



**Hello!**

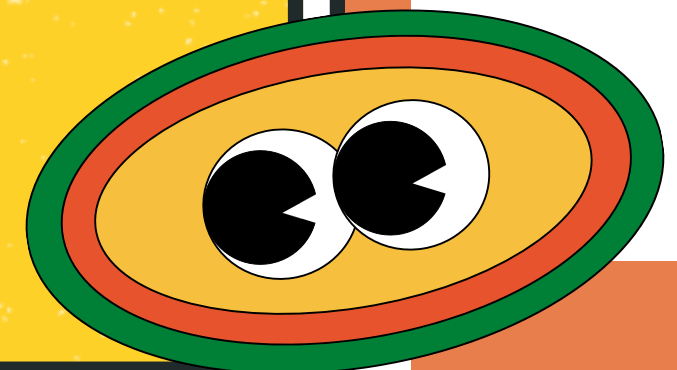
# *Aplikasi pada Aljabar Linear : Konversi Citra RGB ke Grayscale*



# *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))
```



# FOTO ASLI



Dikriani.jpg



Alfira Zalsabila.jpg



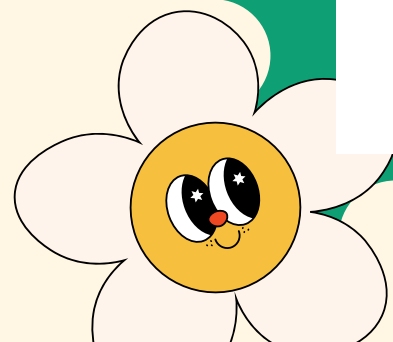
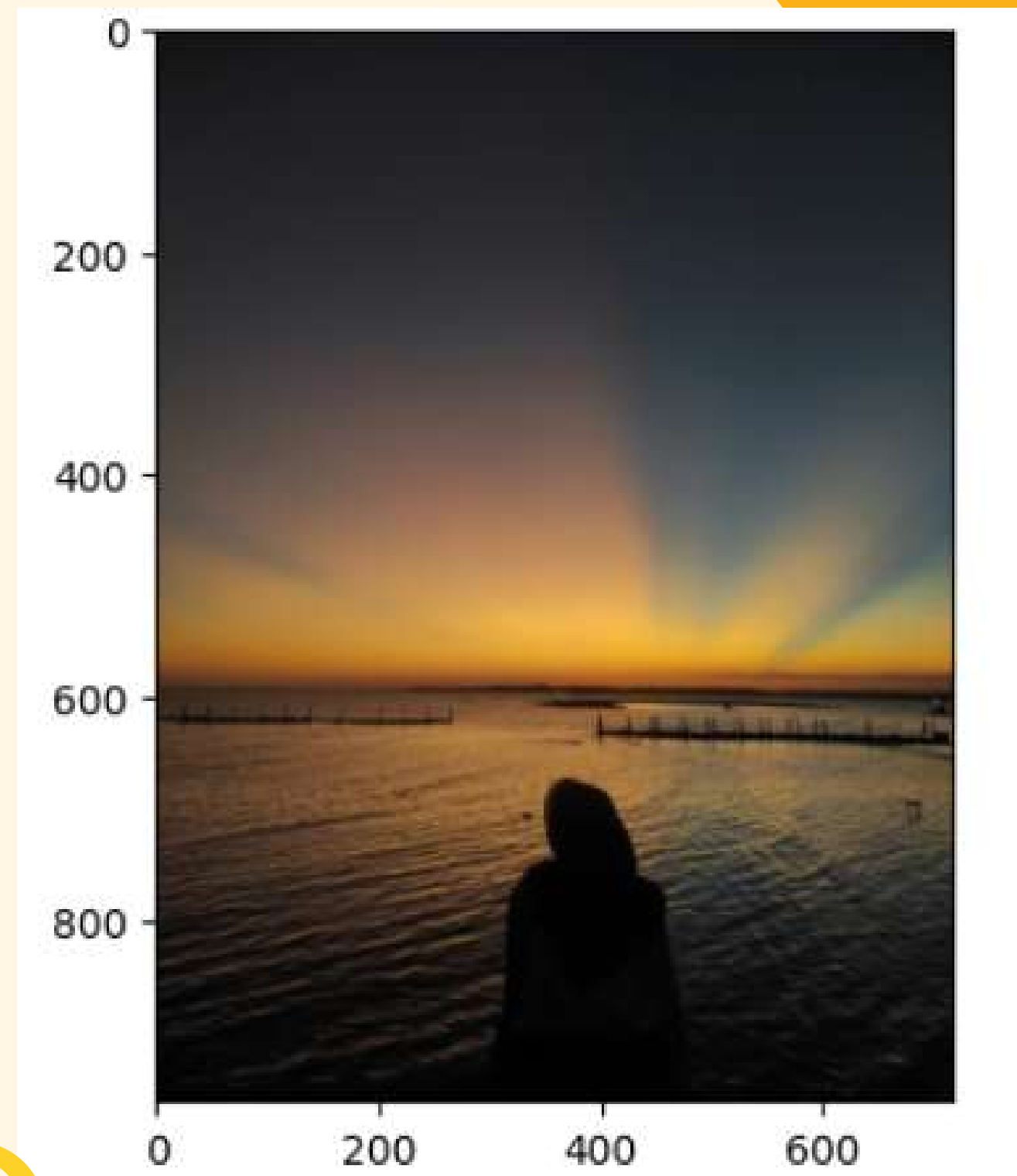
Tristia Desiana.jpg



```
(964, 716, 3)
```

```
[[[ 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]  
 ...  
 [ 1  1  1]  
 [ 0  0  2]  
 [ 1  1  3]]
```

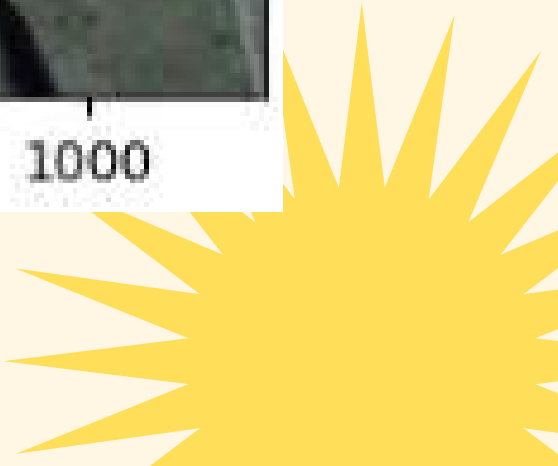
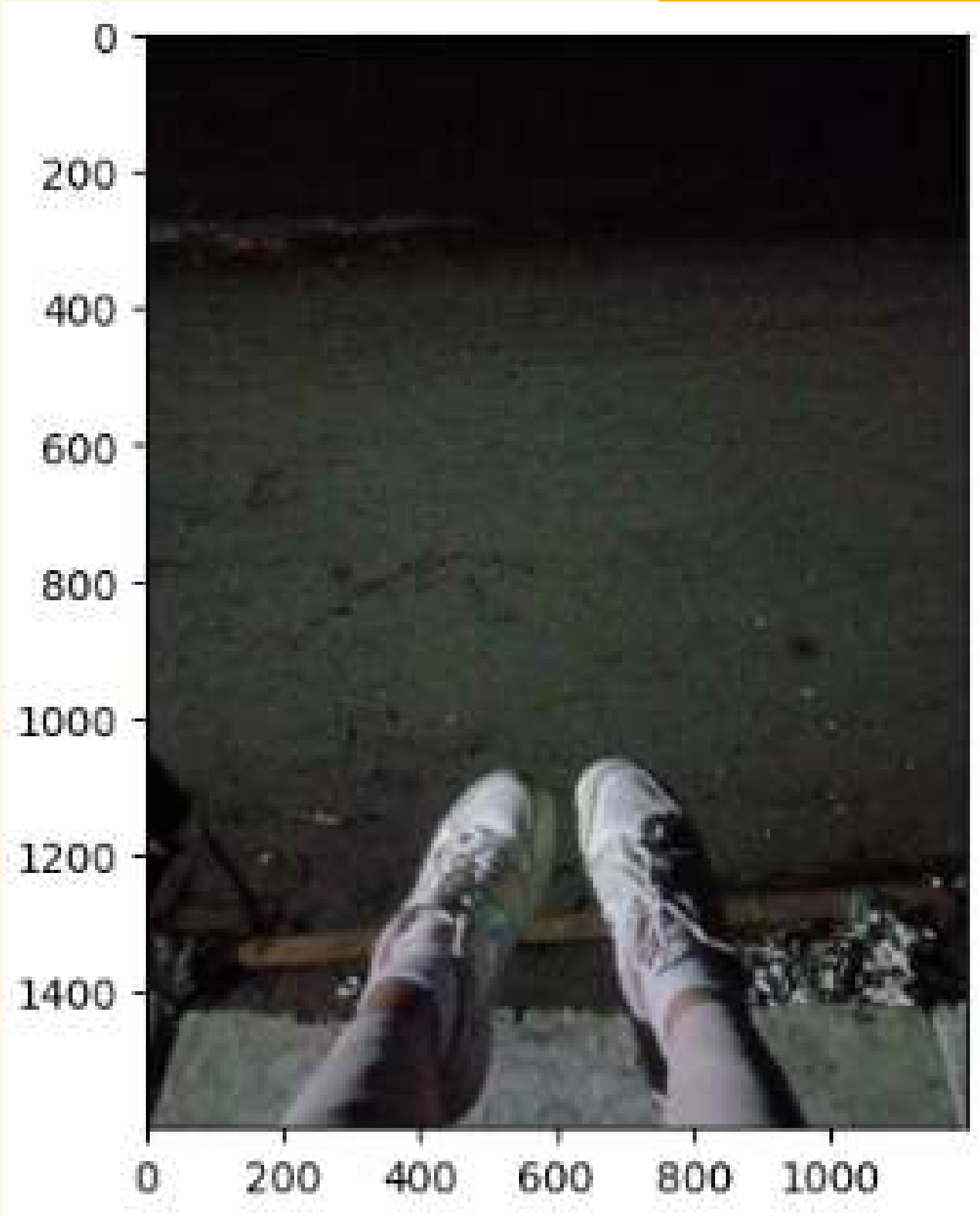
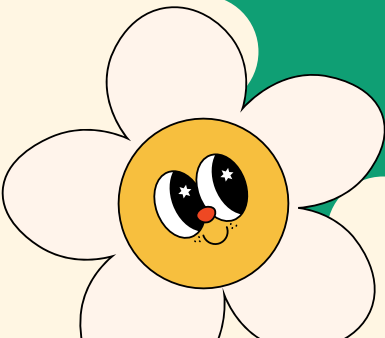




