Aplikasi Pada Aljabar Linear: Konversi Citra RGB Ke Grayscale

Tugas Presentasi Individu

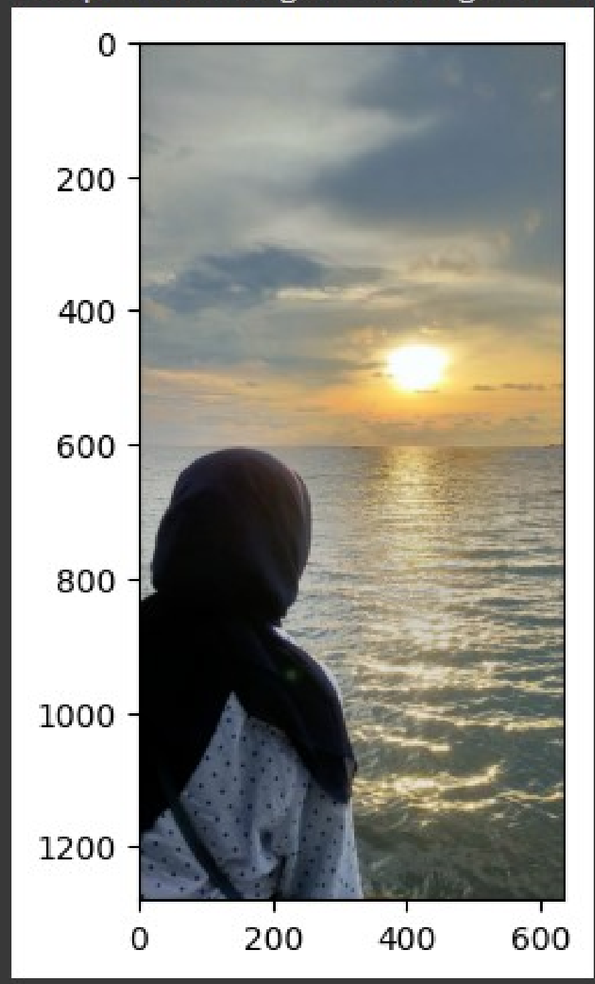
ANGGOTA KELOMPOK

1. Andi Srirahayu Putri Rasyid (221011035)
2. Andi Muhammad Kasyful Anwar (221011113)
3. Disky Fahrul Rifaih (221011068)

Andi Srirahayu Putri Rasyid

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

img_path = 'Andi Srirahayu Putri Rasyid.jpg'
img = cv2.imread(img_path)
print(img.shape)

fix_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(fix_img)
```

```
(1280, 636, 3)
<matplotlib.image.AxesImage at 0x7a92f29c79a0>
```



Andi Muhammad Kasyful Anwar

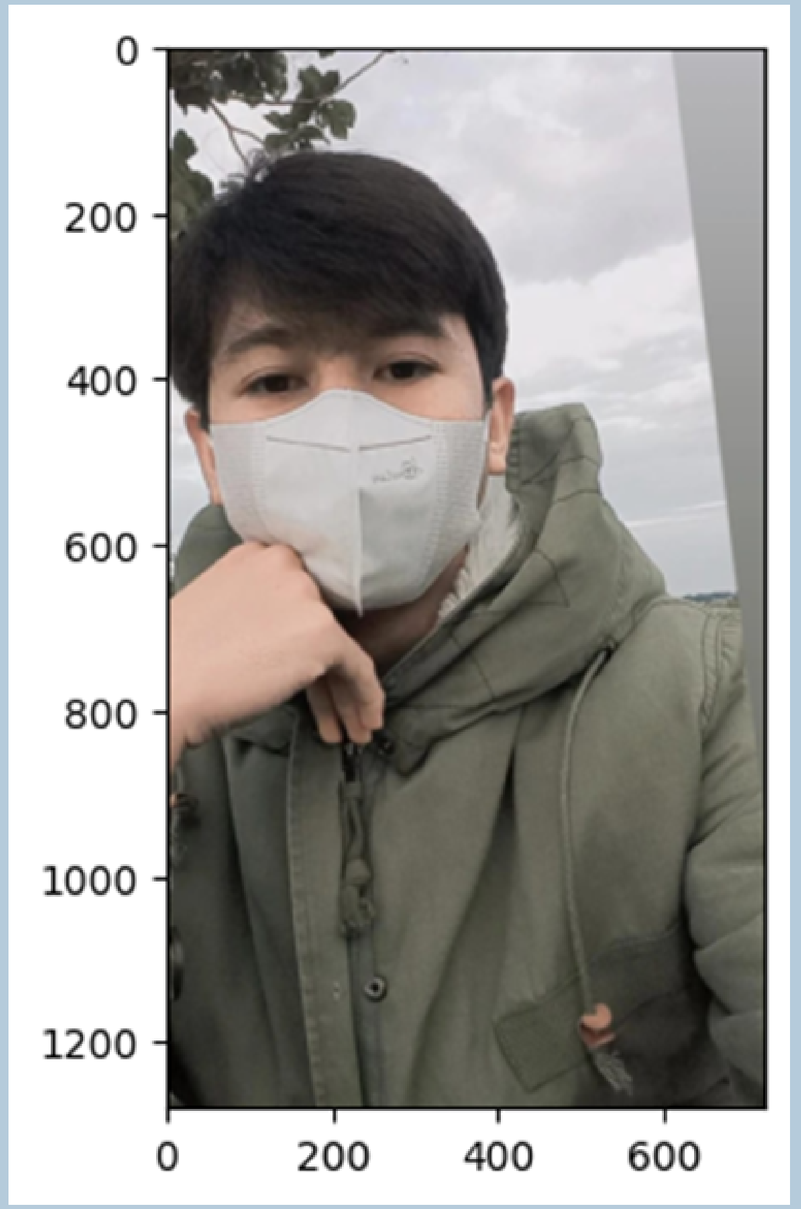
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

img_path = "Kasyful.jpg"
img = cv2.imread(img_path)
print(img.shape)

fix_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(fix_img)
```

