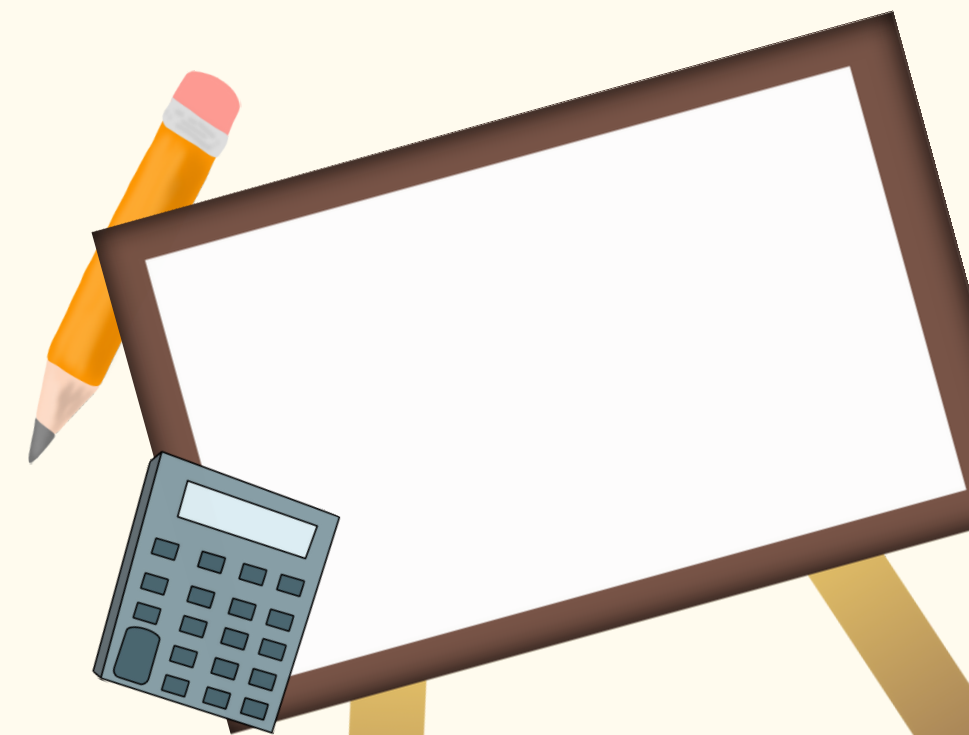
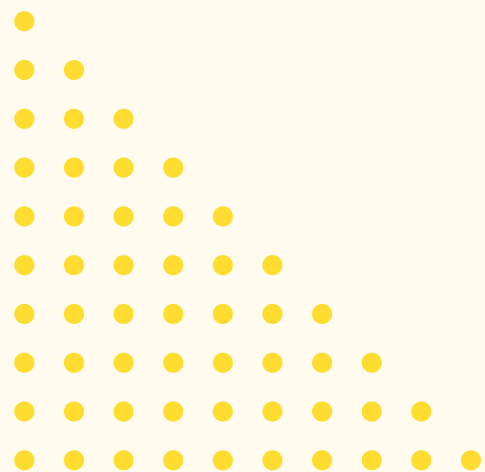


