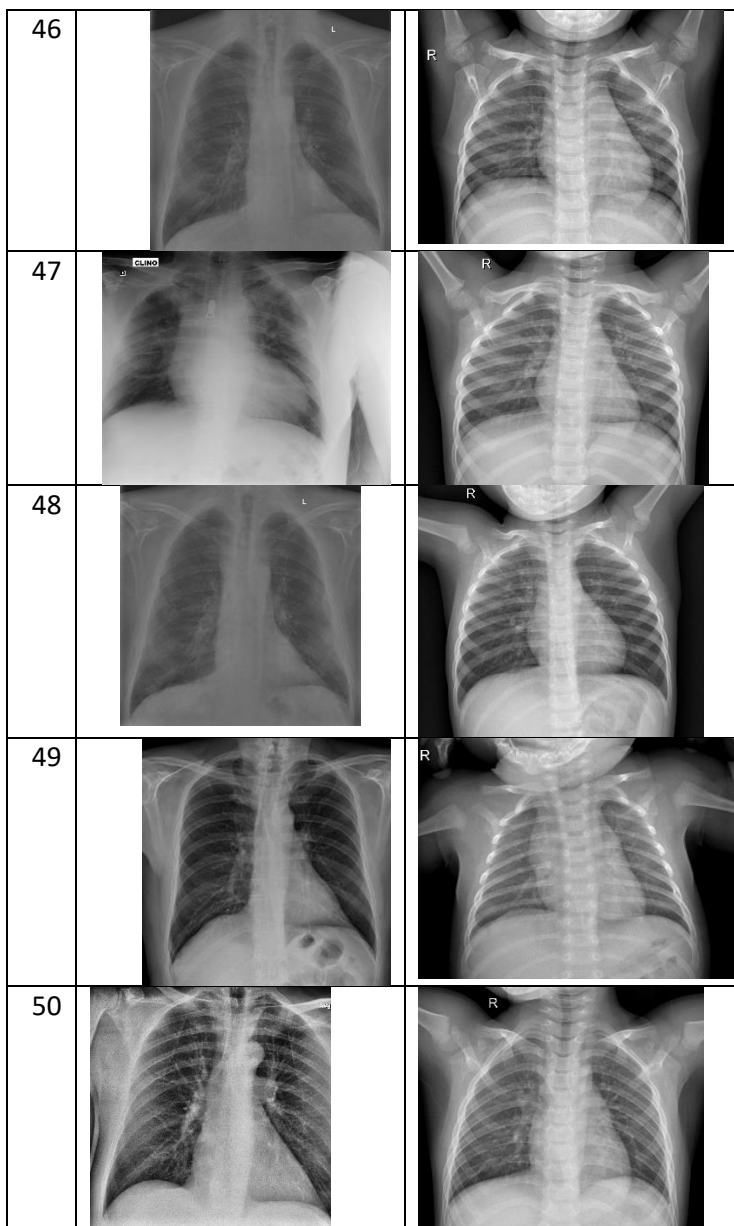


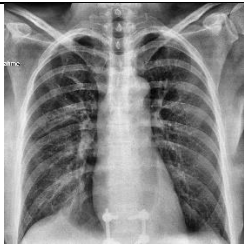




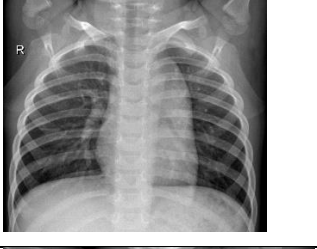




26	 <p>Chest X-ray (PA view) labeled B. The image shows a normal chest X-ray with clear lung fields, a normal heart size, and no visible abnormalities.</p>	 <p>Chest X-ray (AP view) labeled R. The image shows a normal chest X-ray with clear lung fields, a normal heart size, and no visible abnormalities.</p>
27	 <p>Chest X-ray (PA view) labeled C. The image shows a normal chest X-ray with clear lung fields, a normal heart size, and no visible abnormalities.</p>	 <p>Chest X-ray (AP view) labeled R. The image shows a normal chest X-ray with clear lung fields, a normal heart size, and no visible abnormalities.</p>
28	 <p>Chest X-ray (PA view) labeled A. The image shows a normal chest X-ray with clear lung fields, a normal heart size, and no visible abnormalities.</p>	 <p>Chest X-ray (AP view) labeled R. The image shows a normal chest X-ray with clear lung fields, a normal heart size, and no visible abnormalities.</p>
29	 <p>Chest X-ray (PA view). The image shows a normal chest X-ray with clear lung fields, a normal heart size, and no visible abnormalities. There are several medical devices visible, including a central venous catheter and a nasogastric tube.</p>	 <p>Chest X-ray (AP view) labeled R. The image shows a normal chest X-ray with clear lung fields, a normal heart size, and no visible abnormalities.</p>
30	 <p>Chest X-ray (PA view). The image shows a normal chest X-ray with clear lung fields, a normal heart size, and no visible abnormalities. There are several medical devices visible, including a central venous catheter and a nasogastric tube.</p>	 <p>Chest X-ray (AP view) labeled R. The image shows a normal chest X-ray with clear lung fields, a normal heart size, and no visible abnormalities.</p>








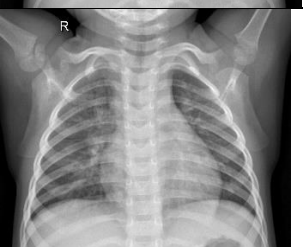






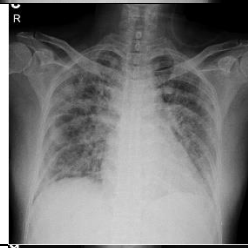
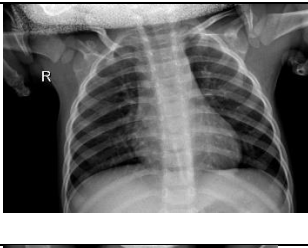




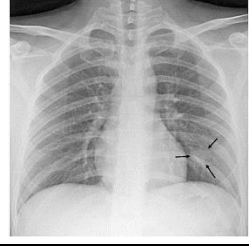

36		
37		
38		
39		
40		

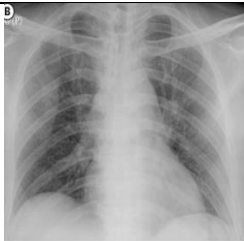




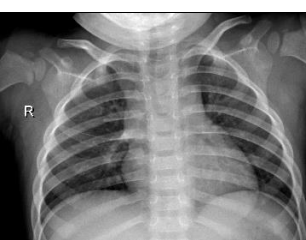
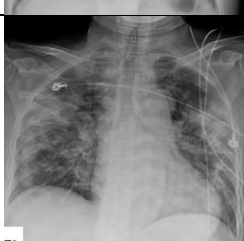



41	 <p>AP-PORT-SEMUPRIGHT L CRP</p>	 <p>R</p>
42	 <p>AP-PORT-SEMUPRIS L CRK</p>	 <p>R</p>
43	 <p>Portabli Semi-Upright</p>	 <p>R</p>
44	 <p>L</p>	
45	 <p>R SUPRO ACRIS</p>	 <p>R</p>

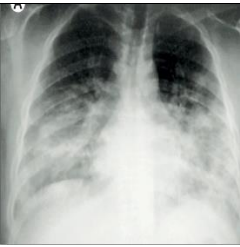
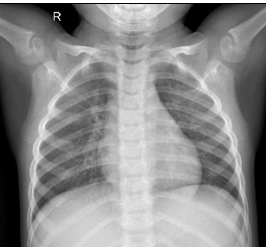



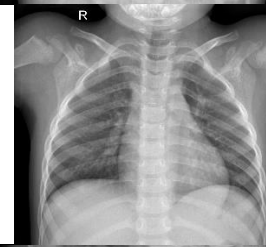

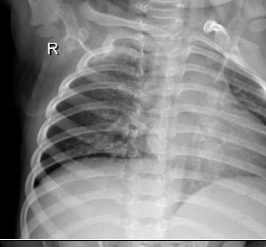




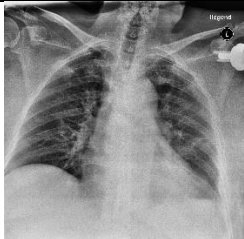






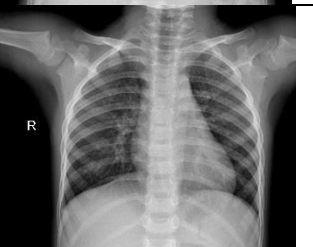


51		
52		
53		
54		
55		


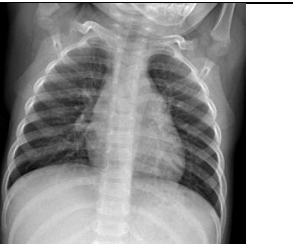

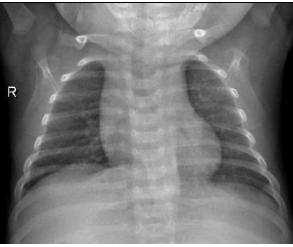






56		
57		
58		
59		
60		


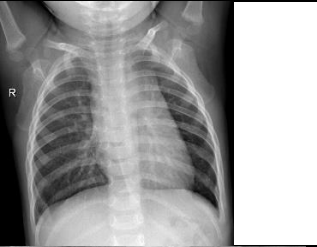




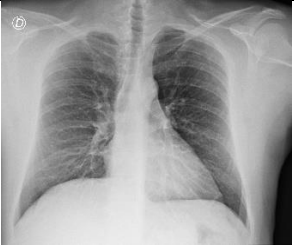
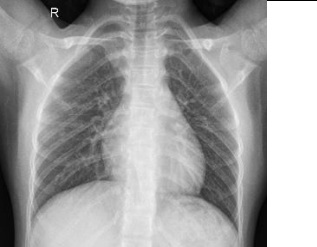

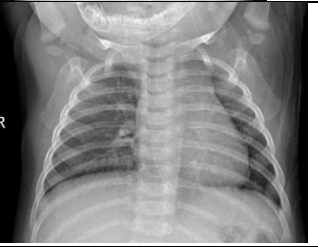
61		
62		
63		
64		
65		

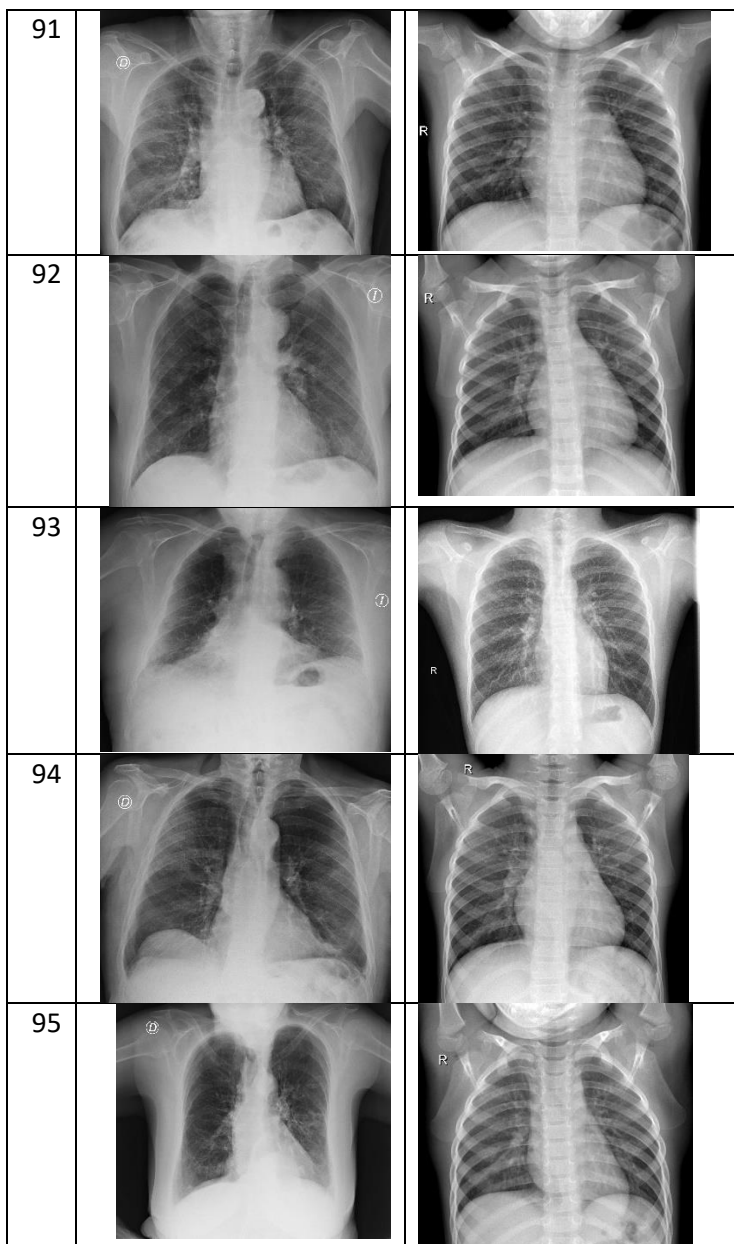
66		
67		
68		
69		
70		


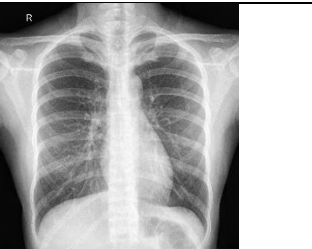
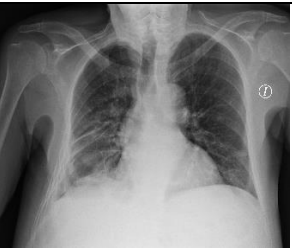

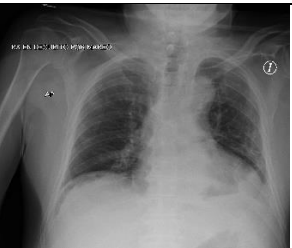


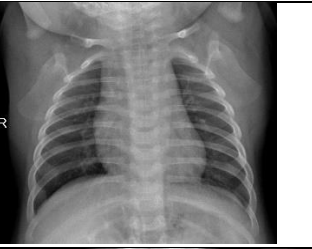
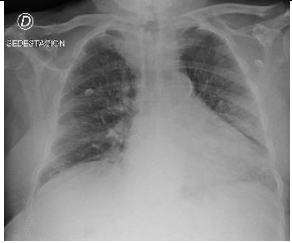

