

Rahma Fairuz Rania J0303201065 TPL 57 A2

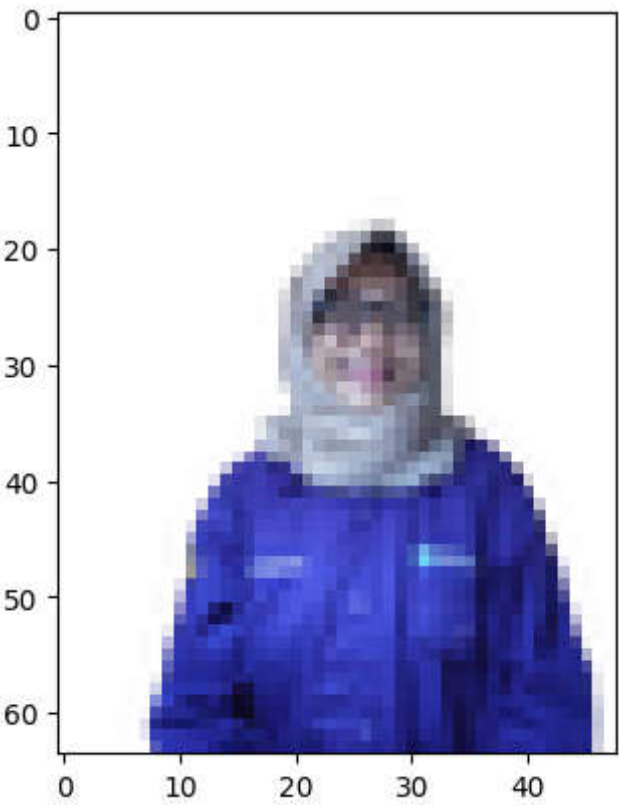
operasi pixel dan histogram

1. ubah ukuran gambar

```
In [ ]: import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline
from PIL import Image
```

```
In [18]: # resampling menjadi 64x64 px
img = Image.open('dt/rahma.png')
img.thumbnail((64, 64), Image.ANTIALIAS)# image in place
imgplot = plt.imshow(img)
```

C:\Users\Asus\AppData\Local\Temp\ipykernel_1804\4246501679.py:2: DeprecationWarning: ANTIALIAS is deprecated and will be removed in Pillow 10 (2023-07-01). Use Resampling.LANCZOS instead.
img.thumbnail((64, 64), Image.ANTIALIAS)# image in place



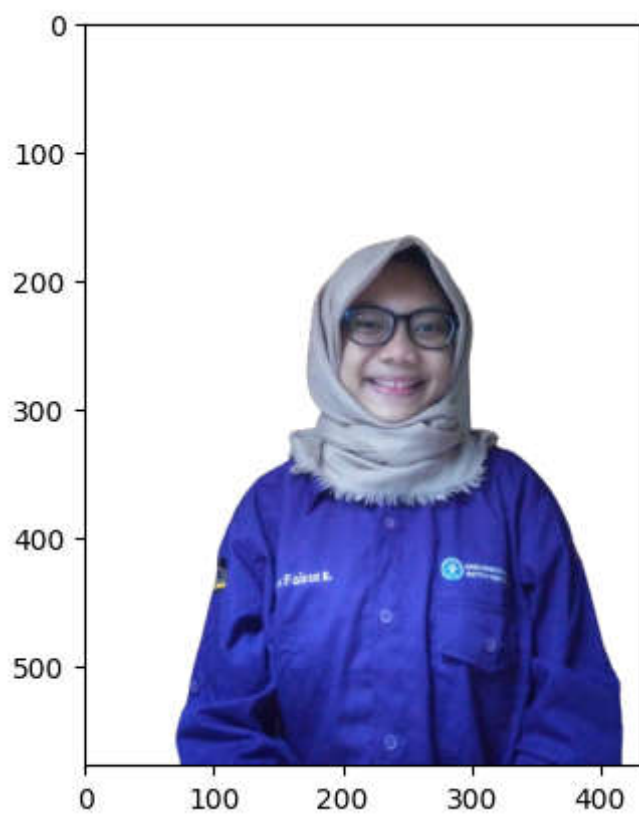
```
In [19]: img.size
```

Out[19]: (48, 64)

