ALIEN

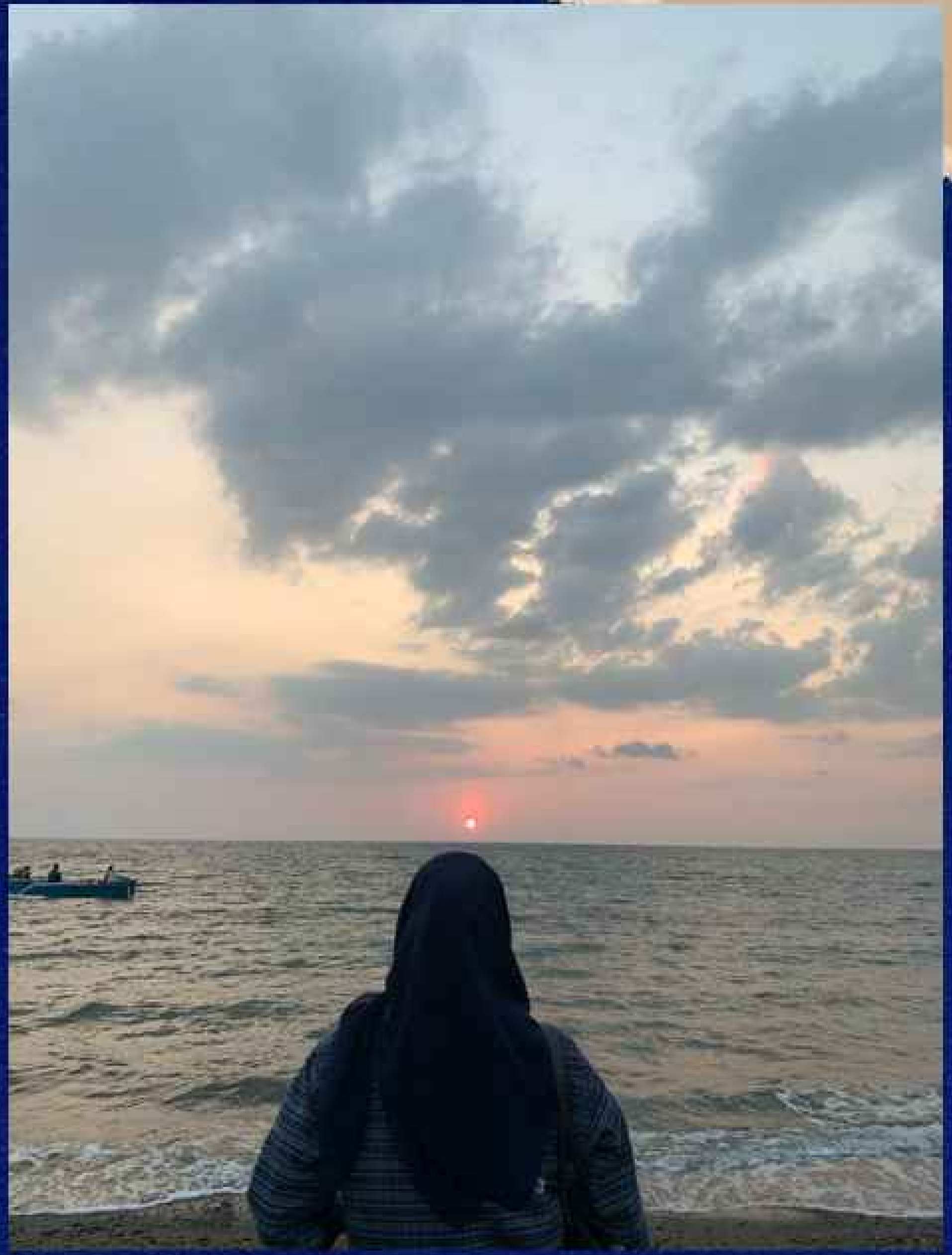
KONVERSI CITRA RGB KE GRAYSCALE

VIEW MY WORK

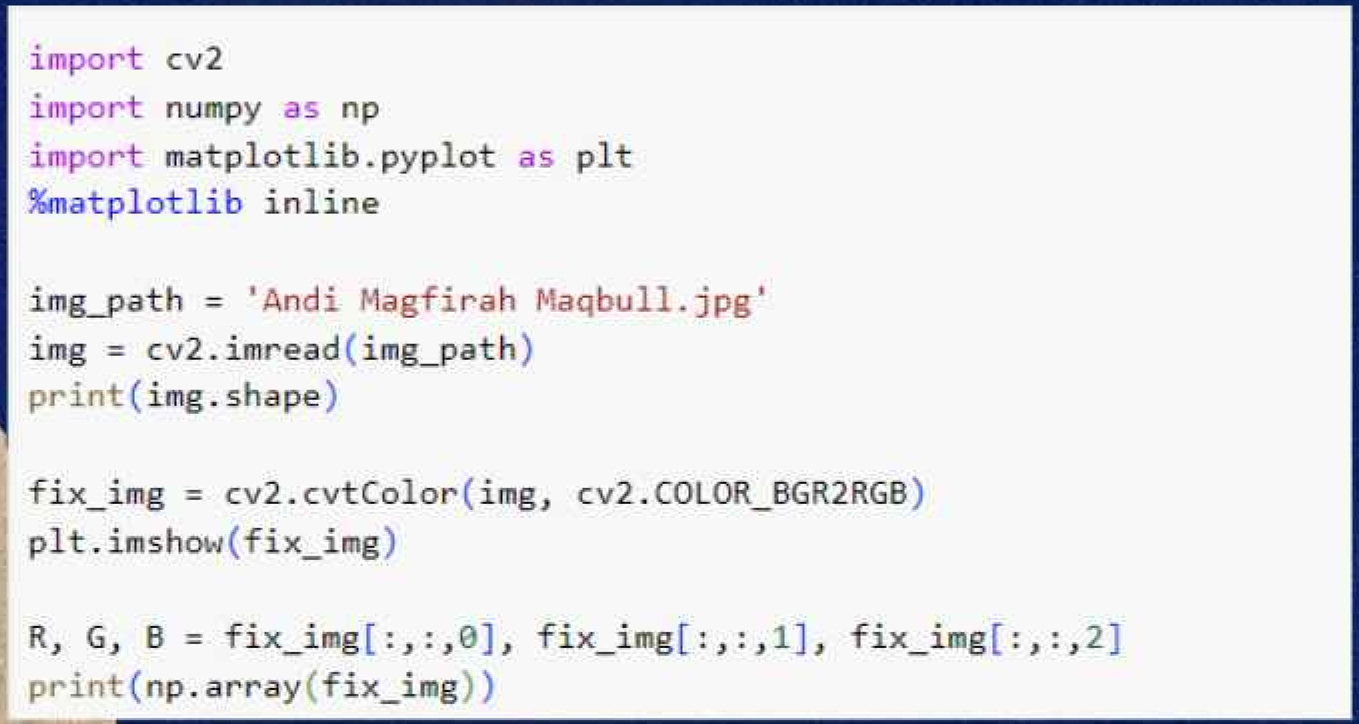
Hello!

Andi Magfirah Maqbul	221011048
Syahraeni Salsabila	221011007
Widya Puspita Sari	221011017

