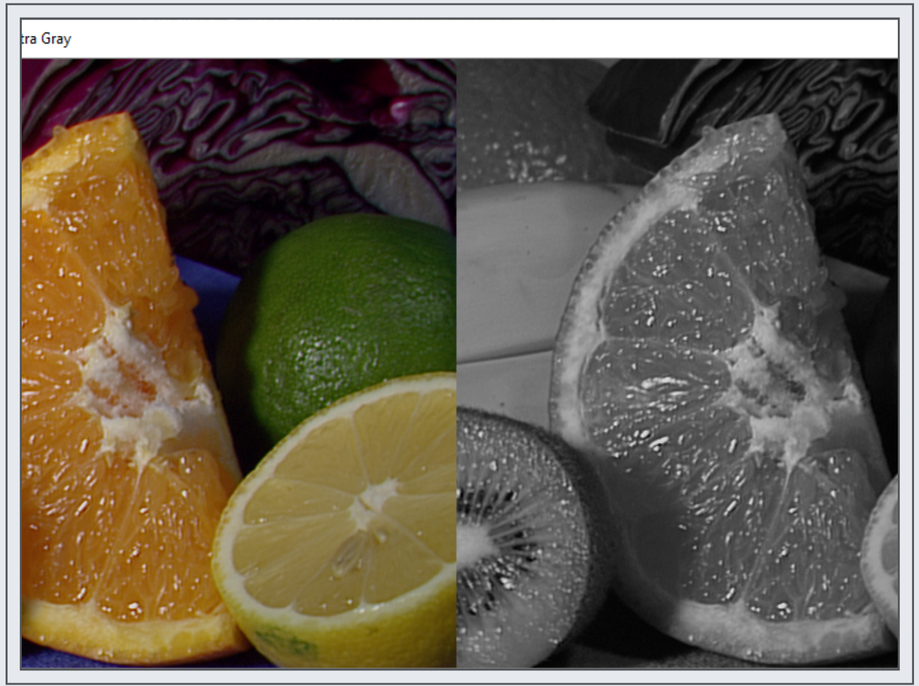
Presentasi **PROYEK AKHIR**

Oleh Kelompok 3:

1. **Muhammad Faruq Al-Fauzi S. (221011015)**
2. **Muhammad Ishak (221011073)**
3. **Nurul Meyti Dea Putri (221011089)**

Mata Kuliah Aljabar Linear | Institut Teknologi BJ Habibie

Konversi Citra RGB ke Grayscale



Pendahuluan

Citra adalah representasi visual dari sesuatu. Citra menggambarkan rupa atau gambaran dari suatu objek. Singkatnya citra adalah gambar!

Citra adalah fungsi dari intensitas cahaya yang dipresentasikan dalam bidang 2 dimensi. Citra dibedakan menjadi 3 jenis, yaitu citra RGB, citra grayscale, dan citra biner. Citra RGB tersusun oleh 3 warna yaitu merah, hijau, dan biru. Citra grayscale adalah citra yang nilai intensitasnya berdasarkan derajat keabuan.

Metode Konversi

Lightness

Mencari nilai tertinggi dan terendah dari nilai R, G, dan B, kemudian nilai tersebut saling ditambahkan dan hasilnya dibagi dengan 2.

$$Grayscale = (max(R, G, B)) + (min(R, G, B)) / 2$$

Average

Mencari nilai rata-rata dari R, G, dan B.

