

## MANUAL BOOK

# APLIKASI PENDETEKSI MASKER MENGGUNAKAN *HAAR CASCADE* DAN *CONVOLUTIONAL NEURAL NETWORK* UNTUK MEMBERI PERINGATAN MEMATUHI PROTOKOL KESEHATAN



Oleh:

NURFAIDAH

NIM.1841720134

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2022**

**Daftar Isi.....**

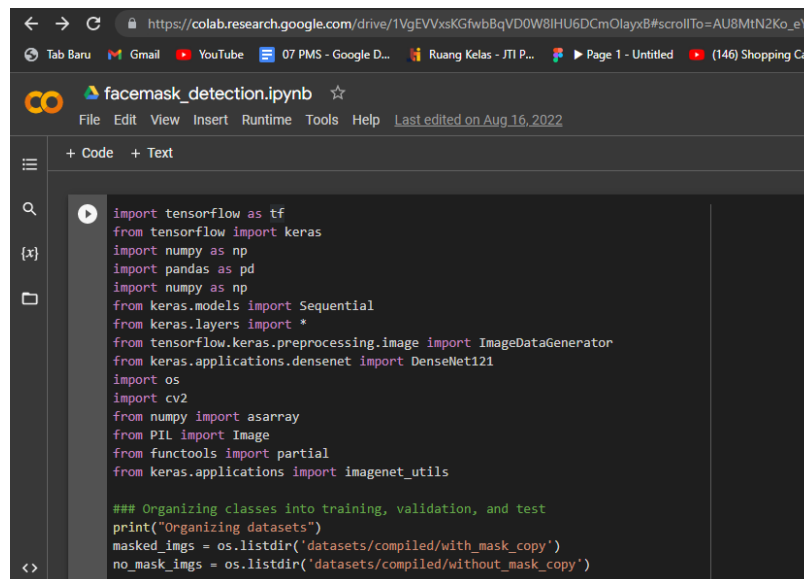
## Petunjuk Penggunaan Aplikasi Deteksi Masker

Aplikasi Pendeteksi Masker adalah aplikasi yang bertujuan untuk mendeteksi apakah seseorang memakai masker atau tidak memakai masker berguna untuk mematuhi protokol kesehatan untuk terhindar dari virus Covid-19.

Berikut adalah langkah-langkah untuk mengakses aplikasi Pendeteksi Masker:

### A. Masuk ke Website Google Colaboratory

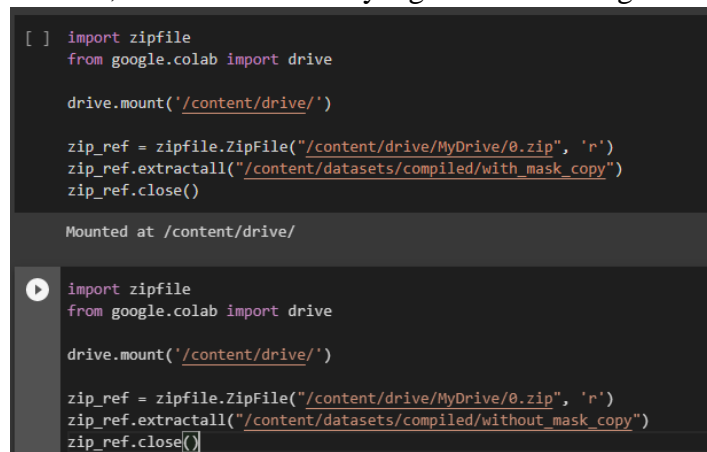
- Untuk mengakses Aplikasi Pendeteksi Masker silahkan Buka aplikasi browser yang ada pada computer anda.
- Akses website dengan copy dan paste alamat url <https://colab.research.google.com/drive/1VgEvvxsKGfwbBqVD0W8IHU6DCmOlayxB?usp=sharing>
- Setelah akses berhasil, akan muncul barisan code yang tersedia seperti dibawah ini



```
import tensorflow as tf
from tensorflow import keras
import numpy as np
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import *
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.applications.densenet import DenseNet121
import os
import cv2
from numpy import asarray
from PIL import Image
from functools import partial
from keras.applications import imagenet_utils

### Organizing classes into training, validation, and test
print("Organizing datasets")
masked_imgs = os.listdir('datasets/compiled/with_mask_copy')
no_mask_imgs = os.listdir('datasets/compiled/without_mask_copy')
```

- Selanjutnya ekstrak file zip untuk compile dataset pada google drive yang sudah tersedia, berikut baris code yang akan di running



```
[ ] import zipfile
from google.colab import drive

drive.mount('/content/drive/')

zip_ref = zipfile.ZipFile("/content/drive/MyDrive/0.zip", 'r')
zip_ref.extractall("/content/datasets/compiled/with_mask_copy")
zip_ref.close()

Mounted at /content/drive/

import zipfile
from google.colab import drive

drive.mount('/content/drive/')

zip_ref = zipfile.ZipFile("/content/drive/MyDrive/0.zip", 'r')
zip_ref.extractall("/content/datasets/compiled/without_mask_copy")
zip_ref.close()
```

Lakukan ekstrak file sampai file zip bisa diakses dan bisa di running.