2. import data gambar ke dlm arr np

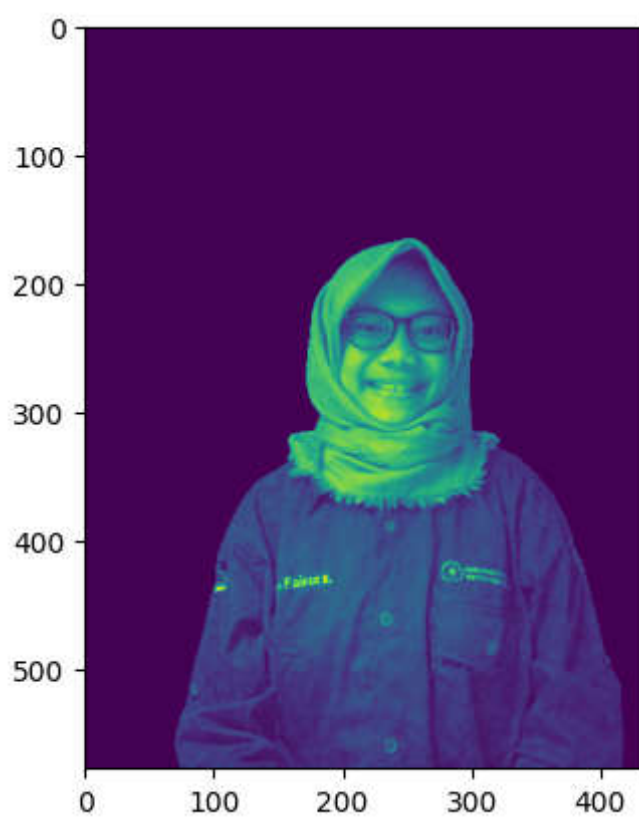
```
In [22]: import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline
from PIL import Image
```

```
In [23]: img = mpimg.imread('dt/rahma.png')
imgplot = plt.imshow(img)
```

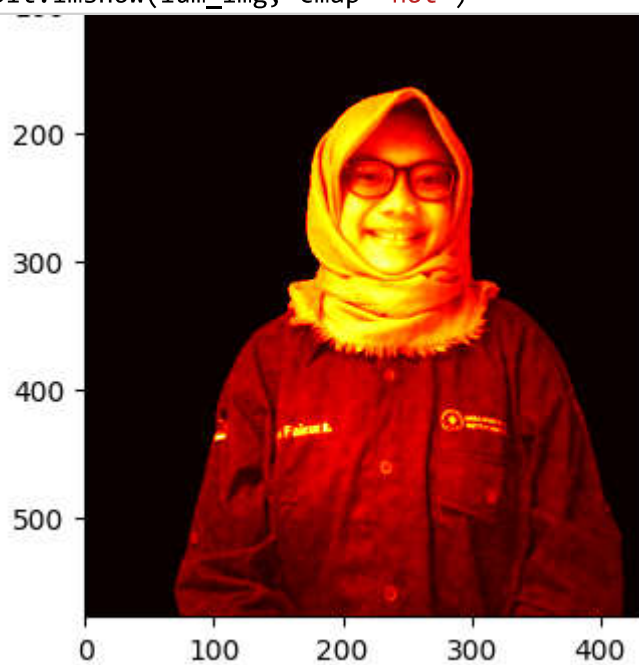


```
In [24]: # slicing channel merah pada gambar, ambil semua channel green dan blue
lum_img = img[:, :, 0]
plt.imshow(lum_img)
```

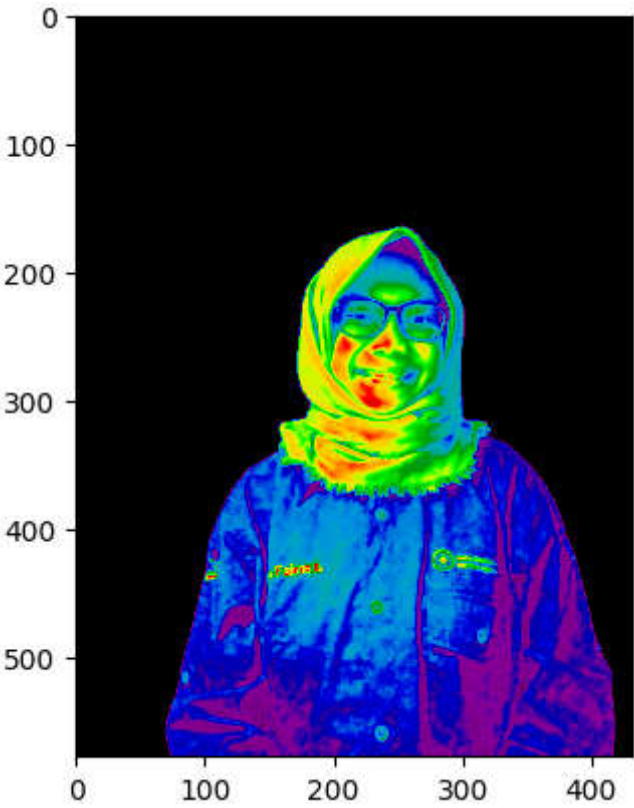
Out[24]: <matplotlib.image.AxesImage at 0x20d3d1d4d00>



```
In [17]: # colormap efek panas
plt.imshow(lum_img, cmap="hot")
```

