



TUGAS PROYEK ALIN

Aplikasi pada Aljabar Linear :
Konversi Citra RGB ke Grayscale



ANGGOTA KELOMPOK

1

Ainun Fatwa

221011012

2

Muhammad fadel Hasyim

221011042



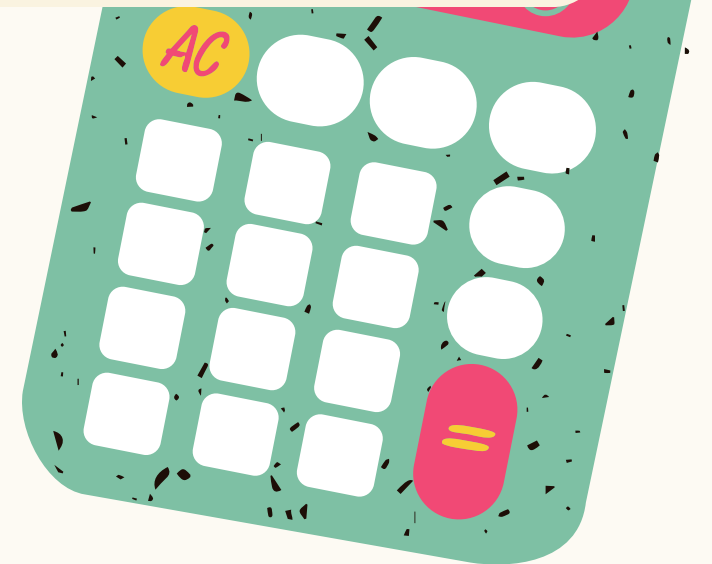
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
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))
```

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

```
img_path = 'Muhammad Fadel Hasyim.JPG'
img = cv2.imread(img_path)
print(img.shape)
R, G, B = fix_img[:, :, 0], fix_img[:, :, 1], fix_img[:, :, 2]
fix_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(fix_img)
print(np.array(fix_img))
```

kode tersebut membaca sebuah gambar, mengonversinya ke format RGB, menampilkannya menggunakan Matplotlib, dan kemudian memisahkan saluran RGB sambil mencetak nilai piksel. Jalur gambar, dan operasi khusus pada gambar dapat bervariasi tergantung pada kebutuhan Anda.



```
(1280, 720, 3)
[[[ 92  81  75]
   [ 92  81  75]
   [ 91  82  75]
   ...
```

```
(3840, 2160, 3)
[[[144 168 194]
   [144 168 194]
   [144 168 194]
```

Gambar 1

- tentang gambar yang telah dibaca:
- Ukuran gambar: dimensi tinggi 1280 piksel ,
720 piksel tinggi dan menggunakan format warna RGB
- 3 : jumlah saluran warna
 - [92 81 75]
 - [92 81 75]
 - [91 82 75]
- R bernilai 92
G bernilai 81
B bernilai 75



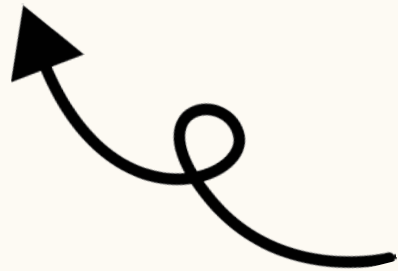
Gambar 2

- Ukuran gambar: tinggi 3840 piksel , lebar 2160 piksel
- 3 = jumlah saluran warna
 - [144 168 194]
 - [144 168 194]
 - [144 168 194]
- R bernilai 144
G bernilai 168
B bernilai 194



```
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

```
[[ [83 83 83]
  [83 83 83]
  [83 83 83]
```

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



Penjelasan kode :
83 merupakan hasil dari
nilai maksimum dan
minimum baris yang ada
pada matriks sebelumnya
yaitu :

- $(75+92)/2 = 83$
- $(75+92)/2 = 83$
- $(75+91)/2 = 83$



Penjelasan kode :
169 merupakan hasil dari nilai
maksimum dan minimum baris
1 yang ada pada matriks
sebelumnya yaitu :
 $(194 + 144) / 2 = 169$
 $(194 + 144) / 2 = 169$
 $(194 + 144) / 2 = 169$