```
(1600, 1200, 3)
[[[ 36  31  38]
   [ 29  24  31]
   [ 20  15  22]
   ...
   [ 10   5   9]
   [ 11   6  10]
   [ 12   7  11]]

 [[ 30  25  32]
   [ 23  18  25]
   [ 15  10  17]
   ...
   [  9   4   8]
   [ 10   5   9]
   [ 11   6  10]]

 [[ 26  21  27]
   [ 19  14  20]
   [ 12   7  13]
   ...
   [  8   3   7]
   [  9   4   8]
   [  9   4   8]]
```

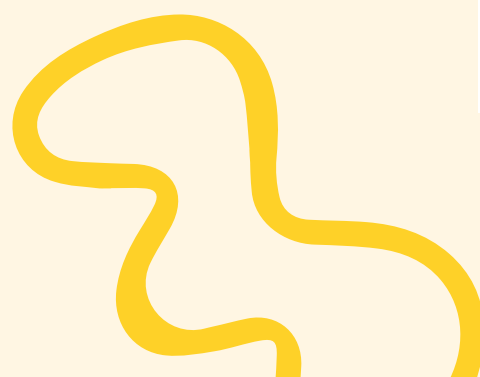
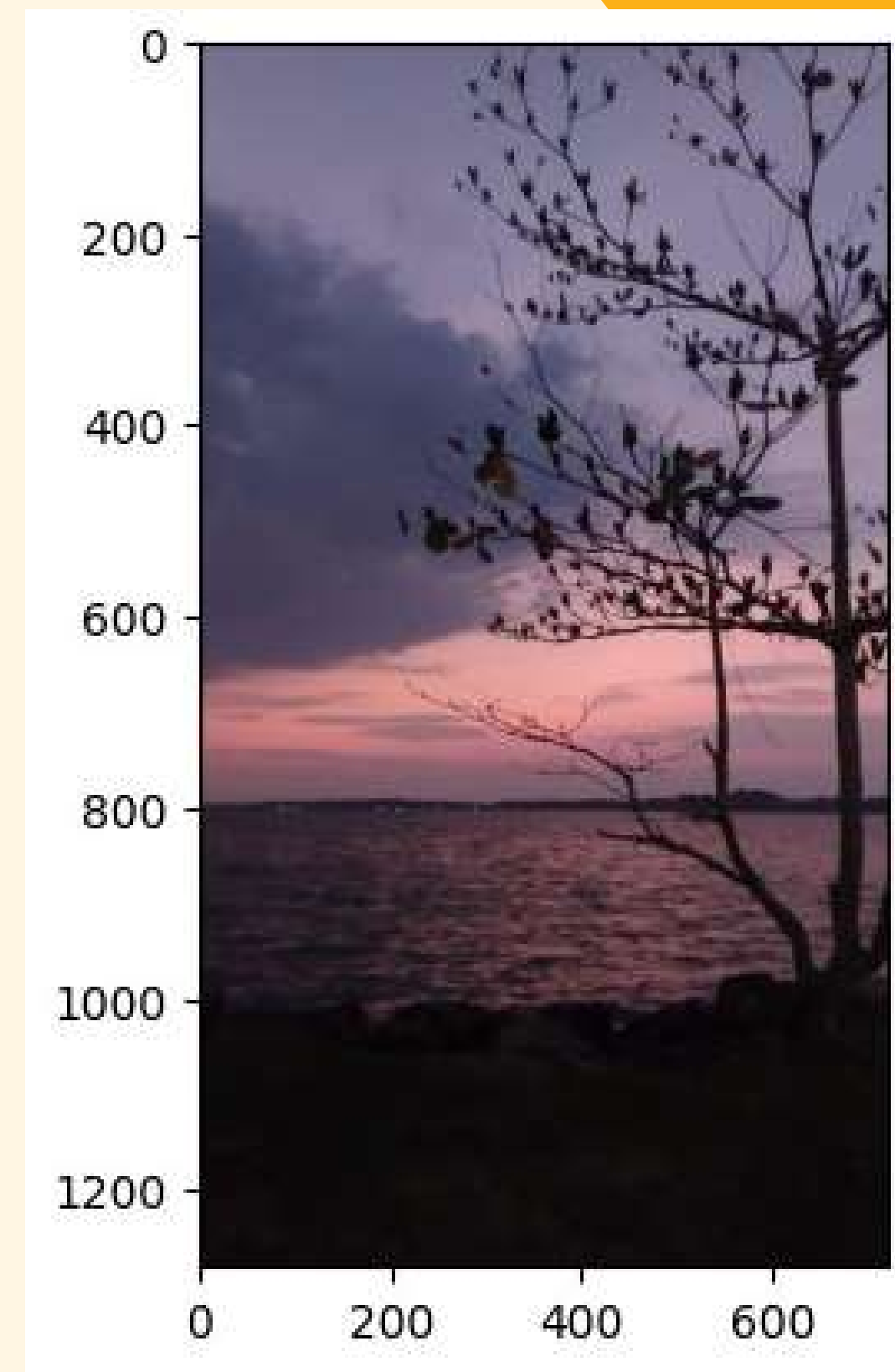
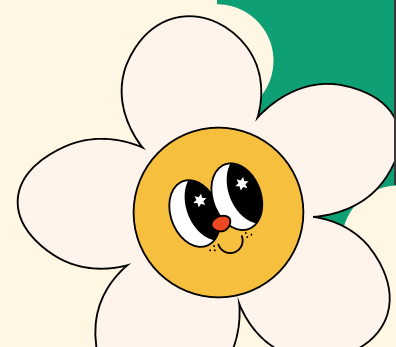




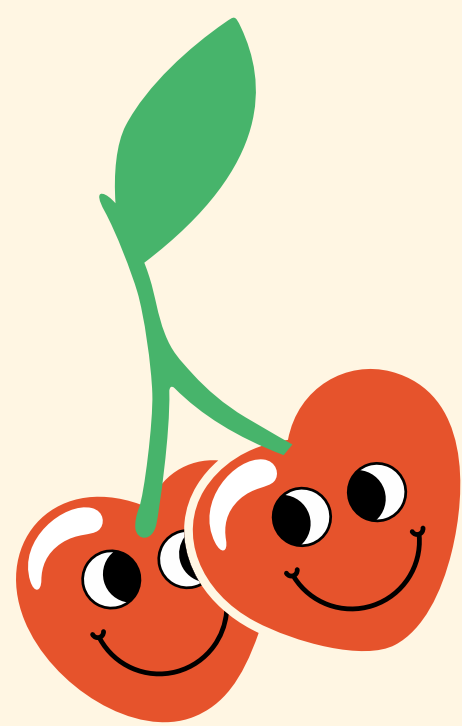
```
(1280, 720, 3)
[[[115 110 140]
  [115 110 140]
  [115 110 140]
  ...
  [125 122 151]
  [126 121 153]
  [127 122 154]]

[[[115 110 140]
  [115 110 140]
  [115 110 140]
  ...
  [125 122 151]
  [127 122 154]
  [127 122 154]]

[[[115 110 140]
  [115 110 140]
  [115 110 140]
  ...
  [126 121 151]
  [128 121 154]
  [128 121 154]]
```



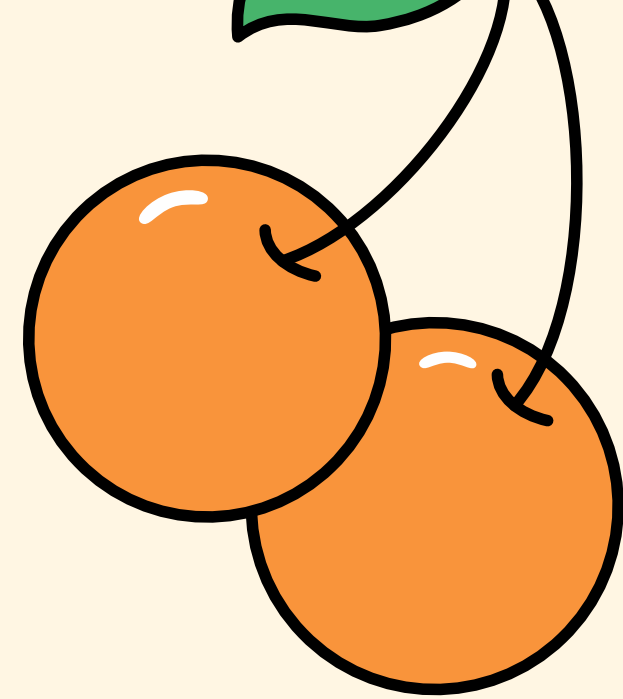




## Metode Lightness

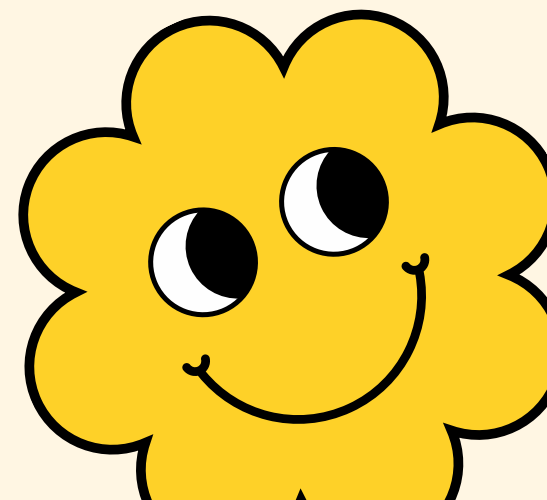
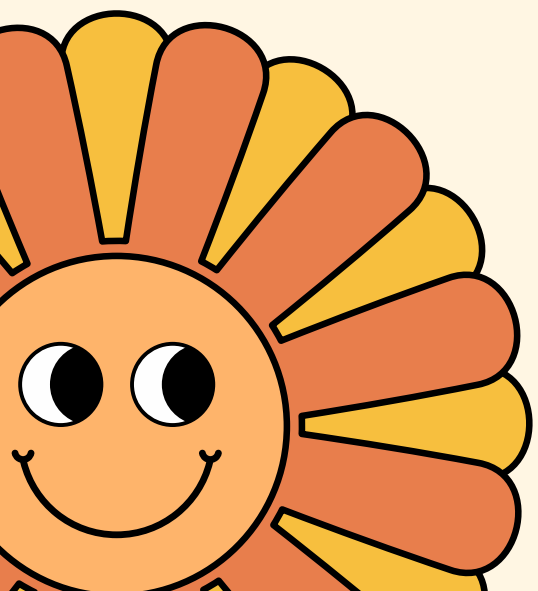
```
fix_img[:] = np.max(fix_img, axis=-1, keepdims=1)/2 + np.min(fix_img, axis=-1, keepdims=1)/2  
print(np.array(fix_img[:]))
```

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

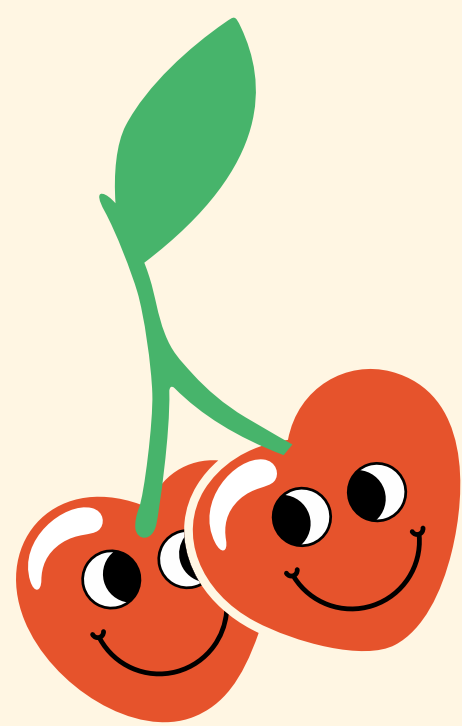


## Outputnya

