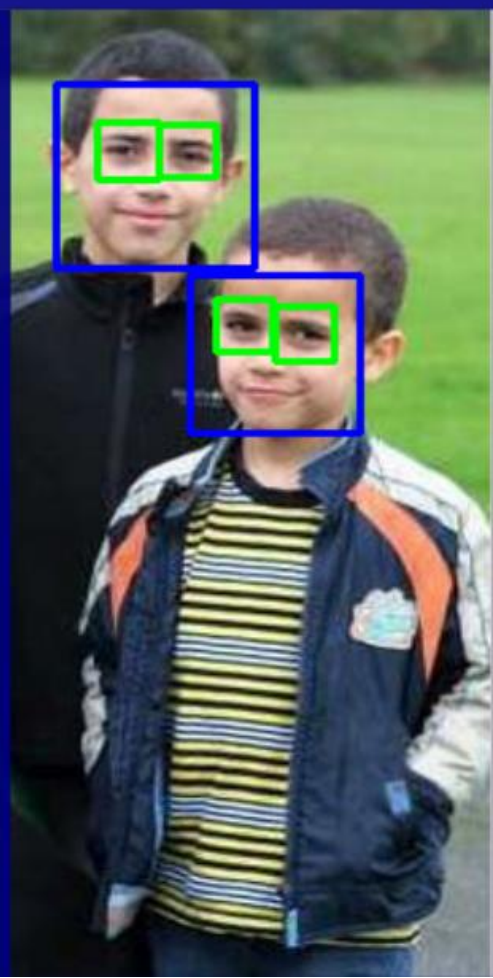
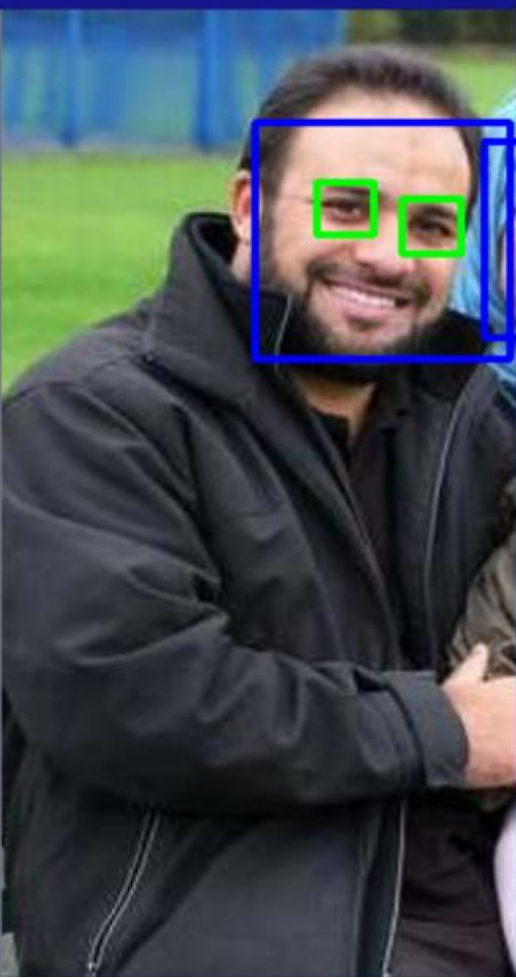




# OpenCV dengan Python

*Nor Kandir*



## **OpenCV dengan Python**

(Dasar pengolahan image dan video yang mencakup load, write, save, capture, hingga deteksi wajah)

Penyusun:

**Nor Kandır**

## **A. Daftar Isi**

OpenCV dengan Python .....	2
<b>A. Daftar Isi.....</b>	<b>3</b>
<b>B. Pendahuluan .....</b>	<b>4</b>
<b>C. Download dan Instalasi .....</b>	<b>6</b>
<b>D. Mengolah Image .....</b>	<b>8</b>
Load dan Menyimpan Image.....	9
Konversi Warna Image.....	10
Menampilkan Image .....	11
Ambil Image dari WebCam .....	12
<b>E. Mengolah Video.....</b>	<b>14</b>
Play Video dari WebCam.....	14
Play Video dari File.....	18
Save Video dari Webcam dan File .....	19
Realtime untuk Image dan Video.....	23
<b>F. Deteksi Wajah dengan Haar Cascade.....</b>	<b>26</b>
Realtime untuk Harr Cascade .....	29
<b>G. Kesimpulan .....</b>	<b>32</b>
<b>H. Daftar Pustaka .....</b>	<b>33</b>
<b>I. Penulis .....</b>	<b>34</b>

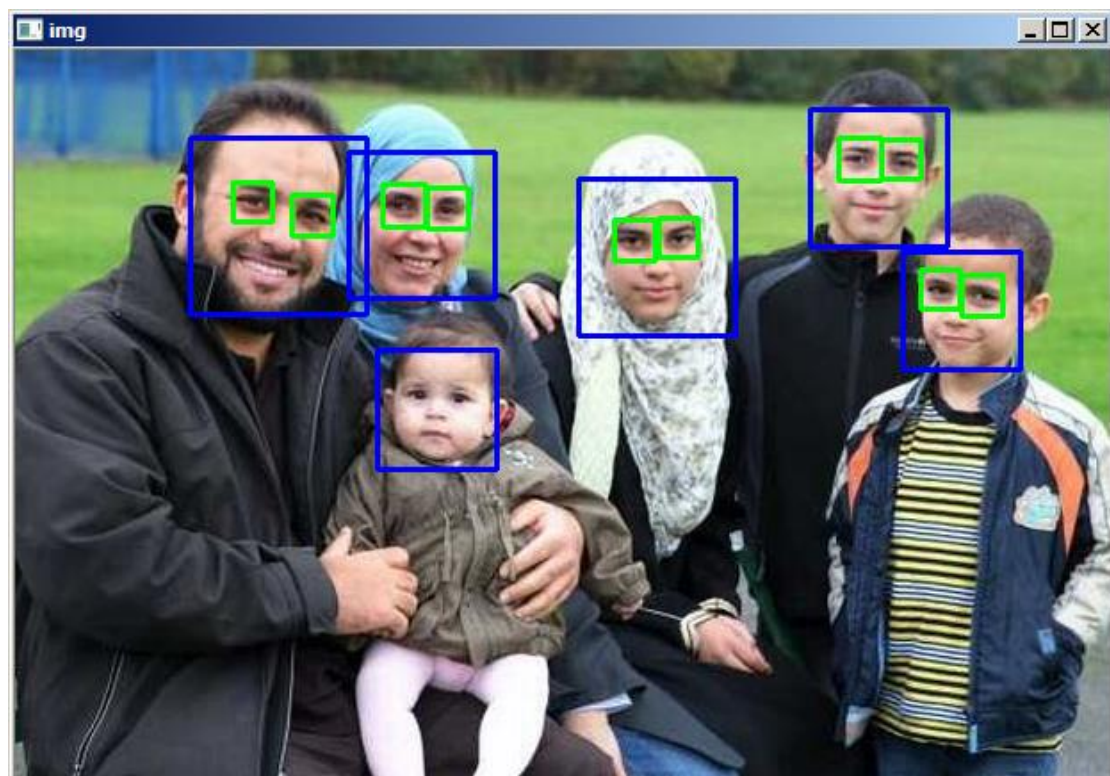
## B. Pendahuluan

Ada dua istilah penting di sini yaitu **OpenCV** dan **Python**. Apa itu OpenCV? **OpenCV** adalah sebuah library (perpustakaan) yang digunakan untuk mengolah gambar dan video. Kata Open pada OpenCV dimaksudkan *opensource* yakni gratis, tidak berbayar, bisa didownload oleh siapa saja. Sementara CV pada kata OpenCV adalah kependekan dari Computer Vision, maksudnya adalah komputer yang digunakan untuk mengolah image (citra/gambar) yang ditangkap oleh alat perekam (seperti kamera atau webcam) yang dikonversi dari analog ke digital lalu diolah di dalam komputer. Tujuan pengolahan image ini ada banyak, di antaranya untuk memperbaiki kualitas gambar atau untuk indentifikasi gambar.

