



Lampiran 1. Daftar program pengambilan referensi

```
import cv2

import sys

from gpiozero import LED

import imutils

#GUI LIBRARY

from tkinter import *

import tkinter.font

###GUI

win = Tk()

win.title("Start")

myFont = tkinter.font.Font(family = 'Helvetica', size = 9, weight = "bold")

def close():

    led.off()

    led2.off()

    win.destroy()

    cv2.destroyAllWindows()

    sys.exit()

def input_ref1():

    scale = 10

    cap=cv2.VideoCapture(0)
```

```

cap.set(cv2.CAP_PROP_FPS,60)

gambarr=5 # membuat variable untuk penamaan file yang akan ditulis nanti

global led2, led

led = LED(12)

led2 = LED(21)

led.on()

led2.on()

while True:
    _,img=cap.read()
    x, y, h, w = (100, 75, 250, 450)
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 20)
    cv2.namedWindow('tekan s untuk capture', cv2.WINDOW_NORMAL)
    cv2.resizeWindow('tekan s untuk capture', 400, 300)
    img_resize = img[50:350, 75:575]
    cv2.imshow("tekan s untuk capture",img_resize)
    k=cv2.waitKey(1) & 0xFF
    if k==27:
        cap.release()
        cv2.destroyAllWindows()
    if k==ord('s'):# jika tombol 's' kecil ditekan
        fileN='ref_1.jpg'# membuat string nama image yang disimpan
        cv2.imwrite(fileN,img_resize)# simpan image di folder yang aktif sekarang
        print (gambarr)# menampilkan nama image yang telah tersimpan di command
        prompt(CMD) / console terminal

        cv2.destroyAllWindows()

```

```
led.off()

led2.off()

text = Label(text="NAMA", font=myFont)

text.grid(row=3,column=0)

ment = StringVar()

mEntry = Entry(win,textvariable=ment)

mEntry.grid(row=4, column=0)

def ok():

    global nama

    nama = ment.get()

    file = open("referensi1","r+")

    file.write(nama)

    file.write("\n")

    file.close()

    close()

mButton = Button(win, text='OK', font=myFont, command=ok, bg='red', height=1,
width=1)

mButton.grid(row=4, column=1)

win.geometry("300x200")

win.mainloop()

cv2.destroyAllWindows()

break
```

```

def input_ref2():

    cap=cv2.VideoCapture(0)

    cap.set(cv2.CAP_PROP_FPS,60)

    gambarr=6 # membuat variable untuk penamaan file yang akan ditulis nanti

    global led2, led

    led = LED(12)

    led2 = LED(21)

    led.on()

    led2.on()

    while True:

        _,img=cap.read()

        x, y, h, w = (100, 75, 250, 450)

        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 20)

        cv2.namedWindow('tekan s untuk capture', cv2.WINDOW_NORMAL)

        cv2.resizeWindow('tekan s untuk capture', 400, 300)

        img_resize = img[50:350, 75:575]

        cv2.imshow("tekan s untuk capture",img_resize)

        k=cv2.waitKey(1) & 0xFF

        if k==27:

            cap.release()

            cv2.destroyAllWindows()

        if k==ord('s'):# jika tombol 's' kecil ditekan

            fileN='ref_2.jpg'# membuat string nama image yang disimpan

            cv2.imwrite(fileN,img_resize)# simpan image di folder yang aktif sekarang

            print (gambarr)# menampilkan nama image yang telah tersimpan di command
prompt(CMD) / console terminal

```

```
gambarr = gambarr+1

cv2.destroyAllWindows()

led.off()

led2.off()

text = Label(text="NAMA", font=myFont)

text.grid(row=3,column=0)

ment = StringVar()

mEntry = Entry(win,textvariable=ment)

mEntry.grid(row=4, column=0)

def ok():

    global nama

    nama = ment.get()

    file = open("referensi2","r+")

    file.write(nama)

    file.write("\n")

    file.close()

close()

mButton = Button(win, text='OK', font=myFont, command=ok, bg='red', height=1,
width=1)

mButton.grid(row=4, column=1)

win.geometry("300x200")

win.mainloop()
```

```
cv2.destroyAllWindows()
```

```
break
```

```
def input_ref3():
```

```
cap=cv2.VideoCapture(0)
```

```
cap.set(cv2.CAP_PROP_FPS,60)
```

```
gambarr=7 # membuat variable untuk penamaan file yang akan ditulis nanti
```

```
global led2, led
```

```
led = LED(12)
```

```
led2 = LED(21)
```

```
led.on()
```

```
led2.on()
```

```
while True:
```

```
    _,img=cap.read()
```

```
    x, y, h, w = (100, 75, 250, 450)
```

```
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 20)
```

```
    cv2.namedWindow('tekan s untuk capture', cv2.WINDOW_NORMAL)
```

```
    cv2.resizeWindow('tekan s untuk capture', 400, 300)
```

```
    img_resize = img[50:350, 75:575]
```

```
    cv2.imshow("tekan s untuk capture",img_resize)
```

```
    k=cv2.waitKey(1) & 0xFF
```

```
    if k==27:
```

```
        cap.release()
```

```
        cv2.destroyAllWindows()
```

```
    if k==ord('s'):# jika tombol 's' kecil ditekan
```

```

fileN='ref_3.jpg'# membuat string nama image yang disimpan
cv2.imwrite(fileN,img_resize)# simpan image di folder yang aktif sekarang
print (gambarr)# menampilkan nama image yang telah tersimpan di command
prompt(CMD) / console terminal