```
[[[ 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]]
```



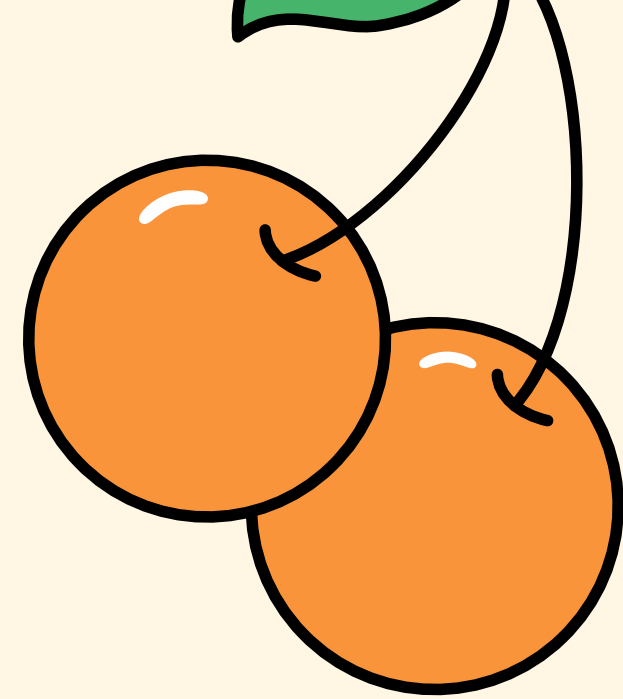




## Metode Lightness

```
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[:]))

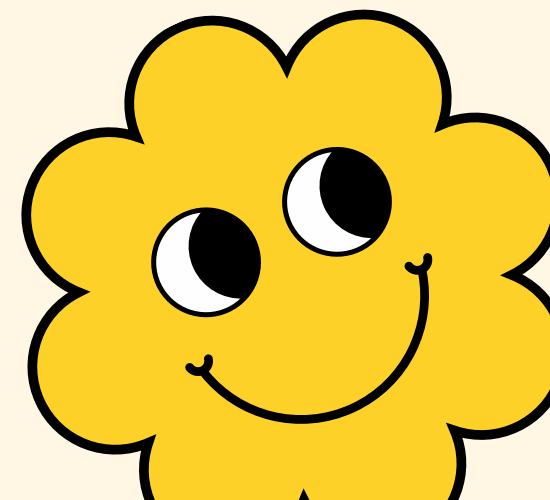
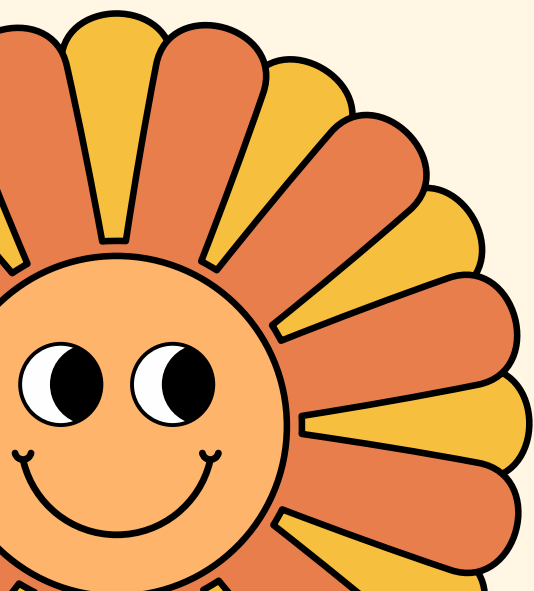
plt.axis('off')
plt.imshow(fix_img[:])
plt.savefig('Metode Lightness')
```

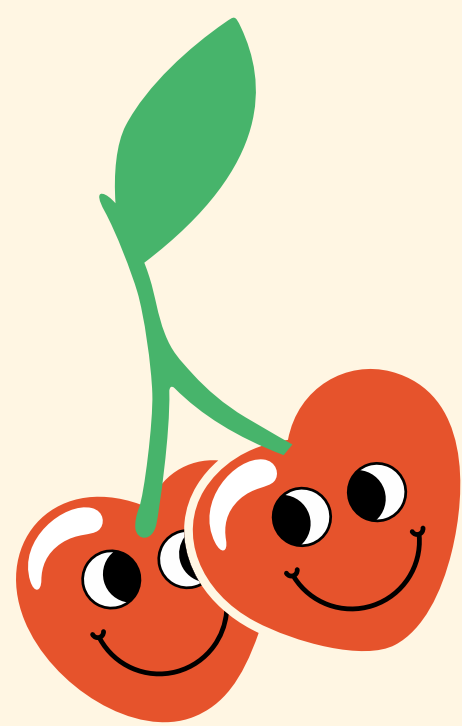


## Outputnya

```
➡ [[ [ 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]
      ...
```

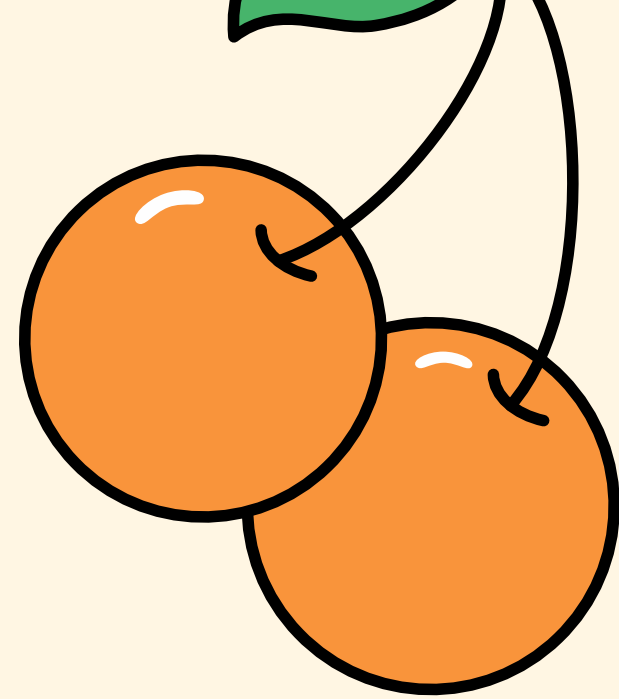




## Metode Lightness

