

DAFTAR PUSTAKA

- Abràmoff, M., & Kay, C. N. (2013). Image Processing. In *Retina* (pp. 151–176). Elsevier. <https://doi.org/10.1016/B978-1-4557-0737-9.00006-0>
- Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-94463-0>
- Astari, P. (2018). Katarak: Klasifikasi, Tatalaksana, dan Komplikasi Operasi. *Cermin Dunia Kedokteran*, 45(10). <https://doi.org/10.55175/cdk.v45i10.584>
- Berrada, L., Zisserman, A., & Kumar, M. P. (2018). *Deep Frank-Wolfe For Neural Network Optimization*.
- Brown, N. A. P. (1993). The morphology of cataract and visual performance. *Eye*, 7(1), 63–67. <https://doi.org/10.1038/eye.1993.14>
- Budianto, A. J., & Saian, P. O. N. (2023). Pengembangan Modul Inventory Management Pada Aplikasi Master Distribution Centre System Menggunakan Framework Flask Di PT XYZ. *Jurnal Jtik (Jurnal Teknologi Informasi Dan Komunikasi)*, 7(2), 201–207. <https://doi.org/10.35870/jtik.v7i2.714>
- Delbarre, M., & Froussart-Maille, F. (2020). Sémiologie et formes cliniques de la cataracte chez l'adulte. *Journal Français d'Ophtalmologie*, 43(7), 653–659. <https://doi.org/10.1016/j.jfo.2019.11.009>
- Dogo, E. M., Afolabi, O. J., Nwulu, N. I., Twala, B., & Aigbavboa, C. O. (2018). A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks. *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, 92–99. <https://doi.org/10.1109/CTEMS.2018.8769211>
- Fuadah, Y. N., Magdalena, R., Palondongan, S., & Kumalasari, N. (2019). OPTIMASI K-NEAREST NEIGHBOR UNTUK SISTEM KLASIFIKASI KONDISI KATARAK. *TEKTRIKA - Jurnal Penelitian Dan Pengembangan Telekomunikasi, Kendali, Komputer, Elektrik, Dan Elektronika*, 4(1), 16. <https://doi.org/10.25124/tektrika.v4i1.1832>
- Gholamalinezhad, H., & Khosravi, H. (2020). *Pooling Methods in Deep Neural Networks, a Review*.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. The MIT Press.
- Grandini, M., Baqli, E., & Visani, G. (2020). Metrics for Multi-Class Classification: an v, *abs/2008.05756*. arxiv.org/abs/2008.05756. antischolar.org/CorpusID:221112671
- Ien, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, hen, T. (2018). Recent advances in convolutional neural networks. *ition*, 77, 354–377. <https://doi.org/10.1016/j.patcog.2017.10.013>



- Irwansyah, E., & Faisal, M. (2015). *Advanced Clustering: Teori dan Aplikasi*. Deepublish.
- Kaka Kamaludin, Woro Isti Rahayu, & Helmi Setywan, M. Y. (2023). TRANSFER LEARNING TO PREDICT GENRE BASED ON ANIME POSTERS. *Jurnal Teknik Informatika (Jutif)*, 4(5), 1041–1052. <https://doi.org/10.52436/1.jutif.2023.4.5.860>
- Kanski, J. J. (2007). *Clinical ophthalmology: a systematic approach* (6th ed.). Butterworth-Heinemann/Elsevier.
- Ketkar, N. (2017). *Deep Learning with Python : A Hands-on Introduction*. Apress Media.
- Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8), 5455–5516. <https://doi.org/10.1007/s10462-020-09825-6>
- Kurniawan, B., Gunawan, R., & Elanda, A. (2022). Redesain Sistem Aplikasi Web Sumber Reload. *Prosiding Seminar Nasional Inovasi Dan Adopsi Teknologi (INOTEK)*, 2(1), 240–249. <https://doi.org/10.35969/inotek.v2i1.246>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Nayak, D. R., Dash, R., & Majhi, B. (2020). Automated diagnosis of multi-class brain abnormalities using MRI images: A deep convolutional neural network based method. *Pattern Recognition Letters*, 138, 385–391. <https://doi.org/10.1016/j.patrec.2020.04.018>
- Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA EKSPRESI MANUSIA. *ALGOR*, 2(1), 12–20. <https://jurnal.buddhidharma.ac.id/index.php/algor/article/view/441>
- Patterson, J., & Gibson, A. (2017). *Deep learning : a practitioner's approach*. Deep learning : a practitioner's approach.
- Putra, A. T., Usman, K., & Saidah, S. (2021). WEBINAR STUDENT PRESENCE SYSTEM BASED ON REGIONAL CONVOLUTIONAL NEURAL NETWORK USING FACE RECOGNITION. *Jurnal Teknik Informatika (Jutif)*, 2(2), 109–118. <https://doi.org/10.20884/1.jutif.2021.2.2.82>
- Shafirra, N. A., & Irhamah, I. (2020). Klasifikasi Sentimen Ulasan Film Indonesia dengan Konversi Speech-to-Text (STT) Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Sains Dan Seni ITS*, 9(1). <https://doi.org/10.12962/j23373520.v9i1.51825>



, Ying, Y., & Yuan, X. (2020). Stability and optimization error of gradient descent for pairwise learning. *Analysis and Applications*, 17. <https://doi.org/10.1142/S0219530519400062>

, & Rahmawati, S. (2018). Aplikasi Pendiagnosa Penyakit Mata menggunakan Metode Certainty Factor. *JURNAL TEKNOLOGI*, 1(1), 85. <https://doi.org/10.36294/jurti.v2i1.412>