71		
72		
73		
74		
75		

76		
77		
78		
79		
80		







81		
82		
83		
84		
85		

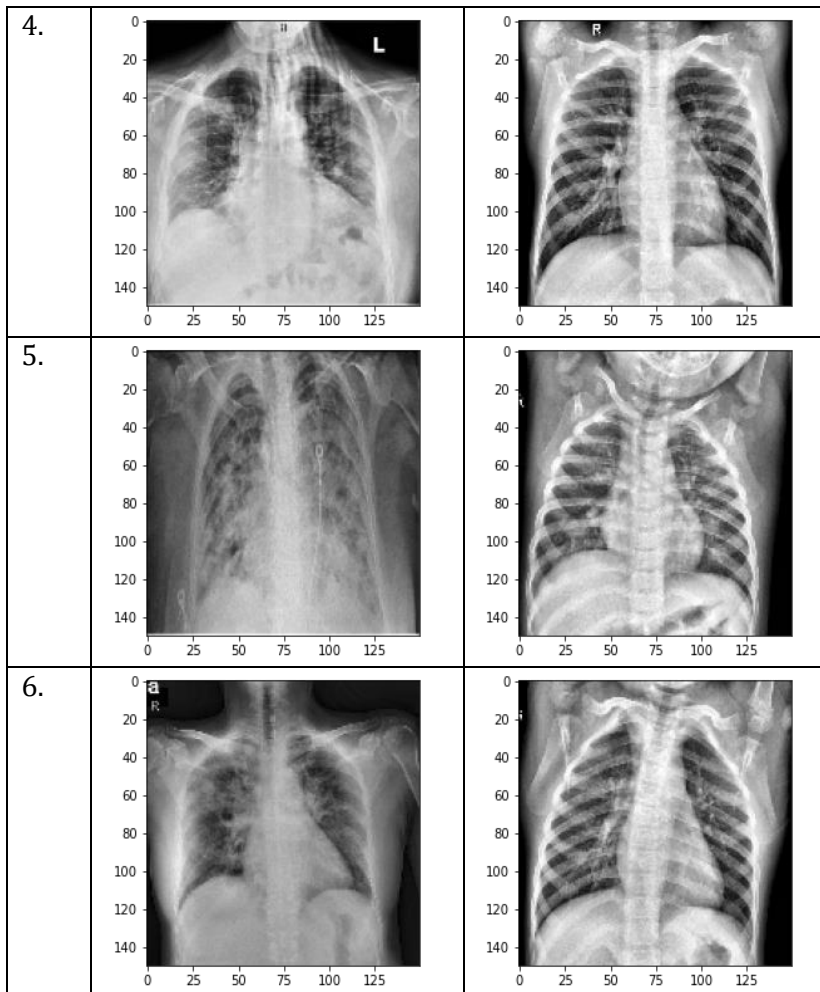
86		
87		
88		
89		
90		

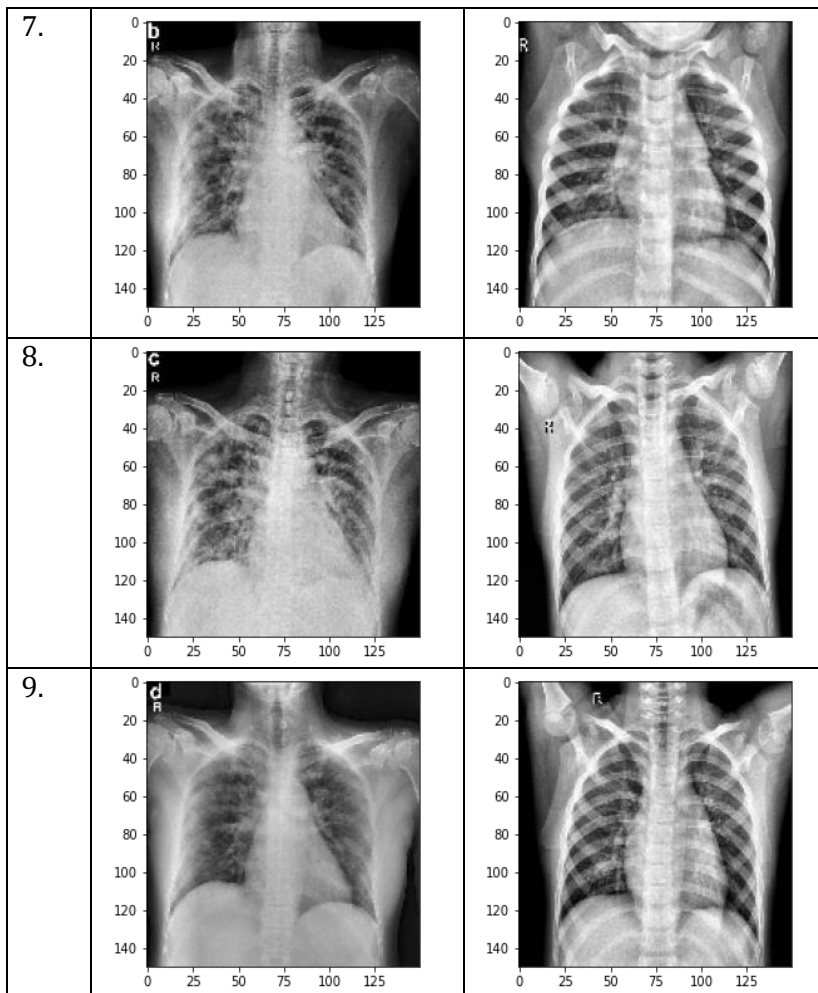


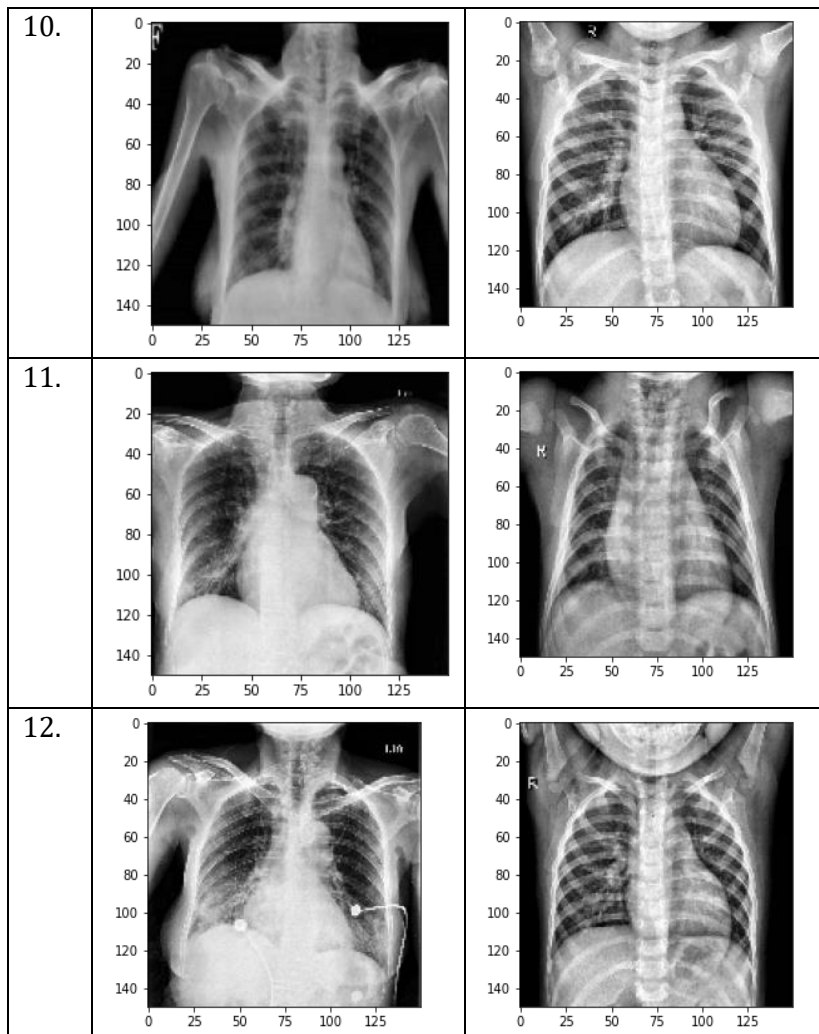
96		
97		
98		
99		
100		

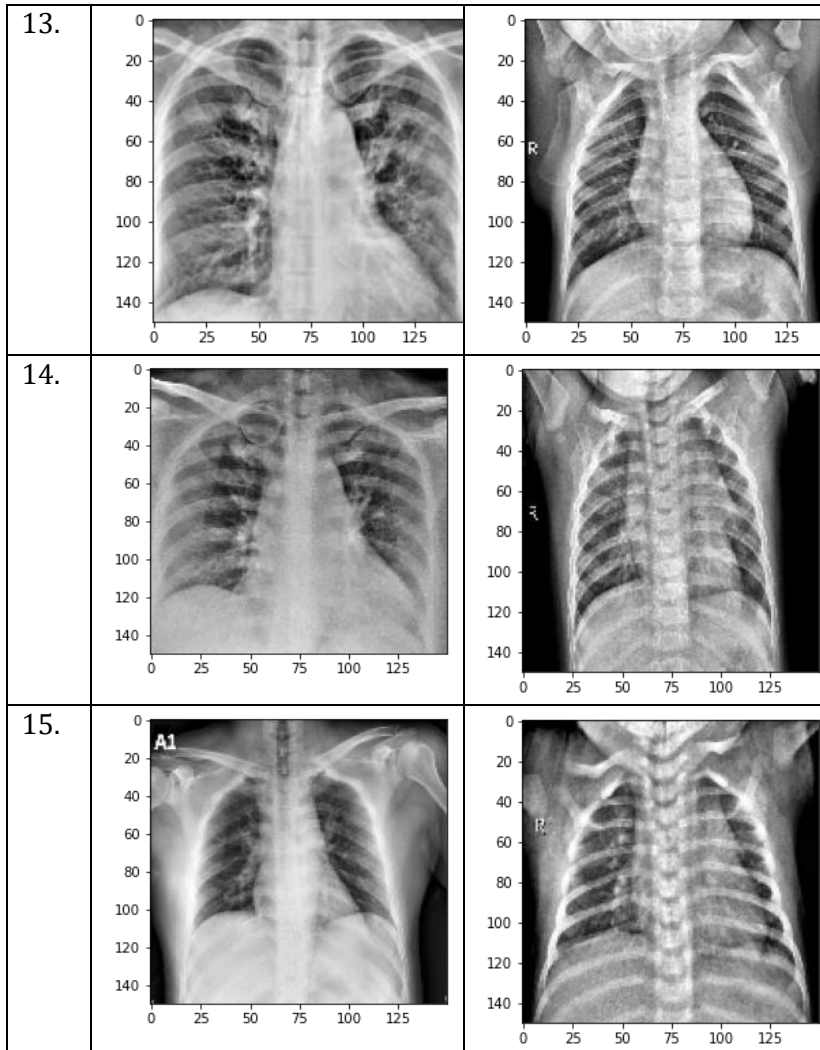
Lampiran 2: Citra Chest X-Ray setelah pre-processing

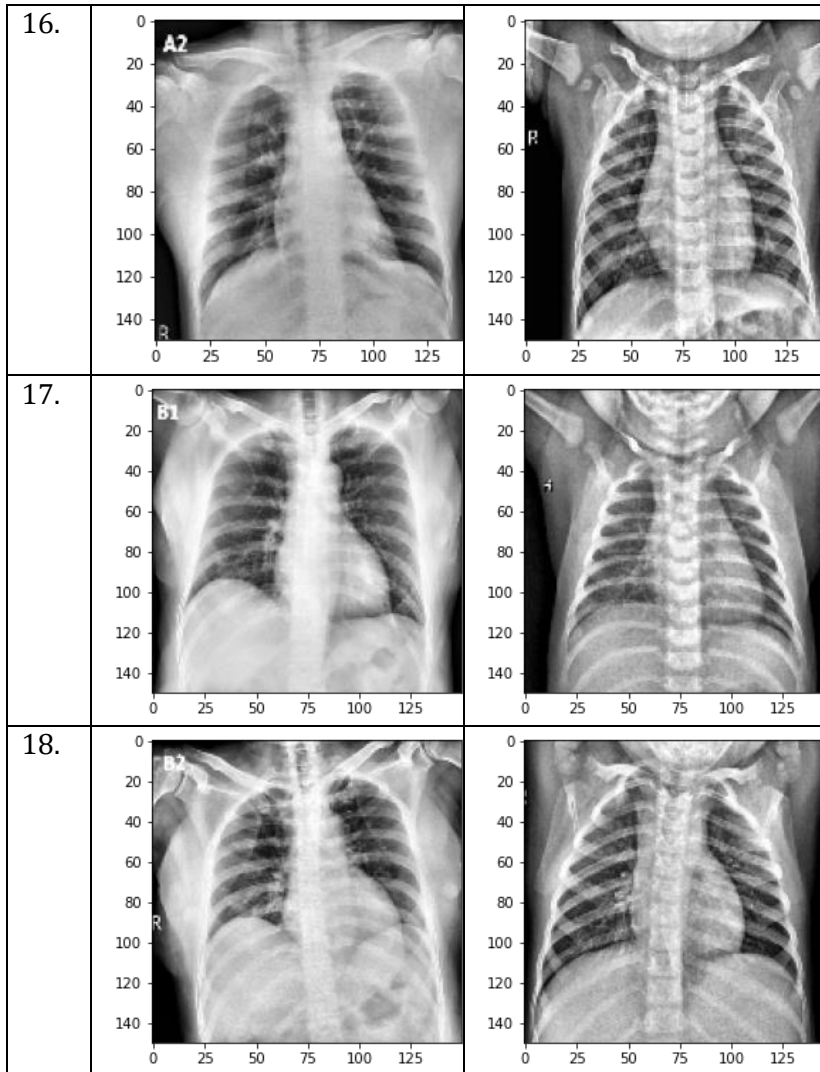
NO	COVID	NORMAL
1.		
2.		
3.		