```
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')
```

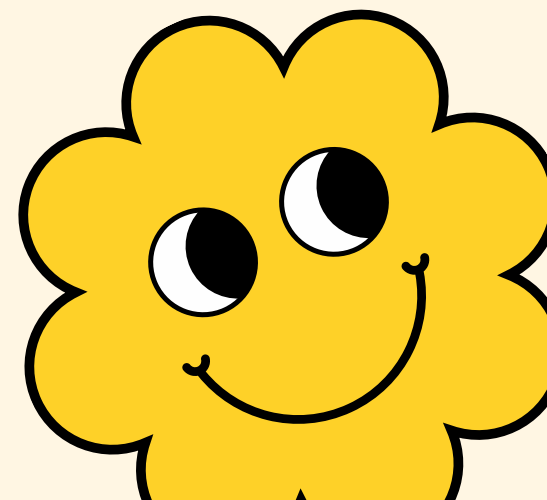
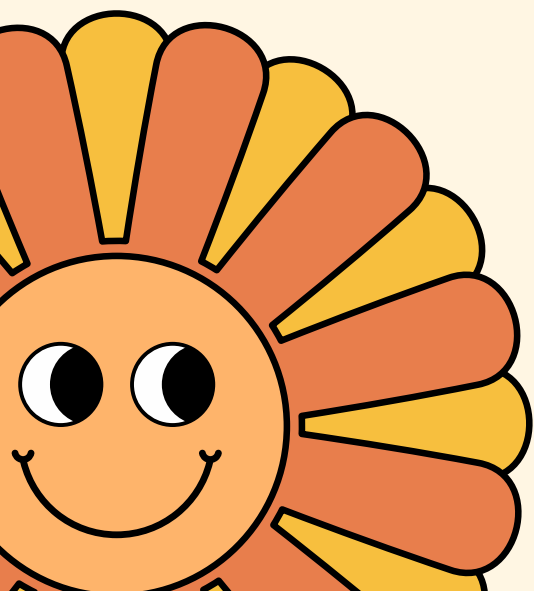


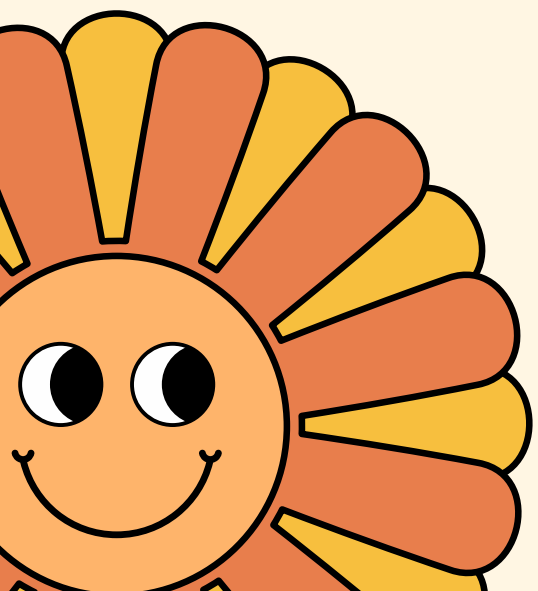
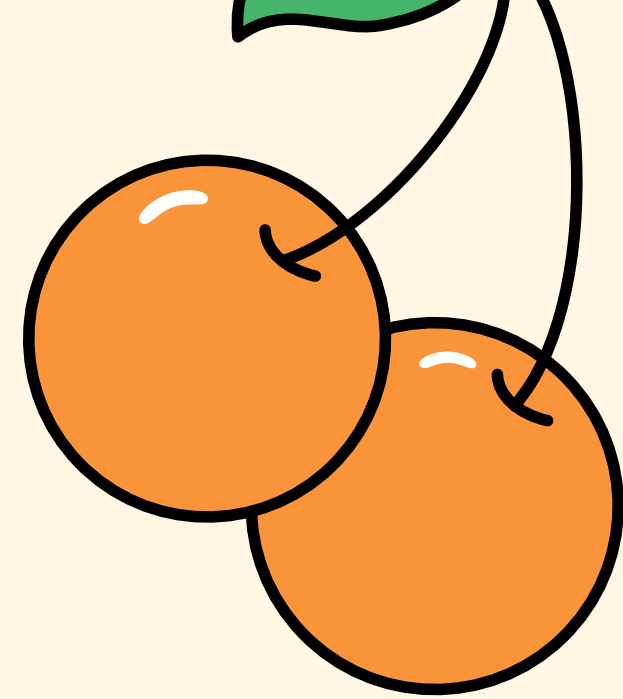
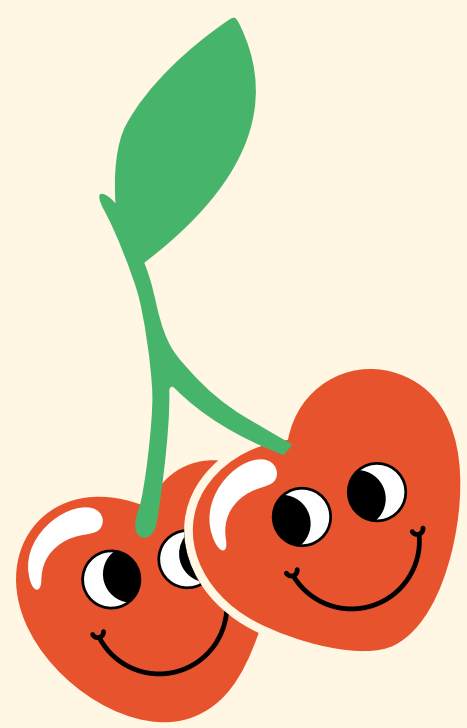
## Outputnya

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

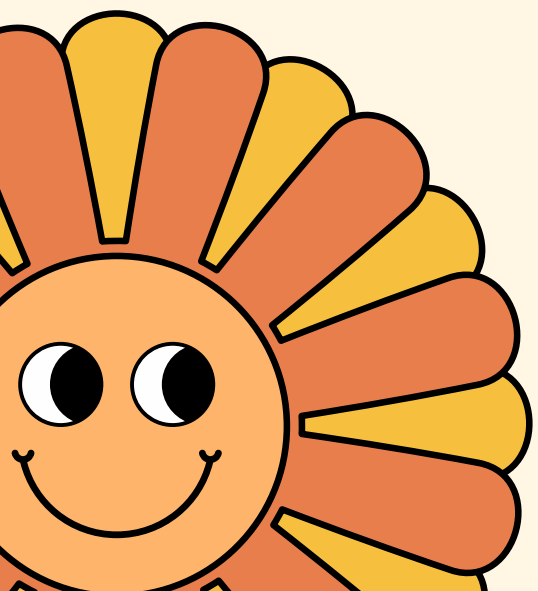
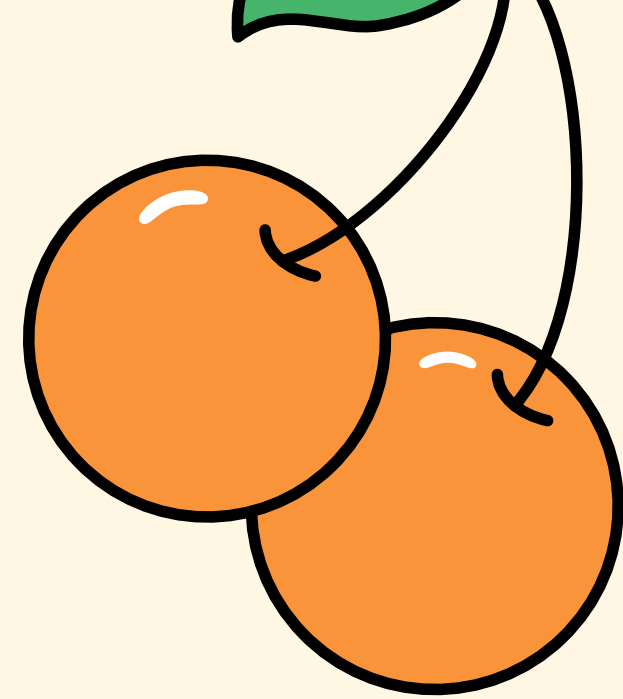
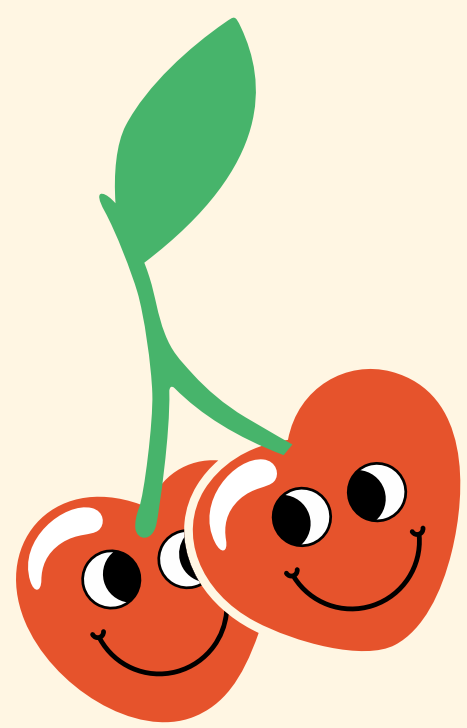
[[125 125 125]
 [125 125 125]
 [125 125 125]
  ...
  [136 136 136]
  [138 138 138]
  [138 138 138]]

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



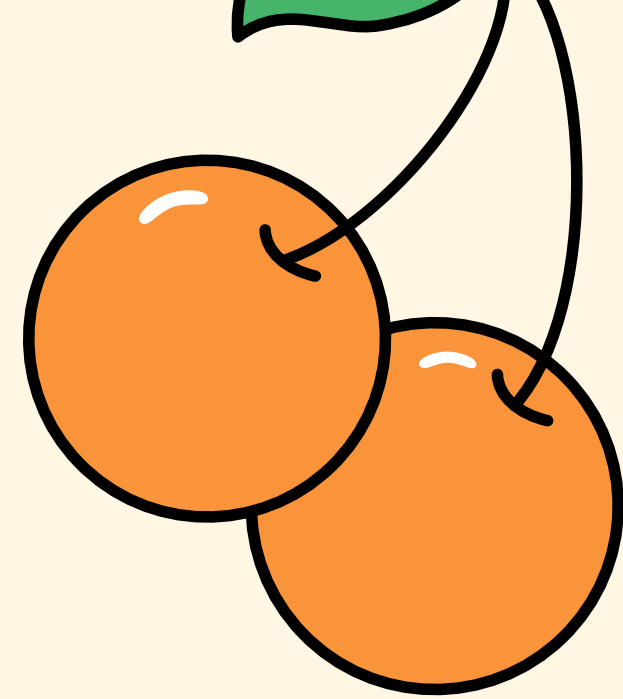
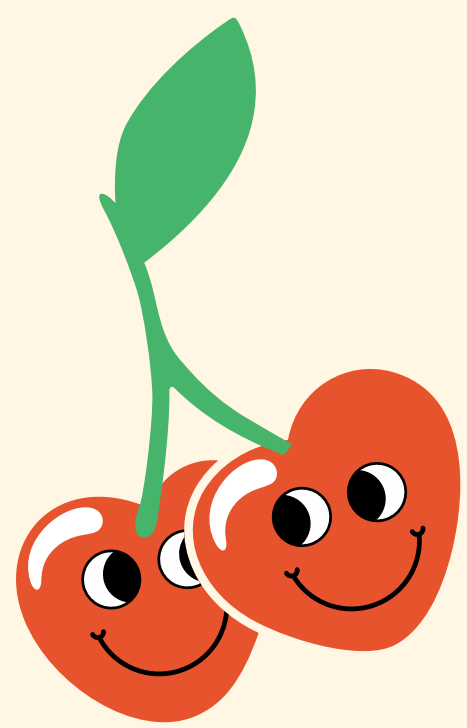


*Hasil fotonya*

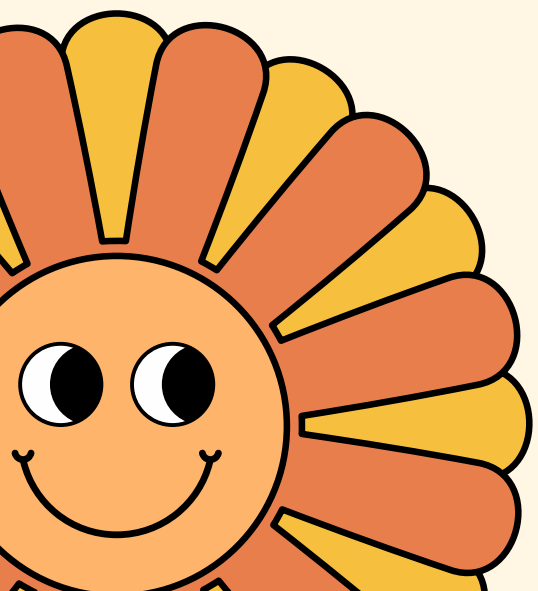


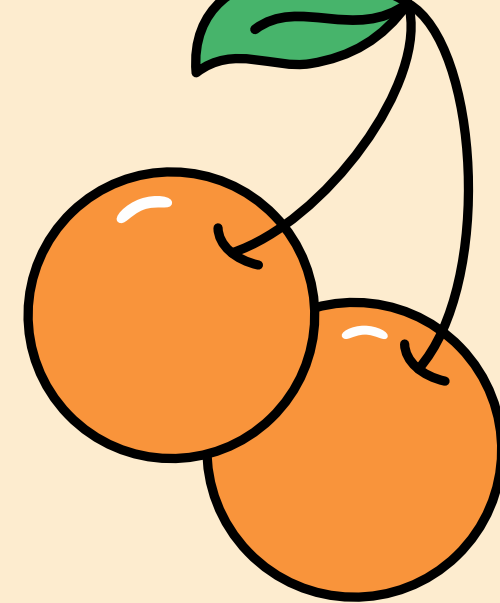
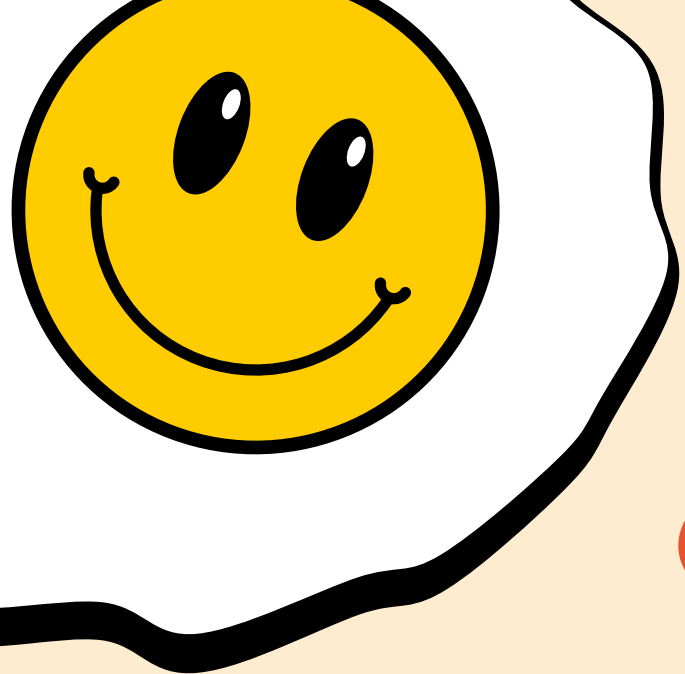
*Hasil fotonya*





*Hasil fotonya*



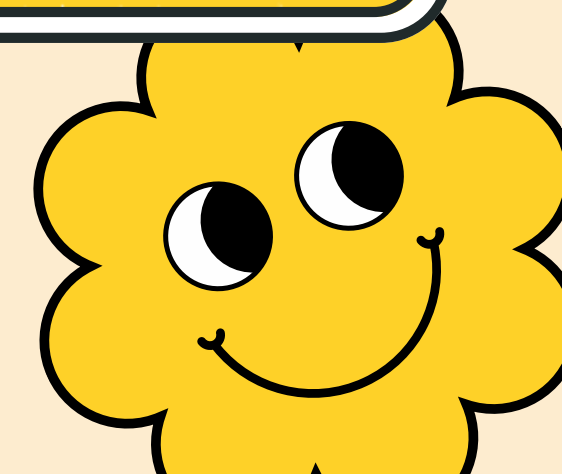


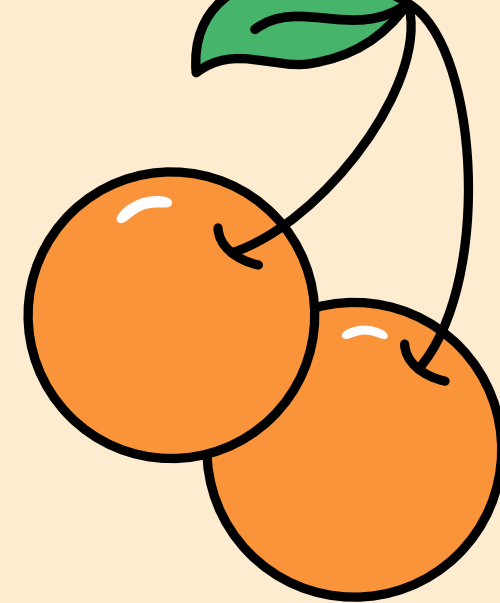
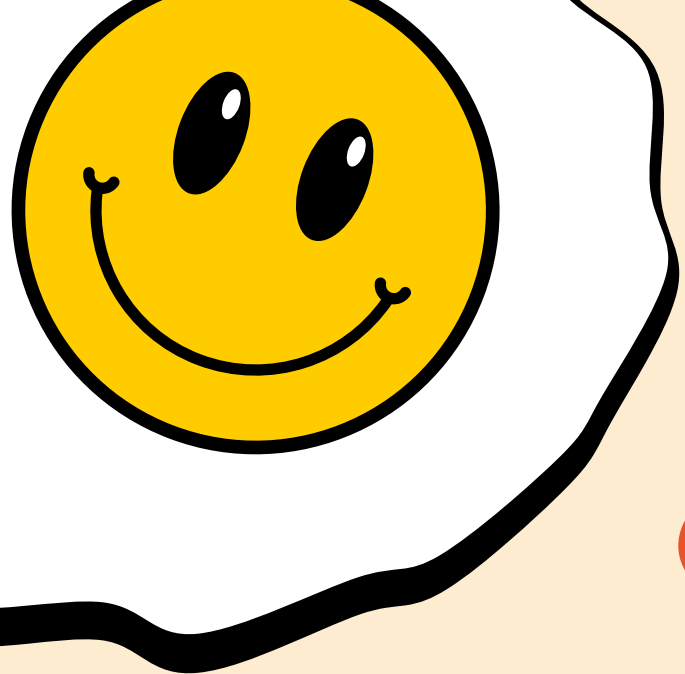
# METHOD AVERAGE

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

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

