

Image Processing: Histograms

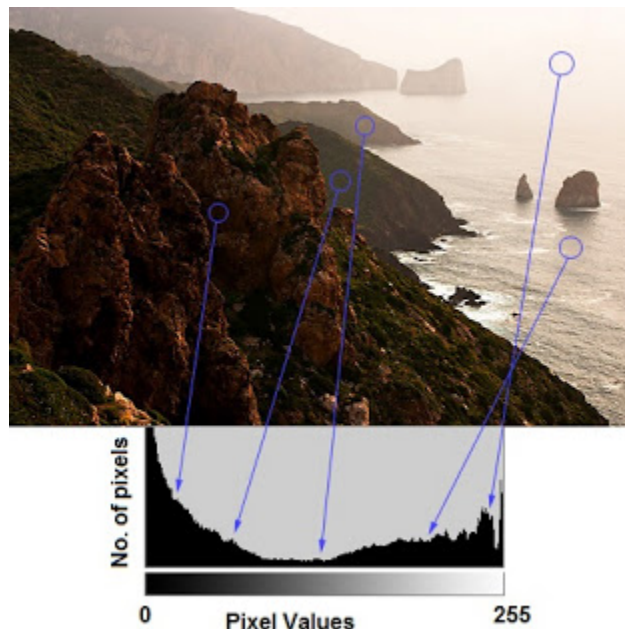
Natalis Ransi

Pada materi kali ini akan disajikan teori dan aplikasi untuk menyajikan gambar (*image*) dalam bentuk histogram. Akan disajikan juga penggunaan OpenCV dan Numpy pada Bahasa pemrograman Python untuk menghitung frekuensi intensitas warna pada sebuah gambar untuk selanjutnya disajikan dalam bentuk grafik. Fungsi seperti `cv2.calcHist()` akan digunakan.

Theory

Histogram dari sebuah *image* dapat disajikan dalam bentuk grafik, yang menunjukkan distribusi (lebih tepatnya frekuensi kemunculan) intensitas (misalnya derajat keabuan sebuah gambar gray. Dimana rentang nilainya antara 0 – 255. Jika disajikan dalam grafik dua dimensi (diagram kartesian) misalnya x dan y, sumbu x akan menyatakan intensitas warna dan sumbu y adalah banyaknya jumlah intensitas warna yang muncul pada sebuah gambar.

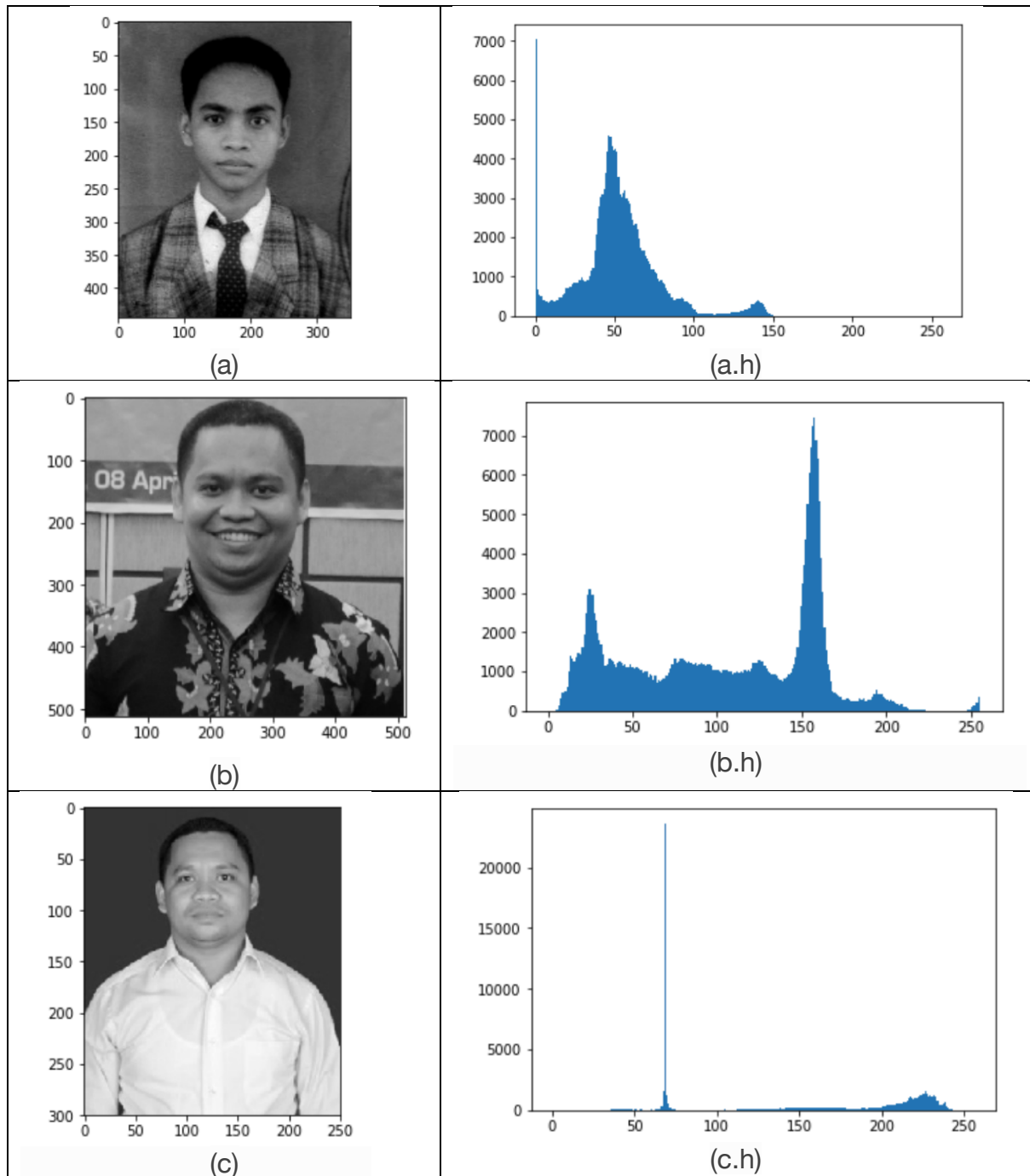
Dengan kata lain bahwa histogram merupakan cara lain untuk memahami sebuah gambar. Dengan melihat histogram sebuah gambar, secara intuisi kita dapat mendeskripsikan *contrast*, *brightness*, ataupun distribusi dari intensitas warna dari sebuah gambar. Hampir semua aplikasi pengolah gambar saat ini dilengkapi dengan fitur ini.