module: cv2

Open in tab View source

OpenCV Python binary extension loader

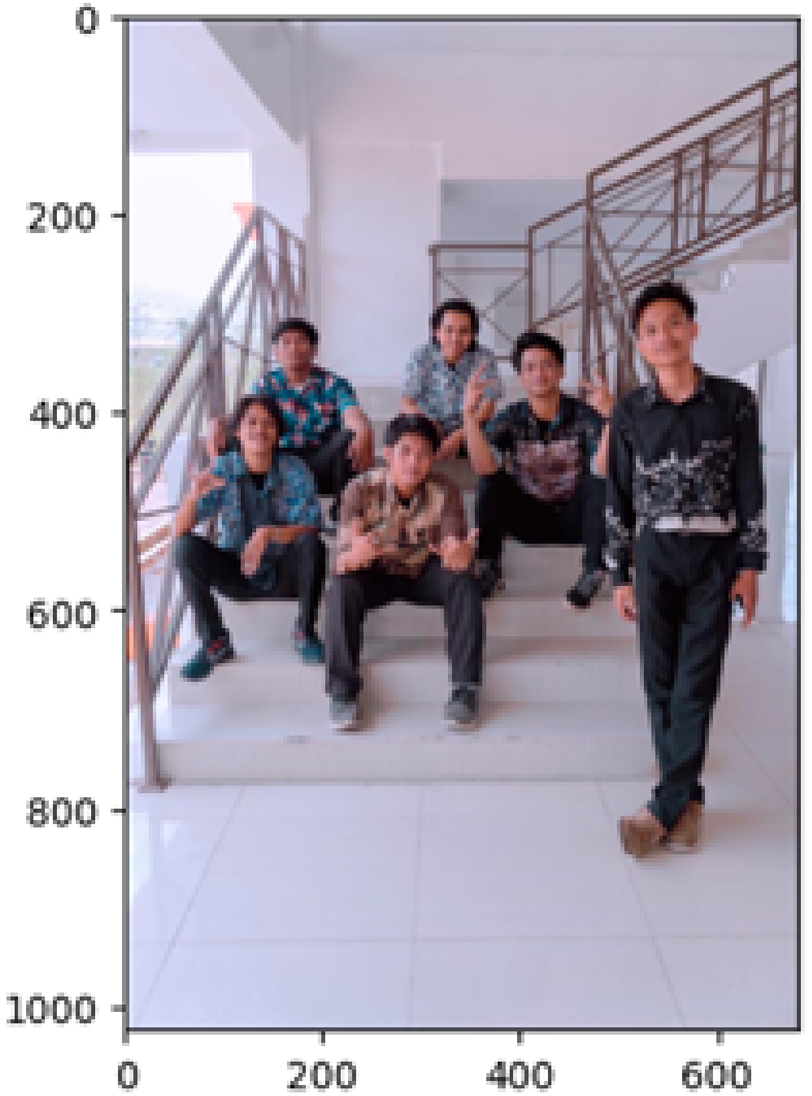
```
(1280, 720, 3)
<matplotlib.image.AxesImage at 0x7d11682039a0>
```



