

APPENDIX

LIBRARY HAARCASCADE

```
face_cascade =
cv2.CascadeClassifier('data\\xml\\haarcascade_frontalface_defa
ult.xml')
eye_cascade =
cv2.CascadeClassifier('data\\xml\\haarcascade_eye.xml')
nose_cascade = cv2.CascadeClassifier('data\\xml\\Nariz.xml')
mouth_cascade =
cv2.CascadeClassifier('data\\xml\\haarcascade_mcs_mouth.xml')
```

CODING CONVERT GRAYSCALE DAN BLACK AND WHITE

```
count = 0
bw_threshold = 80 // mengatur threshold value range
cap = cv2.VideoCapture('Foto\\data\\video.mp4') //read video

while True:
    ret, img = cap.read()
    img = cv2.flip(img,1)

    # Convert Image ke gray
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Convert image ke Black and White
    (thresh, black_and_white) = cv2.threshold(gray,
bw_threshold, 255, cv2.THRESH_BINARY)
```

CODING DETECT FACE, EYES, NOSE, MOUTH

```
# detect wajah
```

```

faces = face_cascade.detectMultiScale(gray, 1.1, 4)

# Prediksi wajah untuk hitam dan putih
faces_bw = face_cascade.detectMultiScale(black_and_white,
1.1, 4)

# Gambar kotak di wajah
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 255,
255), 2)

    roi_gray = gray[y:y + h, x:x + w]
    roi_color = img[y:y + h, x:x + w]

    #deteksi mata
    eye = eye_cascade.detectMultiScale(roi_gray)
    for (ex, ey, ew, eh) in eye:
        cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey +
eh), (0, 255, 0), 2)

    #deteksi hidung
    nose = nose_cascade.detectMultiScale(gray, 1.15, 35)

    # deteksi mulut
    mouth_rects = mouth_cascade.detectMultiScale(gray,
1.18, 35)

```

CODING DETERMINING USE OF MASK

```

if (len (nose) == 0 and len(mouth_rects) == 0 ):
    cv2.putText(img, 'Mask On', (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 2,(0, 255, 0), 2)

elif (len (nose) > 0 and len (mouth_rects) == 0 ):
    for (nx, ny, nw, nh) in nose:
        cv2.rectangle(img, (nx, ny), (nx + nw, ny
+ nh), (255, 0, 0),2)
        cv2.putText(img, 'Wrong', (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 2,(0, 0, 255),
2)

        print("Image "+str(count)+"saved")
        file='C:/Python/Python38/Foto/
Simpan'+str(count)+'.jpg'
        cv2.imwrite(file, img)
        count +=1
        break

    else :
        for (mx, my, mw, mh) in mouth_rects:
            if(y < my < y + h):
                cv2.rectangle(img, (mx, my), (mx + mh,
my + mw), (0, 0, 255), 2)
                break

        for (nx, ny, nw, nh) in nose:
            if(y < ny < y + h):
                cv2.rectangle(img, (nx, ny), (nx + nh,
ny + nw), (255, 0, 0), 2)
                break
            cv2.putText(img, 'Mask Off', (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 2,(0, 255,
0), 2)

```

```
print("Image "+str(count)+"saved")
file='C:/Python/Python38/Foto/
Simpan'+str(count)+'.jpg'
cv2.imwrite(file, img)
count +=1
break
```

```
cv2.imshow('Mask Detection', img)
k = cv2.waitKey(1)
if k == 27:
break
```

```
# Release video
cap.release()
cv2.destroyAllWindows()
```





2.3% PLAGIARISM
APPROXIMATELY

Report #14314667

INTRODUCTION Background Masks are items that cannot be separated from everyday life. Especially during this pandemic, mask must be used when leaving the house or doing activities outside the home. However, there are still many people who underestimate the use of masks. Especially in the office, many employees sometimes don't wear masks according to the rules when in the office area. Even other people who see it sometimes don't care. The mask detection system in the office can detect employees who use masks, don't use masks, and the wrong way to wear masks. If the employee is wearing the mask correctly, the system can detect if the mask is wearing correctly. If you don't wear a mask or use mask incorrectly, the system can detect if you are not wearing a mask correctly and the employee's face will be captured and stored in a computer file. The system for detecting the use of masks in the office was created to monitor the use of masks in the office. It is hoped that with this system