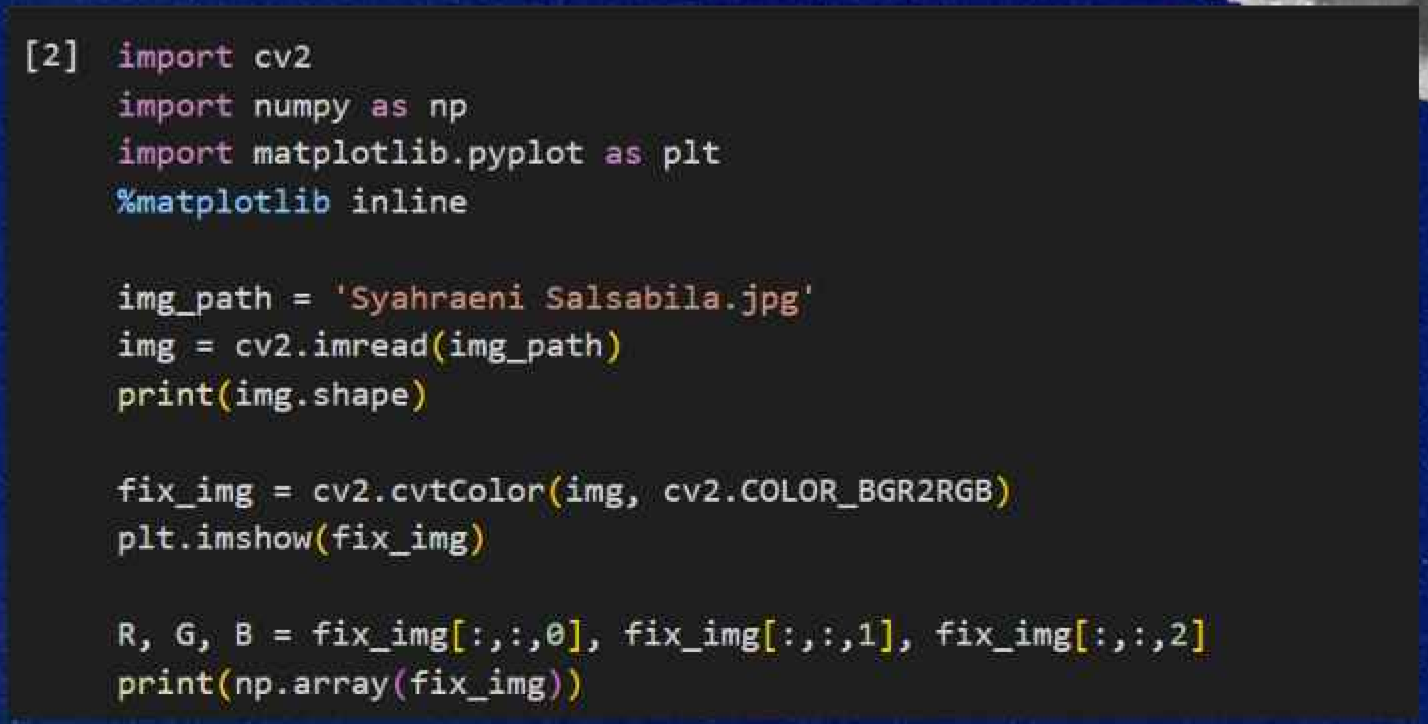
FOTO YANG KAMI GUNAKAN



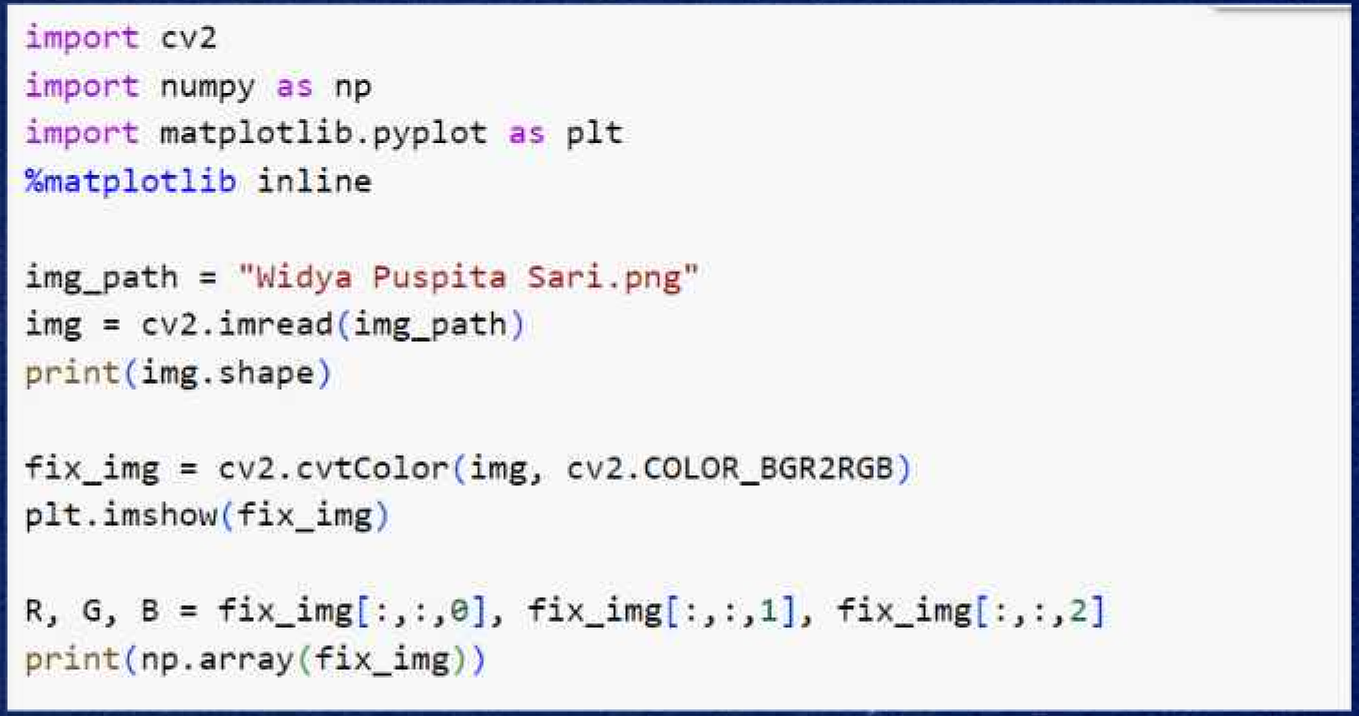
KONVERSI CITRA RGB KE GRAYSCALE



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

img_path = 'Andi Magfirah Maqbull.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))
```

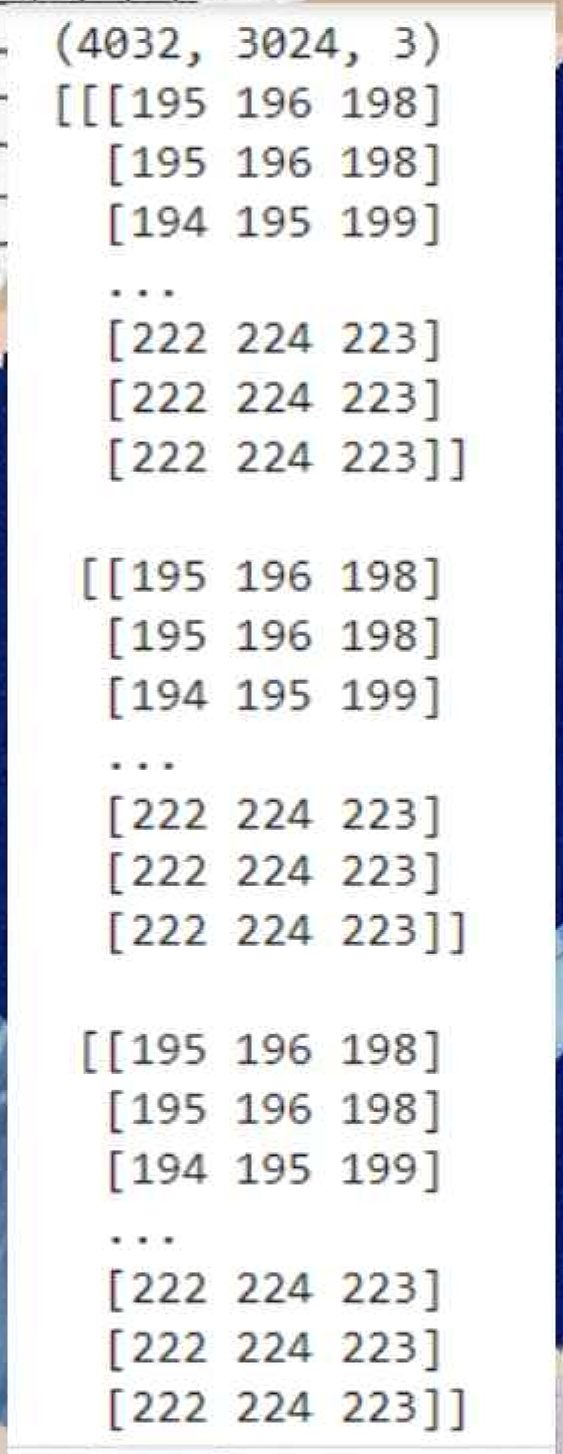


```
[2] import cv2
    import numpy as np
    import matplotlib.pyplot as plt
    %matplotlib inline

    img_path = 'Syahraeni Salsabila.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))
```

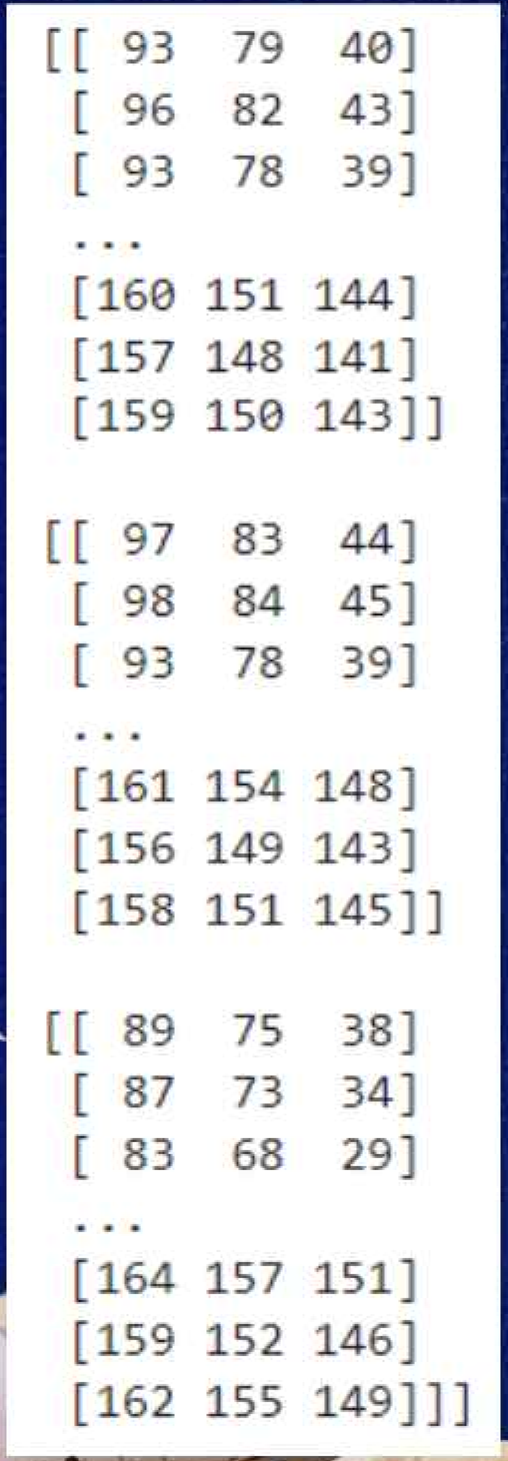


```
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

img_path = "Widya Puspita Sari.png"
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))
```

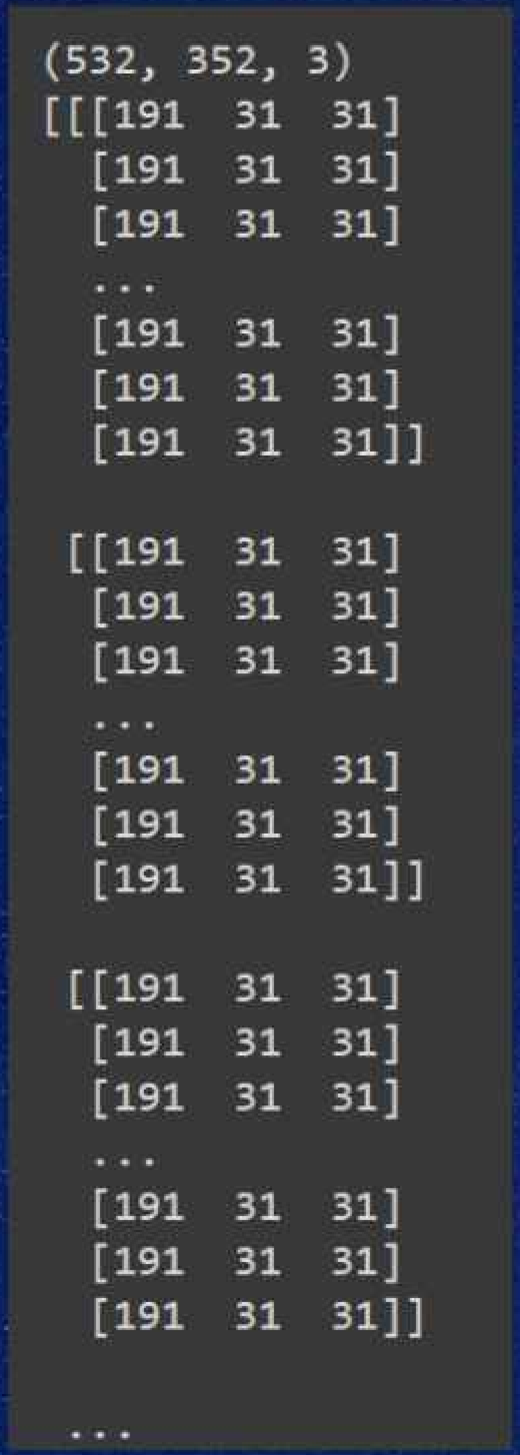
MATRIKS AWAL



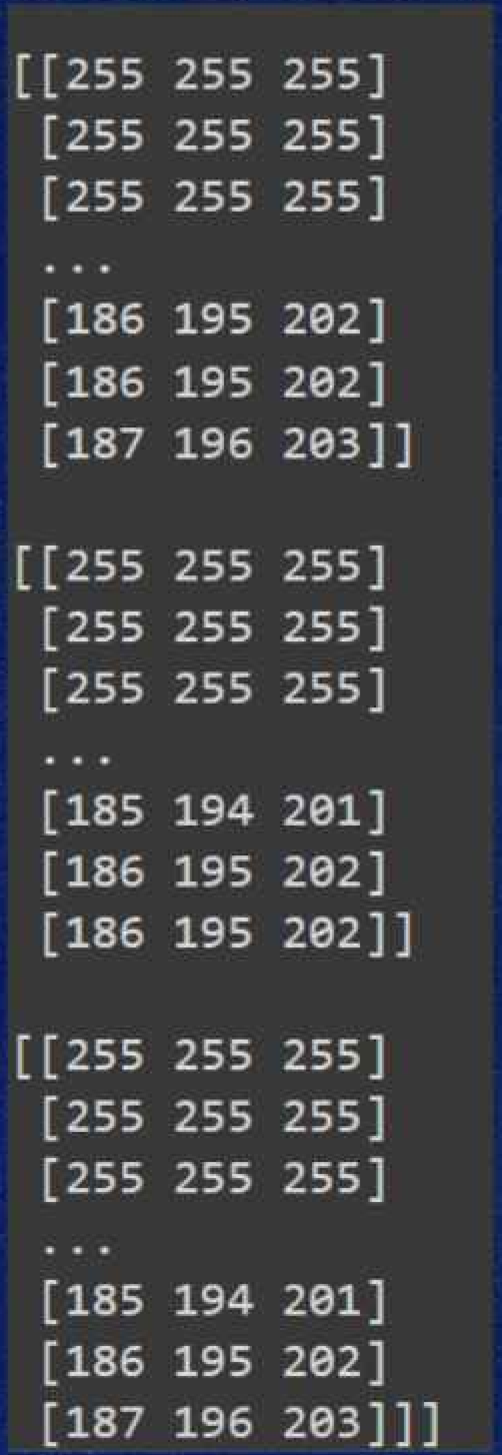
```
(4032, 3024, 3)
[[[195 196 198]
  [195 196 198]
  [194 195 199]
  ...
  [222 224 223]
  [222 224 223]
  [222 224 223]]

