



TUGAS PROYEK ALIN

Aplikasi pada Aljabar Linear :
Konversi Citra RGB ke Grayscale

ANGGOTA KELompok

1

Ainun Fatwa

221011012

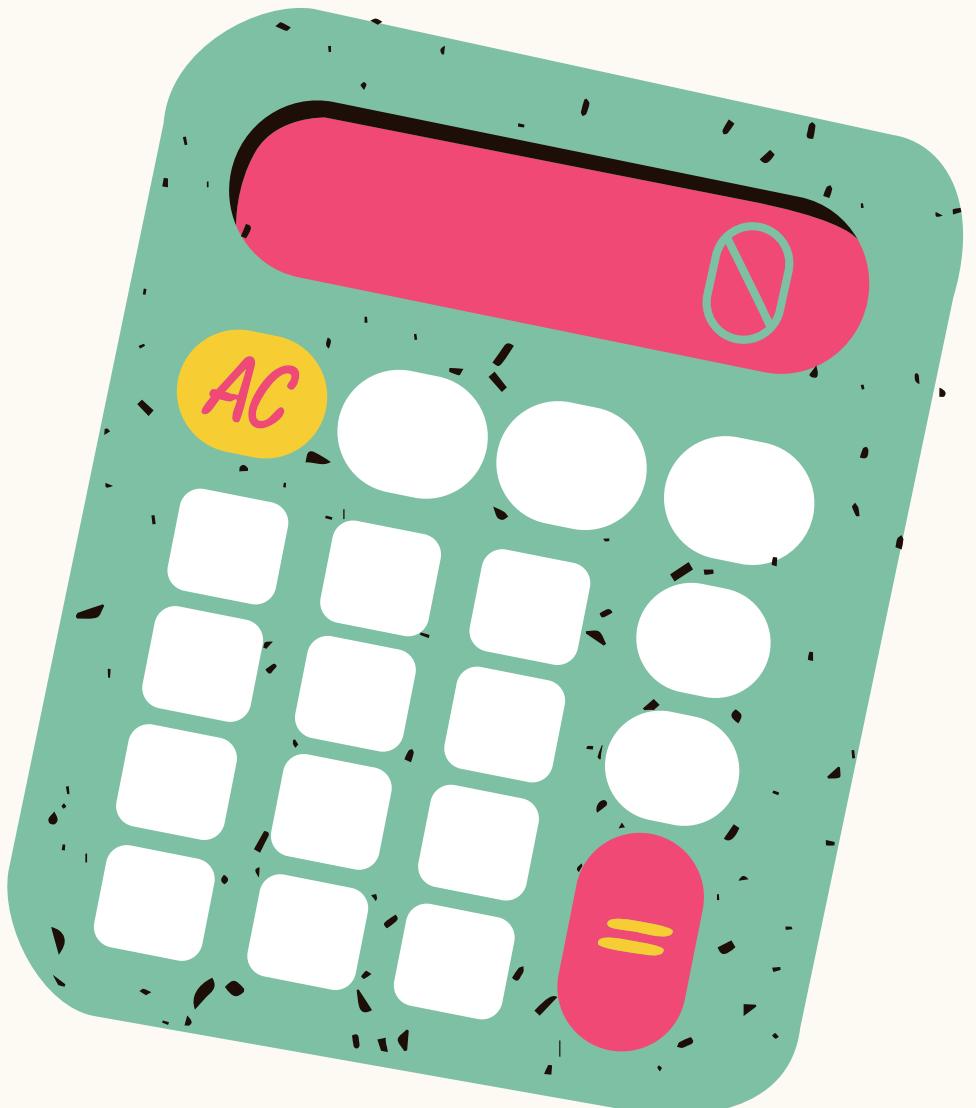
2

Muhammad fadel Hasyim

221011042

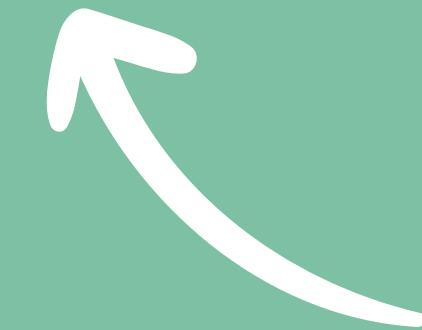
```
import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
%matplotlib inline  
  
img_path = 'Ainun fatwa .png'  
img = cv2.imread(img_path)  
print(img.shape)  
fix_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
plt.imshow(fix_img)  
R, G, B, = fix_img[:, :, 0], fix_img[:, :, 1], fix_img[:,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

Kode ini dirancang untuk membuka,
memanipulasi, dan menyimpan gambar.



(892, 720, 3)
[[[104 102 90]
[95 93 81]
[87 85 73]

Gambar 1



tentang gambar yang telah dibaca:

- Ukuran gambar: 892 piksel lebar x 720 piksel tinggi.
- 3 = jumlah saluran warna
 - [104 102 90]
 - [95 93 81]
 - [98 85 73]
- R bernilai 104 atau 102
G bernilai 95 atau 93
B bernilai 98 atau 85

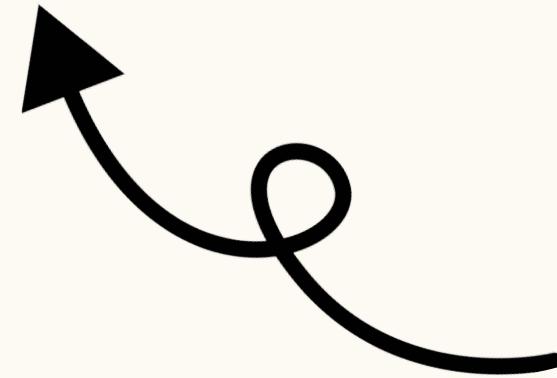


Gambar 2

