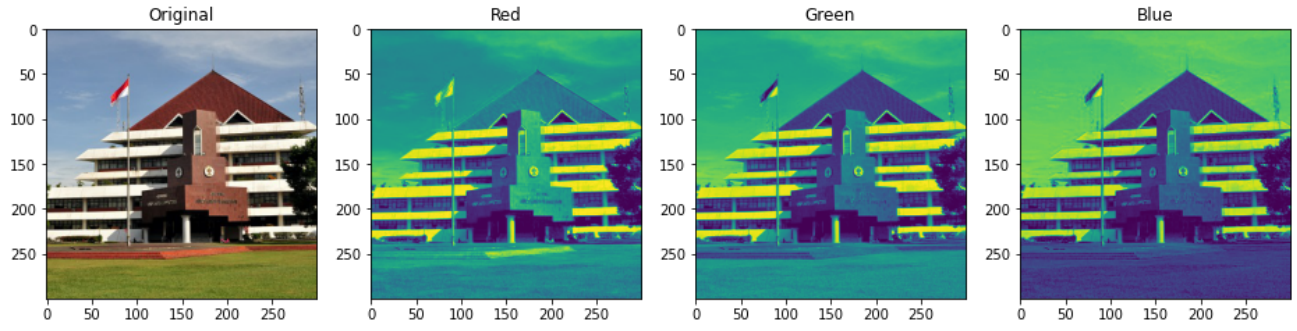
citra

```
import matplotlib.pyplot as plt
import skimage.io as io
from copy import deepcopy
import numpy as np

img = io.imread("https://ipb.ac.id/media/images/event/workshop-event1.jpg")

r_channel=deepcopy(img)
g_channel=deepcopy(img)
b_channel=deepcopy(img)

r_channel= img[:,:,0]
g_channel= img[:,:,1]
b_channel= img[:,:,2]


fig, axes = plt.subplots(1, 4, figsize=(16, 12)) #membuat subplot pada 1 bidang untuk mena
ax = axes.ravel()

ax[0].imshow(img)
ax[0].set_title("Original")
ax[1].imshow(r_channel)#menampilkan red channel dari sebuah pict
ax[1].set_title("Red")
ax[2].imshow(g_channel)#menampilkan green channel dari sebuah pict
ax[2].set_title('Green')
ax[3].imshow(b_channel)#menampilkan blue channel dari sebuah pict
ax[3].set_title('Blue')
```

```
Text(0.5, 1.0, 'Blue')
```