 [[195 196 198]
  [195 196 198]
  [194 195 199]
  ...
  [222 224 223]
  [222 224 223]
  [222 224 223]]

 [[195 196 198]
  [195 196 198]
  [194 195 199]
  ...
  [222 224 223]
  [222 224 223]
  [222 224 223]]
```



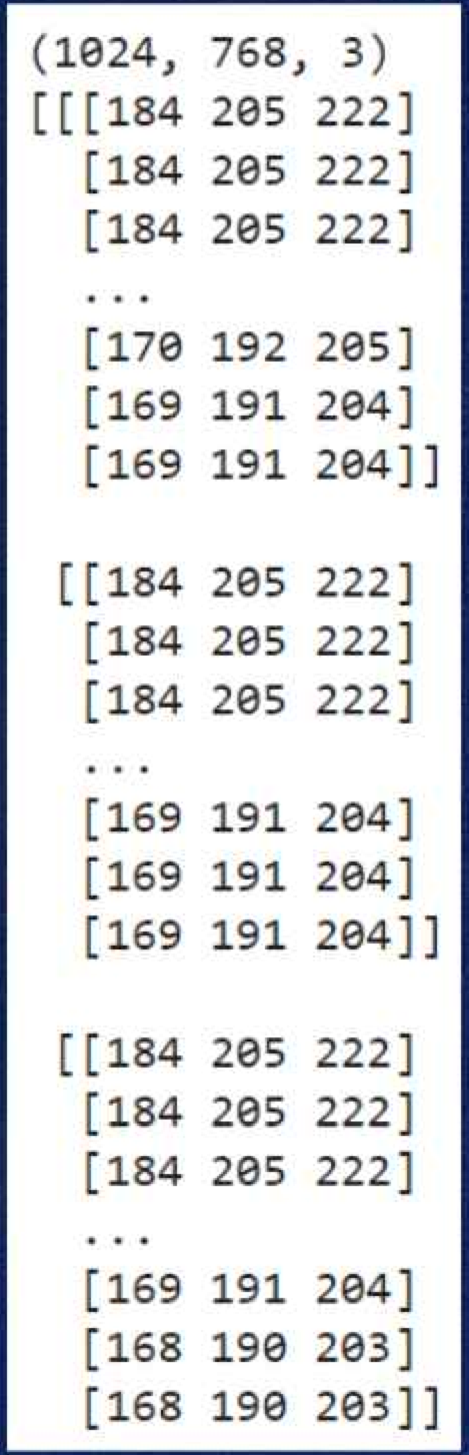
```
[[ 93  79  40]
  [ 96  82  43]
  [ 93  78  39]
  ...
  [160 151 144]
  [157 148 141]
  [159 150 143]]

[[ 97  83  44]
  [ 98  84  45]
  [ 93  78  39]
  ...
  [161 154 148]
  [156 149 143]
  [158 151 145]]

[[ 89  75  38]
  [ 87  73  34]
  [ 83  68  29]
  ...
  [164 157 151]
  [159 152 146]
  [162 155 149]]]
```



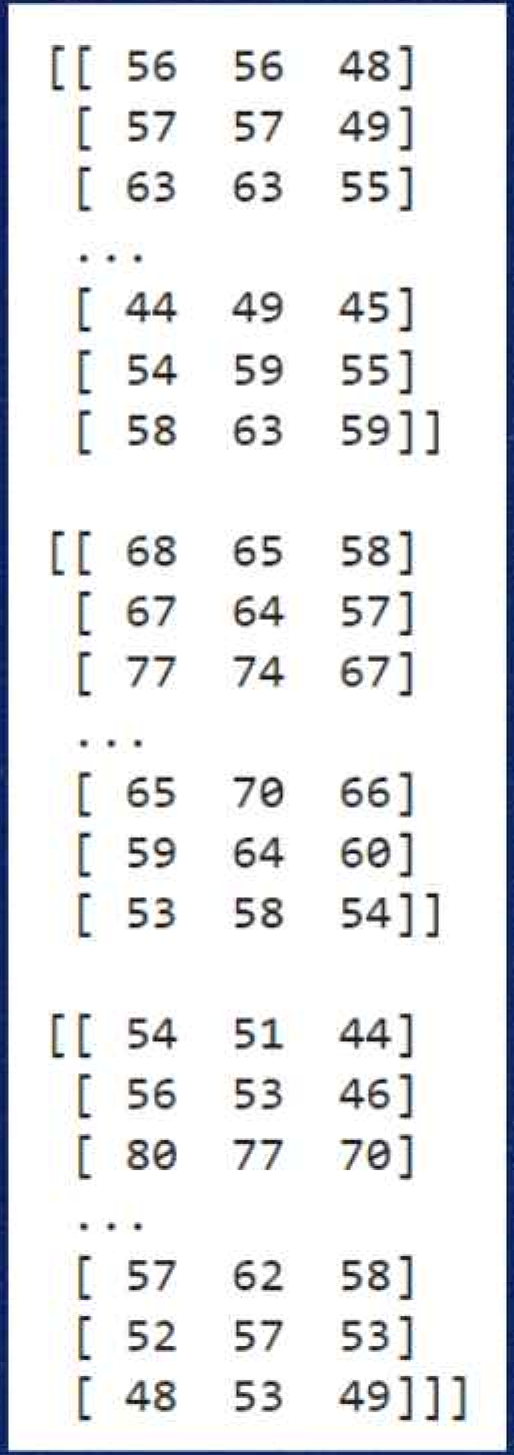
```
(532, 352, 3)
[[[191  31  31]
  [191  31  31]
  [191  31  31]
  ...
  [191  31  31]
  [191  31  31]
  [191  31  31]]

 [[191  31  31]
  [191  31  31]
  [191  31  31]
  ...
  [191  31  31]
  [191  31  31]
  [191  31  31]]

 [[191  31  31]
  [191  31  31]
  [191  31  31]
  ...
  [191  31  31]
  [191  31  31]
  [191  31  31]]

 ...
```



```
[[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [186 195 202]
  [186 195 202]
  [187 196 203]]

[[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [185 194 201]
  [186 195 202]
  [186 195 202]]

[[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [185 194 201]
  [186 195 202]
  [187 196 203]]]
```

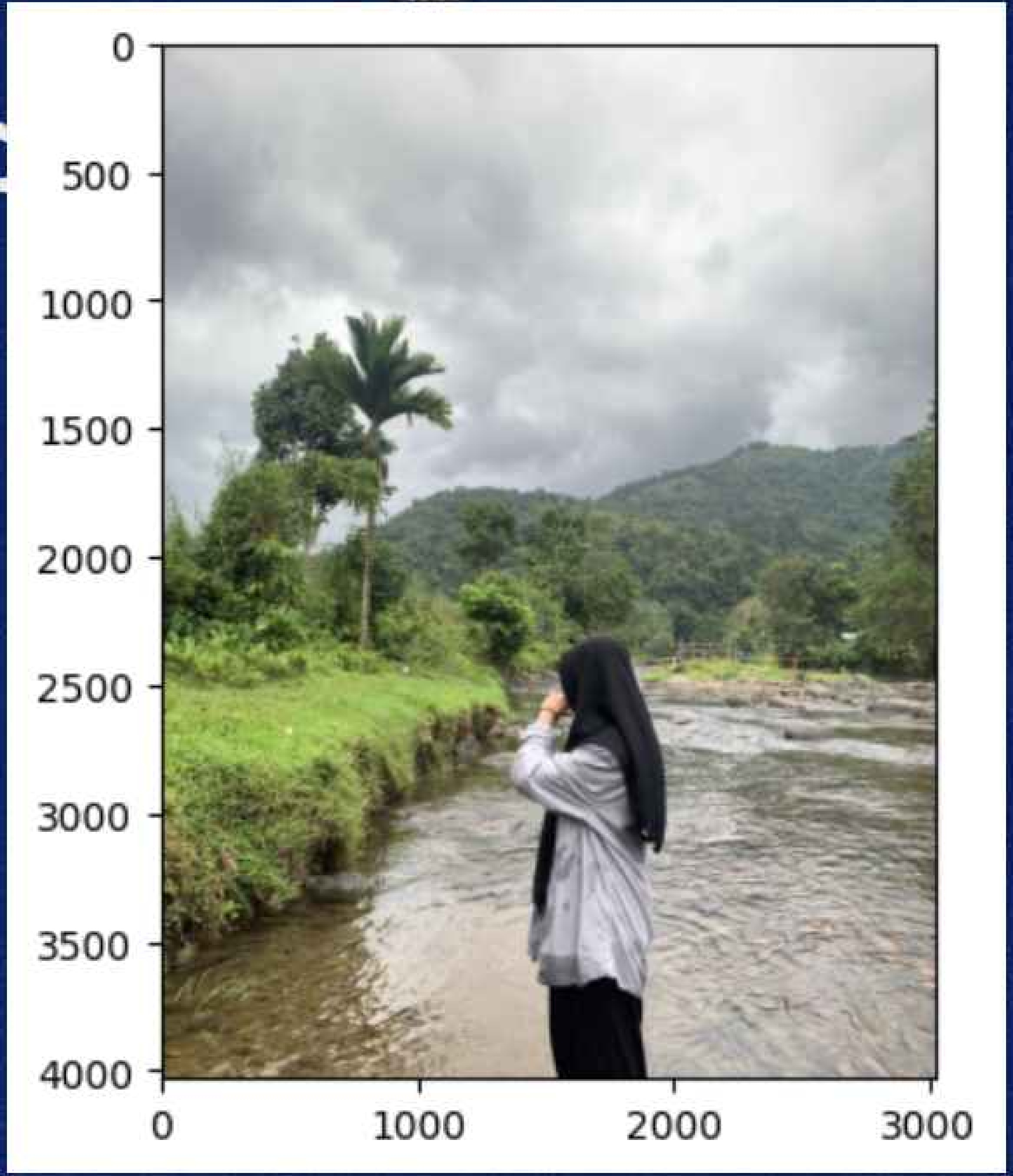


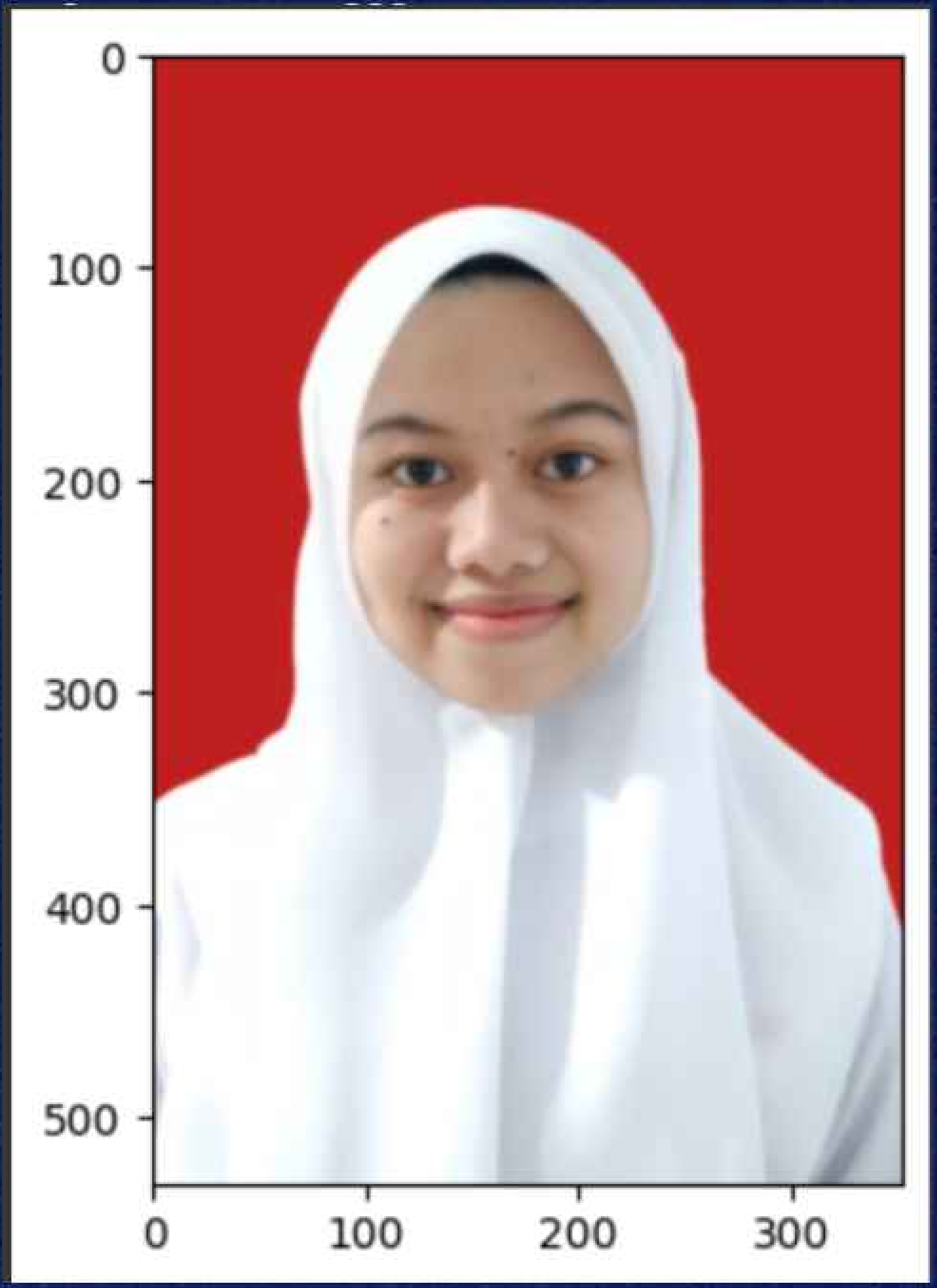
```
(1024, 768, 3)
[[[184 205 222]
  [184 205 222]
  [184 205 222]
  ...
  [170 192 205]
  [169 191 204]
  [169 191 204]]

