

Implementasi Arsitektur Xception Untuk Klasifikasi Citra Covid-19 Radiography

Mohd. Abdul Ghani¹, Fathur Fahrizal², Armin Lawi³

^{1,2,3}Program Studi Sistem Informasi, Departemen Matematika, Universitas Hasanuddin

Email : 1ghanima18h@student.unhas.ac.id, 2fahrizalf18h@student.unhas.ac.id, 3armin@unhas.ac.id

Abstract — Covid-19 is a viral disease that causes serious pneumonia and affects various parts of our body from mild to severe depending on the patient's immune system. This infection was first reported in Wuhan City, China in December 2019, and after that, it became a global pandemic that spread rapidly across the globe. Preventing the spread of the virus requires early detection of positive cases and to treat infected patients as quickly as possible. Based on these problems, this study was conducted to classify radiographic images whether infected with Covid-19 or not using the Xception architecture with transfer learning. Research results are presented with data visualization, confusion matrix, ROC curve, and other reports. The dataset used is the Covid-19 Radiography Database with 4 classes, namely COVID, Lung Opacity, Normal, and Viral Pneumonia. This classification is carried out to help early detection of positive cases, with the initial method of pre-processing data such as converting images into arrays, labeling image data and normalizing data, then training and testing processes are carried out with a comparison of 70% data to train model development and 30% data evaluate the performance of the model. The experimental results obtained the best training accuracy of 92% and the validation accuracy of 92% (best-fitting) with an f1-score of 25%. The details of the results for each class are as follows: Viral Pneumonia with 25% precision and 23% recall, Lung Opacity with 24% precision and 26% recall, Normal with 28% precision and 30% recall, Covid with 24% precision and 21 recall. %. And for the ROC AUC score of 50%.

Keyword — Convolution Neural Network (CNN), Covid-19 Radiography Database, Deep Learning, Pre-Processing, Xception.

Abstrak — Covid-19 adalah penyakit virus yang menyebabkan pneumonia serius dan berdampak pada berbagai bagian tubuh kita dari ringan hingga berat tergantung pada kekebalan tubuh pasien sistem. Infeksi ini pertama kali dilaporkan di Kota Wuhan, China pada Desember 2019, dan setelah itu, menjadi pandemi global yang menyebar dengan cepat di seluruh dunia. Mencegah penyebaran virus membutuhkan deteksi dini kasus positif dan untuk mengobati pasien yang terinfeksi secepat mungkin. Berdasarkan permasalahan tersebut, penelitian ini dilakukan untuk mengklasifikasi citra radiografi apakah terinfeksi Covid-19 atau tidak menggunakan arsitektur Xception dengan transfer learning. Hasil penelitian disajikan dengan visualisasi data, matriks konfusi, kurva ROC, dan laporan lainnya. Dataset yang digunakan ialah Covid-19 Radiography Database dengan 4 class yaitu COVID, Lung Opacity, Normal, dan Viral Pneumonia. Klasifikasi ini dilakukan untuk membantu deteksi dini kasus positif, dengan metode awal dilakukan pra-proses data seperti mengkonversi citra menjadi array, pelabelan data citra dan normalisasi data, kemudian dilakukan proses training dan testing dengan perbandingan 70% data untuk melatih pembangunan model dan 30% data mengevaluasi kinerja model. Hasil percobaan diperoleh akurasi pelatihan terbaik 92% dan akurasi validasinya sebesar 92% (best-

fitting) dengan f1-score 25%. Rincian hasil untuk masing-masing kelas adalah sebagai berikut: ViralPneumonia dengan precision 25% dan recall 23%, Lung Opacity dengan precision 24% dan recall 26%, Normal dengan precision 28% dan recall 30%, Covid dengan precision 24% dan recall 21%. Dan untuk skor ROC AUC sebesar 50%.

Kata kunci—Convolution Neural Network (CNN), Covid-19 Radiography Database, Deep Learning, Pre-Processing, Xception.

I. PENDAHULUAN

Covid-19 atau biasa disebut coronavirus merupakan keluarga besar virus yang mengakibatkan terjadinya infeksi saluran pernapasan. Penyakit ini menjadi sorotan karena kemunculannya di akhir tahun 2019 pertama kali di Wuhan, China. Lokasi kemunculan pertama kali ini, membuat coronavirus juga dikenal dengan sebutan Wuhan virus. Selain China, coronavirus juga menyebar secara cepat ke berbagai negara lain, termasuk Jepang, Thailand, Korea Selatan, Amerika Serikat, bahkan hingga sampai ke Indonesia [1].