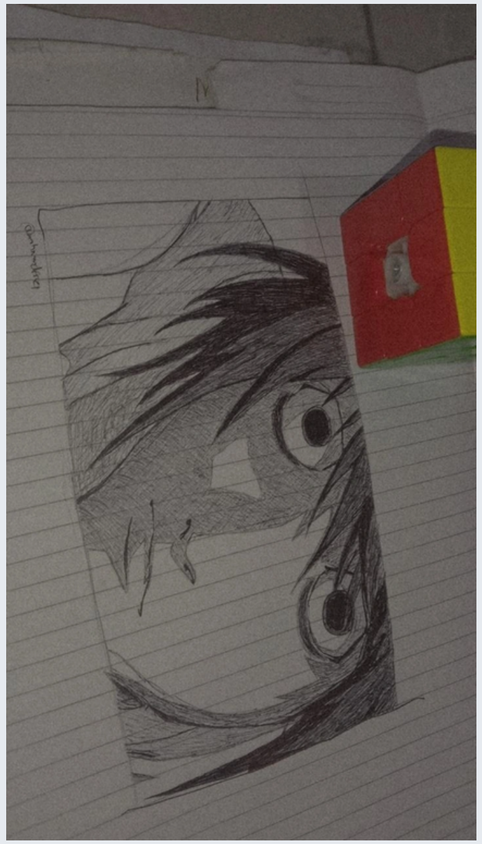
$$Grayscale = (R + G + B)/3$$

Luminosity

Mengalikan setiap nilai R, G, dan B dengan konstanta tertentu yang sudah ditetapkan nilainya, kemudian hasil perkalian semua nilai R, G, dan B dijumlahkan satu sama lain

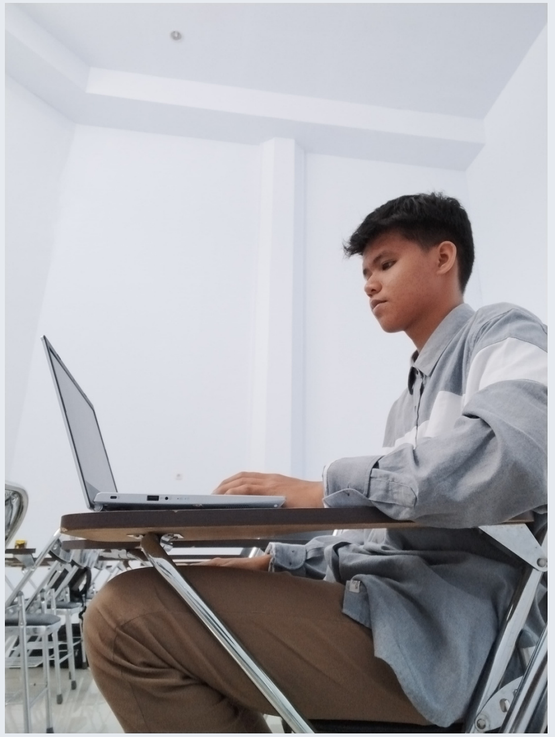
$$Grayscale = (0.2126 \times R) + (0.7152 \times G) + (0.0722 \times B)$$
$$Grayscale = (0.299 \times R) + (0.587 \times G) + (0.114 \times B)$$

Gambar



MUHAMMADFARUQALFAUZI.jpg

NurulMeytiDeaPutri.jpg



Ishak.jpg

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

img_path = "MUHAMMADFARUQALFAUZI.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 digunakan untuk mengonversi warna dari format BGR ke RGB lalu menampilkan matriks piksel yang mewakili kode warna dari gambar yang diunggah.

Output Matriks

```
(1280, 720, 3)
[[[ 87  83  82]
  [ 88  84  83]
  [ 87  83  82]
```

MUHAMMADFARUQALFAUZI.jpg

1280 = tinggi gambar
720 = lebar gambar
3 = jumlah saluran warna

Nilai matriks merepresentasikan pixel warna
R bernilai 87 dan 88
G bernilai 83 dan 84
B bernilai 82 dan 83

```
(4160, 3084, 3)
[[[210 211 215]
  [210 211 215]
  [210 211 215]
```

Ishak.jpg

4160= tinggi gambar
3084 = lebar gambar
3 = jumlah saluran warna

Nilai matriks merepresentasikan pixel warna
R bernilai 210
G bernilai 211
B bernilai 215

```
(1600, 900, 3)
[[[171 158 141]
  [173 160 143]
  [180 167 148]
```

NurulMeytiDeaPutri.jpg

1600 = tinggi gambar
900 = lebar gambar
3 = jumlah saluran warna

Nilai matriks merepresentasikan pixel warna
R bernilai 171, 173, dan 180
G bernilai 171, 173, dan 180
B bernilai 141, 153, dan 148

Metode Lightness

```
# 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')
```

Loading...