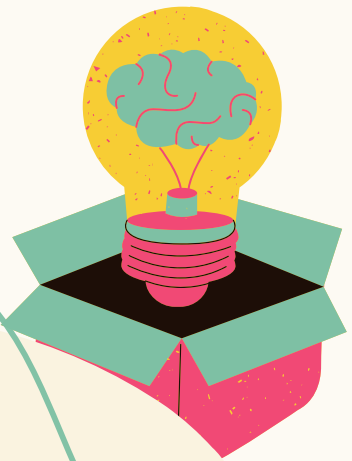
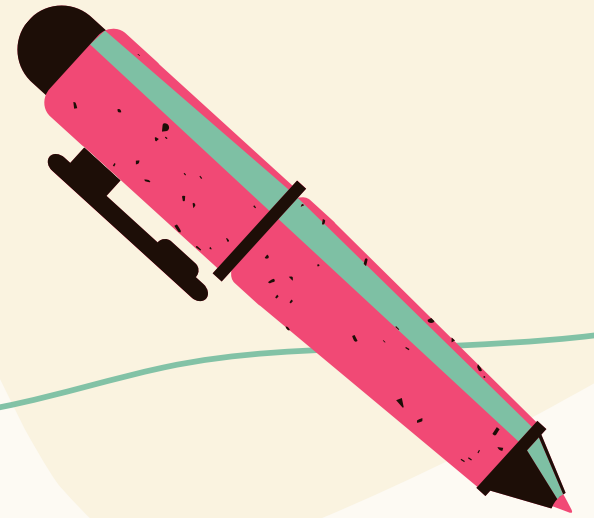
Metode Average

```
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.





```
[[83. 83. 83. ... 56. 56. 56.]  
[83. 83. 83. ... 55. 55. 55.]  
[83. 83. 83. ... 55. 55. 55.]  
...  
[80. 81. 83. ... 61. 59. 58.]  
[87. 89. 90. ... 63. 61. 60.]  
[96. 97. 99. ... 66. 65. 65.]]
```



Penjelasan output gambar 1 :

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

$$(92 + 81 + 75) / 3 = 83$$

$$(92 + 81 + 75) / 3 = 83$$

$$(91 + 81 + 75) / 3 = 83$$



```
[[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

Metode Luminosity



```
[ [83. 83. 83. ... 56. 56. 56.]
  [83. 83. 83. ... 55. 55. 55.]
  [83. 83. 83. ... 55. 55. 55.]
  ...
  [80. 81. 83. ... 61. 59. 58.]
  [87. 89. 90. ... 63. 61. 60.]
  [96. 97. 99. ... 66. 65. 65.] ]
```



gambar 1

83,56 merupakan hasil dari $(0.299 \times R) + (0.587 \times G) + (0.114 \times B)$ dimana $R = 104$, $G = 102$, $B = 90$ sehingga:

- $(0.299 \times 92) + (0.587 \times 81) + (0.114 \times 75) = 83$
- $(0.299 \times 92) + (0.587 \times 81) + (0.114 \times 75) = 83$
- $(0.299 \times 91) + (0.587 \times 81) + (0.114 \times 75) = 83$



```
[ [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.] ]
```



gambar 2

169,255 merupakan hasil dari $(0.299 \times R) + (0.587 \times G) + (0.114 \times B)$ dimana $R = 104$, $G = 102$, $B = 90$ sehingga:

- $(0.299 \times 144) + (0.587 \times 168) + (0.114 \times 194) = 169$
- $(0.299 \times 144) + (0.587 \times 168) + (0.114 \times 194) = 169$
- $(0.299 \times 144) + (0.587 \times 168) + (0.114 \times 194) = 169$



KESIMPULAN

Menurut kami , pada ketiga metode tersebut yang membuat kami tertarik itu luminosity , alasannya karena metode menghasilkan gambar yang lebih akurat kecerahannya sehingga dapat menghasilkan hasil yang lebih memuaskan dalam pengolahan 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!**