Penyebab Corona virus merupakan virus singlestranded RNA yang berasal dari kelompok Coronaviridae. Dinamakan coronavirus karena permukaannya yang berbentuk seperti mahkota. Virus corona umumnya ditemukan pada hewan seperti ular, hewan ternak, dan kelelawar. Manusia dapat tertular virus apabila terdapat riwayat kontak dengan hewan tersebut. Namun, adanya ledakan jumlah kasus di Wuhan, China menunjukkan bahwa corona virus dapat ditularkan dari manusia ke manusia. Virus bisa ditularkan lewat droplet, yaitu partikel air yang berukuran sangat kecil dan biasanya keluar saat batuk atau bersin. Apabila droplet tersebut terhirup atau mengenai lapisan kornea mata, seseorang beresiko untuk tertular penyakit ini [1].

Penyebaran virus 2019-nCoV sangat cepat hingga ke 204 negara di dunia pada akhirnya, bulan Maret 2020 World Health Organization (WHO) menetapkan penyakit ini sebagai pandemi global dan menyebut virus corona ini dengan sebutan Covid-19 (Corona Virus Disease tahun 2019) [2]. Berdasarkan data dari Worldometers per 31 Maret 2020 menunjukkan bahwa dari 204 negara dan 2 kapal pesiar sebanyak 801.117 kasus terkonfirmasi positif, sebanyak 38.771 kasus meninggal dunia, dan sebanyak 172.319 kasus berhasil sembuh [3].

Mencegah penyebaran virus membutuhkan deteksi dini kasus positif dan untuk mengobati pasien yang terinfeksi

secepat mungkin. Berdasarkan permasalahan tersebut, penelitian ini dilakukan untuk mengklasifikasi citra radiografi apakah terinfeksi Covid-19 atau tidak menggunakan arsitektur Xception dengan transfer learning.

II. PENELITIAN TERKAIT

A. Classification of Covid-19 in chest X-ray images using DeTraC deep convolution neural network

Chest X-ray adalah teknik pencitraan pertama yang berperan penting dalam diagnosis penyakit COVID-19. Karena tingginya ketersediaan kumpulan data citrabernotasi skala besar, kesuksesan besar telah dicapai dengan menggunakan jaringan neural konvolusional (CNN) untuk pengenalan dan klasifikasi citra. Namun, karena terbatasnya ketersediaan citra medis bernotasi, klasifikasi citra medis tetap menjadi tantangan terbesar dalam diagnosis medis. Berkat pembelajaran transfer, sebuah mekanisme efektif yang dapat memberikan solusi yang menjanjikan dengan mentransfer pengetahuan dari tugas pengenalan objek generik ke tugas khusus domain. Dalam makalah ini, kami memvalidasi dan deep CNN, yang disebut *Decompose, Transfer, and Compose* (DeTraC), untuk klasifikasi citra rontgen dada COVID-19. DeTraC dapat menangani tidak aturan dalam kumpulan data citra dengan menyelidiki batas kelasnya menggunakan mekanisme dekomposisi kelas. Hasil percobaan menunjukkan kemampuan DeTraC dalam mendeteksi kasus COVID-19 dari kumpulan data komprehensif yang dikumpulkan dari beberapa rumah sakit sekitar Dunia. Akurasi tinggi 93,1% (dengan sensitivitas 100%) dicapai oleh DeTraC dalam mendeteksi COVID-19 Citra sinar-X dari kasus normal, dan sindrom pernapasan akut berat.

B. Classification of Covid-19 chest X-rays with deep learning: new models or fine tuning

Data rontgen dada terbukti sangat menjanjikan untuk menilai pasien COVID-19, terutama untuk mengatasi kelebihan kapasitas gawat darurat dan pusat perawatan darurat. Metode deep learning (DL) dalam kecerdasan buatan (AI) memainkan peran dominan sebagai pengklasifikasi berkinerja tinggi dalam mendeteksi penyakit menggunakan rontgen dada. Mengingat banyak model DL baru yang telah dikembangkan untuk tujuan ini, tujuan dari penelitian ini adalah untuk menyelidiki fine tuning jaringan saraf konvolusional (CNN) pra-pelatihan untuk klasifikasi COVID-19 menggunakan sinar-X dada. Jika CNN terlatih yang disetel dengan baik dapat memberikan hasil klasifikasi yang setara atau lebih baik daripada CNN lain yang lebih canggih, penerapan alat berbasis AI untuk mendeteksi COVID-19 menggunakan data rontgen dada dapat lebih cepat dan hemat biaya.

