



**PRAKTIKUM**  
***PENGOLAHAN SINYAL INFORMASI***

**MODUL 6**  
**FACE DETECTION DAN FACE RECOGNITION**

**1.1 Tujuan Praktikum**

1. Memahami konsep dasar deteksi wajah (face detection) dan pengenalan wajah (face recognition).
2. Mempelajari teknik-teknik yang digunakan dalam deteksi dan pengenalan wajah.
3. Mampu mengimplementasikan deteksi wajah menggunakan library OpenCV.
4. Menguasai dasar-dasar pengenalan wajah dengan berbagai pendekatan.
5. Mampu memvisualisasikan hasil deteksi dan pengenalan wajah menggunakan Python.

**1.2 Alat dan Bahan**

1. Laptop/PC
2. Visual Code
3. Library:
  - a) OpenCV (cv2)
  - b) dlib
  - c) face\_recognition
  - d) numpy
  - e) matplotlib



# PRAKTIKUM

## PENGOLAHAN SINYAL INFORMASI

### 1.3 Pendahuluan

#### 1. Apa itu dlib?

dlib adalah sebuah *library* (perpustakaan kode) open-source di Python (juga tersedia di C++) yang digunakan untuk berbagai tugas *Machine Learning* dan *Computer Vision*.

Beberapa fitur dlib diantaranya:

- a) **Face Detection:** Deteksi wajah dengan metode HOG + SVM atau CNN.
- b) **Facial Landmark Detection:** Menentukan titik-titik penting di wajah seperti mata, hidung, mulut, rahang.
- c) **Face Recognition:** Menghasilkan representasi vektor dari wajah (embedding) dan membandingkannya.

Adapun kelebihan dlib diantaranya:

- a) Akurat untuk deteksi dan pengenalan wajah.
- b) Dokumentasi cukup lengkap.
- c) Mendukung model deep learning.
- d) Gratis dan open source.

Instalasi dlib: `'pip install dlib'`

#### 2. Face Detection

Face detection adalah proses identifikasi lokasi wajah manusia dalam gambar digital. Teknik ini merupakan langkah awal sebelum melakukan pengenalan wajah (face recognition). Face detection bekerja dengan menganalisis pola visual dalam gambar untuk menemukan area yang sesuai dengan karakteristik wajah manusia. Dalam melakukan face detection terdapat beberapa metode yang cukup populer, diantaranya:

- 1) Haar Cascade Classifier (OpenCV)
  - a. Metode ini menggunakan pendekatan statistik yang disebut Haar-like features. Fitur ini merepresentasikan pola terang-gelap yang sering ditemukan di wajah manusia, seperti area sekitar mata yang lebih gelap dibanding pipi.
  - b. Algoritma ini melibatkan pelatihan banyak "weak classifiers" (pohon keputusan kecil) dan menggabungkannya menjadi "strong classifier" menggunakan teknik *AdaBoost*.



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

- c. Setelah dilatih, model ini menggunakan struktur mirip pohon (cascade) untuk menyaring kandidat area wajah secara bertahap — dari kasar ke detail.
  - d. Kelebihan: sangat cepat dan efisien, cocok untuk real-time.
  - e. Kekurangan: Kurang akurat jika pencahayaan buruk atau wajah miring.
- 2) HOG + SVM (dlib)
- a. Histogram of Oriented Gradients (HOG) adalah teknik untuk mengekstrak fitur dari gambar berdasarkan arah tepi (gradient) dalam blok-blok kecil.
  - b. Fitur HOG ini kemudian diklasifikasikan menggunakan Support Vector Machine (SVM) untuk mendeteksi apakah blok tersebut mengandung wajah.
  - c. Kelebihan: Lebih akurat dari Haar, lebih tangguh terhadap rotasi kecil dan perubahan pencahayaan.
  - d. Kekurangan: Tidak secepat Haar untuk real-time.
- 3) CNN-based Face Detector (dlib atau framework deep learning lainnya)
- a. CNN (Convolutional Neural Network) mempelajari fitur kompleks dari data gambar secara otomatis melalui beberapa layer konvolusi.
  - b. dlib memiliki face detector berbasis CNN yang lebih akurat dalam mendeteksi berbagai pose wajah dan kondisi pencahayaan.
  - c. Kelebihan: Akurasi tinggi.
  - d. Kekurangan: Membutuhkan GPU atau CPU yang kuat karena komputasi lebih berat

### **3. Facial Landmark Detection**

Facial Landmark Detection adalah proses menemukan titik-titik penting pada wajah. Titik ini berguna untuk mengenali struktur wajah lebih dalam seperti:

- a. Mata (kanan dan kiri)
- b. Ujung hidung
- c. Bibir atas dan bawah
- d. Ujung rahang, dan lain-lain



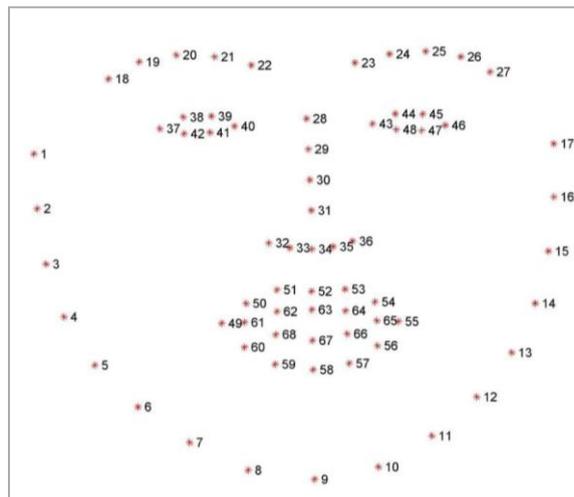
## PRAKTIKUM *PENGOLAHAN SINYAL INFORMASI*

