

DAFTAR PUSTAKA

- Alsing, O. (2018). *Mobile Object Detection using TensorFlow Lite and Transfer Learning*.
- Azmi, K., & Defit, S. (2023). Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat. *Jurnal Unitek*, 16(1), 2580–2582.
- C. pearce, E. (2016). *ebook_Anatomi_dan_Fisiologi_untuk_Paramame.pdf*.
- Dendi Maysanjaya, I. M. (2020). Klasifikasi Pneumonia pada Citra X-rays Paru-paru dengan Convolutional Neural Network (Classification of Pneumonia Based on Lung X-rays Images using Convolutional Neural Network). *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi* |, 9(2), 190.
- Goggin, G. (2012). Google phone rising: The Android and the politics of open source. *Continuum*, 26(5), 741–752.
<https://doi.org/10.1080/10304312.2012.706462>
- Hartono, W. (2017). Segmentasi Paru-Paru Pada Citra X-Ray Thorax Menggunakan K-Means. *Sains Dan Teknologi*.
- Hidayat, A., Darusalam, U., Technology, I., & Jakarta, S. (2019). Jurnal Ilmu Komputer dan Informasi (Journal of a Science and Information). 12 / 1 (2019), 51-56 DOI: <http://dx.doi.org/10.21609/jiki:v12i1:695>
DETECTION OF DISEASE ON CORN PLANTS USING CONVOLUTIONAL NEURAL. *Jurnal Ilmu Komputer Dan Informasi (Journal of a Science and Information)*, 1, 51–56.

- Higaki, T., Akita, K., & Katoh, K. (2020). Coefficient of variation as an image-intensity metric for cytoskeleton bundling. *Scientific Reports*, *10*(1), 22187. <https://doi.org/10.1038/s41598-020-79136-x>
- Hosseini, A., Eshraghi, M. A., Taami, T., Sadeghsalehi, H., Hoseinzadeh, Z., Ghaderzadeh, M., & Rafiee, M. (2023). A mobile application based on efficient lightweight CNN model for classification of B-ALL cancer from non-cancerous cells: A design and implementation study. *Informatics in Medicine Unlocked*, *39*(January), 101244. <https://doi.org/10.1016/j.imu.2023.101244>
- I. Akhlis, S. (2011). IMPLEMENTASI METODE HISTOGRAM EQUALIZATION UNTUK MENINGKATKAN KUALITAS CITRA DIGITAL. *Jurnal Fisika*, *3*, 70–74.
- Liu, K., Kang, G., Zhang, N., & Hou, B. (2018). Breast Cancer Classification Based on Fully-Connected Layer First Convolutional Neural Networks. *IEEE Access*, *6*, 23722–23732. <https://doi.org/10.1109/ACCESS.2018.2817593>
- Natasya, F. A. (2022). *TATALAKSANA PNEUMONIA*. *03*(02).
- Nurkamilah, A., Kudus, A., Statistika, P., Matematika, F., Alam, P., & Bandung, U. I. (2023). Selang Kepercayaan the Generalized Confidence Interval (GCI) untuk Koefisien Variasi dari Distribusi Invers Gaussian. *Bandung Conference Series: Statistics*, 628–634.
- Ode, L., & Sagala, A. S. (2022). *Klasifikasi Cats dan Dogs dengan Metode CNN dalam Fungsi Aktivasi relu , sigmoid , softmax , softplus , softsign , dan selu*. *September*.

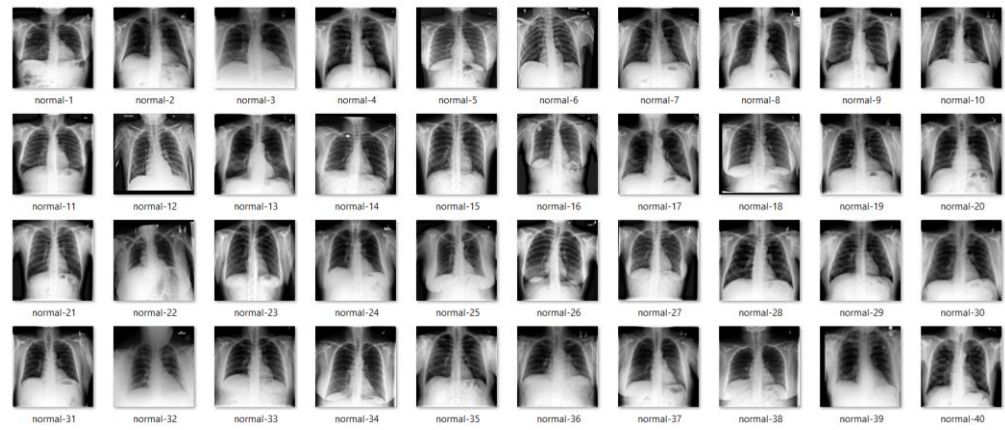
- Ouleddroun, T., Ellahyani, A., & El Ansari, M. (2023). Automated Pneumonia Detection using deep features in chest X-ray images. *2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, 1–6. <https://doi.org/10.1109/ICCT56969.2023.10076157>
- Putra, J. W. G. (2020). *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*. 4.
- Putra, R. A. D. I. (2023). Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN). *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(2), 466–478.
- Rahayu, W. I., Cahyo Prianto, & Novia, E. A. (2021). PERBANDINGAN ALGORITMA K-MEANS DAN NAÏVE BAYES UNTUK MEMREDIKSI PRIORITAS PEMBAYARAN TAGIHAN RUMAH SAKIT BERDASARKAN TINGKAT KEPENTINGAN PADA PT. PERTAMINA (PERSERO). *Jurnal Teknik Informatika*, 13(2), 1–8.
- Rahman, T., Khandakar, A., Member, S., Kadir, M. A., Islam, K. R., Islam, K. F., Mazhar, R., & Hamid, T. (2020). Reliable Tuberculosis Detection Using Chest X-Ray With Deep Learning , Segmentation and Visualization. *IEEE*, 8. <https://doi.org/10.1109/ACCESS.2020.3031384>
- Salmon, B. P., Kleynhans, W., Schwegmann, C. P., & Olivier, J. C. (2015). School of Engineering and ICT , University of Tasmania , Australia Remote Sensing Research Unit , Meraka Institute , CSIR , South Africa Department of Electrical , Electronic and Computer Engineering , University of Pretoria , South Africa. *IEEE*, 3057–3060.

- Sari, G. K., & Setyawati, T. (2022). *TUBERKULOSIS PARU POST WODEC PLEURAL EFUSION : LAPORAN KASUS PULMONARY TUBERCULOSIS POST WODEC PLEURAL EFFUSION : CASE REPORT*. 4(2), 174–182.
- Sia, N., Kong, P., Ibrahim, H., & Hoo, S. C. (2013). *A Literature Review on Histogram Equalization and Its Variations for Digital Image Enhancement*. 4(4), 386–389. <https://doi.org/10.7763/IJIMT.2013.V4.426>
- Tataroglu, O., Erdogan, S. T., Erdogan, M. O., & Tayfur, I. (2018). *Diagnostic Accuracy of Initial Chest X-Rays in Thorax Trauma*. 28(7), 546–548.
- Taye, M. M. (2023). *Understanding of Machine Learning with Deep Learning : Computers*.
- Tobias, R. R., Carlo De Jesus, L., Mital, M. E., Lauguico, S., Guillermo, M., Sybingco, E., Bandala, A., & Dadios, E. (2020). *Android Application for Chest X-ray Health Classification From a CNN Deep Learning TensorFlow Model*. 2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech), 255–259. <https://doi.org/10.1109/LifeTech48969.2020.1570619189>
- Wathani, M. R., Hidayati, N., Islam, U., Muhammad, K., & Al, A. (2023). *Analisis Perbandingan Fungsi Aktivasi CNN Pada Pengelompokan Jenis Beras Berdasarkan Mutu Beras*. *Jurnal Penerapan Kecerdasan Buatan*, 4(2), 144–153.