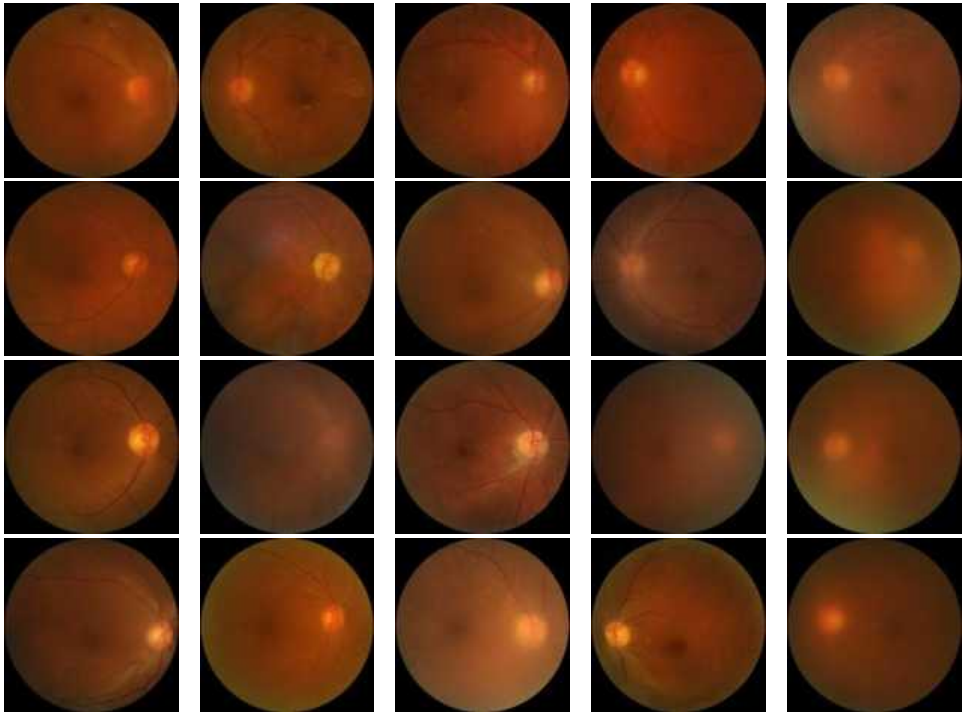
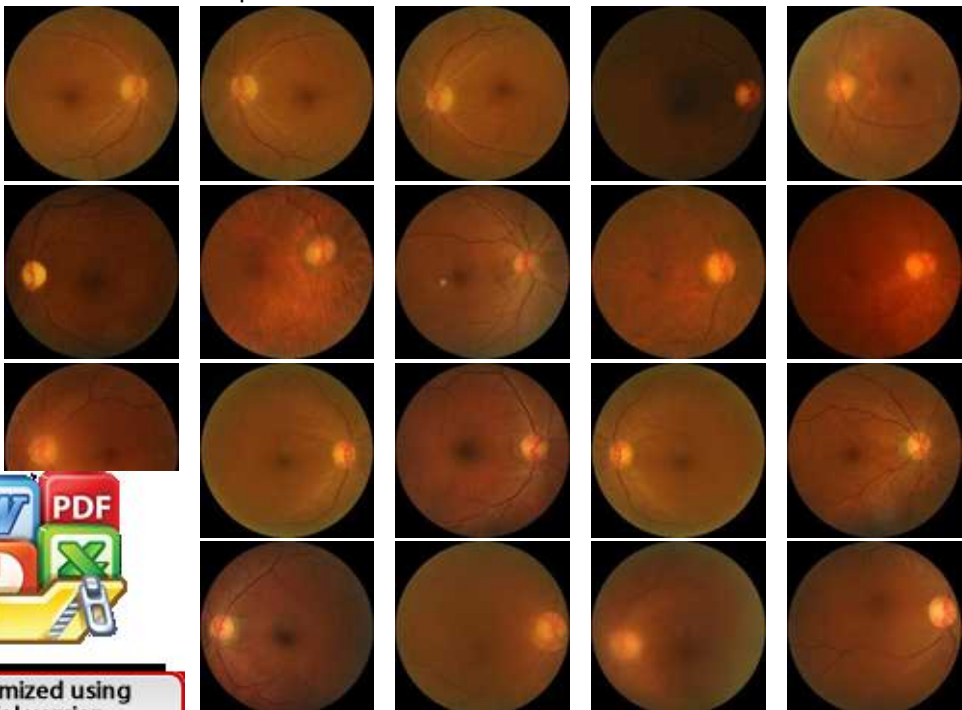
- Wang, Y., Tang, C., Wang, J., Sang, Y., & Lv, J. (2021). Cataract detection based on ocular B-ultrasound images by collaborative monitoring deep learning. *Knowledge-Based Systems*, 231, 107442. <https://doi.org/10.1016/j.knosys.2021.107442>
- Witanto, K. S., Sanjaya ER, N. A., Karyawati, A. E., Kadyanan, I. G. A. G. A., Suhartana, I. K. G., & Astuti, L. G. (2022). Implementasi LSTM Pada Analisis Sentimen Review Film Menggunakan Adam Dan RMSprop Optimizer. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 10(4), 351. <https://doi.org/10.24843/JLK.2022.v10.i04.p05>
- Xie, H., Li, Z., Wu, C., Zhao, Y., Lin, C., Wang, Z., Wang, C., Gu, Q., Wang, M., Zheng, Q., Jiang, J., & Chen, W. (2023). Deep learning for detecting visually impaired cataracts using fundus images. *Frontiers in Cell and Developmental Biology*, 11. <https://doi.org/10.3389/fcell.2023.1197239>
- Yadav, S., & Yadav, J. K. P. S. (2023). Automatic Cataract Severity Detection and Grading Using Deep Learning. *Journal of Sensors*, 2023, 1–17. <https://doi.org/10.1155/2023/2973836>
- Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4), 611–629. <https://doi.org/10.1007/s13244-018-0639-9>
- Zidan Rusminto, M., Adi Wibowo, S., & Santi Wahyuni, F. (2024). Peramalan Harga Saham Menggunakan Metode Arima (Autoregressive Integrated Moving Average) Time Series. *Jati (Jurnal Mahasiswa Teknik Informatika)*, 8(2), 1263–1270. <https://doi.org/10.36040/jati.v8i2.9089>



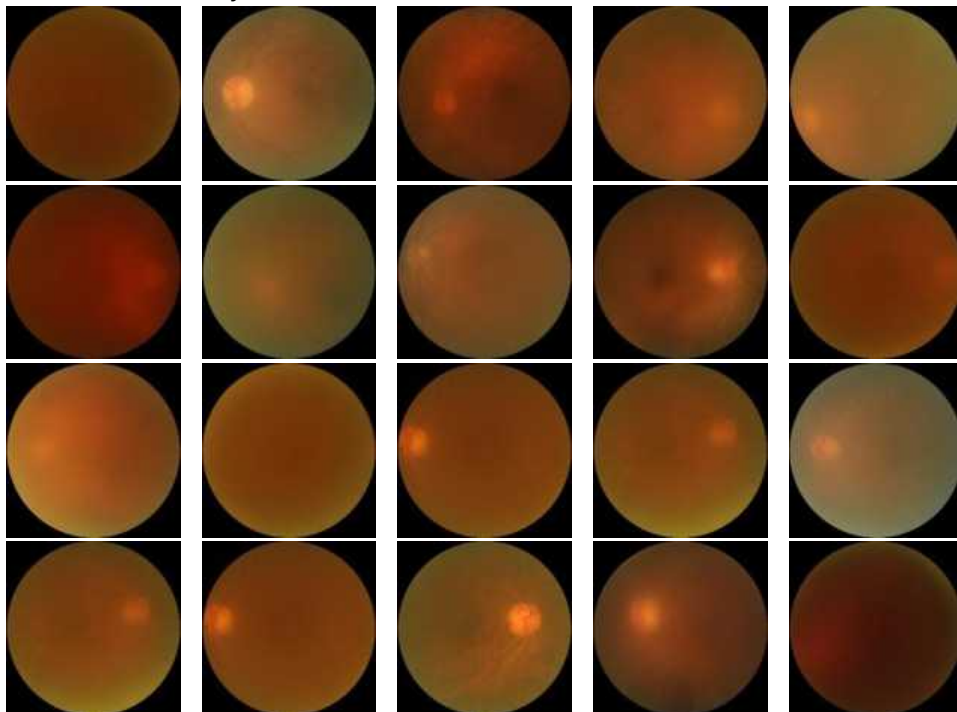
LAMPIRAN



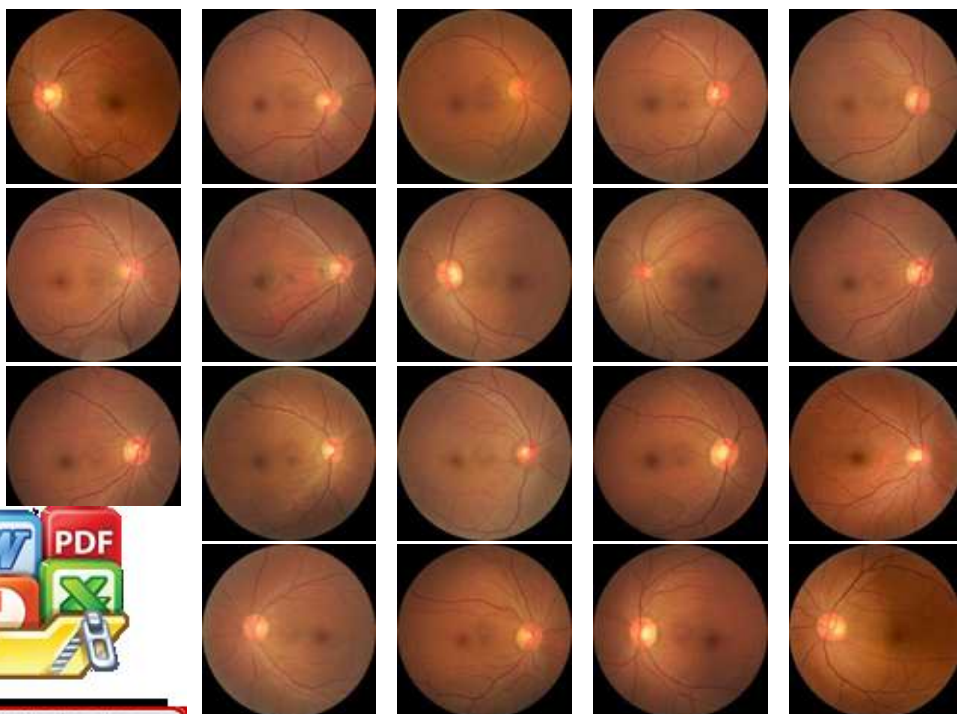
Optimized using
trial version
www.balesio.com

