

# **UNIVERSITAS GUNADARMA**



## **PRAKTIKUM PENGANTAR KECERDASAN BUATAN**

### **MANUAL BOOK**

#### **“Face and eye Detection Using Opencv and Python3”**

**Nama** : M. Kizbudin, Nanda Tri Setyadi, Veri Irvan

**NPM** : 53417413, 54417423, 56417079

**Fakultas** : Teknologi Industri

**Jurusan** : Teknik Informatika

**Penanggung Jawab** :

**Ditulis Guna Melengkapi Sebagian Syarat**

**Praktikum Pengantar Kecerdasan Buatan Jenjang S1**

**Universitas Gunadarma**

**2020**

## DAFTAR ISI

Daftar Isi.....	2
BAB 1 : Pendahuluan.....	3
BAB 2 : Pembahasan.....	5
BAB 3 : Perancangan dan Implementasi .....	8
LAMPIRAN.....	9
Daftar Pustaka.....	13

# **BAB I**

## **PENDAHULUAN**

### **1. Latar Belakang**

Seiring dengan berkembangnya teknologi, maka pengaplikasian deteksi wajah mengalami pengembangan yang sangat cepat dan semakin banyak digunakan, contohnya untuk sistem keamanan, absensi dan lainnya. Deteksi wajah (face detection) merupakan salah satu tahap awal yang sangat penting sebelum dilakukan proses pengenalan wajah (face recognition)[1]. Sistem pengenalan wajah akan digunakan untuk membandingkan citra wajah masukan dengan suatu kumpulan wajah sehingga dapat mengenali mana wajah yang paling cocok dengan citra wajah tersebut.

Dari penelitian ini diharapkan dapat menguji perubahan transformasi pada suatu area wajah. Selanjutnya pengaplikasian deteksi wajah dapat digunakan untuk monitoring pada suatu ruangan. Telah di ketahui bersama bahwa sistem monitoring yang sekarang ini masih jarang digunakan. Untuk itu deteksi wajah dapat dimanfaatkan untuk monitoring keberadaan seseorang pada suatu ruangan, dimana monitoring ruangan dengan deteksi wajah dapat mempermudah manusia dalam hal mengetahui keberadaan seseorang didalam ruangan tanpa memasuki suatu ruangan. Dengan pemanfaatan deteksi wajah ini akan memonitoring setiap objek pada suatu ruangan dengan menggunakan kamera misalnya, yang selanjutnya akan diproses dengan metoda deteksi wajah yang telah ada untuk memberikan informasi apakah objek tersebut berada didalam ruangan atau tidak.

Untuk deteksi wajah itu sendiri telah banyak dikembangkan dengan beberapa metode yang telah ada, pada penelitian sebelumnya deteksi wajah berbasis metode template matching[9], dan metode Gabor Wavelet[10]. Akan tetapi berbeda dengan penelitian sebelumnya, pada penelitian ini akan menggunakan metode Speed-Up Robust Features (SURF). Metoda SURF merupakan sebuah metode deteksi fitur yang menggunakan keypoint dari sebuah gambar, keypoint itu sendiri adalah bagian-bagian dari sebuah gambar yang nilainya tetap ketika mengalami perubahan skala, rotasi, blurring, transformasi 3dimensi dan pencahayaan. Salah satu tujuan penelitian deteksi wajah dengan metoda SURF ini diharapkan agar dapat menganalisa wajah yang tidak utuh pada posisi derajat tertentu berdasarkan transformasi rotasi, skala, dan perubahan dimensi dengan dilakukan kalibrasi posisi wajah terlebih dahulu.