- A white curved arrow points from the text in Gambar 2 to the image of the building.
- Ukuran gambar: 892 piksel lebar x 720 piksel tinggi.
 - 3 = jumlah saluran warna
 - [104 102 90]
 - [95 93 81]
 - [98 85 73]
 - R bernilai 104 atau 102
G bernilai 95 atau 93
B bernilai 98 atau 85



```
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')
```



"Lightness" untuk mengonversi gambar warna menjadi gambar keabuan (grayscale). Metode ini didasarkan pada mengambil rata-rata tertentu dari nilai maksimum dan minimum dalam setiap piksel gambar warna.



Metode Lightness



```
[[[97 97 97]
 [88 88 88]
 [80 80 80]
```

...

```
[[[169 169 169]
 [169 169 169]
 [169 169 169]
```

Penjelasan kode :
97 dan 88 merupakan hasil dari rata - rata nilai maksimum dan minimum baris 1 yang ada pada matriks sebelumnya yaitu :

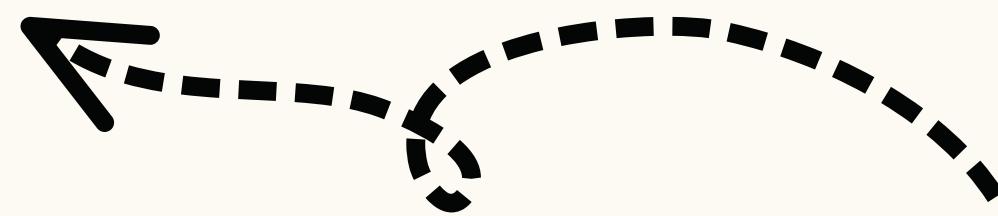
- $(90 + 104)/2 = 97$
- $(81 + 95)/2 = 88$
- $(73 + 87)/2 = 80$



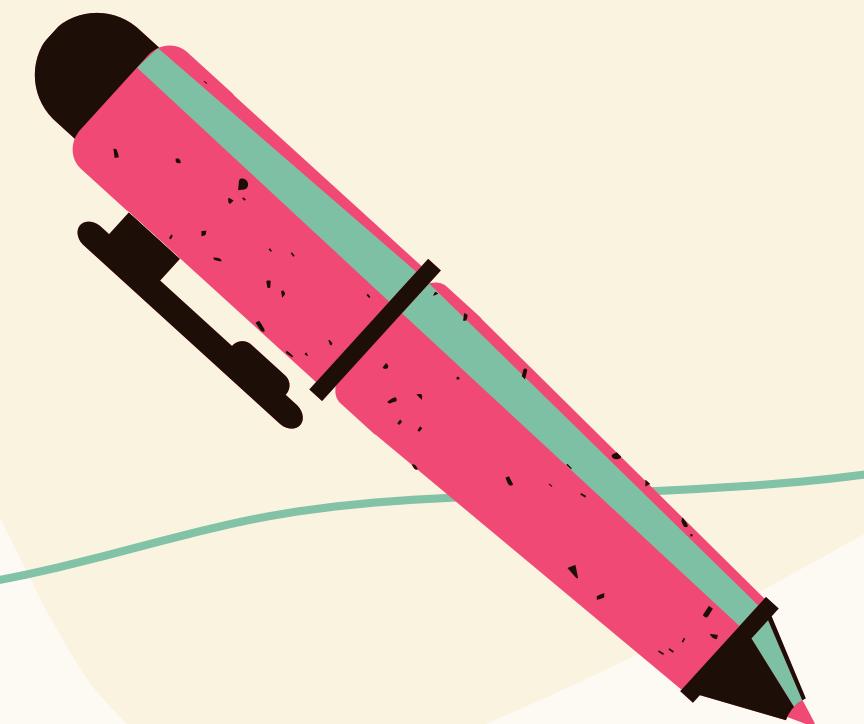
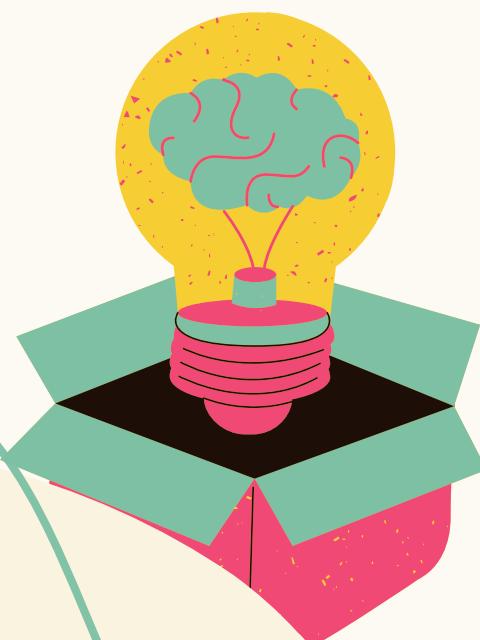
Metode Average

