

Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsi Layanan Teknologi Informasi

---

## DIGITAL SIGNAL PROCESSING

Laporan ini merupakan bagian dari Assignment EL5226 – Jaringan Informasi Sosial ITB2015. Source code dan laporan juga tersedia di link github : [https://github.com/soedomoto/ITB2015/tree/EL5226/Assignment 2 - DSP](https://github.com/soedomoto/ITB2015/tree/EL5226/Assignment%202%20-%20DSP)

### 1. AUDIO FINGERPRINTING AND RECOGNITION

Merekam suara dan mencocokkan dengan potongan suara yang tersedia untuk identifikasi

Audio fingerprinting merupakan suatu teknik identifikasi suara dengan cara memecahnya menjadi beberapa sampel. Sebuah file audio dikonversi menjadi grafik sinus, kemudian sampel diperoleh dengan cara pencarian titik puncaknya. Algoritma pencarian titik puncak ini dikenal dengan FFT ([https://en.wikipedia.org/wiki/Fast\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Fast_Fourier_transform)). Misalnya sebuah file .mp3 stereo (2 channel) dengan durasi 3 menit akan memiliki sampel :

$$3 \text{ min} * 60 \text{ sec} * 44100 \text{ samples per sec} * 2 \text{ channels} = 15,876,000 \text{ samples}$$

Kemudian sebanyak 15,876,000 samples tersebut akan disimpan ke dalam sebuah database. Begitu seterusnya untuk setiap file audio yang akan diidentifikasi. Kemudian ketika terdapat sebuah potongan audio, misalnya 10 detik audio .mp3 ingin diidentifikasi siapakah penyanyi dari potongan audio tersebut, maka potongan audio tersebut akan di-sampling, kemudian setiap sampling akan dicocokkan pada database fingerprint. Metode seperti ini seperti yang diterapkan pada beberapa aplikasi pencarian lagu populer seperti Midomi (<http://www.midomi.com/>).

Salah satu tools atau implementasi audio fingerprinting and recognition adalah Dejavu (<https://github.com/worldveil/dejavu>). Dejavu merupakan aplikasi berbasis python yang bekerja dengan algoritma FFT. Dejavu bekerja dengan memanfaatkan database MySQL, akan tetapi untuk meningkatkan portabilitas, sebuah kelas python yang mengimplementasikan interface `dejavu.database.Database` dibuat agar mendukung database SQLite3 ([dejavu\\_sqlite.py](#)). Berikut penjelasan singkatnya :

Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsional Layanan Teknologi Informasi

---

### System Dependencies :

Aplikasi dejavu merupakan aplikasi yang ditulis dengan bahasa Python, sehingga sebenarnya dejavu bersifat platform independent. Tapi dalam kasus ini, dejavu akan dijalankan pada platform berbasis Linux Ubuntu 14.04 Trusty Tahr. Dejavu memerlukan beberapa dependency terhadap aplikasi tertentu :

1. ffmpeg, untuk konversi audio ke .wav format,
2. portaudio, untuk membaca file audio, termasuk membaca audio lewat microphone,
3. audacity, untuk modifikasi file audio (join, splice, resample)

### Python Dependencies :

1. numpy, untuk membaca signal audio,
2. pydub, python wrapper ffmpeg,
3. matplotlib, untuk menggambarkan spectrogram dan plotting,
4. scipy, untuk pencarian titik puncak dengan algoritma FFT,
5. pyaudio, python wrapper untuk portaudio

### Penjelasan Program :

Download [Dejavu](#), atau modifikasinya dengan menghilangkan dependency terhadap MySQL disini [Modifikasi Dejavu](#). Dejavu yang telah dimodifikasi memiliki dependency terhadap [dejavu\\_sqlite.py](#). Struktur database yang digunakan dalam dejavu-sqlite adalah :

```
CREATE TABLE songs (  
    song_id integer primary key autoincrement,  
    song_name varchar(255) not null,  
    fingerprinted tinyint default 0,  
    file_sha1 binary(10) not null  
);
```

```
CREATE TABLE fingerprints (  
    hash binary(10) not null,  
    song_id unsigned mediumint not null,  
    offset unsigned int not null,  
    UNIQUE(song_id, offset, hash)  
);
```