## **2. Tujuan**

Tujuan dari pembuatan tugas akhir praktikum Pengantar Kecerdasan Buatan ini adalah untuk membandingkan citra wajah dan mata

## **BAB II**

### **PEMBAHASAN**

#### **Pengenalan Wajah**

Pengenalan wajah adalah salah satu teknologi biometrik yang telah banyak diaplikasikan dalam system security selain pengenalan retina mata, pengenalan sidik jari dan iris mata. Dalam aplikasinya sendiri pengenalan wajah menggunakan sebuah kamera untuk menangkap wajah seseorang kemudian dibandingkan dengan wajah yang sebelumnya telah disimpan di dalam database tertentu. Ada beberapa macam metoda pengenalan wajah yaitu neural network, jaringan syaraf tiruan, neuro fuzzy adaptif dan eigenface. Secara khusus dalam Makalah ini metoda yang akan dijelaskan adalah metoda eigenface.

#### **Teknologi Biometrik**

Teknologi biometrik adalah metode otomatis untuk mengidentifikasi manusia berdasarkan beberapa karakteristik biologis dan kebiasaan. Sistem pengenalan wajah adalah salah satu jenis sistem pada teknologi biometrik . Pemodelan komponen wajah pada tingkatan semantik membantu mengungkapkan bagaimana komponen individu berperan dalam pengenalan wajah. Hal ini mewujudkan penangkapan konfigurasi dalam menentukan perbedaan komponen wajah untuk menentukan bobot lokal dari komponen wajah serta memisahkan komponen internal dan eksternal wajah.

## **BAB III**

### **PERANCANGAN DAN IMPLEMENTASI**

#### **1. Perancangan**

Pada praktikum perancangan kecerdasan buatan tentang Pengenalan wajah ada berbagai macam hal yang dibutuhkan, yaitu :

##### **a. Python version 3.0**

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar.

Python mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi.

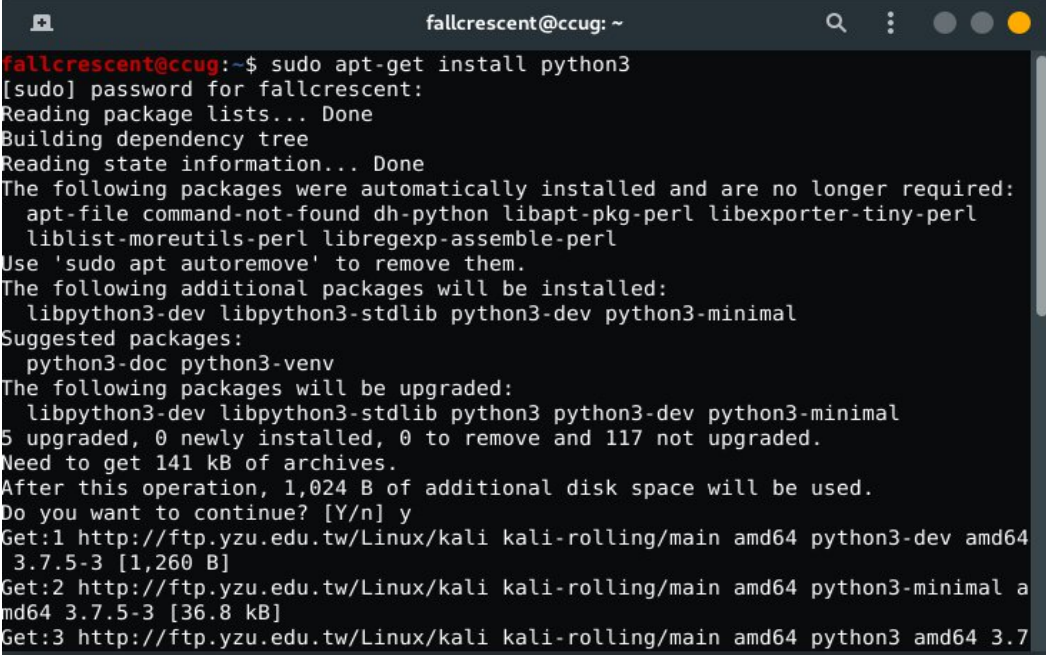
## **b. OpenCV**

**OpenCV** (*Open Source Computer Vision Library*) adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara real-time, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. Program ini bebas dan berada dalam naungan sumber terbuka dari lisensi BSD. Pustaka ini merupakan pustaka lintas platform. Program ini didedikasikan sebagian besar untuk pengolahan citra secara real-time. Jika pustaka ini menemukan pustaka Integrated Performance Primitives dari intel dalam sistem komputer, maka program ini akan menggunakan rutin ini untuk mempercepat proses kerja program ini secara otomatis.