```

```

gambarr = gambarr+1

cv2.destroyAllWindows()

led.off()
led2.off()

text = Label(text="NAMA", font=myFont)
text.grid(row=3,column=0)

ment = StringVar()

mEntry = Entry(win,textvariable=ment)
mEntry.grid(row=4, column=0)

```

```

def ok():

```

```

    global nama
    nama = ment.get()

```

```

file = open("referensi3","r+")

```

```

file.write(nama)

```

```

file.write("\n")

```

```

file.close()

```

```

close()

```

```

mButton = Button(win, text='OK', font=myFont, command=ok, bg='red', height=1,
width=1)

```



```
mButton.grid(row=4, column=1)
```

```
win.geometry("300x200")
```

```
win.mainloop()
```

```
cv2.destroyAllWindows()
```

```
break
```

```
def input_ref4():
```

```
cap=cv2.VideoCapture(0)
```

```
cap.set(cv2.CAP_PROP_FPS,60)
```

```
gambarr=8 # membuat variable untuk penamaan file yang akan ditulis nanti
```

```
global led2, led
```

```
led = LED(12)
```

```
led2 = LED(21)
```

```
led.on()
```

```
led2.on()
```

```
while True:
```

```
_,img=cap.read()
```

```
x, y, h, w = (100, 75, 250, 450)
```

```
cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 20)
```

```
cv2.namedWindow('tekan s untuk capture', cv2.WINDOW_NORMAL)
```

```
cv2.resizeWindow('tekan s untuk capture', 400, 300)
```

```
img_resize = img[50:350, 75:575]
```

```
cv2.imshow("tekan s untuk capture",img_resize)
```

```
k=cv2.waitKey(1) & 0xFF
```

```

if k==27:

    cap.release()

    cv2.destroyAllWindows()

if k==ord('s'):# jika tombol 's' kecil ditekan

    fileN='ref_4.jpg'# membuat string nama image yang disimpan

    cv2.imwrite(fileN,img_resize)# simpan image di folder yang aktif sekarang

    print (gambarr)# menampilkan nama image yang telah tersimpan di command
prompt(CMD) / console terminal

    gambarr = gambarr+1

    cv2.destroyAllWindows()

    led.off()

    led2.off()

    text = Label(text="NAMA", font=myFont)
    text.grid(row=3,column=0)

    ment = StringVar()

    mEntry = Entry(win,textvariable=ment)
    mEntry.grid(row=4, column=0)

def ok():

    global nama

    nama = ment.get()

    file = open("referensi4","r+")

    file.write(nama)

    file.write("\n")

    file.close()

```

```

close()

mButton = Button(win, text='OK', font=myFont, command=ok, bg='red', height=1,
width=1)

mButton.grid(row=4, column=1)

win.geometry("300x200")
win.mainloop()
cv2.destroyAllWindows()
break

def input_ref5():
    cap=cv2.VideoCapture(0)
    cap.set(cv2.CAP_PROP_FPS,60)
    gambarr=9 # membuat variable untuk penamaan file yang akan ditulis nanti
    global led2, led
    led = LED(12)
    led2 = LED(21)
    led.on()
    led2.on()
    while True:
        _,img=cap.read()
        x, y, h, w = (100, 75, 250, 450)
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 20)
        cv2.namedWindow('tekan s untuk capture', cv2.WINDOW_NORMAL)
        cv2.resizeWindow('tekan s untuk capture', 400, 300)

```

```

img_resize = img[50:350, 75:575]

cv2.imshow("tekan s untuk capture",img_resize)

k=cv2.waitKey(1) & 0xFF

if k==27:

    cap.release()

    cv2.destroyAllWindows()

if k==ord('s'):# jika tombol 's' kecil ditekan
    fileN='ref_5.jpg'# membuat string nama image yang disimpan
    cv2.imwrite(fileN,img_resize)# simpan image di folder yang aktif sekarang
    print (gambarr)# menampilkan nama image yang telah tersimpan di command
    prompt(CMD) / console terminal

    gambarr = gambarr+1
    cv2.destroyAllWindows()
    led.off()
    led2.off
    text = Label(text="NAMA", font=myFont)
    text.grid(row=3,column=0)
    ment = StringVar()
    mEntry = Entry(win,textvariable=ment)
    mEntry.grid(row=4, column=0)

def ok():

    global nama

    nama = ment.get()

    file = open("referensi5","r+")

    file.write(nama)