 [[184 205 222]
  [184 205 222]
  [184 205 222]
  ...
  [169 191 204]
  [169 191 204]
  [169 191 204]]

 [[184 205 222]
  [184 205 222]
  [184 205 222]
  ...
  [169 191 204]
  [168 190 203]
  [168 190 203]]
```

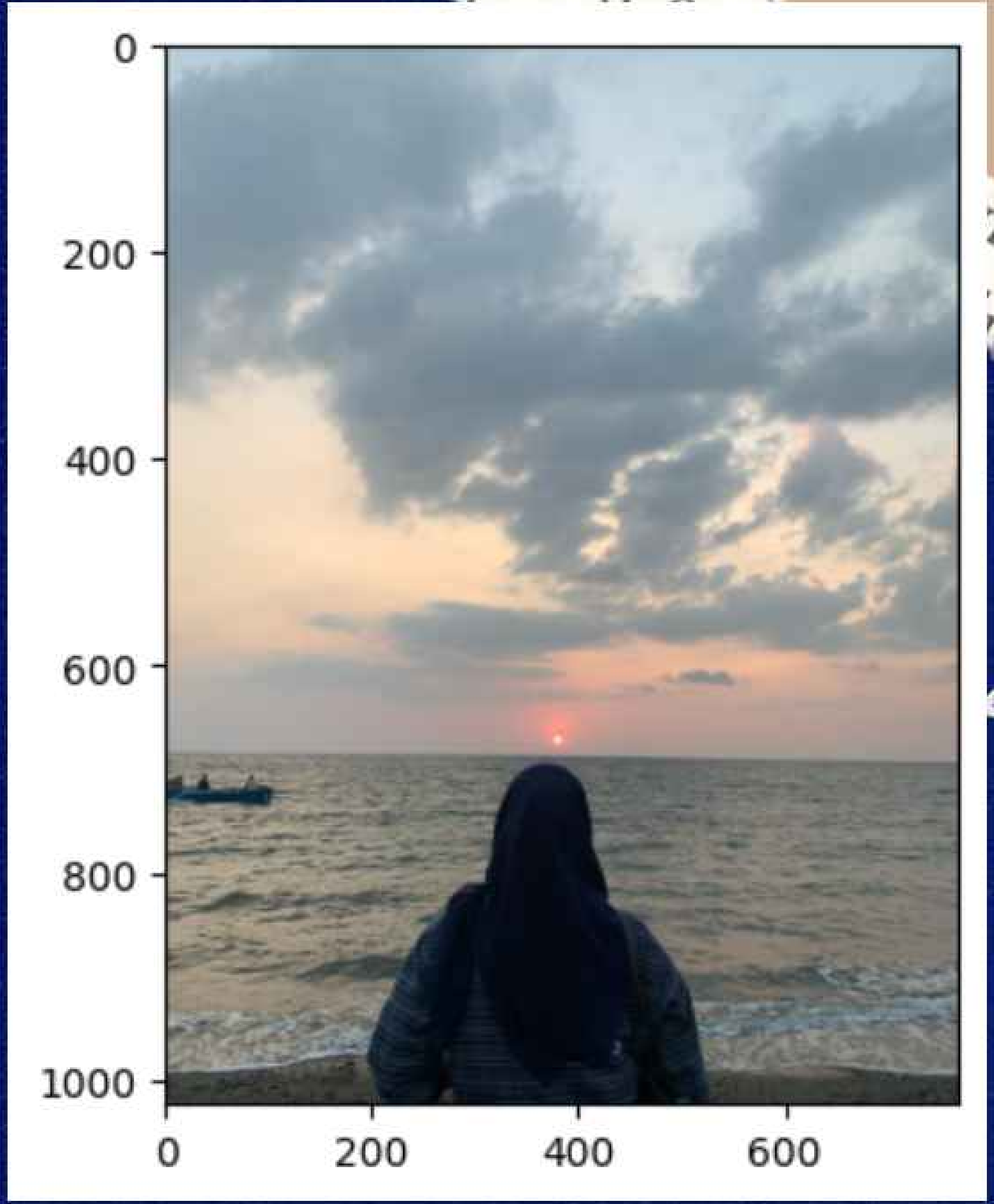


```
[[ 56  56  48]
  [ 57  57  49]
  [ 63  63  55]
  ...
  [ 44  49  45]
  [ 54  59  55]
  [ 58  63  59]]

[[ 68  65  58]
  [ 67  64  57]
  [ 77  74  67]
  ...
  [ 65  70  66]
  [ 59  64  60]
  [ 53  58  54]]

[[ 54  51  44]
  [ 56  53  46]
  [ 80  77  70]
  ...
  [ 57  62  58]
  [ 52  57  53]
  [ 48  53  49]]]
```

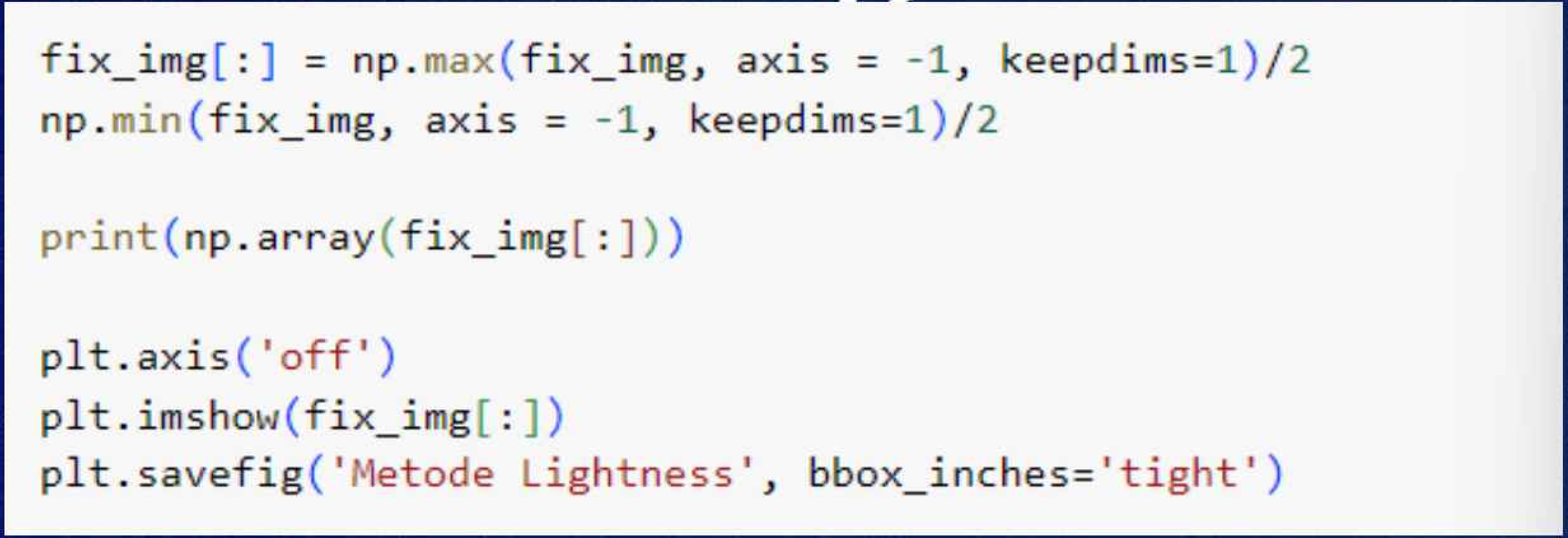
KONVERSI CITRA RGB KE GRAYSCALE







Metode Lightness

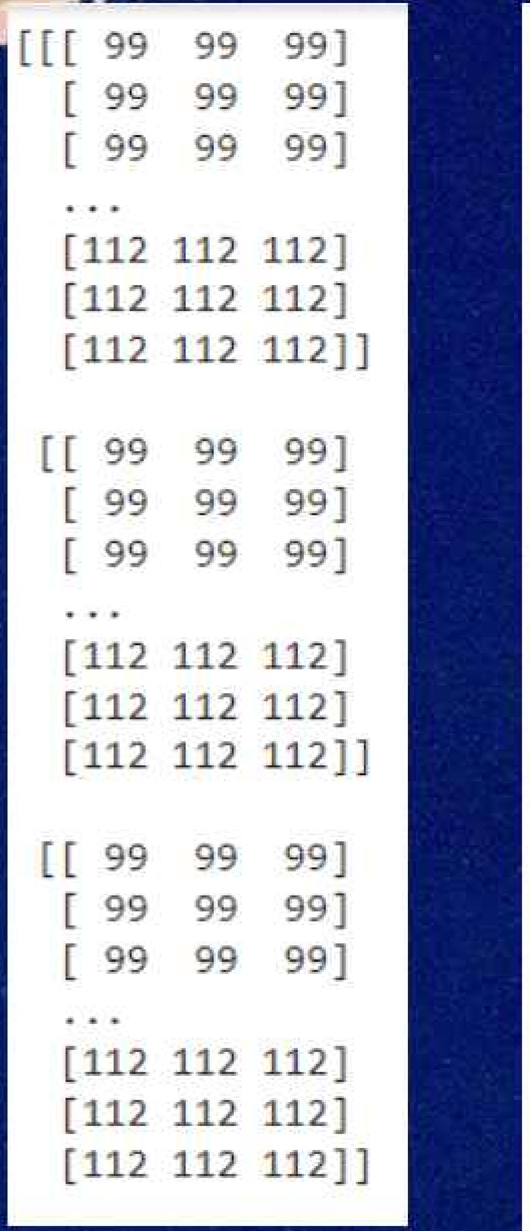


```
fix_img[:] = np.max(fix_img, axis = -1, keepdims=1)/2
np.min(fix_img, axis = -1, keepdims=1)/2

print(np.array(fix_img[:]))


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

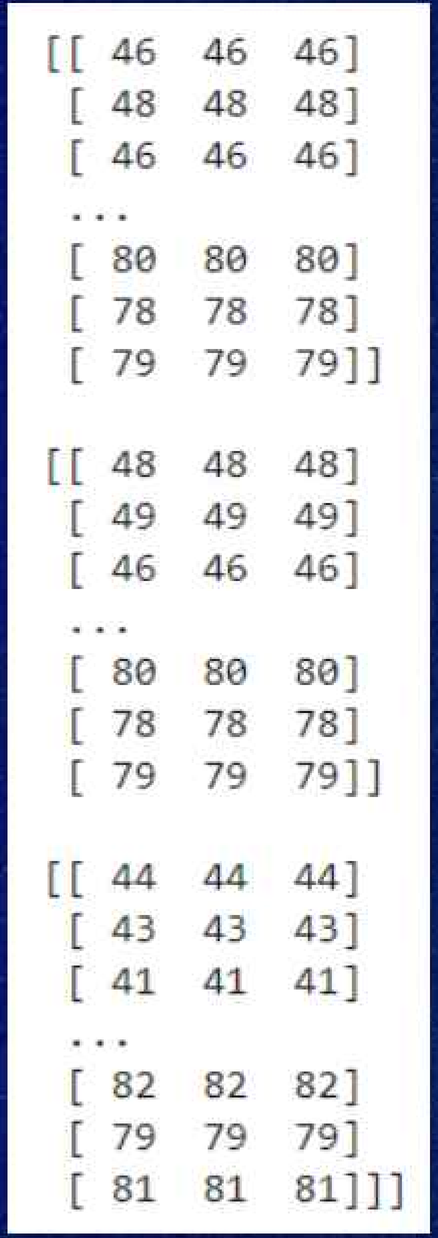
Keuntungan:

Menggunakan nilai rata-rata antara nilai maksimum dan minimum, memberikan gambar dengan tingkat kontras yang baik.