Apa itu Python? **Python** adalah bahasa pemrograman yang banyak dipakai di dunia komputer. Kemampuan Python untuk mengolah citra sangatlah terbatas, untuk itu dia perlu import library dari OpenCV. Di sinilah keterkaitan antara OpenCV dan Python.

Contoh penerapan OpenCV dengan Python adalah camera yang dipasang di parkirana yang mampu membaca plat nomor. Plat nomor ini dikonversi dari analog ke digital lalu diolah menjadi karakter sehingga menjadi data yang bisa dijadikan sebagai informasi penting.

Contohnya pula adalah deteksi wajah, mata, mulut, atau hidung dari suatu image atau video. Bisa juga deteksi semua bagian wajah. Berikut ini contoh hasil dari deteksi wajah dan mata:



Intinya, OpenCV bersama Python dimanfaatkan untuk mengolah image atau video (tumpukan frame/image) sesuai dengan tujuan masing-masing yang melibatkan kamera untuk menangkap gambar lalu diolah di komputer.

Sebenarnya, OpenCV bisa digunakan di bahasa C++ dan Java, selain Python, tetapi bahasa yang paling mudah dari ketiga ini adalah Python karena sederhana. Saya sudah mempelajari C++ untuk OpenCV, ternyata cukup rumit kode-kodenya. Maka, Anda beruntung mempelajari Python dan juga beruntung mendapatkan modul ini.



## C. Download dan Instalasi

Awalnya modul ini disusun dari hasil rangkuman buku *OpenCV Computer Vision with Python*, tetapi karena versi OpenCV 2.4.3 sudah tidak ada, maka terpaksa saya menggunakan OpenCV versi lain (yang saya gunakan adalah OpenCV 2.4.11 karena stabil). Untuk itu, software dependensinya juga menyesuaikan, yaitu numpy versi 1.9.1. Awalnya saya menginstal numpy versi numpy 1.6.2 sesuai yang disarankan oleh penyusun buku di atas, tetapi program mengalami eror karena OpenCV yang saya pakai versinya lebih tinggi dari yang disarankan. Masalah terselesaikan dengan menginstal numpy versi 1.9.1.

Masalah kedua muncul lagi, yaitu skrip (kode) yang ada di buku *OpenCV Computer Vision with Python* tidak bisa dijalankan, akhirnya saya memutuskan mencari skrip lain dan mendapatkannya di situs resmi OpenCV.org. Ternyata di versi terbaru ini mengalami beberapa perkembangan, dan dugaan saya hal ini yang menyebabkan sintak lama tidak kompatibel lagi, dan juga, kabarnya, OpenCV 2.4.3 belum stabil dan disempurnakan di versi 2.4.11 ini.

Untuk bisa menjalankan dan memakai modul ini dengan baik, maka Anda membutuhkan 3 (tiga) file utama, yaitu OpenCV, Python, dan Numpy. Berikut versi masing-masing software dan link untuk mendownloadnya:

1. OpenCV 2.4.11 >> <https://nchc.dl.sourceforge.net/project/opencvlibrary/opencv-win/2.4.11/opencv-2.4.11.exe> (351 MB)
2. Python 2.7.3 >> <https://www.python.org/ftp/python/2.7.3/python-2.7.3.msi> (15,1 MB)
3. Numpy 1.9.1 >> <https://nchc.dl.sourceforge.net/project/numpy/NumPy/1.9.1/numpy-1.9.1-win32-superpack-python2.7.exe> (8,52 MB)

Untuk berjaga-jaga, download juga:

4. Scipy 0.11.0 >> <https://nchc.dl.sourceforge.net/project/scipy/scipy/0.11.0/scipy-0.11.0-win32-superpack-python2.7.exe> (45,6 MB)
5. Matplotlib 1.5.0 >> <https://ncu.dl.sourceforge.net/project/matplotlib/matplotlib/matplotlib-1.5.0/windows/matplotlib-1.5.0.win32-py2.7.exe> (5,88 MB)

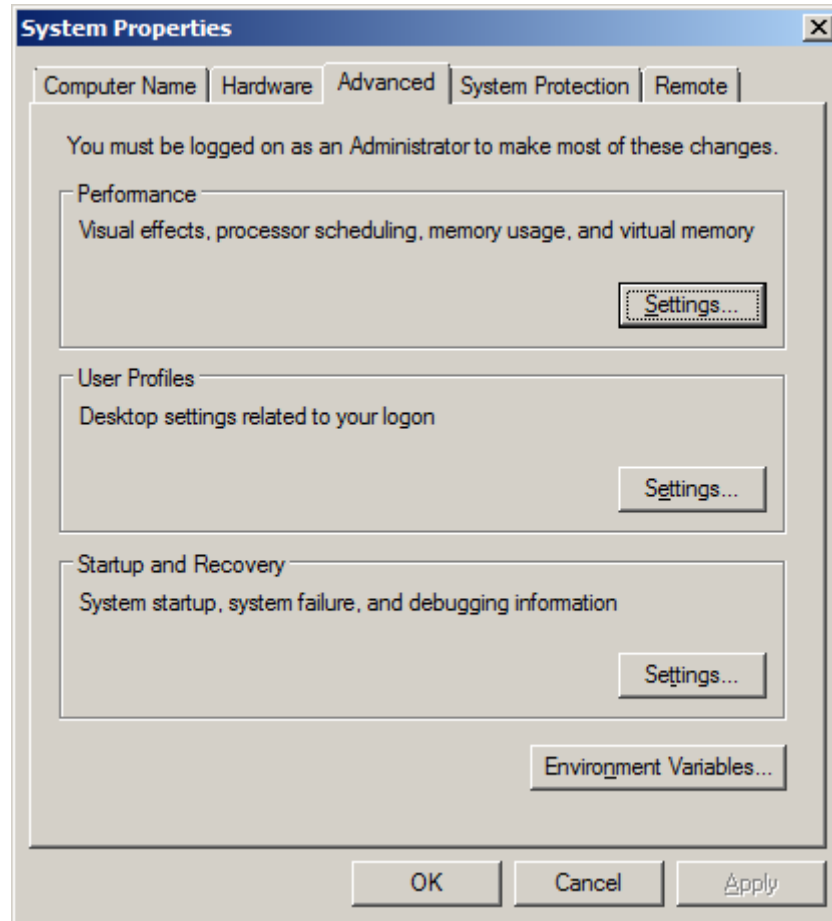
Komputer yang saya gunakan berarsitektur 64 bit Windows 7. Namun, semua software di atas untuk 32 bit (x86/win32).

Setelah didownload semuanya, maka tahapan berikutnya adalah menginstal dengan urutan sebagai berikut:

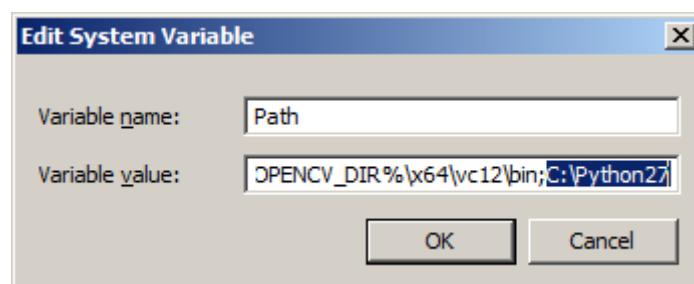
1. Instal OpenCV 2.4.11. Taruh hasil instalasi di drive C atau D atau lainnya, terserah. Maka hasilnya adalah C:/opencv atau D:/opencv. Adapun saya, OpenCV ditaruh di E:/opencv.
2. Instal Python 2.7.3. Secara default akan terinstal di C:/Python27. Setelah selesai, maka copy file cv2.pyd yang ada di "E:\opencv\build\python\2.7\x86\" dan taruh di "C:\Python27\Lib\site-packages\". Hal ini dimaksudkan untuk membuat link antara