C. Literature Review Artificial Intelligence Deeteksi Hasil Ctscan Paru-Paru Pasien Terjangkit Covid-19

Penyakit Virus Corona 19 (COVID-19) merupakan penyakit menular yang disebabkan oleh virus corona yang baru-baru ini ditemukan. Saat ini COVID-19 menjadi fenomena permasalahan untuk kita semua namun hingga sekarang belum ada obat yang ditemukan ampuh dalam mengatasinya. Persoalan lain adalah pada proses pendeteksian orang terjangkit. Hasil pendeteksian Covid-19 menggunakan PCR Swap masih dianggap sangat lambat dan menggunakan Rapid Tes bahkan dianggap kurang meyakinkan dengan melihat beberapa kasus yang ada sebelumnya. Tujuan penelitian ini untuk pendeteksian orang terjangkit COVID-19 lebih cepat dengan tingkat akurasi yang tinggi menggunakan metode Artificial Intelligence yang lebih khusus menggunakan Deep Learning arsitektur *Convolutional Neural Network* (CNN). Metode penelitian yang digunakan adalah literatur review, dimana artikel dikumpulkan dan diproses menggunakan aplikasi mendeley, kriteria artikel yang digunakan adalah yang diterbitkan tahun 2020 yang berkaitan dengan penanganan COVID-19 khususnya yang memanfaatkan Artificial Intelligence dalam pembahasannya. Dengan mengumpulkan dan membahas beberapa penelitian yang ada maka dapat dikatakan bahwa dengan menggunakan Artificial Intelligence sistem dapat mendeteksi terjangkitnya seseorang melalui analisa pola yang ada pada hasil CT Scan Paru dengan memanfaatkan tingkat akurasi data latih yang ada.

D. Xception: Deep Learning dengan Depthwise Separable Convolutions

Kami menyajikan interpretasi modul Inception dalam jaringan saraf konvolusional sebagai langkah perantara di antara konvolusi reguler dan operasi konvolusi terpisah yang dapat dipisahkan secara mendalam (konvolusi mendalam diikuti oleh konvolusi pointwise). Dalam hal ini, konvolusi yang dapat dipisahkan secara mendalam dapat dipahami sebagai modul awal dengan jumlah menara yang sangat banyak. Pengamatan ini mengarahkan kami untuk mengusulkan arsitektur jaringan neural konvolusional mendalam yang baru yang terinspirasi oleh Inception, di mana modul Inception telah diganti dengan konvolusi yang dapat dipisahkan secara mendalam. Kami menunjukkan bahwa arsitektur ini, dijuluki Xception, sedikit mengungguli Inception V3 pada dataset ImageNet (yang dirancang untuk Inception V3 pada dataset klasifikasi citra yang lebih besar yang terdiri dari 350 juta gambar dan 17.000 kelas. Karena arsitektur Xception memiliki jumlah parameter yang sama dengan Inception V3, perolehan kinerja bukan karena peningkatan kapasitas, melainkan karena penggunaan parameter model yang lebih efisien.

III. BAHAN DAN METODE

A. Dataset

Data Covid-19 Radiography Database dapat diakses melalui Repositori Kaggle.

<https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>.

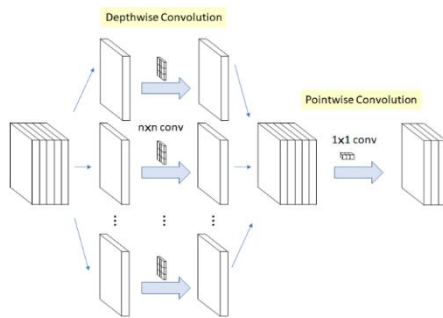
Dataset ini berukuran 740 MB dan memiliki 4 class, yaitu class COVID, Lung Opacity, Normal, dan Viral Pneumonia, untuk lebih detailnya dapat dilihat pada Tabel 3.1.

Tabel 1
DESKRIPSI DATA

No.	Label/Class	Data Citra
1	COVID	3616
2	Lung Opacity	6012
3	Normal	10002
4	Viral Pneumonia	1345
Total		20975

B. Arsitektur Xception

i. Original Depthwise Separable Convolution



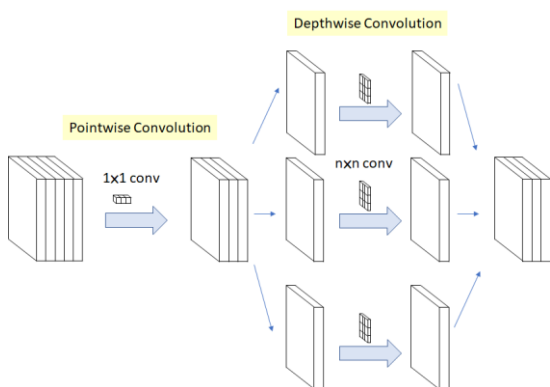
Gambar 1. Original Depthwise Convolution

Original Depthwise convolution adalah konvolusi kedalaman yang diikuti oleh konvolusi pointwise.