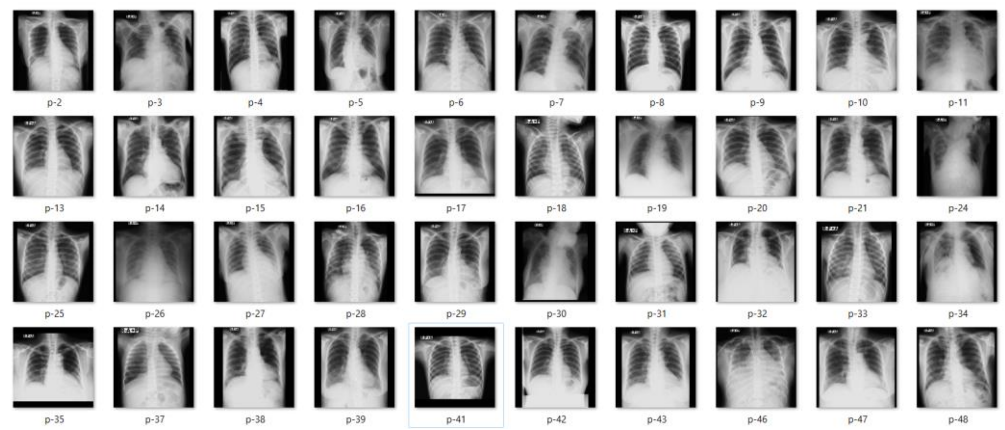
LAMPIRAN

1. Lampiran Dataset

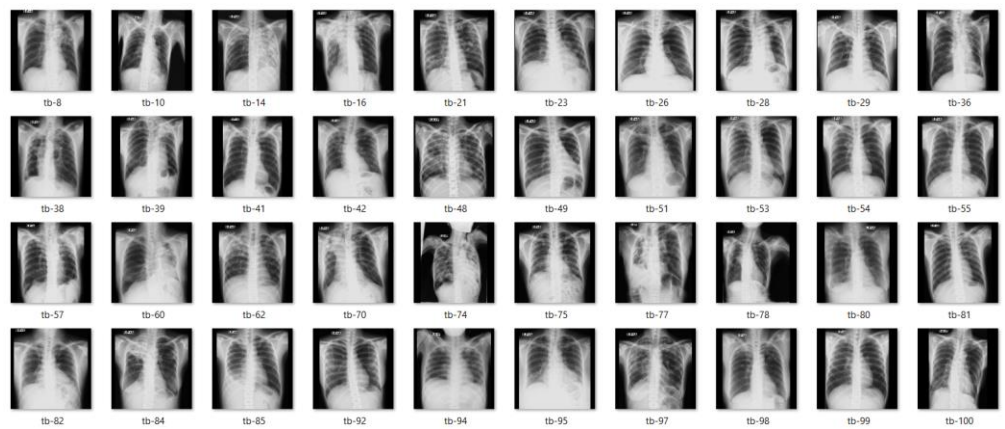
- **Kelas Normal**



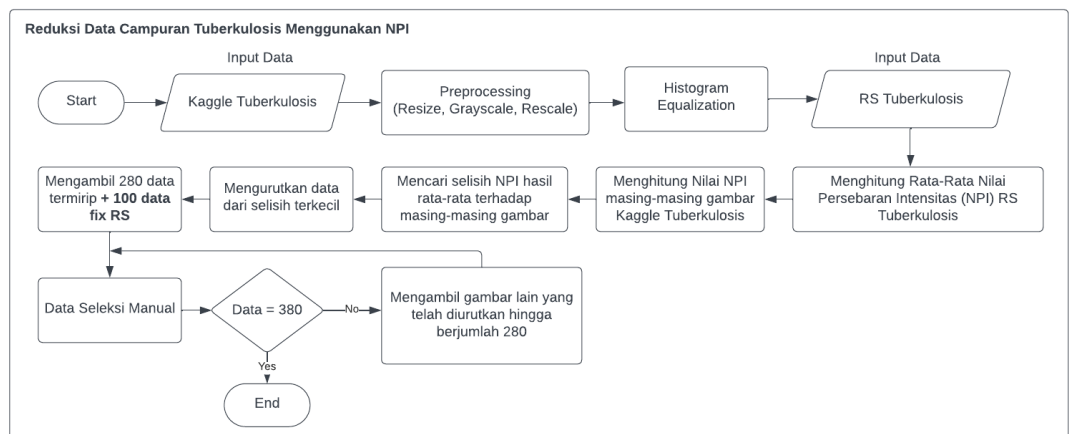
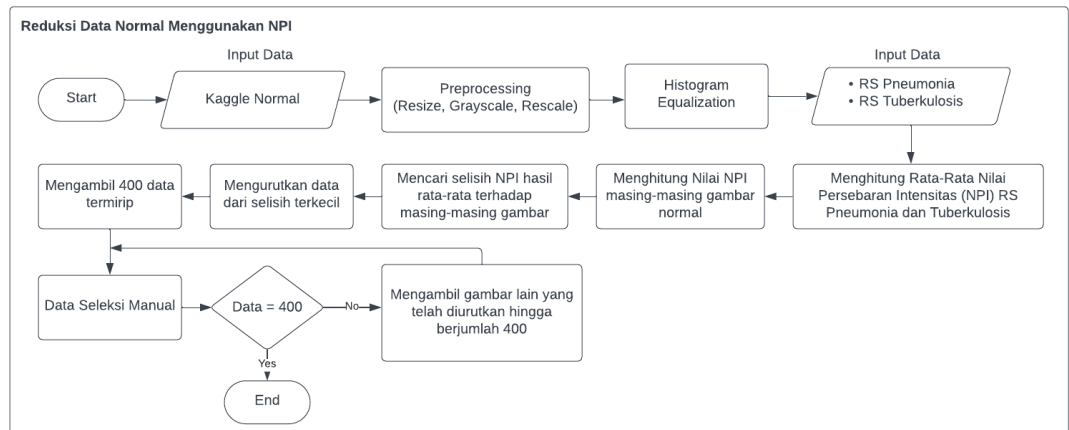
- **Kelas Pneumonia**



- **Kelas Tuberkulosis**



2. Lampiran *flowchart* Reduksi Data



3. Lampiran *Source Code* Program

a. *Resize, Grayscale, dan Histogram Equalization* Paru Normal

```
import os
import cv2

input_folder = "E:/data/PARU NORMAL"
output_folder = "E:/data/Paru HE/Paru Normal Kaggle - 512x512 - JPG - GRAYSCALE"

if not os.path.exists(output_folder):
    os.makedirs(output_folder)

for filename in os.listdir(input_folder):
    if filename.endswith((".jpg", ".jpeg", ".png")):
        image = cv2.imread(os.path.join(input_folder, filename))

        grayscale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

        equalized_image = cv2.equalizeHist(grayscale_image)

        resized_image = cv2.resize(equalized_image, (512, 512))

        output_path = os.path.join(output_folder, f"Normal-preprocessing{len(os.listdir(output_folder)) + 1}.jpg")
        cv2.imwrite(output_path, resized_image)
```

b. *Resize, Grayscale, dan Histogram Equalization* Paru Tuberkulosis

```
import os
import cv2

input_folder = "E:/data/PARU TBC 3000"
output_folder = "E:/data/Paru HE/Paru Tuberkulosis Kaggle - 512x512 - JPG - GRAYSCALE"

if not os.path.exists(output_folder):
    os.makedirs(output_folder)

for filename in os.listdir(input_folder):
    if filename.endswith((".jpg", ".jpeg", ".png")):
        image = cv2.imread(os.path.join(input_folder, filename))

        grayscale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

        equalized_image = cv2.equalizeHist(grayscale_image)

        resized_image = cv2.resize(equalized_image, (512, 512))

        output_path = os.path.join(output_folder, f"Tuberkulosis-preprocessing{len(os.listdir(output_folder)) + 1}.jpg")
        cv2.imwrite(output_path, resized_image)
```