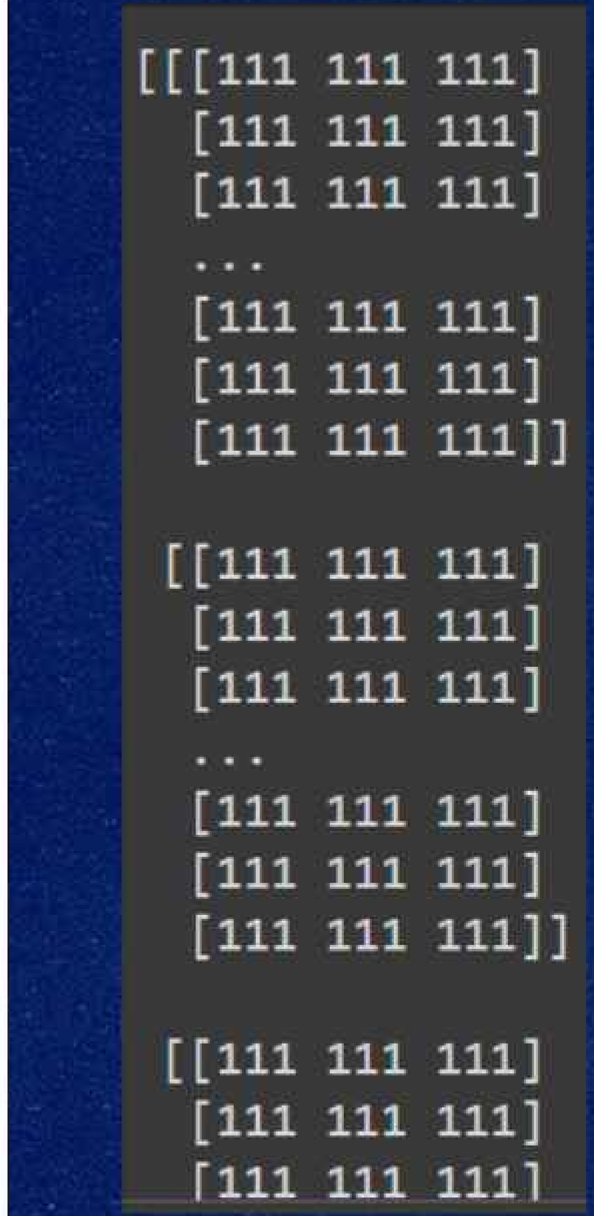
Metode Lightness



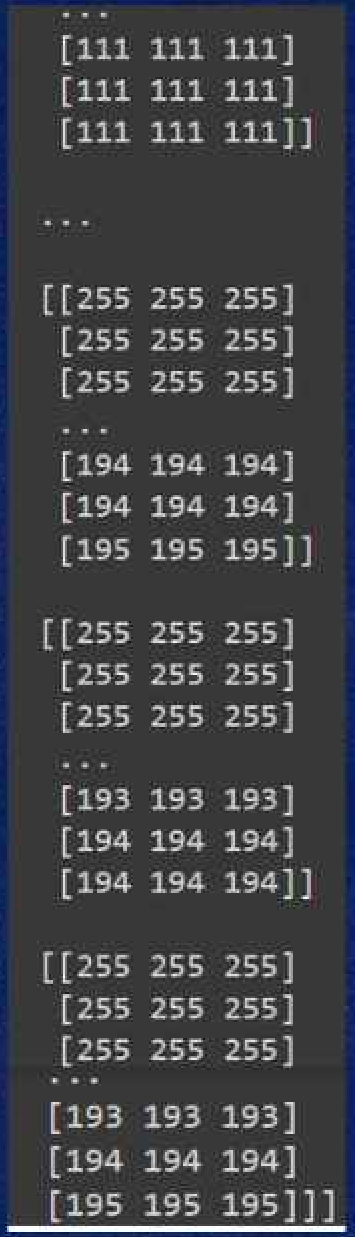
```
[[[ 99  99  99]
  [ 99  99  99]
  [ 99  99  99]
  ...
  [112 112 112]
  [112 112 112]
  [112 112 112]]

 [[ 99  99  99]
  [ 99  99  99]
  [ 99  99  99]
  ...
  [112 112 112]
  [112 112 112]
  [112 112 112]]

 [[ 99  99  99]
  [ 99  99  99]
  [ 99  99  99]
  ...
  [112 112 112]
  [112 112 112]
  [112 112 112]]
```



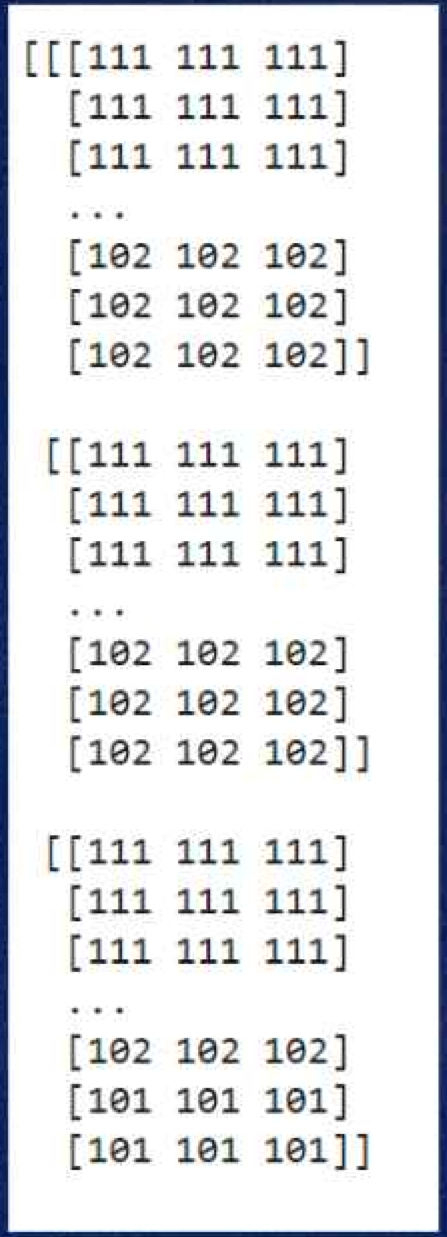
```
[[ 46  46  46]
  [ 48  48  48]
  [ 46  46  46]
  ...
  [ 80  80  80]
  [ 78  78  78]
  [ 79  79  79]]

 [[ 48  48  48]
  [ 49  49  49]
  [ 46  46  46]
  ...
  [ 80  80  80]
  [ 78  78  78]
  [ 79  79  79]]

 [[ 44  44  44]
  [ 43  43  43]
  [ 41  41  41]
  ...
  [ 82  82  82]
  [ 79  79  79]
  [ 81  81  81]]]
```



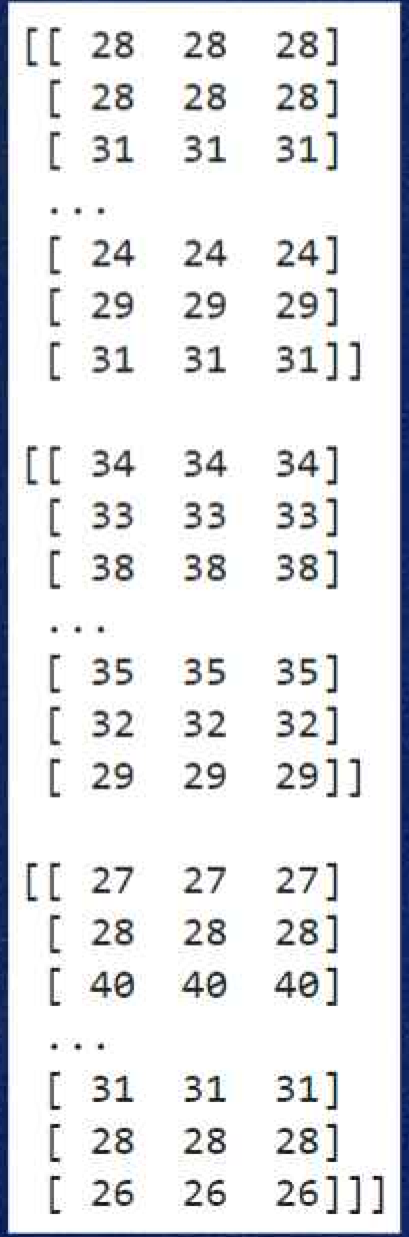
```
[[[111 111 111]
  [111 111 111]
  [111 111 111]
  ...
  [111 111 111]
  [111 111 111]
  [111 111 111]]

 [[111 111 111]
  [111 111 111]
  [111 111 111]
  ...
  [111 111 111]
  [111 111 111]
  [111 111 111]]

 [[111 111 111]
  [111 111 111]
  [111 111 111]
```



```
  ...
  [111 111 111]
  [111 111 111]
  [111 111 111]]

 ...

 [[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [194 194 194]
  [194 194 194]
  [195 195 195]]

 [[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [193 193 193]
  [194 194 194]
  [194 194 194]]

 [[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [193 193 193]
  [194 194 194]
  [195 195 195]]]
```