```

```
file.write("\n")
```

```
file.close()
```

```
close()
```

```
mButton = Button(win, text='OK', font=myFont, command=ok, bg='red', height=1,  
width=1)
```

```
mButton.grid(row=4, column=1)
```

```
win.geometry("300x200")
```

```
win.mainloop()
```

```
cv2.destroyAllWindows()
```

```
break
```

```
ref1_button = Button(win, text='Tanda tangan 1', font=myFont, command=input_ref1,  
bg='red', height=1, width=11)
```

```
ref1_button.grid(row=0, column=0)
```

```
ref2_button = Button(win, text='Tanda tangan 2', font=myFont, command=input_ref2,  
bg='red', height=1, width=11)
```

```
ref2_button.grid(row=0, column=1)
```

```
ref3_button = Button(win, text='Tanda tangan 3', font=myFont, command=input_ref3,  
bg='red', height=1, width=11)
```

```
ref3_button.grid(row=1, column=0)
```

```
ref4_button = Button(win, text='Tanda tangan 4', font=myFont, command=input_ref4,  
bg='red', height=1, width=11)
```

```
ref4_button.grid(row=1, column=1)
```

```
ref5_button = Button(win, text='Tanda tangan 5', font=myFont, command=input_ref5,  
bg='red', height=1, width=11)
```

```
ref5_button.grid(row=2, column=0)
```

```
win.geometry("300x200")
```

```
win.mainloop()
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows
```

```
sys.exit()
```



Lampiran 2. Daftar program penentuan keaslian tanda tangan

```
import cv2

import sys

from gpiozero import LED

from scipy import ndimage

## GUI LIBRARY

from tkinter import *

import tkinter.font

from PIL import Image, ImageTk

## GUI

win = Tk()

win.title("Start")

myFont = tkinter.font.Font(family = 'Helvetica', size = 9, weight = "bold")

def close():

    won.destroy()

    cv2.destroyAllWindows()

    sys.exit()

def f_fake():

    global won

    won = Tk()

    won.title("Result")

    myFont = tkinter.font.Font(family = 'Helvetica', size = 9, weight = "bold")
```



```
#CONCLUSION NEGATIVE
```

```
text = Label(text="tanda tangan dinyatakan palsu", font=myFont)
```

```
text.grid(row=0,column=1)
```

```
print ("tanda tangan dinyatakan palsu")
```

```
exitButton = Button(won, text='Exit', font=myFont, command=close, bg='red', height=1,  
width=1)
```

```
exitButton.grid(row=3, column=1)
```

```
won.geometry("300x200")
```

```
won.mainloop()
```

```
def f_binary(source_image):
```

```
    gray_image = cv2.cvtColor(source_image, cv2.COLOR_BGR2GRAY)
```

```
    #cv2.imshow("Grayscale Ref", gray_image)
```

```
    ret, binary_image = cv2.threshold(gray_image,70,255,cv2.THRESH_BINARY)
```

```
    return binary_image
```

```
def f_canny(source_image):
```

```
    gray_image = cv2.cvtColor(source_image, cv2.COLOR_BGR2GRAY)
```

```
    #cv2.imshow("Grayscale Cap", gray_image)
```

```
    canny_image = cv2.Canny(gray_image,20,200)
```

```
    return canny_image
```

```
def f_crop(source_image):
```



```
#pixel di dalam kotak hijau
```

```
x_start=23
```

```
x_end=607
```

```
y_start=83
```

```
y_end=387
```

```
#asumsi awal
```

```
crop_left=641
```

```
crop_top=481
```

```
crop_right=0
```

```
crop_bottom=0
```

```
#pencarian nilai crop left
```

```
for y in range(y_start, y_end):
```

```
    for x in range(x_start, x_end):
```

```
        if source_image[y,x] == 0:
```

```
            if x <= crop_left:
```

```
                crop_left=x
```