MUHAMMADFARUQALFAUZI.jpg

```
[[[ 84  84  84]
  [ 85  85  85]
  [ 84  84  84]
```

NurulMeytiDeaPutri.jpg

```
[[[156 156 156]
  [158 158 158]
  [164 164 164]
```

Ishak.jpg

```
[[[212 212 212]
  [212 212 212]
  [212 212 212]
```

Bagaimana?

```
(1280, 720, 3)
[[[ 87  83  82]
  [ 88  84  83]
  [ 87  83  82]
```

```
[[[ 84  84  84]
  [ 85  85  85]
  [ 84  84  84]
```

(Nilai max(RGB) + nilai minimal(RGB))/2

(87 + 82) / 2 = 84.5

```
(4160, 3084, 3)
[[[210 211 215]
  [210 211 215]
  [210 211 215]
```

```
[[[212 212 212]
  [212 212 212]
  [212 212 212]
```

(Nilai max(RGB) + nilai minimal(RGB)) /2

(215 + 210) / 2 = 212.2

```
(1600, 900, 3)
[[[171 158 141]
  [173 160 143]
  [180 167 148]
```

```
[[[156 156 156]
  [158 158 158]
  [164 164 164]
```

(Nilai max(RGB) + nilai minimal(RGB))/2

(171 + 141) / 2 = 156

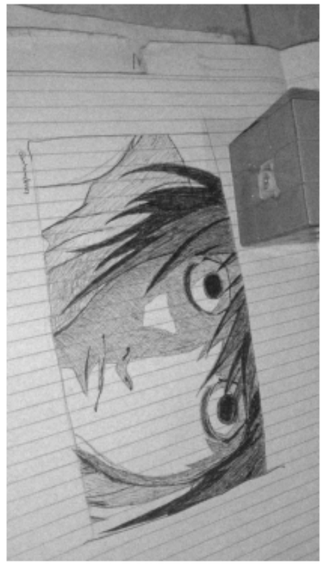
Metode Average

```
# 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')
```

MUHAMMADFARUQALFAUZI.jpg



```
[[ 84.  85.  84. ...  62.  60.  60.]
 [ 85.  88.  86. ...  59.  56.  57.]
 [ 86.  90.  91. ...  57.  57.  59.]
 ...
 [102. 106. 111. ...  89.  90.  92.]
 [104. 106. 109. ...  93.  90.  86.]
 [109. 106. 106. ...  96.  90.  82.]]
```

NurulMeytiDeaPutri.jpg

```
[[156. 158. 164. ... 158. 166. 170.]
 [164. 168. 175. ... 143. 151. 156.]
 [170. 178. 192. ... 134. 141. 146.]
 ...
 [109. 120. 115. ...  29.  28.  27.]
 [111. 123. 116. ...  26.  25.  25.]
 [100. 110. 105. ...  30.  29.  29.]]
```

Ishak.jpg

```
[[212. 212. 212. ... 198. 198. 198.]
 [212. 212. 212. ... 198. 198. 198.]
 [212. 212. 212. ... 198. 198. 198.]
 ...
 [176. 176. 176. ...  38.  38.  38.]
 [175. 175. 176. ...  38.  38.  39.]
 [175. 175. 175. ...  38.  39.  39.]]
```

Bagaimana?

```
(1280, 720, 3)
[[[ 87  83  82]
  [ 88  84  83]
  [ 87  83  82]
```

```
[[ 84.  85.  84. ...  62.  60.  60.]
 [ 85.  88.  86. ...  59.  56.  57.]
 [ 86.  90.  91. ...  57.  57.  59.]
 ...
 [102. 106. 111. ...  89.  90.  92.]
 [104. 106. 109. ...  93.  90.  86.]
 [109. 106. 106. ...  96.  90.  82.]]
```

