



Hello!

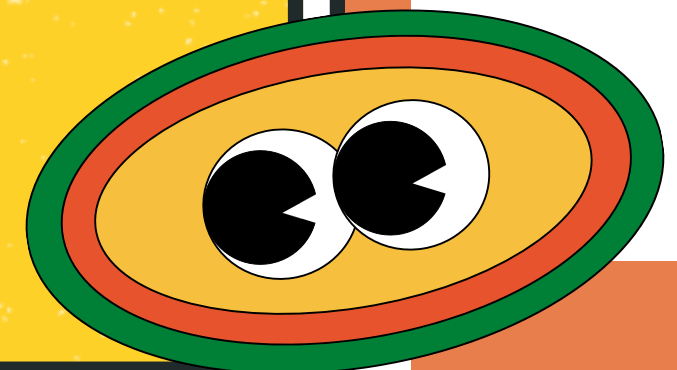
Aplikasi pada Aljabar Linear : Konversi Citra RGB ke Grayscale



Introduction

ANGGOTA KELOMPOK :

1. Dikriani (221011057)
2. Tristia Desiana. P (221011028)
3. Alfira Zalsabila (221011110)



```

import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

img_path = 'alfira zalsabila.jpg'
img = cv2.imread(img_path)
print(img.shape)

fix_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(fix_img)

R, G, B = fix_img[:, :, 0], fix_img[:, :, 1], fix_img[:, :, 2]
print(np.array(fix_img))

```

```

from io import IncrementalNewlineDecoder
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

img_path = 'Tristia Desiana.jpeg'
img = cv2.imread(img_path)
print(img.shape)

fix_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(fix_img)

R, G, B = fix_img[:, :, 0], fix_img[:, :, 1], fix_img[:, :, 2]
print(np.array(fix_img))

```

```

import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

img_path = "Dikriani.jpg"
img = cv2.imread(img_path)
print(img.shape)

fix_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(fix_img)

R, G, B = fix_img[:, :, 0], fix_img[:, :, 1], fix_img[:, :, 2]
print(np.array(fix_img))

```

Kode tersebut membaca gambar dari file, mengonversi skema warna dari BGR ke RGB, dan menampilkan gambar. Selain itu, kode juga mencetak dimensi gambar dan memisahkan saluran warna (merah, hijau, biru) serta mencetak representasi array NumPy dari gambar yang telah dimodifikasi.

FOTO ASLI



Dikriani.jpg



Alfira Zalsabila.jpg

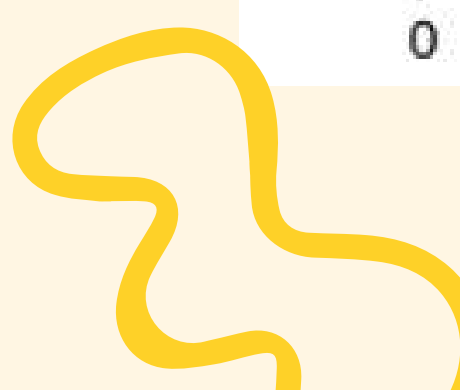
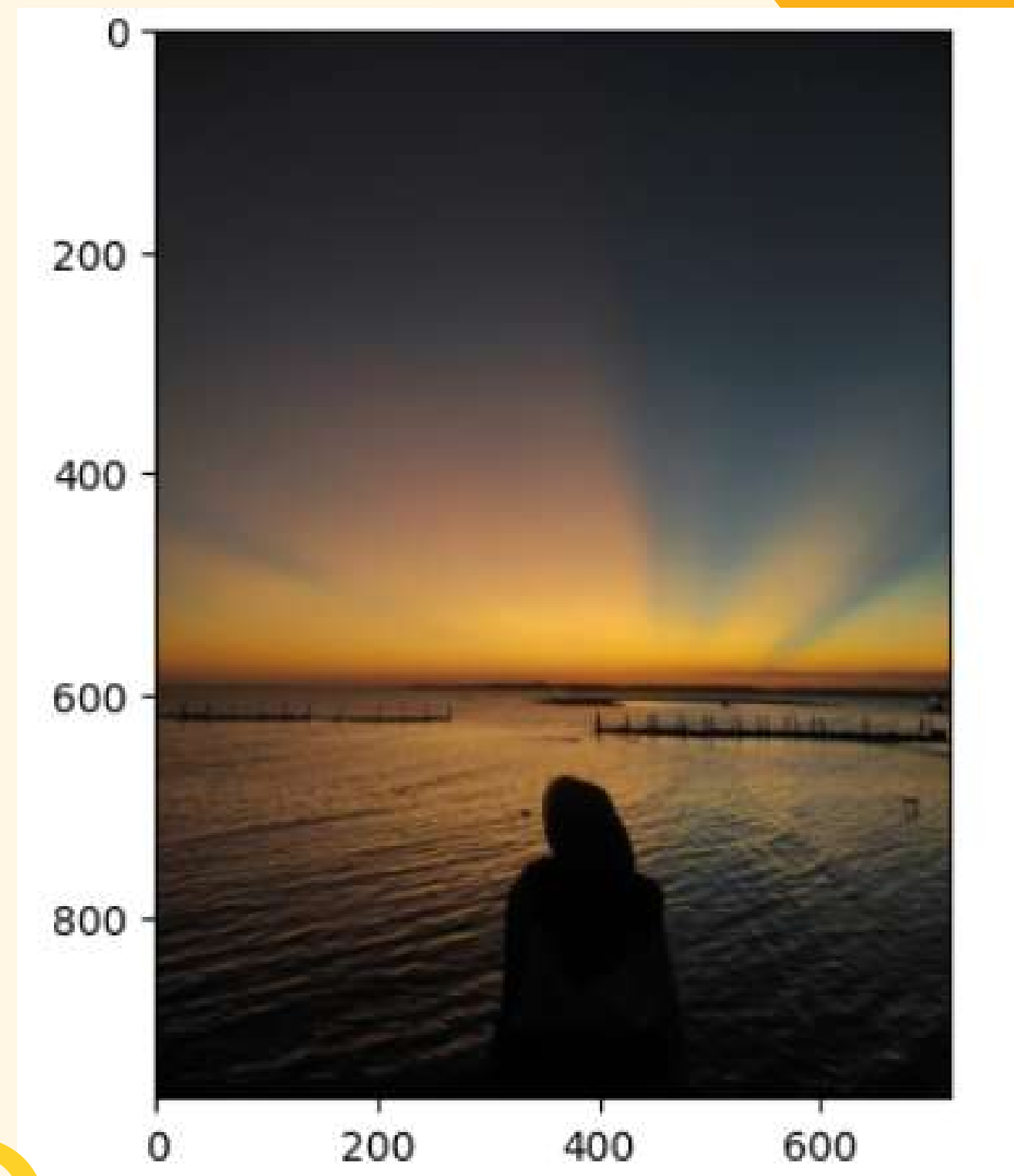
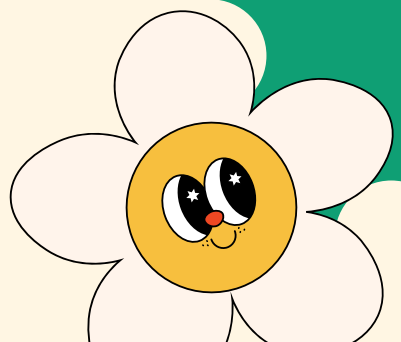


Tristia Desiana.jpg



```
(964, 716, 3)
[[[ 0  0  0]
   [ 0  0  0]
   [ 0  0  0]
```

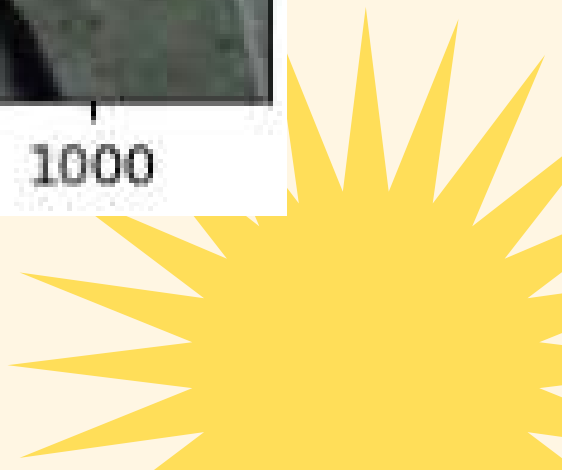
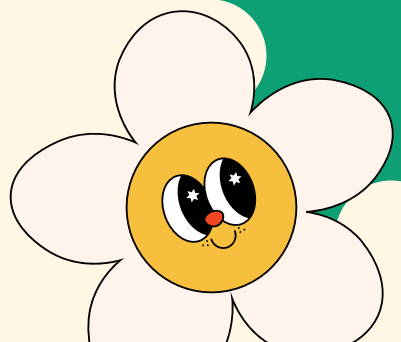
- **Ukuran gambar: dimensi tinggi 964, piksel , 716 piksel tinggi dan menggunakan format warna RGB**
- **3 : jumlah saluran warna atau dimensi**
[0 0 0]
[0 0 0]
[0 0 0]
- **R bernilai 0**
G bernilai 0
B bernilai 0





```
(1600, 1200, 3)
[[[ 36  31  38]
   [ 29  24  31]
   [ 20  15  22]
```

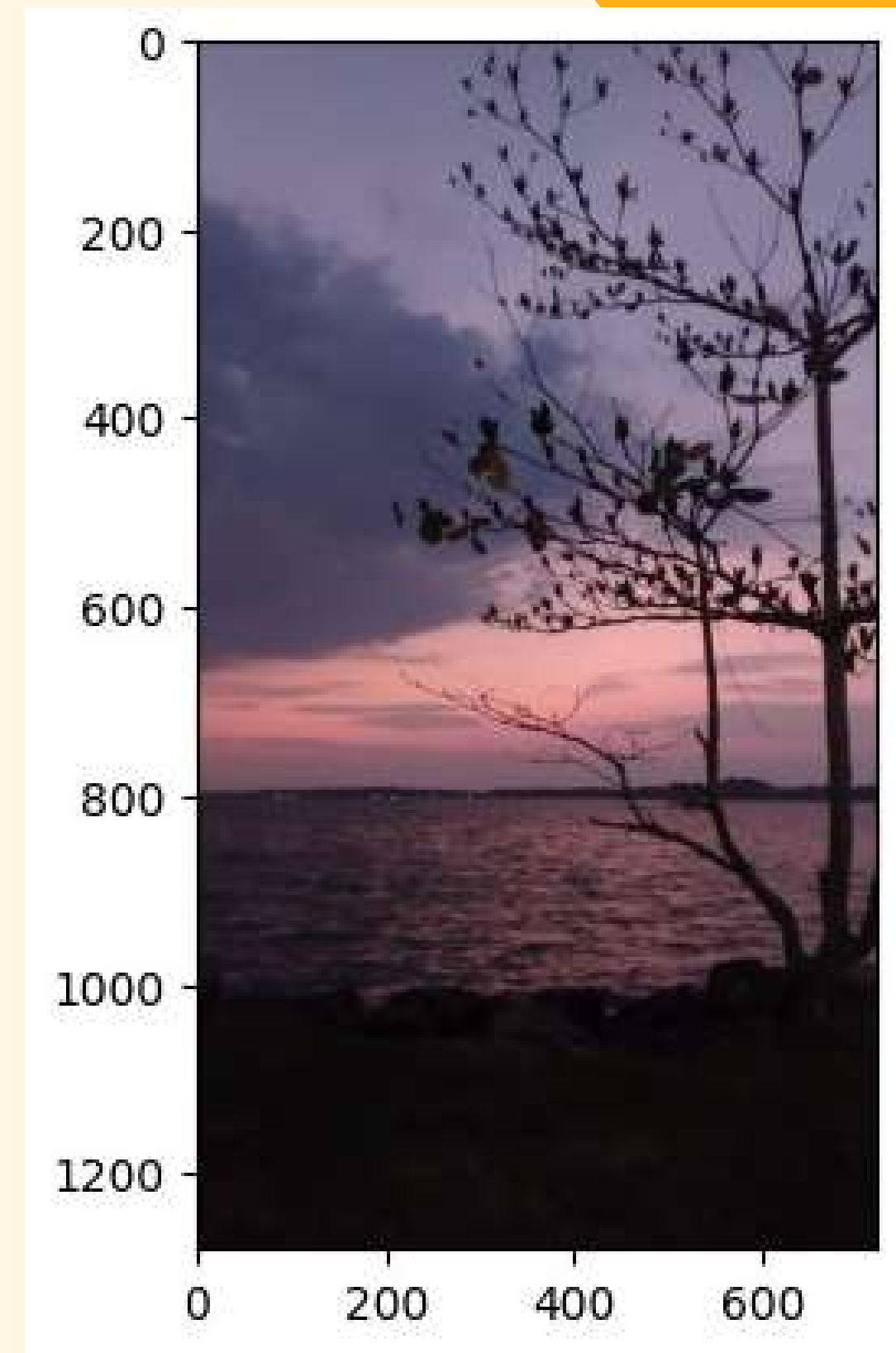
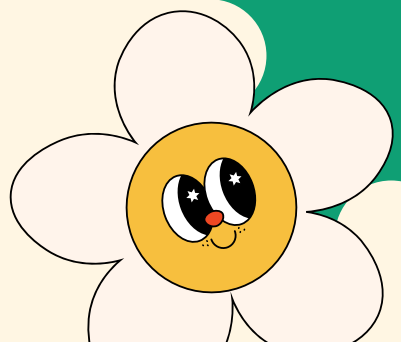
- **Ukuran gambar: dimensi tinggi 1600, piksel , 1200 piksel tinggi dan menggunakan format warna RGB**
- **3 : jumlah saluran warna atau dimensi**
[36 31 38]
[29 24 31]
[20 15 22]
- **R bernilai 36**
G bernilai 31
B bernilai 38