c. Pencarian kontras gambar tuberkulosis termirip dengan kontras data asli

```
import os
import cv2
import numpy as np
import pandas as pd

kaggle_folder = "E:/data/Paru HE/Paru Tuberkulosis Kaggle - 512x512 - JPG - GRAYSCALE"

input_folder = "E:/DataTelahDiSeleksiKesamaan/TuberkulosisSebelumDiSeleksiManual"
mean_values = []
stddev_values = []

for filename in os.listdir(input_folder):
    if filename.endswith((".jpg", ".jpeg", ".png")):
        image = cv2.imread(os.path.join(input_folder, filename))

        grayscale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

        mean_value = np.mean(grayscale_image)
        stddev_value = np.std(grayscale_image)

        mean_values.append(mean_value)
        stddev_values.append(stddev_value)

average_contrast_hospital = np.mean([stddev / mean for stddev, mean in zip(stddev_values, mean_values)])
```



```

import os
import cv2
import numpy as np
import pandas as pd

kaggle_folder = "E:/data/Paru HE/Paru Tuberkulosis Kaggle - 512x512 - JPG - GRAYSCALE"

input_folder = "E:/DataTelahDiSeleksiKesamaan/TuberkulosisSebelumDiSeleksiManual"
mean_values = []
stddev_values = []

for filename in os.listdir(input_folder):
    if filename.endswith((".jpg", ".jpeg", ".png")):
        image = cv2.imread(os.path.join(input_folder, filename))

        grayscale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

        mean_value = np.mean(grayscale_image)
        stddev_value = np.std(grayscale_image)

        mean_values.append(mean_value)
        stddev_values.append(stddev_value)

average_contrast_hospital = np.mean([stddev / mean for stddev, mean in zip(stddev_values, mean_values)])

kaggle_files = [filename for filename in os.listdir(kaggle_folder) if filename.endswith((".jpg", ".jpeg", ".png"))]

# Menghitung kontras rasio untuk gambar-gambar Kaggle
contrast_ratios = []
for filename in kaggle_files:
    image = cv2.imread(os.path.join(kaggle_folder, filename))
    grayscale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    stddev = np.std(grayscale_image)
    mean = np.mean(grayscale_image)
    contrast_ratio = stddev / mean
    contrast_ratios.append(contrast_ratio)

# Menghitung selisih kontras dengan kontras rata-rata rumah sakit
selisih_kontras = [abs(contrast_ratio - average_contrast_hospital) for contrast_ratio in contrast_ratios]

# Membuat DataFrame
data = {
    'Nama File': kaggle_files,
    'Kontras Rasio': contrast_ratios,
    'Selisih Kontras': selisih_kontras
}

df = pd.DataFrame(data)
print(df)

```

```

df = df.sort_values(by='Selisih Kontras')

output_folder = "E:/data/Paru HE/Paru Tuberkulosis FIX"

if not os.path.exists(output_folder):
    os.makedirs(output_folder)

angka = 0
for i, row in df.iterrows():
    image = cv2.imread(os.path.join(kaggle_folder, row['Nama File']))
    output_filename = f"tuberkulosis-{angka + 1}.jpg"
    output_path = os.path.join(output_folder, output_filename)
    cv2.imwrite(output_path, image)
    angka = angka + 1

print("Proses selesai. Gambar telah disimpan sesuai urutan.")

```

d. Pencarian kontras gambar normal termirip dengan kontras data asli

```

import os
import cv2
import numpy as np
import pandas as pd

input_folder = "E:/data/Paru HE/Paru Normal Kaggle - 512x512 - JPG - GRAYSCALE"

mean_values = []
stddev_values = []
image_filenames = []

for filename in os.listdir(input_folder):
    if filename.endswith((".jpg", ".jpeg", ".png")):
        image = cv2.imread(os.path.join(input_folder, filename))

        grayscale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

        mean_value = np.mean(grayscale_image)
        stddev_value = np.std(grayscale_image)

```

```

        image_filenames.append(filename)

contrast_values = [stddev / mean for stddev, mean in zip(stddev_values, mean_values)]

data = {
    'Nama File': image_filenames,
    'Kontras Rasio': contrast_values
}

df = pd.DataFrame(data)

```

```

import os
import cv2
import pandas as pd

df['Selisih Kontras'] = abs(df['Kontras Rasio'] - 0.641)
df = df.sort_values(by='Selisih Kontras')

output_folder = "E:/data/Paru HE/Normal FIX"

# Membuat folder jika belum ada
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

angka = 0
for i, row in df.iterrows():
    image = cv2.imread(os.path.join(input_folder, row['Nama File']))
    output_filename = f"normal-{angka + 1}.jpg"
    output_path = os.path.join(output_folder, output_filename)
    cv2.imwrite(output_path, image)
    angka = angka + 1

print("Proses selesai. Gambar telah disimpan sesuai urutan.")

```

e. Training Model

- *Import library*

```

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Dense, Activation, Dropout, Conv2D, MaxPooling2D, Flatten, Input
from tensorflow.keras import layers, models
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model, load_model, Sequential
from keras.callbacks import ModelCheckpoint
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from matplotlib.pyplot import imshow
import os
import seaborn as sns
sns.set_style('darkgrid')
from sklearn.metrics import confusion_matrix, classification_report
import cv2
import time

```

- *Read Data*

```

sdir = r'D:\PercobaanLast\DataSet'

filepaths = []
labels = []

classlist = os.listdir(sdir)

for klass in classlist:
    classpath = os.path.join(sdir, klass)

    if os.path.isdir(classpath):
        flist = os.listdir(classpath)

        for f in flist:
            fpath = os.path.join(classpath, f)

            img = cv2.imread(fpath, cv2.IMREAD_GRAYSCALE)

            img = img.resize((64, 64))

            filepaths.append(fpath)
            labels.append(klass)

Fseries = pd.Series(filepaths, name='filepaths')
Lseries = pd.Series(labels, name='labels')
df = pd.concat([Fseries, Lseries], axis=1)

print(df.head())
print(df['labels'].value_counts())

```

- *Split Data*

```

train_split = 0.7
val_test_split = 0.15

# Bagi data menjadi data pelatihan (70%), data validasi (15%), dan data tes (15%)
train_df, temp_df = train_test_split(df, train_size=train_split, shuffle=True, random_state=123, stratify=df['labels'])
val_df, test_df = train_test_split(temp_df, train_size=val_test_split / (1 - train_split),
                                  shuffle=True, random_state=123, stratify=temp_df['labels'])

# Tampilkan panjang setiap set data
print('train_df length:', len(train_df), ' val_df length:', len(val_df), ' test_df length:', len(test_df))

```

- *Image Generator*

```

height = 64
width = 64
channels = 1
batch_size = 4
img_shape = (height, width, channels)
img_size = (height, width)
length = len(test_df)

# Membuat generator untuk data augmentasi pada dataset pelatihan
traingen = ImageDataGenerator(
    rescale=1./255,
    brightness_range=[0.8,1.2],
    horizontal_flip=True
)

train_gen = traingen.flow_from_dataframe(
    train_df,
    x_col='filepaths',
    y_col='labels',
    class_mode='categorical',
    color_mode='grayscale',
    shuffle=True,
    batch_size=batch_size,
    target_size=img_size,
)

```

```

val_gen = ImageDataGenerator(
    rescale=1./255
)

val_data = val_gen.flow_from_dataframe(
    val_df,
    x_col='filepaths',
    y_col='labels',
    class_mode='categorical',
    color_mode='grayscale',
    shuffle=False,
    batch_size=batch_size,
    target_size=img_size,
)

# Generator untuk data uji
testgen = ImageDataGenerator(rescale=1./255)

test_gen = testgen.flow_from_dataframe(
    test_df,
    x_col='filepaths',
    y_col='labels',
    target_size=img_size,
    class_mode='categorical',
    color_mode='grayscale',
    shuffle=False,
    batch_size=batch_size,
)

classes = list(train_gen.class_indices.keys())
print(classes)
class_count = len(classes)

```

- *Training tuning hyperparameter 5 skenario*

```

modelLR1 = models.Sequential()

# Layer 1
modelLR1.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same', input_shape=(height, width, 1)))
modelLR1.add(layers.MaxPooling2D((2, 2)))

# Layer 2
modelLR1.add(layers.Conv2D(32, (3, 3), activation='relu', padding='same'))
modelLR1.add(layers.MaxPooling2D((2, 2)))

# Layer 3
modelLR1.add(layers.Conv2D(64, (3, 3), activation='relu', padding='same'))
modelLR1.add(layers.MaxPooling2D((2, 2)))

modelLR1.add(layers.Flatten())

```

```
# Layer Fully Connected
modelLR1.add(layers.Dense(256, activation='relu'))
modelLR1.add(layers.Dense(128, activation='relu'))
modelLR1.add(layers.Dense(3, activation='softmax'))

# Compile model
modelLR1.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=.001),
                 loss='categorical_crossentropy',
                 metrics=['accuracy'])
```

```
checkpoint_filepath = 'D:/PercobaanValidation/Model3/modelLR1.h5'

epochs = 35

model_checkpoint_callback_acc = ModelCheckpoint(
    filepath=checkpoint_filepath,
    save_best_only=True,
    monitor='val_accuracy',
    mode='max',
    verbose=1
)

start_time = time.time()
historyLR1 = modelLR1.fit(
    x=train_gen,
    epochs=epochs,
    validation_data=val_data,
    callbacks=[model_checkpoint_callback_acc]
)
end_time = time.time()

training_time = end_time - start_time
print(f"Waktu pelatihan: {training_time} detik")
```