Aljabar Linear

Konversi Citra RGB ke Grayscale



ANGGOTA KELOMPOK

Nuriyyah Iffah

221011013

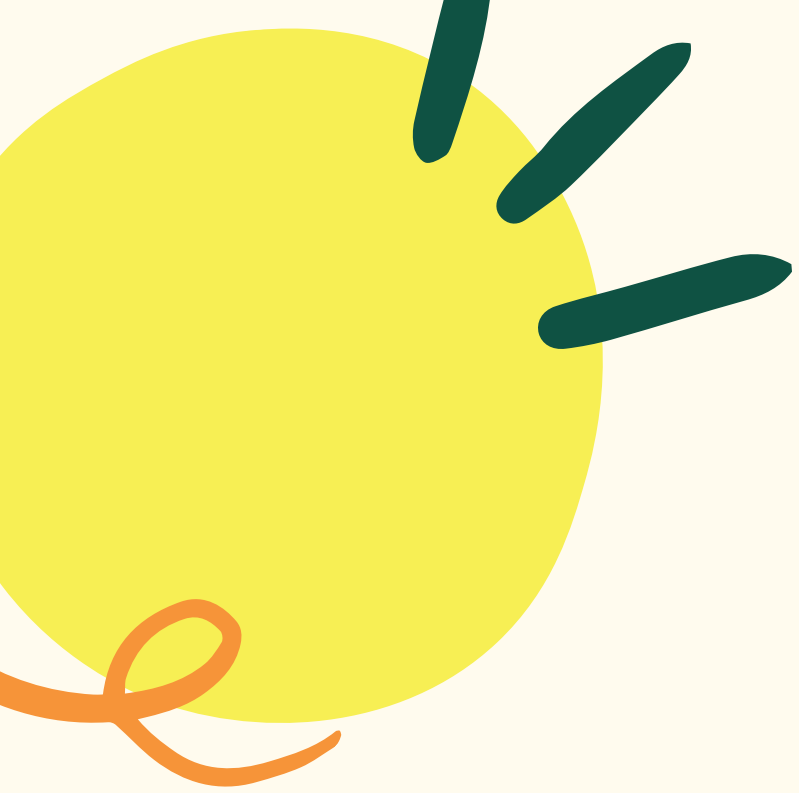
Nur Fadilah Hakim

221011075

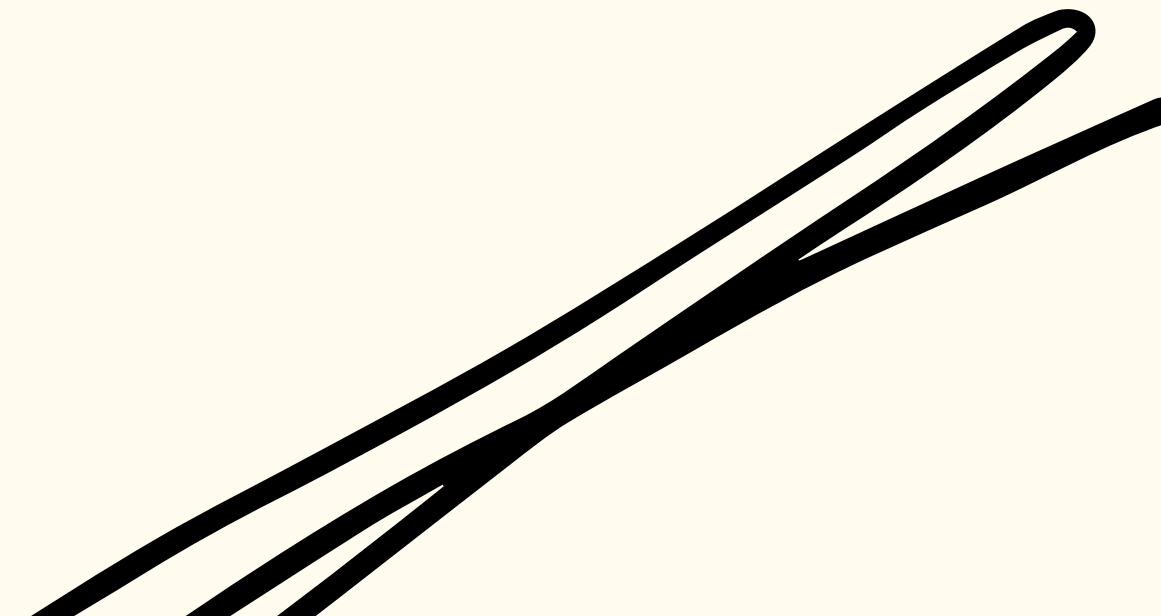
Nurul Aulia

221011084





Metode



Penjelasan

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

img_path = 'iii.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))
```

Kode disamping ini digunakan untuk mengubah format BGR menjadi RGB, dan juga untuk menampilkan susunan matriks yang mendeskripsikan gambar setelah warnanya diubah.

Iffah

```
➡ (600, 800, 3)
  [[[225 224 222]
    [227 226 224]
    [230 229 227]
    ...
```

600 = Tinggi Gambar

800 = Lebar Gambar

3 = Jumlah Saluran Warna

[225 224 222]

[227 226 224]

[230 229 227]

R bernilai 225, 227 atau 230

G bernilai 224, 226 atau 229

B bernilai 222, 224 atau 227

Dila

```
(2208, 1242, 3)
[[[40 38 43]
  [36 34 39]
  [38 36 41]
```

2208 = Tinggi Gambar

1242 = Lebar Gambar

3 = Jumlah Saluran Warna

[40 38 43]

[36 34 39]

[38 36 41]

R bernilai 40, 36 atau 38

G bernilai 38, 34 atau 36

B bernilai 43, 39 atau 41

Lulu