Lampiran 1. Dataset**a. Kelas Katarak Imatur****b. Kelas Katarak Subkapsular**

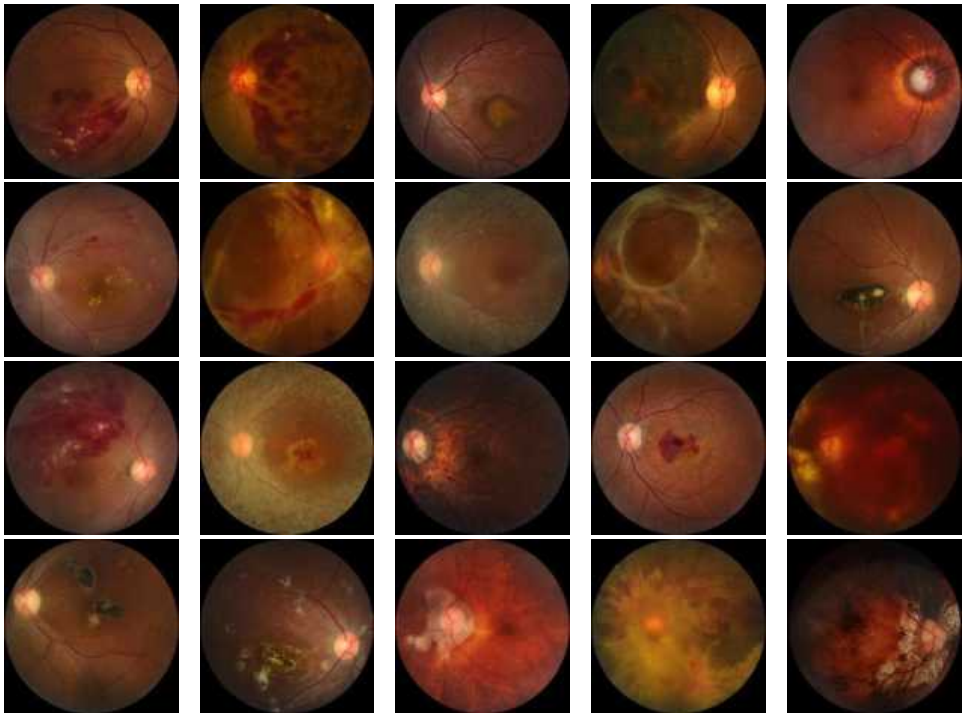
c. Kelas Katarak Lainnya



d. Kelas Normal



e. Kelas Bukan Katarak



Lampiran 2. Source Code Program

a. Ekstraksi Fitur

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
from tensorflow.keras.preprocessing import image
from tqdm import tqdm
import os

model = Sequential([
    Conv2D(16, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D(2, 2),
    BatchNormalization(),

    Conv2D(32, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    BatchNormalization(),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    BatchNormalization(),

    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    BatchNormalization(),

    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])

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

# Memuat dan mempersiapkan dataset
train_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/My Drive/Skripsi/data_mentah/Katarak_Subkapsular/',
    target_size=(150, 150),
    batch_size=10,
    class_mode='categorical',
    subset='training'
)

ator = train_datagen.flow_from_directory(
    '/content/drive/My Drive/Skripsi/data_mentah/Katarak_Subkapsular/',
    target_size=(150, 150),
    batch_size=10,
    class_mode='categorical',
    subset='validation'
)
```



```
# mengekstrak fitur menggunakan model yang telah dilatih
feature_extractor = tf.keras.Model(inputs=model.input, outputs=model.layers[-3].output)

def extract_features(img_path):
    img = image.load_img(img_path, target_size=(150, 150))
    img_data = image.img_to_array(img)
    img_data = np.expand_dims(img_data, axis=0)
    img_data /= 255.0
    features = feature_extractor.predict(img_data)
    return features.flatten()

# Ekstraksi fitur untuk semua gambar
image_directory = '/content/drive/My Drive/Skripsi/data_mentah/Katarak_Subkapsular/'
image_paths = [os.path.join(image_directory, fname) for fname in os.listdir(image_directory)]

features_list = []
for img_path in tqdm(image_paths):
    features = extract_features(img_path)
    features_list.append(features)

features_array = np.array(features_list)
```



b. K-means Clustering

```

from sklearn.cluster import KMeans
from scipy.spatial.distance import euclidean
import pandas as pd
import os
import shutil

# Melakukan K-Means clustering
num_clusters = 3
kmeans = KMeans(n_clusters=num_clusters, random_state=0).fit(features_array)

# Memberikan label cluster pada gambar
cluster_labels = kmeans.labels_
centroids = kmeans.cluster_centers_

# Menghitung jarak Euclidean antara fitur gambar dan centroid cluster
distances = []
for i, features in enumerate(features_array):
    centroid = centroids[cluster_labels[i]]
    distance = euclidean(features, centroid)
    distances.append((distance, image_paths[i]))

# Mengurutkan gambar berdasarkan jarak ke centroid
sorted_distances = sorted(distances, key=lambda x: x[0])
sorted_image_paths = [x[1] for x in sorted_distances]

# Menyimpan koordinat centroid dalam bentuk dataframe
centroid_coordinates = pd.DataFrame(centroids, columns=[f'Feature_{i}' for i in range(centroids.shape[1])])
centroid_coordinates['Cluster'] = [f'Cluster_{i}' for i in range(num_clusters)]

# Menyimpan jarak Euclidean dan label cluster untuk setiap gambar
distance_data = {
    'File_Name': [],
    'Cluster': [],
    'Distance_to_Centroid': []
}

for i, features in enumerate(features_array):
    cluster_label = cluster_labels[i]
    centroid = centroids[cluster_label]
    distance = euclidean(features, centroid)

    image_path = image_paths[i]
    file_name = os.path.basename(image_path) # Mengambil nama file

    distance_data['File_Name'].append(file_name)
    distance_data['Cluster'].append(f'Cluster_{cluster_label}')
    distance_data['Distance_to_Centroid'].append(distance)

distance_df = pd.DataFrame(distance_data)

# Mengurutkan data berdasarkan kolom Distance_to_Centroid
sorted_distance_df = distance_df.sort_values(by='Distance_to_Centroid')

```



```

Google Colab
inates:")
import display
dinates)

a (sorted by Distance_to_Centroid):")
ce_df)

```

```

output_directory_sorted = '/content/drive/My Drive/Katarak/hasil_clustering_sorted/imatur2/'
os.makedirs(output_directory_sorted, exist_ok=True)

# Menyimpan gambar yang telah diurutkan ke dalam folder
for i, img_path in enumerate(sorted_image_paths):
    destination_path = os.path.join(output_directory_sorted, f'sorted_image_{i}.jpg')
    shutil.copy(img_path, destination_path)

```

c. Grayscale, Contour Detection, Masking and Cropping, Resizing

```

import cv2
import numpy as np
from matplotlib import pyplot as plt
import os
from google.colab import drive
from glob import glob

input_folder = '/content/drive/My Drive/Katarak/percobaan3/bahan/imatur/'
output_folder = '/content/drive/My Drive/Katarak/percobaan3/bahan/imatur2/'

def preprocess_image(img):
    gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