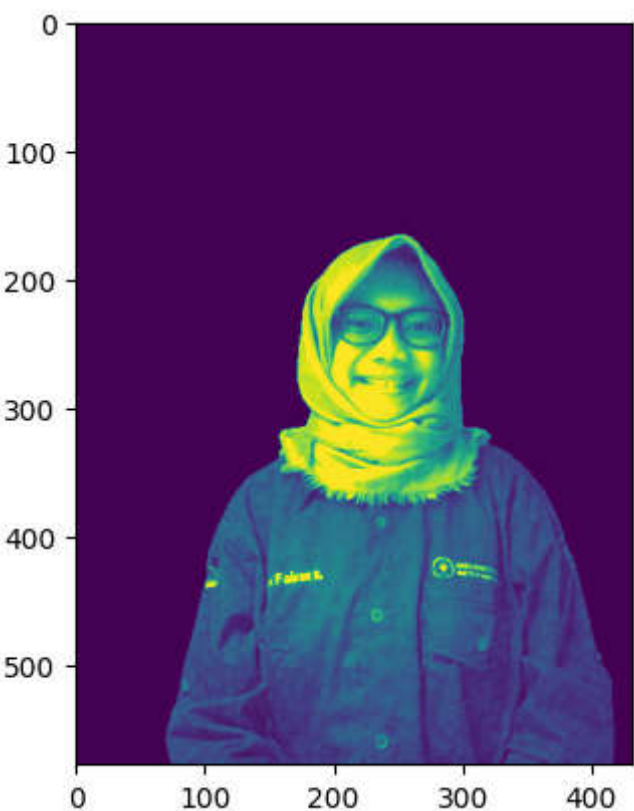


```
In [18]: # colormap efek spectral
imgplot = plt.imshow(lum_img)
imgplot.set_cmap('nipy_spectral')
# mengatur color map
```



