

# **LAPORAN TUGAS RESUME DAN PRAKTIK**

## **COMPUTER VISION PEETEMUAN 11**

**NAMA : LILIS ANGGRAENI**  
**NIM : 42050230009**  
**KELAS : 5A- TEKNOLOGI INFORMASI**

### **BAB 5**

#### **MENDETEKSI DAN MENGENALI WAJAH**

Computer vision membuat banyak tugas yang terdengar futuristik menjadi kenyataan, termasuk deteksi wajah (menemukan lokasi wajah dalam gambar) dan pengenalan wajah (mengidentifikasi wajah sebagai orang tertentu). OpenCV mengimplementasikan berbagai algoritma untuk kedua fungsi ini, dengan aplikasi mulai dari keamanan hingga hiburan. Bab ini menggunakan Python, OpenCV, dan NumPy, serta modul opsional *opencv\_contrib* untuk pengenalan wajah. Beberapa bagian juga menggunakan OpenNI 2 untuk kamera kedalaman.

##### **1. Mengkonseptualisasikan Haar Cascade**

Gambar, bahkan dari webcam, seringkali memiliki terlalu banyak detail untuk dianalisis secara langsung. Detail ini bisa tidak stabil karena pencahayaan, sudut pandang, jarak, guncangan kamera, dan noise digital. Untuk klasifikasi yang stabil, detail gambar diubah menjadi fitur (features), yaitu representasi yang lebih abstrak dari pola penting dalam gambar. Setiap fitur direpresentasikan sebagai vektor, dan kesamaan antar-gambar dapat diukur berdasarkan jarak antar-vektor fitur.

Fitur mirip Haar (Haar-like features) sering digunakan dalam deteksi wajah real-time. Fitur ini pertama kali diperkenalkan oleh Paul Viola dan Michael Jones dalam makalah Robust Real-Time Face Detection (2001). Setiap fitur menggambarkan pola kontras di antara wilayah gambar yang berdekatan, seperti tepi, simpul, dan garis tipis. Fitur khas ini biasanya muncul pada wajah tetapi jarang pada objek lain. Fitur-fitur ini disusun dalam hierarki (cascade), di mana lapisan atas memiliki fitur yang paling khas sehingga detektor dapat menolak area yang jelas bukan wajah dengan cepat.

Untuk membuat pengklasifikasi Haar cascade tidak peka terhadap skala, ukuran jendela deteksi dijaga konstan, tetapi gambar diubah skalanya beberapa kali (image pyramid). Dengan cara ini, wajah dengan ukuran berbeda dapat terdeteksi

pada berbagai tingkat skala. Namun, Haar cascade tidak tahan terhadap rotasi atau perspektif; wajah terbalik atau profil biasanya tidak dikenali. Implementasi yang lebih kompleks dapat menangani rotasi, tetapi memerlukan sumber daya lebih banyak. OpenCV membatasi implementasinya pada versi sederhana.

## 2. Mendapatkan Data Haar Cascade

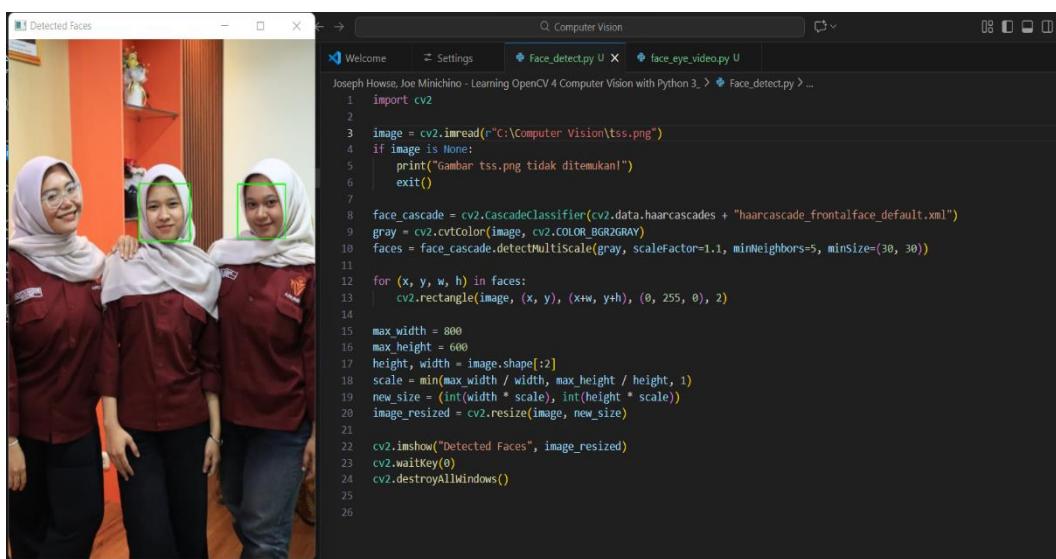
Kode sumber OpenCV 4 atau instalasi build pra-paket OpenCV 4 memiliki sub folder data/haarcascades yang berisi file XML Haar cascade. Jika tidak ditemukan, dapat merujuk ke instruksi instalasi OpenCV. Folder ini berisi file XML yang dimuat dengan cv2.CascadeClassifier. Kelas ini menginterpretasikan file XML sebagai model deteksi objek (misal wajah). File yang biasa digunakan antara lain:

- a. haarcascade\_frontalface\_default.xml → deteksi wajah frontal
- b. haarcascade\_frontalface\_alt.xml → deteksi wajah frontal alternatif
- c. haarcascade\_eye.xml → deteksi mata

File-file ini harus disalin ke folder proyek (misal di subfolder cascades). Untuk membuat cascade sendiri, diperlukan banyak kesabaran dan komputer yang cukup kuat, seperti dijelaskan dalam buku Joseph Howse, OpenCV 4 for Secret Agents. Melakukan

## 3. Deteksi Wajah pada Gambar Diam

Dengan cv2.CascadeClassifier, deteksi wajah dapat dilakukan baik pada gambar diam maupun video. Deteksi wajah pada video hanyalah penerapan deteksi wajah pada setiap frame. Teknik yang lebih canggih memungkinkan pelacakan wajah antar-frame, tetapi pendekatan sekuensial dasar juga efektif.



Kode tersebut berfungsi untuk melakukan deteksi wajah menggunakan OpenCV, dimulai dengan mengimpor library cv2 lalu memuat gambar tss.png dan memastikan gambar berhasil dibaca. Setelah itu, program memanggil model Haar Cascade bawaan OpenCV (haarcascade\_frontalface\_default.xml) sebagai pendekripsi wajah, kemudian mengubah gambar asli menjadi grayscale karena proses deteksi bekerja lebih efektif pada citra hitam putih. Selanjutnya, fungsi detectMultiScale digunakan untuk mendekripsi wajah dengan parameter scaleFactor=1.1, minNeighbors=5, dan ukuran minimum wajah (30,30). Jika wajah ditemukan, program menggambar kotak persegi berwarna hijau di sekitar tiap wajah menggunakan cv2.rectangle. Agar gambar tidak terlalu besar saat ditampilkan, ukuran gambar disesuaikan hingga maksimal 800×600 tanpa mengurangi kualitas apabila sudah lebih kecil. Akhirnya, gambar hasil deteksi ditampilkan melalui cv2.imshow, menunggu input tombol dengan waitKey(0), lalu menutup jendela tampilan menggunakan destroyAllWindows().

#### 4. Melakukan Deteksi Wajah pada Video

Langkah dasar deteksi wajah adalah memuat gambar dan mendekripsi wajah di dalamnya. Untuk visualisasi, wajah yang terdeteksi digambar dengan persegi panjang di gambar asli. Detektor ini bekerja paling baik pada wajah yang tegak dan frontal, seperti gambar deretan orang yang menghadap ke kamera.

```
❶ face_eye_video.py > ...
1  import cv2
2
3 # Load Haar cascades
4 face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_frontalface_default.xml")
5 eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_eye.xml")
6
7 # Buka kamera laptop
8 camera = cv2.VideoCapture(0)
9
10 while True:
11     success, frame = camera.read()
12     if not success:
13         break
14
15     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
16
17     # Deteksi wajah
18     faces = face_cascade.detectMultiScale(gray, 1.3, 5, minSize=(120, 120))
19     for (x, y, w, h) in faces:
20         cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2) # Kotak biru di wajah
21
22         roi_gray = gray[y:y + h, x:x + w]
23         roi_color = frame[y:y + h, x:x + w]
24
25         # Deteksi mata di dalam wajah
26         eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 5, minSize=(40, 40))
27         for (ex, ey, ew, eh) in eyes:
28             cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2) # Kotak hijau di mata
29
30     # Tampilkan hasil
31     cv2.imshow("Face and Eye Detection", frame)
32
33     # Tekan 'q' untuk keluar
34     if cv2.waitKey(1) & 0xFF == ord('q'):
35         break
```



Kode tersebut berfungsi untuk melakukan proses deteksi wajah dan mata secara real-time menggunakan OpenCV. Program diawali dengan mengimpor library cv2, kemudian memuat dua model Haar Cascade bawaan OpenCV, yaitu haarcascade\_frontalface\_default.xml sebagai pendekripsi wajah dan haarcascade\_eye.xml sebagai pendekripsi mata. Kamera laptop diakses melalui fungsi VideoCapture(0) sehingga setiap frame yang tertangkap dapat diproses secara berkelanjutan. Setiap frame terlebih dahulu dikonversi ke citra grayscale untuk meningkatkan efisiensi dan akurasi proses deteksi. Selanjutnya, metode detectMultiScale digunakan untuk mengidentifikasi area wajah pada frame, dan setiap wajah yang terdeteksi diberi anotasi berupa kotak berwarna biru. Pada area wajah tersebut, dilakukan deteksi lanjutan untuk mengenali mata menggunakan model Haar Cascade khusus mata, yang kemudian ditandai dengan kotak berwarna hijau. Hasil deteksi ditampilkan secara real-time melalui cv2.imshow, dan program akan terus berjalan hingga pengguna menekan tombol ‘q’ sebagai perintah untuk menghentikan proses dan menutup seluruh jendela tampilan.