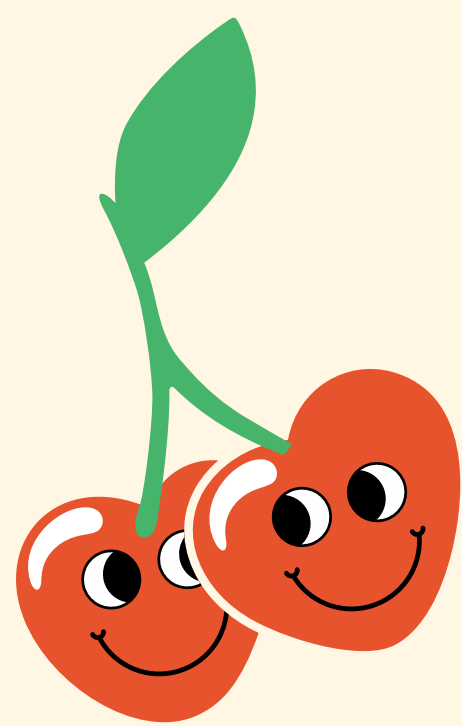




```
(1280, 720, 3)
[[[115 110 140]
  [115 110 140]
  [115 110 140]
```

- **Ukuran gambar: dimensi tinggi 1280, piksel , 720 piksel tinggi dan menggunakan format warna RGB**
- **3 : jumlah saluran warna atau dimensi**
[115 110 140]
[115 110 140]
[115 110 140]
- **R bernilai 115**
G bernilai 110
B bernilai 140

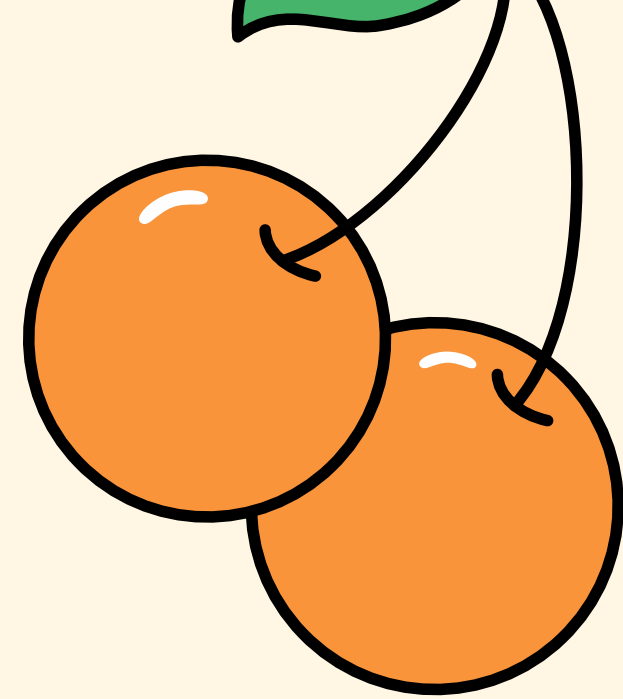




Metode Lightness

```
fix_img[:] = np.max(fix_img, axis=-1, keepdims=1)/2 + np.min(fix_img, axis=-1, keepdims=1)/2  
print(np.array(fix_img[:]))
```

```
plt.axis('off')  
plt.imshow(fix_img[:])  
plt.savefig('Lightness', bbox_inches= 'tight')
```



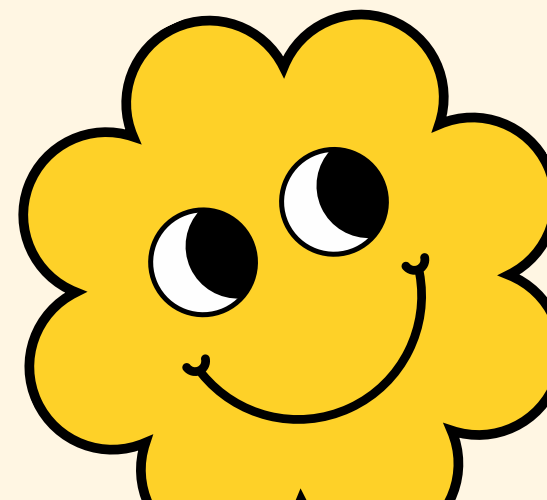
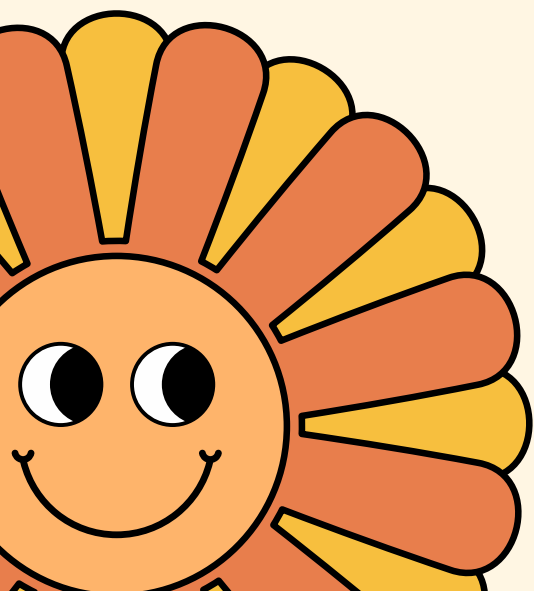
Outputnya

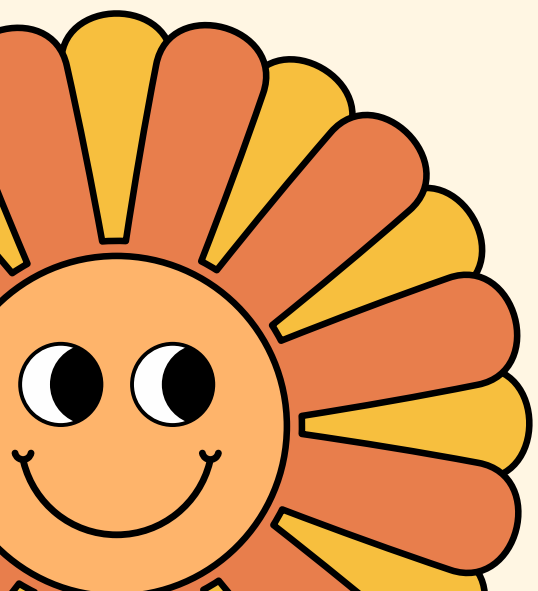
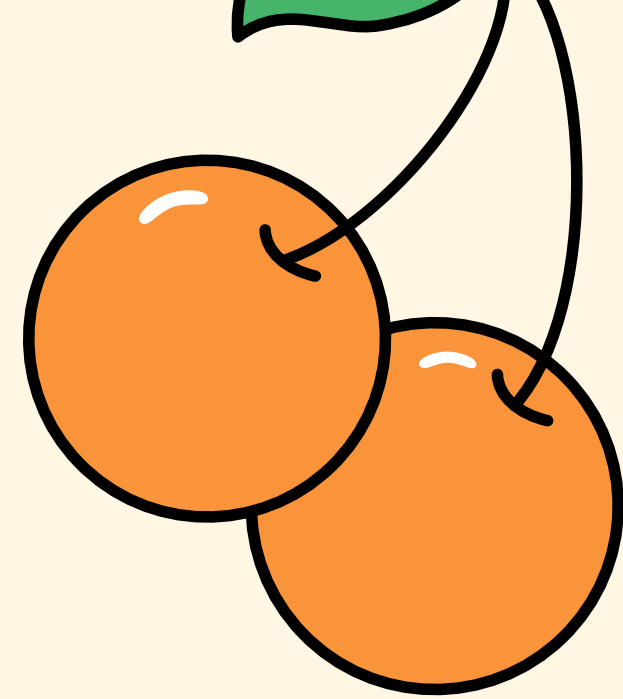
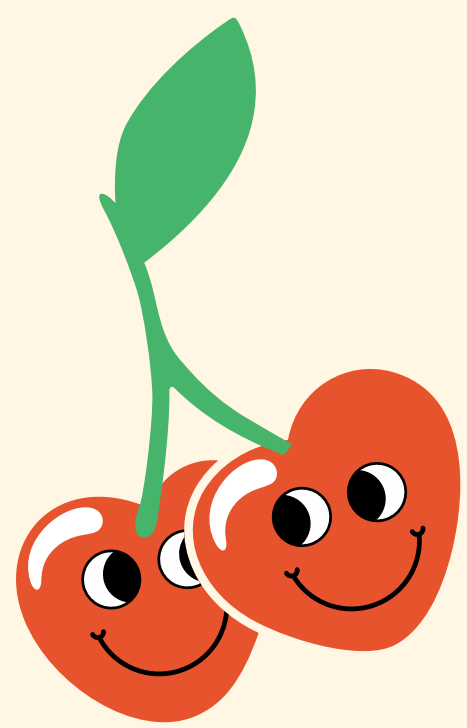
```
[[[ 0  0  0]  
 [ 0  0  0]  
 [ 0  0  0]  
 ...  
 [ 0  0  0]  
 [ 0  0  0]  
 [ 0  0  0]]
```

```
[[ 1  1  1]  
 [ 1  1  1]  
 [ 1  1  1]]
```

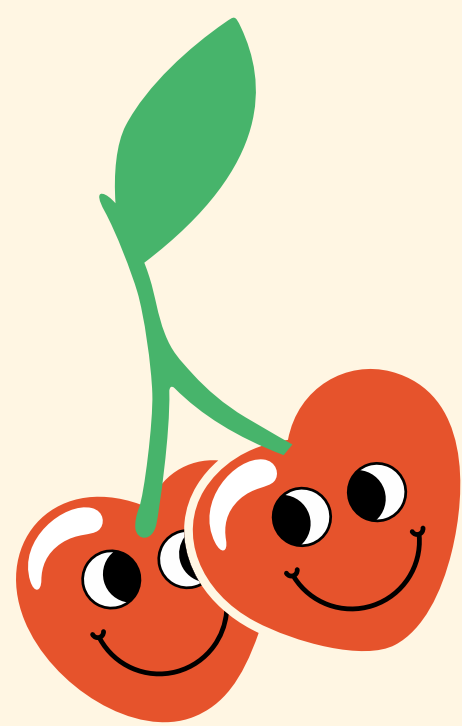
0 merupakan hasil dari rumus lightness
Grayscale = (max(R, G, B)) + (min(R, G, B)) / 2
yaitu :

- **(0+0)/2 = 0**
- **(0+0)/2 = 0**
- **(0+0)/2 = 0**





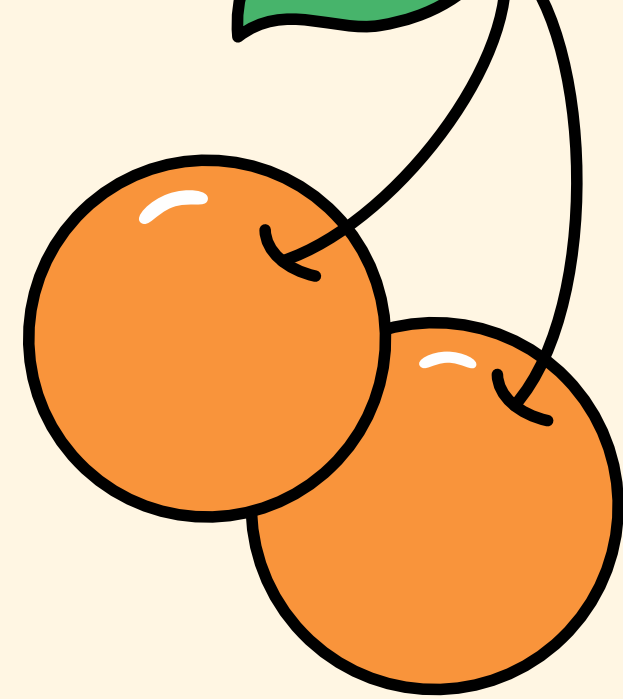
Hasil fotonya



Metode Lightness

```
fix_img[:] = np.max(fix_img, axis=-1, keepdims=1)/2 + np.min(fix_img, axis=-1, keepdims=1)/2
print(np.array(fix_img[:]))

plt.axis('off')
plt.imshow(fix_img[:])
plt.savefig('Metode Lightness')
```