    _, binary = cv2.threshold(gray, 10, 255, cv2.THRESH_BINARY)

    contours, _ = cv2.findContours(binary, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    c = max(contours, key=cv2.contourArea)
    x, y, w, h = cv2.boundingRect(c)

    cropped_img = img[y:y+h, x:x+w]

    mask = np.zeros_like(cropped_img)
    (h, w) = mask.shape[:2]
    center = (w // 2, h // 2)
    radius = min(center[0], w - center[0], h - center[1])
    cv2.circle(mask, center, radius, (255, 255, 255), -1)

    result = cv2.bitwise_and(cropped_img, mask)

    return result

# Memproses semua gambar dalam folder input
for img_path in glob(os.path.join(input_folder, '*')):
    if img_path.endswith(('jpg', 'png')):
        img = cv2.imread(img_path)

        img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        processed_img = preprocess_image(img_rgb)

        dim = (512, 512)
        resized_img = cv2.resize(processed_img, dim, interpolation=cv2.INTER_AREA)

        img_bgr = cv2.cvtColor(resized_img, cv2.COLOR_RGB2BGR)

        filename = os.path.basename(img_path)

        output_path = os.path.join(output_folder, filename)
        cv2.imwrite(output_path, img_bgr)

        print(f"Gambar {filename} sudah diproses dan disimpan ke folder output.")

```



- Horizontal Flip

```
import os
from keras.preprocessing.image import ImageDataGenerator
import cv2
import albumentations as A
from matplotlib import pyplot as plt

save_dir = '/content/drive/My Drive/Katarak/percobaan4/bahan/imatur'

if not os.path.exists(save_dir):
    os.makedirs(save_dir, exist_ok=True)

# Mendapatkan daftar file dalam folder
folder_path = '/content/drive/My Drive/Katarak/percobaan1/bahan/imatur/'
file_list = os.listdir(folder_path)

# Definisikan transformasi augmentasi dengan probabilitas 100%
horizontal_flip = A.HorizontalFlip(p=1.0)

# Fungsi untuk menyimpan gambar hasil augmentasi
def save_augmented_image(image, filename, save_dir):
    save_path = os.path.join(save_dir, filename)
    cv2.imwrite(save_path, image)

# Lakukan augmentasi dan simpan gambar
for file_name in file_list:
    file_path = os.path.join(folder_path, file_name)
    image = cv2.imread(file_path)

    # Simpan gambar asli
    save_augmented_image(image, f'{file_name[:-4]}_original.jpg', save_dir)

    # Terapkan horizontal flip
    for i in range(1): # Sesuaikan jumlah augmentasi yang diinginkan
        augmented = horizontal_flip(image=image)
        ug_image = augmented['image']
        ave_augmented_image(ug_image, f'{file_name[:-4]}_hflip_{i}.jpg', save_dir)

print("Augmentasi selesai")
```

- Vertical Flip

```
import os
from keras.preprocessing.image import ImageDataGenerator
import cv2
import albumentations as A
from matplotlib import pyplot as plt

save_dir = '/content/drive/My Drive/Katarak/percobaan4/bahan/imatur'

# Buat direktori jika belum ada
if not os.path.exists(save_dir):
    os.makedirs(save_dir, exist_ok=True)

# Mendapatkan daftar file dalam folder
folder_path = '/content/drive/My Drive/Katarak/percobaan1/bahan/imatur/'
file_list = os.listdir(folder_path)
```



```

# Definisikan transformasi augmentasi dengan probabilitas 100%
vertical_flip = A.VerticalFlip(p=1.0)

# Fungsi untuk menyimpan gambar hasil augmentasi
def save_augmented_image(image, filename, save_dir):
    save_path = os.path.join(save_dir, filename)
    cv2.imwrite(save_path, image)

# Lakukan augmentasi dan simpan gambar
for file_name in file_list:
    file_path = os.path.join(folder_path, file_name)
    image = cv2.imread(file_path)

    # Simpan gambar asli
    save_augmented_image(image, f'{file_name[:-4]}_original.jpg', save_dir)

    # Terapkan vertical flip
    for i in range(1): # Sesuaikan jumlah augmentasi yang diinginkan
        augmented = vertical_flip(image=image)
        aug_image = augmented['image']
        save_augmented_image(aug_image, f'{file_name[:-4]}_vflip_{i}.jpg', save_dir)

print("Augmentasi selesai")

```



e. Training dan Evaluasi Model

```

from google.colab import drive

drive.mount('/content/drive/')
base_dir = '/content/drive/My Drive/Skripsi/percobaan4/'
!ls "/content/drive/My Drive/Skripsi/percobaan4/"
!ls "/content/drive/My Drive/Skripsi/percobaan4/bahan"

import os
bahan_dir = os.path.join(base_dir, 'bahan')
train_dir = os.path.join(base_dir, 'latih')
validation_dir = os.path.join(base_dir, 'validasi')
testing_dir = os.path.join(base_dir, 'testing')

imatur_dir = os.path.join(bahan_dir, 'imatur/')
subkapsular_dir = os.path.join(bahan_dir, 'subkapsular/')
lainnya_dir = os.path.join(bahan_dir, 'lainnya/')
normal_dir = os.path.join(bahan_dir, 'normal/')
bukankatarak_dir = os.path.join(bahan_dir, 'bukan_katarak/')

# mengumpulkan semua file paths dan labelnya
def gather_files_and_labels(source_dirs, labels):
    files = []
    file_labels = []
    for source_dir, label in zip(source_dirs, labels):
        for filename in os.listdir(source_dir):
            files.append(os.path.join(source_dir, filename))
            file_labels.append(label)
    return files, file_labels

source_dirs = [imatur_dir, subkapsular_dir, lainnya_dir, normal_dir, bukannya_dir]
labels = ['imatur', 'subkapsular', 'lainnya', 'normal', 'bukan_katarak']

files, file_labels = gather_files_and_labels(source_dirs, labels)

from sklearn.model_selection import train_test_split
import os
import random
from shutil import copyfile

# Step 1: Pisahkan menjadi data Latih (80%) dan data sementara (20%)
train_files, temp_files, train_labels, temp_labels = train_test_split(files, file_labels, test_size=0.2, stratify=file_labels, random_state=42)

# Step 2: Pisahkan data sementara menjadi data validasi (50% dari 20% = 10%) dan data uji (50% dari 20% = 10%)
val_files, test_files, val_labels, test_labels = train_test_split(temp_files, temp_labels, test_size=0.5, stratify=temp_labels, random_state=42)

def copy_files(file_list, labels_list, dest_dirs):
    for file, label in zip(file_list, labels_list):
        label_dir = os.path.join(dest_dirs, label)
        os.makedirs(label_dir, exist_ok=True)
        dest = os.path.join(label_dir, os.path.basename(file))
        copyfile(file, dest)

# Menyalin file ke direktori Latih, validasi, dan uji
copy_files(train_files, train_labels, train_dir)
copy_files(val_files, val_labels, validation_dir)
copy_files(test_files, test_labels, testing_dir)

def walk_directory(directory):
    for r, d, files in os.walk(directory):

```



```

def count_files_in_dir(directory):
    return sum([len(files) for r, d, files in os.walk(directory)])

print("Jumlah Data Tiap Kelas di Train, Validasi, dan Test:")
print('Jumlah Train katarak imatur:', count_files_in_dir(os.path.join(train_dir, 'imatur')))
print('Jumlah Val katarak imatur:', count_files_in_dir(os.path.join(validation_dir, 'imatur')))
print('Jumlah Test katarak imatur:', count_files_in_dir(os.path.join(testing_dir, 'imatur')))

print('\nJumlah Train katarak subkapsular:', count_files_in_dir(os.path.join(train_dir, 'subkapsular')))
print('Jumlah Val katarak subkapsular:', count_files_in_dir(os.path.join(validation_dir, 'subkapsular')))
print('Jumlah Test katarak subkapsular:', count_files_in_dir(os.path.join(testing_dir, 'subkapsular')))

print('\nJumlah Train katarak lainnya:', count_files_in_dir(os.path.join(train_dir, 'lainnya')))
print('Jumlah Val katarak lainnya:', count_files_in_dir(os.path.join(validation_dir, 'lainnya')))
print('Jumlah Test katarak lainnya:', count_files_in_dir(os.path.join(testing_dir, 'lainnya')))

print('\nJumlah Train normal:', count_files_in_dir(os.path.join(train_dir, 'normal')))
print('Jumlah Val normal:', count_files_in_dir(os.path.join(validation_dir, 'normal')))
print('Jumlah Test normal:', count_files_in_dir(os.path.join(testing_dir, 'normal')))