```
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 Lightness', 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.



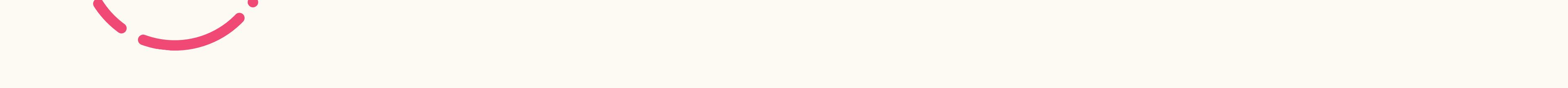
```
[[97. 88. 80. ... 74. 74. 74.]  
[96. 93. 91. ... 74. 74. 74.]  
[92. 92. 93. ... 73. 74. 74.]  
...  
[80. 81. 83. ... 61. 59. 58.]  
[87. 89. 90. ... 63. 61. 60.]  
[96. 97. 99. ... 66. 65. 65.]]
```

```
[[169. 169. 169. ... 255. 255. 255.]  
[169. 169. 169. ... 255. 255. 255.]  
[169. 169. 169. ... 255. 255. 255.]  
...  
[137. 137. 138. ... 143. 137. 139.]  
[139. 133. 122. ... 146. 149. 130.]  
[145. 141. 138. ... 156. 155. 149.]]
```

Penjelasan output gambar 2:

169 merupakan hasil dari rata-rata nilai pada baris 1 yang ada pada matriks sebelumnya yakni :

- $(144 + 168 + 194)/3 = 169$
- $(144 + 168 + 194)/3 = 169$
- $(144 + 168 + 194)/3 = 169$



```
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('Metode 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



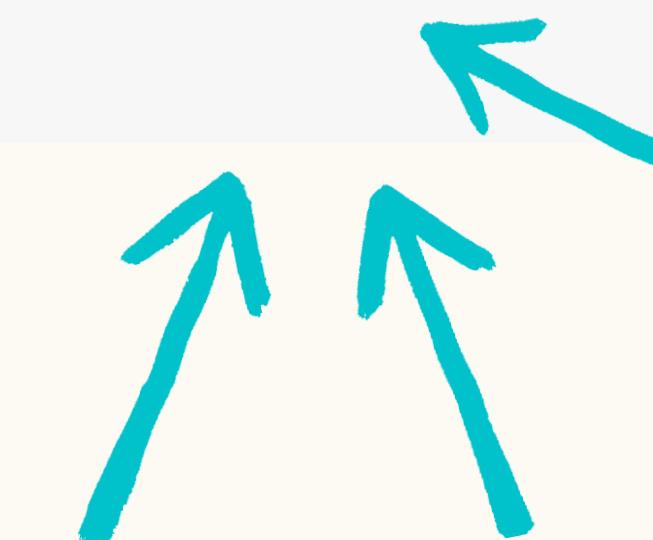
```
[[101.5588 92.5588 84.5588 ... 75.7244 75.7244 75.7244]
 [100.5588 97.5588 95.5588 ... 75.7244 75.7244 75.7244]
 [ 96.5588 96.5588 97.5588 ... 74.7244 75.7244 75.7244]
 ...
 [ 86.5966 87.5966 89.5966 ... 63.5404 61.5404 60.5404]
 [ 93.5966 95.5966 96.5966 ... 65.5404 63.5404 62.5404]
 [102.5966 103.5966 105.5966 ... 68.5404 67.5404 67.5404]]
```