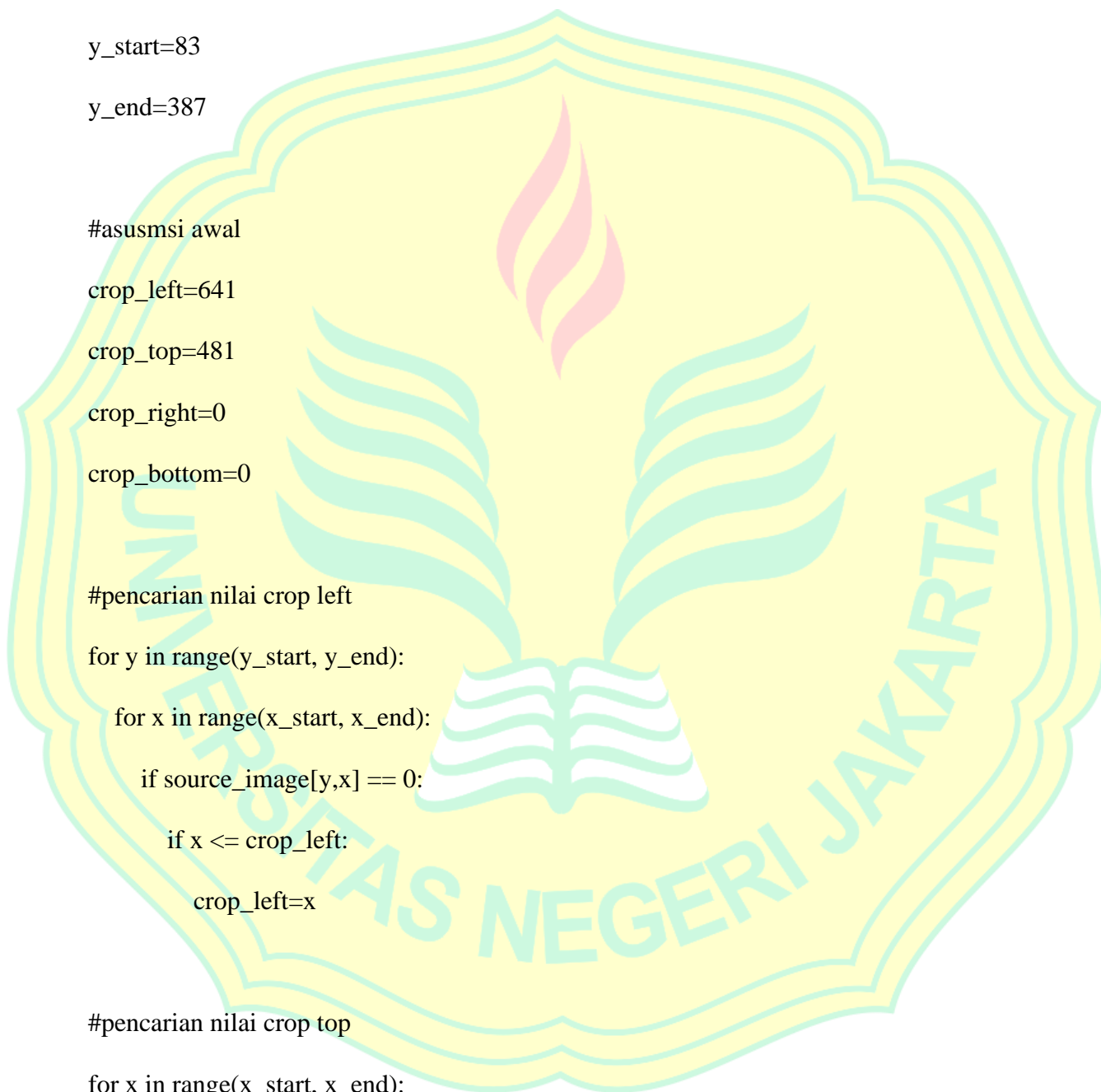
```
#pencarian nilai crop top
```

```
for x in range(x_start, x_end):
```

```
    for y in range(y_start, y_end):
```

```
        if source_image[y,x] == 0:
```

```
            if y <= crop_top:
```



```
crop_top=y

#pencarian nilai crop right
for y in range(y_start, y_end):
    for x in range(x_start, x_end):
        if source_image[y,x] == 0:
            if x >= crop_right:
                crop_right=x

#pencarian nilai crop bottom
for x in range(x_start, x_end):
    for y in range(y_start, y_end):
        if source_image[y,x] == 0:
            if y >= crop_bottom:
                crop_bottom=y

return source_image[crop_top:crop_bottom, crop_left:crop_right], crop_top, crop_bottom,
crop_left, crop_right

def f_compare(source_image, compare_image, type_image):
    count_pixel = 0
    matched_pixel = 0
    unmatched_pixel = 0

    source_height = source_image.shape[0]
    source_width = source_image.shape[1]
```

```
compare_image_resize = cv2.resize(compare_image,(source_width,source_height))
```

```
for x in list(range(source_width)):
```

```
    for y in list(range(source_height)):
```

```
        count_pixel = count_pixel+1
```

```
        if compare_image_resize[y,x] == source_image[y,x]:
```

```
            matched_pixel = matched_pixel+1
```

```
        else:
```

```
            unmatched_pixel = unmatched_pixel+1
```

```
    return (matched_pixel/count_pixel)*100
```

```
def f_real(i):
```

```
    print ("tanda tangan dinyatakan asli")
```

```
##GUI
```

```
global won
```

```
won = Tk()
```

```
won.title("Result")
```

```
myFont = tkinter.font.Font(family = 'Helvetica', size = 9, weight = "bold")
```

```
load = Image.open("ref_"+i+ ".jpg")
```

```
resize = load.resize((200, 100), Image.ANTIALIAS)
```

```
render = ImageTk.PhotoImage(resize)
```

```
img = Label(image=render)
```

```
img.grid (row=1, column=0)

file = open("referensi"+i,"r")

nama = file.read()

text = Label(text="tanda tangan dinyatakan asli", font=myFont)
text.grid(row=0,column=0)

text2 = Label(text=nama, font=myFont)
text2.grid(row=2,column=0)

exitButton = Button(won, text='Exit', font=myFont, command=close, bg='red', height=1,
width=1)
exitButton.grid(row=4, column=0)

won.geometry("300x200")
won.mainloop()
cv2.waitKey(0)
cv2.destroyAllWindows()

def main_program_1():
    win.destroy()

    #Declare Variable

    ref_image = []

    ref_image_binary = []

    ref_image_binary_cropped = []
```

```
ref_image_canny = []  
ref_image_canny_cropped = []  
  
matched_value_binary = []  
matched_value_canny = []  
  
#Import Capture  
cap_image_read = cv2.imread("cap.jpg")  
cap_image = cv2.resize(cap_image_read,(640,480))  
cap_image_binary = f_binary(cap_image)  
  
#SHOWING IMAGE  
#cv2.imshow("Biner cap", cap_image_binary)  
  
cap_image_binary_cropped, crop_top, crop_bottom, crop_left, crop_right =  
f_crop(cap_image_binary)  
fileN='cap biner cropped.jpg'  
cv2.imwrite(fileN, cap_image_binary_cropped)  
print ("Cap image done binary")  
cap_image_canny = f_canny(cap_image)  
  
#SHOWING IMAGE  
#cv2.imshow("canny cap", cap_image_canny)  
  
cap_image_canny_cropped = cap_image_canny[crop_top:crop_bottom,  
crop_left:crop_right]
```

```
fileN='cap canny cropped.jpg'

cv2.imwrite(fileN,cap_image_canny_cropped)

print ("Cap image done canny")

highest_match = float(0)

highest_match_index = 0

ref_image_read = cv2.imread("ref_1.jpg")
ref_image = cv2.resize(ref_image_read,(640,480))

#Binary
ref_image_binary = f_binary(ref_image)