Disky Fahrul Rifaih

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

img_path = "diski.jpg"
img = cv2.imread(img_path)
print(img.shape)

fix_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(fix_img)
```

```
(1024, 683, 3)
<matplotlib.image.AxesImage at 0x7a6ab11314b0>
```

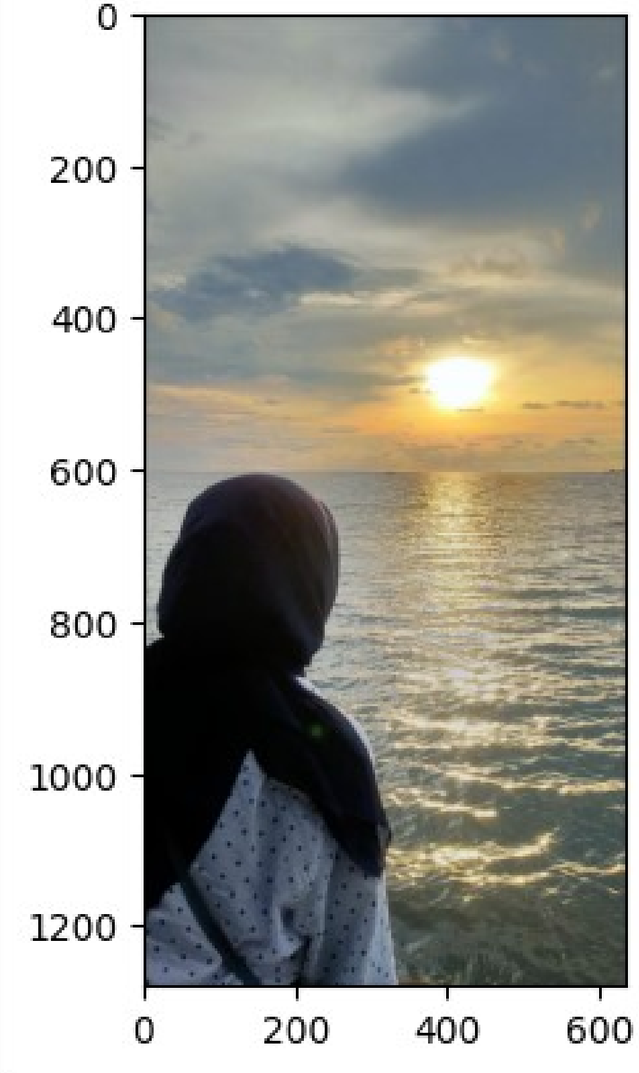


```
from ast import increment_lineno
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