1. Depthwise convolution adalah konvolusi spasial n x n saluran bijaksana. Misal pada gambar di atas, kita memiliki 5 saluran, maka kita akan memiliki konvolusi spasial 5 n x n.
2. Pointwise convolution adalah lilitan 1 x 1 untuk mengubah dimensi.

Dibandingkan dengan konvolusi konvensional, kita tidak perlu melakukan konvolusi di semua saluran. Artinya jumlah koneksinya lebih sedikit dan modelnya lebih ringan.

ii. Modified Depthwise Separable Convolution in Xception

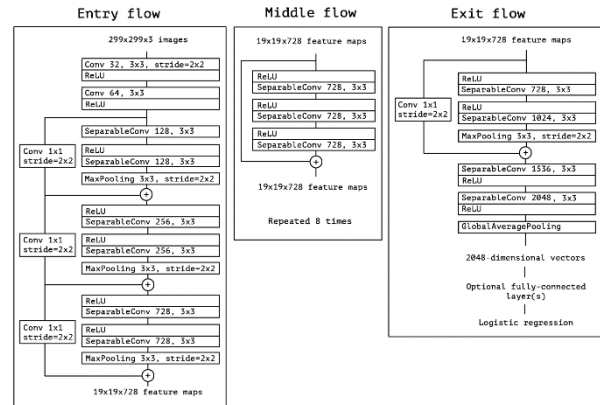


Gambar 2. Modified Depthwise Separable Convolution in Xception

Konvolusi pemisah kedalaman yang dimodifikasi adalah pointwise convolution yang diikuti oleh depthwise convolution. Modifikasi ini dimotivasi oleh modul awal dalam Inception-v3 bahwa konvolusi 1 x 1 dilakukan terlebih dahulu sebelum konvolusi spasial n x n. jadi, ini sedikit berbeda dari yang asli. (n = 3 di sini karena konvolusi spasial 3 x 3 digunakan dalam Inception-v3).

Dua perbedaan kecil:

1. Urutan operasi: seperti disebutkan, konvolusi pemisah kedalaman yang asli seperti yang biasanya diterapkan (misalnya dalam TensorFlow) melakukan konvolusi spasial bijaksana saluran pertama dan kemudian melakukan konvolusi 1 x 1 sedangkan konvolusi pemisah kedalaman yang dimodifikasi 1 x 1 terlebih dahulu kemudian saluran konvolusi spasial yang bijaksana. Ini diklaim tidak penting karena ketika digunakan dalam pengaturan bertumpuk, hanya ada perbedaan kecil yang muncul di awal dan di akhir semua modul awal yang dirantai.
2. Ada/tidaknya Non-Linearitas: dalam InceptionModule yang asli, ada non-linearitas setelah operasi pertama. Dalam Xception, konvolusi pemisah kedalaman yang dimodifikasi, tidak ada IntermediateReLU non-linearity.



Gambar 3. Arsitektur Xception

D. Confusion Matrix

Confusion matrix adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep data mining atau Sistem Pendukung Keputusan. Pada pengukuran kinerja menggunakan confusionmatrix, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah TruePositive (TP), TrueNegative (TN), FalsePositive (FP) dan FalseNegative (FN).

- TrueNegative (TN) merupakan jumlah data negatif yang terdeteksi dengan benar.

- FalsePositive (FP) merupakan data negatif namun terdeteksi sebagai data positif.
- TruePositive (TP) merupakan data positif yang terdeteksi benar.
- FalseNegative (FN) merupakan kebalikan dari TruePositive, sehingga data positif, namun terdeteksi sebagai data negatif.

Tabel 2
CONFUSION MATRIX

		True Values	
		True	False
Prediction	True	TP Correct result	FP Unexpected result
	False	FN Missing result	TN Correct absence of result

i. Presisi

Presisi adalah data yang diambil berdasarkan informasi yang kurang. Dalam klasifikasi biner, presisi dapat dibuat sama dengan nilai prediksi positif.

$$\text{Precision} = (\text{TP} / (\text{TP} + \text{FP})) * 100\% \quad (1)$$

ii. Recall

Recall adalah data penghapusan yang berhasil diambil dari data yang relevan dengan kueri. Dalam klasifikasi biner, recall dikenal sebagai sensitivitas. Munculnya data relevan yang diambil adalah menyetujui dengan query dapat dilihat dengan recall. Berikut ini adalah peran recall.

$$\text{Recall} = (\text{TP} / (\text{TP} + \text{FN})) * 100\% \quad (2)$$

iii. Akurasi dan Validasi Akurasi

Akurasi digunakan untuk mengukur kinerja algoritma dengan cara yang dapat ditafsirkan. Akurasi suatu model biasanya ditentukan setelah parameter model dan dihitung dalam bentuk persentase. Ini adalah ukuran seberapa akurat prediksi model dibandingkan dengan data sebenarnya dan Akurasi (acc) berada pada train. Berikut ini adalah aturan akurasi.