Model yang paling umum digunakan adalah 68-point facial landmark predictor dari dlib, yang memetakan wajah ke dalam 68 koordinat:

- a. Titik 1–17: garis rahang
- b. Titik 18–27: alis
- c. Titik 28–36: hidung
- d. Titik 37–48: mata
- e. Titik 49–68: mulut

Landmark digunakan dalam:

- a. Penyesuaian orientasi wajah (alignment)
- b. Pelacakan ekspresi wajah
- c. Cropping bagian tertentu (misalnya mata saja atau mulut saja)



#### 4. Face Recognition

Face recognition adalah proses identifikasi atau verifikasi identitas seseorang berdasarkan fitur wajahnya. Proses ini lebih kompleks dari face detection karena tidak hanya menemukan wajah tetapi juga mengenali siapa pemilik wajah tersebut.

Langkah-langkah umum:

- a. **Face Detection:** Deteksi lokasi wajah dari gambar.
- b. **Face Alignment (opsional):** Menyesuaikan orientasi agar wajah sejajar (misal: mata sejajar horizontal).
- c. **Face Embedding:** Mengubah wajah menjadi representasi numerik berupa vektor fitur (contoh: 128 dimensi). Vektor ini merepresentasikan karakteristik unik wajah.
- d. **Matching:**



**PRAKTIKUM**  
***PENGOLAHAN SINYAL INFORMASI***

- a) Verification: Apakah wajah A dan wajah B adalah orang yang sama?
  
  - b) Identification: Cari tahu siapa wajah ini dari daftar identitas yang dikenal (database wajah).
- = [[;-.

**a. Model dan Teknik Populer dalam Face Recognition**

- a. dlib face recognition: Model ResNet (deep learning) yang menghasilkan vektor embedding 128 dimensi.
- b. FaceNet: Model dari Google yang mengoptimalkan embedding wajah agar jarak antar wajah yang sama kecil, dan wajah berbeda besar.
- c. DeepFace: Framework berbasis Keras/TensorFlow, mendukung VGG-Face, OpenFace, ArcFace, dsb.
- d. Siamese Network: Jaringan neural khusus yang membandingkan dua input (wajah A dan B) dan memutuskan apakah sama atau tidak.

**5. Penerapan dalam kehidupan sehari-hari**

- a. Absensi otomatis berbasis wajah.
  - b. Sistem keamanan (smart lock, akses area terbatas).
  - c. Social media tagging otomatis.
  - d. Deteksi penipuan di sistem e-KYC (Know Your Customer)
  - e. Penegakan hukum (mengenali wajah pelaku kejahatan dari rekaman CCTV)
1. Contoh penggunaan face detection yaitu menemukan wajah seseorang dalam suatu gambar.

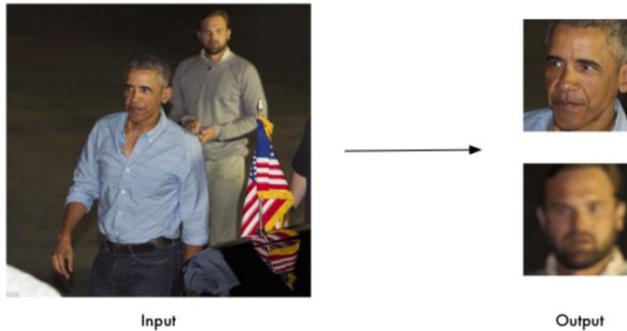


# PRAKTIKUM

## PENGOLAHAN SINYAL INFORMASI

### Find faces in pictures

Find all the faces that appear in a picture:



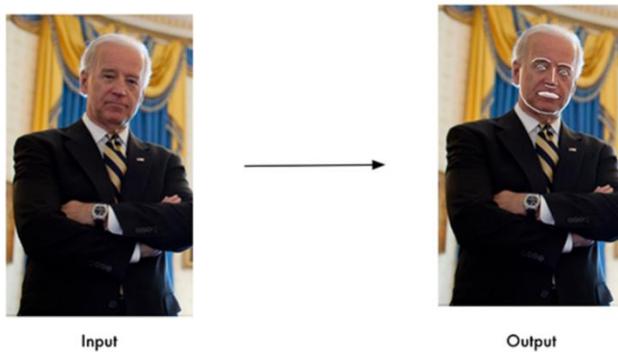
```
import face_recognition
image = face_recognition.load_image_file("your_file.jpg")
face_locations = face_recognition.face_locations(image)
```



## 2. Memanipulasi gambar

### Find and manipulate facial features in pictures

Get the locations and outlines of each person's eyes, nose, mouth and chin.



```
import face_recognition
image = face_recognition.load_image_file("your_file.jpg")
face_landmarks_list = face_recognition.face_landmarks(image)
```



## 3. Mengidentifikasi nama wajah seseorang



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

Identify faces in pictures

Recognize who appears in each photo.



Input



Picture contains  
"Joe Biden"

Output

```
import face_recognition
known_image = face_recognition.load_image_file("biden.jpg")
unknown_image = face_recognition.load_image_file("unknown.jpg")

biden_encoding = face_recognition.face_encodings(known_image)[0]
unknown_encoding = face_recognition.face_encodings(unknown_image)[0]

results = face_recognition.compare_faces([biden_encoding], unknown_encoding)
```



Referensi : [https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)