img_path = 'Andi Srirahayu Putri Rasyid.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))
```

Kode tersebut membaca gambar dari path yang ditentukan, mengubah format warna gambar dari BGR ke RGB, menampilkan gambar, dan memisahkan saluran warna merah, hijau, dan biru.

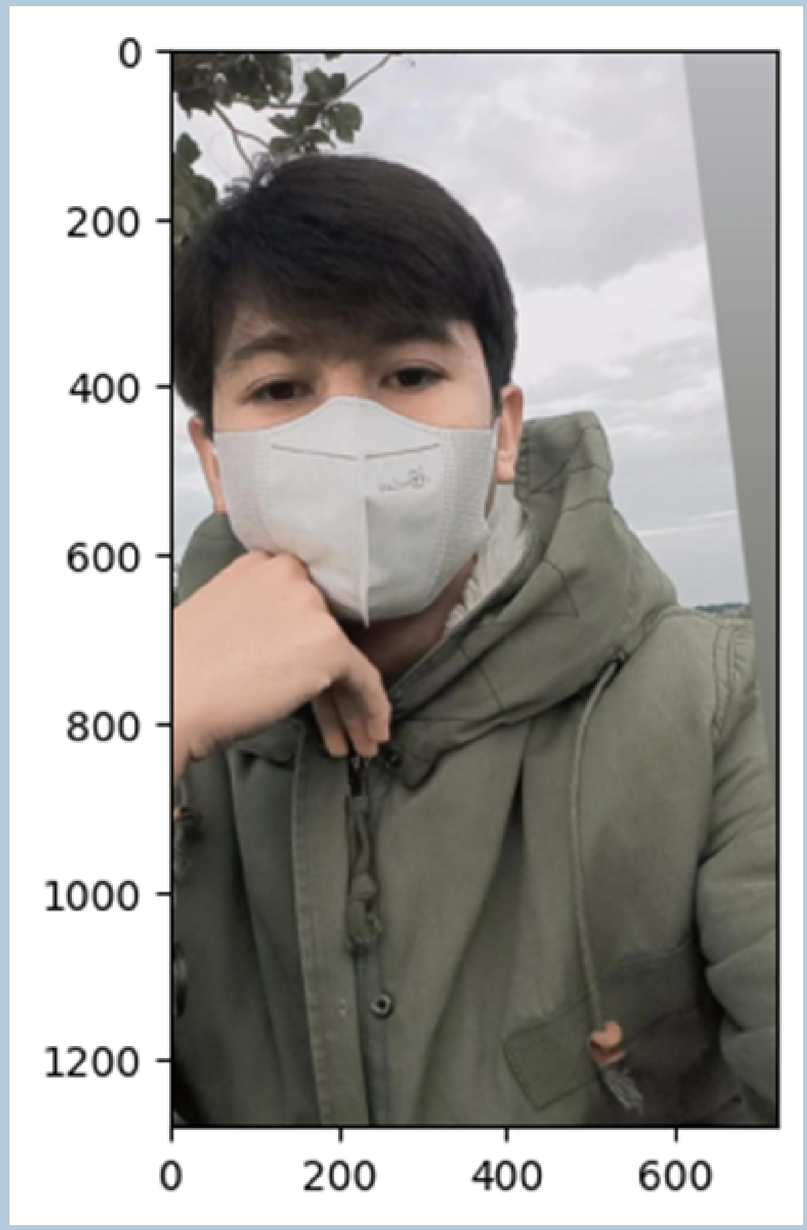
```
(1280, 636, 3)
[[[135 145 144]
  [135 145 144]
  [135 145 144]
  ...
  [102 113 119]
  [102 113 119]
  [102 113 119]]

 [[135 145 144]
  [135 145 144]
  [135 145 144]
  ...
  [102 113 119]
  [102 113 119]
  [102 113 119]]

 [[134 144 143]
  [134 144 143]
  [134 144 143]
  ...
  [102 113 119]
  [102 113 119]
  [102 113 119]]

 ...
```

```
[[ 82  93 115]
 [ 54  65  87]
 [ 14  24  49]
 ...
 [ 66  74  61]
 [ 66  74  61]
 [ 66  74  61]]

[[ 81  92 114]
 [ 60  71  93]
 [ 26  36  61]
 ...
 [ 68  76  63]
 [ 68  76  63]
 [ 68  76  63]]

[[ 80  91 113]
 [ 68  79 101]
 [ 42  52  77]
 ...
 [ 69  77  64]
 [ 69  77  64]
 [ 69  77  64]]]
```



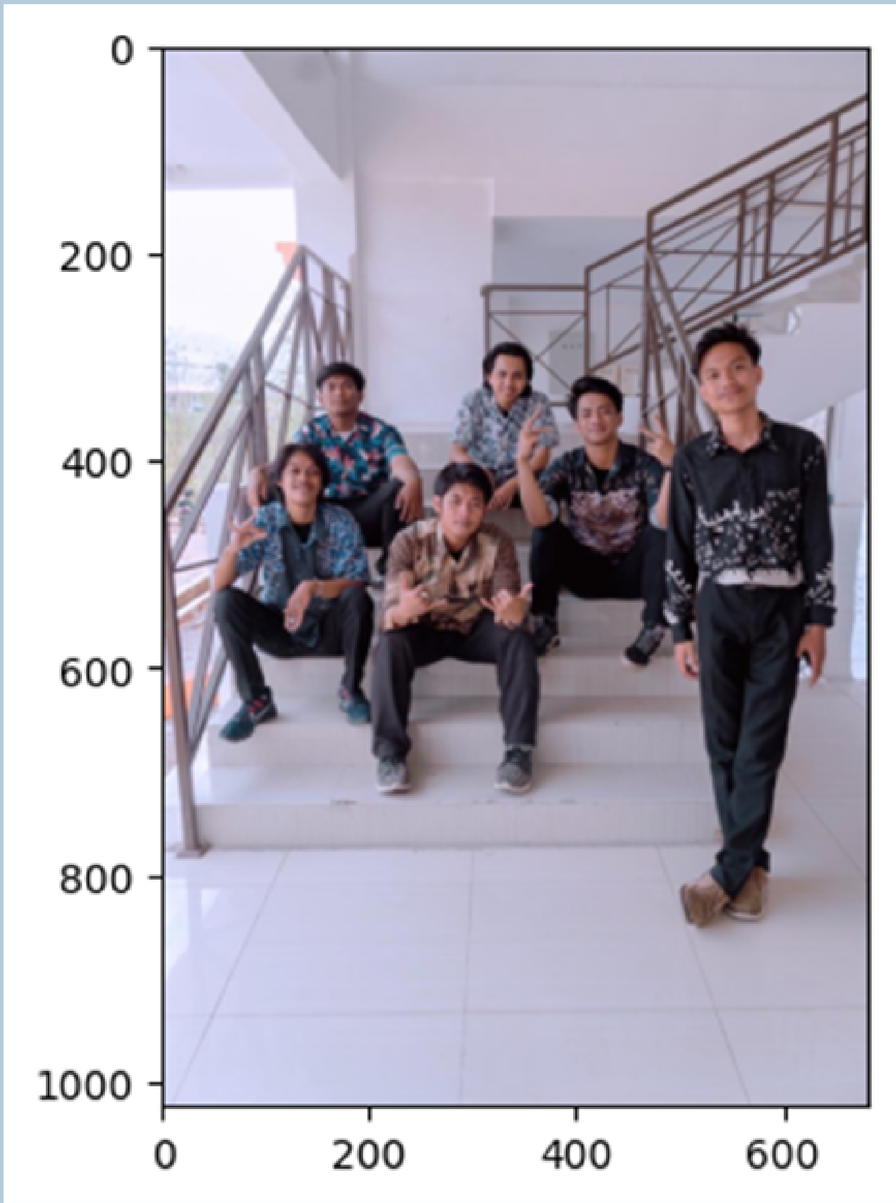
```
[[[ 56  56  44]
  [ 56  56  44]
  [ 59  59  47]
  ...
  [184 183 188]
  [184 183 188]
  [184 183 188]]