$$\text{Akurasi} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) * 100\% \quad (3)$$

Sedangkan validasi acc ada di data validasi. Yang terbaik adalah mengandalkan val_acc untuk representasi yang adil dari kinerja model karena neural network yang baik pada akhirnya akan menyesuaikan data train pada 100%, tetapi hal ini akan dapat berkinerja buruk pada data yang tidak terlihat.

iv. Loss dan Validasi Loss

Lossfunction digunakan untuk mengoptimalkan algoritma MachineLearning. Lossfunction dihitung berdasarkan training data dan validasi dataserta interpretasinya didasarkan pada seberapa baik kinerja model dalam dua set ini. Ini adalah jumlah kesalahan yang dibuat untuk setiap contoh dalam set training atau validasi. Nilai loss menyiratkan seberapa buruk atau baiknya suatu model berperilaku setelah setiap iterasi optimasi.

Validation loss adalah metrik yang sama dengan Training Loss, tetapi tidak digunakan untuk memperbarui bobot. Ini dihitung dengan cara yang sama - dengan menjalankan jaringan maju melalui input xi dan membandingkan output jaringan yi dengan nilai-nilai kebenaran dasar yi menggunakan loss function.

$$J = \frac{1}{N} \sum_{i=1}^N l(\hat{y}_i, y_i) \quad (4)$$

IV. IMPLEMENTASI

A. Lingkungan Komputasi Eksperimen

Dalam melakukan pengimplementasian arsitektur Xception ini, digunakan jenis perangkat laptop dengan spesifikasi Lenovo IdeaPad S340-14API, dengan processor AMD Ryzen 3 3200U, Radeon Vega Mobile Gfx 2.60 GHz, Ram 8 Gb, disertai dengan Windows 10 Home Single Language (20H2, Build 19042.928).

Selain dari jenis perangkat keras yang digunakan, dibutuhkan juga perangkat lunak untuk menguji dataset Covid-19 Radiography database, yang dalam hal ini menggunakan Google Colab atau Google Collaboratory.

B. Alur Kerja Implementasi

Berikut merupakan flowchart atau alur kerja dari penelitian ini.



Gambar 4. Flowchart pengujian Arsitektur Xception

C. Implementasi pada Google Colab

i. Hasil Pre-Processing

Untuk melakukan Pre-Processing data, langkah pertama yang dilakukan ialah Importlibrary yang dibutuhkan dalam pengolahan dataset Covid-19.

```

# Data preprocessing
import matplotlib.image as mpimg
import glob
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
from os import listdir
import cv2
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer

#Deep Learning
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator,load_img, img_to_array
from keras.models import Sequential, Model
from keras.layers import Conv2D, MaxPooling2D,GlobalAveragePooling2D, SeparableConv2D
from keras.layers import Conv3D, MaxPooling3D,GlobalAveragePooling3D, Add
from keras.layers import Activation, Dropout, BatchNormalization, Flatten
from keras.layers import Dense, AvgPool2D,MaxPool2D, Input, ReLU, GlobalAvgPool2D
from keras.optimizers import Adam, SGD, RMSprop
import tensorflow as tf
from tensorflow.keras.applications import EfficientNetB0
from tensorflow.keras.preprocessing import image
from keras.utils.vis_utils import plot_model

from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc
    
```

Gambar 5. Import Library

Selanjutnya upload dataset yang dibutuhkan kemudian menentukan nilai epoch dan batch_sizesnya agar dapat melakukan trainingdataset untuk mwngupload sebuah model, yaitu dataset Covid-19 Radiography Database.

```

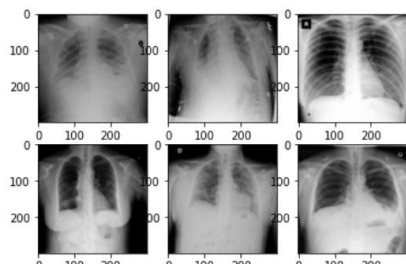
[2] directory_root = "/content/drive/MyDrive/COVID-19_Radiography"
print(os.listdir(directory_root))

['Viral Pneumonia', 'Lung Opacity', 'Normal', 'COVID']

[4] img_list = glob.glob(os.path.join(directory_root, '*/*.png'))

for i, img_path in enumerate(random.sample(img_list, 6)):
    img = image.load_img(img_path, target_size=(299, 299))
    img = image.img_to_array(img, dtype=np.uint8)

    plt.subplot(2, 3, i+1)
    plt.imshow(img.squeeze())
    
```