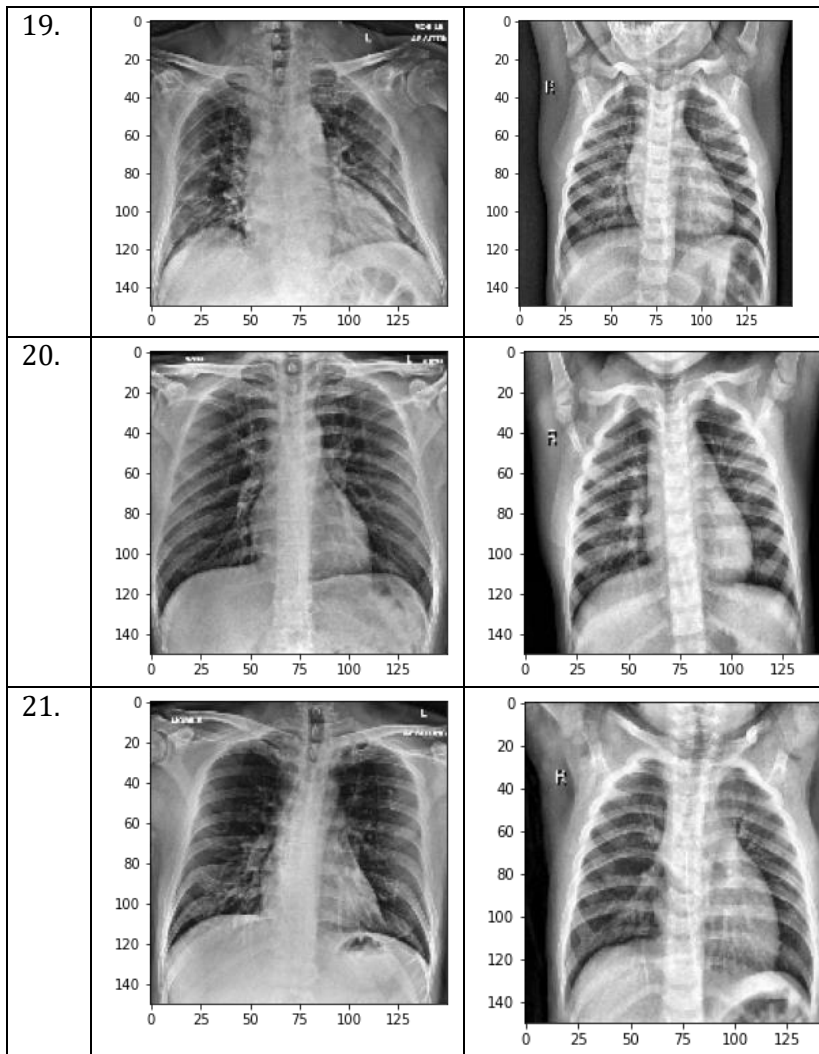


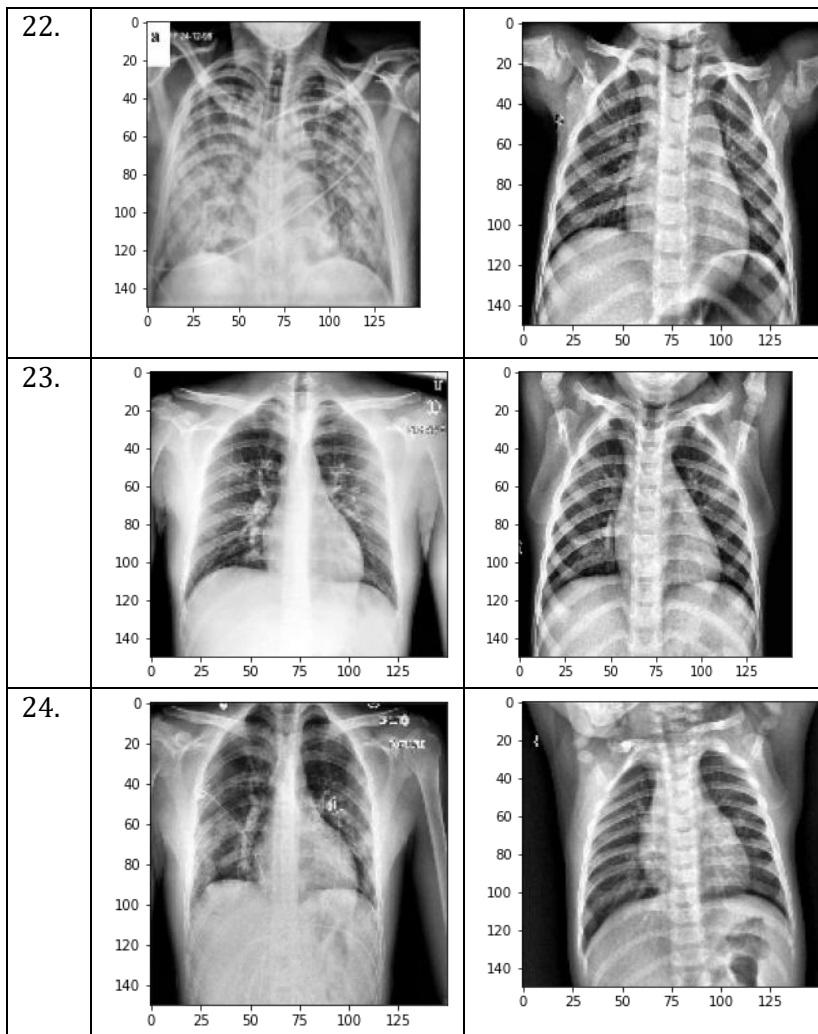


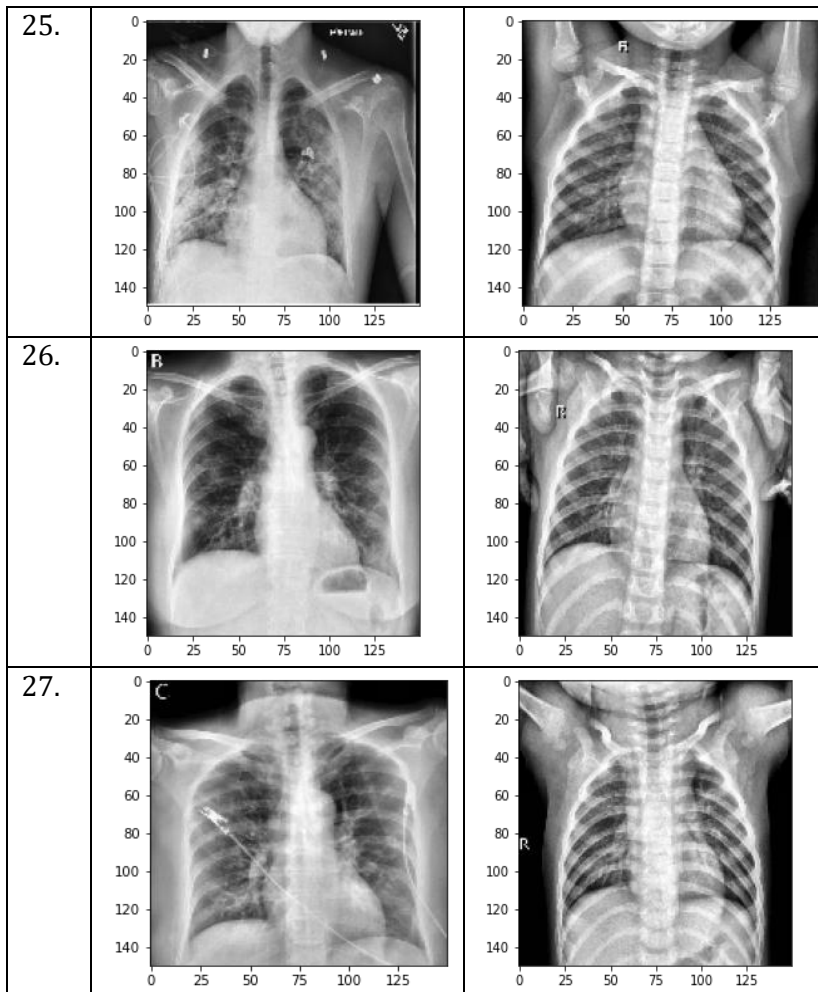


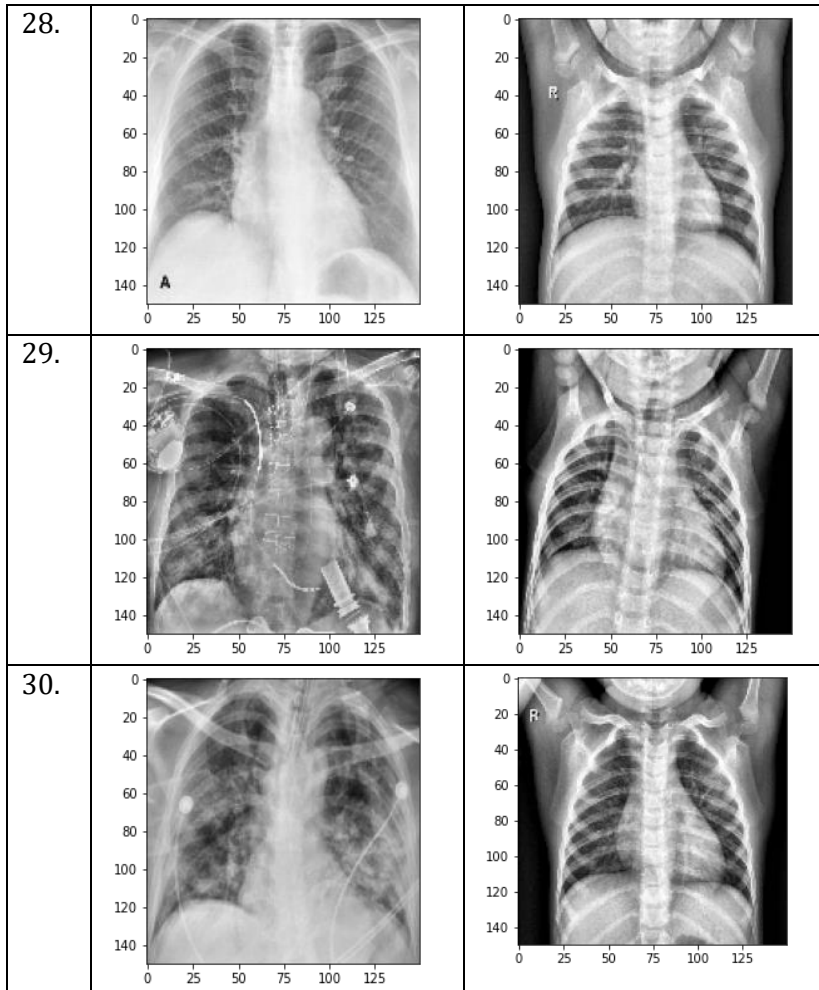


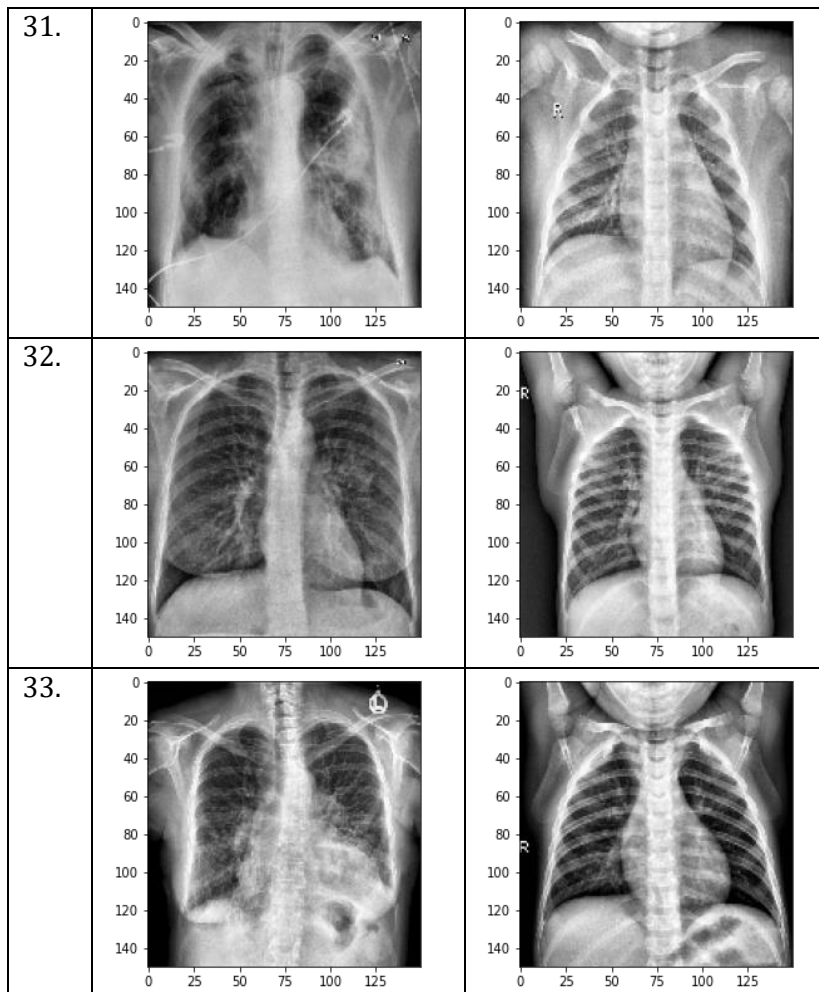


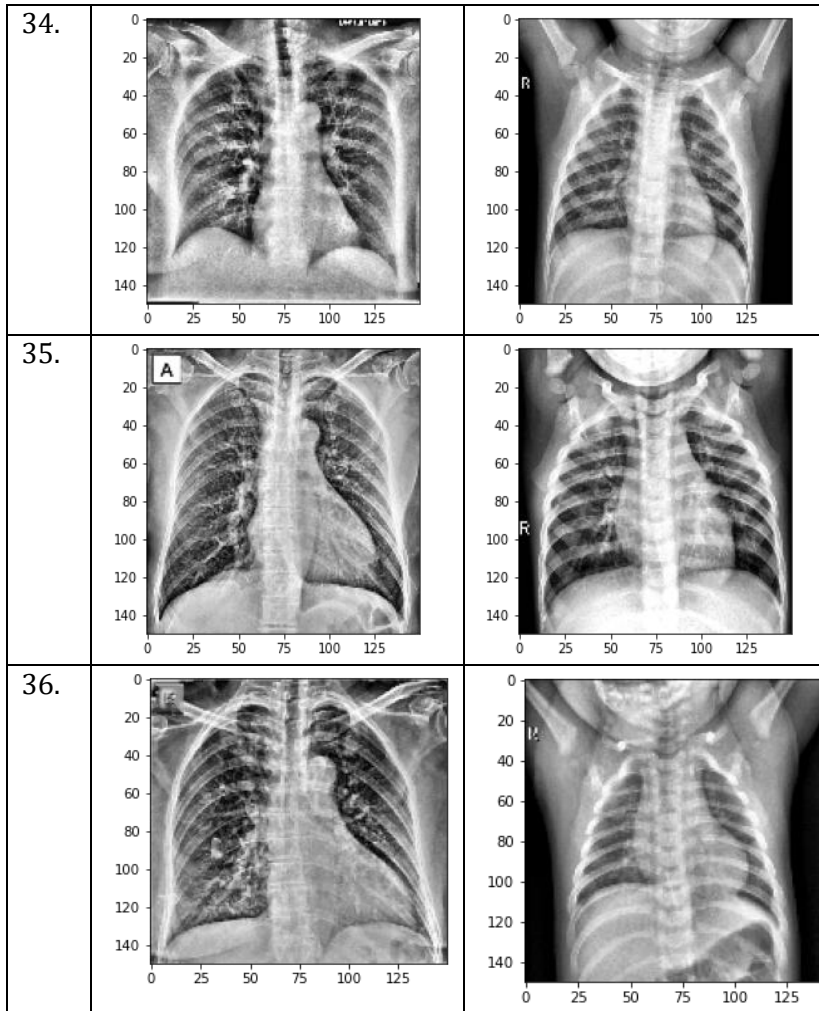


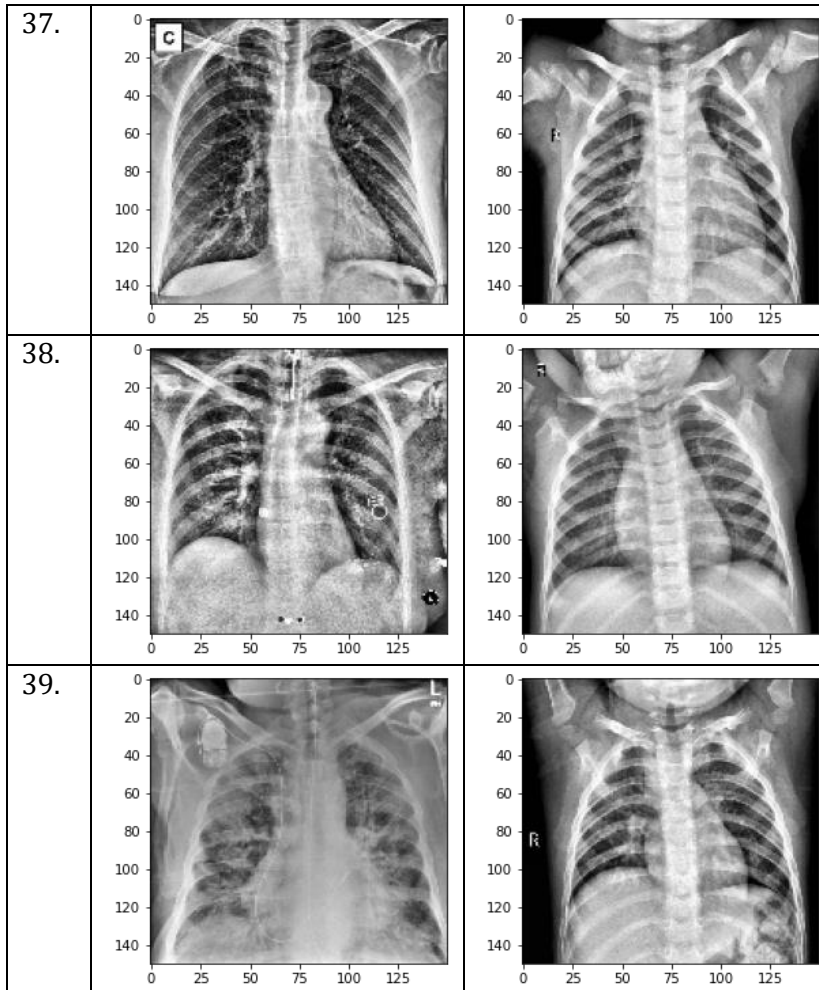


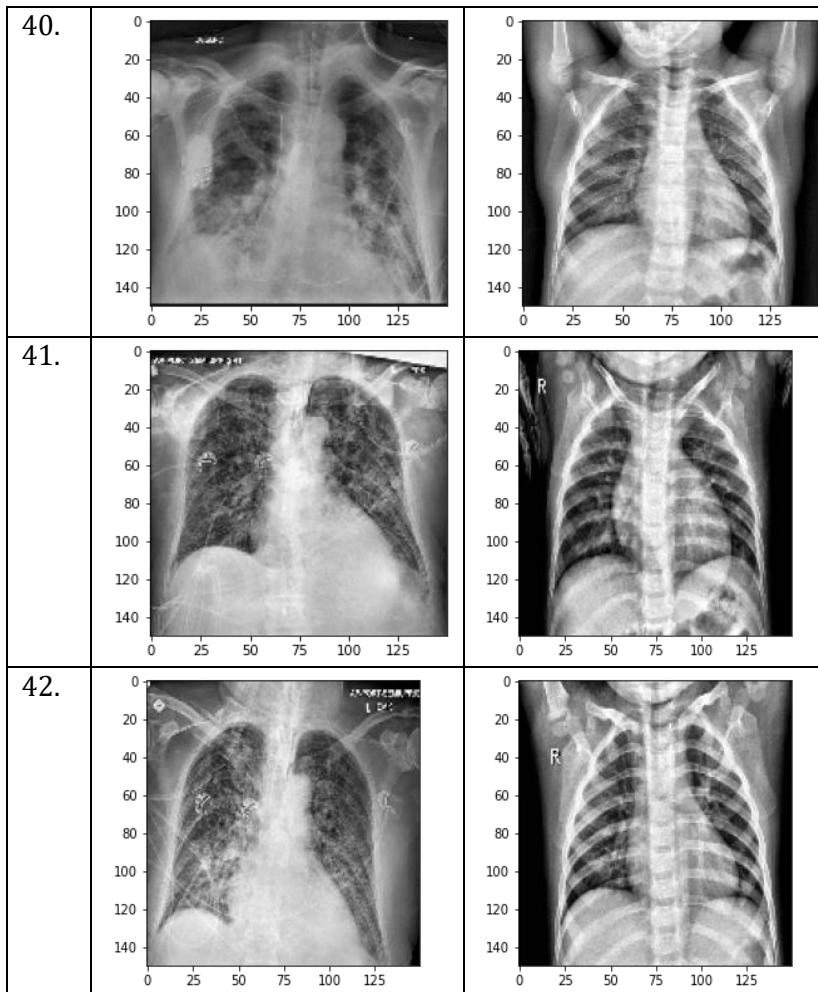


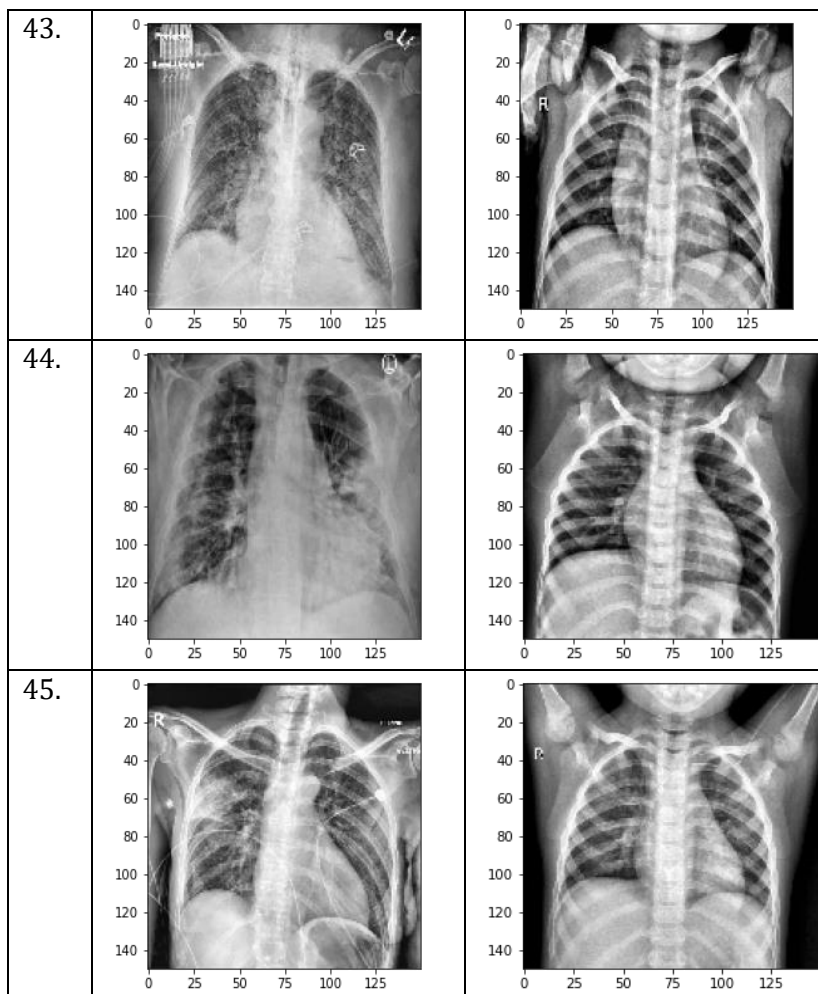


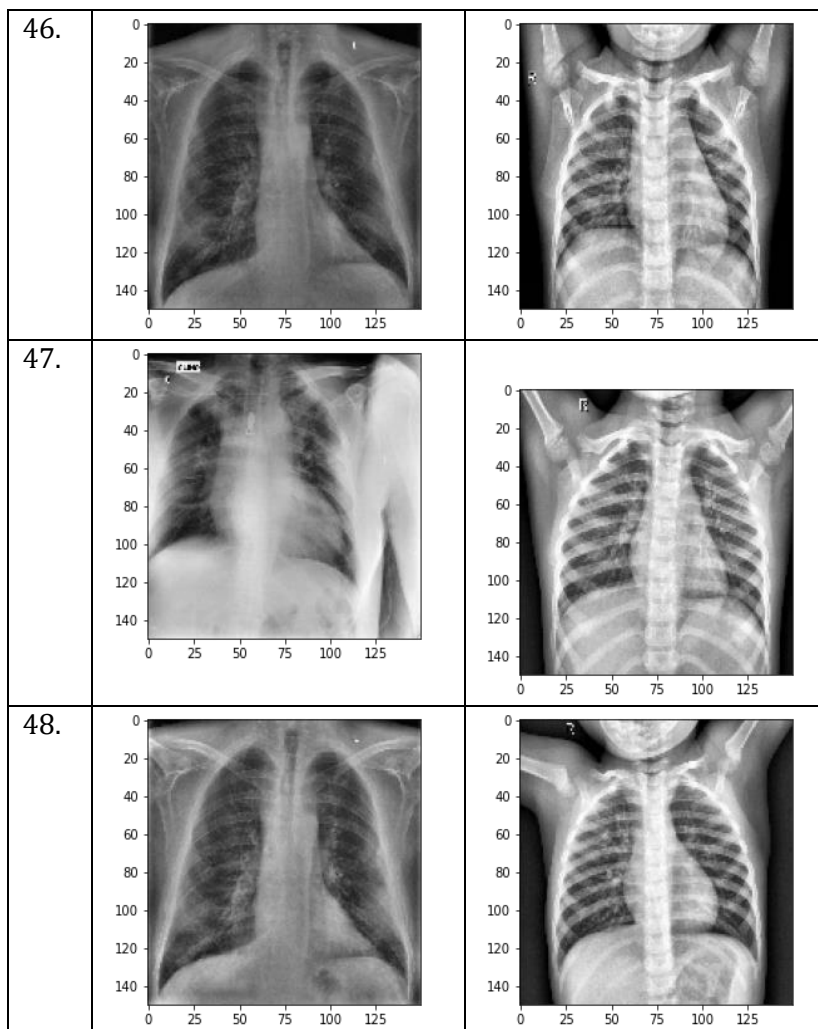


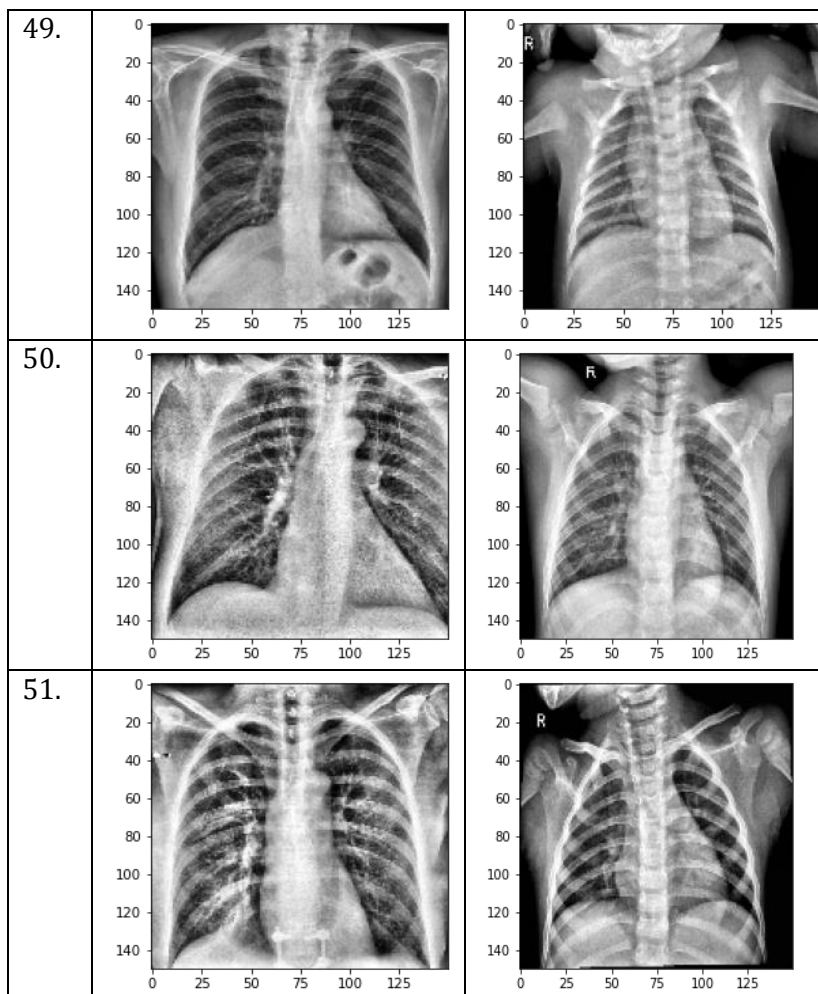


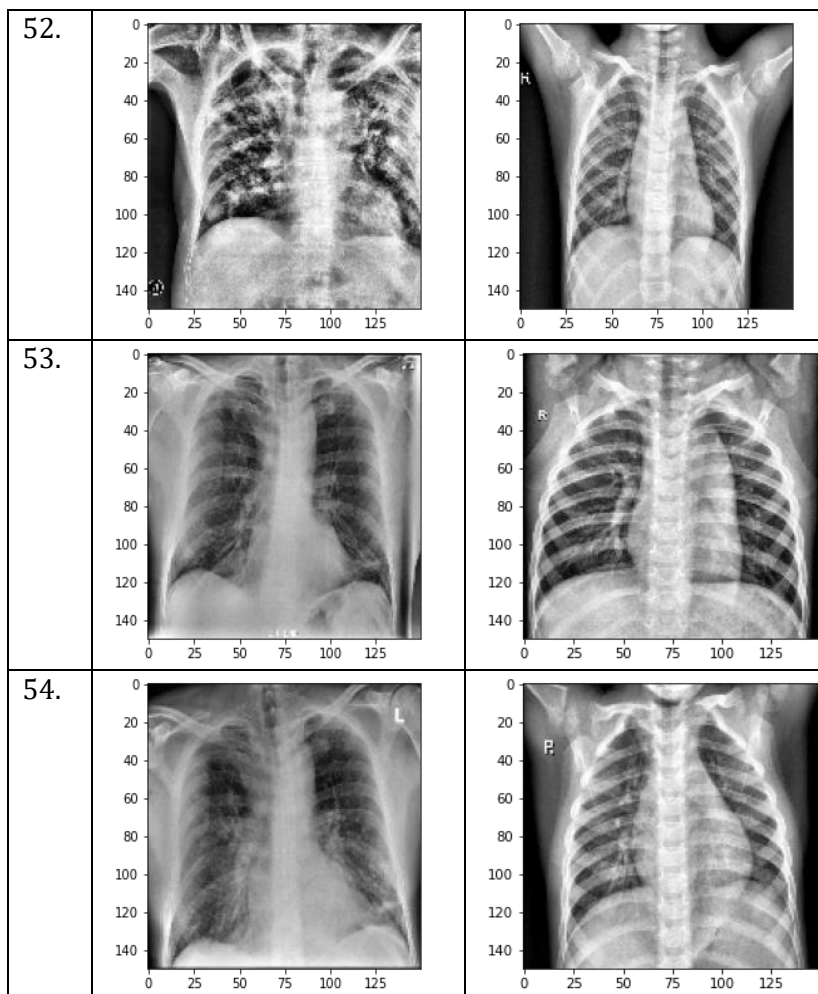


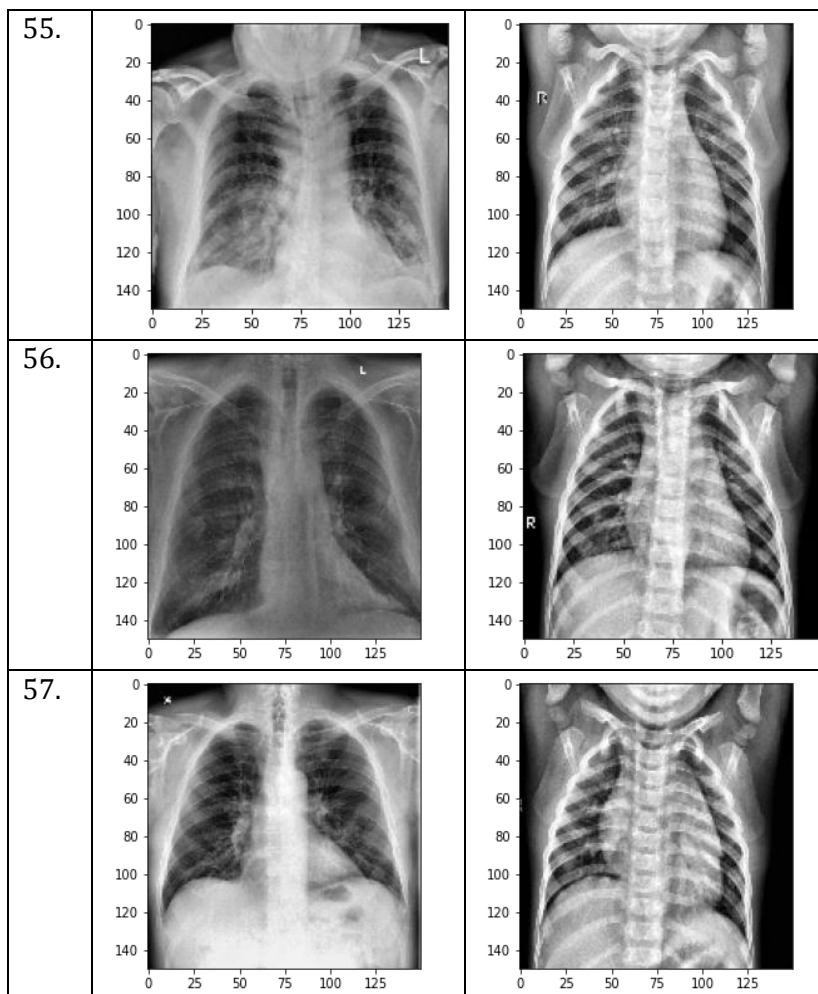


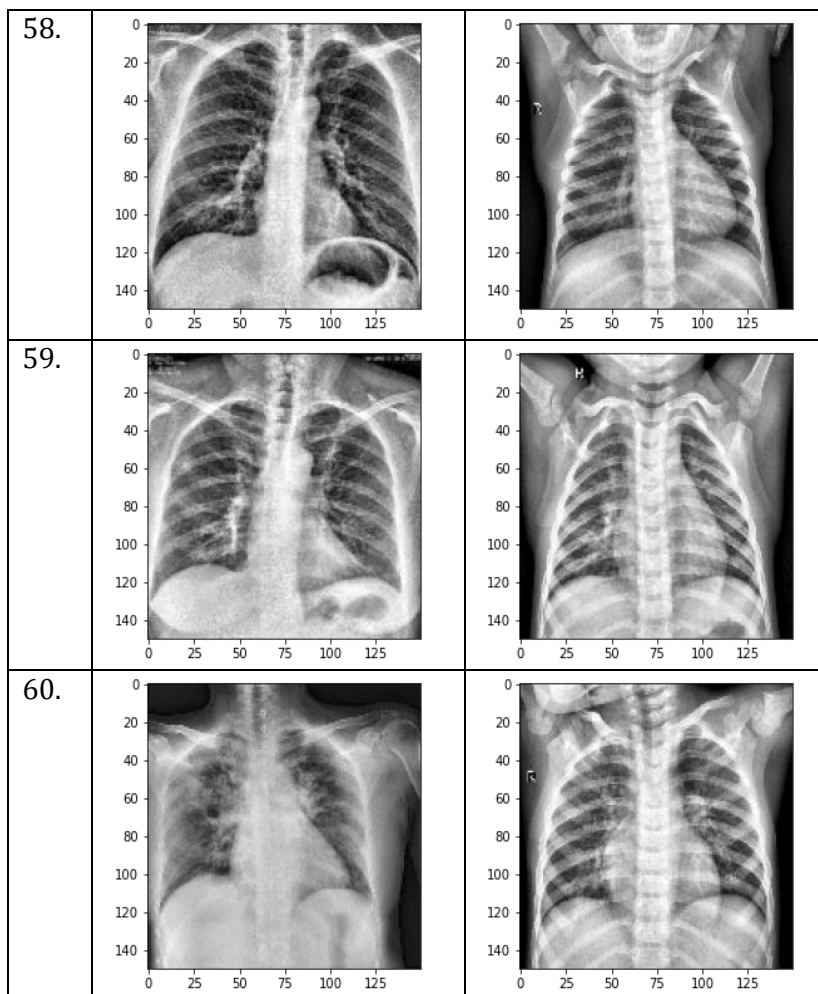


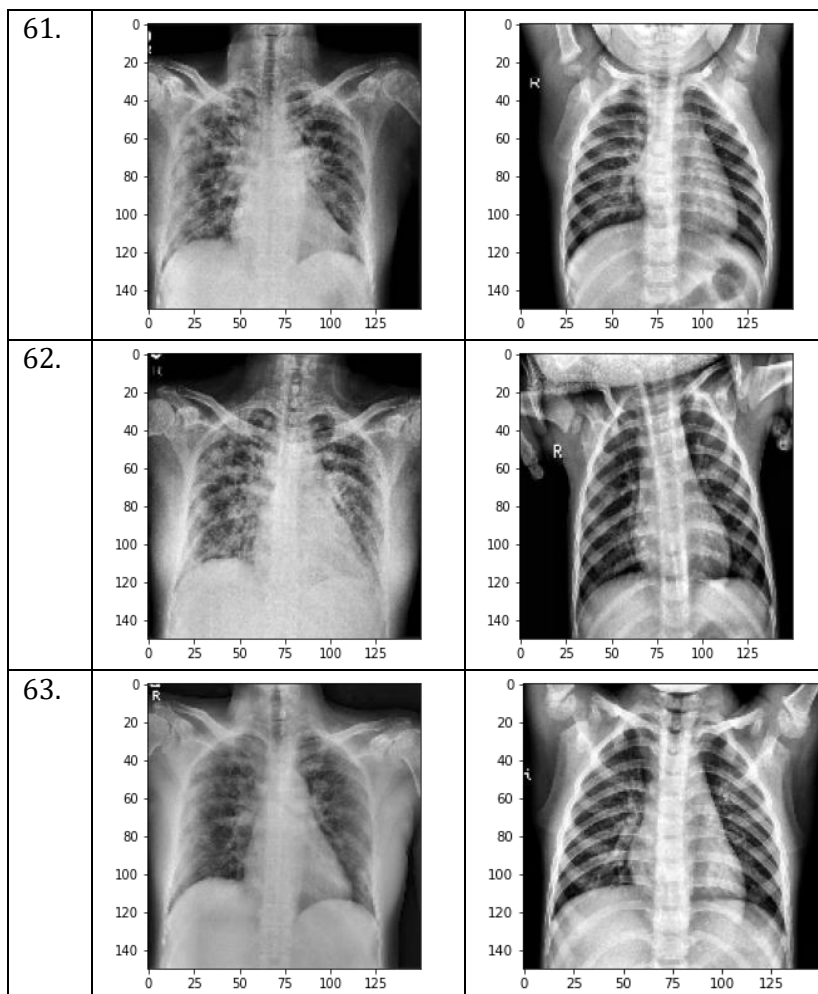


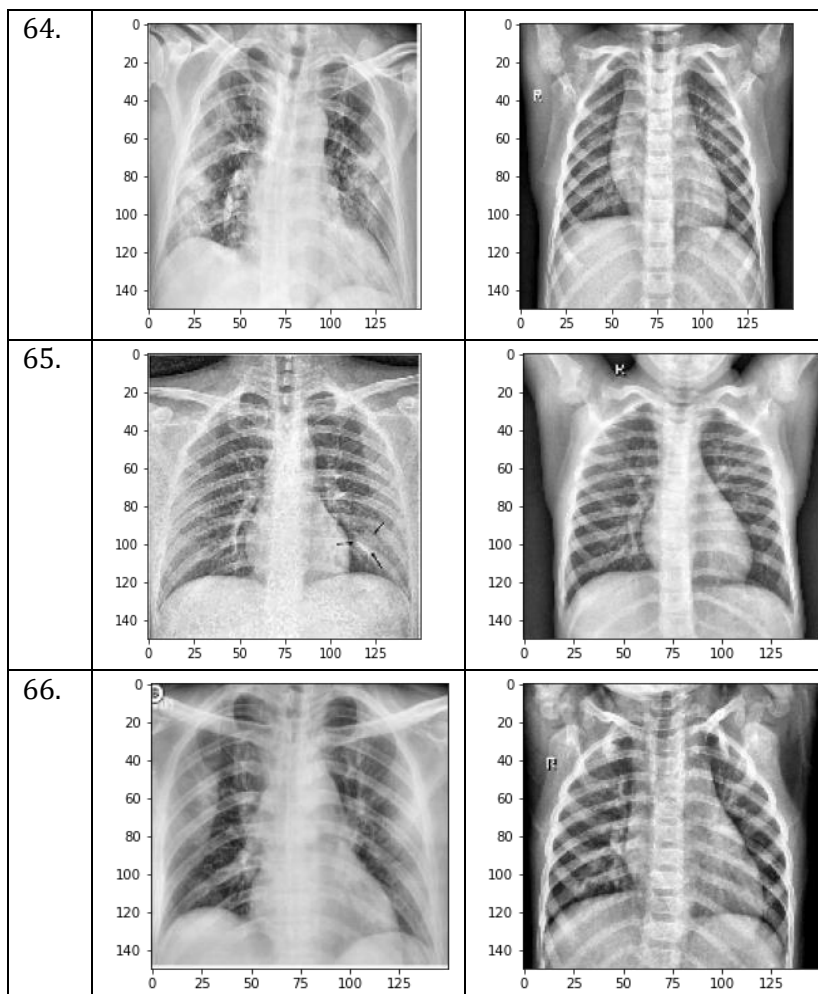


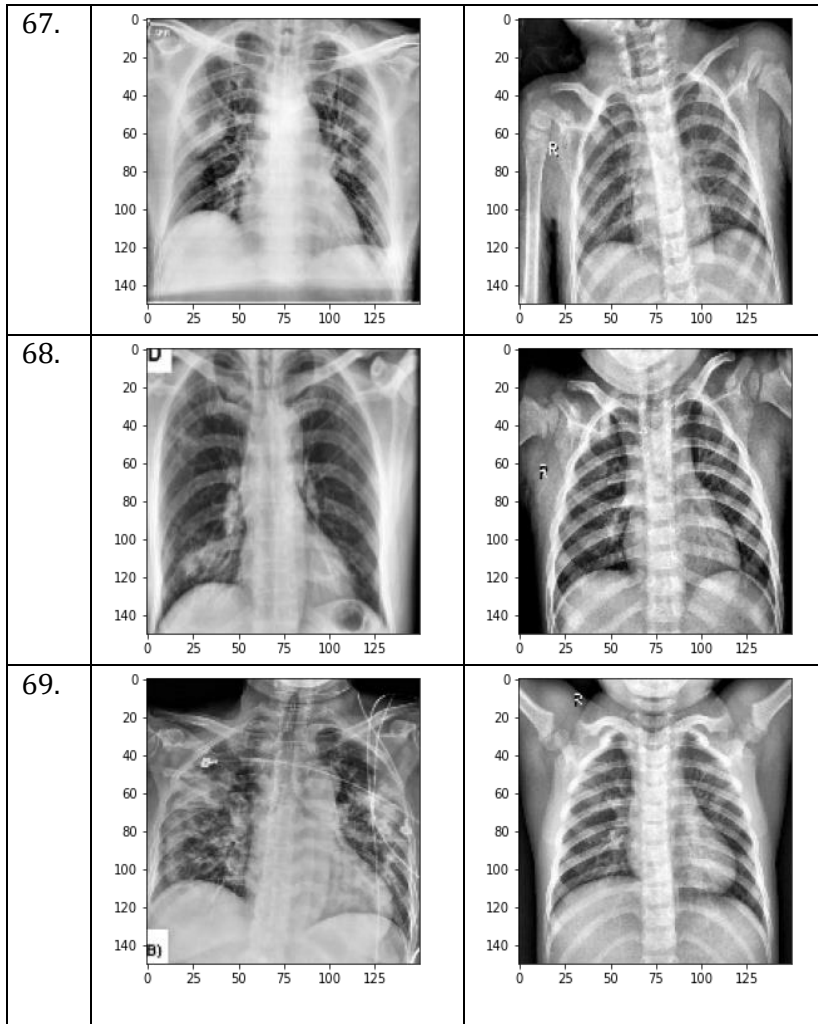


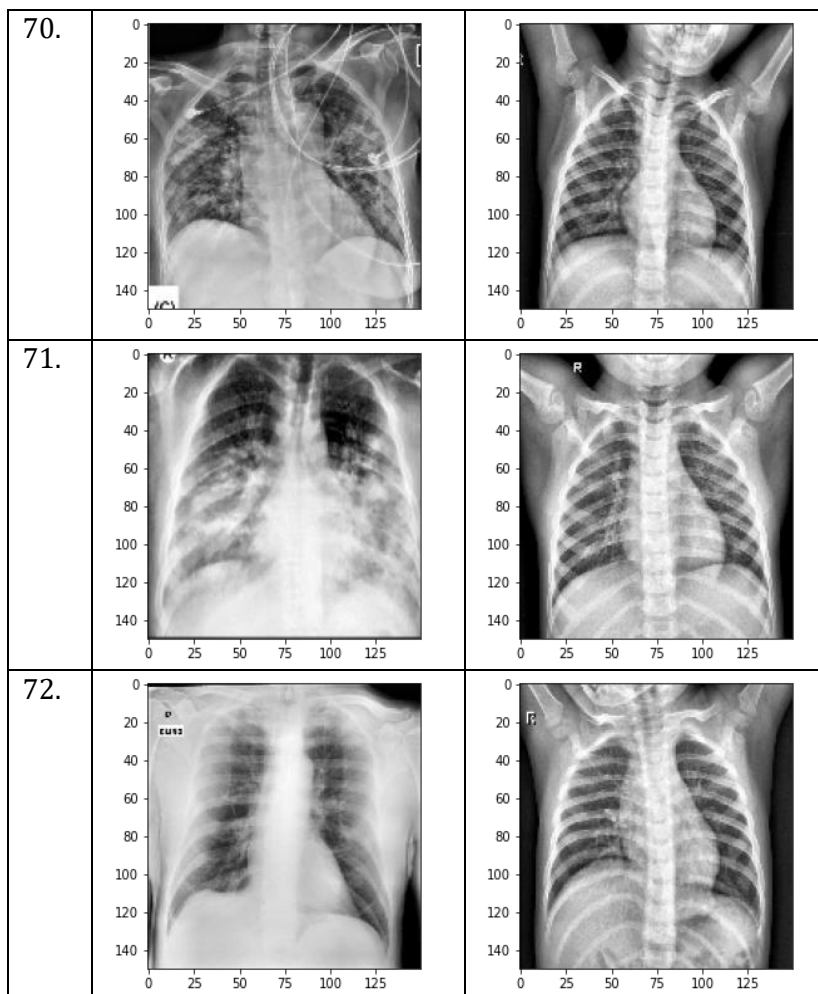


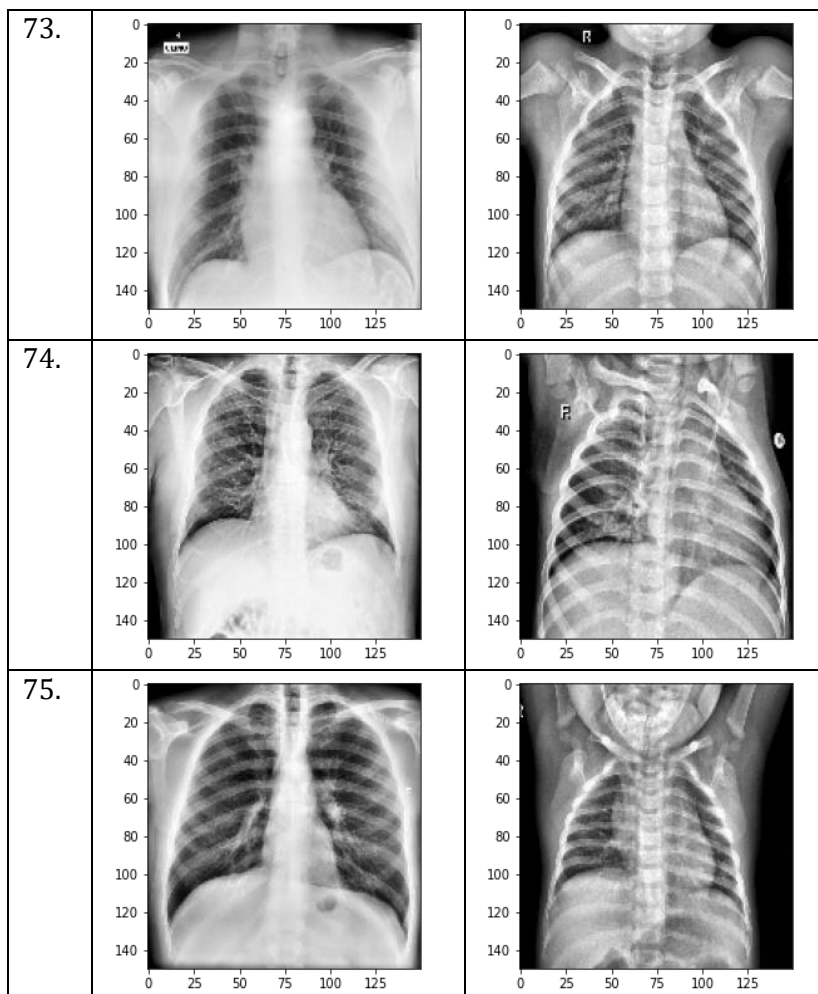


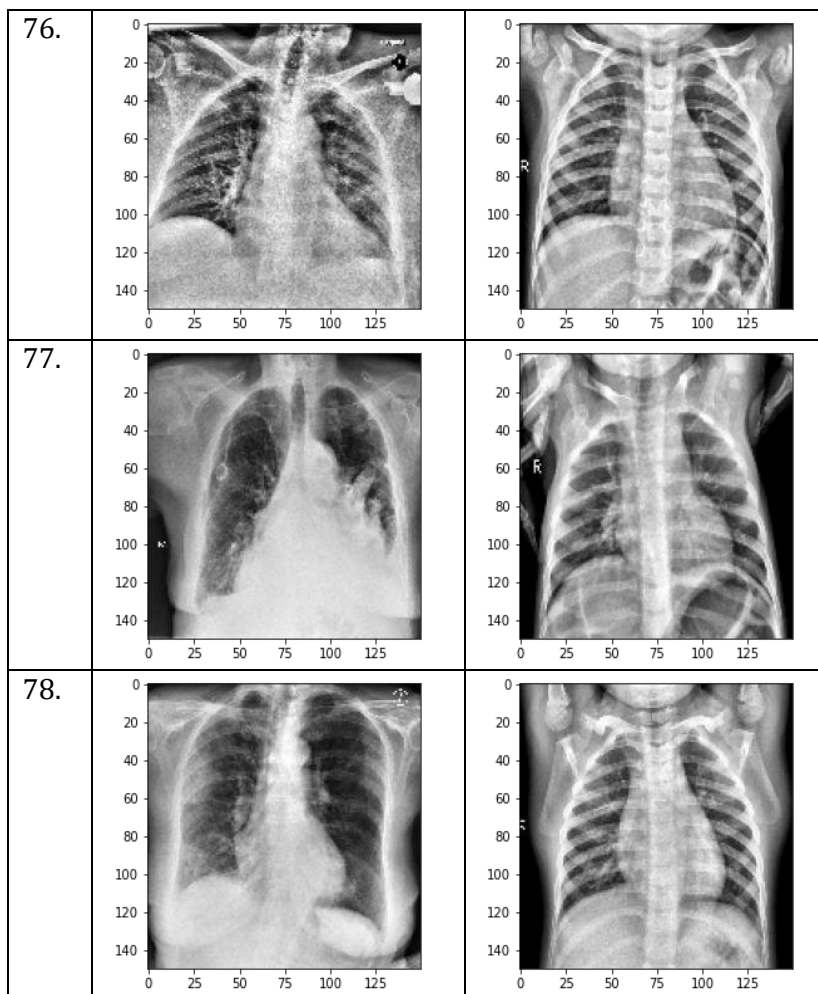


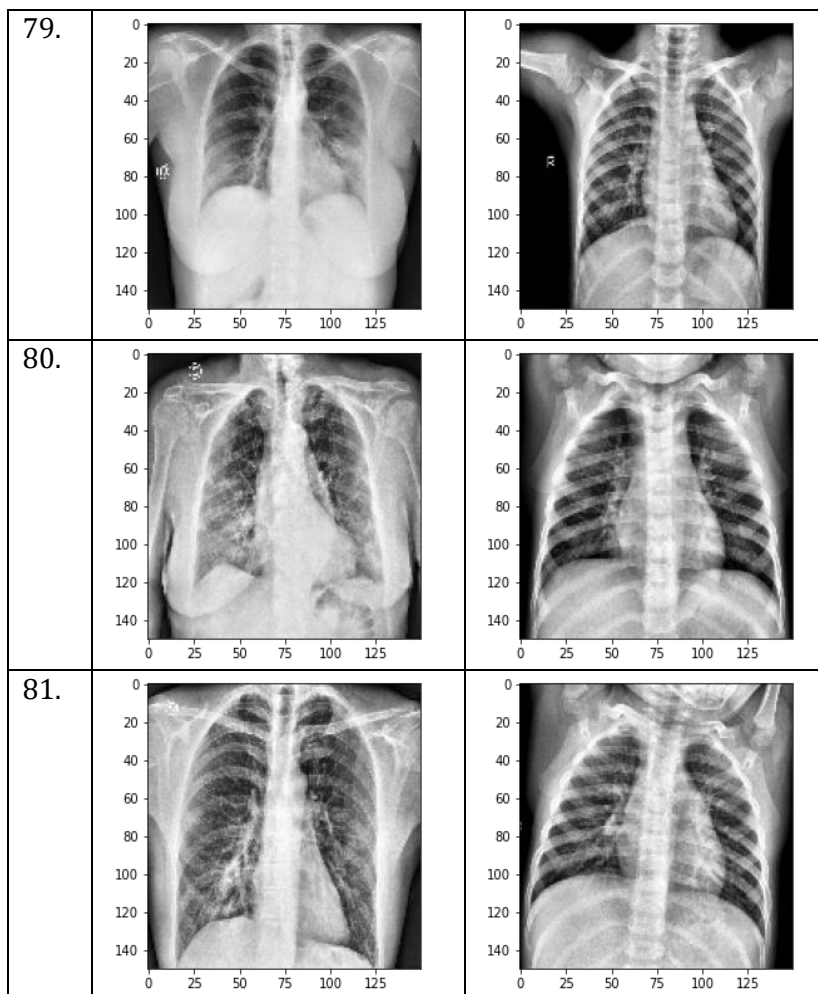


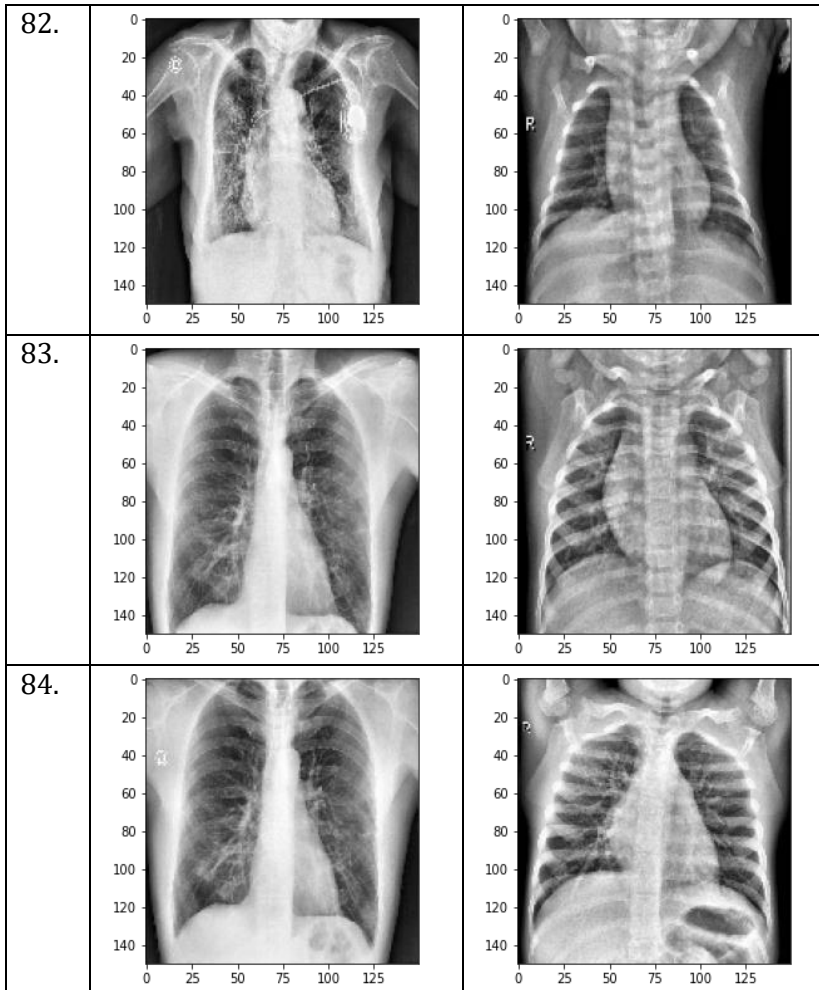


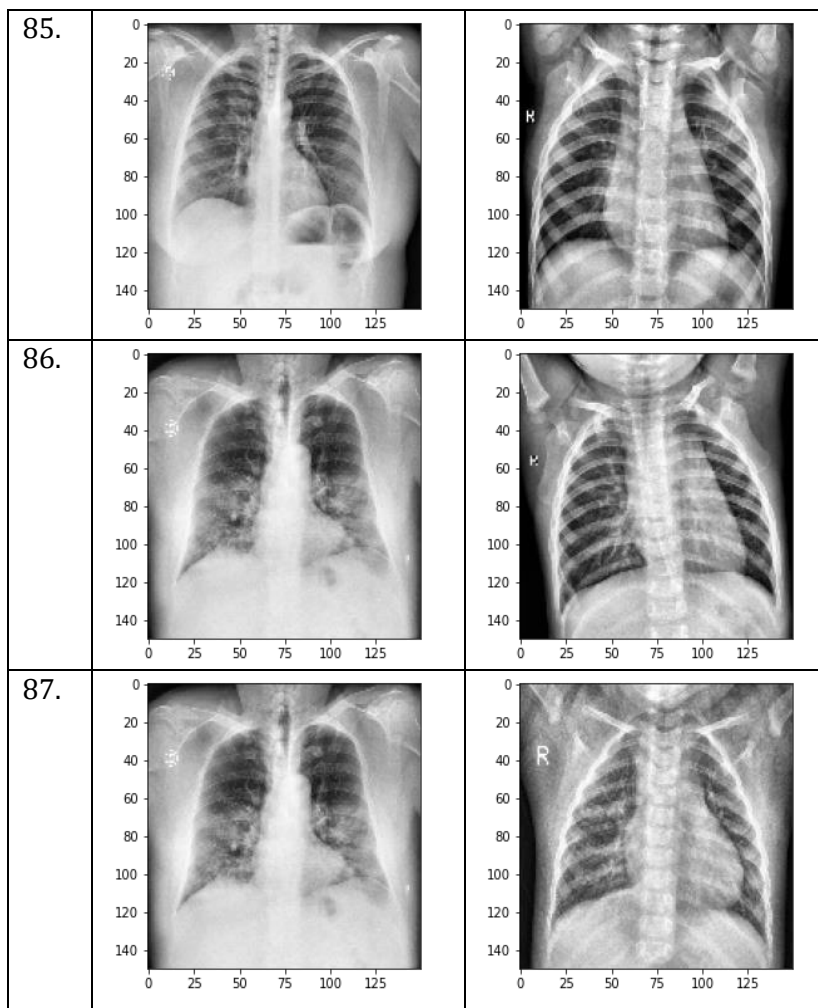


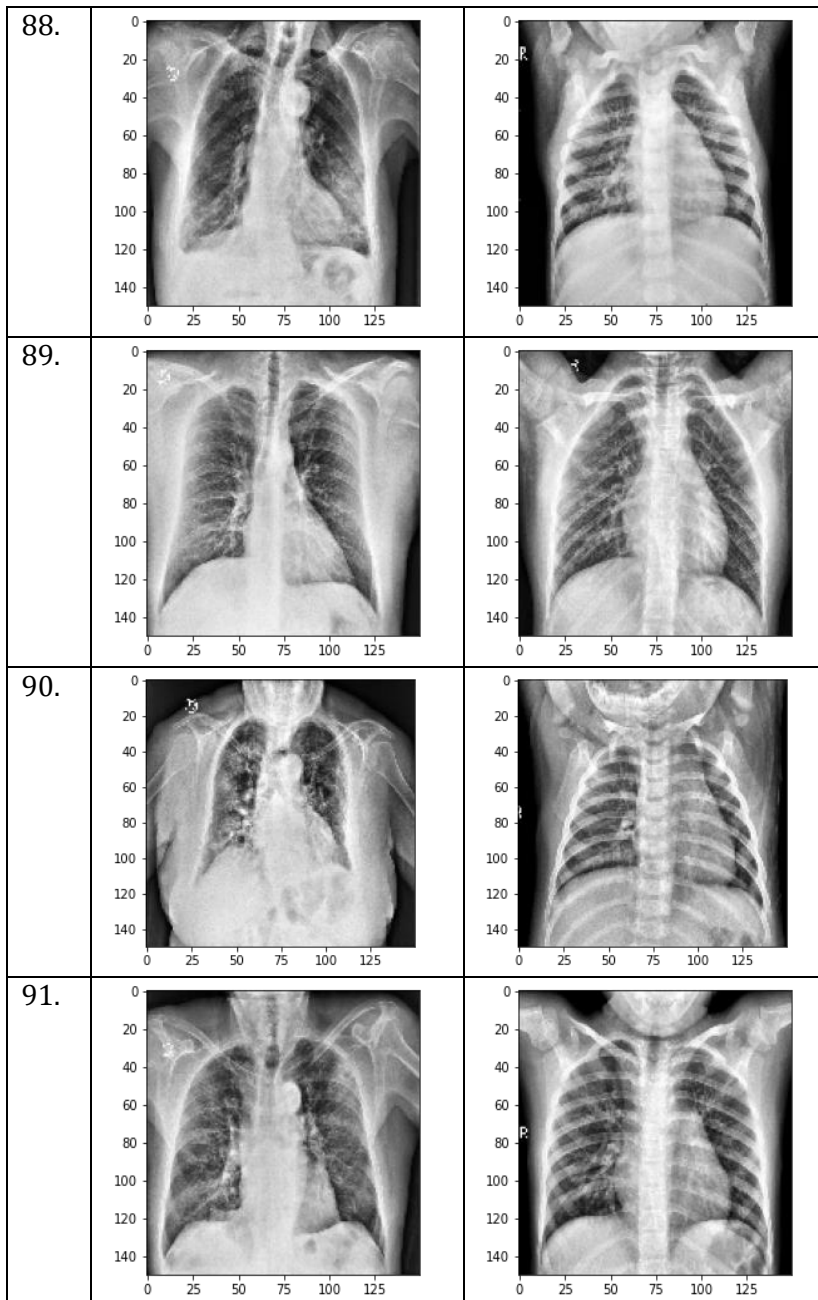


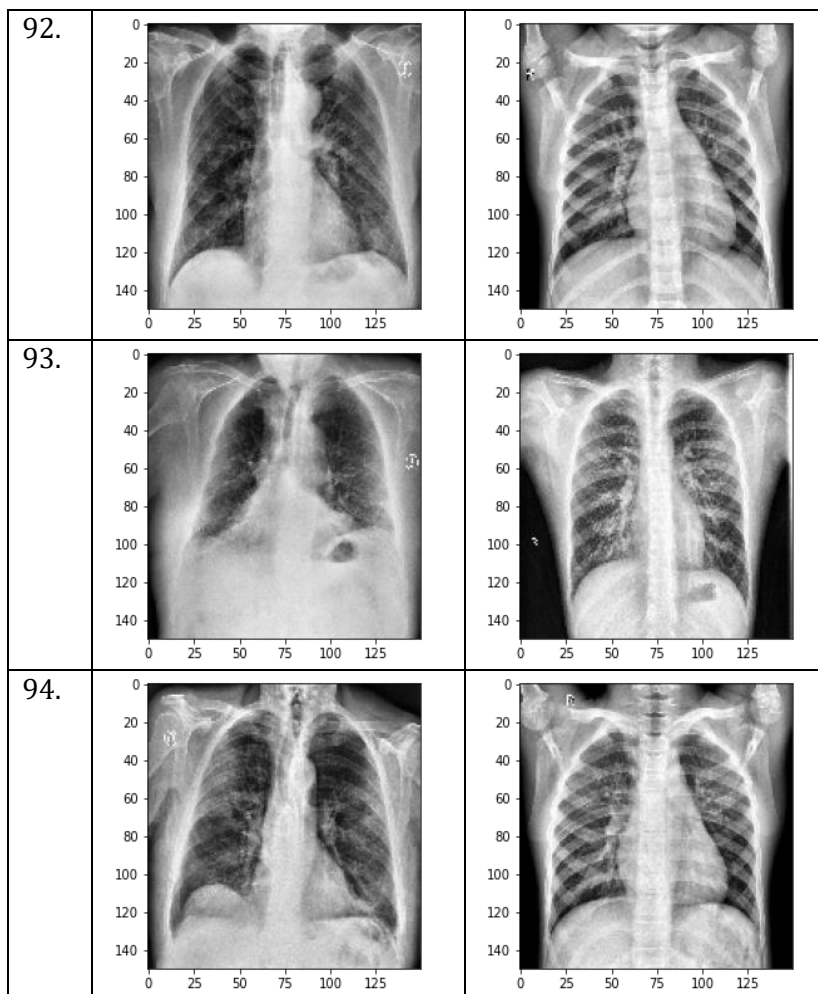


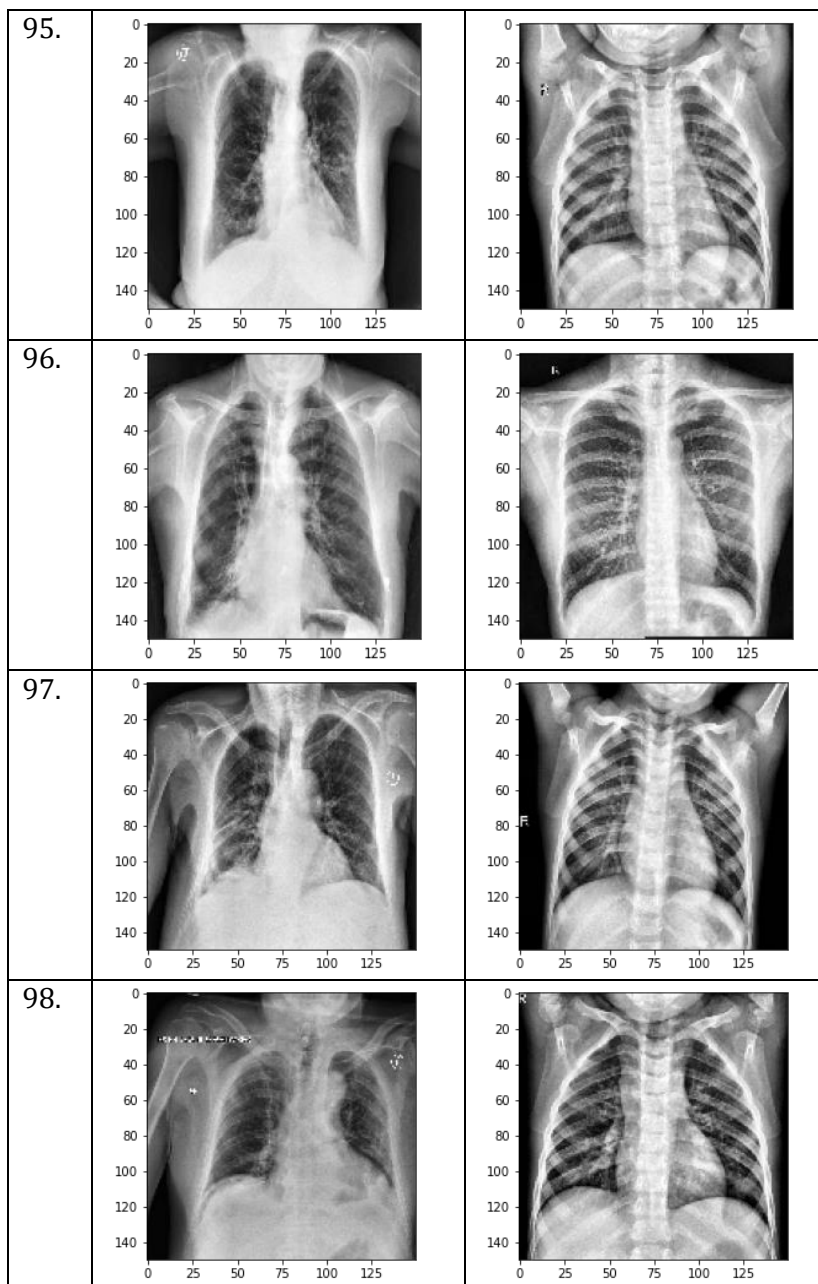


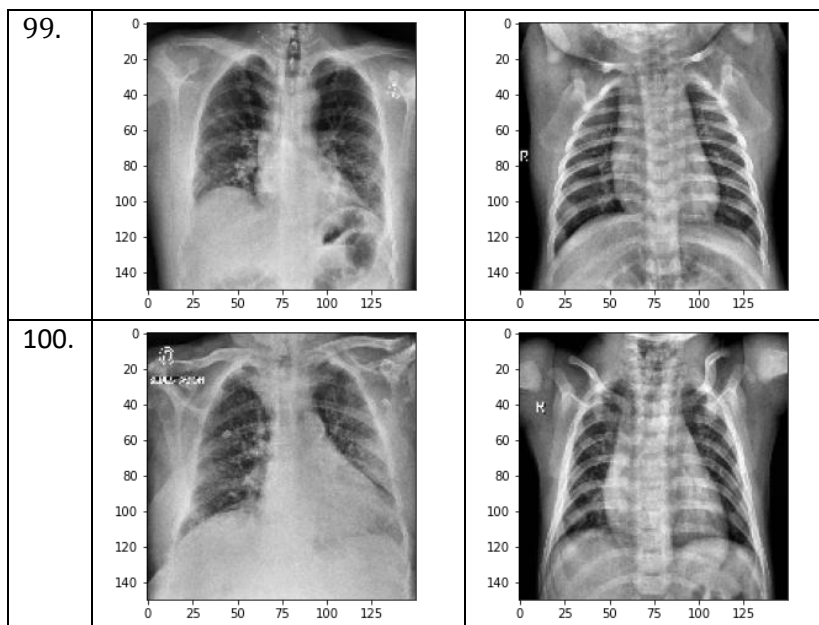












Lampiran 3: Algoritma sistem klasifikasi dengan metode CNN

SISTEM KLASIFIKASI COVID-19 DAN NORMAL DENGAN METODE CNN MENGGUNAKAN SOFTWARE PYTHON

Menambahkan library yang digunakan

```
import numpy as np
import pandas as pd
import os
from skimage.io import imread, imsave, imshow
import matplotlib.pyplot as plt
import seaborn as sn
import cv2
import glob
```

Memanggil dataset

```
datadir= 'C:/Users/ThinkPad X250/Documents/Python
Skripsi/Citra untuk skripsi'

file= 'C:/Users/ThinkPad X250/Documents/Python
Skripsi/hasil'

label = 0

im_arr = []
lb_arr = []

X = []
y = []

for i in os.listdir(datadir): #loop all directory
    count = 0
```

```

for pic in glob.glob('C:/Users/ThinkPad