print('\nJumlah Train bukan katarak:', count_files_in_dir(os.path.join(train_dir, 'bukan_katarak')))
print('Jumlah Val bukan katarak:', count_files_in_dir(os.path.join(validation_dir, 'bukan_katarak')))
print('Jumlah Test bukan katarak:', count_files_in_dir(os.path.join(testing_dir, 'bukan_katarak')))

import time
import tensorflow as tf
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import to_categorical

train_datagen = ImageDataGenerator(
    rescale = 1./255,
    horizontal_flip = True,
    vertical_flip = True,
)

val_datagen = ImageDataGenerator(
    rescale = 1./255,
    horizontal_flip = True,
    vertical_flip = True,
)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size = (150, 150),
    batch_size = 8,
    class_mode = 'categorical',
    shuffle=True
)

val_generator = val_datagen.flow_from_directory(
    validation_dir,
    target_size = (150, 150),
    batch_size = 8,
    class_mode = 'categorical',
    shuffle=False
)

from tensorflow.keras.callbacks import ModelCheckpoint

# Buat callback ModelCheckpoint untuk menyimpan model terbaik
checkpoint = ModelCheckpoint('Model10.keras', monitor='val_accuracy', save_best_only=True, mode='max', verbose=1)

```



```

model = Sequential([
    Conv2D(16, (3, 3), activation='relu', padding='same', input_shape=(150, 150, 3)),
    MaxPooling2D(2, 2),
    BatchNormalization(),
    Dropout(0.5),

    Conv2D(32, (3, 3), activation='relu', padding='same'),
    MaxPooling2D(2, 2),
    BatchNormalization(),
    Dropout(0.5),

    Conv2D(64, (3, 3), activation='relu', padding='same'),
    MaxPooling2D(2, 2),
    BatchNormalization(),
    Dropout(0.5),

    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(5, activation='softmax')
])

learning_rate = 0.0001
optimizer = Adam(learning_rate=learning_rate)

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

model.summary()

# Catat waktu mulai
start_time = time.time()
history = model.fit(
    train_generator,
    steps_per_epoch=steps_per_epoch,
    epochs = 10,
    validation_data = val_generator,
    validation_steps=validation_steps,
    verbose = 1,
    callbacks = [checkpoint]
)

# Catat waktu selesai
end_time = time.time()

# Hitung lama waktu pelatihan
training_duration = end_time - start_time
print(f"Lama waktu pelatihan: {training_duration // 60:.0f} menit {training_duration % 60:.0f} detik")

# Gabungkan history dan waktu pelatihan
# history_data = f'
ry.history,
art_time,
time,
on': training_duration

kle
tory.pkl', 'wb') as f:
ory_data, f)

```



```
import matplotlib.pyplot as plt

# Membaca file pickle
with open('training_history.pkl', 'rb') as file:
    history = pickle.load(file)

# Mendapatkan data dari history
acc = history['accuracy']
val_acc = history['val_accuracy']
loss = history['loss']
val_loss = history['val_loss']

# Membuat plot untuk accuracy
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

# Membuat plot untuk loss
plt.subplot(1, 2, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

# Menampilkan plot
plt.show()
```



```

from tensorflow.keras.models import load_model

# Muat model yang telah disimpan
model = load_model('/content/model_fix.h5')

from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Data generator untuk data testing
test_datagen = ImageDataGenerator(rescale=1./255)
test_generator = test_datagen.flow_from_directory(
    testing_dir,
    target_size=(150, 150),
    batch_size=8,
    class_mode='categorical',
    shuffle=False
)

# Pastikan generator direset
test_generator.reset()

# Evaluasi model untuk mendapatkan loss dan accuracy
test_loss, test_accuracy = model.evaluate(test_generator, verbose=1)

# Tampilkan nilai test loss dan test accuracy dalam bentuk float
print(f"Test Loss: {test_loss:.4f}")
print(f"Test Accuracy: {test_accuracy:.4f}")

```

```

from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Pastikan generator direset
test_generator.reset()

# Prediksi dengan model yang telah dimuat
test_predictions = model.predict(test_generator, verbose=1)

# Ambil Label sebenarnya
test_true = test_generator.classes
# Ambil Label prediksi
test_pred = test_predictions.argmax(axis=1)

# Hitung classification report
report = classification_report(test_true, test_pred, target_names=test_generator.class_indices.keys())
print('Classification Report')
print(report)

# Hitung confusion matrix
cm = confusion_matrix(test_true, test_pred)

# Plot confusion matrix
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=test_generator.class_indices.keys(), yticklabels=test_generator.class_indices.keys())
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix - Testing Data')
plt.show()

```



f. Implementasi Model ke Website

• Frontend

```

1  import React, { useState } from 'react';
2  import './App.css';
3
4  function App() {
5    const [file, setFile] = useState(null);
6    const [confidenceScores, setConfidenceScores] = useState(null);
7    const [processedImageUrl, setProcessedImageUrl] = useState(null);
8    const [predictedLabel, setPredictedLabel] = useState(null);
9
10   const handleFileChange = (event) => {
11     setFile(event.target.files[0]);
12   };
13
14   const handleSubmit = async (event) => {
15     event.preventDefault();
16     const formData = new FormData();
17     formData.append('file', file);
18
19     try {
20       const response = await fetch('http://localhost:5000/upload', {
21         method: 'POST',
22         body: formData,
23       });
24
25       const data = await response.json();
26       console.log(data); // Debug logging
27       setConfidenceScores(data.confidence_scores);
28       setProcessedImageUrl(`http://localhost:5000/processed/${data.processed_filename}`);
29       setPredictedLabel(data.prediction_label);
30     } catch (error) {
31       console.error('Error:', error);
32     }
33   };
34

```



```

35     return (
36         <div className="container">
37             <header>
38                 <h2>CataractDetect</h2>
39             </header>
40             <main>
41                 <h3>Masukkan Gambar Fundus Mata yang Ingin Diprediksi</h3>
42                 <form onSubmit={handleSubmit}>
43                     <div className="upload-box">
44                         <input type="file" name="file" id="file-input" accept="image/*" onChange={handleFileChange} required />
45                         <label htmlFor="file-input">Pilih Gambar</label>
46                         <p id="file-name" className="file-name">{file ? file.name : 'Tidak ada gambar yang dipilih'}</p>
47                     </div>
48                     <div className="button-container">
49                         <button type="submit">Prediksi</button>
50                     </div>
51                 </form>
52                 {processedImageUrl && (
53                     <div className="result">
54                         <div className="image-container">
55                             <img src={processedImageUrl} alt="Processed" />
56                         </div>
57                         <div className="prediksi">
58                             {predictedLabel && (
59                                 <h2>Prediksi: {predictedLabel} === "Confidence score sangat rendah"
60                                 ? "Tidak dapat diprediksi karena confidence score sangat rendah"
61                                 : predictedLabel
62                             )}
63                         </div>
64                     <h3>Confidence Scores:</h3>
65                     <div className="confidence-scores">
66                         {confidenceScores &&
67                             Object.entries(confidenceScores).map(([label, score]) => (
68                                 <p key={label}>
69                                     {label}: {score}%
70                                 </p>
71                             ))
72                         }
73                     </div>
74                 </div>
75             </main>
76         </div>
77     );
78 }
79 };
80 }
81
82 export default App;

```

- Backend Dockerfile

```

1  # Gunakan image tensorflow yang sesuai
2  FROM tensorflow/tensorflow:2.15.0
3
4  # Tentukan direktori kerja di dalam container
5  WORKDIR /app
6
7  # Salin semua file dari direktori proyek ke direktori kerja di dalam container
8  COPY . /app
9
10 # Set up alat build yang diperlukan
11 RUN apt-get update && apt-get install -y build-essential
12
13 # Instal versi numpy yang kompatibel dengan TensorFlow
14 RUN pip install numpy==1.24.3
15
16 # Instal flask tanpa dependensi
17 RUN pip install flask==2.0.1 Jinja2==3.0.1 itsdangerous==2.0.1 click==8.0.1 Werkzeug==2.0.1 flask-cors --timeout=100 --retries=10
18
19 # Instal Pillow
20 RUN pip install pillow --timeout=100 --retries=10
21
22 # Instal OpenCV
23 RUN pip install opencv-python-headless --timeout=100 --retries=10
24
25 # Tentukan port yang akan diekspose oleh container

```



ntah untuk menjalankan aplikasi Flask
"app.py"]

app.py

