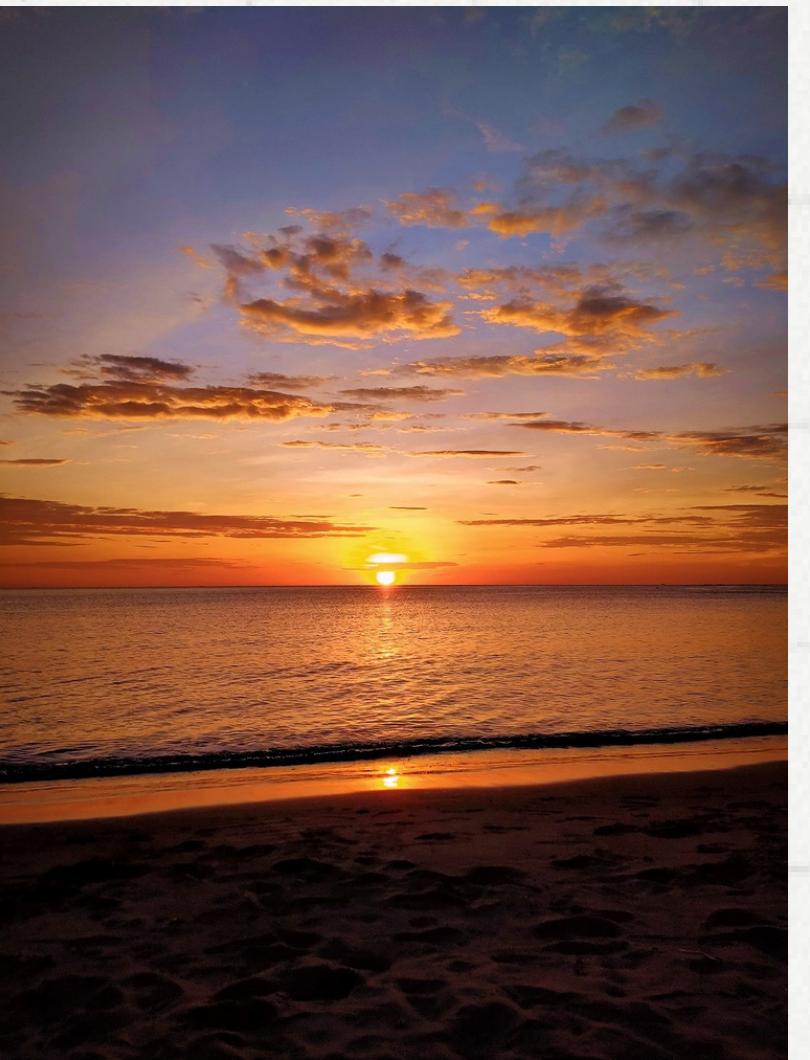


APLIKASI PADA ALJABAR LINEAR: KONVERSI CITRA RGB KE GRayscale

Muhammad Aldi Alfatih - 221011044

Nur Aini Sri Oktaviani Bahar - 221011049