OpenCV dengan Python. Setelah itu, daftarkan Python ke Path Environments agar saat file Python dieksekusi maka komputer bisa mengenalinya. Caranya adalah sebagai berikut:

3. Klik Start > klik kanan Computer > Properties > klik Advanced system setting > Advanced > Environment Variables



4. Kemudian, pada System variables > Path dan klik edit > tambahkan “;C:\Python27” di kotak Variables value, seperti gambar berikut:



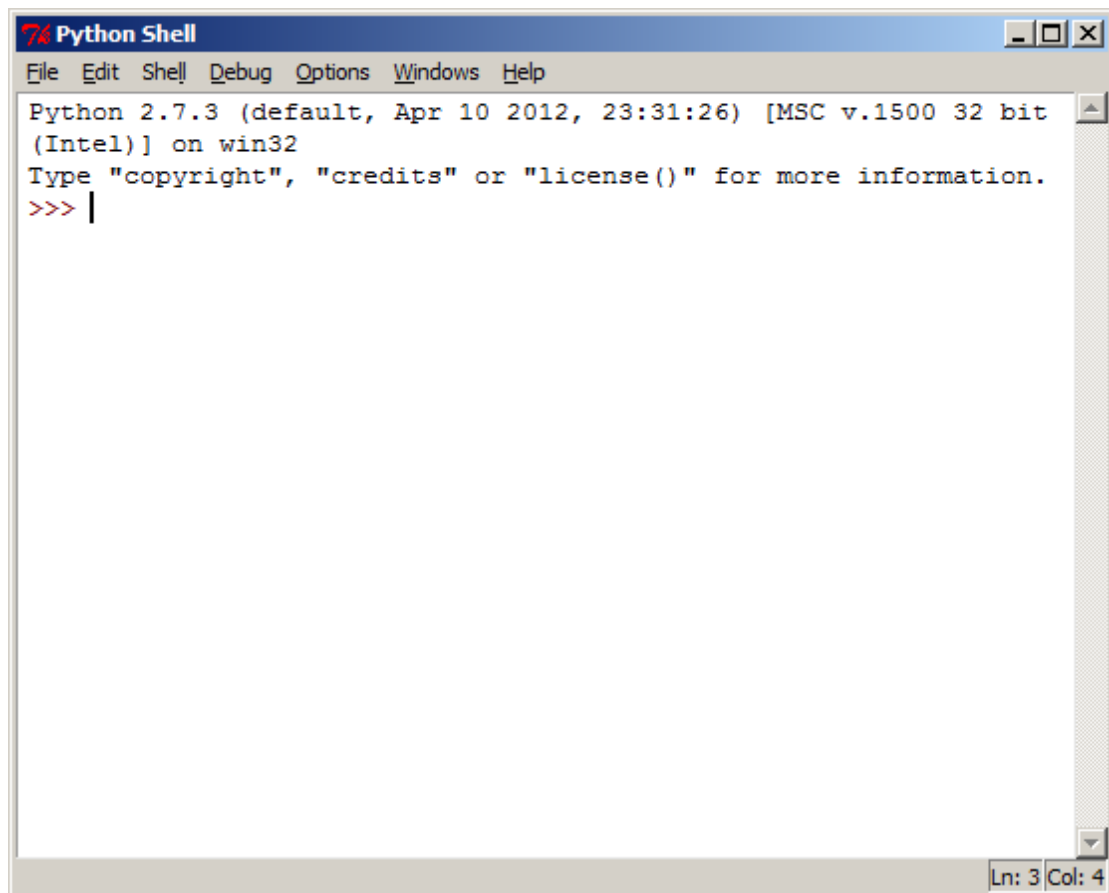
5. Instal Numpy 1.9.1. Biarkan ia secara default. Next dan next saja saat instalasi.

Jika semua tahapan ini sudah Anda lakukan dengan benar, maka sekarang Python yang terintegrasi dengan OpenCV sudah bisa digunakan.

## D. Mengolah Image

Apa itu **image**? Image biasa diartikan citra atau gambar. Image adalah sebutan untuk hasil dari jepretan foto atau kamera yang sudah dikonversi ke digital (berisi angka 0 dan 1). Inilah yang dimaksud image di sini. Contoh image adalah setiap file yang berektensi (berakhiran) \*.jpg dan \*.png.

Silahkan buka Python:

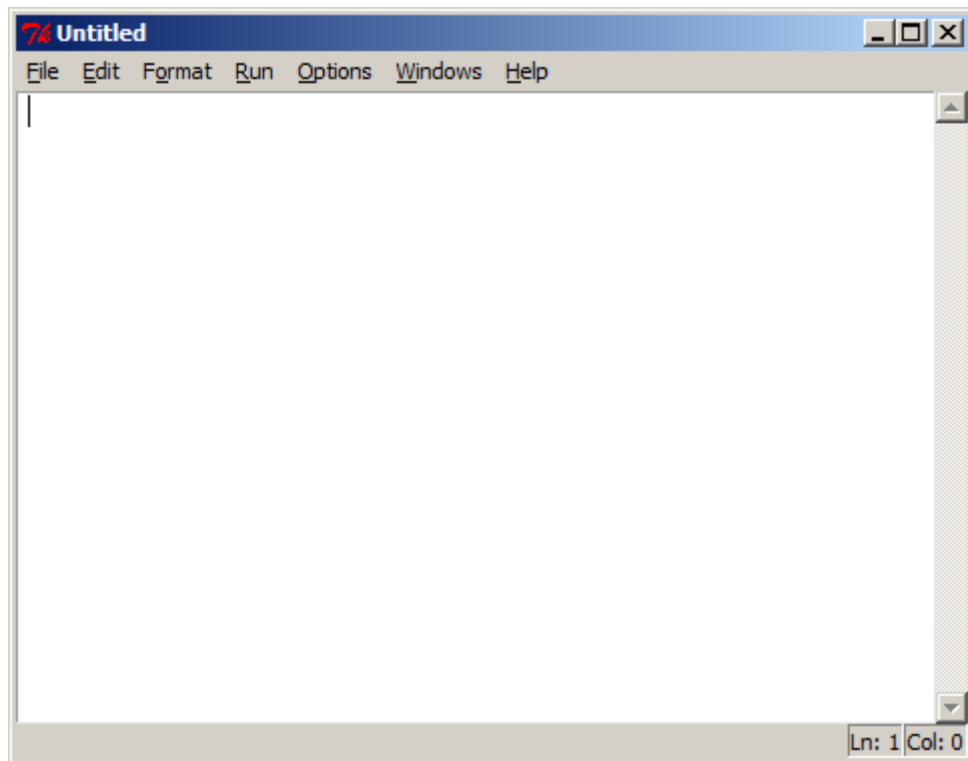


```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7.3 (default, Apr 10 2012, 23:31:26) [MSC v.1500 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```

Ln: 3 Col: 4

Di sinilah kode Python ditulis. Untuk lebih memudahkan dalam praktek, kita buat file dengan cara File > New Window, maka akan terbuka window kosong dan sinilah nanti ditulis kode Python:





Silahkan disave dengan nama, misalnya, **bismillah.py**. Ekstensi py adalah ekstensi untuk Python, sebagaimana cpp adalah ekstensi untuk C++ dan docx adalah ekstensi untuk MS Word. Nama file jangan menggunakan spasi atau karakter khusus seperti # \$ ^ \* & dan seterusnya. Anda bisa menaruh file di C:/Python27 tetapi dianjurkan membuat folder baru (misalnya BelajarPython) sehingga menjadi C:/Python27/BelajarPython. Di sini nanti file di simpan sekaligus menyimpan image dan video dari hasil belajar.

Jika tidak diberi ekstensi \*.py maka kode yang diketik tidak berwarna. Maka jangan lupa untuk menambah \*.py di akhir nama file.

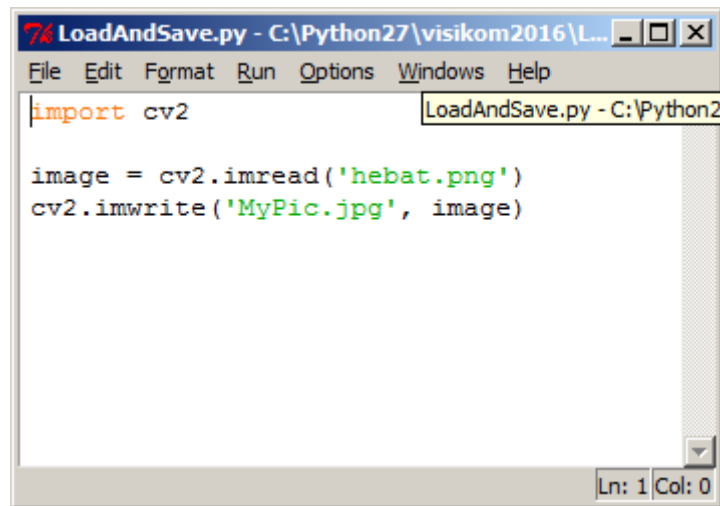
### **Load dan Menyimpang Image**

Read() digunakan untuk membuka image sementara write() digunakan untuk menyimpan image di folder yang sama. Berikut sintaknya:

```
import cv2

image = cv2.imread('hebat.png')
cv2.imwrite('MyPic.jpg', image)
```

Berikut tampilannya:



Kemudian jalankan program ini dengan mengklik **Run > Run Module** atau **F5**. Maka otomatis program akan membuat file MyPic.jpg yang tersimpan di folder yang sama dengan hebat.png

**import cv2** maksudnya Python memanggil library cv2 yaitu OpenCV.

**image = cv2.imread('hebat.png')** maksudnya membaca (load) file 'hebat.png' yang kemudian disimpan di variabel sementara bernama image. Kata sementara ini maksudnya disimpan di memori tidak disimpan di drive komputer. Jika gambar ada di folder lain (misalnya di folder pictures) maka bisa dipanggil dengan sintak "C:\Users\Public\Pictures\Sample Pictures\Koala.jpg"

**cv2.imwrite('MyPic.jpg', image)** maksudnya menyimpan citra yang tersimpan di variabel image tadi, disimpan di drive komputer dengan nama MyPic.jpg. File ini tersimpan di folder yang sama dengan hebat.png.

Note: Jika kode-kode di bawah ini eror (tidak bisa berjalan) silahkan instal Scipy 0.11.0 yang sudah Anda download.

## Konversi Warna Image

Image yang diload bisa dikonversi ke beberapa format, dan format paling umum adalah dari RGB ke GrayScale dan HSV. Apa itu RGB? RGB (Red Green Blue) adalah standar warna suatu image. Semua warna yang dikenal oleh manusia sebenarnya tersusun dari gabungan tiga warna ini. Masing masing warna memiliki intensitas dari 0-255. Artinya semakin tinggi nilai intensitasnya maka warnanya semakin terang.

Dari perpaduan 3 warna ini maka terbentuk warna lainnya. Misalnya warna kuning yang terbentuk dari perpaduan warna R&G dengan komposisi yang sama atau ditulis 255:255:0. Citra RGB terdiri dari 3 channel (kanal) yaitu R, G, B. Masing2 kanal berisi 8 bit intensitas ( $2^8 = 256$  warna), sehingga jumlah warna yang tertampung di citra RGB sebanyak  $3 \times 8$  bit atau 16 juta warna lebih.

GrayScale (skala keabu-abuan) adalah warna perpaduan dari warna hitam dan putih. Citra

gray hanya memiliki satu kanal sehingga hanya berisi 8 bit intensitas atau 256 intensitas warna. Citra Grayscale dibuat dari RGB dengan rumus:  $0,33 \text{ RED} + 0,56 \text{ GREEN} + 0,11 \text{ BLUE}$ .

Sementara HSV (*Hue Saturation Value*) adalah jenis warna lain yang jumlah dari perpaduan warna-warnanya banyak sekali.

Sebenarnya, maka manusia tidak mampu membedakan antara warna kuning asli (255:255:0) dengan kuning jenis lain (misalnya 255:255:4) karena mirip sekali, tetapi bagi komputer hal ini sangat mudah dibedakan karena yang dibaca komputer adalah angka-angkanya.

Ada dua cara untuk konversi dari GRB ke Gray, yaitu dengan sintak

```
grayImage = cv2.imread('MyPic.png', cv2.CV_LOAD_IMAGE_GRAYSCALE)
atau
img =cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Contoh penerapannya adalah:

```
import cv2

image = cv2.imread('hebat.png', cv2.CV_LOAD_IMAGE_GRAYSCALE)
cv2.imwrite('MyPic.jpg', image)
```

Atau

```
import cv2

image = cv2.imread('hebat.png')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imwrite('MyPic.jpg', gray)
```

Konversi warna yang banyak digunakan adalah

1. RGB ke GrayScale > cv2.COLOR\_BGR2GRAY
2. RGB ke HSV > cv2.COLOR\_BGR2HSV

## Menampilkan Image

Untuk menampilkan image menggunakan sintak:

```
cv2.imshow('aa',image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Contoh penerapan:

```
import cv2

image = cv2.imread('hebat.png')
cv2.imshow('Tampil Image', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

**cv2.imshow('Tampil Image', image)** maksudnya menampilkan citra yang tersimpan di variabel image dan diletakkan di window dengan nama Tampil Image.

**cv2.waitKey(0)** maksudnya window ditahan (tidak menutup sendiri) hingga user menekan salah satu tombol di keyboard (misal tombol ESC atau space). Jika user menekan tombol (ESC misalnya) maka window baru menutup.

**cv2.destroyAllWindows()** maksudnya semua window yang terbuka (jika membuka lebih dari window) maka semuanya akan tertutup saat menekan tombol di keyboard. `cv2.waitKey(0)` dan `cv2.destroyAllWindows()` harus disertakan setiap kali menggunakan `cv2.imshow()`. Jika tidak maka window tidak bisa ditutup.

### Ambil Image dari WebCam

WebCam adalah kamera yang terpasang di komputer. Hampir semua laptop memiliki WebCam. Jika belum terdetek, mungkin belum menginstal drivernya. Jika webcam sudah bisa dijalankan maka biasanya akan muncul lampu biru kecil sebagai tanda webcam menyala. Sintak umum menangkap image dari webcam adalah:

```
cap = cv2.VideoCapture(0)

ret, frame=cap.read()
```

Contoh penerapannya adalah:

```
import cv2
```

```
cap = cv2.VideoCapture(0)
ret,frame=cap.read()
cv2.imwrite('a.png',frame)
cv2.imshow('frame',frame)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

**cap = cv2.VideoCapture(0)** maksudnya mengaktifkan kamera Webcam dan melakukan capture (menangkap image) objek di depannya di kali pertama dia hidup. Angka nol menunjukkan webcame utama di laptop. Jika webcame ada dua, maka untuk mengaktifkan webcam kedua, menggunakan angka 1. Image yang tercapture ini disimpan sementara di variabel cap.

**ret,frame=cap.read()** adalah sintak wajib yang menyertai VideoCapture(). Tanpanya VideoCapture() tidak bisa berjalan. Maksud frame di sini adalah image. Image disebut frame karena ia diibaratkan sebagai frame (bingkai). Image yang tersimpan di variabel cap tadi, diload dan disimpan di variabel frame agar bisa diolah.

## E. Mengolah Video

Video adalah gambar bergerak yang ditampilkan di komputer yang meniru pergerakan objek asli. Sebenarnya video adalah tumpukan dari banyak frame (image). Jika sebuah image di tumpuk dengan diberi delay tertentu (jeda waktu tertentu sekian mili detik) maka seolah-olah bergerak. Itulah video.

### Play Video dari WebCam

Maksud play video dengan webcam adalah mengaktifkan webcam lalu menampilkan objek yang ditangkap webcam di komputer. Berikut sintaknya:

```
cap = cv2.VideoCapture(0)

while(True):
    ret, frame = cap.read()
    cv2.imshow('Hasil Capture', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Contoh penerapannya dengan melibatkan simpan capture dan menampilkannya saat menutup webcam:

```
import cv2

cap = cv2.VideoCapture(0)

while(True):
    # Tangkap frame per frame
    ret, frame = cap.read()

    # RGB diubah ke GrayScale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```



```

# Gambar disimpan dengan nama 'tangkap.png'

cv2.imwrite('tangkapRGB.png',frame)
cv2.imwrite('tangkapGray.png',gray)

# Citra ditampilkan per frame
cv2.imshow('frame',gray)

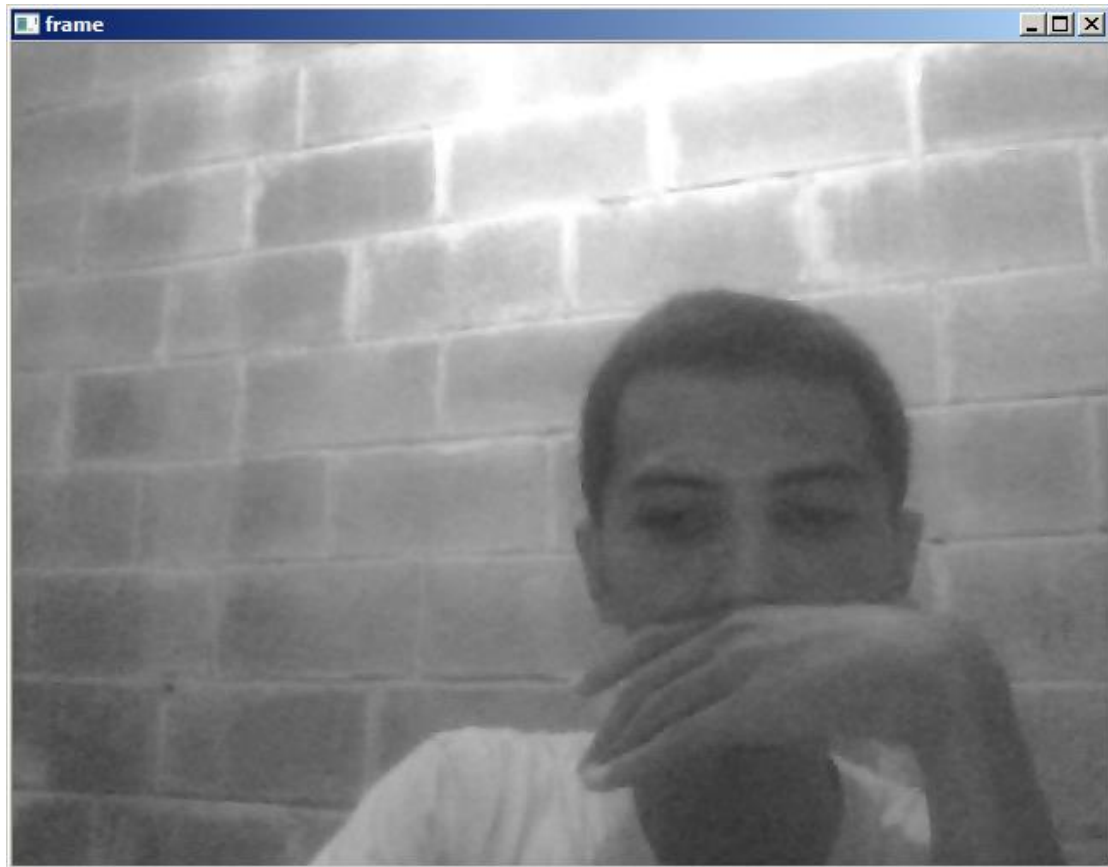
# Tutup Webcam dengan tombol ESC atau q
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Jalankan capture
cap.release()
cv2.destroyAllWindows()

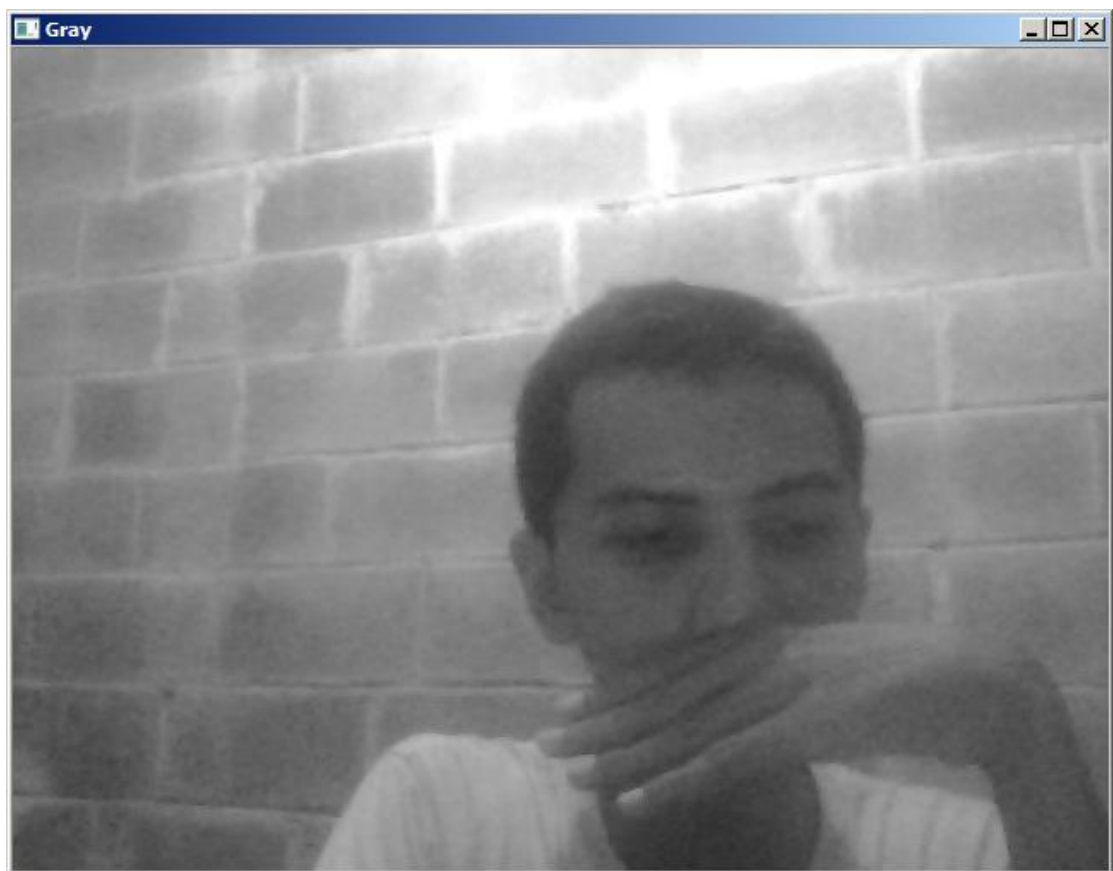
#Tampilkan hasil capture setelah webcam ditutup
hasilRGB=cv2.imread('tangkapRGB.png')
hasilGray=cv2.imread('tangkapGray.png')
cv2.imshow('RGB',hasilRGB)
cv2.imshow('Gray',hasilGray)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Cara kerjanya, program menyalakan Webcam lalu mencapture (menangkap) objek yang ada di depannya dan disimpan dalam frame. Dengan pengulangan while maka objek didepannya ditangkap terus satu demi satu. Dari tumpukan frame inilah maka menjadi gambar bergerak (video). Frame ini lalu ditampilkan dengan bantuan imshow() dengan while.



Jika user menekan tombol ESC atau q maka webcam mati. Kemudian, frame yang tersimpan terakhir ditampilkan dengan bantuan `imshow()` untuk hasil RGB dan Gray.



## Play Video dari File

File yang saya gunakan berformat wmv (juga menerima format lainnya seperti mp4 dan avi), yaitu file video bawaan Window 7 yang terdapat di C:\Users\Public\Videos\Sample Videos\Wildlife.wmv.

