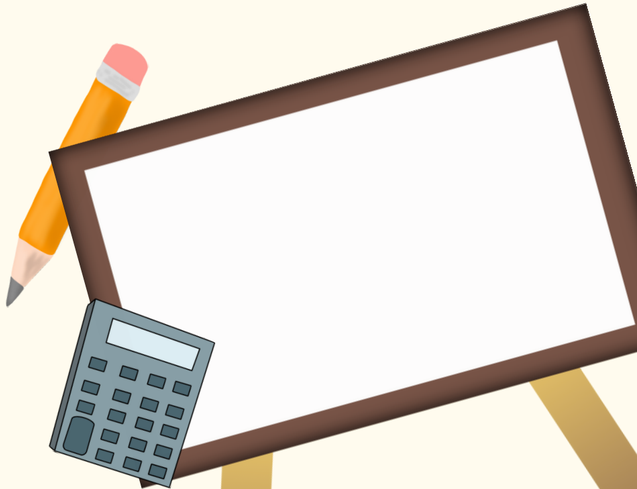
konversi citra RGB ke Grayscale

Tugas Presentasi

ALJABAR LINEAR



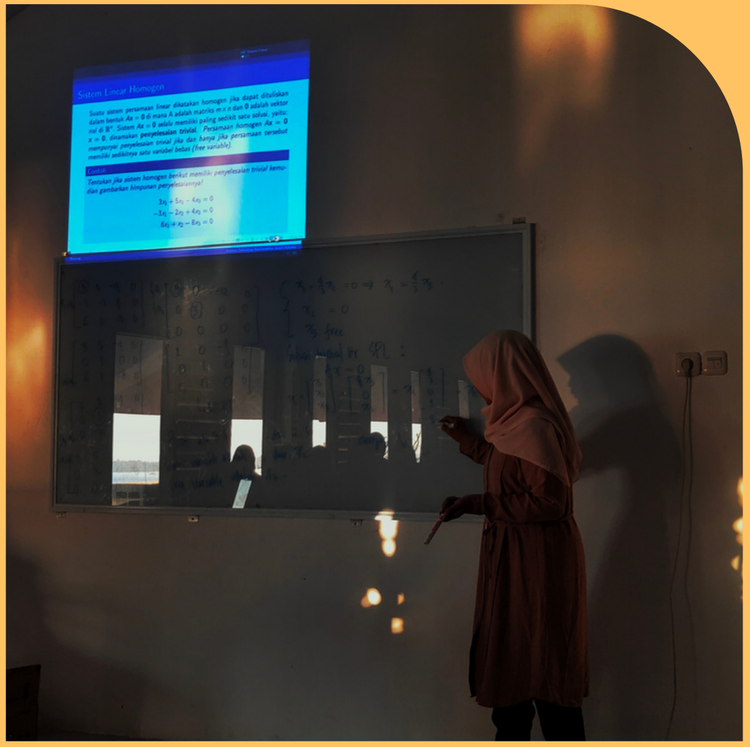
Kelompok 9

ISMAIL

NABILAH PUTRI ALI

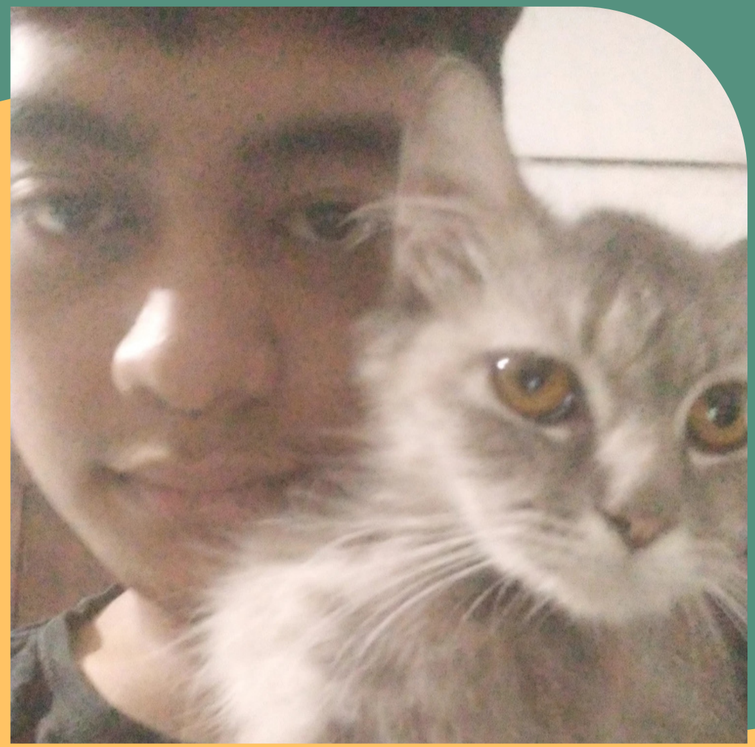
MOHAMAD BINTANG ZAKY ZHAFRAN

ismail.jpg



lala.jpg

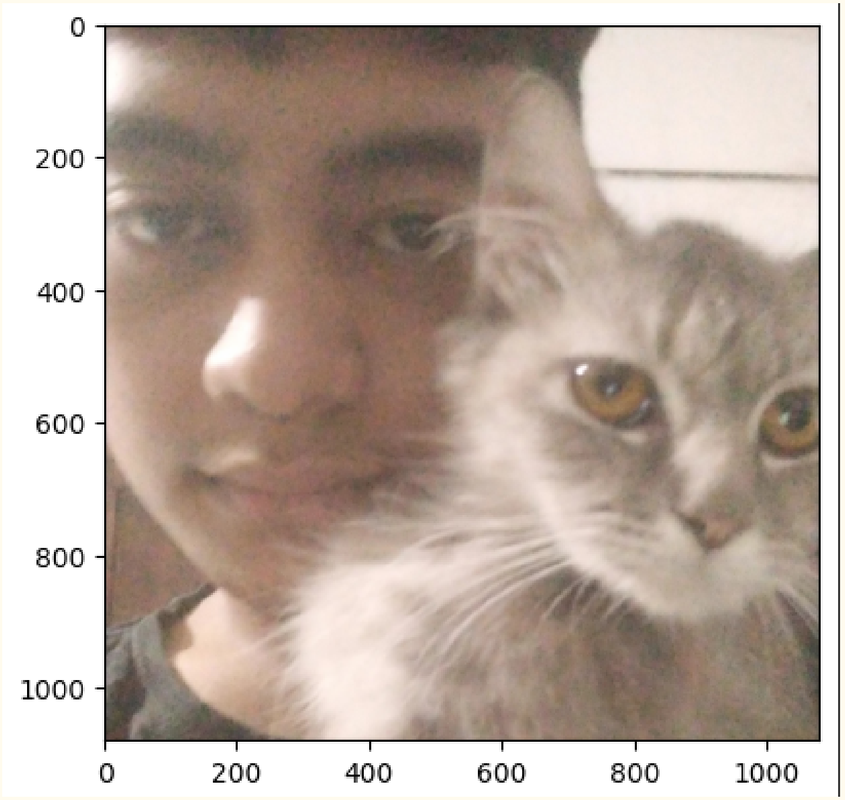
bintang.jpg



```python
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
img_path = 'bintang.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))
```

```
(1080, 1080, 3)
[[[152 139 131]
  [157 144 136]
  [160 147 139]
  ...
  [213 204 199]
  [213 203 201]
  [208 198 196]]

 [[153 140 132]
  [160 147 139]
  [162 149 141]
  ...
  [213 204 199]
  [210 200 198]
  [213 203 201]]

 [[158 145 137]
  [158 145 137]
  [164 151 143]
  ...
  [210 201 196]
  [206 196 194]
  [211 201 199]]

 ...
```

```
[[ 87  76  72]
 [ 78  67  63]
 [ 83  69  66]
 ...
 [112 109  94]
 [109 107  94]
 [115 113 100]]

[[ 96  85  81]
 [ 69  58  54]
 [ 79  65  62]
 ...
 [108 105  90]
 [111 109  96]
 [109 107  94]]

[[ 75  64  60]
 [ 75  64  60]
 [ 91  77  74]
 ...
 [109 106  91]
 [103 101  88]
 [111 109  96]]]
```



