

**ANALISIS PERBANDINGAN MODEL ARSITEKTUR CNN
DALAM PENDETEKSIAN CORONAVIRUS DISEASE
MENGUNAKAN CITRA X-RAY PARU-PARU
(Studi Kasus : *Baseline CNN, InceptionResnetV2, VGG19,*
dan Xception)**

SKRIPSI

**Diajukan oleh:
AULIA SABRI
NIM. 180705005
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2022 M / 1443 H**

**ANALISIS PERBANDINGAN MODEL ARSITEKTUR CNN DALAM
PENDETEKSIAN *CORONAVIRUS DISEASE* MENGGUNAKAN CITRA
X-RAY PARU-PARU
(Studi Kasus : *Baseline CNN, InceptionResnetV2, VGG19,*
dan Xception)**

SKRIPSI

Diajukan kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri Ar-Raniry Banda Aceh
Sebagai Beban Studi Memperoleh Gelar Sarjana dalam Ilmu Teknologi Informasi

Oleh:

**AULIA SABRI
NIM. 180705005**

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui Untuk Dimunaqasyahkan oleh:

Pembimbing I,

Bustami, M.Sc
NIDN: 2008048601

Pembimbing II,

Ima Dwitawati, M.B.A
NIDN: 0113108204

Mengetahui

Ketua Program Studi Teknologi Informasi

Ima Dwitawati, MBA
NIDN: 0113108204

**ANALISIS PERBANDINGAN MODEL ARSITEKTUR CNN DALAM
PENDETEKSIAN *CORONAVIRUS DISEASE* MENGGUNAKAN CITRA
X-RAY PARU-PARU**

**(Studi Kasus : *Baseline CNN, InceptionResnetV2, VGG19,*
dan Xception)**

SKRIPSI

**Telah diuji oleh Panitia Ujian Munaqasyah Skripsi
Fakultas Sains dan Teknologi UIN Ar-Raniry dan dinyatakan Lulus
Serta diterima sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Ilmu Teknologi Informasi**

Pada Hari/Tanggal: Rabu 20 Juli 2022
21 Dzulhijah 1443 H

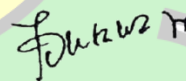
Panitia Ujian Munaqasyah Skripsi

Ketua,



Bustami, M.Sc.
NIDN: 2008048601

Sekretaris,



Ima Dwitawati, M.B.A
NIDN: 0113108204

Penguji I,



Hendri Ahmadian, S.Si., M.IM
NIDN: 2004018303

Penguji II,



Sri Wahyuni, M.T
NIDN: 2027058901

Mengetahui,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Ar-Raniry Banda Aceh



Dr. Azhar Amsal, M.Pd.
NIDN: 2001066802

LEMBAR PERNYATAAN KEASLIAN KARYA ILMIAH/SKRIPSI

Yang bertanda tangan di bawah ini:

Nama : Aulia Sabri

NIM 180705005

Program Studi : Teknologi Informasi

Fakultas : Sains dan Teknologi

Judul Skripsi : Analisis Perbandingan Model Arsitektur CNN Dalam Pendeteksian *Coronavirus Disease* Menggunakan Citra *X-ray* Paru-Paru (Studi Kasus : *Baseline CNN*, *InceptionResnetV2*, *VGG19*, dan *Xception*)

Dengan ini menyatakan bahwa dalam penulisan skripsi ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu mempertanggungjawab atas karya ini;

Bila kemudian hari ini ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat mempertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenakan sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh,

Yang Menyatakan


Aulia Sabri

ABSTRAK

Nama : Aulia Sabri
NIM : 180705005
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi (FST)
Judul : Analisis Perbandingan Model Arsitektur CNN Dalam Pendeteksian *Coronavirus Disease* Menggunakan Citra *X-ray* Paru-Paru (Studi Kasus : *Baseline CNN*, *InceptionResnetV2*, *VGG19*, dan *Xception*)
Tanggal Sidang : 20 Juli 2022/ 21 Dzulhijah 1443 H
Tebal Skripsi : 77 Halaman
Pembimbing I : Bustami, M.Sc.
Pembimbing II : Ima Dwitawati, M.B.A
Kata Kunci : CNN, *COVID-19* dan Citra *X-Ray* Paru-Paru

Convolutional neural network (CNN) merupakan metode yang terdapat pada *deep learning* yang diklaim sebagai metode terbaik atas permasalahan yang berkaitan dengan *image classification*. Perkembangan metode CNN saat ini telah menghasilkan banyak model atau arsitektur CNN terlatih yang dapat digunakan pada berbagai macam data, salah satunya data citra *x-ray* paru-paru untuk pendeteksian *COVID-19*. Penelitian ini menggunakan beberapa model arsitektur CNN yaitu *baseline CNN*, *VGG19*, *InceptionResnetV2*, dan *Xception*. Pertanyaan penelitian dalam skripsi ini adalah bagaimana mengimplementasikan model arsitektur *baseline CNN*, *VGG19*, *InceptionResnetV2*, dan *Xception* dalam pendeteksian *COVID-19* menggunakan citra *x-ray* paru-paru serta mengetahui model mana yang memiliki tingkat keakuratan terbaik.

Penelitian ini merupakan penelitian kualitatif dan penelitian pengembangan (*Research and Development / R&D*). Data citra *x-ray* paru-paru diperoleh dari *website kaggle.com*, kemudian data tersebut melalui tahapan data *preprocessing* agar siap digunakan pada proses *training* dan *testing* model. Setelah *training* dan *testing* selesai maka dilakukan evaluasi terhadap model untuk mengetahui tingkat keakuratan yang dihasilkan.

Berdasarkan pembahasan dan hasil analisis yang telah dilakukan dalam penelitian ini, maka diperoleh *accuracy* dari setiap model arsitektur CNN. *Baseline CNN* memiliki *accuracy* 83,17%, *VGG19* 93,29%, *InceptionResnetV2* 92,34% dan *Xception* 93,33%. Dari penelitian ini disimpulkan bahwa model arsitektur yang memiliki tingkat *accuracy* terbaik adalah *Xception* 93,33%, disusul *VGG19* 93,29%, *InceptionResnetV2* 92,34%, dan yang terakhir *baseline CNN* dengan nilai *accuracy* 83,17%.

ABSTRACT

Name : Aulia Sabri
Student Number : 180705005
Department : Information Technology
Faculty : Science and Technology
Title : Comparative Analysis of CNN Architectural Models in Detecting Coronavirus Disease Using Image Lung X-ray (Case Study: Baseline CNN, InceptionResnetV2, VGG19, and Xception)
Date of Thesis Exam : 20 July 2022/ 21 Dzulhijah 1443 H
Thesis Pages : 77 Halaman
Supervisor I : Bustami, M.Sc.
Supervisor II : Ima Dwitawati, M.B.A
Keywords : CNN, COVID-19 and Image Lung X-ray

Convolutional neural network (CNN) is a method found in deep learning which is claimed to be the best method for problems related to image classification. The current development of the CNN method has produced many CNN models or architectures that can be used on various kinds of data, one of which is lung x-ray image data for the detection of COVID-19. This study uses several CNN architectural models, namely baseline CNN, VGG19, InceptionResnetV2, and Xception. The research question in this thesis is how to implement the CNN, VGG19, InceptionResnetV2, and Xception baseline architecture models in the detection of COVID-19 using lung x-ray images and find out which model has the best level of accuracy.

This research is a qualitative research and research development (R&D). The lung x-ray image data was obtained from the kaggle.com website, then the data went through the data preprocessing stage so that it was ready to be used in the training and testing process of the model. After the training and testing is complete, an evaluation of the model is carried out to determine the level of accuracy.

Based on the discussion and analysis results that have been carried out in this study, the accuracy of each CNN architectural model is obtained. Baseline CNN has an accuracy of 83.17%, VGG19 93.29%, InceptionResnetV2 92.34% and Xception 93.33%. From this research, it is said that the architectural model that has the best accuracy rate is Xception 93.33%, followed by VGG19 93.29%, InceptionResnetV2 92.34%, and the last baseline CNN with an accuracy value of 83.17%.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan penulisan skripsi ini dengan baik. Shalawat dan salam tidak lupa kita sanjung sajikan kepada baginda Nabi besar Muhammad SAW yang telah membawa seluruh umatnya untuk menjadi generasi yang berilmu pengetahuan.

Dengan izin Allah SWT, saya dapat menyelesaikan skripsi yang berjudul **“Analisis Perbandingan Model Arsitektur CNN Dalam Pendeteksian Coronavirus Disease Menggunakan Citra X-ray Paru-Paru (Studi Kasus : Baseline CNN, InceptionResnetV2, VGG19, dan Xception)”** Dengan harapan penulis bahwa skripsi ini dapat memberikan manfaat bagi pihak yang membutuhkan, menambahkan wawasan serta imu pengetahuan.

Penulis menyadari skripsi ini tidak dapat diselesaikan dengan baik tanpa bimbingan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada pihak-pihak yang terlibat secara langsung maupun tidak langsung dalam mendukung kelancaran penulisan skripsi ini baik berupa dukungan, doa maupun bimbingan yang telah diberikan. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada:

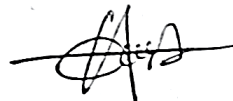
1. Dr. Azhar Amsal, M.Pd. selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.
2. Eriawati, S.Pd., M.Pd. dan Hendri Ahmadian, S.Si., M.IM selaku Ketua dan Sekretaris Program Studi Teknologi Informasi Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.
3. Bustami, M.Sc. selaku pembimbing I dan Ima Dwitawati, M.B.A selaku pembimbing II yang telah mencurahkan waktu, pikiran dan tenaga dalam

membimbing penulis demi kesempurnaan skripsi ini. Terima kasih banyak penulis ucapkan, semoga Bapak dan Ibu selalu mendapat rahmat dan lindungan Allah SWT.

4. Nazaruddin Ahmad, M.T selaku Penasehat Akademik (PA) penulis selama menempuh pendidikan di Program Studi Teknologi Informasi. Terima kasih banyak telah memberi nasehat dan saran selama ini kepada penulis.
5. Seluruh dosen yang mengajar pada Program Studi Teknologi Informasi yang telah memberikan ilmu pengetahuan yang sangat berguna bagi penulis selama proses belajar mengajar.
6. Cut Ida Rahmadiana S,Si. Selaku staf prodi Teknologi Informasi yang telah membantu penulis dalam hal administrasi selama menempuh pendidikan di Program Studi Teknologi Informasi.
7. Orang tua terhebat yang penulis cintai, Bapak Danial, S.P dan Ibu Khairani yang telah mendo'akan serta memberikan semangat, kasih sayang yang tiada henti kepada penulis serta seluruh keluarga besar yang telah memberikan nasihat, semangat dan motivasi sehingga penulis menyelesaikan skripsi ini.
8. Sahabat-sahabat Teknologi Informasi yang telah berjuang bersama, berbagi semangat dan suka duka dalam penyelesaian skripsi ini. Terima kasih banyak penulis ucapkan kepada Darul Fata, Faslul Faizi, Mega Ellyadi, Muhammad Firdaus, Muhammad Ridha, Nisa Afdhila, Rizal Ilham, dan Rizki Mardhatillah. Penulis sangat berterima kasih atas motivasi dan semangat kalian semua.

Banda Aceh, 18 Juli 2022

Penulis



Aulia Sabri

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN KARYA ILMIAH/SKRIPSI	iii
ABSTRAK	iv
ABSTRACT.....	v
KATA PENGANTAR	vi
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	2
I.3 Tujuan Penelitian	3
I.4 Batasan Masalah.....	3
I.5 Manfaat Penelitian	3
I.6 Sistematika Pembahasan	4
BAB II TINJAUAN PUSTAKA.....	6
II.1 COVID-19	6
II.2 CT-Scan dan X-ray.....	6
II.3 Computer Vision	7
II.4 Artificial Inteligence (AI).....	7
II.5 Machine Learning (ML)	8
II.6 Deep Learning.....	8
II.7 Artificial Neural Network (ANN)	9
II.7.1 Neuron.....	9
II.7.2 Perceptron.....	10
II.7.3 Multi Layer Perceptron (MLP).....	11
II.7.4 Backpropagation	11
II.8 Convolutional Neural Network (CNN)	12
II.8.1 Image classification	12
II.8.2 Convolutional layer.....	13
II.8.3 Filter.....	13
II.8.4 Proses konvolusi	14
II.8.5 Max Pooling	14

II.8.6	Arsitektur CNN	15
II.9	<i>Transfer Learning</i>	16
II.10	<i>Python</i>	16
II.11	<i>Keras</i>	16
II.12	<i>Evaluation Measurement</i> (Pengukuran Evaluasi)	18
II.12.1	<i>Confusion Matrix</i>	18
II.12.2	<i>Recall</i> dan <i>Precision</i>	19
II.12.3	<i>F1 Score</i>	19
II.12.4	<i>Accuracy</i>	20
II.13	Penelitian Terdahulu.....	20
II.14	Kerangka Berpikir	22
BAB III	METODE PENELITIAN.....	23
III.1	Jenis Penelitian	23
III.2	Tahapan Penelitian	23
III.3	Jenis Data	31
III.4	Teknik Pengumpulan Data	31
III.5	Alat dan Bahan	32
III.6	Tempat dan Waktu Penelitian	32
BAB IV	HASIL PENELITIAN DAN PEMBAHASAN	33
IV.1	Persiapan <i>Software</i>	33
IV.2	Pengambilan Dataset	34
IV.3	Data <i>preprocessing</i>	36
IV.4	Model.....	37
IV.5	<i>Training & Testing</i>	37
IV.6	Hasil.....	44
IV.7	Pengaruh Jumlah Dataset dan jumlah <i>Class</i>	49
BAB V	KESIMPULAN DAN SARAN.....	51
V.1	Kesimpulan.....	51
V.2	Saran	51
DAFTAR	KEPUSTAKAAN	53
LAMPIRAN-	LAMPIRAN.....	59
RIWAYAT	PENULIS	65

DAFTAR GAMBAR

Gambar II.1 Citra <i>x-ray</i> paru paru.....	7
Gambar II.2 <i>Neuron</i> atau saraf	10
Gambar II.3 <i>Perceptron</i>	10
Gambar II.4 <i>Multi layer perceptron</i>	11
Gambar II.5 <i>Backpropagation</i>	11
Gambar II.6 Contoh Pendeteksian pada gambar.....	13
Gambar II.7 <i>Filter</i>	14
Gambar II.8 Konvolusi	14
Gambar II.9 <i>Pooling</i>	15
Gambar II.10 Arsitektur CNN	15
Gambar II.11 Struktur <i>framework Keras</i>	17
Gambar II.12 Kerangka berpikir	22
Gambar III.1 Tahapan penelitian	24
Gambar III.2 (a) <i>COVID-19</i> (b) Normal (c) <i>Lung opacity</i> (d) <i>Pneumonia</i>	25
Gambar III.3 Arsitektur CNN	26
Gambar III.4 Arsitektur <i>InceptionResnetV2</i>	28
Gambar III.5 <i>InceptionResnetV2</i> layer.....	28
Gambar III.6 Arsitektur <i>Xception</i>	29
Gambar III.7 <i>Xception</i> layer	29
Gambar III.8 Arsitektur <i>VGG19</i>	30
Gambar III.9 <i>VGG19</i> layer	30
Gambar IV.1 <i>Login google colab</i>	33
Gambar IV.2 <i>Notebook</i> baru	33
Gambar IV.3 Ubah jenis <i>runtime</i>	34
Gambar IV.4 Kode untuk menghubungkan <i>google colab</i> dan <i>kaggle</i>	35
Gambar IV.5 Kode download dataset	35
Gambar IV.6 Kode <i>restucturing folder</i>	35
Gambar IV.7 Proses <i>training</i>	38
Gambar IV.8 <i>Validation accuracy</i> setiap model.....	43
Gambar IV.9 <i>Confusion matrix</i> CNN	44
Gambar IV.10 <i>Confusion matrix</i> <i>InceptionResnetV2</i>	45
Gambar IV.11 <i>Confusion matrix</i> <i>Xception</i>	46
Gambar IV.12 <i>Confusion matrix</i> <i>VGG19</i>	47
Gambar IV.13 <i>Recall, precision, f1 score, dan accuracy</i> setiap model	48

DAFTAR TABEL

Tabel II.1 <i>Confusion matrix</i>	18
Tabel II.2 Penelitian terdahulu	20
Tabel IV.1 Hasil <i>training baseline CNN</i>	39
Tabel IV.2 Hasil <i>training InceptionResnetV2</i>	40
Tabel IV.3 Hasil <i>training Xception</i>	41
Tabel IV.4 Hasil <i>training VGG19</i>	42
Tabel IV.5 Hasil <i>testing</i>	43
Tabel IV.6 Pengaruh jumlah dataset dengan 4 <i>class</i>	50
Tabel IV.7 Pengaruh jumlah dataset dengan jumlah 3 <i>class</i>	50



DAFTAR LAMPIRAN

Lampiran 1 Code.....	59
----------------------	----



BAB I

PENDAHULUAN

I.1 Latar Belakang

Coronavirus disease 2019 (COVID-19) adalah suatu penyakit infeksi sistem pernapasan yang disebabkan oleh virus yang bernama *Severe acute respiratory syndrome coronavirus-2 (SARS-CoV-2)* (Yuliana, 2020). Virus ini ditemukan pada Desember tahun 2019 dan mulai menyebar keseluruh dunia pada awal tahun 2020. Menurut data *epidemiologi*, virus ini berasal dari pasar *seafood* yang berada di Wuhan, Tiongkok (Huang dkk., 2020). Data dari WHO menyatakan sejak awal ditemukan hingga sekarang per tanggal 16 november 2021 sudah ada sebanyak 253,163,330 pasien yang terkonfirmasi *COVID-19* di seluruh dunia dan 5,098,174 pasien yang meninggal (Who.int, 2021). Pemerintah Indonesia sendiri menyebutkan ada sebanyak 4,251,076 kasus terkonfirmasi dan 143,670 kasus meninggal yang ada di Indonesia (covid19.go.id, 2021). Rata-rata penderita infeksi *COVID-19* mengalami gejala klinis ringan hingga berat. Beberapa gejala yang ditimbulkan dapat berupa demam tinggi, pilek, batuk, nyeri tenggorokan, kesulitan bernapas, dan infeksi paru-paru dan berdasarkan hasil *rontgen* pada citra *x-ray* paru-paru menunjukkan *infiltrat pneumonia* luas di kedua paru (Fauzi dkk., 2020; PDPI, 2020). Karena gejala yang ditimbulkan banyak maka ada banyak pula cara untuk mendeteksi *COVID-19* salah satunya dengan memanfaatkan citra *x-ray* paru.