NurJamaliyah J. - 221011109



AldiAlfatih.jpg



Aini.jpg

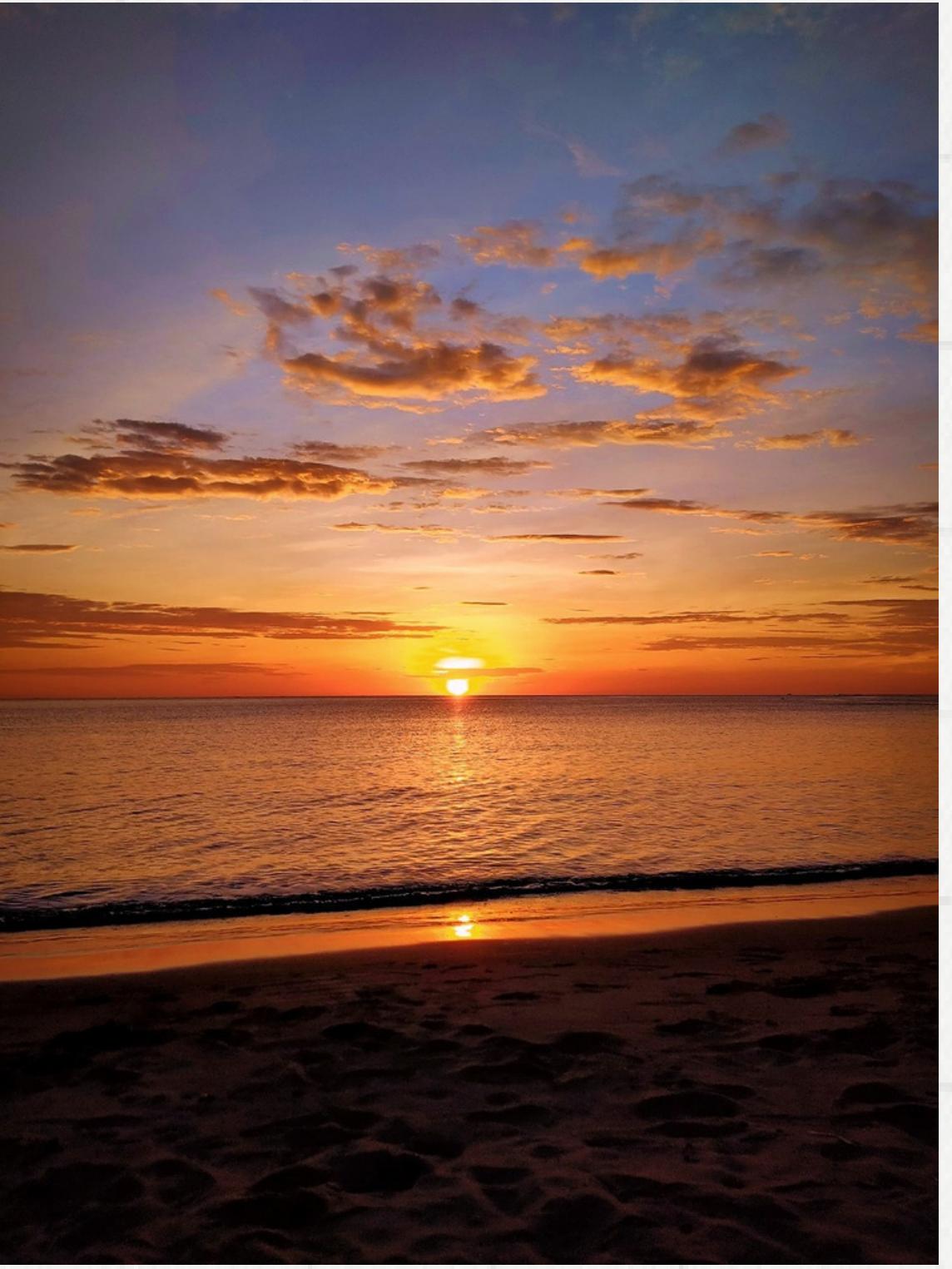


nur.jpg

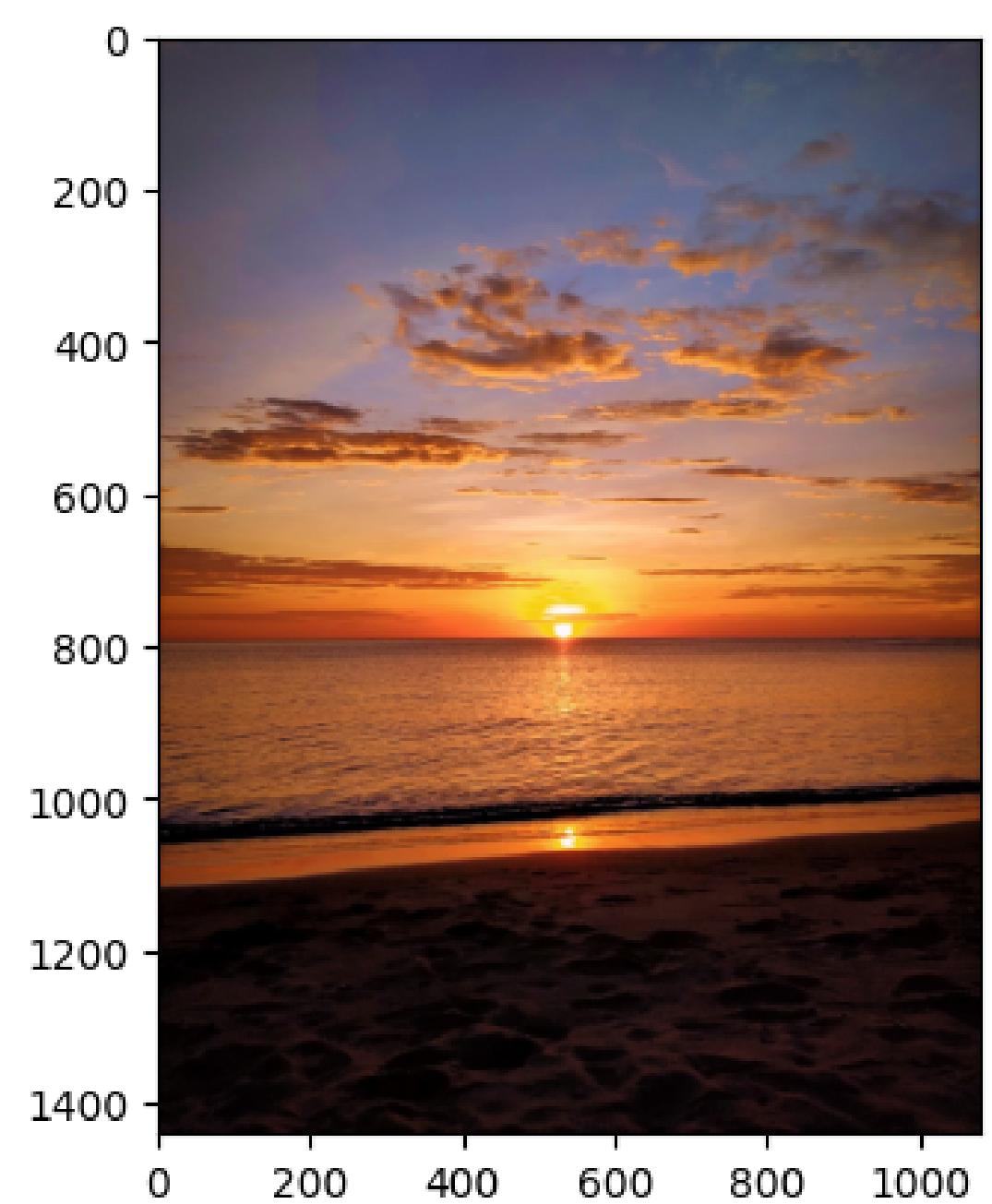
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

img_path = 'AldiAlfatih.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))
```



```
→ (1440, 1080, 3)  
[[[61 54 72]  
[57 50 66]  
[54 50 65]  
...  
[55 55 65]  
[57 55 68]  
[60 58 71]]  
  