### Import library yang diperlukan :

```
from dejavu import Dejavu  
from dejavu.recognize import FileRecognizer, MicrophoneRecognizer  
import dejavu_sqlite
```

Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsi Layanan Teknologi Informasi

---

Buat konfigurasi database dengan menggunakan sqlite :

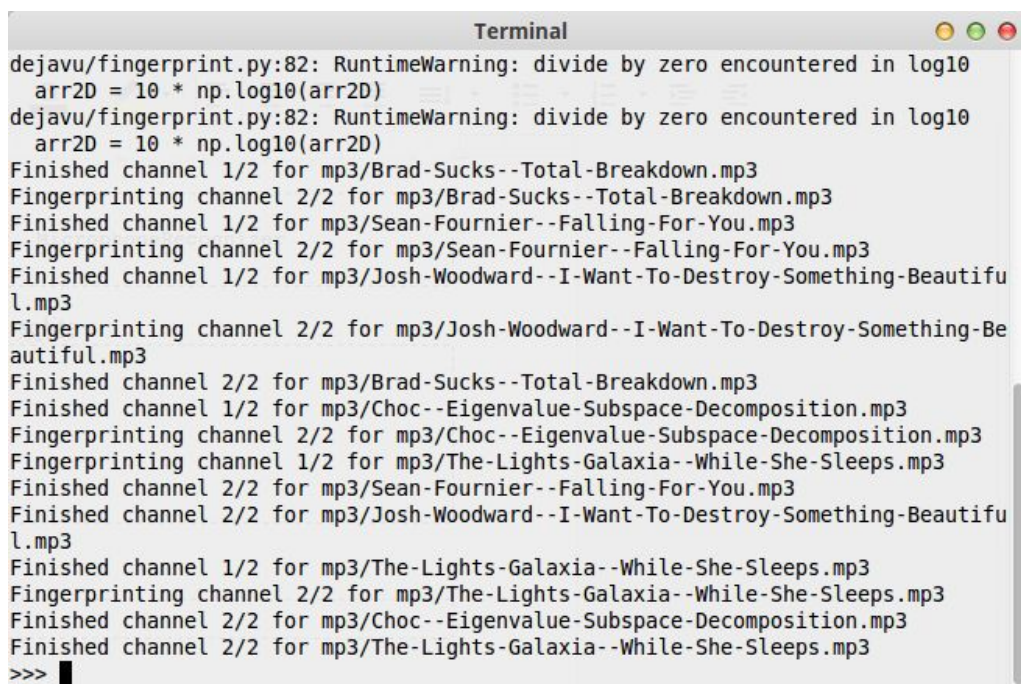
```
config = {  
    "database_type": "sqlite",  
    "database": {  
        "db": "dejavu-test.db"  
    }  
}
```

Instansiasi kelas Dejavu :

```
djv = Dejavu(config)
```

Fingerprint semua file .mp3 dalam sebuah directory mp3 (File audio tersedia [disini](#)) :

```
djv.fingerprint_directory("mp3", [".mp3"])
```



```
Terminal  
dejavu/fingerprint.py:82: RuntimeWarning: divide by zero encountered in log10  
arr2D = 10 * np.log10(arr2D)  
dejavu/fingerprint.py:82: RuntimeWarning: divide by zero encountered in log10  
arr2D = 10 * np.log10(arr2D)  
Finished channel 1/2 for mp3/Brad-Sucks--Total-Breakdown.mp3  
Fingerprinting channel 2/2 for mp3/Brad-Sucks--Total-Breakdown.mp3  
Finished channel 1/2 for mp3/Sean-Fournier--Falling-For-You.mp3  
Fingerprinting channel 2/2 for mp3/Sean-Fournier--Falling-For-You.mp3  
Finished channel 1/2 for mp3/Josh-Woodward--I-Want-To-Destroy-Something-Beautiful.mp3  
Fingerprinting channel 2/2 for mp3/Josh-Woodward--I-Want-To-Destroy-Something-Beautiful.mp3  
Finished channel 2/2 for mp3/Brad-Sucks--Total-Breakdown.mp3  
Finished channel 1/2 for mp3/Choc--Eigenvalue-Subspace-Decomposition.mp3  
Fingerprinting channel 2/2 for mp3/Choc--Eigenvalue-Subspace-Decomposition.mp3  
Fingerprinting channel 1/2 for mp3/The-Lights-Galaxia--While-She-Sleeps.mp3  
Finished channel 2/2 for mp3/Sean-Fournier--Falling-For-You.mp3  
Finished channel 2/2 for mp3/Josh-Woodward--I-Want-To-Destroy-Something-Beautiful.mp3  
Finished channel 1/2 for mp3/The-Lights-Galaxia--While-She-Sleeps.mp3  
Fingerprinting channel 2/2 for mp3/The-Lights-Galaxia--While-She-Sleeps.mp3  
Finished channel 2/2 for mp3/Choc--Eigenvalue-Subspace-Decomposition.mp3  
Finished channel 2/2 for mp3/The-Lights-Galaxia--While-She-Sleeps.mp3  
>>>
```