Untuk bisa menjalankan video, diperlukan sedikit pengaturan, yaitu: copy file **opencv\_ffmpeg2411.dll** yang terdapat di **E:\opencv\build\x86\vc12\bin\** lalu taruh di **C:\Python27**.

Sintak yang digunakan sama persis dengan Play Video dengan Webcam, bedanya hanya di bagian ini:

```
cap = cv2.VideoCapture(0)

diganti dengan

cap = cv2.VideoCapture('C:\Users\Public\Videos\Sample
Videos\Wildlife.wma')
```

Contoh penerapannya adalah:

```
import cv2

cap = cv2.VideoCapture('C:\Users\Writer\Videos\Wildlife.mp4')

while(True):
    # Tangkap frame per frame
    ret, frame = cap.read()

    # RGB diubah ke GrayScale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Gambar disimpan dengan nama 'tangkap.png'
    cv2.imwrite('tangkapGray.png',gray)

    # Citra ditampilkan per frame
```

```
cv2.imshow('frame',gray)

# Tutup Webcam dengan tombol ESC atau q
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Jalankan capture
cap.release()
cv2.destroyAllWindows()
```

Video yang diplay mengalami percepatan gerak. Untuk mengurai percepatan ini bisa digunakan `imread()` dengan asumsi program anak menyimpan image di komputer lalu ditindih image dari frame berikutnya sehingga mengurangi percepatan gerak video. Namun, terkadang sebuah video memiliki bitrate yang besar sehingga bisa berjalan dengan normal, seperti kasus video di atas. Jika Anda memainkan video kecil maka akan terasa percepatannya. Sebenarnya hal ini bisa diatasi dengan sintak khusus tetapi saya sendiri belum sempat mencarinya. Berikut ini hasilnya:



### Save Video dari Webcam dan File

Video yang dicapture oleh Webcam atau yang berasal dari file, bisa disimpan di drive komputer dengan hasil pengolahan tertentu, grayscale misalnya. Berikut sintak umumnya:

```

import cv2

cap = cv2.VideoCapture(0)

# Definisikan codex (format video)
fourcc = cv2.cv.CV_FOURCC(*'XVID')

# cv2.VideoWriter(['nama file',[nilai fourcc],[nilai fps],[nilai
lebar,panjang])
out = cv2.VideoWriter('output.avi', fourcc, 5, (640,480))

while(cap.isOpened()):
    ret, frame = cap.read()
    if ret==True:
        out.write(frame)

        # Tampilkan hasil capture webcam
        cv2.imshow('Hasil Capture',frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break

# Jalankan capture
cap.release()
out.release()
cv2.destroyAllWindows()

jika berupa file dari drive, maka hanya mengganti cap =
cv2.VideoCapture(0) dengan cap =
cv2.VideoCapture('address_file.avi')

```



**fourcc = cv2.cv.CV\_FOURCC(\*'XVID')** digunakan untuk format file dan XVID adalah standar format. Untuk video dari file, jika XVID ditulis DIVX maka video terbaca berkebalikan arah, yang sebelah kanan berada di sebelah kiri, dan sebaliknya. Berikut jenis-jenis format beserta sintaknya:

- `cv2.cv.CV_FOURCC('I','4','2','0')`: This is an uncompressed YUV, 4:2:0 chroma subsampled. This encoding is widely compatible but produces large files. The file extension should be `avi`.
- `cv2.cv.CV_FOURCC('P','I','M','1')`: This is MPEG-1. The file extension should be `avi`.
- `cv2.cv.CV_FOURCC('M','J','P','G')`: This is motion-JPEG. The file extension should be `avi`.
- `cv2.cv.CV_FOURCC('T','H','E','O')`: This is Ogg-Vorbis. The file extension should be `ogv`.
- `cv2.cv.CV_FOURCC('F','L','V','1')`: This is Flash video. The file extension should be `flv`.

**fps (frame per second)** maksudnya kecepatan pergantian dari satu frame ke frame lainnya. Semakin besar nilainya, maka video semakin cepat. Nilai normal adalah 5.

Contoh penggunaannya adalah:

```
import cv2

cap = cv2.VideoCapture(0)

# Definisikan codex (format video)
fourcc = cv2.cv.CV_FOURCC(*'DIVX')

# cv2.VideoWriter(['nama file',[nilai fourcc],[nilai fps],[nilai lebar,panjang])
out = cv2.VideoWriter('output.avi', fourcc, 5, (640,480))

raw_input('Tekan enter untuk mulai dan esc untuk saving dan keluar')
while(cap.isOpened()):
    ret, frame = cap.read()
    if ret==True:
```

```

        # Mengatur derajat putar
        frame = cv2.flip(frame,180)

        # Mencetak frame ke out
        out.write(frame)

        # Tampilkan hasil capture webcam
        cv2.imshow('Hasil Capture',frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break

# Jalankan capture
cap.release()
out.release()
cv2.destroyAllWindows()

###PLAY (Copas dari file PlayVideo_File.py)
#import cv2

cap = cv2.VideoCapture('output.avi')

while(True):
    # Tangkap frame per frame
    ret, frame = cap.read()

    # RGB diubah ke GrayScale

```

```

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# Gambar disimpan dengan nama 'tangkap.png'
cv2.imwrite('tangkapGray.png',gray)

# Citra ditampilkan per frame
cv2.imshow('Hasil Tangkap Frame Hasilkan Video',gray)

# Tutup Webcam dengan tombol ESC atau q
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Jalankan capture
cap.release()
cv2.destroyAllWindows()

```

**raw\_input('Tekan enter untuk mulai dan esc untuk saving dan keluar')** digunakan untuk menampilkan kalimat di command prompt (cmd) dan program dimulai dengan menekan enter.

**frame = cv2.flip(frame,180)** digunakan untuk memutar video. Jika nilainya 0 maka video akan terbalik, yang bawah menjadi atas dan yang atas menjadi bawah.

Cara kerjanya, program menyalakan Webcam lalu mencapture (menangkap) objek yang ada di depannya dan disimpan dalam frame. Dengan pengulangan while maka objek didepannya ditangkap terus satu demi satu. Dari tumpukan frame inilah maka menjadi gambar bergerak (video). Frame ini lalu ditampilkan dengan bantuan imshow() dengan while. Ketika user menekan tombol q maka Webcam mati dan program kemudian memainkan output.avi.

### **Realtime untuk Image dan Video**

Realtime (waktu nyata) adalah istilah untuk merekam suatu objek dan menampilkannya langsung. Maka apa yang tampil adalah apa terekam saat itu juga. Tentunya apa yang ditampilkan ini sudah mengalami pengolahan, warnanya grayscale atau wajah terdeteksi dengan kotak-kotak misalnya.

Pada latihan kali ini, kita akan memanfaatkan semua teori yang sudah kita pelajari yaitu