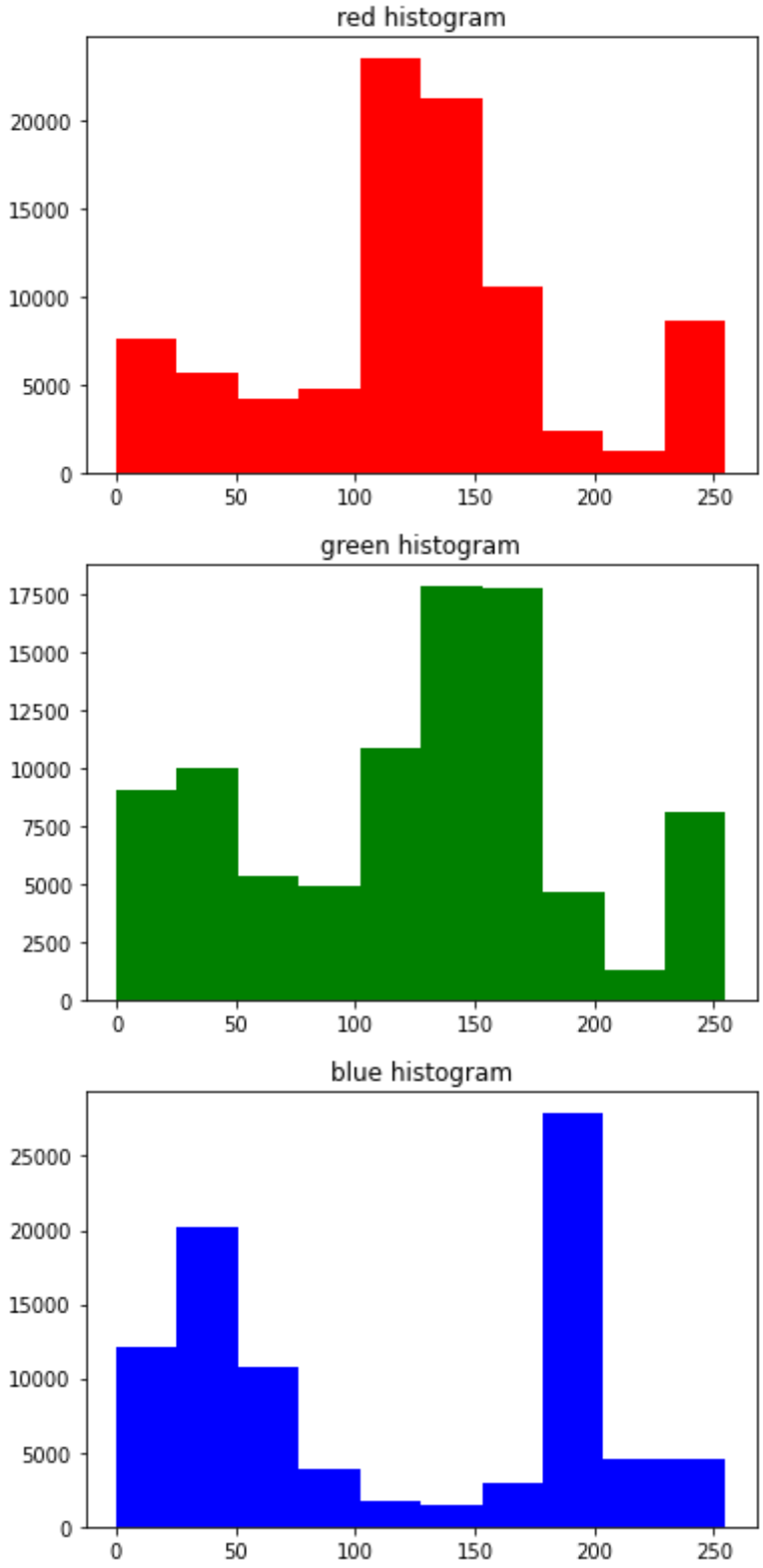


histogram

```
plt.hist(r_channel.flatten(), color ='red')
plt.title('red histogram')
plt.figure()
```

```
plt.hist(g_channel.flatten(), color='green')
plt.title('green histogram')
plt.figure()
plt.hist(b_channel.flatten(), color = 'blue')
plt.title('blue histogram')
```

```
Text(0.5, 1.0, 'blue histogram')
```

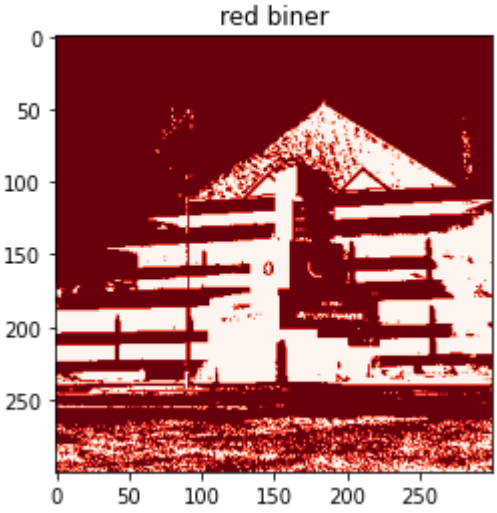


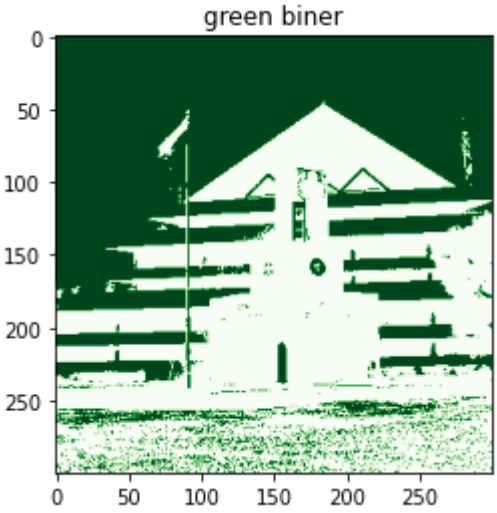
biner

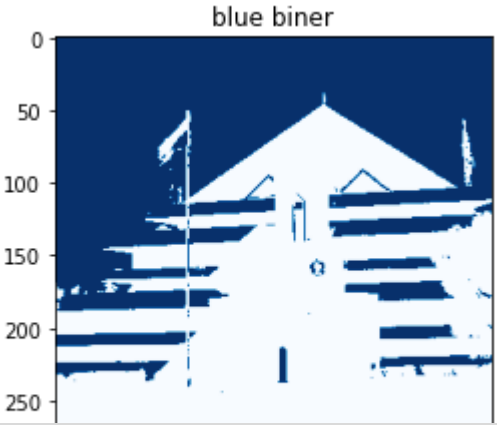
```
# perhitungan beda rumus
img_ngnay = n_channel
```

```
img_rgray = r_channel
img_ggray = g_channel
img_bgray = b_channel

def binerisasi(img,thresshold):
  img_q=np.zeros((img.shape[0], img.shape[1]))
  for r in range(0,img.shape[0]):
    for c in range(0,img.shape[1]):
      if img[r,c]< thresshold:
        img_q[r,c] = 0
      else:
        img_q[r,c] =255
  return img_q

img_rbiner = binerisasi(img_rgray,111)
plt.figure()
plt.imshow(img_rbiner,cmap=plt.cm.Reds)
plt.title('red biner')

img_gbiner = binerisasi(img_ggray,135)
plt.figure()
plt.imshow(img_gbiner, cmap=plt.cm.Greens)
plt.title('green biner')

img_bbiner = binerisasi(img_bgray,170)
plt.figure()
plt.imshow(img_bbiner, cmap=plt.cm.Blues)
plt.title('blue biner')
```

```
Text(0.5, 1.0, 'blue biner')
```







✓ 3s completed at 23:28