- Perbandingan grafik akurasi dan nilai *loss* data validasi

```
# Ambil data validation loss dan accuracy untuk tiga model
validation_loss_model1 = historyLR1.history['val_loss']
validation_accuracy_model1 = historyLR1.history['val_accuracy']

validation_loss_model2 = historyLR2.history['val_loss']
validation_accuracy_model2 = historyLR2.history['val_accuracy']

validation_loss_model3 = historyLR3.history['val_loss']
validation_accuracy_model3 = historyLR3.history['val_accuracy']

# Jumlah epoch
epochs = range(1, len(validation_loss_model1) + 1)

plt.figure(figsize=(12, 4))

# Subplot untuk Validation Loss
plt.subplot(1, 2, 1)
plt.plot(epochs, validation_loss_model1, 'b', label='Validation Loss Learning Rate 0.001', linewidth=2)
plt.plot(epochs, validation_loss_model2, 'g', label='Validation Loss Learning Rate 0.0001', linewidth=2)
plt.plot(epochs, validation_loss_model3, 'r', label='Validation Loss Learning Rate 0.00001', linewidth=2)
plt.title('Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(loc='upper left')

# Subplot untuk Validation Accuracy
plt.subplot(1, 2, 2)
plt.plot(epochs, validation_accuracy_model1, 'b', label='Validation Accuracy Learning Rate 0.001', linewidth=2)
plt.plot(epochs, validation_accuracy_model2, 'g', label='Validation Accuracy Learning Rate 0.0001', linewidth=2)
plt.plot(epochs, validation_accuracy_model3, 'r', label='Validation Accuracy Learning Rate 0.00001', linewidth=2)
plt.title('Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')

plt.tight_layout()
plt.show()
```

- Perbandingan grafik akurasi dan nilai *loss* data *testing*

```
import matplotlib.pyplot as plt

LR1 = 'D:/PercobaanValidation/Model3/modelLR1.h5'
LR2 = 'D:/PercobaanValidation/Model3/modelLR2.h5'
LR3 = 'D:/PercobaanValidation/Model3/modelLR3.h5'

modelLR1Fix = load_model(LR1)
modelLR2Fix = load_model(LR2)
modelLR3Fix = load_model(LR3)
```

```

def test_model(model, test_generator, title):
    test_steps = len(test_generator)
    test_loss, test_accuracy = model.evaluate(test_generator, steps=test_steps)
    print(f"{title} - Test Loss: {test_loss}")
    print(f"{title} - Test Accuracy: {test_accuracy}")
    return test_loss, test_accuracy

# Model 1
test_loss_model1, test_accuracy_model1 = test_model(modelLR1Fix, test_gen, "Learning rate 0.001")

# Model 2
test_loss_model2, test_accuracy_model2 = test_model(modelLR2Fix, test_gen, "Learning rate 0.0001")

# Model 3
test_loss_model3, test_accuracy_model3 = test_model(modelLR3Fix, test_gen, "Learning rate 0.00001")

# Plot grafik akurasi
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(['LR 0.001', 'LR 0.0001', 'LR 0.00001'], [test_accuracy_model1, test_accuracy_model2, test_accuracy_model3], marker='o')
plt.title('Test Accuracy')
plt.ylabel('Accuracy')

# Plot grafik Loss
plt.subplot(1, 2, 2)
plt.plot(['LR 0.001', 'LR 0.0001', 'LR 0.00001'], [test_loss_model1, test_loss_model2, test_loss_model3], marker='o', color='red')
plt.title('Test Loss')
plt.ylabel('Loss')

plt.tight_layout()
plt.show()

```

- Evaluasi model dengan *confusion matrix*

```

model_testing = load_model("D:/PercobaanValidation/Model2/modelkonvolusi4.h5")

y_pred_probs = model_testing.predict(test_gen, steps=test_steps)
y_pred = y_pred_probs.argmax(axis=1)

y_true = test_gen.classes

# Menghitung confusion matrix
cm = confusion_matrix(y_true, y_pred)

# Visualisasi Confusion Matrix dengan Seaborn
class_labels = ['Normal', 'Pneumonia', 'Tuberkulosis']
plt.figure(figsize=(12, 10))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False, xticklabels=class_labels,
            yticklabels=class_labels, annot_kws={"size": 16})
plt.xlabel('Predicted', fontsize=16)
plt.ylabel('Actual', fontsize=16)
plt.title('Confusion Matrix', fontsize=18)
plt.show()

# Menampilkan Classification Report
cr = classification_report(y_true, y_pred, target_names=class_labels)
print("Classification Report:\n", cr)

```

- Konversi model ke *tflite*

```

import tensorflow as tf
from tensorflow.keras.models import load_model

model_testing = load_model("D:/PercobaanValidation/Model2/modelkonvolusi4.h5")

converter = tf.lite.TFLiteConverter.from_keras_model(model_testing)
tflite_model = converter.convert()

output_path = "D:/PercobaanValidation/ModelFix/model.tflite"
with open(output_path, "wb") as f:
    f.write(tflite_model)

print(f"Model TFLite disimpan di: {output_path}")

```

- Implementasi model ke Android

```

1 package com.example.lungx
2
3 import android.content.Intent
4 import android.graphics.Bitmap
5 import android.graphics.Canvas
6 import android.graphics.ColorMatrix
7 import android.graphics.ColorMatrixColorFilter
8 import android.graphics.Paint
9 import androidx.appcompat.app.AppCompatActivity

```

```

10 import android.os.Bundle
11 import android.provider.MediaStore
12 import android.util.Log
13 import android.view.View
14 import android.widget.Button
15 import android.widget.ImageView
16 import android.widget.TextView
17 import com.example.lungx.databinding.ActivityMainBinding
18 import org.tensorflow.lite.DataType
19 import org.tensorflow.lite.support.tensorbuffer.TensorBuffer
20 import java.io.IOException
21 import java.nio.ByteBuffer
22 import java.nio.ByteOrder

```

```

24 class MainActivity : AppCompatActivity() {
25
26     private lateinit var gallery: Button
27     private lateinit var imageView: ImageView
28     private lateinit var result: TextView
29     private lateinit var binding : ActivityMainBinding
30
31     override fun onCreate(savedInstanceState: Bundle?) {
32         super.onCreate(savedInstanceState)
33         binding = ActivityMainBinding.inflate(layoutInflater)
34         setContentView(binding.root)
35
36         gallery = findViewById(R.id.button2)
37
38         result = findViewById(R.id.result)
39         imageView = findViewById(R.id.imageView)
40
41         gallery.setOnClickListener { it: View!
42             val cameraIntent = Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI)
43             startActivityForResult(cameraIntent, requestCode: 1)
44         }
45     }

```

```

50         binding.disclaimer.setOnClickListener{ it: View!
51             val view = View.inflate(context: this, R.layout.panduankonsultasi, root: null)
52             val builder = android.app.AlertDialog.Builder(context: this)
53             builder.setView(view)
54             val dialog = builder.create()
55
56             val mengerti = view.findViewById<Button>(R.id.mengerti)
57
58             dialog.show()
59             dialog.setCancelable(true)
60             dialog.setCanceledOnTouchOutside(true)
61             dialog.window?.setBackgroundDrawableResource(android.R.color.transparent)
62
63
64             mengerti.setOnClickListener{ it: View!
65                 dialog.dismiss()
66             }
67         }
68     }

```

```

71 private fun classifyImage(image: Bitmap) {
72     try {
73         val model = com.example.lungx.ml.Model.newInstance(applicationContext)
74
75         val inputFeature0 = TensorBuffer.createFixedSize(intArrayOf(1, 64, 64, 1), DataType.FLOAT32)
76         val byteBuffer = ByteBuffer.allocateDirect(capacity: 4 * 64 * 64 * 1)
77         byteBuffer.order(ByteOrder.nativeOrder())
78         val intValues = IntArray(size: 64 * 64)
79         image.getPixels(intValues, offset: 0, image.width, x: 0, y: 0, image.width, image.height)
80         var pixel = 0
81         for (i in 0 until < 64) {
82             for (j in 0 until < 64) {
83                 val `val` = intValues[pixel++]
84                 byteBuffer.putFloat(`val` and 0xFF).toFloat() * (1f / 255f))
85             }
86         }