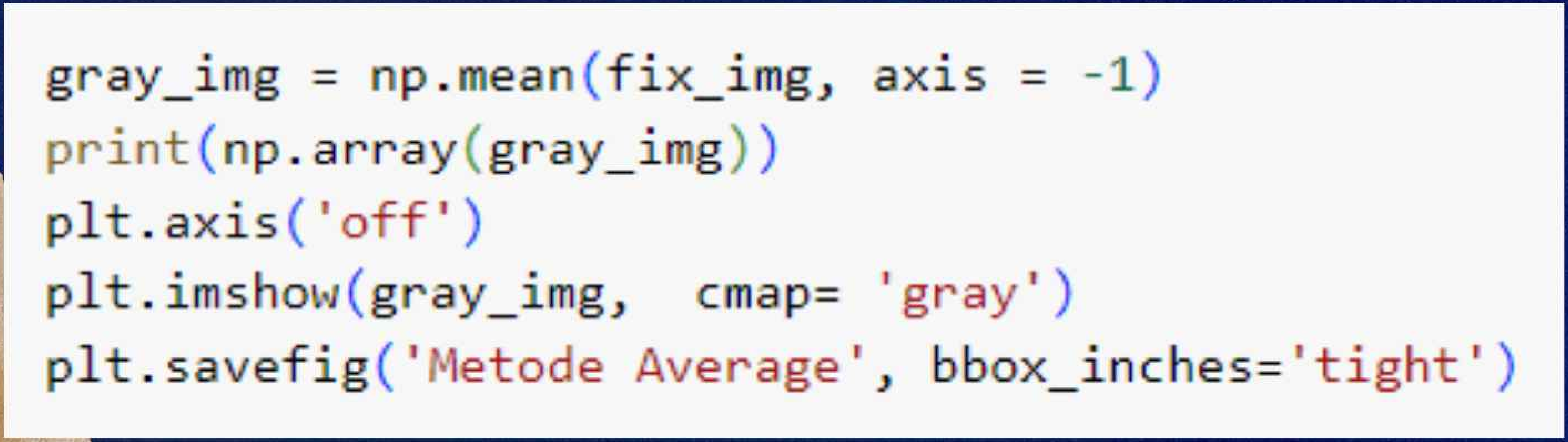
```
[[[111 111 111]
  [111 111 111]
  [111 111 111]
  ...
  [102 102 102]
  [102 102 102]
  [102 102 102]]

 [[111 111 111]
  [111 111 111]
  [111 111 111]
  ...
  [102 102 102]
  [102 102 102]
  [102 102 102]]

 [[111 111 111]
  [111 111 111]
  [111 111 111]
  ...
  [102 102 102]
  [101 101 101]
  [101 101 101]]
```



```
[[ 28  28  28]
  [ 28  28  28]
  [ 31  31  31]
  ...
  [ 24  24  24]
  [ 29  29  29]
  [ 31  31  31]]

 [[ 34  34  34]
  [ 33  33  33]
  [ 38  38  38]
  ...
  [ 35  35  35]
  [ 32  32  32]
  [ 29  29  29]]

 [[ 27  27  27]
  [ 28  28  28]
  [ 40  40  40]
  ...
  [ 31  31  31]
  [ 28  28  28]
  [ 26  26  26]]]
```

Metode Lightness

Metode Avarage

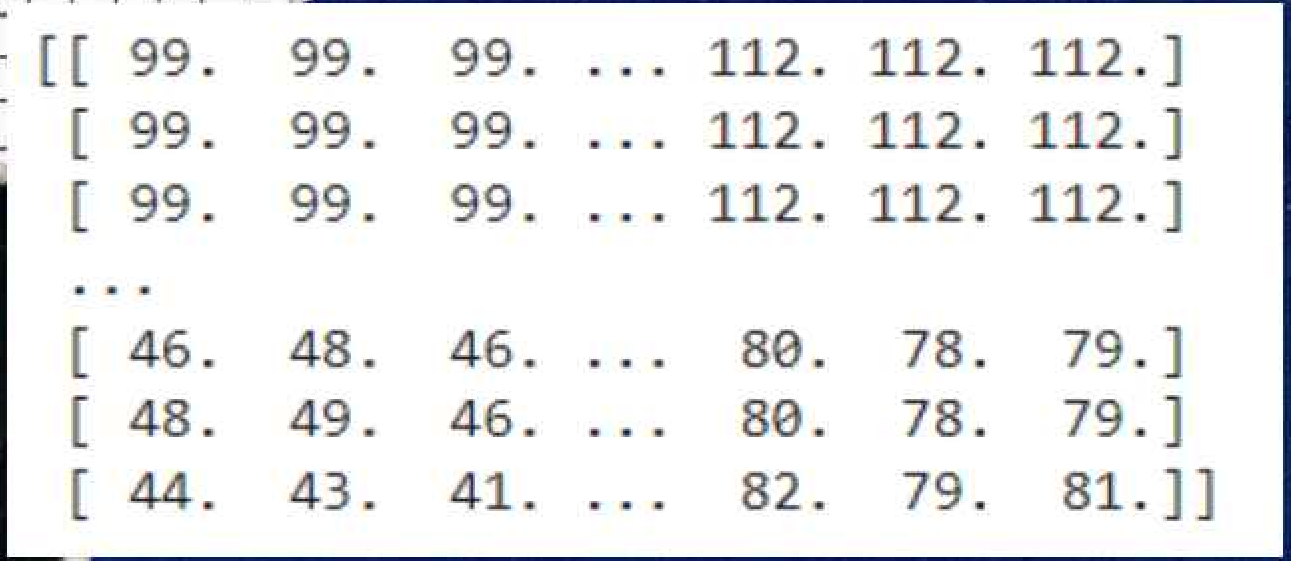


```python
gray_img = np.mean(fix_img, axis = -1)
print(np.array(gray_img))
plt.axis('off')
plt.imshow(gray_img,  cmap= 'gray')
plt.savefig('Metode Average', bbox_inches='tight')
```

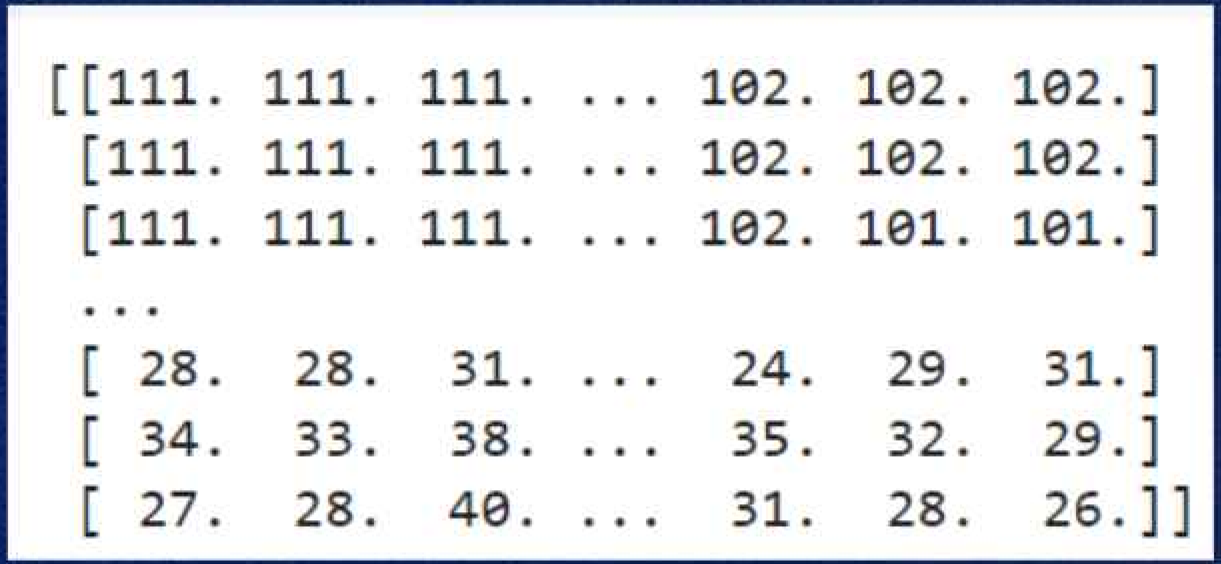
Keuntungan:

Menghasilkan gambar grayscale yang relatif netral.

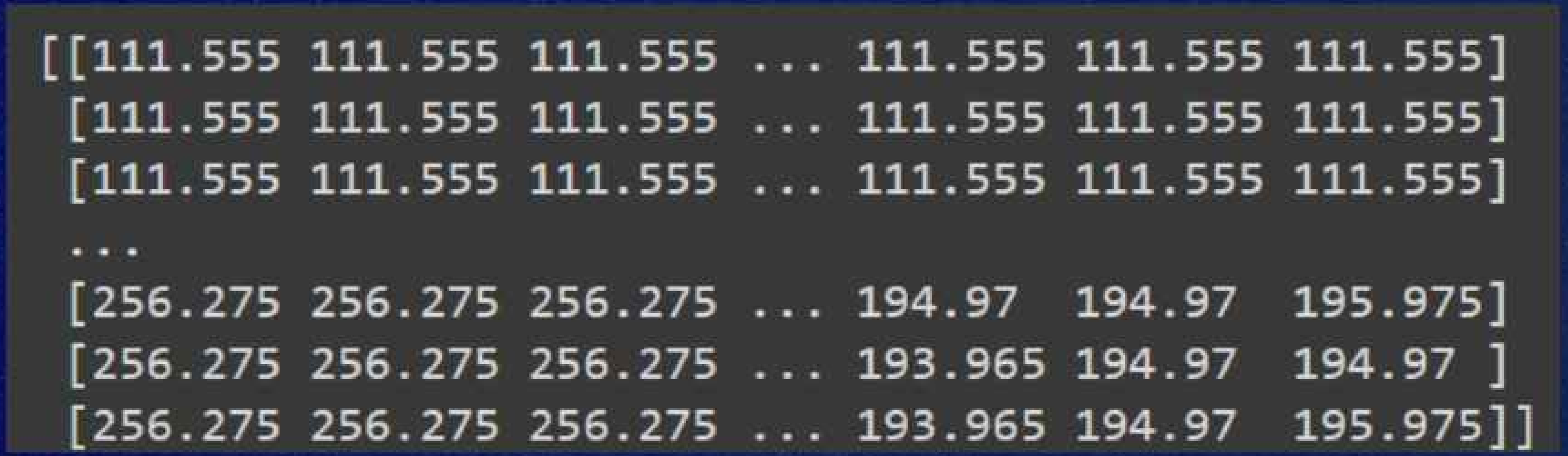
Metode Avarage



```
[[ 99.   99.   99.  ... 112. 112. 112.]
 [ 99.   99.   99.  ... 112. 112. 112.]
 [ 99.   99.   99.  ... 112. 112. 112.]
 ...
 [ 46.   48.   46.  ...  80.   78.   79.]
 [ 48.   49.   46.  ...  80.   78.   79.]
 [ 44.   43.   41.  ...  82.   79.   81.]]
```



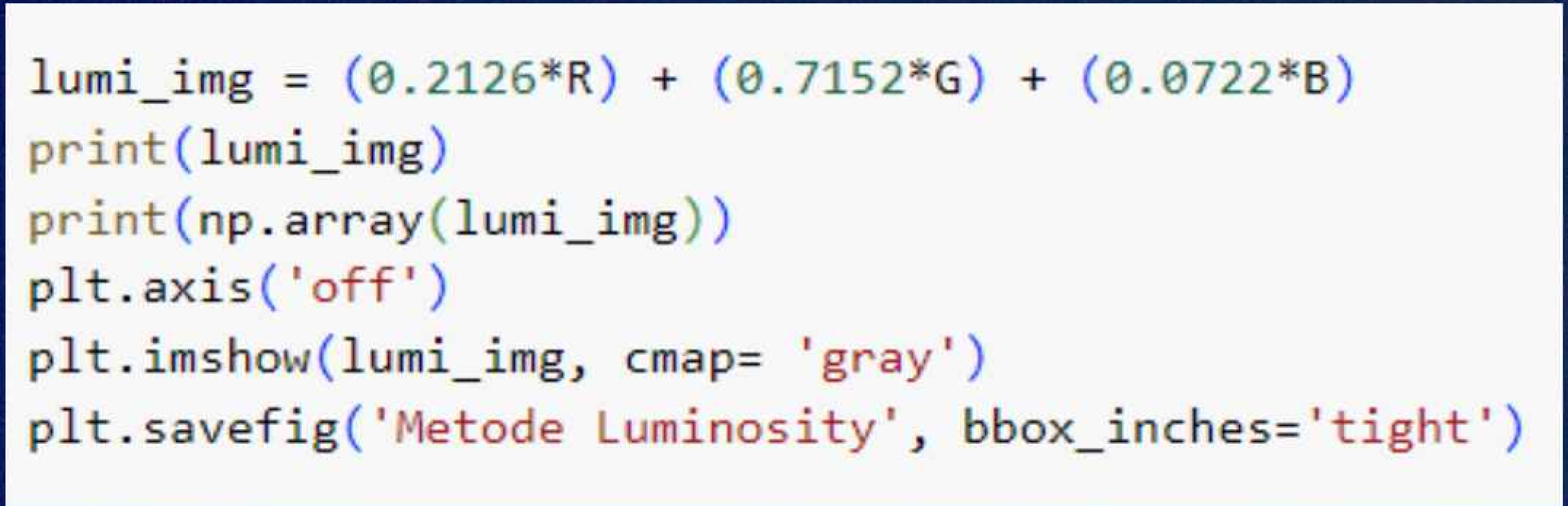
```
[[111. 111. 111. ... 102. 102. 102.]
 [111. 111. 111. ... 102. 102. 102.]
 [111. 111. 111. ... 102. 101. 101.]
 ...
 [ 28.  28.  31. ...  24.  29.  31.]
 [ 34.  33.  38. ...  35.  32.  29.]
 [ 27.  28.  40. ...  31.  28.  26.]]
```



```
[[111.555 111.555 111.555 ... 111.555 111.555 111.555]
 [111.555 111.555 111.555 ... 111.555 111.555 111.555]
 [111.555 111.555 111.555 ... 111.555 111.555 111.555]
 ...
 [256.275 256.275 256.275 ... 194.97  194.97  195.975]
 [256.275 256.275 256.275 ... 193.965 194.97  194.97 ]
 [256.275 256.275 256.275 ... 193.965 194.97  195.975]]
```

