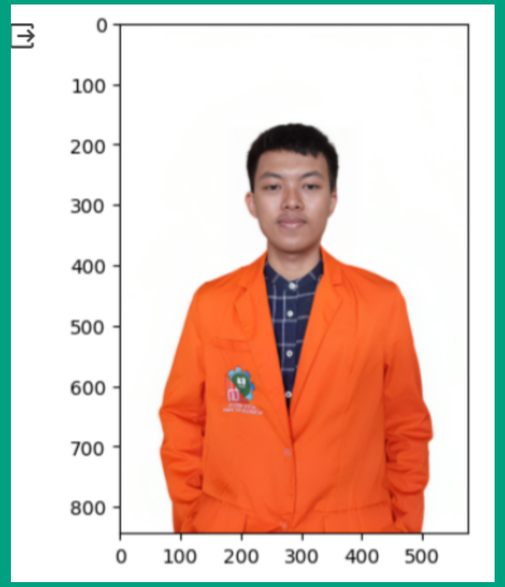
ALJABAR LINEAR

PROYEK AKHIR

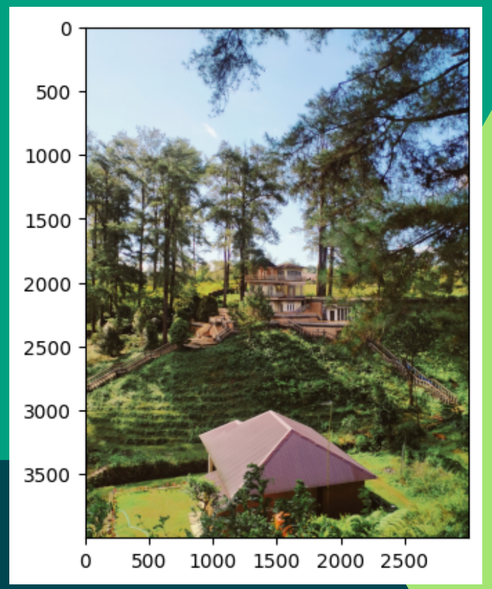
Kelompok :

1. Andri (221011104)
2. Arif Hidayat (221011023)
3. Muhammad Reza Fahlevi Syahrul (221011061)

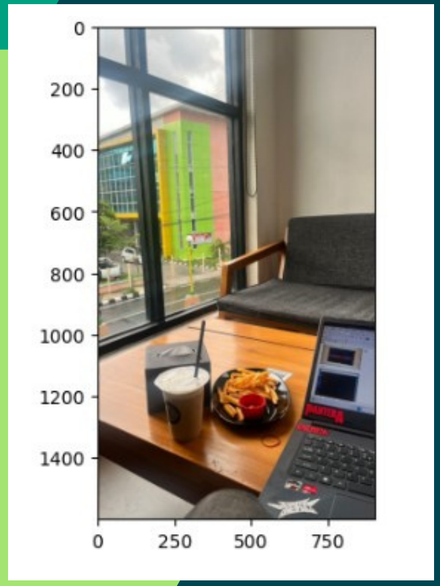



```
(844, 576, 3)
[[[254 255 253]
  [254 255 253]
  [254 255 254]
  ...
  [253 254 253]
  [253 254 253]
  [253 254 253]]

 [[254 255 253]
  [254 255 253]
  [254 255 254]
  ...
  [253 254 253]
  [253 254 253]
  [253 254 253]]

 [[254 255 254]
  [254 255 254]
  [254 255 254]
  ...
  [253 254 253]
  [253 254 253]
  [253 254 253]]
```

```python
from io import IncrementalNewlineDecoder
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

img_path = 'arif.jpg'
img = cv2.imread(img_path)
print(img.shape)

fix_img
R, G, B = fix_img[:,:,0], fix_img[:,:,1], fix_img[:,:,2]
print(np.array(fix_img))

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

print(np.array(fix_img))
```




```
 [210 229 243]
 [210 229 243]
 ...
 [ 43  87 150]
 [129 192 235]
 [120 193 225]]

[[210 229 243]
 [211 230 244]
 [211 230 244]
 ...
 [ 56 104 166]
 [122 189 231]
 [115 191 223]]

[[207 229 242]
 [210 232 245]
 [211 230 244]
 ...
 [ 73 132 188]
```




```
(1600, 900, 3)
[[[230 234 237]
  [229 233 236]
  [229 233 236]
  ...
  [179 171 158]
  [180 172 159]
  [180 172 159]]

 [[230 234 237]
  [229 233 236]
  [229 233 236]
  ...
  [179 171 158]
  [180 172 159]
  [181 173 160]]

 [[230 234 237]
  [229 233 236]
  [229 233 236]
  ...
  [179 171 158]
  [180 172 159]
  [181 173 160]]
```

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')
```

```
[[[254 254 254]
  [254 254 254]
  [254 254 254]
  ...
  [253 253 253]
  [253 253 253]
  [253 253 253]]

 [[254 254 254]
  [254 254 254]
  [254 254 254]
  ...
  [253 253 253]
  [253 253 253]
  [253 253 253]]

 [[254 254 254]
  [254 254 254]
  [254 254 254]
  ...
  [253 253 253]
  [253 253 253]
  [253 253 253]]

 ...
```

```
[[[233 233 233]
  [232 232 232]
  [232 232 232]
  ...
  [168 168 168]
  [169 169 169]
  [169 169 169]]

 [[233 233 233]
  [232 232 232]
  [232 232 232]
  ...
  [168 168 168]
  [169 169 169]
  [170 170 170]]