X250/Documents/Python Skripsi/Citra untuk
skripsi/'+i+'/*'):
    img = cv2.imread(pic)
    img = cv2.resize(img,(150,150))
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    clahe = cv2.createCLAHE(clipLimit=2.0,
tileGridSize=(8,8))
    clahe= clahe.apply(gray)
    count = count + 1
    X.append(clahe)
    y.append(label)
    if(count <= 10):
        im_arr.append({str(i):clahe})
    print("Jumlah "+str(i)+" : "+str(count))
    label = label + 1
    lb_arr.append(i)
X = np.array(X)
y = np.array(y);

```

Out:

```

Jumlah COVID : 100
Jumlah NORMAL : 100

```

X[0]

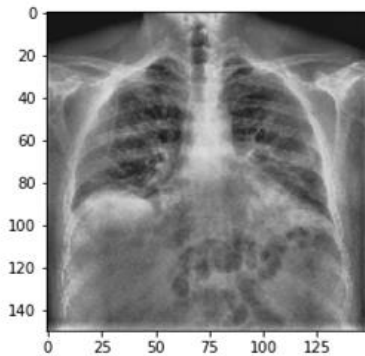
Out:

```
array([[ 13,  5,  5, ...,  6,  6,  6],
       [ 11,  5,  5, ...,  6,  6,  6],
       [ 13,  5,  5, ...,  6,  6,  6],
       ...,
       [ 47, 59, 76, ..., 66, 87, 141],
       [ 63, 88, 105, ..., 90, 112, 155],
       [114, 116, 138, ..., 100, 255, 160]], dtype=uint8)
```

```
plt.imshow(X[0])
```

Out :

```
<matplotlib.image.AxesImage at 0x16cb1ee2640>
```



```
#visualisasi data dengan matplotlib
```

```
fig,axs =plt.subplots (3,7, figsize=(20,10))
```

```
cnt=0
```

```
row=0
```

```
col=0
```

```
for i in im_arr:
```

```
    for key,value in i.items():
```

```
        if (cnt==7):
```

```
            row=row+1
```

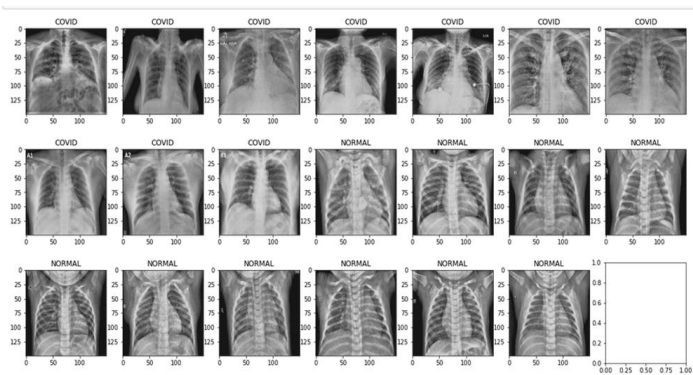


```

col= 0
cnt= 0
axs[row, col].imshow(value)
axs[row, col].set_title(key)
cnt=cnt+1
col=col+1
plt.show()

```

Out:



#Pembagian Dataset

#load library untuk train test split

```

from sklearn.model_selection import train_test_split
from keras.utils import to_categorical

from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

X_train, X_test, y_train, y_test =
train_test_split(X,y,test_size=0.25, random_state=101)

X_train = X_train.astype('float32')#Cara terbaik untuk
mengubah tipe data dari array yang ada, adalah membuat
salinan dari array dengan metode astype()