#SHOWING IMAGE
#cv2.imshow("binary ref", ref_image_binary)

temp_crop, crop_top, crop_bottom, crop_left, crop_right = f_crop(ref_image_binary)
ref_image_binary_cropped = temp_crop

matched_value_binary = f_compare(ref_image_binary_cropped,
cap_image_binary_cropped, "binary")

print ("matched value ref ",1," biner : ",matched_value_binary,"%")

#Canny

ref_image_canny = f_canny(ref_image)

#SHOWING IMAGE
```

```
#cv2.imshow("canny ref", ref_image_canny)

temp_crop, crop_top, crop_bottom, crop_left, crop_right = f_crop(ref_image_canny)

ref_image_canny_cropped=temp_crop

matched_value_canny = f_compare(ref_image_canny_cropped,
cap_image_canny_cropped, "canny")

print ("matched value ref ",1," canny : ",matched_value_canny,"%")

average_value = (matched_value_binary + matched_value_canny) / 2

#Mencari nilai kecocokan tertinggi
if highest_match < average_value:
    highest_match = average_value
    highest_match_index = 1

print ("Compare done")
print ("matched value", highest_match)
print ("matched index", highest_match_index)

#Menentukan keaslian
if highest_match >= 81:
    f_real(str(highest_match_index))
else:
    f_fake()

def main_program_2():
```



```
win.destroy()

#Declare Variable

ref_image = []

ref_image_binary = []

ref_image_binary_cropped = []

ref_image_canny = []

ref_image_canny_cropped = []

matched_value_binary = []
matched_value_canny = []

#Import Capture
cap_image_read = cv2.imread("cap.jpg")
cap_image = cv2.resize(cap_image_read,(640,480))
cap_image_binary = f_binary(cap_image)

#SHOWING IMAGE
#cv2.imshow("Biner cap", cap_image_binary)

cap_image_binary_cropped, crop_top, crop_bottom, crop_left, crop_right =
f_crop(cap_image_binary)

fileN='cap biner cropped.jpg'

cv2.imwrite(fileN, cap_image_binary_cropped)

print ("Cap image done binary")

cap_image_canny = f_canny(cap_image)
```



```
#SHOWING IMAGE
```

```
#cv2.imshow("canny cap", cap_image_canny)
```

```
cap_image_canny_cropped = cap_image_canny[crop_top:crop_bottom,  
crop_left:crop_right]
```

```
fileN='cap canny cropped.jpg'
```

```
cv2.imwrite(fileN,cap_image_canny_cropped)
```

```
print ("Cap image done canny")
```

```
highest_match = float(0)
```

```
highest_match_index = 0
```

```
ref_image_read = cv2.imread("ref_2.jpg")
```

```
ref_image = cv2.resize(ref_image_read,(640,480))
```

```
#Binary
```

```
ref_image_binary = f_binary(ref_image)
```

```
#SHOWING IMAGE
```

```
#cv2.imshow("binary ref", ref_image_binary)
```

```
temp_crop, crop_top, crop_bottom, crop_left, crop_right = f_crop(ref_image_binary)
```

```
ref_image_binary_cropped = temp_crop
```

```
matched_value_binary = f_compare(ref_image_binary_cropped,  
cap_image_binary_cropped, "binary")
```

```
print ("matched value ref ",2," biner : ",matched_value_binary,"%")
```

```
#Canny
```

```
ref_image_canny=f_canny(ref_image)
```

```
#SHOWING IMAGE
```

```
#cv2.imshow("canny ref", ref_image_canny)
```

```
temp_crop, crop_top, crop_bottom, crop_left, crop_right = f_crop(ref_image_canny)
```

```
ref_image_canny_cropped=temp_crop
```

```
matched_value_canny = f_compare(ref_image_canny_cropped,  
cap_image_canny_cropped, "canny")
```

```
print ("matched value ref ",2," canny : ",matched_value_canny,"%")
```

```
average_value = (matched_value_binary + matched_value_canny) / 2
```

```
#Mencari nilai kecocokan tertinggi
```

```
if highest_match < average_value:
```

```
    highest_match = average_value
```

```
    highest_match_index = 2
```

```
print ("Compare done")
```

```