Penelitian menggunakan citra *x-ray* paru pernah dilakukan sebelumnya untuk mendeteksi *COVID-19* menggunakan metode CNN (Apostolopoulos & Mpesiana, 2020; Chouat dkk., 2022; Fauzi dkk., 2020; Malik dkk., 2021). CNN merupakan metode yang terdapat pada deep learning yang diklaim sebagai metode

terbaik atas permasalahan yang berkaitan dengan *image classification* (Nasichuddin, 2017). Perkembangan CNN saat ini telah menghasilkan banyak model atau arsitektur CNN terlatih yang dapat digunakan pada berbagai macam data. Penelitian menggunakan model arsitektur CNN tersebut pernah dilakukan oleh I Apostolopoulos dkk untuk mengetahui model mana yang dapat melakukan pendeteksian *COVID-19* dengan akurasi terbaik (Apostolopoulos & Mpesiana, 2020). Mereka menggunakan model arsitektur CNN berbeda seperti *InceptionResnetV2*, *VGG19*, dan *Xception* dengan 1.428 citra *x-ray* paru.

Oleh karena itu, pada penelitian ini penulis melakukan analisis perbandingan model arsitektur CNN dalam pendeteksian *COVID-19*. Model yang digunakan pada penelitian ini adalah *baseline* CNN, *VGG19*, *InceptionResnetV2*, dan *Xception* dengan menggunakan 21.165 citra *x-ray* paru-paru. Penelitian ini bertujuan untuk mengetahui model mana yang dapat melakukan pendeteksian *COVID-19* dengan akurasi terbaik. Hasil dari penelitian ini diharapkan dapat berkontribusi pada permasalahan yang sedang dihadapi saat ini, khususnya pada bidang *deep learning* dan bidang kesehatan.

I.2 Rumusan Masalah

Berdasarkan uraian pada bagian sebelumnya, maka rumusan masalah pada penelitian ini adalah:

1. Bagaimana mengimplementasikan model arsitektur *baseline* CNN, *VGG19*, *InceptionResnetV2*, dan *Xception* dalam pendeteksian *COVID-19* menggunakan citra *x-ray* paru-paru?

2. Bagaimana hasil deteksi *COVID-19* menggunakan model arsitektur *Baseline* CNN, *VGG19*, *InceptionResnetV2*, dan *Xception* pada citra *x-ray* paru-paru untuk mengetahui model dengan tingkat keakuratan terbaik?

I.3 Tujuan Penelitian

Tujuan dalam penelitian ini adalah:

1. Mengetahui bagaimana mengimplementasikan model arsitektur *Baseline* CNN, *VGG19*, *InceptionResnetV2*, dan *Xception* dalam pendeteksian *COVID-19* menggunakan citra *x-ray* paru-paru.
2. Mengetahui hasil deteksi *COVID-19* menggunakan model arsitektur *Baseline* CNN, *VGG19*, *InceptionResnetV2*, dan *Xception* pada citra *x-ray* paru-paru untuk mengetahui model dengan tingkat keakuratan terbaik.

I.4 Batasan Masalah

Penelitian ini terbatas pada:

1. Data yang digunakan dalam penelitian ini adalah data citra *x-ray* paru-paru.
2. Metode yang digunakan yaitu metode CNN.
3. Bahasa pemrograman yang digunakan adalah *python* dengan menggunakan *framework keras*.

I.5 Manfaat Penelitian

Penulis berharap dengan adanya penelitian ini akan bermanfaat bagi sebagian pihak, antara lain:

1. Secara Teoritis

Hasil penelitian ini secara teoritis diharapkan dapat memberikan sumbangan pemikiran tentang model arsitektur CNN dalam pendeteksian *COVID-19* menggunakan citra *x-ray* paru-paru.

2. Secara Praktis

Hasil penelitian ini secara praktis diharapkan dapat memudahkan suatu pihak untuk mengembangkan sistem pendeteksian *COVID-19* dengan menggunakan bantuan perangkat lain yang mana sistem dapat bekerja terhadap pendeteksian pada *COVID-19* dengan menerapkan model arsitektur CNN yang telah teruji melalui penelitian ini.

3. Secara Akademis

Hasil penelitian ini dapat dijadikan sebagai acuan untuk penelitian sejenis terhadap pendeteksian pada objek-objek lain dengan menggunakan model arsitektur CNN dan untuk pengayaan pengetahuan bidang *deep learning*.

I.6 Sistematika Pembahasan

BAB I PENDAHULUAN Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika pembahasan.

BAB II LANDASAN TEORI Bab ini menjelaskan uraian teori yang relevan dengan topik yang akan dibahas diantaranya yaitu *COVID-19*, *CT-Scan* dan *X-ray*, *computer vision*, *artificial intelligence*, *machine learning* (ML), *deep learning*, *artificial neural network* (ANN), *convolutional neural network* (CNN), *transfer learning*, *python*, *keras*, *evaluation measurement* (pengukuran evaluasi), penelitian terdahulu, dan kerangka berpikir.

BAB III METODE PENELITIAN Bab ini menjelaskan metode penelitian yang relevan dengan judul skripsi, penelitian ini menggunakan penelitian kualitatif dan penelitian pengembangan (*Research and Development/ R&D*) yang meliputi jenis

penelitian, tahapan penelitian, jenis data, teknik pengumpulan data, bahan dan alat, dan tempat dan waktu.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN Bab ini menjelaskan tentang hasil pengolahan data dan pembahasan yang akan menjelaskan temuan dan pembahasan data hasil penelitian, sesuai dengan metode yang digunakan.

BAB V PENUTUP Bab ini menjelaskan kesimpulan dan saran berdasarkan pembahasan yang telah diuraikan dalam penelitian.



BAB II

TINJAUAN PUSTAKA

II.1 COVID-19

Coronavirus disease 2019 (COVID-19) adalah suatu penyakit infeksi sistem pernapasan yang disebabkan oleh virus yang bernama *Severe acute respiratory syndrome coronavirus-2 (SARS-CoV-2)* (Yuliana, 2020). Virus ini ditemukan pada Desember tahun 2019 dan mulai menyebar keseluruh dunia pada awal tahun 2020. Menurut data *epidemiologi*, virus ini berasal dari pasar *seafood* yang berada di Wuhan, Tiongkok (Huang dkk., 2020). Sejak awal ditemukan hingga sekarang per tanggal 16 november 2021 sudah ada sebanyak 253,163,330 pasien yang terkonfirmasi *COVID-19* di seluruh dunia dan 5,098,174 pasien yang meninggal (Who.int, 2021). Di Indonesia sendiri ada sebanyak 4,251,076 kasus terkonfirmasi, 4,098,884 sembuh, dan 143,670 kasus meninggal (covid19.go.id, 2021). Rata-rata penderita infeksi *COVID-19* mengalami gejala klinis ringan hingga berat. Beberapa gejala yang ditimbulkan dapat berupa demam tinggi, pilek, batuk, nyeri tenggorokan, kesulitan bernapas, dan infeksi paru-paru dan berdasarkan hasil *rontgen* pada citra *x-ray* paru-paru menunjukkan infiltrat *pneumonia* luas di kedua paru (Fauzi dkk., 2020; PDPI, 2020).

II.2 CT-Scan dan X-ray

CT-Scan dan *x-ray* merupakan suatu metode yang digunakan pada paru-paru untuk melihat dan mendiagnosis *pneumonia*. *CT-Scan* melakukan pemantauan dan mendeteksi *pneumonia* pada paru-paru dengan menggunakan *x-ray* yang berbasis *artificial intelligence* (Gozes dkk., 2020). Citra *x-ray* paru-paru yang penulis gunakan pada penelitian ini adalah output yang dihasilkan oleh metode *CT-*

Scan dan *x-ray*. Dibidang kedokteran, hasil *CT-Scan* dan *x-ray* mempunyai peran penting untuk mengidentifikasi pasien terkena virus infeksi paru-paru atau tidak. Contoh dari citra *x-ray* dapat dilihat pada gambar 1.



Gambar II.1 Citra *x-ray* paru paru

II.3 *Computer Vision*

Computer vision merupakan ilmu yang membuat komputer dapat mengenali suatu keadaan dan objek fisik nyata dari sebuah gambar atau citra untuk dapat membuat suatu keputusan (Shapiro & Stockman, 2000). *Computer vision* merupakan salah satu bidang pada *artificial intelligence*. *Computer vision* membuat komputer dapat menerima informasi secara visual seperti manusia. Sehingga, menjadikan komputer “*acts like human sight*”. Beberapa Kemampuan itu adalah: *object detection*, *image classification* dan *image segmentation*.

II.4 *Artificial Intelligence (AI)*

Suatu ilmu dan teknik yang membahas tentang pembuatan mesin cerdas merupakan definisi *Artificial intelligence* atau kecerdasan buatan. Menurut Dobrev (Dobrev, 2004) *artificial intelligence* adalah bagaimana agar sebuah komputer mampu melakukan pekerjaan yang masih dilakukan dengan lebih baik oleh manusia pada saat ini. Layaknya manusia, sebuah komputer menjadi cerdas dengan belajar dari pengalaman. komputer mendapatkan pengalaman dengan mempelajari data.

II.5 *Machine Learning* (ML)

Machine learning (ML) diperkenalkan oleh seorang ilmuwan komputer yang bernama Arthur Samuel pada tahun 1959. Menurutnya, *machine learning* merupakan disiplin ilmu yang membuat komputer memiliki kemampuan belajar tanpa harus diprogram secara eksplisit (Samuel, 1959). *Machine learning* merupakan salah satu bidang pada *artificial intelligence* yang membuat suatu prediksi atau keputusan berdasarkan penggunaan komputer menggunakan model statistik. *support vector machine* dan *decision tree* merupakan contoh dari model *machine learning*. *Machine learning* pada umumnya dibagi menjadi tiga kategori besar yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.

Penelitian yang penulis lakukan termasuk dalam kategori *Supervised learning*. Kategori ini menyediakan solusi atau label pada proses pembelajarannya. Algoritma bekerja dengan mempelajari pola pada dataset yang mempunyai pasangan data dan label. *Supervised learning* dapat dianggap sebagai sebuah komputer yang belajar menyelesaikan masalah dengan solusi yang sudah diberikan oleh manusia.

II.6 *Deep Learning*

Deep learning merupakan bagian dari *machine learning* yang mampu beradaptasi dan belajar terhadap sejumlah besar data. Berbeda dengan *machine learning* yang belajar menggunakan model statistik, *deep learning* belajar menggunakan algoritma jaringan syaraf tiruan dalam menyelesaikan masalah. Algoritma jaringan syaraf tiruan pada *deep learning* terinspirasi dari struktur otak manusia. *Deep learning* mempelajari data dengan memanfaatkan banyak *layer*

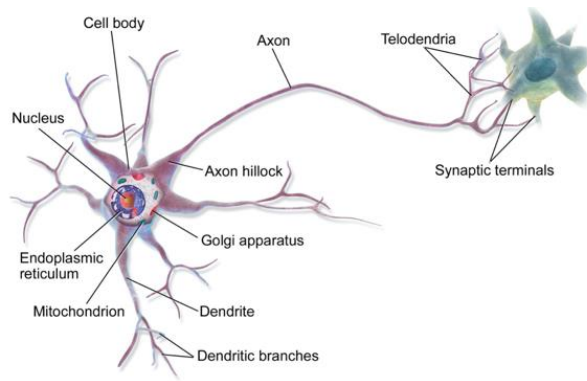
pengolahan informasi (Deng & Yu, 2013). Sun dkk, mengatakan bahwa kelebihan *deep learning* adalah mampu secara otomatis mempelajari fitur-fitur abstrak yang berguna pada data yang kompleks (Sun dkk., 2017). Oleh karena itu, *deep learning* sangat sesuai dijadikan sebagai solusi atas permasalahan yang berhubungan dengan *image classification* pada citra *x-ray* paru-paru.

II.7 Artificial Neural Network (ANN)

Artificial neural network (ANN) atau jaringan saraf tiruan merupakan sebuah algoritma dalam *deep learning* yang terinspirasi dari sistem jaringan neuron/saraf pada otak manusia. Haykin mengatakan (Haykin, 2009), *neuron* adalah dasar dari sebuah jaringan saraf tiruan yang berfungsi sebagai unit pengolah informasi. Saat ini ANN telah digunakan dalam menangani permasalahan seperti klasifikasi gambar dan pengenalan bahasa.

II.7.1 Neuron

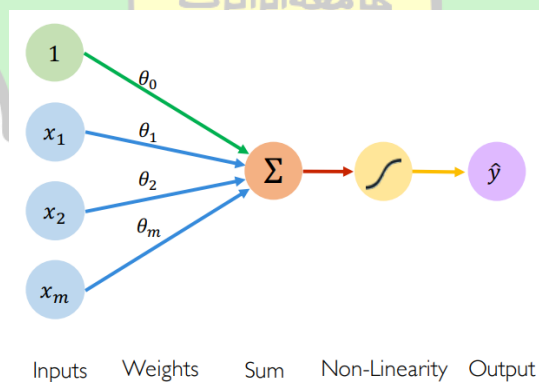
National Institute of Neurological Disorders and Stroke (ninds.nih.gov, 2021) menyebutkan bahwa *neuron* atau saraf adalah pembawa informasi. *Neuron* mengirimkan informasi keseluruh sistem saraf menggunakan *impuls* listrik dan sinyal kimiawi. *Neuron* terdiri atas 3 bagian, yaitu badan sel yang di dalamnya terdapat *nukleus*, *dendrit*, dan *akson*. *Nukleus* berfungsi mengontrol seluruh aktivitas sel dan berisi materi genetik. *Dendrit* adalah cabang pendek yang bertugas menerima pesan untuk sel. Sedangkan *akson* adalah cabang panjang yang berfungsi mengirimkan pesan dari sel.



Gambar II.2 Neuron atau saraf (Kia, 2017)

II.7.2 Perceptron

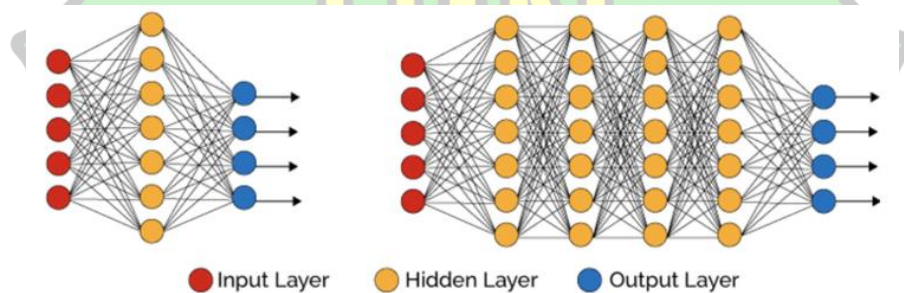
Komponen dasar pembangun jaringan saraf tiruan adalah *perceptron*. Ilmuwan yang pertama kali menemukan *perceptron* adalah Frank Rosenblatt dari *Cornell Aeronautical Library* pada tahun 1957 (Rosenblatt, 1958). *Perceptron* terinspirasi dari neuron pada jaringan saraf manusia. *Perceptron* dan *neuron* merupakan hal yang sama pada jaringan saraf tiruan. Suatu perceptron terdiri dari 5 bagian yaitu Input (x_i) dan Output (y), Penjumlahan (Σ), Bobot atau weights (W_i) dan bias (W_0), dan Fungsi aktivasi (f).



Gambar II.3 Perceptron (Data, 2022)

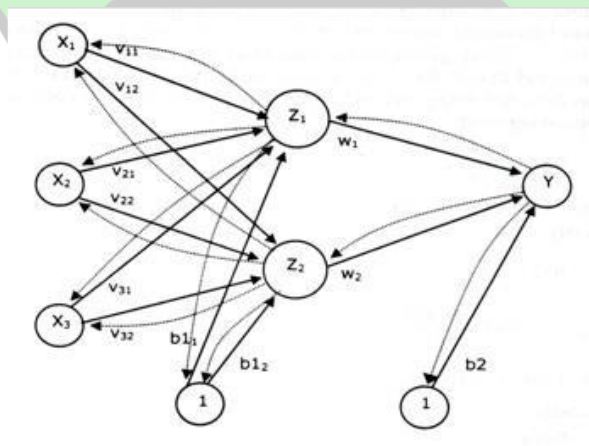
II.7.3 Multi Layer Perceptron (MLP)

Multi layer perceptron (MLP) merupakan jaringan saraf yang terdiri dari *layer input*, *hidden layer*, *layer output*. *Layer input* merupakan layer yang hanya menampung angka-angka. Sedangkan *hidden layer* dan *layer output* adalah layer yang mempunyai beberapa *perceptron*. Pada jaringan saraf tiruan, hidden layer bisa ada atau tidak sama sekali. Namun untuk *layer input* dan *layer output* harus selalu ada. Pada sebuah jaringan saraf tiruan tidak ada aturan baku dalam menetapkan jumlah *perceptron* dan *hidden layer*. Semakin banyak jumlah *hidden layer* yang dipakai, maka semakin kompleks masalah yang dapat diselesaikan, namun juga semakin lama jaringan saraf tersebut untuk belajar.



Gambar II.4 Multi layer perceptron (Johnson & Khoshgoftaar, 2019)

II.7.4 Backpropagation



Gambar II.5 Backpropagation (LINA, 2018)

Propagasi balik atau *backpropagation* dipublikasikan melalui sebuah penelitian yang dilakukan oleh Geoffrey Hinton, David Rumelhart, dan Ronald Williams pada tahun 1985 (Rumelhart & McClelland, 1985). Propagasi balik merupakan suatu metode yang digunakan dalam pelatihan MLP. Metode *backpropagation* bekerja dengan melakukan *forward propagation* pada awal proses dan kemudian menyimpan *error* yang dihasilkan. Lalu *error* tersebut digunakan untuk mengubah nilai bobot setiap perceptron dalam arah mundur (*backward*).