Rata-rata suatu baris
(87 + 83 + 82) / 3 = 84

```
(4160, 3084, 3)
[[[210 211 215]
  [210 211 215]
  [210 211 215]
```

```
[[212. 212. 212. ... 198. 198. 198.]
 [212. 212. 212. ... 198. 198. 198.]
 [212. 212. 212. ... 198. 198. 198.]
 ...
 [176. 176. 176. ...  38.  38.  38.]
 [175. 175. 176. ...  38.  38.  39.]
 [175. 175. 175. ...  38.  39.  39.]]
```

Rata-rata suatu baris
(210 + 211 + 215) / 3 = 212

```
(1600, 900, 3)
[[[171 158 141]
  [173 160 143]
  [180 167 148]
```

```
[[156. 158. 164. ... 158. 166. 170.]
 [164. 168. 175. ... 143. 151. 156.]
 [170. 178. 192. ... 134. 141. 146.]
 ...
 [109. 120. 115. ...  29.  28.  27.]
 [111. 123. 116. ...  26.  25.  25.]
 [100. 110. 105. ...  30.  29.  29.]]
```

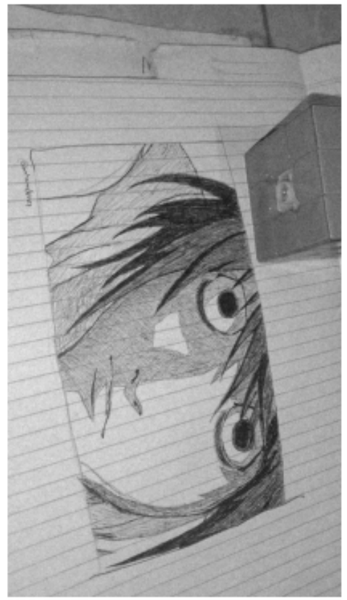
Rata-rata suatu baris
(171 + 158 + 141) / 2 = 156.666

Metode Luminosity

```
# metode luminosity
lumi_img = (0.2126*R) + (0.7152*G) + (0.0722*B)
# print(lumi_img)
print(np.array(lumi_img))

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

MUHAMMADFARUQALFAUZI.jpg



```
[[ 84.  85.  84. ...  62.  60.  60.]
 [ 85.  88.  86. ...  59.  56.  57.]
 [ 86.  90.  91. ...  57.  57.  59.]
 ...
 [102. 106. 111. ...  89.  90.  92.]
 [104. 106. 109. ...  93.  90.  86.]
 [109. 106. 106. ...  96.  90.  82.]]
```

NurulMeytiDeaPutri.jpg

```
[[156. 158. 164. ... 158. 166. 170.]
 [164. 168. 175. ... 143. 151. 156.]
 [170. 178. 192. ... 134. 141. 146.]
 ...
 [109. 120. 115. ...  29.  28.  27.]
 [111. 123. 116. ...  26.  25.  25.]
 [100. 110. 105. ...  30.  29.  29.]]
```

Ishak.jpg

```
[[212. 212. 212. ... 198. 198. 198.]
 [212. 212. 212. ... 198. 198. 198.]
 [212. 212. 212. ... 198. 198. 198.]
 ...
 [176. 176. 176. ...  38.  38.  38.]
 [175. 175. 176. ...  38.  38.  39.]
 [175. 175. 175. ...  38.  39.  39.]]
```

Bagaimana?