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')
```

```
[[[141 141 141]
  [146 146 146]
  [149 149 149]
  ...
  [206 206 206]
  [207 207 207]
  [202 202 202]]

 [[142 142 142]
  [149 149 149]
  [151 151 151]
  ...
  [206 206 206]
  [204 204 204]
  [207 207 207]]

 [[147 147 147]
  [147 147 147]
  [153 153 153]
  ...
  [203 203 203]
  [200 200 200]
  [205 205 205]]

 ...
```

```
[[ 79  79  79]
 [ 70  70  70]
 [ 74  74  74]
 ...
 [103 103 103]
 [101 101 101]
 [107 107 107]]

[[ 88  88  88]
 [ 61  61  61]
 [ 70  70  70]
 ...
 [ 99  99  99]
 [103 103 103]
 [101 101 101]]

[[ 67  67  67]
 [ 67  67  67]
 [ 82  82  82]
 ...
 [100 100 100]
 [ 95  95  95]
 [103 103 103]]]
```

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')
```

```
[[140.66666667 145.66666667 148.66666667 ... 205.33333333 205.66666667
  200.66666667]
 [141.66666667 148.66666667 150.66666667 ... 205.33333333 202.66666667
  205.66666667]
 [146.66666667 146.66666667 152.66666667 ... 202.33333333 198.66666667
  203.66666667]
 ...
 [ 78.33333333  69.33333333  72.66666667 ... 105.         103.33333333
  109.33333333]
 [ 87.33333333  60.33333333  68.66666667 ... 101.         105.33333333
  103.33333333]
 [ 66.33333333  66.33333333  80.66666667 ... 102.          97.33333333
  105.33333333]]
```

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')
```

```
[[141.1862 146.1862 149.1862 ... 205.5524 204.9816 199.9816]
 [142.1862 149.1862 151.1862 ... 205.5524 201.9816 204.9816]
 [147.1862 147.1862 153.1862 ... 202.5524 197.9816 202.9816]
 ...
 [ 78.0498  69.0498  71.7598 ... 108.5548 106.4866 112.4866]
 [ 87.0498  60.0498  67.7598 ... 104.5548 108.4866 106.4866]
 [ 66.0498  66.0498  79.7598 ... 105.5548 100.4866 108.4866]]
```

Metode Weighted Average

```
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')
```

```
[[141.975 146.975 149.975 ... 206.121 205.762 200.762]
 [142.975 149.975 151.975 ... 206.121 202.762 205.762]
 [147.975 147.975 153.975 ... 203.121 198.762 203.762]
 ...
 [ 78.833  69.833  72.844 ... 108.187 106.116 112.116]
 [ 87.833  60.833  68.844 ... 104.187 108.116 106.116]
 [ 66.833  66.833  80.844 ... 105.187 100.116 108.116]]
```

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

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

plt.imshow(img)

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))
```

```
(4000, 1847, 3)
[[[ 35  39  42]
  [ 36  40  43]
  [ 37  38  42]
  ...
  [ 38  38  38]
  [ 39  39  39]
  [ 37  37  37]]

 [[ 36  40  43]
  [ 37  41  44]
  [ 37  38  42]
  ...
  [ 38  38  38]
  [ 38  38  38]
  [ 38  38  38]]

 [[ 35  39  42]
  [ 36  40  43]
  [ 34  38  41]
  ...
  [ 38  38  38]
  [ 41  41  41]
  [ 37  37  37]]

 ...
```

```
[[[128 128 128]
  [128 128 128]
  [128 128 128]
  ...
  [128 128 128]
  [128 128 128]
  [128 128 128]]

 [[128 128 128]
  [128 128 128]
  [128 128 128]
  ...
  [128 128 128]
  [128 128 128]
  [128 128 128]]

 [[128 128 128]
  [128 128 128]
  [128 128 128]
  ...
  [128 128 128]
  [128 128 128]
  [128 128 128]]]
```