```

```

87         inputFeature0.loadBuffer(byteBuffer)
88         val outputs = model.process(inputFeature0)
89         val outputFeature0 = outputs.getOutputFeature0AsTensorBuffer()
90         val confidences = outputFeature0.toFloatArray()
91         val classes = arrayOf("Normal", "Pneumonia", "Tuberkulosis")
92
93         val combinedData = classes.zip(confidences.toList())
94         val sortedCombinedData = combinedData.sortedByDescending { it.second }
95         val sortedClasses = sortedCombinedData.map { it.first }.toTypedArray()
96         val sortedConfidences = sortedCombinedData.map { it.second.toFloat() }.toFloatArray()
97
98         val max = sortedConfidences[0]*100
99         val tengah = sortedConfidences[1]*100
100        val min = sortedConfidences[2]*100
101        val maxclass = sortedClasses[0]
102        val tengahclass = sortedClasses[1]
103        val minclass = sortedClasses[2]

```

```

105        binding.labelNormal.text = maxclass
106        binding.confidenceNormal.text = max.toString()+"%"
107
108        binding.labelPneu.text = tengahclass
109        binding.confidencePneu.text = tengah.toString()+"%"
110
111        binding.labelTuberkulosis.text = minclass
112        binding.confidenceTuber.text = min.toString()+"%"
113
114        binding.gambarawal.visibility = View.GONE
115        binding.hasil.visibility = View.VISIBLE
116        binding.imageView.visibility = View.VISIBLE
117
118        model.close()
119    } catch (e: IOException) {
120        Log.d( tag: "Error",e.toString())
121    }
122    }

```

```

93    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
94        if (resultCode == RESULT_OK) {
95            var image: Bitmap? = null
96
97            val dat = data?.data
98            try {
99                image = MediaStore.Images.Media.getBitmap(contentResolver, dat)
100            } catch (e: IOException) {
101                e.printStackTrace()
102            }
103
104            val resizedImage = Bitmap.createScaledBitmap(image!!, dstWidth: 64, dstHeight: 64, filter: false)
105
106            val grayscaleImage = convertToGrayscale(resizedImage)
107
108            imageView.setImageBitmap(image)
109            classifyImage(grayscaleImage)
110        }
111        super.onActivityResult(requestCode, resultCode, data)
112    }

```

```

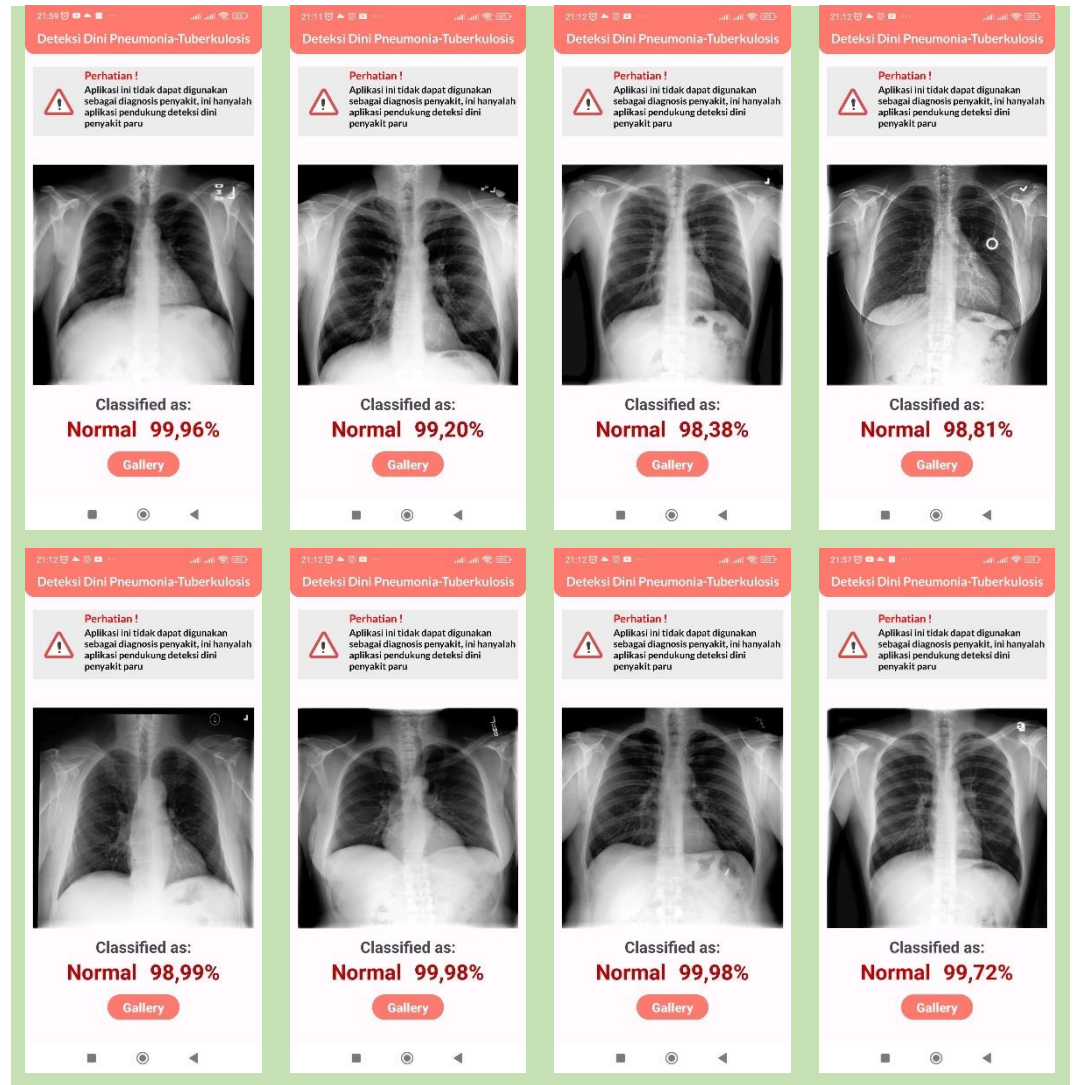
114    private fun convertToGrayscale(inputImage: Bitmap): Bitmap {
115        val grayscaleImage = Bitmap.createBitmap(inputImage.width, inputImage.height, Bitmap.Config.ARGB_8888)
116        val canvas = Canvas(grayscaleImage)
117        val paint = Paint()
118        val colorMatrix = ColorMatrix()
119        colorMatrix.setSaturation(0f)
120        val filter = ColorMatrixColorFilter(colorMatrix)
121        paint.colorFilter = filter
122        canvas.drawBitmap(inputImage, left: 0f, top: 0f, paint)
123        return grayscaleImage
124    }

```

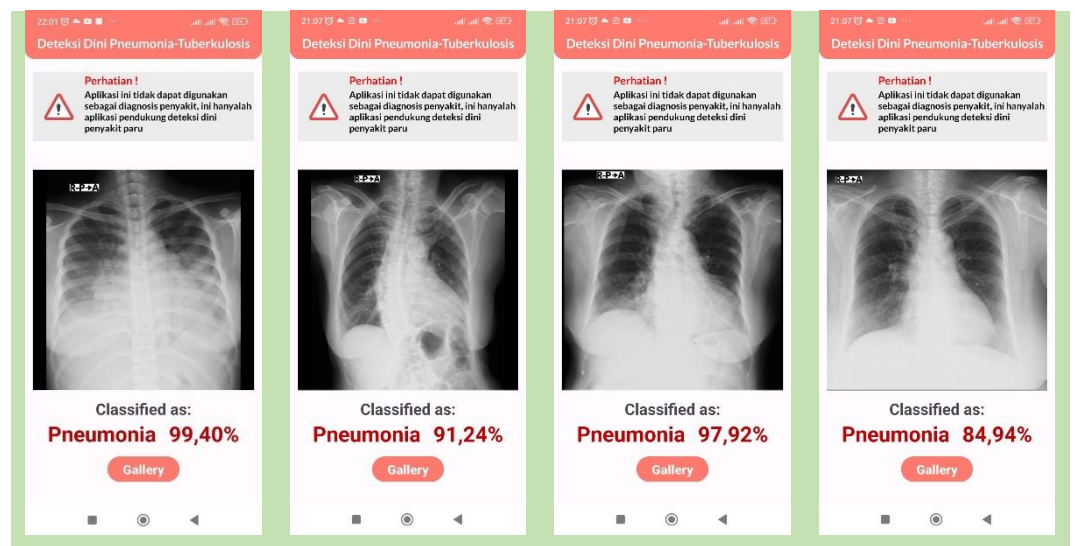
4. Lampiran Testing Aplikasi *LungX* (60 citra)