X_test = X_test.astype('float32') #Set X_test data type as
float32

X_train /= 255 #change x_train value between 0-1
X_test /= 255 #change x_train value between 0-1

X_train = np.reshape(X_train, (len(X_train), 150, 150, 1))
X_test = np.reshape(X_test, (len(X_test), 150, 150, 1))

y_train = to_categorical(y_train, 2) #change label to
binary/categorical: [1 0 0 0]= 0, [0 1 0 0]=1

y_test = to_categorical (y_test, 2) #change label to
binary/categorical

X_train [0]

```

Out:

```

array([[ [0.01960784],
         [0.02745098],
         [0.01960784],
         ...],
       [0.01176471],

```

```
[0.01176471],  
[0.01176471]],  
  
[[0.7529412 ],  
[0.6901961 ],  
[0.68235296],  
... ,  
[0.01960784],  
[0.01960784],  
[0.01960784]],  
  
[[0.99607843],  
[0.4117647 ],  
[0.01960784],  
... ,  
[0.01960784],  
[0.01960784],  
[0.01960784]],  
  
... ,  
  
[[0.05098039],  
[0.01960784],  
[0.01960784],  
... ,  
[0.01960784],  
[0.01960784],  
[0.01960784]],  
  
[[0.05098039],  
[0.01176471],  
[0.01176471],  
... ,  
[0.01176471],  
[0.01176471],  
[0.01176471]],  
  
[[0.07058824],  
[0.01960784],  
[0.01960784],  
... ,  
[0.01960784],  
[0.01960784],
```



```
[0., 1.],
[0., 1.],
[1., 0.],
[1., 0.],
[0., 1.],
[1., 0.],
[1., 0.],
[0., 1.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[0., 1.],
[1., 0.],
[1., 0.],
[0., 1.],
[0., 1.],
[0., 1.],
[0., 1.],
[1., 0.],
[1., 0.],
[1., 0.],
[0., 1.],
[0., 1.],
[1., 0.],
[1., 0.],
[0., 1.],
[1., 0.],
[1., 0.],
[0., 1.],
[0., 1.],
[0., 1.],
[0., 1.],
[1., 0.],
[1., 0.],
[0., 1.],
```

```
[1., 0.],
[0., 1.],
[1., 0.],
[0., 1.],
[1., 0.],
[0., 1.],
[1., 0.],
[0., 1.],
[1., 0.],
[1., 0.],
[0., 1.],
[0., 1.],
[1., 0.],
[1., 0.],
[0., 1.],
[0., 1.],
[0., 1.],
[1., 0.],
[0., 1.],
[0., 1.],
[0., 1.],
[1., 0.],
[0., 1.],
[0., 1.],
[0., 1.],
[1., 0.],
[0., 1.],
[0., 1.],
[1., 0.],
[0., 1.],
[0., 1.],