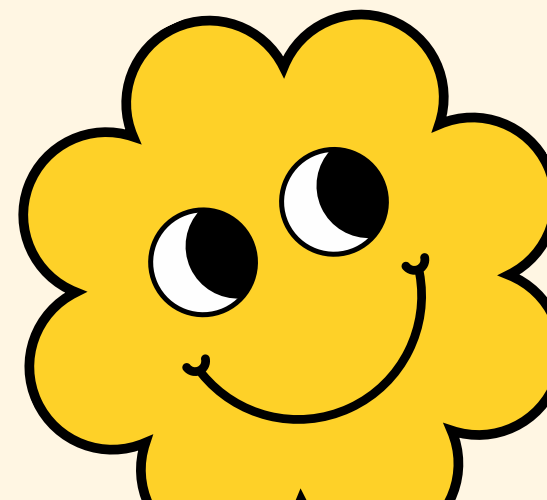
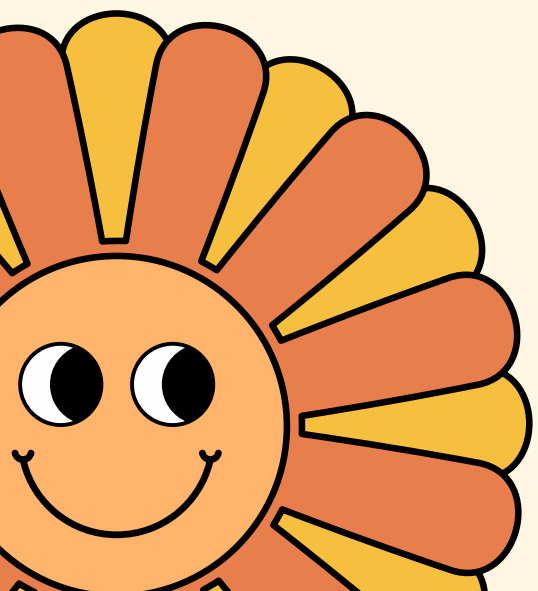
Outputnya

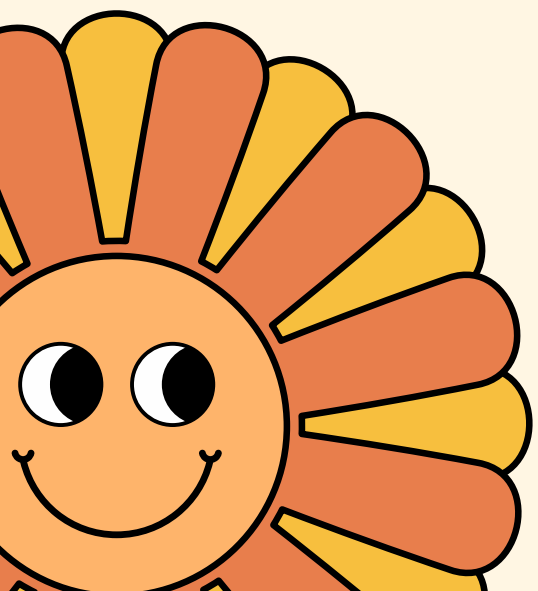
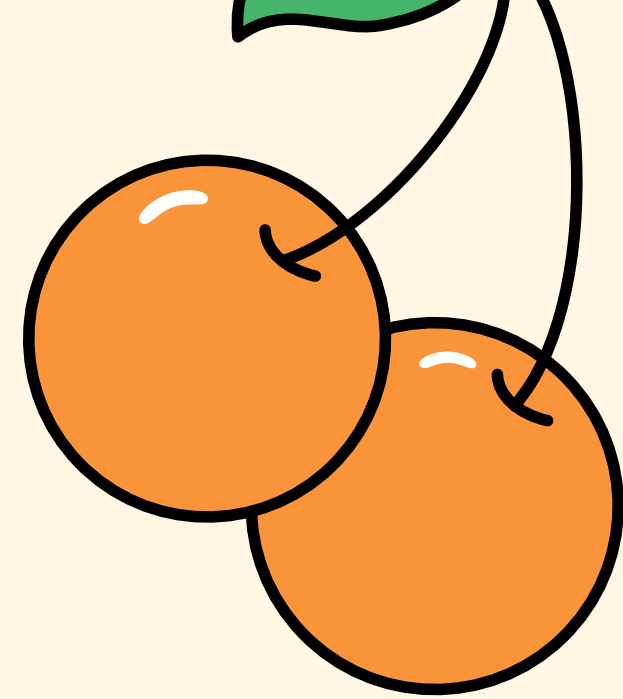
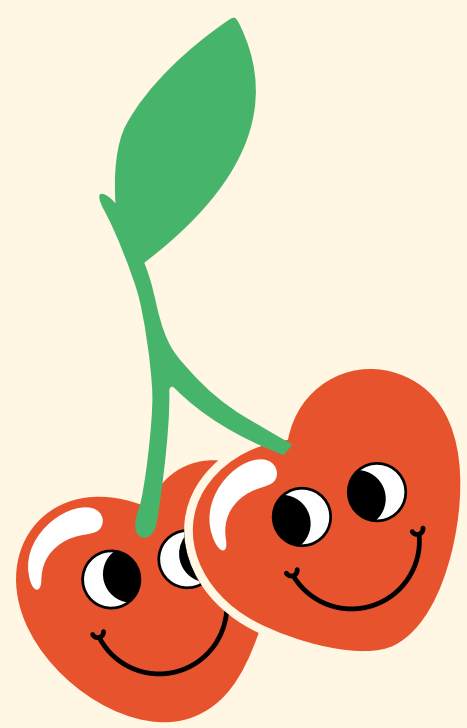
```
→ [[[ 34  34  34]
      [ 27  27  27]
      [ 18  18  18]
      ...
      [  7   7   7]
      [  8   8   8]
      [  9   9   9]]

     [[ 28  28  28]
      [ 21  21  21]
      [ 13  13  13]]
```

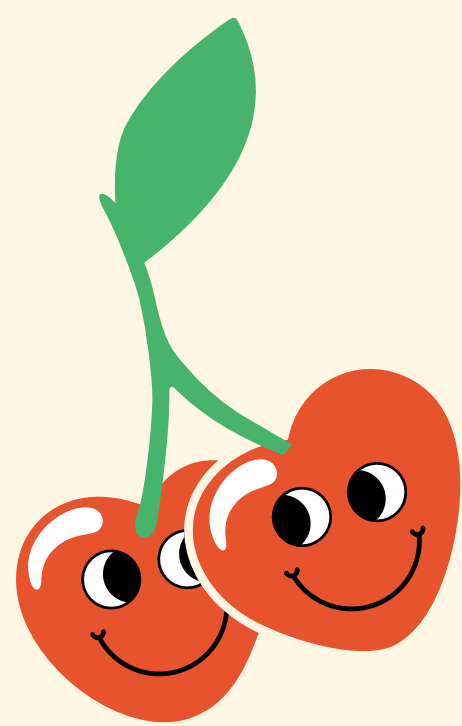
0 merupakan hasil dari rumus lightness
Grayscale = (max(R, G, B)) + (min(R, G, B)) / 2
yaitu :

- **$(31+38)/2 = 37$**
- **$(115+140)/2 = 37$**
- **$(115+140)/2 = 37$**





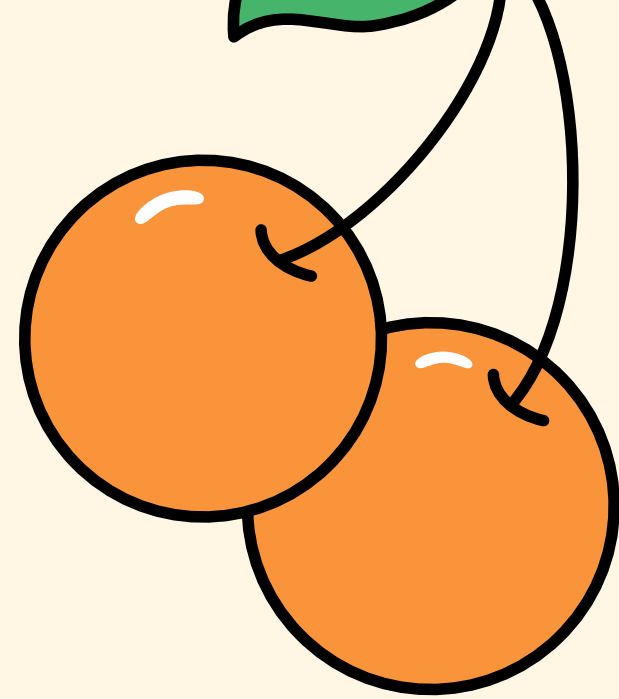
Hasil fotonya



Metode Lightness

```
fix_img[:] = np.max(fix_img, axis=-1, keepdims=1)/2 + np.min(fix_img, axis=-1, keepdims=1)/2
print(np.array(fix_img[:]))

plt.axis('off')
plt.imshow(fix_img[:])
plt.savefig('Lightness', bbox_inches= 'tight')
```



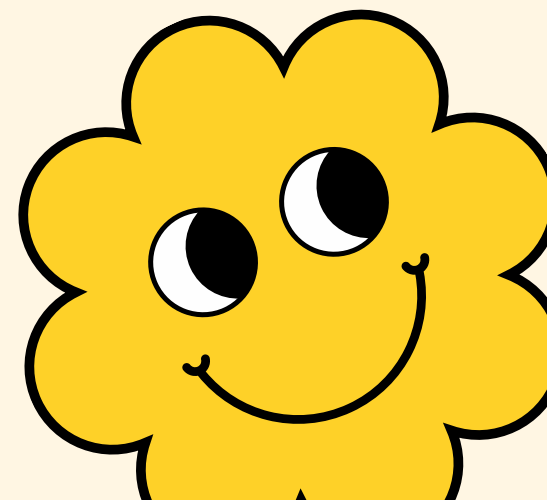
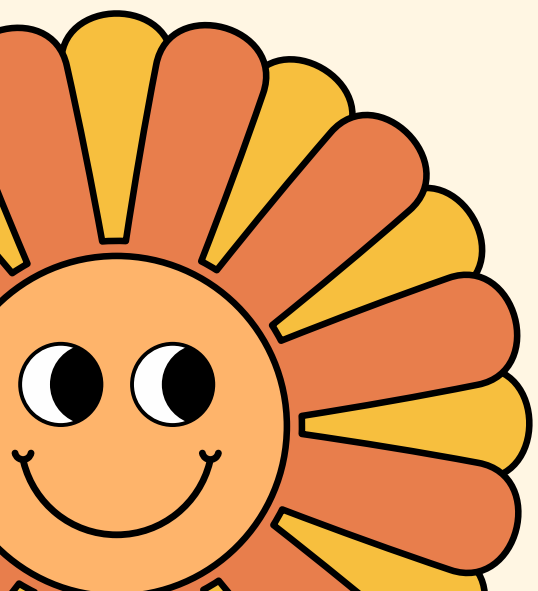
Outputnya

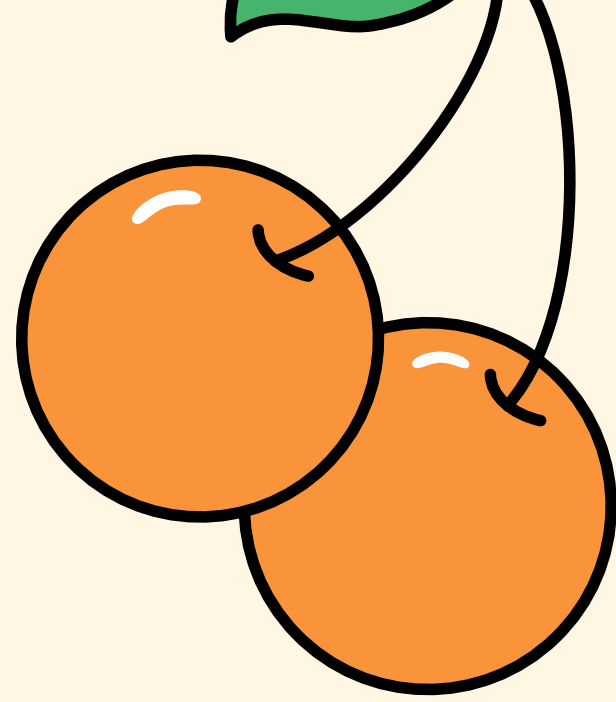
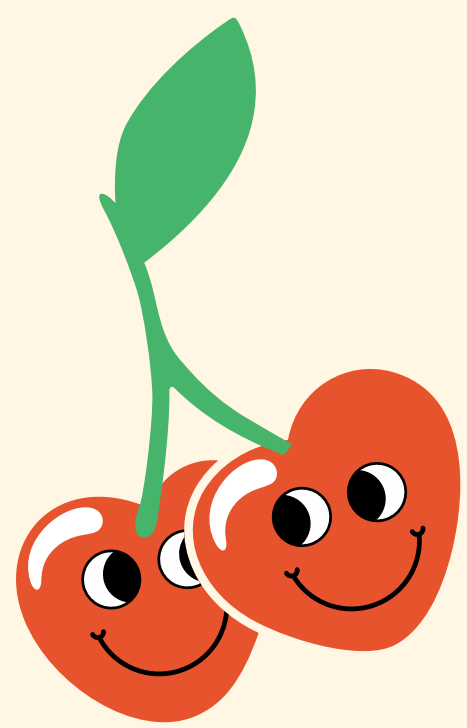
```
[[[125 125 125]
  [125 125 125]
  [125 125 125]
  ...
  [136 136 136]
  [137 137 137]
  [138 138 138]]

 [[125 125 125]
  [125 125 125]
  [125 125 125]]
```