```
➡ (1280, 720, 3)
  [[[ 10 12 9]
    [ 22 24 21]
    [ 27 29 26]
```

1200 = Tinggi Gambar

600 = Lebar Gambar

3 = Jumlah Saluran Warna

[10 12 9]

[22 24 21]

[27 29 26]

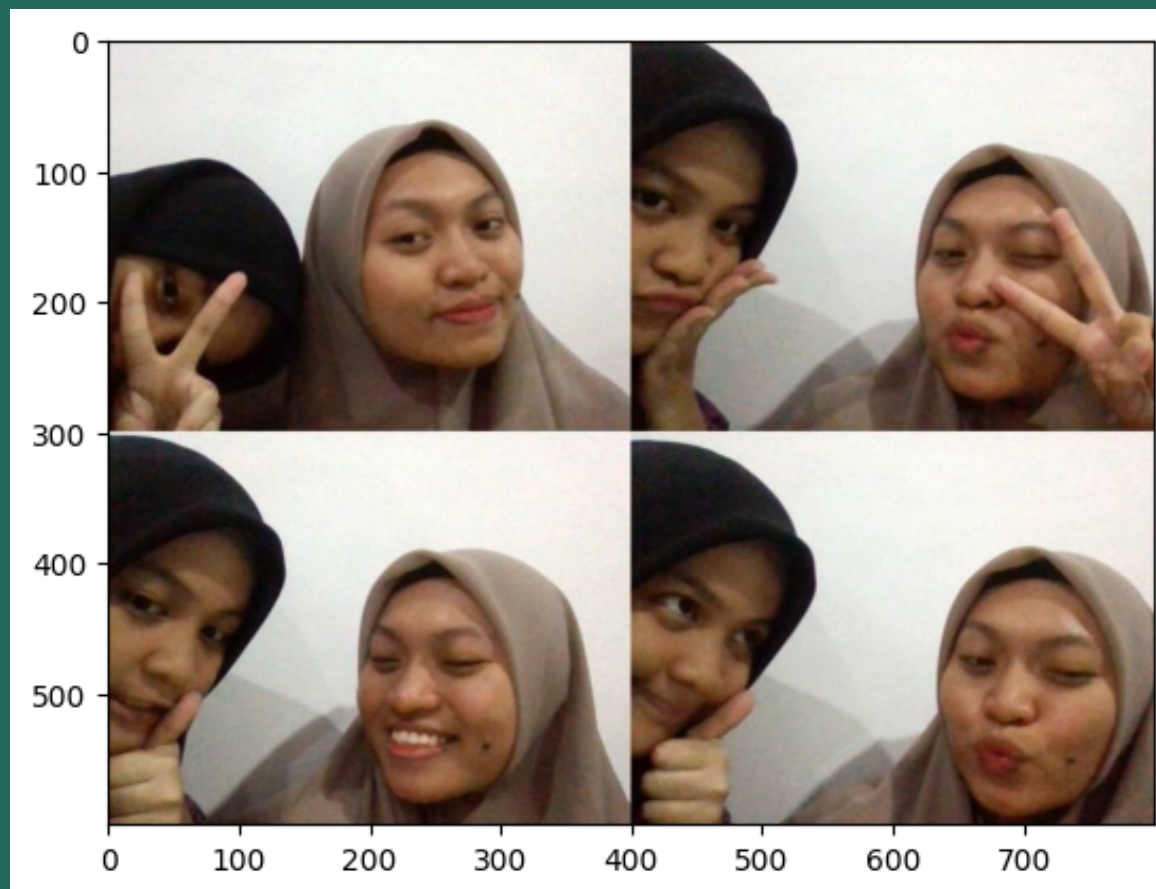
R bernilai 10, 22 atau 27