 [[ 61  61  49]
  [ 60  60  48]
  [ 60  60  48]
  ...
  [184 183 188]
  [184 183 188]
  [184 183 188]]

 [[ 67  67  55]
  [ 66  66  54]
  [ 64  64  52]
  ...
  [184 183 188]
  [184 183 188]
  [184 183 188]]

 ...
```

```
[[ 62  64  53]
  [ 60  62  51]
  [ 58  60  49]
  ...
  [ 42  43  37]
  [ 41  42  36]
  [ 42  43  37]]

 [[ 57  59  48]
  [ 55  57  46]
  [ 53  55  44]
  ...
  [ 44  45  39]
  [ 43  44  38]
  [ 43  44  38]]

 [[ 52  54  43]
  [ 51  53  42]
  [ 49  51  40]
  ...
  [ 45  46  40]
  [ 44  45  39]
  [ 43  44  38]]]
```



```
[[[207 208 238]
  [207 208 238]
  [207 208 238]
  ...
  [172 171 187]
  [172 171 187]
  [172 171 187]]

 [[207 208 238]
  [207 208 238]
  [207 208 238]
  ...
  [172 171 187]
  [172 171 187]
  [172 171 187]]

 [[207 208 238]
  [207 208 238]
  [207 208 238]
  ...
  [171 170 186]
  [171 170 186]
  [171 170 186]]
```

```
      [[209 212 243]
       [209 212 243]
       [209 212 243]
       ...
       [183 181 202]
       [183 181 202]
       [183 181 202]]

      [[209 212 243]
       [209 212 243]
       [209 212 243]
       ...
       [183 181 202]
       [183 181 202]
       [183 181 202]]

      [[209 212 243]
       [209 212 243]
       [209 212 243]
       ...
       [183 181 202]
       [183 181 202]
       [183 181 202]]]
```



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 tersebut mengimplementasikan metode Lightness untuk mengubah tingkat kecerahan gambar. Metode ini menghitung nilai kecerahan baru untuk setiap piksel dengan mengambil rata-rata tertimbang dari nilai maksimum dan minimum saluran warna (RGB) pada piksel tersebut. Hasilnya ditampilkan dan disimpan sebagai file dengan nama "Metode Lightness".