Deteksi potongan lagu (minimal 5 detik potongan lagu), dan kenali penyanyi-nya (File potongan lagu untuk testing tersedia [disini](#)) :

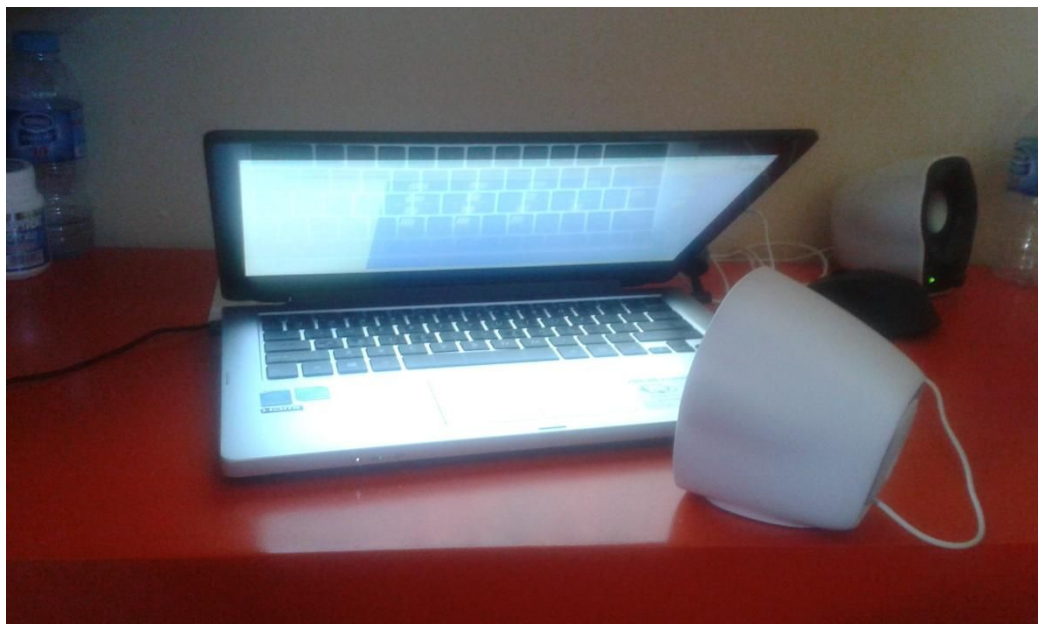
```
song = djv.recognize(FileRecognizer, "test/test1.mp3")  
print "From file we recognized: %s\n" % song
```

Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsional Layanan Teknologi Informasi

```
Terminal
>>> config = {
...     "database_type": "sqlite",
...     "database": {
...         "db": "/home/soedomoto/Documents/dejavu-test.db"
...     }
... }
>>> djv = Dejavu(config)
database opened successfully
>>> song = djv.recognize(FileRecognizer, "test/test1.mp3")
>>> print("From file we recognized: %s\n" % song)
From file we recognized: {'song_id': 2, 'song_name': 'Sean-Fournier--Falling-For-You', 'file_sha1': 'a9d18b9b9daa467350d1b6b249c36759282b962e', 'confidence': 917, 'offset_seconds': 63.20472, 'match_time': 1.6062519550323486, 'offset': 1361}
>>>
>>> warnings
>>> tjson
>>> tjson.filterwarnings("ignore")
>>>
>>> dejavu import Dejavu
>>> dejavu recognize import FileRecognizer, MicrophoneRecognizer
>>>
>>> 
```