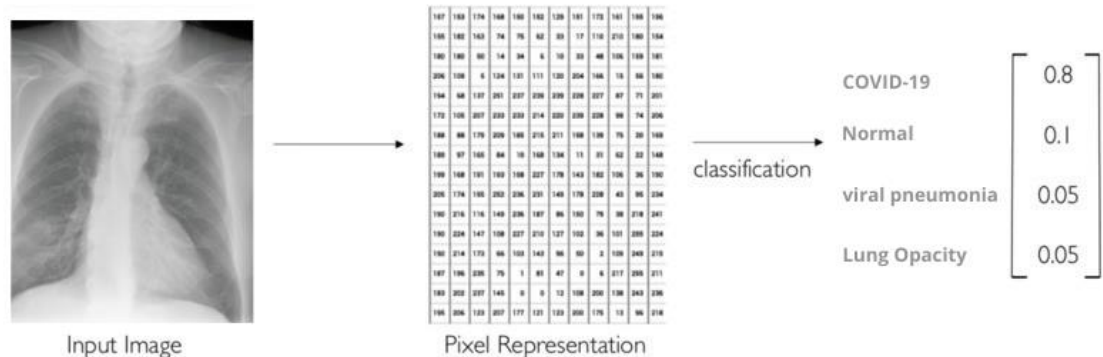
II.8 Convolutional Neural Network (CNN)

Convolutional neural network (CNN) diperkenalkan pada tahun 1998 oleh Yann LeCun pada makalahnya “*Gradient-Based Learning Applied to Document Recognition*” (LeCun dkk., 1998). CNN yang dikenalkan adalah *LeNet* yang berhasil mengenali karakter tulisan tangan. Hingga pada tahun 2012, Alex Krizhevsky memenangkan perlombaan terkenal *ImageNet* dengan memperkenalkan versi lebih canggih dari CNN yaitu *AlexNet* (Krizhevsky dkk., 2012). *AlexNet* ini adalah cikal bakal deep learning. Penamaan *convolutional neural network* diambil karena pada jaringan tersebut terdapat suatu operasi matematika yang disebut konvolusi. CNN merupakan jaringan saraf yang menggunakan konvolusi pada lapisannya dengan jumlah minimal satu lapisan konvolusi (LeCun dkk., 2015). Saat ini CNN diklaim sebagai model terbaik sebagai solusi atas permasalahan yang berkaitan dengan *image classification* (Nasichuddin, 2017).

II.8.1 Image classification

Komputer merupakan mesin yang hanya bisa memahami angka. Komputer melihat sebuah gambar sebagai sebuah matrik yang memiliki nilai pada setiap pixelnya. Matriks 2 dimensi merupakan gambar hitam putih. Sedangkan matriks 2

kali 3 dimensi merupakan gambar berwarna dimana dimensi terakhirnya berisi jumlah kanal dari RGB (red, green, blue).



Gambar II.6 Contoh Pendeteksian pada gambar

Salah satu peran dari machine learning adalah klasifikasi gambar. Contohnya, pada pendeteksian *COVID-19* berdasarkan citra *x-ray* paru dengan output beberapa label penyakit infeksi pada paru. Pada gambar 6 memperlihatkan probabilitas penyakit infeksi paru di dalam gambar. Model harus belajar untuk mengenali ciri-ciri unik yang terdapat pada setiap *class* agar mampu mengklasifikasikan penyakit pada citra *x-ray* dengan benar.

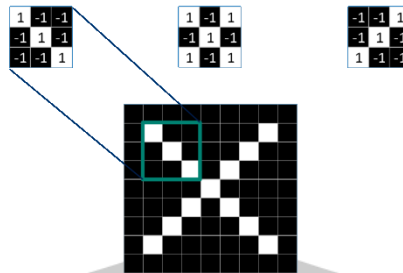
II.8.2 Convolutional layer

Convolutional layer adalah *layer* yang berfungsi untuk mengenali objek pada gambar berdasarkan atribut-atributnya. Atribut-atribut yang lebih tinggi dibentuk dari atribut-atribut yang lebih rendah contohnya atribut hidung dibentuk dari lengkungan dan garis. Atribut mata, hidung, mulut, dan telinga membentuk atribut wajah.

II.8.3 Filter

Filter merupakan matriks yang berisi angka-angka yang digunakan untuk mengenali atribut dari objek pada *convolutional layer*. Seperti yang di perhatikan pada gambar 7. Pada gambar tersebut, terdapat gambar dengan objek huruf X dan

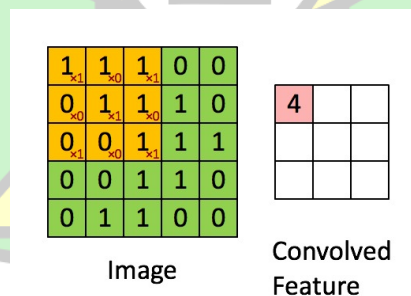
3 buah *filter* matriks 3x3 piksel. Setiap *filter* mengenali atribut yang berbeda-beda. *Filter* sebelah kiri dapat mengenali garis pada kotak biru dan *filter* sebelah kanan dapat mengenali garis pada kotak merah.



Gambar II.7 Filter (<https://www.facebook.com/kdnuggets>, 2016)

II.8.4 Proses konvolusi

Proses penerapan *filter* pada gambar disebut dengan proses konvolusi. Penerapan *filter* terjadi dengan perkalian matriks antara area pada gambar dan filter. Gambar 8 menunjukkan bagaimana proses konvolusi pada sebuah gambar terjadi. Hasil dari proses konvolusi kemudian dijadikan masukan untuk sebuah *multi layer perceptron*(MLP).



Gambar II.8 Konvolusi (“Penjelasan State of the Art Convolutional Neural Networks (CNNs) - DenseNets,” 2020)

II.8.5 Max Pooling

Pooling merupakan proses mengurangi resolusi sekaligus ukuran pada gambar dengan tetap mempertahankan informasi mengenai objek didalamnya. Pada umumnya proses pooling terjadi setelah proses konvolusi. *Max pooling* adalah

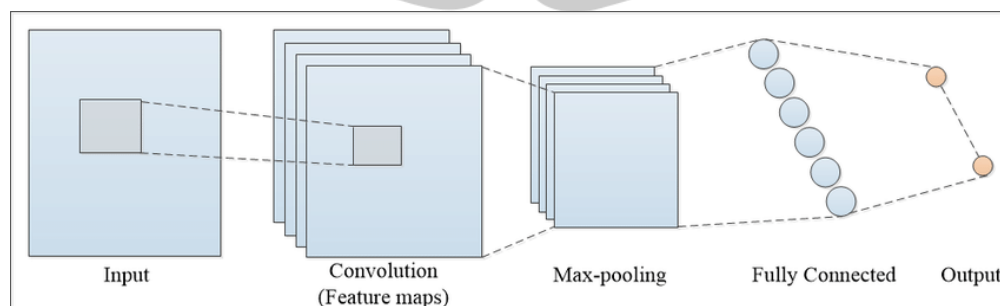
salah satu contoh dari pooling. *Max pooling* bekerja dengan mengambil nilai piksel tertinggi dari luas piksel tertentu. Gambar di bawah menunjukkan contoh dari *max pooling* pada gambar berukuran 4x4 piksel dengan ukuran 2x2 piksel dengan hasil berupa gambar dengan ukuran 2x2 piksel.

Input				maxpool	Output	
7	3	5	2		8	6
8	7	1	6		9	9
4	9	3	9			
0	8	4	5			

Gambar II.9 Pooling (“Kotakode.com | Komunitas Developer Indonesia,” 2022)

II.8.6 Arsitektur CNN

Arsitektur CNN merupakan sebuah jaringan saraf yang menggunakan *layer konvolusi* dan *max pooling*. Pada gambar 10, sebuah gambar melalui proses konvolusi untuk dikenali atribut/fiturnya. Lalu dilanjutkan dengan *max pooling* dengan hasil berupa gambar yang resolusinya lebih kecil. kemudian, hasil dari *max pooling* dimasukkan ke dalam *hidden layer* MLP. Umumnya arsitektur CNN terdiri dari beberapa lapisan konvolusi dan *max pooling*. Penambahan Lapisan konvolusi dan *max pooling* dapat dilakukan dengan cara ditambahkan setelah *max pooling* sebelumnya. Semakin banyak lapisan proses konvolusi maka semakin detail atribut/fitur gambar yang dapat dikenali.



Gambar II.10 Arsitektur CNN (Hussain dkk., 2018)

II.9 *Transfer Learning*

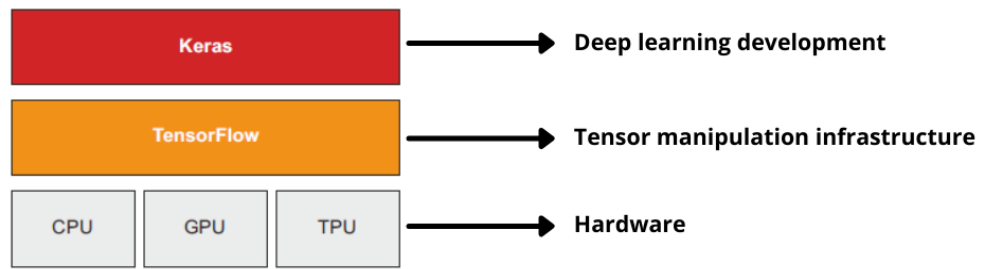
Transfer learning merupakan teknik yang memanfaatkan model yang sudah dilatih sebelumnya (*pretrained model*) dalam hal ini adalah model CNN untuk digunakan mengklasifikasikan dataset yang baru (Rochman & Junaedi, 2020). *Pretrained model* yang digunakan pada penelitian ini adalah *VGG19*, *Xception*, dan *InceptionResnetV2*. *Pretrained model* digunakan untuk melakukan *feature learning* pada dataset sebelum dimasukkan ke *layer* klasifikasi.

II.10 *Python*

Sebuah kecerdasan buatan tentunya mempunyai algoritma yang kompleks dibaliknya. *Python* adalah bahasa pemrograman yang mempunyai sintaks yang mudah dibaca dan dipahami. Sehingga, daripada berfokus pada teknis bahasa pemrograman kita dapat berfokus pada pemecahan masalah. *Python* juga memiliki beragam *library* yang mendukung dalam pembuatan sebuah aplikasi termasuk kecerdasan buatan (Perkovic, 2012).

II.11 *Keras*

Keras merupakan *framework deep learning* tingkat tinggi yang dikembangkan oleh *Google* untuk mengimplementasikan jaringan saraf. *Keras* dibangun di atas *TensorFlow* dan menggunakan *Python* sebagai *interface API* nya (Perkovic, 2012). *Framework* ini memiliki fitur-fitur dengan fokus untuk mempermudah pengembangan model *deep learning* (Santoso & Ariyanto, 2018). Seperti yang terlihat pada gambar 11 *Keras* berdiri di atas *TensorFlow* sehingga dapat berjalan di atas berbagai jenis perangkat keras GPU, TPU, atau CPU biasa.



Gambar II.11 Struktur framework Keras (Chollet, 2021)

Pada awal rilis pada Maret 2015 Keras dibangun di atas Theano yang merupakan sebuah *framework tensor-manipulation* yang dikembangkan oleh *Montréal Institute for Learning Algorithms (MILA)* di *Université de Montréal*. Namun, setelah *TensorFlow* rilis pada November 2015 Keras di-*refactored* menjadi arsitektur *multibackend* yang memungkinkan Keras dapat digunakan dengan *Theano* atau *TensorFlow*. Pada tahun 2017, dua opsi backend tambahan baru ditambahkan ke Keras: *CNTK* (dikembangkan oleh Microsoft) dan *MXNet* (dikembangkan oleh Amazon). Namun *Theano* dan *CNTK* berhenti dalam pengembangan dan *MXNet* jarang digunakan di luar perusahaan Amazon sehingga menjadikan *TensorFlow* satu-satunya opsi *backend* yang tersedia. Hal ini membuat Keras menjadi terkenal sebagai cara yang mudah digunakan untuk mengembangkan aplikasi *TensorFlow*. Akhirnya pada tahun 2018, pimpinan *TensorFlow* memilih Keras sebagai API tingkat tinggi resmi *TensorFlow*. Pada penelitian ini Keras digunakan sebagai *framework* yang membantu dalam pengembangan model deteksi *coronavirus disease* pada citra x-ray paru-paru.

Seiring perkembangan teknologi *deep learning*, Keras menyediakan *pretrained model* atau model yang sudah terlatih yang dapat digunakan untuk menyelesaikan berbagai macam masalah seperti *image classification* dan permasalahan lainnya. Pada Keras, model yang sudah terlatih ini disebut dengan

keras application. Beberapa *keras application* yang digunakan pada penelitian ini adalah *InceptionResnetV2*, *Xception*, dan *VGG19*. Ketiga model yang disebutkan merupakan model yang bertujuan untuk melakukan *image classification*.

II.12 Evaluation Measurement (Pengukuran Evaluasi)

Guna mengetahui hasil atau kesimpulan maka dilakukan evaluasi pada tahap akhir penelitian ini. Evaluasi pada penelitian ini dilakukan dengan menggunakan *confusion matrix*, *recall*, *precision*, *f1 score* dan *accuracy*.

II.12.1 Confusion Matrix

Confusion matrix berisi informasi tentang prediksi dan kondisi aktual dari data yang dihasilkan oleh sebuah model. Informasi yang ada pada *confusion matrix* kemudian digunakan untuk menentukan nilai dari *precision* dan *recall* (Yunus, 2020).

Tabel II.1 Confusion matrix

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Keterangan:

True Positive (TP) : jumlah data positif yang diprediksi benar

False Positive (FP): jumlah data negatif yang diprediksi sebagai data positif

False Negative (FN): jumlah data positif yang diprediksi sebagai data negatif

True Negative (TN) : jumlah data negatif yang diprediksi benar

II.12.2 Recall dan Precision

Recall merupakan pembagian dari jumlah total data positif yang diprediksi bernilai benar dibagi dengan jumlah total data positif. *High Recall* menunjukkan kelas dikenali dengan baik (FN rendah).

Secara matematis *recall* ditulis sebagai berikut:

$$recall = \frac{TP}{TP + FN}$$

Persamaan recall (1)

Sedangkan *Precision* dapat didefinisikan sebagai rasio dari jumlah total data positif yang diprediksi bernilai benar dengan jumlah total data positif yang diprediksi. *High Precision* menunjukkan data berlabel positif memang positif (FP rendah).

Secara matematis *precision* ditulis sebagai berikut:

$$precision = \frac{TP}{TP + FP}$$

Persamaan precision (2)

II.12.3 F1 Score

F1 score adalah *Harmonic Mean* antara *precision* dan *recall*. Nilai *F1 score* berada antara 0 dan 1 dimana 1 adalah nilai terbaik. *F1 Score* yang baik mengindikasikan model klasifikasi memiliki *precision* dan *recall* yang baik.

Secara matematis *f1 score* dapat ditulis sebagai berikut:

$$f1\ score = 2 \cdot \frac{1}{\frac{1}{recall} + \frac{1}{precision}}$$

Persamaan f1 score (3)

II.12.4 Accuracy

Accuracy dapat menunjukkan seberapa akurat model dapat mengklasifikasikan dengan benar berdasarkan pembagian jumlah prediksi yang benar (positif dan negatif) dengan keseluruhan data.

Secara matematis *accuracy* ditulis sebagai berikut:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Persamaan accuracy (4)

II.13 Penelitian Terdahulu

Terkait penelitian yang dilakukan untuk pemodelan deteksi *COVID-19* menggunakan model arsitektur CNN, referensi dari penelitian sebelumnya penting dilakukan agar penelitian dengan tema yang sama akan semakin berkembang dengan kontribusi baru yang dikerjakan pada penelitian ini. Berikut ulasan mengenai penelitian yang pernah dilakukan sebelumnya yang berkenaan dengan penelitian yang penulis lakukan.

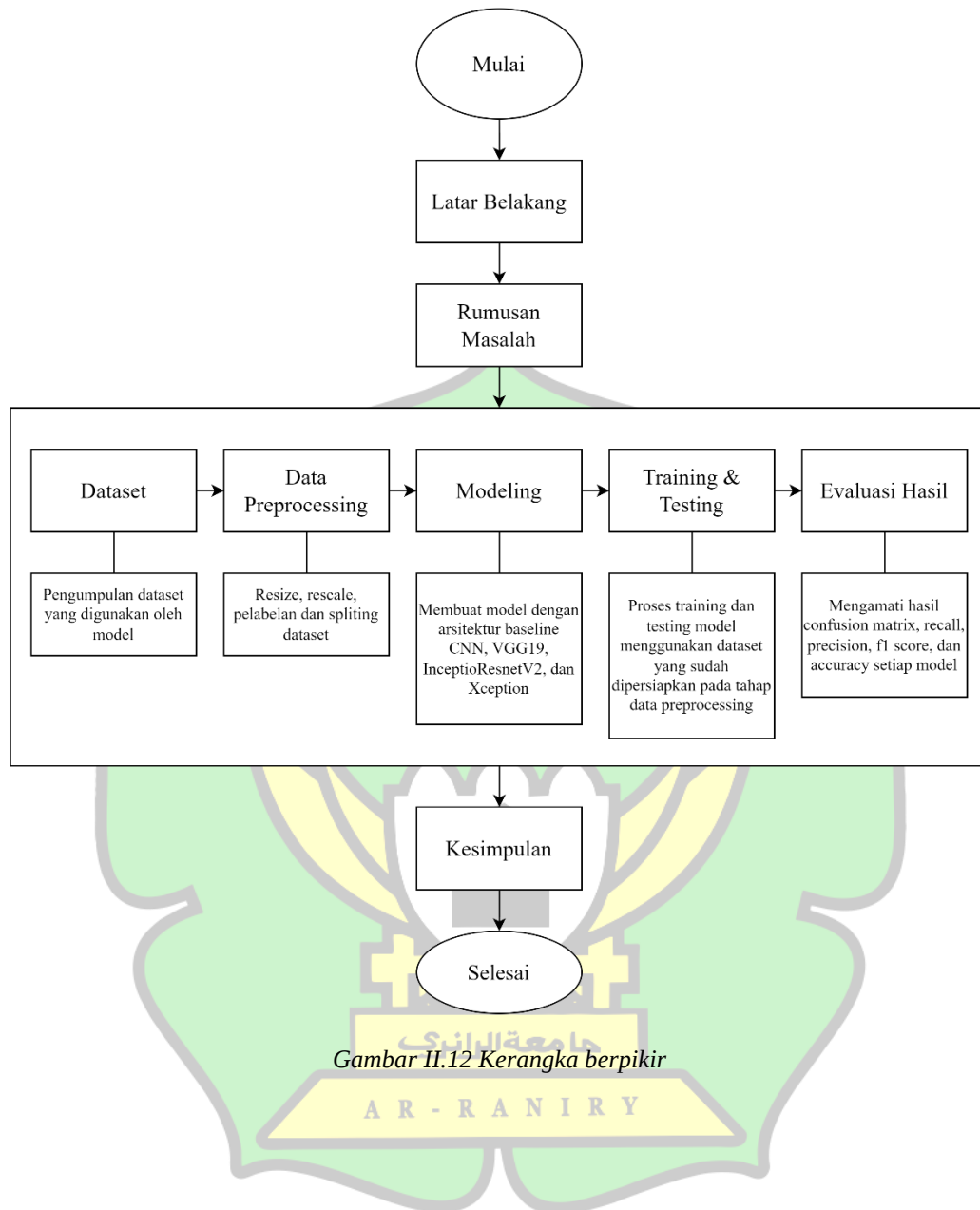
Tabel II.2 Penelitian terdahulu

No	Peneliti	Judul	Model	Sumber Dataset	Jumlah Dataset	Jumlah Class	Akurasi
1	Ioannis D Apos- tolopoulos, dan Tzani A Mpesiana	Covid-19: automatic detec- tion from X-ray images utilizing transfer learning with convolutional neural networks	VGG19, MobileNet, Inception, Xception, InceptionRes- NetV2	Radiologi- cal Society of North America (RSNA), Radiopaed- ia, Italian Society of Medical and Inter- ventional Radiology (SIRM).	1.427	3	97,82%

2	Muhammad Ridho Fauzi, Puspa Eosina, dan Dewi Primasari	DETEKSI CORONAVIRUS DISEASE PADA X-RAY DAN CT-SCAN PARU MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK	Baseline CNN	Tidak disebutkan	5.074	3	95,00%
3	Ines Chouat, Amira Echtioui, Rafik Khemakhem, Wassim Zouch, Mohamed Ghorbel, Ahmed Ben Hamida	COVID-19 detection in CT and CXR images using deep learning models	ResNet50, InceptionV3, VGGNet-19, dan Xception	Github, Kaggle	5.356	2	90,50%
4	Sumit Malik, Shiven-dra Singh, Narendra Mohan Singh, Naman Panwar	Diagnosis of COVID-19 Using Chest X-ray	InceptionV3, Xception, ResNeXt	kaggle	6.432	3	97,00%

Pada penelitian ini akan dilakukan pendeteksian *COVID-19* pada citra *x-ray* paru-paru dengan beberapa model arsitektur yaitu *VGG19*, *Xception*, *InceptionResnetV2*, dan *baseline CNN*. Dataset yang digunakan adalah citra *x-ray* paru-paru yang berjumlah 21.165 yang terdiri dari 4 *class* yaitu *COVID-19* dengan jumlah 3.616, *viral pneumonia* dengan jumlah 1.345, *lung opacity* dengan jumlah 6.012 dan normal dengan jumlah 10.192.