G bernilai 12, 24 atau 29

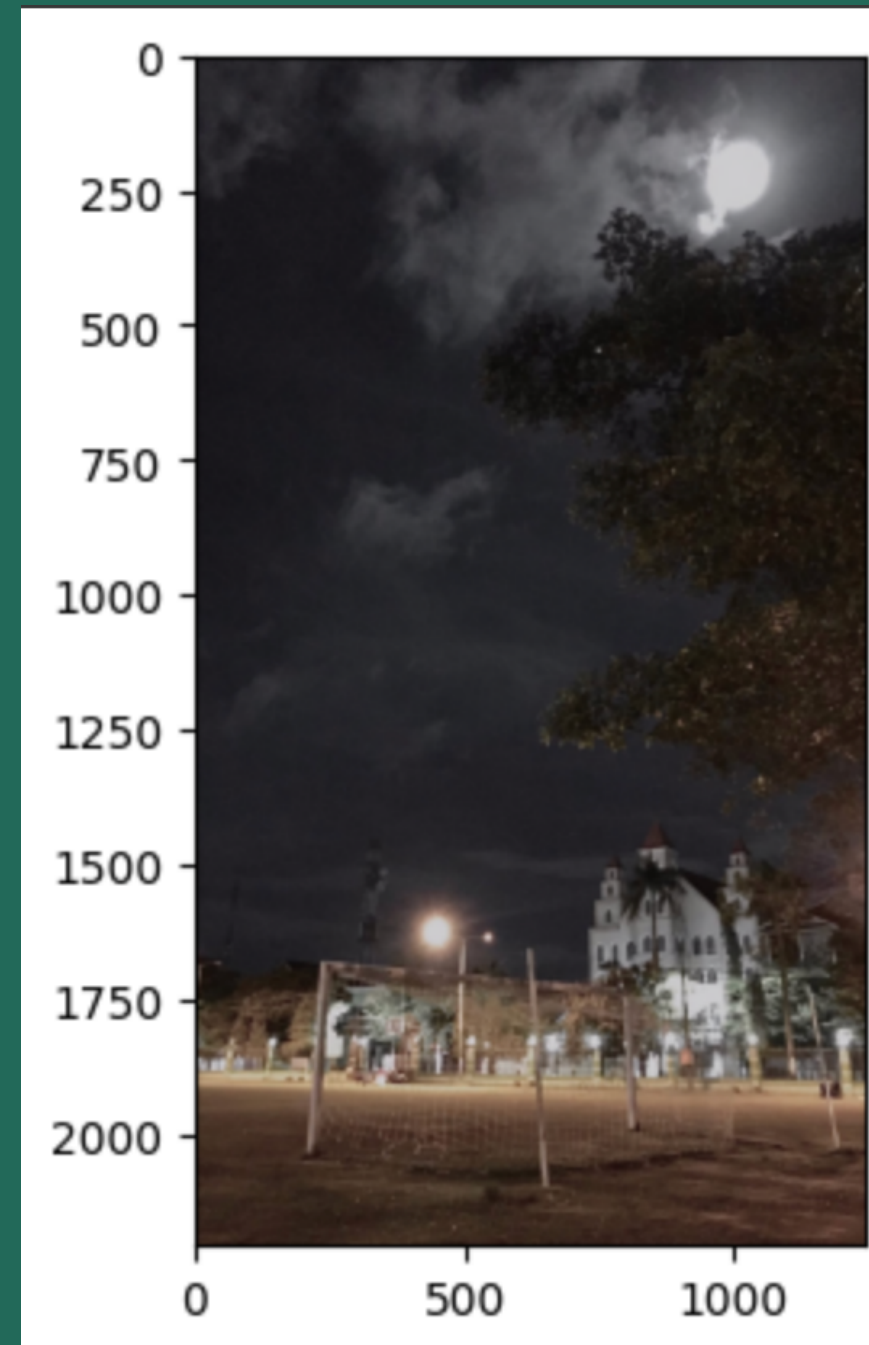
B bernilai 9, 21 atau 26

OUTPUT KODE

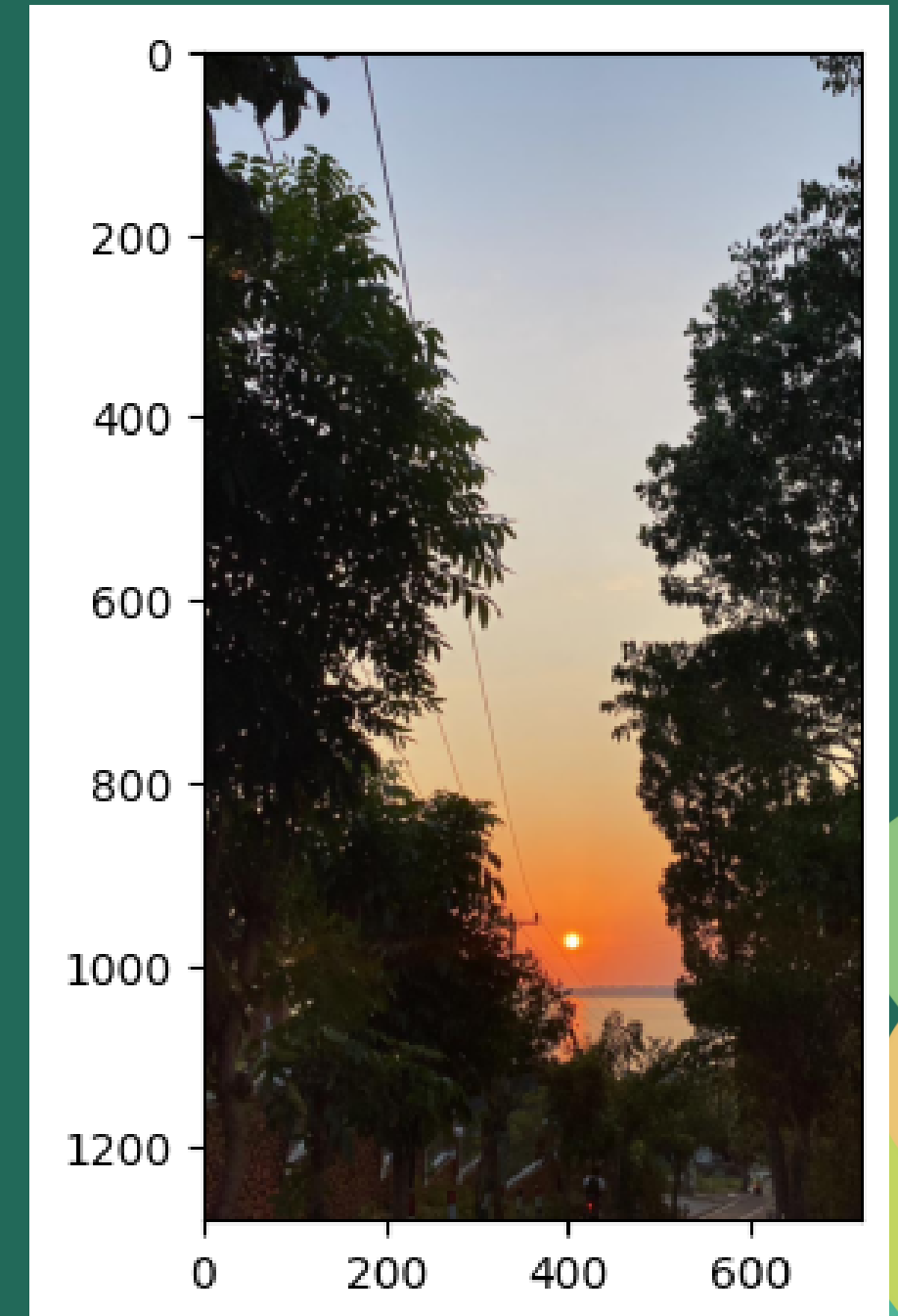
Iffah



Dila



Lulu



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')
```