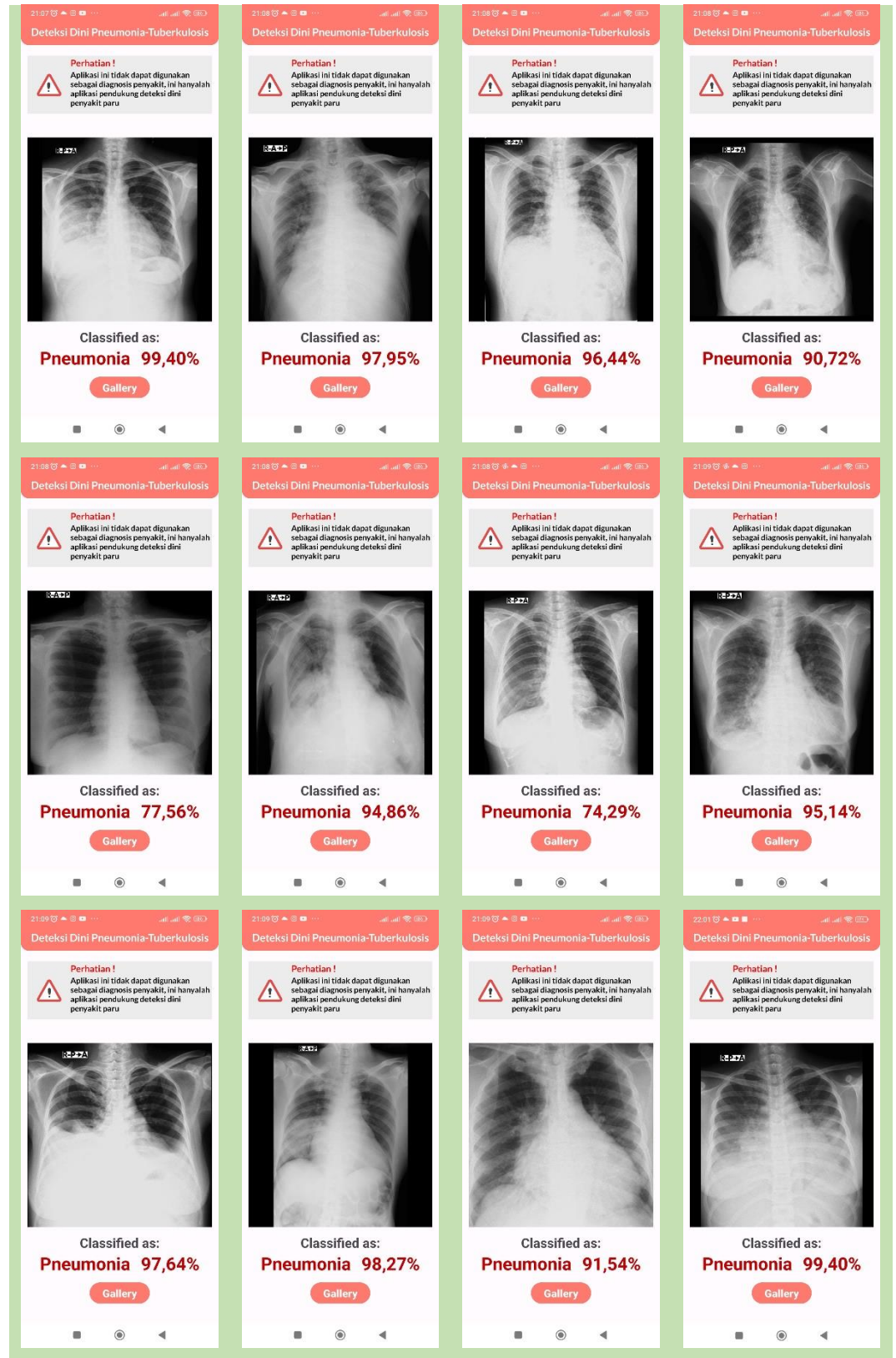
- Label Normal (20 citra)

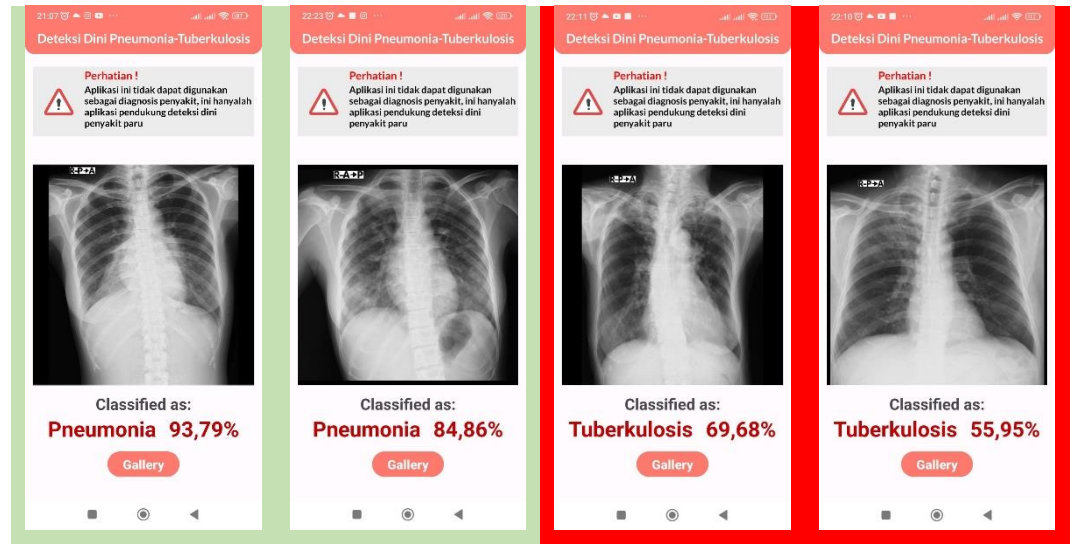




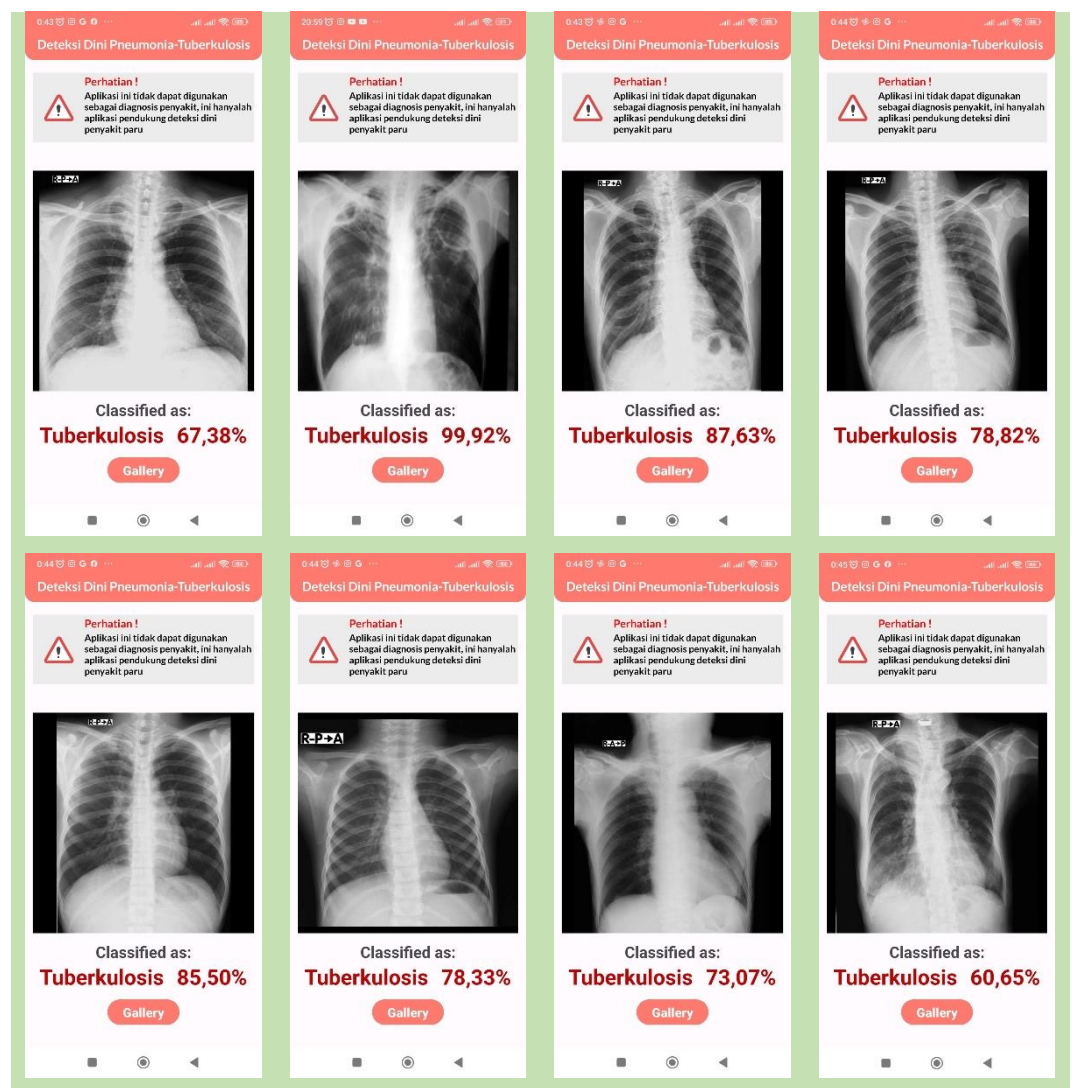
- Label Pneumonia (20 Citra)