Gambar 1. Cambridge in Color website

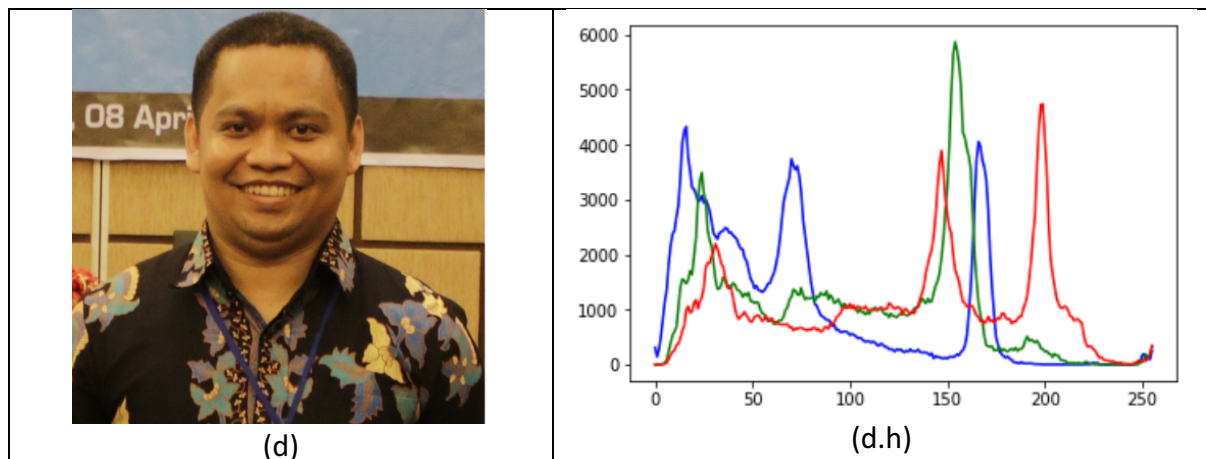
Sumber : <https://www.cambridgeincolour.com/tutorials/histograms1.htm>

Perhatikan Gambar 1 di atas, bagian bawah gambar disajikan grafik histogram dari gambar yang tertera. Jika dilihat dari histogram yang disajikan dapat kita lihat bahwa gambar tersebut memiliki frekuensi kemunculan warna hitam (0) atau yang mendekati 0 lebih banyak. Sehingga dapat disimpulkan bahwa gambar tersebut didominasi oleh warna gelap. Berikut diberikan contoh histogram yang dibentuk dari gambar



Gambar (a) didominasi oleh warna gelap, sehingga histogram (a.h) memperlihatkan distribusi warna sebagian besar berada di bawah 100. Gambar (b) terlihat distribusi intensitas warna lebih menyebar sedangkan gambar (c) terlihat bahwa gambar didominasi warna cerah sehingga dapat dilihat dari (c.h) distribusi warna berada 200 ke atas.

Untuk gambar berwarna (memiliki RGB) kita dapat juga menyajikannya dalam bentuk histogram.