Kode ini berfungsi untuk menerapkan konversi warna gambar ke skala abu-abu dengan metode lightness, di mana nilai abu-abu pada setiap piksel dihitung dengan menjumlahkan nilai tertinggi dan terendah dari nilai R, G, dan B, lalu hasilnya dibagi dua.



Iffah


```
[[[223 223 223]
  [225 225 225]
  [228 228 228]
  ...
```

Dila

```
[ [ [40 40 40]
    [36 36 36]
    [38 38 38]
```

Lulu

```
<built-in function array> [[[ 10  10  10]
  [ 22  22  22]
  [ 27  27  27]
  ...
```

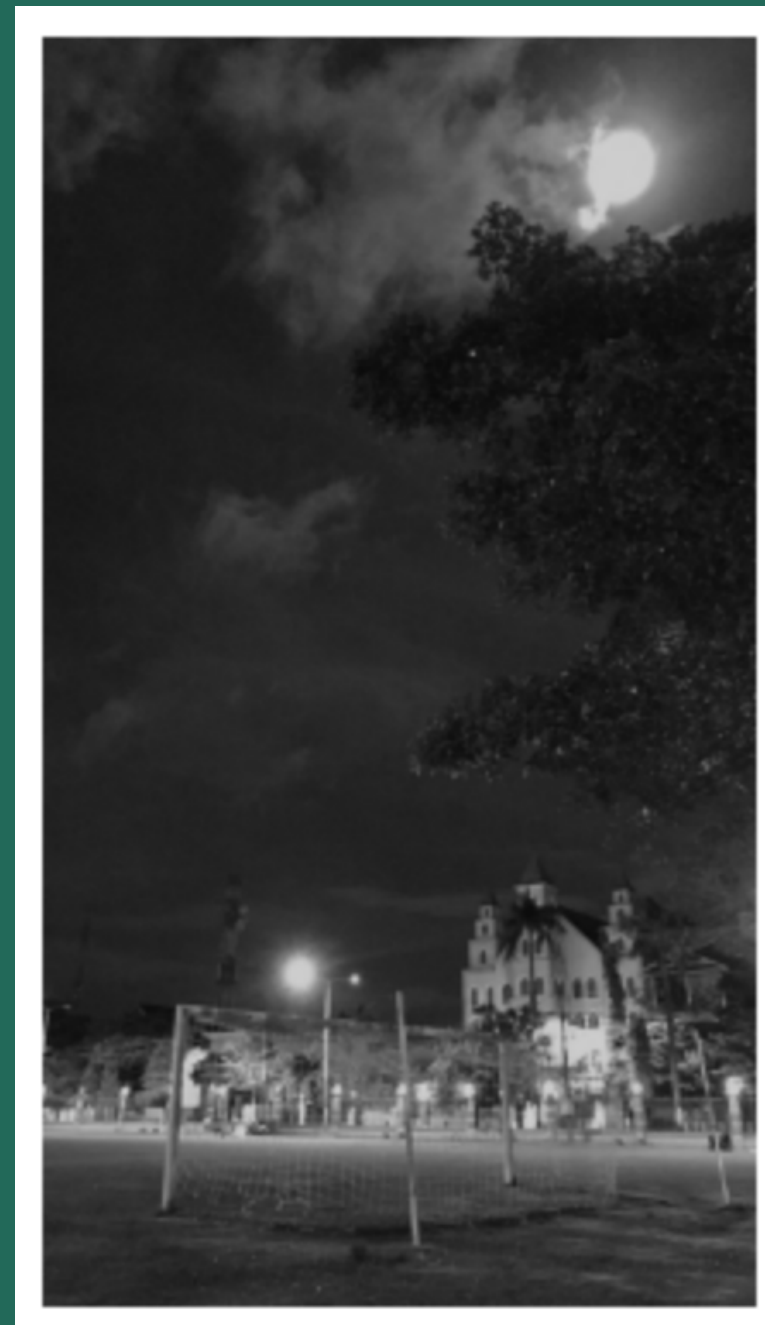


OUTPUT METODE LIGHTNESS

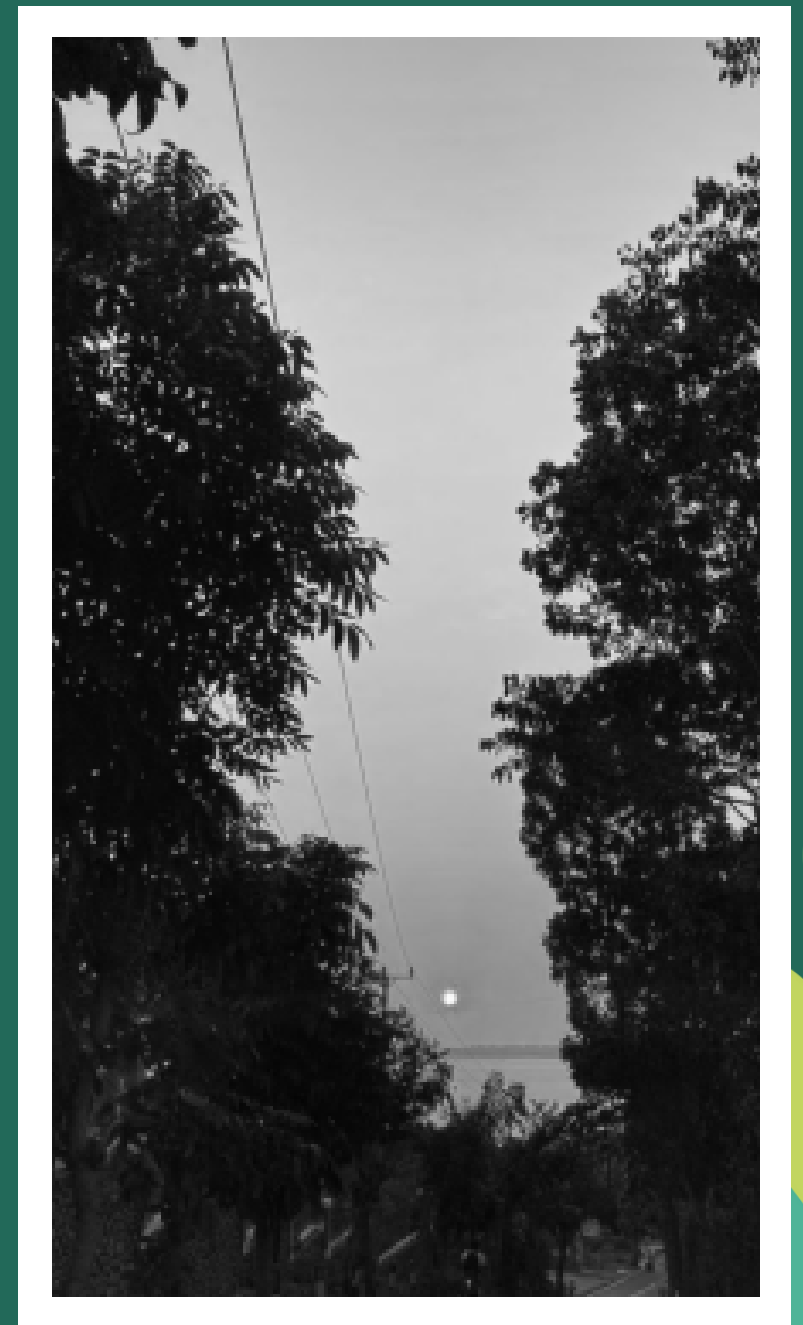
Iffah



Dila



Lulu



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.jpg', bbox_inches = 'tight')
```

Kode ini digunakan untuk mengubah gambar berwarna menjadi gambar berskala abu-abu dengan cara menghitung nilai rata-rata dari komponen warna, yaitu R, G, dan B.

```
[[223. 225. 228. ... 242. 239. 237.]  
 [223. 225. 228. ... 242. 240. 238.]  