II.14 Kerangka Berpikir



Gambar II.12 Kerangka berpikir

BAB III

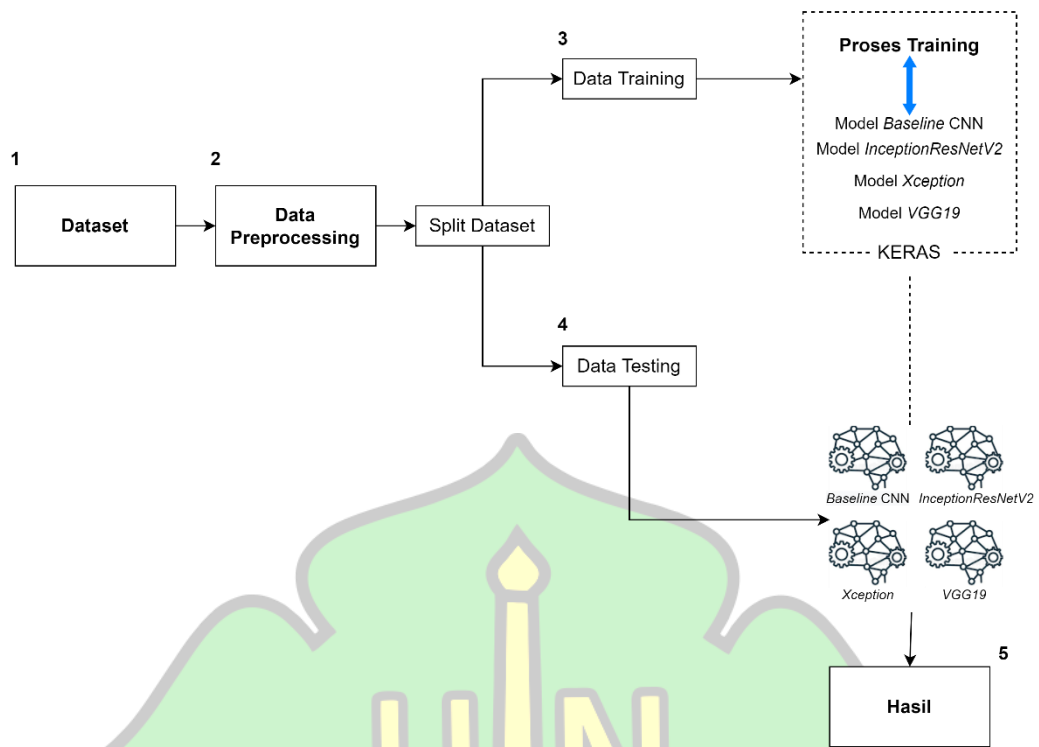
METODE PENELITIAN

III.1 Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah penelitian kualitatif dan penelitian pengembangan (*Research and Development/ R&D*). Penelitian kualitatif merupakan penelitian untuk menjawab permasalahan yang memerlukan analisis data penelitian yang bersifat kualitatif (Arsyam, 2019). Sedangkan penelitian pengembangan (*Research and Development/ R&D*) merupakan penelitian yang ditujukan untuk mengembangkan suatu produk baru atau menyempurnakan produk yang telah ada. Produk tersebut tidak selalu berbentuk benda atau perangkat keras (*hardware*), tetapi bisa juga perangkat lunak (*software*) seperti program komputer (Arsyam, 2019). Penelitian ini dilakukan untuk membangun beberapa model arsitektur CNN yaitu *baseline* CNN, *VGG19*, *InceptionResnetV2*, dan *Xception* untuk mengetahui model arsitektur CNN mana yang melakukan pendeteksian *COVID-19* dengan tingkat akurasi terbaik pada citra *x-ray* paru-paru.

III.2 Tahapan Penelitian

Metode penelitian yang digunakan pada penelitian ini terdiri atas beberapa tahapan pengembangan model *deep learning*. Tahapan-tahapan tersebut dapat dilihat pada gambar 12.



Gambar III.1 Tahapan penelitian

1. Dataset

Tahapan pertama dalam penelitian ini adalah menyiapkan dataset yang terdiri dari kumpulan citra *x-ray* paru-paru. Dataset tersebut digunakan sebagai data yang akan dipelajari oleh model. Dataset citra *x-ray* paru-paru didapatkan dari situs kaggle.com. Dataset terdiri dari citra *x-ray* beberapa jenis penyakit infeksi paru-paru sehingga nantinya membuat model dapat membedakan citra *x-ray* pasien *COVID-19* dengan citra *x-ray* paru-paru normal dan penyakit lainnya. Dataset yang penulis ambil terdiri dari data gambar *COVID-19* berjumlah 3616, *lung opacity* 6012, normal 10,192, dan *pneumonia* yang berjumlah 1345. Semua gambar pada dataset mempunyai resolusi 256x256 piksel dengan format *file portable network graphics* (PNG). <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database> diakses pada tanggal 16 Maret 2022.



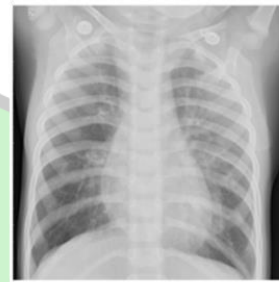
Gambar (a)



Gambar (b)



Gambar (c)



Gambar (d)

Gambar III.2 (a) COVID-19 (b) Normal (c) Lung opacity (d) Pneumonia

2. Data Preprocessing

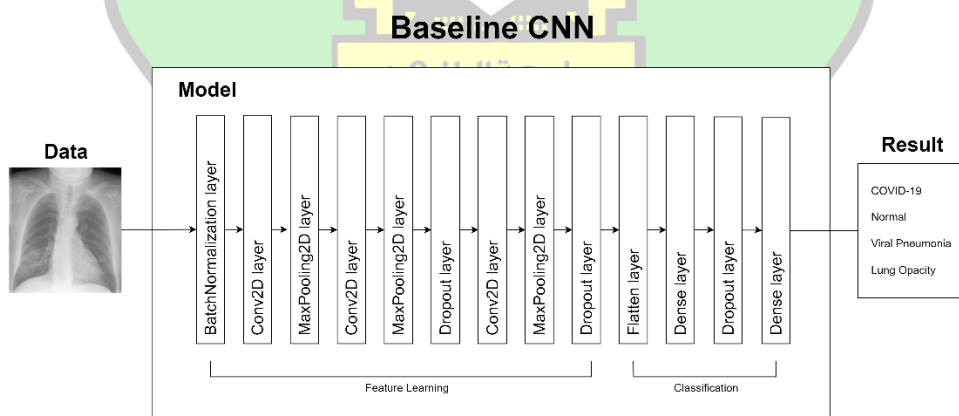
Pada tahap ini dilakukan proses *splitting* dataset menjadi 2 kategori yaitu data *training* sebesar 90% dan data *testing* sebesar 10% (Madiraju, 2020). Berdasarkan jumlah keseluruhan dataset yaitu 21.165 maka data *training* sebesar 19.050 dan data *testing* sebesar 2.115. Data *training* merupakan data yang dipelajari oleh model pada tahap *training*. Sedangkan data *testing* adalah data yang digunakan oleh model untuk mengetahui tingkat keakuratan model pada tahap *testing*. Tahap *testing* dilakukan setelah tahap *training* selesai dilakukan. Selain *splitting* dataset pada tahap *preprocessing* juga dilakukan proses pelabelan, *rescale*, dan *resize* pada gambar. Tujuan dari tahap *preprocessing* adalah agar dataset siap digunakan oleh model untuk dipelajari.

3. Model CNN

Pada tahap ini akan dibangun beberapa model menggunakan Metode CNN dengan *Keras*. Model yang dibangun terdiri dari 4 jenis yaitu model yaitu *Baseline CNN*, *InceptionResnetV2* (Szegedy dkk., 2017), *Xception* (Chollet, 2017), dan *VGG19* (Simonyan & Zisserman, 2015). Secara keseluruhan 4 model tersebut terdiri dari 2 bagian yaitu *feature learning* dan *classification*. *Feature learning* merupakan bagian dimana proses konvolusi terjadi untuk pengenalan data oleh model. Sedangkan *classification* merupakan bagian dimana model memprediksi data.

Perbedaan dari setiap model terletak pada bagian *feature learning*. Pada *Baseline CNN* *feature learning* dibuat berdasarkan konsep dari metode CNN itu sendiri. Sedangkan pada model *InceptionResnetV2*, *Xception* dan *VGG19* bagian *feature learning* menggunakan *layer* yang sudah ada pada model tersebut. Arsitektur dari setiap model dapat dilihat di bawah ini.

a. *Baseline CNN*



Gambar III.3 Arsitektur CNN

Cara kerja dari masing-masing layer yang ada pada arsitektur *Baseline CNN* adalah sebagai berikut:

1) *BacthNormalization layer*

BacthNormalization layer merupakan layer yang menerapkan transformasi untuk mengatur keluaran mean mendekati 0 dan keluaran standard deviation mendekati 1.

2) *Conv2D layer*

Conv2D layer merupakan layer yang berfungsi melakukan proses konvolusi. Pada arsitektur ini proses konvolusi dilakukan sebanyak 4 kali. Konvolusi dilakukan sebanyak 4 kali karena selama penelitian model *Baseline* CNN mendapatkan hasil yang lebih baik jika dilakukan 4 kali proses konvolusi.

3) *MaxPooling2D layer*

MaxPooling2D layer merupakan layer yang berfungsi melakukan proses pooling pada data setelah proses konvolusi selesai dilakukan.

4) *Dropout layer*

Dropout layer merupakan layer yang digunakan untuk mencegah overfitting pada model.

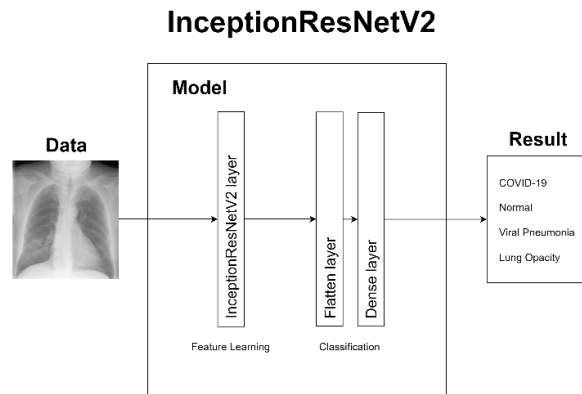
5) *Flatten layer*

Flatten layer merupakan layer yang berfungsi untuk meratakan dimensi dari data.

6) *Dense layer*

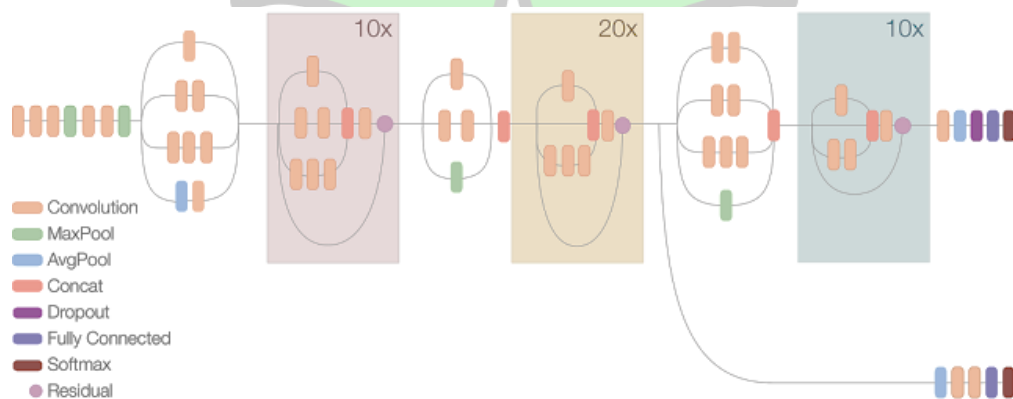
Dense layer merupakan layer *neural network* yang pada hal ini digunakan untuk melakukan klasifikasi pada gambar.

b. *InceptionResnetV2*



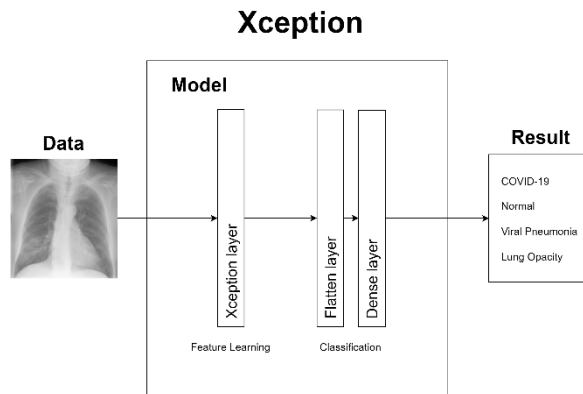
Gambar III.4 Arsitektur InceptionResnetV2

Pada arsitektur ini untuk mendapatkan hasil, data yang kita input di masukkan ke dalam model dengan beberapa *layer*. Berbeda dengan *Baseline CNN* pada Arsitektur ini *feature learning* menggunakan *InceptionResnetV2 layer*. Sebuah *layer* yang berisi model *InceptionResnetV2*. *Layer – layer* yang terdapat di dalam model *InceptionResnetV2* dapat dilihat pada gambar 16. *InceptionResnetV2* bekerja mirip seperti CNN pada umumnya, menggunakan *layer* konvolusi dan *layer max pooling*. Perbedaannya terletak saat proses konvolusi. Gambar masukkan didistribusikan pada setiap cabang *layer* konvolusi untuk dilakukan *feature learning*. Kemudian hasil *feature learning* dari setiap cabang di satukan kembali untuk dilakukan klasifikasi atau pendeteksian.



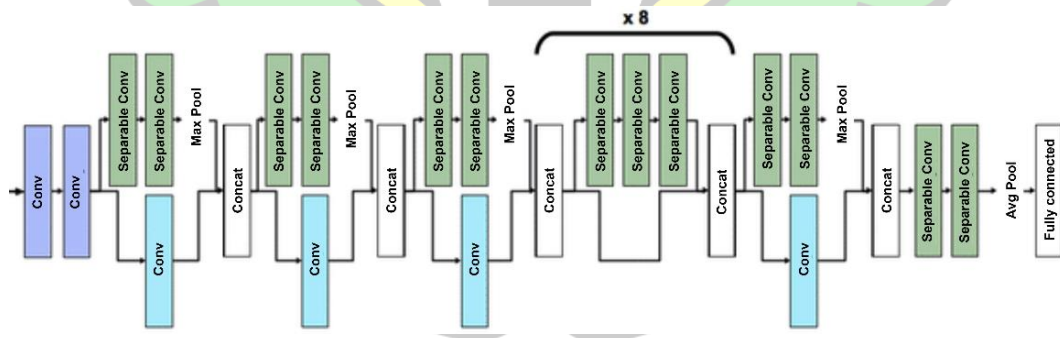
Gambar III.5 InceptionResnetV2 layer (Bockenek, 2014)

c. Xception



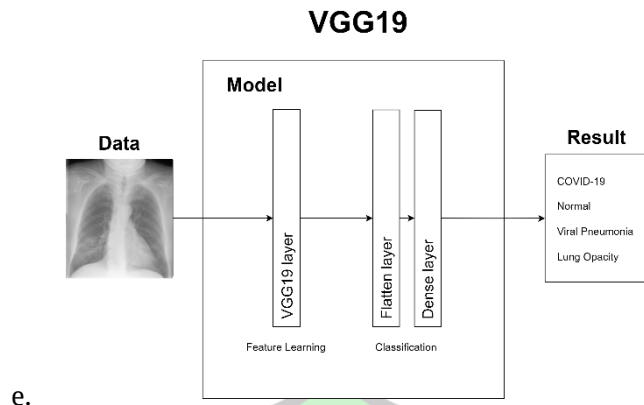
Gambar III.6 Arsitektur Xception

Arsitektur ini mirip dengan arsitektur *InceptionResnetV2* hanya saja pada proses *feature learning* menggunakan *Xception layer*. Sebuah layer yang berisi model *Xception*. Layer yang terdapat di dalam model *Xception* dapat dilihat pada gambar 18. *Xception* bekerja mirip seperti *InceptionResnetV2* dengan melakukan *feature learning* pada layer konvolusi yang terpisah. Kemudian hasil *feature learning* digunakan untuk klasifikasi atau pendeteksian.