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')
```

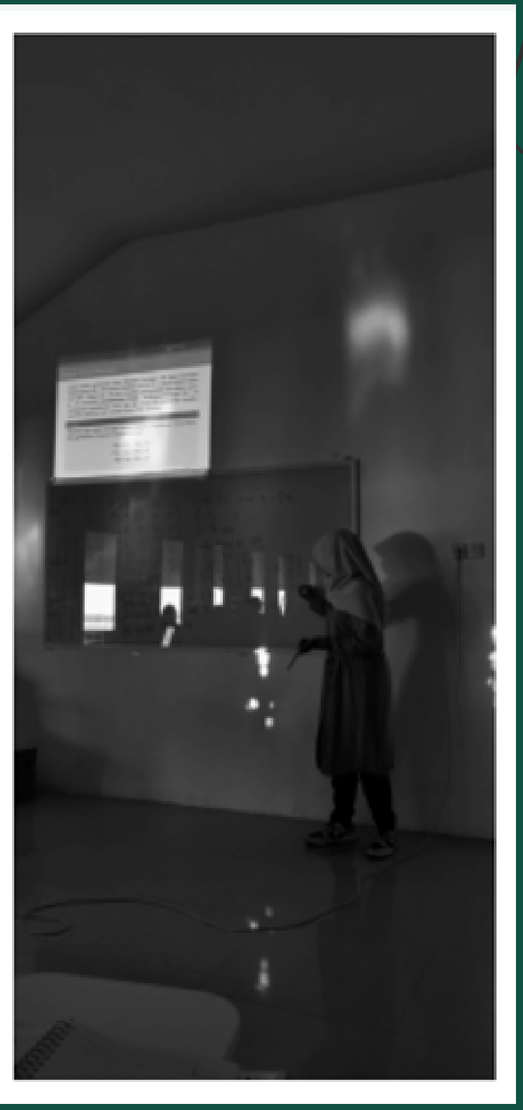
```
[[[ 38  38  38]
  [ 39  39  39]
  [ 39  39  39]
  ...
  [ 38  38  38]
  [ 39  39  39]
  [ 37  37  37]]

 [[ 39  39  39]
  [ 40  40  40]
  [ 39  39  39]
  ...
  [ 38  38  38]
  [ 38  38  38]
  [ 38  38  38]]

 [[ 38  38  38]
  [ 39  39  39]
  [ 37  37  37]
  ...
  [ 38  38  38]
  [ 41  41  41]
  [ 37  37  37]]

 ...
```

```
[[128 128 128]
 [128 128 128]
 [128 128 128]
 ...
 [128 128 128]
 [128 128 128]
 [128 128 128]]

[[128 128 128]
 [128 128 128]
 [128 128 128]
 ...
 [128 128 128]
 [128 128 128]
 [128 128 128]]

[[128 128 128]
 [128 128 128]
 [128 128 128]
 ...
 [128 128 128]
 [128 128 128]
 [128 128 128]]]
```

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')
```

```
[[ 38.66666667  39.66666667  39.         ...  38.         39.
   37.        ]
 [ 39.66666667  40.66666667  39.         ...  38.         38.
   38.        ]
 [ 38.66666667  39.66666667  37.66666667 ...  38.         41.
   37.        ]
 ...
 [128.         128.         128.         ... 128.        128.
  128.        ]
 [128.         128.         128.         ... 128.        128.
  128.        ]
 [128.         128.         128.         ... 128.        128.
  128.        ]]
```



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')
```

```
[[ 38.3662  39.3662  38.0762 ...  38.      39.      37.    ]
 [ 39.3662  40.3662  38.0762 ...  38.      38.      38.    ]
 [ 38.3662  39.3662  37.3662 ...  38.      41.      37.    ]
 ...
 [128.     128.     128.     ... 128.     128.     128.    ]
 [128.     128.     128.     ... 128.     128.     128.    ]
 [128.     128.     128.     ... 128.     128.     128.    ]]
```

Metode Weighted Average