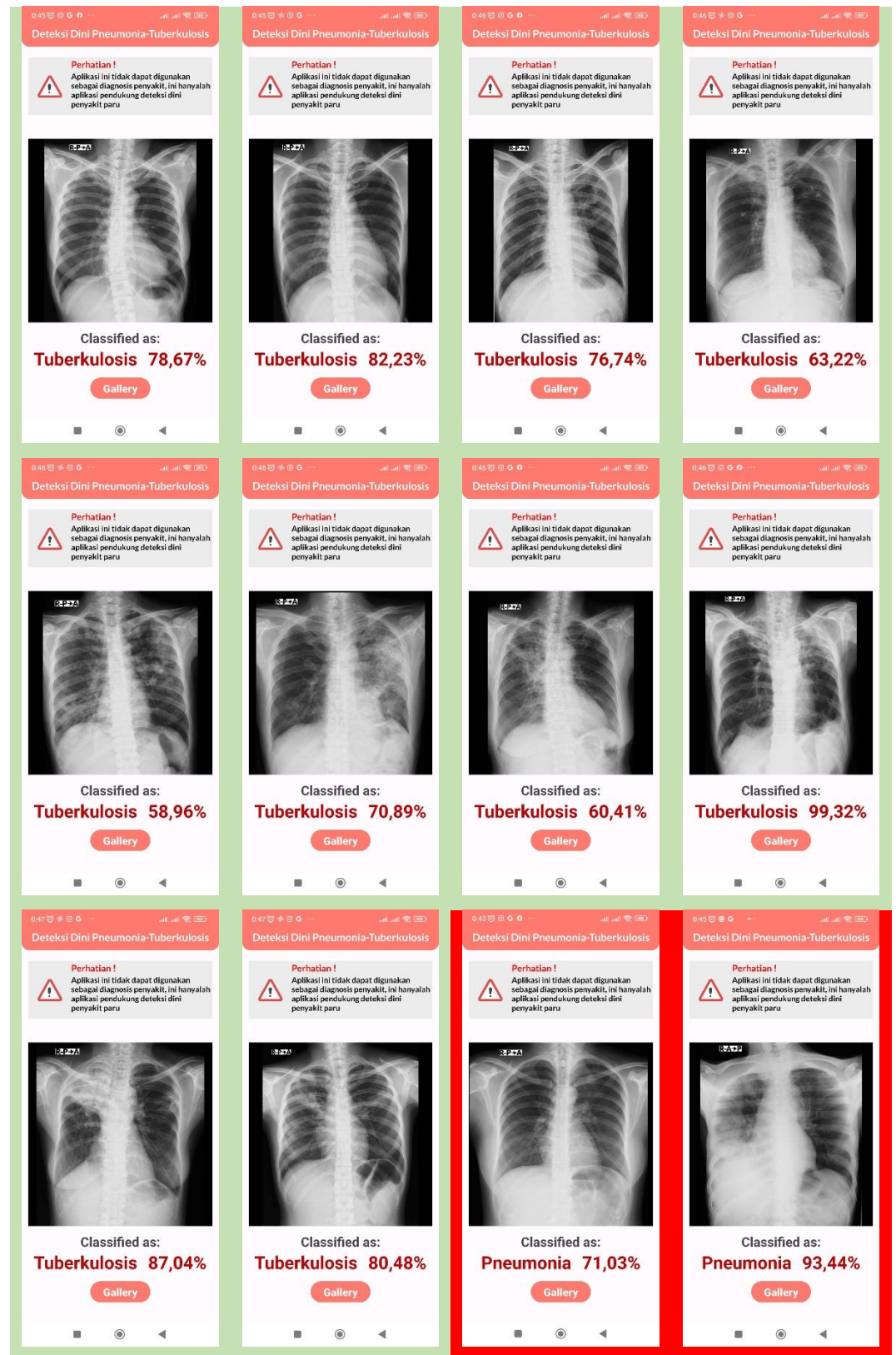






- Label Tuberkulosis (20 Citra)





LEMBAR PERBAIKAN SKRIPSI

**“KLASIFIKASI CITRA SINAR-X PENYAKIT PARU
MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK
BERBASIS ANDROID”**

OLEH:





CALVIN RINALDY LEONARD

D121201079

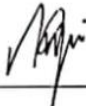

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 5 Februari 2024.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Ir. Ingrid Nurtanio, M.T.	
Sekretaris	Anugrayani Bustamin, S.T., M.T.	
Anggota	Prof. Dr.Eng Intan Sari Areni., ST., MT.	
	Novy N. R. A. Mokobombang, S.T., Ms.TM., PhD	

Persetujuan perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Ir. Ingrid Nurtanio, M.T.	
II	Anugrayani Bustamin, S.T., M.T.	



KEMENTERIAN PENDIDIKAN , KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR SEMINAR HASIL

Nama/Stambuk : 1. Calvin Rinaldy Leonard D121201079

Judul Skripsi/T.A : "Klasifikasi Citra Sinar-X Penyakit Paru Menggunakan Convolutional Neural Network Berbasis Android"

Hari/Tanggal : Rabu, 6 Desember 2023

Jam : 11- 00 Wita – Selesai

Tempat : Ruang Lab. AIMP Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1.Dr. Ir.Ingrid Nurtanio,M.T	1.....
	Pembimbing II	2.Anugrayani Bustamin,ST.,M.T	2.....
II.	Anggota Penguji	3.Novy Nur R.A Mokobombang,ST.,Ms.,TM.,Ph.D	3.....
		4. Prof. Dr. Eng. Intan Sari Areni,ST.,M.T	4.....

PANITIA UJIAN

Ketua,

Dr. Ir.Ingrid Nurtanio,M.T

Sekretaris,

Anugrayani Bustamin,ST.,M.T



KEMENTERIAN PENDIDIKAN , KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA SEMINAR HASIL

Pada hari ini Rabu, tanggal 6 Desember 2023 Pukul 11.00 WITA - Selesai bertempat di Ruang Lab.AIMP Departemen Teknik Informatika Gowa, telah dilaksanakan Seminar Hasil bagi Saudara :

Nama : Calvin Rinaldy Leonard
No. Stambu : D121201079
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : "Klasifikasi Citra Sinar-X Penyakit Paru Menggunakan Convolutional Neural Network Berbasis Android"

Yang dihadiri oleh Tim Penguji Seminar Hasil sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Ir.Ingrid Nurtanio,M.T	Pemb I/Ketua	1...
2.	Anugrayani Bustamin,ST.,M.T	Pemb II/Sekretaris	2...
3.	Novy Nur R.A Mokobombang,ST.,Ms.,TM.,Ph.D	Anggota	3...
4.	Prof. Dr. Eng. Intan Sari Areni,ST.,M.T	Anggota	4...