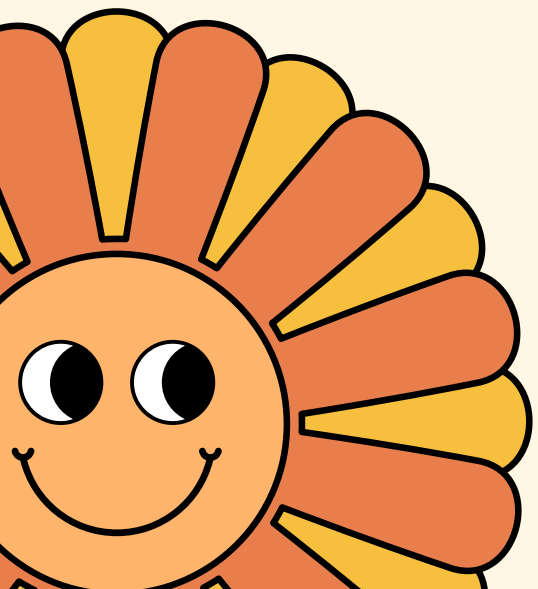
125 merupakan hasil dari rumus lightness
Grayscale = (max(R, G, B)) + (min(R, G, B)) / 2
yaitu :

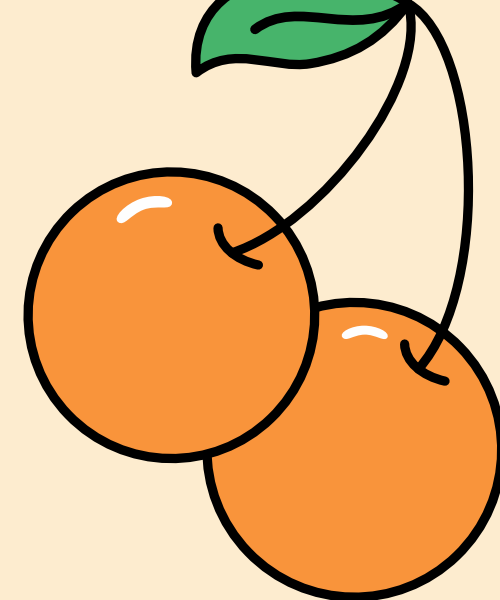
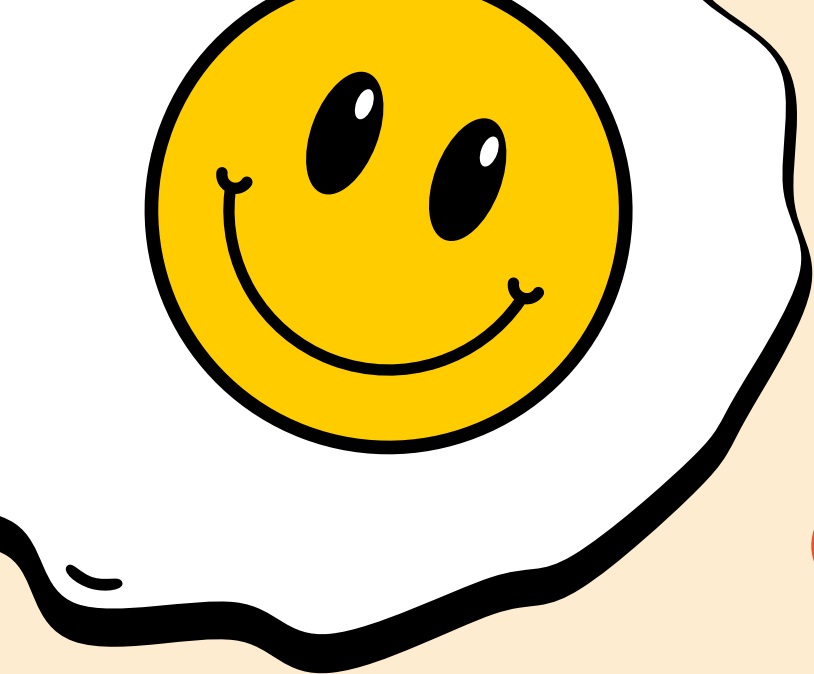
- **(140+110)/2 = 125**
- **(151+122)/2 = 136**
- **(140+110)/2 = 125**





Hasil fotonya



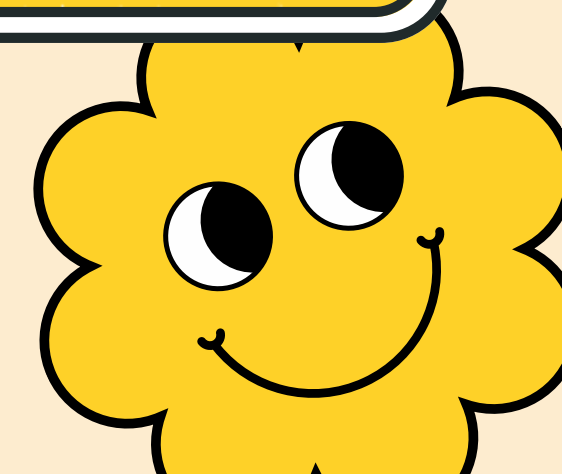


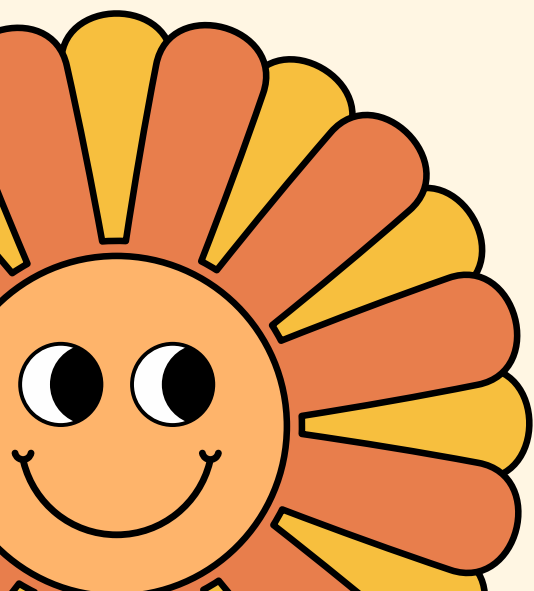
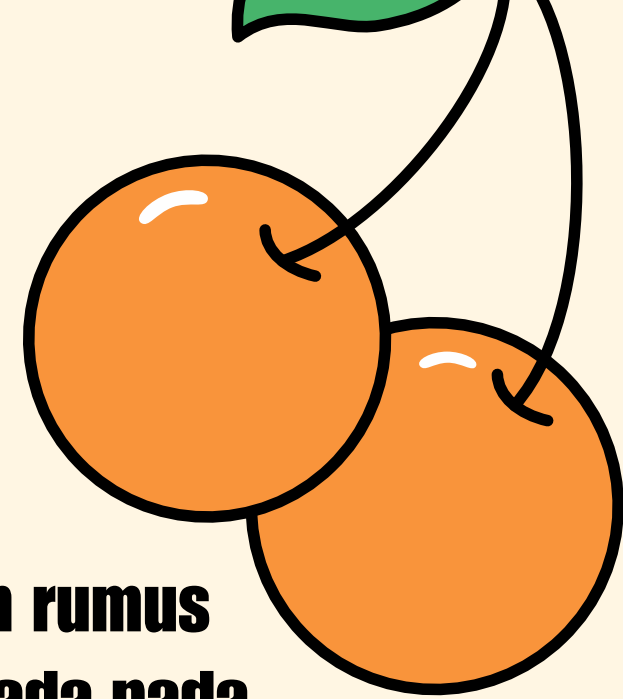
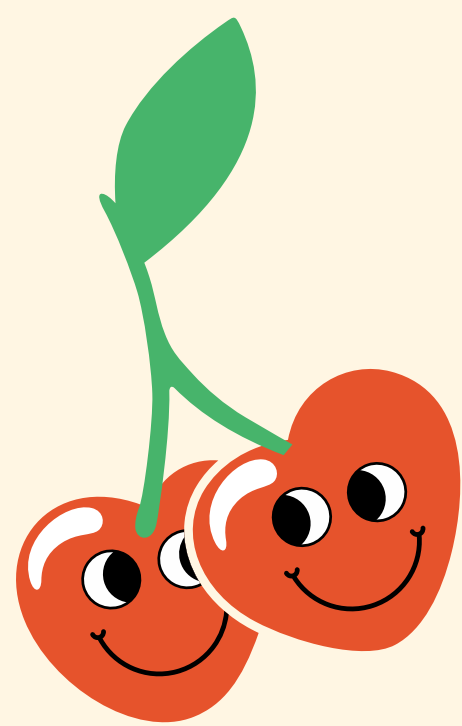
METODE AVERAGE

```
gray_img = np.mean(fix_img, axis=-1)
print(np.array(gray_img))

plt.axis('off')
plt.imshow(gray_img, cmap='gray')
plt.savefig('Average', bbox_inches='tight')
```

KODE INI MENGONVERSI GAMBAR BERWARNA MENJADI CITRA KEABUAN DENGAN MENGAMBIL NILAI RATA-RATA DARI SALURAN WARNA DAN KEMUDIAN MENAMPILKAN SERTA MENYIMPAN HASILNYA DALAM CITRA KEABUAN.





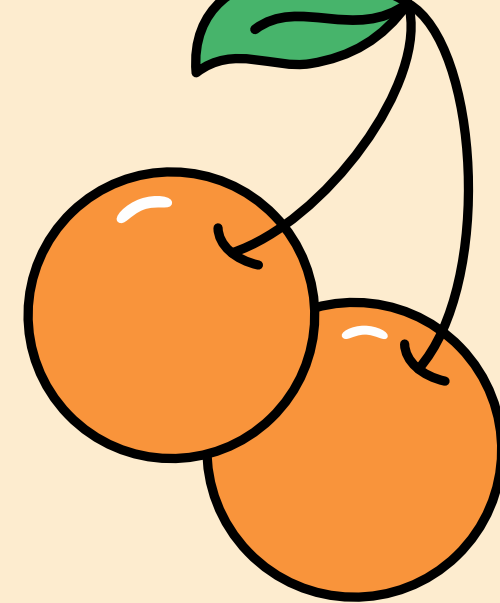
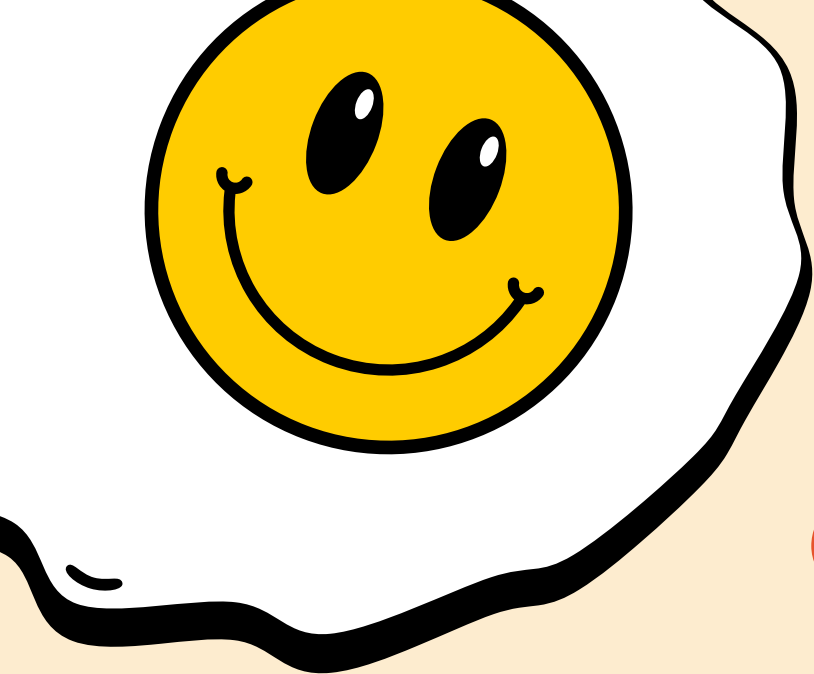
**0 merupakan hasil dari rata-rata nilai dengan rumus
Grayscale = $(R + G + B) / 3$ pada baris 1 yang ada pada
matriks sebelumnya yakni :**

$$(0 + 0 + 0) / 3 = 0$$

$$(0 + 0 + 0) / 3 = 0$$

$$(0 + 0 + 0) / 3 = 0$$

```
[ [ 0.  0.  0.  ...  0.  0.  0. ]  
  [ 1.  1.  1.  ...  1.  1.  2. ]  
  [10. 10. 10.  ...  8.  9.  9. ]  
  ...  
  [ 3.  3.  3.  ... 12. 12. 12. ]  
  [ 3.  3.  3.  ... 11. 12. 13. ]  
  [ 2.  2.  2.  ...  9. 10. 10. ] ]
```