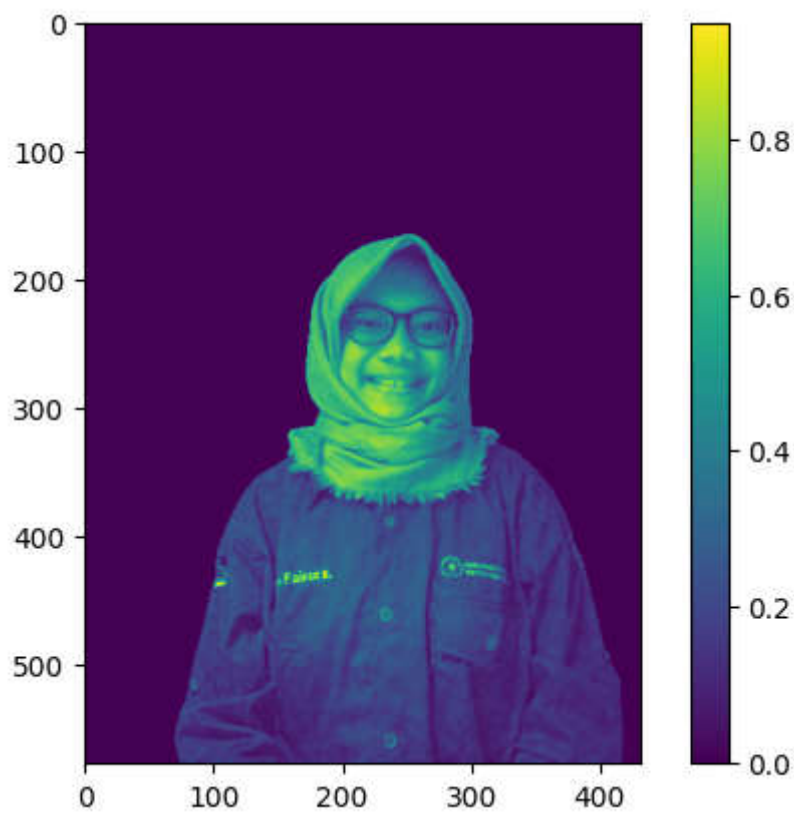
```
In [19]: # cari range intensitas yg tepat utk hist
plt.hist(lum_img.ravel(), bins=256, range=(0.0, 1.0), fc='k', ec='k')
3.69000e+02, 3.63000e+02, 3.16000e+02, 2.70000e+02, 2.37000e+02,
2.20000e+02, 2.37000e+02, 2.10000e+02, 1.80000e+02, 1.92000e+02,
1.94000e+02, 1.45000e+02, 1.76000e+02, 1.73000e+02, 1.50000e+02,
1.62000e+02, 1.42000e+02, 1.28000e+02, 1.56000e+02, 1.44000e+02,
1.69000e+02, 1.54000e+02, 1.53000e+02, 1.43000e+02, 1.41000e+02,
1.58000e+02, 1.53000e+02, 1.59000e+02, 1.75000e+02, 1.90000e+02,
1.80000e+02, 1.46000e+02, 1.31000e+02, 1.67000e+02, 1.70000e+02,
1.37000e+02, 1.47000e+02, 1.63000e+02, 1.54000e+02, 1.44000e+02,
1.29000e+02, 1.37000e+02, 1.70000e+02, 1.25000e+02, 1.47000e+02,
1.43000e+02, 1.57000e+02, 1.40000e+02, 1.62000e+02, 1.50000e+02,
1.57000e+02, 1.60000e+02, 1.57000e+02, 1.42000e+02, 1.87000e+02,
1.72000e+02, 1.73000e+02, 1.77000e+02, 1.85000e+02, 1.79000e+02,
1.49000e+02, 1.77000e+02, 1.87000e+02, 1.62000e+02, 1.87000e+02,
1.79000e+02, 1.53000e+02, 1.85000e+02, 1.84000e+02, 1.82000e+02,
1.71000e+02, 1.91000e+02, 1.96000e+02, 2.35000e+02, 2.21000e+02,
2.43000e+02, 2.16000e+02, 1.96000e+02, 2.15000e+02, 2.25000e+02,
2.45000e+02, 2.64000e+02, 2.36000e+02, 2.56000e+02, 2.17000e+02,
2.79000e+02, 2.50000e+02, 2.57000e+02, 2.31000e+02, 2.33000e+02,
2.40000e+02, 2.31000e+02, 2.05000e+02, 2.30000e+02, 2.09000e+02,
2.05000e+02, 1.98000e+02, 2.03000e+02, 2.05000e+02, 2.36000e+02,
```

```
In [20]: imgplot = plt.imshow(lum_img, clim=(0.0, 0.7))
```



```
In [21]: # menampilkan range colormap pada gambar
imgplot = plt.imshow(lum_img)
plt.colorbar()
```