```

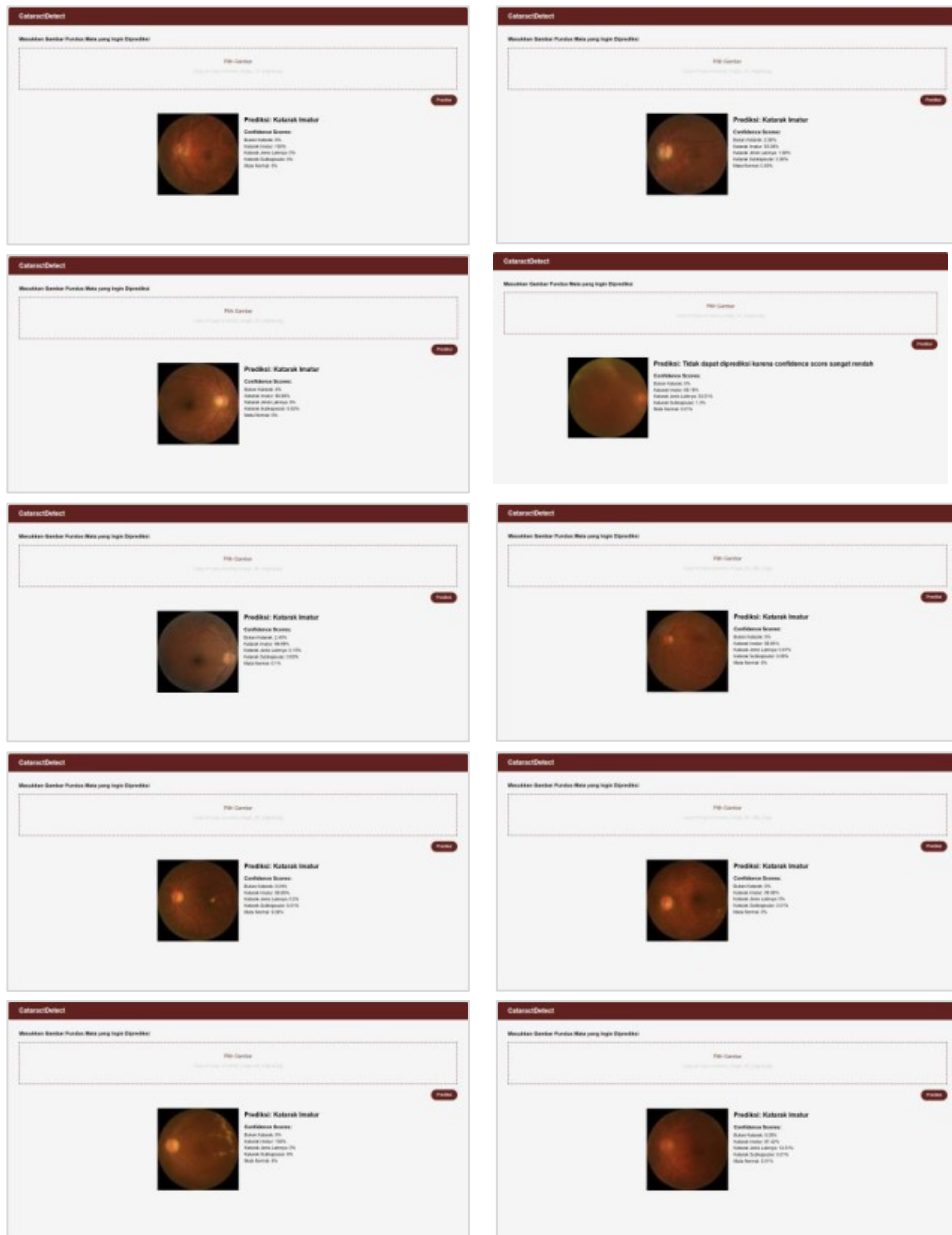
1  from flask import Flask, request, jsonify, send_from_directory, render_template
2  import os
3  import numpy as np
4  from PIL import Image
5  import cv2
6  from keras.models import load_model
7  from keras.preprocessing.image import img_to_array
8  from flask_cors import CORS
9  import logging
10
11  app = Flask(__name__)
12  CORS(app)
13
14  UPLOAD_FOLDER = 'static/uploads'
15  PROCESSED_FOLDER = 'static/processed'
16  app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
17  app.config['PROCESSED_FOLDER'] = PROCESSED_FOLDER
18
19  model = load_model('model1 (1).keras')
20  model.trainable = False
21  CONFIDENCE_THRESHOLD = 70.0
22
23  @app.route('/')
24  def home():
25      return render_template('index.html')
26
27  @app.route('/upload', methods=['POST'])
28  def upload_file():
29      if 'file' not in request.files:
30          return jsonify({"error": "No file part"})
31      file = request.files['file']
32      if file.filename == '':
33          return jsonify({"error": "No selected file"})
34
35      filepath = os.path.join(app.config['UPLOAD_FOLDER'], file.filename)
36      file.save(filepath)
37      logging.info(f'File saved to {filepath}')
38
39      img = Image.open(filepath)
40      img = img.convert('RGB')
41      img = img.resize((150, 150), Image.Resampling.LANCZOS)
42
43      img_array = img_to_array(img)
44      img_array = img_array / 255.0 # Normalisasi
45
46      image = np.expand_dims(img_array, axis=0)
47
48      logging.info(f'Preprocessed image shape: {image.shape}')
49      logging.info(f'Image data type: {image.dtype}')
50      logging.info(f'Image min value: {np.min(image)}, max value: {np.max(image)}')
51
52      processed_filepath = os.path.join(app.config['PROCESSED_FOLDER'], file.filename)
53      img.save(processed_filepath)
54      logging.info(f'Processed file saved to {processed_filepath}')
55
56      labels = ['Bukan Katarak', 'Katarak Imatur', 'Katarak Jenis Lainnya', 'Mata Normal', 'Katarak Subkapsular']
57      prediction = model.predict(image)
58      confidence_scores = {label: round(float(score) * 100, 2) for label, score in zip(labels, prediction[0])}
59
60      max_label = max(confidence_scores, key=confidence_scores.get)
61      max_confidence = confidence_scores[max_label]
62      prediction_label = max_label if max_confidence >= CONFIDENCE_THRESHOLD else "Confidence score sangat rendah"
63
64      logging.info(f'Prediction: {confidence_scores}')
65
66      return jsonify({'confidence_scores': confidence_scores, 'processed_filename': file.filename, 'prediction_label': prediction_label})
67
68  @app.route('/processed/<filename>')
69  def send_processed_file(filename):
70      return send_from_directory(app.config['PROCESSED_FOLDER'], filename)
71
72  if __name__ == '__main__':
73      logging.basicConfig(level=logging.INFO)
74      app.run(debug=True, host='0.0.0.0')

```

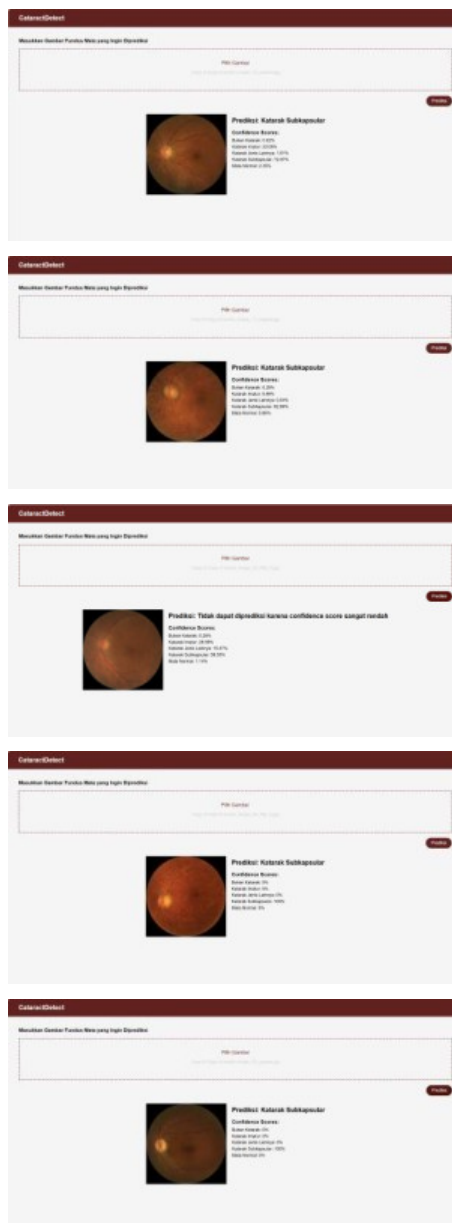
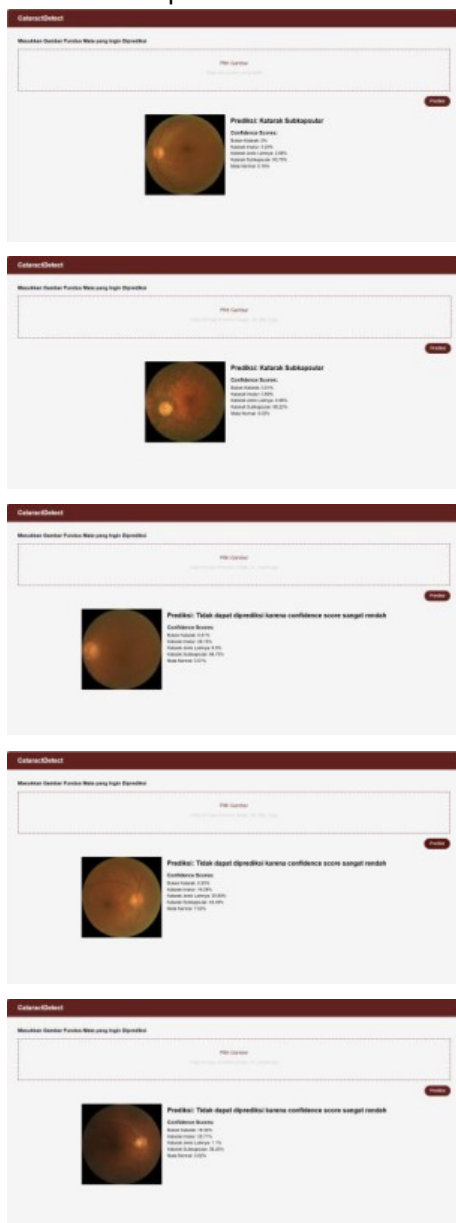


Lampiran 3. Testing Website CataracDetect (60 Citra)

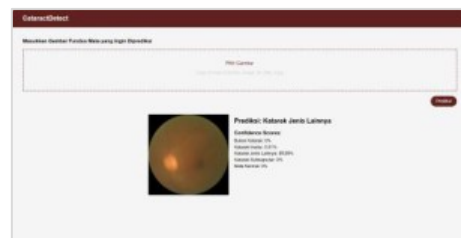
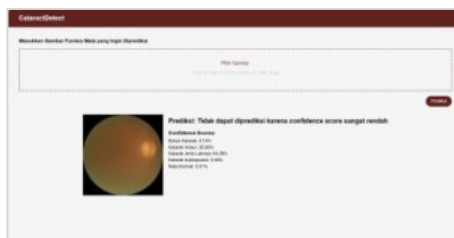
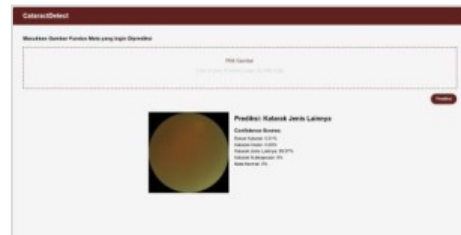
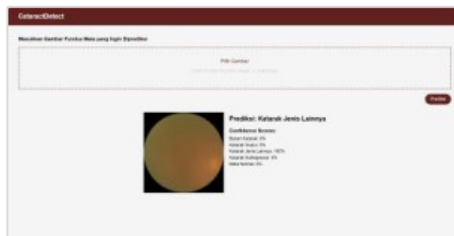
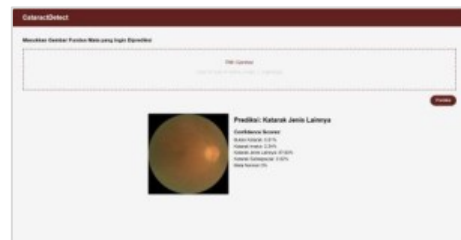
a. Katarak Imatur

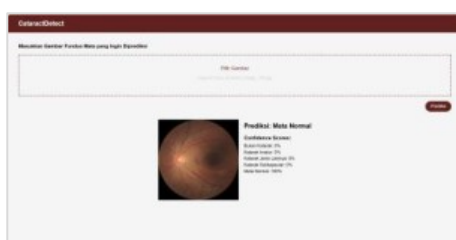


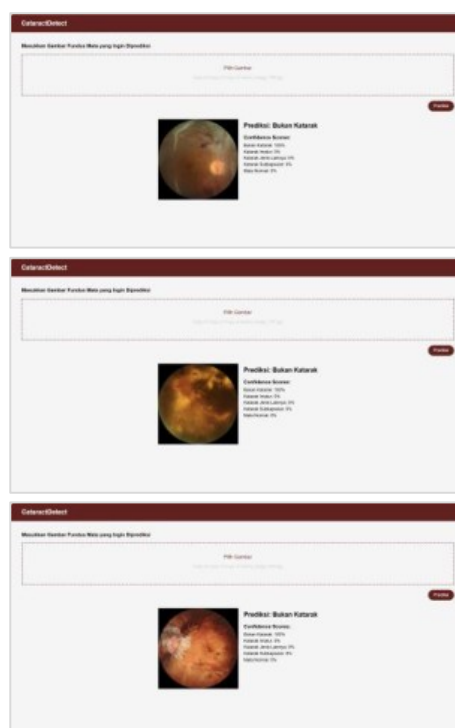
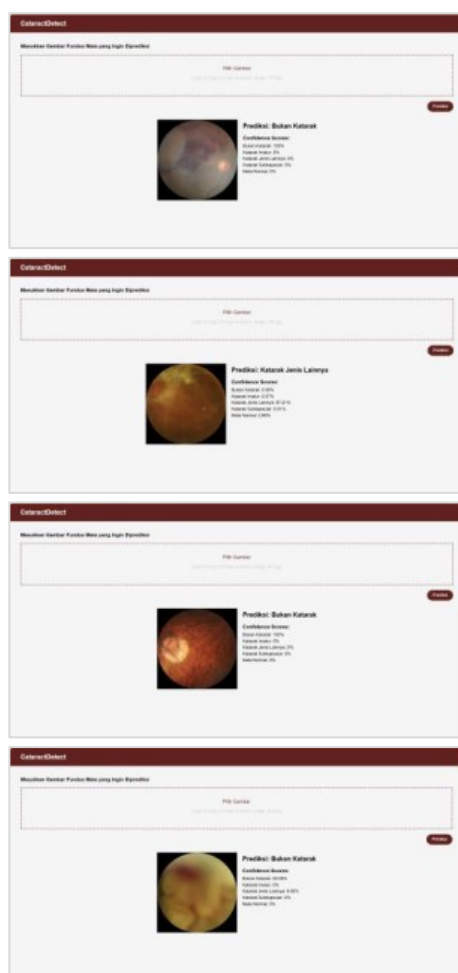
b. Katarak Subkapsular



c. Katarak Lainnya







Lampiran 4. Daftar hadir dan berita acara seminar hasil



**KEMENTERIAN PENDIDIKAN , KEBUDAYAAN
 RISET DAN TEKNOLOGI
 UNIVERSITAS HASANUDDIN
 FAKULTAS TEKNIK