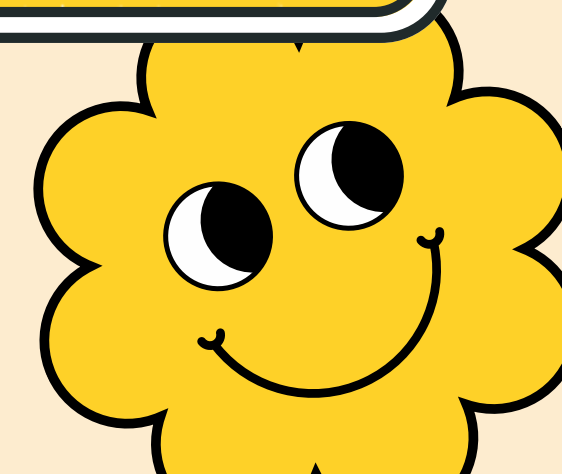


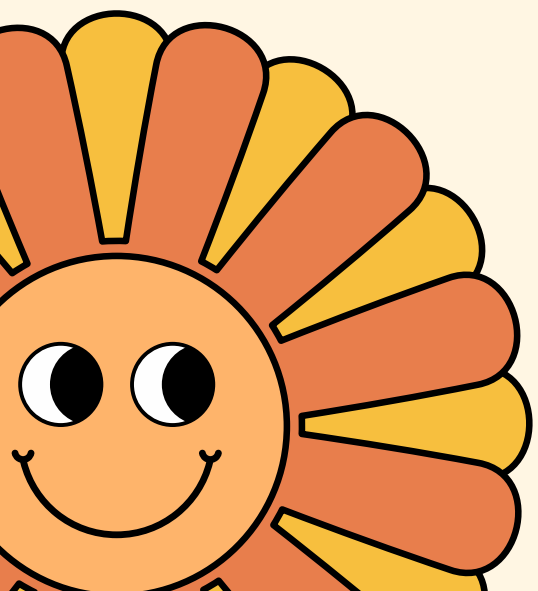
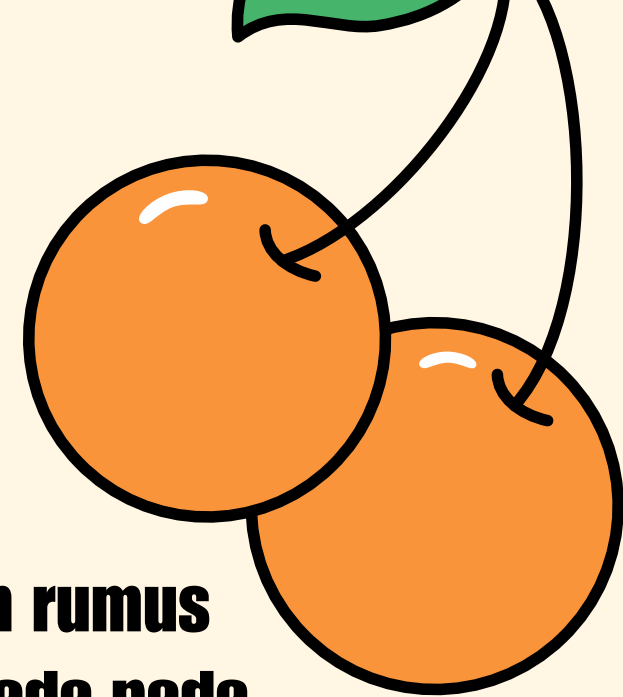
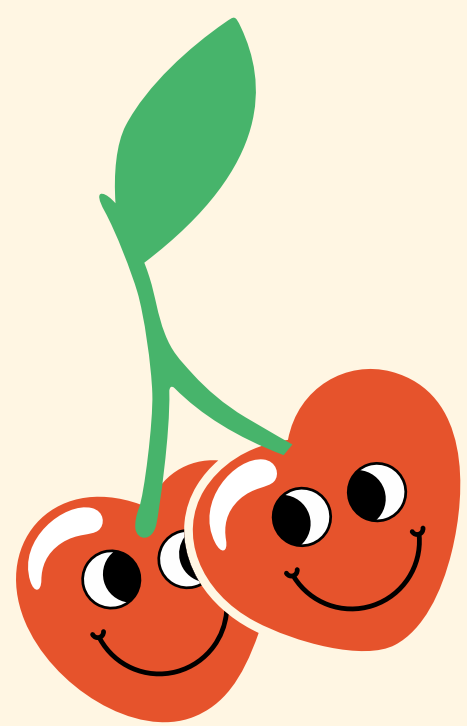
METODE AVERAGE

```
gray_img = np.mean(fix_img, axis=-1)
print(np.array(gray_img))

plt.axis('off')
plt.imshow(gray_img, cmap='gray')
plt.savefig('Average', bbox_inches='tight')
```

KODE INI MENGONVERSI GAMBAR BERWARNA MENJADI CITRA KEABUAN DENGAN MENGAMBIL NILAI RATA-RATA DARI SALURAN WARNA DAN KEMUDIAN MENAMPILKAN SERTA MENYIMPAN HASILNYA DALAM CITRA KEABUAN.





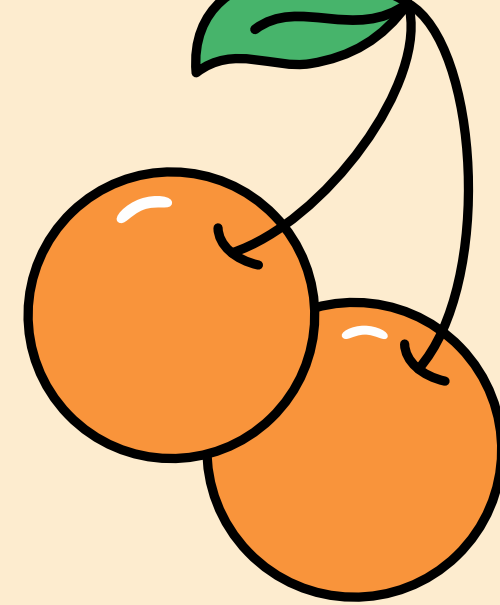
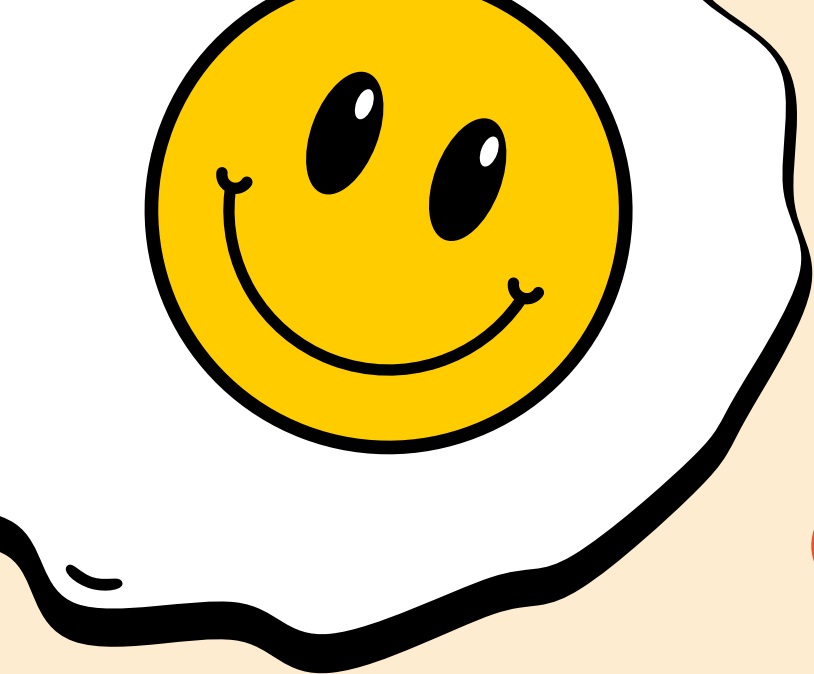
**0 merupakan hasil dari rata-rata nilai dengan rumus
Grayscale = $(R + G + B) / 3$ pada baris 1 yang ada pada
matriks sebelumnya yakni :**

$$(36 + 31 + 38) / 3 = 35$$

$$(29 + 24 + 31) / 3 = 28$$

$$(20 + 15 + 22) / 3 = 19$$

```
[ [ 34.  27.  18. ...  7.   8.   9. ]  
[ 28.  21.  13. ...  6.   7.   8. ]  
[ 24.  17.  10. ...  5.   6.   6. ]  
...  
[ 46.  47.  48. ... 109. 100.  94. ]  
[ 30.  33.  40. ... 114. 109. 104. ]  
[ 29.  31.  40. ... 123. 121. 117. ] ]
```

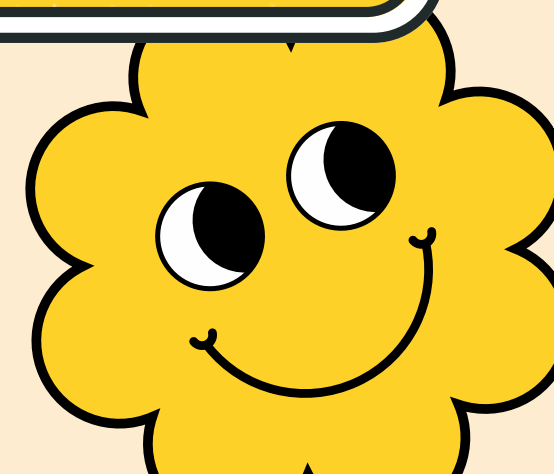


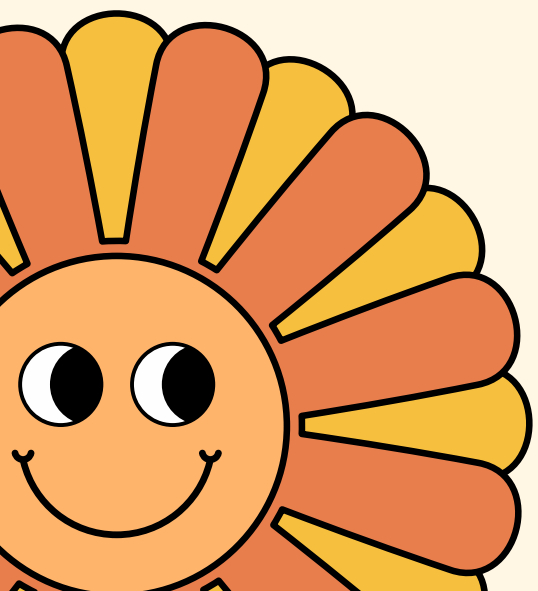
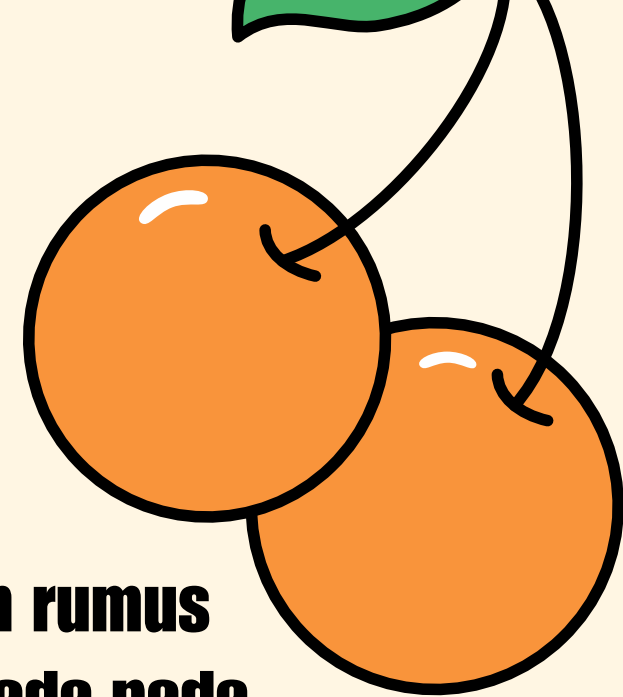
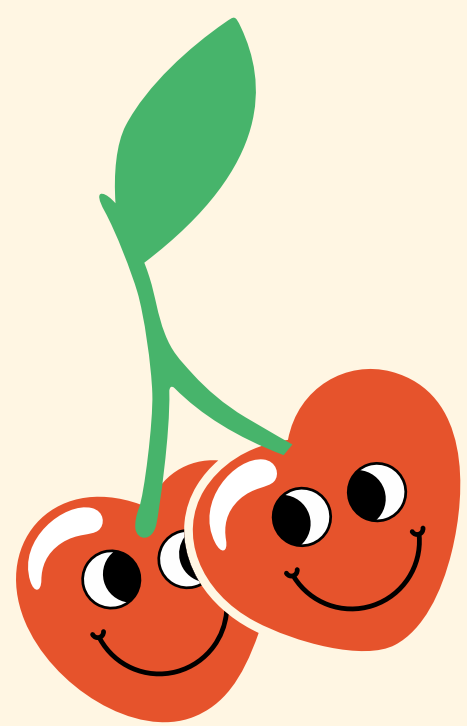
METODE AVERAGE

```
gray_img = np.mean(fix_img, axis=-1)
print(np.array(gray_img))

plt.axis('off')
plt.imshow(gray_img, cmap='gray')
plt.savefig('Average', bbox_inches='tight')
```

KODE INI MENGONVERSI GAMBAR BERWARNA MENJADI CITRA KEABUAN DENGAN MENGAMBIL NILAI RATA-RATA DARI SALURAN WARNA DAN KEMUDIAN MENAMPILKAN SERTA MENYIMPAN HASILNYA DALAM CITRA KEABUAN.