Gambar 6. UploadDataset dari Drive ke Colab

Selanjutnya melakukan normalisasi pada data gambar.

```

np_image_list = np.array(image_list, dtype=np.float16) / 225.0
    
```

Gambar 7. Normalisasi Data

Kemudian split data dengan rasio 7:3 untuk data training sebanyak 70% dan data testing sebanyak 30%.

```

print("[INFO] Splitting data to train, test, and validate")
x_train, x_test, y_train, y_test = train_test_split(np_image_list, image_labels, test_size=0.3, random_state = 42)

[INFO] Splitting data to train, test, and validate

[5] train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.1,
    zoom_range=0.1,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=True,
    validation_split=0.1
)

test_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.1
)

train_generator = train_datagen.flow_from_directory(
    directory_root,
    target_size=(299, 299),
    batch_size=16,
    class_mode='categorical',
    subset='training',
    seed=0
)

validation_generator = test_datagen.flow_from_directory(
    directory_root,
    target_size=(299, 299),
    batch_size=16,
    class_mode='categorical',
    subset='validation',
    seed=0
)
    
```

Found 3922 images belonging to 4 classes.
Found 435 images belonging to 4 classes.

Gambar 8. Proses split data

```

[8] IMG_SHAPE = (299,299,3)
base_model = Xception(input_shape=IMG_SHAPE, include_top=False, weights='imagenet')

[9] base_model.trainable = False

[10] from tensorflow import keras
model = tf.keras.Sequential([
    base_model,
    keras.layers.GlobalAveragePooling2D(),
    keras.layers.Dense(4, activation='sigmoid')
])

[14] base_model.trainable = True
# Let's take a look to see how many layers are in the base model
print("Number of layers in the base model: ", len(base_model.layers))

# Fine tune from this layer onwards
fine_tune_at = 100

# Freeze all the layers before the `fine_tune_at` layer
for layer in base_model.layers[:fine_tune_at]:
    layer.trainable = False

Number of layers in the base model: 132

[15] model.compile(loss='binary_crossentropy',
    optimizer = tf.keras.optimizers.RMSprop(lr=2e-5),
    metrics=['accuracy'])
model.summary()
    
```

Gambar 9. Proses Compile Model Xception Transfer Learning

ii. Hasil Proses Klasifikasi

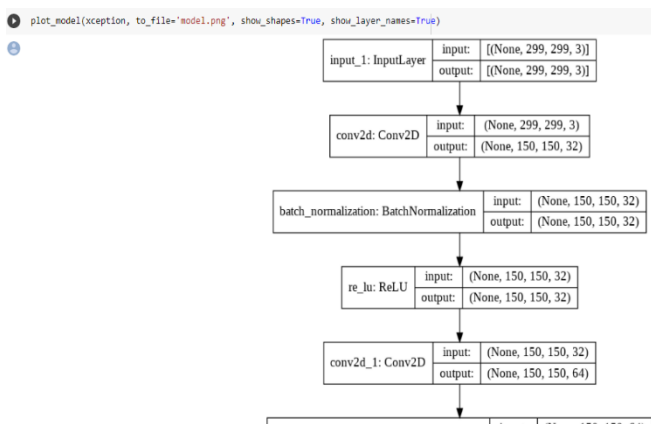
Membangun model menggunakan arsitektur Xception, summary dari model yang dibangun seperti berikut.

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
xception (Functional)	(None, 10, 10, 2048)	20861480
global_average_pooling2d (Gl	(None, 2048)	0
dense (Dense)	(None, 4)	8196

Total params: 20,869,676
 Trainable params: 9,486,540
 Non-trainable params: 11,383,136