Gambar III.7 Xception layer (Leonardo dkk., 2018)

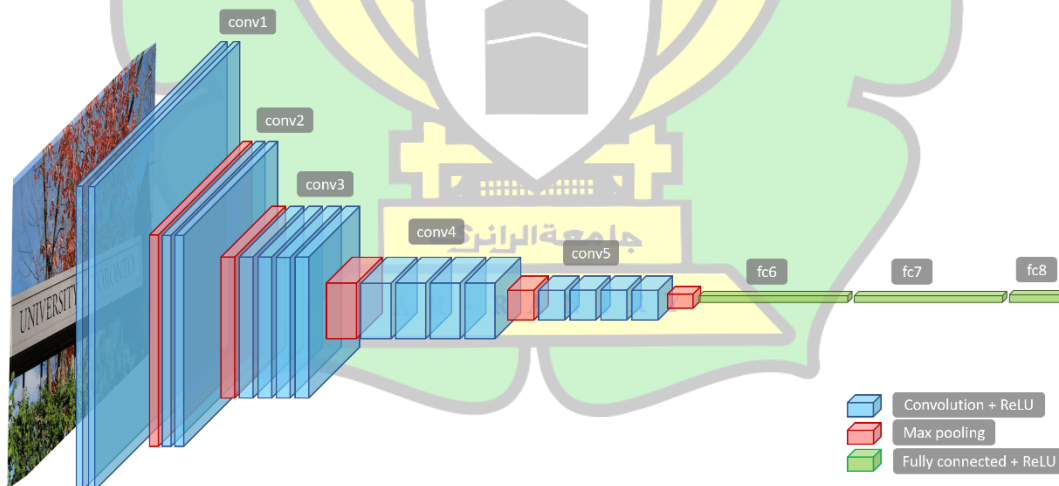
d. VGG19



e.

Gambar III.8 Arsitektur VGG19

Arsitektur ini melakukan proses *feature learning* menggunakan VGG19 layer. Sebuah layer yang berisi model VGG19. Layer yang terdapat di dalam model VGG19 dapat dilihat pada gambar 20. VGG19 mirip seperti *baseline CNN*, namun dengan jumlah layer yang berbeda. Disebut VGG19 karena terdiri dari 16 layer konvolusi dan 3 layer *fully connected*.



Gambar III.9 VGG19 layer (Mohammad Reza Ebrahimi, 2022)

4. Training dan testing model

Pada tahap ini model CNN akan melalui proses *training* dan *testing* dengan data yang sudah diproses sebelumnya pada tahap data *preprocessing*. Kemudian model akan diukur tingkat keakuratannya pada tahapan evaluasi hasil.

5. Evaluasi Hasil

Pada tahapan terakhir dalam penelitian ini dilakukan evaluasi untuk mengetahui hasil kinerja dari setiap model. Hasil kinerja dapat diketahui dari perhitungan menggunakan *confusion matrix*, *recall*, *precision*, *f1 score* dan *accuracy*.

III.3 Jenis Data

Jenis data yang digunakan dalam penelitian ini adalah data sekunder. Data sekunder adalah data yang diperoleh peneliti berdasarkan pengukuran secara tidak langsung dari sumber atau subjek penelitiannya. Umumnya data sekunder diperoleh lewat data yang sudah terkumpul dari pihak lain. Dalam penelitian ini, data sekunder diperoleh dari *website kaggle.com*. Data pada *website kaggle.com* tersebut dibuat oleh para peneliti sebelumnya yang melakukan pendeteksian *COVID-19* pada citra *x-ray* paru-paru (Chowdhury dkk., 2020; Rahman dkk., 2021).

III.4 Teknik Pengumpulan Data

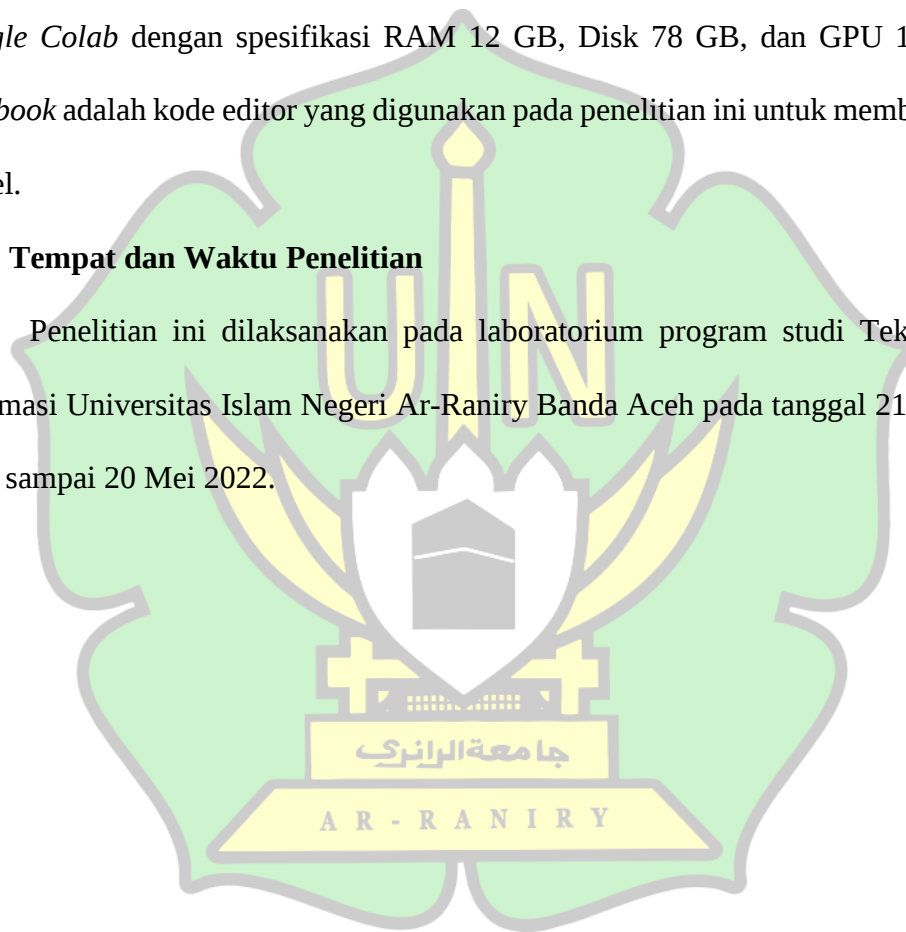
Data yang digunakan pada penelitian ini adalah dataset yang terdiri dari kumpulan citra *x-ray* paru-paru. Dataset citra *x-ray* paru-paru didapatkan dari situs *kaggle.com*. Dataset yang penulis ambil berjumlah 21.165 gambar yang terdiri dari gambar *COVID-19* berjumlah 3.616, *lung opacity* 6.012, normal 10.192, dan *pneumonia* yang berjumlah 1.345. Semua gambar pada dataset mempunyai resolusi 256x256 piksel dengan format *file portable network graphics* (PNG).
<https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>
diakses pada tanggal 16 Maret 2022.

III.5 Alat dan Bahan

Pada penelitian ini bahasa pemrograman yang digunakan adalah *Python* 3.7.12 dengan *framework Keras* 2.7.0. Selain itu pada penelitian ini juga memanfaatkan beberapa fitur dari *numpy* dan *pandas* untuk manipulasi dan analisis data, *matplotlib* dan *seaborn* untuk visualisasi data, dan *scikit-learn* untuk menghasilkan *report* terhadap kinerja model. *Notebook* yang digunakan adalah *Google Colab* dengan spesifikasi RAM 12 GB, Disk 78 GB, dan GPU 15 GB. *Notebook* adalah kode editor yang digunakan pada penelitian ini untuk membangun model.

III.6 Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan pada laboratorium program studi Teknologi Informasi Universitas Islam Negeri Ar-Raniry Banda Aceh pada tanggal 21 Maret 2022 sampai 20 Mei 2022.



BAB IV

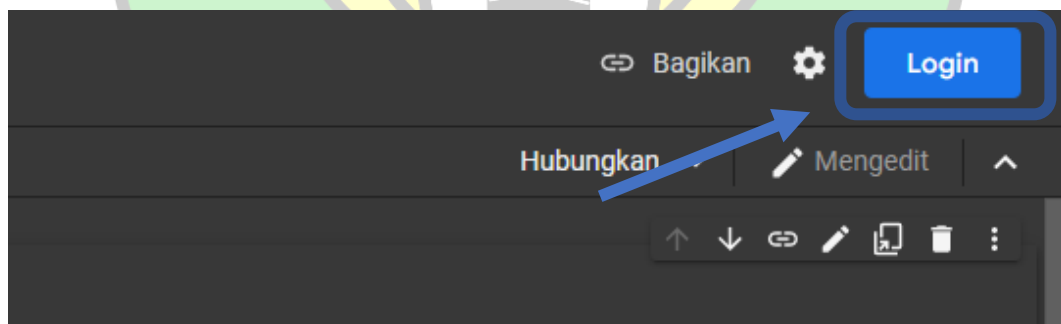
HASIL PENELITIAN DAN PEMBAHASAN

IV.1 Persiapan Software

a. Google Colaboratory

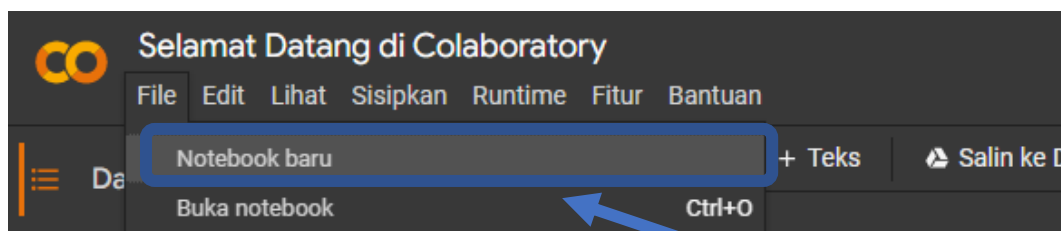
Google colaboratory merupakan sebuah *hosting environment notebook jupyter* yang dapat digunakan untuk kode python. Di dalam *google colaboratory* sudah tersedia berbagai macam *library* yang biasa digunakan untuk keperluan *data science*. Selain itu, untuk mempercepat proses pengembangan suatu model, *Google colaboratory* memberikan akses gratis terhadap GPU dan TPU. Berikut langkah – langkah untuk memulai menggunakan *google colaboratory*:

- 1) Akses halaman website <https://colab.research.google.com/>
- 2) Kemudian pilih login, masukkan *email* dan *password* atau masuk dengan akun *google*.



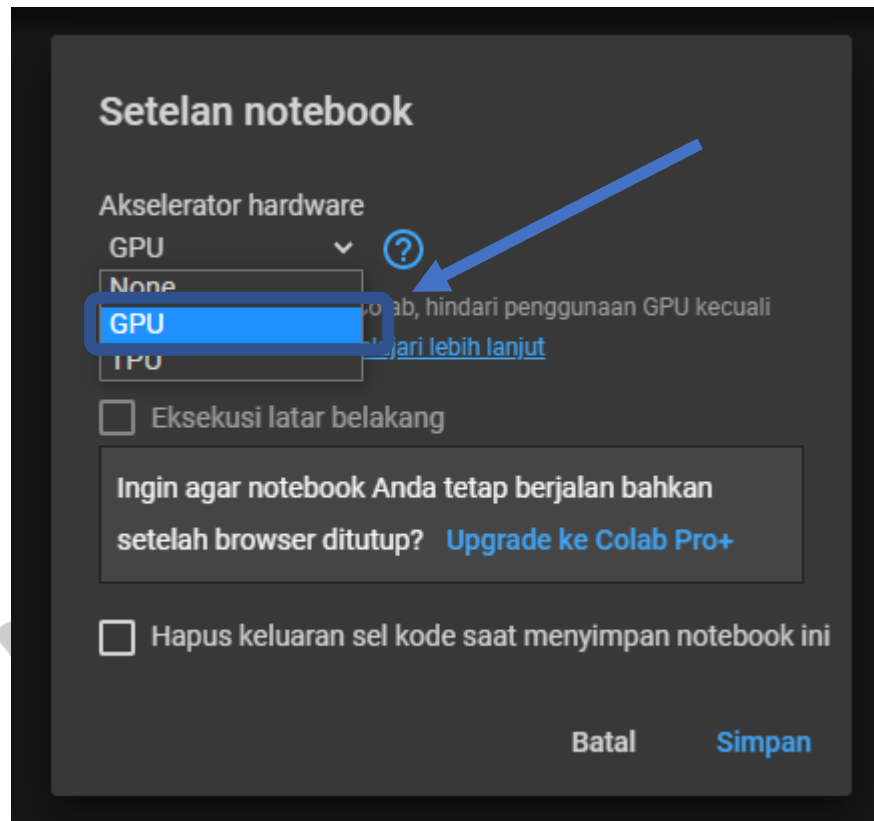
Gambar IV.1 Login google colab

- 3) Setelah berhasil masuk, pilih file kemudian *notebook* baru.



Gambar IV.2 Notebook baru

- 4) Terakhir ubah jenis *runtime* menggunakan GPU. Pilih *runtime*, kemudian pilih ubah jenis *runtime*, pilih GPU.



Gambar IV.3 Ubah jenis runtime

IV.2 Pengambilan Dataset

Setelah persiapan *environment* selesai maka dilanjutkan dengan proses pengambilan dataset citra x-ray paru - paru dari *website kaggle*. Tahapan – tahapan akan dilakukan seperti berikut ini:

- a. Hubungkan *google colaboratory* dengan *kaggle*

Google colaboratory dihubungkan dengan *kaggle* menggunakan *API key* dari *kaggle*. *API key* diperoleh dari akun *kaggle* dan kemudian di *upload* ke dalam *environment google colaboratory*. Berikut kode untuk menghubungkan *google colaboratory* dengan *kaggle*:

```
# Connect to kaggle
!pip install kaggle
!mkdir ~/.kaggle

uploaded = files.upload()
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
```

Gambar IV.4 Kode untuk menghubungkan google colab dan kaggle

b. Download dataset

Selanjutnya proses unduh dataset citra x-ray paru – paru dari *kaggle*. Setelah *download* selesai *file* dataset yang berformat *zip* kemudian di ekstrak. Berikut kode untuk mengunduh dataset kemudian mengekstraknya:

```
[ ] # Download and extract dataset
!kaggle datasets download -d tawsifurrahman/covid19-radiography-database
!unzip covid19-radiography-database.zip
```

Gambar IV.5 Kode download dataset

c. Restructuring folder dataset

Proses *restructuring* folder merupakan proses yang bertujuan untuk merapikan struktur *folder* dataset agar lebih mudah di kelola. Berikut kode yang digunakan dalam *restructuring folder*:

```
# Dataset folder restructuring
!mv COVID-19_Radiography_Dataset/COVID/images/ COVID-19_Radiography_Dataset/
!rm -r COVID-19_Radiography_Dataset/COVID/
!mv COVID-19_Radiography_Dataset/images COVID-19_Radiography_Dataset/COVID

!mv COVID-19_Radiography_Dataset/Lung_Opacity/images/ COVID-19_Radiography_Dataset/
!rm -r COVID-19_Radiography_Dataset/Lung_Opacity/
!mv COVID-19_Radiography_Dataset/images COVID-19_Radiography_Dataset/Lung_Opacity

!mv COVID-19_Radiography_Dataset/Normal/images/ COVID-19_Radiography_Dataset/
!rm -r COVID-19_Radiography_Dataset/Normal/
!mv COVID-19_Radiography_Dataset/images COVID-19_Radiography_Dataset/Normal

!mv COVID-19_Radiography_Dataset/Viral\ Pneumonia/images/ COVID-19_Radiography_Dataset/
!rm -r COVID-19_Radiography_Dataset/Viral\ Pneumonia/
!mv COVID-19_Radiography_Dataset/images COVID-19_Radiography_Dataset/Viral\ Pneumonia
```

Gambar IV.6 Kode restructuring folder

IV.3 Data *preprocessing*

Tahap data *preprocessing* bertujuan agar dataset siap digunakan oleh model untuk dipelajari. Beberapa proses yang terjadi pada tahap ini adalah sebagai berikut:

a. *Splitting dataset*

Proses *splitting* dilakukan untuk membagi dataset menjadi 2 bagian yaitu data *training* sebesar 90% dan data *testing* sebesar 10%. Berdasarkan jumlah keseluruhan dataset yaitu 21.165 maka data *training* sebesar 19.050 dan data *testing* sebesar 2.115.

b. *Rescale*

Pada proses ini nilai dari setiap piksel pada citra diskalakan menjadi nilai antara 0 dan 1. *Rescale* dilakukan agar proses komputasi saat training berjalan lebih cepat karena hanya memproses angka dengan nilai antara 0 dan 1 dimana angka tersebut mempunyai alokasi memori yang kecil (Chollet, 2021).

c. *Pelabelan*

Pada proses ini citra x-ray paru-paru yang berjumlah 21.165 diberikan label sesuai dengan *classnya* masing-masing. 3.616 gambar di beri label *COVID-19*, 1.345 diberi label *viral pneumonia*, 6.012 gambar diberi label *lung opacity* dan 10.192 diberi label *normal*.

d. *Resize*

Pada proses ini ukuran citra di ubah dari 256x256 piksel menjadi 299x299 piksel. *Resize* dilakukan karena model seperti *InceptionResnetV2*, *Xception*, dan *VGG19* menerima gambar dengan ukuran 299x299 secara *default*.

IV.4 Model

Setelah tahap data *preprocessing* selesai maka dilanjutkan dengan tahapan membangun model *deep learning*. Model yang dibangun terdiri dari 4 jenis yaitu model *baseline* CNN, *InceptionResnetV2*, *Xception*, dan *VGG19*. Tahapan awal yang dilakukan untuk membangun setiap model adalah dengan mendefinisikannya terlebih dahulu. Setelah definisi model selesai, selanjutnya dilakukan *compile* model dengan *optimizer*, *loss function*, dan *metrics* sebagai parameternya.

Berikut penjelasan tentang *optimizer*, *loss function*, dan *metrics*:

1. *Optimizer* adalah mekanisme yang digunakan model untuk meningkatkan kinerjanya dengan cara memperbarui dirinya sendiri menggunakan algoritma *backpropagation* berdasarkan data pelatihan yang dilihatnya.
2. *Loss function* adalah fungsi untuk mengukur kinerja model pada data pelatihan. Hasil kinerja model tersebut kemudian digunakan oleh *optimizer* untuk memperbarui model agar bekerja lebih baik.
3. *Metrics* digunakan untuk memantau model selama proses *training* (Chollet, 2021).