 DEPARTEMEN TEKNIK INFORMATIKA**
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://cog.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR SEMINAR HASIL

Nama/Stambuk : I.Thesalonika Aprilia D121201066

Judul Skripsi/T.A : "Klasifikasi Citra Fundus Penyakit Katarak Menggunakan Convolutional Neural Network Berbasis Website"

Hari/Tanggal : Rabu, 28 Agustus 2024
 Jam : 08- 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.
II.	Anggota Penguji	2. Dr.Eng. Ady Wahyudi Paundu,ST.,MT	2.
		3. A.Ais Prayogi Alimuddin,ST.,M.Eng	3.

PANITIA UJIAN

Ketua,


 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://cng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA SEMINAR HASIL

Pada hari ini **Rabu**, tanggal 28 Agustus 2024 Pukul **08.00 WITA** - Selesai bertempat di **Ruang Lab. AIMP** Departemen Teknik Informatika Gowa, telah dilaksanakan Seminar Hasil bagi Saudara :

Nama : Thesalonika Aprilia
No. Stambu : D121201066
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : "Klasifikasi Citra Fundus Penyakit Katarak Menggunakan Convolutional Neural Network Berbasis Website"

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.	Dr. Eng. Ady Wahyudi Paundu, ST., MT	Anggota	2.
3.	A. Ais Prayogi Alimuddin, ST., M. Eng	Anggota	3.

Hasil keputusan Tim Penguji Seminar Hasil : **Lulus / Tidak lulus** dengan nilai angka **8.5**
dan huruf **A**

Gowa 28 Agustus 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 Matene, Gowa
<http://cng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

Nomor : 1275/UN4.7.7/TD.06/2024
Lamp : -
Hal : Penerbitan Surat Penugasan Panitia/Penguji
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/Penguji Seminar Hasil Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik tersebut di bawah ini :

Nama / Stambuk : Thesalonika Aprilia D121201066
Judul TA : Klasifikasi Citra Fundus Penyakit Katarak Menggunakan
Convolutional Neural Network Berbasis Website

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.
Penguji / Anggota : 2. Dr.Eng. Ady Wahyudi Paundu, ST., M.T.
: 3. A. Ais Prayogi Alimuddin, ST., M.Eng.

Untuk dapat diterbitkan surat penugasannya

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

Gowa, 26 Agustus 2024

Ketua Departemen Tek.Informatika,



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

Tembusan :

1. Arsip



Attn, 28 Agustus 2024

08.00

Arip





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

Jalan Poros Malino Km. 6 Bontomarannu, Gowa, 92171, Sulawesi Selatan
☎ +62811 4420 909, E-mail: teknik@unhas.ac.id, <https://eng.unhas.ac.id>

SURAT PENUGASAN
No. 20859/UN4.7.1/TD.06/2024

- Dari : Dekan Fakultas Teknik Universitas Hasanuddin
- Kepada : Mereka yang tercantum namanya dibawah 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 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.
- Penguji / Anggota : 2. Dr.Eng. Ady Wahyudi Paundu, ST., M.T.
- : 3. A. Ais Prayogi Alimuddin, ST., M.Eng.

Untuk menguji bagi mahasiswa tersebut dibawah ini :

- Nama/NIM : Thesalonika Aprilia D121201066
- Program Studi : Teknik Informatika
- Judul thesis/Skripsi : Klasifikasi Citra Fundus Penyakit Katarak Menggunakan Convolutional Neural Network Berbasis Website
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 dikemudian hari terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa
Pada tanggal 26 Agustus 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. Mahasiswa yang bersangkutan



umen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSrE
ITE No 11 Tahun 2008 Pasal 5 Ayat 1
menor Elektronik dan/atau Dokumen Elektronik dan/atau hasil pemrosesan elektronik alat bukti hukum yang sah