```
[[ 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.]]
```



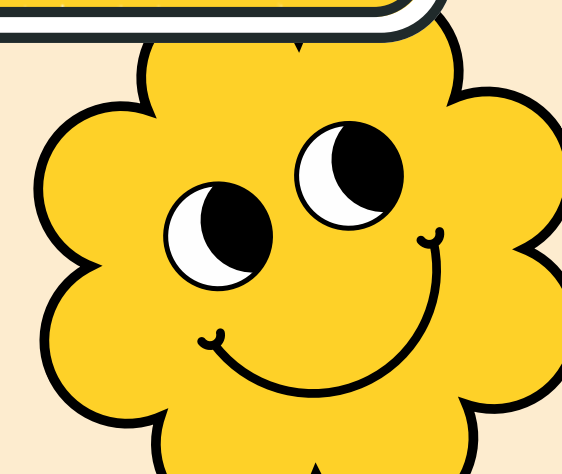


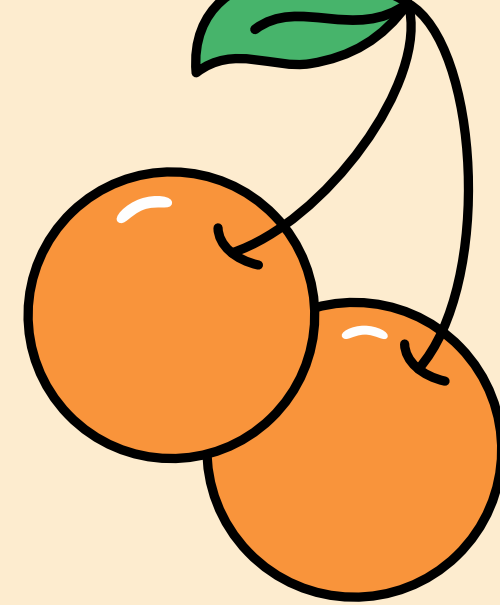
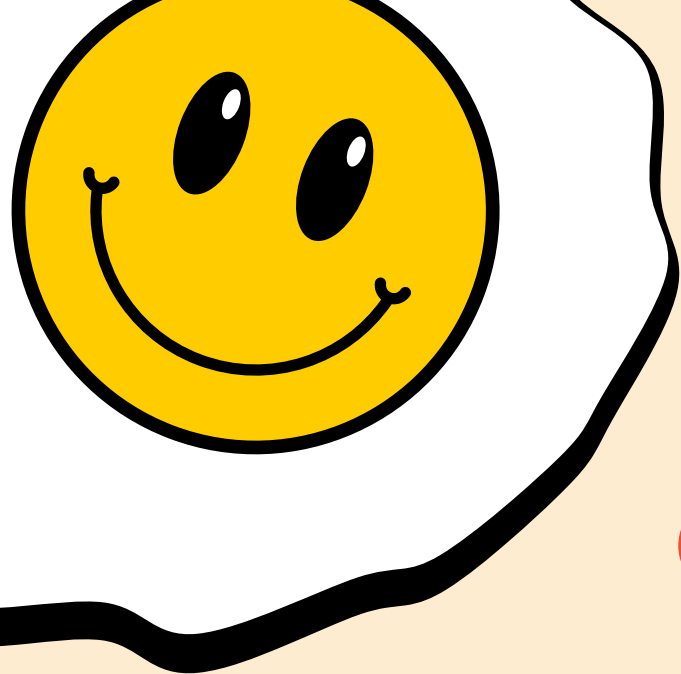
# METHOD AVERAGE

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

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