print ("matched value", highest_match)
```

```
print ("matched index", highest_match_index)
```

```
#Menentukan keaslian
```

```
if highest_match >= 81:
```

```
f_real(str(highest_match_index))
```

```
else:
```

```
f_fake()
```

```
def main_program_3():
```

```
win.destroy()
```

```
#Declare Variable
```

```
ref_image = []
```

```
ref_image_binary = []
```

```
ref_image_binary_cropped = []
```

```
ref_image_canny = []
```

```
ref_image_canny_cropped = []
```

```
matched_value_binary = []
```

```
matched_value_canny = []
```

```
#Import Capture
```

```
cap_image_read = cv2.imread("cap.jpg")
```

```
cap_image = cv2.resize(cap_image_read,(640,480))
```

```
cap_image_binary = f_binary(cap_image)
```

```
#SHOWING IMAGE
```

```
#cv2.imshow("Biner cap", cap_image_binary)
```

```
cap_image_binary_cropped, crop_top, crop_bottom, crop_left, crop_right =  
f_crop(cap_image_binary)
```

```
fileN='cap biner cropped.jpg'

cv2.imwrite(fileN, cap_image_binary_cropped)

print ("Cap image done binary")

cap_image_canny = f_canny(cap_image)

#SHOWING IMAGE

#cv2.imshow("canny cap", cap_image_canny)

cap_image_canny_cropped = cap_image_canny[crop_top:crop_bottom,
crop_left:crop_right]

fileN='cap canny cropped.jpg'

cv2.imwrite(fileN,cap_image_canny_cropped)

print ("Cap image done canny")

highest_match = float(0)

highest_match_index = 0

ref_image_read = cv2.imread("ref_3.jpg")

ref_image = cv2.resize(ref_image_read,(640,480))

#Binary

ref_image_binary = f_binary(ref_image)

#SHOWING IMAGE

#cv2.imshow("binary ref", ref_image_binary)
```

```
temp_crop, crop_top, crop_bottom, crop_left, crop_right = f_crop(ref_image_binary)

ref_image_binary_cropped = temp_crop

matched_value_binary = f_compare(ref_image_binary_cropped,
cap_image_binary_cropped, "binary")

print ("matched value ref ",3," biner : ",matched_value_binary,"%")

#Canny

ref_image_canny=f_canny(ref_image)

#SHOWING IMAGE

#cv2.imshow("canny ref", ref_image_canny)

temp_crop, crop_top, crop_bottom, crop_left, crop_right = f_crop(ref_image_canny)

ref_image_canny_cropped=temp_crop

matched_value_canny = f_compare(ref_image_canny_cropped,
cap_image_canny_cropped, "canny")

print ("matched value ref ",3," canny : ",matched_value_canny,"%")

average_value = (matched_value_binary + matched_value_canny) / 2

#Mencari nilai kecocokan tertinggi

if highest_match < average_value:

    highest_match = average_value

    highest_match_index = 3

print ("Compare done")
```

```
print ("matched value", highest_match)
print ("matched index", highest_match_index)
```

```
#Menentukan keaslian
```

```
if highest_match >= 81:
```

```
    f_real(str(highest_match_index))
```

```
else:
```

```
    f_fake()
```

```
def main_program_4():
```

```
    win.destroy()
```

```
    #Declare Variable
```

```
    ref_image = []
```

```
    ref_image_binary = []
```

```
    ref_image_binary_cropped = []
```

```
    ref_image_canny = []
```

```
    ref_image_canny_cropped = []
```

```
    matched_value_binary = []
```

```
    matched_value_canny = []
```

```
#Import Capture
```

```
cap_image_read = cv2.imread("cap.jpg")
```

```
cap_image = cv2.resize(cap_image_read,(640,480))
```

```
cap_image_binary = f_binary(cap_image)
```

```
#SHOWING IMAGE
```

```
#cv2.imshow("Biner cap", cap_image_binary)
```

```
cap_image_binary_cropped, crop_top, crop_bottom, crop_left, crop_right =  
f_crop(cap_image_binary)
```

```
fileN='cap biner cropped.jpg'
```

```
cv2.imwrite(fileN, cap_image_binary_cropped)
```

```
print ("Cap image done binary")