 [223. 225. 227. ... 242. 240. 239.]
```

Iffah

```
[[40. 36. 38. ... 71. 60. 60.]  
 [38. 35. 37. ... 63. 55. 55.]  
 [37. 35. 37. ... 56. 51. 51.]
```

Dila

```
[[ 10.  22.  27. ... 102. 151. 235.]  
 [ 15.  22.  24. ...   9.  43.  47.]  
 [ 18.  20.  19. ...  53.  30.  42.]
```

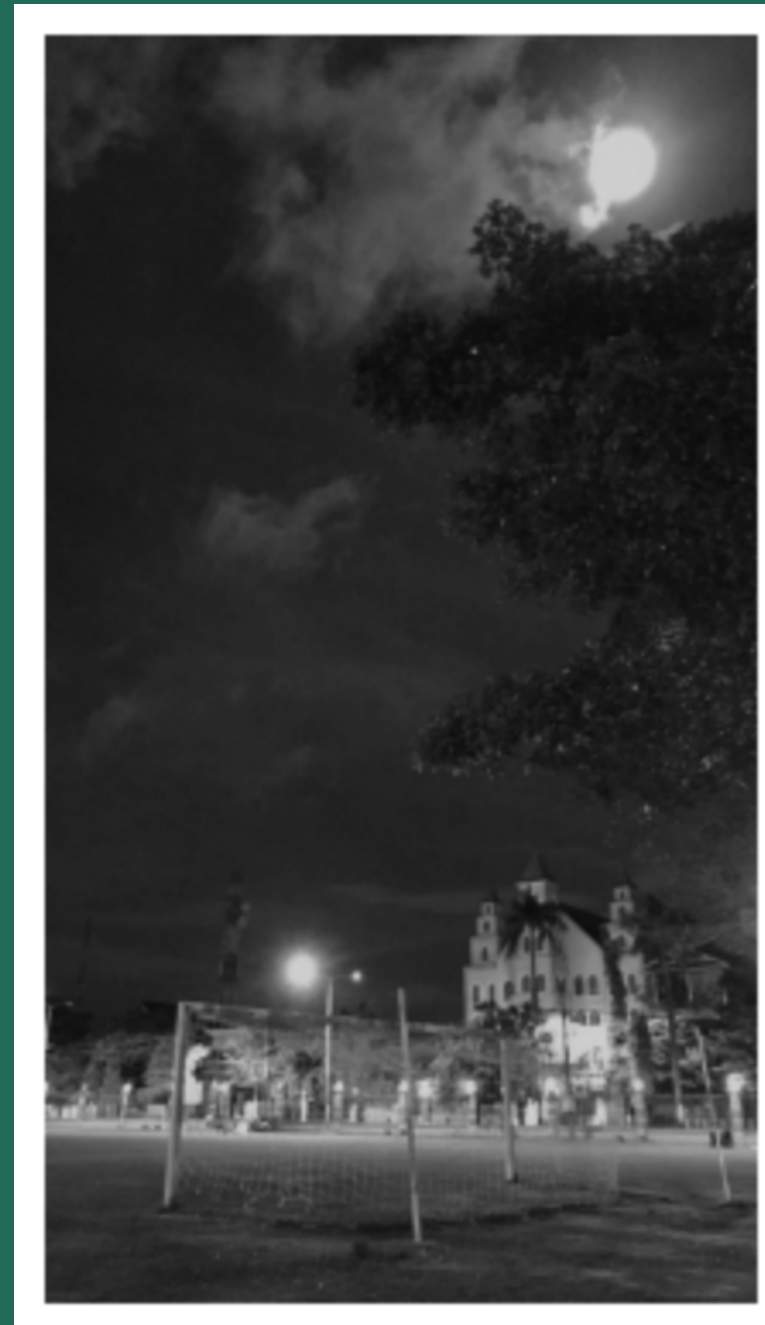
Lulu

OUTPUT METODE AVERAGE

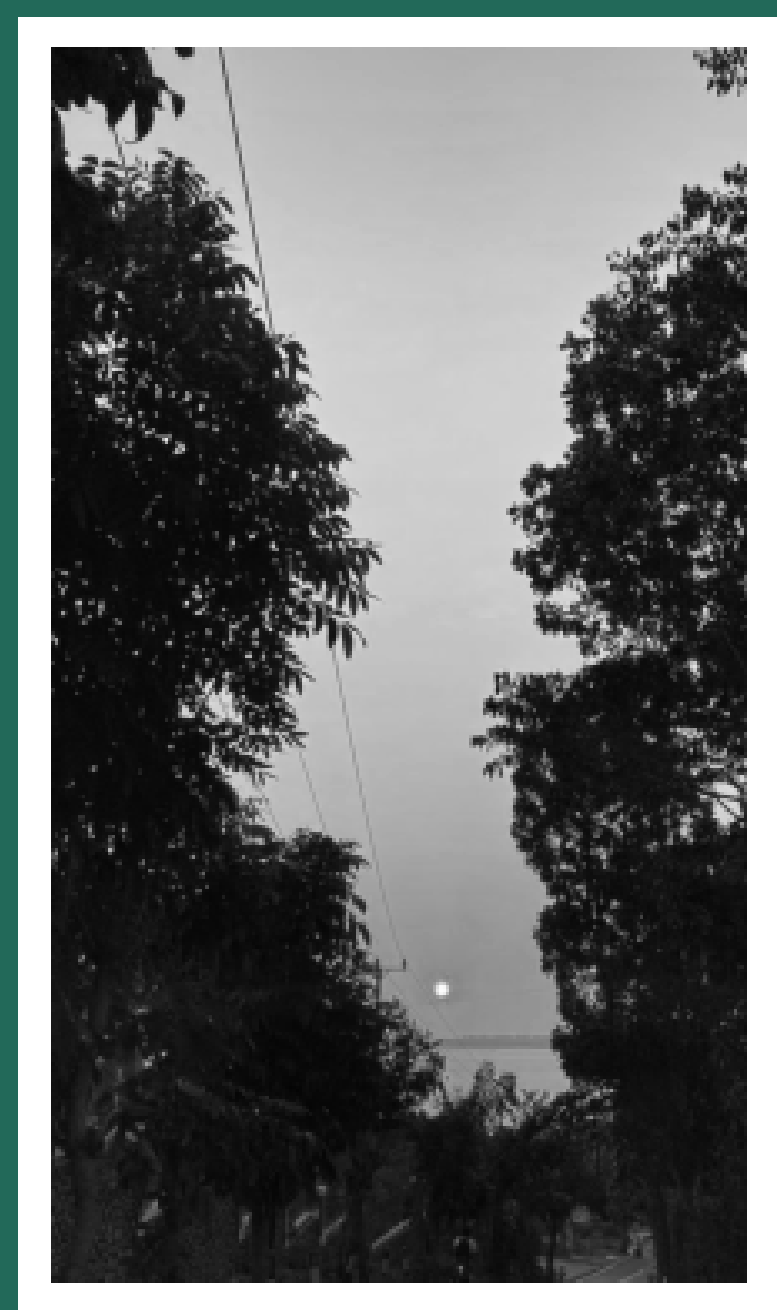
Iffah



Dila



Lulu



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')
```