```
(1280, 720, 3)
[[[ 87  83  82]
  [ 88  84  83]
  [ 87  83  82]
```

```
[[ 84.  85.  84. ...  62.  60.  60.]
 [ 85.  88.  86. ...  59.  56.  57.]
 [ 86.  90.  91. ...  57.  57.  59.]
 ...
 [102. 106. 111. ...  89.  90.  92.]
 [104. 106. 109. ...  93.  90.  86.]
 [109. 106. 106. ...  96.  90.  82.]]
```

Pengalian dengan konstanta tertentu (0.2126 × R) + (0.7152 × G) + (0.0722 × B)

(0.2126 × 87) + (0.7152 × 83) + (0.0722 × 82) = 83.9

```
(4160, 3084, 3)
[[[210 211 215]
  [210 211 215]
  [210 211 215]
```

```
[[212. 212. 212. ... 198. 198. 198.]
 [212. 212. 212. ... 198. 198. 198.]
 [212. 212. 212. ... 198. 198. 198.]
 ...
 [176. 176. 176. ...  38.  38.  38.]
 [175. 175. 176. ...  38.  38.  39.]
 [175. 175. 175. ...  38.  39.  39.]]
```

Pengalian dengan konstanta tertentu (0.2126 × R) + (0.7152 × G) + (0.0722 × B)

(0.2126 × 210) + (0.7152 × 211) + (0.0722 × 215) = 211.0762

```
(1600, 900, 3)
[[[171 158 141]
  [173 160 143]
  [180 167 148]
```

```
[[156. 158. 164. ... 158. 166. 170.]
 [164. 168. 175. ... 143. 151. 156.]
 [170. 178. 192. ... 134. 141. 146.]
 ...
 [109. 120. 115. ...  29.  28.  27.]
 [111. 123. 116. ...  26.  25.  25.]
 [100. 110. 105. ...  30.  29.  29.]]
```

Pengalian dengan konstanta tertentu (0.2126 × R) + (0.7152 × G) + (0.0722 × B)

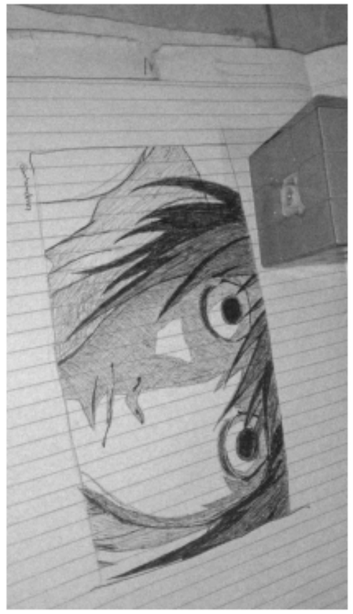
(0.2126 × 171) + (0.7152 × 158) + (0.0722 × 141) = 159.5364

Metode Luminosity 2

```
# metode luminosity 2
wav_img = (0.299*R) + (0.587*G) + (0.114*B)
# print(lumi_img)
print(np.array(wav_img))

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

MUHAMMADFARUQALFAUZI.jpg



```
[[ 84.  85.  84. ...  62.  60.  60.]
 [ 85.  88.  86. ...  59.  56.  57.]
 [ 86.  90.  91. ...  57.  57.  59.]
 ...
 [102. 106. 111. ...  89.  90.  92.]
 [104. 106. 109. ...  93.  90.  86.]
 [109. 106. 106. ...  96.  90.  82.]]
```

NurulMeytiDeaPutri.jpg

```
[[156. 158. 164. ... 158. 166. 170.]
 [164. 168. 175. ... 143. 151. 156.]
 [170. 178. 192. ... 134. 141. 146.]
 ...
 [109. 120. 115. ...  29.  28.  27.]
 [111. 123. 116. ...  26.  25.  25.]
 [100. 110. 105. ...  30.  29.  29.]]
```

Ishak.jpg

```
[[212. 212. 212. ... 198. 198. 198.]
 [212. 212. 212. ... 198. 198. 198.]
 [212. 212. 212. ... 198. 198. 198.]
 ...
 [176. 176. 176. ...  38.  38.  38.]
 [175. 175. 176. ...  38.  38.  39.]
 [175. 175. 175. ...  38.  39.  39.]]
```

Kesimpulan

Metode konversi yang menurut kami menghasilkan citra paling bagus adalah metode average, alasannya adalah karena warna yang dihasilkan tidak terlalu terang dan tidak terlalu gelap.

Terima Kasih!