```

```
cap_image_canny = f_canny(cap_image)
```

```
#SHOWING IMAGE
```

```
#cv2.imshow("canny cap", cap_image_canny)
```

```
cap_image_canny_cropped = cap_image_canny[crop_top:crop_bottom,  
crop_left:crop_right]
```

```
fileN='cap canny cropped.jpg'
```

```
cv2.imwrite(fileN, cap_image_canny_cropped)
```

```
print ("Cap image done canny")
```

```
highest_match = float(0)
```

```
highest_match_index = 0
```

```
ref_image_read = cv2.imread("ref_4.jpg")
```

```
ref_image = cv2.resize(ref_image_read,(640,480))
```



```
#Binary
```

```
ref_image_binary = f_binary(ref_image)
```

```
#SHOWING IMAGE
```

```
#cv2.imshow("binary ref", ref_image_binary)
```

```
temp_crop, crop_top, crop_bottom, crop_left, crop_right = f_crop(ref_image_binary)
```

```
ref_image_binary_cropped = temp_crop
```

```
matched_value_binary = f_compare(ref_image_binary_cropped,  
cap_image_binary_cropped, "binary")
```

```
print ("matched value ref ",4," biner : ",matched_value_binary,"%")
```

```
#Canny
```

```
ref_image_canny=f_canny(ref_image)
```

```
#SHOWING IMAGE
```

```
#cv2.imshow("canny ref", ref_image_canny)
```

```
temp_crop, crop_top, crop_bottom, crop_left, crop_right = f_crop(ref_image_canny)
```

```
ref_image_canny_cropped=temp_crop
```

```
matched_value_canny = f_compare(ref_image_canny_cropped,  
cap_image_canny_cropped, "canny")
```

```
print ("matched value ref ",4," canny : ",matched_value_canny,"%")
```

```
average_value = (matched_value_binary + matched_value_canny) / 2
```



```
#Mencari nilai kecocokan tertinggi
if highest_match < average_value:
    highest_match = average_value
    highest_match_index = 4

print ("Compare done")
print ("matched value", highest_match)
print ("matched index", highest_match_index)

#Menentukan keaslian
if highest_match >= 81:
    f_real(str(highest_match_index))
else:
    f_fake()

def main_program_5():
    win.destroy()
    #Declare Variable
    ref_image = []
    ref_image_binary = []
    ref_image_binary_cropped = []
    ref_image_canny = []
    ref_image_canny_cropped = []

    matched_value_binary = []
```

```
matched_value_canny = []

#Import Capture

cap_image_read = cv2.imread("cap.jpg")

cap_image = cv2.resize(cap_image_read,(640,480))

cap_image_binary = f_binary(cap_image)

#SHOWING IMAGE

#cv2.imshow("Biner cap", cap_image_binary)

cap_image_binary_cropped, crop_top, crop_bottom, crop_left, crop_right =
f_crop(cap_image_binary)

fileN='cap biner cropped.jpg'

cv2.imwrite(fileN, cap_image_binary_cropped)

print ("Cap image done binary")

cap_image_canny = f_canny(cap_image)

#SHOWING IMAGE

#cv2.imshow("canny cap", cap_image_canny)

cap_image_canny_cropped = cap_image_canny[crop_top:crop_bottom,
crop_left:crop_right]

fileN='cap canny cropped.jpg'

cv2.imwrite(fileN,cap_image_canny_cropped)

print ("Cap image done canny")
```

```
highest_match = float(0)

highest_match_index = 0

ref_image_read = cv2.imread("ref_5.jpg")
ref_image = cv2.resize(ref_image_read,(640,480))

#Binary
ref_image_binary = f_binary(ref_image)

#SHOWING IMAGE
#cv2.imshow("binary ref", ref_image_binary)

temp_crop, crop_top, crop_bottom, crop_left, crop_right = f_crop(ref_image_binary)
ref_image_binary_cropped = temp_crop

matched_value_binary = f_compare(ref_image_binary_cropped,
cap_image_binary_cropped, "binary")

print ("matched value ref ",5," biner : ",matched_value_binary,"%")

#Canny
ref_image_canny=f_canny(ref_image)

#SHOWING IMAGE
#cv2.imshow("canny ref", ref_image_canny)