# Implementasi

## a. Install Python

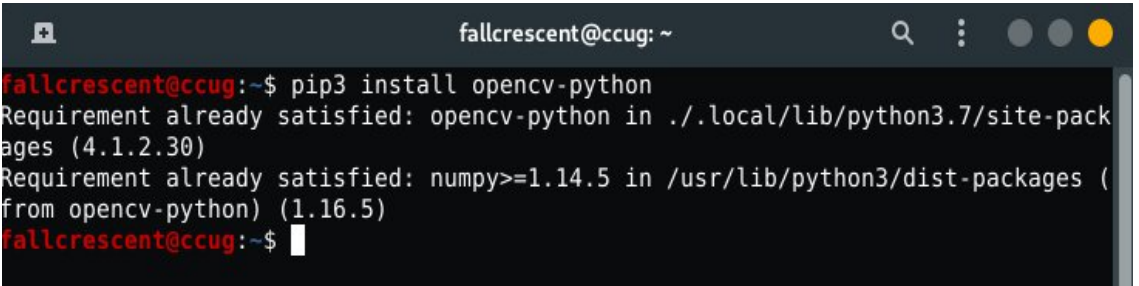
Install Python3 :



```
fallcrescent@ccug: ~  
fallcrescent@ccug:~$ sudo apt-get install python3  
[sudo] password for fallcrescent:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  apt-file command-not-found dh-python libapt-pkg-perl libexporter-tiny-perl  
  liblist-moreutils-perl libregexp-assemble-perl  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  libpython3-dev libpython3-stdlib python3-dev python3-minimal  
Suggested packages:  
  python3-doc python3-venv  
The following packages will be upgraded:  
  libpython3-dev libpython3-stdlib python3 python3-dev python3-minimal  
5 upgraded, 0 newly installed, 0 to remove and 117 not upgraded.  
Need to get 141 kB of archives.  
After this operation, 1,024 B of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 python3-dev amd64  
  3.7.5-3 [1,260 B]  
Get:2 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 python3-minimal a  
md64 3.7.5-3 [36.8 kB]  
Get:3 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 python3 amd64 3.7
```

## b. Install OpenCV

OpenCV diperlukan untuk membuat codingan secara offline. Untuk download dengan melakukan perintah ini di terminal “pip3 install opencv-python,



```
fallcrescent@ccug: ~  
fallcrescent@ccug:~$ pip3 install opencv-python  
Requirement already satisfied: opencv-python in ~/.local/lib/python3.7/site-pack  
ages (4.1.2.30)  
Requirement already satisfied: numpy>=1.14.5 in /usr/lib/python3/dist-packages (  
from opencv-python) (1.16.5)  
fallcrescent@ccug:~$
```



# LAMPIRAN

## Kodingan program

```
1 import cv2
2
3 face = cv2.CascadeClassifier('face.xml')
4 eye = cv2.CascadeClassifier('eye.xml')
5
6 video = cv2.VideoCapture(0)
7
8 while True:
9     _, frame = video.read()
10    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
11
12    muka = face.detectMultiScale(gray, 1.3, 5)
13    for (x,y,w,h) in muka:
14        cv2.rectangle(frame, (x,y),(x+w, y+h), (0,255,0),4)
15
16        roi_warna = frame[y:y+h, x:x+w]
17        roi_gray = frame[y:y + h, x:x+w]
18        mata = eye.detectMultiScale(roi_gray, 3, 3)
19        for (mx,my,mw, mh) in mata:
20            cv2.rectangle(roi_warna, (mx, my), (mx+mw, my+mh), (255,255, 0), 3)
21    cv2.imshow('Face Detection', frame)
22    exit=cv2.waitKey(1) & 0xff
23
24
25 cv2.destroyAllWindows()
26 video.release()
```

## Penjelasan program

```
ai.py > ...  
1  import cv2  
2  
3  face = cv2.CascadeClassifier('face.xml')  
4  eye = cv2.CascadeClassifier('eye.xml')  
5  
6  video = cv2.VideoCapture(0)
```

Maksud dari code diatas adalah kita memasukkan gambar kita dan juga file kita yang sebagai detector atau pendeteksi kedalam variabel.