0 merupakan hasil dari rata-rata nilai dengan rumus
Grayscale = (R + G + B) / 3 pada baris 1 yang ada pada
matriks sebelumnya yakni :

$$(115 + 110 + 140) / 3 = 121.6666667$$

$$(125 + 122 + 151) / 3 = 132.6666667$$

$$(115 + 110 + 140) / 3 = 121.6666667$$

```
[ [121.6666667 121.6666667 121.6666667 ... 132.6666667 133.3333333
  134.3333333]
[121.6666667 121.6666667 121.6666667 ... 132.6666667 134.3333333
  134.3333333]
[121.6666667 121.6666667 121.6666667 ... 132.6666667 134.3333333
  134.3333333]
...
[ 13.6666667 13.6666667 12.6666667 ... 13.6666667 13.6666667
  13.6666667]
[ 13.6666667 13.6666667 12.6666667 ... 13.6666667 13.6666667
  13.6666667]
[ 13.6666667 12.6666667 12.6666667 ... 13.6666667 13.6666667
  13.6666667] ]
```

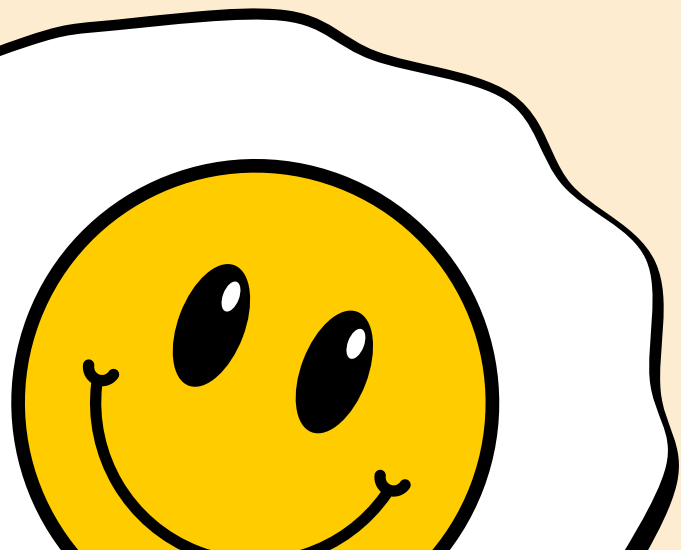
Metode Weighted Average

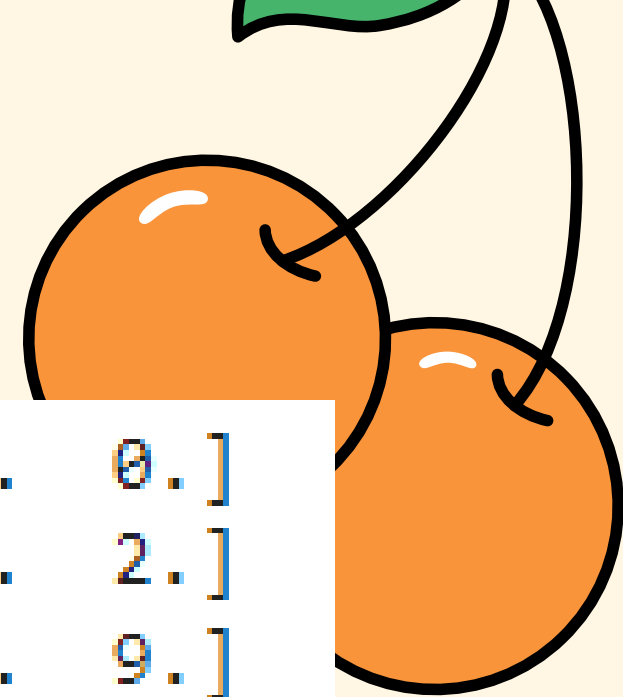
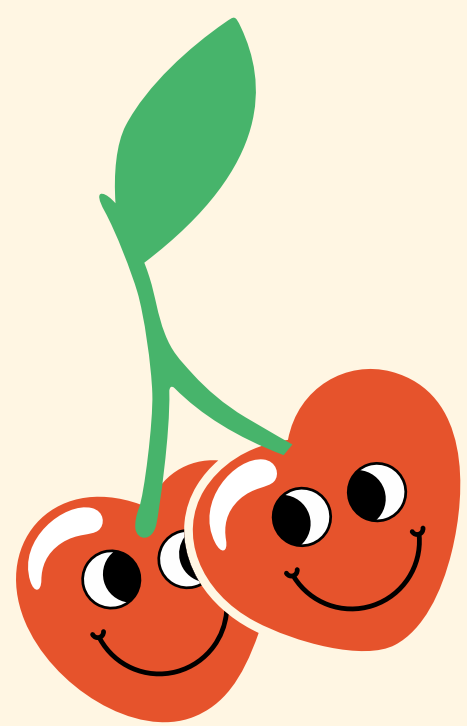


```
wav_img = (0.299*R) + (0.587*G) + (0.114*B)
print(np.array(wav_img))

plt.axis('off')
plt.imshow(wav_img, cmap='gray')
plt.savefig('Weighted Average', bbox_inches= 'tight')
```

KODE INI MENCOBA MEMBUAT CITRA GRAYSCALE (HITAM PUTIH) DARI CITRA BERWARNA DENGAN MENGGUNAKAN FORMULA WEIGHTED AVERAGE. FORMULA TERSEBUT ADALAH $0.299 * \text{NILAI MERAH (R)} + 0.587 * \text{NILAI HIJAU (G)} + 0.114 * \text{NILAI BIRU (B)}$. HASILNYA DILOT MENGGUNAKAN MATPLOTLIB DENGAN `CMAP='GRAY'` UNTUK MENGHASILKAN CITRA GRAYSCALE, DAN KEMUDIAN DISIMPAN SEBAGAI FILE GAMBAR DENGAN NAMA 'WEIGHTED AVERAGE'. `PLT.AXIS('OFF')` DIGUNAKAN UNTUK MENYEMBUNYIKAN SUMBU PADA PLOT.





```
[ [ 0.  0.  0.  ...  0.  0.  0.]  
  [ 1.  1.  1.  ...  1.  1.  2.]  
  [10. 10. 10.  ...  8.  9.  9.]  
  ...  
  [ 3.  3.  3.  ... 12. 12. 12.]  
  [ 3.  3.  3.  ... 11. 12. 13.]  
  [ 2.  2.  2.  ...  9. 10. 10.]]
```

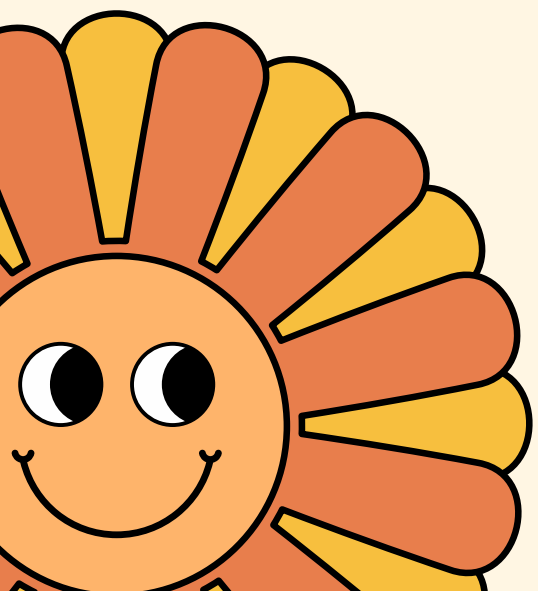
0,0 merupakan hasil dari $(0.299 \times R) + (0.587 \times G) + (0.114 \times B)$ dimana $R = 0$, $G = 0$, $B = 0$ sehingga:

$$(0.299 \times 0) + (0.587 \times 0) + (0.114 \times 0) = 0$$

$$(0.299 \times 0) + (0.587 \times 0) + (0.114 \times 0) = 0$$

$$(0.299 \times 0) + (0.587 \times 0) + (0.114 \times 0) = 0$$

WEIGHTED AVERAGE



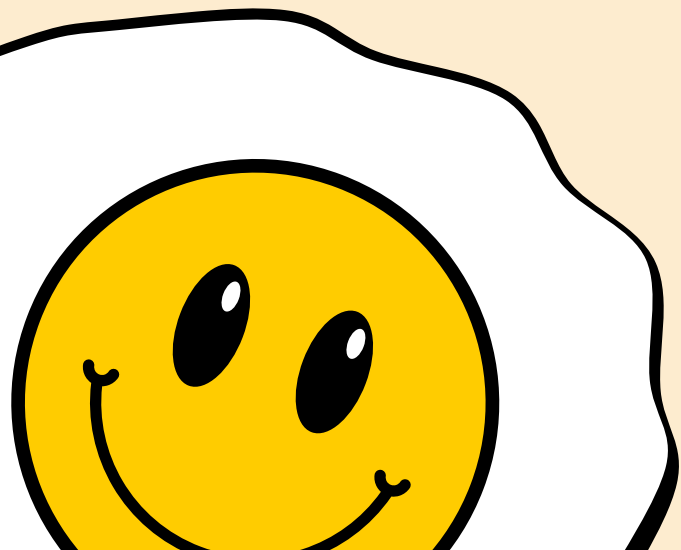


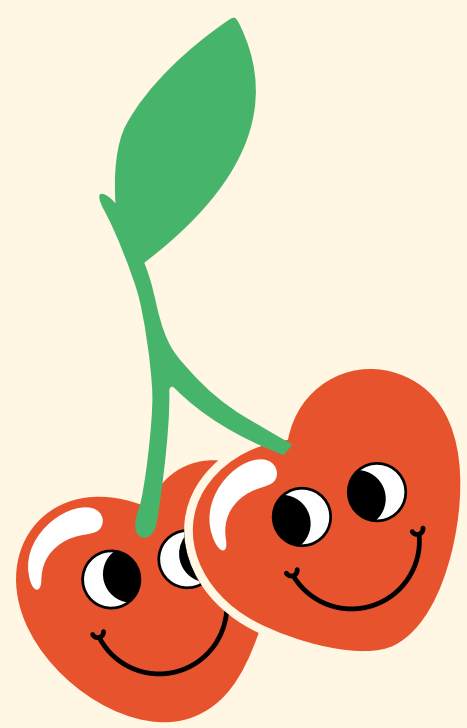
Metode Weighted Average

```
wav_img = (0.299*R) + (0.587*G) + (0.114*B)
print(np.array(wav_img))

plt.axis('off')
plt.imshow(wav_img, cmap='gray')
plt.savefig('Weighted Average', bbox_inches= 'tight')
```

KODE INI MENCOBA MEMBUAT CITRA GRAYSCALE (HITAM PUTIH) DARI CITRA BERWARNA DENGAN MENGGUNAKAN FORMULA WEIGHTED AVERAGE. FORMULA TERSEBUT ADALAH $0.299 * \text{NILAI MERAH (R)} + 0.587 * \text{NILAI HIJAU (G)} + 0.114 * \text{NILAI BIRU (B)}$. HASILNYA DILOT MENGGUNAKAN MATPLOTLIB DENGAN `CMAP='GRAY'` UNTUK MENGHASILKAN CITRA GRAYSCALE, DAN KEMUDIAN DISIMPAN SEBAGAI FILE GAMBAR DENGAN NAMA 'WEIGHTED AVERAGE'. `PLT.AXIS('OFF')` DIGUNAKAN UNTUK MENYEMBUNYIKAN SUMBU PADA PLOT.





```
[[ 34.  27.  18. ...  7.   8.   9.]  
[ 28.  21.  13. ...  6.   7.   8.]  
[ 24.  17.  10. ...  5.   6.   6.]  
...  
[ 46.  47.  48. ... 109. 100.  94.]  
[ 30.  33.  40. ... 114. 109. 104.]  
[ 29.  31.  40. ... 123. 121. 117.]]
```