[0., 1.],
[1., 0.],
[0., 1.],
[0., 1.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[1., 0.],
[0., 1.],
[0., 1.],
[0., 1.],
[1., 0.],
[0., 1.],
[0., 1.],
[1., 0.],
[1., 0.],
[1., 0.],
```

```
[0., 1.],  
[1., 0.],  
[0., 1.],  
[0., 1.],  
[0., 1.],  
[0., 1.],  
[0., 1.],  
[1., 0.],  
[1., 0.],  
[0., 1.],  
[0., 1.],  
[1., 0.],  
[0., 1.],  
[1., 0.],  
[0., 1.],  
[0., 1.],  
[0., 1.],  
[1., 0.],  
[0., 1.],  
[1., 0.],  
[1., 0.],  
[1., 0.],  
[1., 0.],  
[1., 0.],  
[1., 0.],  
[1., 0.],  
[1., 0.], dtype=float32)
```

len(X_test)

Out: 50

Memuat package yang dibutuhkan

```
import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers

from tensorflow.keras.models import Sequential

from tensorflow.keras.preprocessing.image import
ImageDataGenerator

from keras.layers import Conv2D, MaxPooling2D, Dropout,
Flatten, Dense, BatchNormalization, MaxPool2D

# With data augmentation to prevent overfitting and handling
the imbalance in dataset

datagen = ImageDataGenerator(
    featurewise_center=False, # set input mean to 0 over the
dataset
    samplewise_center=False, # set each sample mean to 0
    featurewise_std_normalization=False, # divide inputs by
std of the dataset
    samplewise_std_normalization=False, # divide each input
by its std
    zca_whitening=False, # apply ZCA whitening
    rotation_range = 30, # randomly rotate images in the
range (degrees, 0 to 180)
    zoom_range = 0.2, # Randomly zoom image
    width_shift_range=0.1, # randomly shift images
horizontally (fraction of total width)
    height_shift_range=0.1, # randomly shift images
vertically (fraction of total height)
```



```
horizontal_flip = True, # randomly flip images
vertical_flip=False) # randomly flip images
datagen.fit(X_train)

#arsitektur

model = Sequential() #model = Sequential
model.add(Conv2D(32 , kernel_size=(3,3), activation = 'relu' ,