Deteksi potongan lagu melalui microphone, dan kenali penyanyi-nya :

```
secs = 5
song = djv.recognize(MicrophoneRecognizer, seconds=secs)
if song is None:
    print "Nothing recognized -- did you play the song out loud so your mic could hear it? :)"
else:
    print "From mic with %d seconds we recognized: %s\n" % (secs, song)
```





Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsional Layanan Teknologi Informasi

```
Terminal
>>>
>>>
>>> secs = 5
>>> song = djv.recognize(MicrophoneRecognizer, seconds=secs)
ALSA lib pcm_dsnoop.c:618:(snd_pcm_dsnoop_open) unable to open slave
ALSA lib pcm_dmix.c:1022:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
bt_audio_service_open: connect() failed: Connection refused (111)
bt_audio_service_open: connect() failed: Connection refused (111)
bt_audio_service_open: connect() failed: Connection refused (111)
bt_audio_service_open: connect() failed: Connection refused (111)
ALSA lib pcm_dmix.c:1022:(snd_pcm_dmix_open) unable to open slave
>>> if song is None:
...     print "Nothing recognized -- did you play the song out loud so your mic
could hear it? :)"
... else:
...     ("From mic with %d seconds we recognized: %s\n" % (secs, song))
...
"From mic with 5 seconds we recognized: {'song_id': 5, 'song_name': 'The-Lights-
Galaxia--While-She-Sleeps', 'file_shal': '355223ebf3ee43eed0f45eca688faad186fc31
67', 'confidence': 3, 'offset_seconds': 32.18286, 'offset': 693}\n"
>>>
```

## 2. FACE MOVEMENT RECOGNITION

Merekam video muka dan mengamati pergerakannya (kanan, kiri)

Face recognition adalah suatu proses mengidentifikasi satu atau lebih wajah dalam sebuah gambar atau video dengan menggunakan sebuah algoritma. Tools yang biasa digunakan untuk mengenali wajah adalah Matlab dan OpenCV. Dalam kasus ini akan digunakan OpenCV dan Python yang dijalankan pada platform berbasis Linux Ubuntu 14.04 Trusty Tahr.

### OpenCV

OpenCV (Open Source Computer Vision) adalah sebuah library yang pertama kali dikembangkan oleh Intel untuk pengolahan citra secara real time. OpenCV bersifat platform independent, dan dapat dijalankan di berbagai platform populer seperti Windows, Linux, FreeBSD, MacOS, dan Android. OpenCV hanyalah sebuah library, dan memerlukan programming interface agar dapat berjalan. Programming interface yang didukung oleh OpenCV adalah C, C++, Java, dan Python.

Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsional Layanan Teknologi Informasi

---

### OpenCV Installation

Secara umum, instalasi OpenCV di ubuntu cukup mudah, karena ubuntu telah menyediakan repository opencv ([detail](#)). Command berikut akan otomatis menginstall opencv beserta seluruh dependency-nya :

```
sudo apt-get -qq install libopencv-dev build-essential checkinstall cmake  
pkg-config yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev  
libswscale-dev libdc1394-22-dev libxine-dev libgstreamer0.10-dev  
libgstreamer-plugins-base0.10-dev libv4l-dev python-dev python-numpy  
libtbb-dev libqt4-dev libgtk2.0-dev libfaac-dev libmp3lame-dev  
libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev  
libxvidcore-dev x264 v4l-utils ffmpeg cmake qt5-default checkinstall
```

Akan tetapi, repository ubuntu tidak menyediakan versi terbaru dari OpenCV. Versi terakhir yang disupport oleh ubuntu adalah OpenCV versi 2.4. Sementara itu untuk menginstall OpenCV versi terbaru, OpenCV 3.1, diperlukan untuk mem-build dari source. Berikut cara instalasi-nya :