Anda dapat menggunakan python dalam membuat histogram sebuah gambar (image). Berikut kode program yang bisa digunakan. Kedua kode program (Gambar 2 dan Gambar 3) menggunakan library OpenCV dan numpy sebagai alat bantu.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('Natalis_RansiGray.png',0)
img2 = cv2.imread('noelSmaGray.png',0)
img3 = cv2.imread('noelGrey.png',0)

plt.hist(img2.ravel(),256,[0,256]);
plt.show()
plt.hist(img.ravel(),256,[0,256]);
plt.show()
plt.hist(img3.ravel(),256,[0,256]);
plt.show()
```

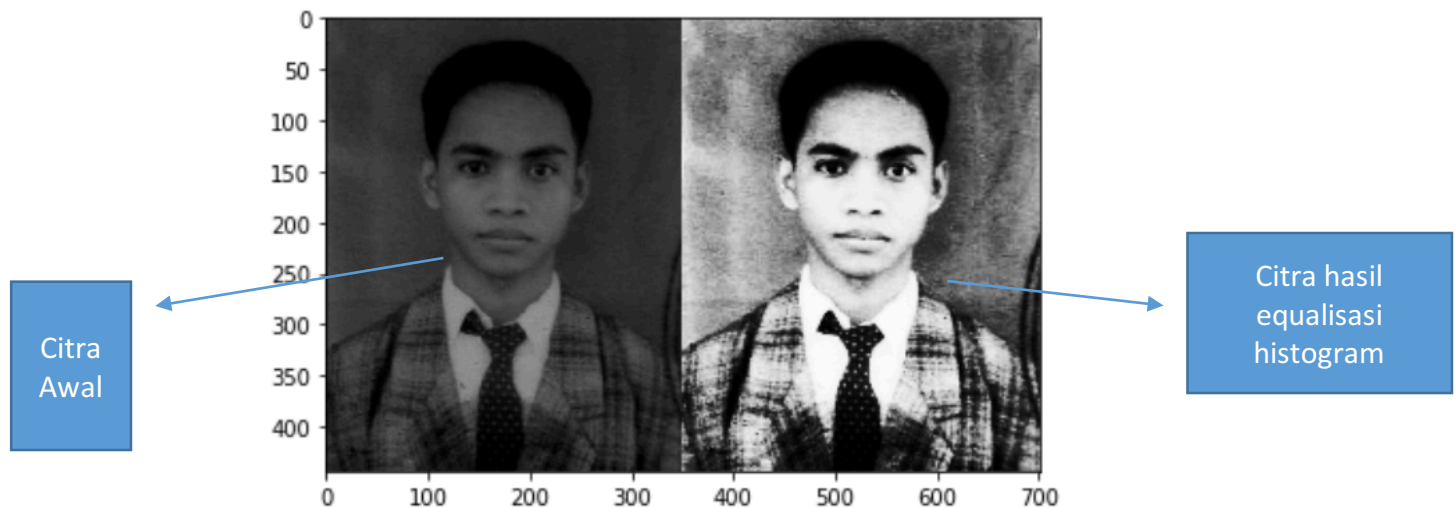
Gambar 2 Kode program menampilkan histogram sebuah gambar grey menggunakan Bahasa pemrograman python

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

imgRgb = cv2.imread('Natalis_Ransi.png')
color = ('b','g','r')
for i,col in enumerate(color):
    histr = cv2.calcHist([imgRgb],[i],None,[256],[0,256])
    plt.plot(histr,color = col)
plt.show()
```

Gambar 3 Kode program menampilkan histogram sebuah gambar berwarna menggunakan Bahasa pemrograman python

Pemahaman tentang histogram menjadi hal penting untuk kita memahami Histogram Equalization. Histogram Equalization bertujuan meningkatkan kontras sebuah gambar. Gambar empat memberikan contohnya. Jika diberikan gambar seperti tampak sebelah kiri (gambar tampak gelap) maka dengan histogram equalization kita dapat membuat gambar menjadi lebih kontras (gambar sebelah kanan)



Gambar 4 hasil equalisasi histogram sebuah gambar

Kita akan membahas ini pada pertemuan selanjutnya. Sebagai bahan bacaan silahkan akses tulisan berikut: https://www.math.uci.edu/icamp/courses/math77c/demos/hist_eq.pdf

Pada Gambar 2 dan Gambar 3 kode program menggunakan librari openCV. Jika Anda mengalami kesulitan silahkan menggunakan kode program seperti pada Gambar 5.

```
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open('noelSmaGray.png')
lebar, tinggi = (img.size)

larikImg = []
for x in range(lebar):
    for y in range(tinggi):
        pikselImg = img.getpixel((x,y))
        larikImg.append(pikselImg)

jml = 255
plt.hist(larikImg, jml, range=(0,255),facecolor='blue', alpha=0.5)
plt.show()
```

Gambar 5 kode program menampilkan histogram sebuah gambar

Catatan: noelSmaGray.png merupakan citra abu-abu.

Tugas: Buat histogram terhadap foto Anda, jika menggunakan python, jelaskan setiap baris kode Anda.