```
[[ 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.]]
```



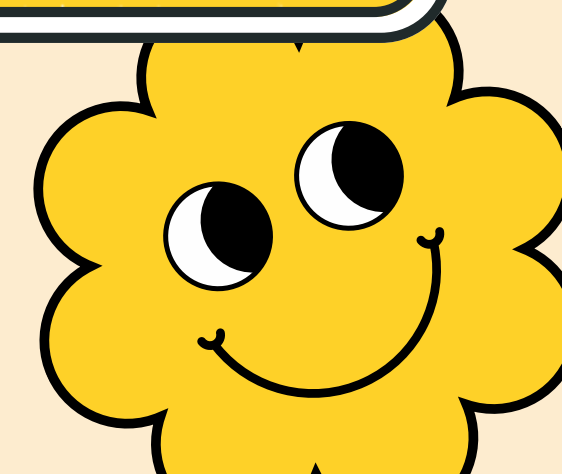


# METHOD AVERAGE

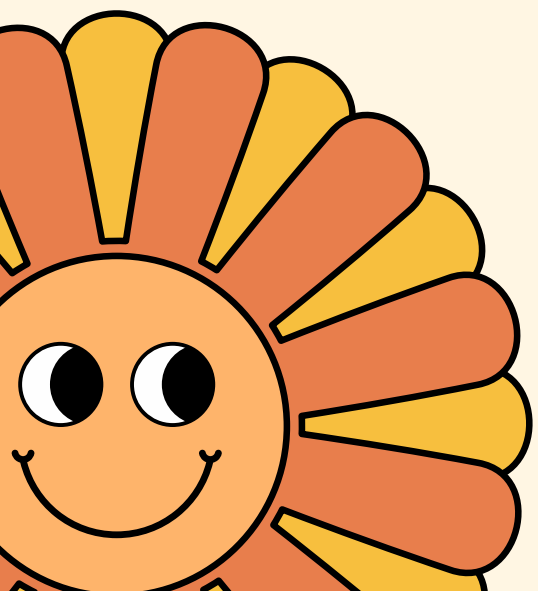
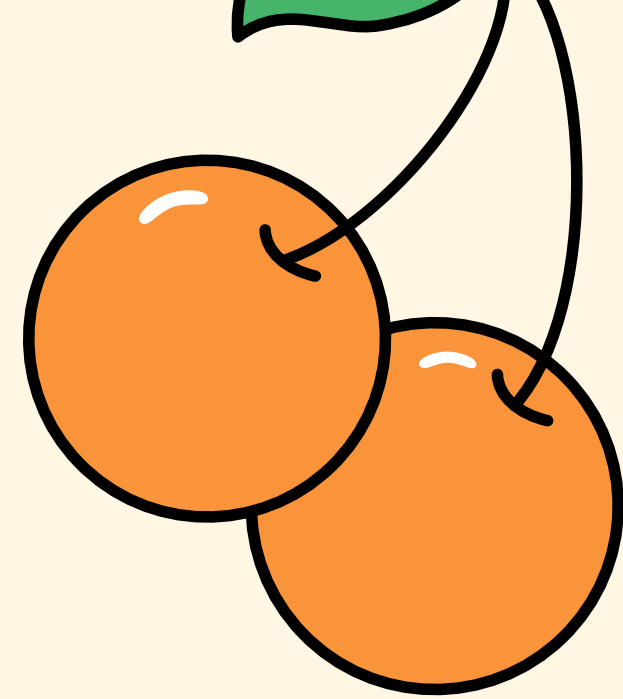
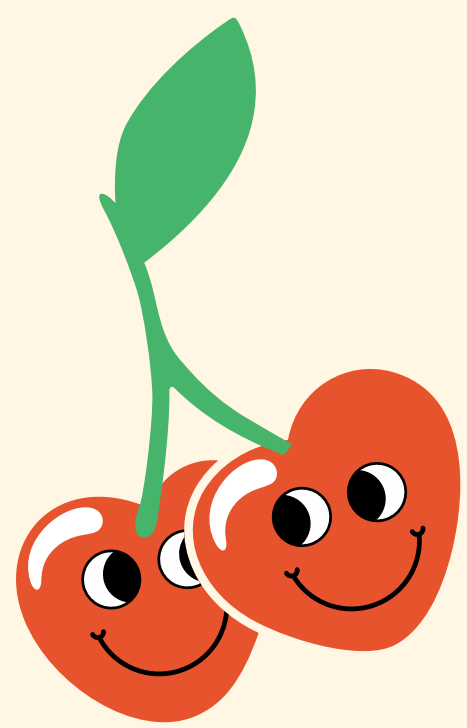
```
gray_img = np.mean(fix_img, axis=-1)
print(np.array(gray_img))

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

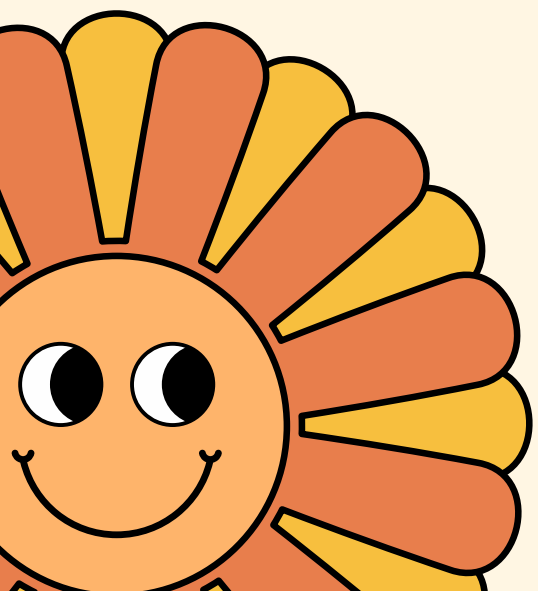
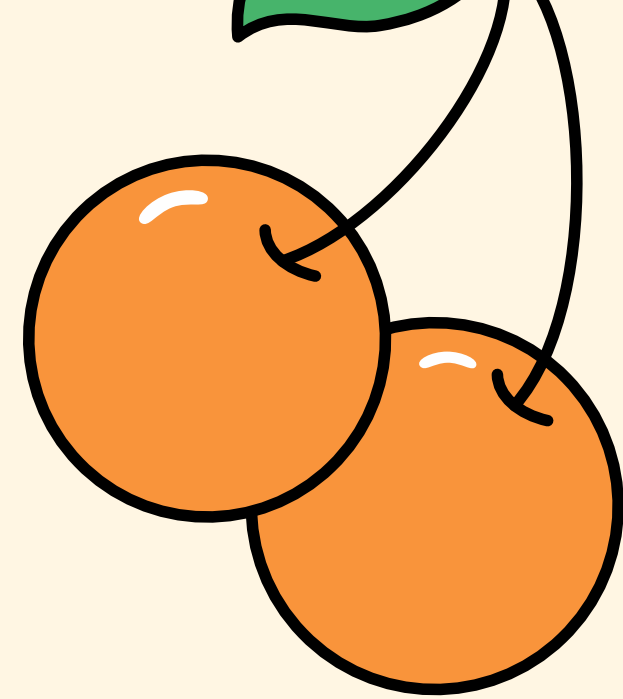
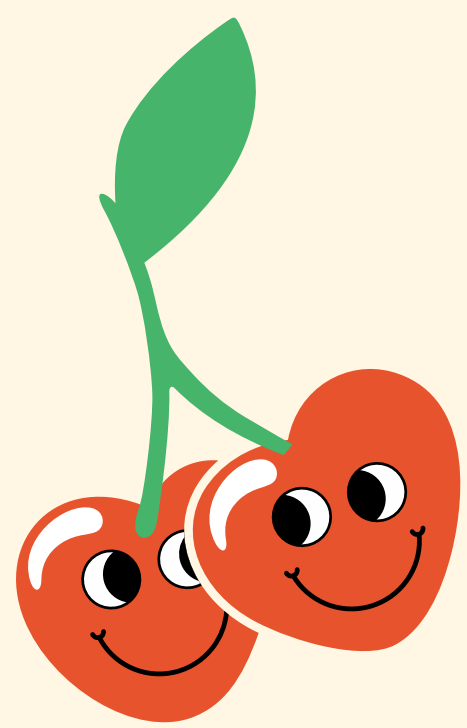
```
[[125. 125. 125. ... 136. 137. 138.]
 [125. 125. 125. ... 136. 138. 138.]
 [125. 125. 125. ... 136. 137. 137.]
 ...
 [ 14.  14.  13. ...  14.  14.  14.]
 [ 14.  14.  13. ...  14.  14.  14.]
 [ 14.  13.  13. ...  14.  14.  14.]]
```



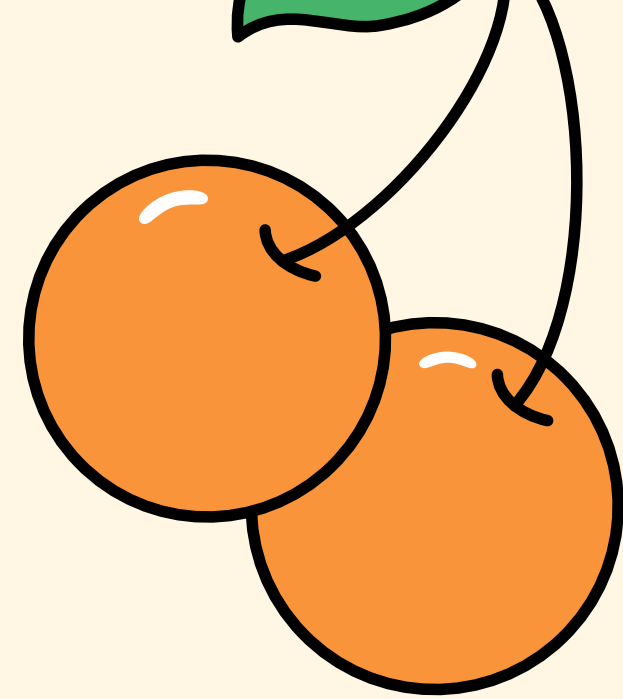
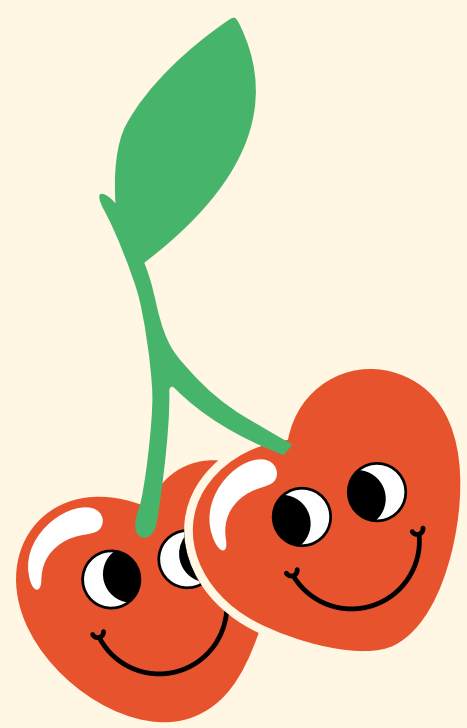