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

sebelumnya, ubah terlebih dahulu gambar awal menjadi hitam putih. Hal ini dilakukan karena sebuah foto dan juga video lebih mudah diproses menggunakan algoritma dalam keadaan hitam putih, Untuk membuat gambar hitam putih, gunakan `gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)`. Code ini akan mengubah gambar menjadi hitam putih yang terletak dalam variabel `gray`

```
12  muka = face.detectMultiScale(gray, 1.3, 5)
```

kode ini menerapkan algoritma dari face kedalam `gray`, yang merupakan versi hitam putih dari gambar. Kemudian 2 parameter terakhir berfungsi sebagai angka keakuratan dari algoritma yang kami gunakan, Kita dapat mengubahnya dan menyesuaikan dengan sesuka hati. Tetapi umumnya digunakan angka 1.3 dan juga 5. variabel `muka` tersebut akan menghasilkan titik X muka, titik Y muka, lebar muka dan juga tinggi muka.

```

13     for (x,y,w,h) in muka:
14         cv2.rectangle(frame, (x,y),(x+w, y+h), (0,255,0),4)

```

Untuk menggambar sebuah kotak di muka yang terdeteksi maka gunakan for loop untuk setiap variabel dalam muka, Nah setelah setiap variabel muka terambil kita akan gambar kotak pada wilayah muka tersebut. OpenCV memiliki banyak fitur Built-In untuk menggambar diatas gambar dan juga video. Tapi kali ini kita gunakan salah satu saja, yaitu rectangle method untuk menggambar kotak pada area wajah, Terlihat pada code diatas kita deklarasikan 5 parameter. Parameter pertama adalah letak dimana kita akan menggambar yaitu pada img bukan gray karena kita gunakan gray untuk mendeteksi muka, tetapi kita gambar kedalam gambar yang berwarna atau gambar asli kita.

Kemudian parameter kedua yaitu tuple (x, y) yang merupakan titik kiri atas dari kotak persegi kita. Parameter ketiga yaitu tuple (x+w, y+h) untuk mendefinisikan titik kanan bawah persegi kita sehingga akan terbentuklah persegi dengan 2 titik tersebut.

Parameter keempat adalah tuple untuk warna. Warna yang dianut OpenCV adalah BGR, yaitu Blue Green Red, kebalikan dari RGB. Valuenya masih sama, warna terkuat adalah value 255, nah karena kami ingin warna hijau maka saya gunakan (0, 255, 0) Kemudian parameter kelima dan terakhir adalah tebal garis kita, disini saya gunakan angka 2 saja.

```

16     roi_warna = frame[y:y+h, x:x+w]
17     roi_gray = frame[y:y + h, x:x+w]

```

Roi didefinisikan dengan bentuk mirip list pada Python. Didalamnya terdapat 2 parameter yang berupa range. Disini urutan untuk menandai wilayahnya adalah Y, X bukannya X, Y seperti standart pada umumnya. Variabel x, y, w, h yang tertulis diatas adalah milik muka yang sudah diambil menggunakan for loop.

Disini juga kita harus mendefinisikan Roi untuk versi berwarna dan juga versi hitam putih. Kita definisikan dalam 2 variabel berbeda yaitu roi\_warna untuk versi yang berwarna dan juga roi\_gray untuk versi hitam putihnya. Urusan isi parameternya sama saja.