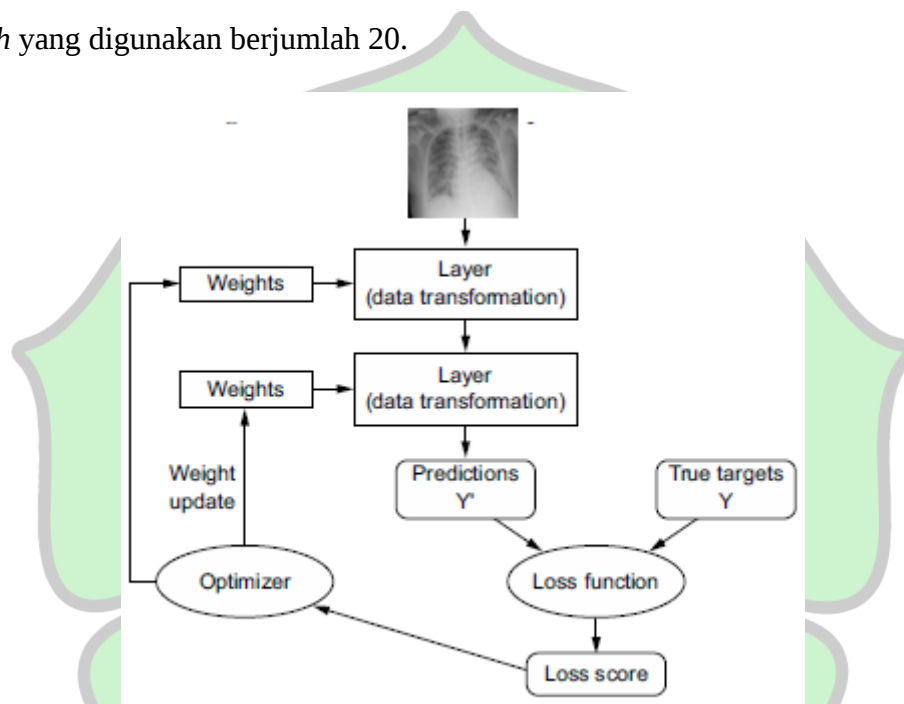
IV.5 Training & Testing

Setelah model didefinisikan dan proses *compile* selesai maka tahap selanjutnya adalah *training* dan *testing*. Berikut tahapan yang akan dilakukan:

a. Training

Training adalah proses bagi model untuk mempelajari data. Data yang digunakan adalah data *training* yang sudah dipersiapkan sebelumnya pada tahap data *preprocessing*. Proses *training* secara keseluruhan di tunjukkan pada gambar 27. *Training* dimulai dengan memasukkan data citra *x-ray* paru – paru kedalam

model. Citra *x-ray* kemudian melalui proses transformasi data dalam *layer* CNN. Selanjutnya model menghasilkan prediksi terhadap data yang kemudian digunakan oleh *loss function* untuk menghasilkan *loss score* dengan membandingkannya dengan hasil yang sebenarnya. *Optimizer* kemudian memperbarui *weight* pada tiap – tiap *layer* CNN berdasarkan *loss score* yang didapatkan dari *loss function*. Proses ini terus berulang sesuai dengan jumlah *epoch* yang digunakan. Pada penelitian ini *epoch* yang digunakan berjumlah 20.



Gambar IV.7 Proses training (Chollet, 2021)

Hasil dari proses training dapat dilihat dari nilai *accuracy* dan *loss score* yang ditunjukkan oleh setiap model. Nilai *accuracy* menunjukkan kemampuan model dalam mengenali dan memprediksi data dengan benar, semakin tinggi nilai *accuracy* maka model semakin baik. Sebaliknya, *loss score* adalah nilai kesalahan model dalam mengenali dan memprediksi data, semakin rendah nilai *loss score* maka model semakin baik. Berikut nilai *accuracy*, *loss score*, serta waktu yang dibutuhkan oleh setiap model pada proses *training*:

1) CNN

Tabel IV.1 Hasil training baseline CNN

<i>Epoch</i>	<i>Training Accuracy</i>	<i>Validation Accuracy</i>	<i>Training Loss</i>	<i>Validation Loss</i>	<i>Time (s)</i>
1	0,570	0,536	2,062	1,235	16
2	0,628	0,641	0,848	0,912	14
3	0,691	0,724	0,758	0,837	14
4	0,705	0,750	0,738	0,67	14
5	0,726	0,770	0,679	0,624	14
6	0,763	0,758	0,603	0,61	14
7	0,772	0,782	0,578	0,555	14
8	0,791	0,795	0,578	0,538	14
9	0,805	0,805	0,516	0,505	14
10	0,793	0,816	0,534	0,483	14
11	0,811	0,816	0,488	0,488	14
12	0,808	0,818	0,507	0,459	14
13	0,813	0,832	0,462	0,45	14
14	0,848	0,827	0,428	0,449	14
15	0,828	0,838	0,447	0,421	14
16	0,836	0,829	0,417	0,461	14
17	0,848	0,844	0,401	0,414	14
18	0,844	0,829	0,414	0,453	14
19	0,829	0,836	0,413	0,458	14
20	0,850	0,831	0,384	0,442	14

Berdasarkan hasil *training* model CNN yang ditunjukkan pada tabel 3. Model di *training* menggunakan 20 *epoch*. Pada *epoch* terakhir model mempunyai nilai *training accuracy* 0,850, *validation accuracy* 0,831, *training loss* 0,384, dan *validation loss* 0,442.

2) *InceptionResnetV2*

Berdasarkan hasil *training* model *InceptionResnetV2* yang ditunjukkan pada tabel 4. Model di *training* menggunakan 20 *epoch*. Pada *epoch* terakhir model mempunyai nilai *training accuracy* 0,959, *validation accuracy* 0,923, *training loss* 0,110, dan *validation loss* 0,327.

Tabel IV.2 Hasil training InceptionResnetV2

<i>Epoch</i>	<i>Training Accuracy</i>	<i>Validation Accuracy</i>	<i>Training Loss</i>	<i>Validation Loss</i>	<i>Time (s)</i>
1	0,61	0,658	0,936	0,847	108
2	0,754	0,753	0,633	0,65	83
3	0,808	0,806	0,521	0,585	82
4	0,833	0,819	0,43	0,634	82
5	0,873	0,84	0,346	0,536	82
6	0,89	0,856	0,32	0,503	82
7	0,896	0,878	0,292	0,449	82
8	0,912	0,901	0,262	0,396	82
9	0,908	0,896	0,257	0,403	82
10	0,912	0,914	0,232	0,339	82
11	0,933	0,914	0,204	0,353	100
12	0,927	0,913	0,198	0,358	82
13	0,945	0,915	0,154	0,356	82
14	0,946	0,921	0,151	0,339	82
15	0,949	0,917	0,158	0,309	82
16	0,956	0,921	0,13	0,315	82
17	0,947	0,926	0,141	0,316	82
18	0,967	0,923	0,115	0,345	82
19	0,96	0,921	0,118	0,331	82
20	0,959	0,923	0,110	0,327	82

3) *Xception*

Berdasarkan hasil training model *Xception* yang ditunjukkan pada tabel 5. Model di training menggunakan 20 *epoch*. Pada *epoch* terakhir model mempunyai nilai *training accuracy* 0,960, *validation accuracy* 0,933, *training loss* 0,127, dan *validation loss* 0,192.

Tabel IV.3 Hasil training Xception

<i>Epoch</i>	<i>Training Accuracy</i>	<i>Validation Accuracy</i>	<i>Training Loss</i>	<i>Validation Loss</i>	<i>Time (s)</i>
1	0,683	0,761	0,787	0,679	84
2	0,818	0,815	0,474	0,52	77
3	0,841	0,786	0,445	0,544	77
4	0,874	0,833	0,361	0,463	77
5	0,885	0,843	0,329	0,421	77
6	0,905	0,855	0,253	0,37	77
7	0,918	0,873	0,246	0,349	77
8	0,912	0,896	0,242	0,275	77
9	0,921	0,916	0,226	0,221	77
10	0,93	0,922	0,205	0,21	77
11	0,943	0,926	0,174	0,209	77
12	0,933	0,93	0,175	0,19	77
13	0,939	0,926	0,18	0,194	77
14	0,937	0,919	0,176	0,222	77
15	0,945	0,931	0,145	0,195	77
16	0,955	0,931	0,141	0,179	77
17	0,942	0,933	0,172	0,188	77
18	0,953	0,927	0,139	0,198	77
19	0,959	0,936	0,117	0,184	77
20	0,960	0,933	0,127	0,192	77

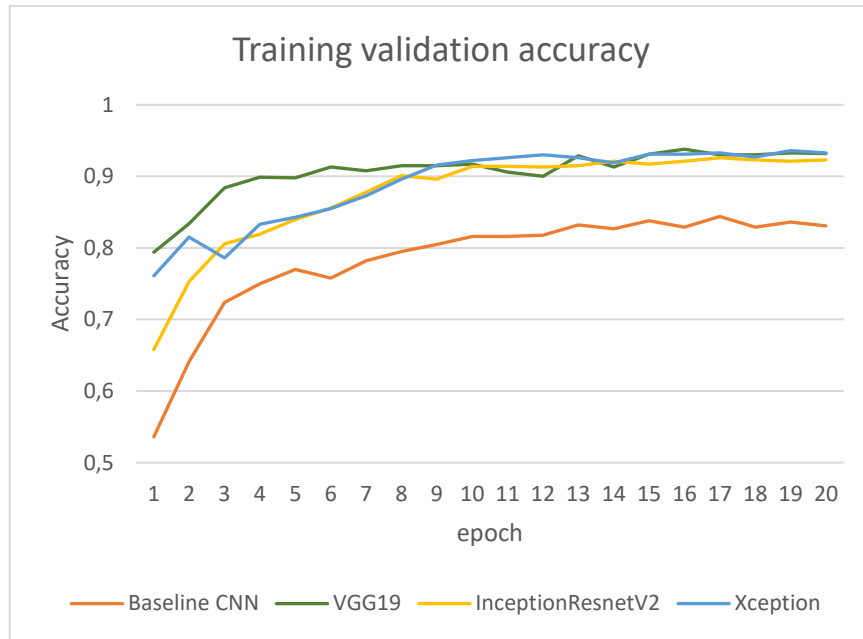
4) VGG19

Berdasarkan hasil *training* model VGG19 yang ditunjukkan pada tabel 6. Model di *training* menggunakan 20 *epoch*. Pada *epoch* terakhir model mempunyai nilai *training accuracy* 0,948, *validation accuracy* 0,932, *training loss* 0,141, dan *validation loss* 0,186.

Tabel IV.4 Hasil training VGG19

<i>Epoch</i>	<i>Training Accuracy</i>	<i>Validation Accuracy</i>	<i>Training Loss</i>	<i>Validation Loss</i>	<i>Time (s)</i>
1	0,633	0,794	0,883	0,571	71
2	0,816	0,834	0,477	0,454	72
3	0,841	0,884	0,424	0,324	73
4	0,866	0,899	0,362	0,272	73
5	0,881	0,898	0,311	0,288	73
6	0,909	0,913	0,272	0,253	74
7	0,896	0,908	0,259	0,266	74
8	0,908	0,915	0,254	0,236	74
9	0,903	0,915	0,258	0,237	73
10	0,928	0,917	0,213	0,222	73
11	0,921	0,906	0,221	0,272	73
12	0,919	0,9	0,231	0,261	74
13	0,928	0,929	0,204	0,19	74
14	0,913	0,913	0,249	0,238	73
15	0,931	0,931	0,186	0,201	73
16	0,923	0,938	0,192	0,189	74
17	0,94	0,93	0,165	0,199	73
18	0,935	0,93	0,186	0,196	73
19	0,936	0,933	0,169	0,191	73
20	0,948	0,932	0,141	0,186	74

Pada gambar 29 menunjukkan diagram dari hasil keseluruhan training model *Baseline CNN*, *InceptionResnetV2*, *VGG19*, dan *Xception*. Diagram digambarkan dengan perolehan nilai *accuracy* oleh model pada tiap *epoch*. Pada *epoch* 1 sampai dengan 10 terlihat setiap model mengalami peningkatan nilai *accuracy*. Kemudian pada *epoch* 10 sampai dengan 20 nilai *accuracy* cenderung datar atau tidak mengalami peningkatan yang signifikan.



Gambar IV.8 Validation accuracy setiap model

b. Testing

Setelah proses *training* selesai maka selanjutnya dilakukan *testing* model. Data yang digunakan adalah data *testing* yang sudah dipersiapkan sebelumnya pada tahapan data *preprocessing*.

Tabel IV.5 Hasil testing

Model	Testing Accuracy	Testing Loss	Time (s)
Baseline CNN	0,831	0,442	8
VGG19	0,932	0,213	24
InceptionResnetV2	0,923	0,327	24
Xception	0,933	0,192	17

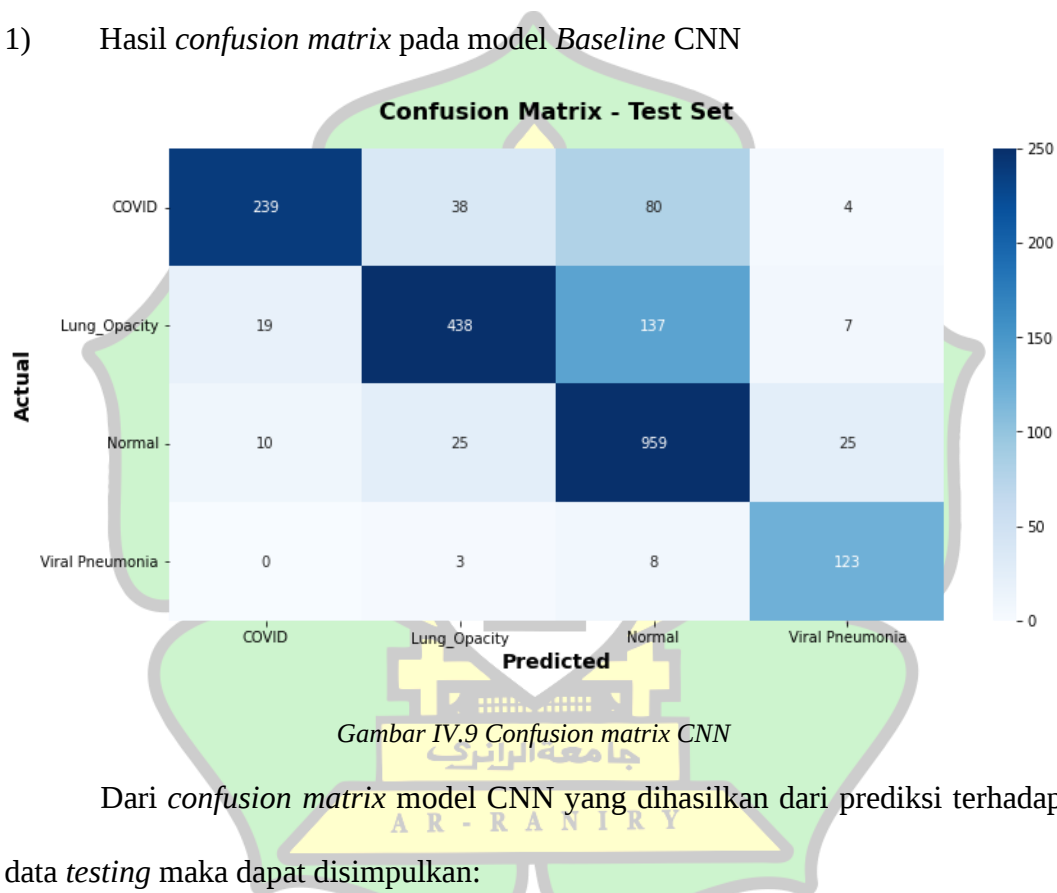
Pada Tabel 7 menunjukkan nilai *accuracy* dan *loss score* yang dihasilkan oleh setiap model pada proses *testing*. Model yang baik adalah model yang memiliki *accuracy* tertinggi dengan *loss score* terendah. Dari gambar 36 dapat dilihat Xception adalah model terbaik yang memiliki *accuracy* 0,933 dan *loss score* 0,192.

IV.6 Hasil

Setelah proses *training* dan *testing* selesai maka selanjutnya dilakukan proses pengukuran. Model melakukan prediksi terhadap data *testing* citra *x-ray* paru-paru. Kemudian hasil prediksi tersebut diukur menggunakan *confusion matrix*, *recall*, *precision*, *f1 score*, dan *accuracy*.

a. Confusion Matrix

- 1) Hasil *confusion matrix* pada model *Baseline CNN*

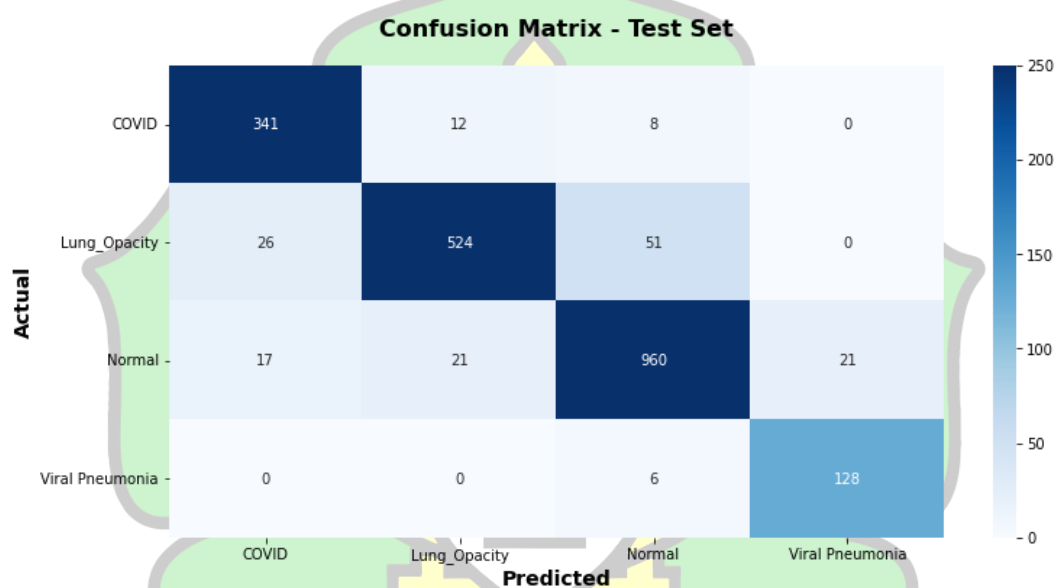


Gambar IV.9 Confusion matrix CNN

Dari *confusion matrix* model CNN yang dihasilkan dari prediksi terhadap data *testing* maka dapat disimpulkan:

- a) Dari total 361 gambar pada *class covid*. Sebanyak 239 gambar di prediksi dengan benar, 38 gambar diprediksi sebagai *lung opacity*, 80 gambar diprediksi sebagai normal, dan 4 gambar sebagai *viral pneumonia*.
- b) Dari total 601 gambar pada *class lung opacity*. Sebanyak 438 gambar di prediksi dengan benar, 19 gambar diprediksi sebagai *covid*, 137 gambar diprediksi sebagai normal, dan 7 gambar sebagai *viral pneumonia*.