temp_crop, crop_top, crop_bottom, crop_left, crop_right = f_crop(ref_image_canny)

ref_image_canny_cropped=temp_crop
```

```
matched_value_canny = f_compare(ref_image_canny_cropped,  
cap_image_canny_cropped, "canny")
```

```
print ("matched value ref ",5," canny : ",matched_value_canny,"%")
```

```
average_value = (matched_value_binary + matched_value_canny) / 2
```

```
#Mencari nilai kecocokan tertinggi
```

```
if highest_match < average_value:
```

```
    highest_match = average_value
```

```
    highest_match_index = 5
```

```
print ("Compare done")
```

```
print ("matched value", highest_match)
```

```
print ("matched index", highest_match_index)
```

```
#Menentukan keaslian
```

```
if highest_match >= 81:
```

```
    f_real(str(highest_match_index))
```

```
else:
```

```
    f_fake()
```

```
#capture_image
```

```
cap=cv2.VideoCapture(0)
```

```
cap.set(cv2.CAP_PROP_FPS,60)
```

```
gambarr=1
```

```
global led2, led
```

```
led = LED(12)
led2 = LED(21)
led.on()
led2.on()
while True :
    _,img=cap.read()
    x, y, h, w = (100, 75, 250, 450)
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 20)
    cv2.namedWindow('tekan s untuk capture', cv2.WINDOW_NORMAL)
    cv2.resizeWindow('tekan s untuk capture', 400, 300)
    img_resize = img[50:350, 75:575]
    cv2.imshow('tekan s untuk capture',img_resize)
    k=cv2.waitKey(1) & 0xFF
    if k==27:
        cap.release()
        cv2.destroyAllWindows()
        break
    if k==ord('s'):
        fileN='cap.jpg'
        cv2.imwrite(fileN,img_resize)
        print (gambarr)
        led.off()
        led2.off()
```

```
cv2.destroyAllWindows()
```

```
break
```

```
file = open("referensi1", "r")
```

```
nama = file.read()
```

```
ref1_button = Button(win, text=nama, font=myFont, command=main_program_1, bg='red',  
height=1, width=15)
```

```
ref1_button.grid(row=0, column=0)
```

```
file = open("referensi2", "r")
```

```
nama = file.read()
```

```
ref1_button = Button(win, text=nama, font=myFont, command=main_program_2, bg='red',  
height=1, width=15)
```

```
ref1_button.grid(row=0, column=1)
```

```
file = open("referensi3", "r")
```

```
nama = file.read()
```

```
ref1_button = Button(win, text=nama, font=myFont, command=main_program_3, bg='red',  
height=1, width=15)
```

```
ref1_button.grid(row=1, column=0)
```

```
file = open("referensi4", "r")
```

```
nama = file.read()
```

```
ref1_button = Button(win, text=nama, font=myFont, command=main_program_4, bg='red',  
height=1, width=15)
```

```
ref1_button.grid(row=1, column=1)
```

```
file = open("referensi5","r")
```

```
nama = file.read()
```

```
ref1_button = Button(win, text=nama, font=myFont, command=main_program_5, bg='red',  
height=1, width=15)
```

```
ref1_button.grid(row=2, column=0)
```

```
win.geometry("300x200")
```

```
win.mainloop()
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```



Lampiran 3. Lembar Uji Sistem Alat Lembar Uji

Alat Penentu Keaslian Tanda Tangan Berbasis Pengolahan Citra Menggunakan Raspberry Pi 3

Tanggal Pengujian :

Ketentuan Uji :

Dibutuhkannya dua(2) orang sebagai penanda tangan dan pemalsu tanda tangan

1. Tanda tangan pada kotak tanda tangan referensi.
2. Tanda tangan pada kotak tanda tangan asli sebanyak lima(5) kali
3. Pemalsu tanda tangan pada kotak tanda tangan sebanyak lima (5) kali
4. Tanda tangan dengan jelas dan jangan keluar dari kotak yang telah disediakan
5. Tanda tangan menggunakan pulpen dengan ketebalan 0,5 MM

Nama penanda tangan :

Nama pemalsu tanda tangan :

TTD Referensi

TTD Asli

TTD Palsu