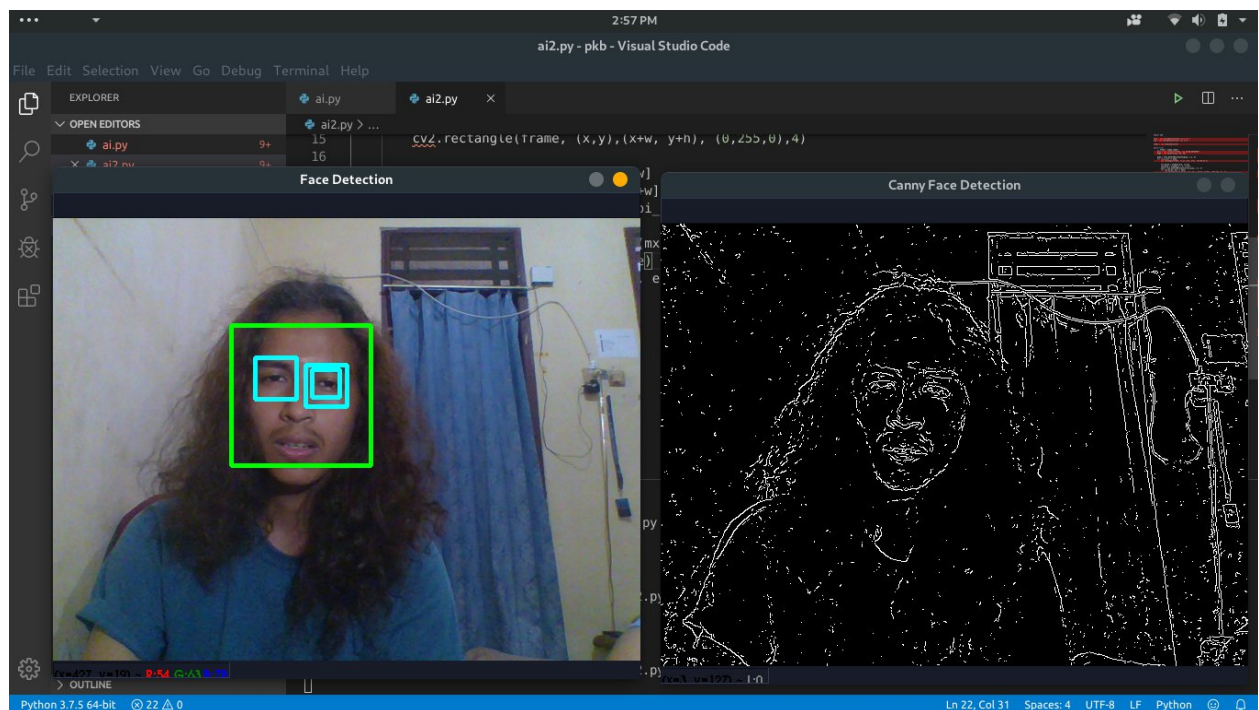
```
18 mata = eye.detectMultiScale(roi_gray, 3, 3)
```

Selanjutnya terapkan juga Eye Detector pada script kita, Masih sama prinsipnya kita terapkan algoritma kita kedalam versi hitam putihnya. Bedanya disini kita tidak perlu deskripsikan lagi nilai keakuratannya.

```
19     for (mx,my,mw, mh) in mata:  
20         cv2.rectangle(roi_warna, (mx, my), (mx+mw, my+mh), (255,255, 0), 3)
```

Kemudian kita masukkan lagi code untuk for loop. Code kali ini untuk menggambar kotak pada mata. Usahakan agar variabelnya berbeda dengan for loop yang pertama agar tidak terjadi error.

## Output program



## DAFTAR PUSTAKA

<http://eprints.umm.ac.id/27682/2/jiptummpp-gdl-ihyaulumud-34975-2-babi.pdf>

[https://elib.unikom.ac.id/files/disk1/661/jbptunikompp-gdl-febryyansy-33013-9-unikom\\_f-n.pdf](https://elib.unikom.ac.id/files/disk1/661/jbptunikompp-gdl-febryyansy-33013-9-unikom_f-n.pdf)

<https://id.wikipedia.org/wiki/OpenCV>