- c) Dari total 1019 gambar pada *class* normal. Sebanyak 959 gambar di prediksi dengan benar, 10 gambar diprediksi sebagai *covid*, 25 gambar diprediksi sebagai *lung opacity*, dan 25 gambar sebagai *viral pneumonia*.
- d) Dari total 134 gambar pada *class viral pneumonia*. Sebanyak 123 gambar di prediksi dengan benar, 3 gambar diprediksi sebagai *lung opacity*, dan 8 gambar sebagai normal.
- 2) Hasil *confusion matrix* pada model *InceptionResnetV2*

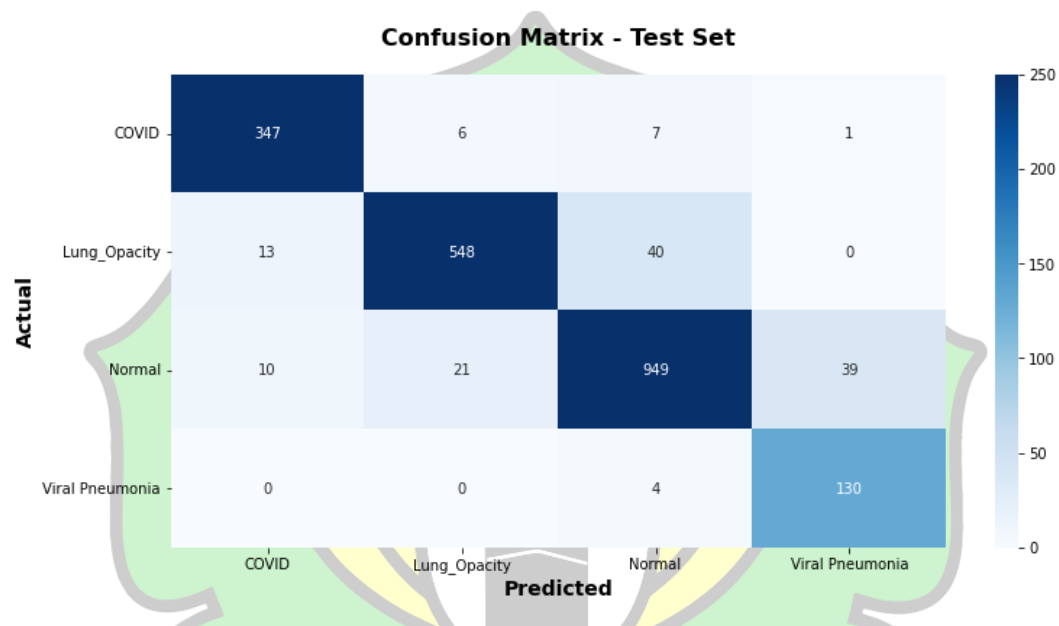


Gambar IV.10 *Confusion matrix* *InceptionResnetV2*

Dari *confusion matrix* model *InceptionResnetV2* yang dihasilkan dari prediksi terhadap data *testing* maka dapat disimpulkan:

- a) Dari total 361 gambar pada *class covid*. Sebanyak 341 gambar di prediksi dengan benar, 12 gambar diprediksi sebagai *lung opacity*, dan 8 gambar diprediksi sebagai normal,
- b) Dari total 601 gambar pada *class lung opacity*. Sebanyak 524 gambar di prediksi dengan benar, 26 gambar diprediksi sebagai *covid*, dan 51 gambar diprediksi sebagai normal.

- c) Dari total 1019 gambar pada *class* normal. Sebanyak 960 gambar di prediksi dengan benar, 17 gambar diprediksi sebagai *covid*, 21 gambar diprediksi sebagai *lung opacity*, dan 21 gambar sebagai *viral pneumonia*.
 - d) Dari total 134 gambar pada *class viral pneumonia*. Sebanyak 128 gambar di prediksi dengan benar dan 6 gambar sebagai normal.
- 3) Hasil *confusion matrix* pada model *Xception*

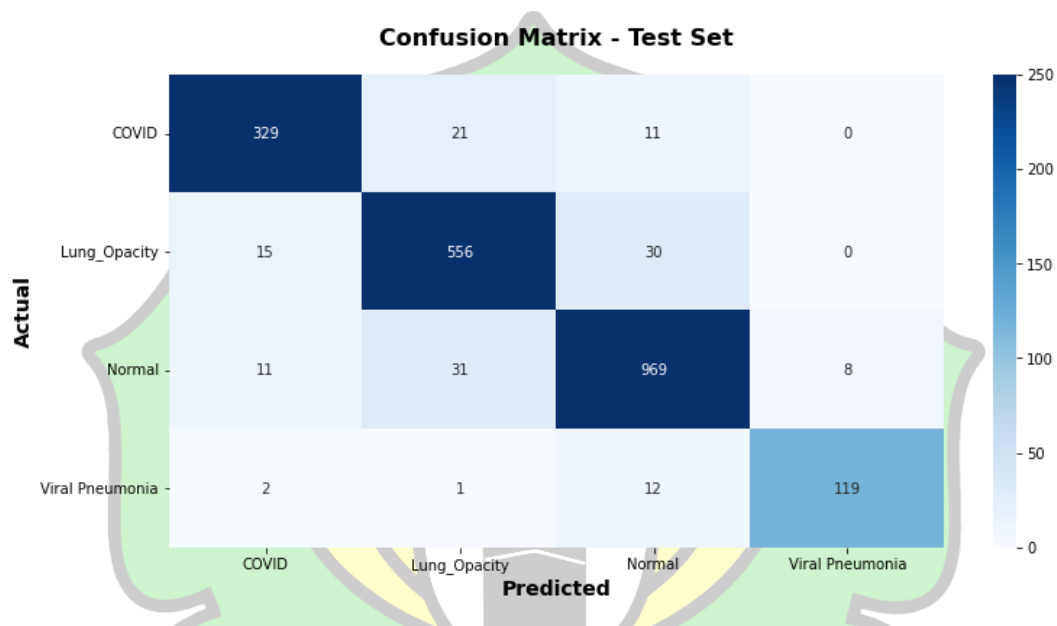


Gambar IV.11 Confusion matrix *Xception*

Dari *confusion matrix* model *Xception* yang dihasilkan dari prediksi terhadap data *testing* maka dapat disimpulkan:

- a) Dari total 361 gambar pada *class covid*. Sebanyak 347 gambar di prediksi dengan benar, 6 gambar diprediksi sebagai *lung opacity*, 7 gambar diprediksi sebagai normal, dan 1 gambar sebagai *viral pneumonia*.
- b) Dari total 601 gambar pada *class lung opacity*. Sebanyak 548 gambar di prediksi dengan benar, 13 gambar diprediksi sebagai *covid*, dan 40 gambar diprediksi sebagai normal.

- c) Dari total 1019 gambar pada *class normal*. Sebanyak 949 gambar di prediksi dengan benar, 10 gambar diprediksi sebagai *covid*, 21 gambar diprediksi sebagai *lung opacity*, dan 39 gambar sebagai *viral pneumonia*.
- d) Dari total 134 gambar pada *class viral pneumonia*. Sebanyak 130 gambar di prediksi dengan benar dan 4 gambar sebagai normal.
- 4) Hasil *confusion matrix* pada model VGG19



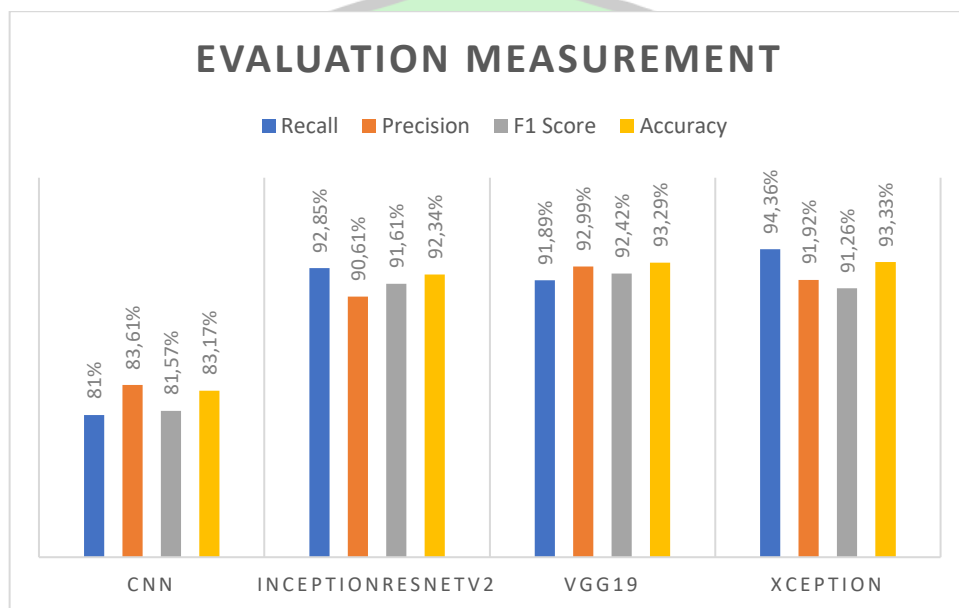
Gambar IV.12 Confusion matrix VGG19

Dari *confusion matrix* model VGG19 yang dihasilkan dari prediksi terhadap data *testing* maka dapat disimpulkan:

- a) Dari total 361 gambar pada *class covid*. Sebanyak 329 gambar di prediksi dengan benar, 21 gambar diprediksi sebagai *lung opacity*, dan 11 gambar diprediksi sebagai normal.
- b) Dari total 601 gambar pada *class lung opacity*. Sebanyak 556 gambar di prediksi dengan benar, 15 gambar diprediksi sebagai *covid*, dan 30 gambar diprediksi sebagai normal.

- c) Dari total 1019 gambar pada *class* normal. Sebanyak 969 gambar di prediksi dengan benar, 11 gambar diprediksi sebagai *covid*, 31 gambar diprediksi sebagai *lung opacity*, dan 8 gambar sebagai *viral pneumonia*.
- d) Dari total 134 gambar pada *class viral pneumonia*. Sebanyak 119 gambar di prediksi dengan benar, 2 gambar diprediksi sebagai *covid*, 1 gambar diprediksi sebagai *lung opacity*, dan 12 gambar sebagai normal.

b. Recall, Precision, F1 Score, Accuracy



Gambar IV.13 Recall, precision, f1 score, dan accuracy setiap model

Berdasarkan *confusion matrix* maka langkah selanjutnya yang dilakukan adalah mencari nilai *Recall*, *Precision*, *F1 Score*, dan *Accuracy*. Nilai *Recall* yang tinggi menunjukkan model dapat mengenali *class* dengan baik. Pada gambar 32 terlihat bahwa model *Xception* mempunyai nilai *recall* tertinggi yaitu 94,36% disusul *InceptionResnetV2* 92,85% *VGG19* 91,89% dan yang terakhir *baseline CNN* dengan nilai *recall* 81%.

Precision adalah nilai yang menunjukkan ketepatan model dalam memprediksi sebuah *class* dengan benar. Pada gambar 32 terlihat bahwa model

VGG19 mempunyai nilai *precision* tertinggi yaitu 92,99% disusul *Xception* 91,92% *InceptionResnetV2* 90,61% dan yang terakhir *baseline CNN* dengan nilai *precision* 83,61%. Sedangkan, *f1 score* merupakan *Harmonic Mean* antara *precision* dan *recall*. Nilai *f1 score* yang tinggi mengindikasikan model memiliki *precision* dan *recall* yang baik. Pada gambar 32 terlihat bahwa model VGG19 mempunyai nilai *f1 score* tertinggi yaitu 92,42% disusul *InceptionResnetV2* 91,61% *Xception* 91,26% dan yang terakhir *baseline CNN* dengan nilai *f1 score* 81,57%.

Selanjutnya adalah nilai *accuracy*. *Accuracy* dapat menunjukkan seberapa akurat model dapat melakukan pendeteksian dengan benar terhadap keseluruhan data. *Xception* mempunyai nilai *accuracy* tertinggi yaitu 93,33% disusul VGG19 93,29% *InceptionResnetV2* 92,34% dan yang terakhir *baseline CNN* dengan nilai *accuracy* 83,17%.

IV.7 Pengaruh Jumlah Dataset dan jumlah Class

Dataset merupakan kumpulan data yang dipelajari oleh model. Dataset yang digunakan pada penelitian ini merupakan kumpulan citra *x-ray* paru-paru yang terdiri dari 4 *class* yaitu *COVID-19* berjumlah 3.616, *lung opacity* 6.012, normal 10.192, dan *pneumonia* yang berjumlah 1.345. Semakin banyak dataset yang digunakan maka semakin banyak informasi yang didapatkan oleh model mengenai data tersebut. Semakin banyak *class* yang digunakan maka model dapat mengenali lebih banyak penyakit infeksi paru yang berbeda. Berikut ini menjelaskan pengaruh jumlah dataset dan jumlah *class* terhadap *accuracy* model yang diperoleh.

a. Pengaruh jumlah dataset dengan 4 class

Pada tabel 8 menunjukkan pengaruh jumlah dataset dan jumlah *class* terhadap *accuracy* yang diperoleh oleh setiap model. Variasi jumlah dataset yang

digunakan adalah 5.000, 10.000, 15.000, dan total keseluruhan data yang ada yaitu 21.165. jumlah *class* yang digunakan adalah 4 yang meliputi *COVID-19*, *lung opacity*, *normal*, dan *pneumonia*. Dari hasil yang ditunjukkan pada tabel 8 terlihat rata-rata model mengalami peningkatan *accuracy* pada saat dataset ditambahkan.

Tabel IV.6 Pengaruh jumlah dataset dengan 4 class

Model	Jumlah Dataset			
	5.000	10.000	15.000	21.165
Baseline CNN	78.60%	82.60%	83.87%	83.17%
VGG19	91.80%	93.00%	91.20%	93.29%
InceptionResnetV2	90.20%	91.30%	91.47%	92.34%
Xception	92.40%	93.80%	93.67%	93.33%

b. Pengaruh jumlah dataset dengan 3 class

Pada tabel 9 menunjukkan pengaruh jumlah dataset dan jumlah *class* terhadap *accuracy* yang diperoleh oleh setiap model. Jumlah *class* yang digunakan adalah 3 yang meliputi *COVID-19*, *normal*, dan *pneumonia*. Variasi jumlah dataset yang digunakan adalah 5.000, 10.000, dan total keseluruhan data yang ada yaitu 15.153. Dari hasil yang ditunjukkan pada tabel 9 terlihat rata-rata model mengalami peningkatan *accuracy* pada saat dataset ditambahkan dan hasil keseluruhan *accuracy* menggunakan 3 *class* lebih besar dari pada menggunakan 4 *class*.

Tabel IV.7 Pengaruh jumlah dataset dengan jumlah 3 class

Model	Jumlah Dataset		
	5.000	10.000	15.000
Baseline CNN	88.28%	89.62%	93.46%
VGG19	95.27%	95.57%	95.97%
InceptionResnetV2	94.90%	95.57%	95.97%
Xception	96.98%	96.13%	96.04%

BAB V

KESIMPULAN DAN SARAN

V.1 Kesimpulan

Pada penelitian ini pendeteksian *COVID-19* pada citra *x-ray* paru-paru dilakukan dengan model *baseline CNN*, *VGG19*, *InceptionResnetV2*, dan *Xception*. Model-model tersebut dilatih menggunakan dataset citra *x-ray* paru-paru yang berjumlah 21.165 dan terdiri dari 4 *class* yaitu *COVID-19*, *viral pneumonia*, *lung opacity* dan normal. Model yang sudah dilatih kemudian di evaluasi untuk mengetahui tingkat keakuratan yang dimiliki oleh setiap model.

Berdasarkan pembahasan dan hasil analisis yang telah dilakukan dalam penelitian ini, maka kesimpulan yang dapat diambil adalah model yang melakukan pendeteksian terbaik adalah *Xception* dengan nilai *accuracy* tertinggi yaitu 93,33% disusul *VGG19* 93,29% *InceptionResnetV2* 92,34% dan yang terakhir *baseline CNN* dengan nilai *accuracy* 83,17%.

V.2 Saran

Dari hasil yang telah diuji dalam penelitian ini, beberapa saran yang mungkin dapat dilakukan untuk penelitian-penelitian selanjutnya adalah sebagai berikut:

1. Menambahkan *class* penyakit infeksi paru lainnya untuk melatih model yang ada agar dapat mengenali lebih banyak jenis penyakit infeksi paru.
2. Menggunakan *pretrained* model lainnya yang tersedia di dalam *keras* untuk melakukan pendeteksian seperti *MobileNet*, *DenseNet*, dan *EfficientNet* serta mengetahui keakuratan model terbaik.

3. Model yang sudah teruji kedepannya harus dapat diimplementasikan untuk melakukan pendeteksian secara *realtime*.



DAFTAR KEPUSTAKAAN

- Apostolopoulos, I. D., & Mpesiana, T. A. (2020). Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, 43(2), 635–640.
<https://doi.org/10.1007/s13246-020-00865-4>
- Arsyam, M. (2019). *Ragam Jenis Penelitian dan Perspektif*.
- Bockenek, J. (2014). Overview of Models. In *Github.io*.
<https://jabocken.github.io/ML2017Fall/docs/models/>
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua*, 1800–1807.
<https://doi.org/10.1109/CVPR.2017.195>
- Chollet, F. (2021). Deep Learning with Python. In *Deep Learning with Python*.
<https://doi.org/10.1007/978-1-4842-5364-9>
- Chouat, I., Echtioui, A., Khemakhem, R., Zouch, W., Ghorbel, M., & Hamida, A. Ben. (2022). COVID-19 detection in CT and CXR images using deep learning models. *Biogerontology*, 23(1), 65–84.
<https://doi.org/10.1007/s10522-021-09946-7>
- Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., Islam, K. R., Khan, M. S., Iqbal, A., Emadi, N. A., Reaz, M. B. I., & Islam, M. T. (2020). Can AI Help in Screening Viral and COVID-19 Pneumonia? *IEEE Access*, 8, 132665–132676.
<https://doi.org/10.1109/ACCESS.2020.3010287>
- covid19.go.id. (2021). *Peta Sebaran | Covid19.go.id*.