[[62 55 73]  
[60 53 69]  
[56 52 66]  
...  
[59 59 69]  
[59 57 70]  
[59 57 70]]
```

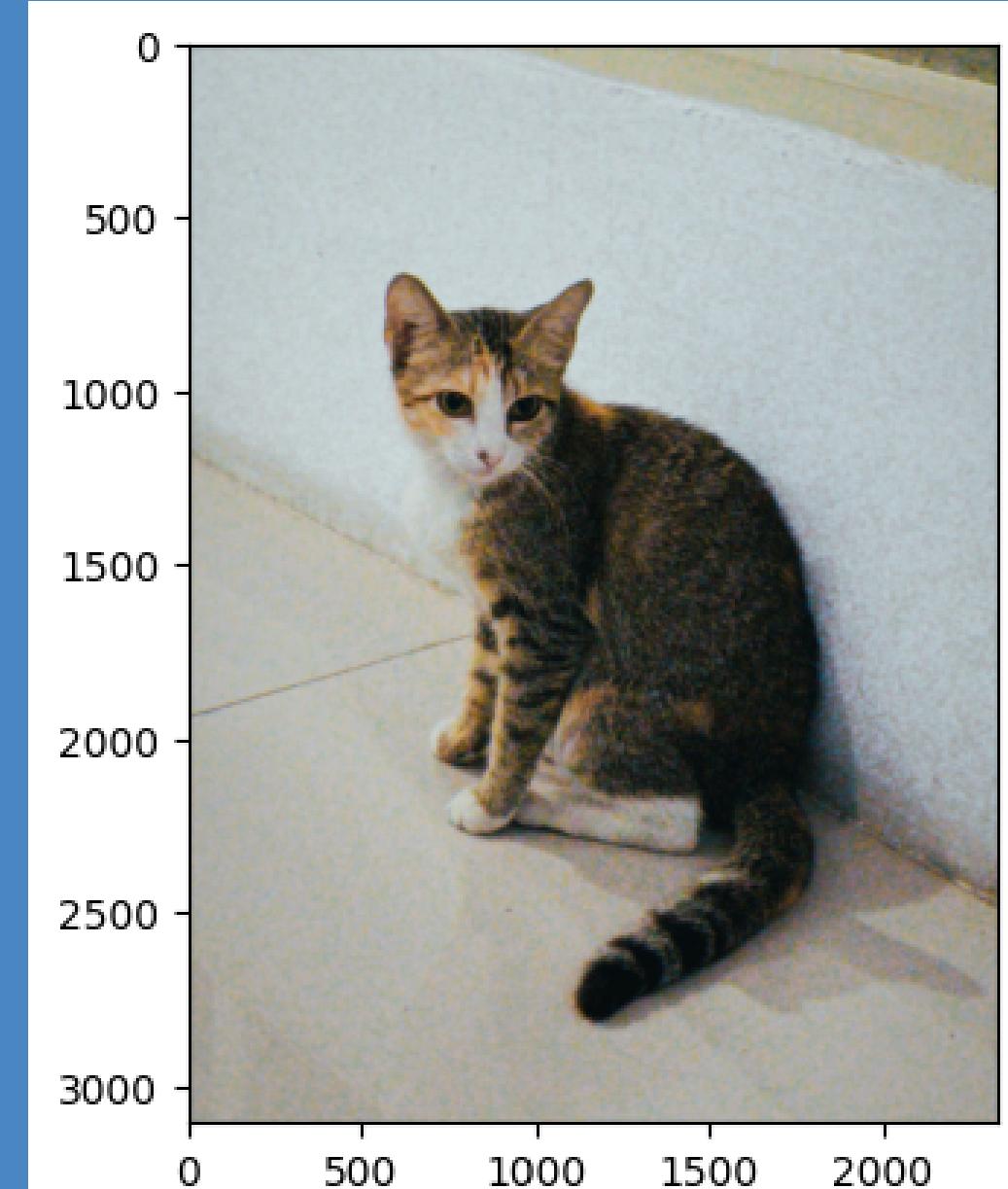


```
▶ import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
%matplotlib inline  
  
img_path = 'Aini.jpg'  
img = cv2.imread(img_path)  
print(img.shape)  
  
plt.imshow(img)  
  
fix_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
plt.imshow(fix_img)  
  
R, G, B = fix_img[:, :, 0], fix_img[:, :, 1], fix_img[:, :, 2]  
print(np.array(fix_img))
```



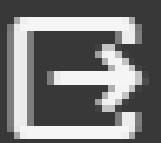


```
(3106, 2330, 3)  
[[[180 199 197]  
 [180 199 197]  
 [182 198 197]  
 ...  
 [ 87 103 76]  
 [ 91 107 78]  
 [ 91 107 78]]  
  
[[180 199 197]  
 [180 199 197]  
 [182 198 197]]
```



```
▶ import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
img_path = 'Nurjamaliyah.jpeg'  
img = cv2.imread(img_path)  
print(img.shape)  
  
plt.imshow(img)  
  
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))
```





Shape gambar: (720, 812, 3)

[[[227 231 232]

[227 231 232]

[227 231 232]]

...]



METODE LIGHTNESS

Lightness, mencari nilai tertinggi dan terendah dari nilai R, G, dan B, kemudian hasil penjumlahan nilai tertinggi dan terendah tersebut dikalikan dengan 0,5. Secara matematis dapat dirumuskan:

$$\text{Grayscale} = (\max(R, G, B)) + (\min(R, G, B)) * 0.5$$

METODE LIGHTNESS

```
fix_img[:] = np.max(fix_img, axis = -1, keepdims = 1) + 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.jpg', bbox_inches = 'tight')
```

```
→ (1440, 1080, 3)
[[[61 54 72]
 [57 50 66]
 [54 50 65]
 ...
 [55 55 65]
 [57 55 68]
 [60 58 71]]