imread(), imwrite(), imshow(), cvtColor(), VideoCapture(), dan lain-lain. Langsung saja, perhatikan skrip di bawah ini:

```
import cv2

cap = cv2.VideoCapture(0)

while(True):
    ret, frame = cap.read()

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    #Canny
    canny = cv2.Canny(frame,100,200)

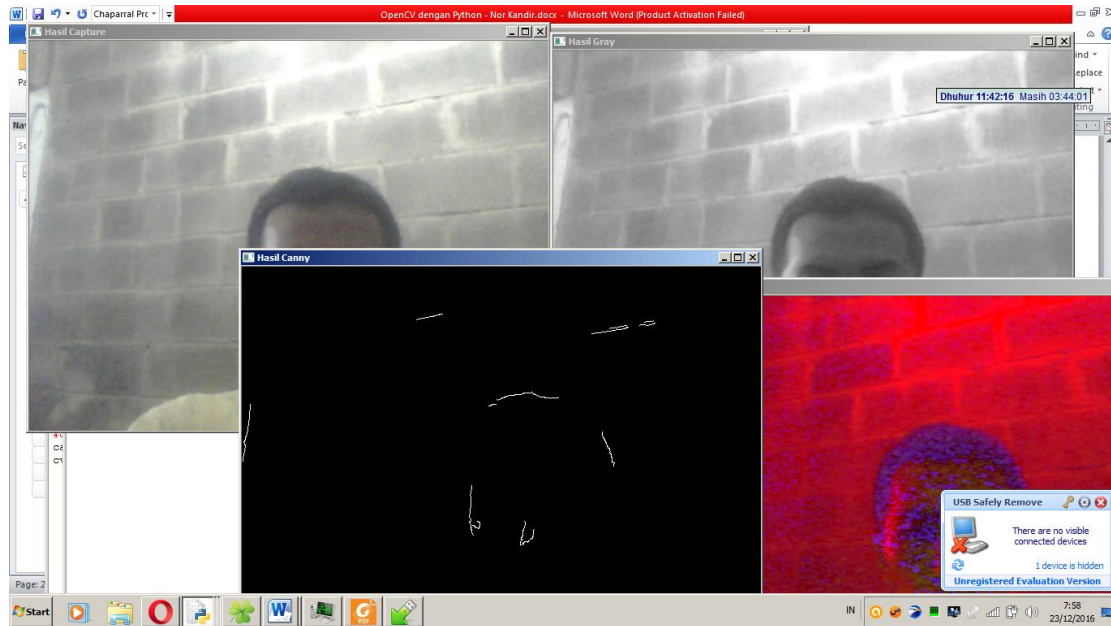
    #Tampilkan semua capture
    cv2.imshow('Hasil Capture', frame)
    cv2.imshow('Hasil Gray',gray)
    cv2.imshow('Hasil HSV',hsv)
    cv2.imshow('Hasil Canny',canny)

    #Simpan capture
    cv2.imwrite('realtimeRGB.jpg',frame)
    cv2.imwrite('realtimeGray.jpg',gray)
    cv2.imwrite('realtimeHSV.jpg',hsv)
    cv2.imwrite('realtimeCanny.jpg',canny)

    #Tutup window saat ditekan tombol "q"
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

```
#Jalankan capture  
cap.release()  
cv2.destroyAllWindows()
```

Jika program ini dirun, maka hasilnya sebagai berikut:



## F. Deteksi Wajah dengan Haar Cascade

Library OpenCV yang menangani deteksi wajah adalah Haar Cascades. Ada empat file utama di OpenCV yang menangani deteksi ini, yaitu:

1. haarcascade\_frontalface\_alt.xml > deteksi wajah
2. haarcascade\_eye.xml > deteksi mata
3. haarcascade\_mcs\_nose.xml > deteksi hidung
4. haarcascade\_mcs\_mouth.xml > deteksi mulut

Sintak umumnya adalah sebagai berikut:

```
import cv2

face_cascade = cv2.CascadeClassifier('E:\opencv\sources\data\haarcascades\haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('E:\opencv\sources\data\haarcascades\haarcascade_eye.xml')

img = cv2.imread('foto.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

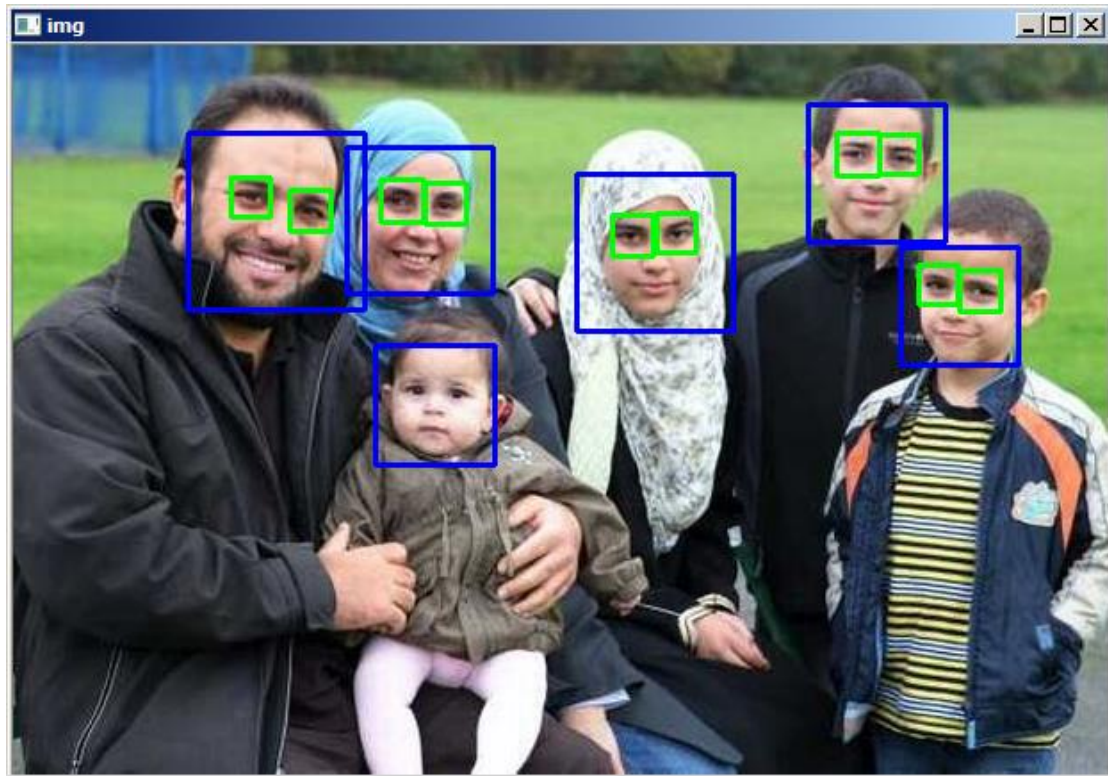
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
for (x,y,w,h) in faces:
    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

cv2.imshow('img',img)
cv2.waitKey(0)
```



```
cv2.destroyAllWindows()
```

Jika program ini dijalankan maka hasilnya adalah sebagai berikut:



Contoh penerapan untuk deteksi wajah yang mencakup mata, hidung, dan mulut. Blok berwarna biru untuk deteksi wajah, blok biru untuk mata, blok merah untuk hidung, dan blok hitam untuk mulut. Berikut sintaknya:

```
import cv2

face_cascade = cv2.CascadeClassifier('E:\opencv\sources\data\haarcascades\haarcascade_frontalface_default.xml')

eye_cascade = cv2.CascadeClassifier('E:\opencv\sources\data\haarcascades\haarcascade_eye.xml')

mouth_cascade = cv2.CascadeClassifier('E:\opencv\sources\data\haarcascades\haarcascade_mcs_mouth.xml')

nose_cascade = cv2.CascadeClassifier('E:\opencv\sources\data\haarcascades\haarcascade_nose.xml')
```

```

de_mcs_nose.xml')

img = cv2.imread('foto.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = face_cascade.detectMultiScale(gray, 1.3, 5)