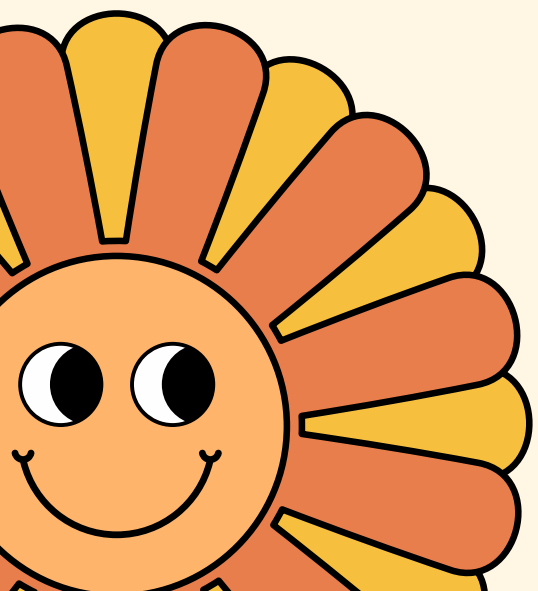
*Hasil fotonya*



*Hasil fotonya*



*Hasil fotonya*



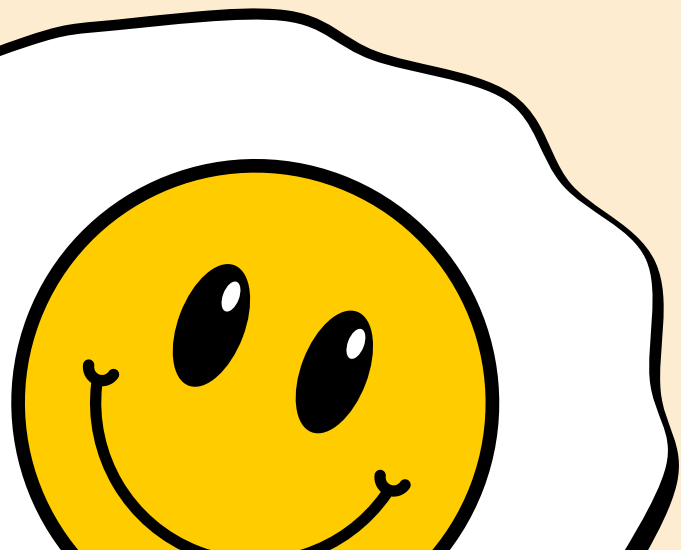
# Metode Luminosity



```
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')
```

```
[[ 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 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')
```

```
[[ 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.]]
```



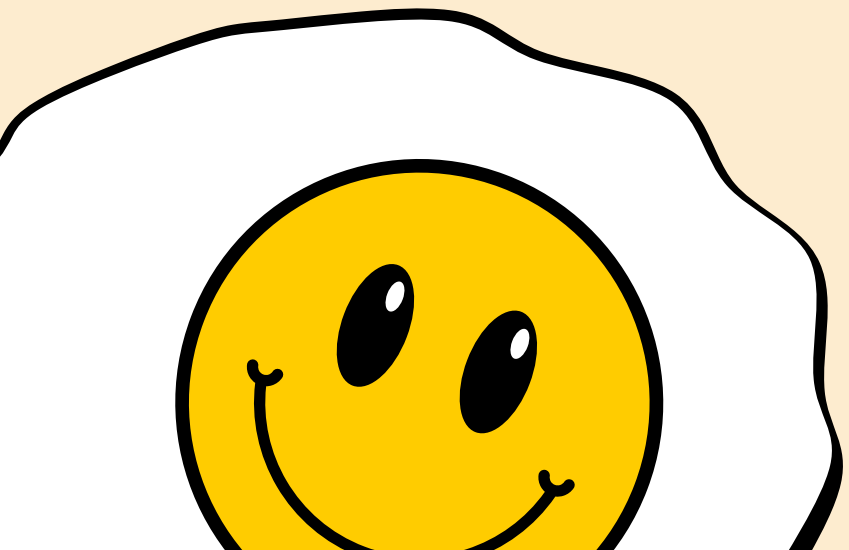
# Metode Luminosity



```
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')
```

```
[[125. 125. 125. ... 136. 137. 138.]
 [125. 125. 125. ... 136. 138. 138.]
 [125. 125. 125. ... 136. 137. 137.]
 ...
 [ 14.  14.  13. ...  14.  14.  14.]
 [ 14.  14.  13. ...  14.  14.  14.]
 [ 14.  13.  13. ...  14.  14.  14.] ]
```



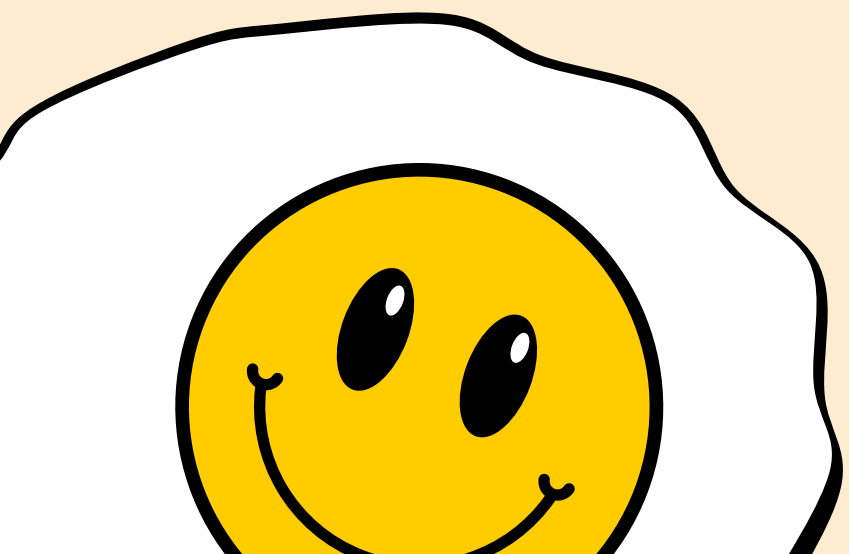


# Metode Weighted Average

```
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')
```

```
[[ 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.]]
```



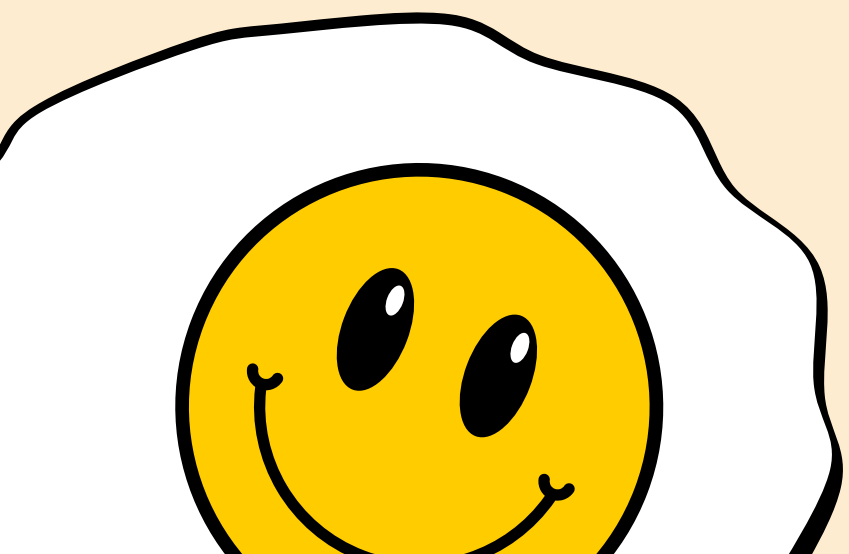


# Metode Weighted Average

```
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')
```

```
[[ 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.]]
```





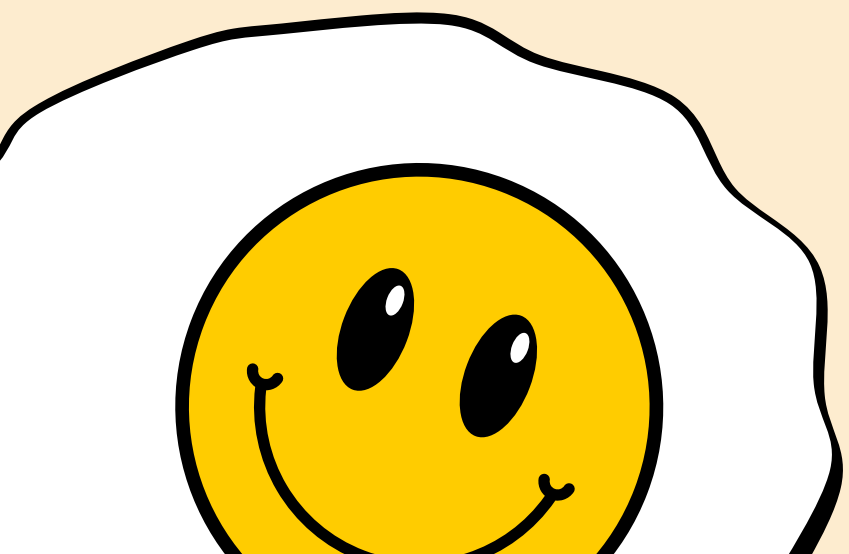


# Metode Weighted Average

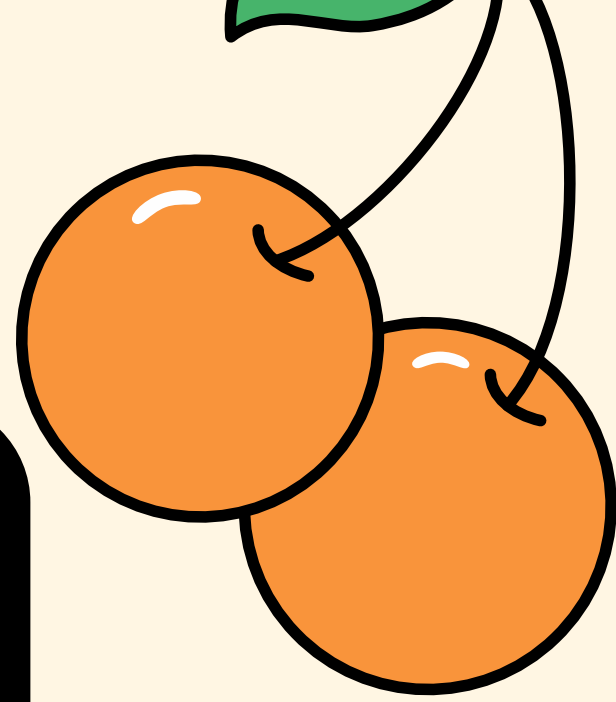
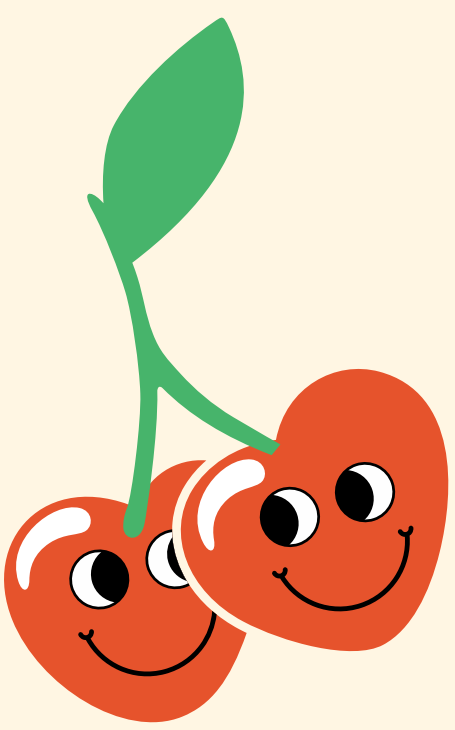
```
wav_img = (0.299*R) + (0.587*G) + (0.114*B)
print(np.array(wav_img))

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