Metode Avarage

Metode Luminosity

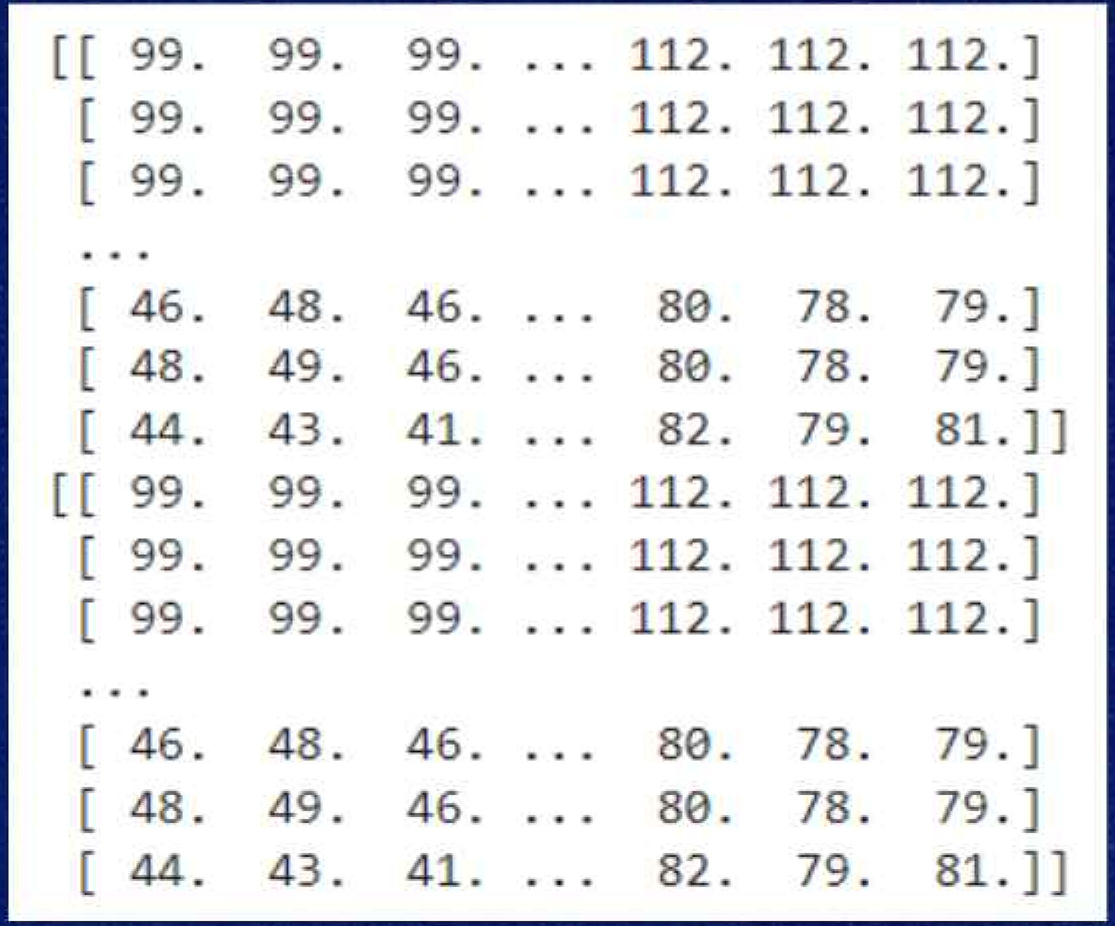


```
lumi_img = (0.2126*R) + (0.7152*G) + (0.0722*B)
print(lumi_img)
print(np.array(lumi_img))
plt.axis('off')
plt.imshow(lumi_img, cmap= 'gray')
plt.savefig('Metode Luminosity', bbox_inches='tight')
```

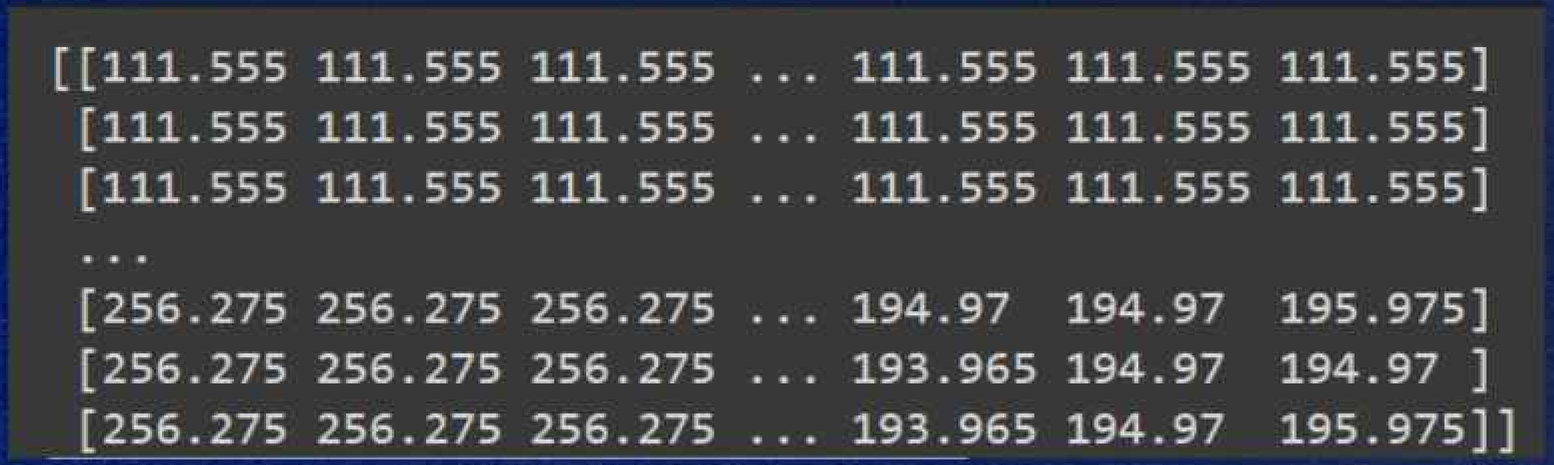
Keuntungan:

Memberikan penekanan pada komponen warna yang lebih terang, lebih sesuai dengan persepsi mata manusia terhadap warna.

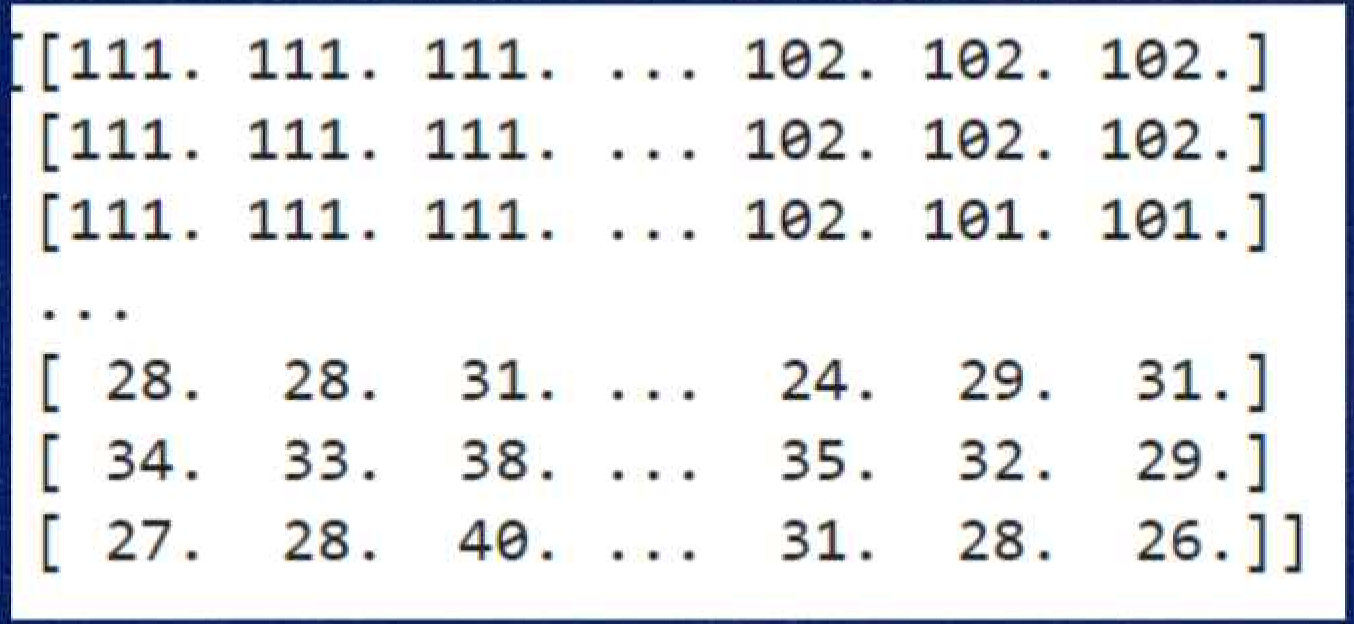
Metode Luminosity



```
[[ 99.   99.   99.  ... 112. 112. 112.]
 [ 99.   99.   99.  ... 112. 112. 112.]
 [ 99.   99.   99.  ... 112. 112. 112.]
 ...
 [ 46.   48.   46.  ...  80.  78.  79.]
 [ 48.   49.   46.  ...  80.  78.  79.]
 [ 44.   43.   41.  ...  82.  79.  81.]]
[[ 99.   99.   99.  ... 112. 112. 112.]
 [ 99.   99.   99.  ... 112. 112. 112.]
 [ 99.   99.   99.  ... 112. 112. 112.]
 ...
 [ 46.   48.   46.  ...  80.  78.  79.]
 [ 48.   49.   46.  ...  80.  78.  79.]
 [ 44.   43.   41.  ...  82.  79.  81.]]
```