Out[21]: <matplotlib.colorbar.Colorbar at 0x2dea4703fd0>



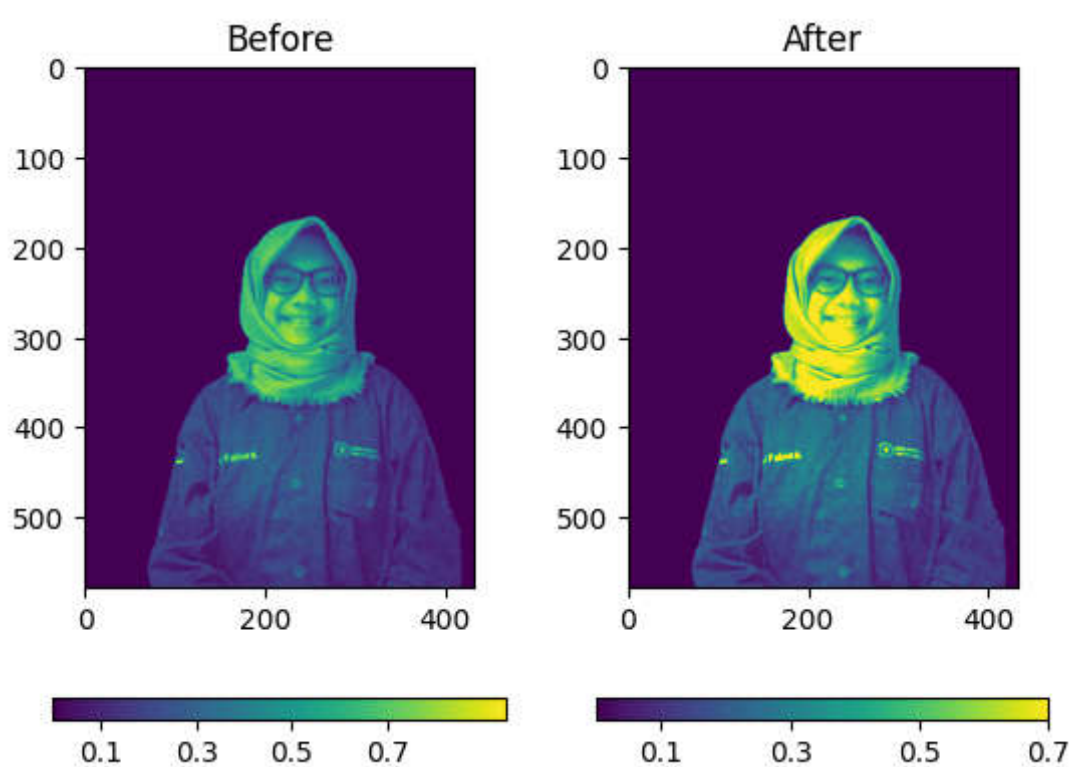
```
In [22]: fig = plt.figure()
ax = fig.add_subplot(1, 2, 1)
imgplot = plt.imshow(lum_img)

ax.set_title('Before')
plt.colorbar(ticks=[0.1, 0.3, 0.5, 0.7],
orientation='horizontal')
ax = fig.add_subplot(1, 2, 2)
imgplot = plt.imshow(lum_img)

# ubah colormap limit scale, limit 0.7 (warna kuning paling tinggi intensitasnya)
imgplot.set_clim(0.0, 0.7)

ax.set_title('After')
plt.colorbar(ticks=[0.1, 0.3, 0.5, 0.7],
orientation='horizontal')
```