Lampiran 5. Daftar hadir dan berita acara ujian skripsi



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. Thesalonika Aprilia D121201066

Judul Skripsi/T.A : “ **Klasifikasi Citra Fundus Penyakit Katarak Menggunakan Convolutional Neural Network Berbasis Website**”

Hari/Tanggal : Rabu, 2 Oktober 2024

Jam : 13- 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.
II.	Anggota Penguji	2. Dr. Eng. Ady Wahyudi Paundu, ST., M.T	2.
		3. A. Ais Prayogi Alimuddin, ST., M.Eng	3.

PANITIA UJIAN

Ketua,

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

BERITA ACARA UJIAN SKRIPSI

Pada hari ini Rabu, tanggal 2 Oktober 2024 Pukul 13.00 WITA - Selesai bertempat di Lab. AIMP Departemen Teknik Informatika Gowa, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Thesalonika Aprilia
No. Stambuk : D121201066
Fakultas/Departemen : Teknik /Teknik Informatika
Judul Skripsi : " Klasifikasi Citra Fundus Penyakit Katarak Menggunakan Convolutional Neural Network Berbasis Website"

Yang dihadiri oleh Tim Penguji Ujian Skripsi sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr. Ir. Ingrid Nurtanio, M.T	Pemb I/Ketua	1...
2.	Dr. Eng. Ady Wahyudi Paundu, ST., M.T	Anggota	2...
3.	A. Ais Prayogi Alimuddin, ST., M.Eng	Anggota	3...

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

Gowa, 2 Oktober 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, 30 September 2024

Nomor : 1488/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 / Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T
Penguji / Anggota : 2. Dr.Eng. Ady Wahyudi Paundu, ST., M.T
3. A. Ais Prayogi Alimuddin, ST., M.Eng

Untuk Bertugas sebagai Penguji/ Penanggap Ujian Sarjana bagi Mahasiswa :

Nama : Thesalonika Aprilia
Stambuk : D121 20 1066

Dengan Judul Skripsi :

“ Klasifikasi Citra Fundus Penyakit Katarak Menggunakan Convolutional Neural
Network Berbasis Website “

Pada :
Hari/Tanggal : Rabu, 2 Oktober 2024
Jam : 13.00 Wita - Selesai
Tempat : Ruang Sidang Lab. AIMP

Demikian penyampaian kami, atas perhatiannya diucapkan terimakasih.

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**

Jalan Poros Malino Km. 6 Bontomarannu, Gowa, 92171, Sulawesi Selatan
☎ +62811 4420 909, E-mail: teknik@unhas.ac.id , <https://eng.unhas.ac.id>

SURAT PENUGASAN

No. 24310/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 / Ketua : 1. Dr. Ir. Ingrid Nurtanio, M.T
Penguji / Anggota : 2. Dr.Eng. Ady Wahyudi Paunda, ST., M.T
3. A. Ais Prayogi Alimuddin, ST., M.Eng

untuk menguji bagi mahasiswa tersebut di bawah ini :

- | | |
|----------------------|---|
| Nama/NIM | : Thesalonika Aprilia D121201066 |
| Program Studi | : Teknik Informatika |
| Judul Thesis/Skripsi | : Klasifikasi Citra Fundus Penyakit Katarak Menggunakan Convolutional Neural Network Berbasis Website |
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 30 September 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



okumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSR E
IU ITE No 11 Tahun 2008 Pasal 5 Ayat 1
Pemerintah Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah

Lampiran 6. Lembar perbaikan skripsi

LEMBAR PERBAIKAN SKRIPSI

KLASIFIKASI CITRA FUNDUS PENYAKIT KATARAK MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK BERBASIS WEBSITE OLEH:

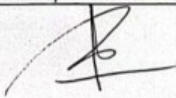
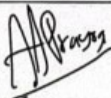
Thesalonika Aprilla

D121 20 1066

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 02 Oktober 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.	
Anggota	Dr. Eng. Ady W. Paundu, S. T., M. T	
	A. Ais Prayogi Alimuddin, S.T., M.Eng.	

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Ir. Ingrid Nurtanio, M.T.	