Data, H. (2022). Introduction to Deep Learning. In *Deep Learning From Scratch*.

https://ziedhy.github.io/Introduction_Deep_Learning.html

Deng, L., & Yu, D. (2013). Deep learning: Methods and applications.

Foundations and Trends in Signal Processing, 7(3–4), 197–387.

<https://doi.org/10.1561/20000000039>

Dobrev. (2004). Artificial Intelligence. *Binus University*.

Fauzi, M. R., Eosina, P., & Primasari, D. (2020). *DETEKSI CORONAVIRUS*

DISEASE PADA X-RAY DAN CT-SCAN PARU MENGGUNAKAN

CONVOLUTIONAL NEURAL NETWORK. 3(2).

Gozes, O., Frid, M., Greenspan, H., & Patrick, D. (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>

Haykin, S. (2009). Neural Networks and Learning Machines. In *Encyclopedia of*

Bioinformatics and Computational Biology: ABC of Bioinformatics (Vol. 1–

3). Pearson.

<https://www.facebook.com/kdnuggets>. (2016). How Convolutional Neural

Networks Work - KDNuggets. In *KDNuggets*.

<https://www.kdnuggets.com/2016/08/brohrer-convolutional-neural-networks-explanation.html>

Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., Zhang, L., Fan, G., Xu, J.,

Gu, X., Cheng, Z., Yu, T., Xia, J., Wei, Y., Wu, W., Xie, X., Yin, W., Li, H.,

- Liu, M., ... Cao, B. (2020). Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *The Lancet*, 395(10223), 497–506.
[https://doi.org/10.1016/S0140-6736\(20\)30183-5](https://doi.org/10.1016/S0140-6736(20)30183-5)
- Hussain, S., Anwar, S. M., & Majid, M. (2018). Segmentation of glioma tumors in brain using deep convolutional neural network. *Neurocomputing*, 282, 248–261. <https://doi.org/10.1016/j.neucom.2017.12.032>
- Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0192-5>
- Kia, S. M. (2017). *Brain Decoding for Brain Mapping: Definition, Heuristic Quantification, and Improvement of Interpretability in Group MEG Decoding*.
- Kotakode.com | Komunitas Developer Indonesia. (2022). In *Kotakode.com*.
[https://kotakode.com/blogs/2707/Convolutional-Neural-Network-\(CNN\)](https://kotakode.com/blogs/2707/Convolutional-Neural-Network-(CNN))
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks Alex. *NIPS*.
- LeChun, Y., Buttou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proc of the IEEE*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Leonardo, M., Carvalho, T., Rezende, E., Zucchi, R., & Faria, F. (2018). *Deep Feature-Based Classifiers for Fruit Fly Identification (Diptera: Tephritidae)*.
<https://doi.org/10.1109/SIBGRAPI.2018.00012>
- LINA, Q. (2018). Implementasi Deep Learning Menggunakan Convolutional

Neural Network untuk Klasifikasi Gambar (Mata Juling dan Mata Normal) dengan R. In *Medium*. Medium.

<https://medium.com/@16611110/implementasi-deep-learning-menggunakan-convolutional-neural-network-untuk-klasifikasi-gambar-mata-87dcc0ad26e0>

Madiraju, N. (2020). Splitting a dataset - Towards Data Science. In *Medium*.

Towards Data Science. <https://towardsdatascience.com/splitting-a-dataset-e328dab2760a>

Malik, S., Singh, S., Singh, N. M., & Panwar, N. (2021). Diagnosis of COVID-19

Using Chest X-ray. *Ojs.Unikom.Ac.Id*, 2(March 2020), 55–64.

<https://ojs.unikom.ac.id/index.php/injiiscom/article/download/4024/2138>

Mohammad Reza Ebrahimi. (2022). cnn-featurevis-ece421. In

colab.research.google.com.

[https://colab.research.google.com/github/mamaj/cnn-featurevis-](https://colab.research.google.com/github/mamaj/cnn-featurevis-ece421/blob/master/CNNFeatureVis_handout.ipynb#scrollTo=tK5mYbtDt0Vd)

[ece421/blob/master/CNNFeatureVis_handout.ipynb#scrollTo=tK5mYbtDt0Vd](https://colab.research.google.com/github/mamaj/cnn-featurevis-ece421/blob/master/CNNFeatureVis_handout.ipynb#scrollTo=tK5mYbtDt0Vd)

Nasichuddin, M. A. (2017). *Implementasi CNN Untuk Klasifikasi Teks*

Menggunakan Tensorflow. [https://medium.com/@arynas92/implementasi-](https://medium.com/@arynas92/implementasi-cnn-untuk-klasifikasi-teks-menggunakan-tensorflow-3a720cc3afbc)

[cnn-untuk-klasifikasi-teks-menggunakan-tensorflow-3a720cc3afbc](https://medium.com/@arynas92/implementasi-cnn-untuk-klasifikasi-teks-menggunakan-tensorflow-3a720cc3afbc)

ninds.nih.gov. (2021). Brain Basics: The Life and Death of a Neuron | National

Institute of Neurological Disorders and Stroke. In *Nih.gov*.

<https://www.ninds.nih.gov/Disorders/patient-caregiver-education/life-and-death-neuron>

PDPI. (2020). Panduan Praktik Klinis: Pneumonia COVID-19. *Journal of the*

American Pharmacists Association, 55(5), 1–67.

Penjelasan State of the Art Convolutional Neural Networks (CNNs) - DenseNets.

(2020). In *ICHI.PRO*. <https://ichi.pro/id/penjelasan-state-of-the-art-convolutional-neural-networks-cnns-densenets-75969814801645>

Perkovic, L. (2012). *Introduction to Computing Using Python: An Application Development Focus*.

Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Abul Kashem, S. Bin, Islam, M. T., Al Maadeed, S., Zughaier, S. M., Khan, M. S., & Chowdhury, M. E. H. (2021). Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Computers in Biology and Medicine*, 132, 104319.

<https://doi.org/10.1016/J.COMPBIOMED.2021.104319>

Rochman, F., & Junaedi, H. (2020). *IMPLEMENTASI TRANSFER LEARNING UNTUK IDENTIFIKASI ORDO TUMBUHAN MELALUI DAUN*. 1(6), 672–679.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. <https://doi.org/10.1037/h0042519>

Rumelhart, D. E., & McClelland, J. L. (1985). Learning Internal Representations by Error Propagation. *Defense Technical Information Center technical report*.

Samuel, A. (1959). *Machine Vision*. McGraw-Hill.

Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah. *Emitor: Jurnal Teknik Elektro*, 18(01), 15–21. <https://doi.org/10.23917/emitor.v18i01.6235>

- Shapiro, L., & Stockman, G. (2000). *Computer Vision* (Vol. 9).
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.
- Sun, T., Zhou, B., Lai, L., & Pei, J. (2017). Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinformatics*, 18(1), 1–8. <https://doi.org/10.1186/s12859-017-1700-2>
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 4278–4284.
- Who.int. (2021). *WHO Coronavirus (COVID-19) Dashboard*. <https://covid19.who.int/>
- Yuliana. (2020). Corona virus diseases (Covid-19): Sebuah tinjauan literatur. *Wellness And Healthy Magazine*, 2(1), 187–192. <https://doi.org/10.30604/well.95212020>
- Yunus, M. (2020). #3 Machine Learning Evaluation - Muhammad Yunus - Medium. In *Medium*. Medium. <https://yunusmuhammad007.medium.com/3-machine-learning-evaluation-239426e3319e>

LAMPIRAN-LAMPIRAN

Lampiran 1 Code

Module import

```
# Dataset
from google.colab import files

# Preprocessing
from keras.preprocessing.image import ImageDataGenerator

# Model
import tensorflow as tf
from keras import layers
from keras.callbacks import ReduceLROnPlateau, EarlyStopping
from tensorflow.keras.optimizers import Adam

# Evaluate
import numpy as np
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import precision_recall_fscore_support, accuracy_score
from keras.preprocessing import image
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

Prepare Dataset

```
# Connect to kaggle
!pip install kaggle
!mkdir ~/.kaggle

uploaded = files.upload()
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
# Download and extract dataset
!kaggle datasets download -d tawsifurrahman/covid19-radiography-database
!unzip covid19-radiography-database.zip
# Dataset folder restructuring
!mv COVID-19_Radiography_Dataset/COVID/images/ COVID-19_Radiography_Dataset/
!rm -r COVID-19_Radiography_Dataset/COVID/
!mv COVID-19_Radiography_Dataset/images COVID-19_Radiography_Dataset/COVID

!mv COVID-19_Radiography_Dataset/Lung_Opacity/images/ COVID-19_Radiography_Dataset/
!rm -r COVID-19_Radiography_Dataset/Lung_Opacity/
!mv COVID-19_Radiography_Dataset/images COVID-19_Radiography_Dataset/Lung_Opacity
```

```

!mv COVID-19_Radiography_Dataset/Normal/images/ COVID-19_Radiography_Da-
taset/
!rm -r COVID-19_Radiography_Dataset/Normal/
!mv COVID-19_Radiography_Dataset/images COVID-19_Radiography_Dataset/Nor-
mal

!mv COVID-19_Radiography_Dataset/Viral\ Pneumonia/images/ COVID-19_Radi-
ography_Dataset/
!rm -r COVID-19_Radiography_Dataset/Viral\ Pneumonia/
!mv COVID-19_Radiography_Dataset/images COVID-19_Radiography_Dataset/Vi-
ral\ Pneumonia

```

Data Preprocessing

```

#add the path general where the classes subpath are allocated
path = 'COVID-19_Radiography_Dataset'

classes=["COVID", "Lung_Opacity", "Normal", "Viral Pneumonia"]
num_classes = len(classes)
batch_size = 16

##No Augmentation on the Test set Images##
datagen = ImageDataGenerator(rescale=1./255,
                             validation_split=0.1)

#loading the images to training set
train_gen = datagen.flow_from_directory(directory=path,
                                       target_size=(299, 299),
                                       class_mode='categorical',
                                       subset='training',
                                       shuffle=True, classes=clas-
ses,
                                       batch_size=batch_size,
                                       color_mode="rgb")

#loading the images to test set
test_gen = datagen.flow_from_directory(directory=path,
                                       target_size=(299, 299),
                                       class_mode='categorical',
                                       subset='validation',
                                       shuffle=False, clas-
ses=classes,
                                       batch_size=batch_size,
                                       color_mode="rgb")

```

Model

```

# baseline CNN
epochs = 20

```

```

## Defining Cnn
model = tf.keras.models.Sequential([
    layers.BatchNormalization(),
    layers.Conv2D(32, 3, activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.3),
    layers.Conv2D(128, 3, activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.2),
    layers.Flatten(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.15),
    layers.Dense(4, activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# InceptionResnetV2
learning_rate = 1e-5
epochs = 20

inception_resnet_v2_layer = tf.keras.applications.InceptionResNetV2(input_shape = (299, 299, 3),

hts = 'imagenet',

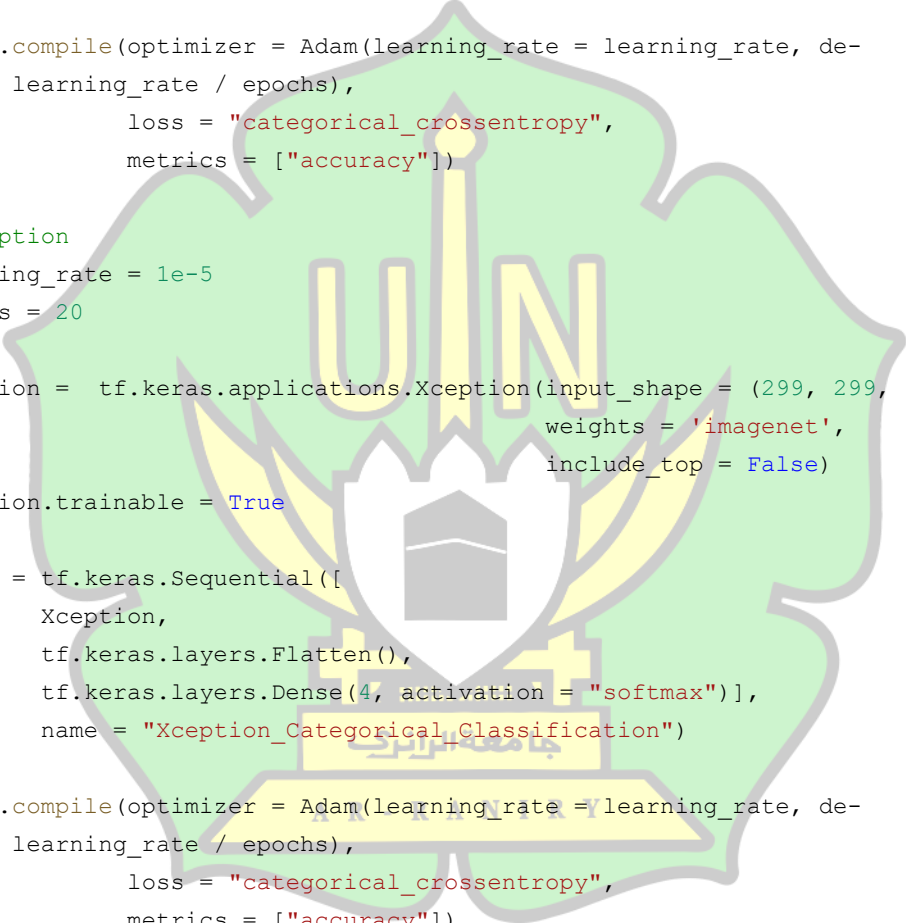
clude_top = False)
inception_resnet_v2_layer.trainable = True

model = tf.keras.Sequential([
    inception_resnet_v2_layer,
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(4, activation = "softmax")],
    name = "InceptionResNetV2_Categorical_Classification")

model.compile(optimizer = Adam(learning_rate = learning_rate, decay = learning_rate / epochs),
    loss = "categorical_crossentropy",
    metrics = ["accuracy"])

# VGG19
learning_rate = 1e-5
epochs = 20

```



```

VGG19 = tf.keras.applications.VGG19(input_shape = (299, 299, 3),
                                     weights = 'imagenet',
                                     include_top = False)

VGG19.trainable = True

model = tf.keras.Sequential([
    VGG19,
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(4, activation = "softmax")],
    name = "VGG19_Categorical_Classification")

model.compile(optimizer = Adam(learning_rate = learning_rate, de-
cay = learning_rate / epochs),
              loss = "categorical_crossentropy",
              metrics = ["accuracy"])

# Xception
learning_rate = 1e-5
epochs = 20

Xception = tf.keras.applications.Xception(input_shape = (299, 299, 3),
                                           weights = 'imagenet',
                                           include_top = False)

Xception.trainable = True

model = tf.keras.Sequential([
    Xception,
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(4, activation = "softmax")],
    name = "Xception_Categorical_Classification")

model.compile(optimizer = Adam(learning_rate = learning_rate, de-
cay = learning_rate / epochs),
              loss = "categorical_crossentropy",
              metrics = ["accuracy"])

```

Training and Testing

```

# fit model
history=model.fit(train_gen,
                  validation_data= test_gen,
                  steps_per_epoch=100,
                  batch_size=batch_size,
                  epochs= epochs)

#evaluate model
model.evaluate(test_gen)

```

Hasil

```
# prediction
y_pred = model.predict(test_gen)

# confusion matrix
# transform the predictions into array such as [0,0,1,2...]
predictions = np.array(list(map(lambda x: np.argmax(x), y_pred)))
#Retrieve the True classes of the test set
y_true=test_gen.classes

# Build Confusion Matrix
CMatrix = pd.DataFrame(confusion_matrix(y_true, predictions), columns=classes, index =classes)

plt.figure(figsize=(12, 6))
ax = sns.heatmap(CMatrix, annot = True, fmt = 'g', vmin = 0, vmax = 250, cmap = 'Blues')
ax.set_xlabel('Predicted', fontsize = 14, weight = 'bold')
ax.set_xticklabels(ax.get_xticklabels(), rotation =0);

ax.set_ylabel('Actual', fontsize = 14, weight = 'bold')
ax.set_yticklabels(ax.get_yticklabels(), rotation =0);
ax.set_title('Confusion Matrix - Test Set', font-size = 16, weight = 'bold', pad=20);

#Accuracy Result
acc = accuracy_score(y_true, predictions)
#Precision, Recall and F-Score (For the whole dataset)
results_all = precision_recall_fscore_support(y_true, predictions, average='macro', zero_division = 1)
#Precision, Recall and F-Score (For each Class)
results_class = precision_recall_fscore_support(y_true, predictions, average=None, zero_division = 1)

#Organise the Results into a Dataframe
metric_columns = ['Precision', 'Recall', 'F-Score', 'S']
all_df = pd.concat([pd.DataFrame(list(results_class)).T, pd.DataFrame(list(results_all)).T])
all_df.columns = metric_columns
all_df.index = ['COVID', 'Lung Opacity', 'Normal', 'Viral Pneumonia', 'Total']

print('**Overall Results**')
print('Accuracy Result: %.2f%%'%(acc*100)) #Accuracy of the whole Dataset
print('Precision Result: %.2f%%'%(all_df.iloc[4,0]*100))#Precision of the whole Dataset
print('Recall Result: %.2f%%'%(all_df.iloc[4,1]*100))#Recall of the whole Dataset
```

```
print('F-Score Result: %.2f%%'%(all_df.iloc[4,2]*100))#F-  
Score of the whole Dataset
```



RIWAYAT PENULIS



Aulia Sabri, lahir 30 April 2000 di Meulaboh, Aceh Barat. Anak keenam dari enam bersaudara. Menyelesaikan sekolah tingkat menengah atas pada SMK Negeri 4 Meulaboh pada jurusan Teknik Survey dan Pemetaan. Melanjutkan studi S1 di Universitas Islam Negeri Ar-Raniry Banda Aceh pada jurusan Teknologi Informasi dengan bidang minat Sistem Web dan *Mobile*. Di samping pendidikan formal, juga pernah mengikuti beberapa program sertifikasi pada bidang *Software Development* dan *Data Science*.

