##Binary Picture DENGAN PERHITUNGAN MANUAL

ts = time.time() #MENGHITUNG WAKTU PEMROSESAN

H,W = Resize.shape[:2] #MENGAMBIL TINGGI DAN LEBAR

BINARY = np.zeros((H,W), np.uint8) #MEMBUAT ARRAY MENJADI ZERO ATAU 0 DENGAN BATAS 8 BIT

for i in range(H):

for j in range(W):

B=Resize[i,j,0] #mengambil nilai BLUE DARI GAMBAR

G=Resize[i,j,1] #mengambil nilai GREEN DARI GAMBAR

R=Resize[i,j,2] #mengambil nilai RED DARI GAMBAR

#RUMUS PERHITUNGAN NORMALISASI RGB

b = B/(B+G+R)

g = G/(B+G+R)

r = R/(B+G+R)

#AMBANG BATAS/ TRESHOLD DIPEROLEH NILAI r dan g dari perhitungan warna LEMON pada umumnya. Untuk B tidak disertakan karena nilai B pada warna lemon adalah 0.

if(r > 0.5 and r < 2) and (g > 0.4 and g < 2):

#JIKA MEMENUHI NILAI TRESHOLD MAKA DIBERI WARNA 255 YAITU PUTIH

BINARY[i,j] = np.clip(255 \* Resize[i,j,0] + 255\* Resize[i,j,1] + 255 \* Resize[i,j,2], 0, 255)

else:

#JIKA TIDAK MEMENUHI NILAI TRESHOLD MAKA DIBERI WARNA 0 YAITU HITAM

BINARY[i,j] = np.clip(0 \* Resize[i,j,0] + 0\* Resize[i,j,1] + 0 \* Resize[i,j,2], 0, 255)

t = (time.time() -ts)

print("Loop: {:} ms".format(t\*1000))

cv2.imshow("BINARY", BINARY)

cv2.waitKey()

##Grayscale Picture DENGAN PERHITUNGAN MANUAL

ts = time.time() #MENGHITUNG WAKTU PEMROSESAN

H,W = Resize.shape[:2]#MENGAMBIL TINGGI DAN LEBAR

GRAY = np.zeros((H,W), np.uint8)#MEMBUAT ARRAY MENJADI ZERO ATAU 0 DENGAN BATAS 8 BIT

for i in range(H):

for j in range(W):

B=Resize[i,j,0] #mengambil nilai BLUE DARI GAMBAR

G=Resize[i,j,1] #mengambil nilai GREEN DARI GAMBAR

R=Resize[i,j,2] #mengambil nilai RED DARI GAMBAR

#RUMUS PERHITUNGAN NORMALISASI RGB

b = B/(B+G+R)

g = G/(B+G+R)

r = R/(B+G+R)

#AMBANG BATAS/ TRESHOLD DIPEROLEH NILAI r dan g dari perhitungan warna LEMON pada umumnya. Untuk nilai b tidak disertakan karena nilai b pada warna lemon adalah 0.

if(r > 0.5 and r < 2) and (g > 0.4 and g < 2):

#JIKA MEMENUHI NILAI TRESHOLD MAKA DIMASUKAN KEDALAM RUMUS GRAYSCALE YAITU SEBAGAI BERIKUT :

GRAY[i,j] = np.clip(0.07 \* Resize[i,j,0] + 0.72 \* Resize[i,j,1] + 0.21 \* Resize[i,j,2], 0, 255)

else:

#JIKA TIDAK MEMENUHI NILAI TRESHOLD JUGA DIMASUKAN KEDALAM RUMUS GRAYSCALE YAITU SEBAGAI BERIKUT WALAUPUN MEMILIKI RUMUS SAMA TETAPI HASIL PERHITUNGAN AKAN MEMISAHKAN ANTAR OBJECT DAN BACKGROUND :

GRAY[i,j] = np.clip(0.07 \* Resize[i,j,0] + 0.72 \* Resize[i,j,1] + 0.21 \* Resize[i,j,2], 0, 255)

t = (time.time() -ts)

print("Loop: {:} ms".format(t\*1000))

cv2.imshow("GRAY", GRAY)

cv2.waitKey()