#### Install dependency :

```
sudo apt-get install build-essential  
sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev  
libavformat-dev libswscale-dev  
sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev  
libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev
```

#### Install OpenCV :

```
git clone https://github.com/Itseez/opencv.git  
cd opencv  
  
mkdir build  
cd build  
  
cmake -D CMAKE_BUILD_TYPE=Release -D CMAKE_INSTALL_PREFIX=/usr/local ..  
make -j7 # runs 7 jobs in parallel  
sudo make install
```

### Python Application

Python merupakan salah satu Programming Language yang didukung oleh OpenCV. Untuk melakukan face recognition dengan OpenCV dalam bahasa python, diperlukan dependency yang dapat diperoleh pada OpenCV example :

1. common.py, dapat diunduh [disini](#),
2. video.py, dapat diunduh [disini](#)

Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsional Layanan Teknologi Informasi

---

### Penjelasan Program

Pada kasus ini, akan digunakan webcam dan file .mp4 sebagai media memperoleh video. Video yang ditangkap dari webcam atau dibaca dari file .mp4 kemudian akan dibaca frame by frame, sehingga yang sesungguhnya dideteksi adalah image. Berikut penjelasannya programnya (Full code dapat diunduh di[sini](#)) :

Import library yang diperlukan :

```
import cv2

# local modules
from video import create_capture
from common import clock, draw_str
```

Buat fungsi-fungsi yang diperlukan :

```
# Detect cascade (pattern) from image
def detect(img, cascade):
    rects = cascade.detectMultiScale(img, scaleFactor=1.3, minNeighbors=4,
                                     minSize=(30, 30),
                                     flags=cv2.CASCADE_SCALE_IMAGE)

    if len(rects) == 0:
        return []
    rects[:,2:] += rects[:,0:2]
    return rects

# Draw rectangle from detected object
def draw_rects(img, rects, color):
    for x1, y1, x2, y2 in rects:
        cv2.rectangle(img, (x1, y1), (x2, y2), color, 2)
```

Definisikan video yang ingin dideteksi pada argument, jika tidak ada argument yang diberikan, maka gunakan webcam sebagai sumber :

```
# Try to get 1st argument as a source for video-file
args, video_src = getopt.getopt(sys.argv[1:], '', ['cascade=',
'nested-cascade='])
try:
    video_src = video_src[0]
except:
    # If no video-file provide, then use webcam as video source
    video_src = 0
args = dict(args)
```

Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsional Layanan Teknologi Informasi

---

Definisikan algoritma yang ingin dipakai (algoritma ditulis pada sebuah file XML) :

```
# Default : use haarcascade algorithm to detect face and eyes
face_fn = args.get('--cascade',
"data/haarcascades/haarcascade_frontalface_alt.xml")
left_eye_fn = args.get('--nested-cascade',
"data/haarcascades/haarcascade_lefteye_2splits.xml")
right_eye_fn = args.get('--nested-cascade',
"data/haarcascades/haarcascade_righteye_2splits.xml")

# Define the classifier for each algorithm
face = cv2.CascadeClassifier(face_fn)
left_eye = cv2.CascadeClassifier(left_eye_fn)
right_eye = cv2.CascadeClassifier(right_eye_fn)
```

Capture video dari sumber yang telah didefinisikan (video-file atau webcam) :

```
# Capture video from source (either webcam or video-file)
cam = create_capture(video_src)
```

Code berikut harus diletakkan dalam iterasi yang tidak pernah berhenti (kecuali diinterupsi) dengan menggunakan while :

```
while True:
    # Next code put after this comment
```

Baca video sumber per frame yang menghasilkan image, dan konversi menjadi grayscale :

```
# Capture each frame as image
ret, vis = cam.read()
# Convert image to gray
gray = cv2.cvtColor(vis, cv2.COLOR_BGR2GRAY)
gray = cv2.equalizeHist(gray)
```