```
[[[140 140 140]
  [140 140 140]
  [140 140 140]
  ...
  [110 110 110]
  [110 110 110]
  [110 110 110]]

 [[140 140 140]
  [140 140 140]
  [140 140 140]
  ...
  [110 110 110]
  [110 110 110]
  [110 110 110]]

 [[139 139 139]
  [139 139 139]
  [139 139 139]
  ...
  [110 110 110]
  [110 110 110]
  [110 110 110]]

 ...
```

```
[[ 98  98  98]
 [ 70  70  70]
 [ 31  31  31]
 ...
 [ 67  67  67]
 [ 67  67  67]
 [ 67  67  67]]

[[ 97  97  97]
 [ 76  76  76]
 [ 43  43  43]
 ...
 [ 69  69  69]
 [ 69  69  69]
 [ 69  69  69]]

[[ 96  96  96]
 [ 84  84  84]
 [ 59  59  59]
 ...
 [ 70  70  70]
 [ 70  70  70]
 [ 70  70  70]]]
```

```
[[[ 50  50  50]
  [ 50  50  50]
  [ 53  53  53]
  ...
  [185 185 185]
  [185 185 185]
  [185 185 185]]

 [[ 55  55  55]
  [ 54  54  54]
  [ 54  54  54]
  ...
  [185 185 185]
  [185 185 185]
  [185 185 185]]

 [[ 61  61  61]
  [ 60  60  60]
  [ 58  58  58]
  ...
  [185 185 185]
  [185 185 185]
  [185 185 185]]


 ...
```


```
[[[ 50  50  50]
  [ 50  50  50]
  [ 53  53  53]
  ...
  [185 185 185]
  [185 185 185]
  [185 185 185]]

 [[ 55  55  55]
  [ 54  54  54]
  [ 54  54  54]
  ...
  [185 185 185]
  [185 185 185]
  [185 185 185]]

 [[ 61  61  61]
  [ 60  60  60]
  [ 58  58  58]
  ...
  [185 185 185]
  [185 185 185]
  [185 185 185]]


 ...
```


```
  ...
  [191 191 191]
  [191 191 191]
  [191 191 191]]

 [[226 226 226]
  [226 226 226]
  [226 226 226]
  ...
  [191 191 191]
  [191 191 191]
  [191 191 191]]]
[[222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 178. 178. 178.]
 ...
 [226. 226. 226. ... 191. 191. 191.]
 [226. 226. 226. ... 191. 191. 191.]
 [226. 226. 226. ... 191. 191. 191.]]
[[222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 178. 178. 178.]
 ...
 [226. 226. 226. ... 191. 191. 191.]
 [226. 226. 226. ... 191. 191. 191.]
 [226. 226. 226. ... 191. 191. 191.]]
[[222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 178. 178. 178.]
```

METODE AVERAGE

```
gray_img = np.mean(fix_img, axis=-1)
print (np.array(gray_img))

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

Kode tersebut mengimplementasikan metode Average untuk mengubah gambar menjadi citra grayscale, menampilkannya, dan menyimpannya dengan nama "Metode Average".

```
[[140. 140. 140. ... 110. 110. 110.]
 [140. 140. 140. ... 110. 110. 110.]
 [139. 139. 139. ... 110. 110. 110.]
 ...
 [ 98.  70.  31. ...  67.  67.  67.]
 [ 97.  76.  43. ...  69.  69.  69.]
 [ 96.  84.  59. ...  70.  70.  70.]]
```

```
[[ 50.  50.  53. ... 185. 185. 185.]
 [ 55.  54.  54. ... 185. 185. 185.]
 [ 61.  60.  58. ... 185. 185. 185.]
 ...
 [ 58.  56.  54. ...  40.  39.  40.]
 [ 53.  51.  49. ...  42.  41.  41.]
 [ 48.  47.  45. ...  43.  42.  41.]]
```

```
[[222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 178. 178. 178.]
 ...
 [226. 226. 226. ... 191. 191. 191.]
 [226. 226. 226. ... 191. 191. 191.]
 [226. 226. 226. ... 191. 191. 191.]]
```