Out[22]: <matplotlib.colorbar.Colorbar at 0x2dea4736c10>

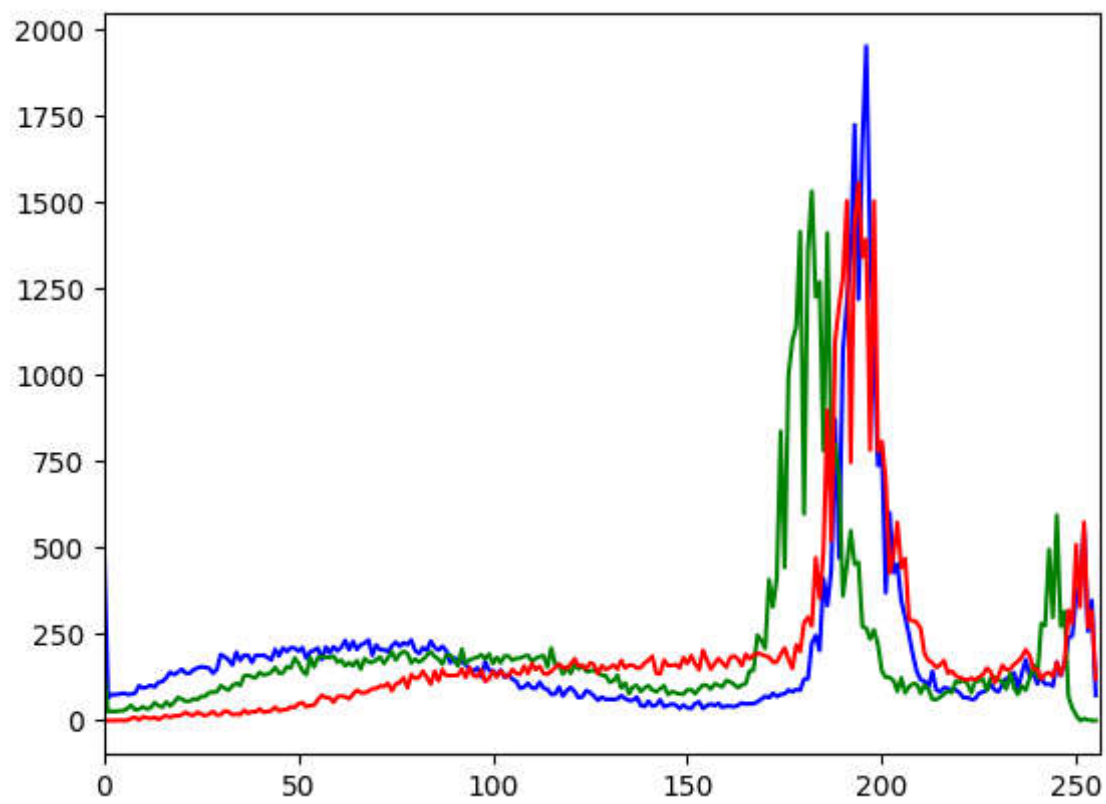


3. kode call hist with plt.hist()

```
In [23]: import cv2
import numpy as np
from matplotlib import pyplot as plt
```

```
In [50]: # grafik utk melihat banyaknya penggunaan channel masing-masing r, g, b pada gambar
img = cv2.imread('dt/meng.jfif')
color = ('b','g','r')
for i,col in enumerate(color):
    # cv2.calcHist([images], [channels], mask, histSize, ranges)
    histr = cv2.calcHist([img],[i],None,[256],[0,256])
    plt.plot(histr,color = col)
    plt.xlim([0,256])

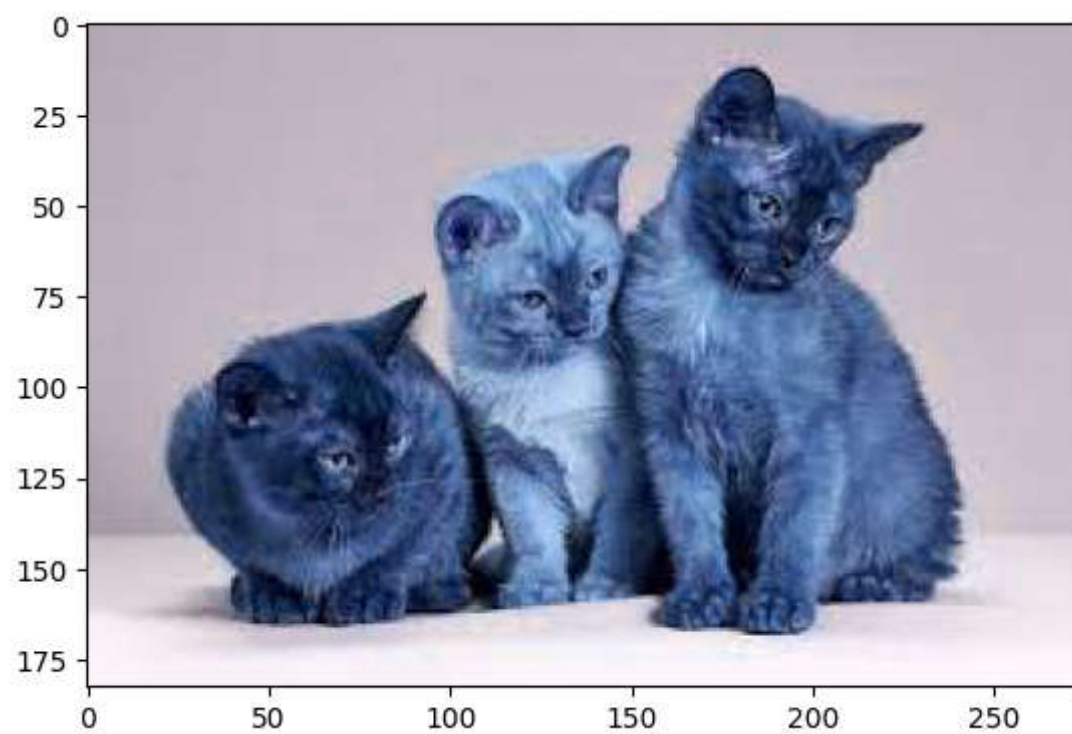
plt.show( )
```



```
In [37]: img
```