Deteksi setiap wajah yang terdapat dalam image, kemudian gambarkan boundary wajah sebagai sebuah kotak berwarna hijau. Tuliskan juga label “Face” pada kotak yang terbentuk :

```
# Detect face from image
rects = detect(gray, face)
# Draw face rectangle in image with green border
draw_rects(vis, rects, (0, 255, 0))
# Read face rectangle coordinates
for face_x1, face_y1, face_x2, face_y2 in rects:
    # Draw face label
    draw_str(vis, (face_x1, face_y1), 'Face')
```



Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsi Layanan Teknologi Informasi

---

Crop image sesuai kotak boundary wajah, sehingga hanya akan diperoleh image wajah :

```
# Get face image
face_gray = gray[face_y1:face_y2, face_x1:face_x2]
vis_face = vis[face_y1:face_y2, face_x1:face_x2]
```

Pada image wajah, deteksi mata kiri, kemudian gambarkan boundary mata kiri sebagai sebuah kotak berwarna biru. Tuliskan juga label “Left eye” pada kotak yang terbentuk. Selain itu, hitung titik tengah pada boundary mata kiri (center of left-eye) :

```
center_of_leye = None
# detect left eye coordinates in face image
left_eye_rect = detect(face_gray, left_eye)
# Draw left eye rectangle in face boundary with blue border
draw_rects(vis_face, left_eye_rect, (255, 0, 0))
# Read left eye rectangle coordinates
for leye_x1, leye_y1, leye_x2, leye_y2 in left_eye_rect:
    # Draw left eye label
    draw_str(vis_face, (leye_x1, leye_y1), 'Left eye')
    # Calculate the center of left eye
    center_of_leye = leye_x1 + ((leye_x2 - leye_x1) / 2)
```

Masih pada image wajah, deteksi mata kanan, kemudian gambarkan boundary mata kanan sebagai sebuah kotak berwarna merah. Tuliskan juga label “Right eye” pada kotak yang terbentuk. Selain itu, hitung juga titik tengah pada boundary mata kanan (center of right-eye) :

```
center_of_reye = None
# detect right eye coordinates in face image
right_eye_rect = detect(face_gray, right_eye)
# Draw right eye rectangle in face boundary with red border
draw_rects(vis_face, right_eye_rect, (0, 0, 255))
# Read right eye rectangle coordinates
for reye_x1, reye_y1, reye_x2, reye_y2 in right_eye_rect:
    # Draw right eye label
    draw_str(vis_face, (reye_x1, reye_y1), 'Right eye')
    # Calculate the center of right eye
    center_of_reye = reye_x1 + ((reye_x2 - reye_x1) / 2)
```

Berdasarkan jarak boundary kiri terhadap center-of-left-eye (left-eye-border), bandingkan dengan jarak boundary kanan terhadap center-of-right-eye (right-eye-border). Jika left-eye-border lebih kecil dari right-eye-border maka dapat disimpulkan wajah sudah menghadap ke kiri, dan sebaliknya. Tuliskan label “Look left” dan “Look right” berdasarkan kesimpulan yang didapat :

Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsi Layanan Teknologi Informasi

```
# Label of status will be placed in the bottom of face rectangle
label_pos = (face_x1, face_y2)
# If both left eye and right eye are not detected
if center_of_leye is None and center_of_reye is None:
    draw_str(vis, label_pos, 'Eyes not detected')
# If only left eye is not detected -> Look left
if center_of_leye is None:
    draw_str(vis, label_pos, 'Look left')
# If only right eye is not detected -> Look right
elif center_of_reye is None:
    draw_str(vis, label_pos, 'Look right')
# If both are detected
else:
    if abs((center_of_leye - face_x1) - (face_x2 - center_of_reye)) < 100:
        draw_str(vis, label_pos, 'Look forward')
    # if right eye is more center than left
    elif (center_of_leye - face_x1) > (face_x2 - center_of_reye):
        draw_str(vis, label_pos, 'Look left')
    # If left eye is more center than right
    else:
        draw_str(vis, label_pos, 'Look right')
```