Link drive file zip untuk di ekstrak:

- [https://drive.google.com/file/d/1kZ0yNau0NVrV-d6R3zVod42MMjdYYaAB/view?usp=share\\_link](https://drive.google.com/file/d/1kZ0yNau0NVrV-d6R3zVod42MMjdYYaAB/view?usp=share_link)
  - [https://drive.google.com/file/d/1Pnqjei\\_jQege5nD-rvm2-sfnFnrtSzND/view?usp=share\\_link](https://drive.google.com/file/d/1Pnqjei_jQege5nD-rvm2-sfnFnrtSzND/view?usp=share_link)
  - [https://drive.google.com/file/d/1Pnqjei\\_jQege5nD-rvm2-sfnFnrtSzND/view?usp=share\\_link](https://drive.google.com/file/d/1Pnqjei_jQege5nD-rvm2-sfnFnrtSzND/view?usp=share_link)
- Setelah semua file selesai di ekstrak, lakukan running code pada baris code program pertama, untuk mendapatkan data untuk di training.
  - Selanjutnya running code program kedua untuk modeling data, modeling data akan memakan waktu yang lama, jadi harap agar laptop selalu menyala dan tidak kehabisan daya.

```
densenet.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.01),
                history = densenet.fit_generator(
                    train_generator,
                    epochs=epochs,
                    validation_data=validation_generator
                )

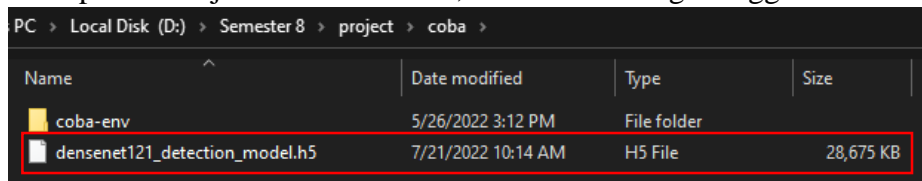
                densenet.save('densenet121_detection_model.h5')

                ### Metrics
                # Testing directories
                test_mask_imgs = os.listdir('datasets/compiled/test/mask')
                test_no_mask_imgs = os.listdir('datasets/compiled/test/no-mask')

                # Numbers of images
                total_mask = len(test_mask_imgs)
                total_no_mask = len(test_no_mask_imgs)
                total = total_mask + total_no_mask

                # Confusion Matrix variables
                mask_correct = 0
                mask_incorrect = 0
                no_mask_correct = 0
                no_mask_incorrect = 0
```

- Setelah proses modeling data selesai, akan muncul file modeling yang siap untuk disimpan dan dijadikan model data, untuk di training menggunakan kamera.

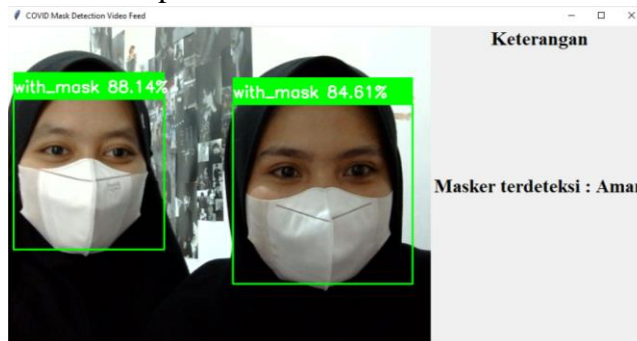


## B. Buka Visual Studio Code

- Unduh file proyek  
[https://drive.google.com/drive/folders/1i0YWLgEx8Q3PD07uaRtB7BF\\_bnUeTtMs?usp=share\\_link](https://drive.google.com/drive/folders/1i0YWLgEx8Q3PD07uaRtB7BF_bnUeTtMs?usp=share_link)  
Lalu buka visual studio code
- Jalankan file GUI.py

```
File Edit Selection View Go Run Terminal Help GUI.py - coba2 - V
GUI.py x coba.py
GUI.py
1 import os
2 import tkinter
3 import cv2
4 from tkinter import *
5 import numpy as np
6 from PIL import ImageTk, Image
7 from tensorflow import keras
8 from keras.models import Sequential
9 from keras.layers import *
10
11 load_model = keras.models.load_model('densenet121_detection_model.h5')
12 face_clsfr=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
13
14 labels_dict={0:'with_mask', 1:'without_mask'}
15 color_dict={0:(0,255,0),1:(0,0,255)}
16
17 size = 4
18 webcam = cv2.VideoCapture(0)
19 classifier = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
20
21 main_window = tkinter.Tk()
22
23 main_window.title("COVID Mask Detection Video Feed")
24 main_window.grid_rowconfigure(0, weight=1)
25 main_window.grid_columnconfigure(0, weight=1)
```

- Tunggu beberapa detik, lalu kamera pada laptop akan terbuka secara otomatis, pada saat kamera menyala, sistem aplikasi akan langsung bekerja.  
Contoh tampilan sistem deteksi:



Catatan: Jika terdapat error pada saat running code program, pastikan semua library yang dibutuhkan sudah terinstall dengan benar.

**Selesai.**