```
[[125. 125. 125. ... 136. 137. 138.]
 [125. 125. 125. ... 136. 138. 138.]
 [125. 125. 125. ... 136. 137. 137.]
 ...
 [ 14.  14.  13. ...  14.  14.  14.]
 [ 14.  14.  13. ...  14.  14.  14.]
 [ 14.  13.  13. ...  14.  14.  14.]]
```



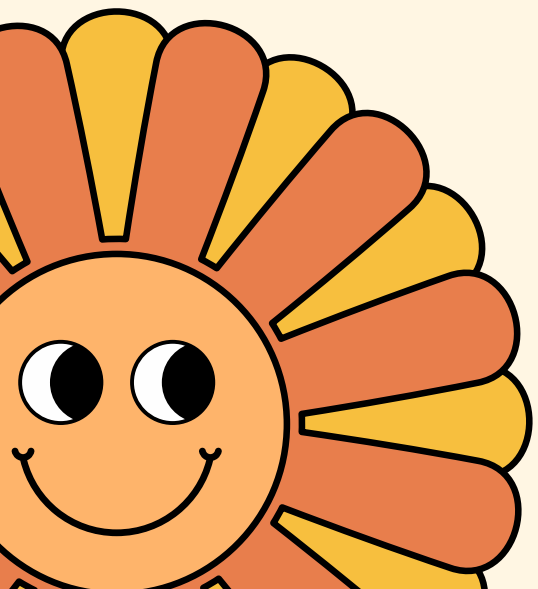
# Hasil Foto



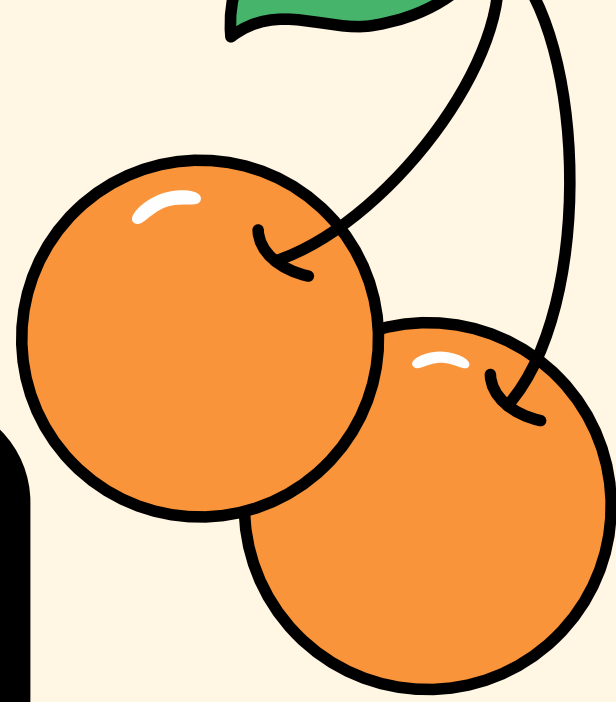
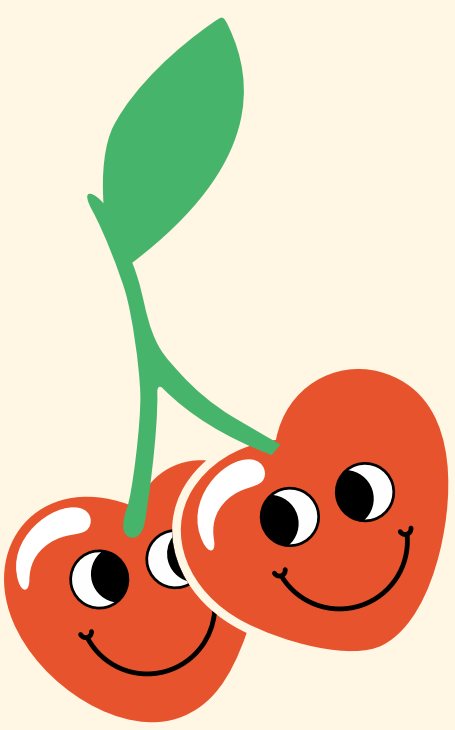
**LUMINOSITY**



**WEIGHTED  
AVERAGE**



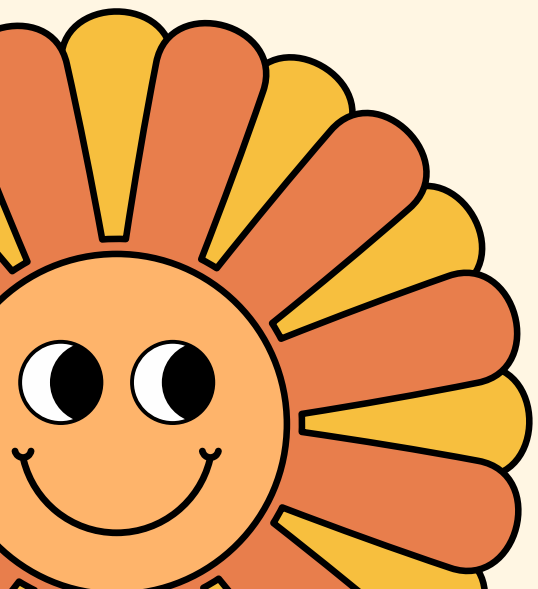
# Hasil Foto



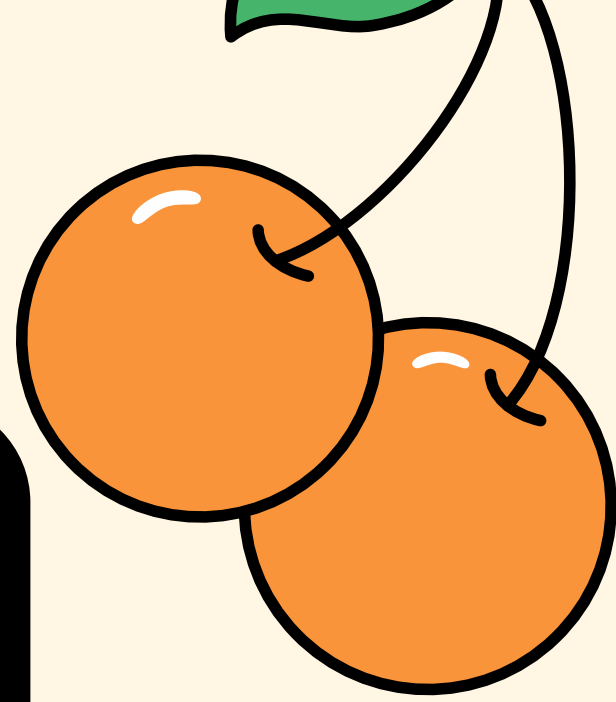
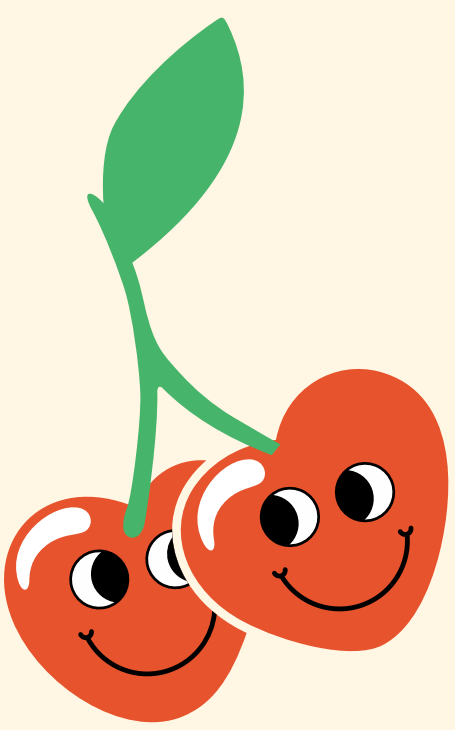
**LUMINOSITY**



**WEIGHTED  
AVERAGE**



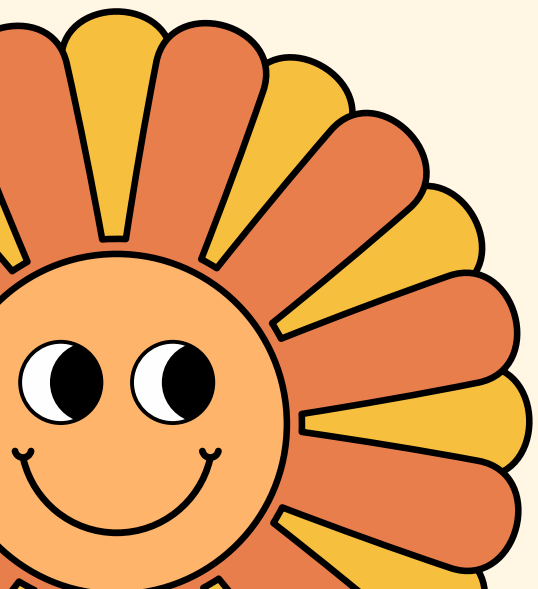
# Hasil Foto



**LUMINOSITY**



**WEIGHTED  
AVERAGE**



გეგმვა  
კავშირ