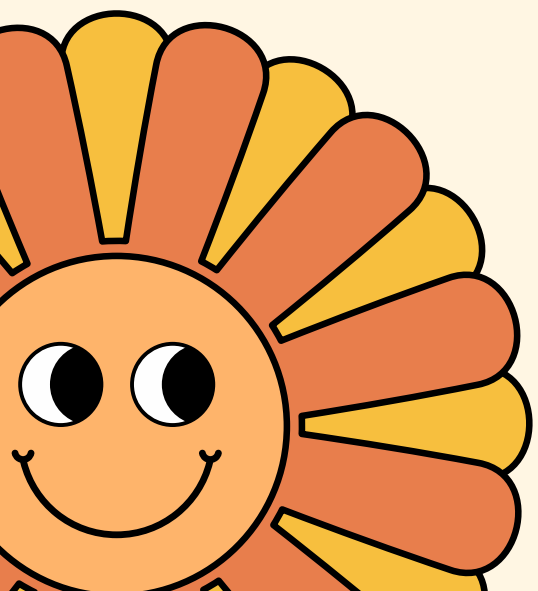
0,0 merupakan hasil dari $(0.299 \times R) + (0.587 \times G) + (0.114 \times B)$ dimana $R = 36$, $G = 31$, $B = 38$ sehingga:

$$(0.299 \times 36) + (0.587 \times 31) + (0.114 \times 38) = 33.293$$

$$(0.299 \times 29) + (0.587 \times 24) + (0.114 \times 31) = 26.293$$

$$(0.299 \times 20) + (0.587 \times 15) + (0.114 \times 22) = 17.293$$

WEIGHTED AVERAGE



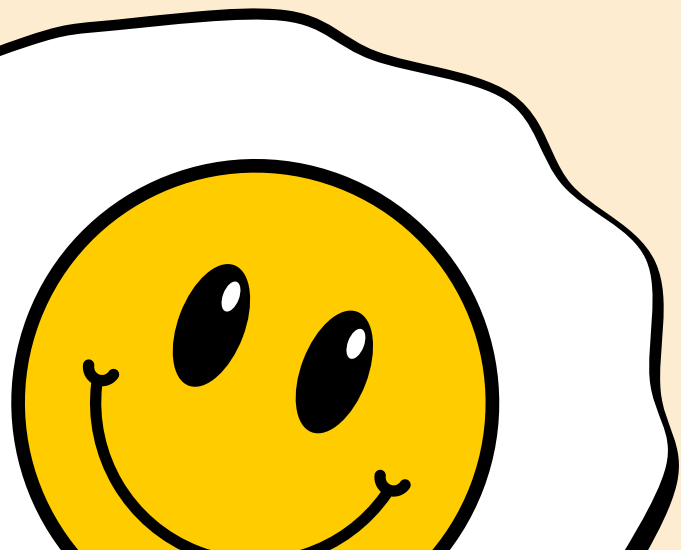


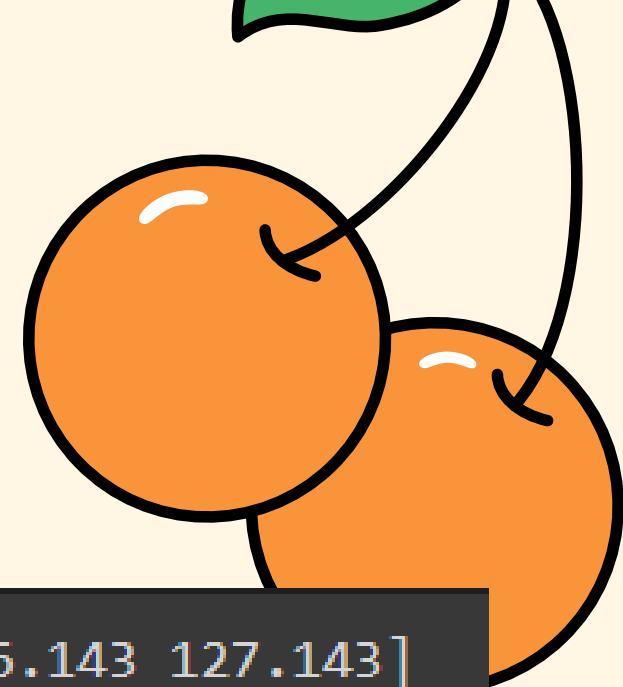
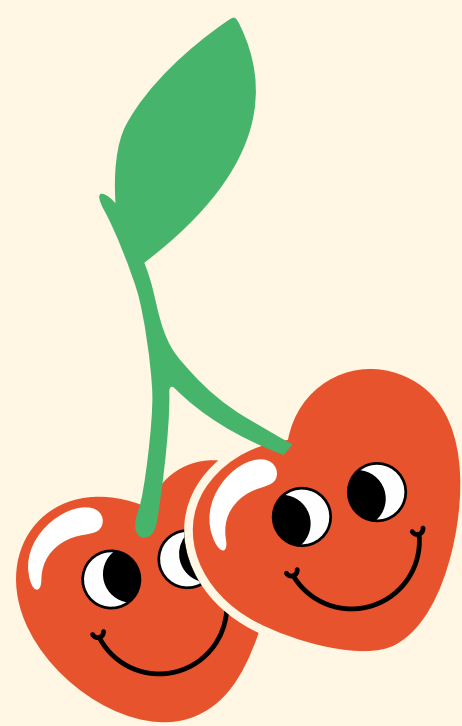
Metode Weighted Average

```
wav_img = (0.299*R) + (0.587*G) + (0.114*B)
print(np.array(wav_img))

plt.axis('off')
plt.imshow(wav_img, cmap='gray')
plt.savefig('Weighted Average', bbox_inches= 'tight')
```

KODE INI MENCOBA MEMBUAT CITRA GRAYSCALE (HITAM PUTIH) DARI CITRA BERWARNA DENGAN MENGGUNAKAN FORMULA WEIGHTED AVERAGE. FORMULA TERSEBUT ADALAH $0.299 * \text{NILAI MERAH (R)} + 0.587 * \text{NILAI HIJAU (G)} + 0.114 * \text{NILAI BIRU (B)}$. HASILNYA DILOT MENGGUNAKAN MATPLOTLIB DENGAN `CMAP='GRAY'` UNTUK MENGHASILKAN CITRA GRAYSCALE, DAN KEMUDIAN DISIMPAN SEBAGAI FILE GAMBAR DENGAN NAMA 'WEIGHTED AVERAGE'. `PLT.AXIS('OFF')` DIGUNAKAN UNTUK MENYEMBUNYIKAN SUMBU PADA PLOT.





```
[ [ 114.915 114.915 114.915 ... 126.203 126.143 127.143 ]  
[ 114.915 114.915 114.915 ... 126.203 127.143 127.143 ]  
[ 114.915 114.915 114.915 ... 125.915 126.855 126.855 ]  
...  
[ 13.31 13.31 12.31 ... 13.31 13.31 13.31 ]  
[ 13.31 13.31 12.31 ... 13.31 13.31 13.31 ]  
[ 13.31 12.31 12.31 ... 13.31 13.31 13.31 ] ]
```

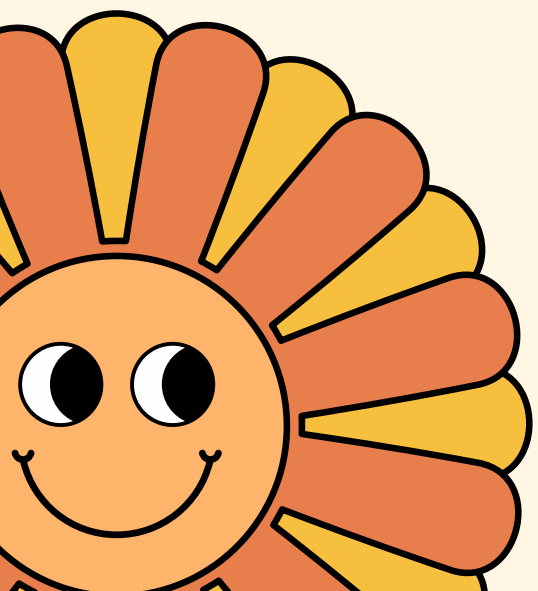
0,0 merupakan hasil dari $(0.299 \times R) + (0.587 \times G) + (0.114 \times B)$ dimana $R = 115$, $G = 110$, $B = 140$ sehingga:

$$(0.299 \times 115) + (0.587 \times 110) + (0.114 \times 140) = 114.915$$

$$(0.299 \times 0) + (0.587 \times 0) + (0.114 \times 0) = 0$$

$$(0.299 \times 0) + (0.587 \times 0) + (0.114 \times 0) = 0$$

WEIGHTED AVERAGE



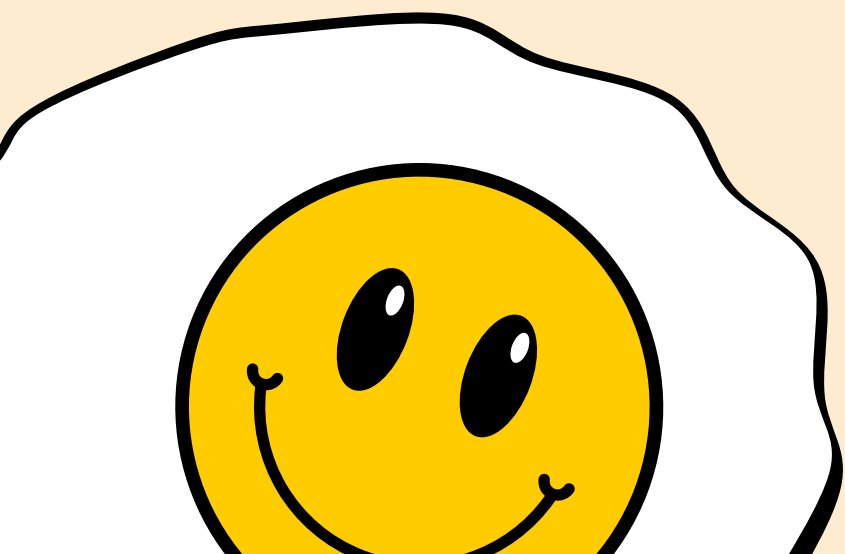
Metode Luminosity

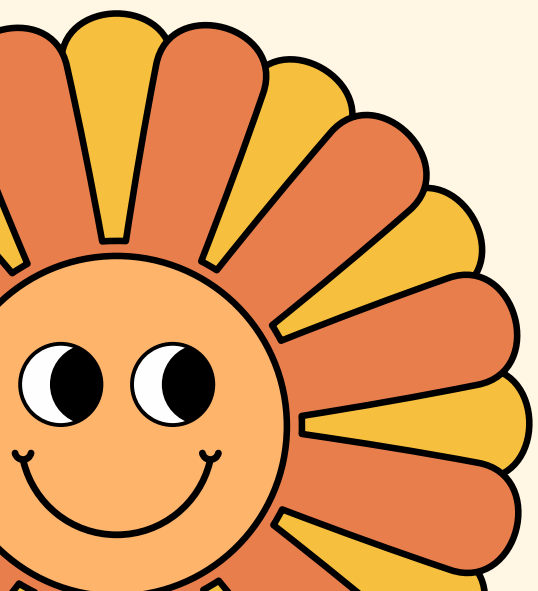
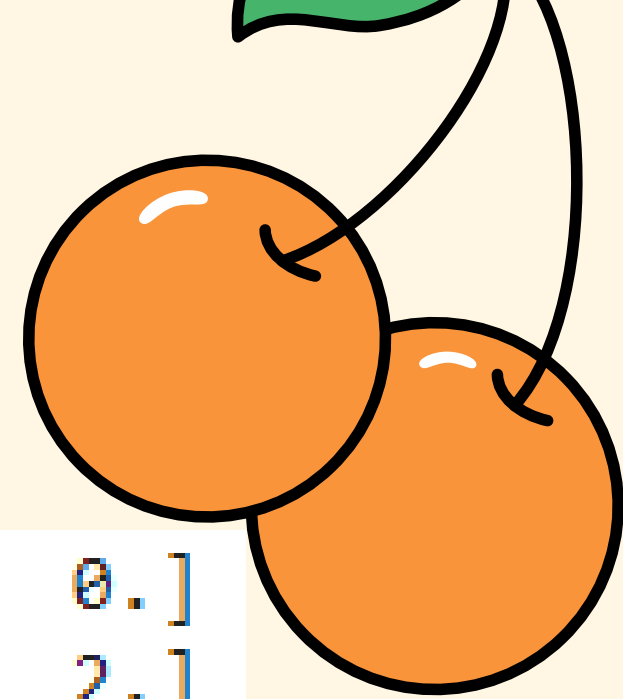
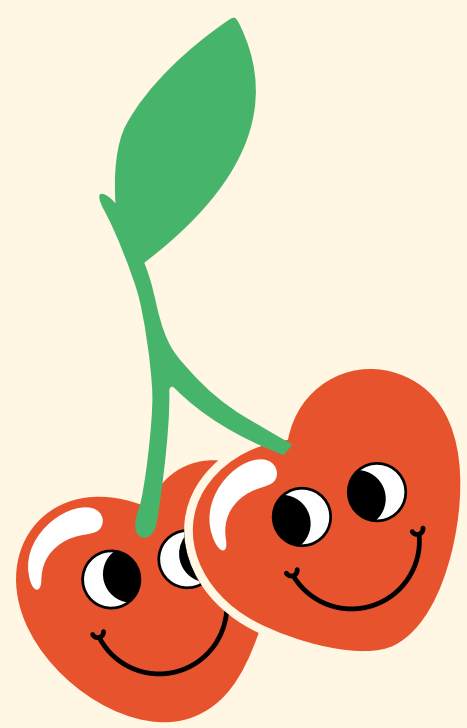


```
lumi_img = (0.2126*R) + (0.7152*G) + (0.0722*B)
print(np.array(lumi_img))

plt.axis('off')
plt.imshow(lumi_img, cmap='gray')
plt.savefig('Luminosity', bbox_inches= 'tight')
```