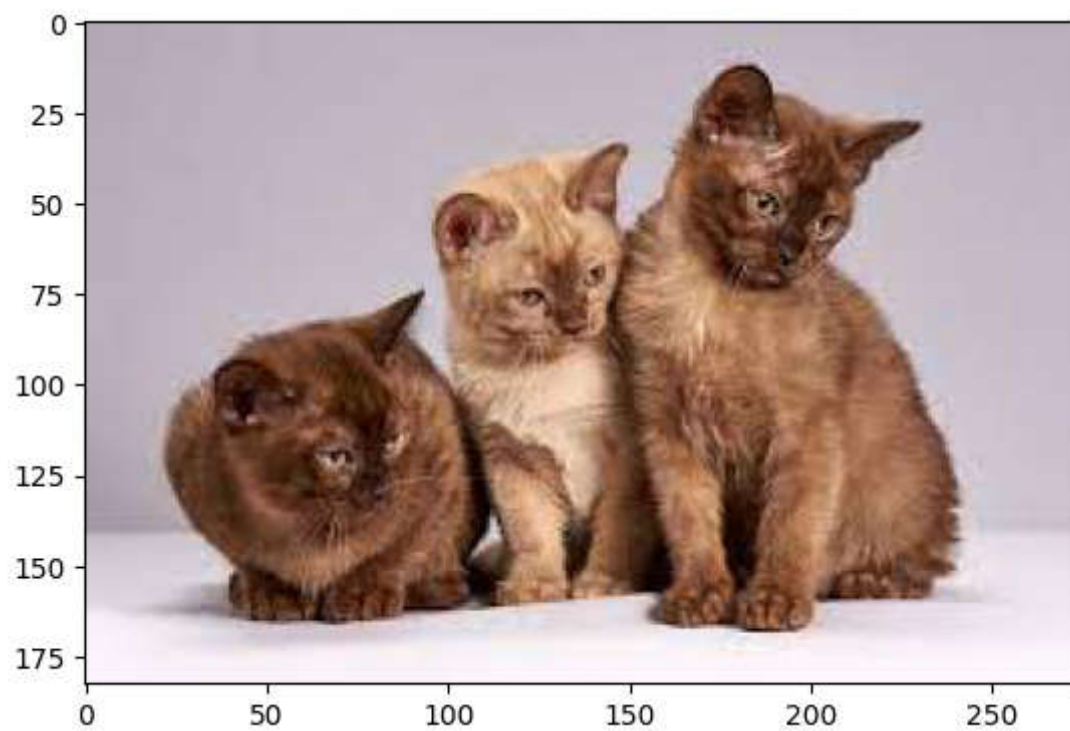
```
In [51]: # default BGR karena menggunakan cv2
plt.imshow(img)
```

```
Out[51]: <matplotlib.image.AxesImage at 0x2dea399f1f0>
```

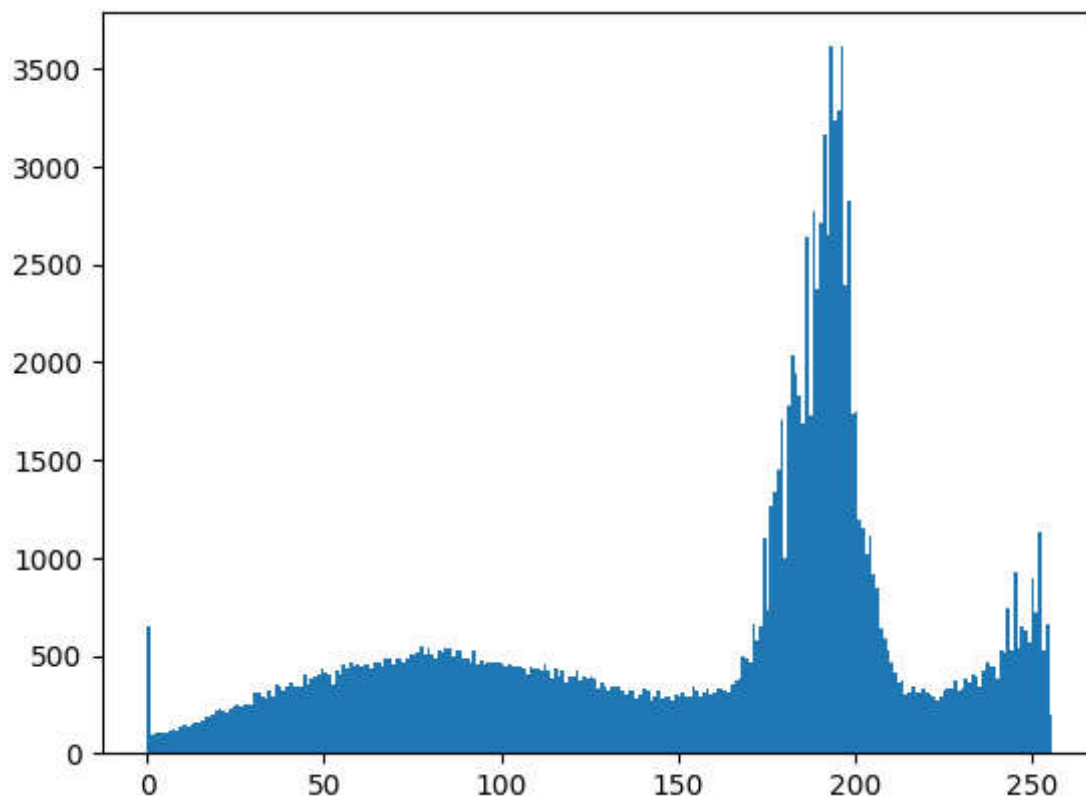



```
In [52]: # convert BGR to RGB
img2 = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img2)
```