 [[233 233 233]
  [232 232 232]
  [232 232 232]
  ...
  [168 168 168]
  [169 169 169]
  [170 170 170]]
```

```
[[[227 227 227]
  [226 226 226]
  [226 226 226]
  ...
  [ 96  96  96]
  [182 182 182]
  [172 172 172]]

 [[226 226 226]
  [227 227 227]
  [227 227 227]
  ...
  [111 111 111]
  [176 176 176]
  [169 169 169]]

 [[224 224 224]
  [227 227 227]
  [227 227 227]
  ...
  [130 130 130]
  [175 175 175]
  [162 162 162]]
```

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', bbox_inches='tight')
```

```
[[254. 254. 254. ... 253. 253. 253.]
 [254. 254. 254. ... 253. 253. 253.]
 [254. 254. 254. ... 253. 253. 253.]
 ...
 [255. 254. 254. ... 254. 254. 254.]
 [254. 254. 254. ... 254. 254. 254.]
 [254. 254. 254. ... 253. 253. 253.]]
```

```
[[233. 232. 232. ... 168. 169. 169.]
 [233. 232. 232. ... 168. 169. 170.]
 [233. 232. 232. ... 168. 169. 170.]
 ...
 [181. 180. 179. ...  73.  72.  71.]
 [180. 180. 180. ...  71.  71.  71.]
 [179. 180. 181. ...  69.  70.  71.]]
```

```
[[227. 226. 226. ...  96. 182. 172.]
 [226. 227. 227. ... 111. 176. 169.]
 [224. 227. 227. ... 130. 175. 162.]
 ...
 [ 39.  38.  40. ...  24.  24.  24.]
 [ 34.  39.  38. ...  21.  23.  26.]
 [ 32.  35.  37. ...  17.  20.  26.]]
```

Metode Luminosity (1)

```
[[254. 254. 254. ... 253. 253. 253.]
 [254. 254. 254. ... 253. 253. 253.]
 [254. 254. 254. ... 253. 253. 253.]
 ...
 [255. 254. 254. ... 254. 254. 254.]
 [254. 254. 254. ... 254. 254. 254.]
 [254. 254. 254. ... 253. 253. 253.]]
```

```
[[233. 232. 232. ... 168. 169. 169.]
 [233. 232. 232. ... 168. 169. 170.]
 [233. 232. 232. ... 168. 169. 170.]
 ...
 [181. 180. 179. ...  73.  72.  71.]
 [180. 180. 180. ...  71.  71.  71.]
 [179. 180. 181. ...  69.  70.  71.]]
```

```
[[226.9714 225.9714 225.9714 ...  82.1942 181.7108 179.7906]
 [225.9714 226.9714 226.9714 ...  98.2716 177.7882 177.1528]
 [225.2614 228.2614 226.9714 ... 123.4998 179.943  171.5964]
 ...
 [ 48.2146  48.5728  50.5728 ...  31.4878  31.1348  31.1348]
 [ 43.069   47.359   47.7172 ...  27.6322  30.1348  32.9222]
 [ 40.1412  43.359   45.359  ...  22.564   27.1348  32.9222]]
```

```
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')
```

Metode Luminosity (2)

```
[[[254 254 254]
  [254 254 254]
  [254 254 254]
  ...
  [253 253 253]
  [253 253 253]
  [253 253 253]]

 [[254 254 254]
  [254 254 254]
  [254 254 254]
  ...
  [253 253 253]
  [253 253 253]
  [253 253 253]]

 [[254 254 254]
  [254 254 254]
  [254 254 254]
  ...
  [253 253 253]
  [253 253 253]
  [253 253 253]]

 ...
```

```
[[226.9714 225.9714 225.9714 ...  82.1942 181.7108 179.7906]
 [225.9714 226.9714 226.9714 ...  98.2716 177.7882 177.1528]
 [225.2614 228.2614 226.9714 ... 123.4998 179.943  171.5964]
 ...
 [ 48.2146  48.5728  50.5728 ...  31.4878  31.1348  31.1348]
 [ 43.069   47.359   47.7172 ...  27.6322  30.1348  32.9222]
 [ 40.1412  43.359   45.359  ...  22.564   27.1348  32.9222]]
```

```
[[225.915 224.915 224.915 ...  81.026 178.065
 [224.915 225.915 225.915 ...  96.716 173.755
 [223.904 226.904 225.915 ... 120.743 175.135
 ...
 [ 45.921  45.981  47.981 ...  30.649  30.165
 [ 41.16   45.149  45.209 ...  26.877  29.165
 [ 38.274  41.149  43.149 ...  21.806  26.165
```

```
wav_img = (0.299*R) + (0.587*G) + (0.114*B)
print(np.array(wav_img))

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

Menurut kami, metode Luminosity adalah metode yang bagus.

Jawab :

Karena mencoba lebih akurat mencerminkan persepsi mata manusia terhadap warna.

memberikan bobot yang berbeda untuk masing-masing saluran warna (merah, hijau, dan biru) berdasarkan pada sensitivitas mata manusia terhadap warna.

telah diadopsi sebagai standar dalam sejumlah aplikasi dan industri

Kelemahan

Tidak selalu sempurna dan tergantung pada konteks aplikasi tertentu.

Kesimpulan :

1. Konversi citra RGB ke grayscale adalah proses mengubah citra berwarna menjadi citra dengan skala keabuan.
2. Metode Luminosity : memberikan bobot berbeda pada setiap saluran warna (merah, hijau, biru).
3. Metode Lightness : menghitung rata-rata dari nilai piksel maksimum dan minimum di antara saluran warna
4. Metode Average : mengambil rata-rata nilai piksel dari semua saluran warna