 [[62 55 73]
 [60 53 69]
 [56 52 66]
 ...
 [59 59 69]
 [59 57 70]
 [59 57 70]]]
```



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('Mode Lightness.jpg', bbox_inches='tight')
```

→ [[[189 189 189]
[189 189 189]
[190 190 190]
...
[89 89 89]
[92 92 92]
[92 92 92]]

[[[189 189 189]
[189 189 189]
[190 190 190]



METODE LIGHTNESS

```
▶ fix_img[:] = np.max(fix_img, axis = -1, keepdims = 1) + 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.jpg', bbox_inches = 'tight')
```

RUMUS : Max (R,G,B) + Min (R,G,B) : 28

Misal : $237 + 227 : 2 = 229,5$

```
[[229.5 229.5 229.5 ... 128. 127. 125. ]  
 [229.5 229.5 229.5 ... 126. 125. 123. ]  
 [229.5 229.5 229.5 ... 123. 122. 121. ]  
 ...  
 [229.5 229.5 229.5 ... 128. 127. 125. ]]
```



METODE AVERAGE

Average, mencari nilai rata-rata dari R, G, dan B. Nilai rata-rata itulah yang dapat dikatakan sebagai grayscale. Rumus matematisnya adalah: Grayscale = (R + G + B) /3

METODE AVERAGE

```
gray_img = np.mean(fix_img, axis = 2)
print(np.array(gray_img))

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

```
[[62.33333333 57.66666667 56.33333333 ... 58.33333333 60.
   63.          ]
 [63.33333333 60.66666667 58.          ... 62.33333333 62.
   62.          ]
 [62.66666667 61.66666667 61.66666667 ... 66.33333333 62.33333333
   60.33333333]
 ...
 [13.          9.          12.          ... 8.          11.
   12.          ]
 [12.33333333 5.33333333 12.          ... 10.          9.66666667
   9.66666667]
 [13.33333333 6.33333333 12.          ... 13.66666667 10.66666667
   8.66666667]]
```



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('Metode Average.jpg' , bbox_inches='tight')
```

```
→ [[192.          192.          192.33333333 ... 88.66666667 92.  
    92.          ]  
   [192.          192.          192.33333333 ... 86.66666667 87.33333333  
    87.33333333]  
   [192.33333333 192.33333333 192.33333333 ... 85.          85.  
    85.          ]  
   ...  
   [156.66666667 156.          157.66666667 ... 157.66666667 157.33333333  
    157.33333333]  
   [157.66666667 156.          156.66666667 ... 158.66666667 158.33333333  
    158.33333333]  
   [157.          156.          156.33333333 ... 158.66666667 158.33333333  
    158.33333333]]
```



METODE AVERAGE

```
▶ gray_img = np.mean(fix_img, axis = 2)
print(np.array(gray_img))

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

```
→ [[229.5 229.5 229.5 ... 128. 127. 125. ]
 [229.5 229.5 229.5 ... 126. 125. 123. ]
 [229.5 229.5 229.5 ... 123. 122. 121. ]
 ...
 [214. 211. 214. ... 46. 46. 46. ]
 [203. 206. 216. ... 46. 46. 46. ]
 [197. 202. 216. ... 48. 48. 48. ]]
```

RUMUS : R + G + B : 3

Misalnya : 227 + 231 + 232 : 3

Hasilnya yaitu : 230



METODE LUMINOSITY

Average, mencari nilai rata-rata dari R, G, dan B. Nilai rata-rata itulah yang dapat dikatakan sebagai grayscale. Rumus matematisnya adalah: Grayscale = $(R + G + B) / 3$

METODE LUMINOSITY

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

```
[[56.7878 52.6434 51.9334 ... 55.722 56.3638 59.3638]
 [57.7878 55.6434 53.8612 ... 59.722 58.3638 58.3638]
 [57.6434 57.2864 57.2864 ... 63.722 59.722 57.722 ]
 ...
 [11.13    7.13    10.13   ... 7.1942 10.0538 11.0538]
 [10.9856  3.9856  10.13   ... 9.1942  8.9816  9.4842]
 [11.9856  4.9856  10.13   ... 12.9816  9.9816  8.4842]]
```



METODE LUMINOSITY

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