input_shape = (150,150,1))) #layer convolutional 2D
model.add(MaxPooling2D(pool_size=(2,2))) #max pooling
with stride (2,2)
model.add(Conv2D(32 , (3,3), activation = 'relu')) #layer
convolutional 2D
model.add(MaxPooling2D(pool_size=(2,2))) #max pooling
with stride (2,2)
model.add(Dropout(0.25)) # delete neuron randomly while
training and remain 75%
model.add(Flatten()) # make layer flatten
model.add(Dense(128 , activation = 'relu'))#fully connected
layer
model.add(Dropout(0.5)) #delete neuron randomly and
remain 50%
model.add(Dense(2 , activation = 'softmax'))

#compile

from keras.optimizers import SGD
epochs=25
lr=0.01
decay=lr/epochs
```

```
model.compile(optimizer = "ADAM", loss =
'categorical_crossentropy', metrics = ['accuracy'])
```

```
model.summary()
```

Out:

```
Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)             (None, 148, 148, 32)       320
-----
max_pooling2d (MaxPooling2D) (None, 74, 74, 32)         0
-----
conv2d_1 (Conv2D)           (None, 72, 72, 32)         9248
-----
max_pooling2d_1 (MaxPooling2 (None, 36, 36, 32)         0
-----
dropout (Dropout)           (None, 36, 36, 32)         0
-----
flatten (Flatten)           (None, 41472)              0
-----
dense (Dense)                (None, 128)                5308544
-----
dropout_1 (Dropout)         (None, 128)                0
-----
dense_1 (Dense)             (None, 2)                  258
-----
Total params: 5,318,370
Trainable params: 5,318,370
Non-trainable params: 0
-----
```

```
from keras.callbacks import ReduceLROnPlateau
```

```
learning_rate_reduction =
ReduceLROnPlateau(monitor='val_accuracy', patience = 2,
verbose=1,factor=0.3, min_lr=0.01)
```

#Training

```
history = model.fit(datagen.flow(X_train,y_train, batch_size =
32), epochs = epochs ,validation_data = datagen.flow(X_test,
y_test) ,callbacks = [learning_rate_reduction])
```