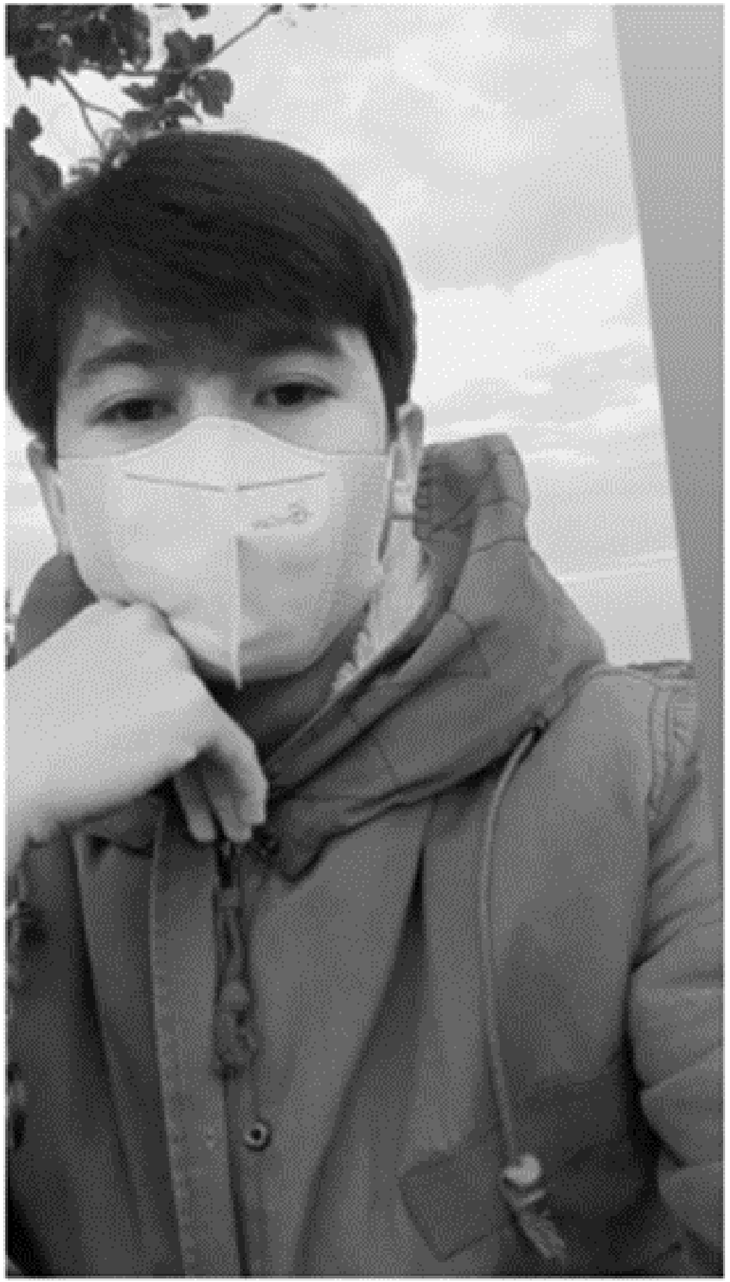
METODE LUMINOSITY

```
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 tersebut menggunakan rumus Luminosity untuk mengonversi gambar ke citra grayscale dengan mempertimbangkan bobot dari setiap saluran warna (R, G, B). Formula yang digunakan adalah (0.2126 * R) + (0.7152 * G) + (0.0722 * B), menggambarkan sensitivitas mata manusia terhadap warna.

```
[[140. 140. 140. ... 110. 110. 110.]
 [140. 140. 140. ... 110. 110. 110.]
 [139. 139. 139. ... 110. 110. 110.]
 ...
 [ 98.  70.  31. ...  67.  67.  67.]
 [ 97.  76.  43. ...  69.  69.  69.]
 [ 96.  84.  59. ...  70.  70.  70.]]
```

```
[[ 50.  50.  53. ... 185. 185. 185.]
 [ 55.  54.  54. ... 185. 185. 185.]
 [ 61.  60.  58. ... 185. 185. 185.]
 ...
 [ 58.  56.  54. ...  40.  39.  40.]
 [ 53.  51.  49. ...  42.  41.  41.]
 [ 48.  47.  45. ...  43.  42.  41.]]
```