Hasil keputusan Tim Penguji Seminar Hasil : **Lulus** / ~~Tidak lulus~~ dengan nilai angka87..... dan huruf **A**.....

Gowa, 6 Desember 2023
Ketua/Sekretaris Panitia Ujian,

Dr. Ir.Ingrid Nurtanio,M.T



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA**

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

Nomor : 1392/UN4.7.7/TD.06/2023
Lamp : -
Hal : Penerbitan Surat Penugasan Panitia
Seminar Hasil Strata Satu (S1)

Kepada Yth :

Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Universitas Hasanuddin

Di-

Gowa

Dengan hormat,
Berdasarkan Persetujuan Pembimbing Mahasiswa, Bersama ini diusulkan susunan Panitia Seminar Hasil Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik tersebut di bawah ini :

Nama / Stambuk : Calvin Rinaldy Leonard D121201079
Judul TA : Klasifikasi Citra Sinar-X Penyakit Paru Menggunakan
Convolutional Neural Network Berbasis Android

Dengan ini kami sampaikan Susunan Panitia Seminar Hasil Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I/ Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T
Pembimbing II / Sekretaris: 2. Anugrayani Bustamin, ST., M.T
Anggota : 3. Novy Nur R.A Mokobombang, ST., Ms.TM. Ph. D
4. Prof. Dr.Eng. Intan Sari Areni, S.T., M.T.

Untuk dapat diterbitkan surat penugasannya

Demikian penyampaian kami, atas perhatian dan kerjasamanya diucapkan terima kasih.

Gowa, 1 Desember 2023
Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu., ST, MT, M.Bus.Sys., IPM, ASEAN.Eng
Nip.19750716 200212 1 004

Tembusan :

1. Arsip



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK**

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 28232/UN4.7.1/TD.06/2023

Dari : Dekan Fakultas Teknik Universitas Hasanuddin

Kepada : Mereka yang tercantum namanya dibawah ini

Isi : 1. Bahwa berdasarkan peraturan Akademik Universitas Hasanuddin Tahun 2018 pasal 18 (SK.Rektor Unhas nomor : 2781/UN4.1/KEP/2018), dengan ini menugaskan Saudara sebagai PANITIA SEMINAR HASIL Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I/ Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T
Pembimbing II / Sekretaris : 2. Anugrayani Bustamin, ST., M.T
Anggota : 3. Novy Nur R.A Mokobombang, ST., Ms.TM. Ph. D
4. Prof. Dr.Eng. Intan Sari Areni, S.T., M.T.

Untuk menguji bagi mahasiswa tersebut dibawah ini :

Nama/NIM : Calvin Rinaldy Leonard D121201079
Program Studi : Teknik Informatika
Judul thesis/Skripsi : Klasifikasi Citra Sinar-X Penyakit Paru Menggunakan Convolutional Neural Network Berbasis Android

2. Waktu seminar ditetapkan oleh Panitia Seminar Hasil Program Strata Satu (S1)
3. Agar Surat Penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasa ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudia hari terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa

Pada tanggal 1 Desember 2023

a.n. Dekan,

Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT
NIP. 197310101998021001

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Mahasiswa yang bersangkutan



• Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSrE
• UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1

"Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah"



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR UJIAN SKRIPSI MAHASISWA
FAKULTAS TEKNIK UNHAS

Nama/Stambuk : 1. Calvin Rinaldy Leonard D121201079

Judul Skripsi/T.A : "Klasifikasi Citra Sinar-X Penyakit Paru Menggunakan Convolutional Neural Network Berbasis Android"

Hari/Tanggal : Senin, 5 Februari 2024

Jam : 09- 00 Wita – Selesai

Tempat : Ruang Lab. AIMP Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr.Ir. Ingrid Nurtanio,M.T	1.....
	Pembimbing II	2. Anugrayani Bustamin,ST.,M.T	2.....
II.	Anggota Penguji	3. Novy Nur R.A. Mokobombang,ST.,Ms. TM.Ph.D	3.....
		4. Prof. Dr. Eng. Intan Sari Areni, ST.,M.T	4.....

PANITIA UJIAN

Ketua,

Dr.Ir. Ingrid Nurtanio,M.T

Sekretaris,

Anugrayani Bustamin,ST.,M.T



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA UJIAN SKRIPSI

Pada hari ini **Senin** tanggal 5 Februari 2024 Pukul 09.00 WITA - Selesai bertempat di **Ruang Lab. AIMP Departemen Teknik Informatika Gowa**, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Calvin Rinaldy Leonard
No. Stambuk : D121201079
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : "Klasifikasi Citra Sinar-X Penyakit Paru Menggunakan Convolutional Neural Network Berbasis Android"

Yang dihadiri oleh Tim Penguji Ujian Skripsi sebagai berikut :

No.	Nama	Jabatan	Tanda tangan
1.	Dr.Ir. Ingrid Nurtanio, M.T	Pemb I/Ketua	1.
2.	Anugrayani Bustamin, ST., M.T	Pemb II/Sekretaris	2.
3.	Novy Nur R.A. Mokobombang, ST., Ms. TM. Ph.D	Anggota	3.
4.	Prof. Dr. Eng. Intan Sari Areni, ST., M.T	Anggota	4.

Hasil keputusan Tim Penguji Ujian Skripsi/Tugas Akhir : **Lulus** / ~~Tidak lulus~~ dengan nilai angka **9.0** dan huruf **A**

Gowa, 5 Februari 2024

Ketua/Sekretaris Panitia Ujian,

Dr. Ir. Ingrid Nurtanio, M.T



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA**

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

Gowa, 3 Februari 2024

Nomor : 212/UN4.7.7.1/TD.06/2024
Lamp : -
Hal : Usulan Susunan Panitia/Penguji Ujian Sarjana

Yth. : Bapak Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas
Di
Gowa

Dalam rangka penyelesaian studi pada Departemen Teknik Informatika Fakultas Teknik Unhas, bersama ini kami usulkan susunan Panitia/Penguji Ujian Sarjana Program Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas nama :

Pembimbing I / Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T
Pembimbing II / Sekretaris : 2. Anugrayani Bustamin, ST., M.T
Anggota : 3. Novy Nur R.A. Mokobombang, ST., Ms.TM. Ph. D
4. Prof. Dr.Eng. Intan Sari Areni, S.T., M.T

Untuk Bertugas sebagai Penguji/ Penanggap Ujian Sarjana bagi Mahasiswa :

Nama : Calvin Rinaldy Leonard
Stambuk : D121201079

Dengan Judul Skripsi

“Klasifikasi Citra Sinar-X Penyakit Paru Menggunakan Convolutional Neural Network Berbasis Android “

Pada :

Hari/Tanggal : Senin, 5 Februari 2024
Jam : 09.00 Wita - Selesai
Tempat : Ruang Sidang Lab. AIMP

Demikian penyampaian kami, atas perhatiannya diucapkan terimah kasih.

Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys., IPM, ASEAN.Eng
Nip.197507016 200212 1 004

Tembusan :
1. Arsip



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK**

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 2767/UN4.7.1/TD.06/2024

Dari : Dekan Fakultas Teknik Universitas Hasanuddin.

Kepada : Mereka yang tercantum namanya di bawah ini.

Isi : 1. Bahwa merujuk kepada Peraturan Rektor Universitas Hasanuddin Nomor : 29/UN4.1/2023 tentang Penyelenggaraan Program Sarjana Universitas Hasanuddin, dengan ini menugaskan Saudara sebagai PENGUJI/PANITIA UJIAN SARJANA Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I / Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T
Pembimbing II / Sekretaris : 2. Anugrayani Bustamin, ST., M.T
Anggota : 3. Novy Nur R.A. Mokobombang, ST., Ms.TM. Ph. D
4. Prof. Dr.Eng. Intan Sari Areni, S.T., M.T.

untuk menguji bagi mahasiswa tersebut di bawah ini :

Nama/NIM : Calvin Rinaldy Leonard D121201079
Program Studi : Teknik Informatika
Judul Thesis/Skripsi : “Klasifikasi Citra Sinar-X Penyakit Paru Menggunakan Convolutional Neural Network Berbasis Android “

2. Waktu Ujian ditetapkan oleh Panitia Ujian Sarjana Program Strata Satu (S1).
3. Agar Surat penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasan ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya Ujian Sarjana tersebut, dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa,
Pada tanggal 3 Pebruari 2024
a.n. Dekan

Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT
NIP.197310101998021001

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Kasubag. Umum dan Perlengkapan FT-UH