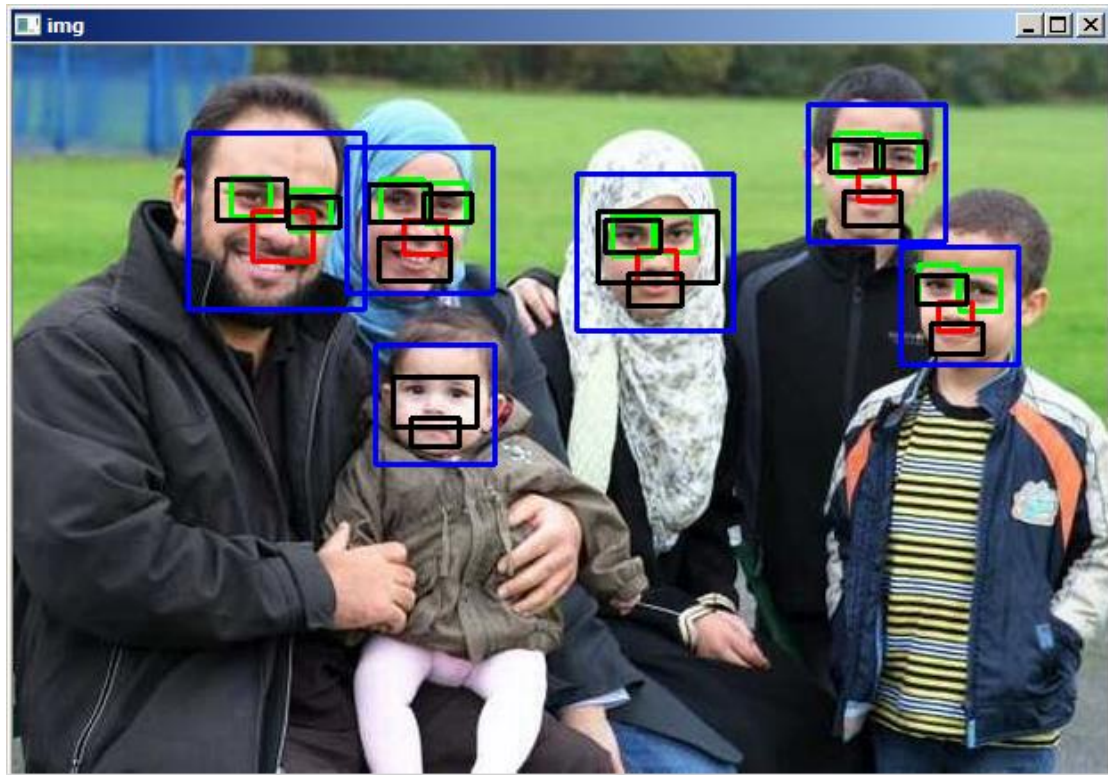
for (x,y,w,h) in faces:
    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    nose = nose_cascade.detectMultiScale(roi_gray)
    mouth = mouth_cascade.detectMultiScale(roi_gray)

    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
    for (nx, ny, nw, nh) in nose:
        cv2.rectangle(roi_color, (nx, ny), (nx + nw, ny + nh),
(0, 0, 255), 2)
    for (mx, my, mw, mh) in mouth:
        cv2.rectangle(roi_color, (mx, my), (mx + mw, my + mh),
(0, 0, 0), 2)

cv2.imshow('img',img)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Jika program dirun maka hasilnya adalah:



### Realtime untuk Harr Cascade

Berikut adalah sintaknya:

```
import cv2

face_cascade = cv2.CascadeClassifier('E:\opencv\sources\data\haarcascades\haarcascade_frontalface_default.xml')

eye_cascade = cv2.CascadeClassifier('E:\opencv\sources\data\haarcascades\haarcascade_eye.xml')

cap = cv2.VideoCapture(0)

while(True):
    # Tangkap frame per frame
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```

faces = face_cascade.detectMultiScale(gray, 1.3, 5)
for (x,y,w,h) in faces:
    cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = frame[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

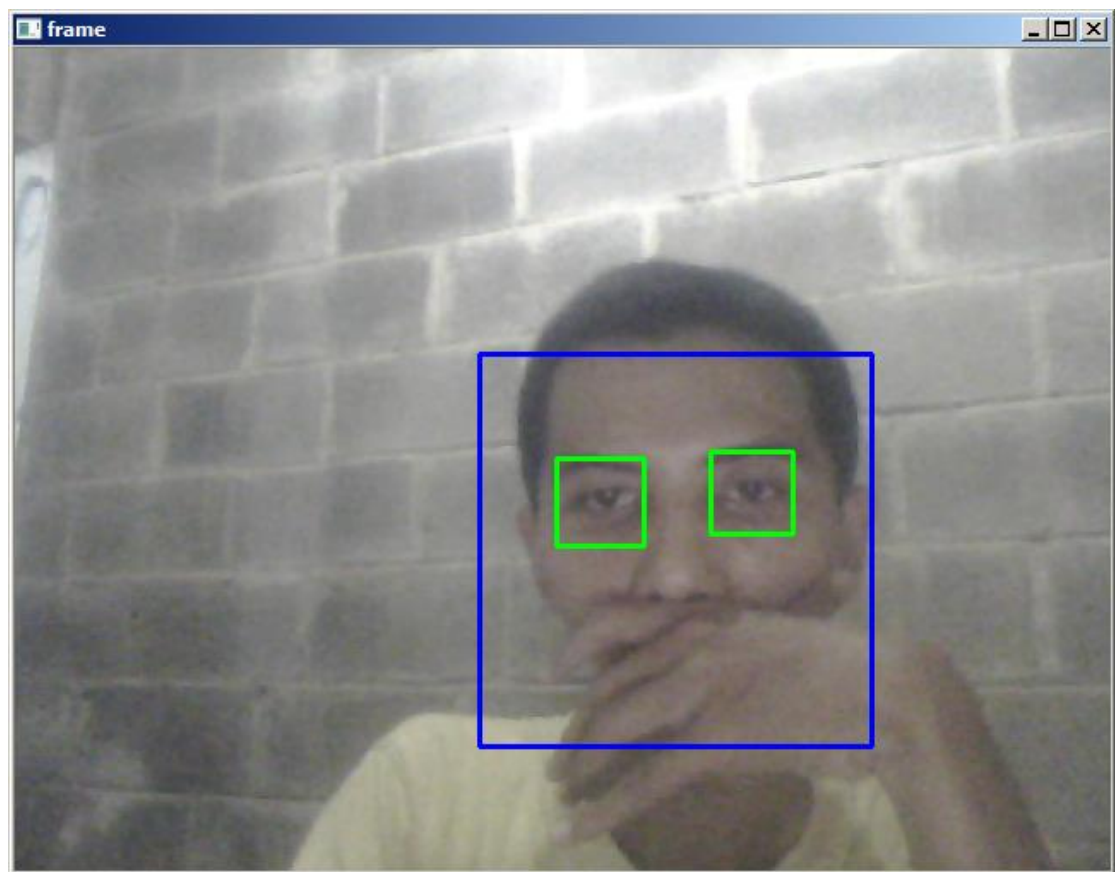
# Citra ditampilkan per frame
cv2.imshow('frame',frame)

# Tutup Webcam dengan tombol ESC atau q
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Jalankan capture
cap.release()
cv2.destroyAllWindows()

```

Haar yang saya gunakan di realtime ini hanya untuk deteksi wajah dan mata. Jika Anda menyertakan hidung dan mulut, silahkan tinggal ditambahkan saja kodenya di atas. Tentunya dengan memahami dahulu sintaksnya. Sintak di atas sudah saya uji coba dan berhasil untuk mendeteksi satu wajah atau lebih. Untuk mengoptimalkan deteksi maka kecerahan layar laptop agar dimaksimalkan. Ini hasilnya:



## **G. Kesimpulan**

1. OpenCV adalah library yang digunakan oleh Python untuk mengolah image dan video.
2. Mengolah image ada beberapa macam yaitu membaca, menyimpan, mengkonversi, menampilkan, dan menangkap image lewat webcam.
3. Webcam bisa mendeteksi wajah beserta mata, hidung, dan mulut dengan bantuan OpenCV dan Python.



## H. Daftar Pustaka

1. OpenCV Computer Vision With Python karya Joseph Howse
2. [http://docs.opencv.org/trunk/d2/d96/tutorial\\_py\\_table\\_of\\_contents\\_imgproc.html](http://docs.opencv.org/trunk/d2/d96/tutorial_py_table_of_contents_imgproc.html)
3. [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_tutorials.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html)
4. Eksperimen sendiri

## **I. Penulis**

Penulis bernama Nor Kandır, alumni Ponpes Raudhlatul Ulum Guyangan Trangkil Pati yang kuliah di ITS Surabaya jurusan Teknik Elektro bidang IT. Modul Python ini disusun untuk memenuhi tugas mata kuliah Visikom. Penulis tertarik di dunia tulis menulis terutama yang berhubungan dengan ilmu agama. Situs penulis bisa dikunjungi di <https://norkandirblog.wordpress.com>. Semoga bermanfaat.[]