out:

```
Epoch 1/25
5/5 [=====] - 3s
666ms/step - loss: 1.2735 - accuracy: 0.5400 -
val_loss: 0.7489 - val_accuracy: 0.4600
Epoch 2/25
5/5 [=====] - 3s
536ms/step - loss: 0.7235 - accuracy: 0.5333 -
val_loss: 0.6776 - val_accuracy: 0.4600
Epoch 3/25
5/5 [=====] - 3s
515ms/step - loss: 0.6457 - accuracy: 0.5533 -
val_loss: 0.6402 - val_accuracy: 0.4800
Epoch 4/25
5/5 [=====] - 3s
593ms/step - loss: 0.5829 - accuracy: 0.6200 -
val_loss: 0.5455 - val_accuracy: 0.8400
Epoch 5/25
5/5 [=====] - 3s
579ms/step - loss: 0.4909 - accuracy: 0.8867 -
val_loss: 0.4397 - val_accuracy: 0.8400
Epoch 6/25
5/5 [=====] - 3s
534ms/step - loss: 0.4053 - accuracy: 0.8533 -
val_loss: 0.3004 - val_accuracy: 0.9200
Epoch 7/25
5/5 [=====] - 3s
536ms/step - loss: 0.3832 - accuracy: 0.8333 -
val_loss: 0.2473 - val_accuracy: 0.9800
Epoch 8/25
5/5 [=====] - 3s
587ms/step - loss: 0.3247 - accuracy: 0.8933 -
val_loss: 0.1544 - val_accuracy: 0.9600
Epoch 9/25
```

```
5/5 [=====] - 3s
661ms/step - loss: 0.2363 - accuracy: 0.9333 -
val_loss: 0.1502 - val_accuracy: 0.9200
Epoch 10/25
5/5 [=====] - 3s
580ms/step - loss: 0.3916 - accuracy: 0.8667 -
val_loss: 0.1382 - val_accuracy: 0.9800
Epoch 11/25
5/5 [=====] - 3s
544ms/step - loss: 0.1692 - accuracy: 0.9467 -
val_loss: 0.1520 - val_accuracy: 0.9400
Epoch 12/25
5/5 [=====] - 3s
525ms/step - loss: 0.1674 - accuracy: 0.9333 -
val_loss: 0.1393 - val_accuracy: 0.9800
Epoch 13/25
5/5 [=====] - 3s
615ms/step - loss: 0.1938 - accuracy: 0.9000 -
val_loss: 0.1059 - val_accuracy: 0.9600
Epoch 14/25
5/5 [=====] - 3s
545ms/step - loss: 0.1403 - accuracy: 0.9467 -
val_loss: 0.1577 - val_accuracy: 0.9200
Epoch 15/25
5/5 [=====] - 3s
608ms/step - loss: 0.1535 - accuracy: 0.9400 -
val_loss: 0.1354 - val_accuracy: 0.9600
Epoch 16/25
5/5 [=====] - 3s
543ms/step - loss: 0.1173 - accuracy: 0.9467 -
val_loss: 0.0911 - val_accuracy: 0.9400
Epoch 17/25
5/5 [=====] - 3s
568ms/step - loss: 0.1870 - accuracy: 0.9000 -
val_loss: 0.1527 - val_accuracy: 0.9400
Epoch 18/25
5/5 [=====] - 3s
533ms/step - loss: 0.1696 - accuracy: 0.9400 -
val_loss: 0.1270 - val_accuracy: 0.9400
Epoch 19/25
5/5 [=====] - 3s
533ms/step - loss: 0.1486 - accuracy: 0.9333 -
val_loss: 0.2122 - val_accuracy: 0.9400
```

```
Epoch 20/25
5/5 [=====] - 3s
541ms/step - loss: 0.1241 - accuracy: 0.9400 -
val_loss: 0.2397 - val_accuracy: 0.9400
Epoch 21/25
5/5 [=====] - 3s
584ms/step - loss: 0.1794 - accuracy: 0.9200 -
val_loss: 0.0814 - val_accuracy: 1.0000
Epoch 22/25
5/5 [=====] - 3s
597ms/step - loss: 0.1728 - accuracy: 0.9467 -
val_loss: 0.2468 - val_accuracy: 0.8800
Epoch 23/25
5/5 [=====] - 3s
562ms/step - loss: 0.1328 - accuracy: 0.9533 -
val_loss: 0.0793 - val_accuracy: 0.9600
Epoch 24/25
5/5 [=====] - 3s
533ms/step - loss: 0.1044 - accuracy: 0.9667 -
val_loss: 0.0837 - val_accuracy: 0.9800
Epoch 25/25
5/5 [=====] - 3s
665ms/step - loss: 0.0973 - accuracy: 0.9533 -
val_loss: 0.1379 - val_accuracy: 0.9800
```

```
print("Loss of the model is - ",
      model.evaluate(X_test,y_test)[0]*100, "%")
```

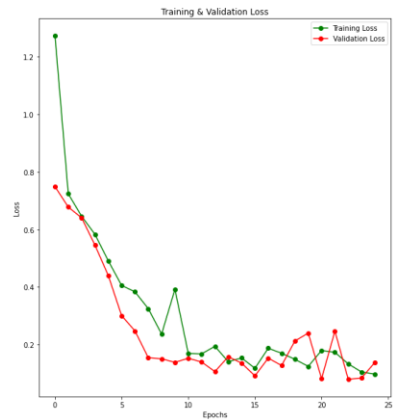
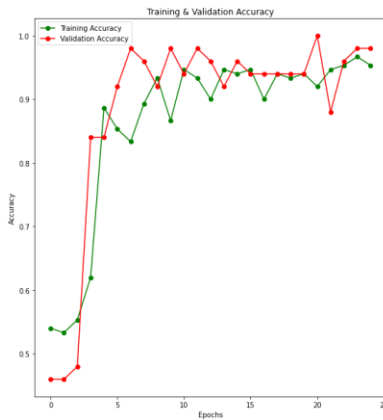
```
print("Accuracy of the model is - ",
      model.evaluate(X_test,y_test)[1]*100, "%")
```

Out:

```
2/2 [=====] - 0s
40ms/step - loss: 0.1061 - accuracy: 0.9600
Loss of the model is - 10.612203925848007 %
2/2 [=====] - 0s
35ms/step - loss: 0.1061 - accuracy: 0.9600
Accuracy of the model is - 95.99999785423279 %
```

```
epochs = [i for i in range(25)]
fig, ax = plt.subplots(1,2)
train_acc = history.history['accuracy']
train_loss = history.history['loss']
val_acc = history.history['val_accuracy']
val_loss = history.history['val_loss']
fig.set_size_inches(20,10)
ax[0].plot(epochs , train_acc , 'go-' , label = 'Training
Accuracy')
ax[0].plot(epochs , val_acc , 'ro-' , label = 'Validation
Accuracy')
ax[0].set_title("Training & Validation Accuracy")
ax[0].legend()
ax[0].set_xlabel("Epochs")
ax[0].set_ylabel("Accuracy")
ax[1].plot(epochs , train_loss , 'g-o' , label = 'Training Loss')
```

```
ax[1].plot(epochs , val_loss , 'r-o' , label = 'Validation Loss')
ax[1].set_title("Training & Validation Loss")
ax[1].legend()
ax[1].set_xlabel("Epochs")
ax[1].set_ylabel("Loss")
plt.show()
Out:
```



```
#Prediksi
```

```
y_pred = model.predict(X_test)
```

```
# iterasi X_test
```

```
y_prediction_n=[]
```

```
y_test_n=[]
```

```
for i in range (0, len (y_pred)):
```

```
    max_n=max(y_pred[i])
```

```
    max_m=max(y_test[i])
```

```

index_n=[index_n for index_n,j in enumerate (y_pred[i]) if
j==max_n]

index_m=[index_m for index_m,j in enumerate (y_test[i]) if
j==max_m]

y_prediction_n.append(index_n)

y_test_n.append(index_m)

```

#CLASSIFICATION REPORT

```

from sklearn.metrics import classification_report

print(classification_report(y_test_n, y_prediction_n,
target_names = ["COVID","NORMAL" ]))

```

Out :

	precision	recall	f1-score	support
COVID	0.93	1.00	0.96	27
NORMAL	1.00	0.91	0.95	23
accuracy			0.96	50
macro avg	0.97	0.96	0.96	50
weighted avg	0.96	0.96	0.96	50

```
y_pred = model.predict_classes(X_test)
```

y_pred

Out :

```

array([0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0,
1, 1, 0, 0, 1, 0, 1, 1,
        1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0,
0, 0, 1, 1, 0, 1, 1, 1,
        0, 0, 0, 0, 0, 0], dtype=int64)

```

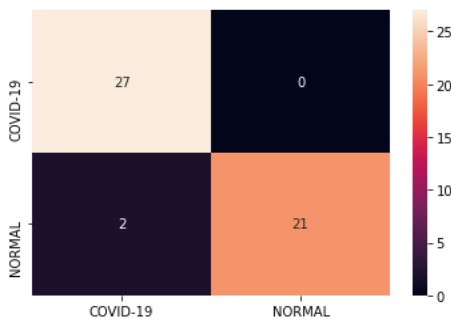


```
y_test_class=np.argmax(y_test,axis=1)
from sklearn.metrics import confusion_matrix,accuracy_score
import seaborn as sn
print (accuracy_score(y_test_class,y_pred))
confusion_matrix (y_test_class,y_pred)
```

```
Out: 0.96
array([[27,  0],
       [ 2, 21]], dtype=int64)
```

```
cm = confusion_matrix(y_test_class,y_pred)
sn.heatmap(cm, annot = True, xticklabels = ["COVID-19",
"NORMAL"],yticklabels = ["COVID-19", "NORMAL"])
```

Out :



Lampiran 4: Daftar Istilah

CNN: *Convolutional Neural Networks*, salah satu kelas *deep feed-forward artificial neural networks* yang banyak diaplikasikan pada analisis citra

Convolutional Layer : Lapisan yang merupakan blok bangunan inti dalam CNN, di mana sebagian besar komputasi dilakukan di lapis ini.

Deep learning : Teknik pembelajaran yang memiliki kemampuan *feature engineering* yang dapat merekayasa fitur secara otomatis sehingga kita tidak perlu bersusah payah membangun model ekstraksi fitur yang seringkali sangat rumit. *Deep learning* terbukti mampu memberikan peningkatan akurasi yang sebanding dengan penambahan jumlah data.

Epoch : *Epoch* adalah ketika seluruh dataset sudah melalui proses training pada CNN sampai dikembalikan ke awal dalam satu putaran.

Fully Connected layer : Lapisan CNN yang terhubung secara penuh, di mana setiap *neurons* memiliki koneksi penuh ke semua aktivasi dalam lapisan sebelumnya.

F1-Score : *harmonic mean* dari *precision* dan *recall*.

Klasifikasi : Menetapkan objek-objek ke kelas-kelas yang berbeda untuk mempermudah identifikasi. Dengan

kata lain, klasifikasi adalah usaha menggeneralisasi struktur yang diketahui untuk diaplikasikan pada data-data baru.

Loss Layer : Lapisan terakhir dalam CNN, menentukan bagaimana pelatihan memberikan penalti atas penyimpangan antara hasil prediksi dan label.

Normalization Layer : Lapis normalisasi pada CNN untuk mengatasi adanya perbedaan rentang nilai yang signifikan pada citra masukan.

Optimasi : Proses pencarian nilai optimum, yang bisa berupa pencarian nilai maksimum (maksimasi) atau minimum (minimasi). pencarian nilai.

Pooling Layer : Lapisan CNN yang berfungsi menjaga ukuran data ketika *convolution*, yaitu dengan melakukan *downsampling* (pereduksian sampel).

Precision : Perbandingan antara *True Positive* (TP) dengan banyaknya data yang diprediksi positif.

Recall : Perbandingan antara *True Positive* (TP) dengan banyaknya data yang sebenarnya positif.

ReLU : *Rectified Linear Units layer*, lapisan yang mengaplikasikan fungsi aktivasi untuk meningkatkan sifat nonlinearitas fungsi keputusan dan jaringan

secara keseluruhan tanpa mempengaruhi bidang-bidang reseptif pada *convolution layer*.

ResNet : *Residual Network*, suatu arsitektur CNN yang memiliki tiga ciri khusus: koneksi lompat (*skip connections*), menggunakan *batch normalization*, dan menghilangkan *fully connected layers* di bagian akhir. Sejak Mei 2016, ResNets menjadi arsitektur CNN terbaik dan menjadi standar dalam penggunaan secara praktis.

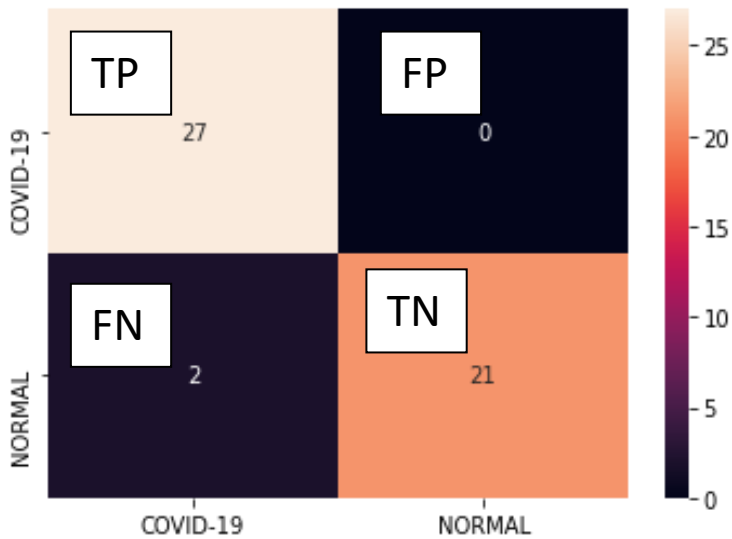
Lampiran 5: Hasil Training model

<i>Epoch</i>	<i>Time (S)</i>	<i>Loss</i>	<i>akurasi</i>	<i>Validation Loss</i>	<i>Validation akurasi</i>
1	4	0.8970	0.5333	0.6689	0.4600
2	3	0.6680	0.5800	0.5931	0.7800
3	3	0.5814	0.7333	0.5161	0.9000
4	3	0.4760	0.8600	0.4044	0.9000
5	3	0.3679	0.8467	0.3123	0.8800
6	3	0.3307	0.8333	0.3556	0.8600
7	3	0.3421	0.8267	0.2858	0.8800
8	3	0.4172	0.7933	0.2777	0.9000
9	3	0.3099	0.9000	0.1930	0.9600
10	3	0.2932	0.9000	0.1890	0.9400
11	3	0.2274	0.9133	0.1459	0.9600
12	3	0.1511	0.9533	0.2595	0.9000
13	3	0.2056	0.9133	0.1451	0.9600
14	3	0.1588	0.9467	0.1802	0.9200
15	3	0.1616	0.9533	0.1759	0.9200
16	3	0.1689	0.9467	0.1331	0.9600
17	5	0.1547	0.9467	0.1436	0.9600
18	3	0.1210	0.9600	0.1991	0.9200
19	3	0.1313	0.9600	0.1430	0.9400
20	3	0.1480	0.9400	0.1762	0.9600
21	3	0.1740	0.9600	0.1322	0.9600
22	3	0.1564	0.9333	0.1042	0.9600
23	7	0.0918	0.9800	0.1117	0.9400
24	4	0.1125	0.9667	0.1149	0.9200
25	6	0.0983	0.9800	0.1563	0.9400

Lampiran 6: Contoh tampilan Confusion matrix 2x2

	precision	recall	f1-score	support
COVID	0.93	1.00	0.96	27
NORMAL	1.00	0.91	0.95	23
accuracy			0.96	50
macro avg	0.97	0.96	0.96	50
weighted avg	0.96	0.96	0.96	50

Lampiran 7: Contoh perhitungan Confusion matrix 2x2



$$\begin{aligned} \text{Presisi} &= \frac{TP}{TP+FP} \times 100\% \\ &= \frac{27}{27+0} \times 100\% = 100\% \end{aligned}$$

$$\begin{aligned} \text{Recall} &= \frac{TP}{TP+FN} \times 100\% \\ &= \frac{27}{27+2} \times 100\% = 93\% \end{aligned}$$

$$\begin{aligned} \text{Accuracy} &= \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \\ &= \frac{27+21}{27+21+2+0} \times 100\% \\ &= \frac{48}{50} \times 100\% \\ &= 96\% \end{aligned}$$

Lampiran 8: Contoh perhitungan Confusion matrix 2x2

Klasifikasi Citra *Chest X-Ray* (CXR) Penyakit COVID-19 dan Normal dengan Metode *Convolutional Neural Network* (CNN) menggunakan *Python*



Lampiran 9: Riwayat Hidup**RIWAYAT HIDUP****A. Identitas Diri**

1. Nama : Nur Chikmi Azizi
2. NIM : 1608026002
3. Tempat,Tgl. Lahir : Bojonegoro, 6 Maret 1999
4. Alamat : Gg Manggis XV No. 1D.
Kel.:Tanjung Duren Selatan,
Kec: Grogol Petamburan,
Jakarta Barat, DKI Jakarta.
5. Email : Chikmiazizi001@gmail.com
6. No. HP : 087834601553

B. Riwayat Pendidikan

1. Pendidikan Formal:
 - a. SDN Tanjung Duren Selatan 07 Pagi Jakarta Lulus 2010
 - b. SMP Negeri 69 Jakarta Lulus 2013
 - c. SMA Negeri 23 Jakarta Lulus 2016
 - d. UIN Walisongo Semarang Lulus 2022
2. Pendidikan Non-Formal:
 - a. Ketua departemen syi'ar ROHIS SMAN 23 2014-2015
 - b. Bendahara Ikatan Mahasiswa Fisika tahun 2017-2019
 - c. Ketua asrama Ikatan Mahasiswa Muhammadiyah 2016-2018
 - d. Tutor MIPA bimbingan belajar Cemerlang 2017-2022

C. Karya Ilmiah

1. Analisis Kualitas Merk Oli Aftermarket berdasarkan parameter viskositas dalam Seminar Nasional Mahasiswa Fisika (SNMF) III tahun 2018 di Universitas Islam Negeri Walisongo Semarang
2. *Sharing and Gathering* Fisika medis Walisongo dalam laman fisika.walisongo.ac.id tahun 2018
3. UIN Walisongo Semarang Terjunkan mahasiswa KKN di Karimunjawa dalam laman javanewsonline.com tahun 2020

Semarang, 12 Desember 2022

Penulis,



Nur Chikmi Azizi

NIM: 1608026002