KODE INI UNTUK MENGONVERSI GAMBAR WARNA MENJADI CITRA KEABUAN. METODE LUMINOSITY MENGGUNAKAN BOBOT TERTENTU UNTUK SETIAP SALURAN WARNA (R, G, B) UNTUK MENGHITUNG NILAI INTENSITAS KEABUAN YANG LEBIH SESUAI DENGAN PERSEPSI MATA MANUSIA TERHADAP WARNA





LUMINOSITY

```
[ [ 0.  0.  0.  ...  0.  0.  0.]  
  [ 1.  1.  1.  ...  1.  1.  2.]  
  [10. 10. 10.  ...  8.  9.  9.]  
  ...  
  [ 3.  3.  3.  ... 12. 12. 12.]  
  [ 3.  3.  3.  ... 11. 12. 13.]  
  [ 2.  2.  2.  ...  9. 10. 10.]]
```

0,0 merupakan hasil dari $(0.2126 \times R) + (0.7152 \times G) + (0.0722 \times B)$ dimana $R = 0$, $G = 0$, $B = 0$ sehingga:

$$(0.2126 \times 0) + (0.7152 \times 0) + (0.0722 \times 0) = 0$$

$$(0.2126 \times 0) + (0.7152 \times 0) + (0.0722 \times 0) = 0$$

$$(0.2126 \times 0) + (0.7152 \times 0) + (0.0722 \times 0) = 0$$

Metode Luminosity

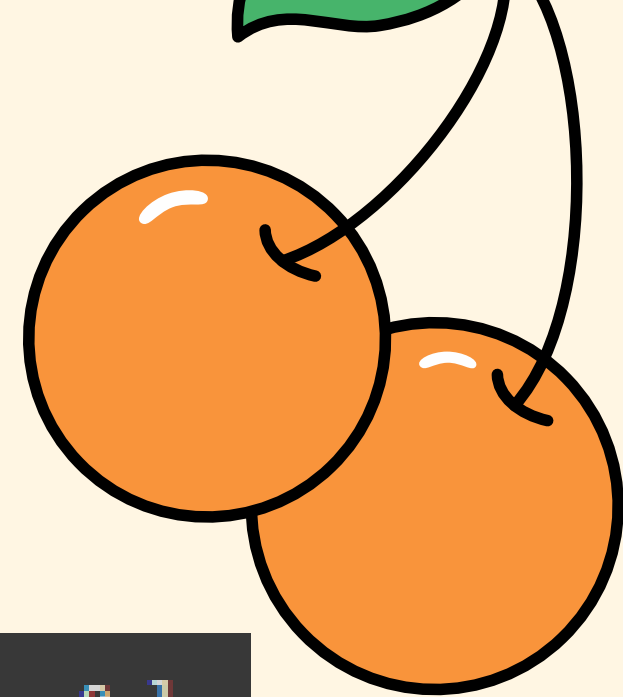
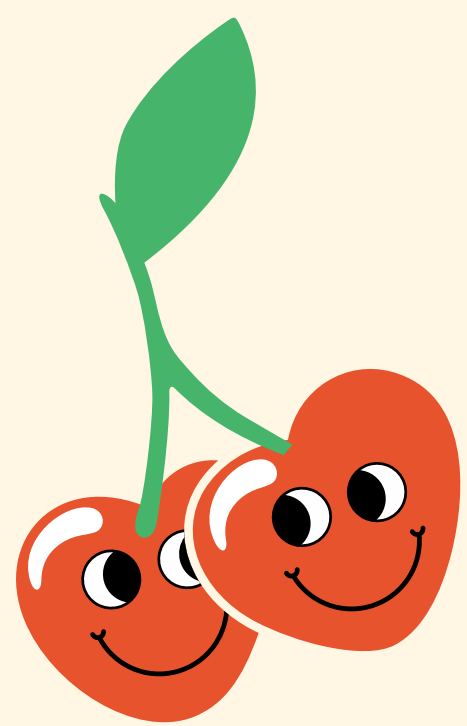


```
lumi_img = (0.2126*R) + (0.7152*G) + (0.0722*B)
print(np.array(lumi_img))

plt.axis('off')
plt.imshow(lumi_img, cmap='gray')
plt.savefig('Luminosity', bbox_inches='tight')
```

KODE INI UNTUK MENGONVERSI GAMBAR WARNA MENJADI CITRA KEABUAN. METODE LUMINOSITY MENGGUNAKAN BOBOT TERTENTU UNTUK SETIAP SALURAN WARNA (R, G, B) UNTUK MENGHITUNG NILAI INTENSITAS KEABUAN YANG LEBIH SESUAI DENGAN PERSEPSI MATA MANUSIA TERHADAP WARNA





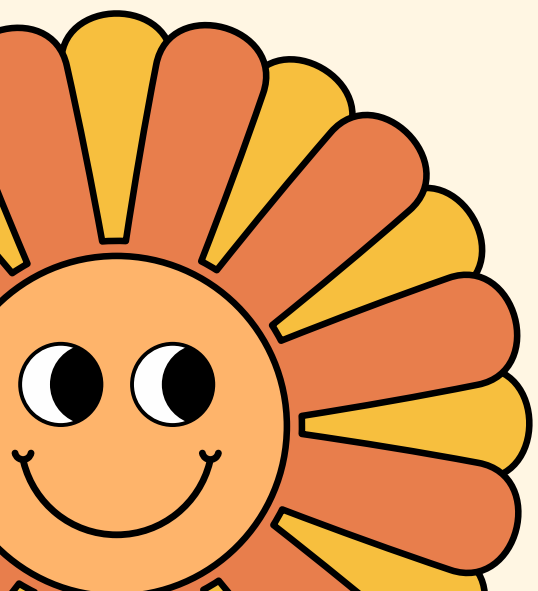
```
[[ 34.  27.  18. ...  7.   8.   9.]  
 [ 28.  21.  13. ...  6.   7.   8.]  
 [ 24.  17.  10. ...  5.   6.   6.]  
 ...  
 [ 46.  47.  48. ... 109. 100.  94.]  
 [ 30.  33.  40. ... 114. 109. 104.]  
 [ 29.  31.  40. ... 123. 121. 117.]]
```

0,0 merupakan hasil dari $(0.2126 \times R) + (0.7152 \times G) + (0.0722 \times B)$ dimana $R = 0$, $G = 0$, $B = 0$ sehingga:

$(0.2126 \times 36) + (0.7152 \times 31) + (0.0722 \times 38) = 32.5684$

$(0.2126 \times 29) + (0.7152 \times 24) + (0.0722 \times 31) = 45.7122$

$(0.2126 \times 20) + (0.7152 \times 15) + (0.0722 \times 22) = 16.5684$



LUMINOSITY

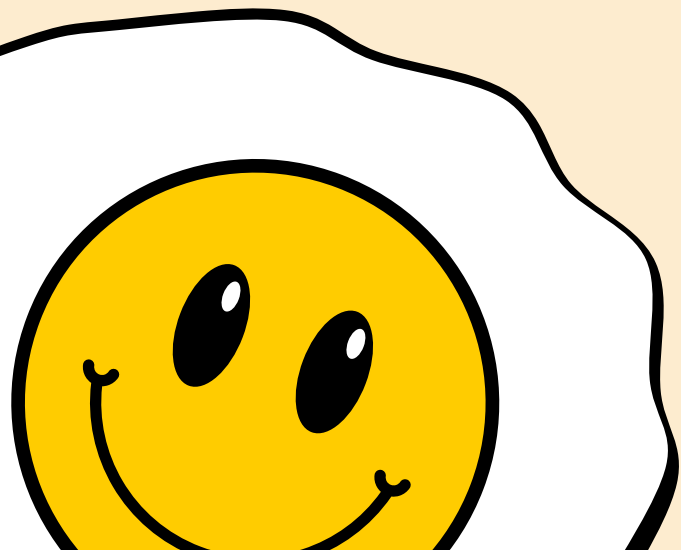
Metode Luminosity

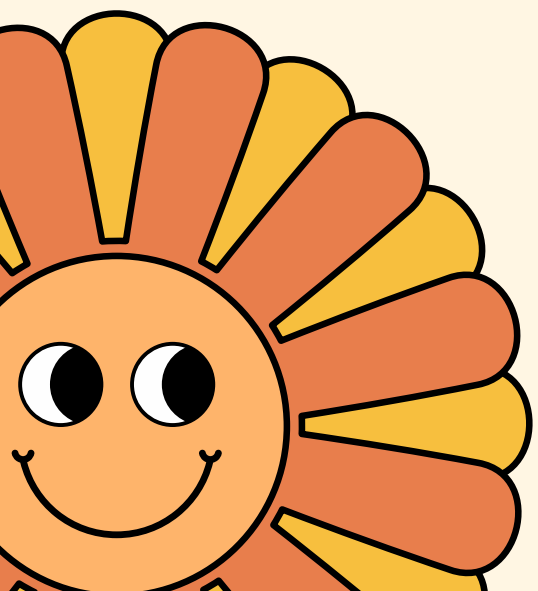
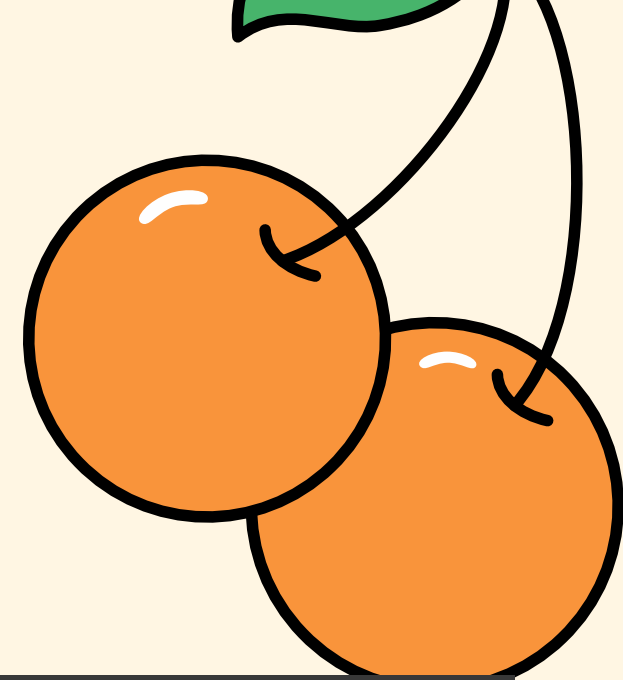
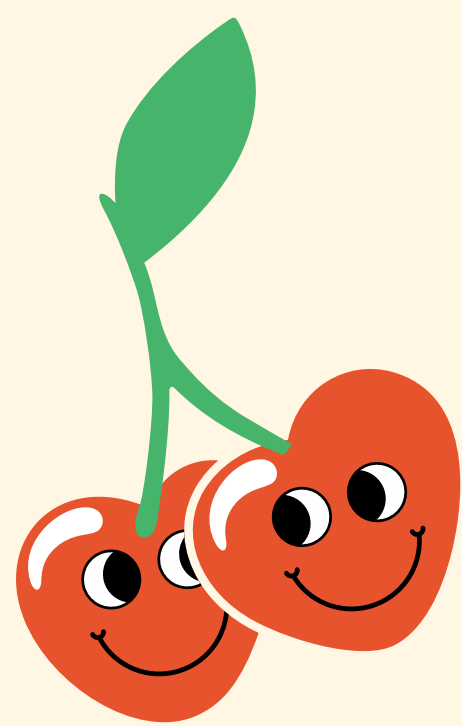


```
lumi_img = (0.2126*R) + (0.7152*G) +(0.0722*B)
print(np.array(gray_img))

plt.axis('off')
plt.imshow(lumi_img, cmap='gray')
plt.savefig('Luminosity', bbox_inches= 'tight')
```

KODE INI UNTUK MENGONVERSI GAMBAR WARNA MENJADI CITRA KEABUAN. METODE LUMINOSITY MENGGUNAKAN BOBOT TERTENTU UNTUK SETIAP SALURAN WARNA (R, G, B) UNTUK MENGHITUNG NILAI INTENSITAS KEABUAN YANG LEBIH SESUAI DENGAN PERSEPSI MATA MANUSIA TERHADAP WARNA





LUMINOSITY

```
[ [ 113.229  113.229  113.229  ... 124.7316 124.3734 125.3734 ]  
[ 113.229  113.229  113.229  ... 124.7316 125.3734 125.3734 ]  
[ 113.229  113.229  113.229  ... 124.229  124.8708 124.8708 ]  
...  
[ 12.9226  12.9226  11.9226 ... 12.9226  12.9226  12.9226 ]  
[ 12.9226  12.9226  11.9226 ... 12.9226  12.9226  12.9226 ]  
[ 12.9226  11.9226  11.9226 ... 12.9226  12.9226  12.9226 ] ]
```

0,0 merupakan hasil dari $(0.2126 \times R) + (0.7152 \times G) + (0.0722 \times B)$ dimana $R = 115$, $G = 110$, $B = 140$ sehingga:

$(0.2126 \times 115) + (0.7152 \times 110) + (0.0722 \times 140) = 113.229$

$(0.2126 \times 0) + (0.7152 \times 0) + (0.0722 \times 0) = 0$

$(0.2126 \times 0) + (0.7152 \times 0) + (0.0722 \times 0) = 0$

გეგმვა
კავშირ