```
→ [[194.8162 194.8162 194.5262 ... 97.649 101.5046 101.5046]  
[194.8162 194.8162 194.5262 ... 95.1464 96.7172 96.7172]  
[194.5262 194.5262 194.5262 ... 93.359 93.359 93.359 ]  
...  
[161.909 161.7646 162.909 ... 160.0562 159.2006 159.2006]  
[162.6872 161.5428 161.1846 ... 159.9106 159.055 159.055 ]  
[162.0402 161.0402 160.7502 ... 159.408 159.055 159.055 ]]
```



METODE LUMINOSITY

```
[ ] lumi_img = (0.2126*R) + (0.7152*G) + (0.0722*B)  
print(lumi_img)  
  
plt.axis('off')  
plt.imshow(lumi_img)  
plt.savefig('Metode Luminosity.jpg', bbox_inches = 'tight')
```

```
[[230.2218 230.2218 230.2218 ... 129.3164 128.3164 126.3164]  
 [230.2218 230.2218 230.2218 ... 127.3164 126.3164 124.3164]  
 [230.2218 230.2218 230.2218 ... 124.3164 123.3164 122.3164]  
 ...  
 [211.0864 208.0864 211.0864 ... 45.1444 45.1444 45.1444]  
 [200.0864 203.0864 213.0864 ... 45.1444 45.1444 45.1444]  
 [194.0864 199.0864 213.0864 ... 47.1444 47.1444 47.1444]]
```

Rumus :

$$0,2126 \times R + 0,7152 \times G + 0,0722 \times B$$

Misalnya :

$$0,2126 \times 214 + 0,7152 \times 211 + 0,0722 \times 214$$

Hasilnya adalah : 211



METODE WEIGHTED AVERAGE

Average, mencari nilai rata-rata dari R, G, dan B. Nilai rata-rata itulah yang dapat dikatakan sebagai grayscale. Rumus matematisnya adalah: Grayscale = (R + G + B) /3

METODE WEIGHTED AVERAGE

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

```
[[ 58.145  53.917  52.906 ...  56.14   57.08   60.08  ],  
 [ 59.145  56.917  54.792 ...  60.14   59.08   59.08  ],  
 [ 58.917  58.39   58.39   ...  64.14   60.14   53.14  ],  
 ...  
 [ 12.033   8.033   11.033 ...   8.061   10.876  11.876],  
 [ 11.805   4.805   11.033 ...  10.061   9.762   10.05  ],  
 [ 12.805   5.805   11.033 ...  13.762   10.762   9.05  ]]
```



METODE WEIGHTED AVERAGE

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

➡ [[193.091 193.091 193.102 ... 95.138 98.91 98.91]
[193.091 193.091 193.102 ... 92.85 94.209 94.209]
[193.102 193.102 193.102 ... 91.149 91.149 91.149]
...
[161.204 160.976 162.204 ... 159.942 159.17 159.17]
[162.286 161.058 160.998 ... 160.181 159.409 159.409]
[161.77 160.77 160.781 ... 159.893 159.409 159.409]]



METODE WEIGHTED AVERAGE

Rumus : **0,299 X R + 0,587 X G + 0,114**

Misalnya : **0,299 X 234 + 0,587 X 206
+ 0,114 X 194**

```
wavr_img = (0.299*R) + (0.587*G) + (0.114*B)  
print(wavr_img)  
  
plt.axis('off')  
plt.imshow(wavr_img, cmap = 'gray')  
plt.savefig('Metode weighted Average.jpg', bbox_inches = 'tight')
```

Hasilnya adalah : **213.004**

```
[[229.918 229.918 229.918 ... 128.313 127.313 125.313]  
 [229.918 229.918 229.918 ... 126.313 125.313 123.313]  
 [229.918 229.918 229.918 ... 123.313 122.313 121.313]  
 ...  
 [213.004 210.004 213.004 ... 45.228 45.228 45.228]  
 [202.004 205.004 215.004 ... 45.228 45.228 45.228]  
 [196.004 201.004 215.004 ... 47.228 47.228 47.228]]
```



HASIL DISKUSI KELOMPOK

Hasil konversi yang memiliki hasil terbaik yakni metode average pada gambar berikut ini:



**Terima
Kasih**