```
[[222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 178. 178. 178.]
 ...
 [226. 226. 226. ... 191. 191. 191.]
 [226. 226. 226. ... 191. 191. 191.]
 [226. 226. 226. ... 191. 191. 191.]]
```

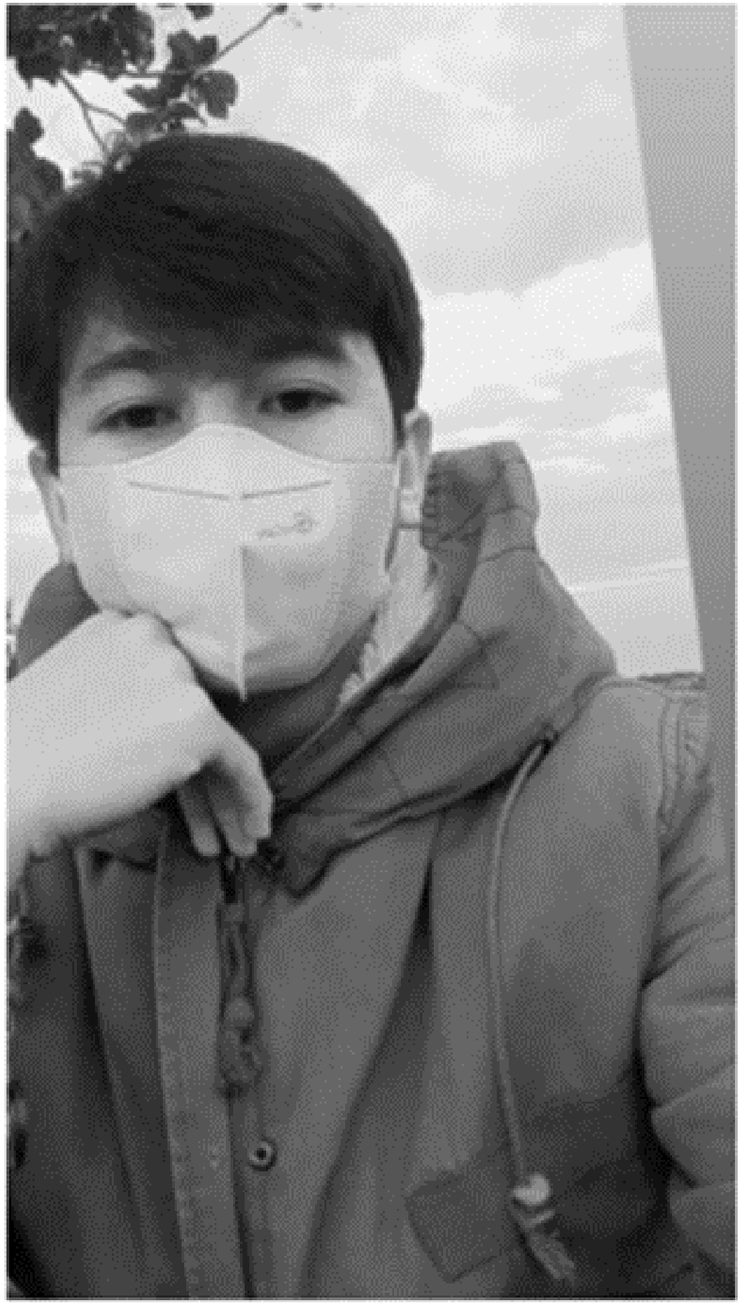
METODE WEIGHTED AVERAGE

```
wav_img = (0.2999*R) + (0.587*G) + (0.114*B)
print(np.array(wav_img))

plt.axis('off')
plt.imshow(wav_img, cmap='gray')
plt.savefig('Metode Weight Average', bbox_inches= 'tight')
```

Kode tersebut menerapkan metode Weighted Average untuk mengonversi gambar ke citra grayscale. Bobot khusus diterapkan pada saluran warna (R, G, B), dan rumus yang digunakan adalah (0.299 * R) + (0.587 * G) + (0.114 * B). Hasil citra grayscale ditampilkan dan disimpan sebagai "Metode Weighted Average".

```
[[140.126  140.126  140.126  ... 110.099  110.099  110.099 ]
 [140.126  140.126  140.126  ... 110.099  110.099  110.099 ]
 [139.1251 139.1251 139.1251 ... 110.099  110.099  110.099 ]
 ...
 [ 98.0882  70.063   31.0279 ...  67.0603  67.0603  67.0603]
 [ 97.0873  76.0684  43.0387 ...  69.0621  69.0621  69.0621]
 [ 96.0864  84.0756  59.0531 ...  70.063   70.063   70.063 ]]
```

```
[[ 50.  50.  53. ... 185. 185. 185.]
 [ 55.  54.  54. ... 185. 185. 185.]
 [ 61.  60.  58. ... 185. 185. 185.]
 ...
 [ 58.  56.  54. ...  40.  39.  40.]
 [ 53.  51.  49. ...  42.  41.  41.]
 [ 48.  47.  45. ...  43.  42.  41.]]
```



```
[[222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 179. 179. 179.]
 [222. 222. 222. ... 178. 178. 178.]
 ...
 [226. 226. 226. ... 191. 191. 191.]
 [226. 226. 226. ... 191. 191. 191.]
 [226. 226. 226. ... 191. 191. 191.]]
```

KESIMPULAN

Metode Average lebih mudah karena hanya menggunakan rata-rata nilai piksel dalam saluran warna (R, G, B) untuk menghasilkan citra grayscale. Tidak ada perhitungan bobot atau pertimbangan sensitivitas mata manusia terhadap warna yang rumit.