Out[52]: <matplotlib.image.AxesImage at 0x2dea66b95b0>



```
In [53]: # melihat persebaran channel RGB secara keseluruhan
plt.hist(img.ravel(),256,[0,256]);
plt.show( )
```



4. manipulasi exposur gambar

```
In [54]: import numpy as np
import matplotlib.pyplot as plt
import skimage.exposure as skie
%matplotlib inline
```

```
In [61]: # import gambar error, file mungkin dipindah/dihapus oleh owner
img = plt.imread('https://github.com/ipythonbooks/cookbook-2nd-data/blob/master/beach.png?raw=true')[..., 0]

525         return response

E:\Anaconda\lib\urllib\request.py in http_response(self, request, response)
630         # request was successfully received, understood, and accepted.
631         if not (200 <= code < 300):
--> 632             response = self.parent.error(
633                 'http', request, response, code, msg, hdrs)
634

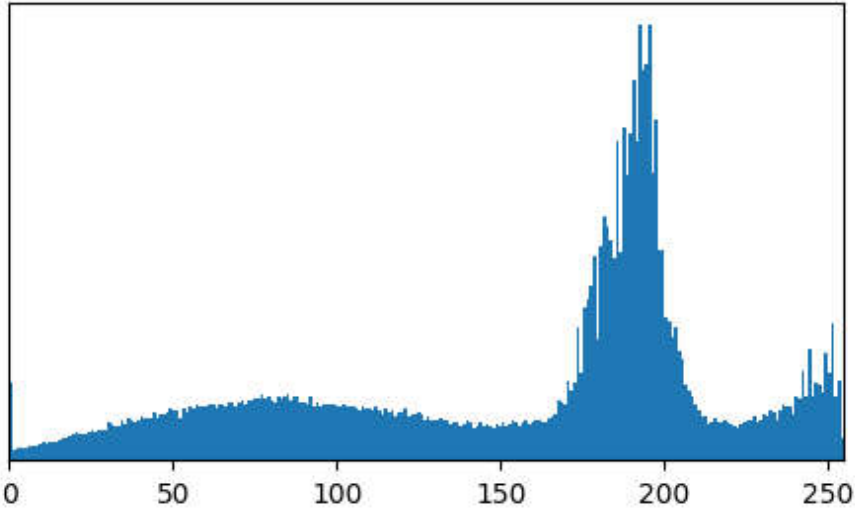
E:\Anaconda\lib\urllib\request.py in error(self, proto, *args)
559         if http_err:
560             args = (dict, 'default', 'http_error_default') + orig_args
--> 561         return self._call_chain(*args)
562
563 # XXX probably also want an abstract factory that knows when it makes

E:\Anaconda\lib\urllib\request.py in _call_chain(self, chain, kind, meth_name, *args)
492         for handler in handlers:
493             func = getattr(handler, meth_name)
...
```

```
In [62]: img = cv2.imread('dt/meng.jfif')
```

```
In [63]: def show(img):
# Display the image.
fig, (ax1, ax2) = plt.subplots(1, 2,
figsize=(12, 3))
ax1.imshow(img, cmap=plt.cm.gray)
ax1.set_axis_off()
# Display the histogram.
ax2.hist(img.ravel(), lw=0, bins=256)
ax2.set_xlim(0, img.max())
ax2.set_yticks([])
plt.show()

show(img)
```



```
In [64]: # menambahkan kontras dan melihatnya pada grafik
show(skie.rescale_intensity(
img, in_range=(0.4, .95), out_range=(0,
1)))
```