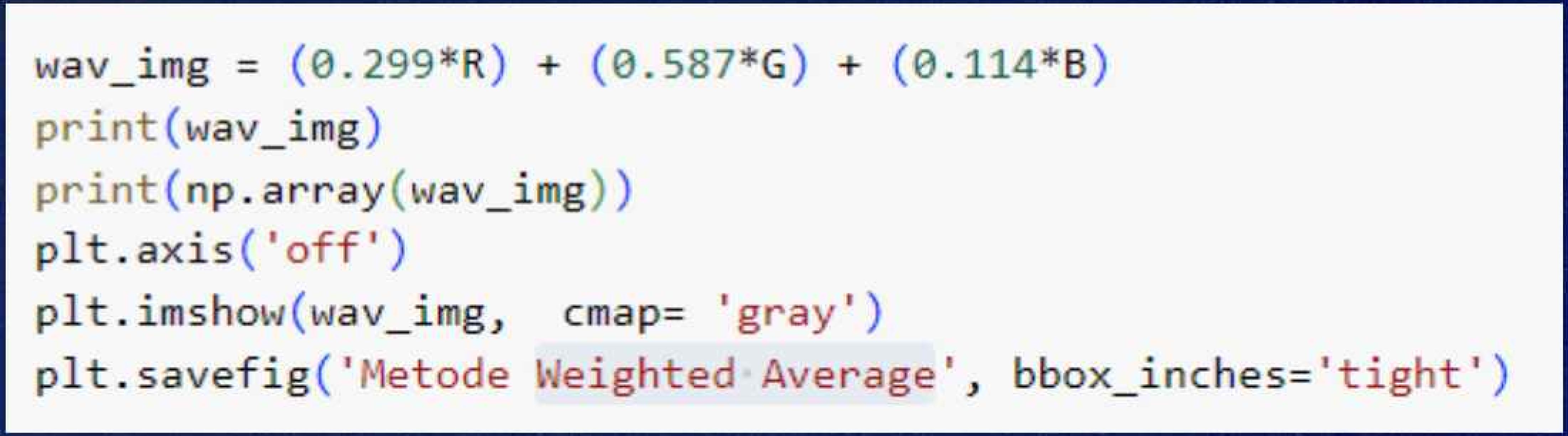
```
[[111.555 111.555 111.555 ... 111.555 111.555 111.555]
 [111.555 111.555 111.555 ... 111.555 111.555 111.555]
 [111.555 111.555 111.555 ... 111.555 111.555 111.555]
 ...
 [256.275 256.275 256.275 ... 194.97  194.97  195.975]
 [256.275 256.275 256.275 ... 193.965 194.97  194.97 ]
 [256.275 256.275 256.275 ... 193.965 194.97  195.975]]
```



```
[[111. 111. 111. ... 102. 102. 102.]
 [111. 111. 111. ... 102. 102. 102.]
 [111. 111. 111. ... 102. 101. 101.]
 ...
 [ 28.  28.  31. ...  24.  29.  31.]
 [ 34.  33.  38. ...  35.  32.  29.]
 [ 27.  28.  40. ...  31.  28.  26.]]
```

Metode Luminosity

Metode Weighted Average

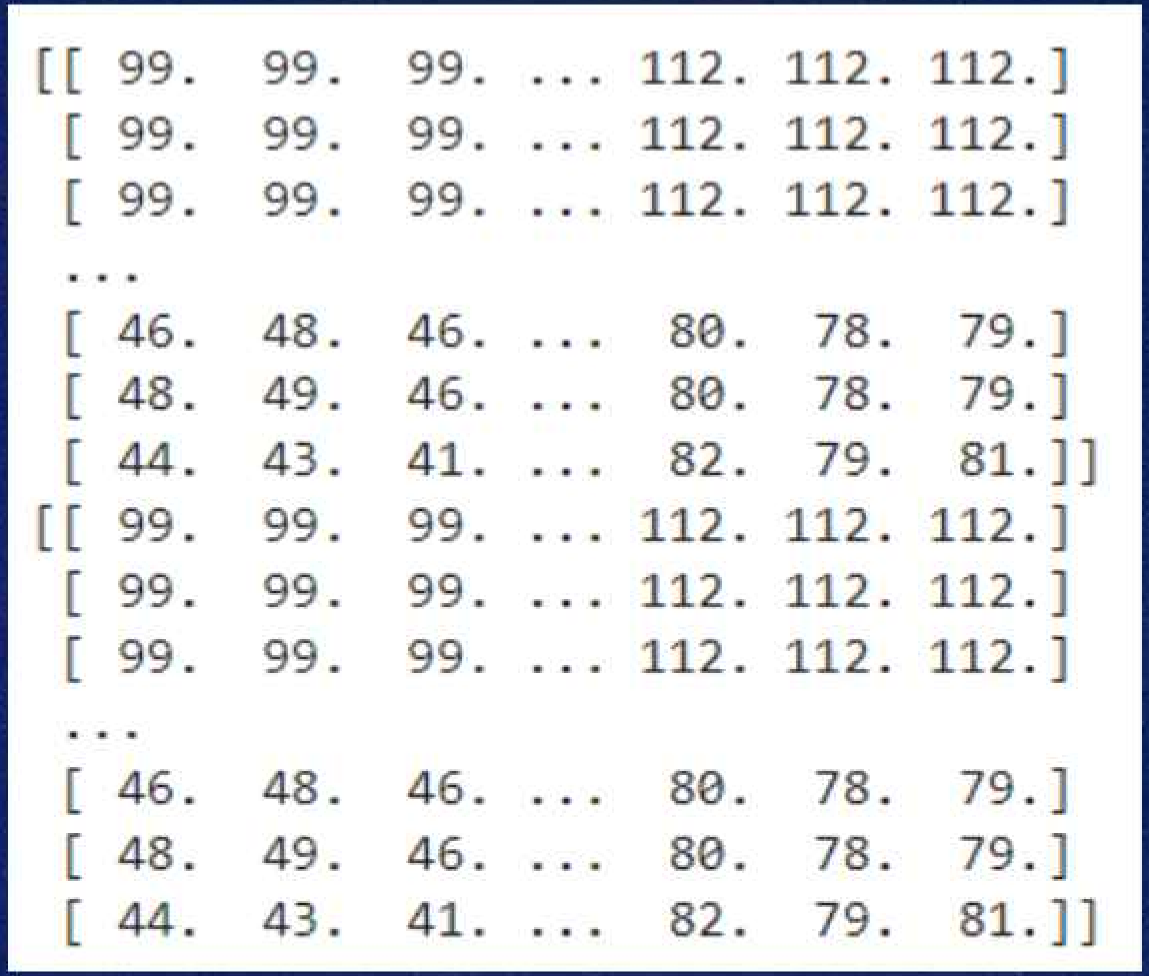


```
wav_img = (0.299*R) + (0.587*G) + (0.114*B)
print(wav_img)
print(np.array(wav_img))
plt.axis('off')
plt.imshow(wav_img,  cmap= 'gray')
plt.savefig('Metode Weighted Average', bbox_inches='tight')
```

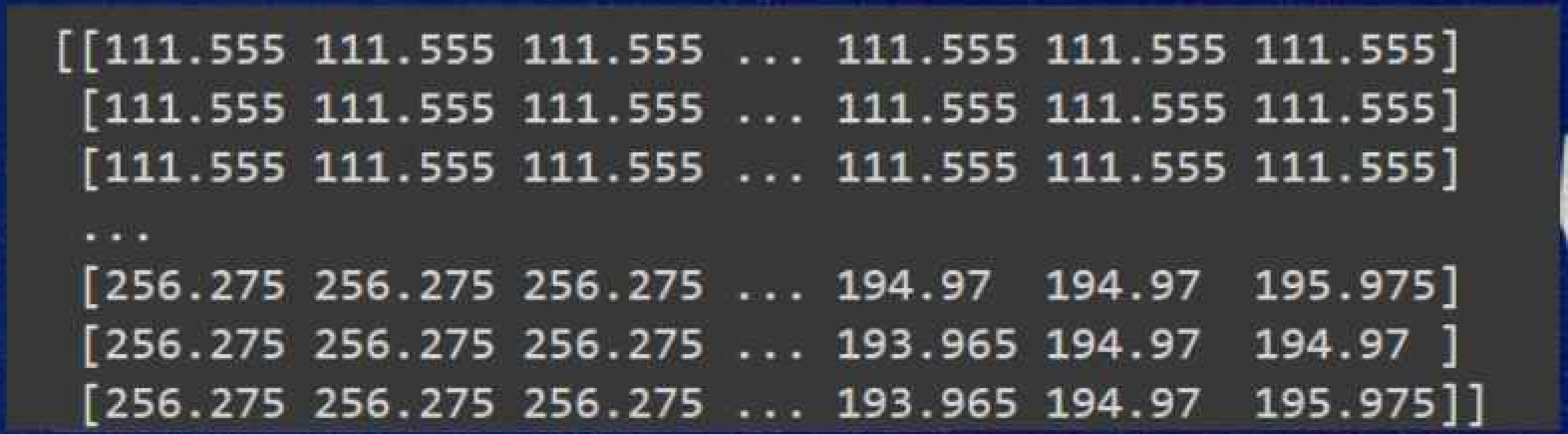
Keuntungan:

Memperhitungkan sensitivitas mata manusia terhadap warna merah, hijau, dan biru.

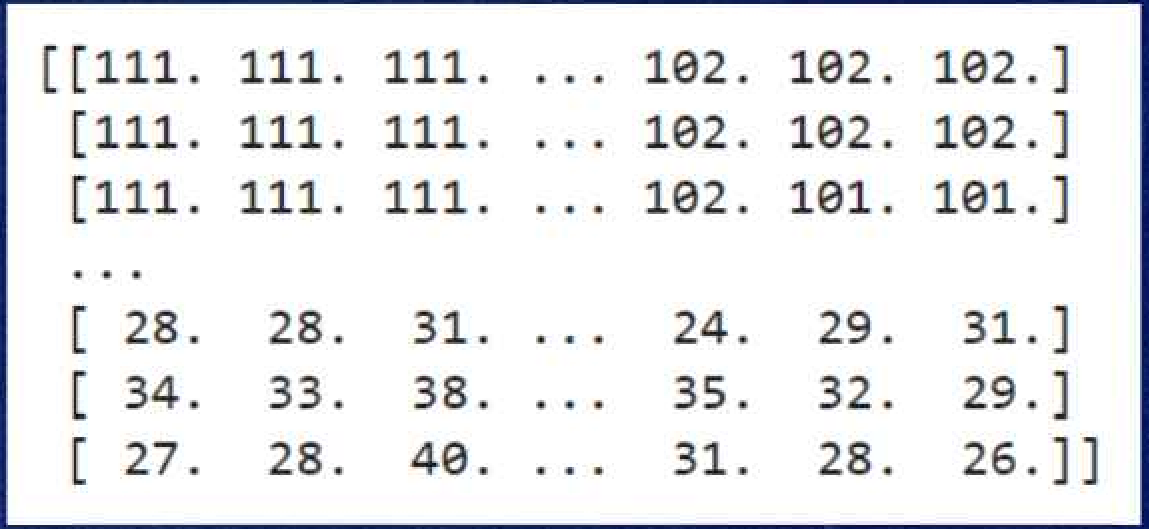
Metode Weighted Average



```
[[ 99.  99.  99. ... 112. 112. 112.]
 [ 99.  99.  99. ... 112. 112. 112.]
 [ 99.  99.  99. ... 112. 112. 112.]
 ...
 [ 46.  48.  46. ...  80.  78.  79.]
 [ 48.  49.  46. ...  80.  78.  79.]
 [ 44.  43.  41. ...  82.  79.  81.]]
[[ 99.  99.  99. ... 112. 112. 112.]
 [ 99.  99.  99. ... 112. 112. 112.]
 [ 99.  99.  99. ... 112. 112. 112.]
 ...
 [ 46.  48.  46. ...  80.  78.  79.]
 [ 48.  49.  46. ...  80.  78.  79.]
 [ 44.  43.  41. ...  82.  79.  81.]]
```



```
[[111.555 111.555 111.555 ... 111.555 111.555 111.555]
 [111.555 111.555 111.555 ... 111.555 111.555 111.555]
 [111.555 111.555 111.555 ... 111.555 111.555 111.555]
 ...
 [256.275 256.275 256.275 ... 194.97  194.97  195.975]
 [256.275 256.275 256.275 ... 193.965 194.97  194.97 ]
 [256.275 256.275 256.275 ... 193.965 194.97  195.975]]
```



```
[[111. 111. 111. ... 102. 102. 102.]
 [111. 111. 111. ... 102. 102. 102.]
 [111. 111. 111. ... 102. 101. 101.]
 ...
 [ 28.  28.  31. ...  24.  29.  31.]
 [ 34.  33.  38. ...  35.  32.  29.]
 [ 27.  28.  40. ...  31.  28.  26.]]
```

Metode Weighted Average

Kesimpulan

Dalam konversi citra RGB ke Grayscale, kami memilih metode Average sebagai favorit kami. Dengan menggunakan nilai rata-rata dari saluran warna RGB, metode ini memberikan hasil gambar grayscale yang sederhana dan netral. Pilihan ini didasarkan pada hasil gambar dari pengolahan citra. Kami merasa bahwa pendekatan ini mencerminkan pendekatan berbasis hasil untuk mencapai efek yang diinginkan dalam representasi keabuan citra sesuai dengan yang kami cari dalam konversi citra.

Thank You