Metode ini untuk mengalikan setiap nilai R, G, dan B dengan konstanta tertentu yang sudah ditetapkan nilainya, kemudian hasil perkalian seluruh nilai R, G, B dijumlahkan satu sama lain.

```
[[223. 225. 228. ... 242. 239. 237.]  
 [223. 225. 228. ... 242. 240. 238.]  
 [223. 225. 227. ... 242. 240. 239.]
```

Iffah

```
[[40. 36. 38. ... 71. 60. 60.]  
 [38. 35. 37. ... 63. 55. 55.]  
 [37. 35. 37. ... 56. 51. 51.]
```

Dila

```
[[ 10.  22.  27. ... 102. 151. 235.]  
 [ 15.  22.  24. ...   9.  43.  47.]  
 [ 18.  20.  19. ...  53.  30.  42.]  
 ...
```

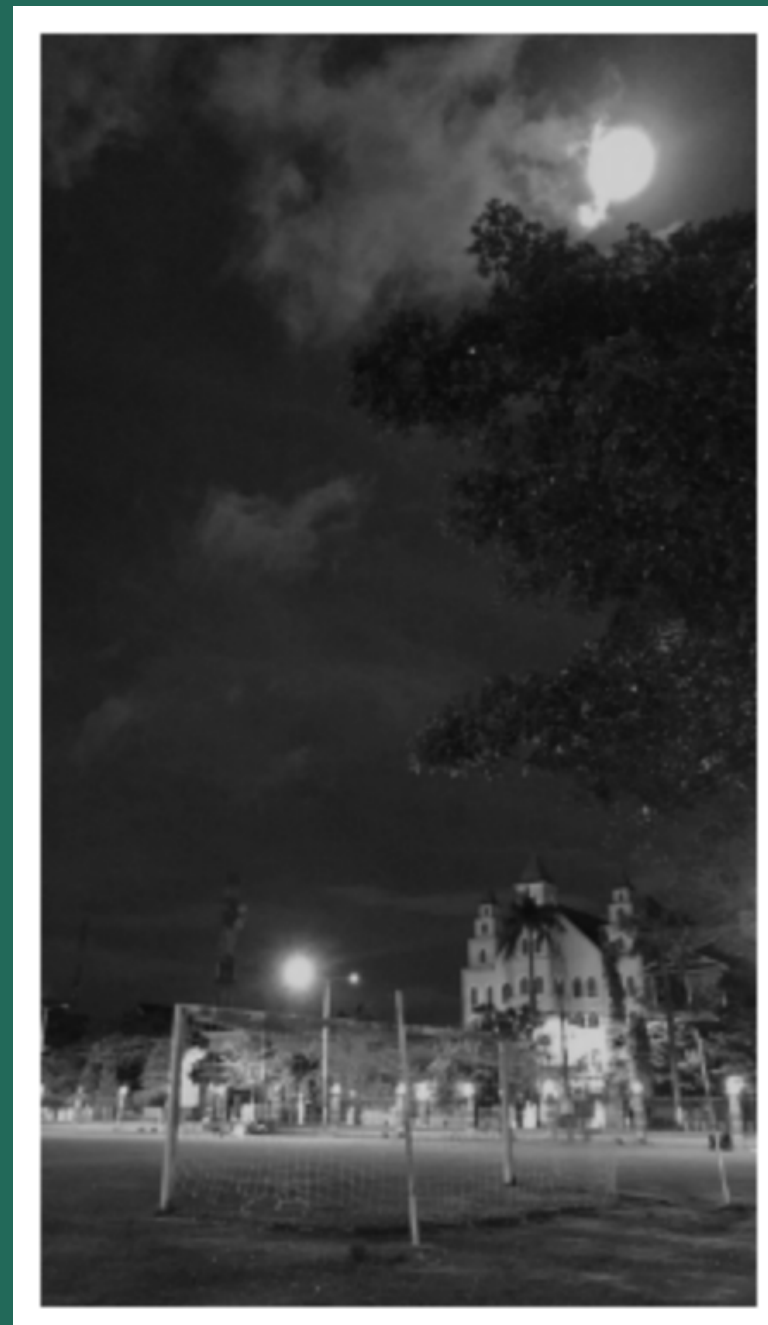
Lulu

OUTPUT METODE LUMINOSITY

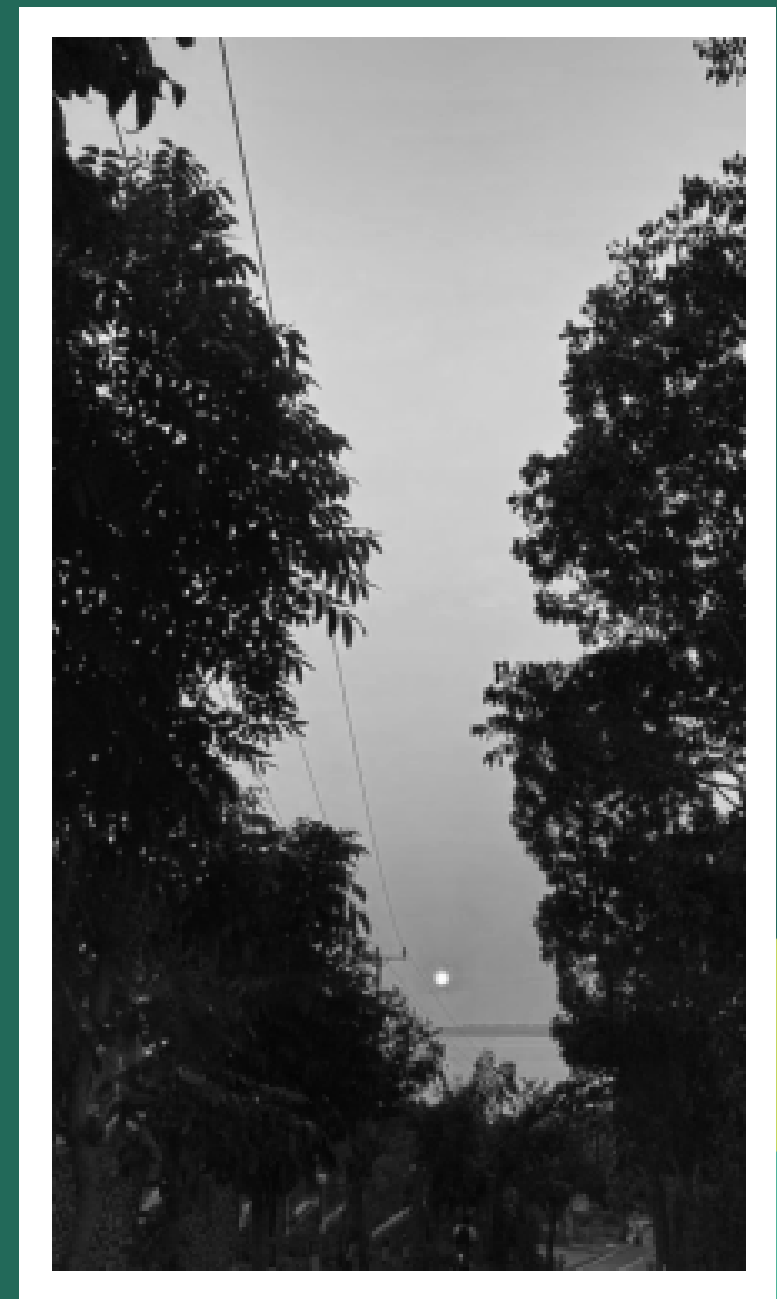
Iffah



Dila



Lulu





Kesimpulan

**Menurut hasil diskusi kelompok kami, metode yang menurut kami bagus adalah metode lightness .
karena Metode Lightness efisien dan sederhana dalam konversi warna ke citra grayscale, mempertahankan kontras warna.**