```
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')
```

```
[[ 38.146  39.146  38.157 ...  38.     39.     37.   ]
 [ 39.146  40.146  38.157 ...  38.     38.     38.   ]
 [ 38.146  39.146  37.146 ...  38.     41.     37.   ]
 ...
 [128.    128.    128.    ... 128.    128.    128.   ]
 [128.    128.    128.    ... 128.    128.    128.   ]
 [128.    128.    128.    ... 128.    128.    128.   ]]
```

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('Mode Lightness.jpg', bbox_inches='tight')
```

```
[[[216 216 216]
  [216 216 216]
  [216 216 216]
  ...
  [174 174 174]
  [174 174 174]
  [174 174 174]]

 [[216 216 216]
  [216 216 216]
  [216 216 216]
  ...
  [174 174 174]
  [174 174 174]
  [174 174 174]]

 [[217 217 217]
  [217 217 217]
  [217 217 217]
  ...
  [174 174 174]
  [174 174 174]
  [174 174 174]]

 ...
```

```
 ...

 [[ 44  44  44]
  [ 81  81  81]
  [106 106 106]
  ...
  [ 97  97  97]
  [ 97  97  97]
  [ 64  64  64]]

 [[ 15  15  15]
  [ 69  69  69]
  [102 102 102]
  ...
  [ 93  93  93]
  [ 97  97  97]
  [ 41  41  41]]

 [[  6   6   6]
  [ 52  52  52]
  [100 100 100]
  ...
  [ 93  93  93]
  [ 81  81  81]
  [ 31  31  31]]]
```

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.jpg' , bbox_inches='tight')
```

```
[[214.33333333 214.33333333 214.33333333 ... 174.33333333 174.33333333
  174.33333333]
 [214.33333333 214.33333333 214.33333333 ... 174.33333333 174.33333333
  174.33333333]
 [215.33333333 215.33333333 215.33333333 ... 174.33333333 174.33333333
  174.33333333]
 ...
 [ 43.66666667  80.66666667 105.         ...  96.          96.33333333
   63.33333333]
 [ 14.          69.         101.66666667 ...  93.          96.33333333
   40.33333333]
 [  5.33333333  52.          99.66666667 ...  92.33333333  80.33333333
   30.66666667]]
```

Metode Luminosity

```
lumi_img = (0.2126*R) + (0.7152*G) + (0.0722*B)
print(lumi_img)

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

```
[[211.4412 211.4412 211.4412 ... 169.5204 169.5204 169.5204]
 [211.4412 211.4412 211.4412 ... 169.5204 169.5204 169.5204]
 [212.4412 212.4412 212.4412 ... 169.5204 169.5204 169.5204]
 ...
 [ 41.2378  78.7404 102.1696 ...  91.906   92.259   59.259 ]
 [ 11.5908  67.0934  99.7404 ...  89.8298  93.4636  37.604 ]
 [  3.0842  50.0934  97.2378 ...  89.1828  77.604   28.7404]]
```

Metode Weighted Average

```
warv_img = (0.299*R) + (0.587*G) + (0.114*B)
print(warv_img)

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

```
[[211.77  211.77  211.77  ... 167.742 167.742 167.742]
 [211.77  211.77  211.77  ... 167.742 167.742 167.742]
 [212.77  212.77  212.77  ... 167.742 167.742 167.742]
 ...
 [ 41.09   78.378 102.019 ...  91.354  91.838  58.838]
 [ 11.574  66.862  99.378 ...  89.197  93.051  37.236]
 [  3.243  49.862  97.09  ...  88.681  77.236  28.378]]
```

Kesimpulan

Metode Average. Menurut kami metode ini sederhana dan metode ini juga hanya mengambil nilai rata-rata dari intensitas warna di seluruh saluran warna. Kami suka dengan kesederhanaan dan kejelasan.

Terimakasih