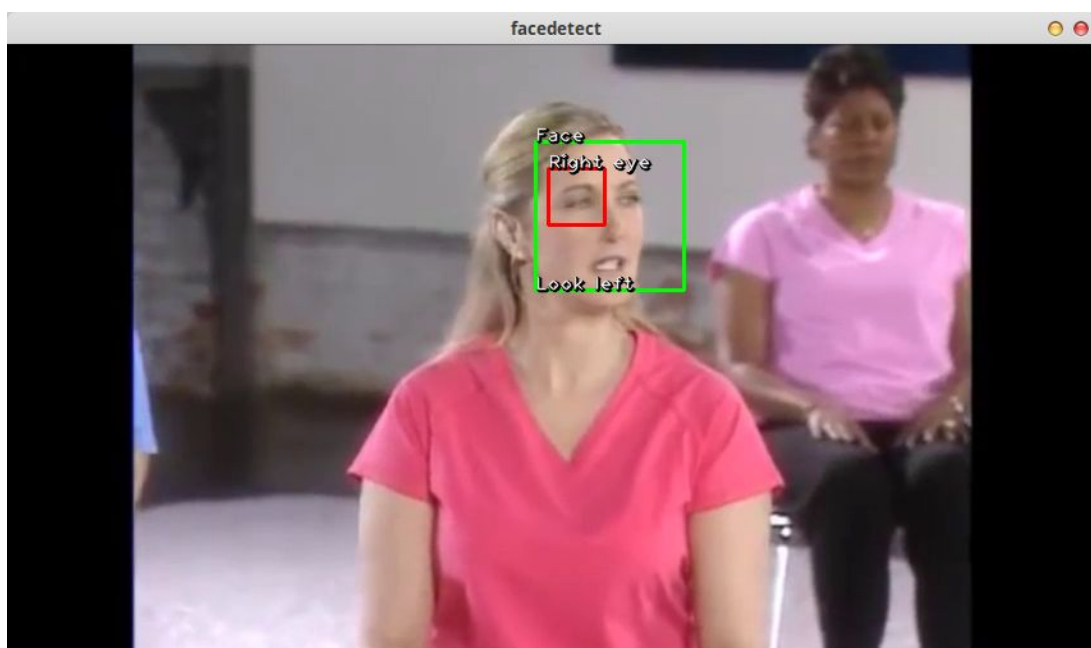
Terakhir, tampilkan setiap image (frame) :

```
# Show image
cv2.imshow('facedetect', vis)
```

### Uji Coba

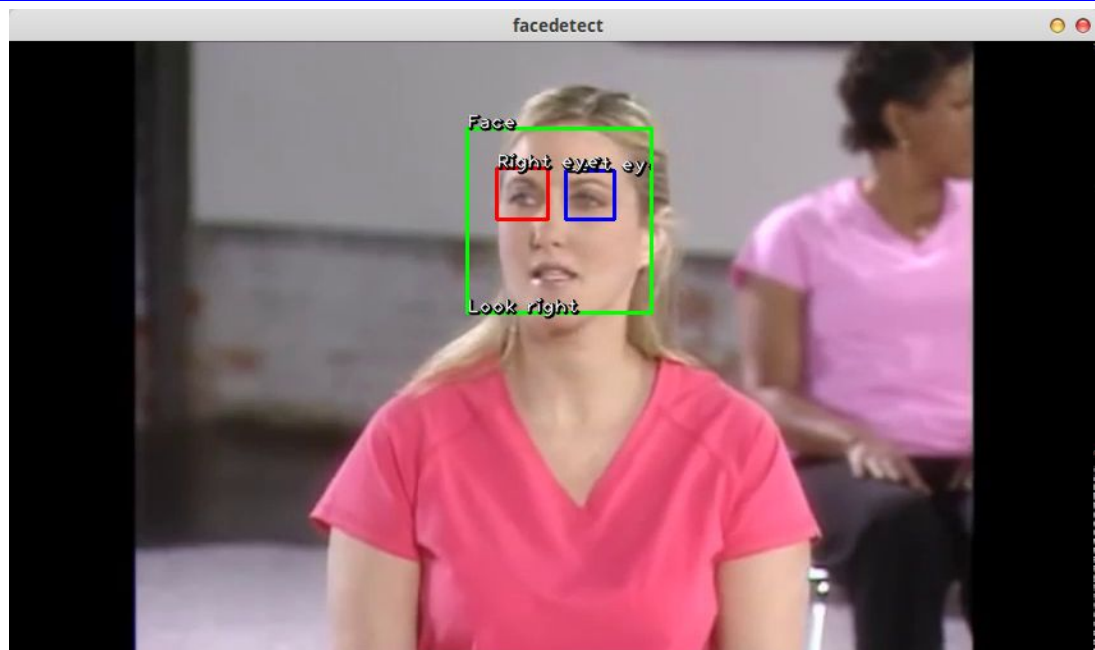
Menggunakan file video .mp4. File untuk ujicoba dapat diunduh [disini](#) :

```
python face_movement_detection.py data/face_video.mp4
```