Gambar 10. Summary dari model arsitektur Xception



Gambar 11. Plot Model Xception

iii. Evaluasi Hasil

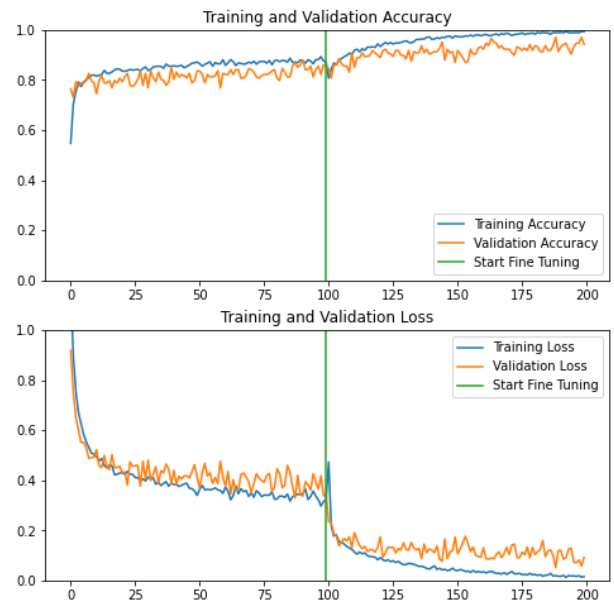
Setelah melakukan model dengan arsitektur Xception, maka diperoleh history dari hasil fitting sebanyak 100 epoch.

```
history_fit = model.fit_generator(train_generator,
                                steps_per_epoch = steps_per_epoch,
                                epochs=epochs,
                                workers=4,
                                validation_data=validation_generator,
                                validation_steps=validation_steps)
```

```
Epoch 20/100: 41s 320ms/step - loss: 0.8669 - accuracy: 0.9582 - val_loss: 0.1460 - val_accuracy: 0.9087
Epoch 30/100: 40s 318ms/step - loss: 0.8691 - accuracy: 0.9583 - val_loss: 0.1104 - val_accuracy: 0.9321
Epoch 40/100: 40s 315ms/step - loss: 0.8410 - accuracy: 0.9764 - val_loss: 0.1255 - val_accuracy: 0.9458
Epoch 41/100: 40s 321ms/step - loss: 0.8023 - accuracy: 0.9952 - val_loss: 0.1211 - val_accuracy: 0.9507
Epoch 42/100: 40s 327ms/step - loss: 0.8844 - accuracy: 0.9827 - val_loss: 0.1108 - val_accuracy: 0.9539
Epoch 43/100: 40s 319ms/step - loss: 0.8395 - accuracy: 0.9735 - val_loss: 0.1376 - val_accuracy: 0.9535
Epoch 44/100: 40s 315ms/step - loss: 0.8576 - accuracy: 0.9638 - val_loss: 0.0953 - val_accuracy: 0.9579
Epoch 45/100: 40s 317ms/step - loss: 0.8435 - accuracy: 0.9683 - val_loss: 0.0758 - val_accuracy: 0.9519
Epoch 46/100: 40s 318ms/step - loss: 0.8438 - accuracy: 0.9683 - val_loss: 0.1660 - val_accuracy: 0.9768
Epoch 47/100: 40s 319ms/step - loss: 0.8395 - accuracy: 0.9762 - val_loss: 0.0960 - val_accuracy: 0.9231
Epoch 48/100: 41s 321ms/step - loss: 0.8420 - accuracy: 0.9703 - val_loss: 0.1127 - val_accuracy: 0.9507
Epoch 49/100: 40s 315ms/step - loss: 0.8689 - accuracy: 0.9329 - val_loss: 0.1189 - val_accuracy: 0.9336
Epoch 50/100: 39s 323ms/step - loss: 0.8162 - accuracy: 0.9959 - val_loss: 0.1288 - val_accuracy: 0.9078
Epoch 51/100: 40s 314ms/step - loss: 0.8177 - accuracy: 0.9853 - val_loss: 0.0734 - val_accuracy: 0.9307
Epoch 52/100: 39s 310ms/step - loss: 0.8172 - accuracy: 0.9902 - val_loss: 0.0702 - val_accuracy: 0.9471
Epoch 53/100: 39s 308ms/step - loss: 0.8186 - accuracy: 0.9808 - val_loss: 0.0818 - val_accuracy: 0.9471
Epoch 54/100: 39s 309ms/step - loss: 0.8122 - accuracy: 0.9930 - val_loss: 0.0578 - val_accuracy: 0.9712
Epoch 55/100: 39s 313ms/step - loss: 0.8135 - accuracy: 0.9944 - val_loss: 0.0617 - val_accuracy: 0.9423
```

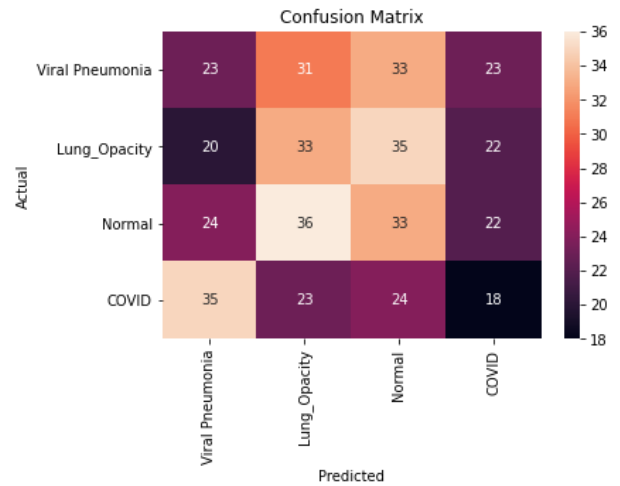
Gambar 12. Proses Fitting pada Model

Kemudian diperoleh hasil visualisasi accuracy dan loss dari hasil fitting sebagai berikut.