```
[[169. 169. 169. ... 255. 255. 255.]
 [169. 169. 169. ... 255. 255. 255.]
 [169. 169. 169. ... 255. 255. 255.]
 ...
 [137. 137. 138. ... 143. 137. 139.]
 [139. 133. 122. ... 146. 149. 130.]
 [145. 141. 138. ... 156. 155. 149.]]
```

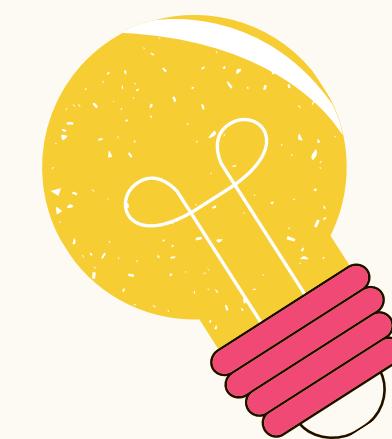
101.5588 merupakan hasil dari $(0.299 \times R) + (0.587 \times G) + (0.114 \times B)$ dimana R = 104 , G= 102, B=90 sehingga:

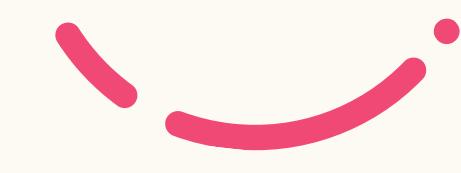
- $(0.299 \times 104) + (0.587 \times 102) + (0.114 \times 90) =$
- $(0.299 \times 95) + (0.587 \times 93) + (0.114 \times 81) =$
- $(0.299 \times 87) + (0.587 \times 85) + (0.114 \times 73) =$

```
lumi2_img = (0.299*R) + (0.587*G) + (0.114*B)  
print(np.array(lumi2_img))  
plt.axis('off')  
plt.imshow(lumi2_img, cmap='gray')  
plt.savefig('python lumi2 gambar', bbox_inches='tight')
```



namun menggunakan bobot yang berbeda untuk menghitung nilai intensitas keabuan.





```
[[97. 88. 80. ... 74. 74. 74.]  
[96. 93. 91. ... 74. 74. 74.]  
[92. 92. 93. ... 73. 74. 74.]  
...  
[80. 81. 83. ... 61. 59. 58.]  
[87. 89. 90. ... 63. 61. 60.]  
[96. 97. 99. ... 66. 65. 65.]]
```



```
[[169. 169. 169. ... 255. 255. 255.]  
[169. 169. 169. ... 255. 255. 255.]  
[169. 169. 169. ... 255. 255. 255.]  
...  
[137. 137. 138. ... 143. 137. 139.]  
[139. 133. 122. ... 146. 149. 130.]  
[145. 141. 138. ... 156. 155. 149.]]
```

KESIMPULAN

Kami memilih metode Luminosity sebagai metode yang lebih efektif dalam menghasilkan gambar grayscale. Karena metode ini memperhitungkan perbedaan sensitivitas mata manusia terhadap warna merah, hijau, dan biru, hal ini menghasilkan representasi yang lebih akurat dari kecerahan dalam gambar berwarna. Dengan demikian, menggunakan metode Luminosity dapat menghasilkan hasil yang lebih memuaskan dalam konteks pengolahan citra atau grafis.

```
lumi2_img = (0.299*R) + (0.587*G) + (0.114*B)
print(np.array(lumi2_img))
plt.axis('off')
plt.imshow(lumi2_img, cmap='gray')
plt.savefig('python lumi2 gambar', bbox_inches='tight')
```

TERIMA KASIH!