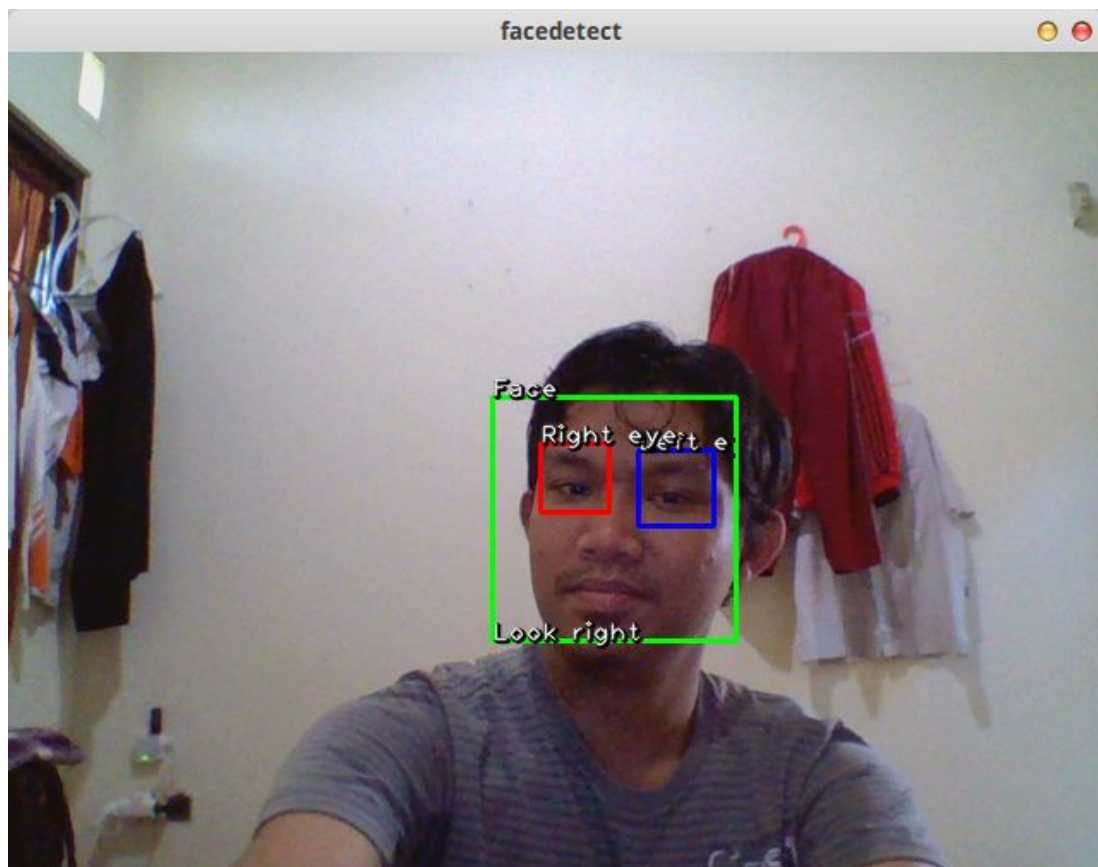
Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsional Layanan Teknologi Informasi

---



Menggunakan webcam :

```
python face_movement_detection.py
```



Nama : Aris Prawisudatama  
NIP : 23215131  
Jurusan : Teknik Elektro  
Opsi Layanan Teknologi Informasi

---