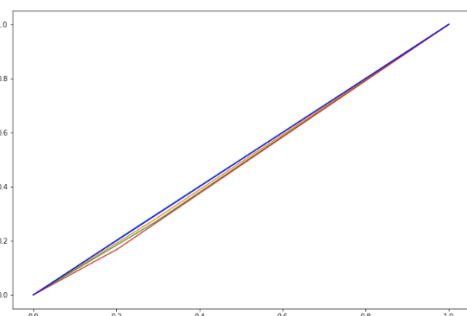
Gambar 13. Visualisasi accuracy dan loss dari tiap epoch

Kemudian diperoleh hasil klasifikasi pada gambar Confusion Matrix seperti berikut.



Gambar 14. Confusion Matrix

Kemudian, divisualisasikan hasil klasifikasi pada kurva ROC berikut.



Gambar 15. ROC curve

V. PEMBAHASAN

Dari hasil penelitian yang kami lakukan pada dataset Covid-19 Radiography Database dengan ukuran file sebesar 740 MB, di mana di dalam dataset tersebut terdapat 4 class diantaranya, class COVID dengan jumlah data citra (3616), Lung_Opacity dengan jumlah data citra (6012), Normal dengan jumlah data citra (10002), Viral Pneumonia dengan jumlah data citra (1345). Upaya dalam meningkatkan accuracy, dilakukan dengan klasifikasi metode deep learning dengan menerapkan arsitektur Xception. Langkah-langkah yang dilakukan pada penelitian ini adalah pre-processing data dengan split data training dan testing dengan perbandingan 70:30, proses klasifikasi, evaluasi kinerja untuk melihat accuracy, loss accuracy, validasi accuracy, confusion matrix, dan ROC curve. Hasil percobaan diperoleh akurasi pelatihan terbaik 92% dan akurasi validasinya sebesar 92% (best-fitting) dengan f1-score 25%. Rincian hasil untuk masing-masing kelas adalah sebagai berikut: Viral Pneumonia dengan precision 25% dan recall 23%, Lung_Opacity dengan precision 24% dan recall 26%, Normal dengan precision 28% dan recall 30%, Covid dengan precision 24% dan recall 21%. Dan untuk skor ROC AUC sebesar 50%.

VI. KESIMPULAN

Penelitian yang kami lakukan pada dataset Covid-19 Radiography Database yang berisi tentang status penderita/bukan penderita Covid-19 yang berjumlah 2000 data citra dengan 4 class di dalamnya di antara lain, Covid, Lung_Opacity, Normal, dan Viral Pneumonia. Upaya dalam meningkatkan accuracy data pada dataset ini dilakukan klasifikasi dan penerapan arsitektur Xception. Pada penelitian ini dilakukan perubahan ukuran gambar sesuai dengan kriteria arsitektur Xception dan setelah itu, dilakukan preprocessing data, dilakukan proses klasifikasi, pada proses ini dilakukan training dan testing pada model dengan perbandingan 70:30, prediksi model terhadap data baru. Kemudian proses selanjutnya melakukan evaluasi kinerja untuk melihat accuracy, validasi accuracy, confusion matrix, dan ROC curve.

DAFTAR ACUAN

- [1] Dr. Septriani Timurtini Limbong, "Pengertian Virus Corona (Covid-19)," 2019. Tersedia [Online]. <https://www.klikdokter.com/penyakit/coronavirus>. Diakses: 22 April 2021.
- [2] WHO, "Coronavirus," 2020. https://www.who.int/healthtopics/coronavirus#tab=tab_1 Diakses: 22 April 2021.
- [3] Worldometer, "Covid-19 Coronavirus Pandemic," 2020. <https://www.worldometers.info/coronavirus/>. Diakses: 22 April 2021.
- [4] Abbas Asmaa, Mohammed M. Abdelsamea, dan Mohamed Medhat Gaber, "Classification of Covid-19 in chest X-ray images using DeTraC deep convolution neural network," 2020. Tersedia [Online]. <https://link.springer.com/content/pdf/10.1007/s10489-020-01829-7.pdf>. Diakses: 22 April 2021.
- [5] Tuan D. Pham, "Classification of Covid-19 chest X-rays with deep learning: new models or fine tuning," 2020. Tersedia [Online]. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7680558/>. Diakses: 23 April 2021.
- [6] Ahyuna, Kalfin Alloto'dang, dkk., "Literature Review Artificial Intelligence Deteksi Hasil Ctscan Paru-Paru Pasien Terjangkit Covid-19," 2021. STMIK Dipanegara Makassar.